INDEX

CHAPTER 1

INTRODUCTION

The Electric Vehicle (EV) Map and Charging Station Locator Platform is designed to enhance the convenience of EV users by providing real-time information about available charging stations, restroom availability, booking slots, and integrated payment solutions. As electric vehicles continue to gain popularity, the need for efficient and user-friendly platforms that facilitate seamless charging experiences has become crucial. This project aims to bridge the gap between EV users and available charging infrastructure by offering a centralized, intelligent system that optimizes user convenience and improves the overall EV charging ecosystem.

1.1 OVERVIEW OF THE PROJECT

In recent years, the adoption of electric vehicles (EVs) has accelerated rapidly as societies seek sustainable and eco-friendly alternatives to traditional combustion engines. However, the widespread transition to electric mobility brings with it significant challenges, particularly in the accessibility, availability, and convenience of charging infrastructure. EV users often struggle to find available charging stations in real time, face uncertainty regarding charging slot availability, and experience delays that disrupt their travel plans. Furthermore, existing navigation applications lack the specialized capabilities required for seamless EV charging experiences.

The **EV Map** project is designed to directly address these challenges by developing an intelligent, user-friendly web-based solution. This platform enables EV users to easily locate nearby charging stations, view real-time slot availability, book charging slots in advance, manage secure payments, and track vehicle delays based on traffic and arrival times. It also introduces an innovative module to check restroom availability and status at charging stations, enhancing user comfort during charging sessions.

By integrating services such as GPS tracking, real-time data analytics, and secure payment gateways, **EV Map** aims to create a smooth, efficient, and enjoyable charging experience for EV owners. The platform supports predictive scheduling, delay notifications, booking rescheduling, and seamless interaction between users and charging infrastructure providers.

The primary focus is to reduce the inconvenience associated with EV charging and contribute towards building a smart, connected electric mobility ecosystem.

Ultimately, **EV Map** seeks to bridge the gap between the growing EV user base and the existing infrastructure, offering a practical solution that combines real-time technology, user experience design, and operational efficiency. This project stands as an important step towards supporting the global shift to sustainable transportation by making EV charging more accessible, predictable, and user-centric.

1.2 LITERATURE SURVEY

1.2.1 Traditional Charging Station Locators

Early EV applications and web platforms focused mainly on providing static maps of charging stations, often lacking real-time availability or slot booking features. While they helped drivers find charging points, these platforms did not guarantee the availability of slots upon arrival, leading to user dissatisfaction and inefficiencies in station usage.

1.2.2 Real-Time Charging Slot Booking Systems

Recent advancements introduced dynamic systems where users can view live slot availability and reserve a charging point before reaching the location. Studies show that integrating real-time booking significantly reduces wait times and improves station utilization. Applications such as PlugShare and ChargePoint have pioneered real-time status tracking and booking functionalities, setting benchmarks for reliability and user satisfaction.

1.2.3 Restroom and Facility Availability Integration

Recent developments in EV service platforms recognize the importance of providing information beyond charging itself. Real-time restroom availability, cleanliness status, and related facility updates have become significant in enhancing user comfort, especially during long journeys. This feature ensures that users can plan their charging and rest breaks efficiently.

1.2.4 Vehicle Delay Tracking and ETA Management

Incorporating real-time vehicle tracking using GPS and traffic data enables platforms to calculate estimated arrival times (ETA) accurately. If a user encounters traffic delays, the system can dynamically update or reschedule their booking based on station availability. This level of adaptive service improves reliability and reduces missed booking slots, significantly improving user convenience.

1.2.5 Scalability, Mobile Optimization, and User Experience

Studies stress the necessity for EV platforms to be highly scalable, mobile-optimized, and intuitive. Responsive interfaces, real-time notifications, and efficient data handling ensure that users can access and manage their EV charging requirements easily from any device. As EV usage grows globally, scalability and mobile-first design have become non-negotiable standards

1.2.6 Security and Data Privacy

Modern EV platforms, dealing with user registrations and location data, must prioritize cybersecurity. Ensuring encrypted communications, secure authentication protocols, and compliance with global data privacy regulations like GDPR protects user trust and data integrity across the platform.

1.3 PROPOSED SYSTEM OBJECTIVES & SCOPES

1.3.1 Objectives of the Proposed System

The primary objective of the "EV Map Platform" project is to create a comprehensive, scalable, and efficient real-time service ecosystem that supports Electric Vehicle (EV) users in managing their charging needs with ease and precision. The system is designed to enable users to search for charging stations, book charging slots based on real-time availability, and track their vehicle's estimated arrival time dynamically, ensuring a seamless and reliable experience. It seeks to eliminate the common challenges faced by EV users, such as uncertainty regarding slot availability, delays, and lack of supporting amenities like restrooms. By integrating GPS

tracking, dynamic booking adjustment, and facility status updates, the platform enhances user convenience and trip planning capabilities. Furthermore, the system incorporates a robust admin management module to oversee station data, restroom facilities, and user bookings, ensuring that the platform remains up-to-date and responsive. The EV Map Platform aims to elevate the standard of EV travel by offering an integrated, user-friendly, and intelligent solution that not only addresses current infrastructure gaps but also lays a scalable foundation for future enhancements and service expansions.

1.3.2 Scope of the Proposed System

The scope of the proposed system incorporates four key modules: Admin Module, Charging Station Module, Vehicle Tracking Module, and User Module. The Admin Module oversees the overall functioning of the platform, managing user registrations, charging station data, restroom facility updates, booking records, and ensuring the platform's operational security and real-time responsiveness. The Charging Station Module enables station operators or administrators to update slot availability, charger types, restroom conditions, and other service-related information dynamically, providing users with accurate and timely updates. The Vehicle Tracking Module integrates GPS and real-time traffic data services to monitor the user's vehicle location, estimate arrival times, and dynamically adjust slot bookings in case of unforeseen delays, ensuring users retain their booking priority or are rescheduled appropriately. The User Module allows EV users to register, search for nearby charging stations, view real-time slot and facility availability, book a charging slot, monitor their booking status, track their vehicle's journey, and receive notifications about delays or changes, all through an intuitive, mobile-optimized interface. This comprehensive scope ensures a collaborative, transparent, and efficient environment for administrators and users alike, delivering a reliable, scalable, and user-friendly EV travel assistance platform that addresses the modern needs of electric vehicle mobility.

CHAPTER 2

REQUIREMENTS SPECIFICATION

The requirement specification for the EV Map Platform entails the development of a secure, scalable, and efficient web-based application using modern web technologies such as React.js for the frontend, Node.js and Express.js for the backend, and MongoDB for the database. The platform enables users to search for nearby charging stations, book available charging slots, track their vehicle's real-time arrival status, and access restroom availability updates during their journey.
Key functionalities include user registration and authentication, real-time charging slot booking, restroom facility monitoring, GPS-based vehicle tracking, booking rescheduling based on delays, and comprehensive admin management of station and facility information. The system leverages the power of a modern tech stack to ensure a responsive, mobile-optimized, and highly scalable platform capable of meeting the growing demands of the electric vehicle ecosystem.

2.1 OVERALL DESCRIPTION

The EV Map Platform is a web-based application built using modern web technologies including React.js, Node.js, Express.js, and MongoDB.

It is designed to provide a real-time, user-friendly environment where Electric Vehicle (EV) users can search for nearby charging stations, book charging slots in advance, and track their vehicle's journey towards the station.

The platform ensures secure user management, seamless booking experiences, dynamic vehicle tracking, and a mobile-responsive interface to support users on the move.

