

BANGALORE UNIVERSITY
ಬ'ೆಂಗಳೂರು ವಿಶ್ವವಿದ್ಯಾಲಯ
Jnanabharathi, Bengaluru, Karnataka – 560 056

SCHEME AND SYLLABUS

For the Course

BACHELOR OF COMPUTER APPLICATIONS
(Artificial Intelligence & Machine Learning/ Data Science/ Full Stack Development)

State Education Policy 2024 SCHEME

Academic Year 2026- 2027 and onwards

Members of the BoS in Computer Science

Sl. No.	Name	Designation
1.	Dr. Muralidhara B. L Coordinator BCA & Senior Professor, Dept. of Computer Science and Applications, Bangalore University, Bangalore -560056.	Chairperson
2.	Mr. Chitty Babu G Consultant, 112, 10th Cross, 29th Main, JP Nagar I Phase, Bangalore - 560078	Member
3.	Dr. Suresh M Associate Professor, Dept. of Computer Science and Applications, Kuvempu University, Jnanasahyadri, Shankaraghatta, Shivamogga Dist.-577451.	Member
4.	Dr. Rajesh B Assistant Professor Department of Management Studies, Mahindra University, Survey no. 62/1A, BahaddurPally, Jeedimetla, Hyderabad, Telangana-500043.	Member
5.	Dr. Nazura Javed Kutty Associate Professor, Dept. of Computer Science and Applications, St. Francis College, 3rd Block, 8th Main, Koramangala 560034	Member
6.	Smt. Radhika E K, Academic Co-ordinator & HOD, Dept. of Computer Science and Applications, Sindhi College, #33/2B, Hebbal, Kempapura, Bengaluru-560024.	Member
7.	Dr. C. Umarani Associate professor, Christ Academy Institute for Advanced studies, Begur Koppa Road, Hullahalli, Sakalavara, Bangalore, 560083	Member
8.	Mr. Hemanth Uppala Assistant Professor, Dayananda Sagar College of Arts, Science and Commerce, Kumaraswamy Layout, Bengaluru 560078.	Member
9.	Mr. Dadavali S.P Associate Professor, Dept. of Computer Science and Applications, Government First Grade College, Kengeri Satellite Town, Kengeri, Bengaluru 560060.	Member
10.	Smt. B. Nazia Hassan Associate Professor, Dept. of Computer Science and Applications, Government First Grade College, Vijayanagar, RPC Layout, Bengaluru-560040.	Member

BANGALORE UNIVERSITY

Proceedings of the meeting of the Board of Studies in Computer Science (UG), held on 15-04-2026 in the Board Room, Department of Chemistry, Bangalore University, Bangalore at 11.00 am

The Following Members were present:

1. Dr. Muralidhara B L, Coordinator, BCA, and Senior Professor, Department of Computer Science and Applications, Bangalore University, Bangalore – 560056 - Chairperson
2. Dr. Chitty Babu G, Consultant, 112, 10th Cross, 29th Main, JP Nagar I Phase, Bangalore – 560078 - Member
3. Dr.Nazura Javed Kutty, Associate Professor, St. Francis College, 3rd Block, 8th Main, Koramangala- 560034 - Member
4. Smt. Radhika E K, Academic Co-ordinator & HoD, Dept. of Computer Science and Applications, Sindhi College, #33/2B, Hebbal, Kempapura, Bengaluru-560024 - Member
5. Dr.C.Umarani, Associate professor, Christ Academy Institute for Advanced studies, Begur Koppa Road, Hullahalli, Sakalavara, Bangalore, 560083 - Member
6. Mr.Hemanth Uppala, Asst. Prof. & Vice-Principal, Dayananda Sagar College of Arts, Science and Commerce, Shavige Malkeswara Hills, Kumaraswamy Layout, Bengaluru 560078. - Member
7. Smt. B. Nazia Hassan, Associate Professor, Dept. of Computer Science and Applications, Government First Grade College, Vijayanagar, RPC Layout, Bengaluru-560040. - Member

The Following Members were participated through online:

1. Dr. Suresh M, Associate Professor, Dept. of Computer Science and Applications, Kuvempu University, Jnanasahyadri, Shankaraghatta, Shivamogga Dist.-577451. - Member
2. Dr.Rajesh B, Assistant Professor, Department of Management Studies, Mahindra University, Survey no. 62/1A, BahaddurPally, Jeedimetla, Hyderabad, Telangana-500043 - Member

The Following Member was absent:

1. Mr. Dadavali S P, Associate Professor, Department of Computer Science and Applications, Government First Grade College, Kengeri Satellite Town , Kengeri, Bengaluru 560060. - Member

The Chairperson welcomed the members.

The members of the Board of Studies thanked all the incumbent Board of Studies members for their contribution for framing the previous scheme syllabus.

The Chairperson briefed the members on the proposed the SEP syllabus for BCA (Artificial Intelligence and Machine Learning), BCA (Data Science), BCA (Full Stack Development), and BSc., (Computer Science). The Committee discussed the syllabus in detail and approved the structure of the following syllabus, and also approved the detailed syllabus:

1. Syllabus for the V and VI semester of BCA (Artificial Intelligence and Machine Learning)
2. Syllabus for the V and VI semester of BCA (Data Science)
3. Syllabus for the V and VI semester of BCA (Full Stack Development)
4. Syllabus for the V and VI semester of BSc., (Computer Science)

The Chairperson thanked all the members for their active participation.

(Dr. Chitty Babu G)

(Dr. Nazura Javed Kutty)

(Ms. Radhika E K)

(Dr.C.Umarani)

(Mr. Hemanth Uppala)

(Ms. B. Nazia Hassan)

(Dr. Muralidhara B L)
Chairperson

BANGALORE UNIVERSITY

Department of Computer Science and Applications

Draft for BCA Course Structure

BCA (with specialization in AI & ML/ Data Science/ Full Stack Development)

AS PER STATE EDUCATION POLICY

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
1	24BCA11	Discrete Structure	03	80	20	100	3
	24BCA12	Problem Solving Technique	03	80	20	100	3
	24BCA13	Computer Architecture	03	80	20	100	3
	24BCA12P	Problem Solving Technique Lab	04	40	10	50	2
	24BCA13P	Computer Architecture Lab	04	40	10	50	2
	24BCA1P	Office Automation Tools	04	40	10	50	2
	24BCAL11	Language L1	04	80	20	100	3
	24BCAL12	Language L2	04	80	20	100	3
	24BCACC1	Environmental Studies	02	40	10	50	2
2	24BCA21	Data Structure	03	80	20	100	3
	24BCA22	Object Oriented Programming Using JAVA	03	80	20	100	3
	24BCA23	Operating Systems	05	80	20	100	5
	24BCA21P	Data Structure Lab	04	40	10	50	2
	24BCA22P	Operating Systems Lab	04	40	10	50	2
	24BCA21P	LINUX and Shell Programming Lab	04	40	10	50	2
	24BCAL21	Language L1	04	80	20	100	3
	24BCAL22	Language L2	04	80	20	100	3
	24BCASEC1	Computer Applications	04	40	10	50	2

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours	Semester End Exam	Internal Assessment	Total Marks	Credits
3	24BCA31	Database Management System	03	80	20	100	3
	24BCA32	Probability and Statistics	03	80	20	100	3
	24BCA33	Artificial Intelligence	04	80	20	100	4
	24BCA31P	Database Management SystemLab	04	40	10	50	2
	24BCA32P	Artificial Intelligence Lab using Python	04	40	10	50	2
	24BCAE1	Elective: I Feature Engineering (AI &ML) Basics of Data Analytics using Spreadsheet (Data Science) Web Programming-I (Full Stack Development)	02	40	10	50	2
	24BCAL31	Language L1	04	80	20	100	3
	24BCAL32	Language L2	04	80	20	100	3
	24BCACC2	Constitution Value 1	02	40	10	50	2
4	24BCA41	Computer Networks	03	80	20	100	3
	24BCA42	Design and Analysis of Algorithms	03	80	20	100	3
	24BCA43	Software Engineering	04	80	20	100	4
	24BCA41P	Computer Networks Lab	04	40	10	50	2
	24BCA42P	Design and Analysis of Algorithms Lab	04	40	10	50	2
	24BCAE2	Elective: II Introduction to ML(AI &ML) Data Visualization (Data Science) Web Programming-II (Full Stack Development)	02	40	10	50	2
	24BCAL41	Language L1	04	80	20	100	3
	24BCAL42	Language L2	04	80	20	100	3
	24BCASEC2	Constitution Value 2	04	40	10	50	2

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V(AI & ML)							
5	24BCA51	ML & Neural Network	04	80	20	100	4
	24BCA52	Digital Image Processing	04	80	20	100	4
	24BCA53	Natural Language Processing	03	80	20	100	3
	24BCA54	Internet of Things	03	80	20	100	3
	24BCA55	Cyber Security	03	80	20	100	3
	24BCA51P	ML & Neural Network Lab	04	40	10	50	2
	24BCA52P	Digital Image Processing Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI (AI & ML)							
6	24BCA61	Deep Learning for Computer Vision	04	80	20	100	4
	24BCA62	Predictive Analysis	04	80	20	100	4
	24BCA63	Research Methodology	03	80	20	100	3
	24BCA64	Generative AI	03	80	20	100	3
	24BCA65	Project Work	08	80	20	100	4
	24BCA66	Internship	04	40	10	50	2
	24BCASEC3	Soft Skills	03	80	20	100	3
Semester – V (Data Science)							
5	24BCA51	Introduction to Data Science	04	80	20	100	4
	24BCA52	Time Series Analysis	04	80	20	100	4
	24BCA53	Machine Learning	03	80	20	100	3
	24BCA54	Internet of Things	03	80	20	100	3
	24BCA55	Cyber Security	03	80	20	100	3
	24BCA51P	Data Science Lab	04	40	10	50	2
	24BCA53P	Machine Learning Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI (Data Science)							
6	24BCA61	Big Data Analytics	04	80	20	100	4
	24BCA62	Exploratory Data Analysis	04	80	20	100	4
	24BCA63	Research Methodology	03	80	20	100	3
	24BCA64	Generative AI	03	80	20	100	3
	24BCA65	Project Work	08	80	20	100	4
	24BCA66	Internship	04	40	10	50	2
	24BCASEC3	Soft Skills	03	80	20	100	3

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V(Full Stack Development)							
5	24BCA51	Frontend Design	04	80	20	100	4
	24BCA52	Backend Development	04	80	20	100	4
	24BCA53	Cloud computing	03	80	20	100	3
	24BCA54	Internet of Things	03	80	20	100	3
	24BCA55	Cyber Security	03	80	20	100	3
	24BCA51P	Frontend Design Lab	04	40	10	50	2
	24BCA52P	Backend Development Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2
Semester – VI (Full Stack Development)							
6	24BCA61	Software Project Management	04	80	20	100	4
	24BCA62	Mobile Application Development	04	80	20	100	4
	24BCA63	Research Methodology	03	80	20	100	3
	24BCA64	Generative AI	03	80	20	100	3
	24BCA65	Project Work	08	80	20	100	4
	24BCA66	Internship	04	40	10	50	2
	24BCASEC3	Soft Skills	03	80	20	100	3

Project Work

Note:

1. The project shall be undertaken individually by each student.
2. The project must be relevant to the student's area of specialization.
3. Students are permitted to use any suitable tools or technologies for project implementation.
4. The project work shall be considered as part of the workload calculation.

Department of Computer Science and Applications

BANGALORE UNIVERSITY, BANGALORE

Program Outcome

PO1	Computational Knowledge	Acquire in-depth computational and mathematical knowledge with an ability to abstract and conceptualise from defined problems and requirements.
PO2	Dynamic Problem-Solving Skill	Identify, formulate, and exhibit strong analytical and dynamic problem-solving skills to address evolving computational challenges.
PO3	Innovative System Analysis and Design/ Development	Design and evaluate solutions for complex problems in Data Science, AI & ML, and Full Stack Development, considering societal, cultural, and environmental factors.
PO4	Investigate complex computing problem	Conduct literature surveys, analyze information, and design experiments using appropriate research methods to derive valid conclusions in relevant domains.
PO5	Use of modern tools/ Adaptive programming proficiency	Select, adapt, and apply modern IT tools and programming languages effectively in Data Science, AI & ML, and Full Stack Development to solve diverse computing challenges.
PO6	Knowledge Optimization	Modify algorithms or software systems to improve efficiency or resource utilization.
PO7	Life Long Continuous learning and Technology Adaptability	Pursue lifelong learning to stay updated with emerging technologies in Data Science, AI & ML, and Full Stack Development for sustained employability.
PO8	Soft skills and collaborative teamwork	Communicate effectively, enhance interpersonal skills, and collaborate in multidisciplinary teams essential for success in professional environments.
PO9	Cyber Security Proficiency	Understand cyber threats, develop secure software, and protect sensitive data in Data Science, AI & ML, and Full Stack Development applications.
PO10	Ethical and Professional Conduct	Adhere to ethical standards and professional practices in Data Science, AI & ML, and Full Stack Development roles and responsibilities.
PO11	Employability	Identify market trends, upgrade skills accordingly, and enhance employability in Data Science, AI & ML, and Full Stack Development careers.
PO12	Innovation and Entrepreneurship	Identify opportunities, innovate, and create value through Data Science, AI & ML, and Full Stack Development projects for personal growth and societal impact.

