

Krantiguru Shyamji Krishna Verma Kachchh University



CURRICULAM AND CREDIT FRAMEWORK FOR

M.Sc (CA&IT) 4th YEAR (Honors) PROGRAMME

Semester 7 and 8

(Effective from 2026-27)

AS PER THE NEP 2020

M. Sc. (CA&IT) Course Outline 7 th and 8 th Semester (NEP)							
Semester	Course Type	Course Code	Name of the Subject	Theory/ Practical	Marks		Credits
					IA	UA	
7	Major	CAIT-701	Machine Learning	Theory	50	50	4
	Major	CAIT -702	Advanced Python Programming	Theory	50	50	4
	Major	CAIT -703P	LAB : Practical based on CAIT701 and CAIT702	Practical	75	75	4
	Minor	CAIT -704	Full stack web development	Theory	25	25	2
		CAIT -704P	Lab: Full stack web development	Practical	25	25	2
	On the Job Training (OJT)	CAIT -705P	On the Job Training	Viva	75	75	6
					300	300	22
8	Major	CAIT-801	Deep Learning	Theory	50	50	4
	Major	CAIT-802	Advanced Mobile Application Development	Theory	50	50	4
	Major	CAIT-803P	LAB : Practical based on CAIT801 and CAIT802	Practical	75	75	4
	Minor	CAIT-804	DevOps Engineering	Theory	25	25	2
		CAIT-804P	Lab: DevOps Engineering	Practical	25	25	2
	On the Job Training (OJT)	CAIT -805P	On the Job Training	Viva	75	75	6
					300	300	22
	<i>External Exam Hours: 2 Hrs – 4 Credit Course – 50 Marks</i> <i>1 Hr – 2 Credit Course – 25 Marks</i> <i>Passing Marks: 35% CAIT 705P and CAIT 805P are On the Job Training which includes company/Industry training, project and internship.</i>						

M.Sc. (CA&IT) – 3 Years and 4 Years Programme Structure of the University or External Exam for 4 Credit Course		
Q-1 All Units	Objective Questions (It can include: definitions, FIBs, True or false, one line answers, MCQs etc)	10
Q-2 (Unit -1)	Answer two short questions carrying 2 marks respectively (Compulsory) Answer two questions, Short notes carrying 3 marks respectively (2 out of 3)	10
Q-3 (Unit -2)	Answer two short questions carrying 2 marks respectively (Compulsory) Answer two questions, Short notes carrying 3 marks respectively (2 out of 3)	10
Q-4 (Unit -3)	Answer two short questions carrying 5 marks respectively OR Any one question which could be a long question, case study, application of concepts, practical problem etc carrying 10 marks	10
Q-5 (Unit -4)	Answer two short questions carrying 5 marks respectively OR Any one question which could be a long question, case study, application of concepts, practical problem etc carrying 10 marks	10

MSc (CA&IT) – 3 Years and 4 Years Programme Structure of the University or External Exam for 2 Credit Course**		
Q-1 All Units	Objective Questions (It can include: definitions, FIBs, True or false, one line answers, MCQs etc)	05
Q-2 (Unit -1)	Answer two short questions carrying 2 marks respectively (Compulsory) Answer two questions, Short notes carrying 3 marks respectively (2 out of 3)	10
Q-3 (Unit -2)	Answer two short questions carrying 2 marks respectively (Compulsory) Answer two questions, Short notes carrying 3 marks respectively (2 out of 3)	10

Structure of the University External and Internal <u>Practical Exam</u>				
2 Credits Course			4 Credits Course	
Sr.No	Contents	Marks	Contents	Marks
1.	Practical	15	Practical	30
2.	Viva	10	Viva	20
Total		25	Total	50

Structure of the University External Viva and Internal of OJT for 6 Credit Course		
Sr.No	Contents	Marks
1.	Documentation	25
2.	Viva	50
Total		75

MSC (CA&IT) - Semester: VII
(Effective from year 2026-27)

Course Code:	CAIT-701		Course Title:	Machine Learning
Course Credits:	04		Hour of Teaching/Week:	04
Internal Assessment Marks:	50		External Exam Marks:	50

- **CO1:** Students will gain a fundamental understanding of Machine Learning concepts, types of learning, applications, and tools used in Machine Learning systems.
- **CO2:** Students will learn to apply supervised learning techniques such as regression and classification algorithms including Linear Regression, KNN, Decision Tree, and Support Vector Machine (SVM) along with model evaluation methods
- **CO3:** Students will develop the ability to analyze and apply unsupervised learning techniques such as clustering and dimensionality reduction, including K-Means and PCA, for discovering patterns in data.
- **CO4:** Students will understand the principles of reinforcement learning, including key components and basic algorithms such as Q-Learning, and explore their applications in real-world scenarios.
- **CO5:** Students will utilize Python libraries such as Scikit-learn, TensorFlow, Keras, and Matplotlib to implement Machine Learning models and perform data analysis through a case study-based approach.

