Tribhuvan University

Orchid International College

A Final Year Project Report

On

ELECTRIC VEHICLE CHARGING STATION RECOMMENDATION SYSTEM USING COLLABORATIVE FILTERING

Under the Supervision of

Er. Utsab Koirala

Lecturer

Orchid International College

Submitted To:

Orchid International College

In partial fulfillment of the requirement for the Bachelor Degree in Computer Science and Information Technology

Submitted By:

Rajat Budhathoki (20820/075)

Bipin Chaudhary (20800/075)

Abhinna Ojha (20788/075)

April, 2023

ORCHID INTERNATIONAL College

SUPERVISOR'S RECOMMENDATION

I hereby recommend that the report prepared under my supervision by Rajat Budhathoki (20820/075), Bipin Chaudhary (20800/075), Abhinna Ojha (20788/075) entitled "Electric Vehicle Charging Station Recommendation System using Collaborative Filtering" in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

……..…….

Er. Utsab Koirala

Lecturer

Orchid International College

Bijayachowk, Gaushala

ORCHID INTERNATIONAL College

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by Rajat Budhathoki (20820/075), Bipin Chaudhary (20800/075), Abhinna Ojha (20788/075) entitled "Electric Vehicle Charging Station Recommendation System using Collaborative Filtering" in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

---------- **Er. Utsab Koirala** Supervisor, Lecturer, Orchid International College, Bijayachowk, Gaushala	---------- **Er. Dhiraj Kumar Jha** Head Of Department, Orchid International College, Bijayachowk, Gaushala
---------- **Ms. Sikha Sharma** Internal Examiner, Full Time Faculty, Orchid International College, Bijayachowk, Gaushala	---------- **External Examiner** Central Department of Computer Science and IT Tribhuvan University Kirtipur, Nepal

ACKNOWLEDGEMENTS

We would like to convey special gratitude towards our supervisors and mentors, Er. Utsab Koirala, Er. Dhiraj Kumar Jha, Ms. Sikha Sharma, and Er. Diwakar Upadhyaya, who provided us with immense guidance and support that not only helped us to complete the project but also learn valuable lessons and guidelines in the process. In spite of their busy schedule, they made sure that our concerns and queries were addressed and were present even on the oddest time of the day.

Furthermore, we would also like to extend our gratitude to the staff and faculty of Orchid International College for their constant support and co-operation throughout the timeframe. Sincere appreciation to all my friends and colleagues who were directly or indirectly helpful for the completion of this project.

ABSTRACT

As charging an EV is not as fast and only some charging stations are equipped with fast charging port, EV drivers and owners face a dilemma on whether a certain location even has a charging station or not, and is the charging station is suitable for them or not. This system aims to provide users a web-based platform that can recommend the potentially suitable charging station to them. The system will be developed in Laravel, utilising its capabilities for potent web development and efficient database management. The proposed system provides personalized recommendations to users based on their charging needs and preferences, making it easier for them to find and use charging stations. The results of our evaluation demonstrate the effectiveness and efficiency of the proposed system in recommending charging stations to users.

Keywords: ***Electric Vehicle (EV), Charging Station, Recommendation System, Cosine Similarity, Weighted Average, Laravel***

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AC	Alternating Current
AI	Artificial Intelligence
CF	Collaborative Filtering
CSS	Cascading Style Sheets
DC	Direct Current
EV	Electric Vehicle
EVCS	Electric Vehicle Charging Station
HTML	Hyper Text Markup Language
ICE	Internal Combustion Engine
IDE	Integrated Development Environment
JS	JavaScript
L-ION	Lithium Ion
ML	Machine Learning
MVC	Model View Controller
ORM	Object Relational Mapping
PHP	Hyper Text Pre-processor
QA	Quality Assurance
SDLC	Software Development Life Cycle
UML	Unified Modelling Language

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1. Introduction

Electric vehicles, EVs, are gaining popularity in recent years. The reduction of taxes by various nations on EVs, the consideration of environment, and the ever-depleting petroleum resources has made EVs a possible vehicle in consideration for many drivers and the easier and cheaper availability of electricity further solidifies the case for the choice of EVs. Similarly, recommendation systems are also nowadays used on almost every walk of digital life. The increasing adoption of electric vehicles (EVs) has led to a growing demand for charging infrastructure. However, finding a charging station that is both conveniently located and available can be a challenge for EV drivers. The aim of this project is to address this challenge by providing a user-friendly and efficient solution for recommending charging stations. The system utilizes the technique of cosine similarity to recommend the most suitable electric vehicle charging stations in a specified location. The report will detail the design, implementation and evaluation of the system, as well as its overall performance and potential future improvements.

1.2. Problem Statement

The big problem of Electric Vehicles (EV), is charging. EV charging is not as easy as filling up fuel as in an internal combustion engine-based vehicle as it takes quite a significant time for the EV to be fully charged. So, picking an EV charging station can be a big personal preference-based choice and finding a right one can be a topic of choice. With the scarce availability of the EV charging stations, it can be very difficult to even look for an EV charging station let alone being recommended one that a person may like, trust, and ultimately prefer.

1.3. Objectives

- To create a web-based recommendation system that recommends EV charging stations in an area
- To implement user-user-based collaborative filtering using cosine similarity

1.4. Scope and Limitation

The project aims to build a system that can be used to provide ratings to various charging stations and receive recommendation of charging station in a user specified location.

The system will not detect the user location of provide routes to the said charging stations.

1.5. Development Methodology

The project is developed using incremental development approach. Incremental development is a method of building software products in which a system is built piece-by-piece. Incremental development was favoured over other approaches due to multitude of factors. The team is not very well skilled or trained and is on a constant learning process during the development process itself. This approach allowed so that the system can be developed in smaller modules and functions can be added over the course of development one by one. Similarly, the system could be developed such that the core requirements of the system could be developed first and then other add-ons could be developed as need be. Furthermore, due to the requirements being set and clear, it was easier to implement the system in incremental approach as client feedback was not a must have requirement for the project. All in all, as requirements are superior, the developers are still learning, and the prioritized requirements need to be developed first, the system is built using incremental approach as small piece-by-piece modules.

1.6. Report Organisation

The report of the Electric Vehicle Charging Station Recommendation System is organised into six chapters.

Chapter 1 is the introductory chapter that encompasses the project introduction, the scope, limitation, and development methodology.

Chapter 2 is the background study and the literature review. This section encompasses the summary of the works done which are similar to the project in consideration. It is research-based analysis and summary of similar works or works that may be relevant to the project. This section summarises the works that have been taken as reference, guideline, or as an influence to the project.

Chapter 3 provides a detailed analysis of the system and the project as a whole. This section focuses on identifying the viability of the project and the requirement of the

system. This section ensures that the system is viable, the platform to be used is viable, and the project is doable.

Chapter 4 highlights the system and how is it to be developed. It is a study of how should the system be built, which approach is to be taken, and what algorithms are to be studied and implemented to ensure smooth and hassle-free development along with providing a system of proper quality.

Chapter 5 provides an outlook on how was the system actually developed and how was it handled. It provides summary on what tools were implemented and why, how was the algorithm implemented, and how was the database implemented. It also summarises the QA aspects of the project.

Finally, chapter 6 provides an insight about the system and the project and how can the system be further improved.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1. Background Study

Electric Vehicle is not a new concept. It has been a topic of discussion and research as an alternative fuel vehicle since the beginning of 21st century. With the breakthrough in research in recent years, and a global awareness of the adverse climatic effects due to fossil fuels, electric vehicles has been a prominent and upcoming field of research.

The biggest problem of the EV is the travel distance as the per charge range is significantly lower than the per tank range of ICE based vehicles. The other fuel system is the use of Hydrogen fuel, but the problematic storage of Hydrogen and Oxygen required for the fuel cell makes the process of fuel cell-based vehicles a challenging concept. Solar power can be used, but solar power produces inconsistent power and may be subject to higher cost due to the innate costs of the panels, maintenance, and repair. So, with careful consideration to all the possibilities, EVs with L-ION batteries and accessible charging stations seems to be the most likely future for the next few decades. Research in EVs have been prominent and research into ways to increase the range of EVs have been a topic of heavy interest. But, for now, it seems though that the availability of suitable EVCS is the better approach towards EV charging.

Furthermore, most of the EVCS recommendation system is based on technicalities. EV charging is treated as ICE vehicle fuelling and recommendations are based on technical parameters only. While technical parameters are important and effective, EV charging is more than just the technicalities. EV charging is not as quick and robust as ICE fuelling and the human aspect is mostly neglected by most researches and systems. While recommendation systems like that of movies is based on human preferences, the recommendation system of EVCS mostly does not consider potential human factors like what will they do during the downtime when their vehicle is charging and how can recommendation be made more human-centric.

2.2. Literature Review

Many prominent researches have been done on the recommendation systems and few of them are based on the electric vehicle charging station.

In a study on Electric Vehicles in Hyderabad [1], the authors conducted research on the EV taxies in Hyderabad, and how can the EVCS recommendation be beneficial to the EV taxies. The EV taxies were compared based on real and simulated recommendations with an accuracy of 94.7%, among which the accuracy of to-station state prediction is 92.2%, 96.8% for recharging, and 95.3% for operating. The comparison of real selections with predicted selections was obtained as 84.7% on integral recharge intention identification, including to-station state, and charging station selection.