Charging station data, slot availability, and restroom status are managed and updated through the admin module, ensuring accurate real-time information for users throughout their journey.

2.1.1 Functional Requirements

In the EV Map Platform, the functional requirements are categorized according to four major roles: User, Admin, Charging Station Module, and Vehicle Tracking Module, each contributing distinct functionalities to ensure a seamless and efficient charging experience.

For Users, the system must allow individuals to register and log in securely. After authentication, users should be able to search for nearby charging stations using an integrated map interface. The platform must enable users to view real-time slot availability, book a preferred charging slot, and receive booking confirmations instantly. Users should also have the ability to monitor restroom availability at selected stations, track their vehicle's estimated arrival time through GPS integration, and receive real-time notifications about booking status changes or possible delays. It is essential for users to manage their personal profiles, view and manage booking history, reschedule bookings if necessary, and access the platform easily through a mobile-optimized interface designed for travel convenience.

The Charging Station Module focuses on managing dynamic updates related to station operations. Charging stations must regularly update real-time slot availability, specify charger types (such as fast or regular charging), and maintain information about restroom facility status, including availability and cleanliness. Charging stations must interact with the platform to ensure users are presented with the most up-to-date and accurate information, enabling efficient scheduling and resource management.

The Vehicle Tracking Module is responsible for integrating GPS and traffic data APIs to track the user's vehicle location and calculate estimated arrival times (ETA) to the booked charging station. If a user experiences delays due to traffic or unexpected issues, the module must dynamically adjust or reschedule the user's booking based on slot availability. Real-time notifications should be sent to inform users of any booking adjustments, ensuring a stress-free travel experience and minimizing wait times at charging stations.

The Admin holds comprehensive authority across the platform. Admins must manage all user accounts, including registration approvals, profile verifications, and complaint handling to ensure platform authenticity and security. Admins oversee the maintenance of station listings by updating or verifying information regarding slot availability, restroom facilities, and general station operations. In addition, admins must monitor booking activities, approve changes submitted by station operators, manage notifications sent to users, and handle escalations related to booking or station complaints. The Admin module should also generate analytical reports summarizing platform usage, station activity trends, and user behavior patterns,

supporting continuous improvement and strategic platform development. Admins are responsible for ensuring the overall integrity, security, and real-time responsiveness of the platform.

2.2 PRODUCT PRESPECTIVE

The EV Map Platform is conceptualized as a comprehensive web-based solution designed to address the growing needs of Electric Vehicle (EV) users for efficient, reliable, and real-time access to charging infrastructure. The platform acts as a centralized service where users can search for nearby charging stations, book available slots in advance, monitor restroom availability, and track their vehicle's estimated arrival time, ensuring a seamless and convenient travel experience. Unlike traditional static station locators, this platform introduces a dynamic, real-time interaction model where users can manage their journeys proactively, making informed decisions based on live data.

The platform follows a modular design built primarily using robust web technologies, ensuring seamless interaction between users, charging stations, and administrators. It integrates GPS-based vehicle tracking and real-time traffic data services to facilitate dynamic booking management and arrival time estimation. Each module—User, Admin, Charging Station, and Vehicle Tracking—operates with role-based access controls, ensuring that each user group interacts only with the features and data relevant to their role, enhancing both usability and security.

The product is intended to be responsive and mobile-friendly, enabling access across desktops, tablets, and smartphones, thereby ensuring a consistent user experience regardless of the device used. Scalability is an integral aspect of the system's design, allowing the platform to accommodate a growing number of users, charging stations, and booking activities without performance degradation. Furthermore, the platform is designed with future extensibility in mind, supporting potential integrations with third-party services such as route planning tools, loyalty programs, and advanced analytics platforms to enhance its operational capabilities. Overall, the EV Map Platform positions itself as a dynamic, secure, and versatile solution capable of transforming the EV charging experience through a user-driven, real-time service ecosystem.

2.3 PRODUCT FUNCTION

2.3.1 User Module

The User Module allows EV users to register, log in securely, and manage their profiles. Users can search for nearby charging stations using a real-time map, view available charging slots, check restroom facility status, and filter stations based on charger type or distance. Once a slot is selected, users can book it instantly and receive confirmation. The system tracks the user's vehicle through GPS integration, calculates estimated arrival time (ETA), and updates the booking if delays occur. Users can also manage active bookings, view booking history, and modify or cancel bookings when needed.
The platform is fully mobile-optimized, providing users with real-time notifications and a smooth, convenient experience for managing their EV charging needs.

2.3.2 Charging Station Module

The Charging Station Module enables charging station operators or administrators to update and manage station details in real time.

Operators can update the number of available slots, types of chargers (fast or regular), and the status of restroom facilities.

They are responsible for keeping slot availability accurate, ensuring users have up-to-date information when booking.

Through a secure login, station admins can also manage operational settings, post important updates, and ensure service quality is maintained.

This module ensures that the platform delivers accurate, reliable information to EV users at all times.

2.3.3 Admin Module

The Admin Module provides administrators with complete control over the platform's operations.

Admins can manage user accounts, monitor charging station updates, verify restroom facility status, and oversee all booking activities.

They are responsible for ensuring that station information remains accurate, user issues are addressed promptly, and platform security is maintained.

Admins can also generate reports on user activity, station usage trends, and booking patterns to support continuous improvements.

Through a centralized dashboard, the Admin Module ensures smooth, secure, and reliable management of the entire EV Map platform.

CHAPTER 3

SYSTEM DESIGN

System design refers to the description of a new system based on the information that is collected during the analysis phase and the process by which it is developed. It is the creative process of inventing and developing new inputs, database procedures, and outputs to meet the system's objectives. System design builds on the information gathered during system analysis. The system analyst must have a clear-cut understanding of the objectives, that the design aims to fulfill. System Design involves translating system requirements and conceptual design into technical specifications and the general flow of processing. After the system requirements have been identified, information has been gathered to verify the problem and after evaluating the existing system, a new system is proposed. System Design is the process of planning a new system or to replace or complement an existing system. It must thoroughly understand the old system and determine how computers can be used to make its operations more effective.

System design sits at the core of system development. Once system requirements have been analyzed and specified, system design is the first of the technical activities-design, code generation, and testing- that require building and verifying the software. System design is the most creative and challenging phase of the system life cycle. The term design describes the final system and the process by which it is to be developed. System design is the high-level strategy for solving the problem and building a solution. System design includes decisions about the organization of the system into subsystems, the allocation of subsystems to hardware and software components, and major conceptual and policy decisions that form the framework for detailed design

3.1 DATA FLOW DIAGRAM (DFD)

A data flow diagram (DFD) is a graphical representation of the flow of data within a system. It depicts how data is input, processed, stored, and outputted in a system or process. DFDs are widely used in software development, system analysis, and business process modeling to understand and communicate the data flow and interactions between different.

A level 0 DFD, also called a fundamental system model or a context model, represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively.

Additional process and information flow parts are represented in the next level i.e., Level 1 DFD. Each of the processes represented at Level 1 are sub-function of the overall system depicted in the context model. Any processes, that are complex in Level 1, will be further represented into subfunctions in the next level. i.e., in level 2. Data flow diagrams illustrate how data is processed by a system in terms of inputs, and outputs. Represent major components or functions with Circles. Actions for input by a user or a system go in Rectangular Boxes. Databases are represented by Parallel lines enclosing a phrase corner.

A data flow diagram consists of various components that collectively represent the flow of data and processes within a system. These components include processes, data stores, external entities, and data flows.

