**EV PLUG**

*Project Report Submitted by*

**Susan Sebastian**

**Reg. No.: AJC21MCA-2105**

*In Partial fulfillment for the Award of the Degree of*

**MASTER OF COMPUTER APPLICATIONS**

**(MCA TWO YEAR)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

# 2021-2023

## DEPARTMENT OF COMPUTER APPLICATIONS

### AMAL JYOTHI COLLEGE OF ENGINEERING

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, **“EV PLUG”** is the bonafide work of **SUSAN SEBASTIAN (Regno: AJC21MCA-2105)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2022-23.

**Ms. Merin Chacko Ms. Meera Rose Mathew**

**Internal Guide Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department External Examiner**

**// CERTIFICATE ON PLAGIARISM CHECK**

**DECLARATION**

I hereby declare that the project report **“EV PLUG”** is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

**Date: SUSAN SEBASTIAN**

**KANJIRAPPALLY Reg: AJC21MCA-2105**

# ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Merin Chacko** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

SUSAN SEBASTIAN

# ABSTRACT

.

In today’s world we can see a drastic raise in the usage of electric vehicles. Due to the high expense of fuel and inflation people are forced to use the electric vehicles even for commuting long distance. Think about a scenario where 10 people are commuting on there EV and they need to charge their vehicles on their way. Since there’re only limited number of chargers available in our state, we need to make sure that the charger is free when we reach the charging place. What if all the chargers are used by some other vehicles. We need to wait till a port is free and our time is precious. This is the scenario where this EVPLUG project has an importance. Using this application, we can view what’s the closest charging station, how many free ports are available and when does these used ports may free. We can even select a time slot to book and arrange our travel based on that. We can create a wallet that can store points that may add up at each charge thus save a few bucks. This will ensure people to use only our application in the future to save more and thus makes profit for the website too.

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL. NO** | | **TOPIC** | **PAGE NO** | |
| **1** | | **INTRODUCTION** |  | |
| **1.1** | | **PROJECT OVERVIEW** |  | |
| **1.2** | | **PROJECT SPECIFICATION** |  | |
| **2** | | **SYSTEM STUDY** |  | |
| **2.1** | | **INTRODUCTION** |  | |
| **2.2** | | **EXISTING SYSTEM** |  | |
| **2.3** | | **DRAWBACKS OF EXISTING SYSTEM** |  | |
| **2.4** | | **PROPOSED SYSTEM** |  | |
| **2.5** | | **ADVANTAGES OF PROPOSED SYSTEM** |  | |
| **3** | | **REQUIREMENT ANALYSIS** |  | |
| **3.1** | | **FEASIBILITY STUDY** |  | |
| **3.1.1** | | **ECONOMICAL FEASIBILITY** |  | |
| **3.1.2** | | **TECHNICAL FEASIBILITY** |  | |
| **3.1.3** | | **BEHAVIORAL FEASIBILITY** |  | |
| **3.1.4** | | **FEASIBILITY STUDY QUESTIONNAIRE** |  | |
| **3.2** | | **SYSTEM SPECIFICATION** |  | |
| **3.2.1** | | **HARDWARE SPECIFICATION** |  | |
| **3.2.2** | | **SOFTWARE SPECIFICATION** |  | |
| **3.3** | | **SOFTWARE DESCRIPTION** |  | |
| **3.3.1** | | **PHP** |  | |
| **3.3.2** | | **MYSQL** |  | |
| **4** | | **SYSTEM DESIGN** |  | |
| **4.1** | | **INTRODUCTION** |  | |
| **4.2** | | **UML DIAGRAM** |  | |
| **4.2.1** | | **USE CASE DIAGRAM** |  | |
| **4.2.2** | | **SEQUENCE DIAGRAM** |  | |
| **4.2.3** | | **STATE CHART DIAGRAM** |  | |
| **4.2.4** | | **ACTIVITY DIAGRAM** |  | |
| **4.2.5** | | **CLASS DIAGRAM** |  | |
| **4.2.6** | | **OBJECT DIAGRAM** |  | |
| **4.2.7** | | **COMPONENT DIAGRAM** |  | |
| **4.2.8** | | **DEPLOYMENT DIAGRAM** |  | |
| **4.2.9** | | **COLLABORATION DIAGRAM** |  | |
| **4.3** | | **USER INTERFACE DESIGN USING FIGMA** |  | |
| **4.4** | | **DATABASE DESIGN** |  | |
| **5** | | **SYSTEM TESTING** |  | |
| **5.1** | | **INTRODUCTION** |  | |
| **5.2** | | **TEST PLAN** |  | |
| **5.2.1** | **UNIT TESTING** | |  |
| **5.2.2** | **INTEGRATION TESTING** | |  |
| **5.2.3** | **VALIDATION TESTING** | |  |
| **5.2.4** | **USER ACCEPTANCE TESTING** | |  |
| **5.2.5** | **AUTOMATION TESTING** | |  |
| **5.2.6** | **SELENIUM TESTING** | |  |
| **6** | **IMPLEMENTATION** | |  |
| **6.1** | **INTRODUCTION** | |  |
| **6.2** | **IMPLEMENTATION PROCEDURE** | |  |
| **6.2.1** | **USER TRAINING** | |  |
| **6.2.2** | **TRAINING ON APPLICATION SOFTWARE** | |  |
| **6.2.3** | **SYSTEM MAINTENANCE** | |  |
| **6.2.4** | **HOSTING** | |  |
| **7** | **CONCLUSION & FUTURE SCOPE** | |  |
| **7.1** | **CONCLUSION** | |  |
| **7.2** | **FUTURE SCOPE** | |  |
| **8** | **BIBLIOGRAPHY** | |  |
| **9** | **APPENDIX** | |  |
| **9.1** | **SAMPLE CODE** | |  |
| **9.2** | **SCREEN SHOTS** | |  |

## List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language.

CSS - Cascading Style Sheet

SQL - Structured Query Language

UML - Unified Modeling Language

EV - Electric Vehicle

# CHAPTER 1

# INTRODUCTION

### PROJECT OVERVIEW

The popularity of electric vehicles depends on convenient and efficient charging services. Currently, none of the existing charging services offer the possibility of reaching the charging station at the desired time and charging directly after arrival without waiting. EVPLUG offers a top-up booking service approach. Users can select a charging station and a congestion-free charging time based on charging information from stations in scattered regions of the city. Dial to ensure the user arrives at the charging station on time and charges on time according to the reservation.This approach can not only greatly reduce user waiting times, but also provide very convenient top-up services for smart users.

### PROJECT SPECIFICATION

The proposed system is a web application that allows the user to see the charging station at a specific location, how many free ports are available and when to free up those ports that are in use. We can even select a time slot to book and plan our trip accordingly.

The system includes 3 modules. They are:

 Admin

The site admin has an overall control on the website. He can perform the functionalities like,

* Login to the website
* Add/View/Update/Delete station/manager/service category/electricity price
* View registered customers
* Manage Profile and password
* Manage available Electric Vehicle Brand and Models
* View and download reports
* View analysis of bookings and feedback of this website.
* Manager

Each charging station is a separate unit aligned with the EV Plug website. There is a

person to manage their respective charging stations. This can include

addition/removal of charging ports,

addition/removal timeslot,

Viewing daily or monthly report

QRCode Scanner and OTP verification,

Apply/Manage Leaves

Chance of fire prediction at station.

* Customer

Customer is the prime user who select and book stations in their day-to-day life

Customer can perform functionalities like,

* Registration/Login and view/manage profile/change password
* Add/View user’s vehicles based on brand names, based on model names etc.
* Search station using location.
* Book the slot in the station by choosing date and time.
* View the bookings so far and cancel it.

# CHAPTER 2

# SYSTEM STUDY

### INTRODUCTION

Systems analysis is the process of gathering and interpreting facts, diagnosing problems and information to recommend system improvements. This is a debugging activity that requires extensive communication between system users and system developers. The system analysis or study is an important phase of any system development process. The system is thoroughly examined and analyzed. The systems analyst takes on the role of an interrogator and examines the functioning of the current system.The system is considered as a whole and the system inputs are identified. The performance of the organization is linked to several processes. Systems analysis is about becoming aware of the problem, identifying the relevant and determining variables, analyzing and synthesizing various factors, and identifying the optimal, or at least satisfactory, solution or course of action. A detailed investigation of this process should be done using various techniques such as interviews, questionnaires, etc. The data collected from these sources must be analyzed in order to draw conclusions.The key is to understand how the system works. This system is referred to as the existing system. The current system is thoroughly reviewed and problem areas identified. The designer now acts as a problem solver and tries to solve the company's difficulties. The solutions are for information only.The proposal is then analyzed against the existing system and the best one is selected. The proposal is submitted to the user for user approval. At the request of the user, the proposal is checked and the appropriate changes are made. It's a cycle that ends when the user is happy with the proposal. Preliminary research is the process of gathering and interpreting facts and using the information to further study the system.Pref-lighting is a troubleshooting activity that requires extensive communication between system users and system developers. Conducting various feasibility studies. In these studies, it is possible to get a high-level picture of system activities, which can be used to decide what strategies to use for effective investigation and analysis of the system.

### EXISTING SYSTEM

**2.2.1 NATURAL SYSTEM STUDIED**

In today’s world we can see a drastic raise in the usage of electric vehicles. Due to the high expense of fuel and inflation people are forced to use the electric vehicles even for commuting long distance. Think about a scenario where 10 people are commuting on their EV and they need to charge their vehicles on their way. Since there’re only limited number of chargers available in our state, we need to make sure that the charger is free when we reach the charging place. What if all the chargers are used by some other vehicles. We need to wait till a port is free and our time is precious.

As per the existing system there are few numbers of charging station available across the state.

Whenever, an EV user came to charge their vehicle what they usually do is directly plug in the charger to the EV if the post is free. The market share of EV is very low compared to fossil fuel vehicles right now. So, there will not be any pay off caused due to the rush to charge this vehicle. EV are charged in two types:

The user goes for a quick charge which their vehicle can be charge up to 50% approximately in less than 30 minutes. Whereas the full charge will take nearly 1 hour. There is a drawback to this system whenever any user come to charge their vehicle and if the charging port is occupied by another user, then he/she has to wait until it gets free. This may cause in wastage of their time.

To overcome this issue and plan a travel using the electric vehicle more conveniently the proposed system is introduced.

### DRAWBACKS OF EXISTING SYSTEM

* Cannot be able to prebook charging slot
* Couldn’t be able to travel long distance using EV
* Cannot prebook any kind of services
* Wastage of time

### PROPOSED SYSTEM

**2.4.1 DESIGNED SYSTEM STUDIED**

The proposed system is a web application is known as EVPLUG is developed using HTML, JavaScript and PHP.In the proposed system, if someone wants to travel long distance in an EV, he/she definitely need to recharge their vehicle on the way. Then he/she can register and check at what time the charging slot is available. If a port is available to his/her convenient time, then they can book a port with an advance payment and make sure that port will be empty upon the user reaches the station on time. This will avoid the wastage of time which the user may spend on waiting for a port. Thus, the travel will be more comfortable. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work. Travelers do not need to wait for anymore without knowing the details of the stations in a particular place. They can ensure whether there will be port without any further interventions.

### ADVANTAGES OF PROPOSED SYSTEM

* **Able to know the station location.**

Our customers will get the opportunity to know about the station available in particular area.

