Ev Charging Station Management System

Prof. B. C. Tandale

Prasad Shrimandilkar

Siddhant Shelke

Bhavesh Gangarde

Lucky Tejwani

Department of Computer Engineering, SKN Sinhgad Institute of Technology and Science, Lonavala, India

Abstract: The shift towards Electric Vehicles (EVs) is driven by the urgent need to address environmental concerns and reduce our reliance on fossil fuels. Our project, a comprehensive Electric Vehicle Station Management System (EVSMS), aims to contribute to this transition. This survey explores the design and implementation of our EVSMS, highlighting its role in promoting the adoption of EVs over traditional fuel-powered vehicles. Our EVSMS offers features such as user registration, owner dashboards, and an admin panel, making it user-friendly for both EV owners and station operators. Through integration with the Google Maps API, users can easily locate nearby charging stations, but what makes our system stand out is its real-time station occupancy information. This feature not only aids users in finding available charging slots but also prompts them to consider alternative stations when their preferred choices are occupied. Furthermore, it keeps users informed through notifications about station availability and promotions, reinforcing the advantages of EV usage.. This survey provides insights into our system's design and its alignment with the global transition towards sustainable, eco-friendly mobility.

Keywords: Electric Vehicles (EV's) , Electric Vehicle Station, Management System (EVSMS)

1. Introduction

We are developing a system in which we going to connect all the electric car charging stations together. By using our system user can find the station according to their choice and it will be useful for those who want to travel for long distance with their EV cars and it will be time saving. It will be very easy to use. If the given time slot is available then your place for the given slot will be booked. Our system will also provide shortest map route to reach at given station. Our system will also provide interface for charging stations to view all available slots as well as booked slot lists and manage slot timing. We are going to develop this system for web based devices. To develop this system, we are going to use time-slot allocation techniques as well as Google maps API for direction sensing. EV charging stations that near-exclusively purchases power from PV systems on smart houses and sells power to electric vehicles (EV) and smart houses is proposed as an aggregator. The EV charging station has the need to utilize a fixed battery for electricity trading.

The development of a centralized web-based system for EV charging stations bulletproofs the entire electric vehicle (EV) recharging process, making it more user friendly and time efficient. Increased adoption of EVs requires infrastructure that can accommodate long-distance travel without substantial delays as a result of charging station capacity.It will be an integrated solution across multiple stations that offers the user a hardware and digital platform to find charging options based on certain likes (from proximity, price or even specific amenities). They will also have the option to see real-time availability of charging slots and pre-book time slots. This in shortens your waiting time, and gets you to where

are trying to go quicker reducing the stress on those longer distance travellers who need a little bit more foresight when planning there recharge stops.Perhaps a more important element of the system, however, is its use with Google Maps API -- not only does this allow users to receive exact directions to their destination station but also provides them with a recommended route that will take the least time and effort. Through real-time navigation on the platform, users will never take leave timely and escape traffic jam or detour to ensure a seamless travel.

2. Literature Survey

The literature on **EV Charging Station Management Systems** reveals diverse approaches to optimizing both user convenience and infrastructure efficiency. Vaidya and Mouftah's work on **smart charging management** demonstrates the potential for scheduling algorithms to minimize wait times and improve slot allocation, although challenges remain in scaling the system for larger networks. Esmaeilian and Samadi explore the **business models** and integration of charging stations into the grid, emphasizing the importance of sustainable infrastructure, but highlight the need for deeper analysis of grid load impacts. Meanwhile, Yu and Liu focus on **real-time slot allocation**, using dynamic scheduling to reduce bottlenecks, though the system's computational demands pose limitations for high- traffic areas. Youssef and Zayed tackle **demand-side management**, optimizing energy distribution during peak times, yet they acknowledge difficulties in handling unpredictable demand surges. Finally, Ho and Nicolaides present a **reservation-based system** that enhances user experience through advance booking, though the approach lacks flexibility for spontaneous users. Collectively, these studies point to key areas of improvement, such as scalability, user flexibility, and real-time grid management, while providing foundational methods for better EV charging station operations

3. Objective

The primary objectives of the **EV Charging Station Registration Management System** are to create a streamlined and user-friendly platform for electric vehicle (EV) owners. This system aims to provide a **slot booking feature**, allowing users to reserve charging slots at their convenience. It will offer a **simple and adaptive interface**, ensuring accessibility for all types of users, regardless of technical expertise. Additionally, the system will **connect multiple charging stations**, integrating them into a **single platform** for easier management and access. Users will be able to view **nearest available stations** and receive **real-time directions** via integrated maps to reach the station efficiently. The system will also make **booking and charging EVs easier** by simplifying the process for car owners, while incorporating a **feedback mechanism**, enabling users to rate and provide insights about their charging experience at various stations. These objectives aim to enhance the overall user experience and optimize charging station utilization.