Research in a similar system [2] was conducted in 2021 which discusses recommendation system of EVCS. The research focused on various parameters like charging request, charging wait time, charging price, and charging failure rate to recommend the most proper charging station with long term goals of minimising the charging wait time, charging price, and charging failure rate.

A similar study in Tamil Nadu [3] proposes a system for the EV charging station recommendation system for load-based taxis. The study suggested that EV taxis' long-time cost at charging stations results from the EV taxi drivers always rushing to the same charging stations during the same period. Furthermore, the preliminary observations and analysis of recharging behaviour patterns suggest the necessity of the charging station recommendation system. The system also suggested the use of real-time recommendation system so that the taxi drivers can get real time suggestions so that they do not rush only to the charging stations they frequently use.

The study [4] in Beijing in 2021 also proposed a similar recommendation system, albeit using federated learning. The research compared the model based on real data and that based on their model and concluded that the model was accurate by a factor of 0.92, 92%.

CHAPTER 3: SYSTEM ANALYSIS

3.1. Requirement Analysis

Requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project. Requirements analysis is the process of determining user expectations for a new or modified product. It involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore, requirements analysis means to analyse, document, validate and manage software or system requirements. High-quality requirements are documented, actionable, measurable, testable, traceable, helps to identify business opportunities, and are defined to a facilitate system design.

The functional requirements specify the core features and the expected behaviours of the system and is modelled using Data Flow Diagrams, UML diagrams, flowcharts, function modelling and so on. The non-functional requirements are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality. These are basically the requirements that outline how well it will operate.

3.1.1. Functional Requirements

The functional requirements specify the core expected behaviour of the system. It is a description of the service that the software must offer and describes a software system or its component. In this system some core functions that have been modelled sing the use-case diagram and use-case descriptions.

The core features of the system are to add and update charging stations by the admin, add and edit charging station ratings by the user, and get recommendation of charging station by user.

3.1.1.1. Use-case diagram

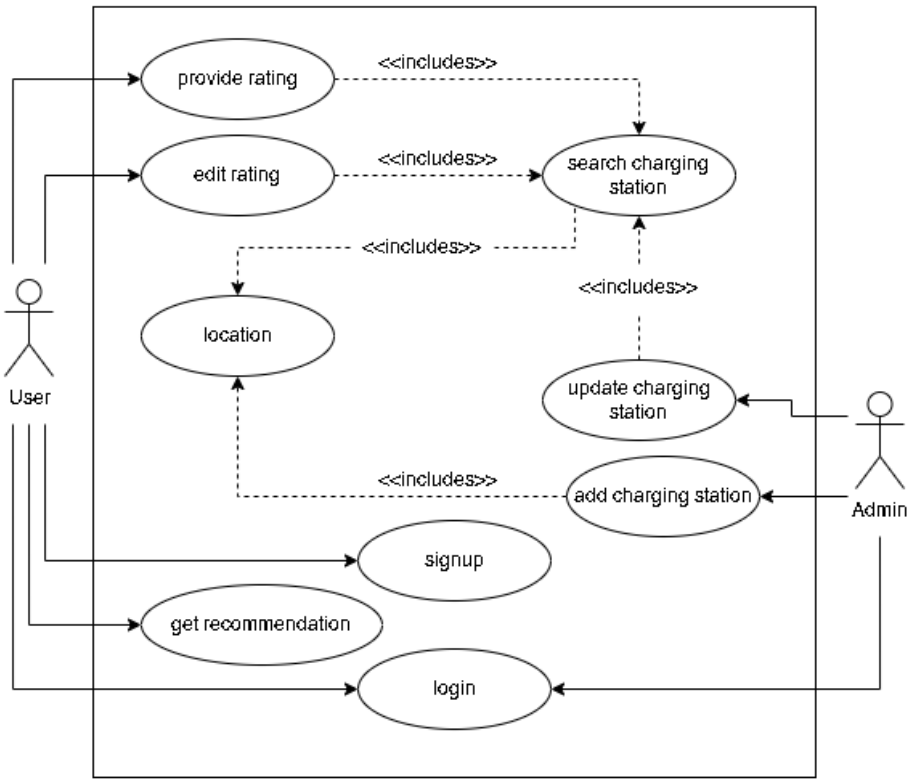


Figure 3.1: Use case of the System

3.1.1.2. Use-case descriptions

Table 3.1: Use case description of rating a charging station

Identifier	UCD1
Name, Description	Providing the rating to a charging station
Actors	User
Pre-condition	The location must be specified and charging station must be selected
Success scenario	Rating is inserted into the database Success message is displayed to user
Failure scenario	Rating is not inserted into the database and failure message is displayed to admin

Table 3.2: Use case description of adding a charging station

Identifier	UCD2
Name, Description	Adding a charging station
Actors	Admin
Pre-condition	The location must be specified
Success scenario	Charging station is inserted into the database Success message is displayed to admin
Failure scenario	Charging station is not inserted into the database and failure message is displayed to admin

Table 3.3: Use case description of updating a charging station

Identifier	UCD3
Name, Description	Updating a charging station
Actors	Admin
Pre-condition	The location must be specified and the charging station to update must be selected
Success scenario	Success message is displayed to admin Charging station is updated in database
Failure scenario	Charging station is not updated in database and message is displayed to admin

Table 3.4: Use case description of getting a charging station recommended

Identifier	UCD4
Name, Description	Get recommendation of charging station
Actors	User
Pre-condition	The location must be specified
Success scenario	Charging stations and their location is displayed to user if the charging station is available Message is displayed to user if the location has no charging station
Failure scenario	Charging station is not recommended as desired

3.1.2. Non-Functional Requirements

- Speed: It is used to determine how fast the system performs certain activities.
- Security: Only registered users can login to the system and provide ratings or get recommendations.
- Availability: The system will be available at all hours, every day of the year once deployed.
- Usability: The system should be easy to access, use, and understand by the users

3.2. Feasibility Study

3.2.1. Technical Feasibility

The dataset to train and test the recommendation system can be available from sources like NEA and Kaggle. Laravel framework would be used to build the recommendation system and the collaborative filtering technique with cosine similarity would be used for the recommendation algorithm. The next step would be to develop the recommendation system using the Laravel framework and the chosen algorithm. This would involve the implementation of various features such as user authentication, data management, and recommendation generation. The developed system would then be tested to ensure its functionality and performance. This would involve evaluating the system's accuracy and efficiency in generating recommendations, as well as user satisfaction with the system. Once the system has been tested and evaluated, it would be deployed for use by EV owners. This would involve the integration of the system with existing EV charging stations and the implementation of any necessary security measures.

All in all, the technical feasibility of this project is high as the Laravel framework is a robust and widely used platform for web development, and collaborative filtering with cosine similarity is a well-established recommendation algorithm. With the appropriate data and resources, the development, testing, and deployment of this system is achievable.

3.2.2. Operational Feasibility

The system will be able to recommend the EV charging station to the user in the specified location. This recommendation will be based on the ratings the user has provided to the EV charging stations that the user has already been to and rated. This solves the problem for the user to manually search the EV charging station and make a guess of whether the one they found may be one of their liking or not and whether they will be able to trust their expensive vehicles on the said charging station or not. Furthermore, having a system of authentication and authorisation will also help prevent fake ratings and make the system more reliable and robust.

3.2.3. Schedule Feasibility

EVCS Recommendation System

ID	Task Name	Duration	Start	Finish	Predecessors
1	**Planning and installation**	**40 hrs**	**Dec 25**	**Dec 30**	
2	Identification of available projects	5 hrs	Dec 25	Dec 25	
3	Identification and analysis of required resources	20 hrs	Dec 25	Dec 28	2
4	Feasibility study	10 hrs	Dec 25	Dec 27	2
5	Installation of required environment	15 hrs	Dec 28	Dec 30	4,3
6	**Designing Database**	**50 hrs**	**Dec 31**	**Jan 07**	**5**
7	Acquisition and study of dataset	30 hrs	Dec 31	Jan 04	5
8	Database Design	20 hrs	Jan 04	Jan 07	7
9	**Prototyping of web pages**	**65 hrs**	**Dec 31**	**Jan 09**	**5**
10	Low-fidelity prototyping	25 hrs	Dec 31	Jan 03	5
11	Resource collection	10 hrs	Jan 03	Jan 05	10
12	High-fidelity prototyping	30 hrs	Jan 05	Jan 09	11
13	**Front end development**	**75 hrs**	**Jan 09**	**Jan 20**	**12**
14	Page structuring	25 hrs	Jan 09	Jan 13	12
15	Page designing	25 hrs	Jan 13	Jan 17	14
16	Page interactivity	25 hrs	Jan 17	Jan 20	15
17	**Back end development**	**160 hrs**	**Jan 07**	**Jan 31**	**8**
18	Database development	40 hrs	Jan 07	Jan 13	8
19	Implementation of algorithms	120 hrs	Jan 13	Jan 31	18
20	**Integration and alpha-deployment**	**50 hrs**	**Jan 31**	**Feb 07**	**16,19**
21	Integration of front and back ends	50 hrs	Jan 31	Feb 07	16,19
22	Release of alpha version	0 hrs	Feb 07	Feb 07	21
23	**Testing**	**180 hrs**	**Feb 07**	**Mar 06**	**22**
24	Algorithm testing	100 hrs	Feb 07	Feb 22	22
25	System testing	80 hrs	Feb 22	Mar 06	22,24
26	**Release of final version**	**0 hrs**	**Mar 06**	**Mar 06**	**24,25**

