SNDT Women's University, Mumbai

Postgraduate Degree / PG Programme (Syllabus as Per NEP) - Faculty of Science & Technology

Master of Computer Applications – Integrated

(MCA Integrated)

As Per NEP – 2020

Semester – V & VI

Syllabus
(W.E.F. Academic Year 2026-27)

Terminologies

Vertical	Full-Form/Definition	Remarks	Related To Major And Minor Courses
Major (Core)	Subject Comprising Mandatory and Elective Courses, Major Specific IKS, Vocational Skill Courses, Internship/ Apprenticeship, Field Projects, Research Projects Connected to Major	Minimum 50% Of Total Credits Corresponding to Three/Four - Year UG Degree- Mandatory Courses	Related To The Major
Minor Course	Course From Same Or Different Faculty	Minimum 18-20 Credits to Be Completed in The First Three Years of UG Programme	Related To the Minor
OEC	Open Elective Courses/ Generic Courses	10-12 Credits to Be Offered in I And/Or II Year. Faculty-Wise Baskets of OEC To Be Prepared	OEC Is to Be Chosen Compulsorily from Faculty Other Than That of the Major
VSC	Vocational Skill Courses, Including Hands On Training Corresponding To The Major And/Or Minor Subject	8-10 Credits, To Be Offered in First Three Years, Wherever Applicable Vocational Courses Will Include Skills Based on Advanced Laboratory Practical's of Major	Related To the Major or Minor
SEC	Skill Enhancement Courses	06 Credits, To Be Offered in I And II Year, To Be Selected from The Basket of Skill Courses Approved by University	Related To the Major or Minor Any Relevant Skill
AEC	Ability Enhancement Courses	08 Credits, To Be Offered in I And II Year, English: 04 Credits to Be Earned in Sem - I, Modern Indian Language Of 04 Credits to Be Offered in II Year	NA
VEC	Value Education Courses	Understanding India, Environmental Science/Education, Digital and Technological Solutions, Health &	NA

		Wellness, Yoga Education, Sports, And Fitness	
IKS	Indian Knowledge System	Generic IKS Course: Basic Knowledge Of The IKS To Be Offered At First Year Level	Major-Specific IKS Courses: Advanced Information About the Major, Part of the Major Credit to Be Offered at Second- Or Third-Year Level
OJT	On-Job Training (Internship / Apprenticeship)	Corresponding To the Major Subject	Related To The Major
FP	Field Projects	Corresponding To the Major Subject	Related To the Major
CC	Co-Curricular Courses	Health And Wellness, Yoga Education Sports, And Fitness, Cultural Activities, NSS/NCC And Fine/ Applied/Visual/ Performing Arts	NA
CE	Community Engagement and Service		Related To Major
RP	Research Project	Corresponding To the Major Subject	Related To Major

Programme Template

Faculty		**Science and Technology**
Programme Degree		Master of Computer Applications-Integrated (MCA-Integrated)
Specialization		NA
Preamble (Brief Introduction to the programme)		The MCA-Integrated programme of SNDT Women's University, Mumbai, offered under the Faculty of Science & Technology as per NEP and AICTE norms, is designed to provide a strong foundation in computer applications, programming, mathematics, systems, and emerging web-based technologies. The first-year curriculum includes Mathematics Foundation to Computer Science I and II, Problem Solving Techniques, Computer Architecture, Fundamentals of Computer Science, C Programming Lab, Data Structures, Operating Systems, Object-Oriented Programming using Java, Web Technologies, and supporting ability/value courses such as General English, Indian Knowledge System, Environmental Science and Sustainability, Indian Constitution, and relevant laboratories. The programme emphasizes conceptual understanding, hands-on programming, laboratory practice, project-based learning, continuous internal assessment, end-semester examination, communication skills, constitutional values, environmental awareness, and professional readiness. It prepares learners for higher-level courses and careers in software development, systems analysis, web application development, database-driven applications, IT services, and research-oriented computing domains.
Programme Specific Outcomes (POs)		**After completing this programme, Learner will**
	1.	Apply foundational mathematical, logical, and computational concepts to analyse and solve problems in computer applications.
	2.	Design algorithms, pseudocode, flowcharts, and structured programs using problem-solving techniques and C programming fundamentals.
	3.	Explain computer architecture, digital logic, memory organisation, operating system concepts, and system-level mechanisms.
	4.	Implement software solutions using core programming paradigms, data structures, object-oriented programming using Java, and web technologies.

	5.	Apply linear and non-linear data structures, searching, sorting, recursion, hashing, trees, and graphs for efficient problem solving.
	6.	Develop interactive, responsive, and secure web applications using HTML, CSS, JavaScript, XML/AJAX, backend scripting, REST APIs, and databases.
	7.	Apply numerical methods, optimization techniques, linear programming, transportation models, and discrete mathematics to computational problems.
	8.	Demonstrate communication proficiency, constitutional awareness, environmental responsibility, and understanding of Indian Knowledge Systems.
	9.	Plan, implement, test, document, and present laboratory-based and project-based computing solutions using appropriate evaluation rubrics.
	10.	Integrate ethical, legal, sustainability, and professional practices in computing and technology-enabled problem solving.
Eligibility Criteria for Programme		A female student who has passed 10+2 examination with Mathematics/Statistics/ Accountancy as compulsory subject. Obtained at least 45% marks in general category and 40% marks in case of candidates belonging to reserved category, in the above subjects taken together and Valid MAH-MCA Integrated-CET score applicable to that academic year.
Intake **(For SNDT WU Departments and Conducted Colleges)**		60
Examination passing criteria		Students have to pass both the internal assessment and external assessment separately for each course with minimum 40% passing marks

Structure with Course Titles

Master of Computer Applications – Integrated

Semester – V

Sr. No.	Course	Type of Course	Credits	Marks	Int Marks	Ext Marks
	Semester – V					
50135111	Quantitative Techniques (Th)	Major (Core)	4	100	50	50
50135112	Software Engineering (Th)	Major (Core)	4	100	50	50
51035121	Python Lab (Pr)	IKS (Major Specific)	2	50	0	50
50235111	Data Science (Th+Pr) (2+2)	Major (Elective) **(Any One)**	4	100	50	50
50235112	Artificial Intelligence (Th+Pr) (2+2)					
50335111	Cyber Security (Th)	Minor Stream	**4**	100	50	50
50635101	Software Testing and Quality Assurance (Pr)	VSC-4	2	50	50	0
51335101	Field Project (Pr)	FP	2	50	50	0
			22	**550**	**300**	**250**

Semester – VI

Sr. No.	Course	Type of Course	Credits	Marks	Int Marks	Ext Marks
	Semester - VI					
60135111	Machine Learning (Th+Pr) (2+2)	Major (Core)	4	100	50	50
60135112	Big Data Analytics (Th+Pr) (2+2)	Major (Core)	4	100	50	50
60235111	Cloud Computing (Th+Pr) (2+2)	Major (Elective) **(Any One)**	4	100	50	50
60235112	Business Analytics Concepts & Applications (Th+Pr) (2+2)					
60335121	PLSQL and NOSQL Lab (Pr)	Minor Stream	2	50	0	50
60335112	Mobile Application Development Technologies (Th+Pr) (2+2)	Minor Stream	4	100	50	50
61235121	On Job Training (Pr)	OJT	4	100	50	50
			22	**550**	**250**	**300**

Course Syllabus

Semester – V

.5.1 Major (Core)

