

# **Smart Electric Vehicle Charging Station Locator with Route Planning & Slot Booking**

## **Abstract:**

The proposed **Smart Electric Vehicle Charging Station Locator with Route Planning & Slot Booking** system offers an intelligent platform that enables EV users to easily locate nearby charging stations, plan optimal travel routes, and reserve charging slots in advance. The application features two primary interfaces: an **Admin module** for managing cities, locations, charging stations, charger types, pricing, and user data, and a **User module** that allows EV owners to register, log in, select cities and locations, view charging stations, check pricing and charger types, and book available time slots. The system enhances convenience by providing real-time slot availability and maintaining a detailed booking history for users. Built using **ReactJS** for the front-end, **Spring Boot (Java)** for the back-end, and **MySQL** for secure data storage, the platform ensures high performance, scalability, and seamless user experience. By integrating route planning and smart station discovery, this solution promotes efficient energy usage, reduces waiting time at charging stations, and supports the growing adoption of electric vehicles.

**Keywords:** Electric Vehicle, Charging Station Locator, Route Planning, Slot Booking, ReactJS, Spring Boot, MySQL.

## **Problem Statement:**

The rapid increase in electric vehicle adoption has created a growing demand for accessible and efficient charging infrastructure, yet users often struggle to locate suitable charging stations in unfamiliar areas. Existing systems lack real-time information on station availability, pricing, and charger types, leading to inconvenience and unnecessary travel. Additionally, the absence of route-based station recommendations makes long-distance planning difficult for EV users. Many stations also operate without a structured slot-booking mechanism, resulting in long waiting times. Administrators face challenges in managing station data, pricing updates, and user information efficiently. These gaps highlight the need for a unified digital platform that

simplifies EV charging discovery and reservation. A comprehensive solution is required to streamline user experience while improving station utilization.

## **Motivation:**

The motivation behind this system arises from the growing necessity to support sustainable transportation through reliable EV charging accessibility. As more users transition to electric vehicles, ensuring a smooth and predictable charging experience becomes crucial. Providing real-time station availability and route-based suggestions encourages confidence in long-distance EV travel. A slot-booking feature reduces waiting times and enhances operational efficiency at charging points. Administrators benefit from centralized management tools that simplify station maintenance and data updates. By integrating advanced technologies like ReactJS and Spring Boot, the system delivers a modern, user-friendly solution. Overall, the project aims to promote increased EV adoption by eliminating common charging-related challenges.

## **Scope:**

The scope of this project includes developing a complete web-based application that enables users to locate charging stations, view pricing and charger types, and book available time slots. The system supports administrators in managing cities, locations, station details, pricing, and user records through a secure dashboard. Users can register, log in, plan routes, and track their slot-booking history. Real-time data access ensures accurate visibility of station availability. The platform focuses on usability, scalability, and smooth integration between front-end and back-end services. It encompasses database design, API development, and interactive user interfaces. The system is intended for EV users, charging station providers, and urban mobility planners.

## **Existing Methodology:**

Existing systems for locating EV charging stations typically rely on basic map-based applications that only display station locations without offering real-time slot availability or detailed charger information. Many platforms do not include route planning features, forcing users to manually determine charging stops during long trips. Slot booking is often unavailable,

resulting in unpredictable waiting times and inefficient station usage. Some systems depend on fragmented datasets that lack accuracy or timely updates. Administrative management tools are limited, making it difficult to handle pricing, charger types, and user data effectively. Users often face incomplete information about station capacity, operating hours, and pricing variations. The lack of an integrated platform leads to inconsistent and inconvenient charging experiences for EV owners. Overall, existing approaches fail to fully support the growing needs of modern EV ecosystems.

### **Disadvantages:**

1. No real-time slot availability or booking feature.
2. Lack of integrated route planning for optimized travel.
3. Incomplete or outdated station information.
4. Poor administrative control over station and pricing data.
5. Inefficient user experience leading to longer wait times and uncertainty.