Figure 3.2: Work breakdown structure

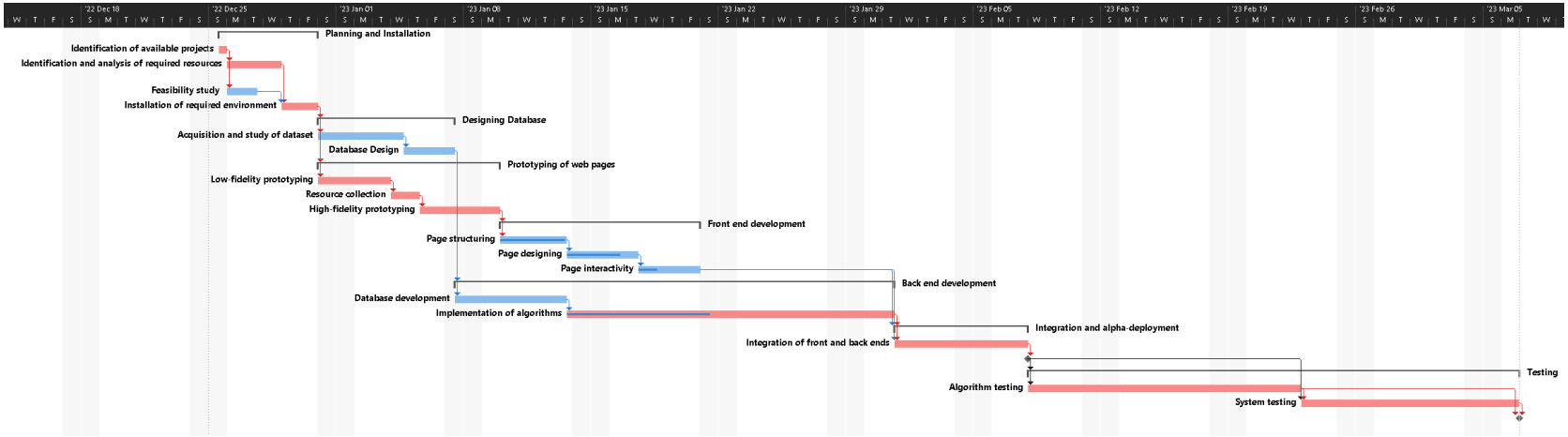


Figure 3.3: Gantt chart

3.3. System Analysis

3.3.1. Class Diagram

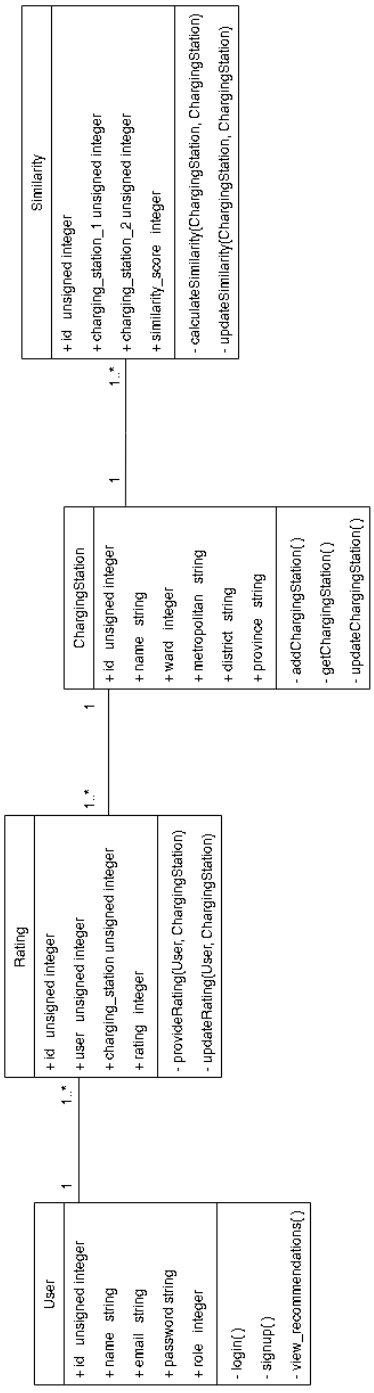


Figure 3.4: Class diagram

3.3.2. Activity Diagram

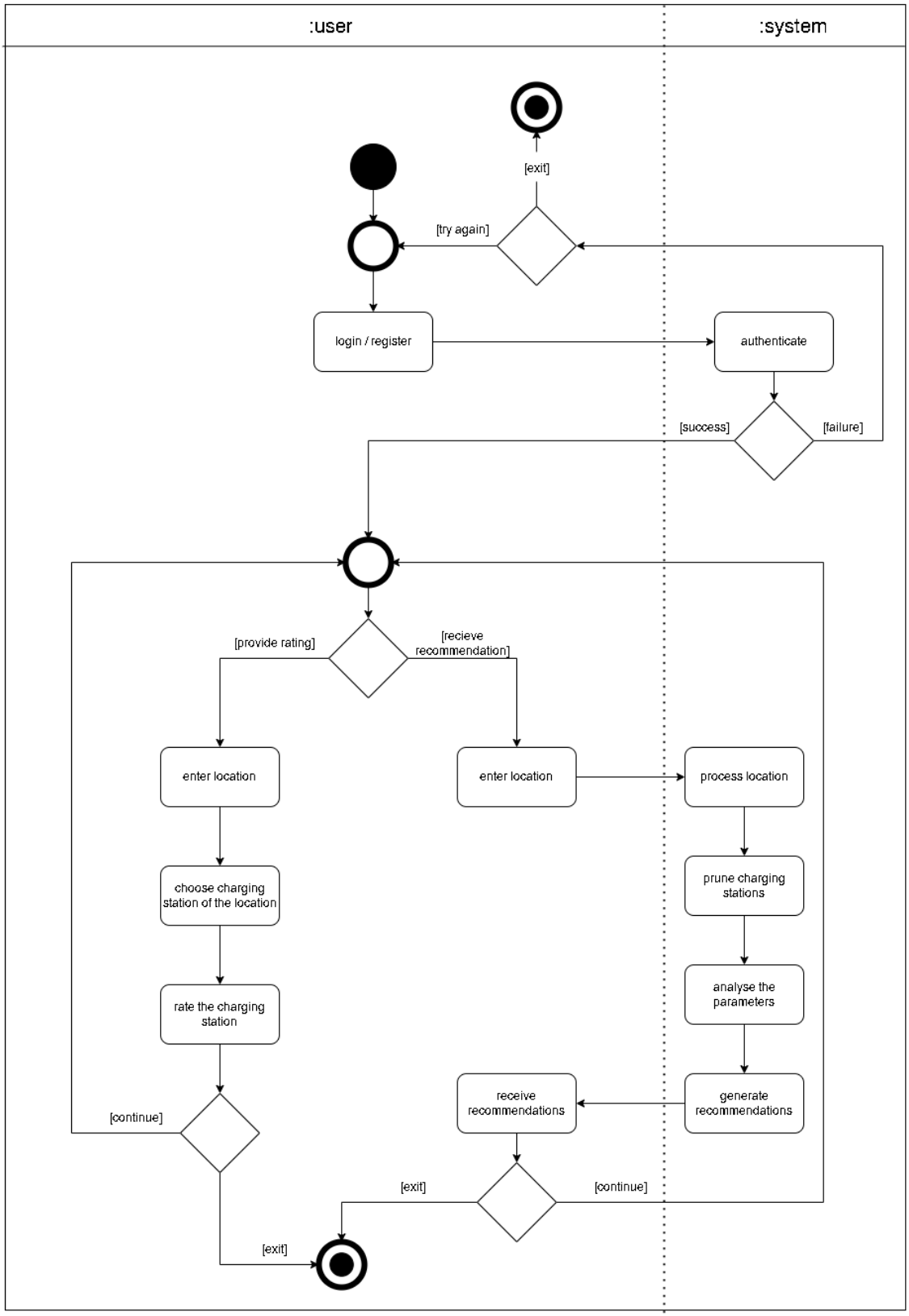


Figure 3.5: Activity diagram to provide rating or receive recommendation

3.3.3. Sequence Diagram

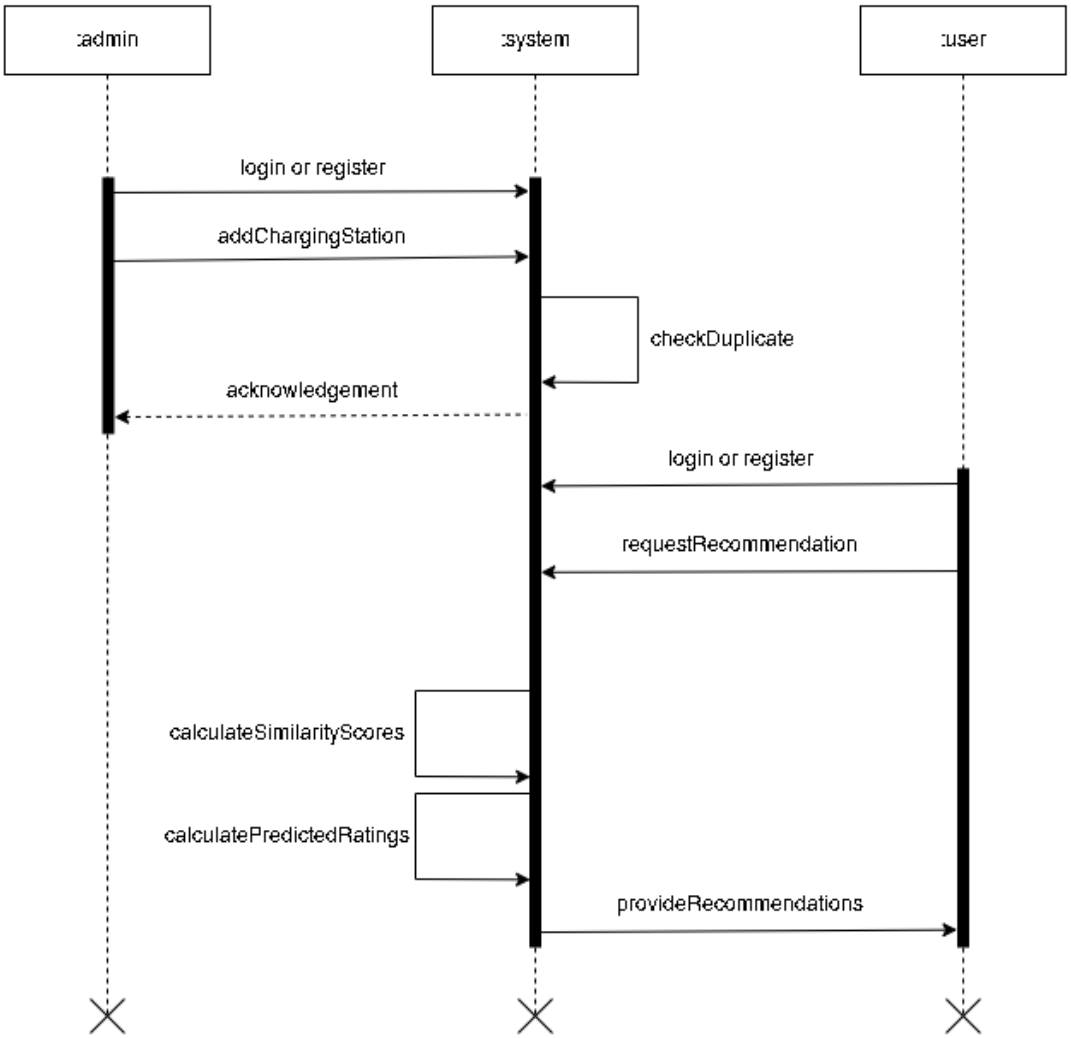


Figure 3.6: Sequence diagram to receive recommendation after adding a station

CHAPTER 4: SYSTEM DESIGN

4.1. Design

4.1.1. Model View Controller Architecture

The system is built following the Model View Controller, MVC, architecture. The model handles the data logic, database communication, and data on database. The controller contains the core business and implementation logic. It also acts as the intermediary between the model and view and helps render views via routes. The view renders the pages and acts as the presentation part of this system. As MVC is one of the most popular and efficient design architectures for web-based systems, we chose to use MVC architecture for this system.

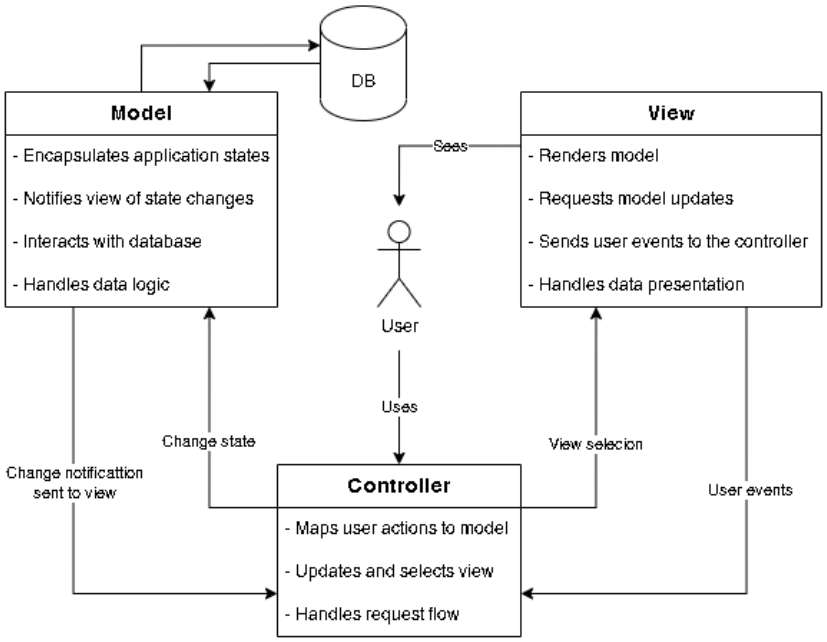


Figure 4.1: Model View Controller architecture

4.2. Study of Algorithms

Collaborative filtering uses algorithms to filter data from user reviews to make personalized recommendations for users with similar preferences. Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users.

Collaborative filtering can broadly be implemented by two approaches; the user based and the item based. The user-based approach uses data from the various users to find similarities and then recommend while the item-based approach uses data from the various items to find similarities and then recommend.

Furthermore, collaborative filtering can be implemented using memory-based, model-based, hybrid, or deep learning methods. Memory-based approach uses similarity calculation and weighted average rating method. Model based approach develops a model using different data mining, machine learning algorithms to predict users' rating of unrated items. The hybrid approach combines the memory-based and the model-based CF algorithms. These overcome the limitations of native CF approaches and improve prediction performance. Deep learning technique is a recent method that uses neural and deep-learning techniques. While deep learning has been applied to many different scenarios, it is not effective when used in a simple collaborative recommendation scenario.

4.2.1. Item-based Collaborative Filtering

The system utilises an item based collaborative filtering algorithm. Rather than matching the user to similar customers, item-to-item collaborative filtering matches each of the user’s purchased and rated items to similar items, then combines those similar items into a recommendation list. Cosine similarity and Pearson correlation are the most commonly used method to calculate the similarity scores used for the collaborative filtering process.

4.2.2. Memory-based Approach

Memory-based approach typically uses neighbourhood-based algorithms to calculate the similarity between two users or items, and produces a prediction for the user by taking the weighted average of all the ratings. Similarity computation between items or users is an important part of this approach. Multiple measures, such as Pearson correlation and vector cosine-based similarity are used for this. Despite of its problems with sparse data, it is a highly effective method due to the results being explainable, method itself being easy to use, new data being easier to facilitate into the system, and having good scalability with co-related items.