Course Titles	**Quantitative Technique (Th)**
Course Credits	**4 Credit's**
Course Outcomes	After Completion of this Course, students will be able to
	1. Apply statistical and probabilistic techniques—including data analysis, regression, forecasting, hypothesis testing, and Bayes' Theorem—to interpret data and support decision-making using statistical software tools.
	2. Formulate and solve optimization and operational problems using models such as Linear Programming, assignment, transportation, inventory (including Economic Order Quantity), and queuing systems, along with appropriate computational tools.
	3. Analyze and optimize real-world systems and projects through techniques like Critical Path Method (CPM), PERT, sensitivity analysis, and evaluation of Markovian and non-Markovian models for effective planning, scheduling, and resource management.
Module1: Statistical Methods (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Apply statistical methods for modeling and inference, including correlation, regression, forecasting, Bayes' Theorem, probability distributions, and hypothesis testing.
	2. Use advanced statistical techniques and tools such as sampling distributions, non-parametric methods, Analysis of Variance (ANOVA), covariance analysis, and statistical software packages for data analysis and decision-making.
Content Outline	• Frequency distribution, measures of central tendency, measures of dispersion, linear correlation and regression, forecasting, Elementary probability theory, Bayes theorem, Some standard discrete and continuous distributions, Testing of statistical hypotheses and tests of significance, sampling distributions, non-parametric methods, Analysis of variance and covariance, Use of some relevant statistical packages
Module2: Optimization models (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Formulate and solve optimization problems using techniques such as Linear Programming, the Simplex Method, assignment, and transportation models.
	2. Apply network-based project management techniques like PERT

	and Critical Path Method to plan, schedule, and optimize resources in complex projects.
	3. Evaluate optimization results using sensitivity analysis and software tools, and analyze trade-offs through network compression and cost-time considerations.
Content Outline	• Introduction to optimisation models, Assignment problem, transportation problems, Linear programming, Simplex method, sensitivity analysis, Use of relevant packages' Network analysis, PERT/CPM, resource and scheduling, network compression and cost consideration, use of relevant packages
Module 3: Inventory model (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Perform sensitivity analysis to evaluate how changes in demand, cost, and lead time affect inventory decisions and total cost.
	2. Analyze inventory systems with non-zero lead time and determine appropriate reorder points to prevent stockouts.
Content Outline	• The classical economic order quantity, Sensitivity analysis, Non-zero lead-time
Module 4: Queuing model (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand the fundamental characteristics of queuing models and explain key performance measures such as waiting time, queue length, and system utilization.
	2. Analyze and apply Markovian queuing model to solve problems involving memoryless arrival and service processes.
	3. Differentiate between Markovian queuing model and Non-Markovian queuing model, and evaluate their applicability in real-world systems.
Content Outline	• General characteristics, Performance measure, Markovian queuing model, Non-Markovian queuing model

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Data Analysis & Statistical Software Lab**
  Work with real or simulated datasets to construct frequency distributions, compute measures of central tendency and dispersion, perform regression and hypothesis testing, and apply Bayes' Theorem using statistical software (e.g., Excel, R, or Python).

- **Optimization & Network Problem-Solving Workshop**
  In groups, students solve assignment, transportation, and Linear Programming problems using the Simplex Method, and develop project schedules using PERT / Critical Path Method with case studies and software tools.

- **Simulation-Based Inventory & Queuing Activity**
  Students simulate real-life systems (e.g., retail inventory or service queues) to apply the Economic Order Quantity model and compare Markovian queuing model vs. Non-Markovian queuing model, analyzing performance measures and decision outcomes.

References Book:

1. PERT and CPM Principles and Application: Srinath
2. Operations Research, Kantiswaroop, Gupta
3. Operations Research Methods and Problems: Sasieni, Yaspan Friedman
4. Mathematical Methods in Operations Research: Wagner
5. Operations Research: Sharma
6. Operations Research: Taha
7. robability and statistical Inference: R.V.Hogg&E.A.Tanis, Macmillan, 1983
8. Introductory Mathematical Statistics: E.Kreyszig Wiley, 1970

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a case study should be developed of 50 marks.

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Semester – V

.5.2 Major (Core)

Course Titles	**Software Engineering (Th)**
Course Credits	**4 Credit's**
Course Outcomes	After Completion of this Course, students will be able to
	1. To Acquire a comprehensive understanding of the software development lifecycle and its application in contemporary software engineering practices.
	2. To Develop proficiency in project management methodologies and strategic decision making for successful software project execution.
	3. To Master the art of software design, development, and testing to produce robust and efficient software solutions.
Module1: Introduction (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Understand the evolution and significance of software engineering, including the changing nature of software, layered technology, and the structure of a generic process framework.
	2. Analyze and compare different software process models, such as waterfall, incremental, evolutionary, and unified process, to determine their suitability for various project requirements.
	3. Apply Agile principles and methodologies, including Scrum and Extreme Programming, and differentiate between plan-driven and agile approaches for effective software development.
Content Outline	• The evolving role of software, changing nature of software, layered technology, a process framework, Process models: The waterfall model, incremental process models, evolutionary process models, the unified process. • Agile software development: Agility Principles, Agile methods, Plan-driven and agile development, Extreme programming, Scrum, A Tool Set for the Agile Process.
Module2:Software Requirements, Risk management (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Identify and classify software requirements, including functional and non-functional requirements, and develop a structured Software Requirements Specification (SRS) document.
	2. Apply requirements engineering processes, such as elicitation, analysis, validation, and management, to ensure accurate and consistent software requirements.
	3. Analyze project risks and plan software projects effectively, including risk identification and mitigation (RMMM), estimation

	techniques, scheduling, and agile vs plan-driven planning approaches.
Content Outline	• **Software Requirements Engineering:** Functional and non-functional requirements, the software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management. • **Risk management:** Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan. • **Project planning:** Software pricing, Plan-driven development, Project scheduling, Agile planning, Estimation techniques.
Module 3:Design,Testing Strategies and Product metrics (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply software design principles and modeling techniques, including architectural, data, and structural design using UML diagrams such as class, sequence, use case, and component diagrams.
	2. Evaluate and implement software testing strategies, including black-box and white-box testing, validation, system testing, and effective debugging practices.
	3. Analyze software quality using product metrics, including metrics for analysis, design, source code, testing, and maintenance to ensure high-quality software systems.
Content Outline	• **Design:** Design process and design quality, design concepts, the design model, software architecture, data design, architectural design, Basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams. • **Testing Strategies:** A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. • **Product metrics:** Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.
Module 4: Quality Management and Release Management (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand and apply software quality management practices, including quality assurance, formal technical reviews, statistical quality assurance, and reliability concepts.
	2. Plan and manage software releases effectively, including release planning, build strategies, risk management, and post-

	deployment monitoring.
	3. Evaluate and implement product sustenance strategies, including software maintenance, updates, migration, and end-of-life management.
Content Outline	• **Quality Management:** Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability. • **Release Management:** Release planning, development and build plans, release strategies, risk management, and post-deployment monitoring. • **Product sustenance:** Maintenance, updates, End of life, migration strategies.

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Activity: Process Model Role-Play Activity**
  Divide students into groups and assign each a model (Waterfall, Incremental, Agile, Unified Process).

- **Activity: Requirements Engineering Case Study**
  Provide a simple problem statement (e.g., Library Management System / Online Food Ordering System).

- **Activity: UML Design Workshop**
  Students design UML diagrams for a given system:
  - Use Case Diagram
  - Class Diagram
  - Sequence Diagram

- **Activity: Testing & Debugging Lab**
  Provide a small buggy program.

References:

1. Software Engineering, N.S. Gill, Khanna Publishing House, 2023
2. Software Engineering, Ian Somerville, 9th edition, Pearson education.
3. Software Engineering A practitioner's Approach, 8th edition, Roger S Pressman, Bruce R. Maxim. McGraw Hill Education, 2015.
4. Stephen Schach, Software Engineering 7th ed, McGraw-Hill, 2007
5. Software Engineering: Principles and Practice Hans van Vliet

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a case study should be developed of 50 marks.