4. System Requirement

4.1 Operating Environment

- **Target OS**: The system should support **Windows**, **Linux**, and **macOS** platforms for both users and administrators.
- **Mobile Compatibility**: A mobile-friendly version of the system will support **Android** and **iOS** devices to ensure accessibility on the go.
- **Browser Support**: The system should be accessible via modern web browsers such as **Chrome**, **Firefox**, **Safari**, and **Edge**.

4.2 Hardware Requirements

- **For Hosting (Charging Stations)**:
  - Minimum of **2 GHz dual-core CPU** for hosting.
  - **4 GB RAM** to ensure smooth operation of server-side processes.
  - **20 GB disk space** for storing system data, logs, and user information.
  - **Stable network connection** with at least **10 Mbps** for communication between stations and the system.
- **For Users**:
  - Basic device with **1 GB RAM**, **1 GHz CPU**, and **internet access** for connecting to the platform.

4.3 Software Requirements

- **Backend**:
  - **Java Development Kit (JDK) 11+** for building the server-side logic.
  - **Spring Boot Framework** for managing the server and backend processes.
  - **MySQL or PostgreSQL** for database management of station information, user accounts, and booking records.
- **Frontend**:
  - **ReactJS/Angular** for building a dynamic, responsive user interface.

- **Google Maps API** for integration of real-time mapping and navigation features.

- **Tools**:

- **IDE**: IntelliJ IDEA or Eclipse for development.

- **Build Automation**: Maven or Gradle for managing dependencies and builds.

- **Containerization**: Docker for packaging and deploying the system in different environments.

4.4 Network Requirements

- **Communication Protocols**: HTTP/HTTPS for secure communication between users, stations, and the backend.

- **Ports**: Ensure the appropriate ports (e.g., 80 for HTTP, 443 for HTTPS) are open for communication.

- **SSL/TLS Encryption**: To secure all data exchanges between users and the system.

4

5. Flow Diagram

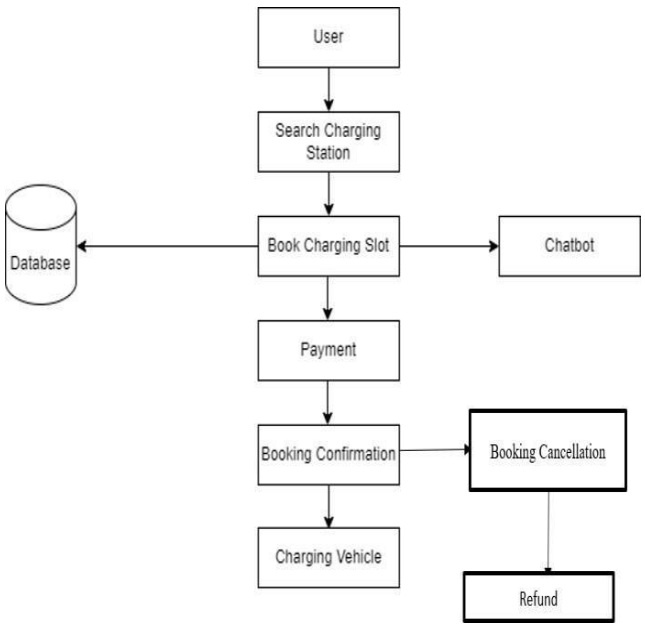

Fig.1.The Flow Diagram

6. Prerequisite

Creating a strong EV Charging Station Registration Management System needs some core technical and programming skills. Deep down is the knowledge of Java Programming Language, while using frameworks like Spring Boot to build backend logic. Java developers also require object-oriented programming (OOP) knowledge, multithreading and exception handling expertise including lists or maps(data structures), file handing etc.Laravel? For frontend part, basic knowledge of web development including HTML5, CSS3 and JavaScript is must be know. The web interface is structured by HTML and styled with CSS; JavaScript adds interactive elements, ensuring a fast responsiveness experience for users.

For building dynamic, responsive interfaces you will need to be familiar with both ReactJS or Angular as the firm uses MVC model in its architecture. They come with a component-based architecture, state management so that users can transition from one page of your application to the next seamlessly. Developers should be familiar with the Spring Boot Framework for backend plaguin. Others are well versed in dependency injection and inversion of control (IoC), making backend development easier. And RESTful APIs are required for frontend to backend communication, and data persistence with Spring Data in databases like MySQL or PostgreSQL creates a sturdy side of the window.