3.1.1 level 0 DFD

The Level 0 Data Flow Diagram (Context Diagram) for the EV Map Platform illustrates the primary interactions between the central system and its external entities: Users, Admins, and Charging Stations.

The EV Map platform acts as the core system, handling incoming requests and sending appropriate responses to all parties.

Users interact with the platform by submitting requests to search for charging stations, book charging slots, view restroom availability, and track vehicle status. In return, the platform provides real-time responses, including station availability, booking confirmations, and ETA updates.

Admins interact with the system by managing user registrations, station data, complaints, and operational settings. They send management requests to the platform and receive responses containing user details, booking records, or system alerts.

Charging Stations communicate with the platform to update real-time slot availability, charger status, and facility information. The platform processes these updates and shares updated responses with users in real time.

This Level 0 DFD shows the high-level flow of data between the EV Map system and its external entities, highlighting how the platform ensures seamless, real-time communication for an efficient EV charging experience.

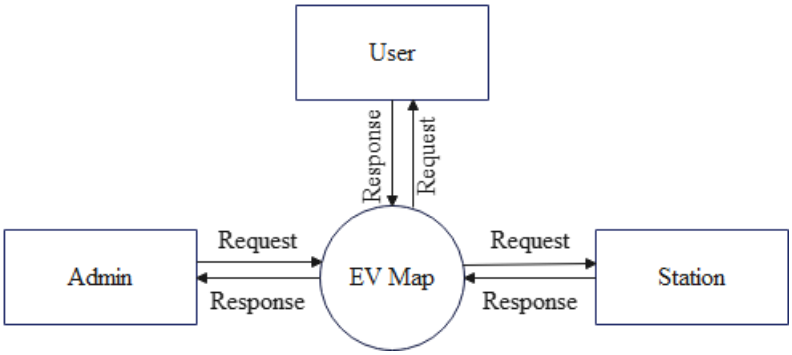

Figure 3.1.1 Level 0 DFD

3.1.2 Level 1 DFD

The Level 1 Data Flow Diagram for the EV Map Platform provides a detailed breakdown of the major processes within the system. Users interact with the platform by registering and managing their profiles through the User Management process. They can then search for nearby charging stations using the Search Station process, where real-time station data is retrieved. After selecting a station, users proceed to book a charging slot through the Booking Station process, securing a slot based on live availability. The Realtime Availability module continuously updates the system with current slot status, ensuring users see the latest information before booking. If users encounter issues or have concerns, they can submit their feedback or issues via the Complaint process, which is managed and reviewed by administrators. Although not a major separate module, the Payment process supports managing transaction data related to booking activities. Administrators interact with the platform primarily through the User Management process, overseeing user registration, complaints, and booking activities to ensure smooth operation. Data flows between users, stations, and administrators include user registration details, station availability data, booking records, complaints, and transaction information. This Level 1 DFD highlights how each external entity interacts with different system components to maintain a real-time, efficient, and user-friendly EV charging management experience.

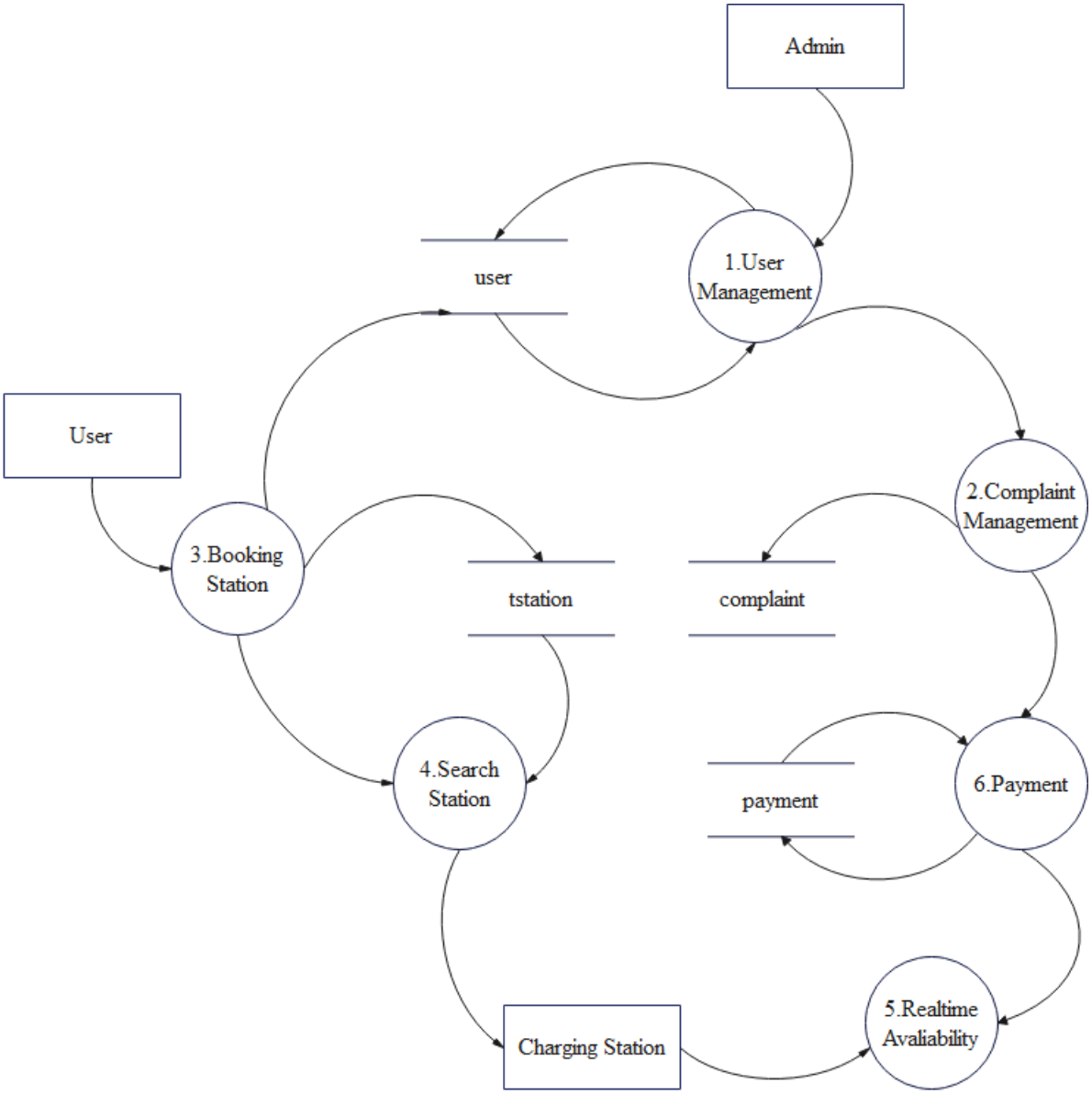

Figure 3.1.2 Level 1 DFD

3.1.3 Level 2 Station DFD

The Level 2 Data Flow Diagram for the Station Module in the EV Map Platform provides a detailed view of the specific activities carried out by station administrators. Stations begin their interaction by completing the Registration process, where they submit their station details to the platform for approval. Upon successful registration, stations use the Login process to access their administrative dashboard securely. Through the Add Slot process, station administrators

can add available charging slots into the system database. The Assign Slot process enables administrators to allocate specific slots to incoming booking requests from users based on real-time availability. Additionally, the Restroom Availability process allows stations to update the status of their restroom facilities, including cleanliness and accessibility information, ensuring users have accurate data when planning their charging stops. Data flows include station login credentials, slot information, restroom facility updates, and assignment of slots to users. This Level 2 DFD highlights the internal operational workflow of station administrators, ensuring that the platform maintains accurate, real-time service information for EV users.