4.2.3. Cosine Similarity

Cosine similarity is a measure of similarity between two sequences of numbers. The cosine similarity always belongs to the interval [-1, 1]. The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$A.B = \left||A|\right|.\left||B|\right|.\cos\theta$$

Given two n-dimensional vectors of attributes, A and B, the cosine similarity, cos(θ), is represented using a dot product and magnitude as:

$$S_c(A,B) = \cos(\theta) = \frac{A.B}{\left||A|\right| . \left||B|\right|} = \frac{\sum_{i=1}^{n} A_i . B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} . \sqrt{\sum_{i=1}^{n} B_i^2}}$$

where A_i and B_i are components of vector A and B respectively.

4.2.4. Weighted Average

Weighted average is a calculation that takes into account the varying degrees of importance of the numbers in a data set. In calculating a weighted average, each number in the data set is multiplied by a predetermined weight before the final calculation is made.

$$W = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}$$

CHAPTER 5: IMPLEMENTATION

5.1. Tools Used

5.1.1. Development Tools

5.1.1.1. HTML and CSS

HTML is the standard markup language for documents designed to be displayed in a web browser. Similarly, CSS is the most commonly used styling sheet used to style the HTML documents. Both HTML and CSS is used this project to create the basic structure and the user interface components in the views.

5.1.1.2. JS along with jQuery

JS is a widely used scripting language for the web. jQuery is a JS framework designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. This project uses JS and jQuery to provide interactivity and visual appeal of the web pages as well as making asynchronous calls to the database.

5.1.1.3. PHP with Laravel

PHP is a fast, flexible, popular, and pragmatic general-purpose scripting language that is especially suited to web development. Laravel is a web application framework for PHP with expressive, elegant syntax. In his project PHP with Laravel framework is used to implement the back-end logic and the ORM structurers. It is the base of the project that handles all the core logic and algorithms as well as providing a MVC structure for development. The system is built using PHP 8.0 and Laravel 9.47.0.

5.1.1.4. PHP Storm IDE

The system is developed using PHP Storm IDE with student license. It was chosen as it provides a better, leaner, and easier development interface and a better IntelliSense compared to other alternatives. The availability of free student license allowed for the project cost to be reduced.

5.1.1.5. MySQL

MySQL is an open-source relational database management system. The core database of the project is built in MySQL using PHP as PHP offers an easy integration of the MySQL databases.

5.1.1.6. GitHub with Git

Git is a version control system. And GitHub is an online tool that uses git for version control and team collaboration. The combination of these tools was used for project versioning and code collaboration within the project team.