Unit	Contents
1	Introduction to Machine Learning - Overview of Machine Learning - Types of Machine Learning – Supervised Learning, Unsupervised Learning and Reinforcement Learning - Applications of Machine Learning - Tools and Technology for Machine Learning
2	Supervised Learning - Introduction to Supervised Learning - Regression Techniques – Linear Regression and Multiple Regression - Classification Techniques – K-Nearest Neighbors (KNN), Decision Tree, Support Vector Machine (SVM) - Model Evaluation Techniques – Accuracy, Precision, Recall, Confusion Matrix
3	Unsupervised Learning - Introduction to Unsupervised Learning - Concepts of Clustering - Difference between Classification and Clustering - Clustering Technique – K-Means Clustering - Dimensionality Reduction – Principal Component Analysis (PCA) - Introduction to Association Rule Mining - Applications of Clustering in real-world problems
4	Reinforcement Learning and ML with Python - Introduction to Reinforcement Learning , - Key Components: Agent, Environment, State, Action, Reward - Basic Concept of Q-Learning, - Exploration vs Exploitation (basic idea) - Applications of Reinforcement Learning: Games, Robotics, Automation - Various Python Libraries and their uses for machine learning applications: Scikit-learn, TensorFlow, Keras, Matplotlib - Data Visualization using Matplotlib - Steps in Machine Learning Model building: Data collection and loading, data preprocessing, model training and testing, - Case Study: Simple dataset (Iris/Housing dataset) project

References

1. Pedro Domingos: *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*, Basic Books
2. Stuart Russell and Peter Norvig: *Artificial Intelligence: A Modern Approach*, Pearson Education
3. Nello Cristianini and John Shawe-Taylor: *An Introduction to Support Vector Machines*, Cambridge University Press

Online Resources:

1. <https://nptel.ac.in/courses/106/106/106106139/> NPTEL Machine Learning Course (useful for conceptual understanding and examples)
2. <https://scikit-learn.org> Official Scikit-learn Documentation (for practical implementation)
3. <https://www.kaggle.com/learn> Kaggle Learn (for hands-on practice and datasets)
4. <https://www.coursera.org/learn/machine-learning> Online ML course for additional learning support

MSC (CA&IT) - Semester: VII

(Effective from year 2026-27)

Course Code:	CAIT 702		Course Title:	Advanced Python Programming
Course Credits:	4		Hour of Teaching/Week:	4
Internal Assessment Marks:	50		External Exam Marks:	50

Course Outcomes

After completion of this course, the student will be capable of developing, managing, and maintaining basic applications using Python.

- CO1: Understand and design GUI for Python Desktop Applications
- CO2: Create Web pages for Python Web Applications
- CO3: Introduction to Web Development Frameworks
- CO4: Take backup of databases and CSV files
- CO5: Perform file operations (read, write, etc.)
- CO6: Connect Python with various types of databases
- CO7: Work with SQLite3 database using Python

Unit	Contents
1	Unit 1: Developing Python GUI with Tkinter • Introduction to GUI programming • Importing Tkinter libraries • Tkinter Widgets: Label, Button, Entry, Text, Frame, Canvas, Checkbutton, Radiobutton, Listbox, Combobox, Spinbox, Scale, Menu, Menubutton, OptionMenu, Scrollbar, Notebook, Message, Progressbar, Treeview • Widget attributes: geometry management, event handling • Layout management: pack, grid, place • Designing forms and menus • Handling user input and validation
2	Unit 2: Introduction to Web Development Frameworks • Flask basics: Introduction, routing, templates, forms & request handling • Django basics: Introduction, models, views, templates, forms & user input • Introduction to Django REST Framework, API Development using Django REST Framework, Introduction to FastAPI, FastAPI Project Structure, Path Parameters and Query Parameters, Request and Response Models using Pydantic, CRUD API Development using FastAPI, API Testing using Postman/Swagger, Comparative Study of Django and FastAPI • GIS with Python: Introduction to python GIS libraries use cases, integration with Python (Example of Demographic Dataset Applied Using GeoPandas Library) • Comparison of Flask vs Django
3	Unit 3: Database with python • SQLite dump: specific table, table structure, entire database • Exporting and importing CSV files • Handling CSV with Python: csv module • Data cleaning and preprocessing with Pandas (importing, handling missing values, transformation, filtering & selection, duplicate identification, integration & export) • Backup strategies for databases • Concepts of modules, packages, namespaces • Setting PYTHONPATH, Importing sqlite3 module, Connecting to database, executing queries • Fetching data: fetchone(), fetchall() • CRUD operations: select, insert, update, delete (using tkinter application), Commit and rollback • Error handling in database operations • MongoDB Fundamentals, Collections and Documents, CRUD Operations in MongoDB, Redis Fundamentals, Caching Concepts, Session Storage, Pub/Sub, Async, Queue Management Concepts
4	Unit 4: Authentication and Authorization • Authentication and Authorization Concepts, Session-Based Authentication, Token-Based Authentication and JWT, Password Hashing and Security, Middleware Concepts, • Introduction to Asynchronous Programming, Async and Await in Python, • Background Tasks and API Optimization, API Documentation • Swagger/OpenAPI, API Testing Fundamentals, Logging and Error • Monitoring Basics, Introduction to Docker, Containers and Images • Dockerfile Basics, Deployment Fundamentals, Introduction to Cloud deployment • Concepts, Backend Mini Project / Case Study Development, Best Practices for Scalable Backend Development