* **Schedule the travel by booking a port in advance.**

Our customers will be able to book charging ports when they plan a travel in an EV.

* **Better Security.**

For data to remain secure measures must be taken to prevent unauthorized access. Security

means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

* **Ensure data accuracy**

The proposed system eliminates the manual errors while entering the details of the users during the registration

* **User friendliness and interactive**

The proposed system’s interface helps the users to perform their operations without any confusions or difficulties. A customer can easily find their bus which has to travel and book a seat.

# CHAPTER 3

# REQUIREMENT ANALYSIS

## FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its ability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development. The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibility. The following are its features: -

### Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

*  The costs conduct a full system investigation.
*  The cost of the hardware and software.
*  The benefits in the form of reduced costs or fewer costly errors

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

The cost of project, EVPLUG was divided according to the system used, its development cost. According to all the calculations the project was developed in a low cost. As it is completely developed using open-source software.

### Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

 Does the existing technology sufficient for the suggested one?

 Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So, there are minimal constraints involved with this project. The system has been developed using HTML, CSS in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using HTML, CSS, PHP and MySQL in server back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core, RAM 4GB and, Hard disk 1TB

### Behavioral Feasibility

The proposed system includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

### Feasibility Study Questionnaire

### How long does it usually take to charge an EV?

That depends on the type of charging station you use, your EV's battery capacity, and how much you drive. Regardless of the type of charging station, the speed at which an EV battery charges is always limited by the maximum amount of power it can handle.

### How fast does my EV go?

### It depends on the model. The average top speed of an electric vehicle is 180 kilometers per hour, but some can reach up to 320 kilometers per hour.

1. What are the problems we’re solving by using EV charging station pre booking site?

* Saves Time: The application allows you to book the charging slot in advance and saves your time from waiting in the queue.
* Emergency: It is needless to point out that the app will locate the nearby stations at the time of urgency.
* Easy Transaction: You can easily transact the amount and keep the history of transactions in the app itself.

1. Does Every EV uses same type of battery?

* The lead-acid battery.
* The nickel-cadmium battery
* The nickel-metal hydride battery
* The lithium-ion battery
* The solid-state battery

Lithium-ion batteries are currently the most widely-used

1. What are the EV branches and their Models in India?

Tata Motors  - [Tigor EV](https://e-vehicleinfo.com/tata-tigor-ev-price-in-india-launch-date-feature-highlights/) and Tiago EV

Mahindra Motors - eVerito, eSupro, Treo, eAlfa Mini, and e2oPlus

MG Motor - MG Hector, Gluster, ZS EV

1. How long does the manufacturer provide a warranty over a battery and how long does it last?

A car battery warranty typically covers for 24 months or 36,000 miles from the date of purchase of the battery. Some warranties have a cover of even 5 years. The warranty can be part of the car insurance, or you can purchase it separately. Most manufacturers have a five to eight-year warranty on their battery. However, the current prediction is that an electric car battery will last from 10 – 20 years before they need to be replaced.

1. How many EV stations are currently working in India?

India currently has 2,826 public charging stations that are operational in the country,

1. What are the payment methods used in existing charging stations?

There are a variety of options for open EV charging payments, including bank cards (credit/debit), smartphone wallets, and eCommerce (website payments/QR codes)

1. How is the electricity generated in charging station?

They’re connected to the larger municipal grid and then you essentially pay the owner of the charging station to hook up through their outlet to that larger power source (grid). Sometimes the charging stations use other energy sources that the owner of the station has utlized. Sometimes these can be solar, natural gas, or other sources.These chargers are powered by renewable energy sources like wind, solar, and hydropower.

1. What is the average cost is used in existing charging station ?

Charging a vehicle between 10 pm and 6 am will cost Rs 10 a unit,while it will be Rs 12 between 6 am and 6 pm. During the peak 6 pm to 10 pm, the rate per unit will be Rs 15.

## 3.2 SYSTEM SPECIFICATION

### Hardware Specification

Processor - Intel core i3

RAM - 4GB

Hard disk - 1TB

### Software Specification

Front End - HTML, CSS

Backend - MYSQL,

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

## SOFTWARE DESCRIPTION

### Eg.PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1994, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym code is interpreted by a web server with a PHP processor module which generates the resulting web page commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP.PHP can be deployed on most web servers and as a standalone shell on almost every operating system and platform, free of charge.

### Eg. MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software

**MySQL is a database management system**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

**MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well- designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

**MySQL software is Open Source*.***

Open-Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

**The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

**MySQL Server works in client/server or embedded systems*.***

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

**A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

# CHAPTER 4

# SYSTEM DESIGN

* 1. **INTRODUCTION**

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel.

## UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language, but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.

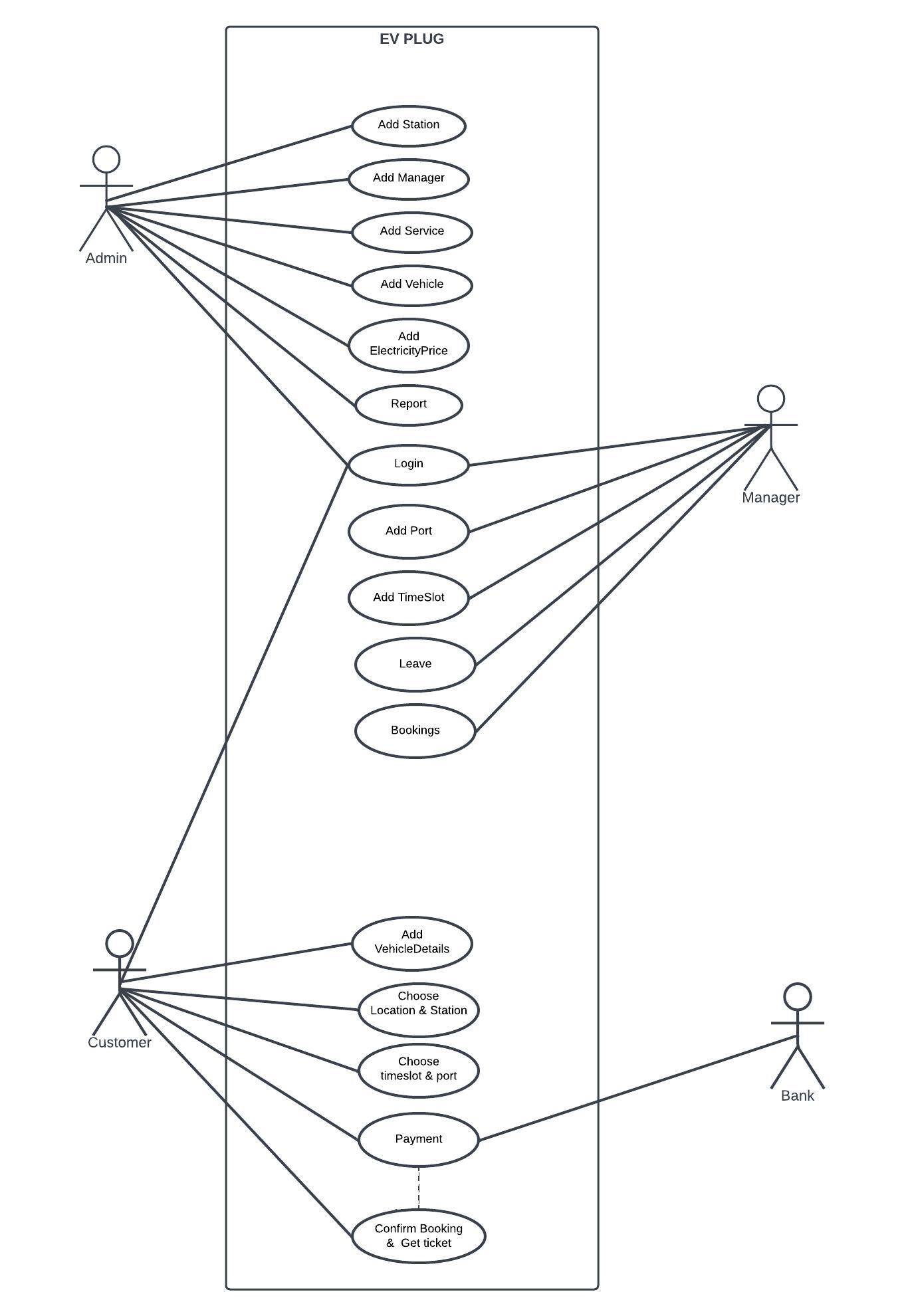
UML includes the following nine diagrams.

*  Activity diagram
*  Use case diagram
*  Sequence diagram
*  Collaboration diagram
*  State chart diagram
*  Class diagram
*  Object diagram
*  Deployment diagram

## USE CASE DIAGRAM

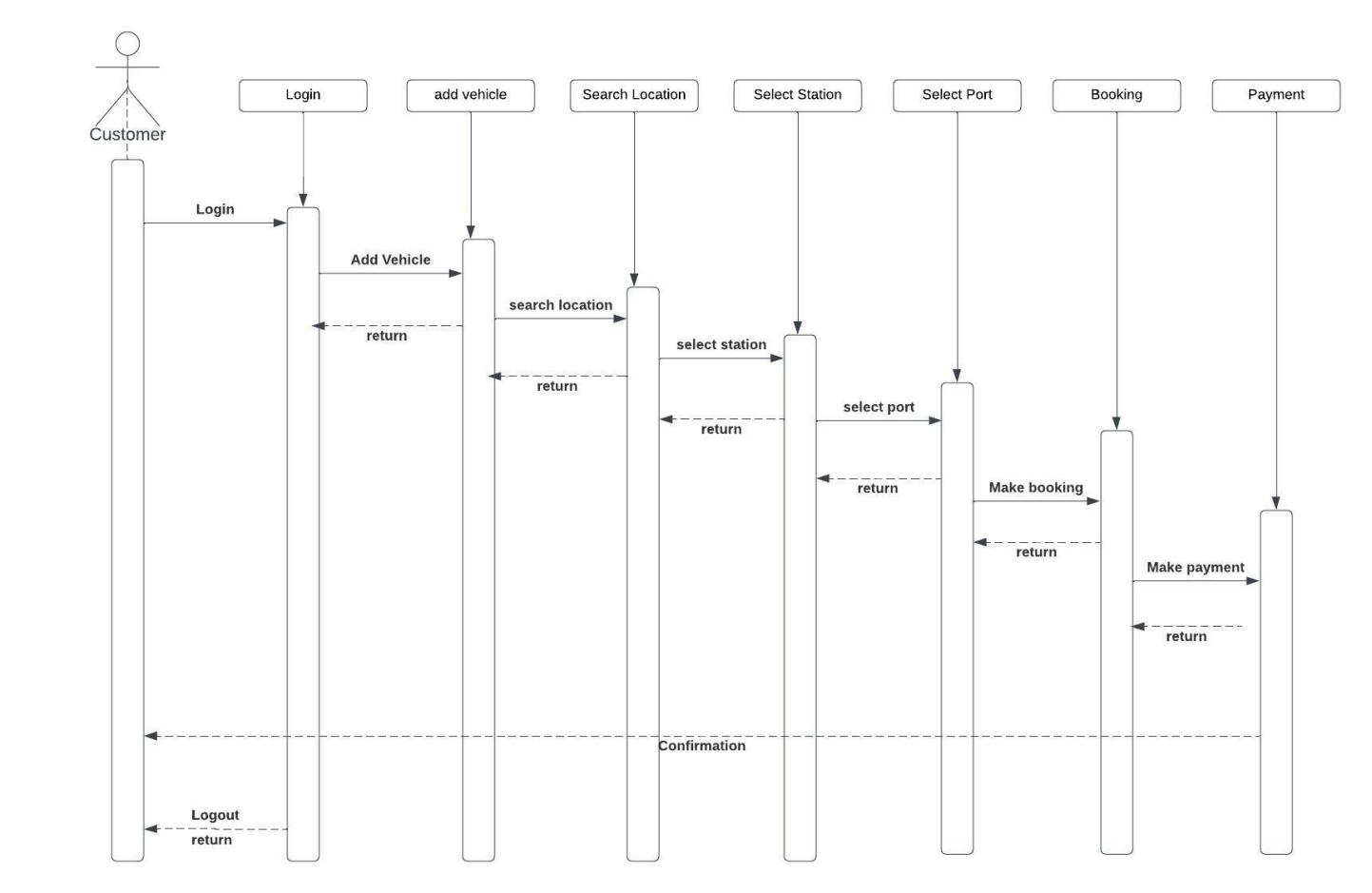
A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real- world objects and systems. System objectives can include planning overall requirements, validating a hardware design,

testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components. The boundary, which defines the system of interest in relation to the world around it. The actors, usually individuals involved with the system defined according to their roles. The use cases, which the specific roles are played by the actors within and around the system. The relationships between and among the actors and the use cases.



## SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



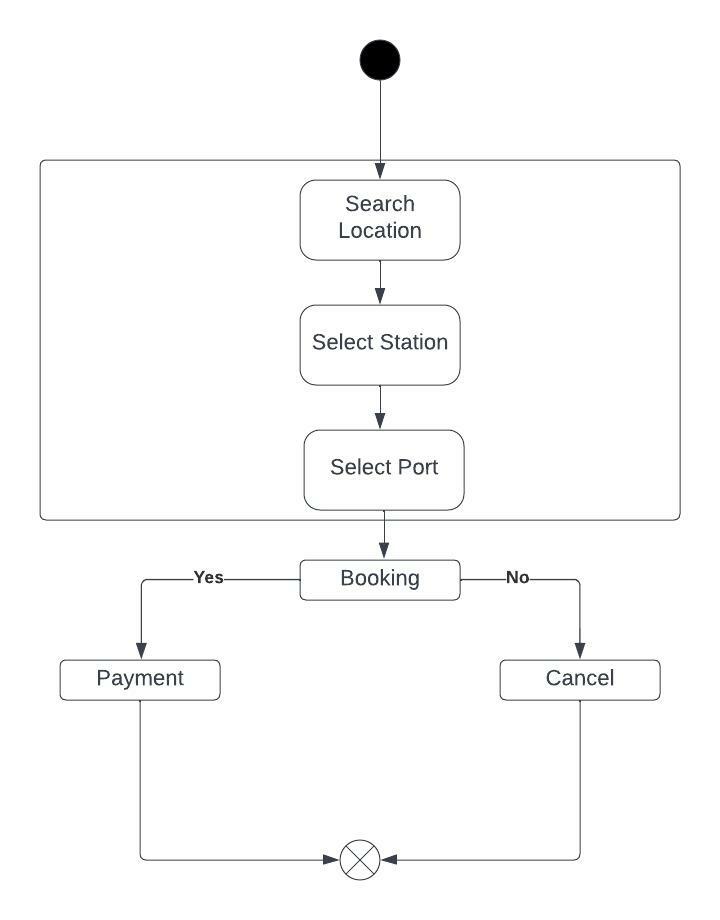
## 4.2.3 State Chart Diagram

It describes different states of a component in a system. The states are specific to a component/object of a system. A statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object, and these states are controlled by external or internal events. They define different states of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists, and it changes when some event is triggered. The most important purpose of statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams −

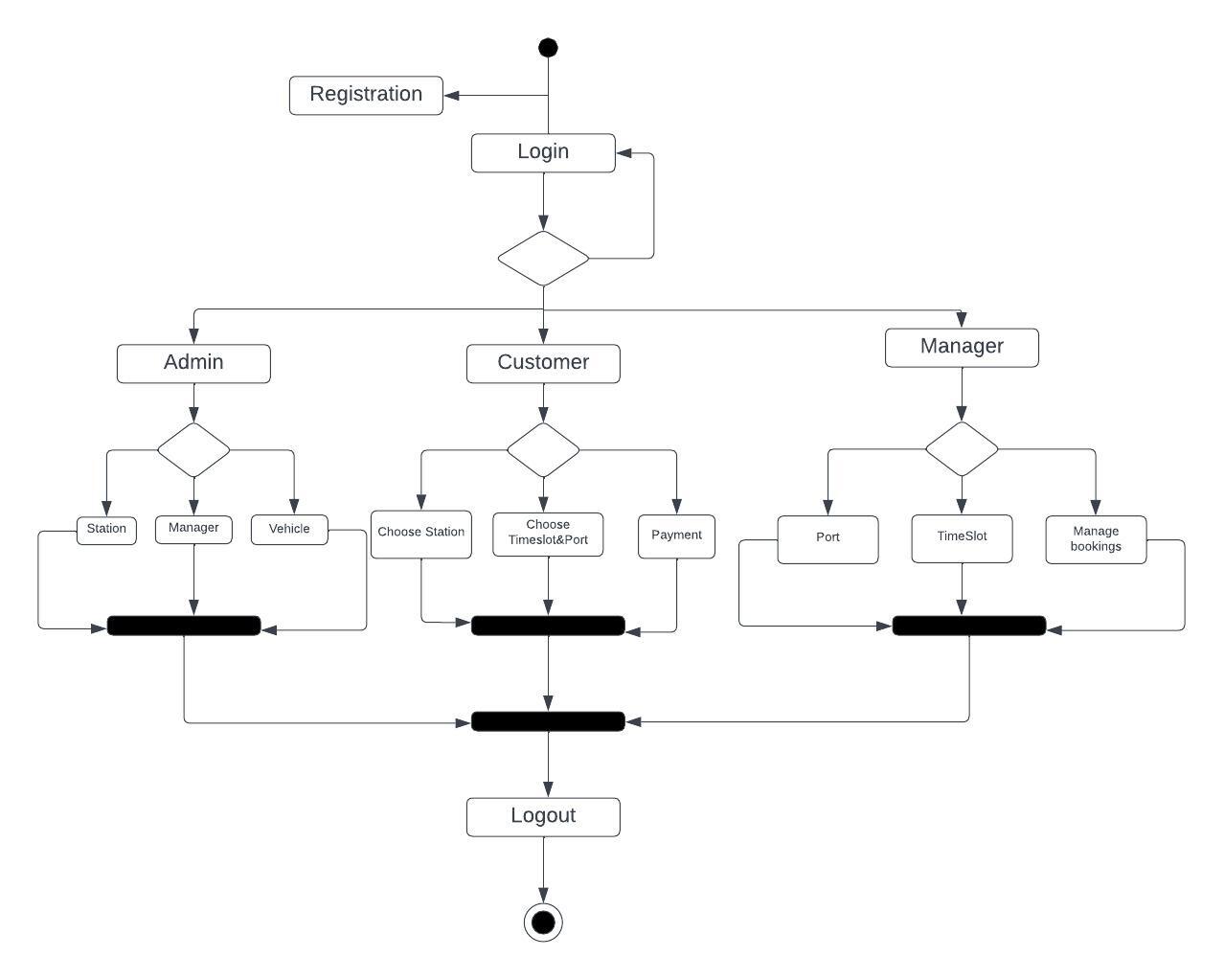
*  To model the dynamic aspect of a system.
*  To model the lifetime of a reactive system.
*  To describe different states of an object during its lifetime.

 Define a state machine to model the states of an object



## 4.2.4 Activity Diagram

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

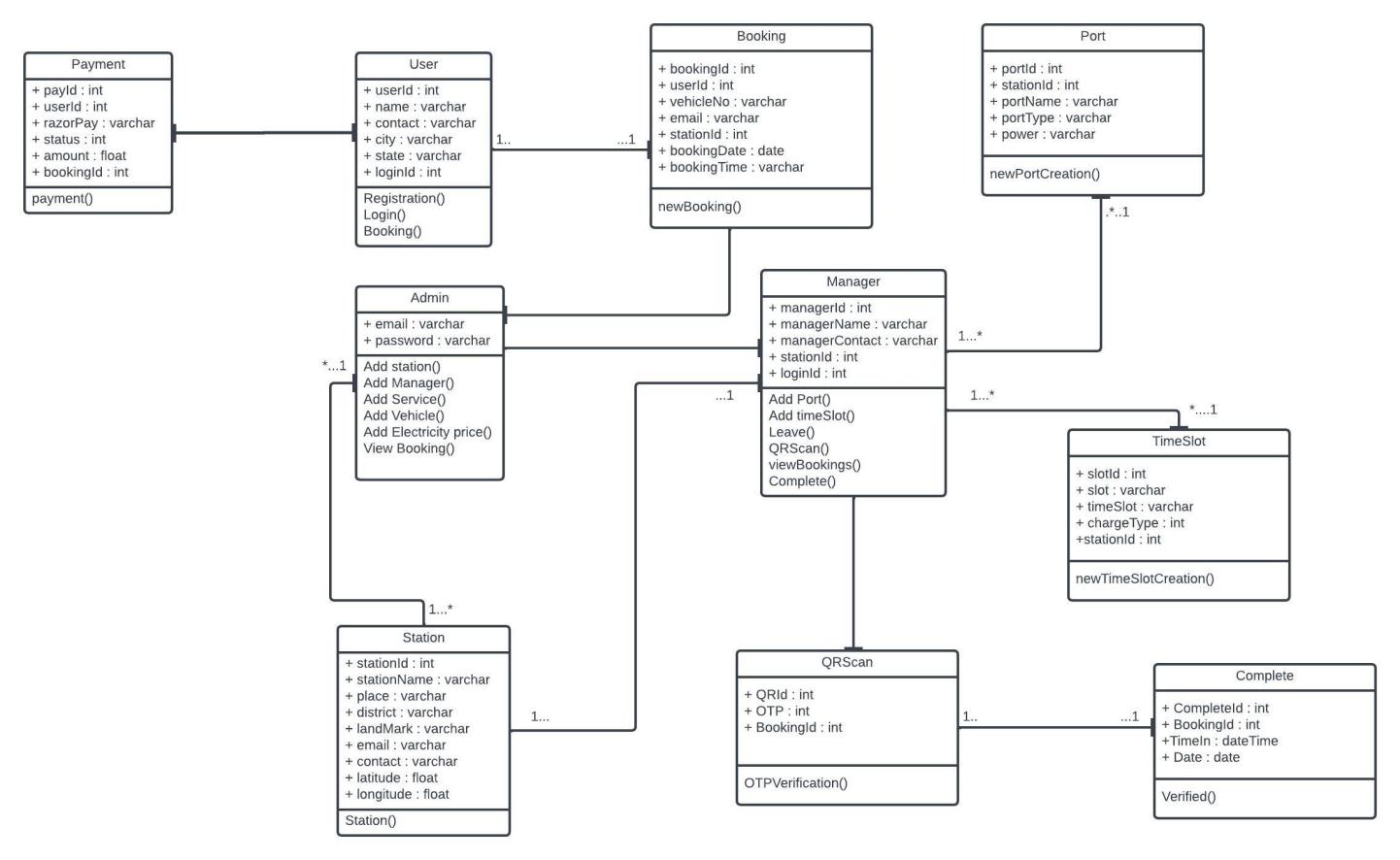


## Class Diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also, for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as −

*  Analysis and design of the static view of an application.
*  Describe responsibilities of a system.
*  Base for component and deployment diagrams.
*  Forward and reverse engineering.