Specialization: AI & ML
SEMESTER –V

	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V(AI & ML)							
5	24BCA51	ML & Neural Network	04	80	20	100	4
	24BCA52	Digital Image Processing	04	80	20	100	4
	24BCA53	Natural Language Processing	03	80	20	100	3
	24BCA54	Internet of Things	03	80	20	100	3
	24BCA55	Cyber Security	03	80	20	100	3
	24BCA51P	ML & Neural Network Lab	04	40	10	50	2
	24BCA52P	Digital Image Processing Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2

24BCA51	ML & Neural Network	60 Hours

Course Objectives

CO1: To understand the fundamentals of Machine Learning and Neural Networks.
CO2: To learn various algorithms for data analysis, classification, and prediction.
CO3: To develop skills in building and evaluating machine learning models.
CO4: To explore neural network architectures and deep learning techniques.

Prerequisites:

1. Basic knowledge of mathematics, including linear algebra, probability, and statistics.
2. Understanding of programming concepts (preferably in Python).
3. Familiarity with basic concepts of data structures and algorithms.

Course content:

Unit-I: Introduction to Machine Learning: [15 hours]

Introduction. Different types of machine learning, Supervised Learning: Regression vs. Classification, Linear Regression, Logistic Regression, Decision Trees and Random Forests. Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Ensemble Methods and Model Evaluation Metrics.

Unit-II: Unsupervised Learning: [15 hours]

Clustering Algorithms (K-Means, Hierarchical Clustering), Principal Component Analysis (PCA), Dimensionality Reduction Techniques, Association Rule Mining, Anomaly Detection, Evaluation of Unsupervised Models.

Unit-III: Artificial Neural Networks (ANN): [15 hours]
ANN in machine learning. Perceptron (linear) Model and Perceptron Learning Algorithm. Overview of Neural Networks, Biological Neurons vs. Artificial Neurons, McCulloch-Pitts Neuron. Activation Functions: Signum, Sigmoid, Tanh, Softmax, ReLU. Learning Error: Loss Functions. Error Minimization.

Unit-IV: Single-Layer Perceptron (SLP) and Multilayer Perceptrons (MLP): [15 hours]
Architecture, applications (AND, OR, and linearly separable data), weight update, Backpropagation Algorithm: Gradient Descent Method, Batch, Stochastic, Mini Batch. Issues with Local Minima, learning rate. Limitation in handling non-linearity (XOR problem). Multilayer Perceptrons (MLP): Why Multilayer Perceptrons (MLP)? Architecture of MLP, Backpropagation in MLP, Application of SLP and MLP in solving classification and estimation problems. LLM, Transformers, Prompt Engineering.

Text Books:

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Hiran, K. K., Doshi, R., Jain, R. K., & Lakhwani, K. (2021). *Machine Learning*. BPB Publications.
3. Ethem Alpaydin. Introduction to machine learning. MIT press, 2020.
4. Haykin, S. (1998). Neural networks: A comprehensive foundation (2nd ed.). Prentice Hall.
5. Goodfellow, I. J., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Reference Books:

1. Kalita, J. K., Bhattacharyya, D. K., & Roy, S. (2023). Fundamentals of data science: Theory and practice. Elsevier. https://doi.org/10.1016/B978-0-323-91778-0
2. Mitchell, T. M. (2017). Machine learning. McGraw-Hill.
3. Aggarwal, C. C. (2023). Neural networks and deep learning: A textbook (2nd ed.). Springer.

Web Resources
1. http://neuralnetworksanddeeplearning.com/
2. https://brilliant.org/courses/intro-neural-networks/

Course Outcomes

COU1: Apply machine learning techniques to solve real-world problems.
COU2: Build and evaluate predictive models effectively.
COU3: Design basic neural network models for classification and estimation tasks.
COU4: Analyze and improve model performance using appropriate techniques.

****

24BCA52	Digital Image Processing	60 Hours

Course Objectives

CO1: Understand the fundamental concepts of Image Processing and its application in various domains.
CO2: Understand and implement various image processing operations for image enhancement/filtering, image restoration, image compression, image segmentation, and morphological image processing.
CO3: Understanding various image transformations and their properties.

CO4: Create an image processing application using Python language.

Prerequisites:

1. Programming Using Python
2. Mathematics for Computing

Course Content

UNIT I: Introduction to Image Processing: [15 hours]

Introduction to Image Processing: Fundamentals: Digital Image Representation, Fundamental Steps in Image Processing, Elements of Digital image processing systems, Sampling and quantization, some basic relationships like neighbours, connectivity, Distance measure between pixels, Imaging Geometry

UNIT II: Image Enhancement: [15 hours]

Image Enhancement: Some Basic spatial and frequency domain transformations: Histogram Equalization, Spatial Correlation and Convolution, Smoothening Spatial Filters: Low pass filters, Sharpening Spatial Filters: Laplacian filter, The Discrete Fourier Transformation (DFT), Frequency Domain Filtering: Ideal and Butterworth Low pass and High pass filters, DCT Transform (1D, 2D).

UNIT III: Image Degradation/Restoration: [15 hours]

Image Degradation/Restoration: Image Degradation/Restoration Process, Noise models, Noise Restoration Filters. Image Compression: Fundamentals of Image Compression, Huffman Coding, Run Length Coding, JPEG.

UNIT IV: Morphological Image Processing: [15 hours]

Morphological Image Processing: Dilation, Erosion, opening and closing, hit and miss transform, boundary extraction. Image Segmentation: Point, Line and Edge Detection, Thresholding, Region Based Segmentation.

Text books:

1. Gonzalez, R. C., & Woods, R. E. Digital Image Processing. 4th edition. Pearson Education, 2017.

Reference Books:

1. Castleman, K. R. Digital Image Processing. 1st edition. Pearson Education,2007
2. Jain, A. K. Fundamentals of Digital Image Processing. 1st edition Prentice Hall of India, 1988.
3. William K. Pratt, Digital Image Processing, John Wiley.

****

24BCA53	Natural Language Processing	45 Hours

Course Objectives:

CO1: Understand NLP fundamentals and preprocessing techniques
CO2: Apply text representation and syntactic analysis methods
CO3: Implement ML and Transformer-based NLP models
CO4: Insights into development of NLP applications and evaluate their performance

UNIT I : Foundations & Text Preprocessing [12 hours]

Introduction to NLP: scope, challenges, applications, Levels of language analysis: morphology, syntax, semantics, pragmatics, Overview of NLP pipeline. Text Preprocessing**:** Tokenization, normalization, Stop-word removal, Stemming and lemmatization, Handling noisy text. Introduction to Regular Expressions and Finite Automata, Introduction to NLP tools and libraries: NLTK, spaCy, Stanford CoreNLP.

UNIT II: Text Representation & Syntactic Analysis [11 hours]

Text Representation: Bag of Words (BoW), TF-IDF, N-grams, Word embeddings (Word2Vec, GloVe). Language Modelling: Statistical language models, N-gram models. Part-of-Speech (POS) Tagging. Syntactic Analysis: Context-Free Grammars (CFG), Parsing (Top-down, Bottom-up), Probabilistic parsing

UNIT III: Semantics and Advanced NLP [11 hours]

Semantic analysis: Meaning representation, Word sense disambiguation, Named Entity Recognition (NER). Machine Learning for NLP: Naïve Bayes, Logistic Regression, Introduction to sequence models: RNN, LSTM, GRU. Transformers for Text: Limitations of RNNs, Attention mechanism, Transformer architecture, Pretrained models: BERT, GPT.

UNIT IV: Advanced Topics & Applications [11 hours]

Discourse analysis and pragmatics, Coreference resolution, Natural Language Generation (NLG), NLP Applications (case study): Machine Translation, Sentiment Analysis, Information Extraction, Question Answering Systems, Text Summarization. Evaluation metrics: Precision, Recall, F1-score, BLEU score

Textbooks:

1. Jurafsky, D., & Martin, J. H. (2009). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd ed.). Pearson Prentice Hall.
2. Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: Analyzing text with the Natural Language Toolkit. O'Reilly Media.
3. Bhattacharyya, P., & Joshi, A. (2023). Natural language processing. Wiley India.

References:

1. Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT Press.
2. Vajjala, S., Majumder, B., Gupta, A., & Surana, H. (2020). *Practical natural language processing: A comprehensive guide to building real-world NLP systems*. O'Reilly Media.

Web Resources:

Stanford NLP: https://nlp.stanford.edu

Stanford Course: https://web.stanford.edu/class/cs224n/

Hugging Face (Transformers) https://huggingface.co
NLTK: https://www.nltk.org
spaCy: https://spacy.io

****

24BCA51P	ML & Neural Network Lab	60 Hours

List of Programs:

1. Perform data preprocessing on a healthcare dataset by handling missing values, encoding categorical variables, and applying normalization techniques. Analyze the dataset using summary statistics and visualizations.
   Dataset: Pima Indians Diabetes Dataset https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database Tools Used: Python, Pandas, NumPy, Matplotlib

2. Develop a Linear Regression model to predict the fuel efficiency (miles per gallon) of automobiles based on various features and evaluate the model using MSE and R^2 score.
   Dataset: Auto MPG Dataset https://archive.ics.uci.edu/ml/datasets/auto+mpg Tools Used: Python, Scikit-learn, Pandas

3. Build a Logistic Regression model to predict whether a tumor is malignant or benign and evaluate the model using a confusion matrix and accuracy score.
   Dataset: Breast Cancer Wisconsin Dataset https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic) Tools Used: Python, Scikit-learn, Pandas

4. Implement the KNN algorithm to classify different types of glass based on their chemical properties and analyze the effect of varying K values on model performance.
   Dataset: Glass Identification Dataset https://archive.ics.uci.edu/ml/datasets/glass+identification Tools Used: Python, Scikit-learn

5. Construct a Decision Tree classifier to predict whether a bank customer will subscribe to a term deposit and evaluate the model using accuracy and precision.
   Dataset: Bank Marketing Dataset https://archive.ics.uci.edu/ml/datasets/bank+marketing Tools Used: Python, Scikit-learn, Matplotlib.

6. Build and train a basic perceptron for binary classification including simple AND, OR functions.

7. Implement and visualize different activation functions: Step, Sigmoid, Tanh, and ReLU.
8. Create a single-layer perceptron to classify a simple dataset (e.g., XOR problem).
9. Use a single-layer perceptron to visualize decision boundaries on 2D data.
10. Build a basic Multilayer Perceptron with one hidden layer to solve a classification problem.
11. Implement the backpropagation algorithm and train an MLP on a small dataset (e.g., Iris dataset).
12. Experiment with different gradient descent optimizers (e.g., SGD, SGD with momentum) to train an MLP.
13. Use MLP to analyze and visualize the changes in training and test errors with different model complexities.
14. Build a simple CNN for image classification.
15. Use appropriate prompt for generating notes for MLP.
16. Summarize the research paper using appropriate prompt.

