VIVEKANAND EDUCATION SOCIETY

INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

Hashu Advani Memorial Complex, Collector's Colony, Chembur,
Mumbai, 400074, Maharashtra, India www.vesit.ac.in

Syllabus Approved By:

Academic Council of V.E.S. Institute of Technology

Preamble

Engineering education forms the cornerstone of technological innovation, industrial advancement, and societal transformation. As the world navigates an era of rapid change driven by automation, artificial intelligence, sustainable technologies, and global connectivity, VESIT positions itself as a catalyst for cultivating a dynamic learning ecosystem.

At VESIT, we strive to foster critical thinking, technical expertise, academic excellence, and holistic development among aspiring engineers. We are committed to shaping professionals who are adaptable, collaborative and are also deeply conscious of their social and environmental responsibilities.

Leveraging its autonomous status and affiliation with the University of Mumbai, VESIT designs a forward-thinking, outcome-based curriculum that emphasizes industry relevance and experiential learning.

The syllabus at VESIT has been structured in alignment with the principles of the National Education Policy (NEP) 2020, focusing on flexibility and multidisciplinary learning. The key thrust areas of education at VESIT include:-

Student-Centric Approach:

The programme is designed to offer students greater autonomy in shaping their academic journey. The syllabus comprises of **Core (Major) and Minor courses** from Diverse Disciplines as well as a wide range of **Open Electives**. Students can tailor their learning paths based on their interests and career aspirations. The curriculum also integrates mentored field projects and on the job training, providing valuable hands-on experience. Additionally, students with a research inclination can also explore **research-based projects** or pursue **Honours by Research.**

Multidisciplinary Approach:

VESIT's curriculum reflects a strong interdisciplinary focus, incorporating emerging fields and cutting-edge technologies. Courses are designed to bridge various domains with offerings such as **Machine learning, Artificial Intelligence & Data Science, Cyber Security, Geographic Information Systems (GIS), Internet of Things (IoT), Register transfer level VLSI, Robotics, Quantum Technologies, Mobile application development, Industrial Automation, Edge Computing and Embedded Intelligence and Information Security.**

This approach encourages broader thinking and prepares students for diverse career paths.

Emphasis on Conceptual Clarity:

The curriculum lays stress also on a strong theoretical foundation, ensuring that students gain deep conceptual understanding, which is essential for mastering advanced topics and solving real-world problems.

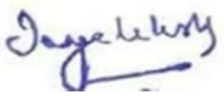
Fostering Creativity and Critical Thinking:

Courses are designed to nurture a critical and creative mindset, promoting analytical reasoning, problem-solving, and innovation. Students are encouraged to question, explore, and think beyond conventional solutions.

Comprehensive Evaluation and Assessment:

Student performance is assessed through a number of assessment tools that includes the Mid-term Tests, Continuous Assessments, End-Semester Examinations. These evaluation tools are designed to measure the knowledge retention of students as well as their ability to apply concepts effectively in practical situations.

Guided by a vision of excellence and inclusivity, and supported by a passionate faculty, VESIT aspires to be a hub where ideas flourish, startups emerge, and industry-academia partnerships thrive. Our goal is to transform students into innovators, entrepreneurs, researchers and responsible leaders poised to drive sustainable growth and meaningful change in society.



Dr. J M Nair

Principal, VESIT

Dr. M Vijayalakshmi

Vice Principal, VESIT

Dr. Mrs. Gresha S Bhatia

Academic Coordinator, VESIT

Preamble

Department of Artificial Intelligence and Data Science

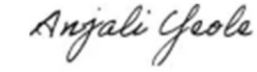
The **Department of Artificial Intelligence and Data Science (AI&DS)** at **Vivekanand Education Society's Institute of Technology (VESIT)** is dedicated to preparing students for the rapidly evolving landscape of intelligent systems and data-driven technologies. In alignment with the **National Education Policy (NEP) 2020**, the department fosters academic excellence, innovation, and holistic development through a dynamic and student-centric curriculum.

The syllabus offers a progressive blend of foundational knowledge and emerging technologies across core areas such as **Data Structures, Probability and Graph Theory, Database Management Systems, Foundations of Data Science, Artificial Intelligence, Analysis of Algorithms, Machine Learning, Data Mining, Deep Learning, Generative AI, Blockchain, Cloud Computing, and Full Stack Development**. Through a combination of theory, practical labs, industry-relevant electives, and value-added courses, students develop strong analytical, programming, and problem-solving skills.

The curriculum integrates **project-based learning, internships, continuous assessments, open electives, and experiential learning opportunities**, thereby promoting interdisciplinary exploration, critical thinking, and real-world readiness. With strong emphasis on **ethics, innovation, entrepreneurship, and sustainability**, the department nurtures future technologists who can responsibly design, build, and deploy intelligent systems for societal advancement.

Through an ecosystem enriched with qualified faculty, industry exposure, hands-on tools, and collaborative research, the AI&DS department aspires to produce graduates who are **technologically proficient, ethically grounded, and globally competitive**, capable of addressing complex challenges in academia, industry, and innovation-driven enterprises.

Dr. M Vijayalakshmi
H.O.D -AIDS



Dr. Anjali Yeole
Dy.H.O.D-AIDS

VIVEKANAND EDUCATION SOCIETY'S
Institute of Technology
(An Autonomous Institute Affiliated to University of Mumbai, Approved by A.I.C.T.E & Recognized by Govt. of Maharashtra)
Department of Artificial Intelligence and Data Science

Department of

Artificial Intelligence and Data science Syllabus (NEP Scheme)

Sem-V

w.e.f. A.Y. 2025-26

Semester V Teaching Scheme

Course Type	Course Code	Course name	Teaching scheme (Contact Hours)			Credits assigned			Total
			Th	Pr	Tu	Th	Pr	Tut	
PCC	NADPC51/ NADPCL51	Machine Learning	3	2	-	3	1	-	4
PCC	NADPC52/ NADPCL52	Data Mining	3	2	-	3	1	-	4
PCC	NADPC53/ NADPCL53	Full Stack Development: Web Dev	3	2	-	3	1	-	4
PE	NADPE5X/ NADPEL5X	PCE-1	3	2	-	3	1	-	4
MDM	NADMM51/ NADMML51	Blockchain - Application Development	3	2	-	3	1	-	4
OE	NOE5XX	Open elective II	3	-	1	3	-	1	4
			18	**10**	**1**	**18**	**5**	**1**	**24**
		Total Hours	**29**			**Total Credits**			**24**

Semester V Examination Scheme

SEM V									
Course Type	Course Code	Course Name	Theory				Term Work	Pract & oral	Total
			Internal Assessment		End Sem Exam	Exam Duration (in Hrs)			
			Mid Test	CA					
PCC	NADPC51/ NADPCL51	Machine Learning	20	20	60	2	25	25	**150**
PCC	NADPC52/ NADPCL52	Data Mining	20	20	60	2	25	25	**150**
PCC	NADPC53/ NADPCL53	Full Stack Development: Web Dev	20	20	60	2	25	25	**150**
PE	NADPE5X/ NADPEL5X	PEC-I	20	20	60	2	25		**125**
MDM	NADMM51/ NADMML51	Blockchain - Application Development	20	20	60	2	25		**125**
OE	NOE5XX	Open elective II	20	20	60	2			**100**
Total			120	120	360		125	75	**800**

Semester V Teaching Scheme

Open Elective- II		
For Departments of AI&DS, CMPN, IT		
Sr.No.	**Course Code**	**Course Name**
1	**NOE506**	**Solid and Hazardous waste management**
2	**NOE507**	**Fundamentals of Sustainability Engineering**
3	**NOE508**	**Energy Audit and Management**
4	**NOE509**	**Electric Vehicles**
5	**NOE510**	**Industrial Automation**
6	**NOE511**	**Fundamentals of Robotics**

Program Elective -I		
Sr.No.	**Course Code**	**Course Name**
1	**NADPE51/NADPEL51**	**User Experience Design**
2	**NADPE52/NADPEL52**	**Statistical and Visualisation techniques**
3	**NADPE53//NADPEL53**	**Computer Vision**

Semester VI Teaching Scheme

Course Type	Course Code	Course name	Teaching scheme (Contact Hours)			Credits assigned			Total
			Th	Pr	Tu	Th	Pr	Tut	
PCC	NADPC61/ NADPCL61	Deep Learning	3	2	-	3	1	-	4
PCC	NADPC62/ NADPCL62	Generative AI	3	2	-	3	1	-	4
PCC	NADP61	Major Project -1	-	4	-	-	2	-	2
PE II	NADPE61X/ NADPEL61X	PEC-II	3	2	-	3	1	-	4
PE III	NADPE62X/ NADPEL62X	PEC-III	3	2	-	3	1	-	4
MD	NADMM61	Secure Software Development	2		-	2		-	2
VC	NADVS61	AWS essential / Azure		2			2		2
			14	**14**		**14**	**8**		**22**
		Total Hours	**28**			**Total Credits**			**22**

Semester VI Examination Scheme

SEM VI									
Course Type	Course Code	Course Name	Theory				Term Work	Prac & oral	Total
			Internal Assessment		End Sem Exam	Exam Duration (in Hrs)			
			Mid Test	CA					
PCC	NADPC61/ NADPCL61	Deep Learning	20	20	60	2	25	25	**150**
PCC	NADPC62/ NADPCL62	Generative AI	20	20	60	2	25	25	**150**
PCC	NADP61	Major Project -1					25	25	**50**
PE II	NADPE61X/ NADPEL61X	PEC-II	20	20	60	2	25		**125**
PE III	NADPE62X/ NADPEL62X	PEC-III	20	20	60	2	25		**125**
MD	NADMM61	Secure Software Development	20	20	60	2			**100**
VC	NADVS61	AWS essential / Azure					25	25	**50**
Total			**100**	**100**	**300**		**150**	**100**	**750**

Program Electives

Program Elective/ Thread	NADPE51	NADPE61	NADPE62
Next Generation Computing (PEC- T I)	User Experience Design	Cloud Computing for AI	Edge Computing
Data Analytics (PEC-T II)	Statistical and Visualisation techniques	Agile Project Management	Game Theory for DS
Applied AI (PEC-T III)	Computer Vision	AI for Healthcare	AI in Finance

Program Elective -2			**Program Elective -3**		
Sr. No.	**Course Code**	**Course Name**	**Sr. No.**	**Course Code**	**Course Name**
1	**NADPE611/NADPEL611**	**Cloud Computing for AI**	1	**NADPE621/NADPEL621**	**Edge Computing**
2	**NADPE612/NADPEL612**	**Agile Project Management**	2	**NADPE622/NADPEL622**	**Game Theory for DS**
3	**NADPE613//NADPEL613**	**AI in Healthcare**	3	**NADPE623//NADPEL623**	**AI in Finance**

COURSE NAME: MACHINE LEARNING

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 51	Machine Learning (Theory)	03	---	---	03	---	---	03
NADPC L51	Machine Learning (Lab)	---	02	---	---	01	---	01

Machine Learning (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 51	Machine Learning (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPC 51	Machine Learning (Theory)	20	20	60	02			100

Prerequisite: Data Structures, Basic Probability and Statistics, Algorithms	
Course Objectives: The course aims:	
1	To understand the fundamental concepts of machine learning, including types, applications, ML pipeline, evaluation metrics, and validation methods.
2	To apply mathematical foundations such as linear algebra, vector operations, matrix decompositions, and SVD in solving ML problems.
3	To implement linear and classification algorithms including Logistic Regression, SVM, KNN, and Regularization techniques for supervised learning.
4	To analyze neural network architectures, learning rules, activation functions, and apply them for solving complex classification problems.
5	To evaluate dimensionality reduction techniques and probabilistic models like PCA, LDA, t-SNE, GMM, and EM for high-dimensional data analysis.

6	To apply MLOps tools and principles for automating ML pipelines, managing data, deploying models, and monitoring them in production.
Course Outcomes: Students will be to	
1	Explain the core concepts of machine learning, including types, applications, ML pipeline, and performance metrics.
2	Apply mathematical concepts such as norms, eigenvalues, SVD, and matrix operations to support machine learning algorithms.
3	Implement and compare linear models, logistic regression, SVM, KNN, and apply regularization techniques in supervised learning tasks.
4	Analyze neural networks, including perceptron, multilayer perceptron, backpropagation, and learning rules for complex problems.
5	Evaluate and apply dimensionality reduction methods like PCA, LDA, t-SNE, and GMM-EM to reduce data complexity and improve analysis.
6	Apply MLOps principles and tools for automating pipelines, managing ML models, version control, and monitoring ML models in production.

Module	Detailed Content	Hours
1	**Introduction to Machine Learning** Overview of machine learning: definitions, types, and applications.Machine learning pipeline: data preprocessing, feature engineering, model training, and evaluation. Training,Test and Validation set, cross validation, k- fold cross validation, Overfitting and Underfitting of model, bias-variance trade off, Performance Metrics: Classification (Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC-AUC), Regression (MSE, MAE, RMSE, R2 Score).	6
2	**Mathematical Foundation for ML** System of Linear equations, Norms, Inner products, Length of Vector, Distance between vectors, Orthogonal vectors. Symmetric Positive Definite Matrices, Determinant, Trace, Eigenvalues and vectors, Orthogonal Projections, Diagonalization, SVD and its applications.	6
3	**Linear Models** Finding best fit line, Gradient Descent Algorithm: Learning Algorithm, First order derivatives, Linear regression using gradient descent, Learning rate, Logistic Regression, Need of Regularization, L1 and L2 Regularization and its implementation. Comparison of L1 and l2 Regularization.Support Vector Machines. Hard and Soft margin in SVM, Hinge Loss function of SVM, Kernel SVM and its type. K-nearest neighbour	7
4	**Neural Networks** Introduction, Fundamental concept, Evolution of Neural Networks, Biological Neuron, Artificial Neural Networks, NN architecture, McCulloch-Pitts Model. Designing a simple network, Non-separable patterns, Perceptron model with Bias, Activation functions: Binary, Bipolar, continuous, Ramp, Limitations of Perceptron. Perceptron Learning Rule. Delta Learning Rule (LMS-Widrow Hoff),	8

	Multi-layer perceptron network Adaline and Madaline network Adjusting weights of hidden layers. Error back propagation algorithm. Hebbian Learning rule.	
5	**Dimensionality Reduction** Curse of Dimensionality,Feature Selection and Feature Extraction Dimensionality Reduction Techniques: Principal Component Analysis, t-distributed Stochastic Neighbor Embedding (t-SNE),Linear Discriminant Analysis,Gaussian Mixture Models (GMM) and Expectation-Maximization (EM) algorithm.	7
6	**MLOps** Introduction to MLOps,Overview of MLOps and its significance,Key challenges in deploying and managing ML models in production, Comparison of traditional software development, DevOps and MLOps, Key components of MLOps, MLOps workflow, Landscape of MLOps tools and technologies.ML Pipelines & Data Management: Overview of data engineering tools and practices, Data management for ML models, ML pipeline automation, Use of version control systems like Git for model development,	5
	TOTAL	**39**

Textbooks:

1	**Pattern Recognition and Machine Learning** – Christopher M. Bishop, Springer, 1st Ed. (2006), ISBN: 978-0387310732.
2	**Machine Learning** – Tom M. Mitchell, McGraw-Hill, 1st Ed. (1997), ISBN: 978-0070428072.
3	**Understanding Machine Learning: From Theory to Algorithms** – Shai Shalev-Shwartz & Shai Ben-David, Cambridge University Press, 1st Ed. (2014), ISBN: 978-1107057135.
4	**Practical MLOps:** Operationalizing Machine Learning Models" by Noah Gift and Alfredo Deza
5	**Building Machine Learning Powered Applications**: Going from Idea to Product - Emmanuel Ameisen - O'Reilly publication

References:

1	**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow** – Aurélien Géron, O'Reilly Media, 3rd Ed. (2022), ISBN: 978-1098125974.
2	**The Elements of Statistical Learning** – Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2nd Ed. (2009), ISBN: 978-0387848570.
3	**Deep Learning** – Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, 1st Ed. (2016), ISBN: 978-0262035613.
4	**Mathematics for Machine Learning** – Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, Cambridge University Press, 1st Ed. (2020), ISBN: 978-1108455145.
5	**Machine Learning Yearning** – Andrew Ng, Self-Published, 1st Ed. (2018) *(Available for free online)*.