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Semester – V

.5.1 Indian Knowledge System (IKS) (Major Specific)

Course Titles	**Python Lab (Pr)**
Course Credits	**2 Credit's**
Course Outcomes	After Completion of this Course, students will be able to
	1. Apply fundamental programming concepts to solve mathematical and logical problems using Python.
	2. Develop and utilize functions, strings, and data structures (lists, tuples, dictionaries) for efficient data processing.
	3. Implement numerical computations and array operations using libraries such as NumPy.
	4. Perform file handling operations and manage data using text and file-based techniques.
Module1:Basic of Python Programming (Credit 1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Understand the basic structure and syntax of Python programs.
	2. Identify and use Python identifiers, keywords, operators, and data types correctly.
	3. Perform input/output operations and apply type conversion techniques.
	4. Manipulate strings using various operations such as slicing, formatting, and built-in functions.
	5. Apply control flow statements (conditional and looping) to solve basic programming problems.
Content Outline	• Structure of a Python program • Identifiers and keywords • Operators and operator precedence • Basic data types and type conversion • Statements and expressions • Input and output operations • Creating and storing strings • Built-in string functions and operators • String slicing, indexing, and joining • String formatting techniques • Conditional statements (if, else) • Looping constructs (for, while)

	• Nested control structures • Loop control statements: break, continue, pass, exit • Built-in functions in Python • Function definition and function calls • Scope and lifetime of variables • Default parameters and command line arguments • Lambda functions • Assert statement • Importing user-defined modules • Lists, tuples, and dictionaries • Common operations on lists, tuples, and dictionaries • Passing lists, tuples, and dictionaries as function arguments
Module2: File Handling and Exceptions (Credit 1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Apply and analyse NumPy and Math modules for numerical computations
	2. Perform file operations including reading, writing, and appending data in text and binary files.
	3. Work with structured data formats such as CSV and JSON files.
	4. Use the Pickle module for object serialization.
	5. Implement exception handling techniques using try-except blocks.
	6. Create and manage user-defined exceptions for robust program design
Content Outline	• Introduction to Math module • Introduction to NumPy for arrays and numerical computations • Types of files (text and binary) • Creating, reading, and writing files • The Pickle module • Working with CSV files • Reading and writing JSON files • Try–except–else–finally blocks • Raising exceptions • Exception hierarchy • Creating user-defined exceptions

References:

1. Venkatesh, Nagaraju Y, Introduction to Python Programming, Khanna Publishing House, 2021.
2. Jeeva Jose, Introduction to Computing & Problem Solving With PYTHON, Khanna Publishing House, 2023.
3. SheetalTaneja& Naveen kumar: Python Programming a Modular approach – A Modular approach with Graphics, Database, Mobile and Web applications, Pearson, 2017.
4. Munesh Chandra Trivedi, Data Science and Data Analytics Using Python, Khanna Publications

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below). Evaluation done at Institute level.

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	20
Implementation & Functionality	10
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	5
Report writing or Documentation	5

Practical Outline:

1. Write a program to find whether a number is a prime number.
2. Write a program to print m raise to power n, where m and n are read from the user.
3. Write a program having a parameterized function that returns True or False depending on whether the parameter passed is even or odd.
4. Write a program to print the summation of the following series upto n terms:1-2+3-4+5- 6+7 - - - - - - - n
5. Write a menu driven program to perform the following operations on strings using string built in functions.
   a) Find the frequency of a character in a string. Replace a character by another character in a string.
   b) Remove the first occurrence of a character from a string.
   c) Remove all occurrences of a character from a string.
6. Write a program that accepts two strings and returns the indices of all the occurrences of the second string in the first string as a list. If the second string is not present in the first string, then it should return -1
7. Using Numpy module write menu driven program to do following
   a) Create an array filled with 1's.
   b) Find maximum and minimum values from an array

c) Dot product of 2 arrays.
d) Reshape a 1-D array to 2-D array.

8. Write a function that takes a sentence as input from the user and calculates the frequency of each letter. Use a variable of dictionary type to maintain the count.
9. Consider a tuple t1=(1,2,5,7,9,2,4,6,8,10). Write a program to perform following operations:
   a) Print contents of t1 in 2 separate lines such that half values come on one line and other half in the next line.
   b) Print all even values of t1 as another tuple t2.
   c) Concatenate a tuple t2=(11,13,15) witht1.
   d) Return maximum and minimum value from t1.
10. Write a function that reads a file file1 and copies only alternative lines to another file file2. Alternative lines copied should be the odd numbered lines.
11. Write a Python program to handle a ZeroDivisionError exception when dividing a number by zero.
12. Write a program that reads a list of integers from the user and throws an exception if any numbers are duplicates.
13. Write a program that makes use of a function to display sine, cosine, polynomial and exponential curves.
    a) Take as input in the months and profits made by a company ABC over a year. Represent this data using a line plot. Generated line plot must include X axis label name = Month Number and Y axis label name = Total profit**.**

Semester – V

.5.4 A. Major (Elective)

Course Titles	**Data Science (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Analyze different types and sources of data, along with the techniques to collect and interpret it.
	2. Apply understanding of various statistical concepts like distributions, hypothesis testing, confidence intervals, and correlation.
	3. Evaluate various machine learning algorithms such as Linear Regression, Logistic Regression, Decision Trees, Clustering, etc.
	4. Design and implement text mining techniques to analyze unstructured data, possibly including hands-on lab sessions for practical understanding.
Module1: Introduction to Data Science (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Analyze various types of data, identifying and differentiating between them based on their characteristics and use cases.
	2. Apply understanding of data analysis to construct models for predictive or descriptive purposes.
	3. Evaluate and implement feature extraction methodologies to enhance model performance and interpretability.
	4. Design illustrations to explain the stages within the Data Science lifecycle and their significance in problem-solving.
Content Outline	• **Introduction to Data:** What is Data? Different kinds of data, Data Sources, Different types of data sources, Exploratory Data Analysis (EDA), Data Science lifecycle, Data Collection, Data Extraction, Data Analysis & Modeling, Data transformations: Dimension reduction, Feature extraction, Smoothing and aggregating.
Module2:Design Strategies (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Analyze and perform various mathematical operations efficiently on NumPy arrays, a powerful tool for mathematical computation in Python.
	2. Apply understanding of the data structures provided by Pandas (Series, DataFrame) and utilize them effectively in data analysis tasks.
	3. Evaluate and execute merging operations to combine datasets

	efficiently for comprehensive analysis.
	4. Design strategies to apply various types of joins effectively to merge datasets based on common columns or indices.
Content Outline	• **The World of arrays with Numpy:** creating an array, Mathematical operations, Indexing and slicing, Shape manipulation. • Empowering Data analysis with pandas: the data structure of pandas, Inserting and exporting data. • **Data Cleansing:** checking missing data, filling missing data, merging operations. • **Data Operations:** Aggregation operations, Joins.
Module 3: Inferential Statistics (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Analyze various probability distributions (normal, binomial, Poisson, etc.), identifying and differentiating between them based on their characteristics and real-world applications.
	2. Apply control over line properties in various chart types for better visualization, enhancing the clarity and effectiveness of data representation.
	3. Evaluate and differentiate between various types of Machine Learning (supervised, unsupervised, reinforcement learning), understanding their unique characteristics and applications.
Content Outline	• **Inferential Statistics:** Various forms of distribution, z-score, p-value, Type 1 and Type 2 errors, Confidence Interval, Correlation, Chi-square distribution, ANOVA. • **Making Sense of Data Through Visualization**: Controlling the line properties of a chart, creating multiple plots, styling your plots, Boxplots, Heatmaps, Scatter plots with histogram, Bubble charts. • **Uncovering Machine Learning:** Different types of Machine Learning, Linear Regression, Logistic Regression, Decision Tree, K-means Clustering, Hierarchical Clustering.
Module 4: Generating Recommendations Systems(Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply understanding of User-Based Collaborative Filtering techniques to generate recommendations based on user similarities.
	2. Analyze and explain the theory behind text mining and its applications in analyzing unstructured data, providing clear and comprehensive insights.
Content Outline	• **Generating Recommendations Systems:** User Based

	collaborative filtering, Item Based collaborative filtering, Context Based filtering. • **Case Study Theory:** Analyzing Unstructured Data using Text mining techniques. (Case Study Practical Implementation to be performed in lab as part of Practical's)

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1: Activity**
  Conduct an exploratory data analysis on a dataset of your choice. Identify the different types of data present, apply your understanding of data analysis to construct predictive models, implement feature extraction methodologies to enhance model performance, and design illustrations to explain your process and findings.

- **Module 2: Activity**
  Using Python, create a NumPy array and perform various mathematical operations on it. Then, use a dataset of your choice and apply your understanding of data structures using Pandas (Series, DataFrame) for data analysis tasks. Perform merging operations and apply various types of joins to combine datasets.

- **Module 3: Activity**
  Analyze different probability distributions (normal, binomial, Poisson, etc.) and their real-world applications. Create different types of charts with controlled line properties for better visualization. Lastly, differentiate between various types of Machine Learning (supervised, unsupervised, reinforcement learning), and implement one of them on a dataset.

- **Module 4: Activity**
  Implement a User-Based Collaborative Filtering technique to generate recommendations based on user similarities on a dataset. Explain the theory behind your approach. Also, find a set of unstructured data and apply text mining techniques to analyze it. Provide comprehensive insights from your analysis.

