ChargeSync: An Intelligent EV Charging Ecosystem with Battery-Aware Recommendation, Smart Reservation and Charging Management

Mayuresh Patil[1], Pranav Mali[2], Rauhan Patel[3], Ramesh Malve[4]
[1,2,3,4]*Department of Computer Science and Engineering, Sharad Institute of Technology College of Engineering, Yadrav, Ichalkaranji, India*

***Abstract*—Finding a place to charge an electric vehicle (EV) is more than a map lookup. A driver with a low battery has to think about whether a station can be reached, whether its connector fits the car, whether a charger is free, how long the queue will be and what the session will cost. This paper presents ChargeSync, an EV charging platform that brings these questions together for three groups of users: EV drivers, station owners and administrators. Drivers use a React Native mobile application, while owners and administrators work through web portals that share a Node.js service layer and a common database. At the core of the platform is a battery-aware recommendation method that screens stations for reachability and connector compatibility and then ranks them with a weighted score built from distance, charging speed, price, expected waiting time, rating and station health. The platform also covers slot reservation with no-show handling, an emergency mode for critically low batteries, owner analytics and a retrieval-augmented review summarizer. We describe the architecture, the development process, the scoring models and the booking workflow, and we work through an example of the reachability analysis. Quantitative performance and user-study results are still to be measured, so we set out the evaluation plan and the limitations of the current design.**

***Index Terms*—Battery-aware recommendation, Electric vehicles, EV charging, Predictive analytics, Retrieval-augmented generation, Smart reservation.**

I. INTRODUCTION

Electric vehicles are moving from early adoption into everyday use, and with that shift the quality of public charging has become a practical concern rather than a side issue. Drivers of battery electric vehicles still worry about running out of charge before they reach a charger, and a second worry has joined the first: the time spent waiting for a charger and then using it. Wang et al. surveyed 217 battery electric vehicle users in mainland China and found that this time-related anxiety is real, and that drivers trade it off against distance-related anxiety when they decide where to charge [3]. In other words, the closest station is often not the best one.

Mobile locator apps have eased part of the problem by letting drivers search for stations, pay and leave reviews. The experience is still far from smooth. Working with user reviews from a popular locator app, Asensio et al. reported that nearly half of the drivers who use mobility applications had faced negative experiences at public charging stations [7]. Anyone who has arrived at a station on a nearly empty battery and found a faulty charger, the wrong connector or a queue will recognize the picture.

A conventional locator answers the question “where is the nearest charging station?” We think a driver with a low battery is really asking something else: which charging station is the safest, most suitable and most efficient choice for me right now? Answering that needs the state of charge of the vehicle, its connector type and range, live charger status, expected waiting time, price and some sense of how trustworthy the station is. It also needs the other side of the market. Station owners want to manage chargers, prices and bookings, and administrators need oversight of users, stations and complaints.

This paper presents ChargeSync, a platform that brings these needs together. Its main contributions are:

1. A battery-aware recommendation method that screens stations for reachability and connector

compatibility before ranking them, with a Safe-to-Reach score and an emergency mode for critically low batteries.
2. A reservation workflow with a defined booking lifecycle, no-show release and slot suggestion based on expected demand.
3. A multi-role ecosystem in which drivers, owners and administrators share one backend, with owner analytics, complaint handling and role-based access control.
4. A design for an AI layer that covers retrieval-augmented review summaries, a composite station health score and waiting-time and demand prediction.

The rest of the paper is organized as follows. Section II reviews related work, Section III outlines the requirements and user roles, Section IV describes the architecture and Section V explains the development process and the methods. Section VI lists the technology stack, Section VII gives the evaluation plan and current status, and Sections VIII and IX discuss limitations, future scope and conclusions.

II. RELATED WORK

A. Station selection and range anxiety

Savari et al. reviewed charging technologies, infrastructure and the schemes used to recommend charging stations to drivers [1]. EL-Fedany et al. proposed a station selection system that chooses the station giving the shortest total trip, using real-time queuing time, existing reservations and predicted future charging requests, and they evaluated it on a Helsinki city scenario [2]. Both works show that a recommendation should look beyond distance, which is the starting point of our design.