Text Books and References:

1. Learning with Python, Allen Downe, DreamTech Press, ISBN: 978-9351198147
2. Python: The Complete Reference, Martin C. Brown, McGraw Hill Education, ISBN: 978-9387572942
3. Python In-Depth, Ahidjo Ayevea, Kamon Ayevea, BPB Publication, ISBN: 978-9389328424
4. The SQLite Handbook, Rita Blackburn, Emereo Publishing, ISBN: 978-1489136459
5. Using SQLite, Jay A. Kreibich, O'Reilly, ISBN: 978-0596521189
6. Android SQLite Essentials, Sunny Kumar Adity, Packt Publishing, ISBN: 978-1783282951

MSC (CA&IT) - Semester: VII
(Effective from year 2026-27)

Course Code:	CAIT 703 P	Course Title:	LAB : Practical based on CAIT701 and CAIT702
Course Credits:	4	Hour of Teaching/Week:	8
Internal Assessment Marks:	50	External Exam Marks:	50

The following activities may be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Programming Proficiency
 - a. Familiarization with Python Programming Fundamentals
 - b. Basic program execution, debugging and error handling
2. Familiarization of Machine Learning environment
 - a. Working with Anaconda environment, Jupyter Notebook and Google Colab
 - b. Understanding libraries such as NumPy, Pandas, Matplotlib, Scikit-learn
 - c. Introduction to datasets and basic data handling techniques
3. Type, Execute, Debug and Implement programs covering Python programming and Machine Learning fundamentals discussed during theory class

List of Sample Programs for CAIT 701

1. Study of Anaconda Environment, Jupyter notebook and Google colab for Machine learning environment
2. Study and implement NumPy and Matplotlib libraries for numerical operations and data visualization
3. Study and implement Pandas library for data manipulation and analysis
4. Implement Linear Regression algorithm using Scikit-learn and plot the graph
5. Implement Logistic Regression algorithm for classification using appropriate dataset
6. Implement K-Nearest Neighbors (KNN) algorithm for classification using Scikit-learn and plot the graph
7. Implement Decision Tree algorithm for classification using suitable dataset
8. Implement Support Vector Machine (SVM) for classification
9. Implement Clustering algorithm using suitable dataset
10. Study and implement basic Reinforcement Learning concepts
11. Develop an Image Recognition system using classification algorithms (CNN/Keras)

List of Practical's Based on CAIT702 Advanced Python Programming

Unit 1: Developing Python GUI with Tkinter

1. Create a simple GUI window with a Label and Button.
2. Design a login form using Entry widgets (username & password) with validation.
3. Build a calculator application using Buttons and Entry.
4. Create a student registration form with multiple widgets (Entry, Radiobutton, Checkbutton, Combobox).
5. Implement a scrollable text editor using Text + Scrollbar.
6. Design a tabbed notebook interface using ttk.Notebook.
7. Create a progress bar demo showing file download simulation.
8. Build a treeview table to display student records.

Unit 2: Introduction to Web Development Frameworks

1. Develop a basic Flask app with routing and HTML templates.
2. Create a Flask form to accept user input and display results.
3. Build a Django project with models, views, and templates for a student database.
4. Implement a Django form for user registration with validation.
5. Use GeoPandas to visualize demographic/geographic dataset on a map.
6. Compare Flask vs Django by building a “Hello World” web app in both frameworks.

Unit 3: Database Backup and CSV Handling

1. Perform SQLite dump of a single table and entire database.
2. Import a CSV file into SQLite and export data back to CSV.
3. Write a Python script using the csv module to read and write CSV files.
4. Use Pandas to clean a dataset:
 - Handle missing values
 - Remove duplicates
 - Filter rows based on conditions
5. Merge two CSV datasets using Pandas and export the cleaned dataset.
6. Demonstrate backup strategy by saving multiple versions of a database.

Unit 4: Python Interaction with SQLite

1. Write a Python script to connect to SQLite and create a table.
2. Perform CRUD operations (insert, update, delete, select) using sqlite3.
3. Fetch single and multiple rows using fetchone() and fetchall().
4. Integrate SQLite with a Tkinter GUI app (e.g., student record management).
5. Demonstrate commit and rollback with transaction handling.
6. Implement error handling for invalid queries or connection issues.
7. Create a modular Python project with packages and namespaces for database operations.

(Effective from year 2026-27)

Course Code:	CAIT 704	Course Title:	Full Stack Development
Course Credits:	2	Hour of Teaching/Week:	2
Internal Assessment Marks:	25	External Exam Marks:	25

CO1: Develop dynamic Single Page Applications using React with components, hooks, routing, and API integration.

CO2: Build scalable backend services using Node.js and Express by implementing RESTful APIs, middleware, and error handling.