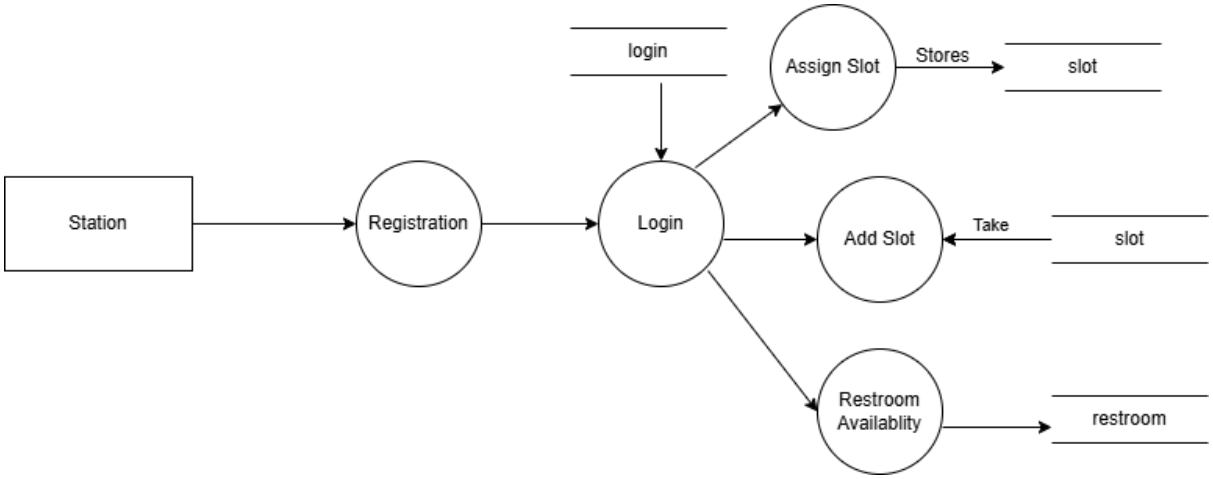

Figure 3.1.3 Station DFD

3.1.4 Level 2 Admin DFD

The Level 2 Data Flow Diagram for the Admin Module in the EV Map Platform provides a detailed view of the administrative operations performed to maintain system integrity. Administrators begin their interaction by securely logging into the platform through the Login process. After authentication, admins can access multiple functionalities, including the View Users process, where they retrieve user registration details to monitor and manage user accounts. Through the View Stations process, administrators can view and verify registered charging station details, ensuring that station information remains accurate and up to date. The View Slots process allows admins to monitor the slots available across different stations, facilitating better oversight of slot management activities. In addition, administrators can add new mechanics to the system via the Add Mechanic process, storing mechanic information into the platform database, and later manage or update these details through the View Mechanic

process. Data flows involve login credentials, user registration data, station records, slot availability details, and mechanic profiles. This Level 2 DFD highlights how the Admin Module oversees user management, station monitoring, service personnel management, and data consistency to ensure the smooth and reliable operation of the EV Map platform.

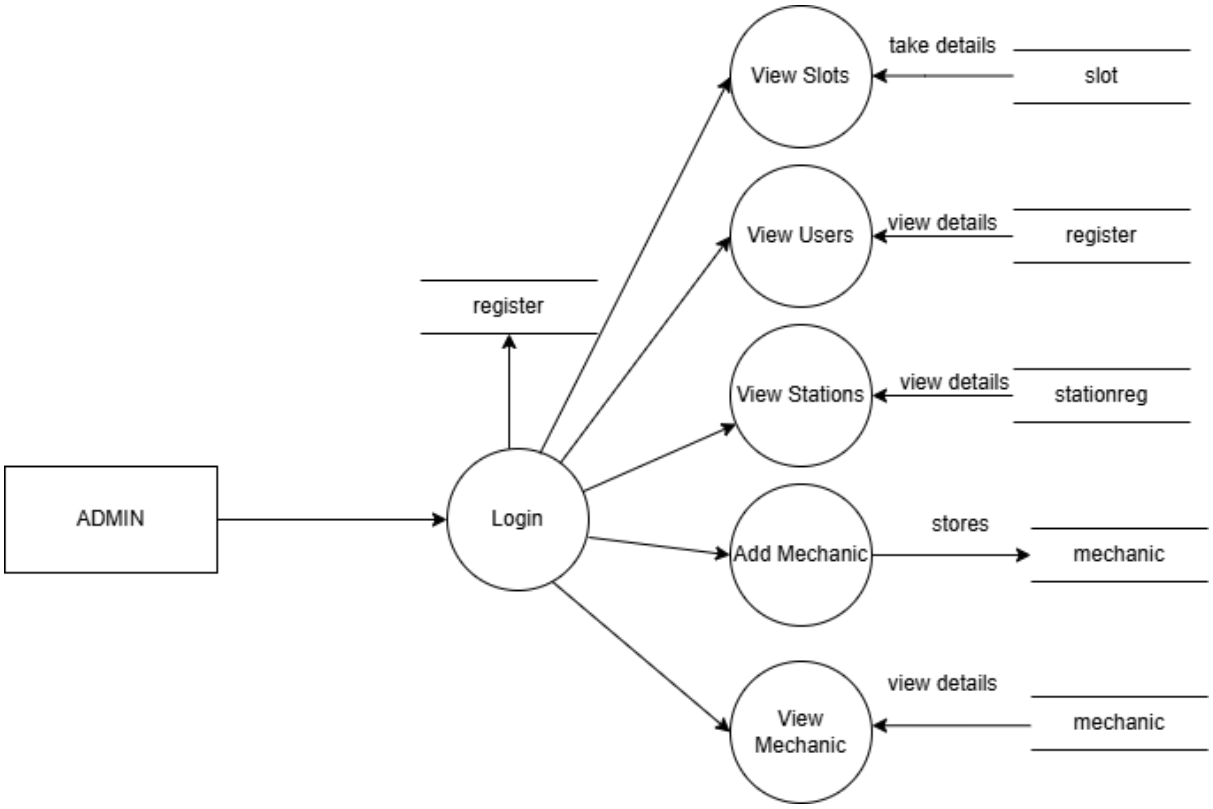

Figure 3.1.5 Admin DFD

3.1.4 Level 2 User DFD

The Level 2 Data Flow Diagram for the User Module in the EV Map Platform outlines the specific activities performed by users to interact with the system. Users begin by completing the Register process, where their personal details are stored securely in the system database. After registration, users can log in to the platform through the Login process by submitting their credentials, which the system verifies before granting access. Once logged in, users can perform several functions, including the View Station process, where they retrieve details of available charging stations from the database. Through the View Slot process, users can view the real-time availability of charging slots at selected stations. The Book Slot process enables users to reserve a charging slot based on current availability, storing the booking details into

the system. Additionally, the Make Payment process allows users to process payments related to their bookings, ensuring that transactions are securely recorded. Data flows include user registration details, login information, station records, slot availability, booking confirmations, and payment data. This Level 2 DFD highlights the step-by-step journey of users through the platform, ensuring a smooth, real-time experience in managing their EV charging activities.