References Book:

1. Provost, F., & Fawcett, T. (2013). Data Science for Business. O'Reilly Publication.
2. McKinney, W. (2017). Python for Data Analysis. O'Reilly Publication.
3. Newbold, P., & Carlson, W. L. (2012). Statistics for Business and Economics. Pearson Publication.
4. Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python. O'Reilly Publication.
5. Madhavan, S. (2015). Mastering Python for Data Science: Explore the World of Data Science Through Python and Learn How to Make Sense of Data. Packt Publishing. Packt Publishing.
6. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly.

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks.

Project based evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Formulation	10
Algorithm / Model Selection	10
Implementation & Functionality	10
Analysis & Result Interpretation	10
Report writing or Documentation	10

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Semester – V

.5.4 B. Major (Elective)

Course Titles	**Artificial Intelligence (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand the fundamental concepts, history, and evolution of Artificial Intelligence, including key domains and applications.
	2. Apply problem-solving techniques such as search algorithms (uninformed and informed searches) and game playing strategies for developing intelligent solutions.
	3. Demonstrate the ability to implement knowledge representation methods like logic-based models, semantic networks, and ontologies for AI problem domains.
Module1: Introduction to Artificial Intelligence (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Define Artificial Intelligence and explain its scope, history, and evolution.
	2. Identify key areas and applications of AI across various industries like healthcare, education, finance, and automation.
	3. Illustrate real-world examples of AI systems and discuss how they improve efficiency and decision-making.
Content Outline	• **Definition, History & Evolution of AI** • **Types of AI:** Narrow AI, General AI, Super AI • Applications of AI in various sectors • **Intelligent Agents:** Types, Structures, and Environments • **Solving by searching:** Travelling salesman Problem, Monkey Banana Problem, Tower of Hanoi and Block World.
Module2:Problem solving by Searching Methods (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Explain the role of search in problem-solving within Artificial Intelligence.
	2. Differentiate between various search strategies, including uninformed (blind) and informed (heuristic) search techniques.
	3. Apply informed search techniques like Greedy Best-First Search and A* Search for efficient problem-solving.
Content Outline	• **Uninformed Search Techniques** o Breadth-First Search (BFS)

	○ Depth-First Search (DFS) ○ Iterative Deepening Search • **Informed Search Techniques** ○ Best-First Search ○ A* Search ○ Hill Climbing • **Adversarial Search** ○ Game Playing: Minimax Algorithm ○ Alpha-Beta Pruning
Module 3: Knowledge Representation and Reasoning (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Describe the importance of knowledge representation in building intelligent systems.
	2. Apply logic-based methods (Propositional and Predicate Logic) to represent facts, rules, and relationships in problem domains.
	3. Implement reasoning techniques such as forward chaining, backward chaining, and resolution to infer new knowledge from existing information.
Content Outline	• **Knowledge Representation Techniques :** Propositional Logic, Predicate Logic, Semantic Networks, Frames and Ontologies • **Reasoning Methods :** Forward Chaining, Backward Chaining, Resolution • **Reasoning under Uncertainty :** Bayesian Networks, Fuzzy Logic and Reasoning • **Structured Knowledge Representation:** Associative Networks, Semantic Nets, Frames Structures, Conceptual, Dependencies & Scripts, Learning – Concept of Learning – Learning Automata, Learning by induction.
Module 4: Expert Systems and Modern Generative AI Applications (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Explain traditional and modern knowledge representation approaches in AI systems.
	2. Understand the fundamentals of Large Language Models (LLMs) and their applications.
	3. Describe the concept and workflow of Retrieval-Augmented Generation (RAG).

	4. Build simple AI applications using prompt-based interaction and document retrieval.
Content Outline	• **Expert Systems** Knowledge acquisition methods. Knowledge engineering process. Architecture of expert systems. Rule-based expert systems. Advantages and limitations • **Large Language Models (LLMs)** Introduction to Generative AI. Concept of Transformer Architecture (overview). Introduction to Large Language Models. Applications of LLMs in education, healthcare, business, and automation. Prompt Engineering basics. Limitations of LLMs: hallucination, bias, privacy concerns. • **Introduction to RAG** Concept of Retrieval-Augmented Generation, Difference between traditional LLM and RAG systems. Components of RAG: document loader, embedding generation, vector database, retriever generator. Use cases of RAG in academic and industrial systems. Simple demonstration using local documents

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1:**
  Present a report on the role of AI in a specific sector of engineering of your choice, and how it impacts daily activities in that sector. Analyze the difference between natural intelligence and artificial intelligence within the context of this sector. Discuss the significant task domains of AI in this sector and the limitations and challenges faced, including ethical considerations, data privacy concerns, and technical hurdles.

- **Module 2:**
  Implement a simple 2-player game (e.g., Tic-Tac-Toe or Nim).
  - Apply:
    - ✓ Minimax Algorithm
    - ✓ Alpha-Beta Pruning
  - Compare:
    - ✓ Nodes explored
    - ✓ Execution efficiency
    - ✓ Visualize game tree.

- **Module 3:**
  - Represent a real-world domain (e.g., medical diagnosis or student performance) using:
    - ✓ Propositional Logic
    - ✓ Predicate Logic
  - Convert statements into logical expressions and rules.
  - Output: Knowledge base and explanation.

- **Module 4:**

- Expert System Development (Mini Project)
- Design a rule-based expert system for:
  - ✓ Career guidance / troubleshooting
- Include:
  - ✓ Knowledge base
  - ✓ Inference engine
  - ✓ Sample rules

  OR
- Develop a mini project using Ollama and introductory RAG

References:

1. Stuart Russell, Peter Norvig (2022). Artificial Intelligence: A Modern Approach (4th ed.). Pearson
2. Introduction to AI and Expert Systems – Patterson
3. Artificial Intelligence - Rich E and Knight K.
4. Principles of Artificial Intelligence - Nilsson.
5. Artificial Intelligence – An Engineering Approach – Schalkoff R J.
6. Introduction to Expert System - Peter Jackson.
7. Hands-On Large Language Models – Jay Alammar & Maarten Grootendorst
8. Natural Language Processing with Transformers – Lewis Tunstall et al.
9. Ollama Documentation (practical manual)

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Project evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	Marks
Problem Understanding & Formulation	10
Implementation & Functionality	10
Knowledge Representation / Logic Design	10
Analysis & Result Interpretation	10
Report writing or Documentation	10

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the university.

Semester – V

.5.5 Minor Stream

Course Titles	**Cyber Security (Th)**
Course Credits	**4 Credit's**
Course Outcomes	After Completion of this Course, students will be able to
	1. Provide an overview of the field of Cyber Security, including its challenges, constraints, and the role of Internet governance.
	2. Understand the imperative for a comprehensive Cyber Security policy, the establishment of a nodal authority, and the importance of an international convention on Cyberspace.
	3. Identify vulnerabilities in software, system administration, network architectures, data access, authentication, broadband communications, and poor awareness.
	4. Identify intrusion types, such as physical theft, privilege abuse, unauthorized access, malware infection, and implement techniques including anti-malware software, network-based intrusion detection/prevention systems with its types.
Module1: Introduction to Cyber Security (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Demonstrate the ability to conduct security audits, identifying potential weaknesses in systems and networks.
	2. Explain the role of cryptography in Cyber Security and apply cryptographic techniques to secure data communication.
	3. Understand the concept of ethical hacking and its role in proactively identifying and addressing vulnerabilities.
Content Outline	• **Introduction to Cyber Security:** Overview of Cyber Security, Internet Governance–Challenges and Constraints, Cyber Threats: - Cyber Warfare-Cyber Crime-Cyber, terrorism-Cyber Espionage, Need for a Comprehensive Cyber Security Policy, Need for a Nodal Authority, Need for an International convention on Cyberspace. • **Cyber Security Vulnerabilities and Safeguards:** Cyber Security Vulnerabilities Overview, vulnerabilities in software, System administration, Complex Network Architectures, Open Access to Organizational Data, Weak Authentication, Unprotected Broadband communications, Poor Cyber Security Awareness. Cyber Security Safeguards Overview, Access control, Audit, Authentication, Biometrics, Cryptography, Deception, Denial of Service Filters, Ethical Hacking, Firewalls, Intrusion Detection Systems, Response, Scanning, Security policy, Threat Management.
Module2:Securing Web Application (Credit1)	

