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

Date	18 February 2026
Team ID	LTVIP2026TMIDS84143
Project Name	Electric Motor Temperature Prediction using Machine Learning
Maximum Marks	4 Marks

Technical Architecture:

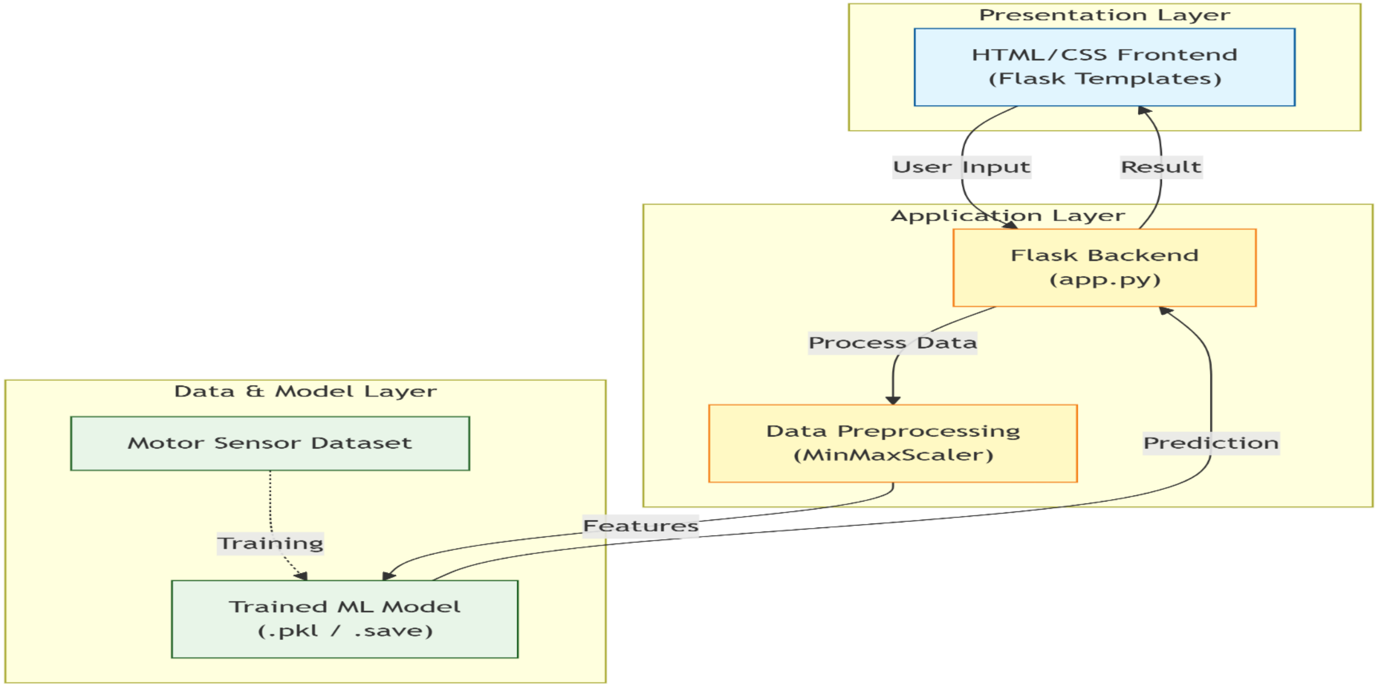

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web interface where user enters motor parameters and views predicted temperature	HTML, CSS, Flask (Jinja2 Templates)
2.	Application Logic-1	Backend logic for handling user input, validation, routing and response handling	Python, Flask
3.	Application Logic-2	Data preprocessing logic including feature scaling and transformation	Python, Pandas, NumPy, Scikit-learn (MinMaxScaler)
4.	Application Logic-3	Machine learning prediction logic using trained regression model	Scikit-learn (Regression Model), Pickle
5.	Database	Dataset storage for training phase (CSV format)	Local CSV File (Pandas DataFrame)
6.	Cloud Database	Not currently implemented (can be extended for real-time data storage)	Future Scope – AWS RDS / Firebase / MongoDB Atlas
7.	File Storage	Storage of trained ML model file and dataset	Local File System (.pkl, .save, .csv files)
8.	External API-1	Not used currently (can integrate real-time sensor API in future)	Future Scope – IoT Sensor API
9.	External API-2	Not applicable in current version	-
10.	Machine Learning Model	Predict Permanent Magnet (PM) Temperature based on motor sensor inputs	Scikit-learn Regression Model
11.	Infrastructure (Server / Cloud)	Application deployed locally using Flask development server	Local Server (Flask), Future Scope – AWS / Render / Heroku

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The application uses open-source frameworks for web development and machine learning	Flask, Scikit-learn, Pandas, NumPy
2.	Security Implementations	Input validation to prevent invalid data submission, basic server-side validation	Flask Validation, Python Input Handling.
3.	Scalable Architecture	Designed using 3-tier architecture (Presentation, Application, Model layer). Can be extended to cloud-based microservices	Flask (Backend), Modular ML Pipeline
4.	Availability	Application is available when deployed on local server; can be deployed on cloud for 24/7 availability	Flask Server, Future – AWS / Render
5.	Performance	Lightweight ML model ensures prediction within 2–3 seconds. Efficient preprocessing pipeline used	Scikit-learn, Optimized Model Loading

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