****

24BCA52P	Digital Image Processing Lab	60 Hours

List of Programs:

1. Write program to read and display digital image using Python
   a. Become familiar with Basic image processing commands and libraries in Python
   b. Read and display image in Python
   c. Resize given image
   d. Convert given color image into gray-scale image
   e. Convert given color/gray-scale image into black & white image
   f. Draw image profile
   g. Separate color image in three R G & B planes
   h. Create color image using R, G and B three separate planes
   i. Write given 2-D data in image file
2. To write and execute image processing programs using point processing method
   a. Obtain Negative image
   b. Obtain Flip image

c. Thresholding

d. Contrast stretching

3. To write and execute programs for image arithmetic operations

a. Addition of two images

b. Subtract one image from another image

c. Calculate mean value of image

4. To write and execute programs for image logical operations

a. AND operation between two images

b. OR operation between two images

c. Calculate intersection of two images. NOT operation (Negative image)

5. To write a program for histogram calculation and equalization using

a. Standard function

b. Program without using standard functions

6. To write and execute program for geometric transformation of image

a. Translation

b. Scaling

c. Rotation

d. Shrinking

e. Zooming

7. To understand various image noise models and to write programs for

a. Image restoration

b. Remove Salt and Pepper Noise

c. Minimize Gaussian noise

d. Median filter

8. Write and execute programs to use spatial low pass and high pass filters

9. Write and execute programs for image frequency domain filtering

a. Apply FFT on given image

b. Perform low pass and high pass filtering in frequency domain

c. Apply IFFT to reconstruct image

10. Write a program for edge detection using different edge detection masks.

11. Write and execute program for image morphological operations erosion and dilation.

****

Specialization: AI & ML
SEMESTER –VI

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – VI (AI & ML)							
6	24BCA61	Deep Learning for Computer Vision	04	80	20	100	4
	24BCA62	Predictive Analysis	04	80	20	100	4
	24BCA63	Research Methodology	03	80	20	100	3
	24BCA64	Generative AI	03	80	20	100	3
	24BCA65	Project Work	08	80	20	100	4
	24BCA66	Internship	04	40	10	50	2
	24BCASEC3	Soft Skills	03	80	20	100	3

24BCA61	Deep Learning for Computer Vision	60 Hours

Course Objectives:

CO1: Understand the history, evolution, and key applications of Computer Vision in various domains, including healthcare, automotive, and robotics.
CO2: Explain the basics and general architecture of Convolutional Neural Networks (CNNs), and describe the functionality of ConvNet layers such as convolutional, pooling, and normalization layers.
CO3: Demonstrate the ability to implement a basic CNN for image classification, including the conversion of fully-connected layers to convolutional layers.
CO4: Analyze and apply advanced CNN architectures, including VGG, ResNet, and Inception, and utilize transfer learning and fine-tuning techniques for pre-trained models.
CO5: Explore and implement Generative Adversarial Networks (GANs), understanding their core concepts, architecture, and key applications, such as image generation and style transfer.
CO6: Introductory idea of 3D point cloud data analysis

Prerequisite: A solid foundation in neural networks, linear algebra, calculus, probability, statistics, basic programming concept in Python.

Course Content:

UNIT I: Introduction to Computer Vision: **[15 hours]**
Introduction to Computer Vision: History and Evolution of Computer Vision, Key Applications in Various Domains (e.g., Healthcare, Automotive, Robotics, Security, Entertainment,

Agriculture, and Manufacturing, etc.) Key activities in Computer Vision: Image Classification, Object Detection, Segmentation, Semantic Segmentation, 3D vision, and point clouds.

UNIT II: Convolutional Neural Networks (CNNs): [15 hours]

Convolutional Neural Networks (CNNs): Basics of CNNs, role in Computer Vision, General Architecture of CNNs. Convolutional Layer: Concept and functionality of convolutional layers, including filters, kernels, stride, padding, feature maps, and receptive fields for capturing spatial hierarchies in images. Pooling Layer: Types and purpose of pooling layers, such as max, average, and global pooling, downsampling, pool size, and stride. Normalization Layer: Role in stabilizing the learning process, including techniques like Batch Normalization, Layer Normalization, Group Normalization, mean-shifting, variance-scaling, and activation scaling. Fully-Connected Layer: parameter sharing, dense connections, weight sharing, flattening, dropout, and activation functions.

UNIT III: Advanced CNN Architectures: [15 hours]

Advanced CNN Architectures: Overview of VGG and ResNet, Concepts of transfer learning, fine-tuning layers to adapt to new tasks. Improving CNN Performance: data augmentation (e.g., rotations, flips, zooms), regularization (e.g., dropout, weight decay), and hyperparameter tuning Object detection algorithm. RCNN, Fast RCNN, and YOLO Image Segmentation using U-Net. 3D point clouds: Overview, Object detection and Semantic segmentation in 3D point clouds.

UNIT IV: Generative Adversarial Networks (GANs): [15 hours]

Core Concepts and Training Process of GANs, Understanding the Architecture: Generator and Discriminator Networks, Key Applications of GANs: Image Generation, Style Transfer, Data Augmentation, and Creating Synthetic Data.

Text Books

1. Szeliski, R. Computer Vision: Algorithms and Applications. Springer, 2011.
2. Chollet, F. Deep Learning with Python (2nd ed.). Manning Publications, 2021.

Reference Books:

1. Forsyth, D. A., & Ponce, J. Computer Vision: A Modern Approach (2nd ed.). Pearson Education, 2015.
2. Fisher, R. B., Breckon, T. P., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., Trucco, E., & Williams, C. K. I. Dictionary of Computer Vision and Image Processing (2nd ed.). Wiley, 2014.
3. Goodfellow, I. J., Bengio, Y., & Courville, A. Deep Learning. MIT Press, 2016.

Web Resources:

1. https://cs231n.stanford.edu/
2. https://www.coursera.org/specializations/deep-learning-computer-vision

****

24BCA62	Predictive Analysis	60 Hours

Prerequisite

Statistics and Probability, Data Analytics, Python Programming, Machine Learning Fundamentals. Familiarity with data wrangling and visualization tools is desirable.

Course Objectives

On completion of this course, students will be able to:

CO1: Understand the theoretical foundations of predictive modelling and the predictive analytics lifecycle.

CO2: Develop proficiency in feature engineering, data preparation, and model selection strategies.

CO3: Build and evaluate supervised learning models for regression and classification tasks.

CO4: Apply advanced ensemble, time-series, and probabilistic forecasting methods to real-world datasets.

CO5: Assess model fairness, interpretability, deployment considerations, and ethical implications of predictive systems.

Course Content

UNIT I: Foundations of Predictive Analytics: [15 hours]

Scope and significance of predictive analytics in business, healthcare, agriculture, finance, and governance; the predictive analytics workflow: problem framing, data acquisition, modelling, evaluation, and deployment; types of prediction tasks: regression, classification, ranking, and probability estimation; distinction between descriptive, diagnostic, predictive, and prescriptive analytics; introduction to supervised and unsupervised learning paradigms; bias–variance tradeoff; overfitting and underfitting; cross-validation techniques; performance metrics: accuracy, precision, recall, F1-score, AUC-ROC, RMSE, MAE, and R^2.

UNIT II: Data Preparation and Feature Engineering: [15 hours]

Exploratory data analysis for predictive modelling; handling missing values: imputation strategies; detection and treatment of outliers; data transformation: normalization, standardization, log and power transforms; encoding categorical variables: label encoding, one-hot encoding, target encoding; feature extraction from text, dates, and structured fields; feature selection methods: filter, wrapper, and embedded methods;

UNIT III: Supervised Learning Models: [15 hours]

Linear and polynomial regression; regularization: Ridge, Lasso, and Elastic Net; logistic regression for binary and multiclass classification. class imbalance handling: oversampling (SMOTE), undersampling, and class weighting; data leakage: identification and prevention; train-validation-test splitting strategies; pipelines for reproducible preprocessing.

UNIT IV: Advanced Predictive Methods and Deployment: [15 hours]

Time-series forecasting: stationarity, decomposition, ARIMA, SARIMA, Exponential Smoothing, and Prophet; survival analysis and hazard modelling; probabilistic forecasting and prediction intervals; model interpretability and explainability: SHAP values, LIME, permutation importance, and partial dependence plots; fairness, accountability, and transparency in predictive models; model deployment concepts: serialization, REST APIs, batch vs. real-time inference, concept drift, and model monitoring; responsible AI considerations and regulatory compliance.

Text Books

1. Hastie, T., Tibshirani, R., and Friedman, J. — The Elements of Statistical Learning, Springer.
2. Géron, A. — Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly.
3. Siegel, E. — Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die, Wiley.

Reference Books

4. James, G., Witten, D., Hastie, T., and Tibshirani, R. — An Introduction to Statistical Learning, Springer.
5. Kuhn, M. and Johnson, K. — Applied Predictive Modeling, Springer.
6. Molnar, C. — Interpretable Machine Learning (Open Access, https://christophm.github.io/interpretable-ml-book/).

Web Resources

7. Scikit-learn Documentation — https://scikit-learn.org/stable/
8. Kaggle Learn — https://www.kaggle.com/learn
9. Google Machine Learning Crash Course — https://developers.google.com/machine-learning/crash-course
10. XGBoost Documentation — https://xgboost.readthedocs.io/
11. SHAP Documentation — https://shap.readthedocs.io/
12. UCI Machine Learning Repository — https://archive.ics.uci.edu/ml/index.php

****

Specialization: Data Science
SEMESTER –V

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V (Data Science)							
5	24BCA51	Introduction to Data Science	04	80	20	100	4
	24BCA52	Time Series Analysis	04	80	20	100	4
	24BCA53	Machine Learning	03	80	20	100	3
	24BCA54	Internet of Things	03	80	20	100	3
	24BCA55	Cyber Security	03	80	20	100	3
	24BCA51P	Data Science Lab	04	40	10	50	2
	24BCA53P	Machine Learning Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2

24BCA51	Introduction to Data Science	60 Hours

Course Objectives

CO1: Understand core data science concepts, data types, lifecycle, and roles.
CO2: Apply data preprocessing including cleaning, transformation, and feature engineering.
CO3: Perform exploratory data analysis using statistics and visualization tools.
CO4: Analyse basic statistical concepts such as probability, distributions, and hypothesis testing.
CO5: Implement machine learning algorithms for regression, classification, and clustering.
CO6: Evaluate data science solutions considering ethics, applications, and deployment.

Prerequisite:

Basic knowledge of programming (preferably Python), Fundamental understanding of Mathematics (Algebra, basic Statistics). Familiarity with data handling concepts, Basic logical and analytical thinking skills.

Unit 1: Foundations of Data Science: [15 hours]

Introduction to Data Science: Definition, scope, applications. Data Science lifecycle (Data collection, Processing, Modelling, and Deployment), Types of Data: Structured, semi-structured, and unstructured. Role of Data Scientist vs Data Analyst vs Data Engineer. Data sources: databases, APIs, web scraping. Data ethics and privacy.

Unit 2: Data Preprocessing and Visualization: [15 hours]

Data cleaning: missing values, noise handling. Data transformation and normalization, Feature engineering basics, Exploratory Data Analysis (EDA) concepts, Descriptive statistics: mean, median,

mode, variance, standard deviation, and skewness. Data visualization principles: Histograms, boxplots, scatter plots, Correlation and covariance, Outlier detection.

Unit 3: Statistical Learning Basics: **[15 hours]**

Introduction to Statistical Learning, Supervised, unsupervised, reinforcement learning. Simple Linear Regression: parameter estimation, t-statistic (method only without derivation). Classification: Logistic Regression.

Unit 4: Data Science Applications and Tools: **[15 hours]**

Data pipelines and workflows, Introduction to Big Data concepts, Cloud platforms for Data Science (overview), Use of SQL for data analysis, Dashboarding tools (e.g., Tableau/Power BI only overview), Application of AI in Data Science, Case studies: e.g., healthcare, finance, agriculture, social media, Model deployment basics (conceptual), Ethics in data science and bias in data.

Text Books:

1. Provost, F., & Fawcett, T. (2013). *Data science for business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media.
2. Cielen, D., Meysman, A. D. B., & Ali, M. (2016). *Introduction to data science*. Dreamtech Press.
3. Maheshwari, A. (2017). *Data analytics*. McGraw-Hill Education.

Reference Books:

1. Han, J., Pei, J., & Tong, H. (2022). Data mining: Concepts and techniques (4th ed.). Morgan Kaufmann.
2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R*. Springer.

Web Link:

NOC: Data Science for Engineers, IIT Madras, Prof. Ragunathan Rengasamy, Prof. Shankar Narasimhan, nptel.ac.in/courses/106106179

****

24BCA52	Time Series Analysis	60 Hours

Course Objectives

CO1: Apply fundamental concepts of time series data, including components, visualization, and decomposition, to analyze real-world datasets.

CO2: Build and implement time series forecasting models such as ARIMA and exponential smoothing using appropriate tools, and evaluate their performance using standard accuracy measures.

CO3: Develop data-driven solutions for real-world applications by applying time series techniques and basic machine learning models.