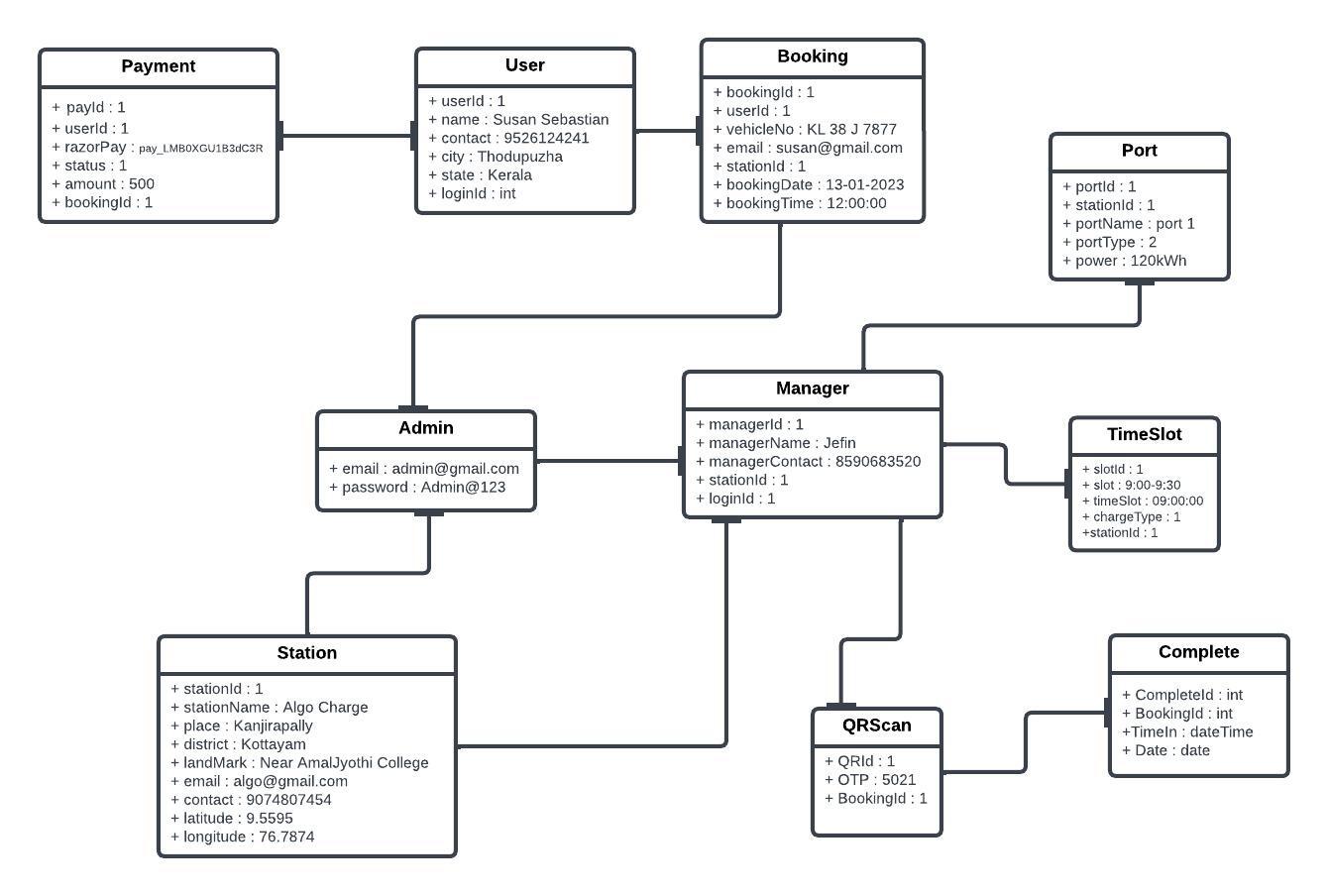


## Object Diagram

Object diagrams are derived from class diagrams, so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system, but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of the object diagram can be summarized as −

*  Forward and reverse engineering.
*  Object relationships of a system
*  Static view of an interaction.
*  Understand object behavior and their relationship from practical perspective

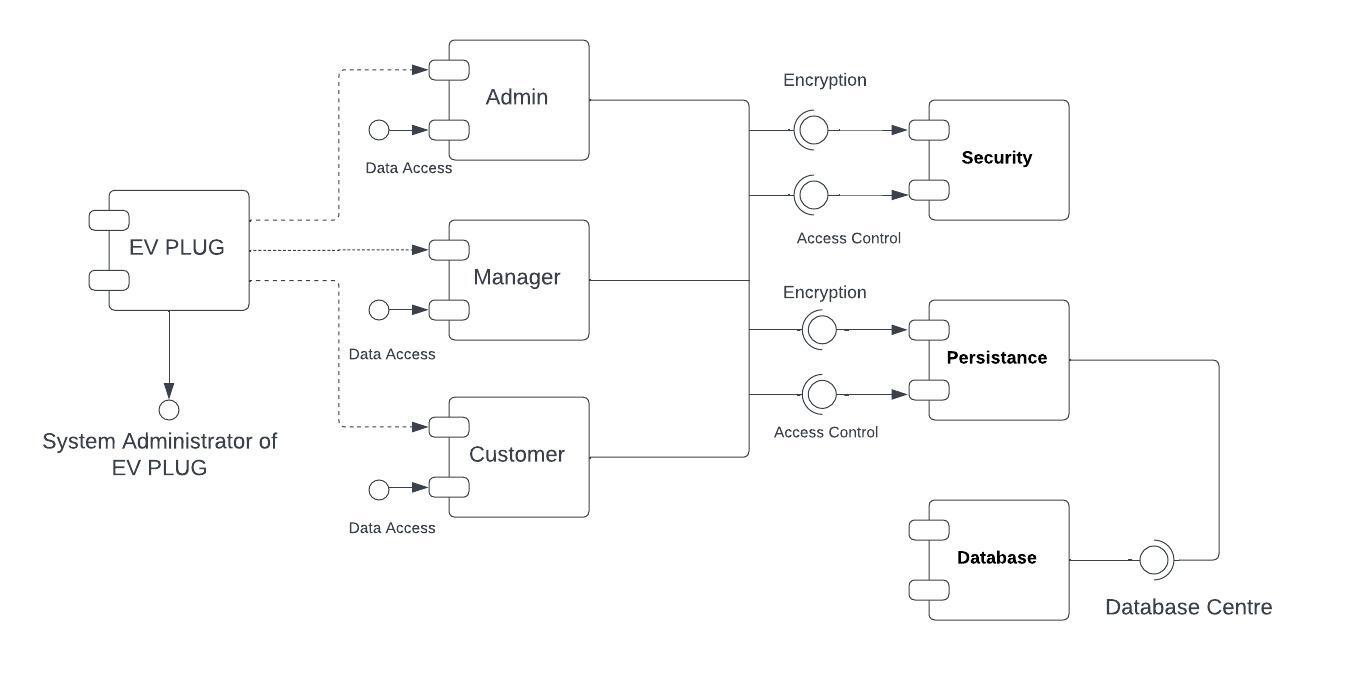


## Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system, but it describes the components used to make those functionalities. Thus, from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc. Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system, but a collection of diagrams is used to represent the whole.

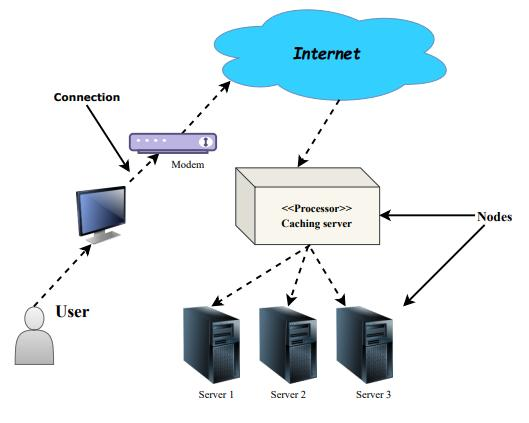
The purpose of the component diagram can be summarized as −

*  Visualize the components of a system.
*  Construct executable by using forward and reverse engineering.
*  Describe the organization and relationships of the components



**4.2.8 Deployment Diagram**

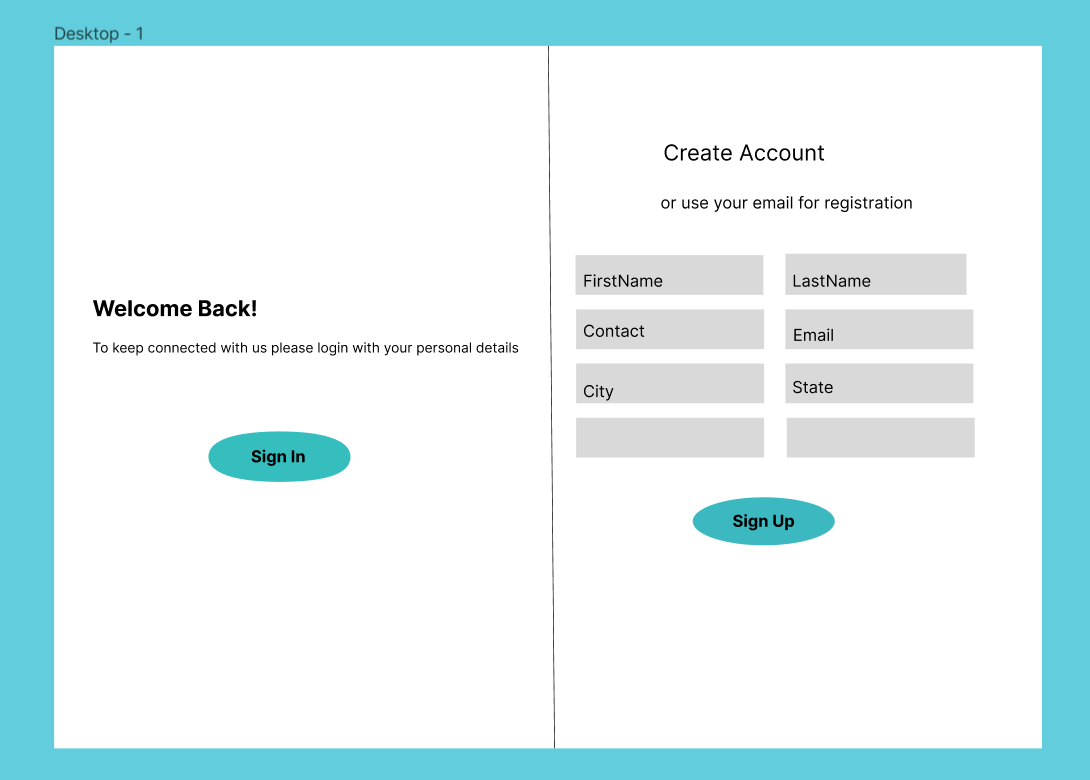
Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.



