MANGALORE UNIVERSITY

State Education Policy – 2024
[SEP-2024]

CURRICULUM

FOR

BCA-ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

MANGALORE UNIVERSITY

Suggested programme structure for the Undergraduate Programmes

[BCA, BCA (A.I & M.I), BCA (D.A)]

Sem	Course 1	Course 2	Course 3	Elective / Optional	Course	Language	Compulsory	Total Credit	Total Working hour
I	5 (3T+2P)	5 (3T+2P)	5 T			3+3	2	23	4+4+4+4+5+4+ 4+2=31
II	5 (3T+2P)	5 (3T+2P)	5T			3+3	2	23	4+4+4+4+5+4+ 4+2=31
III	5 (3T+2P)	5 (3T+2P)	5T	2		3+3		23	4+4+4+4+5+4+ 4+2=31
IV	5 (3T+2P)	5 (3T+2P)	5T	2		3+3	2	25	4+4+4+4+5+2+ 4+4+2=33
V	8[(2x3T)+2P]	8[(2x3T)+2P]	8[(2x3T)+2P]				2	26	3+3+3+3+3+3+ 4+4+4+2=32
VI	3T+2P	3T+2P	Research Methodology + Project Work- 12					22	3+3+4+4+22=36
							Total Credits	**142**	

Note:

- Course1 and Course2: I to IV Semester: Theory 3 credits = 4 contact hours and Practical 2 credits=4 contact hours
- Course 3: I to IV Semester: Theory 5 credits = 5 contact hours
- Course1, Course2 and Course3: V and VI Semester: Theory 3 credit=3 contact hours and Practical 2 credits =4 contact hours
- Elective/Optional: 2 credits=2 contact hours
- Languages: 3 credits=4 contact hours
- Compulsory: 2 credits=2 contact hours

Semester VI								
Sl. No	Course Code	Title of the Course	Category of Courses	Teaching Hours per Week	SEE	IA	Total Marks	Credits
1	BCAAIMS601	Artificial Neural Network	Core	3	80	20	100	3
2	BCAAIMS602	Computer Vision	Core	3	80	20	100	3
~~3~~	BCAAIPS603	Artificial Neural Network Lab	Practical	~~4~~	40	10	50	2
4	BCAAIPS604	Computer Vision Lab	Practical	4	40	10	50	2
5	BCAAIPS605	Research Methodology and Project Work	Project Work	22	400	100	500	12
			Total	**36**	**620**	**180**	**800**	**22**

SEMESTER VI

Program Name	**BCA-AIML**	Semester	**VI**
Course Title	**Artificial Neural Network(THEORY)**		
Course Code:	**BCAAIMS601**	No. of Credits	**03**
Contact hours	**39 Hours**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

Unit	**Description**	**Hours**
I	Introduction to Artificial Intelligence, Machine Learning, and Deep Learning. Types of Learning: Supervised and Unsupervised. Introduction to Artificial Neural Networks: Fundamentals of Neural Networks, Biological Neuron and Artificial Neuron. History and Evolution of Neural Networks. Characteristics and Applications of Neural Networks. Single Layer Perceptron: Basic Architecture, McCulloch-Pitts Neuron Model. Activation Functions: Binary Step, Linear, Sigmoid, Tanh, ReLU and Softmax. Limitations of Single Layer Perceptron.	10
II	Introduction to Multi-Layer Perceptron (MLP). Multi-Layer Neural Networks and Training: **Multi-Layer Perceptron:** Architecture of Multi-Layer Perceptron. Feed Forward Neural Networks. Forward and Backward Propagation. Error Calculation and Weight Update Mechanism.	09

III	**Training Neural Networks:** Training and Testing Datasets. Hyperparameters in Neural Networks. Batch, Stochastic and Mini-Batch Gradient Descent. Epochs and Iterations. Learning Rate and Momentum. **Optimization and Regularization:** Overfitting and Underfitting. Cross Validation. Regularization Techniques: Dropout and Early Stopping. Model Evaluation Metrics: Accuracy, Precision, Recall, F1-Score and Confusion Matrix.	10
IV	**Deep Neural Networks:** Introduction to Deep Learning. Deep Feed Forward Networks. Hidden Layers and Feature Extraction. Vanishing and Exploding Gradient Problems. Advanced Neural Networks and Applications: **Recurrent Neural Networks:** Introduction to Sequential Data. Recurrent Neural Networks (RNN). Working of RNN. Limitations of RNN. Introduction to LSTM and GRU. Applications in Text Processing. **Emerging Neural Network Applications:** Neural Networks in Natural Language Processing (NLP), Recommendation Systems, Generative AI.	10

