"WattWay -Navigate Smarter, Charge Better"

A Major Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Engineering in Computer Science Engineering

Submitted by:

Mayank Piparde (0827CS221160)
Naman Shah (0827CS221174)
Nitya Maheshwari(0827CS221185)
Parth Gupta (0827CS221189)

Guided by:

Prof. Akshay Dubey

Computer Science and Engineering

Acropolis Institute of Technology & Research, Indore

Jul - Dec 2025

EXAMINER APPROVAL

The Major Project entitled ***"WattWay – Navigate Smarter, Charge Better"*** submitted by **Mayank Piparde (0827CS221160), Naman Shah (0827CS221174), Nitya Maheshwari (0827CS221185) and Parth Gupta (0827CS221189)** has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Technology degree in Computer Science Engineering*** discipline, for which it has been submitted. It understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner)   **(External Examiner)**

Date: **Date:**

RECOMMEDATION

This is to certify that the work embodied in this major project entitled **"WattWay–Navigate Smarter, Charge Better"** submitted by **Mayank Piparde (0827CS221160), Naman Shah (0827CS221174), Nitya Maheshwari (0827CS221185) and Parth Gupta (0827CS221189)** is a satisfactory account of the bonafide work done under the supervision of ***Prof. Akshay Dubey***, is recommended towards partial fulfillment for the award of the Bachelor of Technology (Computer Science Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

STUDENTS UNDERTAKING

This is to certify that the major project entitled ***"WattWay"*** has been developed by us under the supervision of ***Prof. Akshay Dubey.*** The whole responsibility of the work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

Mayank Piparde (0827CS221160)

Naman Shah (0827CS221174)

Nitya Maheshwari (0827CS221185)

Parth Gupta (0827CS221189)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely. There are a number of people without whom this project would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Prof. Akshay Dubey,** Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision, and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, Professor & Head CSE, AITR Indore for his support, suggestion, and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of the department for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

We are grateful to **our parents** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

Mayank Piparde (0827CS221160), Naman Shah (0827CS221174), Nitya Maheshwari (0827CS221185) and Parth Gupta (0827CS221189)

Executive Summary

WattWay - Navigate Smarter, Charge Better

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of ***Prof. Akshay Dubey***.

WattWay is a mobile-based EV route planning application designed to overcome the challenges of range anxiety and scattered charging infrastructure faced by electric vehicle users in India. The app calculates energy-feasible routes by considering the vehicle's State of Charge (SoC), reserve limits, battery health, and charger compatibility, while also integrating real-time traffic data to ensure accurate ETAs and dynamic rerouting. Users can explore multiple route options—fastest, cheapest, or with minimum charging stops—and receive suggestions for nearby amenities such as food outlets, restrooms, and ATMs at charging locations. Built using **React Native** for the mobile frontend, **Node.js with Express** for backend services, and **MongoDB** for scalable data management, WattWay delivers a seamless and reliable navigation experience tailored to the needs of EV commuters in India.

Key words: Route planning application

"Where the vision is one

year, cultivate flowers;

Where the vision is ten years,

cultivate trees;

Where the vision is eternity,

cultivate people."

- Oriental Saying

List of Figures

List of Tables

List of Abbreviations

Abbr1: UI – User Interface

Abbr2: DB - Database

Abbr3: DFD - Data Flow Diagram

Abbr4: JSON- Java Script Object Notation

Abbr5: API – Application Programming Interface

Table of Contents

`

Chapter 1

Introduction

In today's rapidly evolving urban mobility landscape, electric vehicle (EV) adoption is rising steadily in India, yet range anxiety and scattered charging infrastructure continue to be major hurdles for EV users. Planning long or even medium-distance journeys often becomes stressful, as drivers must manually locate charging stations, verify compatibility, and estimate charging needs, leading to inefficiencies and unplanned delays. WattWay emerges as a dedicated solution to this challenge by providing an integrated mobile platform for smart EV route planning.

The Indian EV market, projected to reach over $113 billion by 2029, is growing at an accelerated pace, yet navigation tools optimized for EV needs remain limited. While conventional apps offer general routing, they rarely account for EV-specific requirements such as State of Charge (SoC), reserve limits, battery health, or charger compatibility. **WattWay** bridges this gap by offering energy-feasible routes tailored to each vehicle, suggesting compatible charging stations, and integrating real-time traffic data for accurate ETAs. With features like multiple route options (fastest, cheapest, minimal charging stops) and convenience-driven recommendations for nearby amenities, WattWay is designed to make EV journeys more reliable, stress-free, and user-friendly for Indian commuters.

1.1 Overview

WattWay is a mobile application designed to address the growing challenges of range anxiety, scattered charging infrastructure, and inefficient route planning faced by electric vehicle (EV) users in India. By integrating EV-specific parameters with real-time traffic and charging data, WattWay ensures energy-feasible, reliable, and stress-free journeys. With its frontend built on **React Native**, backend powered by **Node.js with Express**, and **MongoDB** for data management, the platform delivers a seamless and scalable user experience tailored to the needs of Indian EV commuters.

Key Objectives:

- To calculate energy-feasible routes by considering vehicle State of Charge (SoC), reserve limits, and battery health.
- To suggest compatible charging stations by matching EV models with charger types and power ratings.
- To provide users with traffic and weather updates. Also the estimated time of arrival.
- To integrate real-time traffic data for accurate ETAs and dynamic rerouting.
- To enhance user convenience by recommending nearby amenities (food, restrooms, ATMs) around charging stations.
- To create a secure, scalable, and user-friendly mobile platform for EV navigation and trip management.

1.2 Background and Motivation

The adoption of electric vehicles (EVs) in India is accelerating as the nation pushes towards sustainable and eco-friendly mobility solutions. However, despite government initiatives and increasing consumer interest, EV users continue to face critical challenges such as range anxiety, scattered charging infrastructure, and **lack of EV-specific navigation support**. Planning even a medium-distance journey often becomes stressful, as drivers must manually check their vehicle's State of Charge (SoC), search for compatible charging stations, and estimate energy consumption—all without reliable digital assistance. Existing navigation apps, while efficient for conventional vehicles, do not account for the unique needs of EVs, leaving a significant gap for users seeking stress-free and reliable journeys.

Our team recognized this gap through careful observation of the EV landscape in India and the limitations of current navigation tools. We noticed that while the EV market is projected to grow rapidly, there are very few solutions tailored specifically to address the needs of EV drivers in terms of **charging compatibility, route optimization, and real-time decision-making**. This realization motivated us to conceptualize **WattWay**, a mobile application designed to provide energy-feasible route planning, suggest compatible charging stops, and integrate real-time traffic data for accurate and adaptable navigation. Unlike generic navigation apps, WattWay focuses exclusively on the EV ecosystem, aiming to make electric mobility more practical, reliable, and user-friendly for modern commuters.

1.3 Problem Statement and Objectives

EV users in India face significant challenges due to range anxiety, limited and scattered charging infrastructure, and the lack of navigation tools tailored to electric mobility. Existing map applications are primarily designed for conventional vehicles and fail to account for EV-specific requirements such as State of Charge (SoC), reserve limits, battery health, and charger compatibility. This forces EV users to rely on manual planning or incomplete information, often leading to inefficient routes, unplanned delays, and uncertainty during journeys. The absence of an integrated, EV-focused navigation solution prevents users from fully leveraging the benefits of electric mobility, creating a major barrier to adoption and convenience.

Thus, the system implemented has the following **objectives**:

- To develop a mobile-based platform that calculates **energy-feasible routes** by considering SoC, reserve limits, and battery health.
- To implement charger compatibility checks based on vehicle model, charger type, and power ratings.
- To provide **multiple route options** including fastest, cheapest, and minimal charging stops.
- To integrate **real-time traffic data** for accurate ETAs and dynamic route adjustments.
- To enhance convenience by suggesting **nearby amenities** (food, restrooms, ATMs) around charging stations.
- To design a secure, scalable, and user-friendly mobile app using React Native, Node.js with Express, and MongoDB, ensuring smooth performance for EV commuters.

1.4 Scope of the Project

WattWay's scope encompasses the development of an intelligent and energy-feasible **EV Route Planning System** designed specifically for Indian road conditions and charging infrastructure. The project aims to assist electric vehicle users in identifying optimal routes, ensuring compatibility with available chargers, and minimizing range anxiety. The scope, boundaries, and deliverables are defined as follows:

- **User authentication and vehicle registration module** for securely storing EV details such as model, battery capacity, connector type, and State of Charge (SoC).

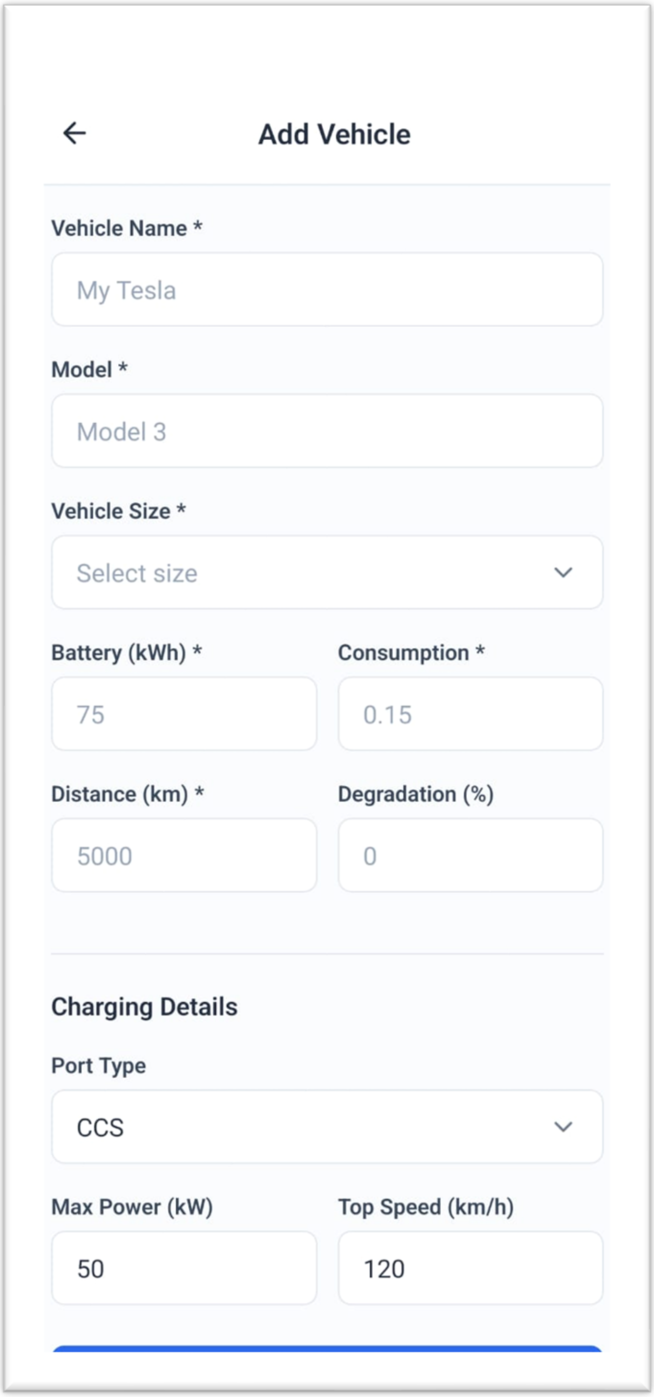


Figure 1-1: Add Vehicle Module

- **Dynamic route planning engine** using Dijkstra's algorithm to compute energy-feasible paths based on SoC, reserve limits, and battery health.

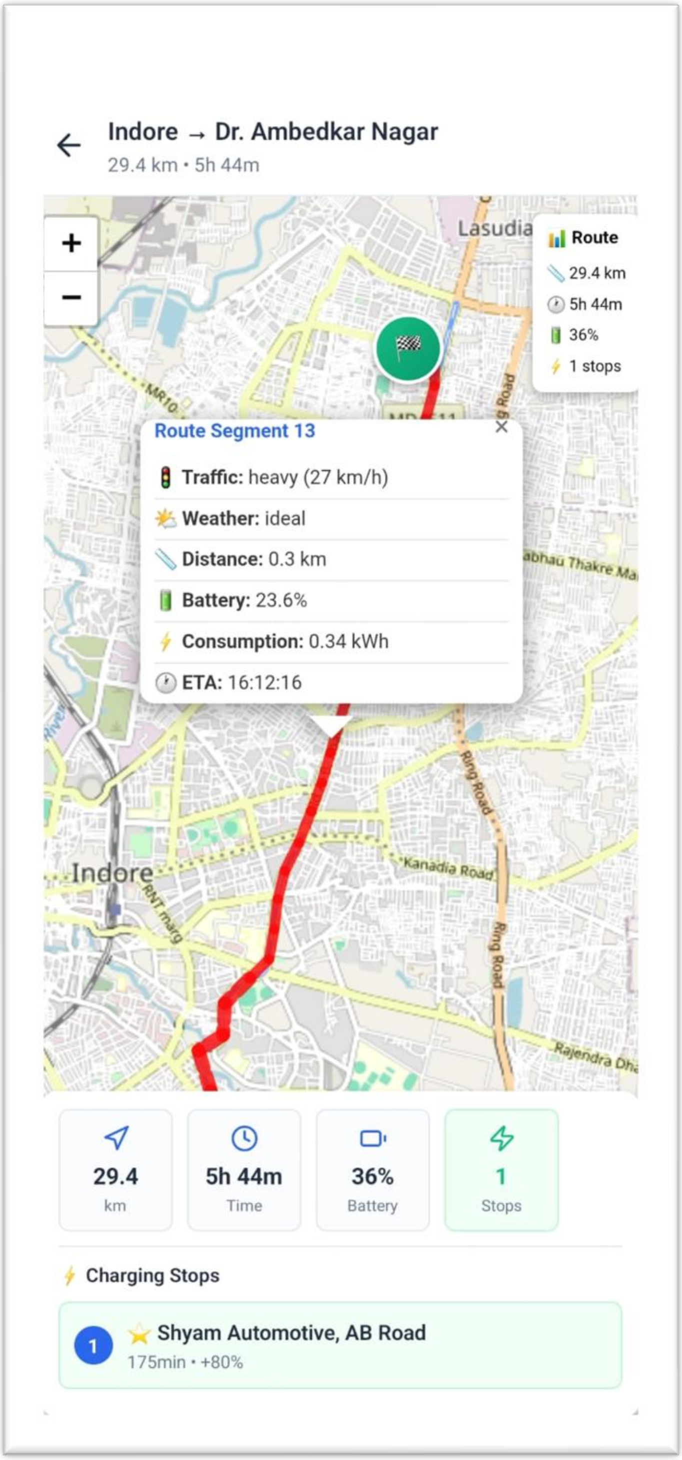


Figure 1-2: Energy-Feasible Route Computation

- **Charging station recommendation system** that identifies compatible stations (connector type, power rating) along the route using integrated datasets from OpenStreetMap, Mapbox, and verified charging networks.