### **Proposed Methodology:**

The proposed methodology introduces a unified digital platform that integrates station discovery, route planning, and slot booking into a seamless user interface. The system collects and displays real-time station availability, pricing, and charger type information to support informed decisions. Users can search by city and location, plan optimized routes based on charging needs, and reserve slots to avoid unnecessary waiting. Administrators manage cities, stations, charger types, and pricing through a secure, centralized dashboard. The back-end, powered by Spring Boot, handles API requests, authentication, and data processing, while ReactJS ensures a responsive and intuitive front-end experience. MySQL stores structured data for users, bookings, stations, and locations. This approach enhances reliability, usability, and scalability across all system components.

### **Advantages:**

1. Real-time slot booking reduces waiting time and enhances user convenience.
2. Route planning ensures efficient and optimized EV travel.
3. Accurate and updated station data improves decision-making.

4. Centralized admin dashboard streamlines management operations.
5. Scalable architecture ensures long-term sustainability and performance.

## Software & Hardware Requirements:

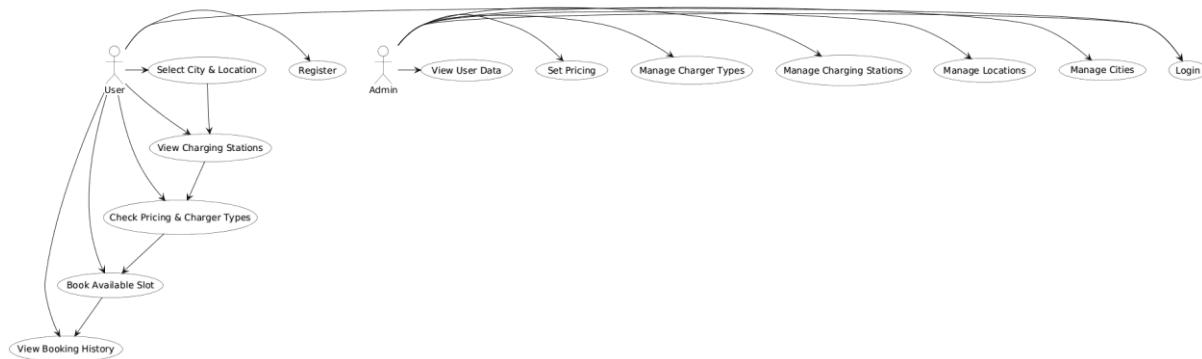
### Software Requirements:

- **Frontend:** ReactJS, HTML, CSS, JavaScript
- **Backend:** Spring Boot (Java), JDK 11+
- **Database:** MySQL
- **Tools:** VS Code/IntelliJ, Postman, Git
- **OS:** Windows / Linux / macOS

### Hardware Requirements

- **For Development:** i5 processor, 8–16 GB RAM, 50+ GB storage
- **For Server:** Quad-core CPU, 8–16 GB RAM, 100 GB SSD
- **For Users:** Any phone/PC with internet and a modern browser

## Project Flow:



## 1. Admin Module

### 1.1 City Management

- Add, update, and delete cities.

### 1.2 Location Management

- Manage locations within cities where charging stations are present.

### **1.3 Charging Station Management**

- Add and manage charging stations, charger types, number of slots, and pricing.

### **1.4 User Management**

- View registered users and manage user-related data.

### **1.5 Booking Management**

- Monitor all bookings, slot status, and station availability.

## **2. User Module**

### **2.1 User Registration & Login**

- Secure sign-up and authentication for EV owners.

### **2.2 Station Discovery**

- View charging stations by selecting city and location.
- See charger types, available slots, and pricing.

### **2.3 Route Planning**

- Suggest optimal routes to selected charging stations.

### **2.4 Slot Booking**

- Check real-time availability and book preferred time slots.

### **2.5 Booking History**

- View past and upcoming bookings.

## **3. System/Backend Module (Core Functionalities)**

### **3.1 Database Management**

- Store user data, stations, charger types, pricing, and booking history.

### **3.2 API Services**

- REST APIs built using Spring Boot for communication between front-end and backend.

### **3.3 Authentication & Security**

- Handles login, session security, and role-based access.

### **3.4 Route Optimization Service**

- Calculates and delivers optimal route suggestions.