B. Reservation

Zhang et al. matched charging requests to time slots through a third-party reservation platform, and their simulations indicated that this eases queuing [4]. We adopt slot-based reservation but pair it with a no-show rule and a slot suggestion step, so that reserved capacity does not sit idle.

C. Occupancy, demand and waiting-time prediction

Hecht et al. used gradient boosting and random forest classifiers to predict next-day charger occupancy and reported 94.8% accuracy on their data, noting that past usage is the most useful input [5]. Ma and Faye predicted multistep occupancy states with a hybrid long short-term memory network [6]. Bhave et al. used a Random Forest model to predict charging demand, which is useful for planning but does not help a driver choose a station in real time [10]. Models like these need historical charging data, which a new platform does not have on its first day. ChargeSync therefore starts with a transparent, reservation-aware waiting-time estimator and leaves learned models as a later extension (Section V-E).

D. Real-time management and security

Zhu et al. used large data from vehicles, charging stations and smart grids for real-time charging management, which helps power management but does not give drivers real-time interaction [9]. ChargeSync pushes live charger status to drivers through Socket.IO. On the security side, Al-Amri et al. studied cyber threats and vulnerabilities in the EV ecosystem and highlighted the need for strong security, but without a full system-level solution [11]. ChargeSync applies secure login with JWT and bcrypt, role-based access control and an SOS emergency mode.

E. Review analysis and retrieval

Asensio et al. classified large-scale user reviews with machine learning to assess charging experiences [7]. Retrieval-augmented generation (RAG) pairs a language generator with an external, searchable memory [8]. That suits review summarization well, because an answer can be tied back to the retrieved text.

Table I summarizes these studies and how ChargeSync responds to each. In the literature we examined, these strands are mostly studied separately and are rarely packaged with the owner and administrator tools that a real charging network needs. ChargeSync is an attempt to put them together in one platform.

Table I: Literature survey and positioning of chargesync

Author & Year	Focus	Method Used	Finding / Gap	Our Work
Savari et al. (2023) [1]	Charging technologies and station	Literature review	A recommendation must look beyond distance	Battery-aware ranking on distance, speed, price,

	recommendation schemes			waiting time, rating and station health
EL-Fedany et al. (2023) [2]	Station selection for the shortest total trip	Real-time queuing, reservations and predicted requests (Helsinki scenario)	Recommendation-focused; owner and admin tools are not part of the system	Recommendation together with owner and admin portals on one backend
Wang et al. (2023) [3]	Range anxiety of battery EV users	Survey of 217 BEV users in mainland China	Both distance and waiting time influence where drivers charge	Safe-to-Reach score and waiting-time estimate shown to the driver
Zhu et al. (2022) [9]	Real-time EV charging management	Large data from vehicles, charging stations and smart grids	Helps in power management but does not provide real-time interaction to drivers	Live station and charger status for drivers using Socket.IO
Hecht et al. (2021) [5]	Charging station availability prediction	Gradient boosting and random forest on historical data	Needs historical data, which a new platform lack	Starts with a transparent, reservation-aware waiting-time estimator
Bhave et al. (2023) [10]	EV charging demand prediction	Machine learning model (Random Forest) to predict charging needs	Useful for planning but does not help users choose stations in real time	Recommendations based on battery level, location and distance
Asensio et al. (2020) [7]	Charging experience from user reviews	Machine-learning classification of large-scale reviews	Analyses reviews but does not summarize them for a driver on demand	Retrieval-augmented review summaries tied to each station
Al-Amri et al. (2024) [11]	Security in the EV ecosystem	Study of cyber threats and system vulnerabilities	Highlights the need for strong security but lacks a full system-level solution	Secure login (JWT, bcrypt), role-based access control and SOS emergency mode

III. REQUIREMENTS AND USER ROLES

ChargeSync serves three roles, summarized in Table II. Drivers need quick decisions on a phone, owners need visibility into operations and revenue, and administrators need oversight of the whole platform.