Web Links:	
1	NPTEL
2	AI and ML Certification - Enroll in PGP AI ML Courses with Purdue (simplilearn.com)
3	https://www.learndatasci.com/out/coursera-machine-learning/
4	https://www.learndatasci.com/out/google-machine-learning-crash-course/
5	***Introduction to Machine Learning***: *Coursera - Machine Learning by Andrew Ng*
6	***Mathematical Foundations for ML***: *MIT OpenCourseWare - Linear Algebra*
7	***Gradient Descent & Linear Models***: *Stanford CS229 - Machine Learning*
8	***Ensemble Learning***: *Towards Data Science - Ensemble Methods*
9	***Neural Networks & Deep Learning***: *Deep Learning Specialization by Andrew Ng*
10	***Dimensionality Reduction***: *Scikit-learn - PCA & Feature Selection*

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Machine Learning (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPCL51	Machine Learning (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPCL51	Machine Learning (Lab)	-	-	-	-	25	25	50

Prerequisite: Database Management Concepts	
Lab Objectives:	
1	To understand the complete ML pipeline, model performance using cross-validation, bias-variance trade-off, and evaluate models using appropriate metrics.
2	To apply various machine learning algorithms including regression, classification, neural networks, and clustering on real-world datasets.
3	To analyze dimensionality reduction technique PCA and t-SNE and LDA.
4	To demonstrate the setup of an MLOps environment by building an automated ML pipeline and using version control tools for model lifecycle management.
Lab Outcomes: Students will be able to	
1	Explain the complete ML pipeline, model performance using cross-validation techniques, bias-variance analysis, and appropriate performance metrics.
2	Implement ML algorithms like Linear Regression, Logistic Regression, SVM, KNN, neural networks, and dimensionality reduction techniques.
3	Analyze and interpret dimensionality reduction techniques PCA,t-SNE,LDA.
4	Demonstrate the creation of an automated ML pipeline integrated with MLOps tools, Git-based version control, and model lifecycle management.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	Introduction to Platform like anaconda and Colab and study Machine learning library and tools.
2	Understand the complete ML pipeline from data preprocessing to model evaluation using a classification / Regression task.
3	Model Evaluation and Cross-Validation with Bias-Variance Analysis.
4	Implement Linear Regression algorithm with L1 and L2 regularization techniques.
5	Implement Logistic regression algorithm.
6	Implement Gradient Descent algorithm for house price prediction data set.
7	Implement Support Vector Machines (SVM) algorithm.
9	Implement logic gates using MP neuron.
10	Implement Multi-layer perceptron network and show the weight changes in backpropagation for sample data set.
11	Implement Hebbian learning algorithm.
12	Implement Dimensionality Reduction using PCA and t-SNE.
13	Implement Expectation Maximization algorithm.
14	Implement Linear Discriminant Analysis.
15	To introduce the foundational concepts of MLOps by setting up a machine learning development environment, implementing version control using Git, and building an automated end-to-end ML pipeline for data processing, model training, and evaluation.

Note: Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 8 -10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)
Evaluation Exam	
1	Practical Exam based on the entire syllabus

COURSE NAME: DATA MINING

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 52	Data Mining (Theory)	03	---	---	03	---	---	03
NADPC L52	Data Mining (Lab)	---	02	---	---	01	---	01

Data Mining (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 52	Data Mining (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPC 52	Data Mining (Theory)	20	20	60	02			100

Prerequisite: Database Management concept	
Course Objectives: The course aims:	
1	To **understand** the architecture, schemas, and evolution of data warehouses and distinguish between OLAP and OLTP systems.
2	To **analyze** various data preprocessing techniques and apply data transformation methods on real-world datasets.
3	To apply classification techniques, including Decision Trees, Naïve Bayes, and ensemble methods such as Random Forest and XGBoost, for solving supervised learning problems.
4	To **examine** different clustering algorithms and distance metrics for unsupervised learning and outlier detection.
5	To **understand** the process of mining frequent patterns and generating association rules using Apriori and pattern-growth techniques.

6	To **evaluate** the efficiency of frequent itemset mining algorithms like PCY and variants in handling large datasets.
Course Outcomes: Students will be be	
1	**Differentiate** between OLAP models and data warehouse schemas like star, snowflake, and fact constellation.
2	**Apply** data cleaning, integration, transformation, and discretization techniques effectively.
3	Implement classification algorithms, including Decision Trees, Naïve Bayes, and ensemble methods such as Random Forest and XGBoost, on labeled datasets and analyze their performance.
4	**Compare** clustering algorithms and **select** appropriate distance measures for various datasets.
5	**Construct** association rules using minimum support, confidence, and lift metrics.
6	**Analyze** the performance of PCY and constraint-based association mining techniques.

Module	Detailed Content	Hours
1	**Data Warehouse and OLAP** Data Warehousing, The Need for Data Warehousing; Data Warehouse Definition Benefits and business Value of Data Warehouse; Features of a Data Warehouse Dimensional Modeling and OLAP, Dimensional Model Vs ER Model; The Star Schema, The Snowflake Schema; Fact Tables and Dimension Tables; Fact less Fact Table, Aggregate Tables; Fact Constellation Schema or Families of Star,Handling Slowly Changing Dimensions (SCD Types 1, 2, and 3. Need for Online Analytical Processing; OLTP vs OLAP; OLAP Operations in a cube: Roll-up, Drilldown, Slice, Dice, Pivot ; OLAP Models: MOLAP,ROLAP, HOLAP. Evolution from Data Warehouses to Modern Architectures Data Lakes vs Data Warehouses, Emerging Concepts : Data Lake (e.g., Delta Lake, Apache Iceberg) , Data Mesh, Data Fabric, Data Islands and Data Swamps	7
2	**Introduction to Data Mining ,Data Exploration and Data Preprocessing** Types of Attributes; Statistical Description of Data; Data Visualization; Measuring similarity and dissimilarity. Data Preprocessing an overview, Data Cleaning: MissingValues, NoisyData Data Cleaning as a Process. Data Integration; Data Reduction: Attribute subset selection, Histograms, Clustering and Sampling; Data Transformation & Data Discretization: Normalization, Binning, Histogram Analysis and Concept hierarchy generation.	5
3	**Classification and Ensemble Learning** Decision Tree Induction: Attribute Selection Measures, Entropy, Information Gain, Gini index and CART, Tree pruning. Bayesian Classification: Naïve Bayes' Classifier,Rule based classifier. Introduction to Ensemble Learning Types of ensembles: Bagging, Boosting, Stacking. Bootstrap Aggregation, Bagging Random Forest: Architecture and working principle. Weak learners and iterative boosting AdaBoost: Working principle and intuition, Introduction to XGBoost.	7
4	**Clustering**	7

	Cluster Analysis and prediction: Basic Concepts; Distance measure:Euclidean, Jaccard,Cosine and hamming distance.Partitioning Methods: K-Means,Elbow method, K Medoids; Hierarchical Methods: Agglomerative, Divisive, Canopy Clustering, DBSCAN Clustering, Outlier Analysis : Types,Challenges and Methods.	
5	**Frequent Pattern Mining** Market Basket Analysis, min support, min confidence, lift. Frequent Item sets, Closed Item sets, and Association Rules; Frequent Pattern Mining, Efficient and Scalable Frequent Itemset Mining Methods, The Apriori Algorithm for finding Frequent Itemsets Using Candidate Generation, Generating Association Rules from Frequent Item sets, Improving the Efficiency of Apriori, A pattern growth approach for mining frequent itemset, Mining Frequent Itemsets Using Vertical Data Format	7
6	**Advance Frequent pattern Mining** Using main memory to count frequent item sets, cost of counting pairs and representation in main memory (triangular matrix, triples), Algorithm for handling large dataset: Algorithm of Park Chen Yu (PCY) & Variants: Multistage PCY and Multihash PCY **Introduction to Mining Multilevel association:** Rules and multidimensional association rule from association mining to correlation analysis lift; Introduction to constraint based association mining**.**	6
	TOTAL	**39**

Textbooks:	
1	Han, Kamber, 'Data Mining Concepts and Techniques', Morgan Kaufmann 3nd Edition
2	P. N. Tan, M. Steinbach, Vipin Kumar, ―Introduction to Data Mining, Pearson Education.
3	Paulraj Ponniah, ―Data Warehousing: Fundamentals for IT Professionals, Wiley India.
4	Raghu Ramakrishnan and Johannes Gehrke, ―Database Management Systems 3rd Edition -McGraw Hill
5	Elmasri and Navathe, ―Fundamentals of Database Systems, 6th Edition, PEARSON Education

References:	
1	Theraja Reema, ―Data Warehousing, Oxford University Press, 2009
2	Ralph Kimball, Margy Ross, ―The Data Warehouse Toolkit: The Definitive Guide To Dimensional Modeling, 3rd Edition. Wiley India.
3	Michael Berry and Gordon Linoff ―Mastering Data Mining- Art & science of CRM, Wiley Student Edition

4	Michael Berry and Gordon Linoff —Data Mining Techniques, 2nd Edition Wiley Publications

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Data Mining(Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPCL52	Data Mining (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPCL52	Data Mining (Lab)	-	-	-	-	25	25	50

Prerequisite: Database Management Concepts	
Lab Objectives:	
1	To design and implement a data warehouse and perform OLAP operations, along with data cleaning, integration, transformation, and visualization.
2	To build, train, and analyze ensemble models like Random Forest and XGBoost, and compare them with single classifiers like Decision Tree and Naïve Bayes.
3	To implement and evaluate unsupervised learning techniques, including K-Means, Hierarchical Clustering, and DBSCAN, and interpret clustering quality and effectiveness.
4	To perform market basket analysis and advanced pattern mining using Apriori, FP-Growth, Multistage, and Multihash PCY on large datasets and interpret the results.
Lab Outcomes:	
1	Apply data warehousing techniques to implement star schema and perform OLAP operations using SQL Server/DataGrip and Power BI.
2	Construct and analyze ensemble models (Random Forest, XGBoost) and compare their performance with base classifiers (Decision Tree, Naïve Bayes) on real-world data using Python/WEKA.
3	Evaluate and visualize clustering results obtained using K-Means, DBSCAN, and Hierarchical Clustering, and interpret cluster quality using elbow plots and dendrograms.
4	Apply association rule mining techniques like Apriori, FP-Growth, and PCY algorithms to extract frequent patterns and generate rules from transactional data.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	Design and Implement a Star Schema using SQL Server/DataGrip: Create fact and dimension tables;
2	Perform OLAP Operations: Roll-up, Drill-down, Slice, and Dice
	Setting data brick community lab (free) using publicly available datasets.
3	Data Cleaning, Integration, Transformation, and Visualization of Real-World Dataset
4	Implement Decision Tree Classifier (ID3, CART)
5	Implement Spam Detection using Naïve Bayes
6	Feature Selection with Random Forest for High-Dimensional Data
7	Implement XGBoost on structured data and evaluate performance.
8	Apply K-Means and use the Elbow plot for optimal K.
9	Perform agglomerative hierarchical clustering on a real-world dataset and visualize the resulting cluster hierarchy using a dendrogram.
10	Apply the DBSCAN algorithm for clustering data points and identifying noise/outliers , demonstrate its effectiveness in unsupervised learning tasks
11	Perform Market Basket Analysis using the Apriori algorithm, generate frequent itemsets, and derive meaningful association rules to uncover patterns in transactional data.
12	Implement the FP-Growth algorithm for efficient pattern mining, extracting frequent itemsets and discovering valuable patterns in large datasets
13	Implement Multistage and Multihash PCY on Large Dataset

Note: Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 8 -10 experiments.

2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)
Evaluation Exam	
1	Practical Exam based on the entire syllabus

COURSE NAME: FULL STACK DEVELOPMENT: WEB DEV

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 53	Full Stack Development : Web Dev (Theory)	03	---	---	03	---	---	03
NADPC L53	Full Stack Development: Web Dev (Lab)	---	02	---	---	01	---	01

Full Stack Development: Web Dev (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 53	Full Stack Development: Web Dev (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPC 53	Full Stack Development: Web Dev (Theory)	20	20	60	02			100

Prerequisite:Basics of web development	
Course Objectives: The course aims to make the learner	
1	Understand core web technologies and JavaScript fundamentals.
2	Master React.js for building dynamic UIs, including state management, hooks, routing, and performance optimization.
3	Develop backend applications using Node.js and Express.js.
4	Work with databases and integrate them with full-stack applications.
5	Design secure RESTful APIs with authentication and error handling.
6	Deploy full-stack applications using modern platforms.
Course Outcomes: Upon completion the learner will be able to	

1	Explain and apply core web technologies (HTTP, DNS, REST) and JavaScript (ES6+) to manipulate the DOM and handle asynchronous operations.
2	Develop interactive web applications using advanced JavaScript features (ES6+, promises, async/await) and implement client-side form validation and event handling.
3	Build dynamic, component-based UIs with React.js using hooks, state management, routing, and performance optimization techniques.
4	Design server-side applications using Node.js, leveraging its event-driven architecture, file system operations, and core modules (HTTP, streams).
5	Develop secure RESTful APIs with Express.js, implementing middleware, authentication (JWT), and error handling for scalable backend services.
6	Integrate SQL/NoSQL databases (PostgreSQL/MongoDB) with React frontends and Express backends, and deploy full-stack applications to cloud platforms.

Module	Detailed Content	Hours
1	**Web programming fundamentals** Working of web browser, HTTP, HTTPS, DNS, TLS, XML introduction, Json introduction, DOM, URL, URI, REST API	5
2	**Javascript** Introduction to JavaScript: JavaScript language constructs, Objects in JavaScript- Built in, Browser objects and DOM objects, event handling, form validation and cookies. Introduction to ES5, ES6, Difference between ES5 and ES6. Variables, Condition, Loops, Functions, Events, Arrow functions, Setting CSS Styles using JavaScript, DOM manipulation, Classes and Inheritance. Iterators and Generators, Promise, async/await, Fetch	6
3	**React Fundamentals** Introduction to React (components, JSX, props, state), Event handling, conditional rendering, Lists and keys, forms, Basic hooks (useState, useEffect). Advanced hooks (useContext, useReducer, custom hooks), Routing with React Router, State management (Redux Toolkit or Context API, Styling (CSS Modules, Styled Components), Performance optimization (memoization, lazy loading).	10
4	**Node.js** Node.js runtime, Asynchronous programming, Callback concept, event loop, Event emitter, Networking module, Buffers, Streams, File system, path, REPL, NPM packages, package.json, Building a basic HTTP server (without Express).	6
5	**Express** Introduction to Express (middleware, routing), RESTful API design (CRUD operations), Error handling, logging, Authentication (JWT, bcrypt).	6

6	**Databases & Full-Stack Integration** SQL (PostgreSQL + Sequelize), NoSQL (MongoDB + Mongoose), Connecting React frontend to Express backend, CORS, environment variables, Deployment basics (Vercel + Render).	6
	TOTAL	39

Textbooks:	
1	Rediscovering JavaScript, Master ES6, ES7, and ES8, By Venkat Subramaniam · 2018
2	Learning React Functional Web Development with React and Redux, Alex Banks and Eve Porcello, O'Reilly
3	Learning Redux, Daniel Bugl, Packt Publication
4	Learning Node.js Development, Andrew Mead, Packt Publishing
5	RESTful Web API Design with Node.js 10, Valentin Bojinov, Packt Publication

References:	
1	Web Development with Node and Express, Ethan Brown, O'Reilly
2	HTML5 Cookbook, By Christopher Schmitt, Kyle Simpson, O'Reilly Media
3	Web Technologies: Black Book, Kogent Learning Solutions, Dreamtech Press
4	Full Stack Web Development with React and Node.js, Shama Hoque, BPB Publications

Useful Links :	
1	https://www.coursera.org/learn/html-css-javascript-for-web-developers?action=enroll
2	ttps://onlinecourses.swayam2.ac.in/ugc19_lb05/preview
3	https://reactjs.org/tutorial/tutorial.html
4	https://nptel.ac.in/courses/106105183

Sr. No.	**Rubrics**	**Marks**
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks

4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Full Stack Development: Web Dev (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPCL53	Full Stack Development: Web Dev (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPCL53	Full Stack Development: Web Dev(Lab)	-	-	-	-	25	25	50

Prerequisite: A MOOC on basics of web development using HTML and CSS.	
Lab Objectives:	
1	Apply core web technologies to build responsive, interactive front-end applications.
2	Develop proficiency in JavaScript for DOM manipulation, API consumption, and form validation.
3	Master React.js fundamentals by creating dynamic UIs with state management and hooks.
4	Design backend systems using Node.js and Express.js for RESTful API development.
5	Integrate databases with full-stack applications and implement authentication.
6	Deploy containerized applications to cloud platforms using modern DevOps tools.
Lab Outcomes:	
1	Construct responsive web interfaces using HTML5, CSS3, and JavaScript.
2	Build React applications with stateful components, API integration, and routing.
3	Create secure backend APIs with Node.js/Express, including CRUD operations and error handling.
4	Implement database solutions for user authentication and data persistence.
5	Connect frontend and backend systems while resolving CORS and environment variables.
6	Deploy full-stack applications using Docker and cloud platforms.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	Responsive Portfolio Website: Build a responsive personal portfolio using HTML5, CSS3 (Flexbox/Grid), and JavaScript DOM manipulation (e.g., dark mode toggle).
2	REST API Data Fetcher: Create a webpage that fetches and displays data from a public API (e.g., JSONPlaceholder) using fetch() and async/await.
3	Interactive Form Validator: Develop a form with real-time validation (email, password) using JavaScript regex and event listeners.
4	CLI Weather App: Build a Node.js CLI tool to fetch weather data from OpenWeather API and display formatted results.
5	Todo List with React: Implement a CRUD todo app using React useState with filtering (active/completed tasks).
6	Movie Search App: Create a React app that searches and displays movies from OMDB API using useEffect and debounced input.
7	E-Commerce Cart: Develop a cart system with global state management (Redux/Context API) and localStorage persistence.
8	Auth-Protected Routes: Set up JWT-based login/logout with React Router, including private/public routes (e.g., /dashboard).
9	File Organizer CLI: Build a Node.js tool to sort files by extension using fs/path modules and log operations.
10	RESTful Blog API: Create a CRUD API for blog posts using Node.js + Express, tested with Postman.
11	User Auth System: Extend the Blog API with PostgreSQL/MongoDB, adding password hashing (bcrypt) and JWT validation.
12	Full-Stack Deployment: Dockerize a React + Express app and deploy it on Vercel (frontend) and Render (backend).

Note: **Suggested List of Experiments is indicative**. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of a minimum 10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)
Evaluation Exam	
1	Practical Exam based on the entire syllabus

COURSE NAME: USER EXPERIENCE DESIGN

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 51	User Experience Design (Theory)	03	---	---	03	---	---	03
NADPE L51	User Experience Design (Lab)	---	02	---	---	01	---	01

Full Stack Development: Web Dev (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 51	User Experience Design (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 51	User Experience Design (Theory)	20	20	60	02			100

Course Objectives:	
1	To understand foundational UX principles, usability evolution, and process models for user-centered design.
2	To apply contextual inquiry, design thinking, and prototyping techniques to develop interaction designs.
3	To evaluate interactive systems using rapid UX assessment methods.
4	To execute rigorous empirical evaluations using usability testing and metrics analysis.
5	To create extended reality (XR) experiences using prototyping and iterative refinement.
6	To understand unique user experience challenges in Metaverse environments.