Course Content:

UNIT I: Introduction to time series data: **[15 hours]**

Definition and examples from business, finance, IoT, and weather domains. Components of time series including trend, seasonal, cyclical, and irregular variations. Additive and multiplicative models with intuitive understanding. Data visualization techniques such as line plots and seasonal plots. Smoothing techniques including simple and weighted moving averages and basic idea of exponential smoothing. Concept of time series decomposition and its tool-based implementation.

Practical: Plotting and decomposition of time series data using Python (Pandas, Matplotlib).

UNIT II: **[15 hours]**

Concept of stationarity and its importance in time series analysis (intuitive approach). Handling non-stationary data using differencing and basic log transformation. Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) for visual interpretation. Introduction to Autoregressive (AR), Moving Average (MA), and ARIMA models (conceptual understanding). ARIMA workflow including model identification using plots, model fitting, and forecasting. Introduction to Auto-ARIMA for automated model selection.

Practical: Identification and fitting of ARIMA models using Python (statsmodels, pmdarima).

UNIT III: **[15 hours]**

Forecasting concepts and steps involved in time series prediction. Exponential smoothing methods including Simple Exponential Smoothing, Holt's Linear Trend method, and Holt-Winters Seasonal method. Model evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Basic residual analysis and understanding of model errors. Model comparison and selection based on performance metrics.

Practical: Forecasting using Holt-Winters method and comparison of models using accuracy metrics.

UNIT IV: **[15 hours]**

Time series applications in data science. Feature engineering techniques including lag variables and rolling statistics (mean and variance). Conversion of time series data into supervised learning format. Application of machine learning models such as Linear Regression and basic tree-based models for forecasting. Introduction to deep learning techniques with a conceptual overview of LSTM. Applications including sales and demand forecasting, stock trend analysis, and anomaly detection.

Practical: Creation of lag features and development of a simple machine learning forecasting model using a real dataset.

Text Books:

1. Aileen Nielsen, *Practical Time Series Analysis*, O'Reilly Media.
2. Manu Joseph, *Hands-On Time Series Analysis with Python*, Packt Publishing.

Reference Books:

1. Brockwell, P.J. and Davis, R.A., *Introduction to Time Series and Forecasting*, Springer.
2. Shumway, R.H. and Stoffer, D.S., *Time Series Analysis and Its Applications*, Springer.

****

24BCA53	Machine Learning	45 Hours

Course Objectives:

CO1: Understand fundamental concepts of machine learning, including ML pipeline, statistical decision theory, and bias–variance trade-off.

CO2: Apply regression and classification algorithms and evaluate their performance using appropriate metrics.

CO3: Analyse and implement unsupervised learning techniques such as clustering and probabilistic models for pattern discovery.

CO4: Understand the principles of neural networks and reinforcement learning, and evaluate their applicability.

Course Content:

UNIT I: Introduction to Machine Learning & Regression Analysis: [10 Hours]
Introduction to Machine Learning: Definition and Types of Machine Learning (Supervised, Unsupervised, Reinforcement Learning), Challenges in Machine Learning , Machine Learning Pipeline: Data Collection, Data Preprocessing, Model Training, Evaluation, Applications of Machine Learning. Ethics in Machine Learning: Bias and Fairness, Privacy Issues. Statistical Foundations: Introduction to Statistical Decision Theory, Bias vs Variance trade-off. Regression Analysis: Simple Linear Regression, Multiple (Multivariate) Linear Regression. Regression Performance Metrics: RMSE, MAE, MAPE, R^2 Score.

UNIT II: Supervised Learning – Classification & Performance Evaluation: [13 Hours]
Similarity-Based Learning: Nearest Neighbour Learning, Weighted k-NN. Classification Algorithms: Decision Trees, Logistic Regression, Support Vector Machines (SVM), Naïve Bayes Classifier. Performance Evaluation of Classifiers: Confusion Matrix, Evaluation Techniques: Cross-validation, Bootstrapping, ROC Curve and AUC. Ensemble Methods: Introduction to Ensemble Learning, Bagging, Boosting, Stacking, Random Forest, Gradient Boosting.

UNIT III: Unsupervised Learning – Cluster Analysis: [12 Hours]
Clustering Techniques: Partitional Clustering (k-Means), Hierarchical Clustering. Advanced Clustering Algorithms: BIRCH, CURE, Density-Based Clustering (DBSCAN). Introduction to Probabilistic Clustering: Gaussian Mixture Models (GMM), Expectation Maximization (EM). Clustering Evaluation Metrics: Silhouette Score, Davies-Bouldin Index, Calinski-Harabasz Index.

UNIT IV: Neural Networks & Reinforcement Learning: [10 Hours]
Neural Networks: Perceptron Model, Multilayer Neural Networks, Backpropagation Algorithm. Deep Learning Basics: Introduction to Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM). Reinforcement Learning: Introduction and Key Components (Agent, Environment, Reward, Policy), Q-Value Concept, Q-Learning Algorithm.

Textbooks:

1. Hiran, K. K., Doshi, R., Jain, R. K., & Lakhwani, K. (2021). *Machine Learning*. BPB Publications.
2. Ethem Alpaydin. Introduction to machine learning. MIT press, 2020.

References:

1. Hiran, K. K., Doshi, R., Jain, R. K., & Lakhwani, K. (2021). *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples*. BPB Publications.
2. Barua, T., Hiran, K. K., Jain, R. K., & Doshi, R. (2024). *Machine Learning with Python*. Walter de Gruyter.
3. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 1997.

Web Resources

1. NPTEL – Machine Learning Courses
2. Coursera – Machine Learning Courses
3. Scikit-learn Official Documentation
4. Kaggle – Machine Learning Platform

****

24BCA51P	Data Science Lab	60 Hours

Note: a) Any programming language can be used. Python is preferable.
b) Dataset: The following link provides the dataset that can be used for this experiment.
https://inria.github.io/scikit-learn-mooc/python_scripts/datasets_ames_housing.html

1. Perform data loading and basic inspection to understand dataset structure, types, and summary for multivariate datasets.
2. Clean a dataset by handling missing values, noise, and inconsistencies.
3. Apply data transformation techniques such as normalization and scaling.
4. Construct new features from existing attributes to improve data representation.
5. Perform exploratory data analysis to identify patterns and trends in a dataset.
6. Compute descriptive statistics such as mean, median, variance, and standard deviation.
7. Visualize data using histograms, boxplots, and scatter plots to interpret distributions.
8. Analyse relationships between variables using correlation and covariance.
9. Detect and interpret outliers in datasets using appropriate techniques.
10. Implement linear regression to model relationships between variables and interpret results, computing t-statistics for coefficients.
11. Implement classification techniques such as Logistic Regression and K-Nearest Neighbours algorithms.
12. Implement K Means clustering algorithm.
13. Evaluate machine learning models using accuracy, precision, recall, and F1-score, and analyse overfitting and underfitting.
14. Design a simple data pipeline showing the steps of data collection, preprocessing, analysis, and reporting.
15. Perform basic data analysis queries on a dataset using SQL concepts such as selection, filtering, and aggregation.
16. Create a simple dashboard to present key insights from a dataset using charts, tables, and summary indicators.

Text Books:

1. McKinney, W. (2017). *Python for data analysis* (2nd ed.). O'Reilly Media.
2. Sharda, R., Delen, D., & Turban, E. (2019). *Business intelligence, analytics, and data science: A managerial perspective* (4th ed.). Pearson.

Reference Books:

1. Bowerman, B. L., O'Connell, R. T., & Murphree, E. S. (2019). *Business statistics in practice: Using data, modeling, and analytics* (8th ed.). McGraw-Hill Education.
2. Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical statistics for data scientists: 50+ essential concepts using R and Python* (2nd ed.). O'Reilly Media.
3. Géron, A. (2022). *Hands-on machine learning with scikit-learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (3rd ed.). O'Reilly Media.
4. Motwani, B. (2017). *Data analytics using Python* (2nd ed.). Wiley.

Web Link:

1. NOC: Python for Data Science, IIT Madras, Prof. Ragunathan Rengasamy, nptel.ac.in/courses/106106212

****

24BCA53P	Machine Learning Lab	60 Hours

Note: The programs can be implemented in either JAVA or Python. Data sets can be taken from standard repositories.

Course Objectives:

CO1: Implement and evaluate supervised learning algorithms for regression and classification tasks.
CO2: Apply unsupervised learning techniques such as clustering to analyse and interpret data patterns.
CO3: Develop and assess classification models using random forests, support vector machines.
CO4: Demonstrate the working neural networks and implement reinforcement learning using Q-learning.

LAB PROGRAMS

1. Write a program to implement Simple Linear Regression**.** Train the model and plot the regression line.
2. Write a program to implement the KNN classifier. Experiment with different values of K.
3. Write a program which uses Weighted KNN. Compare the results with those obtained in program 2.
4. Write a program to implement the decision tree based ID3 algorithm. Use an appropriate dataset and check with test data.
5. Write a program to implement the naïve Bayesian classifier. Evaluate the performance of the classifier.
6. Write a program to implement Support Vector Machine (SVM). Use linear kernel for classification.
7. Build and evaluate a random forest classifier using a numerical dataset.
8. Write a program to apply clustering (K-means).
9. Write a program to perform Hierarchical Clustering**.** Plot dendrogram.
10. Build a single layer perceptron model to perform binary classification (may use TensorFlow/Keras) and evaluate its performance.

11. Write a program to implement image classification using Keras.
12. Write a program to implement Q-Learning.

Web Resources

- Scikit-learn Official Documentation – https://scikit-learn.org
- Kaggle – https://www.kaggle.com
- UCI Machine Learning Repository – https://archive.ics.uci.edu/ml
- DataCamp ML Tutorial – https://www.datacamp.com/tutorial/machine-learning-python
- Real Python ML Tutorials – https://realpython.com/tutorials/machine-learning/

****

Specialization: Data Science
SEMESTER –VI

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – VI (AI & ML)							
6	24BCA61	Big Data Analytics	04	80	20	100	4
	24BCA62	Exploratory Data Analysis	04	80	20	100	4
	24BCA63	Research Methodology	03	80	20	100	3
	24BCA64	Generative AI	03	80	20	100	3
	24BCA65	Project Work	08	80	20	100	4
	24BCA66	Internship	04	40	10	50	2
	24BCASEC3	Soft Skills	03	80	20	100	3

24BCA61	Big Data Analytics	60 Hours

Course Outcomes:

CO1: Understand the fundamental concepts of big data, its characteristics, evolution, and the importance of big data analytics in modern applications.

CO2: Understand the Hadoop ecosystem including HDFS, YARN, and MapReduce, and apply them for distributed data processing.

CO3: Apply NoSQL concepts using MongoDB and perform basic data operations and queries in big data environments.

CO4: Understand and apply big data processing tools such as Hive and Pig for managing and querying large datasets.

CO5: Understand the concepts of Apache Spark and apply it for basic data analysis in big data environments.

CO6: Understand text and web analytics concepts including web mining techniques and PageRank for analyzing web data.

Course Content:

UNIT –I: Introduction to Big Data: **[15 hours]**

Classification of data, characteristics of data, evolution and definition of big data, concept of big data, need for big data, traditional business intelligence versus big data, typical data warehouse and Hadoop environment. Big data analytics: definition of big data analytics, classification of analytics, importance of big data analytics, technologies used in big data environments, overview of analytical tools, introduction to NoSQL databases.

UNIT – II: Introduction to Hadoop and MapReduce Programming: [15 hours]
Introduction to Hadoop: introducing Hadoop, need for Hadoop, limitations of RDBMS, comparison of RDBMS and Hadoop, history of Hadoop, Hadoop overview, use cases of Hadoop, Hadoop Distributed File System (HDFS), data processing with Hadoop, resource management and application management using YARN. Introduction to MapReduce programming: introduction, Mapper, Reducer, Combiner, Partitioner, searching, sorting, compression, illustrating batch-oriented distributed data processing in Hadoop.

UNIT – III: Introduction to MongoDB and Pig: [15 hours]
Introduction to MongoDB: concept of MongoDB, need for MongoDB, terminology comparison between RDBMS and MongoDB, data types in MongoDB, MongoDB query language. Introduction to Hive: concept of Hive, Hive architecture, Hive data types, Hive file formats, Hive query language (HQL), user defined functions (UDF). Introduction to Pig: concept of Pig, anatomy of Pig, Pig on Hadoop, Pig philosophy, use cases of Pig, Pig Latin overview, data types in Pig, execution modes of Pig, HDFS commands, relational operators, evaluation functions, complex data types, comparison of Pig and Hive, emphasizing high-level data processing and querying in big data environments.