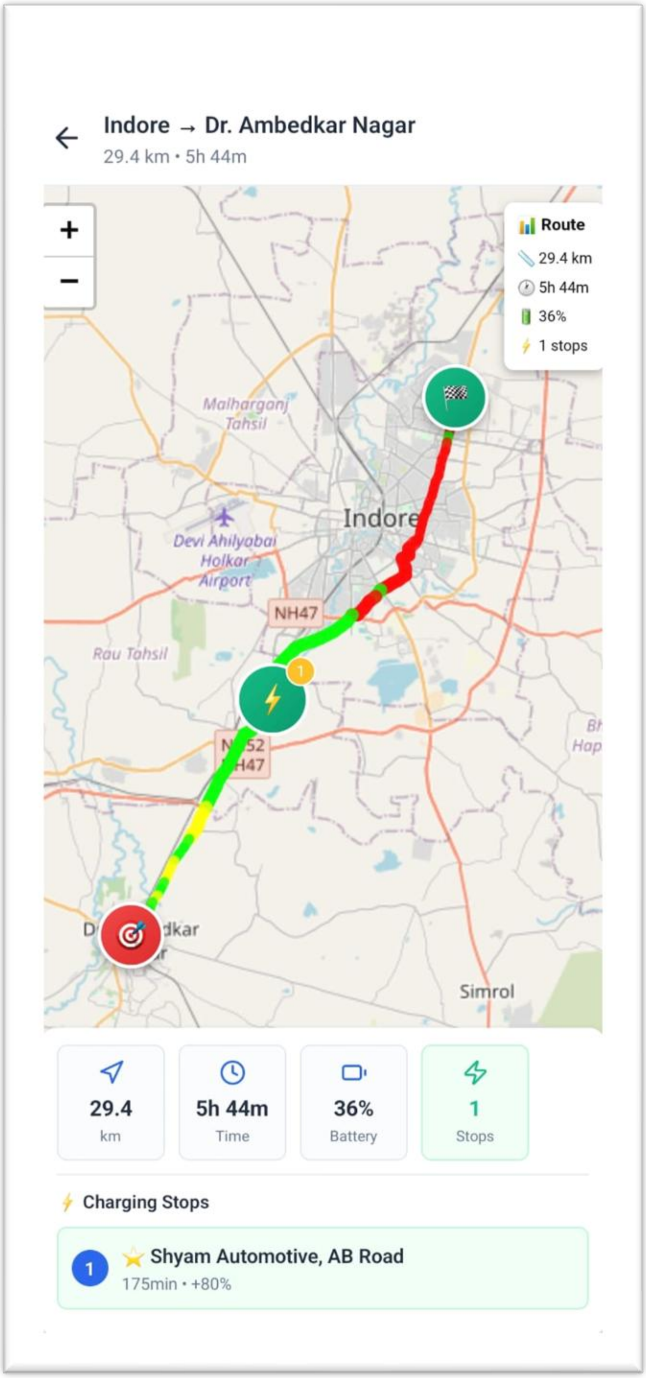


Figure 1-3: Charging Station Suggestion

- **Integration of real-time traffic data** to ensure accurate Estimated Time of Arrival (ETA) and adaptive re-routing when congestion or unexpected delays occur.

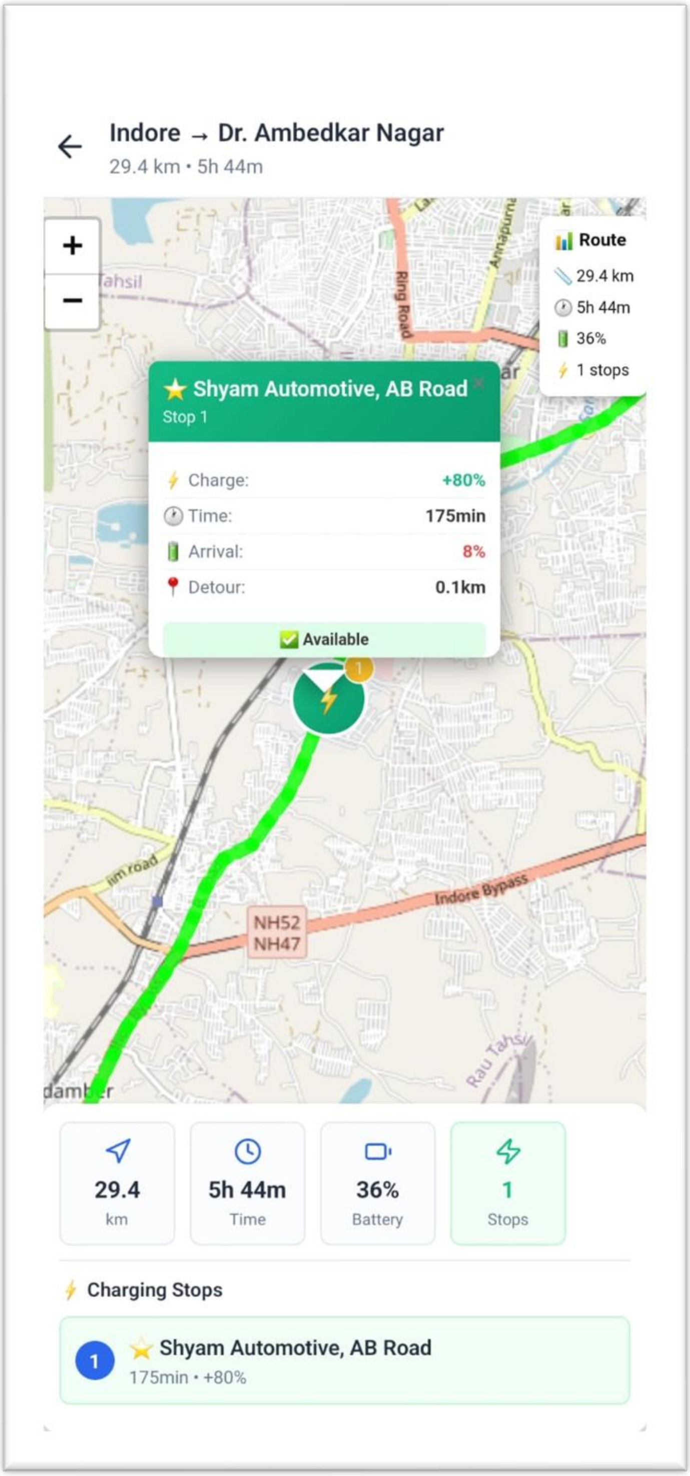


Figure 1-4: ETA Suggestion

- Suggest nearby amenities to the user for their convenience to the user.

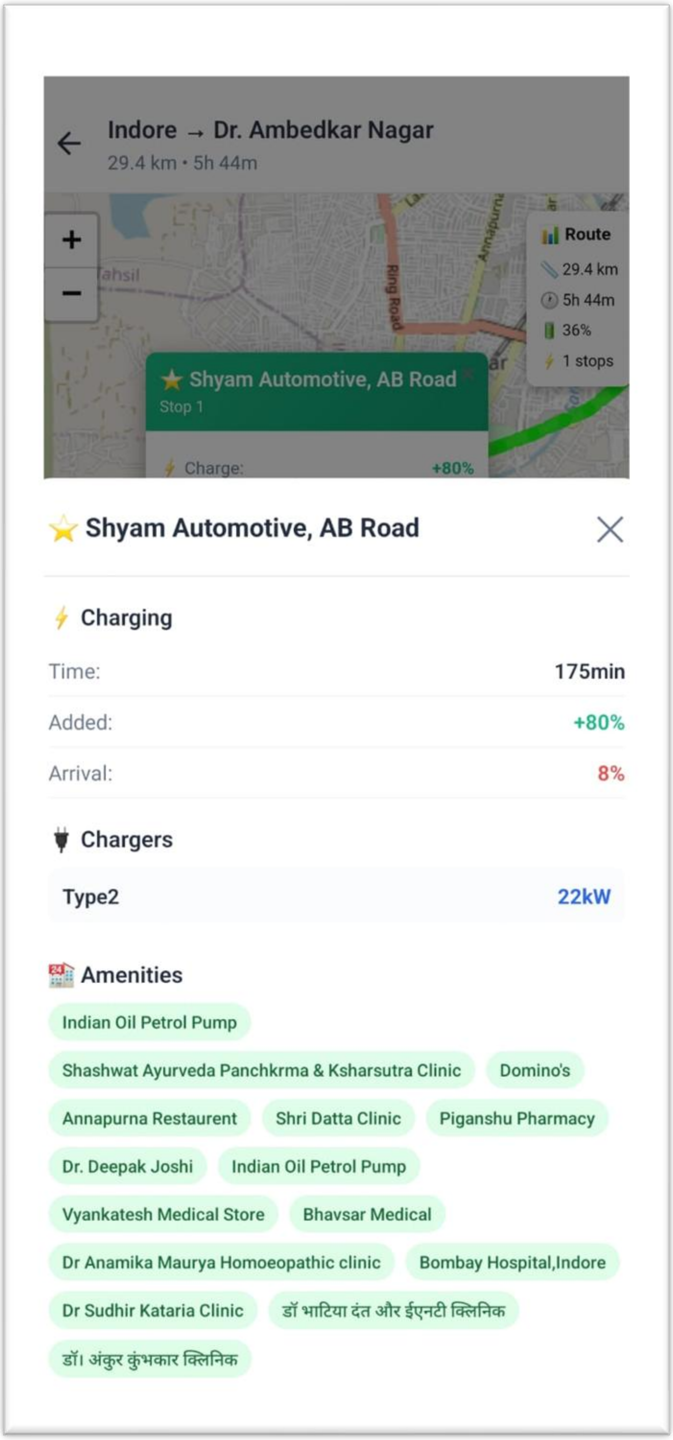


Figure 1-5: Amenities Suggestion

- **Limitations:** The current system model of WattWay faces certain limitations. Real-time charger availability, dynamic electricity pricing, and live SoC data integration are currently out of scope and will be addressed in future iterations.

1.5 Team Organization

- **Mayank Piparde:**

  Mayank has been instrumental in building the frontend of our web application. Leveraging technologies like React Native, he created several beautiful and responsive pages. He also merged all the pages, ensuring the website is fully functional and cohesive, making the vision of the project a reality.

- **Naman Shah:**

  Naman led the research and analysis of existing systems, identifying key gaps that shaped our solution. He co-authored the research paper and designed UML diagrams that guided development. He actively contributed to system testing and played a major role in creating the technical poster and presentation.

- **Nitya Maheshwari:**

  Nitya contributed extensively to both the analytical and presentation aspects of the project. Alongside Naman, she prepared a comprehensive research paper and collaboratively developed UML diagrams that structured the development process. Nitya also took an active role in testing the system to ensure performance accuracy and robustness. Her creativity and communication skills were instrumental in designing the technical poster and presentation, delivering a polished and impactful project showcase.

- **Parth Gupta:**

  Parth played a vital role in developing the backend and managing the database. He played a key role in implementing the logic that brought the platform to life. Additionally, Parth contributed innovative features and functionalities that enhance the overall usability of the website.

1.6 Report Structure

The project ***WattWay*** is primarily concerned with providing an integrated mobile platform for smart EV route planning. It is designed to make EV journeys more reliable, stress-free, and user-friendly for Indian commuters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2

Review of Literature

The development of **WattWay** requires a detailed understanding of electric vehicle adoption trends, charging infrastructure availability, and digital navigation solutions in India. This chapter examines existing research and industry reports related to EV mobility, range optimization techniques, and navigation applications to establish the groundwork for our energy-feasible route planning system. By analyzing current offerings such as **Google Maps, PlugShare, Statiq, and ChargePoint**, along with studies on EV range anxiety and charging behavior, we identify the critical gaps that WattWay seeks to address. Unlike generic mapping tools, WattWay is designed to specifically cater to the needs of EV users in the Indian context, ensuring reliable navigation, compatible charging options, and enhanced commuter confidence.

2.1 Preliminary Investigation

2.1.1 Current System

The existing landscape for electric vehicle (EV) navigation in India is dominated by generic map applications such as Google Maps, which are primarily designed for conventional vehicles and do not address EV-specific requirements. EV users currently rely on a combination of navigation apps, third-party charging station locators, and manual planning to estimate range and identify charging points. This fragmented approach often leads to inefficiencies such as detours to incompatible charging stations, unplanned delays due to limited infrastructure, and uncertainty in calculating energy-feasible routes. The lack of an integrated system tailored for EVs creates significant barriers for users, particularly in long-distance travel or in regions with limited charging networks, reinforcing range anxiety and reducing confidence in EV adoption.

1. **Google Maps:**
   - **Description:**
     Widely used navigation platform offering real-time traffic updates, route optimization, and location services for conventional vehicles.
   - **Challenges:**
     - Not designed for EV-specific requirements such as SoC monitoring, battery health, or reserve thresholdsNo curation specific to ritual requirements or ceremony types
     - No integration of charger compatibility with specific vehicle models.
     - Inability to suggest charging stops dynamically based on real-time consumption
   - **Limitations:**
     - Routes may not be energy-feasible for EVs, especially on long trips.
     - No recommendations for nearby amenities at charging stations.
     - Limited data on charger availability and operational status
     - Cannot prevent users from being routed to incompatible or non-functional chargers

2. **PlugShare:**
   - **Description:**
     A global platform for locating EV charging stations with user-generated reviews and basic trip planning features.

- **Challenges:**
  - Primarily focuses on charging station mapping rather than complete route optimizationNo capability for service bookings of any kind
  - Limited integration with live traffic data for accurate ETAs
  - Does not account for battery health, SoC thresholds, or reserve margins
- **Limitations:**
  Lack of seamless navigation integration compared to mainstream map apps.

3. **Statiq:**

- **Description:**
  An Indian EV charging network platform that provides charging station discovery and booking services.
- **Challenges:**
  - Primarily focused on charging station networks, not holistic EV route planning
  - Restricted station database (mostly tied to Statiq's own network)
- **Limitations:**
  Works best in metros; sparse coverage in smaller cities. Cannot provide optimized multi-stop routes for long-distance trips

4. **ChargePoint:**

- **Description:**
  An international EV charging network with an app for locating and accessing its stations.
- **Challenges:**
  - Focuses mainly on its proprietary charging infrastructure.
  - Does not support cross-network charger compatibility in India.

    - No comprehensive navigation features integrating traffic data with charging needs.
    - Limited to users within its own ecosystem, reducing overall usability.
- **Limitations:**
    - Incomplete station coverage for Indian EV users.
    - No personalized route planning based on SoC or reserve limits.
    - Cannot adapt dynamically to unexpected detours or changes in traffic.
    - Not tailored to the Indian EV ecosystem with its unique infrastructural challenges.

2.2 Limitations of Current System

The limitations of existing systems include:

1. **Generic Routing:** Current navigation apps are designed for conventional vehicles and don't optimize routes based on EV-specific energy consumption.

2. **Range Anxiety:** Limited awareness of nearby charging stations and insufficient consideration of SoC thresholds increases the risk of running out of battery.

3. **Charger Compatibility Issues:** Users often face incompatibility between their vehicle and available chargers (type, power level), causing unplanned delays.

4. **Battery Health Ignorance:** Existing tools do not factor in battery health or reserve limits, which can lead to inefficient or unsafe routing decisions.

5. **Lack of Multiple Optimization Options:** Standard navigation apps rarely provide route alternatives considering fastest, cheapest, or minimal charging stops.

6. **Static Traffic Data:** Current systems may not integrate real-time traffic and charger occupancy data, leading to inaccurate ETAs and increased waiting times.

7. **Limited Convenience Features:** Apps don't suggest nearby amenities like food, restrooms, or ATMs around charging stations, reducing overall trip comfort.

8. **Urban EV Challenges:** In congested cities, scattered charging infrastructure and lack of dynamic routing exacerbate delays and stress for EV users.

These gaps highlight the need for a dedicated solution like **WattWay**, which provides an integrated, energy-feasible route planning system tailored for EV users in India.

2.3 Requirement Identification and Analysis for Project

Significant research and commercial work has been conducted in the field of electric vehicle (EV) routing and charging platforms. However, creating an integrated, India-specific solution that ensures energy-feasible routes, considers SoC thresholds, battery health, charger compatibility, and driver convenience presents unique challenges. The review of literature leads to several major findings which are as under:

- As outlined in the comparative analysis of existing platforms such as Google Maps, PlugShare, Tata Power EZ Charge, Ather Grid, and Mappls [6–10], no single system currently offers a comprehensive EV route planner that integrates *real-time traffic, SoC-aware routing, charger compatibility checks, and multiple optimization modes* *(fastest, cheapest, fewest stops).* This highlights the opportunity for **WattWay** to establish a unique value proposition in the EV navigation ecosystem.

- Industry reports [1], [2] emphasize that Indian EV users cite *range anxiety and lack of charging infrastructure* as the primary obstacles to adoption. Although India had nearly 8,000 fast-charging stations by 2024 [2], their uneven distribution, especially outside Tier-1 cities, creates stress for long-distance EV travel. A reliable route planner that eliminates this anxiety by predicting charging needs is therefore essential.

- Research on EV navigation algorithms like eDijkstra [3] and predictive routing models [4] demonstrates that shortest path computation must integrate *battery constraints, charging times, and connector compatibility* to ensure feasibility. These studies establish critical benchmarks for any new routing platform, including **WattWay**, which adapts Dijkstra's algorithm to EV-specific constraints.

- Commercial solutions like Mapbox EV Routing [5] already show the importance of integrating *live vehicle telemetry, traffic conditions, and charger availability* into route computation. Without these, ETAs become unreliable and drivers face delays. WattWay leverages OpenStreetMap/Mapbox datasets with live feeds to dynamically update travel times and re-route when necessary.