CO3: Design and manage MongoDB databases with schema modeling, CRUD operations, and backend integration using Mongoose.

CO4: Implement secure authentication and authorization using JWT, role-based access, and apply web security best practices.

CO5: Deploy full stack applications with proper configuration, ensuring performance, scalability, and real-world usability.

Unit	Description
I	Modern Frontend Development • Introduction to SPA architecture, React fundamentals: Components, JSX, Props, State, Hooks (useState, useEffect), Event handling & forms, Routing using React Router, API integration (Axios/Fetch), State management basics (Context API), UI frameworks (Bootstrap / Tailwind CSS) Backend Development with Node.js Node.js runtime & event-driven architecture, Express.js fundamentals, Routing, middleware, controllers, RESTful API design principles, Request/Response lifecycle, Error handling & validation
II	Database & Integration (MongoDB) • NoSQL concepts, MongoDB CRUD operations, Schema design & relationships, Mongoose ODM, Indexing & performance basics, Connecting frontend + backend Authentication, Security & Deployment Authentication using JWT, Role-based access control, Password hashing (bcrypt), Web security: (XSS, CSRF, Injection attacks), Environment variables & config, Deployment: (Netlify / Vercel (frontend), Render / AWS basics (backend))

Basic Text & Reference Books :-

Reference books

- Banks, A., & Porcello, E. (2020). Learning React: Modern patterns for developing React apps (2nd ed.). O'Reilly Media.
- Holmes, S. (2019). Getting MEAN with Mongo, Express, Angular, and Node (2nd ed.). Manning Publications.
- Tilkov, S., & Vinoski, S. (2020). Node.js: Design patterns (3rd ed.). Packt Publishing.
- Freeman, E., Robson, E., Sierra, K., & Bates, B. (2020). Head first design patterns (2nd ed.). O'Reilly Media.

MSC (CA&IT) - Semester: VII
(Effective from year 2026-27)

Course Code:	CAIT-705P		Course Title:	On the Job Training (OJT)
Course Credits:	6		Hour of Training/Week:	12
Internal Assessment Marks:	75		External Exam Marks:	75

Rules and Regulations for a 6-Credit Individual Software Project (UG Course):

1. On the Job Training - Project Eligibility and Registration:

- Students may be required to have cleared certain prerequisites like approval of the project proposal from competent authority of the institute.
- Project registration will be conducted within a specified timeframe announced by the department.
- **Internship-Based Individual Software Project or Training :** Undertaking a substantial software development project or training module as part of a departmentally approved internship with an external organization.
- All students must individually register their project choice and be assigned a faculty supervisor.
- For internship-based projects, the internship organization and the proposed project/tasks must be approved by the department to ensure academic rigor and relevance.
- Work for Home or Online On the Job Training is allowed based on the type of work to be followed by the students. College principal or Head of Department will be the final authority to approve the Online On the Job Training of the particular student.

2. Training Proposal and Approval:

- **Internship-Based Training or Projects:** Each student must submit a project proposal that includes:
 - Internship organization/Company details and contact person.
 - Description of the Training role and responsibilities.
 - Clear outline of the specific software project or training to be undertaken during the internship.
 - Expected learning outcomes and alignment with coursework.
 - Letter of confirmation/offer from the internship organization.
 - Name and contact details of the industry supervisor/mentor.

- All project or training proposals will be reviewed and approved by a committee constituted by the department.
- Students may be required to present their project proposal individually to the committee for approval.
- Supervisors will be assigned based on the project topic/internship and faculty expertise.

3. Roles and Responsibilities:

- **Student:**
 - Take primary and sole responsibility for all aspects of the project (planning, design, development, testing, documentation, and for internships, active participation).
 - Regularly consult with their faculty supervisor and seek guidance. For Training or internships, also communicate regularly with the industry supervisor.
 - Adhere to the training or project timeline and meet deadlines.
 - Maintain a detailed project logbook/diary documenting their work.
 - Conduct all work ethically and with academic integrity.
 - Prepare and submit progress reports, a comprehensive final project report, and other required deliverables.
 - Present their project work and training experience.
 - Participate in the individual viva voce examination.
- **Faculty Supervisor:**
 - Provide guidance and mentorship to the student throughout the project.
 - Help in defining the project scope and objectives (in collaboration with the industry supervisor for training).
 - Provide feedback on proposals, progress reports, and final reports.
 - Monitor project progress and provide timely advice. For training, maintain communication with the industry supervisor.
 - Assess the student's work and provide feedback for evaluation.
 - Ensure the project aligns with academic standards and ethical guidelines.
- **Industry Supervisor/Mentor:**
 - Provide guidance and mentorship to the student during the internship.
 - Assign relevant software development project tasks and provide necessary resources.
 - Provide feedback on the student's performance and progress to both the student and the faculty supervisor (if required).
 - May contribute to the evaluation process in consultation with the faculty supervisor (e.g., providing a performance assessment).

4. Training Deliverables and Timeline:

- A clear timeline for project completion will be established and communicated. This will be adapted for internship-based projects to accommodate the internship duration.