Table II User Roles and Main Functions

Role	Interface	Main functions
EV user	Mobile app	Station search, vehicle profile, recommendations, booking, navigation, payment, reviews, SOS mode, charging history
Station owner	Web portal	Station, charger and connector management; pricing; bookings; revenue and utilization analytics; complaints
Administrator	Web panel	User and owner management; station and booking monitoring; complaints; audit logs; platform statistics

Beyond these role-specific functions, the system must check connector compatibility, show each charger as AVAILABLE, OCCUPIED, RESERVED, OFFLINE or MAINTENANCE, keep every role inside its own permissions, and send notifications for bookings, charging events, payments and complaint updates.

IV. SYSTEM ARCHITECTURE

A. Layered design

The mobile app and the two web applications talk to a central backend through REST APIs (Fig. 1). Socket.IO pushes live changes, such as a charger switching from AVAILABLE to OCCUPIED, to connected clients. The backend is split into services for authentication, stations and chargers, bookings and payments, and reviews and complaints. A data layer persists everything, and an AI layer built with Brain.js, machine learning algorithms and retrieval-augmented generation hosts the recommendation, prediction and review-summarization components.

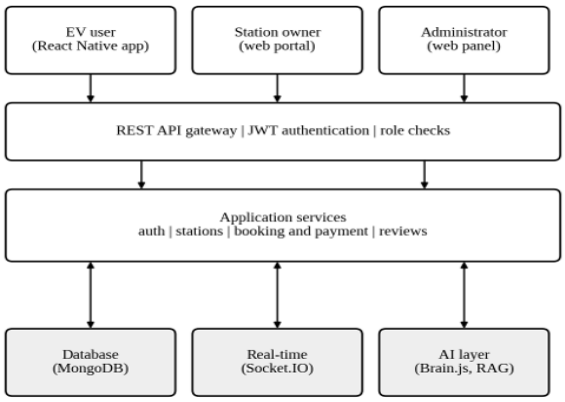


Fig. 1. Layered architecture of ChargeSync.

B. Data model

The data layer holds users, owners, administrators, vehicles, stations, chargers, connectors, bookings, charging sessions, payments, reviews, complaints, notifications, pricing rules, maintenance records, stored recommendations and station analytics. A station belongs to an owner and contains chargers, each charger exposes one or more connectors, and bookings, reviews and analytics hang off the station.

C. Security

The design relies on bcrypt password hashing, JSON Web Token (JWT) authentication and role-based access control with three roles: USER, OWNER and ADMIN. Each protected route checks the role of the caller against the permissions that role needs. Inputs are validated at the API boundary, and database access goes through the service layer only.

V. METHODOLOGY

A. Development process

ChargeSync is developed in the six phases listed in Table IV. The subsections that follow describe the recommendation, prediction and booking methods built in the design and integration phases.

Table IV. Development Phases

No.	Phase	Activities
1	Requirement analysis	Identify the key problems faced by EV users and charging station owners, and define the system requirements for the three roles.
2	System design	Design the overall architecture, the database structure and the user interfaces for web and mobile.
3	Development	Build the application using the MERN stack and integrate mobile access.
4	Smart integration	Implement real-time updates and the AI/ML-based recommendation features.
5	Testing	Test the APIs and system functionality, including the SOS feature and data accuracy (Table VI). No results have been measured yet.
6	Deployment	Deploy the system on cloud platforms and improve it from user feedback. This phase is planned.

B. Inputs and notation

A driver's vehicle is described by its battery capacity C (kWh), rated full-charge range Rfull (km), connector type and maximum charging power Pveh (kW). The driver's current state of charge is σ (%). For each candidate station i the system knows the road distances di (km), the highest compatible charger power Pi (kW), the price pi (INR/kWh), the estimated waiting time Wi (min), the rating ri (0 to 5) and the health score Hi (0 to 100).

