INDEX

Title	Page No
Abstract	4
Acknowledgement	5
Introduction	6
System Analysis	7
Purpose of document	9
Project Members	10
Proposed systems	11
Objective and scope of project	13
System Requirement Specification	15
1.1 Overview	15
1.2 Study of the System	11

1. Functional Requirements 1.1 Hardware Requirements 1.2 Software Requirements 1.3 Supported Tools	16
2 Non-Functional Requirements 2.1 Performance Requirements 2.2 Safety Requirements 2.3 Security Requirements	19
3 Software Quality Attributes	20
4. System Design	21
Future Scope	24
References	26

LIST OF FIGURES

FIG. NO.	TITLE	PAGE NO.
1	E-R Diagram	17
3	Use Case Diagram 3.1 Admin Use Case Diagram 3.2 Faculty Use Case Diagram 3.3 Student Use Case Diagram	18, 19, 20
4	Sequence Diagram	21

ABSTRACT

The EV Charging Application is a comprehensive web-based platform designed to facilitate electric vehicle (EV) owners in locating, booking, and paying for charging services. Developed using modern technologies like React for the frontend and Node.js for the backend, this application integrates seamlessly with Razorpay to provide secure and efficient payment processing.

The application features a user-friendly interface that allows users to register, log in, and manage their accounts effortlessly. Users can search for nearby EV charging stations, view availability, and book charging slots based on their preferences. The integrated payment gateway, Razorpay, ensures smooth and secure transactions for users, enhancing the overall experience.

On the backend, the system leverages a MySQL database to manage user data, bookings, and transaction records, ensuring reliability and scalability. The Node.js server, coupled with Express.js, handles API requests and communicates efficiently with the frontend to deliver a seamless user experience.

This project addresses the growing demand for accessible and convenient EV charging infrastructure, providing a robust solution for both users and service providers in the electric vehicle ecosystem.

ACKNOWLEDGMENT

Success will be crowned to people who made it a reality but the people whose constant guidance and encouragement made it possible will be crowned first on the eve of success.

This acknowledgement transcends the reality of formality when we would like to express deep gratitude and respect to all those people behind the screen who guided, inspired and helped us for the completion of our project work.

We consider our self lucky enough to get such a good project. This project would add as an asset to our profile. We would like to express our thankfulness to all the Staff, Colleague who helped us to complete this project, for there constant motivation and valuable help through the project work. We also extend our thanks Infoway Technologies members for their co-operation during the course.

Finally we would like to thank our team members for their cooperation to complete this project.

Introduction

As electric vehicles (EVs) continue to gain popularity, the need for accessible and efficient charging infrastructure has become increasingly critical. The EV Charging Application addresses this need by providing a comprehensive platform that simplifies the process of locating, booking, and paying for EV charging services.

The application is designed with a focus on user experience, ensuring that EV owners can easily find charging stations and manage their charging needs with minimal effort. Leveraging cutting-edge technologies like React for the frontend and Node.js for the backend, the platform offers a responsive and intuitive interface that meets the demands of modern users.

One of the key features of the EV Charging Application is its integration with Razorpay, a secure payment gateway, which facilitates seamless transactions for users booking charging slots. This integration not only enhances the convenience of the application but also ensures that all payments are processed safely and efficiently.

In addition to its user-centric design, the application is built on a robust backend infrastructure powered by a MySQL database, ensuring that user data, bookings, and transaction records are managed reliably. The Node.js server, along with the Express.js framework, handles all API requests and provides smooth communication between the frontend and backend.

By offering a streamlined solution for EV charging, this project contributes to the broader effort to support the adoption of electric vehicles and promote sustainable transportation options.

System Analysis

1. Problem Definition:-

The rapid adoption of electric vehicles (EVs) has created a significant demand for reliable and accessible charging infrastructure. However, EV owners often face challenges such as locating nearby charging stations, checking availability, and managing payments. The lack of a unified platform to address these issues can lead to inconvenience and inefficiency, discouraging potential EV users and slowing down the transition to sustainable transportation.

2. Objectives:-

User-Friendly Interface: Develop an intuitive and responsive web application that allows users to easily search for EV charging stations, book slots, and manage payments.
Secure Payment Processing: Integrate a secure payment gateway (Razorpay) to ensure that all financial transactions are handled safely and efficiently.
Real-Time Availability: Provide real-time information on the availability of charging stations and slots to help users make informed decisions.
Scalability and Reliability: Build a robust backend system capable of handling a growing user base and managing data securely and efficiently.