Key deliverables include:

- Project Proposal (as per the chosen option).
- Training Logbook/Internship Diary.
- Progress Reports (frequency may vary based on the option).
- Draft Final Report.
- Final Project Report (content will vary slightly based on the option, with training or internship reports including details of the organization and the student's role).
- Software Demonstration (for both options).
- Letter of Completion/Performance Certificate from the training or internship organization.
- Specific guidelines for the format and content of the project report and software documentation will be provided by the department.
- Submission deadlines must be strictly adhered to. Late submissions may be subject to penalties.

5. Project Assessment and Evaluation:

- The individual project will be evaluated based on a combination of factors, with slight variations depending on the option:
 - Relevance and academic rigor of the training or internship project, student's engagement and contribution during the internship (based on diary and industry supervisor feedback), integration of training/internship experience in the final report, feedback from the industry supervisor.
- The evaluation will be conducted by a panel of examiners.
- A detailed marking scheme outlining the weightage for each component will be provided.
- Plagiarism in code and documentation is strictly prohibited.

MSC (CA&IT) - Semester: VIII
(Effective from year 2026-27)

Course Code:	CAIT-801	Course Title:	Deep Learning
Course Credits:	4	Hour of Teaching/Week:	4
Internal Assessment Marks:	50	External Exam Marks:	50

- **C01:** Students will gain a fundamental understanding of Deep Learning concepts, including neural networks, architecture, tools, and real-world applications.
- **C02:** Students will learn to design and implement Artificial Neural Network (ANN) models using concepts such as forward propagation, loss functions, optimization techniques, and model evaluation methods.
- **C03:** Students will develop the ability to build and apply Convolutional Neural Networks (CNN) for image processing tasks, including image classification and transfer learning techniques.
- **C04:** Students will understand and apply sequential models such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) for handling time-series and textual data.
- **C05:** Students will explore generative models such as Autoencoders and GANs, and analyze the applications, ethical issues, and future trends of deep learning in various domains.

Unit	Contents
1	Introduction to Deep Learning - Overview of AI, Machine Learning, and Deep Learning - Applications of Deep Learning - Basics of Neural Networks and its architecture - Tools and Libraries for Deep Learning: TensorFlow, Keras, Jupyter, Google Colab
2	Artificial Neural Networks (ANN) - Structure of ANN and Working Mechanism - Forward Propagation - Loss Functions: MSE, Cross Entropy - Training Process and Model Evaluation - Overfitting and Underfitting - Regularization Techniques: Dropout, Early Stopping - Optimizers: SGD, Adam
3	Convolutional Neural Networks (CNN) - Introduction to Image Processing in Deep Learning - CNN Architecture and Components (Convolution, Pooling, Flattening, FC Layer) - Image Classification using CNN, - Transfer Learning - Applications: Object Detection, Face Recognition Sequential Models (RNN & LSTM) - Introduction to Sequential Data, Recurrent Neural Networks (RNN): Concept and Working - Limitations of RNN, - Long Short-Term Memory (LSTM) - Applications: Text Prediction, Time Series forecasting
4	Generative Models & Applications of Deep Learning - Introduction to Autoencoders - Generative Adversarial Networks (GAN): Generator & Discriminator - Deep Learning Applications in: Computer Vision, Natural Language Processing, Healthcare and Finance - Ethical Issues and Future Trends in Deep Learning

Text Books and References:

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville: *Deep Learning*, MIT Press (Latest Edition)
2. François Chollet: *Deep Learning with Python*, Manning Publications
3. Aurélien Géron: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Publication
4. Simon Haykin: *Neural Networks and Learning Machines*, Pearson Education
5. Michael Nielsen: *Neural Networks and Deep Learning*, Determination Press
6. Charu C. Aggarwal: *Neural Networks and Deep Learning*, Springer
7. Josh Patterson and Adam Gibson: *Deep Learning: A Practitioner's Approach*, O'Reilly Publication
8. <https://nptel.ac.in/courses/106/106/106106139/> NPTEL Deep Learning Course
9. <https://scikit-learn.org> Official TensorFlow Documentation
10. <https://keras.io> Keras Documentation and Tutorials
11. <https://www.kaggle.com/learn> Kaggle Learn (Hands-on practice and datasets)

MSC (CA&IT) - Semester: VIII
(Effective from year 2026-27)

Course Code:	CAIT-802		Course Title:	Advanced Mobile Application Development
Course Credits:	4		Hour of Teaching/Week:	4
Internal Assessment Marks:	50		External Exam Marks:	50

- **CO1:** Design and develop responsive cross-platform mobile applications using Flutter widgets, Material UI components, and navigation routes.
- **CO2:** Implement secure Firebase services including authentication, cloud storage, push notifications, and analytics for mobile applications.
- **CO3:** Develop and integrate Node.js + MySQL backend systems to manage data processing, database connectivity, and server responses.
- **CO4:** Build and consume RESTful APIs using CRUD operations, HTTP methods, JSON parsing, and Flutter API integration techniques.
- **CO5:** Create complete data-driven mobile applications using Flutter state management, error handling, token-based authentication, and dynamic CRUD interfaces.