UNIT – IV: Spark and big data analytics: [15 hours]
Spark and big data analytics: introduction to Spark as an advanced distributed computing framework overcoming limitations of MapReduce, introduction to data analysis using Spark. Text, web content and link analytics: introduction, text mining, web mining, web content analytics, web usage analytics, PageRank algorithm, structure of the web, analysis of web graph, focusing on modern large-scale data analytics techniques.

References

1. Raj Kamal & Preeti Saxena, *Big Data Analytics: Introduction to Hadoop, Spark, and Machine Learning*
2. Tom White, *Hadoop: The Definitive Guide*
3. Mohammed Guller, *Big Data Analytics with Spark*
4. Venkat Ankam, *Big Data Analytics*
5. Ambrish Kumar Sharma et al., *A Brief Guide to Big Data and Hadoop*

****

24BCA62	Exploratory Data Analysis	60 Hours

Course Objectives
CO1: Understand the role of statistical reasoning in data exploration.
CO2: Apply multivariate and relational analysis techniques conceptually.
CO3: Explore data from real-world domains such as business, HR, and social media.
CO4: Communicate EDA findings through structured reports and visualizations.
CO5: Identify patterns, trends, and anomalies in datasets systematically.
CO6: Generate meaningful insights and hypotheses from exploratory analysis.

Prerequisite: Completion of Data Science. Basic knowledge of descriptive statistics, data types, and Python fundamentals.

Unit 1: Statistical Reasoning for EDA: [15 hours]
Probability distributions: Normal, Binomial, and Poisson - properties and identification in data. Probability Density Function (PDF) and Cumulative Distribution Function (CDF) - concept and

interpretation. Central Limit Theorem and its relevance to data analysis. Sampling strategies: simple random, stratified, and systematic sampling. Confidence intervals: concept and interpretation. Hypothesis testing basics: null and alternate hypothesis, z-test, t-test, and Chi-Square test. Statistical significance and p-value interpretation.

Unit 2: Multivariate and Relational Analysis: [15 hours]
Multivariate data: concept and challenges. Cross-tabulation and contingency tables. Partial correlation and multicollinearity concept. Stratified analysis and subgroup exploration. Dimensionality reduction: Principal Component Analysis (PCA) - concept, scree plot, and variance explained (conceptual only). Association measures: Cramér's V and phi coefficient. Mosaic plots and their interpretation.

Unit 3: Domain-Specific EDA: [15 hours]
Time series data: trend, seasonality, and irregular variation; autocorrelation concept and stationarity. Text data exploration: word frequency, TF-IDF concept, and sentiment polarity. Survey data: Likert scale and ordinal data exploration. Domain overviews: HR analytics (attrition and performance); financial data (risk and return concepts); social media analytics (engagement and sentiment). Imbalanced datasets: concept and impact on EDA.

Unit 4: EDA Reporting and Data Storytelling: [15 hours]
Principles of visual encoding: position, colour, size, and shape. Chart selection framework: choosing the right chart for the right question. Advanced chart types: violin plots, funnel charts, and waterfall charts — use and interpretation. Common visualization mistakes and corrections. Data storytelling: context, conflict, and resolution. Insight pyramid: data – observation - insight - recommendation. Automated EDA tools: Pandas Profiling and Sweet viz - features and use (conceptual only). Bias in EDA: selection bias and sampling bias. Case studies: student performance, customer segmentation, and employee attrition.

Text Books:

1. Mukhiya, S. K., & Ahmed, U. (2020). *Hands-on exploratory data analysis with Python.* Packt Publishing.
2. Downey, A. B. (2014). *Think stats: Exploratory data analysis in Python.* O'Reilly Media.
3. Knaflic, C. N. (2015). *Storytelling with data: A data visualization guide for business professionals.* Wiley.

Reference Books:

1. Tukey, J. W. (1977). *Exploratory data analysis.* Addison-Wesley.
2. Wilke, C. O. (2019). *Fundamentals of data visualization.* O'Reilly Media.
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R.* Springer.

Online Resources:

1. NPTEL: Exploratory Data Analysis for Data Science with R Software — https://onlinecourses.nptel.ac.in
2. DataCamp: Exploratory Data Analysis in Python — https://www.datacamp.com
3. Kaggle: EDA notebooks and datasets — https://www.kaggle.com
4. Towards Data Science: EDA articles — https://towardsdatascience.com

*****

Specialization: Full Stack Development

SEMESTER –V

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – V(Full Stack Development)							
5	24BCA51	Frontend Design	04	80	20	100	4
	24BCA52	Backend Development	04	80	20	100	4
	24BCA53	Cloud computing	03	80	20	100	3
	24BCA54	Internet of Things	03	80	20	100	3
	24BCA55	Cyber Security	03	80	20	100	3
	24BCA51P	Frontend Design Lab	04	40	10	50	2
	24BCA52P	Backend Development Lab	04	40	10	50	2
	24BCASEC2	Quantitative Techniques	02	40	10	50	2

24BCA51	Frontend Design	60 Hours

Course Objectives:

CO1: Understand the fundamental concepts of web page design using HTML, CSS, and responsive design techniques for building structured and accessible user interfaces.

CO2: Develop knowledge of client-side scripting using JavaScript and dynamic content manipulation through DOM for creating interactive and user-driven web applications.

CO3: Understand asynchronous communication using Fetch API and data exchange formats such as JSON for integrating external data and handling real-time UI updates.

CO4: Gain knowledge of component-based frontend development using modern frameworks (e.g., React) and reusable UI design principles for building scalable applications.

CO5: Understand modern frontend development practices including UI frameworks, styling systems, performance optimization, and AI-assisted (prompt-based) development techniques.

CO6: Understand frontend application architecture, API communication, version control, debugging, and deployment concepts for building and delivering production-ready web applications.

Prerequisite:

Basic understanding of HTML, CSS, and JavaScript at a foundational level. Familiarity with fundamental web concepts such as web browsers, DNS, and client-server architecture. Basic knowledge of programming concepts including variables, loops, functions, and data structures. Exposure to problem-solving approaches and a willingness to use modern development tools and AI-assisted coding platforms for learning and building applications.

Course Content:

UNIT I: Core Web Foundations: **[15 Hours]**
Semantic Structure and Layout Design: HTML5 semantic elements, forms, accessibility basics. Layout design using Flexbox and CSS Grid. Structuring clean and maintainable web pages. Responsive Design and Modern Styling: Responsive design using media queries (mobile-first). Introduction to modern styling tools (Tailwind CSS via CDN). Basics of web concepts: browsers, HTTP/HTTPS, DNS.

UNIT II: JavaScript and Dynamic UI Development: **[15 Hours]**
JavaScript and DOM Manipulation: JavaScript fundamentals, events, DOM selection and updates. Form validation and interactive UI behavior. Asynchronous Data Handling: Fetch API for API integration. JSON handling, loading states, and error handling in UI. Introduction to breaking problems into smaller steps.

UNIT III: Modern Frontend Development: **[15 Hours]**
Component-Based Development: Introduction to React (via CDN). Components, props, basic state (useState). Building reusable UI components. Modern UI and Resource Usage: Using Tailwind CSS for rapid UI design. Basic UI principles and consistency. Using CDN and simple package tools for integrating libraries.

UNIT IV: Production, AI-Assisted Development and Deployment: **[15 Hours]**
AI-Assisted Development and Debugging: Using AI tools (ChatGPT, Gemini, Claude, Copilot) for code generation. Writing minimal prompts, refining output, and debugging generated code. Deployment and Real-World Practices: Version control using Git basics. Hosting frontend apps using Vercel/Netlify. Basic performance optimization and debugging in production.

Frontend Textbooks:

1. Duckett, Jon, HTML & CSS: Design and Build Websites, Wiley, 2011.

2. Robbins, Jennifer Niederst, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, O'Reilly Media, 2018.

3. Haverbeke, Marijn, Eloquent JavaScript: A Modern Introduction to Programming, 3rd Edition, No Starch Press, 2018.

Web Resources

Core Web (HTML, CSS, Basics)

1. https://developer.mozilla.org/en-US/docs/Web/HTML
2. https://developer.mozilla.org/en-US/docs/Web/CSS
3. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form
4. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table

Responsive Design & Layout

1. https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout
2. https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries
3. https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout

Bootstrap (Framework)

1. https://getbootstrap.com/docs/5.3/getting-started/introduction/

2. https://getbootstrap.com/docs/5.3/layout/grid/
3. https://getbootstrap.com/docs/5.3/components/

JavaScript & DOM

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript
2. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model
3. https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector

AJAX / Fetch API

1. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
2. https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

JSON & XML

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON
2. https://developer.mozilla.org/en-US/docs/Web/XML

PHP (Server Side Basics)

1. https://www.php.net/manual/en/
2. https://www.w3schools.com/php/

CDN & External Resources

3. https://developer.mozilla.org/en-US/docs/Glossary/CDN
4. https://www.jsdelivr.com/
5. https://cdnjs.com/

Material Design & UI Concepts

1. https://m3.material.io/
2. https://getbootstrap.com/

Markdown

1. https://www.markdownguide.org/
2. https://commonmark.org/help/

Web APIs & Architecture

1. https://developer.mozilla.org/en-US/docs/Web/API
2. https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Client-side_APIs/Introduction

Security

1. https://developer.mozilla.org/en-US/docs/Web/Security
2. https://owasp.org/www-project-top-ten/

Deployment & Hosting

1. https://vercel.com/docs
2. https://docs.netlify.com/

****

24BCA52	Backend Development	60 Hours

Course Objectives:

CO1: Understand the fundamental concepts of backend development, client-server architecture, HTTP protocols, and REST API design.

CO2: Develop knowledge of building server-side applications using Python (Flask) and handling requests, responses, and JSON data exchange.

CO3: Understand database integration and perform CRUD operations through backend APIs with proper data handling and validation.

CO4: Gain knowledge of backend development using Node.js (Express) and asynchronous programming for scalable application development.

CO5: Understand modern backend tools including no-code/low-code platforms (e.g., n8n), AI-assisted (vibe coding) development, and integration of AI applications such as chatbots.

CO6: Learn deployment, version control, and production practices along with building AI-powered workflows and agent-based backend systems.

Prerequisite:

Basic knowledge of programming concepts including variables, loops, functions, and data structures. Familiarity with fundamental web concepts such as client-server architecture, HTTP/HTTPS, and APIs. Basic understanding of JavaScript and any programming language (Python preferred). Awareness of databases (DBMS concepts) and willingness to learn modern development tools, automation platforms, and AI-assisted coding techniques.

Course Content:

UNIT I: Backend Fundamentals and REST API Basics: [15 Hours]

Server-Side Concepts and API Foundations: Introduction to backend and client-server architecture. HTTP methods, request-response cycle. Designing simple REST APIs and understanding endpoints. Building APIs using Python (Flask): Creating basic REST APIs using Flask. Routing, handling requests/responses, returning JSON data.

UNIT II: Database Integration and Backend Development: [15 Hours]

Database and CRUD Operations: Introduction to databases and integration. Performing CRUD operations using backend APIs. Handling user input and basic validation. Backend using Node.js and Express: Introduction to Node.js and Express. Creating REST APIs, middleware usage, and handling asynchronous operations.

UNIT III: Modern Backend Tools and AI-Assisted Development: [15 Hours]

No-Code/Low-Code Backend and Automation Tools: Using n8n for building backend workflows, API integrations, and automation without heavy coding. AI-Assisted Development (Vibe Coding): Using AI tools to generate backend logic, APIs, and integrations. Writing minimal prompts, refining outputs, and debugging AI-generated code.

UNIT IV: AI Applications, Agents and Deployment: [15 Hours]

Building AI Applications and Chatbots: Creating simple AI-based applications and shopping cart using APIs. Integrating backend with AI services for automation and user interaction. Agents, Deployment and Production Practices: Introduction to AI agents and workflow-based systems. Deployment of backend services using cloud platforms (Render,Vercel, PythonAnywhere). Basic version control and production practices.