3. System Requirements

Functional Requirements:

User Registration and Login: Users must be able to create accounts, log in securely, and manage their profiles.
Charging Station Search: The system should allow users to search for charging stations based on location, availability, and other criteria.
Booking System: Users should be able to book charging slots in advance and view their booking history.
Payment Gateway Integration: The system should securely process payments through Razorpay and provide users with transaction receipts.

Admin Panel: An administrative interface for managing charging stations, bookings, and user accounts.

Non-Functional Requirements:

Scalability: The system should be capable of scaling to accommodate a growing number of users and charging stations.
Performance: The application should load quickly and respond to user inputs with minimal delay.
Security: User data, including payment information, should be protected through encryption and other security measures.
Reliability: The system should be robust, with minimal downtime and reliable data storage.

Purpose of the Document

The purpose of this document is to provide a comprehensive guide for the development, deployment, and maintenance of the EV Charging Application. It serves as a critical resource for developers, stakeholders, and users by offering detailed information about the project's architecture, design, implementation, and usage. This document outlines the project’s scope and objectives, ensuring a shared understanding among all stakeholders.

It provides in-depth technical documentation, including system architecture, data flow, API endpoints, and integration points like the payment gateway, enabling developers to efficiently work on and extend the application. Additionally, the document offers step-by-step instructions for installation and setup, describes the key features and functionalities of the application, and provides guidelines for ongoing maintenance and troubleshooting. By serving as a central reference point, this document facilitates collaboration, promotes consistency, and ensures the sustainability and adaptability of the EV Charging Application over time.

Project Members:

PRN No.	Name
240343120011	Apurv Pramod Pannase
240343120072	Sarvesh Manohare
240343120090	Subhash Singh
240343120040	Mayank Jairaj

Proposed System

The proposed system is a web-based EV Charging Application designed to streamline the process of locating, booking, and paying for electric vehicle (EV) charging services. The system addresses the growing need for efficient and user-friendly charging solutions by offering an integrated platform that connects EV owners with charging stations. The application leverages modern technologies and secure payment processing to enhance user experience and ensure reliability.

Key Features of the Proposed System:

User Registration and Authentication:

The system allows users to create accounts and log in securely using their credentials. User profiles are managed through a secure authentication process, ensuring that personal and payment information is protected.

Charging Station Locator:

Users can search for nearby EV charging stations based on their current location or a specified address. The system provides a list of available stations, along with real-time information on availability and pricing, helping users make informed decisions.

Slot Booking and Reservation:

The system enables users to book charging slots at their preferred stations. Users can view available time slots, reserve a spot, and receive confirmation of their booking, ensuring that they have access to charging services when needed.

Payment Integration:

Secure payment processing is integrated into the system via Razorpay, allowing users to pay for charging services directly through the application. The payment process is seamless and includes options for various payment methods, providing flexibility and convenience.

Transaction and Booking History:

Users can view their past transactions and booking history within the application. This feature allows users to track their usage, manage expenses, and rebook frequently used stations with ease.

Admin Management:

The system includes an administrative interface where station owners or managers can add, update, and manage charging stations, view booking reports, and handle customer inquiries. This feature ensures that the system remains up-to-date and responsive to user needs.

Scalability and Performance:

The proposed system is designed to be scalable, capable of handling a growing number of users and charging stations as the adoption of electric vehicles increases. The backend architecture is optimized for performance, ensuring that the application remains responsive even under heavy usage.

Security and Compliance:

The system incorporates robust security measures to protect user data and ensure compliance with relevant regulations, including data encryption, secure authentication, and regular security audits.

Objective & Scope of Project

The primary objective of the EV Charging Application is to create a user-friendly and efficient platform that simplifies the process of finding, booking, and paying for electric vehicle (EV) charging services. The application aims to bridge the gap between EV owners and charging station providers by offering a seamless, integrated solution that addresses the growing demand for accessible EV charging infrastructure. Specifically, the objectives include enhancing user convenience by providing an intuitive interface that allows users to quickly locate nearby charging stations, check availability, and reserve charging slots with ease. Securing transactions is another key objective, achieved through the integration of a reliable payment gateway like Razorpay, which ensures that all financial transactions are processed securely, giving users confidence in the safety of their payments.