SQL (Structured Query Language) — to work with the databases, hospitals will want you to know how data is created in a database: details on users, types of charging stations and bookings or feedback. You must be expert in querying, designing tables for the databases (MySQL or PostgreSQL), relationships and transactions. Developers should also know Hibernate, an ORM tool that enables Java objects to be mapped onto database tables in a way which makes data management easier.

These include a basic understanding of how to working with location, for example knowledge in the Google Maps API (or any other maps service) so that you can show charging station locations as well as quickly draw on map directions between two points and availability status updates about someone's nearest station. It is very important in order to provide user friendly navigation path of nearest stations which Ola Play does. Also important is writing a REST API, which allows for the communication between the frontend and backend as well as mobile apps. Familiarity with HTTP methods (GET, POST, PUT & DELETE) and JSON for data interchange. It is also a key concern for APIs to be secured by OAuth 2.0 for authentication and secure data exchange.

For authentication and security protocols, developers must have a command on OAuth 2.0 or JWT (JSON Web Tokens) for securing user sessions and operations. Role Based Access Control (RBAC) is implemented then all roles(admin, station owner,user..etc.) are allowed to specify permissions along with Secure HTTPS for server-client communication.

Working with the real-time data everywhere in the system. Developers are required to have a thorough understanding of WebSocket and polling as weather-side if both need to be implemented so that the system keeps up-to-date checking for real-time charging station statuses. WebSockets could certainly be used for real-time communication, and they very well might be the solution here; conversely polling may also suffice if station availability updates relatively seldomly. Second, we need to have a quite good understanding of the system architecture on which your product operates or being developed i.e. microservices that make many charging stations scalable and modularly connected up into one single coherent system. For managing events, eg.: booking confirmations and slot updatesEvent-driven architecture pub-sub concepts out be helpful. Knowledge of design patterns (MVC, or Model-View-Controller and Observer amongst others) makes it easier to structure your system.

Besides the development, deploying and maintain of Rewarded Ads System is crucial. Without this chapter, an interesting containerized and orchestrated application is useless as it cannot be efficiently deployed or scaled due to lack of deployment in a docker containers definitions. CI/CD pipelines are fundamental to continuous integration and deployment, which aim for automation of testing as well updates.

7. Results/Output/Screenshots

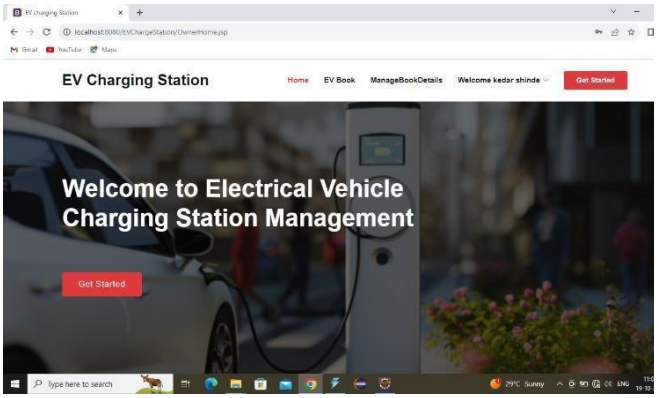

Fig.2. The front (Starting)page.

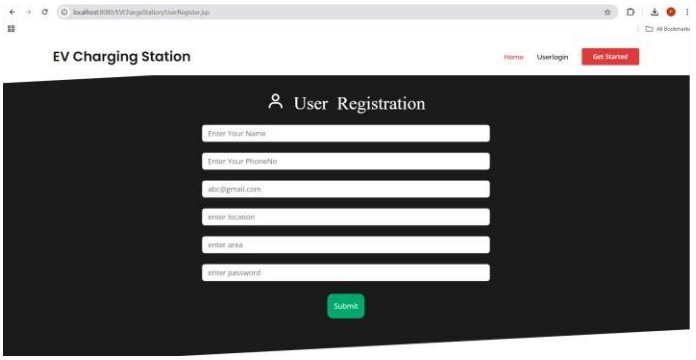

Fig3. New Registration Form.