- Market projections [2] reveal that India's EV ecosystem is growing rapidly, with sales and charging demand expected to scale sharply by 2030 under national policies like FAME-II. This underscores a rising demand for EV support services such as WattWay, which can reduce friction in adoption by enabling stress-free long-distance journeys.

- Existing platforms like PlugShare and Tata Power EZ Charge [7], [9] primarily focus on charging station discovery, but do not suggest *amenities, rest areas, or cost-optimized routes*. WattWay addresses this by offering nearby amenities (food, ATMs, restrooms) around charging stops, enhancing usability and driver satisfaction.

- While Mappls [6] and Ather Grid [8] offer partial features, no existing system integrates **SoC tracking, charger-vehicle compatibility, battery health, and multi-criteria optimization** into a single, adaptive routing engine. WattWay positions itself to fill this unmet requirement in the Indian EV ecosystem.

2.3.1 Conclusion

The comprehensive review of existing literature and current EV routing systems highlights significant gaps in the electric mobility ecosystem, particularly concerning reliability, energy feasibility, and integrated service offerings. While platforms such as Google Maps and Mappls provide navigation, and applications like PlugShare, Tata Power EZ Charge, and Ather Grid facilitate charging station discovery, they fall short in delivering a holistic solution that simultaneously considers State of Charge (SoC), battery health, charger compatibility, traffic conditions, and user convenience. These insights underscore the necessity for an innovative system like **WattWay**, which aims to integrate EV-specific constraints with real-time traffic, energy-aware route optimization, multi-criteria planning (fastest, cheapest, or minimum stops), and nearby amenity suggestions to ensure reliable and stress-free EV journeys in the Indian context.

Chapter 3

Proposed System

3.1 The Proposal

The proposal for **WattWay** is to develop a dedicated EV route planning platform that addresses the unique challenges faced by Indian EV users, such as range anxiety, charger compatibility, and scattered charging infrastructure. The system combines intelligent energy-based routing with dynamic traffic integration and real-time charging station data to ensure stress-free and reliable EV journeys. Additionally, WattWay incorporates personalized vehicle profiling, charger availability prediction, and amenity recommendations to enhance overall convenience and trip planning for EV drivers.

The key functionalities of the proposed system include:

1. **Energy-feasible routing** that considers State of Charge (SoC), reserve limits, and battery health before suggesting routes.
2. **Smart charger matching** that ensures compatibility between vehicle models and available charger types/power levels.
3. **Traffic and Weather** –provide users with traffic and weather updates thus allowing users to find their estimated time of arrival.
4. **Real-time traffic and charger data integration** for accurate ETAs, reduced waiting times, and adaptive route recalculations.
5. **Nearby amenities suggestions** (food, restrooms, ATMs) around charging stations for enhanced user comfort.
6. **User profiles with vehicle details and preferences** to personalize routing and station recommendations.

3.2 Benefits of the Proposed System

The proposed **WattWay** system offers the following advantages:

- **Range Confidence**: Minimizes EV users' range anxiety by providing energy-feasible routes that account for SoC, reserve limits, and battery health.
- **Charger Compatibility Assurance**: Ensures that suggested charging stations match the vehicle's model, connector type, and power requirements, reducing unplanned delays.
- **Vehicle type**: Allows users to enter their vehicle details and preferences so they can personalize their routing experience.
- **Real-Time Adaptability**: Integrates live traffic and charger occupancy data to dynamically recalculate routes, ensuring accurate ETAs and smoother travel.
- **Enhanced User Convenience**: Suggests nearby amenities (food, restrooms, ATMs) around charging stations to make charging breaks more comfortable.
- **Time and Cost Optimization**: Reduces unnecessary detours, charging delays, and energy wastage, making EV journeys more efficient and economical.

3.3 Block Diagram

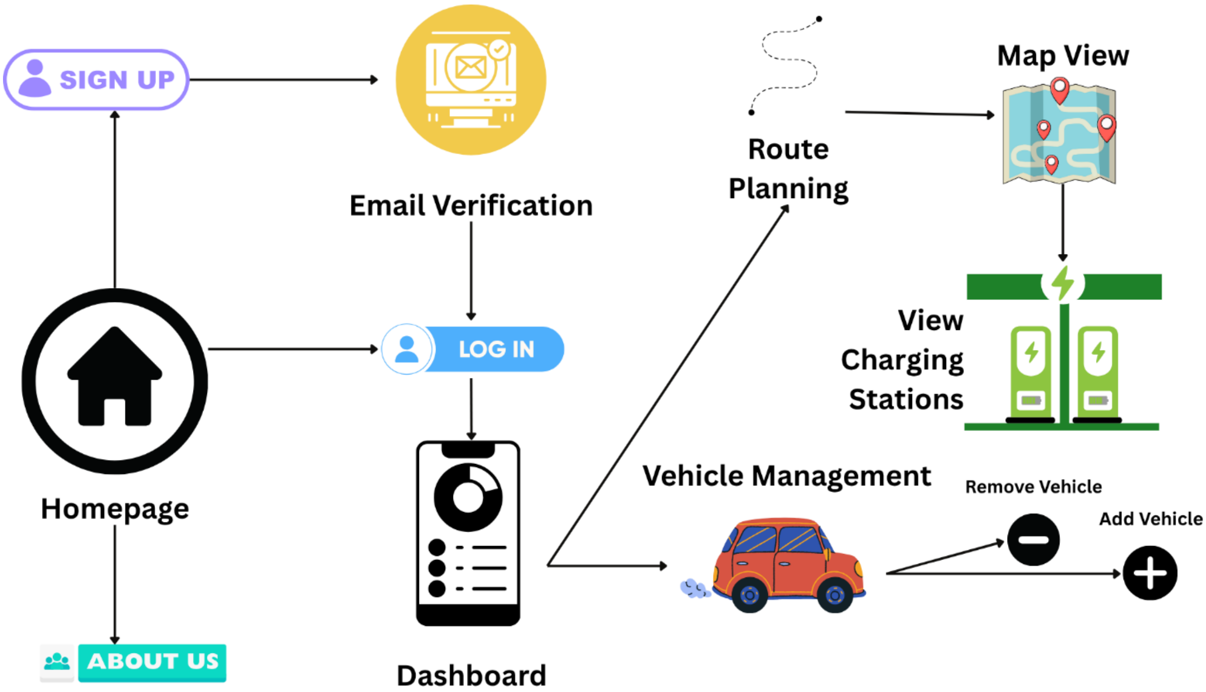


Figure 3-1: Block Diagram

3.4 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.4.1 Technical

The **WattWay** system is technically feasible due to its reliance on modern technologies, real-time data integration, and proven navigation frameworks to ensure accurate route planning, energy optimization, and user satisfaction.

1. **Platform Architecture:**

- The system employs a modular architecture combining EV-specific route planning, charging station data integration, and real-time traffic monitoring.
- This layered approach ensures flexibility for handling diverse EV models while maintaining scalability for future enhancements.

2. **Frameworks and Tools:**

- Backend development utilizes **Node.js** and **Express.js**, ensuring fast, scalable, and event-driven processing of routing algorithms.
- The frontend, built with **React Native**, delivers an intuitive and responsive interface for users.
- Integration with **MongoDB** supports efficient data storage for user profiles, vehicle specifications, and charging station databases.

3. **Automation Capabilities:**

- Real-time route recalculations based on SoC, charger availability, and traffic conditions minimize travel delays.
- Automated charger matching ensures compatibility between the EV's connector type and available charging infrastructure.

4. **Hardware Requirements:**

- Secure hosting infrastructure to ensure data privacy and fast response times for users.

5. **Reliability:**

- Designed for **24/7 uptime**, especially critical for intercity EV travel and long-distance routes.
- Advanced error-handling, redundancy mechanisms, and optimized database queries reduce risks of downtime, inaccurate routes, or failed charging recommendations.

3.4.2 Economical

The economic feasibility of the **WattWay** system lies in its potential to reduce EV travel costs, minimize range anxiety, and require a reasonable initial investment in technology and partnerships.

1. **Market Opportunity:**

- The platform targets India's rapidly growing EV market, projected to exceed 10 million EVs by 2030, with increasing demand for EV-friendly infrastructure.
- With limited existing solutions focused specifically on EV route optimization in India, WattWay presents a unique value proposition with low direct competition.

2. **Infrastructure Investment:**

- Initial costs include developing the software platform, integrating EV-specific routing algorithms, and partnering with charging networks.
- Investment is also needed for data acquisition, including real-time traffic, charging station availability, and vehicle compatibility databases.

3. **Operational Economics:**

- Revenue generation can come from subscription models, premium app features, partnerships with charging stations, and advertising opportunities for nearby amenities.
- Operational costs are optimized through scalable cloud-based services, minimizing the need for heavy physical infrastructure.

4. **Scalability:**

- WattWay's cloud-native architecture allows expansion to additional cities and regions without proportional increases in cost.

- Integration with additional EV models and charging networks can be done incrementally, keeping investments flexible and controlled.

5. **Network Effects:**

- As more EV users adopt WattWay, the platform's routing accuracy and recommendations improve through data aggregation.

3.4.3 Operational

The **WattWay** system is designed to be highly operationally feasible, with features that ensure ease of use, reliability, and adaptability.

1. **Ease of Use:**

- User-friendly mobile interface ensures EV drivers can plan routes and charging stops without extensive technical knowledge.
- Clear route options (fastest, cheapest, minimum charging stops) simplify decision-making for users.

2. **Accuracy and Efficiency:**

- Real-time traffic integration and SoC-based routing ensure trips are energy-efficient and timely.
- Compatible charging stop recommendations reduce unplanned delays and range anxiety.

3. **Adaptability:**

- Dynamic route recalculation adapts to changing traffic, battery health, and charger availability.
- Supports multiple EV models and varying battery capacities for personalized routing.

4. **Continuous Availability:**

- The system operates 24/7, providing uninterrupted route planning support across urban and rural areas.
- Redundant cloud infrastructure ensures high availability even during peak usage.

5. **EV-Specific Considerations**:

- Route planning respects battery health, reserve thresholds, and charger compatibility.
- Offers suggestions for nearby amenities (food, restrooms, ATMs) at charging stops to enhance user convenience.

6. **Quality Assurance:**

- Charging station and route data are continuously verified for accuracy.
- Feedback mechanisms allow users to report station issues, improving the platform's reliability over time.