Supporting scalability is also a critical objective, with the system designed to accommodate an increasing number of users and charging stations as the adoption of electric vehicles grows, ensuring long-term viability and performance. The application also aims to improve operational efficiency by enabling charging station providers to manage their stations, bookings, and customer interactions efficiently through a dedicated admin interface. Finally, the project contributes to the broader goal of promoting electric vehicle adoption by making EV charging more accessible and convenient.

The scope of the EV Charging Application encompasses the entire lifecycle of locating, booking, and paying for EV charging services, with features tailored for both end-users (EV owners) and service providers (charging station operators). This includes user management for registration, login, and secure storage of user data; a charging station locator with search functionality and real-time availability display; a slot booking system with reservation management; and secure payment processing through Razorpay, supporting multiple payment methods and transaction history tracking. The scope also covers an admin interface for station management, a scalable and secure

backend built with Node.js and Express.js, a responsive frontend interface designed with React, and robust security measures to ensure compliance with data protection regulations. Future enhancements may include third-party integrations for route planning, EV maintenance, and additional features like loyalty programs, dynamic pricing, and user reviews.

SYSTEM REQUIREMENT SPECIFICATION

Overview

- User Registration and Authentication:

User signup/login functionality with email verification and password recovery

Search and Mapping:

Location-based search for nearby charging stations within a 5 km radius

Integration with Google Maps for displaying station locations and directions

Dynamic filtering by charger type (Level 1, Level 2)

Booking and Payment:

Charging slot booking system with real-time availability

Secure payment gateway integration with transaction history

Pricing tiers for different charger levels (e.g., 30 rupees for Level 1, 65 rupees for Level 2)

Notification System:

Email and SMS notifications for booking confirmations, payment receipts, and station updates

Push notifications for real-time updates and reminders

- Admin Dashboard:
    - Administrative interface for managing users, stations, payments, and reports
    - Analytics dashboard for tracking usage patterns, revenue, and performance metrics

2.2 Non-Functional Specifications

- Performance:

- The system should handle at least 1,000 concurrent users without performance degradation
- API response time should be under 500ms for standard queries

- Scalability:
  - The platform should be easily scalable to handle increasing user loads and expanding geographic coverage
  - Support for horizontal scaling of servers and databases

- Security:
  - Data encryption at rest and in transit using industry-standard protocols (e.g., AES-256, SSL/TLS)
  - Regular security audits and vulnerability assessments

    Role-based access control (RBAC) to restrict access to sensitive functionalities

- Usability::
  - Intuitive UI/UX design optimized for both desktop and mobile devices
  - Accessibility features complying with WCAG 2.1 standards

1. Functional Requirements

Operating Environment:

1.1 Hardware Platform:

- Processor: Above Pentium 4, with clock speed of 2.0 GHz
- RAM: 1GB or Above
- Hard Disk: Free disc space above 1GB

1.2 Software Platform:

- Front End: ReactJS / JSP, HTML, CSS, Bootstrap.
- Back End: MySQL, Spring and Spring Boot Framework, JPA.

1.3 Supported Tools:

- MySQL Workbench, Eclipse, STS.
- Web Server: Tomcat 9.0.

J2EE: Java 2 Enterprise Edition is a programming platform part of the Java Platform for Developing and running multitier architecture Java applications, based largely on modular software components running on an application server.

TOMCAT: It's an application server which is mostly used in the web-applications. It implements the Servlet2.5 & JSP2.1 specifications. It's a cross-platform application Server.

ECLIPSE: In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to

develop applications. By means of various plugins, Eclipse may also be used to develop applications in other programming languages: C, C++, and JavaScript. It can also be used to develop packages for the software Mathematical. Software Requirements Specification for "Virtual Learning Environment" 10 Development environments includes the Eclipse Java development tools (JDT) for Java.

SPRING BOOT: Java Spring Boot (Spring Boot) is a tool that makes developing web application and micro services with Spring Framework faster and easier through three core capabilities: Auto configuration. An opinionated approach to configuration. The ability to create standalone applications.

MySQL: MySQL is an open source 'Relational Database Management System' in which all the data are stored in the form of tables. Each table is connected to some other table i.e., has a Software Requirements Specification for "Virtual Learning Environment".

Relation with another table and this relationship is established through integrity constraints. These tables have columns which represents the attributes of an entity and there are rows of data for each column. This is called the database and is connected to the frontend or user interface with the help of controller. This is a fast and highly scalable database management System.

2. Non- Functional Requirements

2.1 Performance Requirements:-

The system should store all the database records of each instructor and student properly and the application should be available for use 24*7 through the server. Also, the application should be user friendly with a proper user interface which makes it easy for the user to understand. All the options should be present in properly accessible for user convenience.