C. Reachability and the Safe-to-Reach score

The first question is whether a station can be reached at all. We define the reachability ratio as the share of the usable range that the trip would consume:

$$\rho i = di / [\ \kappa\ (\sigma/100)\ Rfull] \quad (1)$$

Here κ ($0 < \kappa \leq 1$) is a derating factor that accounts for real driving conditions such as traffic, air-conditioning load and terrain. The estimated charge on arrival is $\sigma arr,i = \sigma\ (1 - \rho i)$. Stations with $\rho i \geq 1$ are unreachable and are dropped. For the rest we compute a Safe-to-Reach score that stays at 100 while the margin is comfortable and then falls to zero as the trip approaches the limit of the range:

$$SRi = 100 \cdot \min\ \{1, \max\ [\ 0, (1 - \rho i) / (1 - \rho s)]\} \quad (2)$$

A station is labeled SAFE when $\rho i \leq \rho s$, CAUTION when $\rho s < \rho i \leq \rho r$, and HIGH RISK above ρr. Both thresholds are configurable. Table III applies the method with illustrative values $\rho s = 0.75$ and $\rho r = 0.90$ to a driver whose battery is at 18% and whose estimated remaining range is 55 km. The figures in Table III come from evaluating the formulas, not from an experiment.

Table III. Worked Example of Reachability Analysis

Station	d (km)	ρ	Arrival charge	SR	label
A	12	0.22	14.1%	100	safe
B	39	0.71	5.2%	100	safe
C	51	0.93	1.3%	29	high risk

D. Weighted ranking

Stations are first removed if the connector is incompatible, every compatible charger is OFFLINE or in MAINTENANCE, or the station is unreachable. The remaining stations are scored on six criteria, each normalized to the range 0 to 1 across the candidate set: proximity ($1 - di/dmax$), charging speed (the lower of Pi and Pveh, divided by the best value in the set), price (cheapest scores 1), waiting time ($1 - Wi/Wmax$), rating ($ri/5$) and station health ($Hi/100$). The overall score is

$$Scorei = 100\ \Sigma k\ wk\ nk,i \quad (3)$$

where nk,i is the normalized value of criterion k and the weights wk sum to one. The weights come from a stored preference profile. A driver who mostly wants speed might use 0.40 for speed, 0.20 for price, 0.25 for waiting time and 0.15 for rating. The system returns three views of the ranked list: the best overall score, the highest Safe-to-Reach score, and the lowest estimated session cost. The cost view uses the energy needed to reach a target charge σt:

$$E_i = C\,(\sigma_t - \sigma_{arr,i})/100, \quad Cost_i = p_i E_i \qquad (4)$$

This ignores charging losses and assumes a single price band during the session.

E. Waiting-time estimate

Instead of showing only "two chargers available", ChargeSync estimates how long the driver would wait on arrival. Let τj be the time at which compatible charger j is expected to be free, taken as the end of the current session or of the last confirmed booking, and let ti be the expected arrival time at station i (now plus travel time). Then

$$W_i = \min_j \max(0, \tau_j - t_i) \qquad (5)$$

so, the estimate is zero whenever a charger is free at arrival. This needs no history. Once enough charging sessions have been logged, learned models of the kind reported in [5] and [6] can refine the estimate by correcting τj for time of day and day of week.

F. Station health score

Star ratings alone say little about reliability. The station health score combines five components, each scaled to 0–100: availability reliability, charger uptime, user rating, waiting-time performance and complaint rate (inverted, so that fewer complaints give a higher value):

$$H = \Sigma_m\, v_m h_m \qquad (6)$$

with weights vm that sum to one and can be set by the administrator. The score feeds the ranking in (3) and is shown to drivers and owners.

G. Booking, slot suggestion and no-shows

A booking passes through the states shown in Fig. 2. The driver picks a station, charger, date and time, the price is calculated from the owner's pricing rules (normal, peak and off-peak rates), and the booking is confirmed after payment and an owner notification. To suggest a slot, the system looks at start times near the one requested and computes the booked share ut of the compatible chargers in each; it recommends the nearest slot whose ut is below a configurable threshold, or the least busy slot if none qualifies. If the driver has not started charging within a grace period after the slot begins, the booking becomes NO-SHOW and the slot is released.

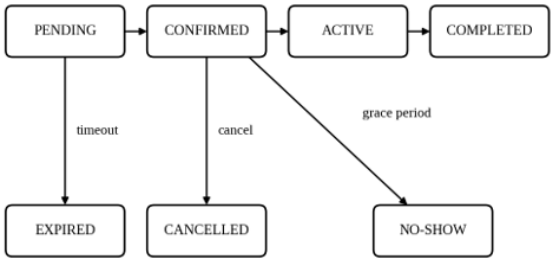


Fig. 2. Booking lifecycle.