3.5 Design Representation

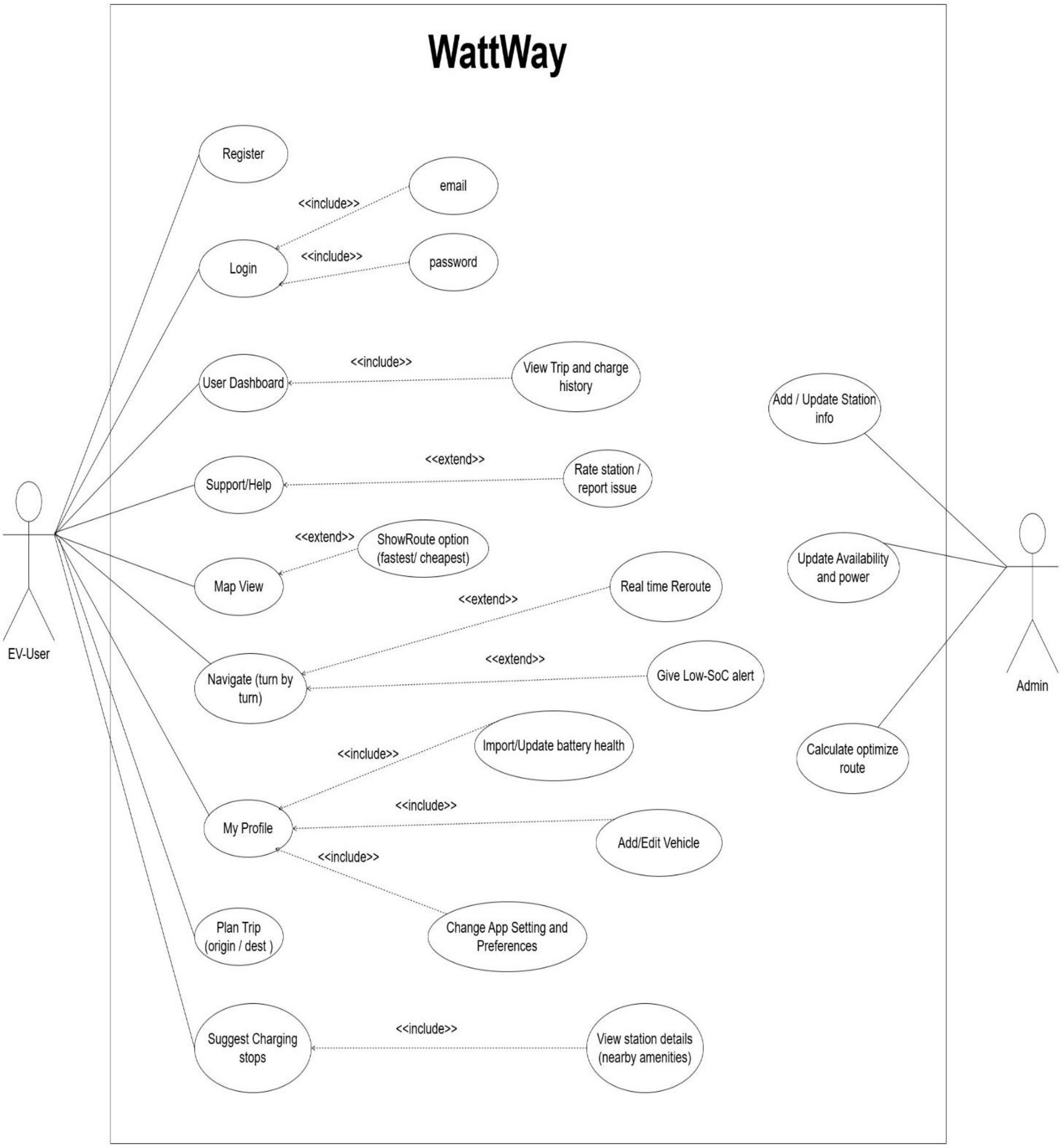


Figure 3-2: Use Case Diagram

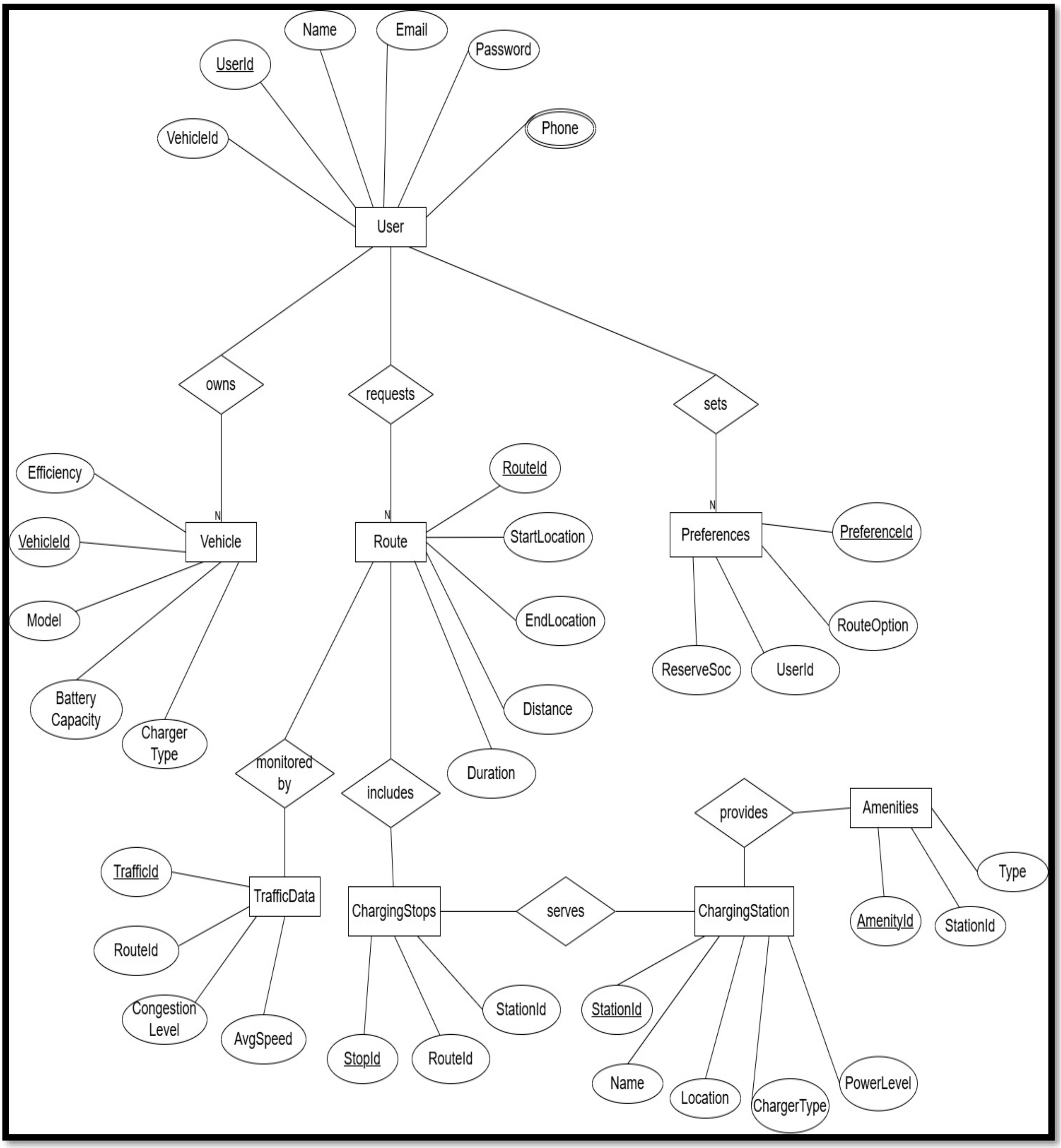


Figure 3-3: ER Diagram

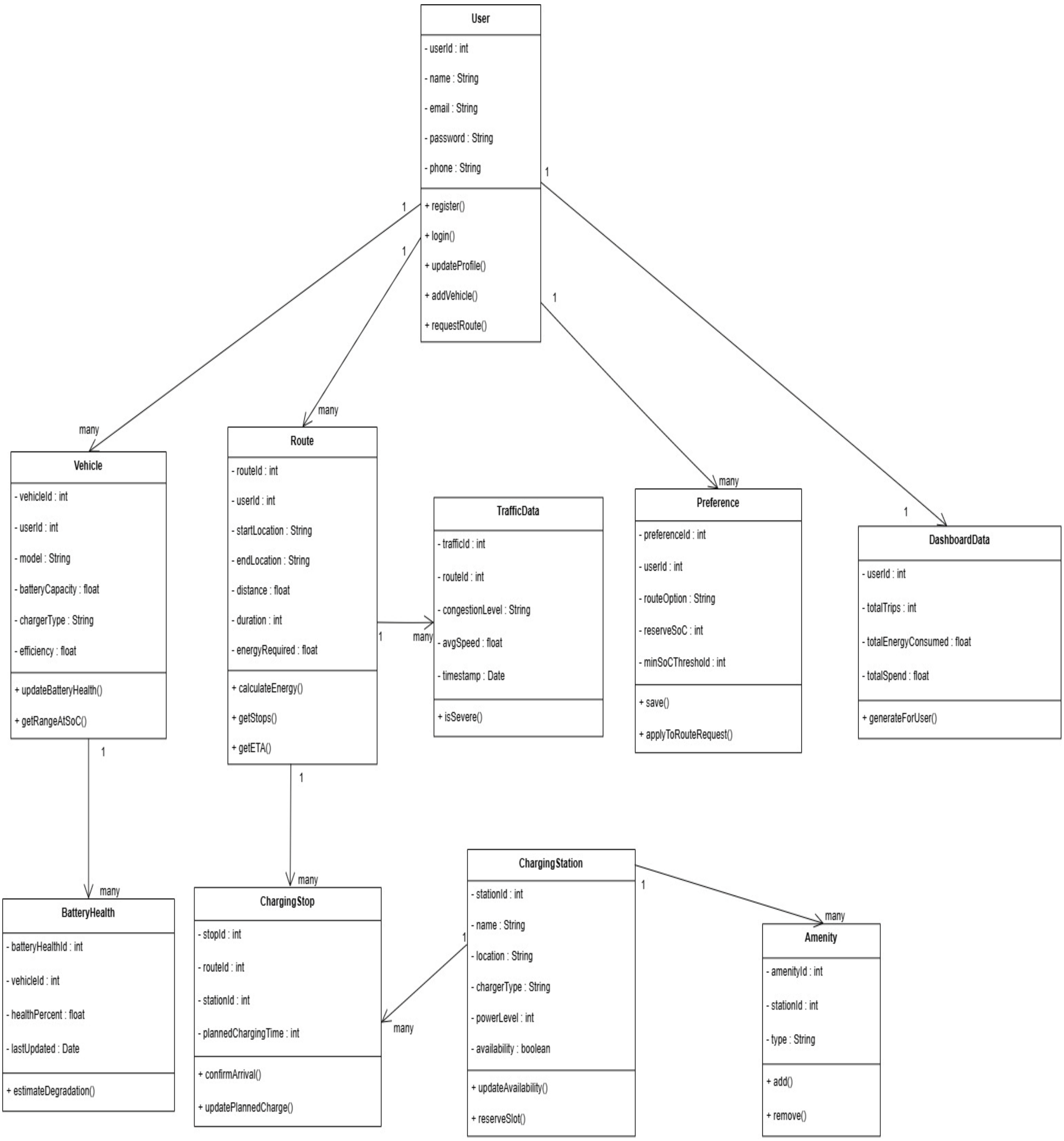


Figure 3-4: Class Diagram

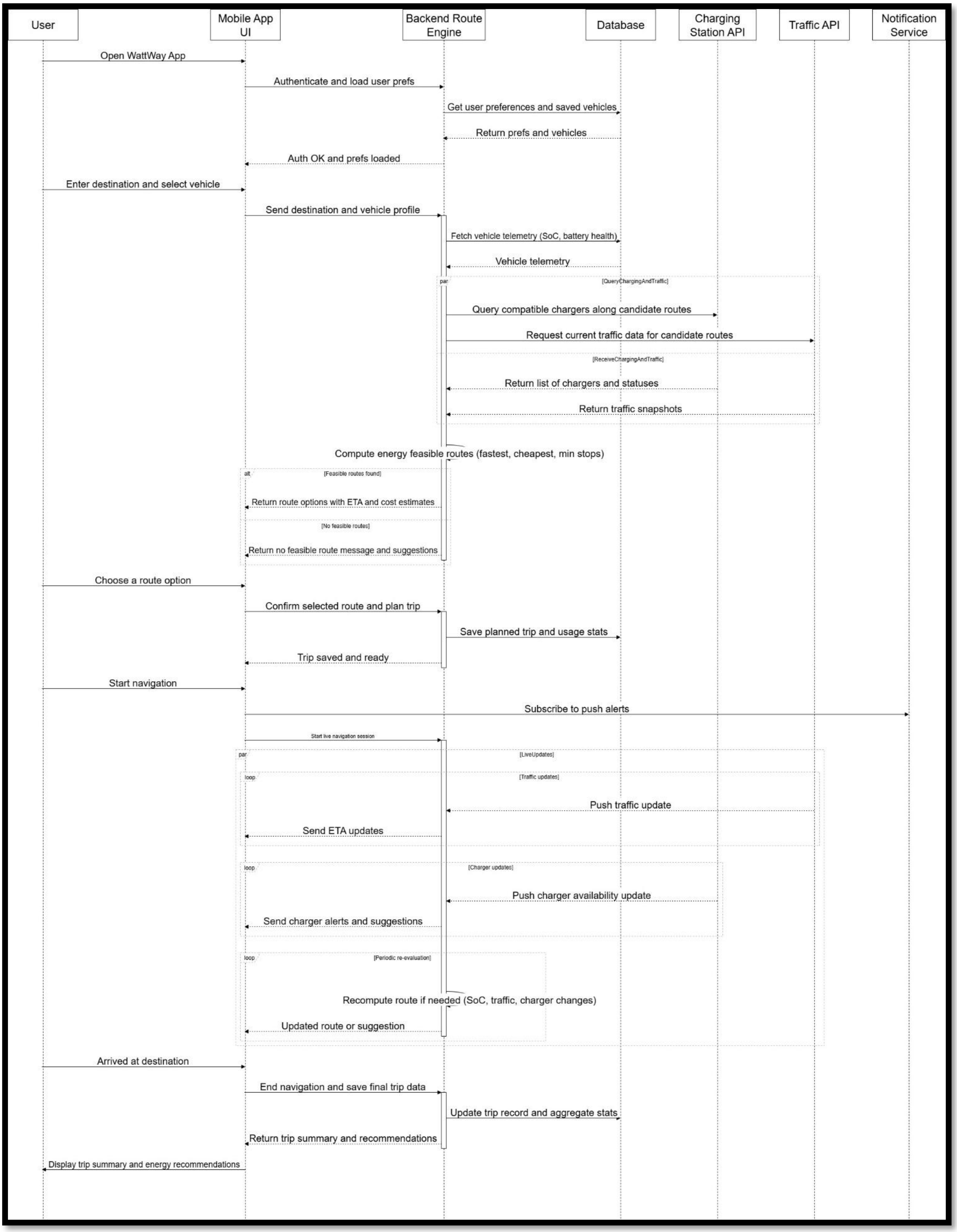


Figure 3-5: Sequence Diagram

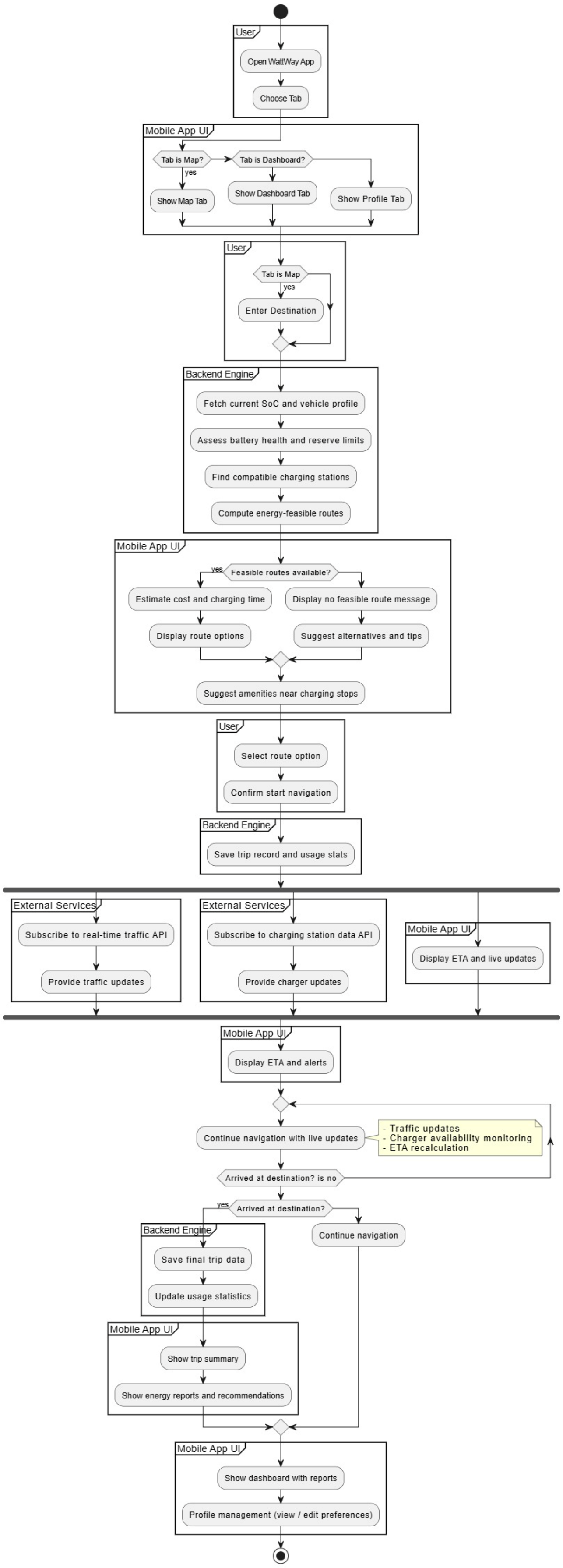


Figure 3-6: Activity Diagram

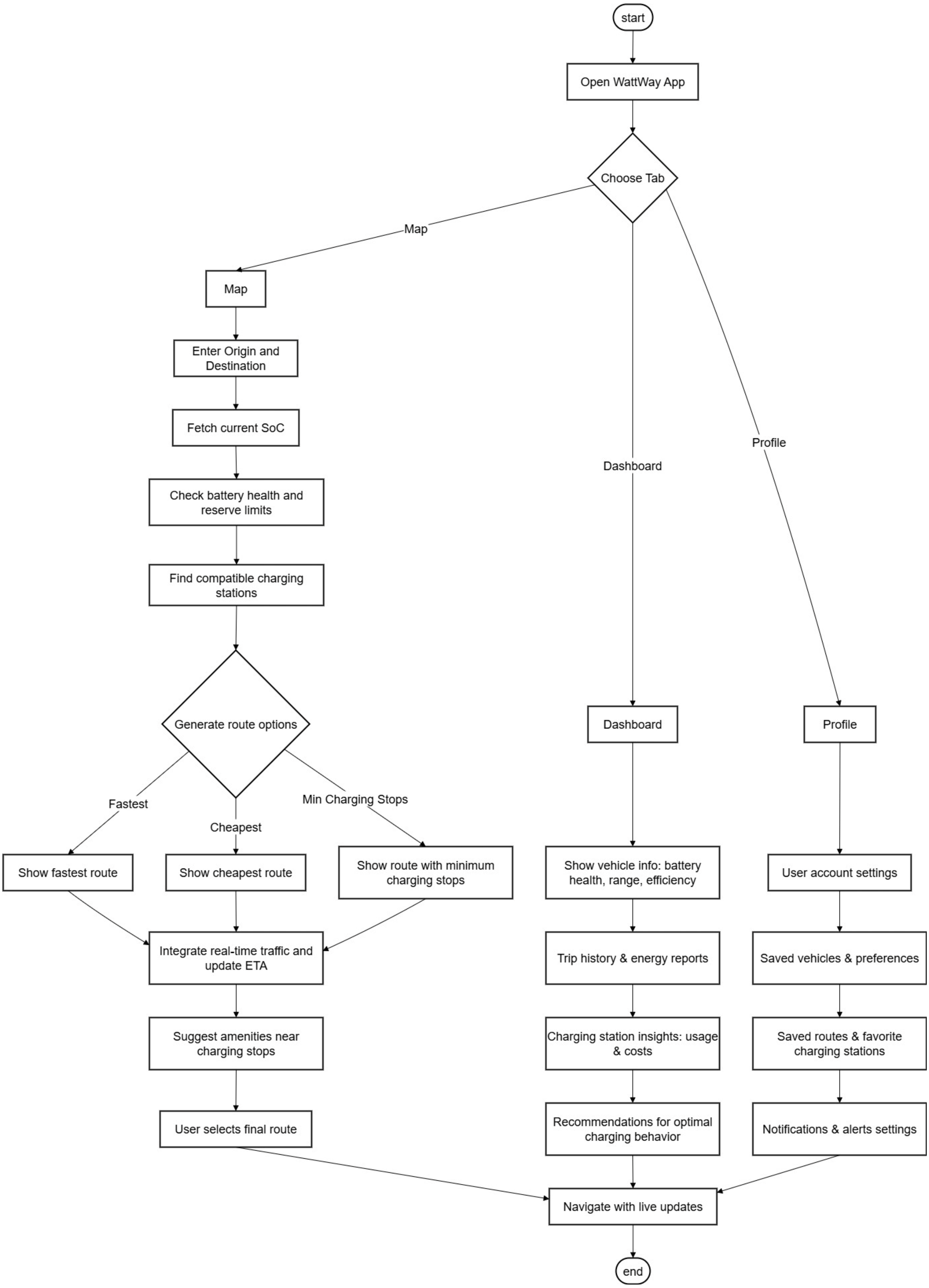


Figure 3-7: Flow Chart

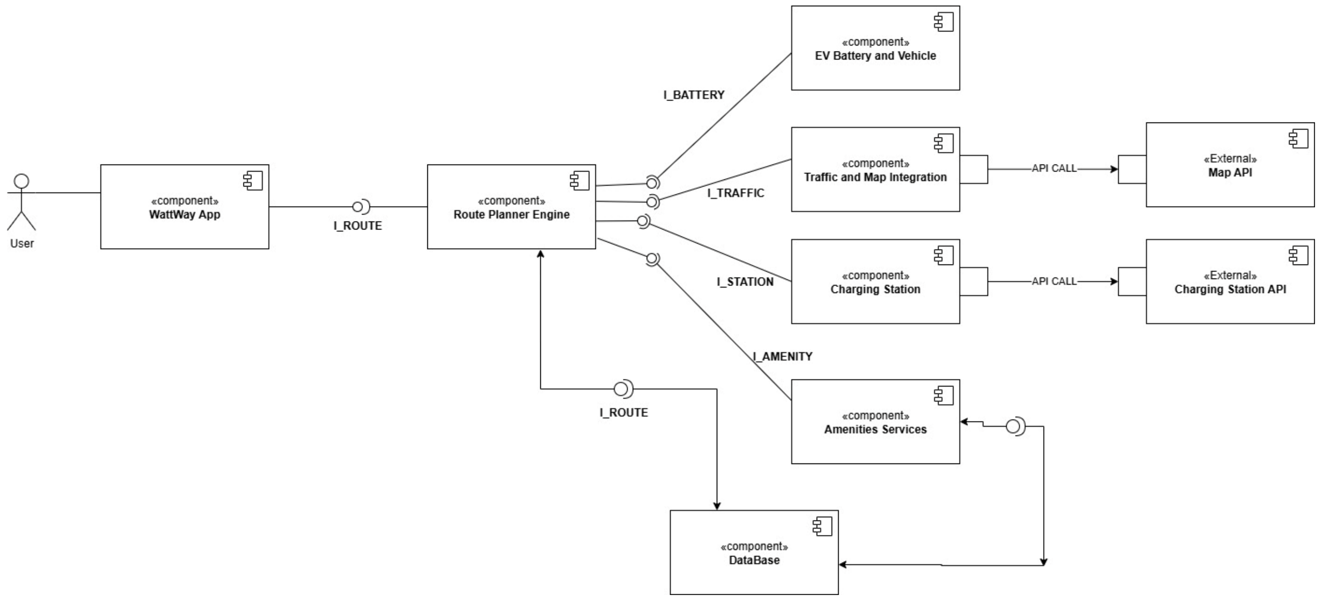


Figure 3-8: Component Diagram

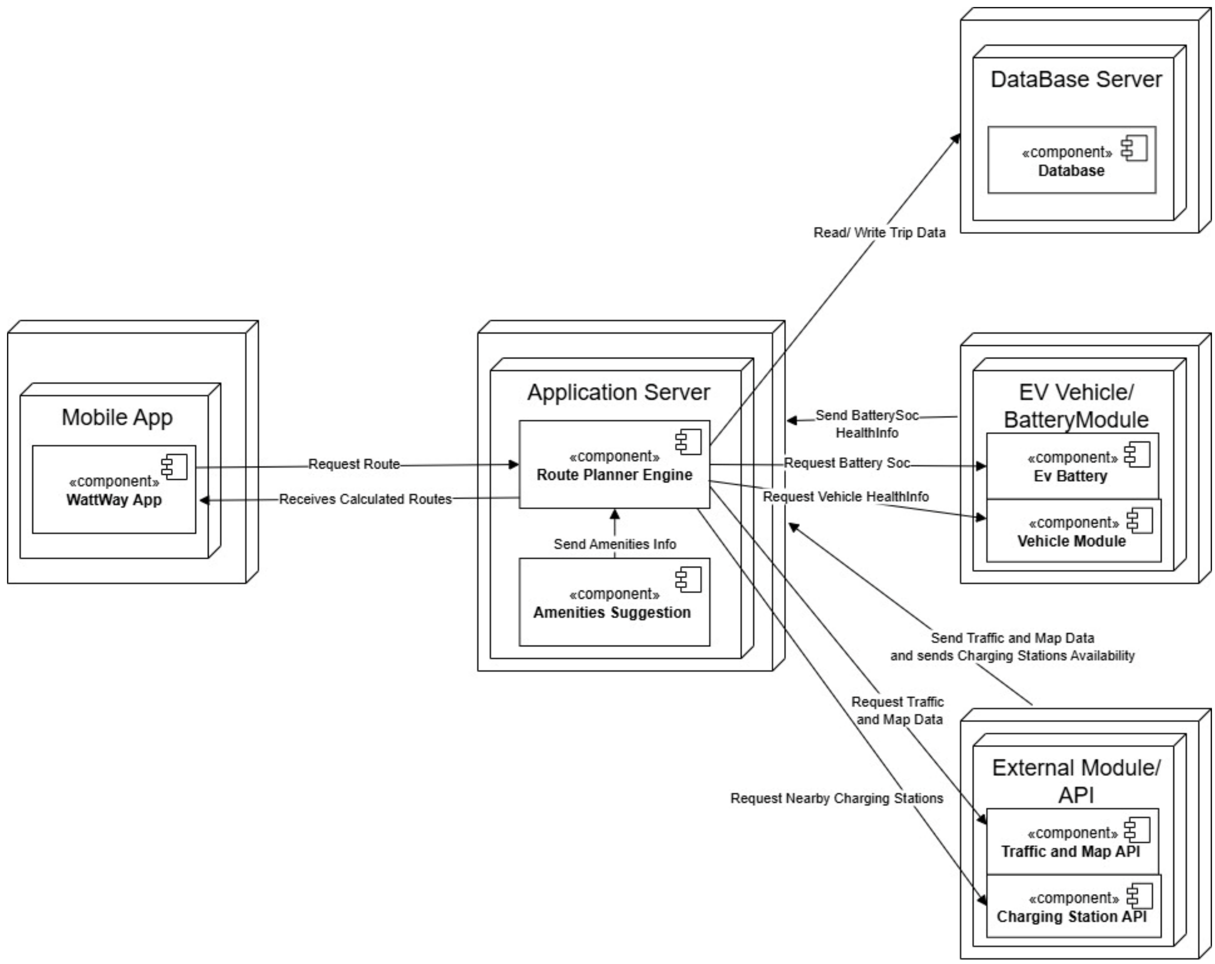


Figure 3-9: Deployment Diagram

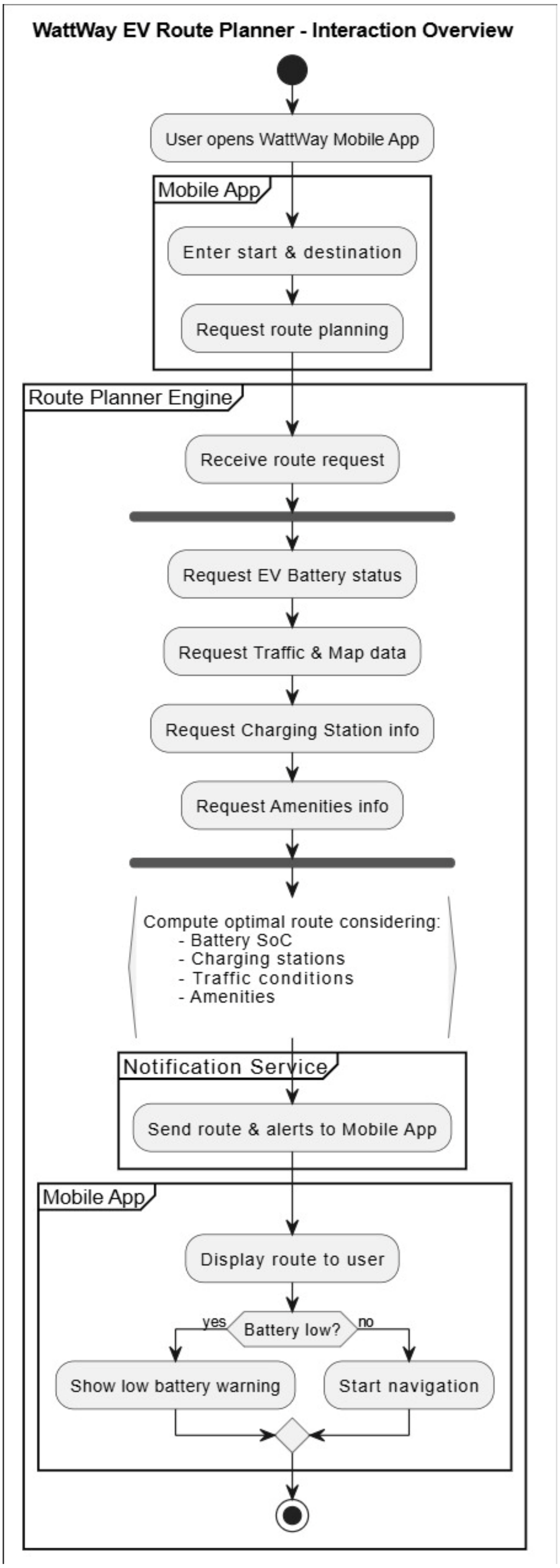


Figure 3-10: Interaction Overview Diagram

SYSTEM ARCHITECTURE

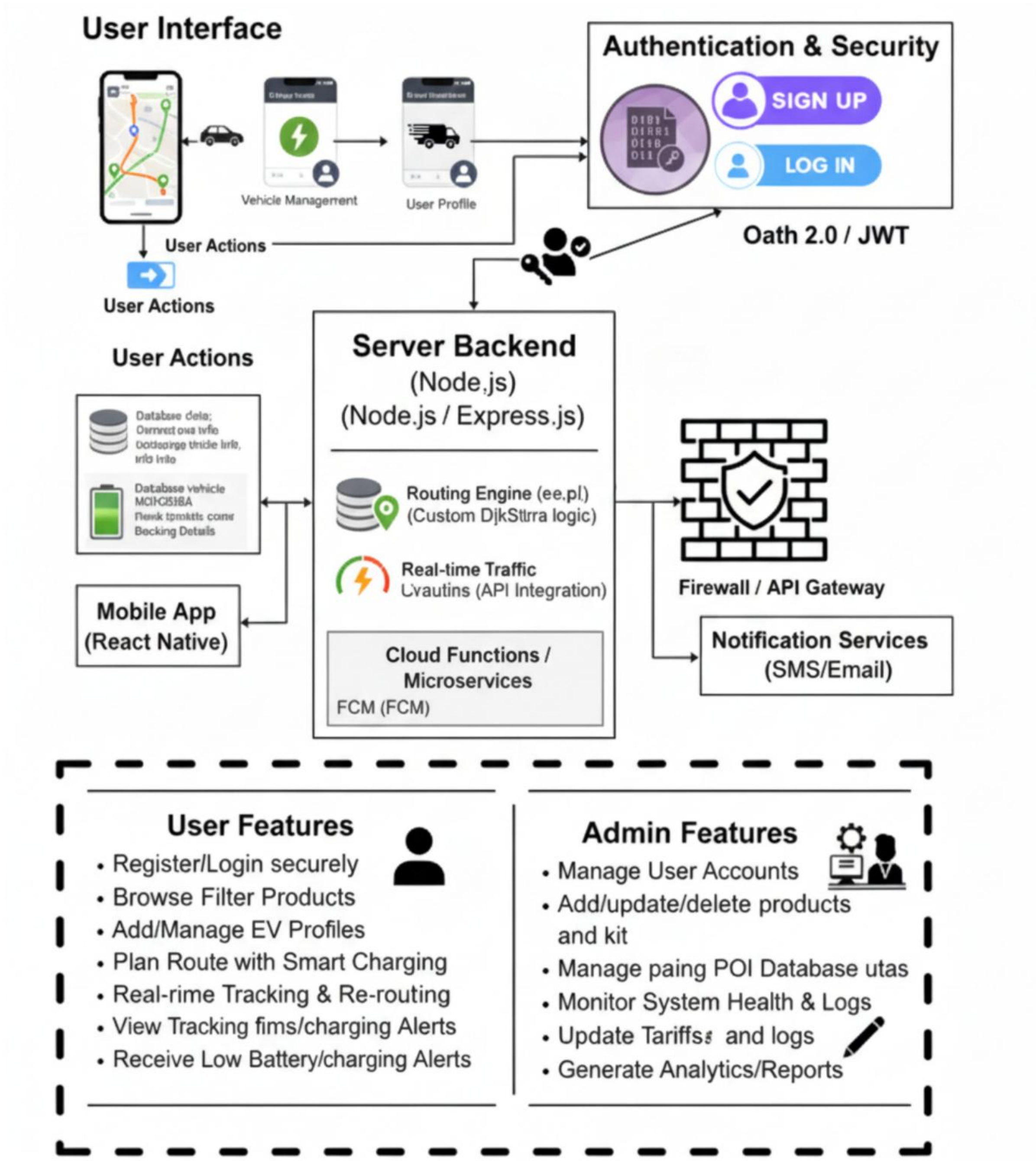


Figure 3-11: System Architecture

NAME	DATA TYPE	DESCRIPTION
session_id	varchar	Unique identifier for each charging session transaction
user_id	varchar	Unique identifier for registered EV users
station_id	varchar	Identifier for EV charging stations within the network
vehicle_id	varchar	Unique identifier for registered vehicles
session_timestamp	datetime	Records date and time when charging session was initiated
station_location	varchar	Geographical location/address of charging station
charging_status	enum	Current status of session (initiated, charging, completed, cancelled)
payment_status	enum	Status of payment (pending, completed, refunded, failed)
energy_consumed	float	Amount of electricity consumed during session (in kWh)
charging_cost	Float	Total cost calculated for the charging session
user_rating	float	Rating given by the user for the station/service (1–5 scale)