5.1.1.7. XAMPP server

XAMPP is a free and open-source cross-platform web server solution stack package. In this project the XAMPP server allowed the simulation of the Apache server to run the PHP scripts and implement the MySQL database.

5.1.2. Design and Documentation Tools

5.1.2.1. diagrams.net

Diagrams.net is a free and open-source cross-platform graph drawing software that can be used to create diagrams such as flowcharts, wireframes, UML diagrams, organizational charts, and network diagrams. The project uses diagrams.net to prepare the various UML diagrams like use-case diagram and activity diagram.

5.1.2.2. Microsoft Project

Microsoft Project is a project management software designed to assist a project manager in developing a schedule, assigning resources to tasks, tracking progress, managing the budget, and analysing workloads. In this project it is used as the project analysis tool that allowed for the schedule analysis and planning using work breakdown structure and Gantt charts.

5.1.2.3. Microsoft Word

Microsoft Word is a word processor software. It is used during the report and proposal preparation for the project.

5.1.2.4. Microsoft PowerPoint

Microsoft PowerPoint is a slide preparation and presentation software. In this project, it is used for preparing the presentation slides during the proposal submission and defence, mid-term defence, and final defence.

5.2. Database Implementation

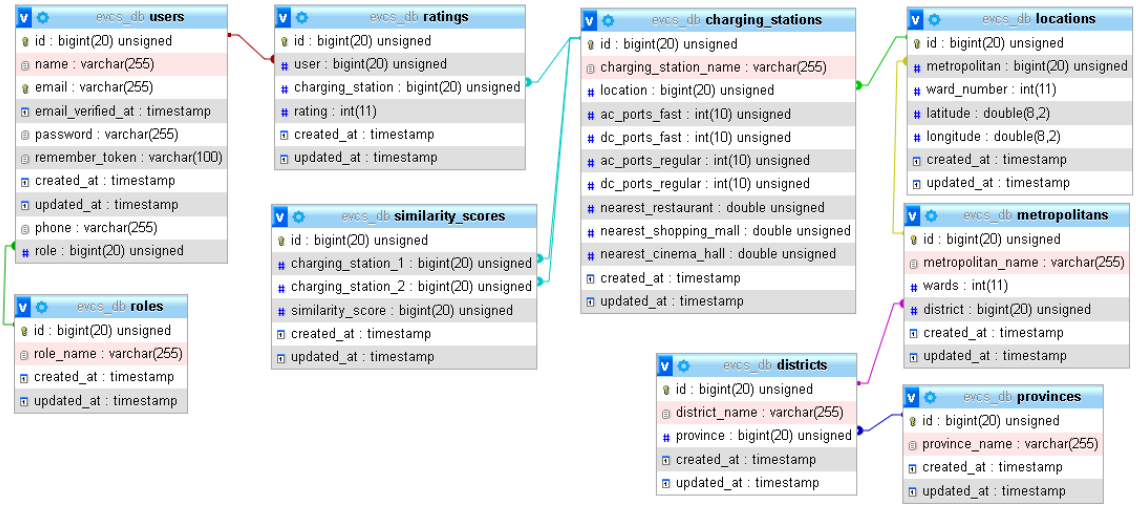


Figure 5.1: Database implementation

5.3. Algorithm Implementation

This system uses item-based approach and a memory-based method to implement collaborative filtering. The system implements vector cosine-based similarity to compute the similarity score required for the memory-based approach. The algorithm is implemented in two phases. The first phase is the computation of similarity scores between two charging stations while the second phase is the execution of the recommendation system using weighted average.

5.3.1. Phase 1: Similarity between charging stations using cosine similarity

The first phase in the implementation of algorithm in the system is the implementation of cosine similarity between the charging stations. Unlike the general approach of using the user ratings itself for the computation of similarity scores to generate a similarity matrix, this system used the attributes of the charging stations itself to factor into the computation of the similarity matrix. The system calculates the similarity scores using the cosine similarity as mentioned in section 4.2.3 of this document.

$$S_c(A,B) = \frac{\vec{A}.\vec{B}}{\left|\left|\vec{A}\right|\right| . \left|\left|\vec{B}\right|\right|} = \frac{\sum_{i=1}^{n} \vec{A}_i . \vec{B}_i}{\sqrt{\sum_{i=1}^{n} \vec{A}_i^2} . \sqrt{\sum_{i=1}^{n} \vec{B}_i^2}}$$

Here, A and B are the two charging stations vectors whose similarity scores are to be computed and S_c is the similarity score obtained. The attributes of the charging stations that contribute to the computation of similarity scorers are:

- Number of Type 1 AC charging ports, AC regular charging ports
- Number of Type 2 AC charging ports, AC fast charging ports
- Number of CHAdeMO charging ports, DC regular charging ports
- Number of Combined Charging System (CCS) charging ports, DC fast charging ports
- Distance from the nearest restaurant
- Distance from the nearest shopping mall
- Distance from the nearest cinema hall

The distances are first transformed into discrete values using a discretisation method. The distances are discretised based on the distance value. The discretisation transformation method is implemented so that the large distance values can be transformed into smaller values so that it is easier to compute. The distance is not as sensitive so the discretisation induced data loss is not a major issue. The discretisation is done as:

Table 5.1: Discretisation of distance

S.N.	Distance	Discretised
1	>500	0
2	>450 and <=500	1
3	>400 and <=450	2
4	>350 and <=400	3
5	>300 and <=350	4
6	>250 and <=300	5
7	>200 and <=250	6
8	>150 and <=200	7
9	>100 and <=150	8
10	>50 and <=100	9
11	>0 and <=50	10
12	=0	0

The dot product of the two vectors is then calculated using the above-mentioned formula. There are 7 attributes involved in the similarity calculation so n = 7. Thus, the above formula can be simplified as:

$$S_c(A,B) = \frac{\sum_{i=1}^{7} \vec{A}_i . \vec{B}_i}{\sqrt{\sum_{i=1}^{7} \vec{A}_i^2} . \sqrt{\sum_{i=1}^{n} \vec{B}_i^2}}$$

If the 7 attributes of $\vec{A}$ is named as $a_1, a_2, a_3, a_4, a_5, a_6,$ and a_7 and that of $\vec{B}$ is named as $b_1, b_2, b_3, b_4, b_5, b_6,$ and b_7 then the similarity score is computed as:

$$S_c(A,B) = \frac{a_1.b_1 + a_2.b_2 + a_3.b_3 + a_4.b_4 + a_5.b_5 + a_6.b_6 + a_7.b_7}{\sqrt{{a_1}^2 + {a_2}^2 + {a_3}^2 + {a_4}^2 + {a_5}^2 + {a_6}^2 + {a_7}^2} . \sqrt{{b_1}^2 + {b_2}^2 + {b_3}^2 + {b_4}^2 + {b_5}^2 + {b_6}^2 + {b_7}^2}}$$

5.3.2. Phase 2: Predicted rating generation using weighted average

The second phase is the implementation of the predicted rating calculation to find which charging station is to be recommended. The system calculates the weighted average using the method as mentioned in section 4.2.4 of this document.

$$W = \frac{\sum_{i=1}^{n} w_i X_i}{\sum_{i=1}^{n} w_i}$$

Here, w_i is the weight value, X_i is the item value and W is the weight obtained after the computation. For this system, the weight is the similarity score between the charging stations. The item value is the rating provided to the charging station by the user. So, if we consider $P(c,u)$ as the predicted rating of charging station c for user u, S_{c,c_i} as the similarity score between the charging station c and c_i, and R_{u,c_i}as the rating provided by user u to charging station c_i then we can rewrite the formula as:

$$P(c,u) = \frac{\sum_{i=1}^{n} S_{c,c_i} . R_{u,c_i}}{\sum_{i=1}^{n} S_{c,c_i}}$$

5.4. Testing

Testing is the process of finding errors and bugs in the system. Testing is the act of examining the elements and the behaviour of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Testing is performed parallelly to coding. If any bugs or errors are found during the testing phase, it is fixed via editing the code or re-coding.

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

Verification is the process of confirming if the software is meeting the business requirements, and is developed adhering to the proper specifications and methodologies.