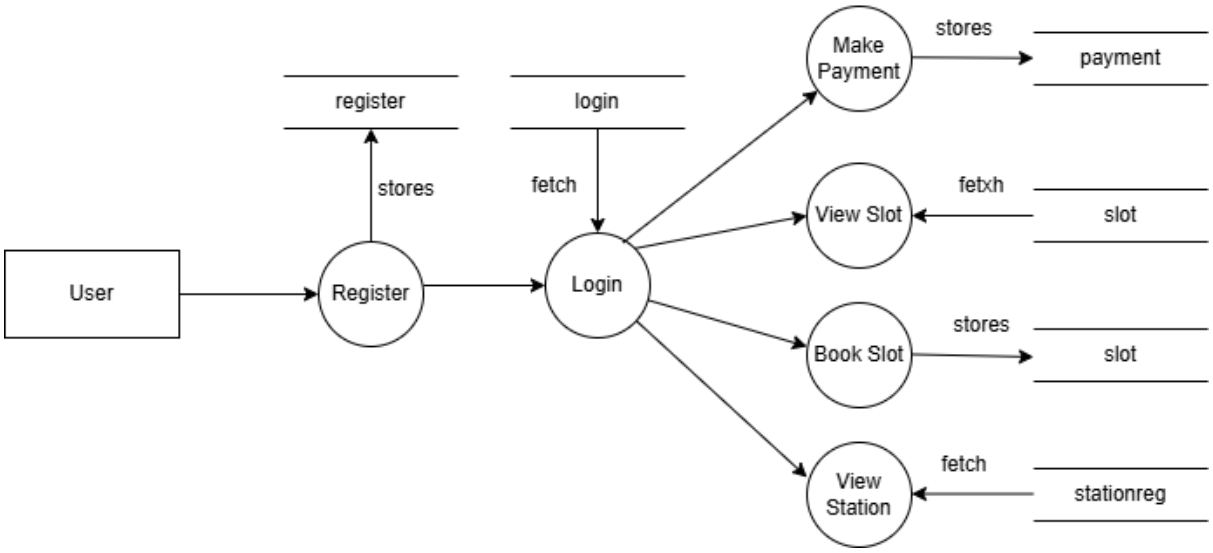

Figure 3.1.4 User DFD

3.2 USE CASE DIAGRAM

The use case diagram for the EV Map Platform illustrates the interactions between users, station administrators, and system administrators with the platform. Users can register, log in, manage their profiles, search for nearby charging stations, view real-time slot and restroom availability, book charging slots, track vehicle location, and manage their bookings. Station administrators interact with the platform by registering their stations, updating available charging slots, maintaining restroom facility status, and monitoring booking activities. System administrators are depicted with functionalities to manage user registrations, oversee station operations, verify slot and facility updates, handle user complaints, and generate platform. This diagram provides a clear overview of how different user roles interact with the EV Map Platform to deliver a seamless, real-time, and user-friendly EV charging management experience.

3.2.1 Admin Use Case Diagram

The Admin Use Case for the EV Map Platform defines the core functionalities available to system administrators. Administrators can manage users and charging stations by verifying registrations, updating station details, and overseeing slot information. They can generate platform reports through the Analytics and Reporting feature to monitor user activities and station performance. The Notification and Alert System enables sending real-time updates to users and stations about bookings and service changes. Additionally, through the Complaint and Dispatch Resolution process, administrators handle user complaints and ensure quick issue resolution. This use case ensures that administrators maintain the reliability, security, and smooth operation of the EV Map Platform.

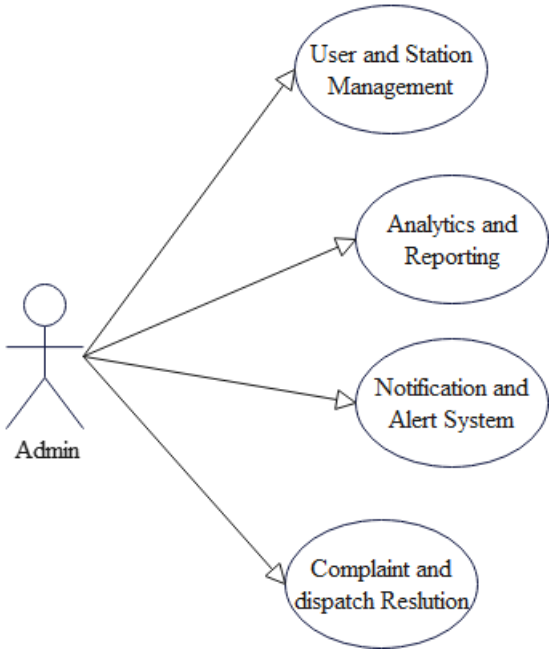

Figure 3.2.1 Admin Use case

3.2.2 Station Use Case Diagram

The Station Use Case for the EV Map Platform describes the main activities handled by charging station operators. Stations start with the Registration process to join the platform and create their operational profiles. Through Profile Management, stations can update information such as contact details, address, and facilities. The Real-Time Availability feature allows stations to update slot status and restroom conditions dynamically. Stations can also manage Pricing and Subscription plans for their offered services. Additionally, the Energy Monitoring

function enables stations to track and optimize their energy consumption. This use case ensures that stations provide up-to-date service data to users while maintaining efficient operations on the platform

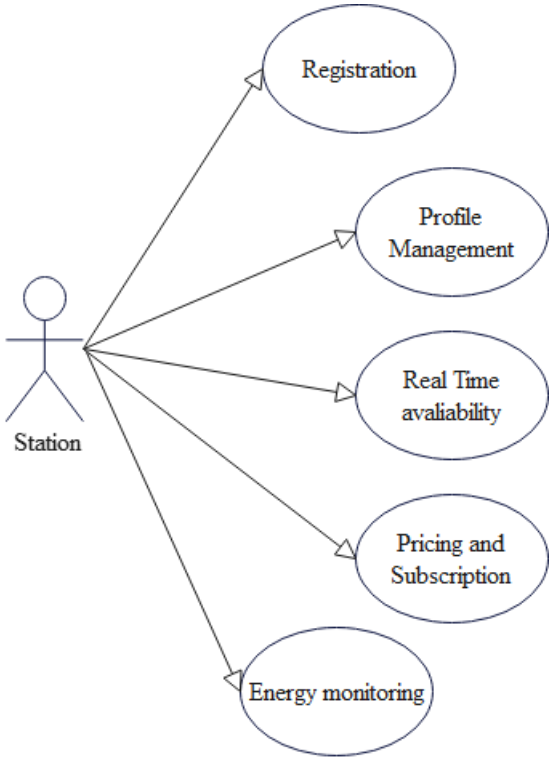

Figure 3.2.2 Station Use case

3.2.3 User Use Case Diagram

The User Use Case for the EV Map Platform outlines the key activities that EV users perform on the system. Users begin by completing the Registration process to create their accounts. Through the Search and Navigation functionality, users can locate nearby charging stations and navigate efficiently. The Booking and Payment process allows users to reserve charging slots and complete necessary payments securely. Route Optimization helps users find the best travel paths to reach their selected stations, considering real-time traffic and distance. Additionally, the Profile Management feature enables users to update personal details, view booking history, and manage their account settings. This use case ensures that users experience a seamless and efficient journey while interacting with the EV Map Platform.