Table 1: Types of Data

3.5.1 Data Flow Diagrams

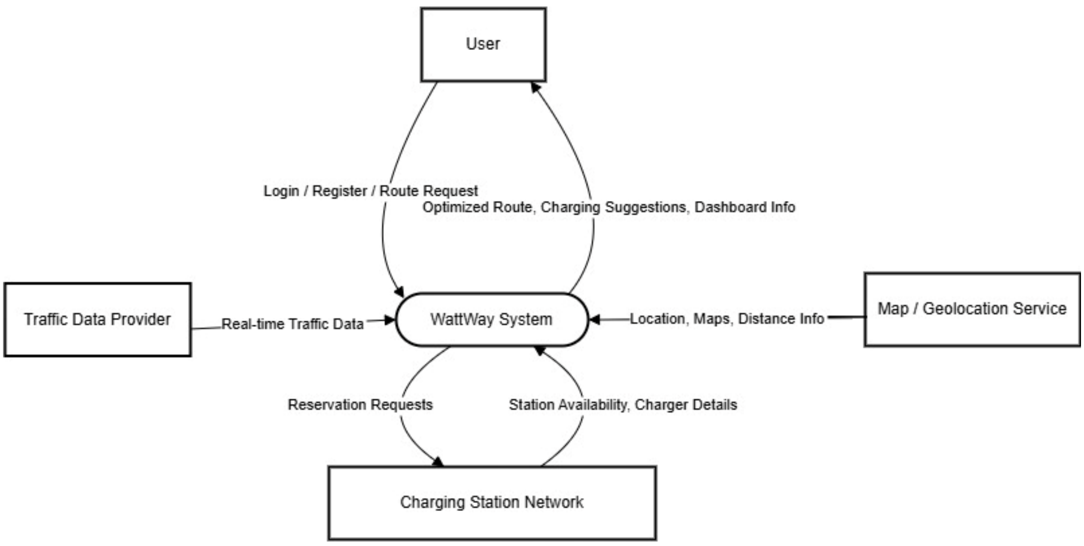


Figure 3-12: DFD Level 0

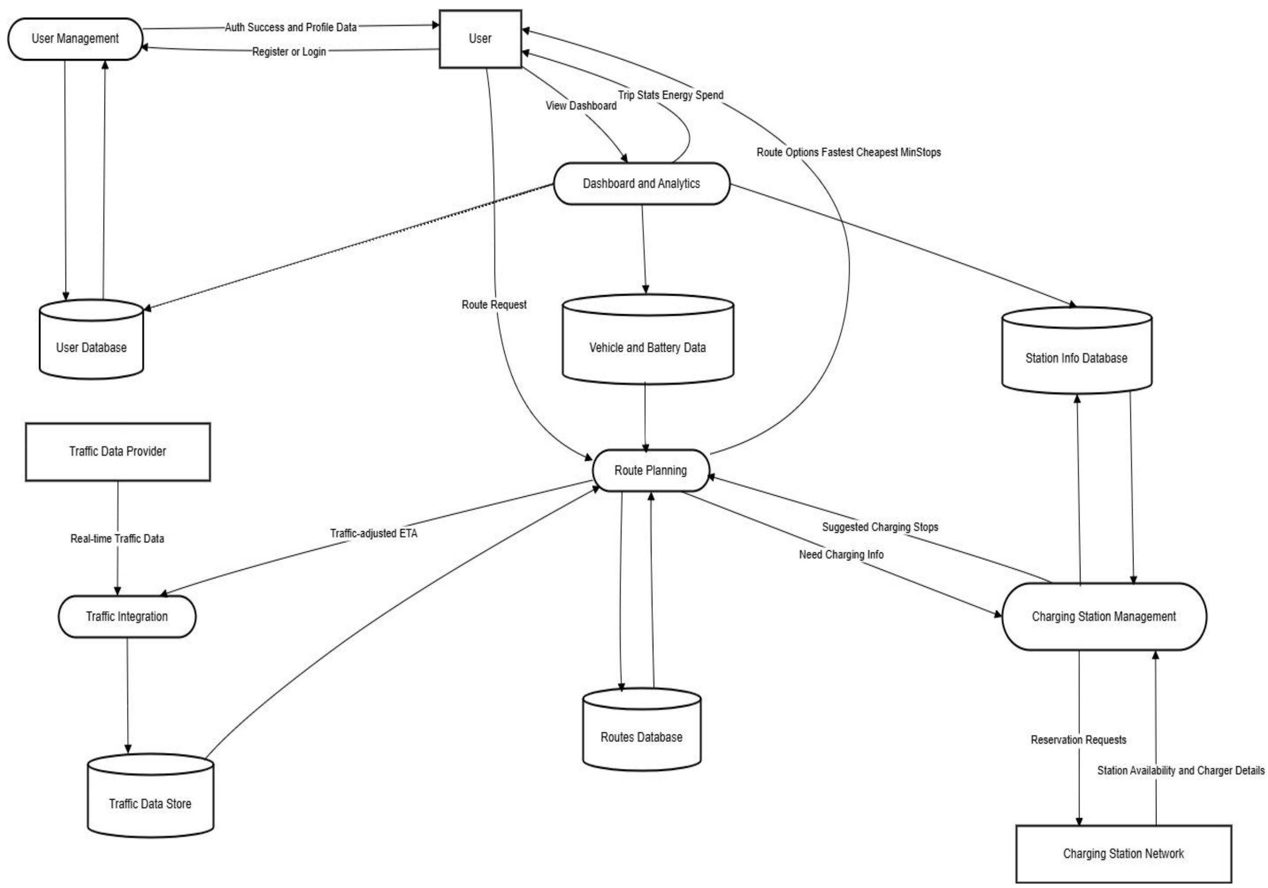


Figure 3-13: DFD Level 1

`

3.5.2 Database Structure

The **WattWay** project incorporates a structured database to manage EV user profiles, vehicle information, charging stations, and route data efficiently. The database schema ensures fast retrieval and reliable storage to support accurate route planning and real-time updates. Below are the details of the database structure:

1. Vehicle Data Schema

The Vehicle Data schema stores information about each registered EV, its battery specifications, and compatibility with charging infrastructure.

FIELD	TYPE	DESCRIPTION
vehicleId	String	Unique identifier for the EV.
ownerId	String	ID of the vehicle owner.
model	String	Vehicle model name.
batteryCapacity	Number	Maximum battery capacity in kWh.
currentSoC	Number	Current State of Charge (SoC) percentage.
chargerType	String	Supported charger types.
rangePerCharge	Number	Estimated distance on a full charge..
registrationNo	String	Vehicle registration number.
lastMaintenance	Date	Last service/maintenance date.

Table 2: Vehicle Data Schema

2. Charging Station Data Schema

The Charging Station schema manages all relevant information about available chargers and facilities.

FIELD	TYPE	DESCRIPTION
stationId	String	Unique identifier for the charging station.
name	String	Name or identifier of the station.
location	Object	Geolocation coordinates (latitude, longitude).
chargerTypes	Array	Types of chargers available at the station.
availableUnits	Number	Number of chargers currently operational.
powerRating	Number	Charger power in kW.
amenities	Array	Nearby amenities such as food, restrooms, ATMs.
operatingHours	Object	Opening and closing times.
ratings	Number	Average user rating of station (1–5). .

Table 3: Charging Station Data Schema

3. Route Data Schema

The Route Data schema stores computed routes, including energy feasibility, traffic, and charging stops.

FIELD	TYPE	DESCRIPTION
routeId	String	Unique identifier for the route.
userId	String	ID of the user requesting the route.
startLocation	Object	Starting geolocation coordinates.
endLocati on	Object	Destination geolocation coordinates.
routeOptions	Array	List of route options: fastest, cheapest, minimal charging stops.
chargingStops	Array	List of recommended charging stations along the route.

Table 4 : Route Data Schema

4. User Data Schema

FIELD	TYPE	DESCRIPTION
userId	String	Unique identifier for the user.
name	String	Full name of the user.
contactInfo	Object	Phone number, email, and preferred contact method.
registeredVehicles	Array	List of vehicle IDs owned by the user.
subscriptionPlan	String	Type of subscription or premium service plan.
preferences	Object	Preferred route options, charger preferences.
tripHistory	Array	Records of previous routes and charging stops.

Table 5: User Data Schema

Database Overview

The database is organized into following collections:

1. **User Collection**: Stores EV user profiles, preferences, and trip history.
2. **Vehicle Collection**: Maintains details of registered EVs, battery health, and charger compatibility.
3. **Charging Station Collection:** Contains all information about chargers, location, availability, and amenities.
4. **Route Collection:** Records computed routes, estimated energy consumption, and traffic conditions.
5. **Trip Collection**: Stores completed trips, charging stops, travel times, and user ratings for stations/routes.

6. **Subscription Collection**: Manages premium features, billing, and subscription plans.

7. **Traffic Data Collection**: Maintains real-time and historical traffic data for accurate route calculations.

8. **Feedback Collection**: Captures user feedback and ratings for stations, routes, and the app experience.

3.6 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below:

3.6.1 Hardware

1. Processor: Intel i5 or higher / AMD Ryzen 5 or higher.
2. RAM: Minimum 8GB (16GB recommended for smoother multitasking).
3. Storage: At least 256GB SSD for quick read/write operations.

3.6.2 Software

- **Backend:** NodeJs and Express Js.
- **Frontend:** React Native for an intuitive user interface.
- **Database Management:** MongoDB for efficient data storage and retrieval.
- **Integration Tools:** APIs for real-time traffic data, EV charging station availability, and route optimization.
- **Testing Tool:** Postman for API testing.

`

Chapter 4

Implementation

The implementation of the **WattWay** project involves a comprehensive process of integrating EV-specific route optimization, charging station coordination, and user experience design to create a reliable platform for EV users in India. This chapter outlines the techniques, tools, programming languages, and testing methodologies employed to build a system that ensures energy-efficient, safe, and stress-free EV journeys.

4.1 Technique Used

4.1.1 Hybrid EV Route Planning Model

The Hybrid EV Route Planning Model allows WattWay to manage two interconnected but independently optimized workflows—route computation and charging station coordination. This model separates the calculation of energy-efficient routes from the management of charger availability, while ensuring synchronization by recommending charging stops that match the vehicle's battery capacity, SoC, and charger compatibility. In essence, the system ensures that EV users reach their destinations punctually and without unexpected delays, balancing both route efficiency and energy feasibility. This hybrid approach reduces dependency on any single factor (like traffic or charger availability) while maintaining overall trip reliability.

4.1.2 Real-Time Route and Energy Optimization

To provide stress-free and energy-efficient EV journeys, WattWay incorporates several smart optimization strategies. These include dynamic calculation of most efficient route based on real-time traffic data. Additionally, the system leverages predictive energy modeling by analyzing battery health, vehicle type, and driving patterns to anticipate energy requirements and recommend timely charging stops. For the journey itself, adaptive route recalculation ensures that the driver always follows the most efficient path in response to traffic changes or charger availability. This intelligent routing framework minimizes range anxiety and ensures reliable, optimized EV travel.

4.2 Tools Used

The **WattWay** system relies on a carefully selected stack of tools and frameworks that enable seamless integration between the frontend, backend, database, and API layers. These tools ensure scalability, real-time responsiveness, and accurate EV-specific route planning for users.

4.2.1 Tailwind CSS

` Tailwind CSS is a **modern utility-first CSS framework** that allows developers to style their applications directly within the markup using pre-defined utility classes. In the **KaaryaKram** project, Tailwind CSS was used extensively to build a **clean, responsive, and highly customizable UI** with minimal CSS bloat. It enabled faster prototyping and consistent design, helping maintain visual harmony across components like booking cards, service listings, and forms. Its mobile-first approach also ensured that the platform looks great on both desktop and mobile devices.

4.3 Language Used

The **WattWay** system utilizes a multi-language stack to ensure robust performance, flexibility, and scalability. Each language plays a specific role in delivering an energy-optimized route planning system for EVs.

1. MongoDB

MongoDB is a flexible, document-oriented **NoSQL database** used to store and manage data in a JSON-like format. In WattWay, MongoDB is used to manage user profiles, vehicle details, charging station information, and trip histories. Its schema-less nature allows rapid feature additions, while its scalability ensures reliable performance even with large volumes of route and traffic data.

2. Express.js

Express.js is a **lightweight** and **fast Node.js web application framework** that simplifies server-side development. In WattWay, Express.js handles API development, middleware integration, and server-side routing. It powers critical operations such as user authentication, route calculations, charging station lookups, and third-party API integrations (maps, payment gateways, charging station databases).

3. React Native

React Native is a popular framework for cross-platform mobile app development. WattWay uses React Native to build a single codebase that runs seamlessly on both Android and iOS. Its component-based architecture enables reusable UI components such as route cards, station details, and navigation menus, ensuring a smooth and responsive experience for EV users.

4. Node.js

Node.js is a high-performance, **event-driven JavaScript runtime environment** that enables the execution of JavaScript on the server side. WattWay uses Node.js to handle asynchronous requests such as real-time traffic updates, dynamic route recalculations, and charger availability checks. Its scalability and non-blocking I/O operations make it well-suited for delivering a real-time EV route planning service.

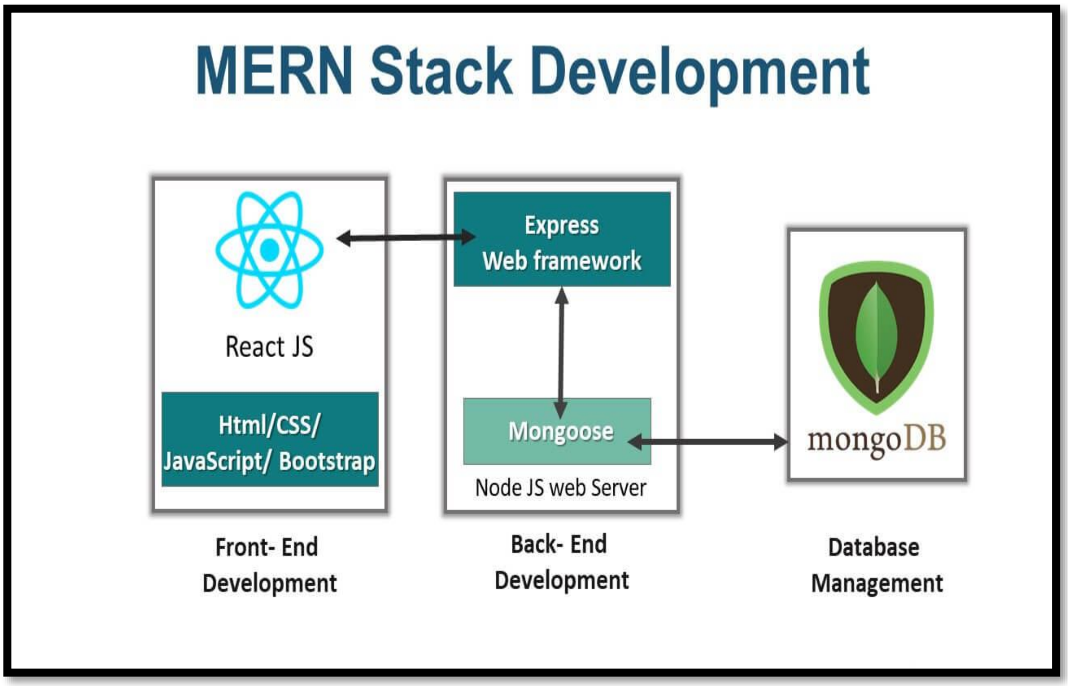


Figure 4-1: Working of MERN Stack

4.4 Screenshots

The Following are the screenshots of the result of the project :

