**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 31 January 3035 |
| Team ID | LTVIP2025TMID35292 |
| Project Name | **TrafficTelligence**: advanced traffic volume estimation with machine learning |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>**

Guidelines

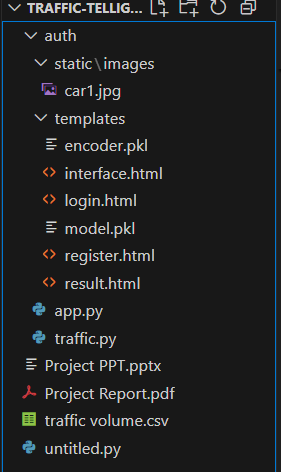
**Frontend (HTML Templates)**  
interface.html, login.html, register.html, result.html

**Backend Block**

app.py block for backend and database integration

**ML Models**

encoder.pkl and model.pkl files for program modelling.In this it also includes the data preprocessing,scaling blocks.



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for user registration, login, and prediction input | HTML, CSS, JavaScript |
|  | Application Logic-1 | Handles routing, user registration, OTP, and model processing | Java / Python |
|  | Application Logic-2 | OTP generation and email delivery | IBM Watson STT service |
|  | Application Logic-3 | User data validation and MongoDB CRUD operations | IBM Watson Assistant |
|  | Database | Stores user data and credentials | MySQL, NoSQL, etc. |
|  | Cloud Database(Optional) | Database Service on Cloud | IBM DB2, IBM Cloudant etc. |
|  | File Storage | Stores model.pkl and encoder.pkl files | Local Filesystem (`model.pkl`, `encoder.pkl`) |
|  | External API-1 | Sending OTP emails | Gmail SMTP API |
|  | Machine Learning Model | Predicts traffic volume based on input parameters | XGBoost Regressor + Scikit-learn (Scaler/Encoder |
|  | Infrastructure (Server / Cloud) | Local deployment with Flask server | Localhost (Flask on port 5001) |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Flask used for web server,  Scikit-learn & XGBoost for ML,  Pandas, NumPy for data processing | Flask, Scikit-learn, XGBoost, Pandas, NumPy |
|  | Security Implementations | OTP-based login via email,  password stored securely,  email auth through SMTP | OTP, Email Verification, Gmail SMTP |
|  | Scalable Architecture | 3-tier architecture with UI, Application Logic, and ML Model layers. Easily extendable to cloud | Flask MVC, XGBoost model as separate logic |
|  | Availability | Application hosted on Flask local server | Flask (Local) |
|  | Performance | Lightweight backend, quick predictions (<2s), pre-loaded model and scaler in memory | Scikit-learn, XGBoost, Flask, Pickle |

**References:**

**<https://c4model.com/>**

**<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>**

**<https://www.ibm.com/cloud/architecture>**

**<https://aws.amazon.com/architecture>**

**<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>**