TEXT BOOKS:

1. Charu C. Aggarwal "Neural Networks and Deep Learning" –, Springer.
2. Ian Goodfellow, Yoshua Bengio, Aaron Courville "Deep Learning" – MIT Press. 2016
3. Laurene V. Fausett "Fundamentals of Neural Networks" Prentice Hall. 1994

REFERENCE BOOKS:

1. B. Yegnanarayana "Artificial Neural Networks" – PHI Learning 1999.

2. Simon Haykin., “Neural Networks: A Comprehensive Foundation” – Prentice Hall.
3. Aurélien Géron., “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow” 3rd Edition – O’Reilly Media.
4. François Chollet., “Deep Learning with Python” 2nd Edition 2021 – Manning Publications.

Pedagogy: Lecture/ PPT/ Videos/ Animations/ Quizzes/ Mini Projects/Group Discussion/ Experiential Learning.

Program Name	**BCA-AIML**	Semester	**VI**
Course Title	**Computer Vision(THEORY)**		
Course Code:	**BCAAIMS604**	No.of Credits	**03**
Contact hours	**39 Hours**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**20**	Summative Assessment Marks	**80**

1. Course Outcomes: After completing this course, the student must demonstrate the knowledge and ability to:
2. Recognize and describe both the theoretical and practical aspects of image formation and computing with images.
3. Understand the basics of 2D and 3D Computer Vision and Connect issues from Computer Vision to Human Vision.
4. Become familiar with the significant technical approaches in computer vision, including registration, alignment, and image matching.
5. Get exposure to advanced concepts leading to object categorization and segmentation in images.
6. Build computer vision applications.
7. Implement computer vision applications using OpenCV, integrating machine learning algorithms for advanced tasks**.**

UNIT – I	**9 Hours**
Fundamentals of Image Processing and Computer Vision: Brief history of Computer Vision Image formation: Geometric primitives,2D transformations, 3D transformations, 3D rotations, 3D to 2D projections, Lens distortions Photometric image formation- Lighting, Reflectance and shading. Camera projections: Sampling and aliasing	Book1: Page 29 to 91.
UNIT – II	**10 Hours**
Image processing Local operators -Pixel transforms, Colour transforms,	Book1: Page 97

Compositing and matting, Histogram equalization Neighbourhood operators- Linear filtering, Non-linear filtering, Morphology, Distance transforms, Connected components. Pyramids and wavelets: Interpolation and decimation, multi-resolution representations, Wavelets.	to 160
UNIT – III	**10 Hours**
Image Features and Recognition Techniques: Image Features, Image Descriptors, and Features, Texture Descriptors, colour Features, Edge Detection, Object Boundary and Shape Representations, Interest or Corner Point Detectors, Histogram of Oriented Gradients, Scale Invariant Feature Transform.	Book1: Page 203 to 253, Study Material
UNIT – IV	**10 Hours**
OpenCV and Applications of Computer Vision: Introduction to OpenCV and Applications of Computer Vision: What Is OpenCV? The Origin of OpenCV, Downloading and Installing OpenCV, First Program—Display a Picture, Second Program—AVI Video, A Simple Transformation OpenCV Primitive Data Types, CvMat Matrix Structure. Machine Learning Algorithms and their Applications in Medical, Image Segmentation, Motion Estimation and Object Tracking, Face and Facial Expression Recognition, Gesture Recognition, Image Fusion, Programming Examples.	Book 3
TEXTBOOKS: 1. Computer Vision: Algorithms and Applications, Richard Szeliski, February 27, 2010 2. Computer Vision and Image Processing Manas Kamal Bhuyan CRC Press 1st Edition 2020 3. Gary Bradski and Adrian Kaehler. “Learning OpenCV: Computer Vision with the OpenCV Library,” First Edition, O’Reilly Media, 2008. 4. Computer Vision: A Modern Approach David A. Forsyth, Jean Ponce Pearson 2nd 2012 5. Machine Vision R. Jain, R. Kasturi, and B. G. Schunk McGrawHill 1995	