Unit	Contents
1	Frontend Development with Flutter - Introduction to Flutter, Widgets (Stateless & Stateful) , UI design using Material components , Navigation (routes) , Provider , Riverpod , Bloc / Cubit , MVVM architecture , Clean architecture basics Authentication (Very Important) - Firebase Email/Password Login, Google Sign-In ,Phone Authentication (OTP) , Anonymous Login , Password reset & email verification , User session management Firebase Storage - Upload images/files, Download files , Profile picture handling , File security rules
2	Push Notifications - Firebase Cloud Messaging (FCM), Sending notifications , Handling background messages , Topic-based notifications Firebase Analytics - Track user activity, Events tracking, User engagement analysis Node.js + MySQL Integration - Connecting Node.js with MySQL, Using mysql2 package, Executing queries, Handling responses
3	REST API & CRUD Operations - Create → POST API, read → GET API, Update → PUT/PATCH API, delete → DELETE API, Status codes (200, 404, 500) API Integration in Flutter - Making HTTP requests, GET, POST, PUT, DELETE methods, JSON parsing, Model class creation
4	CRUD Operations in Flutter UI - Display data using ListView, Add data using forms, Update data using edit screen , Delete data using button action State Management & Error Handling - setState (basic), Loading indicators, Error handling (try-catch) Basic Authentication - Login system using API, Token-based authentication , APK / AAB build generation

Book and References

Brown, E. (2019). Web development with Node and Express: Leveraging the JavaScript stack (2nd ed.). O'Reilly Media.

Casciaro, M., & Mammino, L. (2020). Node.js design patterns (3rd ed.). Packt Publishing.

Kamal, R. (2007). Mobile computing. Oxford University Press.

Napoli, M. L. (2019). Beginning Flutter: A hands-on guide to app development. Wiley.

Payne, R. (2019). Beginning app development with Flutter: Create cross-platform mobile apps. Apress.

Tahaghoghi, S. M. M., & Williams, H. E. (2007). Learning MySQL. O'Reilly Media.

Talukder, A. K., Yavagal, R. R., & Ahmed, H. (2010). Mobile computing: Technology, applications, and service creation (2nd ed.). McGraw-Hill Education.

MSC (CA&IT) - Semester: VIII
(Effective from year 2026-27)

Course Code:	CAIT-803P	Course Title:	Lab: Practical based on CAIT-801 and CAIT-802
Course Credits:	4	Hour of Teaching/Week:	8
Internal Assessment Marks:	50	External Exam Marks:	50

The following activities may be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Programming and Environment Setup
 - a. Familiarization with Python programming fundamentals required for deep learning
 - b. Installation and configuration of Anaconda, TensorFlow, Keras, and Jupyter Notebook
 - c. Understanding basic program execution, debugging, and error handling
2. Familiarization with Deep Learning Environment and Tools
 - a. Working with Google Colab for executing deep learning models and checking GPU availability
 - b. Understanding libraries such as NumPy, Pandas, Matplotlib, TensorFlow, and Keras
 - c. Introduction to datasets such as MNIST and CIFAR-10 and basic data preprocessing techniques
3. Implementation of Deep Learning Models
 - a. Development of Artificial Neural Network (ANN) models for classification tasks
 - b. Implementation of Convolutional Neural Networks (CNN) for image classification
 - c. Understanding and applying transfer learning using pre-trained models such as VGG16 and ResNet50
4. Sequential and Generative Models
 - a. Implementation of Recurrent Neural Networks (RNN) for sequence prediction and text classification
 - b. Implementation of Long Short-Term Memory (LSTM) networks for sentiment analysis
 - c. Introduction to Generative Adversarial Networks (GAN) and Autoencoders for data generation and reconstruction
5. Program Execution and Model Evaluation
 - a. Type, execute, debug, and analyze deep learning programs
 - b. Evaluate model performance using accuracy and loss metrics
 - c. Interpret results and improve model performance through experimentation

List of Sample Programs

1. Installation and configuration of Deep Learning environment using Anaconda, TensorFlow, Keras, and Jupyter Notebook
2. Study and implementation of Google Colab environment for running deep learning models and checking GPU availability
3. Write a program to implement a basic Artificial Neural Network (ANN) for handwritten digit classification using the MNIST dataset
4. Write a program to design and train a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset
5. Write a program to implement Transfer Learning using a pre-trained VGG16 model for Image Classification
6. Write a program to implement Transfer Learning using a pre-trained ResNet50 model for Image Classification
7. Write a program to implement a Recurrent Neural Network (RNN) for sequence Prediction and Text Classification
8. Write a program to implement a Long Short-Term Memory (LSTM) network for Sentiment Analysis on Text Data
9. Write a program to implement a Generative Adversarial Network (GAN) for generating handwritten digit images
10. Write a program to implement an Autoencoder for Data Compression and Reconstruction

Practical based on Advanced Mobile Application Development

1: Flutter Environment Setup and First App

- Install Flutter SDK and Android Studio/VS Code
- Create and run a basic Flutter application

