COURSE TITLE	**INTELLIGENT WEB DESIGNING**
COURSE CODE	**01AL0501**
COURSE CREDITS	**2**

Objective:

1 This course aims to equip students with practical skills to bridge the gap between AI/ML models and user-facing applications. The objective is to teach students how to develop modern web interfaces, build robust backend APIs, and successfully integrate and deploy intelligent features (such as chatbots, recommendation engines, and computer vision models) into web applications. Students will learn to build full-stack web applications that integrate AI/ML models, provide interactive data visualizations, and utilize industry-standard version control and deployment workflows.

Course Outcomes: After completion of this course, student will be able to:

1 Implement version control workflows using Git and GitHub for collaborative development.

2 Develop responsive web interfaces featuring interactive data visualizations (using D3.js, Chart.js, or Plotly)

3 Develop backend RESTful APIs to serve machine learning models

4 Integrate pre-trained AI models (NLP, Computer Vision, Speech) into web environments

5 Deploy containerized intelligent web applications on cloud platforms

Pre-requisite of course: Basics of HTML and CSS, Programming in Python, Basics of Machine Learning.

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
0	0	4	50	0	0	25	25

Contents : Unit	Topics	Contact Hours
Total Hours		

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Web Fundamentals & Version Control** Git workflow (init, commit, branch, merge, GitHub collaboration; HTML5/CSS3 and JavaScript (ES6+), Introduction to Frontend Frameworks, State management, and building responsive UI components., Experiment 1: Initialize a Git repository and host a responsive static "AI Project Portfolio" on GitHub.	4

DR. TARANNUM ALIMAHMAD BLOCH
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
2	**Data Visualization for the Web** Integrating visualization libraries (Chart.js, Plotly.js, or D3.js), Binding web components to live data, Creating interactive dashboards and AI model performance charts., Experiment 2: Build a responsive UI that dynamically updates text based on user input using JavaScript., Experiment 3: Visualization: Create an interactive Line and Bar chart dashboard using Chart.js to display dataset statistics.	4
3	**Backend APIs & AI Integration** Introduction to Python web frameworks (Flask or FastAPI), Building RESTful APIs (FastAPI/Flask), Serving machine learning models via APIs, integrating NLP : sentiment analysis, chatbots using Dialogflow/OpenAI APIs, implementing computer vision features: image upload and processing and Web Speech APIs, Connecting frontend to ML models :NLP/Vision, Handling real-time data streams and JSON communication, Experiment 5: Develop a FastAPI/Flask backend to serve a simple Sentiment Analysis model, Experiment 6: Connect the Frontend to the Backend API to display model predictions in real-time, Experiment 7: Integrated Visuals: Build a "Model Performance Dashboard" showing Confusion Matrices and Accuracy scores via the web, Experiment 8: Implement an Image Classification interface where uploaded images are processed by an AI backend, Experiment 9: Implement the Web Speech API for voice-to-text input within the web application	14
4	**Deployment and Cloud Hosting** GitHub Pages for static hosting, Introduction to containerization (Docker basics), packaging web apps and models, and deploying applications to cloud platforms e.g., Heroku, Render, AWS, or Vercel, Experiment 11: Containerize the entire application (Frontend + Backend + ML Model) using Docker., Experiment: Deployment: Deploy the final application to a cloud platform with a live URL. (e.g., Render/Vercel), Experiment 12: Final Lab Project: End-to-end Intelligent Web App with Data Viz, hosted on GitHub.	12
Total Hours		**34**

Textbook :

1 Building Machine Learning Powered Applications: Going from Idea to Product, Emmanuel Ameisen, O'Reilly Media, 2020

2 Flask Web Development: Developing Web Applications with Python, Miguel Grinberg , O'Reilly Media, 2018

3 Practical Deep Learning for Cloud, Mobile, and Edge, Anirudh Koul, Siddha Ganju, Meher Kasam, O'Reilly Media, 2019

4 Learning React: Modern Patterns for Developing React Apps, Alex Banks & Eve Porcello, O'Reilly Media, 2020

References:

1 Interactive Data Visualization for the Web, Interactive Data Visualization for the Web, Scott Murray, O'Reilly Media, 2017

2 Designing Web APIs, Designing Web APIs, Brenda Jin, Saurabh Sahni, Amir Shevat, O'Reilly Media , 2018

3 Docker Deep Dive, Docker Deep Dive, Nigel Poulton, Leanpub, 2023

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
10.00	15.00	30.00	15.00	10.00	20.00

Instructional Method:

1 The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

2 The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.

3 Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

4 Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory

DR. TARANNUM ALIMAHMAD BLOCH
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)