Backend Textbooks:

1. Hillar, Gastón C., Building RESTful Python Web Services, Packt Publishing, 2016.

2. Brown, Ethan, Web Development with Node and Express: Leveraging the JavaScript Stack, O'Reilly Media, 2014.

3. Kumar, P. Sivakumar and others, AI Agents Using n8n for Beginners, CSharpCorner Publishing, 2025.

4. Subramanian, Harihara and Raj, Pethuru, RESTful API Design Patterns and Best Practices, O'Reilly Media, 2025.

Resources

Backend Fundamentals & REST APIs

1. https://developer.mozilla.org/en-US/docs/Web/HTTP
2. https://restfulapi.net/

Python Backend (Flask)

1. https://flask.palletsprojects.com/en/stable/quickstart/
2. https://docs.python.org/3/tutorial/
3. https://realpython.com/tutorials/flask/

Node.js and Express.js

1. https://nodejs.org/en/docs
2. https://expressjs.com/en/starter/installing.html
3. https://www.freecodecamp.org/news/tag/nodejs/

Database and CRUD Operations

1. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose
2. https://www.w3schools.com/sql/
3. https://www.mongodb.com/docs/

No-Code / Automation Tools

1. https://docs.n8n.io/
2. https://zapier.com/learn/
3. https://make.com/en/help

AI Applications and Chatbots

1. https://platform.openai.com/docs
2. https://python.langchain.com/docs/
3. https://learn.microsoft.com/en-us/azure/ai-services/

Version Control and Deployment

1. https://git-scm.com/docs
2. https://docs.github.com/en/get-started

3. https://render.com/docs
4. https://vercel.com/docs

Backend Security and Best Practices

1. https://owasp.org/www-project-top-ten/
2. https://developer.mozilla.org/en-US/docs/Web/Security

****

24BCA53	Cloud computing	45 Hours

Course Objectives

CO1: Understand the foundations of cloud computing and its relevance to data analytics.
CO2: Learn cloud service models, deployment models, and cloud data architectures.
CO3: Develop the ability to use cloud platforms for data storage, processing, and analytics workflows.
CO4: Study scalable analytics using distributed and cloud-native tools.
CO5: Analyze security, governance, cost, and ethical concerns in cloud-based analytics systems.

Prerequisite:

Basic understanding of HTML, CSS, JavaScript, and PHP at a theoretical level. Familiarity with fundamental web concepts such as web browsers, DNS, and client-server architecture. Basic knowledge of programming concepts including variables, loops, functions, and data structures.

Course Content:

UNIT I: Introduction to Cloud Computing and Data Analytics: [12 Hours]

Evolution of distributed computing to cloud platforms, characteristics of cloud computing including on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, virtualization and containerization, cloud service models such as IaaS, PaaS, SaaS, CaaS, FaaS, AIaaS, cloud deployment models including public, private, hybrid, community, need for scalability in analytics, benefits and limitations of cloud-based analytics.

UNIT II: Cloud Fundamentals and Data Management: [11 Hours]

Cost models and pay-as-you-go pricing, workload optimization, vendor lock-in, shared responsibility principle, identity and access management, authentication and authorization, encryption at rest and in transit, data privacy and compliance, ethical issues in cloud-hosted data systems, environmental sustainability in cloud computing, overview of major cloud ecosystems such as AWS, Microsoft Azure, and Google Cloud. Cloud storage concepts including object storage, block storage, and file storage, data lakes and data warehouses in cloud environments, relational and NoSQL cloud databases, distributed file systems, data ingestion pipelines, ETL and ELT concepts, role of distributed computing in handling large-scale datasets.

UNIT III: Cloud Data Processing and Analytics Workflow: [11 Hours]

Data ingestion in cloud environments including batch and streaming data concepts, overview of distributed data processing, introduction to Apache Spark, concepts of RDD and DataFrames, data preprocessing and transformation techniques including filtering, aggregation, and cleaning, handling large-scale datasets. Data storage for analytics including data lake and data warehouse concepts, querying large datasets, introduction to cloud-based notebook environments, basic workflow automation and job scheduling concepts.

UNIT IV: Scalable Analytics, Machine Learning and Visualization on Cloud: [11 Hours]
Scalable analytics including parallel processing and distributed querying concepts, introduction to cloud-based machine learning workflows, overview of MLlib, basic concepts of model building and evaluation in cloud environments. Data visualization and business intelligence in cloud platforms, dashboard creation concepts, applications of analytics in business, healthcare, governance, and social media, monitoring and logging of analytics workloads, backup, disaster recovery, and high availability concepts.

Text Books

1. Rajkumar Buyya, Christian Vecchiola, and S. Thamarai Selvi, *Mastering Cloud Computing*, McGraw Hill.
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, Pearson.
3. Jules S. Damji, Lee Stultz, Ted Malone, et al., *Data Analytics on Microsoft Azure*, Microsoft Press.

Reference Books

1. Holden Karau, Andy Konwinski, Patrick Wendell, and Matei Zaharia, *Learning Spark*, O'Reilly.
2. Martin Kleppmann, *Designing Data-Intensive Applications*, O'Reilly.

Web Resources

1. AWS Academy Training and Documentation
2. Microsoft Learn for Azure Data and Analytics
3. Google Cloud Training for Data Analytics
4. Apache Spark Documentation
5. Docker and Kubernetes Documentation

****

24BCA51P	**Frontend Design Lab**

Note: From 6th Question onwards usage of AI can be encouraged for faster development

1. Create a Structured Webpage using Semantic HTML:
   a. Design a webpage using <header>, <nav>, <section>, <article>, <footer> for a college event page.
   b. Include a schedule table, speaker list, images, and navigation links.

2. Design a Responsive Layout using CSS (Flexbox & Grid):

a. Create a webpage layout with header, sidebar, content, and footer using Flexbox and CSS Grid.
b. Apply box model, spacing, and alignment properly.

3. Create a Mobile-First Responsive Webpage:
   a. Design a product landing page using media queries (mobile-first approach).
   b. Adjust layout, font sizes, and visibility for mobile, tablet, and desktop.

4. Build a Styled UI using Tailwind CSS (CDN):
   a. Use Tailwind CSS CDN to design a responsive card layout.
   b. Create reusable components like buttons, cards, and navbar.

5. Build a Registration Form with Validation:
   a. Create a form with input types: text, email, password, radio, checkbox, select.
   b. Validate using JavaScript (pattern, required, length) and show dynamic error messages.

6. Build a Calculator using JavaScript and Enhance Features:
   a. Create a calculator for basic operations (+, −, ×, ÷) using JavaScript.
   b. Enhance it with:
      - Keyboard input support
      - Error handling (divide by zero)
      - Clear/reset functionality

7. Implement DOM Manipulation for a Dynamic UI:
   a. Create a task list where users can add, edit, and delete items using querySelector.
   b. Dynamically update styles and content on user actions.

8. Fetch and Display API Data with Error Handling:
   a. Fetch data from a public API (e.g., users/posts) using Fetch API.
   b. Display data in a table with:
      - Loading indicator
      - Error message if API fails

9. Convert Static Page into React Components (CDN):
   a. Take an existing HTML page and convert it into React components using CDN.
   b. Use props and basic state (useState) for dynamic data.

10. Build a Mini Dashboard using React:
    a. Create a dashboard with components (navbar, sidebar, content cards).
    b. Display dynamic data (API or static JSON) and manage state.

11. Solve UI Problem using AI with Minimal Prompts:
    a. Given a broken UI (layout or styling issue), fix it using AI assistance.
    b. Reduce prompt size step-by-step and document improvements.

12. Deploy a Production-Ready Frontend Application:
    a. Build a complete frontend app (React + API integration).
    b. Use Git for version control and deploy using Vercel or Netlify.

c. Submit:
- Live deployed link
- Code repository
- Prompts used and improvements made

****

24BCA52P	Backend Development Lab	60 Hours

1. Build a User List API using Flask:
   a. Create a Flask server with endpoint /users returning a static list of 5 users (JSON).
   b. Add endpoint /users/<id> to return details of a specific user.

2. Create a Student Registration API:
   a. Build a Flask API with endpoint /register to accept name and email using POST.
   b. Validate email format and return success or error message.

3. Implement CRUD API with Database:
   a. Create a "students" table in SQLite/MySQL.
   b. Implement API to insert and fetch student records.

4. Extend CRUD with Update and Delete:
   a. Add API endpoints to update student name by ID.
   b. Add endpoint to delete a student record.

5. Build Product API using Node.js (Express):
   a. Create an Express server with endpoint /products returning a list of products.
   b. Add POST endpoint to add a new product.

6. Add Middleware and Validation:
   a. Add middleware in Express to log request method and URL.
   b. Validate product data (name not empty, price > 0).

AI-ASSISTED + REAL-WORLD PROBLEMS

7. Build a Task Manager API:
   a. Create API with endpoints to add and view tasks (title, status).
   b. Use AI to generate initial structure and refine it.

8. Build a Login API:
   a. Create API /login that checks username and password (hardcoded).
   b. Return success or failure response in JSON.

9. Build Weather Data API Integration:
   a. Fetch weather data from a public API.
   b. Create backend endpoint /weather to return formatted response.

10. Create Workflow using n8n:
   a. Use n8n to create a webhook.
   b. Send data from your API to n8n and log/store the response.

11. Build a Simple Shopping Cart:
    a. Create endpoint /addOrders that takes user order.
    b. Create endpoint /showOrders that shows recent orders.

12. Improve API using AI:
    a. Take an existing API and improve response format using AI.
    b. Add proper status messages and structure.

13. Version Control using Git:
    a. Initialize Git repository and commit all files.
    b. Push code to GitHub.

14. Deploy Backend API:
    a. Deploy your API on Render/Vercel/PythonAnywhere.
    b. Test all endpoints after deployment.

*****

Specialization: Full Stack Development

SEMESTER –VI

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours / week	Semester End Exam	Internal Assessment	Total Marks	Credits
Semester – VI (AI & ML)							
6	24BCA61	Software Project Management	04	80	20	100	4
	24BCA62	Mobile Application Development	04	80	20	100	4
	24BCA63	Research Methodology	03	80	20	100	3
	24BCA64	Generative AI	03	80	20	100	3
	24BCA65	Project Work	08	80	20	100	4
	24BCA66	Internship	04	40	10	50	2
	24BCASEC3	Soft Skills	03	80	20	100	3

24BCA61	Software Project Management	60 Hours

Prerequisite:
Basic knowledge of Software Engineering concepts such as software life cycle, Project planning, Design and Testing.

Course Objectives:

CO1: Apply project management concepts and AI-assisted techniques to IT projects.
CO2: Identify issues leading to project success/failure using predictive analytics.
CO3: Explain project management in AI-enhanced software development processes. **CO4:** Describe responsibilities of IT project managers in AI-driven environments.

Unit-I:

Introduction to Software Project Management and AI-Driven Project Evaluation: [15 hours]

Importance of Software Project Management, Activities, Methodologies, Categorization of Software Projects. Setting objectives, Management Principles, Management Control, Project Portfolio Management. Cost-benefit evaluation technology with AI-based analytics. Risk evaluation using predictive analytics. Project Planning: Top-Down and Bottom-Up Planning, Types of Activities, Project Duration Estimation using traditional and AI-based prediction techniques. Schedule Monitoring Tools – Gantt Chart, PERT Chart, Critical Path Method.

Introduction to AI-assisted project planning and automated scheduling tools.

Unit-II: Project Life Cycle and AI-based Effort Estimation: [15 hours]
Project Life Cycle and AI-Based Effort Estimation: Software Process and Process Models, Choice of Process Models, Rapid Application Development (RAD), Dynamic System Development Method (DSDM), Extreme Programming (XP), Managing iterative processes. Basics of Software Estimation, Effort and Cost Estimation Techniques, COSMIC Full Function Points, COCOMO II – Parametric Productivity Model. AI-based estimation models using machine learning for accurate effort prediction.

Unit-III: Activity Planning and AI-Driven Analytics: [15 hours]
Objectives of Activity Planning, Project Scheduling, Activities, Sequencing and Scheduling, Network Planning Models, Forward Pass & Backward Pass Techniques, Critical Path Method (CPM). Software Measurements – Monitoring and Measurement of Software Development (Cost, Size, Time Metrics). Methods and tools for metrics, issues in multi-project environments. AI-based analytics for predictive monitoring, and intelligent dashboards.