Program Name	**BCA-AIML**	Semester	**VI**
Course Title	**Artificial Neural Networks Lab**		
Course Code:	**BCAAIPS603**	No.of Credits	**02**
Contact hours	**52 Hours**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART-A

1. Write a Python program to implement a single-layer Perceptron to determine attendance eligibility of students. Train the network using the Perceptron learning rule and test it with new attendance values. Display the eligibility status for each test input.
2. Write a Python program to implement a single-layer Perceptron to realize the OR logic gate. Display the final weights, bias, and predicted outputs.
3. Write a Python program to implement the LMS algorithm for identifying an unknown system defined by $d(n) = 0.5x(n) + 0.3x(n-1)$ and plot the convergence.
4. Write a Python program to implement a Competitive Learning Neural Network for clustering two-dimensional input patterns.
5. Write a Python program to train a single-layer perceptron using given input patterns and plot the learning curve.
6. Using the learning rate annealing technique on $f(x) = x^2 + 4x + 4$, plot a graph showing learning the convergence.

PART-B

1. Write a Python program to implement the Perceptron learning algorithm with a step activation function and train it to classify the AND, OR, and XOR logic gates and plot the results.
2. Write a Python program to design and train a Multi-Layer Perceptron using TensorFlow/Keras for a simple classification problem.
3. Write a Python program to implement a single-layer neural network for the OR gate. Train the network using different activation functions. Compute the Mean Squared Error (MSE), and compare the performance of the activation functions
4. Implement a Hopfield Network in Python to store and recall multiple binary patterns. Test it with noisy inputs and visualize the energy convergence over iterations.
5. Write a Python program to implement a Self-Organizing Map (SOM) for clustering two-dimensional input data. Display the final neuron weight vectors representing clusters.
6. Write a Python program to demonstrate the Hopfield Network approach for solving a simple optimization problem such as the Traveling Salesman Problem (TSP) with a small number of cities.

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	PART-A Writing: 5 Marks; Logic:5 marks Execution: 5 Marks	15 Marks
Program-2	PART-B Writing: 5 Marks; Logic:10 marks Execution: 5 Marks	20 Marks
Practical Record		05 Marks
Total		40Marks

Program Name	**BCA-AIML**	Semester	**VI**
Course Title	**Computer Vision Lab**		
Course Code:	**BCAAIPS604**	No.of Credits	**02**
Contact hours	**52 Hours**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**10**	Summative Assessment Marks	**40**

PART - A

1. Write a Python program using the OpenCV library to upload a digital image and perform operations related to image brightness enhancement, contrast adjustment, illumination variation. The program should display both the original and processed images.
2. Write a Python program to perform image translation, rotation, scaling, and shearing using OpenCV.
3. Write a Python program to Implement a simple image reconstruction technique from multiple projections using Radon Transform concepts.
4. Write a Python program to perform point processing operations such as image negative, thresholding, brightness enhancement, and contrast stretching.
5. Write a Python program to Implement smoothing and sharpening filters such as Mean Filter, Gaussian Filter and Median Filter
6. Write a Python program to Implement RGB←→HSV and RGB←→Grayscale image conversions and perform color enhancement and histogram equalization.

PART -B

1. Write a Python program to detect edges using Sobel and Prewitt edge detection algorithms.
2. Write a Python program to Develop a simple image classification system using Linear Regression.
3. Write a Python program to Build and train a CNN model using TensorFlow/Keras for handwritten digit or object classification.
4. Write a Python program to Create applications using OpenCV for Face detection and facial expression recognition.
5. Write a Python program to upload an image from the computer using File open dialog box and Implement Harris Corner Detection and extract basic image features.
6. Write a Python program to Create a simple real-time application such as Webcam edge detection

Evaluation Scheme for Lab Examination:

Assessment Criteria		
Program-1	PART-A Writing: 5 Marks; Logic:5 marks Execution: 5 Marks	15 Marks
Program-2	PART-B Writing: 5 Marks; Logic:10 marks Execution: 5 Marks	20 Marks
Practical Record		05 Marks
Total		40Marks