The system was subject to various scenarios and the outcome was noted. The system was tested using test cases which have been documented as:

Table 5.2: Test case for signup

Test Case ID	TC-1
Test Scenario	Register new user
Actions	1. Open Register page 2. Input the name, email, and password. 3. Click “Register” button.
Input	Name: Ram Dhami Email: ramdhami@gmail.com Password: 23morang@33 Confirm password: 23morang@33
Expected Results	• Redirected to recommendations page. • Name displayed on top right of screen.
Observed Results	As expected
Assertation	Pass

Table 5.3: Test case for login

Test Case ID	TC-2
Test Scenario	Login existing user
Actions	1. Open Login page 2. Input the email and password. 3. Click “Login” button.
Input	Email: ramdhami@gmail.com Password: 23morang@33
Expected Results	• Redirected to recommendations page. • Name displayed on top right of screen.
Observed Results	As expected
Assertation	Pass

Table 5.4: Test case for addition of rating with valid data

Test Case ID	TC-3
Test Scenario	Addition of rating using valid data.
Actions	1. Open rate page. 2. Input the location, the charging station, and rating. 3. Press “Add Rating” button.
Input	Province: Bagmati District: Kathmandu Metropolitan: Kathmandu Ward: 22 Charging Station: Jagat Charging Station Rating: 4
Expected Results	• Success message is displayed. • “ratings” table is updated.
Observed Results	As expected
Assertation	Pass

Table 5.5: Test case for addition of charging station with valid data

Test Case ID	TC-4
Test Scenario	Addition of charging station with valid data.
Actions	1. Open Add Charging Station page. 2. Input the locations and the charging station details. 3. Press "Add Charging Station"
Input	Name: Shiva Shakti Charging Station Province: Bagmati District: Kathmandu Metropolitan: Kathmandu Ward: 30 Fast Charging AC Ports: 2 Fast Charging DC Ports: 3 Regular AC Ports: 4 Regular DC Ports: 4 Nearest Restaurant: 312 Nearest Shopping Mall: 569 Nearest Cinema Hall: 452
Expected Results	• Success message is displayed • Database table "charging_stations" has a new entry. • Redirect to index page.
Observed Results	As expected
Assertation	Pass

Table 5.6: Test case for addition of charging station with invalid data

Test Case ID	TC-5
Test Scenario	Addition of charging station using invalid data.
Actions	1. Open Add Charging Station page. 2. Input the locations and the charging station details. 3. Press "Add Charging Station" button.
Input	Name: Jagat Charging Station Province: Bagmati District: Kathmandu Metropolitan: Kathmandu Ward: 22 Fast Charging AC Ports: 2 Fast Charging DC Ports: 3 Regular AC Ports: 4 Regular DC Ports: 4 Nearest Restaurant: 312 Nearest Shopping Mall: 569 Nearest Cinema Hall: -1
Expected Results	• System does not allow the invalid data and error message is displayed
Observed Results	As expected
Assertation	Pass

Table 5.7: Test case for getting recommendation without ward

Test Case ID	TC-6
Test Scenario	Get recommendation of charging station without inputting ward.
Actions	1. Open Recommendations page. 2. Input the location and check the Exclude Ward checkbox. 3. Press “Get recommendation” button.
Input	Province: Bagmati District: Kathmandu Metropolitan: Kathmandu
Expected Results	• Top 3 charging stations displayed.
Observed Results	As expected
Assertation	Pass

Table 5.8: Test case for getting recommendation without ward

Test Case ID	TC-7
Test Scenario	Get recommendation of charging station.
Actions	1. Open Recommendations page. 2. Uncheck the Exclude Ward checkbox. 3. Input the location. 4. Press “Get recommendation” button.
Input	Province: Bagmati District: Kathmandu Metropolitan: Kathmandu Ward: 3
Expected Results	• Top 3 charging stations displayed.
Observed Results	As expected
Assertation	Pass

CHAPTER 6: CONCLUSION AND FUTURE RECOMMENDATIONS

6.1. Conclusion

The product of this work, Electric Vehicle Charging Station Recommendation System, is a web-based application that provides a platform that recommends EV charging stations to the user in their desired location. The memory-based approach of cosine similarity and weighted average allows the system to have decent scalability on co-related items. The system has functional authentication, similarity calculation during insertion, rating, and recommendation. The system is able to recommend the charging station in the said location and the algorithm functions as intended. Existing recommendation systems had a problem of specialisation. They were either too technical and thus overlooked the human aspect as EV charging is not a quick endeavour, or overlooked the technical aspect and were recommending based on ratings of users alone, which in times recommended incompatible charging stations. The system incorporates both the technical and human aspects of the charging station to provide a better recommendation for the users.

6.2. Future Recommendations

In spite of the system having good performance, the system could still be improved upon. The system can incorporate maps and use dynamic distance computation to provide a more dynamic system to the users as well as the admin. PHP is not well suited for implementation of high-end ML and AI algorithms. A change in the choice of the backend framework could allow implementation of a hybrid filtering approach, despite being highly complex. Finally, in the era of real-time technology, the system could be subject to further improvements by adding real-time booking and scheduling of the EV charging station ports.

REFERENCES AND BIBLIOGRAPHY

[1] R. Urooj and K. Annamma, “Persistent Charging Station Recommendation System for Electric-Vehicle Taxis,” International Journal of Scientific Engineering and Technology Research, Hyderabad, India, 2017.

[2] W. Zhang, H. Liu, F. Wang, T. Xu, H. Xin, D. Dou and H. Xiong, “Intelligent Electric Vehicle Charging Recommendation Based on Multi-Agent Reinforcement Learning,” International World Wide Web Conference Committee, Ljubljana, Slovenia, 2021.

[3] R.Aarthi and P. Prasath, “Enhanced Real-Time Charging Station Recommendation System For Load Base Electric-Vehicle Taxis,” South Asian Journal of Engineering and Technology, 2017.

[4] X. Wang, X. Zheng and X. Liang, “Charging Station Recommendation for Electric Vehicle Based on Federated Learning,” Artificial Intelligence on Electric Power System State Grid Corporation Joint Laboratory, Beijing, China, 2021.

[5] F. Ricci, L. Rokach and B. Shapira, Introduction to Recommender Systems Handbook, 2011.

[6] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” 2009.

[7] A. Singhal, “Modern Information Retrieval: A Brief Overview,” Google, Inc..

APPENDIX

Snippets of major source code components

Computation of similarity scores

```
function calculateSimilarityScores($cs1, $cs2) {
   $chargingStationModel = new ChargingStation();
   $cs_att_1 = $chargingStationModel->getChargingStationAttributes($cs1);
   $cs_att_2 = $chargingStationModel->getChargingStationAttributes($cs2);
   $ab = $cs_att_1[0]->ac_ports_fast * $cs_att_2[0]->ac_ports_fast +
      $cs_att_1[0]->dc_ports_fast * $cs_att_2[0]->dc_ports_fast +
      $cs_att_1[0]->ac_ports_regular * $cs_att_2[0]->ac_ports_regular +
      $cs_att_1[0]->dc_ports_regular * $cs_att_2[0]->dc_ports_regular +
      $this->distance_scale($cs_att_1[0]->nearest_restaurant) *
         $this->distance_scale($cs_att_2[0]->nearest_restaurant) +
      $this->distance_scale($cs_att_1[0]->nearest_shopping_mall) *
         $this->distance_scale($cs_att_2[0]->nearest_shopping_mall) +
      $this->distance_scale($cs_att_1[0]->nearest_cinema_hall) *
         $this->distance_scale($cs_att_2[0]->nearest_cinema_hall);
   $aSquared = $cs_att_1[0]->ac_ports_fast * $cs_att_1[0]->ac_ports_fast +
      $cs_att_1[0]->dc_ports_fast * $cs_att_1[0]->dc_ports_fast +
      $cs_att_1[0]->ac_ports_regular * $cs_att_1[0]->ac_ports_regular +
      $cs_att_1[0]->dc_ports_regular * $cs_att_1[0]->dc_ports_regular +
      $this->distance_scale($cs_att_1[0]->nearest_restaurant) *
         $this->distance_scale($cs_att_1[0]->nearest_restaurant) +
      $this->distance_scale($cs_att_1[0]->nearest_shopping_mall) *
         $this->distance_scale($cs_att_1[0]->nearest_shopping_mall) +
      $this->distance_scale($cs_att_1[0]->nearest_cinema_hall) *
         $this->distance_scale($cs_att_1[0]->nearest_cinema_hall);
   $bSquared = $cs_att_2[0]->ac_ports_fast * $cs_att_2[0]->ac_ports_fast +
      $cs_att_2[0]->dc_ports_fast * $cs_att_2[0]->dc_ports_fast +
      $cs_att_2[0]->ac_ports_regular * $cs_att_2[0]->ac_ports_regular +
      $cs_att_2[0]->dc_ports_regular * $cs_att_2[0]->dc_ports_regular +
      $this->distance_scale($cs_att_2[0]->nearest_restaurant) *
         $this->distance_scale($cs_att_2[0]->nearest_restaurant) +
```

```
        $this->distance_scale($cs_att_2[0]->nearest_shopping_mall) *
            $this->distance_scale($cs_att_2[0]->nearest_shopping_mall) +
        $this->distance_scale($cs_att_2[0]->nearest_cinema_hall) *
            $this->distance_scale($cs_att_2[0]->nearest_cinema_hall);
    if ($aSquared == 0 || $bSquared == 0) {
        $similarityScore = 0;
    } else {
        $similarityScore = $ab / (sqrt($aSquared) * sqrt($bSquared));
    }
    return $similarityScore;
}
```

Computation of missing rating and recommendation

```
function getRecommendation(Request $request) {
    $ratingsModel = new Ratings();
    $chargingStationModel = new ChargingStation();
    $recommendationRating = [0, 0, 0];
    $recommendations = [0, 0, 0];
    $user_rating = $ratingsModel->userRatings();
    $chargingStationLocation = '';
    if($user_rating->count() <= 3) {
        request()->session()->flash('error', 'Please rate at least 3 charging stations.');
        return redirect()->route('recommendations.index');
    } else {
        if($request->get('ward_enabled') == 0) {
            $chargingStationLocation =
                $chargingStationModel->getChargingStationNoWard($request);
        } elseif ($request->get('ward_enabled') == 1) {
                $chargingStationLocation =
            $chargingStationModel->getChargingStationWard($request);
        }
        foreach ($chargingStationLocation as $csl) {
            $ratingEstimateNum = 0;
            $ratingEstimateDen = 0;
```