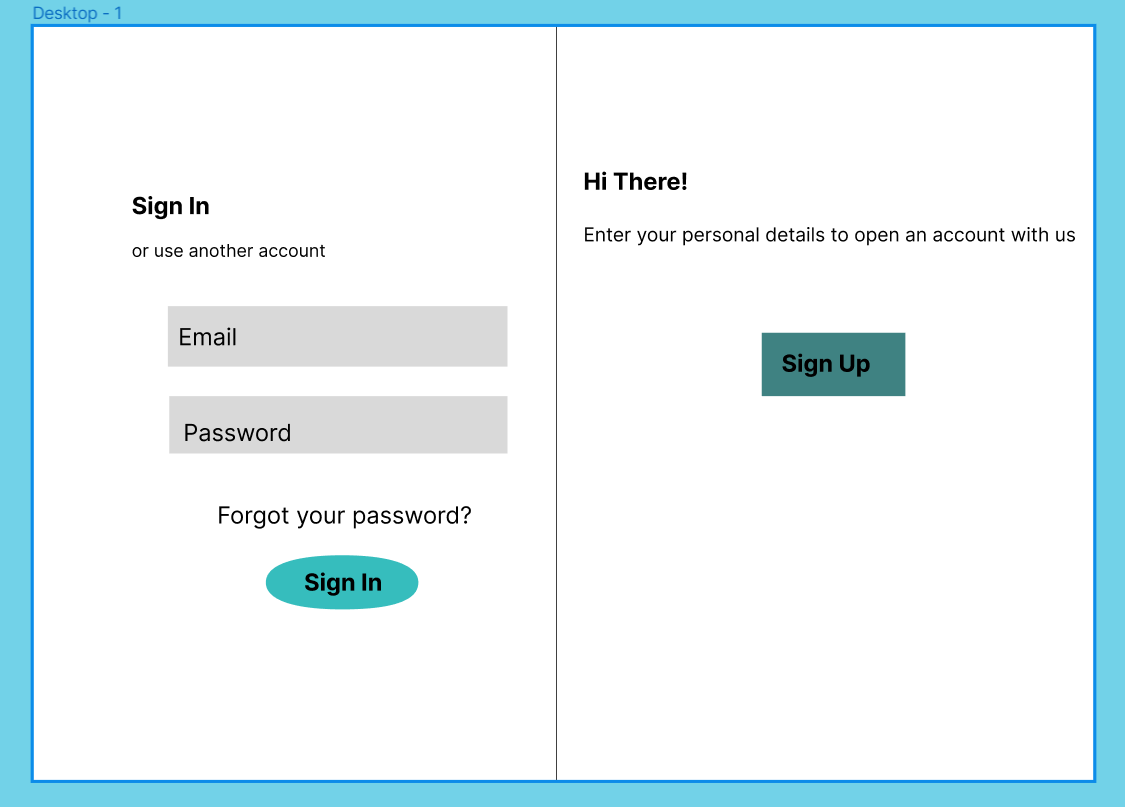
**4.2.9 Collaboration Diagram**

## 4.3 USER INTERFACE DESIGN USING FIGMA

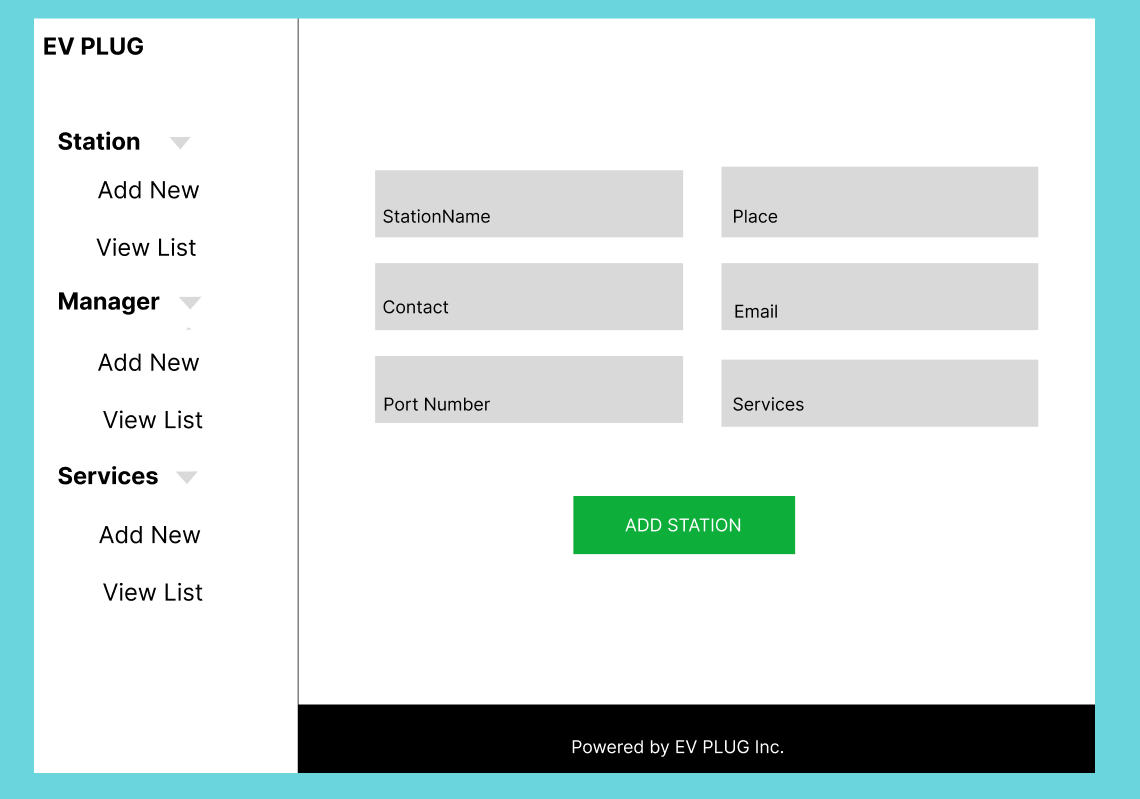
**Form Name: db\_register**



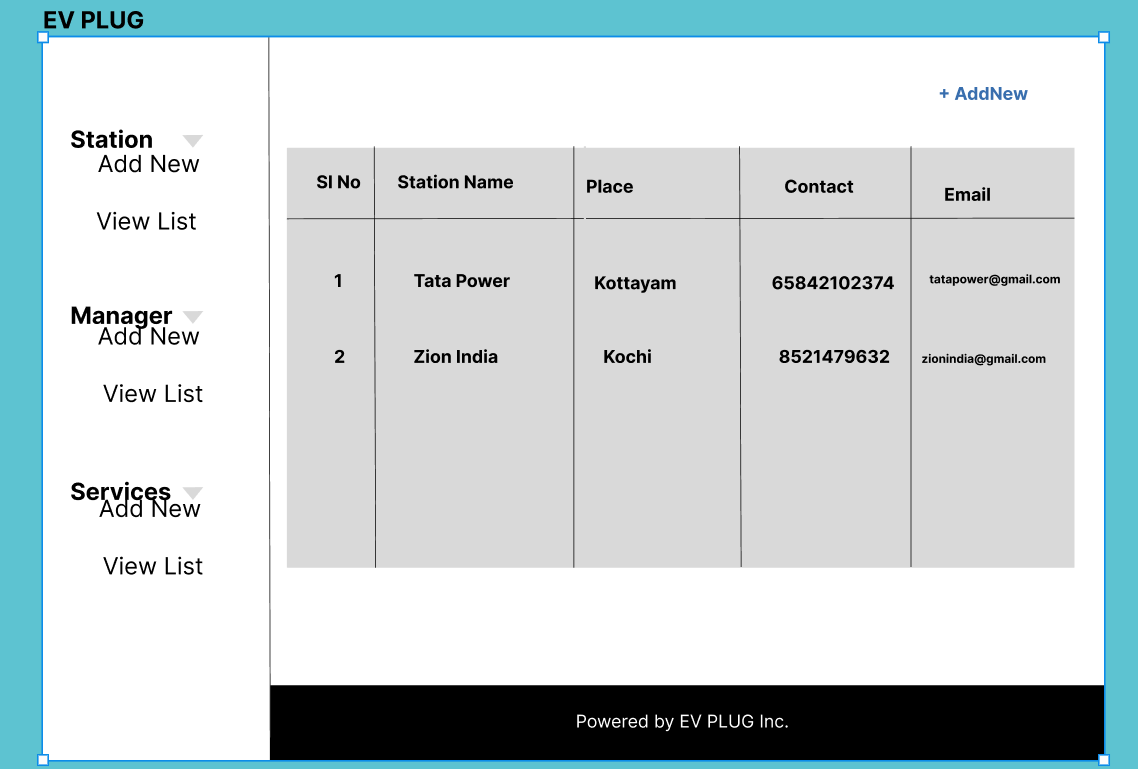
**Form Name: db\_login**



**Form\_name : add\_Station**



**Form\_name : view\_Station**



## 4.3 DATABASE DESIGN

A database is a structured system with the ability to store information and allow users to quickly and effectively access that information. Any database must be protected because its primary goal is its data. The database design process has two stages. User requirements are obtained in the first step, and a database is created to as clearly as possible meet these objectives. Information Level Design is the name of this stage, which is carried out independently of any specific DBMS. The second phase involves converting this information level design into a design for the DBMS that will be used to implement the system in issue. Physical Level Design is the stage where the properties of the DBMS are considered. Data Integrity Data independence

### 4.3.1 Relational Database Management System (RDBMS)

In a relational model, the database is shown as a set of relations. each connection like a file or table of records with values. formal terminology for a relational model, A column header is known as an attribute, a row is known as a tuple, and the table is known as a relation. Each table in a relational database is made up of data that is stored in rows and columns. assigned an arbitrary name. In a story, each row represents a group of associated values.

**Domains, Relations, and Attributes**

A relation is a table. Tuples are the units of a table's rows. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D. Specifying the data type from which the domain's data values are derived is a standard way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

### 4.3.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. Data normalization is a formal process. structures in ways that encourage integrity and remove duplication. Normalization is a method of dividing large datasets into smaller ones and removing superfluous fields. Into a smaller table. Additionally, it serves to prevent additions, deletions, and updates. Anomalies. Keys and relationships are two notions commonly used in data modelling. A table row is uniquely identified by its key. key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized means placing things in their natural form, as the name suggests. By using normalization, the application developer aims to establish a coherent arrangement of the data into appropriate tables and columns, where names may be quickly related to the data by the user. By removing recurring groups from the data, normalization prevents data redundancy, which puts a heavy strain on the computer's resources. include:

 Normalize the data.

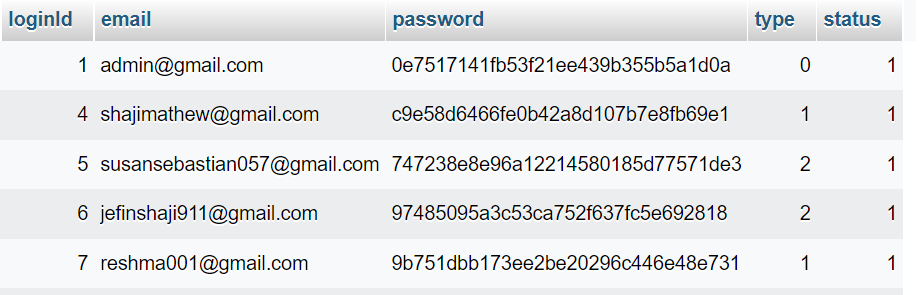
 Choose proper names for the tables and columns.