Program Name	**BCA-AIML**	Semester	**VI**
Course Title	**Research Methodology and Project Work**		
Course Code:	**BCAAIPS605**	No. of Credits	**12**
Contact hours	**22 Hours/Week**	Duration of SEA/Exam	**3 Hours**
Formative Assessment Marks	**100**	Summative Assessment Marks	**400**

PROJECT GUIDELINES

Preamble: The project work has been incorporated into the BCA programme to provide students with practical exposure to the design, development, and implementation of modern computing solutions. The primary objective of the project work is to enable students to understand and apply the principles of software engineering, problem-solving, innovation, and emerging technologies in real-world scenarios. During the development of the project, students are expected to actively participate in all phases of the Software Development Life Cycle (SDLC), including problem identification, requirements analysis, system design, development/coding, testing, deployment, documentation, and maintenance, with emphasis on building reliable, scalable, and user-oriented systems.

In view of rapid advancements in technology, students are encouraged to undertake projects in contemporary domains such as Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Internet of Things (IoT), Robotics, Drone Technology, Data Analytics, Cloud Computing, Cyber Security, Mobile Applications, and Web Technologies. AI/ML/DL projects may focus on intelligent systems, predictive analytics, automation, computer vision, natural language processing, or recommendation systems. IoT-based projects may involve sensor integration, smart automation, real-time monitoring, and connected devices. Robotics and Drone projects may include autonomous systems, embedded controllers, navigation, surveillance, industrial automation, or smart agriculture applications. Students may

also integrate hardware and software components wherever applicable to develop interdisciplinary and innovative solutions.

Since the project work spans the entire final semester, students are encouraged to select projects addressing the needs of industry, research organizations, startups, government agencies, or socially relevant real-life problems suggested by the faculty in charge of BCA project work in the institution. The selected project topic must be meaningful, analytical, application-oriented, and technically significant. It should demonstrate originality, creativity, and substantial development effort based on the chosen theme.

Use of ready-made, copied, off-the-shelf, pirated, or previously implemented work is strictly prohibited. Any form of plagiarism, fabrication, or use of unfair means shall result in rejection of the project work. All stages of project development must be carried out in a systematic and time-bound manner, and equal participation and contribution from all team members are expected throughout the project lifecycle.

GENERAL GUIDELINES TO THE INSTITUTIONS

- The Project Work Calendar shall be announced before the commencement of the VI Semester.
- The calendar shall include tentative schedules for:
    - Project Proposal Submission
    - Project Approval
    - Synopsis Submission
    - Problem Analysis Document
    - System Design Document
    - Database Design /Detailed Design
    - Coding and Testing
    - Project Demonstration and Internal Viva-Voce
    - Final Report Submission
- A project team shall consist of a maximum of four students. A project supervisor (from among the faculties of the institution) must be assigned for each team, before the commencement of the project work.
- Students shall undertake projects based on real-life, application-oriented, or research problems relevant to industry, society, or emerging technologies.
- Projects may be carried out within the institution or in collaboration with industries, research laboratories, startups, or

software companies, subject to approval by the faculty in charge/ project supervisor and the Head of the Department.

- The faculty in charge/ Project Supervisor shall ensure that industry-based projects are genuine, original, and academically relevant.
- Institutions shall conduct at least two Internal Assessment evaluations during the project period.
- Internal Assessment may include:
  - Project Presentations
  - Document Verification
  - Coding Demonstration
  - Viva-Voce
  - Group Discussion
  - Progress Review
- Internal Assessments shall evaluate the students' technical understanding, contribution, implementation skills, documentation, and overall project progress.

EVALUATION OF THE PROJECT

- Project evaluation shall be conducted jointly by one Internal Examiner and one External Examiner appointed by the University during the End Semester Examination.
- The examiners shall understand and follow the evaluation procedure, guidelines and marking scheme before commencement of the evaluation process.
- Each student in the project team shall individually present and explain their contribution to the project work.
- Evaluation shall include:
  - Project Demonstration
  - Viva-Voce
  - Technical Presentation
  - Verification of Documentation
  - Assessment of Individual Contribution
- Students shall demonstrate the working of the developed system, model, application, device, or prototype.
- For projects in domains such as Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Internet of Things (IoT), Robotics, Drone Technology, Data Analytics, and Embedded Systems, evaluation may include:
  - Dataset Preparation and Usage
  - Model Training and Testing