H. Emergency mode

When the state of charge falls below a critical level, the ranking changes (Fig. 3). Only reachable stations with a compatible, available fast charger are kept. They are ordered by Safe-to-Reach score, then by waiting time and then by distance, and the interface shows the estimated arrival charge so that the driver can see the margin. Safe reachability takes priority over price and rating.

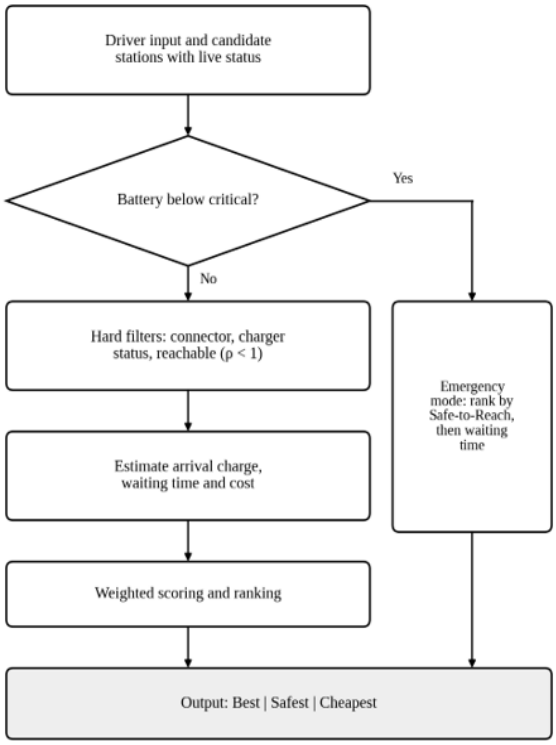


Fig. 3. Recommendation flow, including the emergency branch.

I. Review intelligence

Reading hundreds of reviews is not realistic for a driver in a hurry. ChargeSync cleans the reviews of each station, splits them into chunks, converts the chunks to embeddings and stores them in a vector index tagged by station. When a driver asks a question such as whether a station is reliable in the evening, the system retrieves the most similar chunks for that station and passes them to a language model, which is instructed to answer only from the retrieved text and to separate positive from negative points [8]. Three safeguards keep the summary honest: retrieval is limited to the station in question, a minimum number of reviews is required before a summary is shown, and the supporting reviews, rating and complaint count are displayed beside it so that the driver can check the claim.

J. Sustainability estimate

A charging history page can report an estimate of the emissions avoided by driving an EV instead of a conventional vehicle over distance D:

$$\Delta CO2 = D\,(EFice - e\,EFgrid) \quad (7)$$

where EFice is the emission factor of a reference petrol or diesel vehicle (kg CO2/km), e is the energy consumption of the EV (kWh/km) and EFgrid is the emission factor of the electricity grid (kg CO2/kWh). The assumptions behind these factors must be stated wherever the number is shown, and it should be read as an estimate. An optional points-and-badges scheme rewards completed and off-peak charging sessions.

VI. TECHNOLOGY STACK

Table V summarizes the technology choices for each part of the platform.

Table V. Technology Stack

Component	Technologies
Mobile app	React Native, TypeScript, React Navigation, Redux Toolkit
Web portals	React.js, Vite, JavaScript
Backend	Node.js, Express.js, REST API, Socket.IO
Database	MongoDB (MongoDB Atlas for hosting)
Security	JWT, bcrypt, role-based access control
Maps	Leaflet.js with OpenStreetMap (web); Google Maps API and react-native-maps (mobile)
AI and ML	Brain.js, machine learning algorithms; embeddings, vector database and a language model for RAG
API testing	Thunder Client
Configuration	Dotenv (.env)
Media and version control	Cloudinary, GitHub

Parts of the design depend on physical charger hardware, a live payment provider or large historical charging datasets. Unless stated otherwise, those parts are presented in this paper as design-level proposals.