 Choose the proper name for the data.

**First Normal Form**

According to the First Normal Form, any attribute's tuple's value must be a single value from its domain, which must only contain atomic values. the territory of that property. To put it another way, 1NF forbids "relations within relations. "Alternatively, "relations as attribute values within tuples." The sole attribute values that are allowed by1NF are indivisible or single-atom values. The data must be entered into Initial as the first step. Standard Form. By putting the data in separate tables, you may donate this.in each table is of a similar type. A primary key or foreign key is assigned to each table as per the project's requirements. For each non-atomic relationship, we create new ones in this. attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only. connection or nested attribute. This got rid of data groups that were repeated. If a relation meets the requirements that include only the primary key, it is said to be in first normal form.

Example :



**Second Normal Form**

Accordance with Second Normal Form No non-key attribute should be functionally dependent on a portion of the primary key for relations when the main key has several attributes. This involves breaking down each partial key into its dependent characteristics and setting up new relation for each one. Keep the original primary key and any properties that are entirely dependent on it in your database. This procedure aids in removing data that depends only on a small portion of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and every non-primary key attribute of the connection is completely dependent on its primary key alone, then that relation is said to be in second normal form.

**Third Normal Form**

According to the Third Normal Form relation should not have a non-key attribute that is functionally determined by another non-key attribute or by a collection of non-key attributes. The primary key should not be transitively dependent, in other words. The non-key attributes that functionally determine other non-key attributes are decomposed in this way put up in relation. This action is made to remove anything that is not completely dependent on the Primary Key. Only when a relation is in second normal form and, more importantly, when its non-key characteristics do not depend on those of other non-key attributes, is it considered to be in third no

### 4.4.3 Sanitization

An automated procedure called "sanitization" is used to get a value ready for use in a SQL query. This process typically involves checking the value for particular characters that have a special significance for the target database. To prevent a SQL injection attack, you must sanitize (filter) the input string while processing a SQL query based on user input. For instance, the user and password input is a typical scenario. In that particular scenario, the server response would provide access to the 'target user' account without requiring a password check

**4.4.4 Indexing**

By reducing the number of disk accesses needed when a query is completed, indexing helps databases perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes. The primary key or candidate key of the table is duplicated in the first column, which is the Search key. To make it easier to find the related data, these values are kept in sorted order. Recall that the information may or may not be kept in sorted order.

### 4.5 TABLE DESIGN

**1.tbl\_login**

Eg.Primary Key: **loginId**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | loginId | Int(11) | Primary Key | Primary key of the table |
| 2 | email | Varchar(50) | Not null | EmailId of the user |
| 3 | password | Varchar(50) | Not null | Password of the user |
| 4 | type | Int(11) | Not null | Type of user |
| 5 | status | Int(11) | Not null | Status of user whether it is active or not. |

**2.tbl\_user**

Eg.Primary key: **userId**

Eg.Foreign key: **loginId** references table **tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | userId | Int(11) | Primary Key | Id of the user |
| 2 | userFirstName | Varchar(50) | Not null | First name of the user |
| 3 | userLastName | Varchar(50) | Not null | Last name of the user |
| 4 | userContact | Varchar(50) | Not null | Contact\_no |
| 5 | City | Varchar(50) | Not null | Current city of the user |
| 6 | state | Varchar(50) | Not null | Contact\_no |
| 7 | loginId | Int(11) | Foreign Key | Login id of the user |

**3.tbl\_station**

Eg.Primary key: **stationId**

Eg.Foreign key: **mapId** references table **tbl\_mapping**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | stationId | Int(11) | Primary Key | Id of the station |
| 2 | stationName | Varchar(50) | Not null | Name of the station |
| 3 | place | Varchar(50) | Not null | Place of the station |
| 4 | district | Varchar(50) | Not null | District of the station |
| 5 | landmark | Varchar(50) | Not null | Landmark of the station |
| 6 | email | Varchar(50) | Not null | Email address of the station |
| 7 | contact | Varchar(50) | Not null | Contact number of the station |
| 8 | mapId | Int(11) | Foreign Key | Id of the service mapped to the station |
| 9 | latitude | Float | Not null | Latitude of station location |
| 10 | longitude | Float | Not null | Longitude of station location |
| 11 | status | Int(11) | Not null | Defines station is active or not. |

**4.tbl\_Manager**

Eg.Primary key: **managerId**

Eg.Foreign key: **loginId** references table **tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | managerId | Int | Primary Key | Id of the manager |
| 2 | managerName | Varchar | Not null | Name of manager |
| 3 | managerContact | Varchar | Not null | Contact of manager |
| 4 | stationId | Int | Not null | Id of the station which manager is mapped |
| 5 | status | Int | Not null | Defines whether the manager is active or not |
| 6 | loginId | Int | Foreign Key | Login Id of the manager |

**5.tbl\_service**

Eg.Primary key: **serviceId**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | ServiceId | Int | Primary Key | Primary Key of the table |
| 2 | ServiceType | Int | Not null | Type of the service |
| 3 | ServiceCost | Int | Not null | Cost of the service |
| 4 | Status | Int | Not null | Defines whether the service is active or not. |

**6.tbl\_mapping**

Eg.Primary key: **mapId**

Eg.Foreign key: **ServiceId** references table **tbl\_mapping.**

Eg.Foreign key: **StationId** references table **tbl\_service**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | mapId | Int | Primary Key | Primary Key of the table |
| 2 | ServiceId | Int | ForeignKey | Id of the service |
| 3 | StationId | Int | Not null | Id of the station which service is mapped |

1. **tbl\_userVehicle**

Eg.Primary key: **vehicleId**

Eg.Foreign key: **loginId** references table **tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | vehicleId | Int | Primary Key | Id of the manager |
| 2 | vehicleRegNo | Varchar | Not null | Vehicle Registration Number |
| 3 | vehicleBrand | Varchar | Not null | Brand of the vehicle |
| 4 | vehicleModel | Varchar | Not null | Model of the vehicle |
| 5 | vehicleType | Int | Not null | Defines whether the manager is active or not |
| 6 | loginId | Int | Foreign Key | LoginId of the user |

1. **tbl\_Password\_reset\_temp**

Eg.Primary key: **resetId**

Eg.Foreign key: **email** references table **tbl\_login**

Eg.Foreign key: **loginId** references table **tbl\_login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | resetId | Int | Primary Key | Id of the table |
| 2 | email | Varchar | Foreign Key | Email address of the user |
| 3 | token | Varchar | Not null | Token generated for reset |
| 4 | status | Int | Not null | Defines whether password reset or not |
| 5 | loginId | Int | Foreign Key | Login Id of the user |

1. **tbl\_vehicleBrand**

Eg.Primary key: **BrandId**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | BrandId | Int | Primary Key | Id of the Vehicle Brand |
| 2 | BrandName | Varchar | Not null | Name of the vehicle |
| 3 | type | Int | Not null | Defines whether vehicle is 2,3 or 4 wheeler |
| 4 | Status | Int | Not null | Defines whether the brand is active or not |

**10.tbl\_vehicleModel**

Eg.Primary key: **modelId**

Eg.Foreign key: **BrandId** references table **tbl\_vehicleBrand**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | modelId | Int | Primary Key | Unique Id of the table |
| 2 | modelName | Varchar | Not null | Name of the vehicle model |
| 3 | BrandId | Int | Foreign Key | Id of the vehicle brand |
| 4 | vehicleType | Int | Not null | Type of the vehicle |
| 6 | capacity | float | Not null | Capacity of each vehicle |
| 7 | status | Int | Not null | Defines whether the model is active or not |

**11.tbl\_timeslot**

Eg.Primary key: **slotId**

Eg.Foreign key: **StationId** references table **tbl\_station**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Column\_name** | **Type** | **Key Constraint** | **Description** |
| 1 | slotId | Int | Primary Key | Id of the slot |
| 2 | slot | Varchar | Not null | Time slot for booking |
| 3 | timeSlot | Varchar | Not null | Description of timeSlot |
| 4 | type | int | Not null | Type of charging |
| 5 | StationId | int | Foreign Key | Id of the Station |
| 6 | status | Int | Not null | Defines whether the slot is booked or not |

1. **tbl\_booking**

Eg.Primary key: **booking\_id**

Eg.Foreign key: **StationId** references table **tbl\_station**

Eg.Foreign key: **CustomerId** references table **tbl\_user**

Eg.Foreign key: **VehicleNo** references table **tbl\_uservehicle**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Column\_name** | **Type** | **Constraint** | **Description** |
| 1 | booking\_id | Int | Primary Key | Id of the booking |
| 2 | Customer\_id | Int | Foreign key | Id of the customer |
| 3 | Vehicle\_no | Varchar | Foreign key | Registration number of the Vehicle |
| 4 | Station\_id | Int | Foreign key | Id of the station |
| 5 | Booking\_date | Varchar | Not null | Date of booking |
| 6 | Booking\_time | Varchar | Not null | Time of booking |
| 7 | type | int | Not null | Type of charging |
| 8 | port | int | Not null | Defines which port is booked |
| 9 | price | float | Not null | Price of charging |
| 10 | status | int | Not null | Status of booking |

**13.tbl\_electricity**

Eg.Primary key: **electricityId**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | electricityId | Int | Primary Key | Primary Key of the table |
| 2 | electricityPrice | float | Not null | Defines the price of electricity. |

**14.tbl\_feedback**

Eg.Primary key: FeedbackId

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | FeedbackId | Int | Primary Key | Primary Key of the table |
| 2 | Name | Varchar | Not null | Name of person |
| 3 | Email | Varchar | Not null | emailId of the person |
| 4 | Rating | Varchar | Not null | Rating of the website |
| 5 | Comment | Varchar | Not null | Comments about the website |

**14.tbl\_complete**

Eg.Primary key: **completeId**

Eg.Foreign key: **bookingId** references table **tbl\_booking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | completeId | Int | Primary Key | Primary Key of the table |
| 2 | bookingId | Varchar | Foreign Key | Id of booking |
| 3 | Time\_in | Varchar | Not null | emailId of the person |
| 4 | Completed\_date | Varchar | Not null | Rating of the website |

**15.tbl\_qr\_auth**

Eg.Primary key: **qrId**

Eg.Foreign key: **bookingId** references table **tbl\_booking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Datatype (Size)** | **Key Constraints** | **Description of the Field** |
| 1 | qrId | Int | Primary Key | Primary Key of the table |
| 2 | otp | int | Not null | Otp send to the user |
| 3 | bookingId | Varchar | Foreign Key | Id of booking |

# CHAPTER 5

# SYSTEM TESTING

* 1. **INTRODUCTION**

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Checking that what has been specified matches what the user truly desired is the process of validation. The processes of static analysis and dynamic analysis are additional ones that are frequently related to software testing. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis. Static analysis examines the software's source code, searching for issues and obtaining statistics without actually running the code. Dynamic analysis examines how software behaves while it is running to provide information including test coverage details, time profiles, and execution trails. A series of activities known as tests can be prepared in advance and carried out. systematically. beginning with the module level, testing progresses toward integration. a system entirely based on computers. Testing is necessary and cannot be done without. There are numerous guidelines that can be used to ensure the accomplishment of the system testing objectives.