    - Accuracy and Performance Metrics
    - Hardware Integration
    - Sensor/Device Connectivity
    - Real-Time Monitoring or Automation
    - Simulation or Prototype Demonstration
    - Innovation and Practical Relevance
- AI/ML/DL projects shall include proper validation of models using suitable evaluation metrics such as accuracy, precision, recall, F1-score, confusion matrix, or similar performance measures wherever applicable.
- IoT, Robotics, and Embedded System projects shall include demonstration of hardware functionality, communication protocols, device integration, and system reliability.
- If live demonstration is not possible due to confidentiality, copyright, licensing, cloud restrictions, or industry policies, students may present:
    - Screenshots
    - Recorded Demonstrations
    - Simulation Results
    - Deployment Reports
    - Performance Analysis
    - Technical Documentation
- In such cases, students shall submit valid supporting documents or authorization letters issued by the concerned organization or company.
- The evaluation shall primarily focus on:
    - Originality of the Work
    - Problem Identification and Solution Design
    - Technical Knowledge and Understanding
    - Implementation Skills
    - Innovation and Creativity
    - Testing and Validation
    - Documentation Quality
    - Practical Utility and Social/Industrial Relevance
- Examiners shall verify whether appropriate development methodologies and engineering practices have been followed throughout the project lifecycle.
- For software-based projects, adherence to Software Development Life Cycle (SDLC) principles shall be examined.
- For AI, ML, DL, IoT, Robotics, and interdisciplinary projects, suitable design, development, testing, deployment, and validation

practices relevant to the domain shall be considered during evaluation.

- Project evaluation shall be fair, objective, and based on the scope, complexity, originality, and outcomes of the project work.

SCHEME OF VALUATION and Marks Distribution

	Particulars	**Marks**
Internal Assessment		
	Progressive assessment for two times with award of 50 marks each time	100
Project Report Valuation (During End Semester Examination):		300
Viva-Voce: 100 marks		
1	Live Demonstration (Software execution) or Dry runs (Presentation of authentic screenshots or captured videos may be used to walk through complete scenarios)-consistency and completeness	80
2	Question and Answer (Oral)	20
Grand Total Marks		500

FORMAT OF PROJECT SYNOPSIS

The Project Synopsis is a concise summary of the proposed project work. It shall clearly present the objectives, methodology, technologies used, implementation plan, and expected outcomes of the project. The synopsis shall not exceed 20 pages.

The synopsis shall include the following:

1. Title of the Project
2. Introduction, Objectives, and Scope
    - Problem Statement
    - Objectives of the Project
    - Scope and Expected Outcomes
3. Project Domain/Category
    - Software Application
    - Web/Mobile Application
    - Database Management System
    - Client-Server System
    - Networking
    - Cloud Computing
    - Artificial Intelligence (AI)
    - Machine Learning (ML)

- Deep Learning (DL)
- Internet of Things (IoT)
- Robotics/Drone Technology
- Data Analytics
- Embedded Systems
- Simulation/Game Development
- Cyber Security
- Any other relevant category

4. Tools, Platforms, and Requirements
   - Programming Languages
   - Frameworks/Libraries
   - Development Tools and Platforms
   - Hardware Requirements
   - Software Requirements
   - Dataset Requirements (for AI/ML/DL projects)
   - Sensors/Controllers/Communication Modules (for IoT/Robotics projects)
   - Cloud or Deployment Requirements, if any
5. System Analysis and Design
   - Architecture Diagram
   - DFD/UML/Class Diagrams/ER Diagrams/System Flowcharts as applicable
   - Database Design
   - Hardware Architecture (for IoT/Embedded/Robotics projects)
   - Model Architecture (for AI/ML/DL projects)
6. Project Structure and Methodology
   - Module-wise Description
   - Functional Requirements
   - Data Structures/Algorithms Used
   - Process Flow and Logic
   - Training and Testing Methodology (for AI/ML/DL projects)
   - Communication Protocols and Device Interaction (for IoT projects)
   - Simulation/Prototype Details, if applicable
7. Testing and Validation
   - Testing Methods and Tools
   - Performance Evaluation Metrics
   - Accuracy/Precision/Recall or other relevant measures for AI/ML projects
   - Reliability and Functional Testing for IoT/Robotics projects

8. Project Type
    - Industry Defined
    - Client Defined
    - User Defined
    - Research/Innovation Based
      (Mention the name and address of the Industry/Client/Organization, if applicable.)
9. Limitations of the Project
10. Future Scope and Enhancements
11. References
    - Books
    - Research Papers
    - Websites/APIs/Datasets/Documents referred to during project planning.

