

# Charge Locator: EV Charger Locator

Harshal Mitkari<sup>1</sup>, Nikhil Mogare<sup>2</sup>, Shivp Arasad Phadatare<sup>3</sup>, Akshay Hedau<sup>4</sup>, H. Thengil<sup>5</sup>

<sup>1,2,3,4</sup> *B.E, Information Technology, Sinhgad College of Engineering, Pune, India*

<sup>5</sup> *Guide – Prof B.E, Information Technology, Sinhgad College of Engineering, Pune, India*

**Abstract**—The rise in electric vehicle (EV) ownership has created a demand for accessible and efficient charging solutions. Charge Locator is a mobile application designed to simplify the EV charging experience by offering a platform for users to locate and book nearby charging stations. Built with Flutter and a Spring Boot backend, the app features real-time data updates on station availability, slot booking, user ratings, and navigation assistance, all integrated with GPS and third-party APIs for seamless location tracking. The app's responsive interface ensures a smooth user experience, allowing EV owners to easily find and reserve charging slots. Charge Locator's database is regularly updated to reflect new station availability, ensuring users have the latest information. By offering a streamlined, user-friendly platform for managing EV charging needs, Charge Locator supports broader EV adoption and contributes to a sustainable urban transportation ecosystem.

**Index Terms** - Charging Stations, Slot Booking, Route Planning, Charger Type, Flutter.

## I. INTRODUCTION

The shift towards sustainable transportation has led to a substantial increase in electric vehicle (EV) ownership worldwide. This growth has emphasized the critical need for accessible and efficient EV charging infrastructure to support the expanding EV user base. While EV technology advances, many drivers still face challenges such as locating available charging stations and securing charging slots in high-demand areas. These limitations can hinder the user experience, creating unnecessary delays and inefficiencies for EV owners.

Charge Locator was developed to address these challenges by providing an intuitive mobile application that enables EV drivers to locate, book, and navigate to charging stations in real-time. Built with Flutter for cross-platform compatibility and utilizing a robust Spring Boot backend, the Charge Locator app offers features such as real-time station availability, slot booking, and user-generated reviews. With GPS integration and support for third-party APIs, the app provides seamless location tracking,

enabling users to identify nearby charging stations efficiently.

The Charge Locator platform follows an Agile development process, which enables continuous improvements based on user feedback and evolving market demands. By streamlining the charging experience, Charge Locator not only enhances convenience for EV drivers but also contributes to the growth and accessibility of EV infrastructure, thus supporting sustainable urban transportation. This report outlines the system architecture, key functionalities, implementation, testing, and future enhancements, highlighting the platform's impact on the EV ecosystem.

## II. LITERATURE REVIEW

The development of EV charging station locator applications has gained momentum in recent years, driven by the rapid growth of electric vehicle adoption and the corresponding need for accessible charging infrastructure. Existing solutions such as Plug Share and Charge Point have set a benchmark for EV charging platforms, with each offering unique features and architecture tailored to enhance user experience and optimize accessibility.

Plug Share utilizes a client-server model with a cloud-based backend hosted on Amazon Web Services (AWS), which provides scalability and real-time data processing capabilities for a growing user base. Its frontend, built with React Native, ensures cross-platform functionality, enabling users on both iOS and Android to search and book nearby charging stations seamlessly. Plug Share's backend relies on a RESTful API to manage requests and employs a PostgreSQL database to store detailed charging station data, user profiles, and user reviews. Integration with Google Maps API allows for real-time navigation, while user authentication mechanisms ensure data security. Plug Share has demonstrated how robust backend architecture, combined with a user-friendly interface, can significantly improve the EV charging experience. Similarly, ChargePoint employs a client-server architecture backed by a cloud-based system on AWS,

with a mobile application that features native frontends for iOS and Android. ChargePoint leverages a RESTful API to handle user requests and uses a NoSQL database, such as MongoDB, to store information on charging stations, user accounts, and booking details. Like Plug Share, Charge Point integrates with the Google Maps API to provide real-time location tracking and navigation, while secure user authentication ensures data integrity. In addition to offering ratings and reviews, ChargePoint incorporates advanced features such as user feedback, creating a community-driven approach to charging station management.