Course Outcomes: students will be able to	
1	Understand UX concepts, processes, and business value for interactive systems.
2	Apply user-centered prototypes through contextual analysis and design methodologies.
3	Evaluate using heuristic evaluations and rapid usability assessments to improve designs.
4	Execute formal usability studies, analyze results, and refine interfaces.
5	Create XR prototypes using appropriate tools while addressing ethical considerations.
6	Understand immersive Metaverse interfaces that overcome spatial, social and accessibility challenges.

Module	Detailed Content	Hours
1	**Introduction** Introduction to UX, Ubiquitous interaction, Emerging desire for usability, From usability to user experience, Emotional impact as part of the user experience, User experience needs a business case, Roots of usability, A UX process lifecycle template, Choosing a process instance for your project, The system complexity space, Meet the user interface team, Scope of UX presence within the team.	6
2	**Contextual Inquiry and Contextual Analysis** The system concept statement, User work activity gathering, Look for emotional aspects of work practice, Data-driven vs. model driven inquiry, Contextual Analysis, Extracting Interaction Design Requirements, Constructing Design Information Models, Information ,Architecture and Interaction Design and Prototyping Introduction, Design paradigms, Design thinking, Design perspectives, User personas, Ideation, Sketching, Mental Models and Conceptual Design, Wireframe, Prototyping	6
3	**UX Evaluation:Rapid Evaluation Method** UX Evaluation and Improve UX Goals, Metrics and Targets, UX Evaluation Techniques.- Formative vs summative ,types of formative and informal summative evaluation methods, types of evaluation data, some data collection techniques, Rapid Evaluation Methods: Design walkthroughs and reviews UX Inspection, Heuristic evaluation, a UX inspection method, practical approach to UX Inspection, Do UX Evaluation rite, Quasi-empirical UX evaluation Questionnaires.	7
4	**UX Evaluation: Rigorous Empirical Evaluation** Plan for rigorous empirical UX evaluation, Team roles for rigorous evaluation , Prepare an effective range of tasks, Select and adapt evaluation method and data collection techniques, Select participants, Recruit participants, Prepare for participants, Rigorous Empirical Evaluation: Running the Session,Rigorous Empirical Evaluation: Analysis, Evaluation Reporting	6
5	**XR Design Process**	8

	XR Design Process: Components involved in creating XR experiences, Design Thinking, Ethics, and Guidelines for XR: Building empathy with users, Ethical and responsible design in XR, Design guidelines and best practices, Storyboarding & Physical Prototyping for XR Experiences: Sketching and storyboarding for XR, Physical prototyping using materials like paper, cardboard, and Play-Doh, Enhancing physical prototypes with AR techniques, Digital Prototyping & Immersive Authoring: Utilizing digital tools for XR prototyping, Implementing immersive authoring techniques, Testing and refining XR prototypes on actual devices	
6	**Designing the Metaverse: User Experience** The Importance of User Experience (UX) and Interface Design in the Metaverse: Enhancing Immersion and Presence, Seamless Navigation and Orientation, Personalization and Customization, Social Interaction and Collaboration, Accessibility and Inclusivity, Iterative Design and User Feedback, **Challenges of Designing Intuitive and Immersive Experiences:** Hardware Limitations and Fragmentation, User Interface and Interaction Design, Spatial Design and User Orientation, Comfort and Ergonomics, Content Creation and Scalability, Ethical Considerations.	6
	TOTAL	**39**

Textbooks:	
1	The UX Book Process and Guidelines for Ensuring a Quality User Experience by Rex Hartson,
2	Smashing UX Design by Jesmond Allen and James Chudley, John Wiley & Sons
3	UX for XR Cornel Hillmann
4	The Metaverse: And How It Will Revolutionize Everything Matthew L. Ball

References:	
1	The Elements of User Experience by Jesse James Garrett
2	Observing the User Experience: A Practitioner's Guide to User Research by Mike Kuniavsky
3	https://www.coursera.org/learn/user-experience-interaction-design-augmented-virtual-mixed-ext en ded-reality
4	https://www.cambridgescholars.com/resources/pdfs/978-1-5275-6577-7-sample.pdf

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

User Experience Design (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL51	User Experience Design (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPEL51	User Experience Design (Lab)	-	-	-	-	25		25

Lab Objectives:	
1	To understand foundational knowledge of user-centered design principles, including requirements gathering, user research, and problem analysis for virtual and augmented reality projects.
2	To create wireframes, prototypes, and comprehensive design systems using tools like Figma, with an emphasis on responsive and accessible interfaces.
3	To implement UI/UX interactions in VR/AR environments, including gesture, gaze, and voice controls.
4	To evaluate VR/AR and metaverse interfaces, focusing on usability, navigation, and user engagement strategies.
Lab Outcomes: Students will be able	
1	To understand comprehensive user research, create user personas and stories, and define clear problem statements and project scope for VR/AR applications.
2	To create low- and high-fidelity prototypes and establish cohesive UI design systems that ensure responsiveness and accessibility in VR/AR environments.
3	To implement intuitive VR/AR interfaces incorporating multimodal input controls such as gestures, gaze, and voice commands, ensuring seamless user interaction.
4	To evaluate VR/AR and Metaverse interfaces, providing insights into navigation efficiency, user engagement, and overall user experience improvements.

Suggested Experiments Students are required to ceast 8-10 experiments.	
Sr. No.	**Name of the Experiment**
1	Project Proposal and Requirement Gathering (Choose the project). Briefly state the problem(s) that the project will seek to solve. Take the user's point of view. Consider what the user's goals are, and what obstacles lie in the way.
2	Create User Personas, User Stories, and Use Case Diagrams to define a Problem Statement, Scope, and understand the problem.
3	User research: Document & Conduct user, market, product, and competitive research to find possible solutions
4	Ideation: Create sketches and low-fidelity wireframes of the scoped solution
5	Execution: Create High-Fidelity Mockups & Prototypes from the wireframe
6	Figma basics: Creating basic responsive elements like buttons, input elements, etc. to understand frames, groups, layout, constraints, texts, vector, color palette, etc.
7	Basic Clickable Prototyping using figma.
8	Create a Design System for a selected project using Grid and Spacing, Typography, Color System, and UI elements like icons, images, buttons, Inputs, Cards, search bar etc.
9	Create low-fidelity wireframes or storyboards for a VR or AR feature (e.g., gesture-based menu, voice interaction, object manipulation).
10	Design and implement a user interface (UI) system for a VR environment in Unity, ensuring that interactions are intuitive and non-intrusive.
10	Evaluate the use of multiple input modes (gesture, gaze, voice) in VR/AR interfaces.
11	Test and evaluate the usability of a browser-based metaverse platform like Mozilla Hubs or FrameVR.
12	Analyze how navigation aids (signs, maps, minimaps, floor labels) influence user experience in 3D virtual spaces.
13	Understand how avatar customization impacts user engagement and emotional connection in virtual spaces.
14	Design a low-fidelity prototype of an interface or interaction for a metaverse experience using 2D tools like Figma or Miro.
15	Microproject on UX design for Metaverse.

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 10 -12 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: STATISTICAL AND VISUALIZATION TECHNIQUES

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 51	Statistical and Visualization Techniques (Theory)	03	---	---	03	---	---	03
NADPE L51	Statistical and Visualization Techniques (Lab)	---	02	---	---	01	---	01

Statistical and Visualization Techniques (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 51	Statistical and Visualization Techniques (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 51	Statistical and Visualization Techniques (Theory)	20	20	60	02			100

Course Objectives: The course aims:	
1	To understand fundamental statistical concepts for data analysis.
2	To explain core concepts of probability theory and probability distributions, and use them in basic statistical analysis.
3	To apply the principles of hypothesis testing and compare different groups.
4	To implement sampling methods and nonparametric statistical tests for data analysis without strict distributional assumptions.
5	To evaluate regression models and modern statistical measures for high dimension data.

6	To apply data visualization techniques for exploring and presenting data insights effectively.
Course Outcomes: Students will be to	
1	Understand data using statistical measures and visualize relationships using plots and correlation techniques.
2	Explain probability models and distributions to analyze random variables and infer population characteristics.
3	Apply hypothesis tests, including t-tests and ANOVA, to draw conclusions from data.
4	Implement resampling techniques and nonparametric tests like Kruskal-Wallis and Friedman's test for data analysis.
5	Evaluate linear regression and advanced statistical metrics to interpret model performance.
6	Apply various basic and advanced visualizations plots for data exploration.

Module	Detailed Content	Hours
1	**Introduction to statistics** Elements of Structured Data,Rectangular Data Data Frames and Indexes ,Non rectangular Data Structures, Population vs sample, Estimates of Location:Mean,Median and Robust Estimates , Measures of dispersion: range, quartile deviation, mean deviation, standard deviation, variance, empirical relations between measures of dispersion, absolute and relative dispersion, coefficient of variation, outlier, moments, Pearson's β and γ coefficients, skewness, kurtosis, population parameters and sample statistics. Measures of position: quartiles, interquartile range, semi-interquartile range, percentiles, percentile rank, 10–90 percentile range, box and whisker plot. Correlation: Scatter plot, covariance, Karl Pearson's coefficient of correlation.	6
2	**Random Variable and Probability Distributions** Fundamentals of Probability theory: Probability spaces and Random Variable Discrete and continuous, Joint, marginal, and conditional distributions, Bayes theorem, Markov Network : Motivation and factors , Discrete-time vs. continuous-time,Markov chain,The Central Limit Theorem, Distributions of the sample mean and the sample variance for a normal population. Common probability distributions: Normal Distribution ,Student's t-Distribution ,Binomial Distribution ,Chi-SquareDistribution ,F-Distribution, Poisson Distributions.	8
3	**Hypothesis Testing and ANOVA** A/B Testing,Hypothesis Tests,The Null Hypothesis,Alternative Hypothesis , significance levels, Confidence Interval, One-WayVersusTwo-Way Hypothesis Tests P-values and their interpretation, Type 1 and Type 2 Error, t-Tests, F Statistics,ANOVA - One- Way and Two-Way ANOVA.	7
4	**Sampling and Non Parametric Methods** Motivation for sampling The idea of a sampling distribution,Resampling with replacement,Bias estimation using bootstrap, Resampling Versus Bootstrapping,	6

	Degree of Freedom, Parametric and NonParametric Tests, Comparison of Parametric and nonparametric methods, Assumptions of Nonparametric Statistics, Nonparametric Method:The Kruskal-Wallis , Friedman's Test, Advantages and Disadvantages of Non-parametric Statistical Tests.	
5	**Regression and Modern Statistical Techniques** Simple Linear Regression, Statistical Properties of the Estimated Slope and Intercept,Assessing the Fit,Linear Least Squares, Multiple Linear Regression, properties of multiple linear regression, Modern Statistical Evaluation Techniques BLEU, ROUGE, Likelihood and log-probability usage ,Top-k sampling, Temperature scaling as a form of variance control	6
6	**Visualization** Acquiring and Visualizing Data,Introduction to Matplotlib, Basic Plotting create Bar Chart, Pie Chart, Box Plot, violin plot, Matrix charts and heat maps, Frequency Tables and Histograms ,Density Plots,Scatter plots , visualizing two or more Variables, Hexagonal Binning and Contours,Exploring advancedVisualizations (Candlestick Charts, Bubble Charts, Surface Charts, Map Charts), Visualization with Seaborn. Narrative visualization and digital storytelling, Infographics and interactive dashboards.	6
	TOTAL	**39**

Textbooks:	
1	Bruce, Peter, and Andrew Bruce. Practical statistics for data scientists: 50 essential concepts. Reilly Media, 2017.
2	Statistical Methods, S. P. Gupta, Sultan Chand, 2021, 46th revised edition.
2	An Introduction to Statistics with Python, Thomas Hasalwanter, Springer, 2016
3	Think Stats: Probability and Statistics for Programmers, Allen B. Downey, Green Tea Press, 2011.
4	Ben Fry, 'Visualizing data: Exploring and explaining data with the processing environment', O'Reilly, 2008.

References:	
1	Fundamentals of mathematical statistics, S. C. Gupta, V. K. Kapoor, Sultan Chand, 2020, 12th edition
2	Statistics, Freedman, David, Robert Pisani, Roger Pervis, W. W. Norton, 2007
3	Python for Data Analysis: 3rd Edition, Wes McKinney ,Publisher(s): O'Reilly Media, Inc.

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Statistical and Visualization Techniques (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL51	Statistical and Visualization Techniques (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPEL51	Statistical and Visualization Techniques (Lab)	-	-	-	-	25		25

Lab Objectives:	
1	To understand descriptive statistics for data distributions using summary measures like mean, variance,correlation, covariance,
2	To apply different probability distribution techniques, joint and conditional probabilities on given data
3	To analyze statistical inference techniques including confidence intervals, bootstrapping, ANOVA and hypothesis techniques
4	To implement linear regression and data visualization techniques using Python tools.
Lab Outcomes:	
1	To understand statistical measures like mean, variance, skewness, percentiles and kurtosis, interquartile range on data.
2	To apply different probability distributions techniques and joint/conditional probability distribution
3	To analyze statistical inference using confidence intervals, bootstrap methods, Chi-square tests, ANOVA and hypothesis techniques.
4	To implement linear regression techniques and create interactive visualizations plots.

Suggested Experiments Students are required to complete at least 8-10 experiments using Python/Excel/R programming.	
Sr. No.	**Name of the Experiment**
1	Descriptive Statistics : Calculating summary statistics (mean, median, mode, variance, standard deviation)
2	Measure and analyse correlation and covariance between two variables.
3	Measure skewness, kurtosis, interquartile range, semi-interquartile range, percentiles.
4	Conduct experiments to understand conditional probability and independence.
5	Simulate and analyze discrete distributions like Binomial, and Poisson / continuous distributions such as Uniform, Exponential, and Normal to datasets.
6	Calculate and interpret joint and conditional probabilities from given data
7	To perform hypothesis tests using Z statistics/ t statistics/ F statistics.
8	To calculate confidence intervals for different parameters.
9	To prove the central limit theorem.
10	To study sampling distributions and their parameters
10	Implement bootstrap resampling to estimate sampling variability.
11	To perform a hypothesis test using Chi square.
12	To perform ANOVA on given data.
13	Simple linear regression and Multiple linear regression implementation
14	Creating basic plots (Histogram,Pie Chart, Boxplot, bar, scatter, Density plot,heatmaps, pair plots, violin plots)) also Customize plots (labels, titles, legends)
15	Advanced visualizations (Candlestick Charts, Bubble Charts, Surface Charts, Map Charts)
16	Creating interactive visualizations with Plotly, Dashboards and Storytelling with data.

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 10 -12 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.

3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: COMPUTER VISION

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 51	Computer Vision (Theory)	03	---	---	03	---	---	03
NADPE L51	Computer Vision (Lab)	---	02	---	---	01	---	01

Full Stack Development: Web Dev (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 51	Computer Vision (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 51	Computer Vision (Theory)	20	20	60	02			100

Prerequisite: Probability and Statistics, Programming Python/C/C++/OpenCV	
Course Objectives: The course aims	
1	Understand Fundamental Concepts and Theories of Computer Vision
2	Understand fundamental image processing operations such as filtering, edge detection, and histogram equalization to extract and analyze features from images.
3	Apply techniques to track objects and analyze motion within video sequences.
4	Compare various stereo vision algorithms to establish correspondences between image pairs, facilitating depth estimation and 3D reconstruction.
5	Design systems capable of detecting and localizing objects within images or video streams.
6	Develop systems that can recognize objects across different viewpoints and reconstruct 3D structures.

Course Outcome: Students will be able to	
1	Understand the fundamental concepts and theories of computer vision.
2	Understand basic image processing techniques for feature extraction and analysis.
3	Apply advanced methods such as scale-invariant feature detection and optical flow for object tracking and motion analysis.
4	Evaluate different stereo vision correspondence algorithms.
5	Develop computer vision systems for object detection.
6	Design computer vision systems for object recognition and 3D reconstruction.