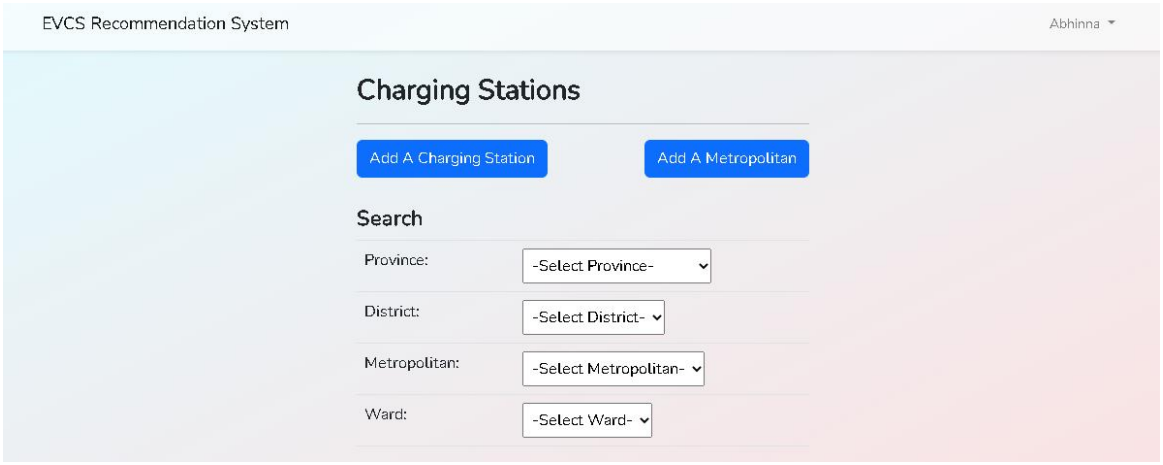
```
        foreach ($user_rating as $ur) {
            $similarityScore = $this->calculateSimilarityScores(
                $ur->charging_station, $csl->charging_station);
            $ratingEstimateNum = $ratingEstimateNum + $ur->rating * $similarityScore;
            $ratingEstimateDen += $similarityScore;
        }
        if ($ratingEstimateDen == 0) {
            $ratingEstimate = 0;
        } else {
            $ratingEstimate = $ratingEstimateNum / $ratingEstimateDen;
        }
        if ($ratingEstimate >= $recommendationRating[0]) {
            $recommendationRating[2] = $recommendationRating[1];
            $recommendationRating[1] = $recommendationRating[0];
            $recommendationRating[0] = $ratingEstimate;
            $recommendations[2] = $recommendations[1];
            $recommendations[1] = $recommendations[0];
            $recommendations[0] = $csl->charging_station;
        } elseif ($ratingEstimate >= $recommendationRating[1]) {
            $recommendationRating[2] = $recommendationRating[1];
            $recommendationRating[1] = $ratingEstimate;
            $recommendations[2] = $recommendations[1];
            $recommendations[1] = $csl->charging_station;
        } elseif ($ratingEstimate >= $recommendationRating[2]) {
            $recommendationRating[2] = $ratingEstimate;
            $recommendations[2] = $csl->charging_station;
        }
    }
}
$provinceModel = new Provinces();
$data['provinces'] = $provinceModel->selectProvinces();
$data['user'] = Auth::id();
$recommendation1 =
    $chargingStationModel->getFinalRecommendation($recommendations[0]);
```

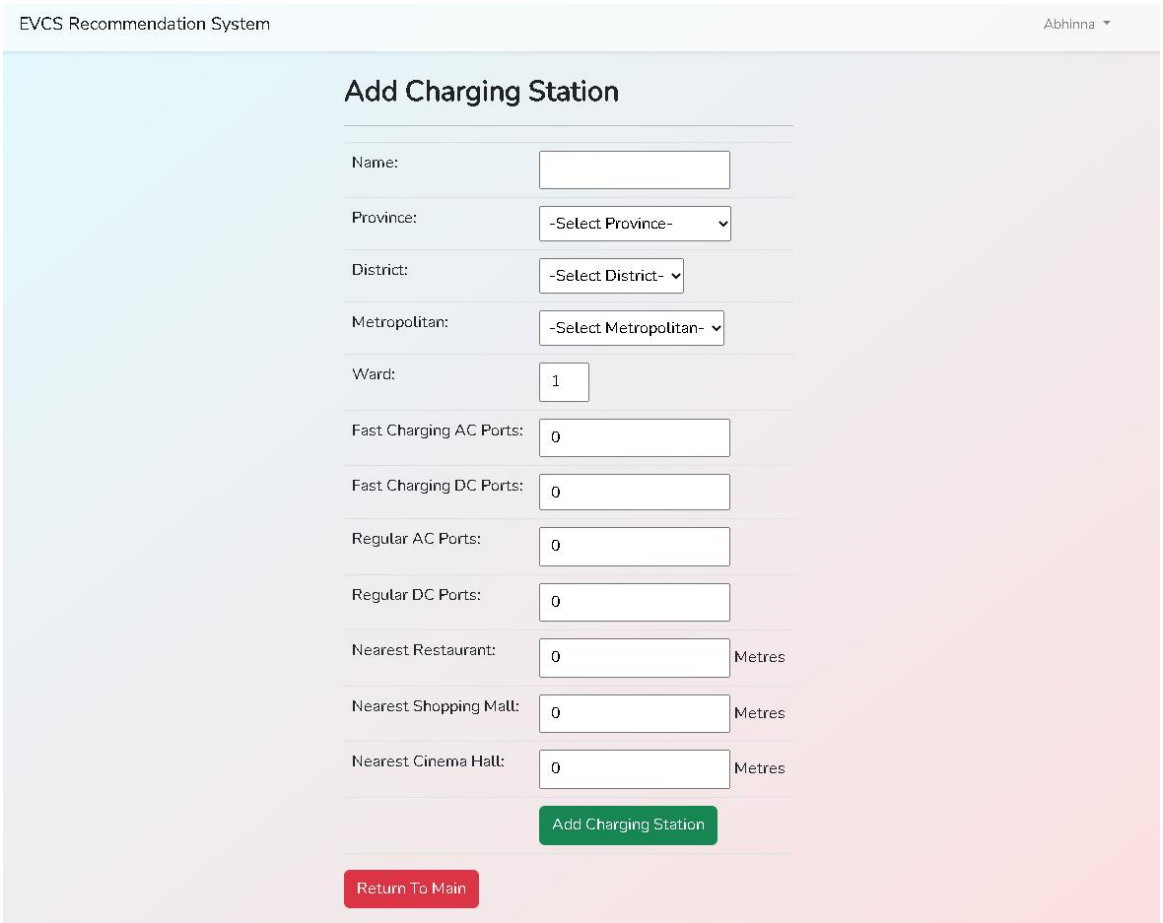
```
    $recommendation2 =
        $chargingStationModel->getFinalRecommendation($recommendations[1]);
    $recommendation3 =
        $chargingStationModel->getFinalRecommendation($recommendations[2]);
    $data['recommendations'] =
        collect([$recommendation1, $recommendation2, $recommendation3]);
    $data['actual_cs'] = $recommendations;
    $data['estimated_rating'] = $recommendationRating;
    return view('recommend.recommend', compact('data'));
}
```

