Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	13 May 2023
Team ID	NM2023TMID02708
Project Name	Project – IoT Based Weather Adaptive Street lighting System

Technical Architecture:

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

Guidelines:

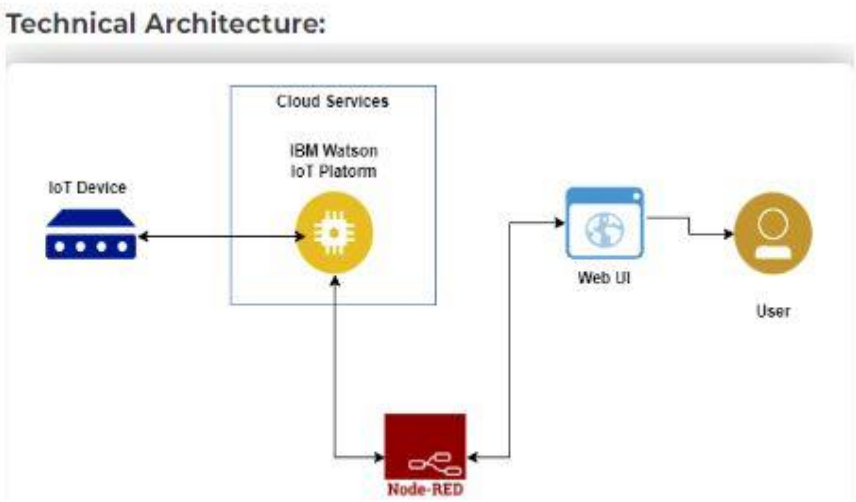

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Provides the interface for users to interact with the street lighting system through the mobile application.	Mobile application, Web application
2.	Application Logic-1	Manages the logic and functionality related to street light control and monitoring.	Programming languages (e.g., Python, Java, C++)
3.	Application Logic-2	Handles user authentication, registration, and login functionalities for the mobile application.	Programming languages (e.g., Python, Java, C++)
4.	Application Logic-3	Implements the logic for data capturing, processing, and sending data to the mobile application.	Programming languages (e.g., Python, Java, C++)
5.	Database	Stores and manages the data related to street lighting, user accounts, and system configurations.	Relational database management systems (e.g., MySQL)
6.	Cloud Database	Provides a cloud-based storage solution for scalable and reliable data storage and retrieval.	Cloud database services (e.g., Amazon RDS, Google Cloud SQL)
7.	File Storage	Stores and manages files such as images, documents, and logs associated with the application.	Cloud storage services (e.g., Amazon S3, Google Cloud Storage)
8.	External API-1	Integrates with external services or APIs for additional functionalities or data retrieval.	API integration frameworks (e.g., RESTful APIs)

9.	External API-2	Integrates with external services or APIs for weather data, location services, or other relevant information.	API integration frameworks (e.g., RESTful APIs)
10.	Machine Learning Model	Utilizes machine learning algorithms to analyze and predict street lighting requirements and optimize energy consumption.	Machine learning frameworks (e.g., TensorFlow, PyTorch)
11.	Infrastructure (Server / Cloud)	Provides the underlying infrastructure and hosting environment for the application:	Cloud service providers (e.g., Amazon Web Services, Google Cloud Platform).

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	The system utilizes open-source frameworks for its development.	Open-source frameworks (e.g., Django, Flask, Ruby on Rails, Node.js)
2.	Security Implementations	The system incorporates robust security measures to protect user data and prevent unauthorized access	Encryption protocols (e.g., SSL/TLS), secure authentication (e.g., OAuth, JWT), secure communication (e.g., HTTPS)
3.	Scalable Architecture	The system is designed with a scalable architecture to handle increasing users, devices, and data.	Scalable architecture (e.g., microservices, containerization), horizontal scaling, load balancing
4.	Availability	The system aims for high availability to ensure uninterrupted access to the application.	Redundancy, fault tolerance, disaster recovery planning, monitoring and alerting systems
5.	Performance	The system is optimized for performance to deliver fast response times and efficient data processing.	Performance optimization techniques, caching mechanisms (e.g., Redis), database optimization