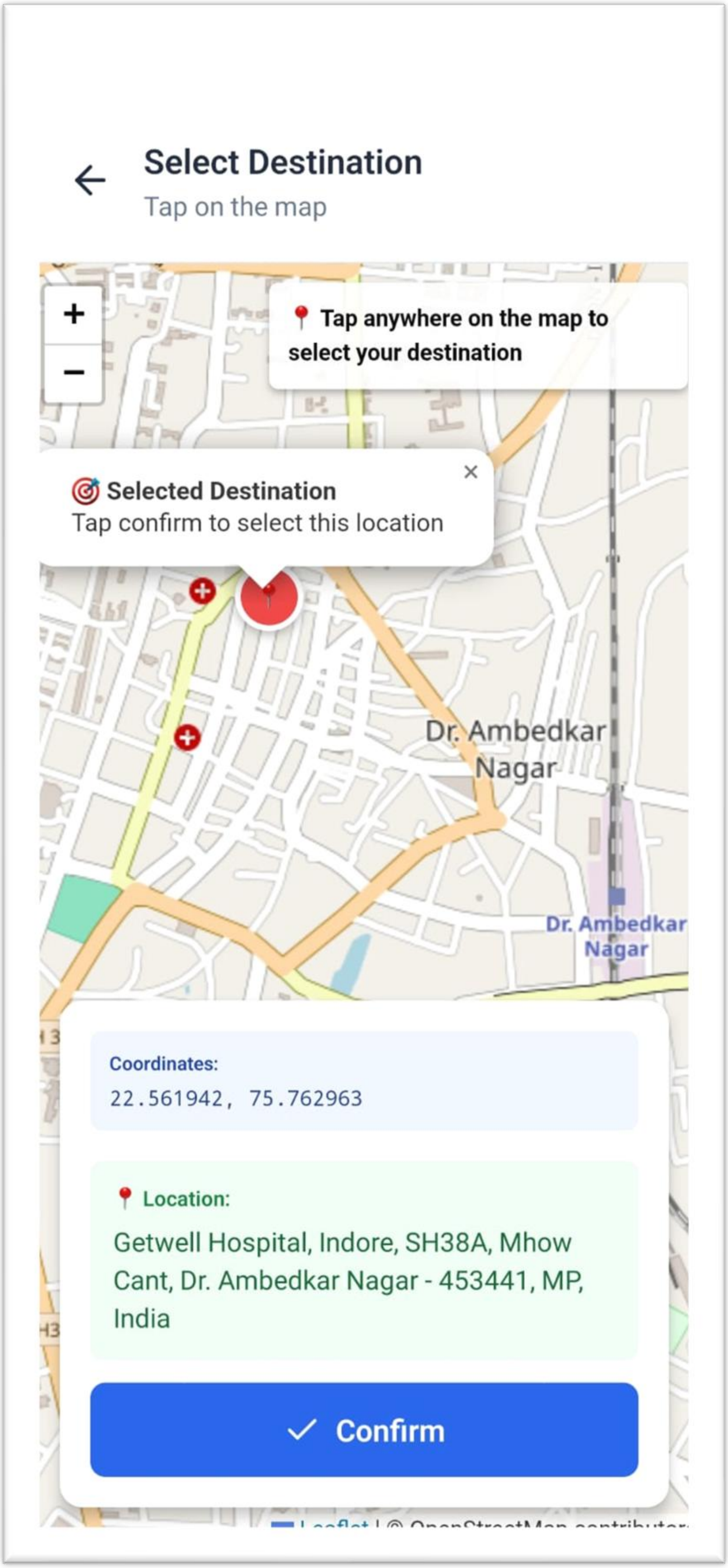


Figure 4-2: Screenshot 1-Map View

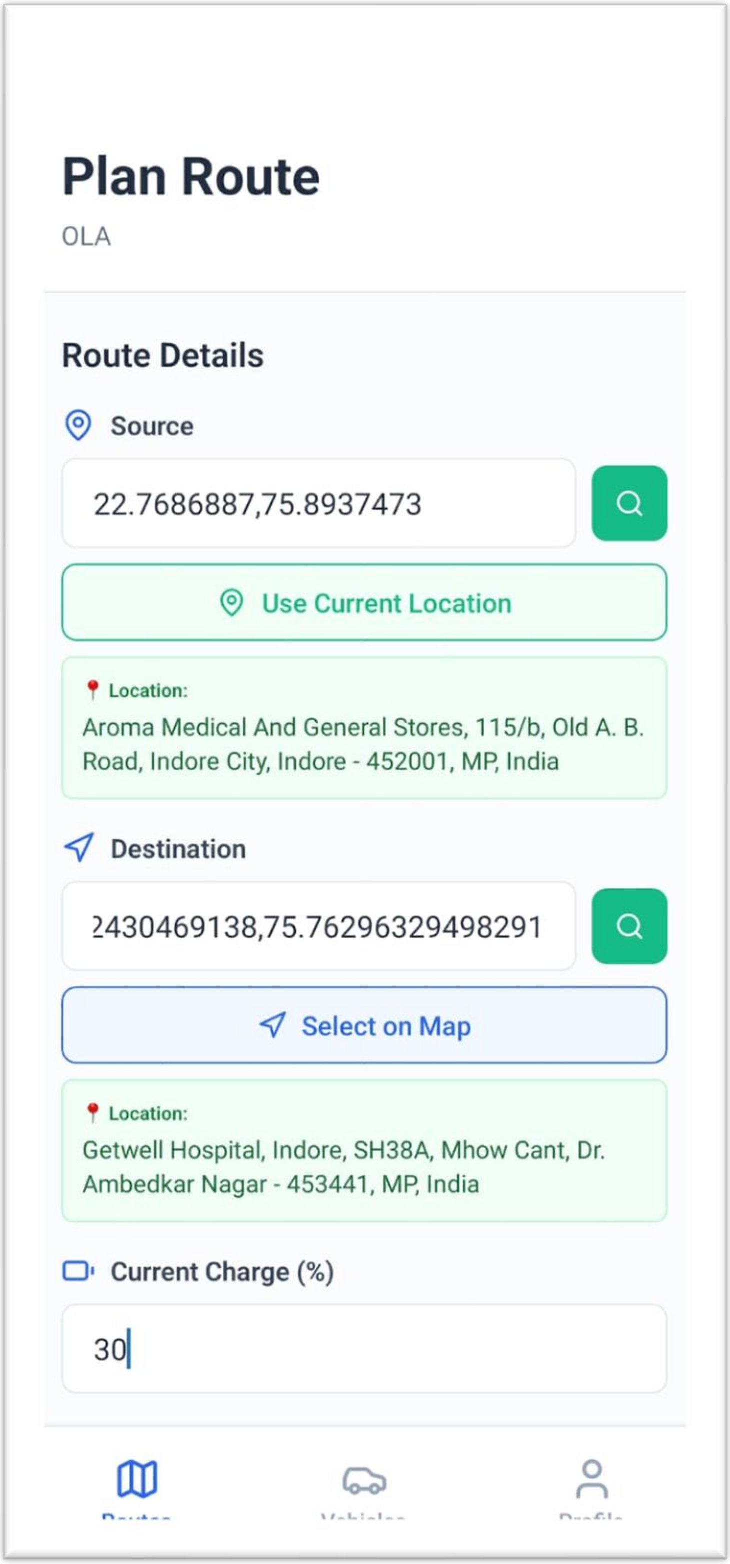


Figure 4-3: Screenshot 2 – Plan Route

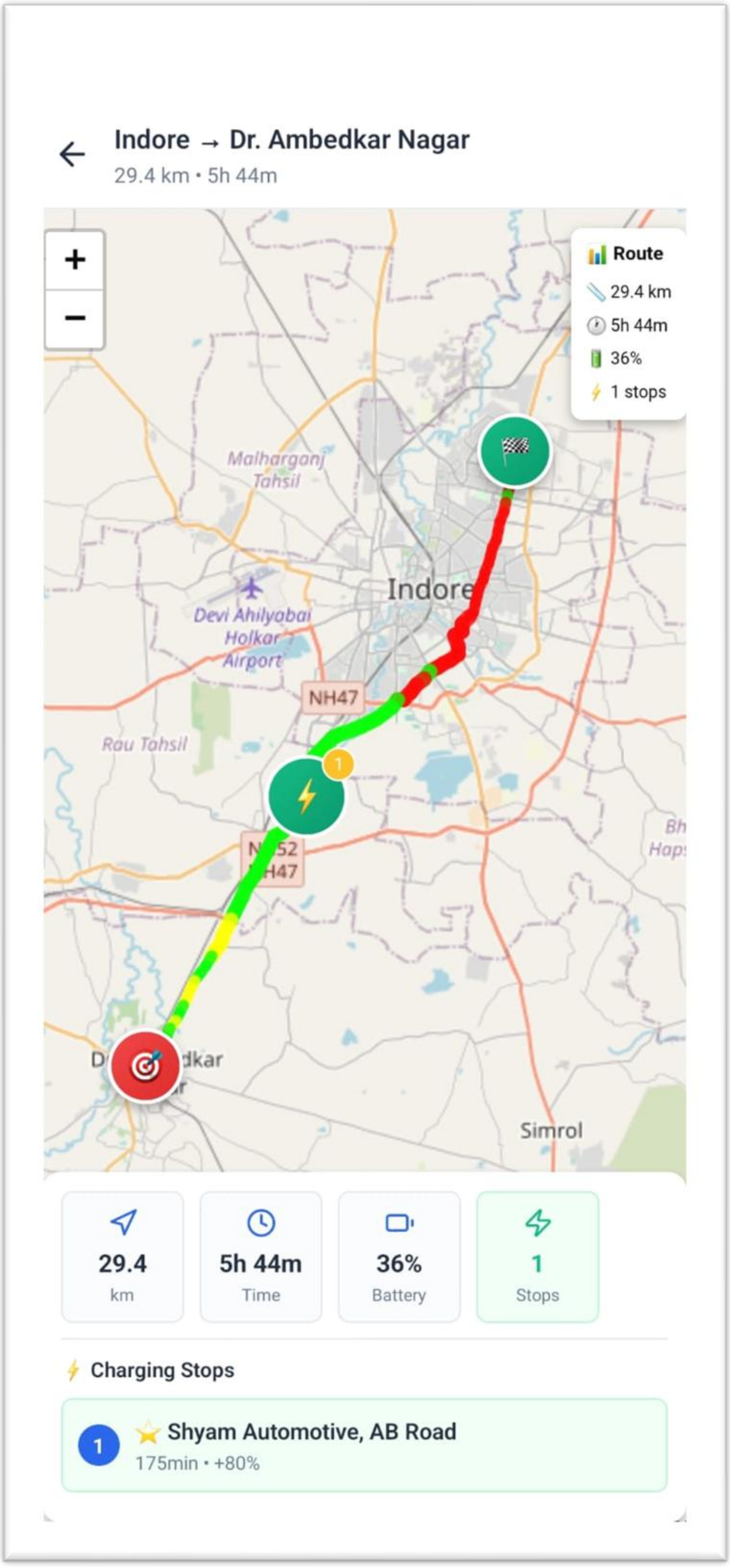


Figure 4-4: Screenshot 3- Optimized Route and Charging Stop Suggestion

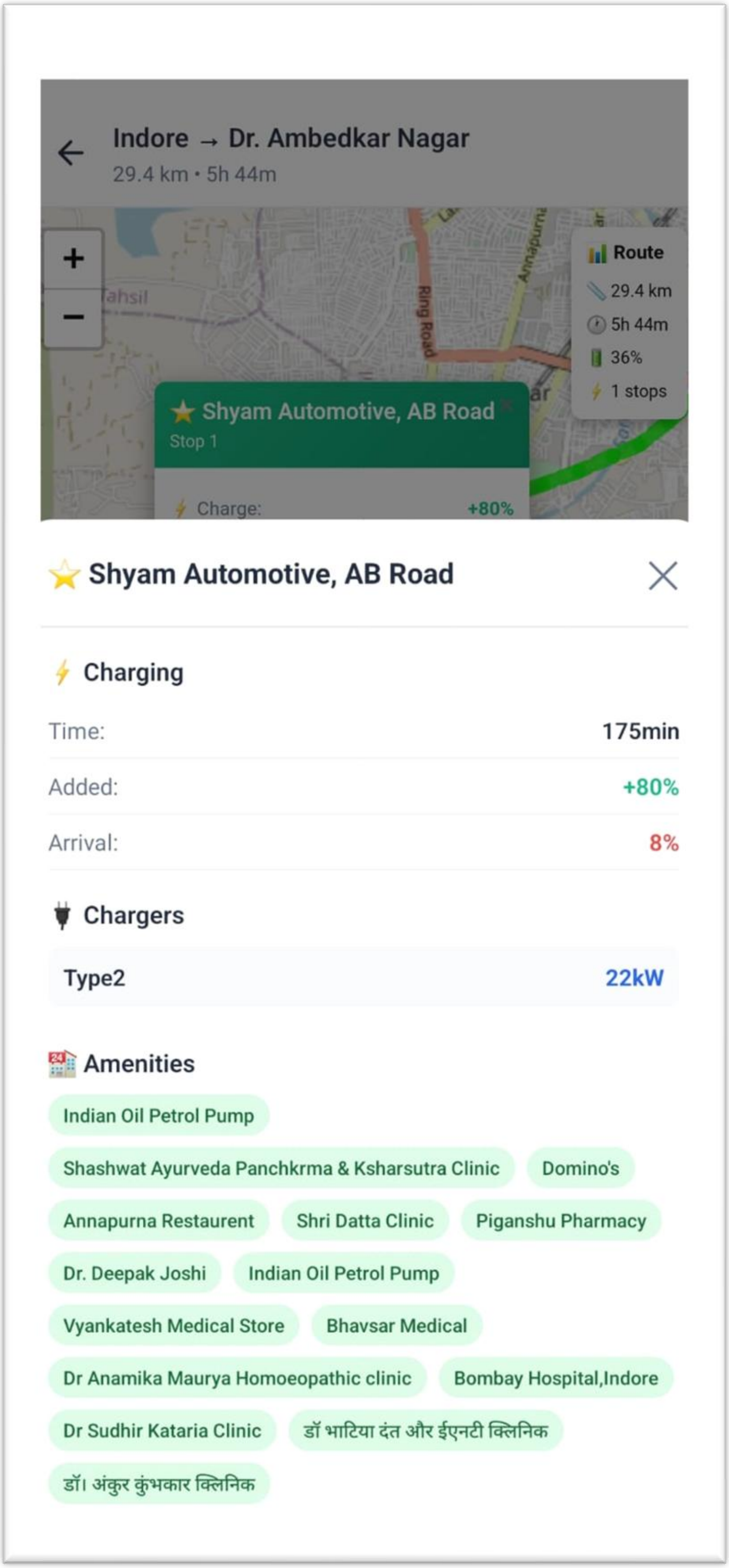


Figure 4-5: Screenshot 4 – Amenities Suggestion

4.5 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process. .

4.5.1 Strategy Used

The testing strategy ensures the system's robustness and reliability:

1. **Unit Testing:** Verifies the functionality of individual components like algorithms, API routes, and database queries.
2. **Integration Testing:** Ensures smooth communication between the frontend, backend, and database.
3. **System Testing:** Validates the entire system under real-world conditions, including large datasets and multiple users.
4. **Load Testing:** Tests the system's performance under high concurrent API requests to ensure scalability.

4.5.2 Test Case and Analysis

Test Case 1:

TestCase ID	**TC001**
Test Case	Verify nearby amenities suggestion from charging stop.
Summary	Ensure that nearby amenities are correctly displayed for the selected charging stop.
Test Steps	1. Select a route with charging stops. 2. Tap on a charging stop (e.g., Shyam Automotive, AB Road). 3. Scroll to the “Amenities” section.
Expected Result	Relevant nearby amenities are listed under the “Amenities” section.
Actual Result	Nearby amenities such as Petrol Pump, Restaurant, and Hospitals are displayed.

Table 5: Test Case 1 – Amenities Suggestion

Test Case 1 Output:

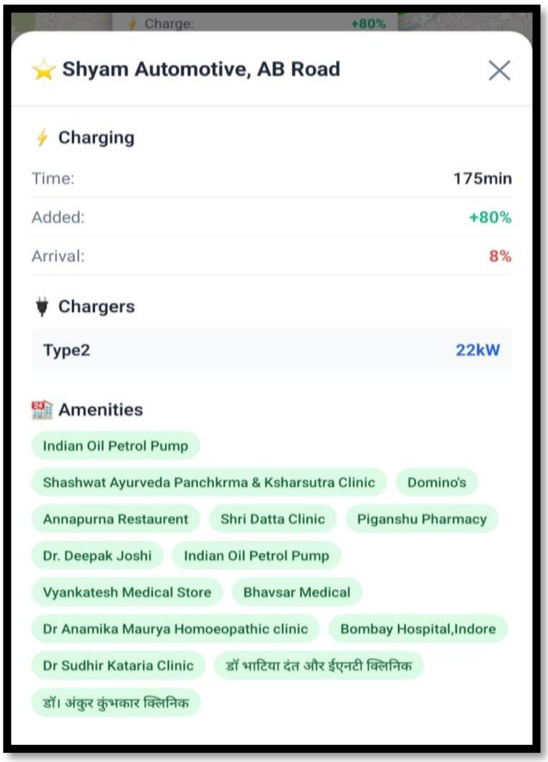


Figure 4-6: Test Case 1 Output

Test Case 2:

TestCase ID	TC002
Test Case	Verify charging stop details are displayed correctly.
Summary	Check if charging stop info (charge, time, arrival, detour, availability) shows correctly.
Test Steps	1. Select a route with a charging stop. 2. Tap the stop to view details.
Procedure	Open the stop details and verify the displayed data.
Expected Result	All charging stop details should be correctly displayed and match backend data.
Actual Result	Charging stop details displayed correctly — Charge +80%, Time 175min, Arrival 8%, Detour 0.1km, Available.
Status	Pass

Table 6: Test Case 2 – Charging Station Details

Test Case 2 Output:

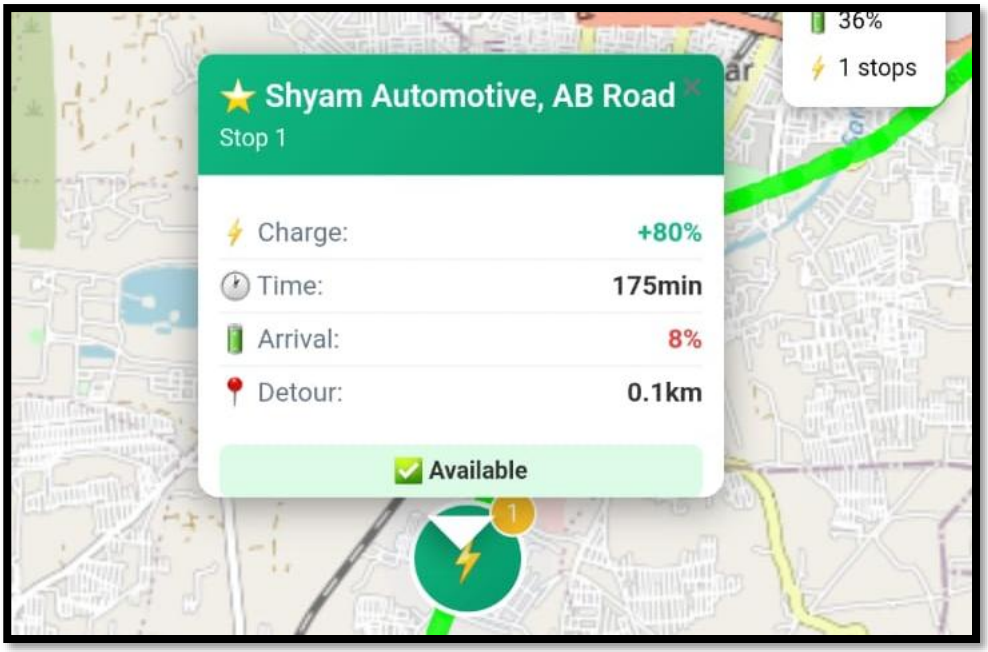


Figure 4-7: Test Case 2 Output

Test Case 3:

TestCase ID	TC003
Test Case	Verify addition of a new vehicle to "My Vehicles".
Summary	Check if a new vehicle is added successfully.
Test Steps	1. Click "+" on My Vehicles screen. 2. Enter valid details (e.g., OLA S1, 100 kWh, 0.7 kWh/km, 80 km). 3. Click **Save/Add**.
Procedure	Add vehicle via UI and check network tab for response.
Expected Result	200 OK response ; vehicle appears as **Active** in list.
Actual Result	200 OK received; OLA S1 added successfully.
Status	Pass

Table 7: Test Case 3 - Adding vehicles information

Test Case 3 Output:

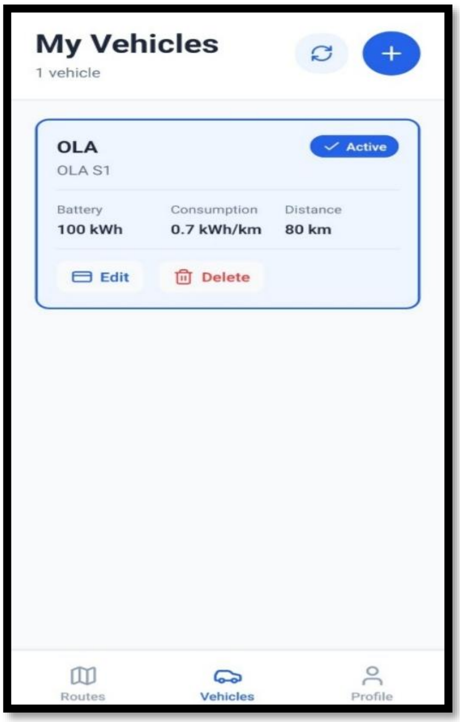


Figure 4-8: Test Case 3 Output

Test Case 4:

TestCase ID	TC004
Test Case	Verify route and charging stop details.
Summary	Ensure the system shows correct route, distance, time, SoC, and charging stop info.
Test Steps	1. Enter source: *Indore*, destination: *Dr. Ambedkar Nagar*. 2. Tap **Find Route** and view details.
Expected Result	Route (~29.4 km, 5h 44m), SoC 36%, and charging stop "Shyam Automotive (+80%)" shown correctly.
Actual Result	All route and charging details displayed accurately.
Status	Pass

Table 8: Test Case 4 – EV route planner

Test Case 4 Output:

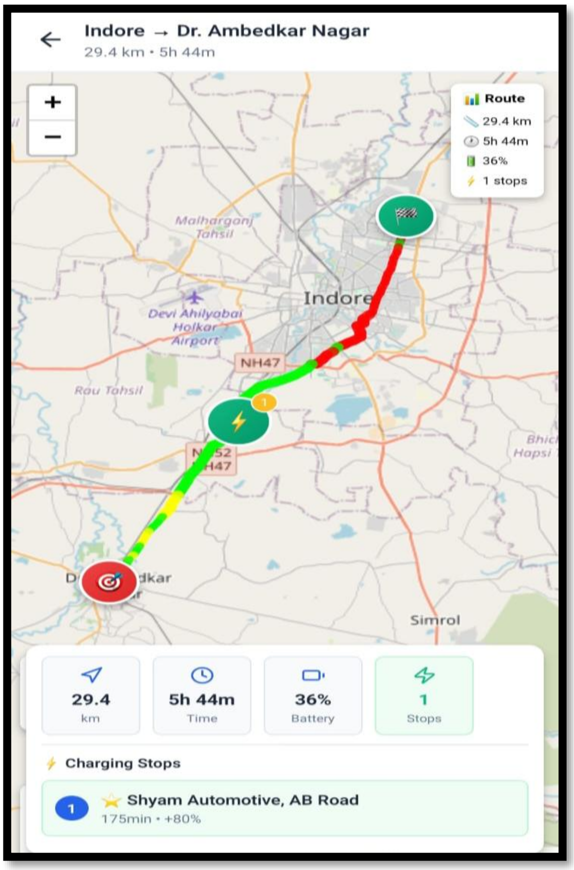


Figure 4-9: Test Case 4 Output

`

Chapter 5

Conclusion

5.1 Conclusion

The **WattWay project** presents a smart, EV-centric solution that bridges the gap between traditional navigation systems and the specific needs of electric vehicle users in India. By integrating modern technologies such as React.js, Node.js, MongoDB, and real-time traffic APIs, the platform ensures energy-feasible route planning alongside intelligent charging station recommendations. WattWay addresses a critical challenge — minimizing range anxiety while optimizing EV journeys — and successfully automates this process to meet both logistical and user experience expectations.

Through its energy-aware routing model, WattWay efficiently coordinates between vehicle State of Charge (SoC), battery health, charger compatibility, and dynamic traffic conditions. Real-time features like charger occupancy prediction, adaptive route recalculations, and nearby amenity suggestions significantly enhance reliability, efficiency, and comfort.

By unifying EV navigation with smart energy management, **WattWay** offers a contextually relevant, technologically advanced solution. It not only simplifies long-distance and intra-city EV travel but also redefines convenience, safety, and confidence for EV users. This project demonstrates how modern web technologies and thoughtful system design can come together to solve real-world mobility challenges with innovation, sustainability, and user-centric efficiency.

5.2 Limitations of the Work

- **Regional Data Constraints:**

  WattWay currently relies on charging infrastructure data that is limited in coverage and accuracy across India. In smaller towns and rural areas, incomplete or outdated data about charging stations reduces the reliability of route planning, impacting the system's nationwide usability.

- **Charger Availability & Occupancy:**
  While WattWay suggests compatible charging stations, real-time data on charger occupancy and downtime is not always available or standardized. This can lead to unexpected waiting times or unavailability at charging stations, affecting trip reliability.

- **Battery Model Variability:**
  Different EV manufacturers follow varying standards for SoC reporting, battery health estimation, and energy consumption patterns. The lack of uniformity makes it challenging to ensure consistent accuracy across all vehicle models, especially for older or less-documented EVs.

These limitations highlight areas where future improvements could be made to enhance the flexibility, accuracy, and efficiency of the **WattWay** system.

5.3 Suggestion and Recommendations for Future Work

To improve and extend the functionality of the **WattWay** system, the following suggestions and recommendations for future work are proposed:

1. **Expand Charging Network Coverage:**
   Extend the platform's database to include more cities, highways, and rural areas, ensuring comprehensive visibility of charging stations across India.

2. **Partnerships with Charging Providers:**
   Collaborate with EV charging networks to access real-time data on charger availability, occupancy, and pricing, improving route accuracy and reliability.

3. **AI-Powered Battery and Route Optimization:**
   Incorporate advanced predictive models to factor in battery degradation, terrain, and driving behavior, enabling more precise and personalized route planning.

4. **Multilingual and Voice Support:**
   Introduce multilingual interfaces and voice-assisted navigation to enhance accessibility for diverse users across India.

Bibliography

[1] M. Seetharaman *et al.*, *India Electric Vehicle Report 2023*, Bain & Company, 2023.

[2] IEA, *Global EV Outlook 2025*, International Energy Agency, 2025.

[3] C. Vogt and K. Klausarson, "eDijkstra: Time-Optimal Route Planning for Electric Vehicles," *Proc. ETFA 2020*, 2020.

[4] Y. Cheng *et al.*, "Predictive Routing For EV Charging Stations," *Int. J. Smart Elect. Mobil.*, vol. 10, no. 1, 2022.

[5] Mapbox, "Mapbox for EV: Seamless EV Navigation," Mapbox Blog, 2024.

[6] MapmyIndia, "Electric and Efficient Mobility | Connected Vehicle," Mappls (MapmyIndia) website, 2024.

[7] Tata Power, "Fast EV Charging Stations in India – Tata Power EZ Charge," 2025.

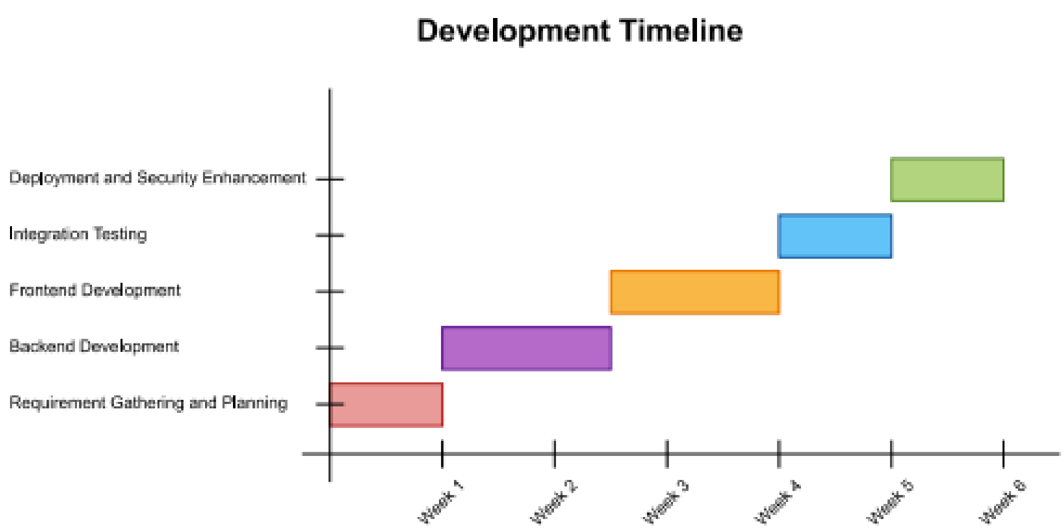
[8] Ather Energy, "Ather Grid – Fast Charging Network for Electric Scooters," Ather Energy Official Website, 2024.

[9] PlugShare, "PlugShare – EV Charging Station Map," PlugShare Official Platform, 2024.

[10] Google, "EV Charging Stations on Google Maps," Google Maps Help Center, 2024.

Project Plan

<u>Gantt Chart</u>



Guide Interaction Sheet

Sr. No.	Meeting Date	Summary of Work & Discussion	Member Present	Remarks/ Suggestions Given
1	01-08-2025	Idea discussion with Faculty mentor and finalizing the functionalities of the project along with the title	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Comparison with similar platforms taking into account their limitations
2	12-08-2025	Discussion on synopsis and various functionalities that can be added in the mobile app.	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Synopsis was finalized
3	21-08-2025	Presentation of the Idea with the introduction, algorithm, impacts, limitations and future enhancements	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Technology stack was discussed.
4	22-24-2025	Completed remaining part of synopsis and worked on technical poster	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	More graphics need to be added in the technical poster and remove limitations from it.
5	29-08-2025	Finalized the technical poster and started working on research paper.	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Changes were suggested in the formatting of the research paper. And more figures needed to be added.

Sr. No.	Meeting Date	Summary of Work & Discussion	Member Present	Remarks/ Suggestions Given
6	09-09-2025	Research paper got finalized and then we worked on uml diagrams.	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Publish the research paper before your internal exam.
7	19-09-2025	Nitya and Naman worked on the UML diagrams, Mayank and Parth worked on the project implementation.	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Changes were suggested in the uml diagrams.
8	01-10-2025	Naman worked on the changes suggested in the UML diagram, nitya worked on the project report and Mayank and Parth worked on project implementation and deployment.	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Further changes were suggested in UML diagrams.
9	08-10-2025	Project poster, report, ppt, video, research paper and final implementation shown	Mayank Piparde, Parth Gupta, Nitya Maheshwari & Naman Shah	Submission Done

`

Source Code

1. Map View

```
"use client"

import { View, Text, StyleSheet, TouchableOpacity, ScrollView, Modal, ActivityIndicator,
Dimensions, Platform } from 'react-native';
import { useState, useEffect, useRef } from 'react';
import { useRouter, useLocalSearchParams } from 'expo-router';
import { WebView } from 'react-native-webview';
import {
  ArrowLeft, Clock, Battery, MapPin, Zap, AlertTriangle, Info, CheckCircle, XCircle,
  Coffee, Utensils, Wifi, ShoppingCart, Home, Fuel, WashingMachine, ParkingCircle,
  Hotel, Heart, DollarSign, Filter, Navigation, X
} from 'lucide-react-native';
import { Amenity, ChargingStation, Output } from "@/utils/mapTypes";
import { routeApi } from '@/api/routeApi';

const { width: SCREEN_WIDTH } = Dimensions.get('window');

export default function MapViewScreen() {
  const router = useRouter();
  const params = useLocalSearchParams<{ routeData: string; source: string; destination: string
}>();
  const webViewRef = useRef<WebView>(null);

  const [routeData, setRouteData] = useState<Output | null>(null);
  const [mapHtml, setMapHtml] = useState('');
  const [selectedStation, setSelectedStation] = useState<ChargingStation | null>(null);
  const [showStationModal, setShowStationModal] = useState(false);
  const [showBatteryModal, setShowBatteryModal] = useState(false);
  const [showFilterModal, setShowFilterModal] = useState(false);
  const [sourceAddress, setSourceAddress] = useState('');
  const [destAddress, setDestAddress] = useState('');
  const [loading, setLoading] = useState(true);

  const [maxDistance, setMaxDistance] = useState(1000);
  const [selectedCategories, setSelectedCategories] = useState<string[]>([]);
  const [amenityRoute, setAmenityRoute] = useState<any>(null);
  const [isLoadingAmenityRoute, setIsLoadingAmenityRoute] = useState(false);
  const [selectedAmenity, setSelectedAmenity] = useState<Amenity | null>(null);

  const amenityCategories = ['food', 'washroom', 'medical', 'hotel', 'wifi', 'parking', 'cafe',
'restaurant', 'fuel', 'atm'];

  useEffect(() => {
    if (params.routeData) {
```

```
    try {
     const data: Output = JSON.parse(params.routeData);
     setRouteData(data);
     generateEnhancedMapHtml(data);

     if (data.routeCoordinates && data.routeCoordinates.length > 0) {
      const start = data.routeCoordinates[0];
      const end = data.routeCoordinates[data.routeCoordinates.length - 1];
      fetchAddressNames(start.lat, start.lng, end.lat, end.lng);
     }

     setLoading(false);
    } catch (error) {
     console.error('Failed to parse route data:', error);
     setLoading(false);
    }
   }
  }, [params.routeData]);

  const fetchAddressNames = async (startLat: number, startLng: number, endLat: number,
endLng: number) => {
   try {
    const [startRes, endRes] = await Promise.all([

fetch(`https://api.geoapify.com/v1/geocode/reverse?lat=${startLat}&lon=${startLng}&apiKe
y=5ffe1f1598ac467dafc8789f5e787a3e`),

fetch(`https://api.geoapify.com/v1/geocode/reverse?lat=${endLat}&lon=${endLng}&apiKey
=5ffe1f1598ac467dafc8789f5e787a3e`)
    ]);

    const [startData, endData] = await Promise.all([startRes.json(), endRes.json()]);

    if (startData.features?.[0]) {
     const props = startData.features[0].properties;
     setSourceAddress(props.city || props.county || props.formatted || 'Start');
    }

    if (endData.features?.[0]) {
     const props = endData.features[0].properties;
     setDestAddress(props.city || props.county || props.formatted || 'Destination');
    }
   } catch (error) {
    console.error('Failed to fetch address names:', error);
   }
  };

  const generateEnhancedMapHtml = (data: Output) => {
   const coordinates = data.routeCoordinates || [];
   if (coordinates.length === 0) return;

   const centerLat = coordinates[Math.floor(coordinates.length / 2)].lat;
   const centerLng = coordinates[Math.floor(coordinates.length / 2)].lng;
```

```
const routeSegments = coordinates.map((coord, index) => {
  const nextCoord = coordinates[index + 1];
  if (!nextCoord) return '';

  const color = coord.trafficColor || '#10b981';
  const batteryPercent = coord.batteryLevelPercent || 100;
  const isCritical = batteryPercent < 20;
  const isCharging = coord.isChargingStop;

  const weight = isCharging ? 12 : (isCritical ? 8 : 5);
  const dashArray = isCritical ? '10, 5' : 'none';
```

GitHub Repo Link:

https://github.com/ParthGupta84616/Route-Wise