VII. EVALUATION PLAN AND CURRENT STATUS

We have not yet collected quantitative measurements for ChargeSync, and this paper reports none. Table VI lists the planned experiments, and each result is marked “to be measured” until real tests have been run.

Table VI. Planned Evaluation

Experiment	What is measured	Result
Functional testing	Registration, login, search, booking, cancellation, payment, owner station management, admin and complaint flows (pass or fail)	To be measured
Scenario tests	Low and high battery, price versus speed preference, long distance, unavailable charger, incompatible connector, long wait, emergency	To be measured
Performance	Response time of API, search, booking and recommendation calls; database query time	To be measured
User study	Ease of use, recommendation satisfaction, booking experience, interface, overall satisfaction	To be measured

For the recommendation tests we plan to fix a seeded set of stations, run each scenario and check whether the ranked output matches what a sensible driver would choose, using a nearest-station list as the baseline. For the user study we will recruit real users and report the questionnaire, the number of participants and the scores once the study is complete.

VIII. LIMITATIONS

Several limits should be kept in mind. Live charger status and charging telemetry need integration with charger hardware or operator APIs. The quality of any recommendation depends on the quality of the station data behind it, and map and GPS accuracy depend on the chosen mapping provider. Learned waiting-time and demand models need enough historical data to be useful. Generated review summaries need careful retrieval and validation, and payment integration must follow the rules of the chosen provider. Finally, a production deployment would need further security and scalability testing.

IX. CONCLUSION AND FUTURE SCOPE

Choosing where to charge is a decision problem, and the nearest station is only one input to it. ChargeSync builds the decision around the driver's battery and vehicle, checks whether a station can safely be reached, and gives owners and administrators the tools to run the other side of the network. The immediate next step is to carry out the experiments in Table VI and report the results.

Beyond that, we see three directions. The first is IoT integration, in which a charger sends voltage, current, power, energy and fault status to the backend through MQTT or an API. The second is learned models for waiting time, demand, station failure and energy demand, together with dynamic pricing. The third is a conversational assistant for natural-language charging search and trip planning.

ACKNOWLEDGMENT

The authors thank Ms. V. V. Patil, Associate Professor, CSE Department, Sharad Institute of Technology College of Engineering, Yadrav, Ichalkaranji, for their guidance and support during this project.

REFERENCES

[1]. Savari GF, et al. Assessment of charging technologies, infrastructure and charging station recommendation schemes of electric vehicles: A review. Ain Shams Engineering Journal. 2023;14(4): Article 101938.

[2]. EL-Fedany I, Kiouach D, Alaoui R. A smart system combining real and predicted data to recommend an optimal electric vehicle charging station. Indonesian Journal of Electrical Engineering and Computer Science. 2023;30(1):394–405. doi:10.11591/ijeecs. v30.i1. pp394-405.

[3]. Wang J, Huang C, He D, Tu R. Range anxiety among battery electric vehicle users: Both distance and waiting time matter. arXiv:2306.05768. 2023.

[4]. Zhang X, Li Y, Wang W, Zhao Y, Xu T. Research on the reservation matching algorithm for electric vehicles charging. Journal of Electrical Engineering. 2020;15(2):18–23. doi:10.11985/2020.02.003.

[5]. Hecht C, Figgener J, Sauer DU. Predicting electric vehicle charging station availability using ensemble machine learning. Energies. 2021;14(23): Article 7834. doi:10.3390/en14237834.

[6]. Ma TY, Faye S. Multistep electric vehicle charging station occupancy prediction using hybrid LSTM neural networks. arXiv:2106.04986.

[7]. Asensio OI, et al. Real-time data from mobile platforms to evaluate sustainable transportation infrastructure. Nature Sustainability. 2020;3(6):463–471. doi:10.1038/s41893-020-0533-6.

[8]. Lewis P, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Advances in Neural Information Processing Systems 33 (NeurIPS 2020); 2020.

[9]. Zhu et al. 2022. [Incomplete reference: full bibliographic details required.]

[10]. Bhave et al. 2023. [Incomplete reference: full bibliographic details required.]

[11]. Al-Amri et al. 2024. [Incomplete reference: full bibliographic details required.]