Module	Detailed Content	Hours
1	**Introduction to Computer Vision** Overview of Computer vision, History of computer vision and its application, Geometric primitives in image processing, Image transformations, 2D & 3D transformation and rotation, 3D to 2D projection, Image restoration, linear filtering, Homogeneous coordinates, camera projection and its models, Basic image processing techniques filtering, edge detection, frequency domain analysis, Frequency domain analysis, Interest point detection, Blob and ridge detection.	6
2	**Image formation and transformation** Low-level vs High-level processing, Two View Geometry, Binocular Stereopsis: Camera and Epipolar Geometry, Planar Scenes and Homography, Depth estimation and multi-camera views, Robust Correspondence Estimation, Auto-calibration, Structure from Motion, Fourier Transform, Interest Point Detection, Edge Detection, Local Binary Pattern, Convolution and Filtering, Gaussian derivative filters, Gabor Filters, DWT.	7
3	**Feature Extraction and Analysis** Scale-invariant feature transform (SIFT), Applications of SIFT, histogram of oriented gradients and its parameters, Bag of words and histogram of oriented gradients (HOG), Feature representation using HOG, Gaussian and Hough transform, Gaussian and Hough transform, Difference of Gaussian and Hough transform for feature detection, Feature extraction, Corner detection using Harris operator, Random sample consensus (RANSAC), RANSAC for robust estimation.	7
4	**Motion Analysis and Stereo Vision** Optical flow and its application in motion tracking, Pyramids and motion models for multi resolution analysis, Global motion estimation, Kanade-Lucas-Tomasi (KLT) feature tracking, Fundamental matrix, Epipolar geometry for stereo vision, Camera models, Calibration and structure from motion, introduction to Stereo vision, Structure from motion, Stereo vision and reconstruction technique.	6
5	**Object Detection** Object detection: General object detection, Face detection, pedestrian detection, 3D reconstruction: shape from X, shape from shading and photometric stereo, shape from texture, shape from focus, 3D scanning, surface representations:	7

	surface interpolations, surface simplification, geometry images, point based representations, volumetric representations, model based reconstruction: architecture, facial modeling and tracking, Human body modeling and tracking.	
6	**Object Recognition** Recognition: Instance recognition, Image classification: feature based methods, face recognition, Application: Visual similarity search, semantic segmentation, instance segmentation, panoptic segmentation, pose estimation, video understanding and vision and language.	7
	TOTAL	**39**

Textbooks:	
1	"Computer Vision: Algorithms and Applications" by Richard Szeliski
2	"Computer Vision: A Modern Approach" by David A. Forsyth and Jean Ponce
3	"Learning OpenCV 4 Computer Vision with Python 3" by Joseph Howse, Prateek Joshi, and Vinícius G. Mendonça

References:	
1	Introduction to Computer Vision" by Tariq Rashid
2	"Programming Computer Vision with Python" by Jan Erik Solem
3	"Computer Vision: Principles, Algorithms, Applications, Learning" by Richard E. Woods and Steven L. Eddins

Sr. No.	**Rubrics**	**Marks**
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Computer Vision (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL51	Computer Vision (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPEL51	Computer Vision (Lab)	-	-	-	-	25		25

Prerequisite: Linear Algebra, Probability and Statistics, Programming - Python/C/C++/OpenCV	
Lab Objectives:	
1	Understand Fundamental Concepts and Theories of Computer Vision
2	Understand fundamental image processing operations such as filtering, edge detection, and histogram equalization to extract and analyze features from images.
3	Apply techniques to track objects and analyze motion within video sequences.
4	Compare various stereo vision algorithms to establish correspondences between image pairs, facilitating depth estimation and 3D reconstruction.
5	Design systems capable of detecting and localizing objects within images or video streams.
6	Develop systems that can recognize objects across different viewpoints and reconstruct 3D structures.
Lab Outcomes:	
1	Understand the fundamental concepts and theories of computer vision.
2	Implement basic image processing techniques for feature extraction and analysis.
3	Apply advanced methods such as scale-invariant feature detection and optical flow for object tracking and motion analysis.
4	Evaluate different stereo vision correspondence algorithms.
5	Design and develop computer vision systems for object detection.
6	Design and develop computer vision systems for object recognition and 3D reconstruction.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	Image Restoration and Feature Detection on Degraded Images Take a noisy/blurry image, restore it using filtering techniques, and detect interest points, edges, blobs, and ridges.
2	3D Object Detection and Projection Using a Monocular Camera Detect a simple 3D object (e.g., a cube or a bottle), estimate its pose in 3D space, and project it onto a 2D image from a monocular camera.
3	3D Scene Reconstruction from Stereo Images Reconstruct a 3D scene from two calibrated cameras using binocular stereopsis and estimate depth maps.
4	Image Texture & Feature Analysis Toolbox Create a GUI or script that allows image analysis using filtering, feature detection, and texture descriptors.
5	Object Recognition Using SIFT, HOG, and Bag of Words (BoW) Build a pipeline to detect and classify objects (e.g., cars, logos, faces) using feature descriptors like SIFT and HOG with Bag of Words for representation.
6	Robust Feature Matching with Harris Corners, Hough Transform, and RANSAC Detect corners in images, find geometric shapes using Hough Transform, and robustly match features between images using RANSAC.
7	Real-Time Motion Tracking and Optical Flow using KLT Tracker Track the movement of objects or feature points in a video using Optical Flow and KLT, and visualize their trajectories.
8	Stereo Vision and 3D Reconstruction Using Epipolar Geometry and SfM Estimate depth from a stereo pair and reconstruct 3D scene structure using stereo geometry and Structure from Motion.
9	Multi-Object Detection and 3D Human Pose Estimation Detect faces, pedestrians, and general objects in a scene and reconstruct **3D human poses** using model-based approaches.
10	Shape-from-X: Depth & 3D Surface Reconstruction Toolbox Reconstruct 3D geometry of objects using multiple shape-from-X techniques: **shading**, **texture**, **focus**, and **photometric stereo**.
11	Unified Visual Recognition System (Multi-Task Image Analysis) Build a visual analysis tool that performs multiple recognition tasks: classification, segmentation (semantic + instance), face recognition, and pose estimation — all from a single image.
12	Vision + Language: Visual Search Engine with Captioning and Similarity Matching Implement a visual search engine where a user uploads an image and the system finds similar images, classifies the content, and auto-generates a caption.

Note: **Suggested List of Experiments is indicative**. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments,

(limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: BLOCKCHAIN APPLICATION DEVELOPMENT

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADMM51	Blockchain Application Development (Theory)	03	---	---	03	---	---	03
NADMML51	Blockchain Application Development (Lab)	---	02	---	---	01	---	01

Blockchain Application Development (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADMM51	Blockchain Application Development (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADMM51	Blockchain Application Development (Theory)	20	20	60	02			100

Prerequisite: Cryptography and System Security	
Course Objectives: The course aims:	
1	To get acquainted with the concept of Block and Blockchain.
2	To learn the concepts of consensus and mining in Blockchain and acquire the knowledge of Bitcoin network, nodes and their roles.
3	To analyze different types of blockchain platforms.
4	To understand the blockchain platform and its terminologies.

5	To design and develop decentralized applications using Ethereum, and Hyperledger.
6	To analyze the applications & case studies of Blockchain.

Course Outcomes: Student will be able to:

1	Describe the basic concept of Block chain.
2	Analyze the concepts of consensus and mining in blockchain, and apply the principles of keys, wallets, and transactions within the Bitcoin network.
3	Analyze Public and Private Blockchain Platforms.
4	Create Public Blockchain using Ethereum.
5	Apply the concept of private blockchain using Hyperledger.
6	Illustrate the applications of Block chain and analyze case studies.

Module	Detailed Content	Hours
1	**Introduction to Blockchain:** Origin of blockchain (cryptographically secure hash functions), basics of blockchain, Foundation of blockchain: Merkle trees. Components of blockchain, Block in blockchain, Types: Public, Private, and Consortium, Consensus Protocol, Limitations and Challenges of blockchain.	4
2	**Cryptocurrency:** Cryptocurrency: Bitcoin, Altcoin, and Tokens (Utility and Security), Cryptocurrency wallets: Hot and cold wallets, Cryptocurrency usage, Transactions in Blockchain, UTXO and double spending problem. Bitcoin Blockchain, Consensus in Bitcoin, Proof of Work (PoW), Proof of Burn(PoB), Proof of Stake (PoS), Proof of Elapsed Time (PoET), Life of a miner, Mining Difficulty, Mining Pools and its methods.	8
3	**Programming for Blockchain:** Introduction to Smart Contracts, Types of Smart Contracts, Structure of a Smart Contract, Smart Contract Approaches, Limitations of Smart Contracts. Introduction to Programming: Solidity Programming – Basics, functions, Visibility and Activity Qualifiers, Address and Address Payable, Bytes and Enums, Arrays-Fixed and Dynamic Arrays, Special Arrays-Bytes and strings, Struct, Mapping, Inheritance, Error handling. Case Study – Voting Contract App, Preparing for smart contract development.	8
4	**Public Blockchain**: Introduction to Public Blockchain, Ethereum and its Components, Mining in Ethereum, Ethereum Virtual Machine (EVM), Transaction, Accounts, Architecture and Workflow, Comparison between Bitcoin and Ethereum. Types of test-networks used in Ethereum, Transferring Ethers using Metamask, Mist Wallet, Ethereum frameworks, Case study of Ganache for Ethereum blockchain, Exploring etherscan.io and ether block structure.	8
5	**Private Blockchain:** Introduction, Key characteristics, Need of Private Blockchain, Smart Contract in a Private Environment, State Machine Replication, Consensus Algorithms for Private	8

	Blockchain - PAXOS and RAFT, Byzantine Faults: Byzantine Fault Tolerant (BFT) and Practical BFT, Byzantine Faults: Byzantine Fault Tolerant (BFT) and Practical BFT. Introduction to Hyperledger, Tools and Frameworks, Hyperledger Fabric, Comparison between Hyperledger Fabric & Other Technologies.	
6	**Tools and Applications of Blockchain:** Blockchain in Action: Use Cases Financial Services, Insurance, Government, Supply Chain Management, Healthcare, Healthcare payments pre-authorization, The Internet of Things (IoT)	3
	TOTAL	**39**

Textbooks:	
1	Blockchain Technology, Chandramouli Subramanian, Asha A. George, Abhilash K. A and Meena Karthikeyen, Universities Press.
2	Mastering Ethereum, Building Smart Contract and Dapps, Andreas M. Antonopoulos Dr. Gavin Wood, O'reilly.
3	Imran Bashir, Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, 3rd Edition, Packt Publishing.
4	"Mastering Bitcoin, PROGRAMMING THE OPEN BLOCKCHAIN", 2nd Edition by Andreas M. Antonopoulos, June 2017, Publisher(s): O'Reilly Media, Inc. ISBN: 9781491954386.
5	"Blockchain for Enterprise Application Developers", Ambadas, Arshad SarfarzAriff, Sham – Wiley.

References:	
1	Blockchain for Beginners, Yathish R and Tejaswini N, SPD.
2	Blockchain Basics, A non Technical Introduction in 25 Steps, Daniel Drescher, Apress.
3	Blockchain with Hyperledger Fabric,Luc Desrosiers, Nitin Gaur, Salman A. Baset, Venkatraman Ramakrishna, Packt Publishing.

Online References:	
1	https://andersbrownworth.com/blockchain/
2	https://andersbrownworth.com/blockchain/public-private-keys/
3	https://www.coursera.org/learn/cryptocurrency
4	https://coinmarketcap.com/

5	NPTEL: https://onlinecourses.nptel.ac.in/noc19_cs63/preview
6	Blockchain for Business, https://www.ibm.com/downloads/cas/3EGWKGX7

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Blockchain Application Development (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADMML51	Blockchain Application Development (lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADMML51	Blockchain Application Development (lab)	-	-	-	-	25		25

Prerequisite: Cryptography and System Security, Expertise in Programming, Basic knowledge of Computer Security, Networking.	
Lab Objectives: The course aims:	
1	To build and test Private Ethereum Blockchain.
2	To learn the concept of the genesis block and Account in the Blockchain.
3	To get familiar with the mining blocks to create an ether.
4	To understand and apply the concepts of keys, wallets.
5	To acquire the knowledge of gateway and desktop applications.
6	To analyze the applications & case studies of Blockchain.
Lab Outcomes:After successful completion of the course, the student will be able to:	
1	Understand the architecture and functionality of blockchain systems, with a focus on Ethereum.
2	Design and implement the genesis block using Puppeth and develop an account with Smart Contracts.
3	Create mining blocks, and verify account status and Proof of Work (PoW) mechanisms.
4	Demonstrate the safe and secure usage of cryptocurrency exchanges and digital wallets.
5	Develop and deploy a gateway interface for interacting with blockchain applications.

6	Integrate blockchain functionality into mobile and cloud platforms for distributed application access.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	Cryptography in Blockchain, Merkle root tree hash.
2	Create a Blockchain using Python.
3	Create a Crypto Currency using Python for the blockchain implemented experiment 2.
4	Case Study on different blockchain platforms.
Identify a Domain as per your choice and perform the below experiments with respect to the selected domain	
5	Creating Smart Contract and performing transactions using Solidity and Remix IDE.
6	Implement the embedding wallet and transaction using Solidity.
7	Implement the Blockchain platform ethereum using Geth.
8	Implement the Blockchain platform Ganache.
9	Testing Interoperability and Cross-Chain Communication between platforms.
10	Presentation on a suitable platform that meets the needs of the Mini Project.
11	Posting IoT device data to a blockchain ledger.
12	Deploying PKI based identity with blockchain.
13	Setup the Ganache Tool in the system.
14	Write a smart contract using a solidity program to perform the balance transfer from contract to other accounts.
15	Build a multi-signature wallet smart contract to require multiple signatures for transaction approval.
16	Develop a block chain-based healthcare data management system using Hyperledger Fabric or Ethereum, ensuring privacy and compliance.

17	Implement block chain-based notary services for document verification, timestamping, and proof of existence.
18	Develop smart contracts to tokenize real-world assets (e.g., real estate or artwork) on Ethereum or another suitable blockchain.

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 10 -12 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 10-marks, Micro Project: 10-marks, Attendance: 05 marks)

VIVEKANAND EDUCATION SOCIETY'S
Institute of Technology
(An Autonomous Institute Affiliated to University of Mumbai, Approved by A.I.C.T.E & Recognized by Govt. of Maharashtra)
Department of Artificial Intelligence and Data Science

Department of

Artificial Intelligence and Data science

Syllabus (NEP Scheme)

Sem-VI

w.e.f. A.Y. 2025-26

COURSE NAME: DEEP LEARNING

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 61	Deep Learning (Theory)	03	---	---	03	---	---	03
NADPC L61	Deep Learning (Lab)	---	02	---	---	01	---	01

Deep Learning (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 61	Deep Learning (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPC 61	Deep Learning (Theory)	20	20	60	02			100

Prerequisite: Basic mathematics and Statistical concepts, Linear algebra, Machine Learning	
Course Objectives: The course aims:	
1	To understand the fundamentals of deep learning, including neural network basics, activation functions, and loss functions.
2	To apply optimization techniques and regularization techniques such as AdaGrad, Adam, and RMSProp ,data augmentation to train deep neural networks.
3	To analyze the architecture of Convolutional Neural Networks (CNNs) and their applications in image classification.
4	To implement and evaluate Autoencoders, including Variational Autoencoders (VAEs), for unsupervised learning tasks such as image reconstruction.
5	To design and implement sequence learning models such as LSTM and GRU.

6	To explore advanced deep learning techniques, such as Generative Adversarial Networks (GANs), transfer learning, and domain adaptation.
Course Outcomes: Students will be	
1	Explain the basic concepts of deep learning, including activation functions, loss functions, and learning factors.
2	Apply optimization and regularization techniques like momentum based gradient descent, AdaGrad, Adam and data augmentation to optimize deep learning models.
3	Analyze and design CNN architectures for tasks such as image classification using modern deep learning architectures.
4	Implement Autoencoders and VAEs for unsupervised learning tasks like image compression and reconstruction.
5	Apply LSTM and GRU based models for sequence learning problems.
6	Analyze and Apply advanced techniques such as GANs and transfer learning to solve complex deep learning problems.

Module	Detailed Content	Hours
1	**Fundamentals of Deep Learning** Deep Learning Basics, ANN Fundamentals, Representation Power of MLPs, Thre Classes of Deep Learning : Supervised, Unsupervised and reinforcement learning. Learning Factors , Activation functions: Tanh, Logistic, Linear, Softmax, ReLU Leaky ReLU, Loss functions: Squared Error loss, Cross Entropy, Choosing outpu function and loss function, Features, Weights.	4
2	**Optimization and Regularization of Deep Neural Network Optimization**: Learning with backpropagation, Learning Parameters: Gradient Descent (GD), Stochastic and Mini Batch GD, Momentum Based GD, Nestrov Accelerated GD, AdaGrad, Adam, RMSProp Regularization: Overview of Overfitting, Types of biases, Bias Variance Tradeoff Regularization Methods: L1, L2 regularization, Parameter sharing, Dropout, Weight Decay, Batch normalization, Early stopping, Data Augmentation Techniques: Cropping, Flipping, Rotation, Brightness, Contrast, Color Augmentation, Saturation; , Adding noise to input and output, Hyper parameter tuning.	8
3	**Convolutional Neural Networks (CNN): Supervised Learning** Convolutional Neural Networks (CNNs): Convolution, Striding, Padding, Pooling, Relation between input, output and filter size, CNN architecture: Convolution layer, Pooling Layer, Weight Sharing in CNN, Fully Connected NN vs CNN,Image Classification (ImageNet Challenge), Modern Deep Learning Architectures:LeNET, AlexNET, VGG 16. Advanced CNN Concepts: Residual Block, ResNet50, 1x1 Convolution, XceptionNet, EfficientNet.	8
4	**Autoencoders: Unsupervised Learning**	8

	Introduction to Autoencoder,Undercomplete Autoencoder, Overcomplete Autoencoders, Regularization in Autoencoders. Types of Autoencoders:Denoising Autoencoders, Sparse Autoencoders, Contractive Autoencoders. Variational Autoencoders (VAEs): Theory of VAEs: Encoder-decoder architecture, latent variables, Training VAEs Loss functions, optimization, and regularization Application of Autoencoders.	
5	**Recurrent Neural Networks (RNN)** Sequence Learning Problem, Recurrent Neural Network, Bidirectional RNN Backpropagation Through Time (BTT), Limitation of vanilla RNN Vanishing and Exploding Gradients, Truncated BTT. Long Short Term Memory(LSTM): Selective Read, Selective write, Selective Forget.Gated Recurrent Unit (GRU).	7
6	**Advanced Topics in Deep Learning** Generative Adversarial Networks (GANs): Theory of GANs: Generator and discriminator networks, adversarial training, and convergence issues, Transfer learning and domain adaptation techniques.	4
	TOTAL	**39**

Textbooks:

1	Ian Goodfellow, Yoshua Bengio, Aaron Courville. ―Deep Learning, MIT Press Ltd, 2016
2	Li Deng and Dong Yu, ―Deep Learning Methods and Applications, Publishers Inc.
3	Satish Kumar "Neural Networks A Classroom Approach" Tata McGraw-Hill.
4	JM Zurada ―Introduction to Artificial Neural Systems‖, Jaico Publishing House
5	M. J. Kochenderfer, Tim A. Wheeler. ―Algorithms for Optimization‖, MIT Press.

