

Project Design Phase-II Technology Stack (Architecture & Stack)

Date	27 February 3035
Team ID	SWTID-2026-5418
Project Name	Insurance Fraud Detection Using Maching Learning
Maximum Marks	4 Marks

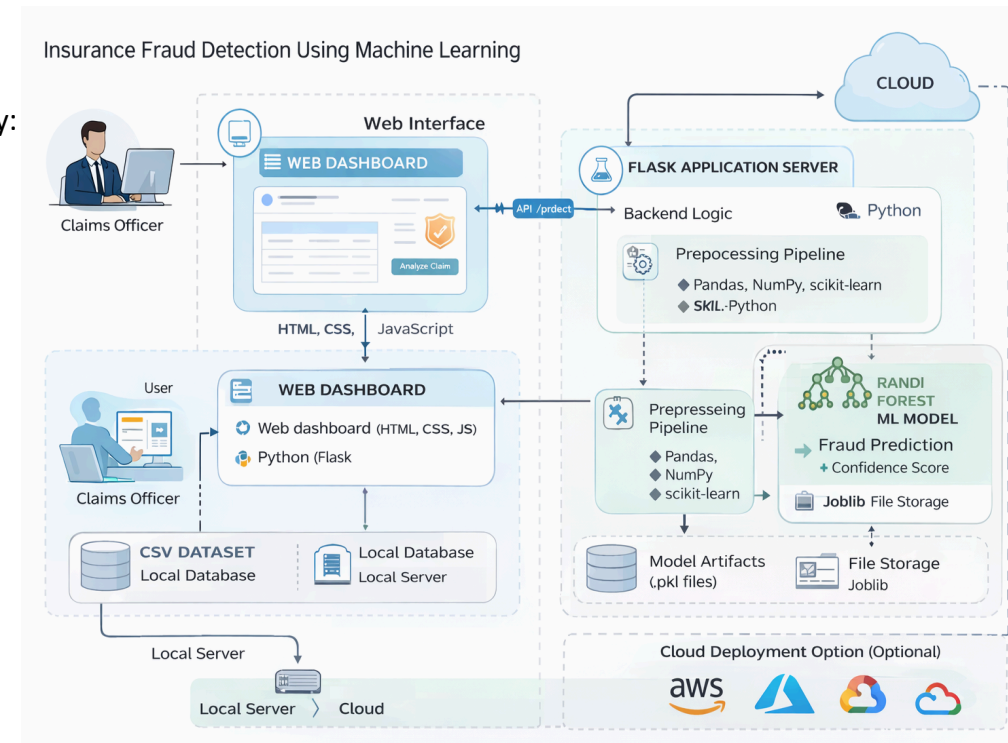
Technology Stack and Architecture Overview

The solution employs a **3-Tier Architecture** for robust and scalable fraud detection, utilizing a Python-based machine learning model and a web interface.

1. High-Level

The system follows a sequential data flow from user input to result display:

1. **User (Claims Officer)** inputs claim data via the Web Dashboard.
2. **Web Dashboard (HTML, CSS, JS)** sends the data to the server.
3. **Flask Application Server (Python)** receives the request.
4. **Preprocessing Pipeline** cleans and transforms the data (Pandas, NumPy, scikit-learn).
5. **Random Forest ML Model** generates a prediction.
6. **Prediction + Confidence Score** is determined.
7. **Results Displayed on Dashboard** for the user.



2. Key Components & Technologies (Table-1)

S.No	Component	Description	Technology
1	User Interface (Presentation Layer)	Web dashboard for claim detail entry and fraud prediction visualization.	HTML, CSS, JavaScript (Chart.js for visualization)
2	Application Logic (Application Layer)	Backend logic for request handling, data processing, and serving the REST API.	Python (Flask Framework), Flask API (/api/predict)
3	Data Preprocessing	Essential data preparation: encoding, scaling, and feature selection.	Pandas, NumPy, scikit-learn
4	Machine Learning Model (Data Layer)	Classification algorithm for insurance fraud prediction.	Random Forest (scikit-learn)
5	Data & Model Storage	Storage for the claims dataset and ML model artifacts (trained model, scaler).	CSV Dataset (Local Storage), Joblib (.pkl files)
6	Infrastructure & Deployment	Environment for application runtime and future deployment strategy.	Local Server (Flask), Future: Docker, Cloud (AWS/Azure/GCP)
7	Model Evaluation	Performance analysis and metric calculation for the ML model.	scikit-learn Metrics

3. Application Characteristics (Table-2)

S.No	Characteristic	Description	Technology/Implementation
1	Architecture Type	Modular structure separating UI, logic, and data/ML components.	3-Tier Architecture
2	Frameworks	Utilizes universally accessible and modifiable software libraries.	Open-Source: Flask, scikit-learn, Pandas
3	Performance	Optimized for quick fraud inference and efficient data transformation.	Fast inference (<1 sec prediction), Optimized Random Forest
4	Reliability & Consistency	Standardized data preparation to eliminate issues like data leakage.	StandardScaler + Stratified Split preprocessing pipeline
5	Optimization	Improved model efficiency by significantly reducing the number of input features.	Feature Importance Analysis (Reduced 40+ features to 8)
6	Security	Measures implemented to handle user input and API data safely.	Flask validation, Structured JSON handling
7	Availability & Scalability	Designed for flexible deployment environments and modular expansion.	Flask + Cloud Deployment Ready