ORGANISATION OF THE DISSERTATION

The Dissertation Report shall be organized into clearly defined chapters and sections. The report shall present the complete development process, implementation details, testing, results, and conclusions of the project work. The structure may be adapted suitably depending on the project domain such as Software Development, AI, ML, DL, IoT, Robotics, Drone Technology, Embedded Systems, Data Analytics, or related interdisciplinary areas.

Sequence of Items in the Dissertation Report

- Cover Page (Hard Bound Cover)
- Title Page
- Certificate from the Institution
- Certificate from the Company/Industry (if applicable)
- Declaration by Students
- Acknowledgement
- Table of Contents
- List of Figures
- List of Tables
- List of Abbreviations and Acronyms
- Chapters
- Bibliography / References
- Appendices (if any)

CHAPTERS

1. Introduction

a. Project Overview

- Project Title
- Project Domain/Category
- Problem Statement
- Overview of the Proposed System

b. Background Study
- Organization/Industry Profile (if applicable)
- Existing System and Limitations
- Need for the Proposed System

c. Objectives
- Project Objectives
- Expected Outcomes

d. Scope of the Project

e. System Architecture
- Overall Architecture Diagram
- Workflow/Process Flow

f. End Users and Applications

g. Tools and Technologies Used
- Software Tools
- Programming Languages
- Frameworks/Libraries
- Hardware Components (if applicable)

h. Deployment Requirements
- Hardware Requirements
- Software Requirements
- Network/Cloud Requirements

2. Software Requirement Specification (SRS)

a. Introduction to SRS

b. Overall Description
- Product Perspective
- Product Features and Functions
- User Characteristics
- Assumptions and Constraints

c. Functional Requirements
- Module-wise Functional Requirements

d. Non-Functional Requirements
- Performance
- Security
- Reliability
- Scalability
- Usability

e. Special Requirements
- Dataset Requirements (AI/ML/DL Projects)
- Sensor/Device Requirements (IoT/Robotics Projects)
- Communication Protocols
- Cloud/API Requirements

3. System Design
- Introduction to System Design
- Functional Design
- Functional Decomposition
- Module Description

c. Design Diagrams
- Context Diagram
- Data Flow Diagrams (DFD)
- UML/Class Diagrams
- Use Case Diagrams
- Sequence/Activity Diagrams (if applicable)

d. Architecture Design
- Software Architecture
- Hardware Architecture (for IoT/Embedded/Robotics Projects)
- Model Architecture (for AI/ML/DL Projects)

4. Database Design / Data Design

a. Introduction

b. Database/Data Structure Design (For Web Application/ App Development Projects)
- Table Definitions
- ER Diagram
- Schema Design

c. Data Collection and Preprocessing (For AI/ML/DL Projects)
- Dataset Description
- Data Cleaning
- Feature Engineering

d. Sensor/Data Acquisition Design(For IoT/Robotics Projects)

5. Detailed Design and Methodology

a. Module-wise Design

For each module:
- Inputs
- Process Logic/Algorithms
- Outputs
- Interfaces

- Implementation Details

b. Algorithms and Techniques Used
- AI/ML Algorithms
- Deep Learning Models
- Data Processing Techniques
- Control Logic/Automation Logic

c. Communication and Integration
- API Integration
- Device Communication
- Cloud Connectivity

d. Structure Charts / Workflow

6. Implementation and Code Description

a. Coding Structure

b. Important Code Modules
- Authentication/Authorization
- Database Operations
- Data Processing
- Search and Retrieval
- API/Device Integration
- Parameter Passing
- Error Handling
- Backup and Recovery

c. Model Training and Deployment
(For AI/ML/DL Projects)

d. Embedded/Hardware Programming
(For IoT/Robotics Projects)

e. Internal Documentation

7. User Interface and Outputs

a. User Interface Screens
- Login Page
- Dashboard/Home Page
- Data Entry and Management Screens
- Reports and Visualizations

b. System Outputs
- Reports
- Graphs/Charts
- Prediction Results
- Alerts/Notifications

c. Prototype Demonstration (For IoT/Robotics/Drone Projects)