References:

1	Deep Learning from Scratch: Building with Python from First Principles- Seth Weidman by O`Reilley
2	François Chollet. ―Deep learning with Python ―(Vol. 361). 2018 New York: Manning.
3	Douwe Osinga. ―Deep Learning Cookbook‖, O'REILLY, SPD Publishers, Delhi.
4	Simon Haykin, Neural Network- A Comprehensive Foundation- Prentice Hall International, Inc
5	S.N.Sivanandam and S.N.Deepa, Principles of soft computing-Wiley India

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Deep Learning (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPCL61	Deep Learning (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPCL61	Deep Learning (Lab)	-	-	-	-	25	25	50

Prerequisite: Database Management Concepts	
Lab Objectives:	
1	To apply various neural network architectures with optimization and regularization techniques and analyze their impact on model performance.,
2	To implement CNNs, to solve digit and fashion image classification problems.
3	To design and implement Autoencoders (Vanilla, Sparse, Denoising, and Variational) for image reconstruction, compression, and feature learning tasks.
4	To design and train advanced neural network architectures like LSTM, GRU, GANs, and apply transfer learning for specific real-life applications.
Lab Outcomes:Students will be able to	
1	Analyze the performance of fully connected neural networks and different optimizers regularization techniques on deep neural network training.
2	Implement CNN architecture for image classification tasks using MNIST and Fashion MNIST datasets.
3	Design and implement various types of Autoencoders (Vanilla, Sparse, Denoising, and Variational) for image reconstruction and compression.
4	Apply advanced neural network models such as LSTM, GRU, GANs, and Transfer Learning for various real-world applications.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	Train a fully connected neural network (with hidden layers) for digit classification using MNIST dataset.
2	Implement Momentum based Gradient descent / Stochastic and Mini Batch GD.
3	Analyze the impact of different optimizers (AdaGrad, Adam, RMSProp) on model performance.
4	Implement Regularization Techniques to reduce overfitting in deep neural networks.
5	Implement a CNN model for classification on publicly available dataset
6	Design the architecture and implement a Vanilla Autoencoder for image reconstruction.
7	Implement a sparse/ denoising autoencoder to learn useful feature representations.
8	Train a variational autoencoder using Tensorflow on publicly available dataset
9	Design and implement LSTM model for machine translation, predictive maintenance, robot control, time series forecasting etc.
10	Implement GRU for chatbot-style sequence classification or sentiment analysis.
11	Design and train a basic Generative Adversarial Network (GAN) for any real life applications.

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 10 -12 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

Evaluation Exam	
1	Practical Exam based on the entire syllabus

COURSE NAME: GENERATIVE AI

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 62	Generative AI (Theory)	03	---	---	03	---	---	03
NADPC L62	Generative AI (Lab)	---	02	---	---	01	---	01

Generative AI (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPC 62	Generative AI (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPC 62	Generative AI (Theory)	20	20	60	02			100

Prerequisite: Python Programming, Machine Learning, Deep Learning	
Course Objectives: The course aims:	
1	To introduce students to the fundamentals, evolution, and key application areas of Generative AI.
2	To build foundational knowledge in deep learning architectures relevant to generative models, including Autoencoders, VAEs, GANs, and Transformers.
3	To explain the architecture, components, and training strategies of Large Language Models (LLMs), including fine-tuning and parameter-efficient methods.
4	To develop skills in prompt engineering and expose students to frameworks like Retrieval-Augmented Generation (RAG).
5	To provide an understanding of diffusion-based generative models and their use in image and content generation.

6	To explore recent advancements in generative AI including multimodal systems, agentic AI, small language models, and graph-based LLMs.
Course Outcomes: Students will be	
1	To explain the motivation, evolution, key applications, and challenges in generative AI.
2	To describe foundational models like AE, VAE, GANs, Transformers, and their role in generative systems.
3	To outline the LLM architecture and pipeline, explain foundation models, and apply fine-tuning and PEFT techniques.
4	To design prompts for LLMs and explain the RAG framework and its components
5	To summarize the working of diffusion models and their applications in generative tasks.
6	To identify trends in multimodal, agentic, and graph-based generative AI and discuss their implications.

Module	Detailed Content	Hours
1	**Introduction to Generative AI** Generative AI - (Definitions, vs. Discriminative AI); Motivation & need for generative models; History & Evolution - Timeline of GenAI evolution, (GANs → Transformers → ChatGPT); Mathematical Foundations (Probability, Latent Space); Taxonomy of generative models - explicit vs implicit, deterministic vs probabilistic) Real-World Applications - Healthcare, Art, Code, media, synthetic data Core Challenges & Ethical Concerns- hallucination, bias, misuse, deepfakes, scaling, alignment, evaluation, multimodality, Current challenges and trends. Foundation Technologies for GenAI - Overview of AE (Autoencoders), VAE (Variational Autoencoders), GAN (Generative Adversarial Networks), Transformers, and Diffusion models Introduction to Gen AI ecosystem tools : Hugging Face, LangChain, Diffusers, OpenAI APIs, GPT-4, DALL·E, Claude, Gemini, MusicLM	6
2	**Transformers & NLP Foundations** NLP Foundations - Tokenization (WordPiece, BPE), Text representation (Bag-of-Words, TF-IDF), Word embeddings , Word2Vec, GloVe → BERT Transformer Architecture - Seq2Seq Limitations, Need for Attention, Scaled Dot-Product Attention (Math + Code), Attention mechanisms (Self-attention, Multi-head), Encoder (BERT) vs. Decoder (GPT) architectures Positional encoding, layer normalization Vector Databases: Vector Embeddings, Vector DBs: FAISS, Pinecone, Weaviate, Approximate Nearest Neighbor (ANN) search algorithms and their optimization.	6
3	**Large Language Models (LLMs) and Fine-Tuning** Introduction to Large Language Models (LLMs), Concept and importance of pre-trained foundation models, Overview of LLM use cases,	8

	Transformer-based LLM architecture and end-to-end pipeline, Tokenization to decoding, Inference and decoding strategies (greedy, beam search, top-k, top-p, temperature), Issues under LLMs. Need for fine-tuning in LLMs, Full fine-tuning, Instruction Fine tuning, Parameter-efficient fine-tuning (PEFT), LoRA (Low-Rank Adaptation), QLoRA (Quantized LoRA), Prefix tuning Introduction to Small Language Models (SLMs) Definition and characteristics of small language models (typically <1B parameters). Comparison of SLMs versus LLMs Examples of SLMs: DistilGPT2, TinyLlama, Alpaca, Phi-2, Real-world applications of SLMs in mobile apps, IoT devices, and specialized domains	
4	**Prompt Engineering and Retrieval-Augmented Generation (RAG)** Concept of prompting in language models, Analysis of prompt components and their impact on model outputs, Prompt structure and design patterns, Zero-shot, one-shot, and few-shot prompting Prompt optimization techniques (clarity, role assignment, constraints), Chain-of-thought prompting for reasoning tasks Prompt engineering use cases in classification, summarization, and code generation Challenges in prompting large models (hallucination, verbosity, inconsistency) Introduction to Retrieval-Augmented Generation (RAG), Motivation for RAG, RAG architecture and pipeline, Components of RAG systems: retriever, generator, and knowledge base. Vector databases and embedding models for efficient retrieval. Integration of external knowledge sources with LLMs. Building end-to-end RAG pipelines using frameworks, Applications of RAG in knowledge-based QA, enterprise chatbots, and legal/medical assistants	6
5	**Diffusion Models** Introduction to generative models for image and text generation, Limitations of GANs and VAEs leading to diffusion models, Core idea of diffusion: forward (noise addition) and reverse (denoising) processes Overview of popular diffusion models like DDPM and Stable Diffusion, Key components: noise scheduler, UNet model, denoising steps text-to-image generation using prompts, Applications of diffusion models in art, design, and media, Examples: Stable Diffusion, DALL·E 2, and Midjourney Safety and ethical concerns in AI image generation	5
6	**Advanced Topics in Generative AI** Overview of recent trends in Generative AI Agentic AI: Introduction, Definition and capabilities of self-directed AI systems, Autonomous AI agents and their components, Examples of agentic AI: Chatbots with self-improvement (e.g., AutoGPT, BabyAGI), Multi-agent systems and collaboration Multimodal Systems: Introduction, Integrating multiple data types (text, images, audio) into AI models, Text-to-image, image-to-text, and text-to-video generation, Applications in interactive media, healthcare, and autonomous systems,	8

	Graph LLMs: Introduction, Graph Neural Networks (GNNs) basics, Knowledge graphs and LLM integration, Graph Transformers (GraphGPS), Graph recommendation systems, drug discovery Emerging Architectures-Mixture-of-Experts models, Long-context processing techniques, Neuro-symbolic integration Ethical concerns and the future of agentic AI	
	TOTAL	**39**

Textbooks:	
1	Rehmani, Altaf. Generative AI for Everyone: Understanding the Essentials and Applications of This Breakthrough Technology. 2024.
2	Applied Generative AI for Beginners: Practical Knowledge on Diffusion Models,ChatGPT, and Other LLMs by Akshay Kulkarni, Adarsha Shivananda, Anoosh Kulkarni and Dilip Gudivada
3	Introduction to GenerativeAI, Numa Dhamani, Kindle Edition, 2024
4	Prompt Engineering for Generative AI. Phoenix, James, and Mike Taylor. 2024.
5	Generative AI Foundations in Python Paperback – 26 July 2024, by Carlos Rodriguez, Samira Shaikh
6	Introduction to Large Language Models, Tanmoy Chakraborty

References:	
1	Generative AI in Action, By Amit Bahree, Manning
2	Transformers for Machine Learning by Uday Kamath
3	Generative Deep Learning: A Practical Guide by David Foster
4	Generative AI for Everyone: Deep learning, NLP, and LLMs for creative and practical applications - Karthikeyan Sabesan, Sivagamisundari, Nilip Dutta

Websites and Courses:	
1	Hugging Face – https://huggingface.co/learn
2	DeepLearning.AI – https://www.deeplearning.ai
3	Designing Autonomous Agents (online MIT lectures & blog series)
4	Multimodal Machine Learning (Carnegie Mellon course notes)
5	Prompt Engineering Guide – https://github.com/dair-ai/Prompt-Engineering-Guide

6	Illustrated Transformer & Diffusion – https://jalammar.github.io
7	https://www.coursera.org/learn/generative-ai-prompt-engineering-for-everyone
8	https://elearn.nptel.ac.in/shop/iit-workshops/completed/leveraging-generative-ai-for-teaching-prog ramming-courses/?v=c86ee0d9d7ed
9	https://elearn.nptel.ac.in/shop/iit-workshops/completed/introduction-to-language-models/?v=c86ee0d9d7ed

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Generative AI (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPCL62	Generative AI (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPCL62	Generative AI (Lab)	-	-	-	-	25	25	50

Prerequisite: Python Programming, Machine Learning, Deep Learning	
Lab Objectives:	
1	To investigate and experiment with core Generative AI frameworks such as OpenAI, Hugging Face, and LangChain to build familiarity with development environments.
2	To construct and refine prompts for tasks like summarization, generation, and translation using LLMs, aligning with model behavior and prompt structure.
3	To design retrieval-augmented generation (RAG) workflows using vector databases for domain-specific knowledge retrieval and response generation.
4	To perform fine-tuning and parameter-efficient adaptation of pre-trained language models for specific use cases.
5	To integrate prompting, RAG, fine-tuning, and agent-like behavior into a functional mini-project that demonstrates a real-world GenAI application.
Lab Outcomes: Sstudents will be able to:	
1	Demonstrate the ability to explore and compare foundational GenAI tools and frameworks such as OpenAI, Gemini, LangChain, and Hugging Face.
2	Design and refine prompts for language models to perform tasks such as summarization, text generation, classification, and question answering.
3	Construct retrieval-augmented generation (RAG) systems by integrating vector databases and embedding models.

4	Apply fine-tuning and parameter-efficient techniques like LoRA to adapt pre-trained language models for specific domains.
5	Analyze the design and implementation of generative applications involving multimodal inputs, diffusion techniques, or graph-based structures.
6	Develop mini-projects that integrate multiple GenAI components including prompting, RAG, fine-tuning, and agent-based reasoning.

Suggested Experiments Students are required to complete at least 10 experiments.

Sr. No.	Name of the Experiment
1	Explore generative AI frameworks (OpenAI, Gemini, Copilot) – OpenAI ChatGPT, Google Gemini, GitHub Copilot
2	Generate simple text using a pre-trained model – Hugging Face Transformers, Google Colab
3	Visualize tokenization process – Hugging Face Tokenizers, NLTK
4	Compare generative vs non-generative models – Scikit-learn, PyTorch, Hugging Face
5	Analyze outputs from generative models – OpenAI API, Stability AI, Hugging Face
6	Denoising Autoencoder – PyTorch, TensorFlow
7	Visualize Transformer attention – PyTorch, Transformers, Matplotlib
8	Document retrieval using vector DB – FAISS, ChromaDB, SentenceTransformers
9	Trace LLM pipeline – Hugging Face, BERTviz
10	Fine-tune a pre-trained LLM – Hugging Face, Open LLaMA, Alpaca
11	Apply PEFT (e.g., LoRA, adapters) – PEFT (Hugging Face), QLoRA
12	Compare LLM vs SLM – TinyLLM, DistilBERT, Mistral, OpenChat
13	Design effective prompts – OpenAI Playground, Hugging Face Inference API
14	Implement a RAG pipeline – LangChain (optional), FAISS, ChromaDB
15	Text-to-image generation with diffusion – Hugging Face, Diffusers, Stability AI
16	Multimodal generation (image-text) – CLIP, BLIP, Hugging Face models

17	Build a planning-based agent – LangGraph, AgentLite, custom Python scripts
18	Micro Project - on topics of – LLMs(Fine tuning prompts RAG), Agents, GraphLLMs

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Suggested Micro Project Ideas.

Sr. No.	Title
1	Personalized AI Study Assistant *(RAG, Agentic AI)*
2	Goal-Oriented Travel Planner Agent *(Agentic AI, Planning)*
3	Voice-to-Story Multimodal Generator *(Multimodal, Diffusion + LLM)*
4	Domain-Specific Chatbot with Memory *(RAG, LLM + Memory)*
5	Job Application Assistant *(LLM, Prompt Engineering)*
6	AI-based Code Explainer & Debugger *(LLM Prompting, Programming)*
7	Resume Builder and Skill Gap Analyzer *(LLM, Fine-tuning optional)*
8	AI News Digest Generator from Multiple Sources *(Summarization, RAG)*
9	Mental Wellness Companion Chatbot *(Prompting, LLM Dialogue)*
10	Graph-based Academic Paper Recommender *(Graph LLM)*
11	Knowledge Graph Question Answering Agent *(Graph LLM, RAG)*
12	Timetable Planner using Natural Language Input *(LLM Reasoning)*
13	Story Generator from a Single Sentence Prompt *(Text Generation)*
14	Summarizer for Class Notes or PDFs *(LLM Summarization)*

15	Basic Image Caption Generator *(Multimodal - CLIP, BLIP)*
16	Code Comment Generator *(Code LLM Prompting)*
17	Quiz Generator from Textbook Chapters *(LLM Prompting)*

Term Work:	
1	Term work should consist of 6 to 8 experiments. And one Micro Project on topics – LLMs(Fine tuning prompts RAG), Agents, GraphLLMs
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)
Evaluation Exam	
1	Practical Exam based on the entire syllabus

COURSE NAME: MAJOR PROJECT -1

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tut	Theory	TW/PR	Tut	Total
NADP61	Major Project -I	--	04	-	--	02	-	02

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADP61	Major Project -I	—	—	—	–	25	25	50

Prerequisite:	
Course Objectives: The course aims:	
1	To get acquainted with the process of identifying the needs and converting it into the problem.
2	To familiarize the process of solving the problem in a group.
3	To acquaint with the process of applying basic engineering fundamentals to attempt solutions to the problems.
4	To inculcate the process of self-learning and research.
5	To develop the ability to analyze and interpret results obtained through theoretical studies, experiments, or simulations for making informed decisions.
6	To familiarize with project planning, execution, and management principles for effective completion of project work within defined timelines.
Course Outcomes: Students will be	
1	Identify problems based on societal /research needs.
2	Apply Knowledge and skill to solve societal problems in a group
3	Draw the proper inferences from available results through theoretical/ experimental/simulations
4	Analyse the impact of solutions in societal and environmental context for sustainable development.
5	Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
6	Demonstrate project management principles during project work.

Guidelines:

1. Project Topic Selection and Allocation:

Project topic selection Process to be defined and followed:

- Project orientation can be given at the end of sixth semester.
- Students should be informed about the domain and domain experts whose guidance can be taken before selecting projects.
- Student's should be recommended to refer papers from reputed conferences/journals like IEEE, Elsevier, ACM etc. which are not more than 3 years old for review of literature.
- Dataset selected for the project should be large and realtime
- Students can certainly take ideas from anywhere, but be sure that they should evolve them in the unique way to suit their project requirements. Students can be informed to refer to Digital India portal, SIH portal or any other hackathon portal for problem selection.