Collectively, these applications underscore the importance of real-time data, cloud-based scalability, and user interaction in developing effective EV charging solutions. While these existing platforms have successfully addressed the need for EV station location and booking, Charge Locator distinguishes itself by focusing on a lightweight, cost-effective solution with seamless real-time features, a simplified booking process, and robust data synchronization via Firestore. By adapting proven architectural frameworks and integrating user feedback mechanisms, Charge Locator seeks to bridge gaps in accessibility and functionality, offering a streamlined and scalable solution tailored to the needs of the modern EV driver.

### III. METHODOLOGY

#### A. System Architecture

The Charge Locator system is designed with a modular architecture, dividing key functionalities across various backend services, a central API Gateway, and multiple databases. This architecture supports the core functionalities of locating, booking, and managing electric vehicle (EV) charging stations, while providing a seamless and user-friendly experience through a mobile app developed in Flutter.

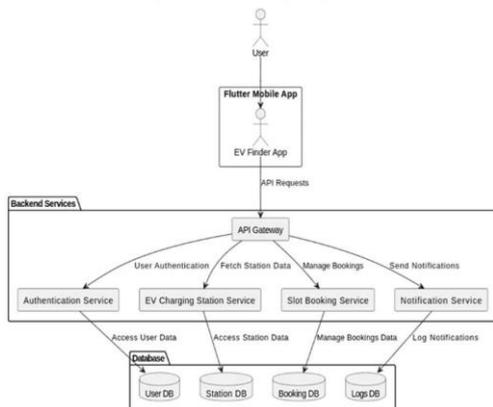


Fig 1. System Architecture of Charge Locator

#### 3.1 Client-Side (Flutter Mobile App)

The client-side of Charge Locator is a mobile application built using Flutter, allowing it to run on both iOS and Android platforms. The app enables users to:

1. Register and authenticate their accounts.
2. Search for nearby EV charging stations based on their location.
3. View station details, including availability, charger type, and ratings.
4. Book available slots at selected charging stations.
5. Receive notifications about booking confirmations, reminders, and updates.

The app sends requests to the backend services through the API Gateway, which routes requests to the appropriate service. The user interface is interactive and optimized for real-time responses, ensuring users receive the latest information on charging station availability and bookings.

#### 3.2 API Gateway

The API Gateway serves as the central communication point between the Flutter mobile app and the backend services. It handles incoming API requests from the client, determines the appropriate backend service, and forwards the requests accordingly. The API Gateway provides a single access point to the backend, enhancing security, scalability, and manageability.

#### 3.3 Backend Services

The backend is composed of the following microservices, each dedicated to a specific function within the Charge Locator system:

-Authentication Service: Handles user registration, login, and authentication. This service ensures that only authorized users have access to app features. It interacts with the User Database to manage user credentials and profiles securely.

-EV Charging Station Service: Manages the retrieval and update of charging station data. This service responds to user requests for finding nearby stations by fetching data from the Station Database, which stores detailed information on station locations, types of chargers, ratings, and real-time availability.

-Slot Booking Service: Responsible for managing the

booking and scheduling of charging slots at stations. This service allows users to book available slots and keeps track of booking statuses. It interacts with the Booking Database to store and retrieve booking details, ensuring real-time synchronization of availability across all users.

-Notification Service: Sends push notifications to users regarding their bookings, including confirmations, reminders, and updates. This service uses Firebase Cloud Messaging (FCM) for real-time notifications and logs each notification interaction in the Logs Database for tracking and analysis.

### 3.4 Databases

The system uses four distinct databases, each dedicated to storing data for a specific service:

- User Database: Stores user profiles, authentication credentials, and any additional user-related information.
- Station Database: Holds data on charging stations, including location, available chargers, and real-time occupancy status.
- Booking Database: Manages booking records, including user reservations and time slots for each station.
- Logs Database: Records all notification activities and system logs, providing a history of notifications sent to users for tracking and auditing purposes.

### 3.5 Process Flow

1. User Authentication: The user initiates registration or login on the mobile app, which sends an API request to the Authentication Service via the API Gateway. After successful authentication, the user gains access to app features.