**There are three ways to test program.**

**•** For correctness

• For implementation efficiency

• For computational complexity

## TEST PLAN

A test plan outlines a set of steps to be taken when implementing various testing techniques. It serves as a blueprint for the actions that will be carried out. During the software development process, engineers create computer programs, their associated documentation, and data structures. Developers are responsible for testing each individual unit of the program, verifying that it functions as intended. To mitigate the potential problems associated with builders testing their own work, an independent test group is often involved. The various levels of testing include:

* Unit testing
* Integration Testing
* Data validation Testing
* Output Testing

### Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. This type of testing is white-box oriented and can be performed simultaneously across multiple components. The modular interface undergoes testing to confirm that the program unit being tested appropriately transfers information in and out. Temporary data storage is scrutinized to ensure that data integrity is preserved throughout each step of an algorithm's execution. Boundary conditions are evaluated to guarantee that each module's statements have been executed at least once. Lastly, all paths for error handling are tested.

### Integration Testing

Integration testing involves systematically building the program structure while simultaneously conducting tests to identify any errors associated with interfacing. The goal is to create a program structure based on design specifications, using components that have already undergone unit testing. The entire program is tested as a whole, which can make it difficult to isolate the cause of any issues that arise. In order to address this, the modules are integrated after unit testing, in order to identify any inconsistencies in the interfaces. Any differences in program structures are eliminated, resulting in a unique program structure.

### Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all code, class modules, forms, and modules. Popular names for this type of testing include system tests and black box testing. The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every programme requirement. The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

### Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm’s need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

* Input Screen Designs.
* Output Screen Designs.

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

* + 1. **Automation Testing**

Automated testing is a software testing process that ensures that the software functions correctly and meets the requirements before it is released for production. It involves the use of testing tools to execute scripted sequences. UI automation testing is a type of automated testing where test scripts are written for each test case, and a series of steps are followed to verify data and action flows. This eliminates the need for testers to manually click through the application to verify the data. Automated testing is necessary when the same test cases need to be run across multiple machines simultaneously.

* + 1. **Selenium Testing**

Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python, etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

Selenium Integrated Development Environment (IDE) is the simplest framework in the Selenium suite and is the easiest one to learn. It is a Chrome and Firefox plugin that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a prototyping tool. If you want to create more advanced test cases, you will need to use either Selenium RC or Web Driver.

Selenium RC was the flagship testing framework of the whole Selenium project for a long time. This is the first automated [web testing](https://www.guru99.com/web-application-testing.html) tool that allows users to use a programming language theyprefer. RC can support the following programming languages [Java](https://www.guru99.com/java-tutorial.html)[C#](https://www.guru99.com/c-sharp-tutorial.html), [PHP](https://www.guru99.com/php-tutorials.html), [Python](https://www.guru99.com/python-tutorials.html), [Perl](https://www.guru99.com/perl-tutorials.html), [Ruby](https://www.guru99.com/ruby-on-rails-tutorial.html). Selenium can be used to automate functional tests and can be integrated with automation test tools such as Maven, Jenkins, & Dockers to achieve continuous testing. It can also be integrated with tools such as TestNG, & JUnit for managing test cases and generating reports.

For example, Selenium is used to test the login functionality of the patient portal by automating the entry of login credentials and verifying that the correct patient dashboard is displayed after successful authentication. Selenium can also be used to automate the testing of appointment scheduling by programmatically entering various input scenarios and verifying that the resulting output matches the expected behavior.

# CHAPTER 6

# IMPLEMENTATION

## INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. Gaining users' trust that the new system will function is perhaps the stage most important to the success of a new system. will be precise and effective. It is mostly focused on user education and documentation. Normally, conversion happens about the same time the user is being or later, training. Simply said, implementation is the gathering of a new system design into operation, the procedure for transforming a fresh, updated system design into a functional one The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result. Implementation encompasses all of the steps used to switch from the old system to the new one. The new system could be entirely different, take the place of an existing manual or automated system, or it could be modified to work better. A reliable system that satisfies organizational needs must be implemented properly. System implementation refers to the process of actually using the built system. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability.

**The implementation state involves the following tasks:**

 Careful planning.

 Investigation of system and constraints.

 Design of methods to achieve the changeover

## IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People first have their doubts about the software, but we must make sure that they do not become resistant. buildup, as one has to make sure that:

The active user must be aware of the benefits of using the new system.

* Their confidence in the software is built up.
* Proper guidance is imparted to the user so that he is comfortable in using the application. Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. The actual process won't happen if the server object is not active and functioning on the server.

### User Training

The purpose of user training is to get the user ready to test and modify the system. The people who will be involved must have faith in their ability to contribute to the goal and benefits anticipated from the computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports and execute other important tasks through user training.

### Training on the Application Software

The user will need to receive the essential basic training on computer awareness after which the new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screen work, what kind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the date that was entered. Then, while imparting the program's training on the application, it should cover the information required by the particular user or group to operate the system or a certain component of the system. It's possible that this training will vary depending on the user group and the level of hierarchy.

### System Maintenance

The mystery of system development is maintenance. A software product works effectively during the maintenance phase of the software cycle. after a system has been implemented successfully, it should be appropriately maintained. System maintenance is a crucial phase in the software development life cycle. Maintenance is necessary for a system to be adaptable to changes in the system environment. Of all, maintaining software entails much more than merely "Finding Mistakes."

### Hosting

Hosting a website refers to the process of making a website accessible and available on the World Wide Web. The hosting service provider is responsible for keeping the server up and running, ensuring that the website or application is available to users at all times, and providing technical support. The steps involved in hosting a website typically include choosing a hosting provider, selecting a hosting plan, registering a domain name, configuring DNS settings, uploading website files, setting up databases and email accounts, and configuring security settings. Overall, hosting is an important aspect of website development and maintenance, as it allows users to access the website or application from anywhere in the world.

**Eg.000Webhost**

000WebHost is among the top [free web hosting providers](https://cybernews.com/best-web-hosting/free-web-hosting/) offering reliable hosting services. It includes a custom control panel as well as the possibility to connect your own domain. It’s good for learning purposes, although I cannot recommend it for professional use.There are many features for this:

* **Pricing** : 000WebHost is absolutely free, there are no costs involved.
* **Ease of Use** : The custom control panel is modern and easy to navigate, although a little slow.
* **Performance** : 000WebHost uptime results were poor (99.05%), while the website’s load speed did surprise – 000WebHost handles small websites and average levels of traffic speedily.
* **Security** : The provider includes just the basic server security modules, while essentials like an SSL certificate are missing.
* **Support** : 000WebHost has no professional customer support options, yet you can use a knowledge base and a community forum.

**Procedure for hosting a website on 000Webhost:**

The steps to host the website named EVPLUG on 000webhost:

**Step 1: Create an account on 000webhost:** The first step is to create an account on the 000webhost website. Provide your basic information such as name, email address, and password.

**Step 2: Verify your email address:** After creating the account, verify your email address by clicking on the verification link sent to your registered email.

**Step 3: Login to the 000webhost cPanel:** Once you have verified your email address, login to the 000webhost cPanel using your registered email and password

**Step 4: Add your domain name:** After logging in to the cPanel, add your domain name by clicking on the 'Add Website' button. Enter the domain name you want to use for your website.

**Step 5: Upload the website files:** After adding the domain name, upload the website files to the 000webhost server using FTP or File Manager.

**Step 6: Configure the website settings:** After uploading the website files, configure the website settings such as the database, email, and SSL.

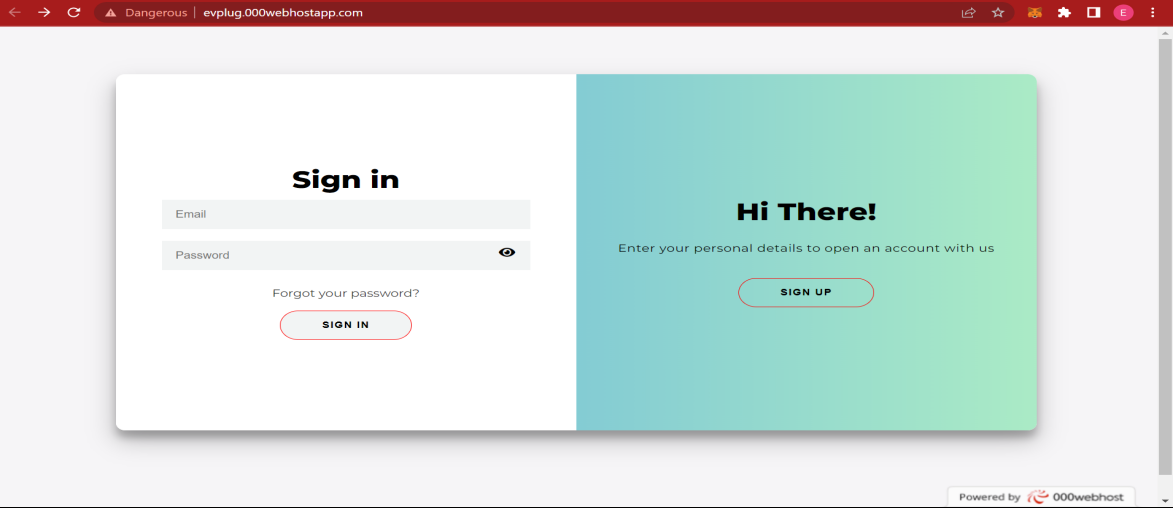
**Step 7: Publish the website:** After configuring the website settings, publish the website by clicking on the 'Publish Website' button.

**Step 8: Publish the website:** After configuring the website settings, publish the website by clicking on the 'Publish Website' button.

**Hosted Website:**

**Hosted Link:** <https://evplug.000webhostapp.com/>

**Screenshot**



# CHAPTER 7

# CONCLUSION AND FUTURE SCOPE

## CONCLUSION

## 

.The current system working technology is old fashioned and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for customer to book charging slot. Provides lots of advantages like user can add their vehicle, viewing the schedule and book, availability of ports in specified station and many more.

* 1. **FUTURE SCOPE**
* Develop a Mobile Application for the website
* Provide option to track the location of station using Google Map
* Sending notification to users’ mobile phone after completing specified amount of

charging.

* We can implement latest technologies like blockchain for data storage.

# CHAPTER 8

# BIBLIOGRAPHY

### REFERENCES:

• Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.

• Roger S Pressman, “*Software Engineering*”, 1994.