Topics can be finalized with respect to following criterion:

- **Topic Selection:** The topics selected should be novel in nature (Product based, Application based or Research based) or should work towards removing the lacuna in currently existing systems.**Technology Used:** Use of latest technology or modern tools can be encouraged. AI, ML, DL, NNFS, NLP based algorithms can be implemented
  - Students should not repeat work done previously (work done in the last three years).
  - Project work must be carried out by the group of at least 3 students and maximum
  - The project work can be undertaken in a research institute or organization/Industry/any business establishment. (out-house projects)
  - The project proposal presentations can be scheduled according to the domains and should be judged by faculty who are experts in the domain.
  - Head of department and senior staff along with project coordinators will take decisions regarding final selection of projects.

- Guide allocation should be done and students have to submit a weekly progress report to the internal guide.
- Internal guide has to keep track of the progress of the project and also has to maintain an attendance report. This progress report can be used for awarding term work marks.
- In case of industry/ out-house projects, visit by internal guide will be preferred and external members can be called during the presentation at various levels

2. Project Report Format:

At the end of semester, each group needs to prepare a project report as per the guidelines issued by the University of Mumbai.

A project report should preferably contain following details:

- Abstract
- Introduction
- Literature Survey/ Existing system
- Limitation Existing system or research gap
- Problem Statement and Objective
- Proposed System
- Analysis/Framework/ Algorithm
- Design details
- Methodology (your approach to solve the problem) Proposed System
- Experimental Set up
- Details of Database or details about input to systems or selected data
- Performance Evaluation Parameters (for Validation)
- Software and Hardware Setup
- Implementation Plan for Next Semester
- Timeline Chart for Term l and Term-II (Project Management tools can be used.)
- References

Desirable: Students can be asked to undergo some Certification course (for the technical skill set that will be useful and applicable for projects.)

COURSE NAME: CLOUD COMPUTING FOR AI

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 611	Cloud Computing for AI (Theory)	03	---	---	03	---	---	03
NADPE L611	Cloud Computing for AI (Lab)	---	02	---	---	01	---	01

Cloud Computing for AI (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 611	Cloud Computing for AI (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 611	Cloud Computing for AI (Theory)	20	20	60	02			100

Prerequisite: Fundamentals of Computer Networks	
Course Objectives: The course aims	
1	Understand cloud computing models (IaaS/PaaS/SaaS) and their role in AI scalability.
2	Explore virtualization technologies and containerization strategies that support scalable AI model development and deployment.
3	Analyze AI workflows on major cloud platforms (AWS/Azure/GCP) with AutoML and serverless tools.
4	Examine cloud deployment techniques, including automation, network considerations, and edge computing for AI applications.
5	Evaluate security concerns in cloud environments, with a focus on data privacy, integrity, and compliance in AI systems.

6	Design cloud-native architectures for AI applications that support distributed training, high availability, and efficient inference.
Course Outcomes: Students will be able to	
1	Explain the core concepts of cloud computing and assess how different cloud models and services support AI development.
2	Apply virtualization and containerization tools (e.g., Docker, Kubernetes, GPU Virtualization) for creating scalable AI environments.
3	Compare cloud service providers based on their AI capabilities and choose appropriate services for building AI pipelines.
4	Implement deployment strategies for AI applications in the cloud, including mobile and edge AI use cases.
5	Identify and address security challenges in cloud-based AI systems, including data confidentiality and ethical AI risks.
6	Design and evaluate cloud-native architectures for AI workloads, leveraging distributed computing and real-time processing techniques.

Module	Detailed Content	Hours
1	**Introduction to Cloud Computing** Cloud Computing Models (IaaS, PaaS, SaaS), Deployment Models (Public, Private Hybrid, Community), Benefits and Limitations of Cloud Computing, Major Cloud Providers Overview (AWS, Azure, GCP), Types of Cloud Environments: Public Private, Hybrid. Role of Cloud in AI: Enabling scalable training, model hosting, and data processing , Comparison with Grids, Business Drivers and Use Cases in AI Adoption.	5
2	**Virtualization & Scalability** Virtualization Basics and Benefits, Levels of Virtualization and their role in AI clusters, Virtual Machine Monitors, OS-Level Virtualization, and Middleware Virtualization Technologies: VMs vs Containers, Hypervisors: Type I & II, Xen Architecture, Full vs Para-Virtualization, Containerization using Docker, Kubernetes, Auto-scaling and Load Balancing, GPU Virtualization and Containerization for AI Role of CUDA and Hardware Accelerators in Virtualized Environments	6
3	**Cloud Services & Business Value in AI** XaaS: IaaS, PaaS, SaaS, DBaaS, Azure ML, Google AI Platform, AI-Driven PaaS Environments (AutoML, MLflow, Vertex AI), Training ML Models on Cloud storage & Databases for AI Workloads, Serverless AI Services, Language and Framework Support (TensorFlow, PyTorch on cloud platforms), Comparative Study of AI Capabilities in Cloud Providers (AWS, Azure, GCP), Scalability	8
4	**Cloud Deployment & Edge AI**	7

	Deployment Considerations for AI Applications, Network Requirements and Cloud Topologies, Common Issues in Large-Scale AI Data Pipelines, Automation in Cloud AI Workflows (CI/CD for ML), Edge Computing Concepts, Federated Learning & AI on Edge Devices (privacy-preserving AI) , Use Cases: Smart Cities, Autonomous Systems, Using Kubernetes and Kubeflow for AI Deployment	
5	**Security & Ethics** Cloud Security Fundamentals,Virtualization Security Essentials, SaaS, PaaS, IaaS Security Models, Data Privacy in AI (e.g., GDPR, differential privacy in cloud ML), Encryption and Secure Data Pipelines, Ethical Implications of AI on Cloud Cloud Firewalls and Gateways, Security as a Service for AI Applications Identity and Access Management (IAM), Bias, Fairness, and Explainability in AI as Emerging Security/Trust Concerns	**7**
6	**Architectures for Cloud-Based AI Applications** Requirements for AI-Centric Cloud Applications, Traditional vs AI-Optimized Architectures, Multi-Tier Architectures for ML Pipelines SOA: Resource-Oriented,Method-Oriented and Event-driven Architectures Distributed Training and Data Parallelism , Use of In-Memory Operations for Real-Time AI Inference (e.g., RedisAI)	**6**
TOTAL		**39**

Textbooks:	
1	Cloud computing: concepts, Technology and architecture : The Pearson Service Technology Series from Thomas Erl) 1st Edition 2 Cloud computing for Dummies
2	Cloud computing for Dummies
3	Michael J. Kavis, "Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)",Wiley.
4	Himanshu Singh, "Practical Machine Learning with AWS: Process, Build, Deploy, and Productionize Your Models Using AWS", Apress.

References:	
1	Rajkumar Buya,' Cloud computing principles and Paradigms', Wiley.
2	Kai Hwang,' Distributed and cloud computing', MK Publications.
3	Cloud computing, black book, Dreamtech publication.
4	Using Google Apps engine O'reilly Publication

5	Daniel Situnayake and Pete Warden, AI at the Edge: Solving Real World Problems with Embedded Machine Learning" O'Reilly Media
6	Giuseppe Ciaburro, "Hands-On Machine Learning on Google Cloud Platform: Implementing Smart and Scalable Models Using the Google Cloud AI Platform", O'Reilly Media.
7	Thomas Erl, Zaigham Mahmood, and Ricardo Puttini, "Cloud Computing: Concepts, Technology & Architecture", O'Reilly Media.

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Cloud Computing for AI (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL611	Cloud Computing for AI (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPEL611	Cloud Computing for AI (Lab)	-	-	-	-	25		25

Prerequisite: Operating System	
Lab Objectives:	
1	To provide hands-on experience with virtualization technologies and containerization tools (e.g., VirtualBox, Docker, Kubernetes) relevant to AI model deployment.
2	To enable students to experiment with various cloud deployment models (public, private, hybrid, community) and evaluate their suitability for AI and machine learning workloads.
3	To expose students to practical implementation of different cloud service models, including IaaS, PaaS, SaaS, SECaaS, and DBaaS, in the context of developing and deploying AI applications.
4	To familiarize students with security, privacy, and ethical issues in cloud computing and implement mechanisms such as encryption, access control, and identity management in cloud-based AI environments.
Lab Outcomes:	
1	Set up and manage virtualized and containerized environments for AI development using industry tools like Docker and Kubernetes.
2	Deploy AI applications on different types of cloud infrastructures and analyze their performance, cost-efficiency, and scalability.
3	Configure and utilize cloud services such as AWS EC2 (IaaS), Google Colab or Azure ML Studio (PaaS), and SaaS-based AI APIs for practical AI tasks.
4	Implement basic security mechanisms for cloud-based AI systems, including role-based access control, encryption for data-at-rest and in-transit, and secure model serving.

Suggested Experiments In the lab work, the students will Implement the state-of-the-art	
1	To study and Implement Infrastructure as a Service using AWS/Microsoft Azure.
2	To study and Implement Platform as a Service using AWS Elastic Beanstalk/ Microsoft Azure App Service.
3	To study and Implement Storage as a Service using Own Cloud/ AWS S3, Glaciers/ Azure Storage.
4	To study and Implement Database as a Service on SQL/NOSQL databases like AWS RDS, AZURE SQL/ MongoDB Lab/ Firebase.
5	To study and Implement Security as a Service on AWS/Azure
6	To study and implement Identity and Access Management (IAM) practices on AWS/Azure cloud.
7	To study and Implement Containerization using Docker
8	To study and implement container orchestration using Kubernetes
9	**Micro-project:** Design a Web Application hosted on a public cloud platform [It should cover the concept of IaaS, PaaS, DBaaS, Storage as a Service, Security as a Service etc.]

Term Work:	
1	Term work should consist of 8 experiments + Micro project.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Micro Project: 10-marks)

COURSE NAME:AGILE PROJECT MANAGEMENT

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 612	Agile Project Management (Theory)	03	---	---	03	---	---	03
NADPE L612	Agile Project Management (Theory)	---	02	---	---	01	---	01

Agile Project Management (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 612	Agile Project Management (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 612	Agile Project Management (Theory)	20	20	60	02			100

Course Objectives: The course aims:	
1	Understand the fundamentals of software engineering, software processes, and the evolution toward agile methodologies.
2	Comprehend agile principles, practices, and frameworks like Scrum, Kanban, and XP.
3	Apply agile planning techniques, including sprint planning, product roadmap development, and adaptive strategies.
4	Analyze the roles and responsibilities within agile teams and effectively manage time, cost, scope, and risks in agile environments.
5	Apply PMBOK principles and W5HH framework to design project charters, evaluate feasibility, and develop RFPs for IT projects.

6	Utilize CPM/PERT, WBS, and Earned Value Analysis to construct optimized schedules and monitor project performance.

Course Outcomes: Students will be able to	
1	Describe Agile principles and contrast them with traditional models.
2	Develop product roadmaps and sprint plans using prioritization techniques.
3	Implement risk management strategies (RMMM) for Agile projects and analyze team dynamics.
4	Design scaling strategies (SAFe/LeSS) and measure agility using KPIs.
5	Create project charters and feasibility reports using PMBOK/W5HH principles.
6	Construct optimized schedules using CPM/PERT and track progress via Earned Value Analysis.

Module	Detailed Content	Hours
1	**Introduction** Introduction to Software Engineering: Software, Evolving role of software, Modernizing Project Management, Software Process Models Understanding Agility, Agile Manifesto and Principles, Benefits of Agility, Focussing on the Customer, Agile Frameworks – Scrum, Kanban, XP	7
2	**Agile Planning and Execution** Agile Approaches, Environments, Behaviours and Team Product Vision and Roadmap, Planning releases and sprints, Daily Planning, Showcasing Work, Inspecting and adapting.	8
3	**Agility Management** Managing Portfolio, Scope and Procurement, Time and Cost, Team Dynamics and Communication, RMMM, Quality and Risk.	7
4	**Ensuring Success** Building a foundation, De-Scaling Across Teams, Being a change Agent, Key benefits of Agile Project Development, Key factors for Agile Project Development, Assuring agility	6
5	**IT Project Management** Introduction, 4 P's, W5HH Principle, Need for Project Management, Project Life cycle and ITPM, Project Feasibility, RFP, PMBOK Knowledge areas, Business Case, Project Planning, Project Charter and Project Scope.	6
6	**Project Scheduling**	5

	Project Scheduling:Defining a Task Set for the Software Project, Timeline charts WBS, Developing the Project Schedule, Network Diagrams (AON, AOA), CPM and PERT, Gantt Chart , Tracking the Schedule, Earned Value Analysis	
TOTAL		**39**

Textbooks:	
1	Mark C. L, Steven J. O, Dean J. K, Agile Project Management For Dummies, 3rd Edition ,Wiley Publication
2	Information technology project management 4th Edition, Jack T. Marchewka
3	Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill
4	Jeff Patton, User Story Mapping: Discover the Whole Story, Build the Right Product, O'Reilly Media, 2014.

References	
1	The Art of Agile Development – by James Shore and Shane Warden
2	Robert C. Martin, Clean Agile: Back to Basics, Prentice Hall, 2019.
3	Henrik Kniberg, Kanban and Scrum – Making the Most of Both, C4Media/InfoQ, 2010
4	Alistair Cockburn, Agile Software Development: The Cooperative Game, 2nd Edition, Addison-Wesley, 2006.

Useful Links	
1	Agile Alliance Comprehensive resource on all Agile methodologies, frameworks, case studies. https://www.agilealliance.org
2	Scrum.org Great source for Scrum training, certifications, guides, and free learning paths. https://www.scrum.org
3	Mountain Goat Software – by Mike Cohn Blog, videos, and articles on Agile planning, estimation, and user stories. https://www.mountaingoatsoftware.com

4	Agile Modeling Explains lightweight modeling and documentation strategies in Agile. https://agilemodeling.com
5	Jira Software Guides (Atlassian) Official tutorials and guides on how Jira supports Agile workflows. https://www.atlassian.com/agile

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming-based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Agile Project Management (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL612	Agile Project Management (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NCMPEL612	Agile Project Management (Lab)	-	-	-	-	25		25

Lab Objectives:	
1	Understand DevOps practices which aims to simplify Software Development Life Cycle.
2	Study different Version Control tools like GIT, CVS or Mercurial
3	Integrate and deploy tools like Jenkins and Maven, which is used to build, test and deploy applications in DevOps environment
4	Understand the importance of Jenkins to Build and deploy Software Applications on server environment
5	Use Docker to Build, ship and manage applications using containerization
6	Understand the concept of Infrastructure as a code and install and configure Ansible tool
Lab Outcomes: Students will be able to	
1	Understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements
2	Obtain complete knowledge of the ―version control system‖ to effectively track changes augmented with Git and GitHub
3	Understand the importance of Selenium and Jenkins to test Software Applications
4	Understand the importance of Jenkins to Build and deploy Software Applications on server environment
5	Understand the concept of containerization and Analyze the Containerization of OS images and deployment of applications over Dockerk.
6	To Synthesize software configuration and provisioning using Ansible.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr.No.	**Name of the Experiment**
1	To understand DevOps: Principles, Practices, and DevOps Engineer Role and Responsibilities
2	To understand Version Control System / Source Code Management, install git and create a GitHub account
3	To Perform various GIT operations on local and Remote repositories using GIT Cheat-Sheet
4	To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job
5	To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.
6	To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes.
7	To Setup and Run Selenium Tests in Jenkins Using Maven.
8	To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers
9	To learn Dockerfile instructions, build an image for a sample web application using Dockerfile.
10	To install and Configure Pull based Software Configuration Management and provisioning tools using Puppet
11	To learn Software Configuration Management and provisioning using Puppet Blocks(Manifest, Modules, Classes, Function)
12	To provision a LAMP/MEAN Stack using Puppet Manifest

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 10 -12 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: AI FOR HEALTHCARE

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 613	AI for Healthcare (Theory)	03	---	---	03	---	---	03
NADPE L613	AI for Healthcare (Lab)	---	02	---	---	01	---	01

AI for Healthcare (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 613	AI for Healthcare (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 613	AI for Healthcare (Theory)	20	20	60	02			100

Prerequisites Desire to learn Artificial Intelligence and machine learning, Knowledge of higher school level math	
Course Objectives: The course aims :	
1	Recall the fundamental concepts of AI and ML and describe their applications in healthcare.
2	Explain advanced AI techniques including deep learning, reinforcement learning, and natural language processing applicable to medicine.
3	Apply AI methods for medical imaging, diagnostics, disease prediction, and treatment planning.
4	Analyze ethical, legal, and social issues associated with the use of AI in healthcare, including bias, privacy, and equity.

5	Evaluate research studies and AI applications to determine their effectiveness and relevance in healthcare settings.
6	Design AI-based solutions for healthcare challenges using appropriate tools, platforms, and methodologies.
Course Outcomes: Student will be able to:	
1	Understand the fundamentals of artificial intelligence and machine learning and their applications in medicine.
2	Explore advanced AI techniques, including deep learning, reinforcement learning, and natural language processing.
3	Learn about the use of AI in medical imaging, diagnostics, disease prediction, and treatment planning.
4	Examine the ethical, legal, and social implications of AI in medicine, including issues of bias, privacy, and equity.
5	Gain hands-on experience with AI tools and platforms through practical exercises and projects.
6	Critically evaluate research studies and applications of AI in healthcare.