Learning Outcomes	After learning this module, learners will be able to
	1. Define and explain the roles of services and servers in the context of web applications.
	2. Understand identity management principles and implement secure identity practices within web services.
	3. Apply security measures to prevent and detect unauthorized access attempts by external entities.
Content Outline	• **Securing Web Application:** Services and Servers Introduction, Basic security for HTTP Applications and Services, Basic Security for SOAP Services, Identity Management and Web Services, Authorization Patterns, Security Considerations, Challenges. • **Intrusion Detection and Prevention:** Intrusion, Physical Theft, Abuse of Privileges, Unauthorized Access by Outsider, Malware infection, Intrusion detection and Prevention Techniques, Anti-Malware software, Network based Intrusion detection Systems, Network based Intrusion Prevention Systems, Host based Intrusion prevention Systems, Security Information Management, Network Session Analysis, System Integrity Validation.
Module 3: Cryptography and Network Security (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Define cryptography and explain its role in securing information and communication.
	2. Differentiate between symmetric and asymmetric key cryptography, and understand their applications in securing data.
	3. Understand and apply cryptography in various applications, demonstrating proficiency in securing data in different contexts.
Content Outline	• **Cryptography and Network Security** o Introduction to Cryptography, Symmetric key Cryptography, Asymmetric key Cryptography, Message Authentication, Digital Signatures, Applications of Cryptography. o Overview of Firewalls- Types of Firewalls, User Management, VPN Security, Security Protocols:-security at the Application Layer-PGP and S/MIME, Security at Transport Layer-SSL And TLS, Security at Network Layer-IPSec.
Module 4: Cyber jurisprudence & Forensic (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Analyze the specificities of the Indian cyber space, including its regulatory framework, challenges, and initiatives.
	2. Define Cyber Forensics and understand its significance in

	investigating cybercrimes and digital incidents.
	3. Demonstrate the ability to initiate and conduct preliminary investigations in response to suspected cyber incidents, ensuring the preservation of digital evidence.
Content Outline	• **Cyberspace and the Law**: Introduction, Cyber Security Regulations, Roles of International Law, the state and Private Sector in Cyber space, Cyber Security Standards. The INDIAN Cyber space, National Cyber Security Policy 2013. • **Cyber Forensics:** Introduction to Cyber Forensics, Handling Preliminary Investigations, Controlling an Investigation, conducting disk-based analysis, Investigating Information-hiding, Scrutinizing E-mail, Validating E-mail header information, Tracing Internet access, Tracing memory in real-time.

Activities towards Comprehensive Continuous Evaluation (CCE)

- These activities are designed to engage students in practical applications of cyber security concepts, fostering critical thinking and skill development throughout the course.
- Divide students into groups and assign each group a specific cyber security vulnerability (e.g., software vulnerabilities, weak authentication). Have them research, analyze, and present strategies to mitigate the assigned vulnerability.
- Organize a workshop where students present and demonstrate various cyber security safeguards. This can include access control, encryption, firewalls, and intrusion detection systems. Encourage hands-on activities and practical demonstrations.
- Assign students a case study involving a web application. They should conduct a security assessment, identify vulnerabilities, and propose safeguards. Emphasize securing HTTP and SOAP services, identity management, and authorization patterns.
- Provide case studies related to intrusion incidents. Students should analyze each case, identify the type of intrusion, and propose effective prevention and detection techniques. Encourage discussion on ethical hacking and security policy enforcement.
- Provide legal cases related to cyber space and cyber security. Students should analyze the legal implications, court decisions, and the role of international law and regulations.
- Assign students to review and critique the National Cyber Security Policy of 2013. They should assess its effectiveness, identify areas for improvement, and propose updated recommendations.
- Encourage students to write research papers on cyber security and spreading awareness among society.

References:

1. Cybersecurity Essentials Charles J. Brooks et al., Publisher: Wiley, 2022.
2. Cryptography and Network Security Atul Kahate, Publisher: McGraw Hill Education.
3. Cybersecurity in Context: Technology, Policy, and Law Chris Jay Hoofnagle & Golden G. Richard III, Publisher: Wiley, 2024 1st Edition.

4. Digital Forensics and Incident Response Gerard Johansen, Publisher: Packt Publishing, 3rd Edition, 2023.
5. Digital Privacy and Security Using Windows: A Practical Guide Nihad Hassan, Rami Hijazi, Apress.
6. Securing the Future: The Role of AI in Cybersecurity Nimeshkumar Patel.

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a case study should be developed for 50 marks.

Given below are two sample case study, but it is expected to work on similar sort of Case study.

- **Case study 1**:-The Data Breach at Any Bank, Identify the cyber vulnerabilities that led to the breach. Discuss what kind of Intrusion Detection/Prevention Systems should have been used. Propose a Cyber Security policy and technical safeguards to prevent such attacks.
- **Case study 2:-**E-commerce Website Under DDoS Attack Explain the working of a DDoS attack and how it affects server availability. Suggest network-level and application-level safeguards to mitigate DDoS risks. Recommend changes in server architecture to prevent similar issues in the future.

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted by university.

Semester – V

.5.6 Vocational Skill Courses (VSC-4)

Course Titles	**Software Testing and Quality Assurance- LAB (Pr)**
Course Credits	**2 Credit's**
Course Outcomes	**After Completion of this Course, students will be able to**
	1. Understand the fundamental principles and concepts of software testing, including its purpose, objectives, and the role it plays in software development.
	2. Apply various testing methods such as functional, non-functional, unit, integration, system, regression, and acceptance testing to diverse software systems.
Module1: Fundamentals of Software Testing (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. To understand key testing concepts, including error, fault, failure, defect, and bug.
	2. Recognize the principles and objectives of testing, including early testing, defect clustering.
	3. Differentiate between black-box and white-box testing techniques and identify when to apply each in real-world scenarios.
	4. To analyze code and specifications to design appropriate test cases using both black-box and white-box methods.
Content Outline	• **Introduction to Software Testing** o Goals and Principles of Testing o Software Development Life Cycle (SDLC) vs Software Testing Life Cycle (STLC) o V-Model and Agile Testing Overview o Types of Testing: Manual vs Automated, Static vs Dynamic o Defect Lifecycle, Severity & Priority • **Black Box Techniques:** o Equivalence Partitioning o Boundary Value Analysis o Cause-Effect Graphing o Decision Table Testing • **White Box Testing Techniques:** o Statement, Branch, Condition, and Path Coverage o Cyclomatic Complexity

	○ Code Reviews and Walkthroughs
Module2: Testing Techniques (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Apply SQA practices such as reviews, audits, and inspections to improve software quality and reduce defects.
Content Outline	• **Automated testing:** ○ Functional Test Automation using Selenium ○ Performance Testing using Apache JMeter ○ Defect Tracking & Test Management using GitLab ○ Concepts of Software Quality ○ Quality Factors (McCall's and ISO 9126) ○ SQA Activities and Audits

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1:**
  - Explain the software testing life cycle (STLC) with diagrams.
  - Describe the role of a test plan. What are its key components?
  - Explain the significance of early testing in SDLC.
  - Provide a case study of an organization's testing process and have students analyze and propose improvements based on maturity model principles.
  - Case study -A software product is nearing completion. As a test engineer, what approach will you take to ensure it meets quality standards?

- **Module 2:**
  - Differentiate between black-box and white-box testing techniques.
  - What is boundary value analysis? Why is it important?
  - State any two advantages of using decision table-based testing.
  - Explain the following black-box testing techniques with examples.
    - ✓ Equivalence Class Partitioning
    - ✓ Boundary Value Analysis
  - Describe the following white-box testing techniques with examples:
    - ✓ Statement Coverage
    - ✓ Branch Coverage
  - Design test cases using equivalence partitioning and boundary value analysis for a function that accepts age input between 18 and 60.
  - Given a simple program to calculate grades based on percentage, create a decision table and generate test cases.

References Book:

1. Software Testing Principles and Practices – Srinivasan Desikan, Gopalaswamy Ramesh

2. Foundations of Software Testing – Dorothy Graham, Erik van Veenendaal, Isabel Evans
3. Software Engineering – Ian Sommerville (Relevant QA chapters)
4. Effective Software Testing – Elfriede Dustin
5. Software Quality Assurance – Daniel Galin
6. Testing Computer Software – Cem Kaner

Evaluation Scheme:

External Assessment:

End Semester practical examination of 50 marks will be conducted by the university.