2.2 Safety Requirements:-

All login ids and passwords of the admins, instructor and students should be protected for privacy using whatever constraints required in the database or the application. Admins, instructor, properties and student' records are to be backed up securely across database servers. Incase database is hacked by someone, and data is deleted a backup server should be present for such purpose.

2.3 Security Requirements:-

All passwords of the administrators should be protected for privacy using whatever constraints required in the database or the application. Transactions regarding properties should be carried out properly. Only admin will have access rights to the student and instructor data according to the need. The database should be protected from attacks and unauthorized access.

3. Software Quality Attributes

3.1 Availability

The system should run on a variety of operating systems that support the JavaScript language. The system should run on a variety of hardware.

3.2 Accessibility

The software will be accessible to admins, builders and users.

3.3 Compatibility

The software will be compatible with multiple platforms.

3.4 Durability:

The software will be tested for working with multiple users.

3.5 Effectiveness

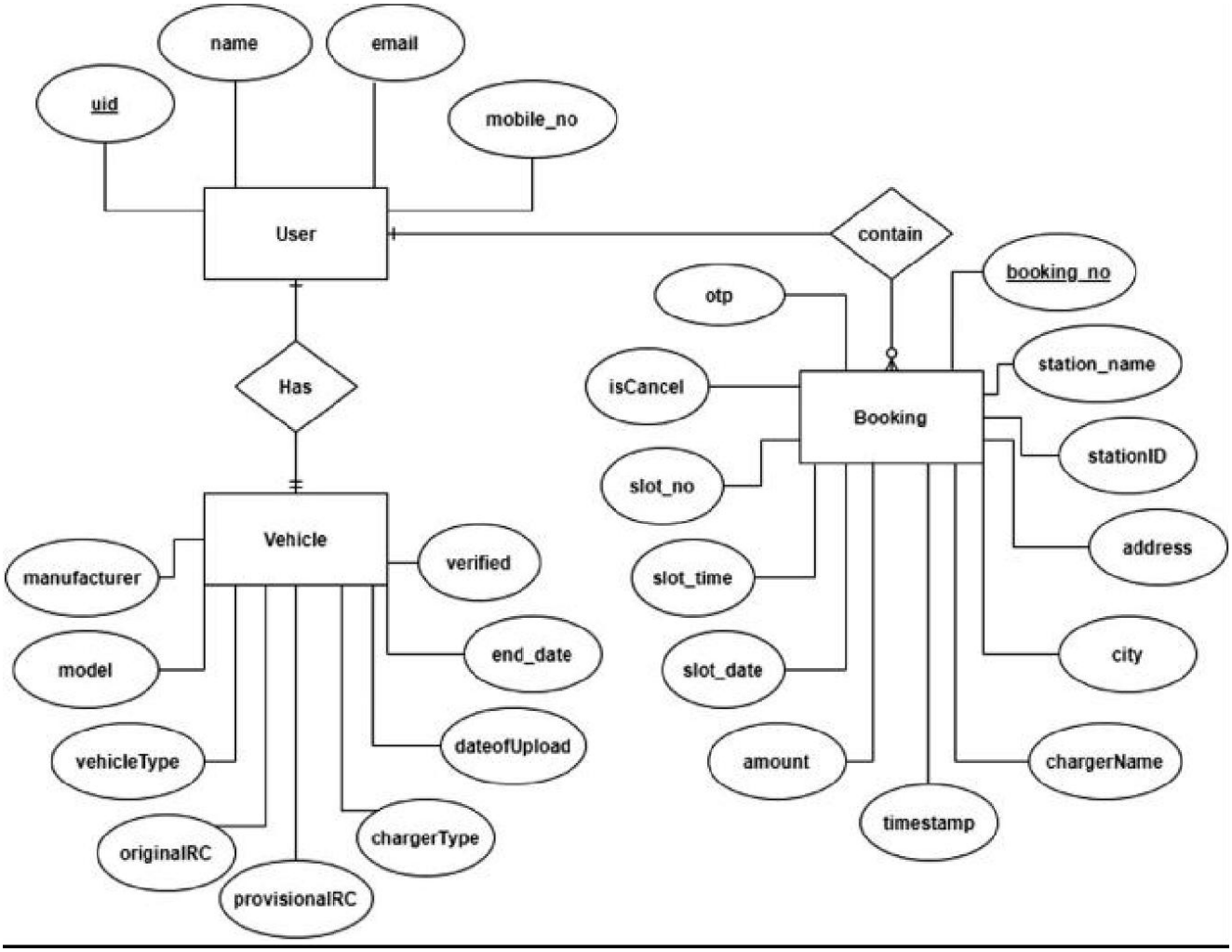
The software will be made to handle operations effectively.