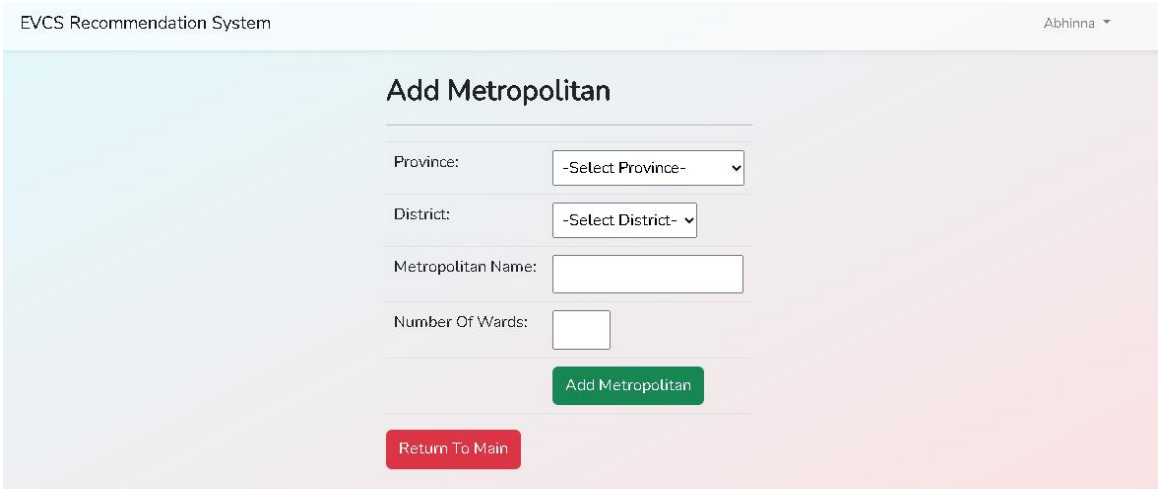
Discretisation of Distances

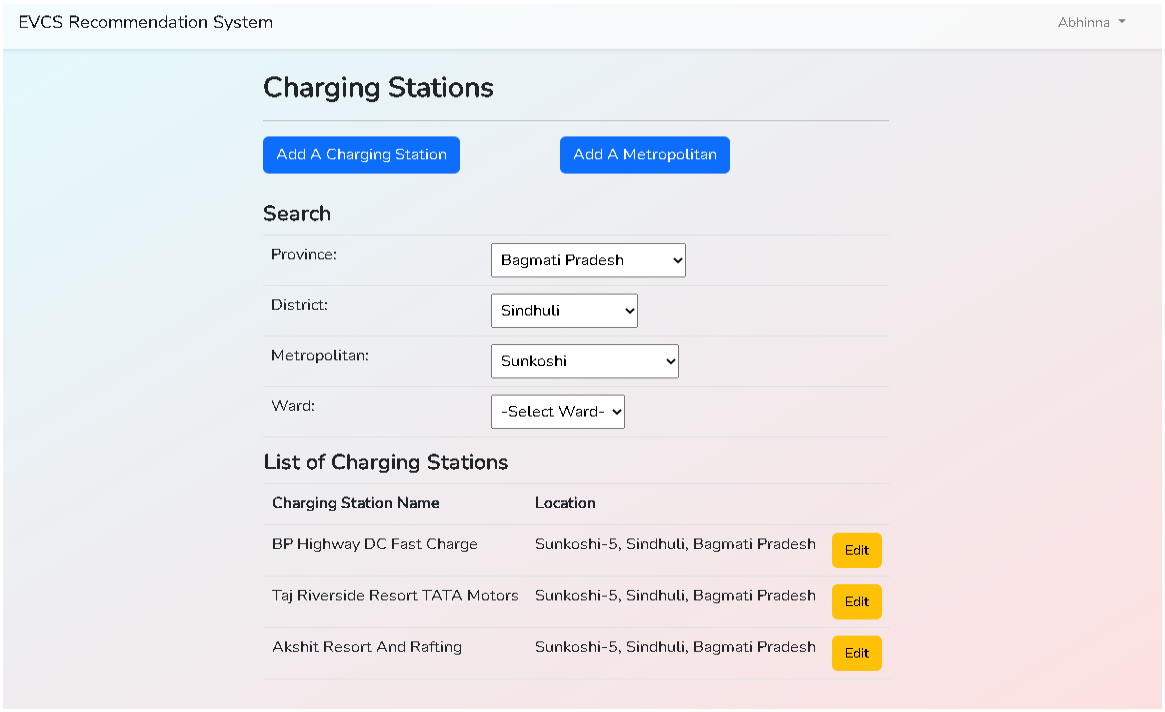
```
function distance_scale($distance_str) {
    $distance = (float)$distance_str;
    if ($distance == 0) {return 0;}
    else if ($distance <= 50) {return 10;}
    else if ($distance <= 100) {return 9;}
    else if ($distance <= 150) {return 8;}
    else if ($distance <= 200) {return 7;}
    else if ($distance <= 250) {return 6;}
    else if ($distance <= 300) {return 5;}
    else if ($distance <= 350) {return 4;}
    else if ($distance <= 400) {return 3;}
    else if ($distance <= 450) {return 2;}
    else if ($distance <= 500) {return 1;}
    else {return 0;}
}
```

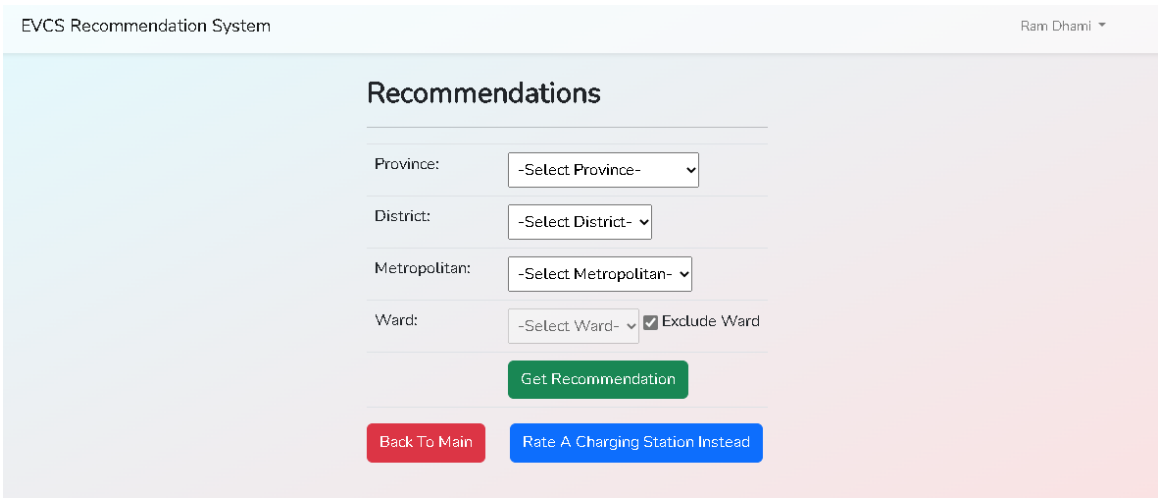
Screenshots

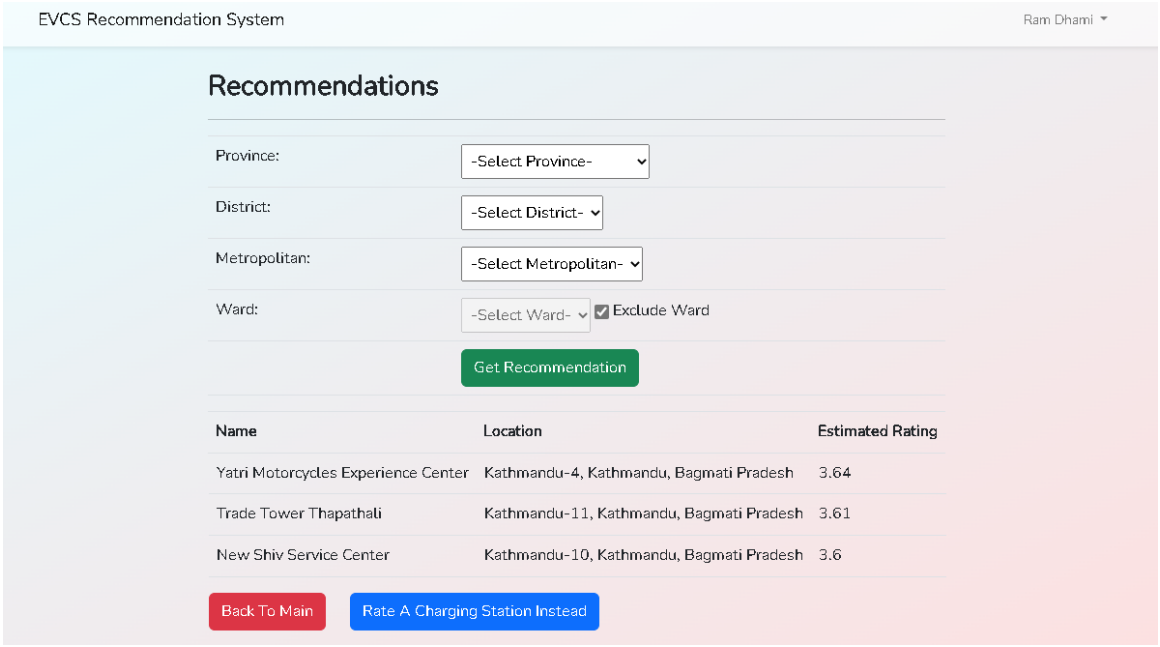





EVCS Recommendation System
Abhinna
Add Metropolitan
Province:
-Select Province-
District:
-Select District-
Metropolitan Name:
Number Of Wards:
Add Metropolitan
Return To Main


EVCS Recommendation System
Abhinna
Charging Stations
Add A Charging Station
Add A Metropolitan
Search
Province:
Bagmati Pradesh
District:
Sindhuli
Metropolitan:
Sunkoshi
Ward:
-Select Ward-
List of Charging Stations
Charging Station Name
Location
BP Highway DC Fast Charge
Sunkoshi-5, Sindhuli, Bagmati Pradesh
Edit
Taj Riverside Resort TATA Motors
Sunkoshi-5, Sindhuli, Bagmati Pradesh
Edit
Akshit Resort And Rafting
Sunkoshi-5, Sindhuli, Bagmati Pradesh
Edit


EVCS Recommendation System
Ram Dhami
Recommendations
Province:
-Select Province-
District:
-Select District-
Metropolitan:
-Select Metropolitan-
Ward:
-Select Ward-
Exclude Ward
Get Recommendation
Back To Main
Rate A Charging Station Instead


EVCS Recommendation System
Ram Dhami
Recommendations
Province:
-Select Province-
District:
-Select District-
Metropolitan:
-Select Metropolitan-
Ward:
-Select Ward-
Exclude Ward
Get Recommendation
Name
Location
Estimated Rating
Yatri Motorcycles Experience Center
Kathmandu-4, Kathmandu, Bagmati Pradesh
3.64
Trade Tower Thapathali
Kathmandu-11, Kathmandu, Bagmati Pradesh
3.61
New Shiv Service Center
Kathmandu-10, Kathmandu, Bagmati Pradesh
3.6
Back To Main
Rate A Charging Station Instead