Unit-IV: Risk Management Resource Optimization and AI-based Control: [15 hours]
Risk Management, Resource Optimization and AI-Based Control:
Risk Identification, Risk Assessment, Risk Planning, Risk Management using AI-based prediction models. PERT Technique, Monte Carlo Simulation with AI enhancements. Resource Allocation using optimization and AI-based recommendations. Creation of Critical Paths and Cost Schedules. Project Management and Control: Framework for Management and Control, Data Collection, Visualization of Progress, Cost Monitoring, Earned Value Analysis, Project Tracking, Change Control, Software Configuration Management, Contract Management.

Case Study Tools: Primavera, Redmine, and AI-enabled tools for intelligent project management.

Text and Reference Books:

1. Bob Hughes, Mike Cotterell and Rajib Mall: Software Project Management – Fifth Edition, Tata McGraw Hill, New Delhi, 2012.
2. Robert K. Wysocki, Effective Software Project Management – Wiley Publication,
3. Walker Royce, Software Project Management- Addison-Wesley, 1998.
4. Gopalaswamy Ramesh, Managing Global Software Projects – McGraw Hill Education (India), Fourteenth Reprint 2013.2011.

Course Outcomes:
After completing the course, students will be able to:
1. Identify the different project contexts and suggest an appropriate management strategy.
2. Describe the key phases of project management with AI-based estimation models.
3. Use predictive analytics for risk and cost management
4. Determine an appropriate project management approach using optimization and AI-based recommendations.
5. Analyze project data using AI-based analytics, and intelligent dashboards
6. Understand ethical and practical implications of AI in project management

****

24BCA62	Mobile Application Development	60 Hours

Course Objectives:

CO1: Understand the fundamentals of mobile computing, mobile application types, and mobile operating system architecture.

CO2: Learn various mobile application development approaches and tools including native and cross-platform frameworks such as React Native and Flutter.

CO3: Understand mobile UI/UX design principles, usability concepts, and prototyping techniques for building user-friendly applications.

CO4: Gain knowledge of data management in mobile applications, including local storage, backend integration, REST APIs, and cloud services.

CO5: Understand the use of device features, system services, and resource management in mobile applications.

CO6: Learn mobile application security concepts, deployment processes, and modern development trends such as low-code platforms and AI-assisted development.

Prerequisite:

Basic understanding of programming, object-oriented concepts, database management systems, computer networks, and web development (frontend and backend).

Course Content:

UNIT I: Introduction to Mobile Application Development: [15 Hours]

Mobile Ecosystem and Architecture: Introduction to mobile computing, evolution of mobile applications, types of mobile apps (native, web, hybrid, cross-platform). Overview of mobile operating systems and architecture. Understanding mobile device constraints such as memory, processing power, battery, and network limitations. Development Approaches and Tools: Introduction to development approaches including native and cross-platform frameworks. Overview of modern tools and frameworks such as React Native, Flutter, and Progressive Web Apps (PWA). Role of frontend and backend in mobile applications.

UNIT II: UI/UX Design for Mobile Applications: [15 Hours]

Mobile UI Design Principles: Principles of mobile UI design including layout, navigation patterns, responsiveness, and usability. Concepts of touch interaction, gestures, and accessibility. Material Design and Human Interface Guidelines (conceptual). User Experience and Prototyping: Designing user-friendly interfaces, wireframing, and prototyping concepts. Understanding user flows, feedback systems, and performance considerations. Introduction to design tools and best practices.

UNIT III: Data Management and Backend Integration: [15 Hours]

Local Data and Storage: Concepts of local storage in mobile applications including files, key-value storage, and lightweight databases. Offline-first approach and synchronization concepts. Backend Integration and APIs: Integration of mobile apps with backend services. REST API concepts, JSON-based communication, authentication basics, and handling asynchronous data. Role of cloud services in mobile applications.

UNIT IV: Advanced Concepts and Modern Trends: [15 Hours]

Device Features and System Services: Use of device features such as sensors, location services, camera, and notifications (conceptual understanding). Background services and resource management. Security, Deployment and Modern Practices: Mobile application security concepts, data protection, and safe API usage. Deployment concepts including app stores and distribution.

Introduction to modern trends such as low-code/no-code platforms, AI-assisted development (vibe coding using agents like Claude, Codex, Gemini), and continuous integration.

Text Books

1. Singh, Michael J., *Learning Mobile App Development: A Hands-On Guide to Building iOS and Android Apps*, Pearson, 2014.
2. Eisenman, Bonnie, *Learning React Native, 2nd Edition*, O'Reilly Media, 2017.
3. Richards, Mark and Ford, Neal, *Fundamentals of Software Architecture: An Engineering Approach*, O'Reilly Media, 2020.
4. Glasser, J.D., *Secure Development for Mobile Apps*, Manning, 2023.

Reference Books

1. Android Programming: The Big Nerd Ranch Guide by Bill Phillips, Chris Stewart, Big Nerd Ranch Guides, 2019.
2. Learning React Native by Bonnie Eisenman, O'Reilly Media, 2017.
3. Flutter in Action by Eric Windmill, Manning Publications, 2019.

Web Resources

1. React Native official documentation https://reactnative.dev/
2. Flutter official documentation https://flutter.dev/
3. Android Developers https://developer.android.com/
4. Apple Developer Documentation https://developer.apple.com/
5. Material Design Guidelines https://m3.material.io/
6. Apple Human Interface Guidelines https://developer.apple.com/design/human-interface-guidelines/
7. MDN Web Docs (HTTP / REST) https://developer.mozilla.org/en-US/docs/Web/HTTP
8. OWASP Mobile Security Project https://owasp.org/www-project-mobile-security/

****

Skill Enhancement Course

Semester –V

24BCASEC2	Quantitative Techniques	30 Hours

Course Overview:

This course is designed to build strong quantitative reasoning and problem-solving skills among BCA students. Spanning three comprehensive units, it progresses from foundational arithmetic, algebra, and logical reasoning, all essential for technical interviews, competitive examinations, and real-world analytical tasks.

Course Objectives:

Upon successful completion of this course, students will be able to:

- **CO1:** Solve problems involving number systems, percentages, ratios, and basic algebra with speed and accuracy.
- **CO2:** Apply algebraic concepts to practical problem scenarios.
- **CO3:** Apply logical reasoning and deductive thinking to solve puzzles and analytical problems.
- **CO4:** Develop examination readiness for aptitude tests used in campus placements and competitive exams.

Unit I: Number System and Arithmetic: [10 hours]

Number Systems: Natural numbers, Whole numbers, Integers, Rational and Irrational numbers, Real numbers, Prime numbers, Composite numbers, Co-prime numbers, Tests of divisibility: 2, 3, 4, 5, 6,
8, 9, 10, 11. Factors and Multiples: HCF and LCM methods and applications.

Number Series & Patterns: Arithmetic progressions and geometric progressions, Finding missing terms in number sequences, Square numbers, Cube numbers, Fibonacci series, Applications of number series in problem solving. BODMAS rule and examples.

Percentages, Profit & Loss: Concept of percentage conversion between fractions, decimals, and percentages, Percentage increase and decrease, successive percentage changes, Applications: population growth, income tax, discount calculations, Percentage error and related problems. Cost price (CP), Selling price (SP), Marked price (MP) relationships. Profit percentage and Loss percentage formula derivations, Discount and successive discounts, Problems involving dishonest traders, faulty weights, and trade discounts.

Ratio, Proportion & Averages: Simple ratio, Compound ratio, Inverse ratio. Direct proportion and Inverse proportion. Partnership problems-simple and compound partnerships, mixing and alligation, rule of alligation and weighted mixtures.

Averages: Arithmetic mean, Weighted average, Average of combined groups, Problems on age, speed, run-rate, and examinations.

Simple & Compound Interest: Simple Interest (SI)-formula, derivation, and applications. Compound Interest (CI)-annual, half-yearly, quarterly compounding.

UNIT II: Time, Work, and Speed: [10 hours]

Time & Work: Concept of work-rate of work, total work, Work done by individuals and groups, Fraction of work and number of days, Problems where persons join/leave midway.

Pipes & Cisterns: Inlet and outlet pipes-filling and emptying, Combined rate when multiple pipes

are open, Leak problems and time to fill/empty.

Time, Speed & Distance: Relationship: Speed = Distance / Time-unit conversions (km/h ↔ m/s), Average speed for different cases (equal distance, equal time), Relative speed-same direction vs. opposite direction. Problems on Trains: Crossing a pole, a platform, or a stationary/moving object.

Two trains crossing each other-same/opposite directions, Length and speed calculations.

Unit III: Statistics and Logical Reasoning : [10 hours]

Permutations & Combinations: Fundamental principle of counting, multiplication and addition principles, Factorial notation and its properties. Permutations: Permutation of n objects taken r at a time: nPr, Permutations with repetition allowed, Circular permutations and necklace arrangements,

Permutations with identical/repeated elements. Combinations: Combination formula: nCr - derivation and properties, Complementary combinations, division into groups, Applications in team selection, committee formation, geometry (collinear points).

Clocks: Angle between hands formula and calculation, Coincidence and right angle problems, Gaining and losing clocks-fast/slow clocks. *Calendars:* Odd days concept-Zeller's rule (conceptual), Day of the week for a given date, Leap year conditions and century rules.

Reference Books:

1) R. S. Aggarwal, Quantitative Aptitude for Competitive Examinations, S. Chand & Co., 7th Ed. 2020.
2) Arun Sharma, How to Prepare for Quantitative Aptitude for CAT, McGraw-Hill, 9th Ed.
3) R. V. Praveen, Quantitative Aptitude & Reasoning, PHI Learning, 3rd Ed. 2016.

Learning Outcomes:

On completing this course, students will demonstrate the ability to:

COU1: Compute numerical expressions and estimate values with efficiency.

COU2: Solve time-speed-distance, work, and mixture problems analytically.

COU3: Construct logical arguments and evaluate reasoning patterns.

COU4: Apply quantitative skills in technical aptitude assessments.

*****

Semester -VI

BCASEC3	Soft Skills	45 Hours

Course Objectives:

CO1: To enable students to develop logical thinking and structured approaches for solving real-life and workplace problems.

CO2: To enhance communication and teamwork skills required in professional environments.

CO3: To build awareness of ethical practices and responsible use of technology.

CO4: To prepare students for career opportunities through employability and interpersonal skills training.

Course Content:

Unit- I: Critical Thinking and Problem Framing [15 hours]

Understanding Critical Thinking, Problem Identification and Problem Framing, Logical Reasoning and Analytical Thinking, Decision-Making Techniques, Problem-Solving Frameworks for IT Professionals, Evaluating Information and Evidence.

Unit- II: Communication and Collaboration [15 hours]

Fundamentals of Professional Communication, Workplace Verbal Communication Skills, Written Communication for IT Professionals, Non-Verbal and Interpersonal Communication, Teamwork and Collaboration Skills, Digital Collaboration Tools and Etiquette, Conflict Management and Negotiation, Presentation and Public Speaking Skills.

Unit- III: Ethical and Responsible Use of Technology [15 hours]

Introduction to Ethics in Technology, Professional Codes of Ethics, Data Privacy and Data Protection, Cybersecurity Awareness and Safe Practices, Intellectual Property Rights (IPR) in IT, Responsible Use of Social Media and Digital Platforms, Emerging Ethical Issues in Technology, Ethical Decision-Making Frameworks.

Reference Books

1. A Practical Guide to Soft Skills, Communication, Psychology, and Ethics for Your Professional Life. Richard Almonte, Routledge, 2021.
2. Soft Skills: Enhancing Employability. M. S. Rao, I.K. International Publishing House, 2010.
3. Emotional Intelligence: Why It Can Matter More Than IQ. Daniel Goleman, Bantam Books, 2005.

****

Common Subjects

Semester -V

24BCA54	Internet of Things	45 Hours

Course Objectives:

CO1: Understand the fundamental concepts, architecture, enabling technologies, communication protocols, and domain-specific applications of the Internet of Things (IoT).

CO2: Analyze IoT system architectures, Machine-to-Machine (M2M) communication, software-defined networking (SDN), network function virtualization (NFV), and IoT system management using NETCONF-YANG.

CO3: Examine IoT data acquisition, storage, processing, analytics, and cloud-based computing platforms to support intelligent IoT applications.

CO4: Evaluate IoT-enabled business models and real-world IoT/IIoT applications in domains such as smart homes, smart cities, healthcare, agriculture, logistics, and connected vehicles.

Course Content

UNIT I: **[12 hours]**

Introduction to Internet of Things(IoT): Introduction, Definition and Characteristics of IoT, Physical Design of IoT, Things in IoT, IoT Protocols, Logical Design of IoT, IoT Functional Blocks, IoT Communication Models, IoT Communications APIs, IoT Enabling Technologies, Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Communication Protocols, Embedded Systems.