Semester – V

.5.7 Field Project

Course Titles	**Field Project (Pr)**
Course Credits	**2 Credit's**
Content Outline	MCA Integrated students undertaking field projects in Machine Learning (ML), Mobile Applications, and Big Data Analytics should focus on building a complete working application or tool that solves a real-world problem using intelligent data-driven approaches. The project must integrate data processing, model building, and user interaction (mobile/web app). 1. **Problem Identification & Objective Definition** • Select a real-world problem such as: o Smart health monitoring app o AI-based recommendation system o Traffic prediction system o E-learning performance analytics o Fraud detection system • Clearly define: o Problem statement o Objectives o Target users o Expected outcomes (prediction, automation, insights) 2. **Technology Stack Selection** • Machine Learning: Python, Scikit-learn, TensorFlow, PyTorch • Mobile Application Development: Android (Java / Kotlin in Android Studio), Flutter • Big Data Analytics: Apache Hadoop, Apache Spark, MongoDB, Firebase • Backend & APIs: Flask / Django (Python-based APIs), RESTful API integration 3. **System Architecture Design** • Design a multi-layer architecture • System architecture diagram • Data flow diagram (DFD) • UML diagrams 4. **Data Collection & Management** • Large-scale datasets for big data projects

	5. **Data Processing & Analysis** 6. **Machine Learning Model Development** 7. **Application Development** 8. **Big Data Integration** 9. **Testing & Validation** 10. **Deployment** 11. **Documentation** 12. **Final Report & Demonstration**

Internal Assessment:

Internal Assessment Practical/Presentation examination of 50 marks will be conducted at institute level.

Course Syllabus

Semester – VI

.6.1 Major (Core)

Course Titles	**Machine Learning (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand the foundational concepts and algorithms of Machine Learning.
	2. Implement binary, multiclass classification and regression models using real-world datasets.
	3. Apply logic-based and algebraic models to structured learning problems.
	4. Develop end-to-end ML solutions including preprocessing, model selection, evaluation, and reporting.
Module1: Introduction (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Explain basic concepts of supervised, unsupervised, and reinforcement learning.
	2. Identify real-world problems where ML can be applied.
	3. Understand data preprocessing techniques.
Content Outline	• What is Machine Learning? Types and Applications • Supervised vs Unsupervised Learning • Data Cleaning, Feature Scaling, Label Encoding • Overfitting and Underfitting, Bias-Variance Tradeoff • Introduction to Scikit-Learn
Module2:Binary and Multiclass Classification (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Build and evaluate classification models.
	2. Handle imbalanced datasets and multiclass problems.
Content Outline	• **Binary Classification**: Logistic Regression, Naïve Bayes, k-NN • **Multiclass Classification**: One-vs-Rest, One-vs-One strategies, Support Vector Machines (Linear & Kernel) • **Model Evaluation**: Confusion Matrix, Accuracy, Precision, Recall, F1 Score, ROC-AUC • Cross-Validation and Grid Search
Module 3:Regression (Credit1)	

Learning Outcomes	After learning the module, learners will be able to
	1. Develop and evaluate regression models for prediction tasks
Content Outline	• Simple and Multiple Linear Regression • Polynomial Regression • Decision Tree Regression • Evaluation Metrics: MSE, RMSE, MAE, R^2 Score • Feature Selection & Regularization: Lasso, Ridge, ElasticNet
Module 4: Logic Based and Algebraic Models, Trends in Machine Learning (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand logic-based approaches and ensemble methods.
	2. Identify and explore current trends in ML.
Content Outline	• Decision Trees, Rule-Based Systems • Ensemble Methods: Bagging, Random Forest, Boosting (AdaBoost, XGBoost) • Introduction to Clustering: K-Means, Hierarchical • Dimensionality Reduction: PCA, t-SNE • Trends: AutoML, Explainable AI (XAI), MLOps Basics, Ethical AI

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1:** Preprocessing real datasets using Python (Mini-lab)
- **Module 2**: Implement binary and multiclass classification models on UCI datasets
- **Module 3:** Predict housing prices using regression techniques
- **Module 4:** Compare ensemble models on real data and present findings

References:

1. Tom Mitchell, Machine Learning, McGraw Hill
2. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow, O'Reilly
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning
4. Jason Brownlee, Machine Learning Mastery with Python
5. Python and Scikit-learn official documentation

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Project evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

Evaluation Parameters	**Marks**
Functionality (Core Features)	10
Code Quality (Structure, Readability)	10
GUI Design & User Experience	10
Input Handling & Error Management	10
Report writing or Documentation	10

The project internal assessment will be done by conducting project presentation at the department level.

Two sample projects are given below and it is expected from students to choose a good and advanced project topic that aligns with their interests and abilities to work on**.**

Project 1: Disease Diagnosis using ML (Multi-class classification)

Project 2: Stock Price Prediction using Regression Techniques

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

Semester – VI

.6.2 Major (Core)

Course Titles	**Big Data Analytics (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Apply big data analytics approaches, including conceptualization, summarization, and machine learning techniques.
	2. Analyze the characteristics of datasets to distinguish between trivial data and big data for various applications.
	3. Evaluate solutions for problems related to batch learning, online learning, and big data characteristics like high dimensionality and scalability issues.
	4. Design scalable solutions to manage and process dynamically growing big data.
Module1: Introduction (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Apply understanding of fundamental Big Data concepts to address modern data challenges effectively.
	2. Analyze traditional and Big Data approaches to select appropriate technologies for large dataset processing and analysis.
	3. Evaluate infrastructure requirements for efficient Big Data handling, focusing on data analytics for valuable insights extraction.
	4. Design strategies to overcome challenges inherent in Big Data, considering volume, velocity, and variety factors for effective data management and analysis.
Content Outline	• Introduction to Big Data, Big Data Characteristics, Types of Big Data • Traditional Versus Big Data Approach, Technologies Available for Big Data • Infrastructure for Big Data, Use of Data Analytics, Big Data Challenges, Desired Properties of a Big Data System, Case Study of Big Data Solutions
Module2: Analytical Theory and Methods (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply clustering algorithms, association rules, and the Apriori Algorithm to recognize patterns effectively across diverse

	datasets.
	2. Analyse real-world scenarios by applying association rules, mastering validation, and testing for model reliability.
	3. Evaluate linear and logistic regression models, implementing and interpreting regression analyses with proficiency.
	4. Design additional regression models to enhance analytical capabilities for addressing diverse data-driven challenges.
Content Outline	• Clustering and Associated Algorithms, Association Rules, Apriori Algorithm, Candidate Rules • Applications of Association Rules, Validation and Testing, Diagnostics • Regression: Linear Regression, Logistic Regression, Additional Regression Models
Module 3: Modern Big Data Processing Architecture - Spark (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply concepts of modern distributed data processing using Apache Spark.
	2. Analyze architecture of in-memory processing and unified analytics platforms.
	3. Evaluate advantages over traditional systems in terms of speed, scalability, and real-time processing.
	4. Design scalable big data pipelines using modern architectures.
	After learning the module, learners will be able to
Content Outline	• Introduction to Modern Big Data Architecture • Limitations of Traditional Systems (Hadoop-based batch processing) • Overview of Apache Spark • Spark Architecture: Driver, Executors, Cluster Manager • Resilient Distributed Datasets (RDDs), DataFrames, and Datasets • Spark Components: Spark SQL, Spark Streaming, MLlib, GraphX • Real-time Data Processing and Stream Analytics • Data Pipelines and ETL using Spark
Module 4: NoSQL (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply comprehensive learning to grasp NoSQL, covering its definition, business drivers, case studies, and data architectural patterns for informed decision-making in data management.

	2. Analyse the practical application of MapReduce within the new software stack, understanding its role and implementing algorithms for efficient big data processing.
	3. Evaluate the effectiveness of MapReduce in handling large-scale data processing tasks, considering its scalability and performance.
	4. Design strategies to optimize MapReduce algorithms and workflows for specific big data processing requirements, ensuring efficient and effective data processing.
Content Outline	• What is NoSQL?, NoSQL Business Drivers, NoSQL Case Studies, NoSQL Data Architectural Patterns • Variations of NoSQL Architectural Patterns, Using NoSQL to Manage Big Data • **MapReduce:** MapReduce and The New Software Stack, MapReduce, Algorithms Using MapReduce

Activities towards Comprehensive Continuous Evaluation (CCE)

These activities aim to apply theoretical concepts to practical application and critical thinking.