Module	Detailed Content	Hours
1	**Introduction to Artificial Intelligence in Medicine** Definition and scope of AI in healthcare, Role of AI in modern healthcare systems, AI-assisted diagnostics and treatments, Applications of AI in clinical decision support systems (CDSS),AI in medical research and drug discovery	4
2	**Healthcare Data Analysis** Overview of Machine Learning and Deep Learning, Transfer learning and pre-trained models, Overview of tools like R, Python, Statistical and Visualization tools, Sources of the healthcare data, Pre-processing of the healthcare data, Handling of the healthcare data, Creation of analysis-ready datasets, Diagnostic decision support systems (DDSS),Transfer learning and pre-trained models, Applications of AI in radiology, pathology, and ophthalmology	12
3	**Natural Language Processing (NLP) in Healthcare** Basics of NLP: Tokenization, stemming, and lemmatization,Named Entity Recognition (NER) for medical texts, Information extraction from Electronic Health Records (EHR), - Applications of NLP in medical analysis	5
4	**Data Mining and Big Data Analytics in Healthcare** Introduction to Data Mining in Healthcare,Techniques for data extraction,	11

	preprocessing, and cleansing, Predictive and descriptive data mining applications **Big Data Analytics in Healthcare:** Characteristics of big data in healthcare,Tools for big data management (e.g., Hadoop, Spark), Real-time analytics for patient monitoring and treatment **IBM Watson in Healthcare:** Watson's role in medical diagnostics and decision-making,Case studies of IBM Watson applications	
5	**Ethical, Legal, and Social Implications (ELSI) of AI in Medicine** Bias and fairness in AI algorithms.Privacy and security of healthcare data,Data anonymization techniques,Regulations: GDPR, HIPAA, and regional policies, Ethical considerations in AI healthcare solutions	4
6	**Case Studies and Future Trends in AI Healthcare** Real-world case studies on disease detection, treatment optimization, and hospital management,Emerging trends in AI healthcare: Federated learning for data privacy, Generative AI for medical imaging, Explainable AI (XAI)	3
	TOTAL	39

** Healthcare-related case studies should be discussed in relevance to the topics mentioned in the syllabus.*

References:	
1	Russell, S. and Norvig, N. Artificial Intelligence: A Modern Approach. Prentice Hall Series in Artificial Intelligence 3
2	AI-First Healthcare by Kerrie L. Holley, Siupo Becker Released April 2021, Publisher(s): O'Reilly Media, Inc.ISBN: 9781492063155
3	"AI and Machine Learning for Healthcare" – Arvind Rajan (O'Reilly)
4	Adam Gibson, Josh Patterson, Deep Learning, O'Reilly Media, Inc.
5	"Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes" – by Arjun Panesar
6	"Healthcare Analytics Made Simple: Techniques in Healthcare Computing Using Machine Learning and Python" – by Vikas (Vik) Kumar

Sr. No.	**Rubrics**	**Marks**
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks

2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming-based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

AI for Healthcare (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL613	AI for Healthcare (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPEL613	AI for Healthcare (Lab)	-	-	-	-	25		25

Prerequisite: Concepts of AI and Machine Learning	
Lab Objectives:	
1	To familiarize students with AI algorithms used in healthcare diagnostics, imaging, and predictive analytics.
2	To provide hands-on experience in applying machine learning, deep learning, and NLP techniques to healthcare problems.
3	To develop skills in implementing AI tools and platforms for medical data analysis and decision support systems.
4	To create awareness about ethical, legal, and privacy concerns related to AI applications in healthcare.
Lab Outcomes:	
1	Implement AI algorithms for disease diagnosis, medical imaging, and healthcare analytics.
2	Demonstrate the ability to process and analyze clinical text data using NLP techniques.
3	Design basic AI-powered healthcare applications like chatbots and decision support systems.
4	Critically evaluate AI models for bias, fairness, explainability, and ethical compliance in healthcare scenarios.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	Name of the Experiment
1	Disease Diagnosis using Decision Trees.
2	Medical Image Classification using CNN
3	Medical Image Segmentation for Lesion Detection
4	Disease Prediction using Logistic Regression
5	Personalized Treatment Recommendation using KNN
6	Symptom Extraction from Clinical Text using Named Entity Recognition (NER) using spaCy/NLTK
7	AI Chatbot for Healthcare Queries (Rule-Based Chatbot using Rasa/Dialogflow)
8	Big Data Analysis in Healthcare :MapReduce with Apache Spark (Basic Analytics)
9	Medical Diagnosis Simulation using Cognitive Computing API like IBM Watson
10	Bias and Fairness Analysis in AI Models using SHAP (SHapley Additive exPlanations) for Explainability

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of a minimum 10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: EDGE COMPUTING

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 621	Edge Computing (Theory)	03	---	---	03	---	---	03
NADPE L621	Edge Computing (Lab)	---	02	---	---	01	---	01

Edge Computing (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 621	Edge Computing (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 621	Edge Computing (Theory)	20	20	60	02			100

Prerequisite: Cloud computing	
Course Objectives: The course aims	
1	Understand the fundamentals of edge computing and its role in IoT systems.
2	Analyze and compare different edge computing architectures, platforms and frameworks.
3	Analyze and evaluate data processing at the edge and Edge analytics.
4	Understand the fundamentals of fog computing and its frameworks.
5	Demonstrate effective communication and collaboration skills in developing edge computing projects.
6	Demonstrate the edge computing for specific IoT use
Course Outcomes: Students will be able to	
1	Understand the fundamentals of Edge Computing, its evolution, use cases, and role in IoT ecosystems.

2	Learn Edge Computing architectures and platforms, including virtualization and containerization techniques.
3	Explore data acquisition, processing, and analytics at the edge, including Edge Machine Learning and resource management.
4	Understand Fog Computing concepts, architecture, and its comparison with cloud and edge computing.
5	Gain knowledge of Fog Computing programming models, middleware, performance metrics, and IIoT applications.
6	Analyze real-world applications and case studies of Edge and Fog Computing in industries like smart cities and healthcare

Module	Detailed Content	Hours
1	**Introduction to Edge Computing** Understanding Edge Computing: Evolution, Use cases, advantages, disadvantages, Overview of edge computing and its significance in IoT, Challenges and opportunities in edge computing.	7
2	**Edge Computing Infrastructure** Edge computing architectures and components: Requirements and views for Edge architecture, Edge Computing Reference Architecture, critical elements for Edge architecture, Challenges for Edge application Development. Setting up Edge computing environments: development tools, python libraries Edge computing platforms and frameworks: AWS IoT Greengrass, Azure IoT Edge Google Cloud IoT Edge, IBM Edge Application Manager, KubeEdge. Virtualization and containerization for edge computing: Introduction to Virtualization and containerization. Advantages of Virtualization and Containerization in Edge Computing. Resource Efficiency, Faster Time to Market.	9
3	**Data Processing at the Edge** Data Acquisition and Processing: Data handling, python data handling, data storage and cloud connectivity, Data Aggregation, Data Timestamping and Synchronization, Data Security and Privacy. Edge analytics and machine learning at the edge: Introduction to Edge Analytics. Edge Machine Learning. Mode Selection and Optimization. Collaborative Edge Learning, Resource managemen and task offloading strategies: Task Offloading, Edge-Cloud Collaboration Dynamic Resource Provisioning, Edge caching and data synchronization: Introduction to Edge caching and data synchronization, Benefits of Edge Caching and Data Synchronization, Challenges in Edge Caching and Data Synchronization.	7
4	**Introduction to Fog Computing** Introductions, Comparison with cloud computing and IoT, Data Management in Fog Computing. Comparison with cloud computing and edge computing. Fog Computing Architecture. Fog node and infrastructure components. Hierarchical and distributed models. Programming Models and Tools for Fog Computing	6
5	**Fog computing programming languages and frameworks**	6

	Middleware and software platforms. Development and deployment considerations Industrial Internet of Things (IIoT). Performance Evaluation and Metrics in Fog Computing. Simulation and modelling techniques. Applications and Use Cases of Fog Computing	
6	**Applications and Case Studies** High-Potential Use cases, Edge computing for smart cities. Industrial IoT and edge computing. Edge computing in Healthcare.	4
	TOTAL	39

Textbooks:	
1	"Fog and Edge Computing" by Rajkumar Buyya, Satish Narayana Srirama, Wiley Publications
2	"Edge Computing: Models, Technologies, and Applications" by Mung Chiang, Bharath Balasubramanian, and H. Vincent Poor.
3	Edge Computing with Python: End-to-end Edge Applications, Python Tools and Techniques, Edge Architectures, and AI Benefit" by Abhinandan Bhadauria, BPB publications.
4	"Edge Computing: Simply in Depth" by Ajit Singh,
5	Edge Computing: Fundamentals, Advances and Applications (Advances in Industry 4.0 and Machine Learning) by K. Anitha Kumari, G. Sudha Sadasivam, D. Dharani, M Niranjanamurthy CRC Press.

References:	
1	"Edge Computing for IoT: Architectures and Applications" by Bharat Bhargava, Sudip Misra, Valentina E. Balas, and Raghvendra Kumar
2	"Practical Industrial Internet of Things Security: A practitioner's guide to securing connected industries" by Sravani Bhattacharjee and Rajdeep Chowdhury
3	"Edge Computing: An Introduction to the Next Generation of Networked Systems" by Kiran Chitturi, Bharadwaj Veeravalli, and Satish Narayana Srirama
4	"Building the Web of Things: With examples in Node.js and Raspberry Pi" by Dominique D. Guinard and Vlad M. Trifa
5	"Internet of Things (IoT): Technologies, Applications, Challenges, and Solutions" edited by Balamuralidhar P., Bharadwaj Veeravalli, and V. Raghu
6	"Fog and Edge Computing: Principles and Paradigms" by Rajkumar Buyya, Satish Srirama, and Pradeep Kumar S.
7	"IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes and Gonzalo Salgueiro
8	"Edge Analytics in IoT" by Shivashankar B. Nair, Siddhartha Bhattacharyya, and Thomas Edward Joshua
9	"Edge Computing: The Convergence of Big Data and Internet of Things" by Samee U. Khan, Albert Y. Zomaya, and Salman A. Baset

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Edge Computing (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL621	Edge Computing (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPEL621	Edge Computing (Lab)	-	-	-	-	25		25

Prerequisite: Cloud Computing and IoT concepts	
Lab Objectives: The course aims	
1	To study the concepts of virtual servers and AWS Edge platforms
2	To get acquainted with different Aws IoT Core Services.
3	To learn message passing between AWS IoT Core devices
4	To study different simulators and development of Edge Computing networks
5	To study different simulators and development of FogComputing networks
Lab Outcomes: Students will be able tp	
1	Understand the concept of virtual servers deployed on AWS Edge platform
2	Analyze the use of AWS IoT Core and related services.
3	Examine the message passing techniques in AWS IoT Core devices and assess the security using VPC on AWS.
4	Implement method for message passing between AWS IoT Core devices
5	Deploy Edge computing networks using Edge computing simulators
6	Deploy Fog computing networks using Fog computing simulators

Suggested Experiments	
Sr. No.	**Name of the Experiment**
1	Create and deploy virtual servers on AWS / Azure
2	To deploy Raspberry Pi on AWS IoT Core
3	To implement MQTT messaging between Raspberry Pi and AWS IoT Core
4	To implement virtual private cloud (VPC) on AWS for IoT services.
5	Study assignment on Edge and Fog Simulators
6	Design and deploy an edge computing architecture using edge simulators such as Mobius / EdgeCloudSim.
7	Develop and evaluate edge-based data analytics algorithms in an edge simulator.
8	Design and deploy a fog computing architecture using simulators such as iFogSim / CloudSim.
9	Explore collaboration between edge and fog nodes for IoT applications.

Term Work:	
1	Term work should consist of 8 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: GAME THEORY FOR DATA SCIENCE

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 622	Game Theory for Data Science (Theory)	03	---	---	03	---	---	03
NADPE L622	Game Theory for Data Science (Lab)	---	02	---	---	01	---	01

Game Theory for Data Science (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 622	Game Theory for Data Science (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 622	Game Theory for Data Science (Theory)	20	20	60	02			100

Prerequisite: Basics of Probability Theory	
Course Objectives: The course aims:	
1	To introduce students to foundational concepts of rational decision-making, strategic interaction, and equilibrium in normal- and extensive-form games.
2	To develop the ability to analyze games with perfect and imperfect information, including repeated and Bayesian games.
3	To equip students with techniques to compute equilibrium strategies and apply them in oligopolies, auctions, and algorithmic contexts.
4	To provide insights into mechanism design, social choice theory, and truthful information aggregation mechanisms.

5	To apply game-theoretic models in machine learning, AI systems, and real-world strategic computational settings.
6	To critically evaluate fairness, behavioral deviations, and ethical challenges in strategic decision-making systems.
Course Outcomes: Students will be to	
1	Analyze strategic and extensive-form games and determine pure and mixed Nash equilibria.
2	Solve games with sequential moves, incomplete information, and repeated interactions using advanced equilibrium concepts.
3	Compute Nash equilibria and apply strategic reasoning in economic models and algorithmic game settings.
4	Design mechanisms and apply social choice theory to achieve fairness, efficiency, and incentive compatibility.
5	Model and analyze strategic behavior in multi-agent learning, federated systems, and GANs.
6	Evaluate fairness, rationality limits, and ethical implications using tools from behavioral game theory and cooperative fairness models.

Module	Detailed Content	Hours
1	**Foundations of Game Theory**- Rational Choice and Strategic Thinking, Cooperative vs. Non-Cooperative Games, Normal-form and Extensive-form Representations, Nash Equilibrium (Pure & Mixed), Dominant Strategies, Pareto Efficiency, Zero-sum Games and Minimax Theorem, Illustrative Games: Prisoner's Dilemma, Matching Pennies, Battle of the Sexes	6
2	**Games with Perfect and Imperfect Information**- Extensive-form Games with Perfect Information, Subgame Perfect Equilibrium, Backward Induction, Imperfect Information and Bayesian Games, Sequential Equilibrium and Belief Systems, Repeated Games (finite & infinite), One-Deviation Property and Long-run Payoffs, Bargaining Games	6
3	**Computation of Equilibria & Strategic Behavior**- Computing Nash Equilibria (2-player zero-sum & general-sum), Identifying Dominated Strategies, Algorithmic Approaches to Equilibrium Computation, Strategic Behavior in Economic Models (Cournot & Bertrand), Mixed Strategy Equilibria	6
4	**Mechanism Design & Social Choice**- Mechanism Design Principles: Incentive Compatibility, Truthfulness, Vickrey and VCG Mechanisms (Shortest Paths), Myerson's Lemma, Social Choice Models: Voting, Condorcet Paradox, Existence of Social Welfare Functions, Fair Division: Cake Cutting, Proportionality, Envy-freeness	6
5	**Game Theory in Machine Learning & AI**- Multi-agent Reinforcement Learning (Markov Games), Auction-based Resource Allocation, Federated Learning as Cooperative Game, Stackelberg Security Games, GANs:	6

	Generator vs Discriminator, Collusion Detection in Pricing, K-armed Bandits, Exploration-Exploitation	
6	**Behavioral Game Theory, Ethics & Applications**- Bounded Rationality, Level-k Thinking, Cognitive Hierarchy, Quantal Response Equilibrium, Hyperbolic Discounting, Herding and Emotion Modeling in AI Interaction, Price of Anarchy in Networks, Shapley Value for Attribution, Ethical Dilemmas in AI & Autonomous Systems, Contract Theory & Data Privacy, Applications: Blockchain, Social Networks.	9
TOTAL		39

Textbooks:	
1	An Introduction to Game Theory – Martin J. Osborne
2	Algorithmic Game Theory – Nisan, Roughgarden, Tardos, and Vazirani
3	The Logic of Strategy – Bonanno, G.
4	Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations – Yoav Shoham and Kevin Leyton-Brown

Sr. No.	**Rubrics**	**Marks**
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

Game Theory for DS (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL622	Game Theory for Data Science (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPEL622	Game Theory for Data Science (Lab)	-	-	-	-	25		25

Prerequisite: Basics of Probability Theory, Python Programming	
Lab Objectives:	
1	To model and analyze strategic interactions in normal-form and extensive-form games using payoff matrices and game trees.
2	To simulate games involving uncertainty, including mixed strategies, repeated games, and Bayesian settings.
3	To implement and compare auction mechanisms, including first-price, second-price, and combinatorial auctions.
4	To design and evaluate mechanisms using VCG and social choice rules ensuring efficiency and fairness.
5	To apply game-theoretic approaches to machine learning settings such as K-armed bandits and Stackelberg games.
6	To explore fairness, ethical challenges, and bounded rationality in strategic environments through behavioral models.
Lab Outcomes:	
1	Construct and solve normal-form and extensive-form games using programming tools and equilibrium concepts.
2	Apply mixed and Bayesian strategy frameworks to simulate incomplete and repeated information games.

3	Evaluate auction outcomes and strategic behavior under different bidding mechanisms using simulations.
4	Implement truthful mechanisms and voting protocols, and assess outcomes for efficiency, fairness, and manipulation.
5	Develop and test game-theoretic models in AI/ML settings, including bandit problems and security games.
6	Simulate behavioral deviations from rationality and analyze fairness metrics in strategic decision-making scenarios.

Suggested Experiments Students are required to complete at least 10 experiments.