2. Station Search and Retrieval: The user searches for charging stations. The API Gateway routes the request to the EV Charging Station Service, which retrieves the relevant data from the Station Database and returns it to the app for display.

3. Slot Booking: When a user selects a time slot at a specific station, the request goes through the API Gateway to the Slot Booking Service, which updates the Booking Database. This service manages the booking status and synchronizes availability in real-time.

4. Notifications: The Notification Service sends confirmations and reminders to the user based on booking activity. Each notification is logged in the Logs Database for future reference.

## IV. RESULT

The Charge Locator project has been successfully implemented as a robust platform for electric vehicle (EV) owners to locate and reserve charging stations with ease. Built using Flutter for the frontend, the app provides a seamless user experience with intuitive navigation for account registration, login, station search, and booking. Integrated GPS functionality enables real-time location tracking, making it simple for users to find nearby charging stations. The backend, developed with Spring Boot, manages essential features like user authentication, station data, and booking services via well-structured RESTful APIs. Real-time updates on station availability and booking status are achieved through WebSocket integration, enhancing responsiveness. Firebase Cloud Messaging supports timely notifications, ensuring users stay informed about their reservations. Additionally, the integration with Google Maps API enriches the experience by providing precise mapping and navigation capabilities. A secure payment gateway has been incorporated, streamlining transactions during the booking process. Feedback post-launch has been very positive, with users praising the app's usability and the convenience it brings to EV owners. Charge Locator addresses critical challenges in the EV charging ecosystem, supporting infrastructure growth and simplifying the charging experience for users. Future updates based on user feedback are planned to refine and expand its functionality.

## V. CONCLUSION

In conclusion, the Charge Locator project has successfully delivered a comprehensive solution for electric vehicle (EV) owners to efficiently locate, book, and manage charging station reservations. By leveraging a modular system architecture, the application seamlessly integrates a Flutter-based frontend with a robust Spring Boot backend, enabling smooth interactions and real-time updates. The implementation of GPS tracking, WebSocket communication, Firebase Cloud Messaging, and Google Maps API has significantly enhanced user experience, allowing EV owners to find nearby stations, receive instant updates, and navigate efficiently. Additionally, the secure payment gateway

facilitates hassle-free transactions, adding another layer of convenience.

User feedback highlights the platform's usability, reliability, and positive impact on the EV ecosystem by addressing key pain points in finding and reserving charging stations. Charge Locator has proven to be an effective tool in supporting the growth of EV infrastructure and fostering a more accessible charging network for EV drivers. As EV adoption continues to rise, future enhancements to the app will focus on further refining functionality, incorporating user suggestions, and expanding features to adapt to the evolving needs of EV owners. ChargeLocator stands as a valuable contribution toward sustainable transportation solutions and the development of smart cities.

## VI. REFERENCES

- [1] A. Joshi, K. M. Vishall Somaiya, A. Tharakaram Hariram, and M. Hussain, "Electric Vehicle Charging Station," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 9, no. 1, pp. 56-70, Mar. 2023.
- [2] N. Omase, S. K. Mittal, S. Palaniraja, P. Guchhait, M. Patil, and P. Mundra, "A Comprehensive Review of Electric Vehicle Charging Infrastructure and Associated Challenges," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 345-360, May 2022.
- [3] D. A. Hussien, W. A. Omran, and R. M. Sharkawy, "Smart Charging of Electric Vehicles in Charging Stations," *IEEE Transactions on Electrical Engineering and Control*, vol. 69, no. 4, pp. 789-800, Apr. 2021
- [4] S. S. G. Acharige, M. E. Haque, M. T. Arif, N. Hosseinzadeh, K. N. Hasan, and A. M. Than Oo, "Review of Electric Vehicle Charging Technologies, Standards, Architectures, and Converter Configurations," *IEEE Access*, vol. 8, pp. 123456- 123478, 2020.
- [5] Charzer. *Charzer App*, Google Play Store. [Online]. Available: <https://play.google.com/store/apps/details?id=com.first.faebikes.faeconsumerapp>.
- [6] PlugShare. *PlugShare App*, Google Play Store. [Online]. Available: <https://play.google.com/store/apps/details?id=com.xato ri.Plugshare>.