2: UI Design using Widgets

- Design a multi-screen mobile UI using:
 - Stateless Widgets, Stateful Widgets, Material Design Components

3: Navigation and Routing in Flutter

- Implement:
 - Named Routes, Navigation between multiple screens
 - Passing data between screens

4: State Management using Provider / Riverpod

- Create a counter or cart management application
- Implement centralized state management

5: Firebase Project Configuration

- Configure Firebase in Flutter project, Connect Android app with Firebase Console
- Test Firebase initialization

6: Firebase Email/Password Authentication

- Implement:
 - User Registration, User Login, Logout functionality
 - Session management

7: Google Sign-In Authentication

- Integrate Google Authentication using Firebase
- Retrieve authenticated user details

8: Phone Authentication using OTP

- Implement mobile number login
- Verify OTP using Firebase Authentication

9: Firebase Storage Integration

- Upload profile images/files to Firebase Storage
- Retrieve and display uploaded files, Apply storage security rules

10: Push Notifications using Firebase Cloud Messaging (FCM)

- Configure FCM in Flutter, Receive foreground and background notifications
- Display local notifications

11: Firebase Analytics Implementation

- Track:
 - Screen visits, Button click events, User engagement activities

12: Node.js and MySQL Connectivity

- Create a simple Node.js server, Connect Node.js with MySQL database
- Execute CRUD SQL queries using mysql2 package

Practical 13: REST API Development using Node.js

- Develop REST APIs:
 - GET, POST, PUT/PATCH, DELETE
- Test APIs using Postman

Practical 14: API Integration in Flutter

- Consume REST APIs in Flutter using HTTP package
- Perform:
 - JSON parsing, GET and POST requests, Display dynamic data in UI

Practical 15: Complete CRUD Mobile Application

- Develop a Flutter application with:
 - Login Authentication
 - REST API integration
 - Add/Edit/Delete operations
 - Loading indicators
 - Error handling using try-catch
 - APK/AAB generation

MSC (CA&IT) - Semester: VIII

(Effective from year 2026-27)

Course Code:	CAIT-804		Course Title:	DevOps Engineering
Course Credits:	2		Hour of Teaching/Week:	2
Internal Assessment Marks:	25		External Exam Marks:	25

CO1: Explain DevOps lifecycle, CI/CD, and collaborative development practices.

CO2: Describe cloud, container, orchestration, and monitoring architectures.

CO3: Use Git and Docker for software delivery workflows.

CO4: Deploy applications using Kind / Minikube and automate tasks using Ansible.

CO5: Implement monitoring dashboards using Prometheus and Grafana.

Unit	Contents
1	Introduction to DevOps - Traditional SDLC vs DevOps, DevOps lifecycle, Agile and collaboration culture CI/CD Concepts - Continuous Integration, Continuous Delivery, Continuous Deployment Linux Fundamentals - Linux architecture, File systems, Users and permissions Networking Basics - IP Addressing, DNS, Ports, HTTP / HTTPS / SSH Version Control Concepts - Source control systems, Branching strategies, Merge workflows
2	Virtualization vs Containerization - Introduction to virtualization and containers, Virtual Machines and Hypervisors, Containers as lightweight virtualization, Difference between VMs and Containers, Benefits and use cases Container Concepts - Images, Containers, Registries Orchestration Concepts - Cluster architecture, Pods, Services, Scaling Infrastructure as Code - Declarative infrastructure, Automation concepts Monitoring & Security - Logs, Metrics, Alerts, Secrets management

Text Books & References

- Kim, G., Humble, J., Debois, P., & Willis, J. (2021). The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations (2nd ed.). IT Revolution Press.
- Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley.
- Turnbull, J. (2014). The Docker book: Containerization is the new virtualization. James Turnbull.
- Burns, B., Beda, J., & Hightower, K. (2019). Kubernetes: Up and running: Dive into the future of infrastructure (2nd ed.). O'Reilly Media.

MSC (CA&IT) - Semester: VIII
(Effective from year 2026-27)

Course Code:	CAIT-804P		Course Title:	LAB: DevOps Engineering
Course Credits:	2		Hour of Teaching/Week:	4
Internal Assessment Marks:	25		External Exam Marks:	25

List of Practical for CAIT 804P**Linux Practical**

- Basic Linux commands: ls, cd, pwd, mkdir, rm, cp, mv
- File permissions: chmod, chown
- User and group management

- Process management and package installation
- SSH remote login basics

Git Practical

- Install and configure Git
- Create local repository
- Add, commit and view logs
- Branch creation and merging
- Push and pull from remote repository
- Conflict resolution basics

Docker Practical

- Install Docker engine, Basic Docker commands, Pull and run images
- Create Dockerfile, Build custom images, Container start, stop, remove
- Volumes and networking basics

CI/CD Practical

- Introduction to CI/CD pipeline
- Create workflow using GitHub Actions / Jenkins
- Build automation basics, Test automation basics
- Deployment workflow basics