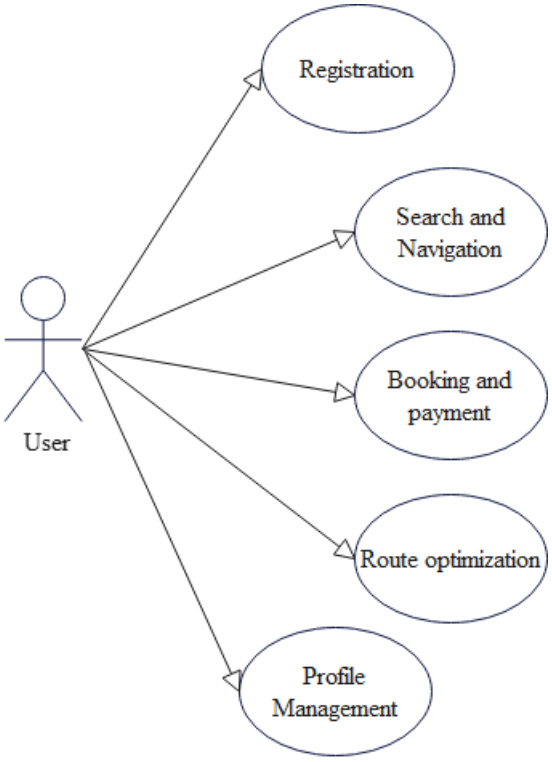

Figure 3.2.3 User Use Case

3.3 TABLE DESIGN

3.3.1 User Table

This table stores the user details

Field	Data Type	Description	Constraints
id	VARCHAR(24)	Unique identifier for each station (MongoDB ObjectID).	PRIMARY KEY, NOT NULL, UNIQUE
name	VARCHAR(100)	Name of the station owner or station entity.	NOT NULL, CHECK (LENGTH(name) > 0)
phone	BIGINT	Phone number associated with the station.	NOT NULL, UNIQUE, CHECK (phone > 0)
address	TEXT	Address of the station location.	NOT NULL, CHECK (LENGTH(address) > 0)

license	VARCHAR(255)	Filename of the station's license document or image.	NOT NULL, CHECK (LENGTH(license) > 0)
version	INT	Version control field (typically used for internal tracking).	NOT NULL, DEFAULT 1, CHECK (version >= 0)

3.3.2 Station Table

This table stores the station registration details.

Field	Data Type	Description	Constraints
id	VARCHAR(24)	Unique identifier for each user (MongoDB ObjectID).	PRIMARY KEY, NOT NULL, UNIQUE
name	VARCHAR(100)	Name of the user.	NOT NULL, CHECK (LENGTH(name) > 0)
longitude	DOUBLE	Longitude coordinate of the user's location.	NOT NULL, CHECK (longitude >= -180 AND longitude <= 180)
latitude	DOUBLE	Latitude coordinate of the user's location.	NOT NULL, CHECK (latitude >= -90 AND latitude <= 90)
phone	BIGINT	User's phone number.	NOT NULL, UNIQUE, CHECK (phone > 0)
version	INT	Version control field (typically used internally).	NOT NULL, DEFAULT 1, CHECK (version >= 0)

3.3.3 Slot Table

The table store the slot details.

Field	Data Type	Description	Constraints
id	VARCHAR(24)	Unique identifier for each slot record (MongoDB ObjectID).	PRIMARY KEY, NOT NULL, UNIQUE
booking_date	DATE	Date of the slot booking.	NOT NULL, CHECK (booking_date >= CURRENT_DATE)
start_time	TIME	Slot start time.	NOT NULL, CHECK (start_time >= '00:00:00' AND start_time < '24:00:00')
end_time	TIME	Slot end time.	NOT NULL, CHECK (end_time > start_time)
station_id	VARCHAR(24)	Reference to the associated station (Station ID).	NOT NULL, FOREIGN KEY (station_id) REFERENCES stations(id)
version	INT	Version control field (typically used internally).	NOT NULL, DEFAULT 1, CHECK (version >= 0)

3.3.4 Mechanic Table

This table stores the mechanic added by the Station

Field	Data Type	Description	Constraints
id	VARCHAR(24)	Unique identifier for each mechanic (MongoDB ObjectID).	PRIMARY KEY, NOT NULL, UNIQUE
name	VARCHAR(100)	Name of the mechanic.	NOT NULL, CHECK (LENGTH(name) > 0)
contact	BIGINT	Mechanic’s phone number.	NOT NULL, UNIQUE, CHECK (contact > 0)
experience	VARCHAR(50)	Mechanic's experience details.	NOT NULL, CHECK (LENGTH(experience) > 0)
address	TEXT	Address of the mechanic.	NOT NULL, CHECK (LENGTH(address) > 0)

time	TIME	Available time or preferred working time.	NOT NULL, CHECK (time >= '00:00:00' AND time < '24:00:00')
version	INT	Version control field (used internally).	NOT NULL, DEFAULT 1, CHECK (version >= 0)

3.3.4 Login Table

Stores login details

Field	Data Type	Description	Constraints
id	VARCHAR(24)	Unique identifier for each login record (MongoDB ObjectID).	PRIMARY KEY, NOT NULL, UNIQUE
email	VARCHAR(255)	Email address used for user login.	NOT NULL, UNIQUE, CHECK (email LIKE '%_@__%.__%')
password	VARCHAR(255)	Password for authentication (should be securely encrypted).	NOT NULL, CHECK (LENGTH(password) >= 8)
user_type	INT	Type of user (1 = Station, 0 = User, 2 = Admin).	NOT NULL, CHECK (user_type IN (0, 1, 2))
user_id	VARCHAR(24)	Reference ID linking to the User or Station.	NOT NULL, FOREIGN KEY (user_id) REFERENCES users(id)
reg_type	VARCHAR(50)	Type of registration (e.g., stationreg or userreg).	NOT NULL, CHECK (reg_type IN ('stationreg', 'userreg'))
version	INT	Internal versioning field for system tracking.	NOT NULL, DEFAULT 1, CHECK (version >= 0)