Sr. No.	Name of the Experiment
1	**Simulating Normal-form Games and Dominant Strategies** Implement a 2-player normal-form game using payoff matrices. Identify strictly/weakly dominant strategies, visualize best responses, and find pure strategy Nash equilibria.
2	**Mixed Strategy Equilibrium in Zero-Sum Games** Simulate games like Rock-Paper-Scissors or Matching Pennies. Compute mixed strategy equilibria using support enumeration or linear programming techniques.
3	**Game Tree Construction and Backward Induction Analysis** Construct an extensive-form game with perfect information. Use backward induction to compute subgame perfect equilibrium and analyze strategy changes under different tree structures.
4	**Bayesian Game Simulation with Belief Updates** Model a Bayesian game with private types. Use player beliefs to compute Bayesian Nash equilibrium and explore how information asymmetry affects strategy.
5	**Auction Simulation: First-Price vs Vickrey** Simulate sealed-bid auctions with multiple bidders. Compare outcomes in first-price and second-price settings, and analyze incentive compatibility.
6	**Cournot and Bertrand Competition Models** Implement Cournot (quantity competition) and Bertrand (price competition) models. Analyze firm profits, consumer surplus, and equilibria in duopoly markets.
7	**VCG Mechanism for Shortest Path Auctions** Simulate a network routing problem where agents bid on edges. Use Vickrey-Clarke-Groves mechanism to compute efficient paths and truthful payments.
8	**Voting Systems and Condorcet Paradox Simulation** Implement Borda, Plurality, and Condorcet voting rules. Analyze social choice outcomes and demonstrate paradoxes through preference profiles.
9	**K-armed Bandit Exploration Strategies** Implement epsilon-greedy and UCB algorithms for K-armed bandits. Track regret over time and compare the effectiveness of different exploration strategies.

10	**Stackelberg Security Game Simulation** Model a leader-follower game between a defender and attacker. Compute optimal defender strategy using commitment, and evaluate attacker response.
11	**Fair Cake-Cutting Protocols** Simulate cake-cutting algorithms such as Divide-and-Choose or Last Diminisher. Evaluate proportionality, envy-freeness, and strategy-proofness of the outcomes.
12	**Modeling Bounded Rationality with Level-k Thinking** Simulate games using level-k reasoning and Quantal Response Equilibrium. Compare outcomes with standard Nash predictions under different rationality levels.

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of a minimum 10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: AI IN FINANCE

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 623	AI in Finance (Theory)	03	---	---	03	---	---	03
NADPE L623	AI in Finance (Lab)	---	02	---	---	01	---	01

AI in Finance (Theory)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPE 623	AI in Finance (Theory)	03	-	-	03	-	-	03

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPE 623	AI in Finance (Theory)	20	20	60	02			100

Prerequisite: Foundational understanding of financial concepts, proficiency in Python programming and machine learning, basic knowledge of statistics and mathematics, and familiarity with financial data and tools.	
Course Objectives: The course aims:	
1	Understand the fundamental concepts, historical evolution, and driving forces behind the adoption of Artificial Intelligence in financial services, along with associated ethical, regulatory, and governance considerations.
2	Develop the ability to identify, preprocess, and engineer structured and unstructured financial data from diverse sources, while ensuring compliance with data privacy and security regulations.

3	Apply supervised, unsupervised, and reinforcement learning techniques to financial forecasting, sentiment analysis, anomaly detection, and evaluate model performance using standard metrics.
4	Design and implement deep learning models—including neural networks and transformers—for tasks such as stock price prediction, robo-advisory, and natural language processing of financial content, while emphasizing model interpretability.
5	Analyze and build AI solutions for real-world financial services applications in banking, investment, insurance, and regulatory technology, including risk scoring, fraud detection, and digital advisory platforms.
6	Evaluate the ethical, governance, and fairness challenges of AI in finance and explore emerging technologies like generative AI and the evolving role of human-AI collaboration.
Course Outcomes: Students will be	
1	Understand the fundamental concepts of Artificial Intelligence and its significance in transforming the financial services industry.
2	Analyze and preprocess various types of financial data, including market, transactional, and alternative datasets, while adhering to data privacy and regulatory compliance standards.
3	Apply AI techniques to analyze financial data and solve problems like forecasting, sentiment analysis, and fraud detection, while effectively evaluating model performance.
4	Develop and implement advanced AI solutions such as neural networks, NLP, and generative models for financial forecasting, advisory, and reporting, with an emphasis on explainability and interpretability.
5	Design and deploy AI-driven solutions for financial services, including credit scoring, trading, robo-advisory, insurance automation, fraud detection, and compliance monitoring.
6	Assess ethical, governance, and fairness challenges in financial AI systems and explore the emerging role of generative AI and human-AI collaboration in the industry.

Module	Detailed Content	Hours
1	**Introduction to AI in Finance** Overview of AI and Machine Learning (ML), Machine learning Categories, Key Algorithms and Role , History and evolution of AI in financial services , Key drivers for AI adoption in finance. Use cases across banking, insurance, asset management, and fintech, Ethical, regulatory, and governance considerations Financial Data Types & Sources	4
2	**Data Foundations in Financial Services** Types of financial data: Structured (market data, transactions) vs. Unstructured (news, reports) **Data sources & Platform**: market data, transaction data, alternative data, data collection and preprocessing and feature engineering, **Data privacy & Governance** , security, and compliance (e.g., GDPR, CCPA)	6

3	**Machine Learning Techniques in Finance** Role of machine learning in finance, **Supervised Learning Algorithms**: Linear Regression, Logistic Regression (Stock prediction) , Decision Trees: Random Forests (Credit Scoring & return prediction) , SVMs, and KNN. Hands-on applications in credit scoring, stock price prediction, etc **Unsupervised Learning Algorithms**: Clustering (Customer Segmentation) Anomaly (Insider Trading) Fraud analytics **Reinforcement Learning** : Algorithmic Trading, Portfolio Management, Market Making. **Deep Learning in Finance**: Neural Networks (FNN, RNN, CNN), Transformers and Attentional Mechanisms.	10
4	**Time-Series Analysis for Financial Forecasting & High-Frequency Trading** **Introduction to Time-Series Data**: Key components: Trend, seasonality, cyclic patterns, noise. **Time-Series Models**: ARIMA, SARIMA, Exponential Smoothing for forecasting. **Advanced Time-Series Forecasting**: LSTM networks for stock price prediction and financial forecasting. **Evaluation and Validation of Time-Series Models**: Metrics like MAE, MSE, RMSE, and MAPE. **Algorithmic & High-Frequency Trading**: Statistical Arbitrage, Market-Making & Execution Strategies, Backtesting & Optimization	10
5	**AI Applications in Banking and Investment, Insurance and Regulatory Tech** Credit scoring and risk assessment, Algorithmic and high-frequency trading, Robo-advisory platforms, Portfolio optimization using ML, Chatbots and virtual financial assistants**,** Underwriting automation and claims processing, Fraud detection in insurance, Predictive analytics for policy pricing and risk modeling, AI for compliance monitoring and reporting, KYC/AML powered by AI	5
6	**Challenges, Trends, and the Future of AI in Finance** Model explainability and interpretability, AI governance and ethical considerations, AI fairness and bias in financial decisions, Generative AI and its emerging role, The future of human-AI collaboration in finance	8
	TOTAL	39

Textbooks:	
1	Machine Learning in Finance by M. Dixon, I Halperin, and P. Bilokon, Springer, 1st Edition
2	Advances in Financial Machine Learning, Marcos Lopez, Wiley, 1st Edition

3	Machine Learning for Algorithmic Trading, Stefan Jansen, 2nd Edition, Packt
4	Reilly, Frank,K., "Investment Analysis and Portfolio Management," 5th Edition, Dryden.

References:	
1	Artificial Intelligence in Finance" – Yves Hilpisch
2	Machine Learning for Asset Managers Marcos López de Prado
3	Machine Learning for Financial Risk Management with PythonAbdullah Karasan
4	Google's TensorFlow Finance tutorials

Journals:	
1	IEEE-T-PAMI (IEEE Transactions on Pattern Analysis and Machine Intelligence).
2	IJCV (International Journal of AI In Finance) - Springer.

Sr. No.	**Rubrics**	**Marks**
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

AI in Finance (Lab)

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADPEL623	AI in Finance (Lab)	-	02	-	-	01	-	01

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADPEL623	AI in Finance (Lab)	-	-	-	-	25		25

Prerequisite: **Need foundational knowledge in finance, statistics, Python programming, machine learning concepts, and experience with tools like Jupyter Notebooks and financial data APIs.**	
Lab Objectives:	
1	Develop skills in cleaning, preprocessing, and transforming financial datasets (structured and unstructured) to make them ready for machine learning applications.
2	Build, train, and evaluate machine learning models such as regression, classification, anomaly detection, and time-series forecasting to solve financial problems like credit scoring, fraud detection, and stock prediction.
3	Implement advanced AI methods including deep learning, sentiment analysis with NLP, and algorithmic trading strategies for real-world financial applications
4	Assess model performance using key metrics and visualizations, and enhance transparency by applying model interpretability tools like SHAP for better understanding and fairness in financial AI systems.
Lab Outcomes:Students will be able to	
1	Clean, preprocess, and handle both structured and unstructured financial data, ensuring it's ready for machine learning models (e.g., handling missing values, normalization, and feature engineering).
2	Implement various machine learning techniques such as regression, classification, anomaly detection, and time-series forecasting to solve financial problems like credit scoring, fraud detection, and stock price prediction.
3	Explore advanced techniques such as deep learning for stock prediction, sentiment analysis using NLP, and algorithmic trading strategies, applying these to real financial datasets.

4	Evaluate the performance of their models using key metrics, visualize results, and gain insights into model interpretability using tools like SHAP, ensuring transparency and fairness in financial AI systems.

Suggested Experiments Students are required to complete at least 10 experiments.	
Sr. No.	**Name of the Experiment**
1	a. Introduction to Financial Datasets and Pandas a. Exploratory Data Analysis (EDA) for Financial Datasets
2	a. Credit Scoring with Logistic Regression a. Predicting Stock Price Movement Using Logistic Regression a. Credit Scoring Using Decision Trees
3	a. Sentiment Analysis on Financial News (NLP) a. Predict market sentiment from financial headlines.
4	a. Feature Engineering on Financial Time Series a. Algorithmic Trading with Time-Series Forecasting
5	Anomaly Detection in Financial Transactions
6	Portfolio Optimization Using Machine Learning
7	Building a Simple Robo-Advisor
8	Automating KYC with Named Entity Recognition
9	Explainable AI in Credit Decisioning
10	Case Study presentation on Banking/ Investment/Insurance working with AI

Note: Suggested List of Experiments is indicative. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Term Work:	
1	Term work should consist of 8-10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 25 Marks (Experiments: 15-marks, Term work Assessment: 10-marks)

COURSE NAME: SECURE SOFTWARE DEVELOPMENT

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADM M61	Secure Software Development (Theory)	02	---	---	02	---	---	02

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADM M61	Secure Software Development (Theory)	02	-	-	02	-	-	02

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADM M61	Secure Software (Theory)	20	20	60	02			100

Prerequisite: Prerequisite:Object Oriented Programming, Cryptography and system security, Ethical Hacking	
Course Objectives: The course aims to:	
1	Understand fundamental concepts and properties in security.
2	Understand various security process models.
3	Identify and prioritize software security requirements.
4	Learn the software security practices.
5	Understand the Secure Software Deployment and Maintenance process.
6	Gain Software Security Knowledge for Architecture and design concepts of Software Security Testing
Course Outcomes: After successful completion of the course, the student will be able to:	
1	Identify the key concepts and properties of security.
2	Summarize the core concepts and principles underlying different security process models.

3	Explain the importance of identifying and prioritizing software security requirements in the software development lifecycle.
4	List common software security practices.
5	Identify common security considerations within each stage of the software deployment and maintenance lifecycle.
6	Explain how software architecture and design choices impact the feasibility and effectiveness of different security testing methods.

Module	Detailed Content	Hours
1	**Importance of software security** Software assurance definition, Properties of secure software, Secure Design Principles. Software vulnerabilities: Zero days, Design, Implementation, and Environment vulnerabilities, CVE, CVSS, NVD Causes of vulnerabilities, Seven pernicious kingdoms	5
2	**Security in the software lifecycle** Software Development Life Cycle (SDLC) models, Maturity models, BSIMM, OpenSAMM, NIST SSDF	5
3	**Software security requirements** The Importance of Requirements Engineering, Quality Requirements, Security Requirements Engineering. Misuse and Abuse Cases: Security Is Not a Set of Features, Thinking About What You Can't Do, Creating Useful Misuse Cases, An Abuse Case Example. The SQUARE Process Model: A Brief Description of SQUARE, Tools, Expected results. SQUARE, Sample Outputs: Output from SQUARE Steps, SQUARE Final Results.	4
4	**Secure Coding Practices** Static analysis, Defensive coding, Secure coding standard Ethics in software security development and Consequences of Ethical Failures. Industry standards - GDPR, HIPAA, PCI DSS, and ISO 27001.	4
5	**Secure Software Deployment and Maintenance** Secure software deployment: Risk and Best practices, secure software maintenance.	4
6	**Secure Software Testing** Fuzzing, Contrasting Software Testing and Software Security Testing,	4

	Functional Testing, Risk-Based Testing. Secure API development -OpenID connect, API gateway security.	
TOTAL		26

Textbooks:	
1	Software Security Engineering A Guide for Project Managers by Julia H.Allen, ean J. Barnum, Robert J. Ellison and Gary McGraw, May 11, 2008
2	John Musa D, "Software Reliability Engineering", 2nd Edition, Tata McGraw-Hill, 2005
3	"Cyber Security and Laws",by Madhumati Chatterjee, Sangita Chaudhary, Gaurav Sharma, Staredu solutions, edition 2019.
4	Gary McGraw. Software Security: Building Security In. Addison-Wesley Professional; 1 edition (February 2, 2006)

References:	
1	Software Security Engineering: A Guide for Project Managers by Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead, Addison Wesley Publishers.
2	Cyber Security Engineering: A Practical Approach for Systems and Software Assurance by Nancy R. Mead and Carol Woody, Addison Wesley Professional.
3	Michael Howard, David LeBlanc & John Viega . 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them (Networking & Comm - OMG). McGraw-Hill Education; 1 edition (September 24, 2009)
4	Ross Anderson. Security Engineering. Wiley. 2nd edition. (https://www.cl.cam.ac.uk/~rja14/book.html)

Journals:	
1	http://study.com/articles/List_of_Free_Online_Software_Engineering_Courses.html
2	https://www.coursera.org/course/softwaresec
3	Mathias Paye. Software Security Principles, Policies, and Protection. (January 2019, v0.33) (https://nebelwelt.net/SS3P/softsec.pdf)
4	Secure Coding Practice Guideline:

	https://security.berkeley.edu/secure-coding-practice-guidelines
5	Seed Labs: https://seedsecuritylabs.org/

Sr. No.	Rubrics	Marks
1	*Certificate course for 4 weeks or more: NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Content beyond syllabus presentation	10 marks
3	Creating Proof of concept	10 marks
4	Mini Project / Extra Experiments/ Virtual Lab / Competitive programming- based event / Group Discussion	10 marks
5	Multiple Choice Questions (Quiz)	5 marks
6	GATE Based Assignment/Tutorials etc.	10 marks

*For sr. no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification, the grading has to be done accordingly.

End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will have a total of five questions
3	All questions have equal weightage and carry 20 marks each
4	Any three questions out of five need to be solved.

COURSE NAME: AWS ESSENTIAL/ AZURE

Course Code	Course Name	Teaching Scheme (Teaching Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	TW/PR	Tut	Total
NADVS61	AWS Essential /Azure (Lab)		02	-	-	02	-	02

Course Code	Course Name	Examination Scheme						
		Theory			Exam Duration (in Hrs)	Term Work	Practical & Oral	Total
		Internal Assessment		End Sem Exam				
		Mid-Term Test	Continuous Assessment					
NADVS61	AWS Essential /Azure (Lab)	-	-	-	-	25	25	50

Note: Suggested Objectives, Outcomes and List of Experiments is indicative and is based on Amazon Web Services. However, flexibility lie with individual course instructors to design and introduce new, innovative and challenging experiments based on Azure / GCP or a similar MOOC course may be offered.

Prerequisite:	
Lab Objectives: By the end of this course, learners will be able to:	
1	Navigate AWS Core Services – Use AWS CloudShell, S3, DynamoDB, and SDKs to interact with AWS programmatically.
2	Develop Serverless Applications – Build REST APIs with API Gateway and Lambda functions using Python.
3	Containerize and Deploy Applications – Migrate web apps to Docker and run them on managed services (e.g., ECS/EKS).
4	Optimize Performance & Security – Implement caching (ElastiCache, CloudFront) and secure apps with Cognito.
5	Design Event-Driven Architectures – Use SNS, SQS, and Step Functions for messaging and workflow orchestration.
6	Apply AWS Best Practices – Follow AWS Well-Architected principles for scalability, cost-efficiency, and reliability.
Lab Outcomes: After completing the experiments, learners will:	
1	Demonstrate the ability to use AWS CLI, SDKs, and consoles for real-world tasks.
2	Build and deploy serverless APIs using Lambda, API Gateway, and Python.

3	Dockerize applications and deploy them on AWS managed services.
4	Configure ElastiCache and CloudFront to improve application performance.
5	Architect solutions using SNS, SQS, and Step Functions for decoupled systems.
6	Secure applications using Cognito for authentication and authorization.

Suggested Experiments	
Sr. No.	Name of the Experiment
1	Exploring AWS CloudShell and an IDE
2	Working with Amazon S3
3	Working with DynamoDB
4	Developing REST APIs with API Gateway
5	Creating Lambda Functions Using the AWS SDK for Python
6	Creating Lambda Functions Using the AWS SDK for Python
7	Migrating a Web Application to Docker Containers
8	Running Containers on a Managed Service
9	a) Caching Application Data with ElastiCache b) Implementing CloudFront for Caching and Application Security
10	Implementing a Messaging System Using Amazon SNS and Amazon SQS
11	Orchestrating Serverless Functions with Step Functions
12	Implementing Application Authentication Using Amazon Cognito
13	Automating Application Deployment Using a CI/CD Pipeline

Term Work:	
1	Term work should consist of a minimum 10 experiments.
2	The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work.
3	Total 50 Marks for Term Work based on the completion of the course assigned by the instructor..