8. Testing and Validation

a. Introduction to Testing

b. Testing Methods

- Unit Testing
- Integration Testing
- System Testing
- Performance Testing

c. AI/ML/DL Validation

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- Loss/Performance Analysis

d. IoT/Robotics Testing

- Hardware Testing
- Sensor Validation
- Connectivity Testing
- Reliability Testing

e. Test Reports and Results

9. Results and Discussion

- Results Obtained
- Performance Analysis
- Comparison with Existing System
- Interpretation of Outputs

10. Conclusion and Future Scope

- Conclusion
- Limitations
- Future Enhancements and Scope

REFERENCES

- Books
- Research Papers
- Journals
- Websites
- APIs/Datasets
- Technical Documentation

APPENDICES (if any)

- Source Code
- User Manual
- Sample Outputs
- Installation Guide
- Published Papers/Certificates (if applicable)

Do not include any header or footer on any page of the report other than foot notes. Only page numbers should be mentioned at the bottom center of each page. 'n+1' copies of dissertation should be prepared by n number of candidates in the team.

DISSERTATION FORMAT

The dissertation shall be prepared in a clear, professional, and standardized format suitable for project works in Software Development, Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), Internet of Things (IoT), Robotics, Drone Technology, Embedded Systems, Data Analytics, and related domains.

1. PAPER AND PRINTING FORMAT

1.1 Paper Quality

- The dissertation shall be printed on white A4 bond paper of minimum 70 GSM quality.
- Paper whiteness shall preferably be 95% or above.

1.2 Paper Size

- Standard A4 Size:
  - Height: 297 mm
  - Width: 210 mm

1.3 Printing and Font Specifications

- The dissertation shall be printed using LaserJet or Inkjet printers.
- Text shall be prepared using standard word processing software.

Recommended Formatting

- Font Style: Times New Roman/Bookman Old Style/ Book Antiqua / Adobe Garamond Pro
- Font Size:
  - Main Text: 12 pt
  - Chapter Titles: 18 pt Bold
  - Section Headings: 16 pt Bold
  - Subsection Headings: 14 pt Bold
- Line Spacing: 1.5

- Text Alignment: Justified

1.4 Margins

The following margins shall be maintained:

- Top Margin: 0.5 inch
- Bottom Margin: 0.5 inch
- Left Margin: 1 inch
- Right Margin: 0.75 inch

1.5 Pagination

- Page numbers shall be placed at the centre of the footer.
- Preliminary pages shall use lower-case Roman numerals (i, ii, iii, iv, …).
- Main chapters shall use Arabic numerals starting from 1.

1.6 Length of Dissertation

- The dissertation shall normally contain a minimum of 60 pages excluding:
  - References
  - Appendices
  - Source Code Listings
  - Annexures
  - Questionnaires
  - Datasheets
- AI/ML/DL, IoT, Robotics, and Embedded projects may additionally include:
  - Dataset Samples
  - Model Architectures
  - Sensor Specifications
  - Hardware Schematics
  - Experimental Results
  - Performance Metrics

2. PARAGRAPH AND TEXT FORMAT

2.1 Paragraph Spacing

- A vertical spacing of approximately 2.5-line spaces shall be maintained between paragraphs.

2.2 Paragraph Indentation

- The first line of each paragraph may be indented by approximately 12 mm or 5 characters.
- Indentation may be omitted if adequate spacing between paragraphs is maintained.

2.3 Text Formatting Guidelines

- Avoid single-line paragraphs.

- Avoid leaving a single line at the top or bottom of a page (widows/orphans).
- Excessive hyphenation shall be avoided.

3. CHAPTER AND SECTION FORMAT

3.1 Chapter Format

- Each chapter shall begin on a new page.
- Chapter titles shall be centred and printed in 18 pt bold font.
- Chapters shall be numbered using Arabic numerals:
  - Example: Chapter 1, Chapter 2
- Adequate spacing shall be maintained:
  - Between chapter number and title
  - Between title and chapter content

3.2 Sections and Subsections

- Chapters may be divided into:
  - Sections
  - Subsections
  - Sub-subsections

Numbering Format

- Section: 2.1
- Subsection: 2.1.1

Formatting

- Section Headings:
  - 16 pt Bold
  - Left aligned
- Subsection Headings:
  - 14 pt Bold
  - Left aligned
- Further subdivisions may use smaller bold or italic fonts.