CHAPTER 4

APPENDIXES

Appendix- A Screenshots

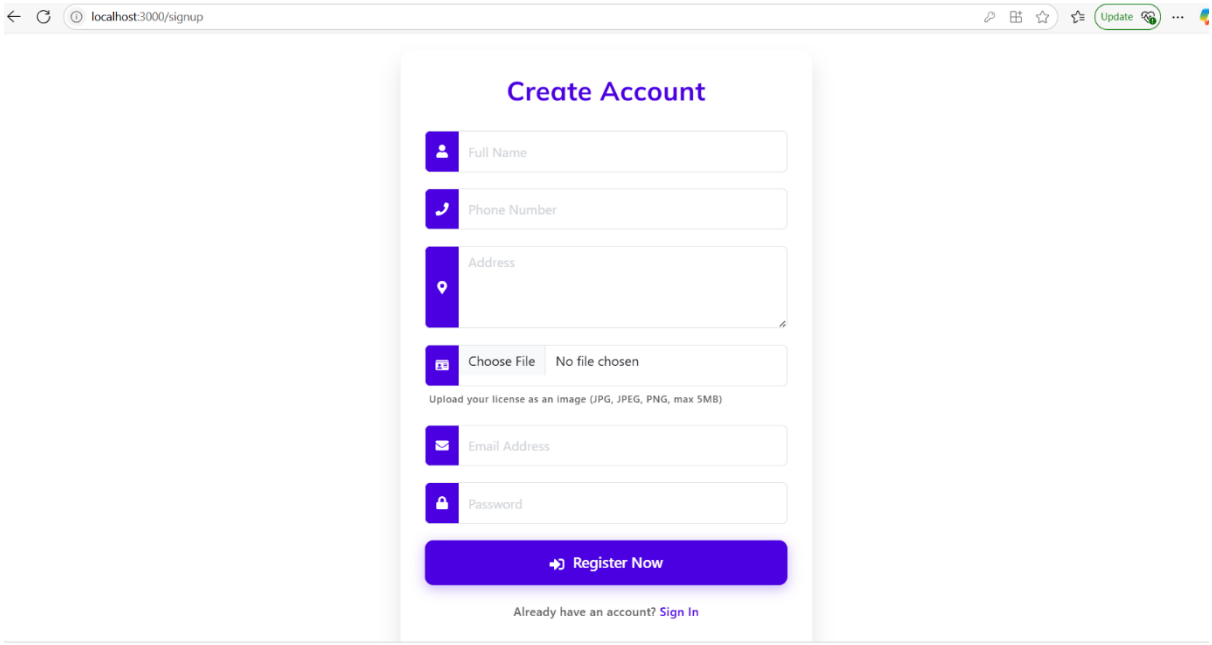

Figure A.1 User Registration

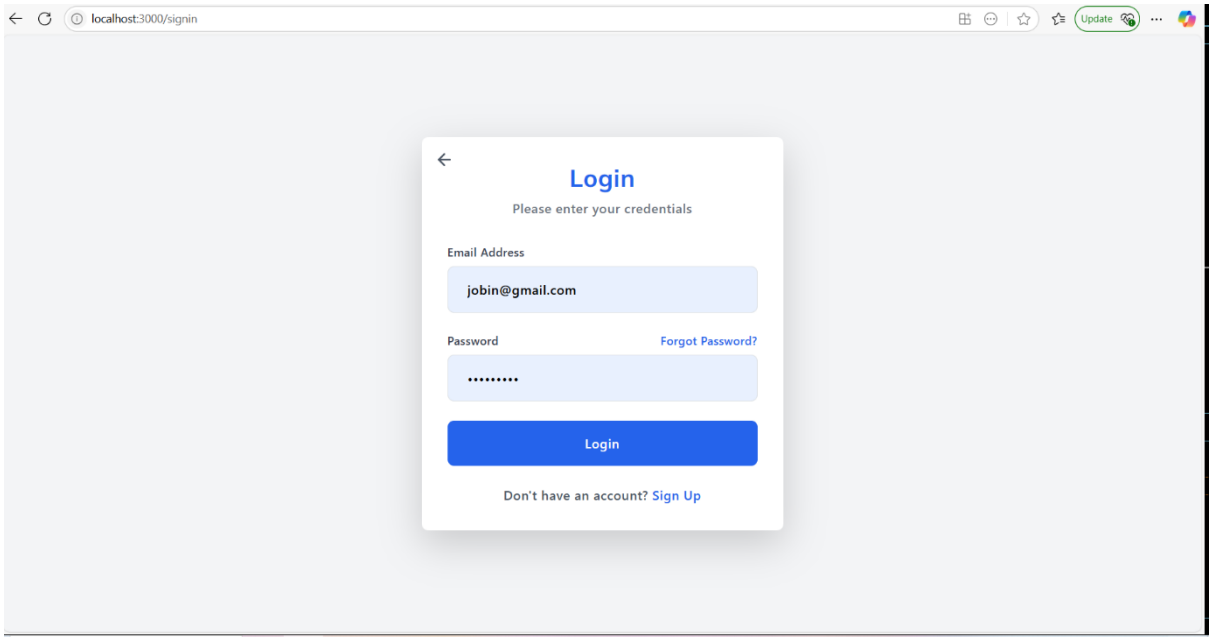

Figure A.2 Login

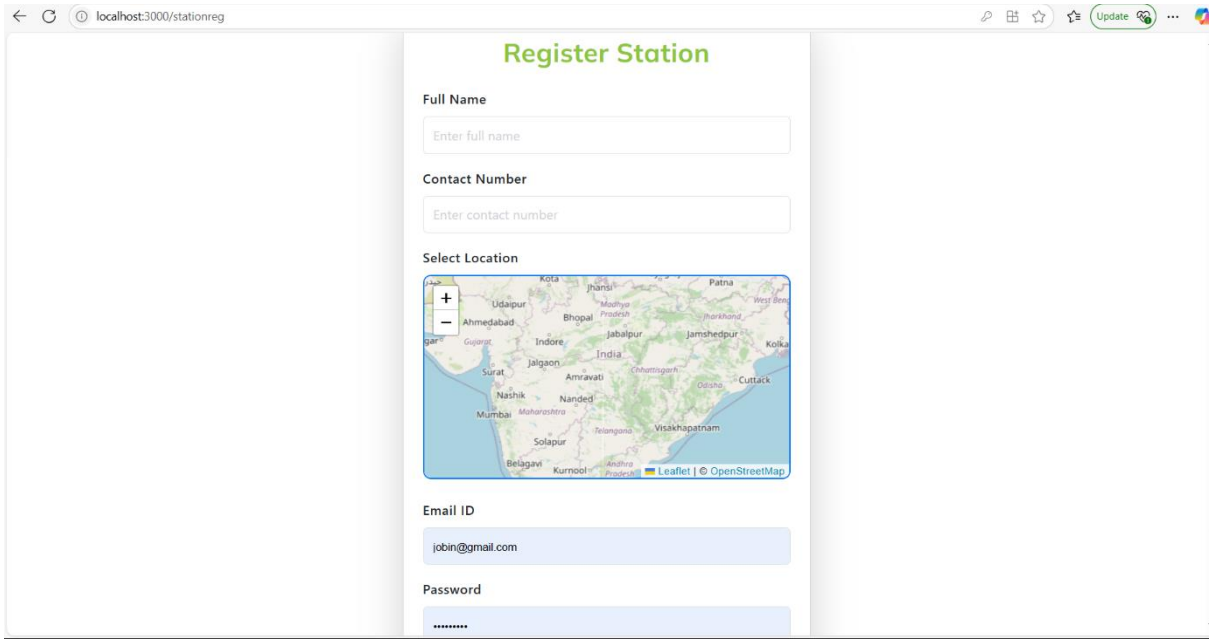

Figure A.3 Station Registration

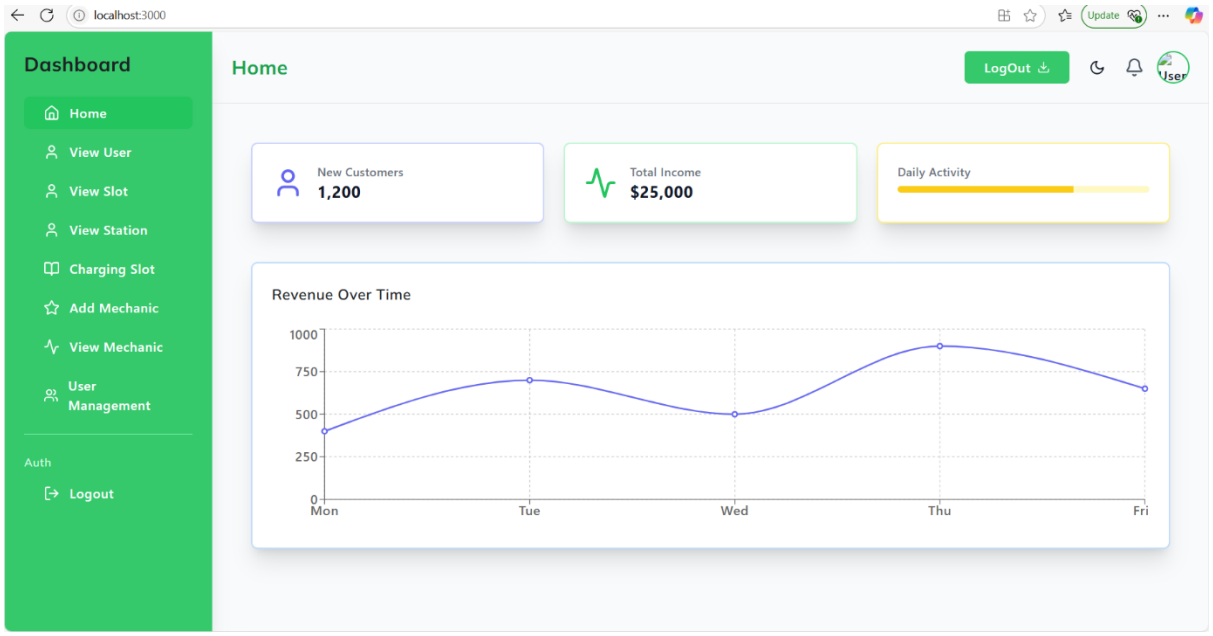

Figure A.4 Admin Dashboard

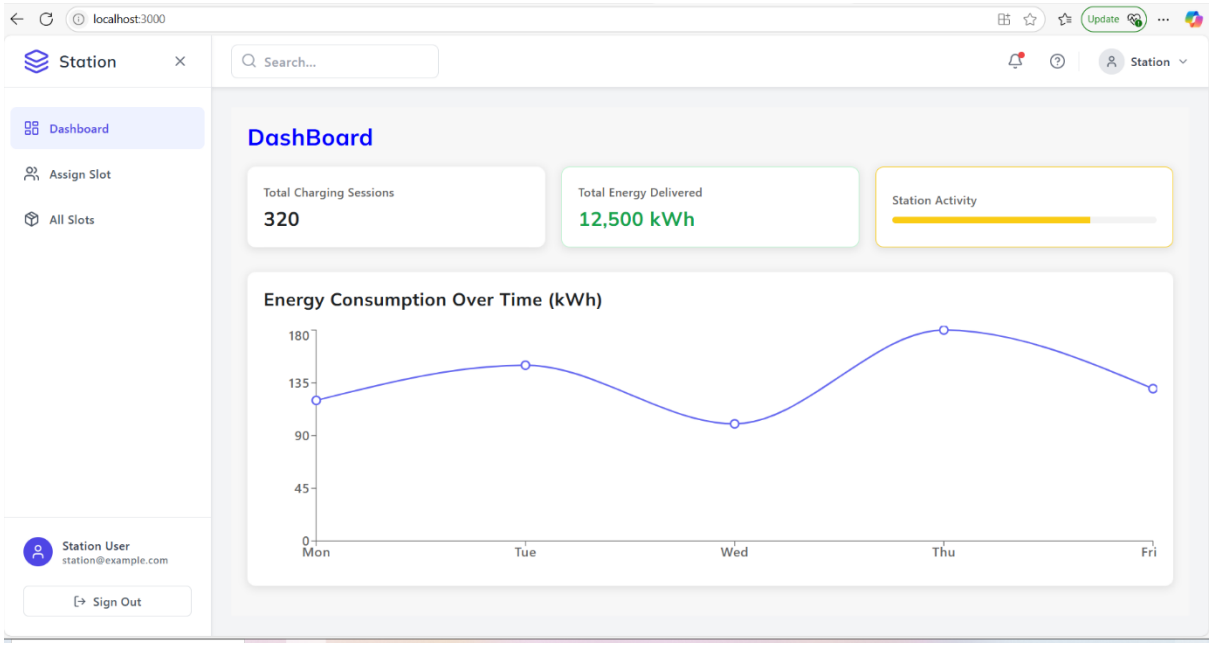

Figure A.5 Station Dashboard

Appendix- B Sample Code

Login.jsx

```
import React, { useState,useEffect } from 'react';
import { ArrowLeft } from 'lucide-react';

export default function LoginForm() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    console.log('Login attempt with:', { email, password });
    // Here you would typically handle authentication
  };

  const handleBack = () => {
    console.log('Back button clicked');
    // Handle navigation back
  };
  const [tailwindReady, setTailwindReady] = useState(false);

  useEffect(() => {
```

```
  // Check if Tailwind is already loaded
  const existingScript = document.querySelector('script[src="https://cdn.tailwindcss.com"]');
  if (!existingScript) {
    const script = document.createElement("script");
    script.src = "https://cdn.tailwindcss.com";
    script.onload = () => setTailwindReady(true);
    document.head.appendChild(script);
  } else {
    setTailwindReady(true);
  }

  // Optional: Remove script when component unmounts
  return () => {
    const script = document.querySelector('script[src="https://cdn.tailwindcss.com"]');
    if (script) {
      document.head.removeChild(script);
      setTailwindReady(false);
    }
  };
}, []);

if (!tailwindReady) {
  return <div>Loading form styles...</div>;
}
```

```
const handleLogin = () => {
  const data = {
    email: email,
    password: password,
  };

  fetch('http://localhost:4000/ev/login', {
    method: 'POST',
    headers: {
      'Accept': 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(data),
  })
    .then((res) => res.json())
    .then((result) => {
      if (result !== "invalid") {
        localStorage.setItem("yourstorage", JSON.stringify(result));
        window.location.href = "/";
      } else {
        console.log("Invalid credentials");
        // Optionally, set an error state here for user feedback in UI
      }
    })
```

```
    .catch((err) => {
      console.error("Error:", err);
      // You could also show this error to the user
    });
};

return (
  <div className="min-h-screen bg-gray-100 flex items-center justify-center p-4">
    <div className="bg-white rounded-lg shadow-lg p-8 w-full max-w-md relative">