3.6 Maintainability

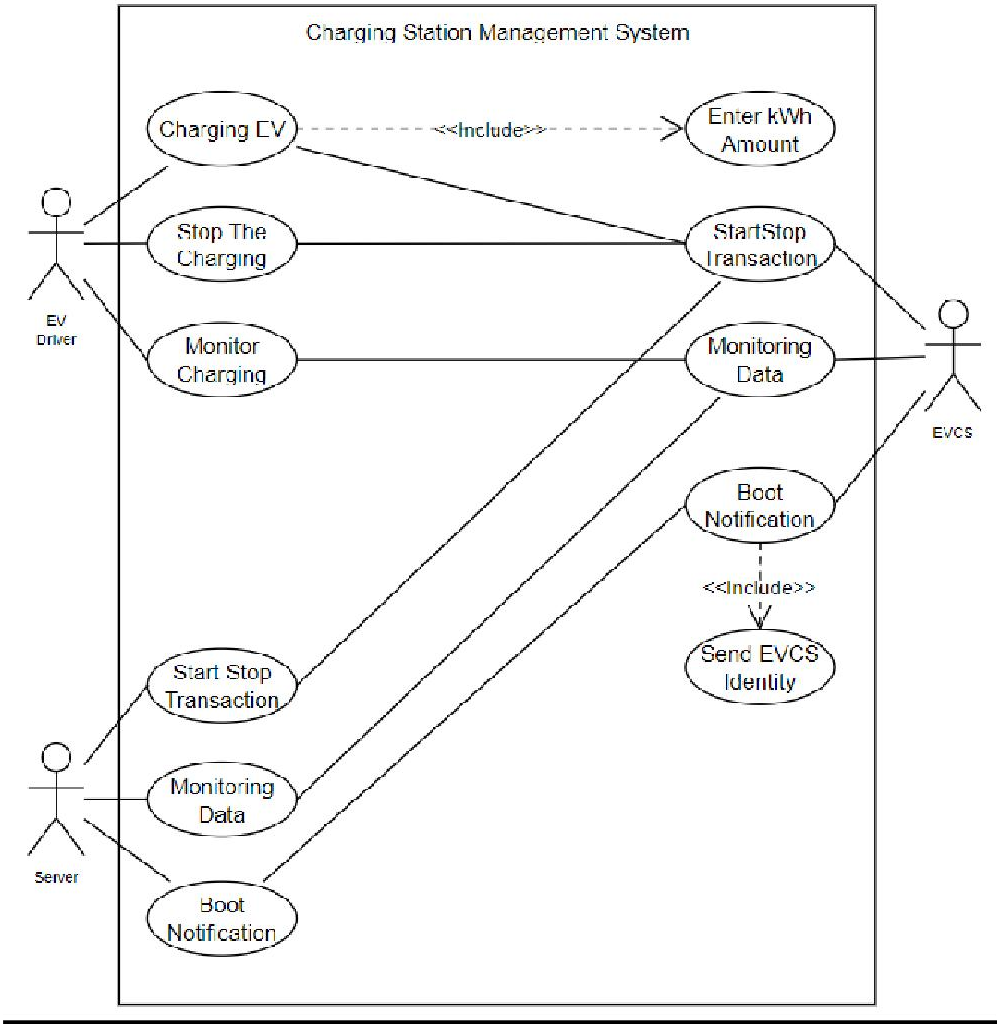
The system should be easy to maintain. There should be a clear separation between the interface and the business logic code. There should be a clear separation between the data access objects that map the database and the business logic code.

