

Project design phase-II Technology Stack (Architecture and stack)

Date	10 th November 2023
Team ID	TEAM-592016
Project Name	Smart Home – Temperature Prediction
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 and Table 2.

Table-1: Components & Technologies:

S.No	component	Description	Technology
1.	User Interface	Interface for user interaction (e.g., mobile app, web UI)	HTML, CSS, JavaScript / React Js, etc.
2.	Data Collection	Gathering sensor data for temperature prediction	IoT Devices, Smart Wi-Fi Thermostats
3.	Data Preprocessing	Cleaning and preparing sensor data for modelling	Python, Pandas, NumPy
4.	Machine Learning Model	Predictive model for temperature	Linear Regression, Random Forest, XGBoost, etc.
5.	Model Training	Training the machine learning model	Python, Scikit-Learn, TensorFlow, PyTorch, etc.
6.	Cloud Database	Storing sensor data and model outputs	Cloud-based Database Service (e.g., AWS DynamoDB, IBM Cloudant)
7.	File Storage	Storing additional files or model artifacts	Cloud Storage (e.g., AWS S3, IBM Cloud Object Storage) or Local Filesystem
8.	User Interaction API	Handling requests from the user interface	Flask, Django (Python frameworks), RESTful API

9.	External API-1	External service for weather information	Weather API (e.g., IBM Weather API)
10.	External API-2	Additional external service for integration	External APIs for Home Automation (e.g., Smart Home API)
11.	Infrastructure (Server / Cloud)	Application Deployment	Local Server, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used in application development	Flask, React, TensorFlow, Scikit-Learn, Pandas, etc.
2.	Security Implementations	Security and Access Controls Implemented	Encryption (e.g., SHA-256), Role-Based Access Control (RBAC), Firewalls, HTTPS, etc.
3.	Scalable Architecture	Scalability considerations and architecture	Microservices Architecture, Cloud-based Scaling (e.g., AWS Auto Scaling)
4.	Availability	Strategies for ensuring high availability	Load Balancers, Redundant Servers, Automated Failover, etc.
5.	Performance	Design considerations for application performance	Caching Mechanisms, CDN Utilization, Asynchronous Processing, etc.