      {/* Back Button */}
      <button
        onClick={handleBack}
        className="absolute top-4 left-4 text-gray-600 hover:text-gray-900 transition-colors"
        aria-label="Go back"
      >
        <ArrowLeft size={24} />
      </button>

      <div className="text-center mb-8">
        <h1 className="text-3xl font-bold text-blue-600">Login</h1>
        <p className="text-gray-500 mt-2">Please enter your credentials</p>
      </div>

      <form onSubmit={handleSubmit}>
```

```
<div className="mb-6">
  <label
    htmlFor="email"
    className="block text-sm font-medium text-gray-700 mb-2"
  >
    Email Address
  </label>
  <input
    type="email"
    id="email"
    value={email}
    onChange={(e) => setEmail(e.target.value)}
    placeholder="Enter your email"
    className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-transparent transition-all"
    required
  />
</div>

<div className="mb-6">
  <div className="flex justify-between items-center mb-2">
    <label
      htmlFor="password"
      className="block text-sm font-medium text-gray-700"
    >
```

```
            Password
          </label>
          <a href="#" className="text-sm text-blue-600 hover:text-blue-800">
            Forgot Password?
          </a>
        </div>
        <input
          type="password"
          id="password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          placeholder="Enter your password"
          className="w-full px-4 py-3 rounded-lg border border-gray-300 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-transparent transition-all"
          required
        />
      </div>

      <button
        type="submit"
        onClick={handleLogin}
        className="w-full bg-blue-600 hover:bg-blue-700 text-white font-medium py-3 px-4 rounded-lg transition-colors focus:outline-none focus:ring-2 focus:ring-blue-500 focus:ring-offset-2"
      >
```

```
            Login
          </button>

          <div className="text-center mt-6">
            <p className="text-gray-600">
              Don't have an account?{' '}
              <a href="#" className="text-blue-600 hover:text-blue-800 font-medium">
                Sign Up
              </a>
            </p>
          </div>
        </form>
      </div>
    </div>
  );
}
```

Server :Database.js

```
var mongoose=require('mongoose')

function database(){
    mongoose.connect("mongodb://localhost:27017/evmap")
    .then(()=>{
        console.log("successfull");})
```

```
    .catch(err=>{
        console.log(err);
    })
}
module.exports=database
```