4. System Design

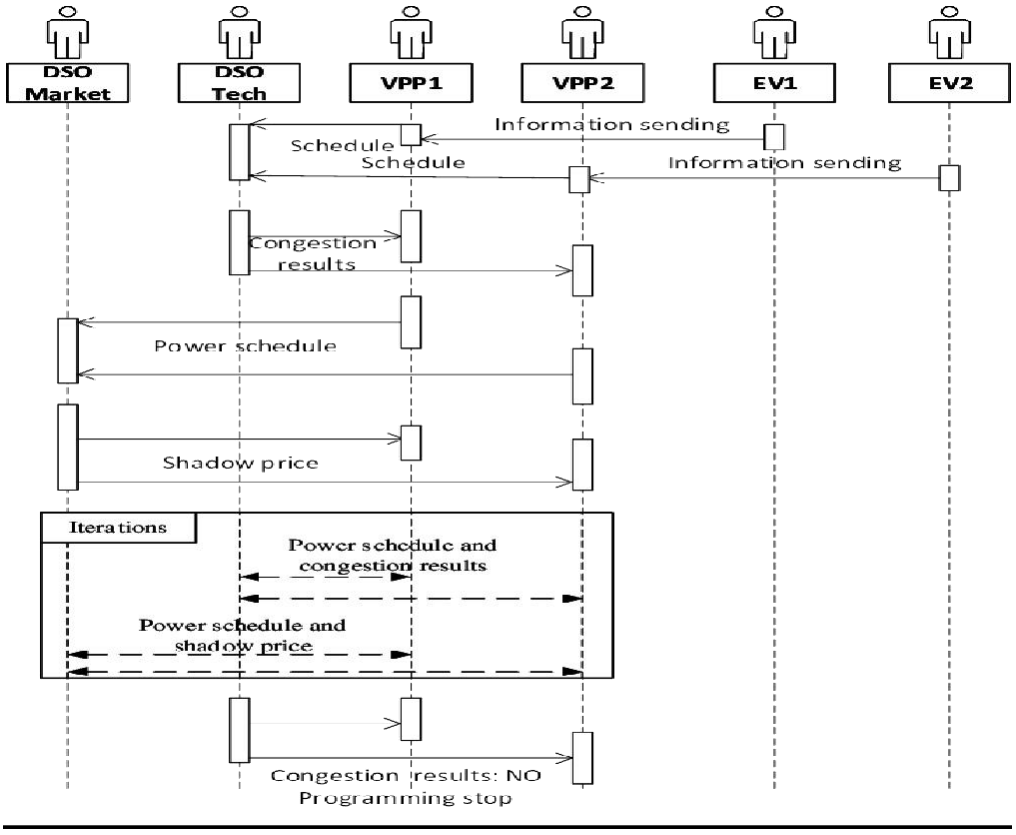
4.1 E-R Diagram



Use Case Diagrams



Sequence Diagrams



Future Scope:

The future scope of your EV charging station platform, EV Link Pune, is broad and promising, given the rapidly growing adoption of electric vehicles (EVs) worldwide. Here are several key areas where your project could expand and evolve:

1. Scaling and Geographic Expansion

- National Expansion: Expand the platform to cover more cities and regions within India, targeting areas with high EV adoption rates.
- International Expansion: If successful locally, consider expanding to other countries where EV adoption is growing and the charging infrastructure is still developing.

2. Integration with Renewable Energy Sources

- Solar-Powered Charging Stations: Integrate solar panels and other renewable energy sources to power charging stations, reducing the carbon footprint and operational costs.
- Battery Storage Solutions: Implement battery storage solutions to store excess renewable energy, ensuring a stable power supply for chargers even during peak times.

3. Advanced Payment Systems

- Subscription Models: Introduce subscription-based plans for frequent users, offering discounts or priority access to charging stations.
- Cryptocurrency Payments: Consider integrating cryptocurrency payment options to attract tech-savvy users and provide additional payment flexibility.

4. Enhanced User Experience

- AI and Predictive Analytics: Use AI to predict user behavior and suggest the best times and locations for charging based on historical data and real-time analytics.
- Smart Routing: Integrate smart routing that not only provides directions but also factors in real-time traffic, availability, and user preferences for the best charging station options.
- User Community Features: Add social features like user groups, forums, or reviews to build a community around your platform and encourage user engagement.

5. Partnerships and Collaboration

- Automotive Partnerships: Partner with EV manufacturers to integrate your platform directly into their vehicles' navigation systems.
- Utility Companies: Collaborate with utility companies to offer special tariffs for users charging during off-peak hours, reducing costs for users and balancing grid demand.
- Hospitality Industry: Partner with hotels, malls, and other businesses to offer exclusive charging deals to attract customers to their locations.

References

Libraries & Frameworks:

Spring Boot documentation, React documentation, RazorPay X documentation.

Tools:

Eclipse, Visual Studio Code, MySQL Workbench.

External Services:

Integration with payment gateway Razorpay, Google Maps.
APIs:

Google Maps API for GPS mapping and location service.