FRONTEND–BACKEND INTEGRATION FOR FIRE WEATHER INDEX (FWI) PREDICTION

1. Introduction

The Fire Weather Index (FWI) is used to predict the risk of fire based on environmental conditions. During my internship, I worked on developing a system that connects a frontend application with a backend server to predict FWI values. The main goal of this project is to provide real-time FWI prediction through a user-friendly web interface.

2. Objective of the Project

- To design a frontend interface for user input
- To connect the frontend with the backend using APIs
- To predict Fire Weather Index using a trained model
- To display the predicted FWI result on the frontend

3. System Architecture

The system follows a client–server architecture.

- Frontend acts as the client where users enter input data
- Backend processes the data and performs prediction
- Communication happens through REST API

4. Frontend Development

The frontend is developed using **HTML, CSS, and JavaScript**. It contains input fields for environmental parameters such as temperature, humidity, wind speed, and rainfall. JavaScript is used to collect user input and send it to the backend server using HTTP requests.

5. Backend Development

The backend is developed using **Python and Flask**. It receives input data from the frontend in JSON format. The backend performs data preprocessing and passes the data to the trained FWI prediction model. After prediction, the result is sent back to the frontend.

6. Frontend–Backend Communication

The frontend communicates with the backend using **HTTP POST requests**.

- User inputs are sent as JSON data
- Backend processes the request
- Predicted FWI value is returned as a response
  This ensures smooth and real-time interaction between both layers.

7. Fire Weather Index Prediction

The backend uses a machine learning model trained on environmental data to predict the Fire Weather Index. The model analyzes the input parameters and generates an FWI value, which indicates the level of fire risk.

8. Result Display

Once the prediction is completed, the backend sends the FWI value to the frontend. The frontend dynamically displays the result in a clear and readable format, making it easy for users to understand the fire risk level.

9. Technologies Used

- Frontend: HTML, CSS, JavaScript
- Backend: Python, Flask
- Machine Learning: FWI Prediction Model
- API: REST API
- Data Format: JSON

10. Advantages of the System

- Real-time FWI prediction
- User-friendly interface
- Efficient frontend–backend integration
- Scalable and easy to extend

11. Conclusion

This project helped me gain practical experience in frontend and backend integration. I learned how to build APIs, handle client–server communication, and deploy a machine learning prediction system. The system successfully predicts Fire Weather Index and displays the result through a web interface.