Mini Application Deployment

- Deploy simple web application in container
- Expose application ports, Update containerized application

Troubleshooting container issues

Kind / Minikube Practical

- Create local cluster, Cluster status and node management
- Deploy pods and deployments, Scale applications
- Delete and recreate resources

Kubernetes Services Practical

- Create services for applications, Expose applications internally and externally
- Load balancing basics, Rolling update basics

Terraform Practical

- Install Terraform, Initialize project, Create basic configuration files
- Variables and outputs, Apply and destroy resources

Ansible Practical

- Install Ansible, Inventory file creation
- Ping managed nodes, Ad-hoc commands
- Create simple playbooks

Monitoring & Server Management

- Install NGINX, Reverse proxy configuration
- Install Prometheus, Configure basic monitoring targets
- Create dashboards in Grafana

MSC (CA&IT) - Semester: VIII
(Effective from year 2025-26)

Course Code:	CAIT-805P		Course Title:	On the Job Training (OJT)
Course Credits:	6		Hour of Training/Week:	12
Internal Assessment Marks:	75		External Exam Marks:	75

Rules and Regulations for a 6-Credit Individual Software Project (UG Course):

1. On the Job Training - Project Eligibility and Registration:

- Students may be required to have cleared certain prerequisites like approval of the project proposal from competent authority of the institute.
- Project registration will be conducted within a specified timeframe announced by the department.
- **Internship-Based Individual Software Project or Training :** Undertaking a substantial software development project or training module as part of a departmentally approved internship with an external organization.
- All students must individually register their project choice and be assigned a faculty supervisor.
- For internship-based projects, the internship organization and the proposed project/tasks must be approved by the department to ensure academic rigor and relevance.
- Work for Home or Online On the Job Training is allowed based on the type of work to be followed by the students. College principal or Head of Department will be the final authority to approve the Online On the Job Training of the particular student.

2. Training Proposal and Approval:

- **Internship-Based Training or Projects:** Each student must submit a project proposal that includes:
 - Internship organization/Company details and contact person.
 - Description of the Training role and responsibilities.
 - Clear outline of the specific software project or training to be undertaken during the internship.
 - Expected learning outcomes and alignment with coursework.
 - Letter of confirmation/offer from the internship organization.
 - Name and contact details of the industry supervisor/mentor.

- All project or training proposals will be reviewed and approved by a committee constituted by the department.
- Students may be required to present their project proposal individually to the committee for approval.
- Supervisors will be assigned based on the project topic/internship and faculty expertise.

3. Roles and Responsibilities:

- **Student:**
 - Take primary and sole responsibility for all aspects of the project (planning, design, development, testing, documentation, and for internships, active participation).
 - Regularly consult with their faculty supervisor and seek guidance. For Training or internships, also communicate regularly with the industry supervisor.
 - Adhere to the training or project timeline and meet deadlines.
 - Maintain a detailed project logbook/diary documenting their work.
 - Conduct all work ethically and with academic integrity.
 - Prepare and submit progress reports, a comprehensive final project report, and other required deliverables.
 - Present their project work and training experience.
 - Participate in the individual viva voce examination.
- **Faculty Supervisor:**
 - Provide guidance and mentorship to the student throughout the project.
 - Help in defining the project scope and objectives (in collaboration with the industry supervisor for training).
 - Provide feedback on proposals, progress reports, and final reports.
 - Monitor project progress and provide timely advice. For training, maintain communication with the industry supervisor.
 - Assess the student's work and provide feedback for evaluation.
 - Ensure the project aligns with academic standards and ethical guidelines.
- **Industry Supervisor/Mentor:**
 - Provide guidance and mentorship to the student during the internship.
 - Assign relevant software development project tasks and provide necessary resources.
 - Provide feedback on the student's performance and progress to both the student and the faculty supervisor (if required).
 - May contribute to the evaluation process in consultation with the faculty supervisor (e.g., providing a performance assessment).

4. Training Deliverables and Timeline:

- A clear timeline for project completion will be established and communicated. This will be adapted for internship-based projects to accommodate the internship duration.

Key deliverables include:

- Project Proposal (as per the chosen option).
- Training Logbook/Internship Diary.
- Progress Reports (frequency may vary based on the option).
- Draft Final Report.
- Final Project Report (content will vary slightly based on the option, with training or internship reports including details of the organization and the student's role).
- Software Demonstration (for both options).
- Letter of Completion/Performance Certificate from the training or internship organization.
- Specific guidelines for the format and content of the project report and software documentation will be provided by the department.
- Submission deadlines must be strictly adhered to. Late submissions may be subject to penalties.

5. Project Assessment and Evaluation:

- The individual project will be evaluated based on a combination of factors, with slight variations depending on the option:
 - Relevance and academic rigor of the training or internship project, student's engagement and contribution during the internship (based on diary and industry supervisor feedback), integration of training/internship experience in the final report, feedback from the industry supervisor.
- The evaluation will be conducted by a panel of examiners.
- A detailed marking scheme outlining the weightage for each component will be provided.
- Plagiarism in code and documentation is strictly prohibited.