3.3 Table of Contents

The Table of Contents shall include:

- Chapters
- Sections
- Subsections
- Corresponding Page Numbers

4. TABLES, FIGURES, AND DIAGRAMS

4.1 General Guidelines

- Tables and figures shall preferably be in portrait orientation.
- Large figures or tables may be placed on separate pages.

4.2 Numbering

- Tables and figures shall be numbered chapter-wise.

Examples

- Table 3.2
- Fig. 5.4

4.3 Captions

- Table captions shall appear above the table.
- Figure captions shall appear below the figure.

4.4 Types of Figures

Depending on the project domain, figures may include:

- DFDs/UML Diagrams
- ER Diagrams
- Architecture Diagrams
- Flowcharts
- Neural Network Architectures
- Training Graphs
- Accuracy/Loss Curves
- IoT Block Diagrams
- Sensor Layouts
- Circuit Diagrams
- Drone/Robot Design Schematics
- Dashboard Screenshots
- Data Visualization Charts

4.5 References for Figures/Tables

- Sources reproduced from books, journals, websites, or research papers shall be properly cited.

Example:

(Source: After McGregor [12])

5. CODE, DATASET, AND EXPERIMENTAL RESULTS FORMAT

5.1 Source Code

- Important source code snippets may be included in the report.
- Large code files shall be included in appendices or digital media.

5.2 AI/ML/DL Project Requirements

Projects involving AI/ML/DL shall include:

- Dataset Description
- Training Parameters
- Model Architecture
- Performance Metrics
- Confusion Matrix/Graphs
- Experimental Results

5.3 IoT/Robotics/Embedded Project Requirements

Such projects shall include:

- Hardware Specifications
- Sensor Details
- Circuit Diagrams
- Communication Protocols
- Prototype Images
- Testing Results

6. BINDING AND COVER FORMAT

6.1 Binding

- The dissertation shall be hard-bound using rexine or equivalent material.

6.2 Front Cover

The front cover shall contain:

- Title of the Dissertation
- Name(s) of Student(s)
- Register Number(s)
- College Emblem/Logo
- Department Name
- Institution Name
- Degree and Semester
- Academic Year

6.3 Lettering

- Lettering on the cover and spine shall preferably be embossed in Gold/White/Silver color.

6.4 Spine Content

The following shall appear on the bound side:

- Student Name
- Degree Name
- Year of Submission

7. PRELIMINARY PAGES

7.1 Blank Pages

- One blank page may be inserted at the beginning and end of the dissertation for binding purposes.

7.2 Title Page

The title page shall include:

- Title of the Project
- Statement of Submission
- Degree Name
- Student Name and Register Number

- Name(s) of Guide/Supervisor
- Department and Institution Name
- Academic Year

7.3 Certificates and Declarations

The dissertation shall include:

- Institution Certificate
- Industry Certificate (if applicable)
- Student Declaration
- Plagiarism Declaration (optional/recommended)

8. REFERENCES AND CITATIONS

- References shall follow only IEEE citation style.
- All datasets, APIs, research papers, software libraries, frameworks, and online resources used in the project shall be properly acknowledged.
- AI-generated tools or services used during development/documentation, if any, shall be appropriately disclosed.

****

Questions Paper Pattern for Core Subjects

Duration:3 Hours Max.Marks:80

Note: Answer any ten Questions from Part-A. And one full Question from each unit in Part-B

Part-A

1. 10*2=20
   a.
   b.
   c.
   d.
   e.
   f.
   g.
   h.
   i.
   j.
   k.
   l.

Part-B

Each unit contains two main questions, and each carry 15 Marks. Each main question contains two or more sub question. Answer any ONE full main question from each Unit. 4*15=60

UNIT-I

2.
   a.
   b.
   c.

3.
   a.
   b.
   c.

UNIT-II

4.

a.
b.
c.

5.
a.
b.
c.

UNIT-III

6.
a.
b.
c.

7.
a.
b.
c.

UNIT-IV

8.
a.
b.
c.

9.
a.
b.
c.

****