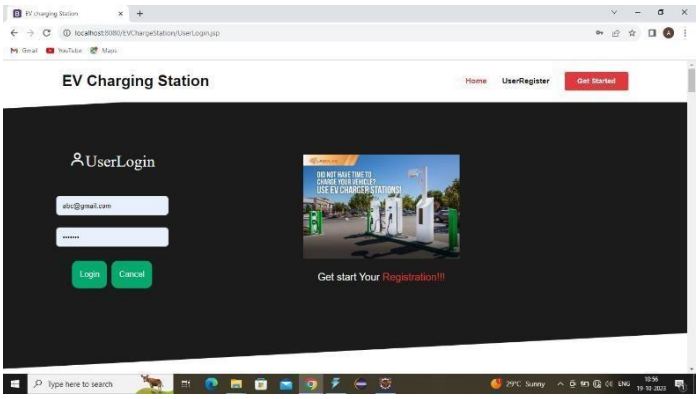

Fig.4. User Login Form.

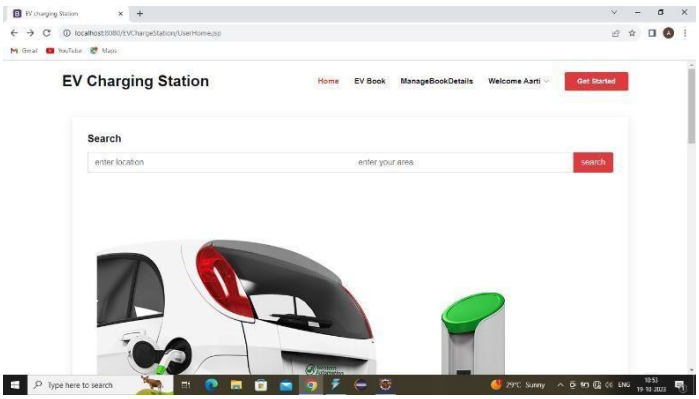

Fig.5. Main Product Page

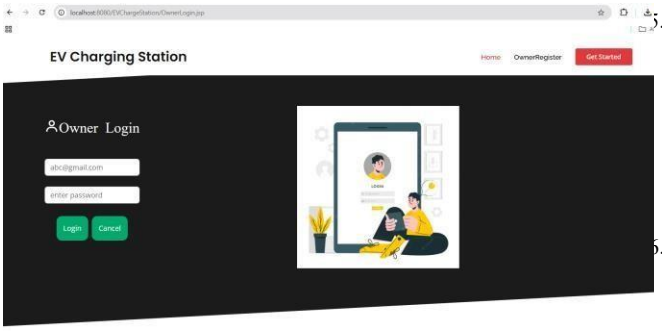

Fig.6.The EV Station Owner login Page

8. Conclusion

System is developed as a “Smart Management of EV Charging Stations” with hybrid approach of android application development. The system also proposes the booking of charging slot according to the type of charging socket to car. This system is also contains the AI chatbot for query solving as well as GMAPS API for direction sensing.In this study, we have introduced a web-based platform to address the challenge of long waiting times for Electric Vehicles (EVs) at charging stations through a pre- reservation system. Our approach was informed by an extensive literature review, a comparative analysis of the most widely used EV charging applications in the Indian market, and a detailed examination of customer feedback. This research unveiled several common shortcomings in existing EV charging platforms, such as intricate and perplexing user interfaces, a lack of real-time information on charging port availability, the inability to check the operational status of charging ports, limited payment options, and a deficiency of user reviews

9. References

1. Binod Vaidya, Hussein T.Mouftah, "Smart electric vehicle charging management for smart cities",School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa,Canada,2020.

2. Rohit Tamrakar, Niraj Wani, "Design and Development of
CHATBOT: A Review", Mechanical Engineering Department Sardar Vallabhabhi National Institute of Technology, Surat –INDIA, April 2021.

3. Achmad Fitro,Retno Kusumaningrum "Shortest Route at Dynamic Location with Node Combination-Dijkstra Algorithm",Magister Program of Information System,School of Postgraduate Studies, Diponegoro University,16-18 October 2019

4. Haysam Selim, Justin Zhan "Towards shortest path identification on large networks" Department of Computer Science, University of Nevada Las Vegas, Las Vegas, NV, USA, 2022

5. Minhang Zhou, Nina Gao "Research on Optimal Path based on Dijkstra Algorithms" Chengdu University of Information Technology Chengdu, China, 3rd International Conference on Mechatronics Engineering and Information
Technology,2021

6. MartinGjoreski, Hristijan Gjoreski, Rok Piltaver, Matjaž Gams, "Predicting the Arrival and the Occupancy of Electric Vehicle Charging Stations Using Machine Learning," Proceedings of the IEEE Intelligent
Transportation Systems Conference (ITSC), 2017,

7. DEPARTURE TIME OF AN EMPLOYEE", Department of Intelligent Systems, Jožef Stefan Institute,Faculty of Computer Science and Engineering - Skopje Jozef Stefan International Postgraduate School,2019.