Domain Specific IoTs – Home Automation, Smart Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health and Life Style.

UNIT II: **[11 hours]**

IOT and M2M: Introduction: M2M, Difference between IoT and M2M, SDN and NFV for IOT, IOT System Management with NETCONF-YANG, Need for IOT Systems Management, Simple Network Management Protocol (SNMP), Network operator requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG.

UNIT III: **[11 hours]**

Data Acquiring, Organising, Processing and Analytics: Introduction, Data acquiring and storage, organizing the data, transactions, business processes, integration and enterprise systems, Analytics.
Data Collection, Storage and Computing Using a Cloud Platform: Cloud Computing Paradigm for Data Collection, Storage and computing, Everything as a service and cloud service models, IoT Cloud-Based Services Using the Xively, Nimbits and other platforms.

UNIT IV:

Business Models and Processes Using IoT: Introduction, Business models and business model Innovation, Value creation in the IoT, Business model scenarios for IoT.

IoT Case Studies: Introduction, IoT/IIot Applications in the Premises, Supply-Chain and customer monitoring, Connected car and its applications and services, IoT applications for homes, cities, environment-monitoring and agriculture.

Text Books:

1. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015.
2. Internet of Things: Architecture, Design Principles And Applications, Rajkamal, McGraw Hill Higher Education, 2017.

Reference Books:

1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017
2. Pethuru Raj and Anupama C. Raman , "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press, 2017.
3. Peter Waher, "Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3", Packt Publishing, 2018.

******

24BCA55	Cyber Security	45 Hours

Course Objectives:

CO1: Understand the fundamental concepts of cybercrime, information security, types of cybercrimes, and the legal framework governing cybercrime from Indian and global perspectives.

CO2: Analyze security threats and vulnerabilities associated with mobile and wireless devices, and evaluate appropriate security mechanisms and organizational policies to mitigate cyber risks.

CO3: Examine the impact of cyber threats on organizations, including cloud computing, social media, privacy, endpoint security, and incident handling, and recommend suitable cyber security best practices.

CO4: Investigate real-world cybercrime cases, digital forensic investigations, financial frauds, online scams, and digital signature-related crimes to understand practical applications of cyber security principles.

Course Content:

UNIT I: **[12 Hours]**

Introduction to Cybercrime: Introduction, Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Who are Cyber criminals? Classifications of Cybercrimes, Cybercrime: The Legal Perspectives, Cybercrimes: An Indian Perspective, Cybercrime and the Indian ITA 2000, A Global Perspective on Cybercrimes.

UNIT II: **[11 Hours]**

Cybercrime: Mobile and Wireless Device: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/Cell Phones, Mobile Devices: Security Implications for Organizations, Organizational Measures for Handling Mobile, Organizational Security Policies and Measures in Mobile Computing Era, Laptops.

UNIT III: **[11 Hours]**

Cyber security: Organizational Implications, Introduction, Cost of Cybercrimes and IPR Issues: Lessons for Organizations, Web Threats for Organizations: The Evils and Perils, Security and Privacy Implications from Cloud Computing, Social Media Marketing: Security Risks and Perils for Organizations, Social Computing and the Associated Challenges for Organizations, Protecting People's Privacy in the Organization, Organizational Guidelines for Internet Usage, Safe Computing Guidelines and Computer Usage Policy, Incident Handling: An Essential Component of Cybersecurity, Forensics Best Practices for Organizations, Media and Asset Protection: Best Practices for Organizations, Importance of Endpoint Security in Organizations.

UNIT IV: **[11 Hours]**

Cybercrime: **Illustrations, Examples and Mini-Cases**:· Introduction, Real-Life Examples, Mini-Cases, Illustrations of Financial Frauds in Cyber Domain, Digital Signature-Related Crime Scenarios, Digital Forensics Case llustrations, Online Scams

Text book:

1. Nina Godbole & Sunit Belapure (2011). Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives. Wiley, India.

Reference Book:

1. Rao, U. & Nayak, P. (2014). Cyber Security. Cengage Learning. Moeti J. (2021). Cybersecurity for Beginners. Amazon Digital Services (for simple reading-level coverage).
2. Chirag Shah (2018). A Hands-On Introduction to Cybersecurity. Wiley.

*****

Common Subjects

Semester -VI

24BCA63	Research Methodology	45 Hours

Course Objectives

CO1: Understand the meaning, significance, and types of research, and apply the research process in computing contexts.

CO2: Identify and evaluate research problems through systematic literature review using academic search engines and databases.

CO3: Apply descriptive and inferential statistical methods for data collection, analysis, and interpretation.

CO4: Develop skills to write and present structured research reports and papers following accepted academic conventions.

CO5: Demonstrate awareness of research ethics, intellectual property rights, plagiarism, and publication standards.

CO6: Utilize ICT tools including reference managers, statistical software, and LaTeX for research documentation.

Prerequisites: Mathematics for Computing, Programming Using Python / C, Basic Knowledge of Spreadsheets and Data Handling.

UNIT I: **[8 hours]**

Research Fundamentals

Meaning and significance of research; requirements and characteristics of good research; types of research — basic/fundamental, applied, analytical, conceptual, empirical, experimental, non-experimental, prospective, retrospective, exploratory/descriptive, qualitative, quantitative, and mixed methods. Research process: problem identification, literature review, hypothesis formulation, data collection, analysis, and reporting. Induction and deduction in research. Introduction to research tools and instruments.

UNIT II: **[7 hours]**

Problem Identification and Literature Review

Choosing an appropriate problem area relevant to computing; identifying sources of research articles journals, conference proceedings, and technical reports. Academic search engines: Google Scholar, Scopus, Web of Science, DBLP, IEEE Xplore, and ACM Digital Library. Literature review: purpose, process, and critical evaluation; stating and evaluating the research problem; identifying techniques, methodologies, and state of the art. Journal quality metrics: CiteScore, SCImago Journal Rank (SJR), Source Normalized Impact per Paper (SNIP). Conference rankings: CORE, Qualis. Writing a research proposal.

UNIT III: **[6 hours]**

Research Design and Data Collection

Meaning and need for research design; types — exploratory, descriptive, and causal. Quantitative vs. qualitative research designs. Identifying variables: concept vs. variable, types of variables, types of measurement scales (nominal, ordinal, interval, ratio). Sources of data — primary and secondary. Methods of data collection: surveys, interviews, observation, and experiments. Sampling concepts: population, sample, sampling frame; sampling methods — simple random, purposive, convenience, snowball, and stratified sampling. Attitudinal scales: Likert, Semantic Differential.

UNIT IV: **[10 hours]**

Data Analytics and Statistics

Exploring and organizing datasets; data pre-processing: cleaning and transformation. Descriptive statistics: measures of central tendency (mean, median, mode), dispersion (range, variance, standard deviation) and correlation (correlation and covariance). Inferential statistics: hypothesis testing - null and alternate hypothesis, type I and type II errors, t-test, chi-square test, ANOVA. Regression and interpretation of results. Probability: conditional probability, Bayes' theorem, probability distributions. Visualisation: histograms, bar plots, scatter plots. Use of Excel and SPSS.

UNIT V: **[8 hours]**

Presenting Research Outcomes

Essential elements of a research paper: title, abstract, keywords, introduction, related work, methodology, results, discussion, conclusion, and references. Structure of an abstract. Analyzing and interpreting data; limitations; conclusions. Citation styles: APA, IEEE, MLA, Chicago. Document preparation using LaTeX: papers, thesis, reports, BibTeX.

UNIT VI: **[6 hours]**

Publication Process and Research Ethics

Where to publish: journals and conferences; peer review — single-blind and double-blind. Scientometric analysis: citation metrics, h-index, impact factor. Plagiarism: forms, consequences, and detection tools; similarity index. Professional code of ethics: informed consent, confidentiality, avoiding bias. Intellectual property rights: patents, copyrights, and trademarks. Fraud and misconduct in science.

Text Books

1. Thomas, C. G. (2021). Research Methodology and Scientific Writing, 2nd Ed. Springer.
2. Leedy, P. D., & Ormrod, J. E. (2016). Practical Research: Planning and Design, 11th Editon. Pearson.

Reference Books

1. Ghezzi, C. Being a Researcher: An Informatics Perspective. Springer.
2. Kumar, R. (2009). Research Methodology: A Step-by-Step Guide for Beginners, 2nd Ed. Pearson Education.
3. Locharoenrat, K. (2018). Research Methodologies for Beginners. PAN Stanford Publication.
4. Cooper, D. R., & Schindler, P. S. (2009). Business Research Methods, 9th Ed. Tata McGraw Hill.

Web Resources

1. Google Scholar: https://scholar.google.com
2. IEEE Xplore Digital Library: https://ieeexplore.ieee.org
3. ACM Digital Library: https://dl.acm.org
4. Scopus: https://www.scopus.com
5. NPTEL - Research Methodology Course: https://nptel.ac.in

*****

24BCA64	Generative AI	45 Hours

Course Learning Objectives:

CO1: Explain AI/ML/GenAI distinctions; write Python to process a dataset; classify an AI application into its ML paradigm.

CO2: Describe a neural network by analogy; explain learning as weight adjustment; train a no-code classifier with Teachable Machine.

CO3: Explain tokenisation and Self-Attention by analogy; interpret embeddings; make live API calls to an LLM in Python.

CO4: Design chain-of-thought prompts; build a RAG pipeline; evaluate GenAI outputs for risks; discuss ethical responsibilities.

Pre requirements:

1. Knowledge of Python/Matlab Programming.
2. Knowledge of basic Machine learning.

Unit I: How AI Thinks - Foundations & Concepts: [12 Hours]

AI vs. ML vs. GenAI: Understanding the difference between traditional software, Machine Learning, and Generative AI through real-world analogies. **Python Basics:** Variables, data types, loops, functions, and File I/O (csv, json) as practical tools for AI workflows. **Data with Pandas:** Loading, inspecting, and cleaning datasets understanding what a model learns from. **Classical ML Intuition:** Supervised vs. unsupervised vs. reinforcement learning (concept only); Scikit-learn.

Unit II: Neural Networks - The Brain Behind GenAI: [11 Hours]

The Neuron & Layers: Perceptron as a decision unit; input, hidden, and output layers explained as levels of abstraction — no calculus. How Models Learn: Weight adjustment by feedback (Gradient Descent as walking downhill); overfitting vs. underfitting with loss curves. Activation Functions: ReLU and Sigmoid — why they exist (allow/block signal) without mathematical derivation. CNN vs. RNN Intuition: CNNs for spatial patterns, RNNs for sequences; why the vanishing gradient problem led to the Transformer.

Unit III: Large Language Models (LLMs) & Transformers: [11 Hours]

Tokenisation: How raw text becomes numerical IDs the model processes; vocabulary, special tokens, subword splitting. The Transformer & Self-Attention: How a model weighs word importance by context — "The bank of the river" explained without vector math. Embeddings: Meaning encoded as distance in vector space — the king − man + woman ≈ queen analogy; semantic similarity. Pre-training vs. Fine-

tuning: How models learn from the internet first, then specialise; prompt engineering as a third strategy. Hands-on API Lab: Python API calls to OpenAI / Gemini — send a prompt, parse and display the response.

Unit IV: Building with GenAI — Prompt to Product: [11 Hours]

Prompt Engineering: Zero-shot, few-shot, chain-of-thought, role prompting, and structured output formatting with real examples. RAG (Retrieval-Augmented Generation): Connecting an LLM to a private document; vector databases, chunking, and retrieval — build a PDF Q&A bot with LangChain. Agentic AI: LLM + tools + planning loop; giving a model access to web search or calculator for multi-step autonomous tasks. Evaluating GenAI Output: Hallucination checklist, bias identification, and factual accuracy review — structured evaluation framework. AI Ethics & Responsibility: Privacy, copyright, deepfakes, misinformation, and the environmental cost of large-scale AI.

References

1. *Python Crash Course* by Eric Matthes — No Starch Press.
2. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* by Aurélien Géron — O'Reilly.
3. *Build a Large Language Model (From Scratch)* by Sebastian Raschka.
4. *Natural Language Processing with Transformers* by Lewis Tunstall, Leandro von Werra.
5. *AI for Everyone* — Andrew Ng, Coursera (free online course).
6. *Attention Is All You Need* — Vaswani et al., 2017 (arxiv.org/abs/1706.03762).

*****