- **Module 1:** Students will conduct research on clustering algorithms, association rules, and the Apriori Algorithm.
- **Module 2:** They will analyze real-world applications of association rules and perform validation and testing of these algorithms.
- **Module 3:** Students will delve into regression analysis, including linear regression, logistic regression, and additional regression models.
- **Module 4:** They will evaluate the effectiveness and limitations of regression models in different data-driven scenarios.

Finally, students will design and implement regression models for specific analytical tasks, applying the learned theories and methods.

References:

1. Shankarmani, R. (2016). Big Data Analytics. Wiley.
2. Chellappan, S., & Acharya, S. (2015). Big Data and Analytics. Wiley.
3. Prajapati, V. (2013). Big Data Analytics with R and Hadoop. Pack Publishing.
4. Dasgupta, N. (2018). Practical Big Data Analytics. Pack Publishing**.**

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Project evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Functionality (Core Features)	10
Code Quality (Structure, Readability)	10
GUI Design & User Experience	10
Input Handling & Error Management	10
Report Writing / Documentation	10

The project Internal assessment will be done by conducting project presentation at the Institute level.

Two sample projects are given below and it is expected from students to choose a good and advanced project topic that aligns with their interests and abilities to work on.

Project 1: Use Apache Spark to analyze a large crime dataset from a major city (e.g., Chicago Crime Data). Extract trends, frequent crimes, and visualize hotspots.

Project 2: Analyze a large dataset of COVID-19-related tweets to find trending hashtags, sentiment patterns, and regional discussions.

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

Course Syllabus

Semester – VI

.6.3 A. Major (Elective)

Course Titles	**Cloud Computing (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Identify security aspects of each cloud model.
	2. Develop a risk-management strategy for migrating to the Cloud. Implement a public cloud instance with a public cloud service provider.
	3. Apply a trust-based security model to different layers.
Module1: Introduction to Cloud Computing (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Define cloud computing and its key characteristics, service models, and deployment models.
	2. Provide an overview of the historical development of cloud computing and its evolution from traditional models.
	3. Compare major cloud service providers like AWS, Azure, and GCP.
	4. Identify and analyze potential security risks and challenges in cloud computing.
Content Outline	• Introduction to Cloud Computing • Online Social Networks and Applications • Cloud introduction and overview • Different clouds, Risks, Novel applications of cloud computing
Module2: Cloud Computing Architecture, Cloud Deployment Models (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Define the requirements driving the emergence of cloud computing and explain CPU virtualization's role.
	2. Provide an overview of basic cloud computing principles, discuss hypervisors, and explain the SPI framework.
	3. Identify key drivers motivating cloud adoption and assess the impact on end-users and businesses.
	4. Explore best practices for establishing effective governance structures in cloud environments.
Content Outline	• Cloud Computing Architecture o Requirements, Introduction Cloud computing architecture

	○ On Demand Computing, Virtualization at the infrastructure level ○ Security in Cloud computing environments, CPU Virtualization ○ Discussion on Hypervisors, Storage Virtualization, Cloud Computing Defined ○ The SPI Framework for Cloud Computing ○ The Traditional Software Model, The Cloud Services Delivery Model • **Cloud Deployment Models** ○ Key Drivers to Adopting the Cloud ○ The Impact of Cloud Computing on Users ○ Governance in the Cloud ○ Barriers to Cloud Computing Adoption in the Enterprise
Module 3: Security Issues in Cloud Computing and Access Management (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply knowledge of infrastructure security in cloud computing to identify key components involved.
	2. Analyse network-level security measures and protocols relevant to cloud environments.
	3. Evaluate application-level security practices and challenges specific to cloud-based applications.
	4. Design strategies to ensure data security and storage in cloud computing environments.
Content Outline	• **Security Issues in Cloud Computing** ○ Infrastructure Security: Network Level, Host Level, Application Level ○ Data Security and Storage, Aspects of Data Security ○ Data Security Mitigation, Provider Data and Its Security • **Identity and Access Management (IAM)** ○ Trust Boundaries and IAM, IAM Challenges ○ Relevant IAM Standards and Protocols for Cloud Services ○ IAM Practices in the Cloud, Cloud Authorization Management
Module 4: Security Management in the Cloud, Privacy Issues (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply security management standards relevant to cloud computing to ensure robust security measures.

	2. Analyse availability management practices for SaaS, PaaS, and IaaS to optimize service availability.
	3. Evaluate risk assessments specific to cloud security and propose effective mitigation strategies.
	4. Design and implement tailored incident response plans for cloud computing scenarios to ensure swift and effective responses.
	After learning the module, learners will be able to
Content Outline	• Security Management in the Cloud • Security Management Standards • **Security Management in the Cloud, Availability Management**: SaaS, PaaS, IaaS • **Privacy Issues** o Data Life Cycle, Key Privacy Concerns in the Cloud o Protecting Privacy, Changes to Privacy Risk Management and Compliance o Legal and Regulatory Implications o U.S. Laws and Regulations, International Laws and Regulations

Activities towards Comprehensive Continuous Evaluation (CCE)

These activites aim to apply theoretical concepts to practical application and critical thinking.

- **Module 1:**
  Research and analyze the evolution of cloud computing and its impact on modern technological landscapes. Identify and explore various online social networks and applications that leverage cloud computing technologies. Develop a comprehensive overview of cloud computing, highlighting different types of clouds, associated risks, and novel applications in diverse domains.

- **Module 2**:
  Investigate the architecture of cloud computing systems, focusing on requirements and essential components. Explore virtualization at the infrastructure level and its role in cloud computing. Evaluate security measures implemented in cloud environments. Examine various cloud deployment models and discuss key drivers influencing cloud adoption.
- **Module 3:**
  Conduct a comprehensive assessment of security issues inherent in cloud computing, covering infrastructure, network, host, and application levels. Explore data security and storage considerations, including mitigation strategies. Investigate identity and access management (IAM) challenges and relevant standards and protocols for cloud services.

- **Module 4:**
  Examine security management standards and practices applicable to cloud computing environments. Analyze availability management for different cloud service models (SaaS, PaaS, IaaS). Investigate privacy concerns related to cloud computing, including data life cycle, key privacy concerns, and compliance with legal and regulatory frameworks.

References:

1. Erl, Z. Mahmood, R. Puttini, Cloud Computing: Concepts, Technology & Architecture, Pearson
2. Bahga, V. Madisetti, Cloud Computing: A Hands-On Approach
3. Sosinsky, Cloud Computing Bible, Wiley
4. Comer, The Cloud Computing Book, CRC Press
5. Fehling et al., Cloud Computing Patterns, Springer
6. J. Garrison, K. Nova, Cloud Native Infrastructure, O'Reilly
7. K. Morris, Infrastructure as Code, O'Reilly
8. T. Mather et al., Cloud Security and Privacy, O'Reilly

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Project evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Functionality (Core Features)	10
Code Quality (Structure, Readability)	10
GUI Design & User Experience	10
Input Handling & Error Management	10
Report Writing / Documentation	10

One sample project is given below and it is expected from students to choose a good and advanced project topic that aligns with their interests and abilities to work on.

Project 1: Develop and deploy a simple web application (e.g., a student feedback system) on the cloud using AWS services. The app must be scalable, fault-tolerant, and include monitoring and logging.

