23CS038	PYTHON FULL STACK DEVELOPMENT WITH MACHINE LEARNING	L	T	P	C
		2	0	2	3

COURSE OBJECTIVES:

- To learn foundational backend development concepts using Python's Flask for API development
- To integrate advanced backend features to ensure secure, efficient, and scalable APIs.
- To build and train machine learning models using Scikit-Learn, focusing on data preprocessing, model evaluation, and tuning
- To integrate machine learning models within backend APIs to enable real-time predictions
- To deploy machine learning applications on Render with CI/CD pipelines and monitoring for production stability

UNIT I	PYTHON FOR BACKEND DEVELOPMENT	6

Backend Fundamentals and REST API Concepts – RESTful architecture, HTTP methods (GET, POST, PUT, DELETE), resource-based endpoint design, best practices for REST API design; **Flask Essentials** – Setting up Flask, routing and request handling, working with JSON, custom error handling; **Building CRUD APIs:** Implementing create, read, update, and delete operations using Flask-Introduction to database interactions using SQLite or in-memory data handling for testing.

UNIT II	ADVANCED BACKEND TECHNIQUES	6

API Security and Authentication – JWT authentication, Flask-JWT-Extended, role based access control;
Data Processing and Serialization – Handling large datasets in FLASK, using JSON and XML data serialization formats;
Implementing Caching and Redis– Introduction to Redis, Flask-

Redis integration, managing cache expiry and invalidation.

UNIT III	MACHINE LEARNING FUNDAMENTALS	6

Types of Machine Learning – Supervised, unsupervised, and reinforcement learning, Supervised Learning Models; **Data Preprocessing and Feature Engineering**– Data cleaning techniques, Scaling and Normalization, Feature Selection and Engineering; **Building Machine Learning Models** – Linear regression and decision trees, Random Forest and SVM; **Model Evaluation and Optimization** – Metrics for evaluation, cross-validation techniques, hyperparameter tuning.

UNIT IV	MACHINE LEARNING MODEL INTEGRATION	6

Exposing ML Models through APIs - Creating prediction endpoints in Flask, Formatting input data for predictions and handling JSON requests; **Data Processing for Model Inference -** Data Formatting and Validation , Batch Processing for Efficiency: **Optimizing and Scaling Model Serving-** Techniques for faster inference, asynchronous processing for handling large volumes of requests; **Monitoring and Logging Predictions** - Logging incoming prediction requests and analyzing data distribution, Health Checks and Error Tracking.

UNIT V	DEPLOYMENT AND PRODUCTION READINESS	6

Render Deployment Essentials– Setting up a Render account and deploying Flask applications, Environment Configuration; **Preparing ML Models for Deployment**- Packaging models and dependencies for production, Creating Docker containers for scalable deployments; **CI/CD with GitHub Actions -** Setting up GitHub Actions for automated builds and deployments,

Monitoring and Logging for Production APIs- Real-time Logging, Error Handling and Alerting.

	TOTAL: 30 PERIODS

PRACTICAL EXERCISES:

LIST OF EXPERIMENTS

1. Basic CRUD API Creation: Develop a CRUD API for managing a library of books with operations for adding, viewing, editing, and deleting records.

2. Implementing JWT Authentication: Set up JWT authentication to secure the library API.

3. Using Redis Caching: Add Redis caching to cache frequently accessed endpoints, such as the "View All Books" endpoint

4. Data Cleaning and Feature Engineering: Clean a housing dataset and create engineered features to improve predictive performance.

5. Model Building and Evaluation: Train a classification model using a dataset, evaluating it with accuracy and F1 score metrics.

6. Model Prediction API: Develop a Flask API to serve predictions from a trained ML model.

7. Prediction Logging: Set up basic logging to track incoming requests and analyze prediction patterns.

8. Deploying Flask API on Render: Deploy a Flask-based API on Render, including environment configuration and monitoring setup.

9. CI/CD Setup with GitHub Actions: Automate deployment of the API with CI/CD, ensuring consistent updates on each code commit

Mini Projects

1. Book Recommendation API: Build an API using Flask that

provides book recommendations based on genre and author. Integrate data validation to ensure API requests have the required fields.

2. User Profile API with JWT and Redis: Create a Flask API where users can view and update their profiles. Implement JWT-based authentication and use Redis to cache user data for improved performance.

3. Movie Rating Predictor: Develop a regression model to predict user ratings for movies based on genre, director, and other features. Tune the model using cross-validation to optimize accuracy.

4. Spam Detection API: Develop an API using a pre-trained spam detection model to classify messages. Implement logging to track prediction accuracy over time.

5. Sentiment Analysis API with CI/CD on Render: Develop and deploy a sentiment analysis API, set up CI/CD on Render to automate redeployment, and implement monitoring.

	TOTAL: 30 PERIODS
COURSE OUTCOMES:	
	After completion of the course, the students will be able to:
CO1:	Design and implement RESTful APIs using Python and Flask framework.
CO2:	Apply authentication, authorization, and caching mechanisms to secure and optimize backend applications.
CO3:	Preprocess data and build machine learning models using Scikit-Learn for regression and classification tasks.
CO4:	Integrate trained machine learning models into Flask APIs for real-time prediction and analysis.
CO5:	Monitor and log backend systems to ensure robustness and performance in API services.

CO6:	Deploy full-stack machine learning applications using Render and GitHub Actions with CI/CD practices
TEXT BOOKS:	
1	Miguel Grinberg, Flask Web Development, 2nd Edition, O'Reilly Media, 2018.
2	Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, O'Reilly, 2019.
3	Sebastian Raschka, Python Machine Learning, 3rd Edition, Packt Publishing, 2019.
REFERENCES:	
1	Mark Bates, Programming Flask, Pragmatic Bookshelf, 2022.
2	Jason Brownlee, Machine Learning Mastery With Scikit-Learn, 2021.
3	GitHub Docs: https://docs.github.com/
4	Flask Docs: https://flask.palletsprojects.com/
5	Render Docs: https://render.com/docs

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	2	1	-	-	-	-	-	-	-	-	-	-	2	-	-
2	3	2	1	1	1	-	-	1	-	1	-	1	3	1	1
3	3	2	1	1	1	-	-	1	-	1	-	1	3	1	1
4	3	2	1	1	-	-	-	-	-	-	-	-	3	-	-
5	3	2	1	1	-	-	-	-	-	-	-	-	3	-	-
6	3	2	1	1	1	-	-	1	-	1	-	1	3	1	1
Overall Correlation	3	2	1	1	1	-	-	1	-	1	-	1	3	1	1