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

Course Syllabus

Semester – VI

.6.3 B. Major (Core)

Course Titles	**Business Analytics Concepts & Applications (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand the role of business analytics in organizational decision-making
	2. Analyze and visualize business data using appropriate tools and techniques
	3. Apply statistical and predictive models to solve business problems
	4. Interpret analytical results and provide actionable business insights
Module 1: Introduction to Business Analytics (Credit 1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Define business analytics and its importance
	2. Identify different types of analytics (descriptive, predictive, prescriptive)
	3. Understand data sources and data-driven decision-making
	4. Explore real-world applications of analytics
Content Outline	• Introduction to business analytics • **Types of analytics:** descriptive, diagnostic, predictive, prescriptive • Role of analytics in business decision-making • Data sources and data collection methods • Case studies of analytics applications in business
Module2: Data Preprocessing and Visualization (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Clean and preprocess data for analysis
	2. Apply data transformation techniques
Content Outline	• Data cleaning and preprocessing • Handling missing values and outliers • Data transformation and normalization • Data visualization techniques (charts, graphs, dashboards) • Tools overview (Excel, Python, Tableau/Power BI)

Module 3: Statistical and Predictive Analytics (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply statistical techniques for data analysis
	2. Understand probability and distributions
	3. Build predictive models
	4. Evaluate model performance
Content Outline	• Descriptive statistics (mean, median, variance) • Probability concepts and distributions • Correlation and regression analysis • Introduction to predictive modeling • Model evaluation techniques
Module 4: Analytics-Driven Decision Making (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply analytics in different business domains
	2. Understand prescriptive analytics and optimization
	3. Interpret analytical results for decision-making
	4. Develop business analytics projects
Content Outline	• Prescriptive analytics and optimization • Analytics applications in marketing, finance, HR, operations • Decision-making using analytics • Ethical issues in business analytics • Mini project development

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1:**
  - Case study analysis on analytics-driven companies
  - Identify analytics applications in real-world scenarios
- **Module 2:**
  - Data cleaning and preprocessing exercise
  - Create visual dashboards using sample datasets
- **Module 3:**
  - Perform statistical analysis using datasets
  - Build simple predictive models
- **Module 4:**
  - Mini Project (Compulsory):
    - ✓ Sales analysis
    - ✓ Customer segmentation

- ✓ Market trend analysis

References:

1. Evans, J. R. – Business Analytics
2. Provost, F., & Fawcett, T. – Data Science for Business
3. Albright, S. C., & Winston, W. – Business Analytics: Data Analysis & Decision Making
4. Sharda, R., Delen, D., & Turban, E. – Business Intelligence and Analytics

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Project evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Analysis	10
Model Building & Implementation	10
Visualization & Interpretation	10
Innovation & Insight Generation	10
Report Writing / Documentation	10

One sample project is given below and it is expected from students to choose a good and advanced project topic that aligns with their interests and abilities to work on.

Projects:

- Customer churn prediction
- Sales forecasting
- Market basket analysis
- Financial risk analysis

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.

Semester – VI

.6.4 Minor Stream

Course Titles	**PLSQL and No SQL LAB (Pr)**
Course Credits	**2 Credit's**
Course Outcomes	After Completion of this Course, students will be able to
	1. Develop database programs using PL/SQL constructs such as procedures, functions, cursors, and triggers.
	2. Apply exception handling and advanced SQL concepts for efficient database operations.
	3. Perform CRUD operations and data modeling using NoSQL databases such as MongoDB.
	4. Compare relational and NoSQL databases and design small real-world applications using both approaches
	After Completion of this Course, students will be able to
Module1: PL/SQL Programming (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Implement control structures such as IF, LOOP, and CASE
	2. Execute SQL statements within PL/SQL blocks
	3. Understand the structure of PL/SQL programs
	4. Develop stored procedures and functions, cursors and triggers
Content Outline	• Structure of PL/SQL block • Variables, constants, and data types • SQL within PL/SQL • Conditional statements (IF, CASE) • Looping statements (FOR, WHILE, LOOP) • Stored procedures and functions • Parameter passing techniques • Cursors (implicit and explicit) • Exception handling • Triggers (BEFORE/AFTER INSERT, UPDATE, DELETE)
Module2: Introduction to NoSQL Databases and Data Modeling (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Create and manage collections in NoSQL databases
	2. Design efficient NoSQL data models
	3. Apply aggregation and indexing techniques

	4. Develop small NoSQL-based applications
Content Outline	• Introduction to NoSQL databases • Types: Key-Value, Document, Column, Graph • Working with MongoDB • CRUD operations • Data modeling (embedded vs referenced documents) • Aggregation framework • Indexing and performance tuning • Mini project using MongoDB

Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- **Write PL/SQL programs for**
  - Arithmetic Operations,
  - Conditional Statements,
  - Loop-Based Computations

- **Develop Database Programs For**
  - Stored Procedures and Functions,
  - Cursor-Based Applications, T
  - Rigger-Based Auditing System

Module 2:

- **Perform operations using NoSQL database:**
  - Create database and collections
  - Insert, update, delete, and retrieve documents
  - Query using operators

- **Mini Project (Compulsory):**

  Design a small application such as:

  - Student database system
  - Product catalog system
  - Social media data storage

References

1. Feuerstein, S. – Oracle PL/SQL Programming
2. Beaulieu, A. – Learning SQL
3. Chodorow, K. – MongoDB: The Definitive Guide
4. Sadalage, P. J. – NoSQL Distilled

Evaluation Scheme

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Problem Understanding & Design	10
Implementation & Functionality	15
Use of Database Concepts	10
Code Quality (Structure, Readability)	5
Testing, Debugging & Error Handling	5
Report Writing / Documentation	5

The project Internal assessment will be done by conducting project presentation at the department level.

Projects

- Banking System using PL/SQL
- E-commerce Catalog using MongoDB
- Library Management System

Semester – VI

.6.5 Minor Stream

Course Titles	**Mobile App Development Technologies (Th+Pr)**
Course Credits	**4 Credit's (2 Th + 2 Pr)**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand mobile app architecture, platforms (Android, iOS), and development environments.
	2. Develop responsive, user-friendly mobile applications using cross-platform frameworks.
	3. Integrate device features like camera, location, and sensors into mobile applications.
	4. Deploy mobile apps and understand app publishing, testing, and security practices.
Module1: Mobile App Fundamentals and Development Tools (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Explain mobile platform differences and app types (native, hybrid, web).
	2. Set up development environments (Android Studio, Flutter, etc.)
	3. Understand app architecture, UI guidelines, and lifecycle.
Content Outline	• Mobile ecosystem: • Android vs iOS, hybrid vs native vs web apps • App architecture (MVC, MVVM) • IDEs: Android Studio, Visual Studio Code • Flutter, React Native, Kotlin, Swift overview • App structure: activities, fragments, views, layouts • Emulator setup and debugging basics
Module2: UI Design, Layouts & Navigation (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Design and implement interactive and responsive UI.
	2. Design interaction with forms, buttons, and menus.
Content Outline	• **Flutter widgets:** text, images, buttons, input fields • **Layouts:** Column, Row, Stack, GridView • Navigation and routing between screens • State management (Provider, setState, Bloc – overview) • App theme and localization

Module 3: Device Integration & Data Handling (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Access mobile hardware features like GPS, camera, and accelerometer.
Content Outline	• **Accessing sensors**: camera, GPS, accelerometer • Handling permissions • **Local storage**: SharedPreferences, SQLite, Hive • **REST APIs**: fetching and sending data using HTTP • JSON parsing • Firebase overview (Auth, Realtime DB)
Module 4: App Testing, Deployment & Security (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Understand app testing strategies and performance optimization.
	2. Prepare and publish apps to Play Store/App Store.
	3. Apply security and privacy measures in mobile apps.
Content Outline	• Debugging and logging • Unit testing and UI testing (Flutter Test, Espresso – overview) • **App performance tips:** lazy loading, image optimization • App signing, APK generation • Publishing to Google Play Store (steps, guidelines) • Best practices: secure API handling, data encryption

Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1:** Build a basic "Hello World" app in Flutter and explore its UI structure
- **Module 2:** Create a multi-screen user registration/login app with form validation
- **Module 3:** Build a location-aware app that captures photos and stores them with GPS data
- **Module 4:** Final mini-project: Fully functional app (e.g., To-do app, Weather app, Expense Tracker) with storage, navigation, and API integration

References:

1. Flutter in Action – Eric Windmill
2. Android Programming: The Big Nerd Ranch Guide – Bill Phillips
3. Beginning iOS Programming with Swift – Simon Ng
4. Pro Android with Kotlin – Peter Späth

Evaluation Scheme:

Internal Assessment:

Depending on the activities mentioned above a project should be developed of 50 marks. Project evaluation to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below).

The Rubric will have the following Evaluation Parameters:

The Rubric will have the following Evaluation Parameters:	
Evaluation Parameters	**Marks**
Functionality (Core Features)	10
Code Quality (Structure, Readability)	10
GUI Design & User Experience	10
Input Handling & Error Management	10
Report Writing / Documentation	10

Two sample projects are given below and it is expected from students to choose a good and advanced project topic that aligns with their interests and abilities to work on.

Project 1: Design a mobile application to help users track their daily health and fitness activities like steps, calories burned, water intake, and sleep hours.

Project 2: Create a mobile app to manage daily expenses, categorize spending, and generate simple reports for budgeting.

External Assessment:

End Semester examination of 50 marks for 2 hours duration will be conducted by the University.