• PankajJalote, “So*ftware engineering*: a precise approach”, 2006

### WEBSITES:

• www.w3schools.com

• www.jquery.com

• http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf

• www.agilemodeling.com/artifacts/useCaseDiagram.html

# CHAPTER 9

# APPENDIX

## Sample Code

**Connection code**

<?php

$mysqli=mysqli\_connect("localhost","root","","db\_evplug")or die ("Couldn't connect");

?>

**Admin add new station - addStation.php**

<div id="addStation">

<!-- station Management -->

<?php

$db = mysqli\_connect("localhost","root","","db\_evplug");

$query = "SELECT \* from tbl\_service";

$result = mysqli\_query($db,$query);

?>

<form method="POST" name="addStation">

<center>

<h1 style="margin-top: 10%; margin-left: 175px">New Station</h1>

<input type="text" placeholder="Station Name" id="sname" name="sname" onchange="return stationnameValidate()" style="width: 34%;padding: 12px 20px;margin-left:250px;display: inline-block;border: 1px solid #ccc;

border-radius: 4px;box-sizing: border-box;" />

<table>

<tr>

<td><span id="stationnameValidate" class="validate"></span></td>

</tr>

<tr>

<td><input type="text" placeholder="Contact" maxlength="13" id="contact" name="contact" onchange="return cnoValidate()"/>

</td>

<td><input type="email" placeholder="Email" id="email" name="email" onchange="return emailValidate()" /></td>

</tr>

<tr>

<td><span id="cnoValidate" class="validate"></span></td>

<td><span id="emailValidate" class="validate"></span></td>

</tr>

<tr>

<td><input type="text" placeholder="Place" id="place" name="place" onchange="return placeValidate()"/></td>

<td><select name="district" id="district" required>

<option selected disabled hidden>Select a District</option>

<option value="Trivandrum">Trivandrum</option>

<option value="Kollam">Kollam</option>

<option value="Pathanamthitta">Pathanamthitta</option>

<option value="Alappuzha">Alappuzha</option>

<option value="Kottayam">Kottayam</option>

<option value="Idukki">Idukki</option>

<option value="Ernakulam">Ernakulam</option>

<option value="Thrissur">Thrissur</option>

<option value="Palakkadu">Palakkadu</option>

<option value="Malappuram">Malappuram</option>

<option value="Kozhikode">Kozhikode</option>

<option value="Wayanadu">Wayanadu</option>

<option value="Kannur">Kannur</option>

<option value="Kasargod">Kasargod</option>

</select></td>

<!-- <td><input type="file" class="custom-file-input" name="StationImage"></td>

--> </tr>

<tr>

<td><span id="placeValidate" class="validate"></span></td>

</tr>

<tr>

<td><input type="text" placeholder="Latitude" id="lat" name="lat" onchange="" />

</td>

<td>

<input type="text" style="width: 100%;padding: 12px 20px;margin: 8px 0;display: inline-block;border: 1px solid #ccc;

border-radius: 4px;box-sizing: border-box;" name="lng" id="lng" min="0" placeholder="Longitude">

</td>

<tr>

<td><span id="latValidate" class="validate"></span></td>

<td><span id="lngValidate" class="validate"></span></td>

</tr>

</tr>

</table>

<center><textarea id="landmark" name="landmark" required placeholder="Enter a Landmark" style="width: 34%;padding: 12px 20px;margin-left:250px;display: inline-block;border: 1px solid #ccc;

border-radius: 4px;box-sizing: border-box;"></textarea></center>

<button class="Submitbutton" name="addStation" id="addStation" style="width: 10%; margin-left: 20%" onclick="return addstationValid()">Add Station</button>

</center>

</form>

</div>

**Customer booking details - booking\_details.php**

<div id="Bookingdetails">

<center>

<div style="width: 70%;" class="w3-container">

<h3>Booking Details</h3>

<form method="POST">

<div class="w3-card-4" style="width:50%">

<div class="w3-container w3-center">

<table>

<tr>

<td>Date</td>

<td><?php echo $\_SESSION['date'] ?></td>

</tr>

<tr>

<td>Charging Type</td>

<?php $chargeType = $\_SESSION['chargingtype'];

if($chargeType==1){

$type='Full Charge';

}

if($chargeType==2){

$type='Quick Charge';

}

?>

<td class="second-row"><?php echo $type ?></td>

</tr>

<tr>

<td>Charging Slot</td>

<td><?php echo $\_SESSION['time'] ?></td>

</tr>

<tr>

<td>Vehicle</td>

<td><?php echo $\_SESSION['vehicleno']; ?></td>

</tr>

<tr>

<td>Price</td>

<td><?php echo $\_SESSION['price']; ?></td>

</tr>

</table>

<input type="submit" value="Proceed to Booking" name="continue" id="continue" style="width: 40%">

</div>

</div>

</form>

</div>

</center>

</div>

**Manager AddTimeSlot- addTimeSlot.php**

<form method="POST">

<table>

<tr>

<td><Label>Select Charging Type</Label></td>

</tr>

<tr>

<td>

<select name="chargingType" id="chargingType" style="width:150px; margin-top: 10px">

<option selected disabled>Select Charging type</option>

<option value="1" name="1">Full Charge</option>

<option value="2" name="2">Quick Charge</option>

</select>

</td>

</tr>

<tr>

<td><select name="slot" id="slot" style="width:150px; margin-top: 10px">

<option value="" selected disabled>Select Time Slot</option>

</select></td>

</tr>

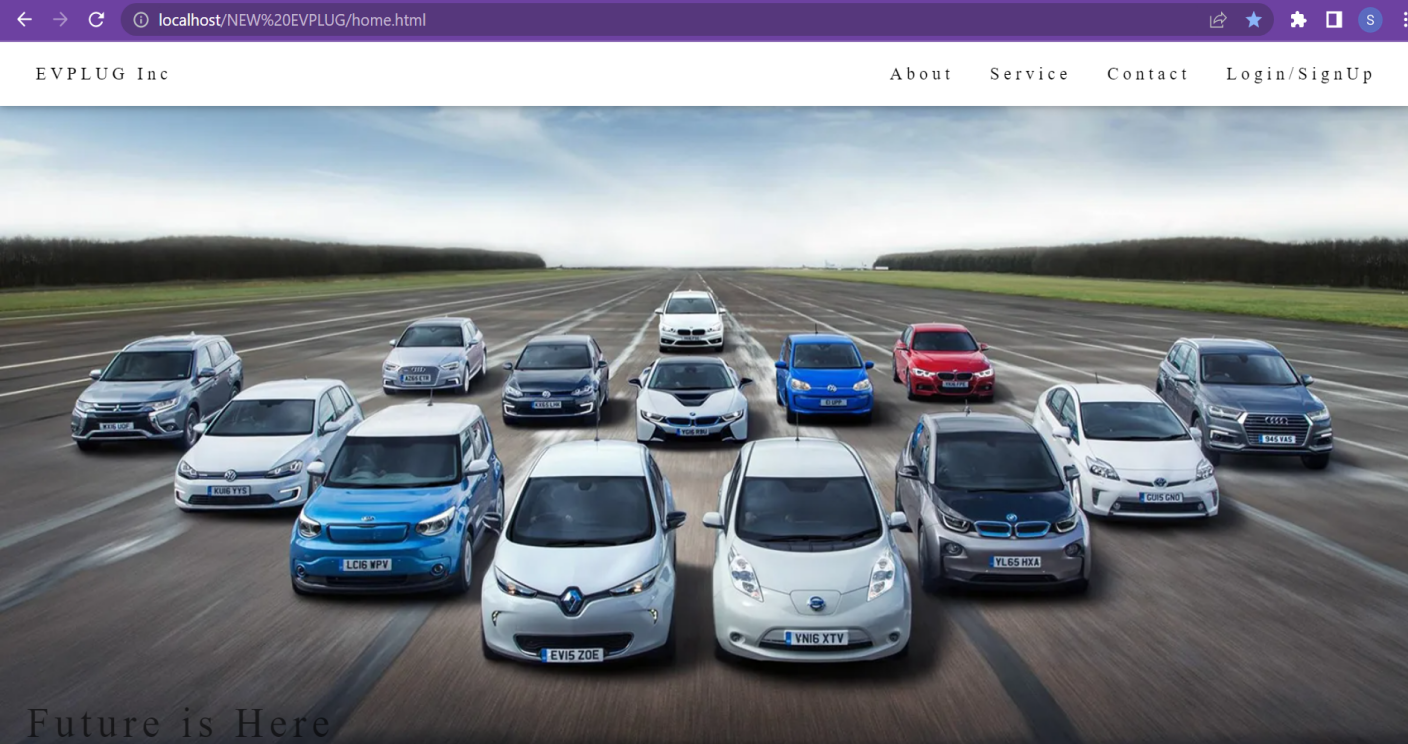
</table>

<button style="margin-left: 0px;" name="save" id="save" class="btnAdd success">Save</button>

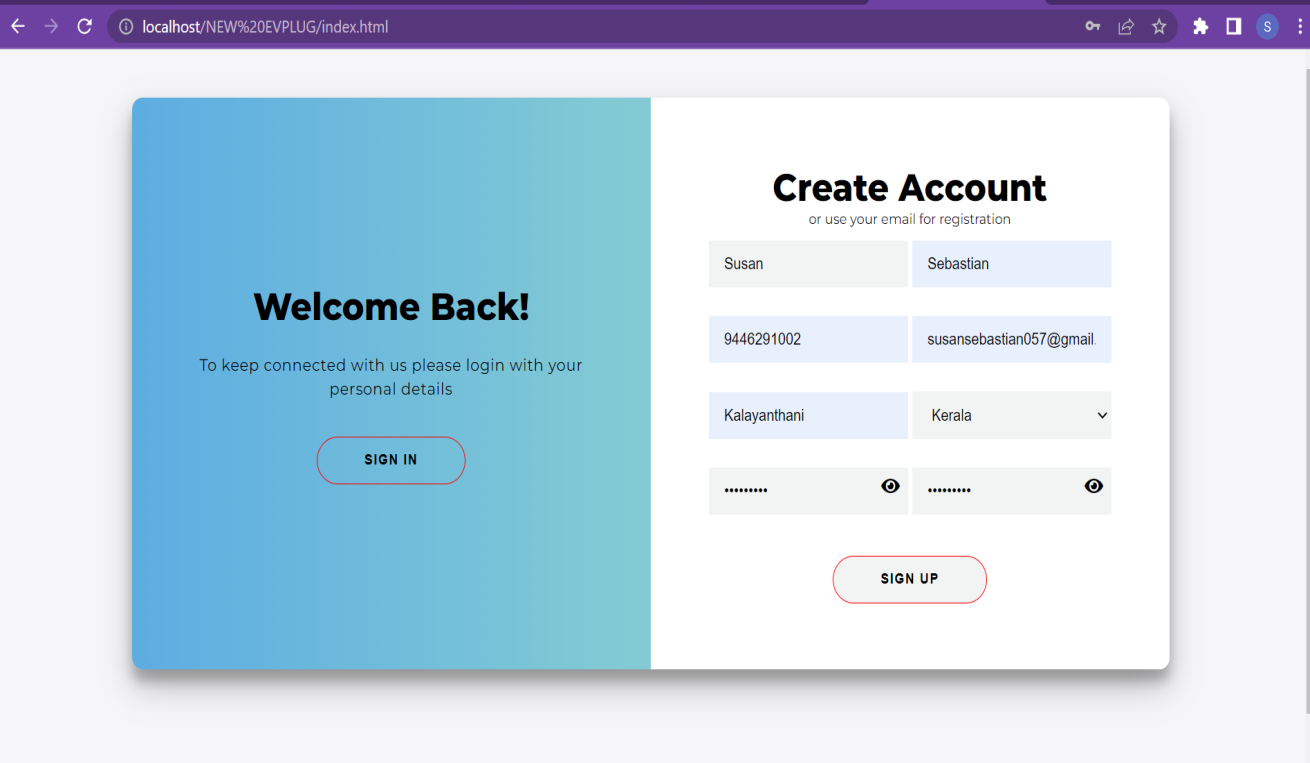
</form>

## 9.2 Screen Shots

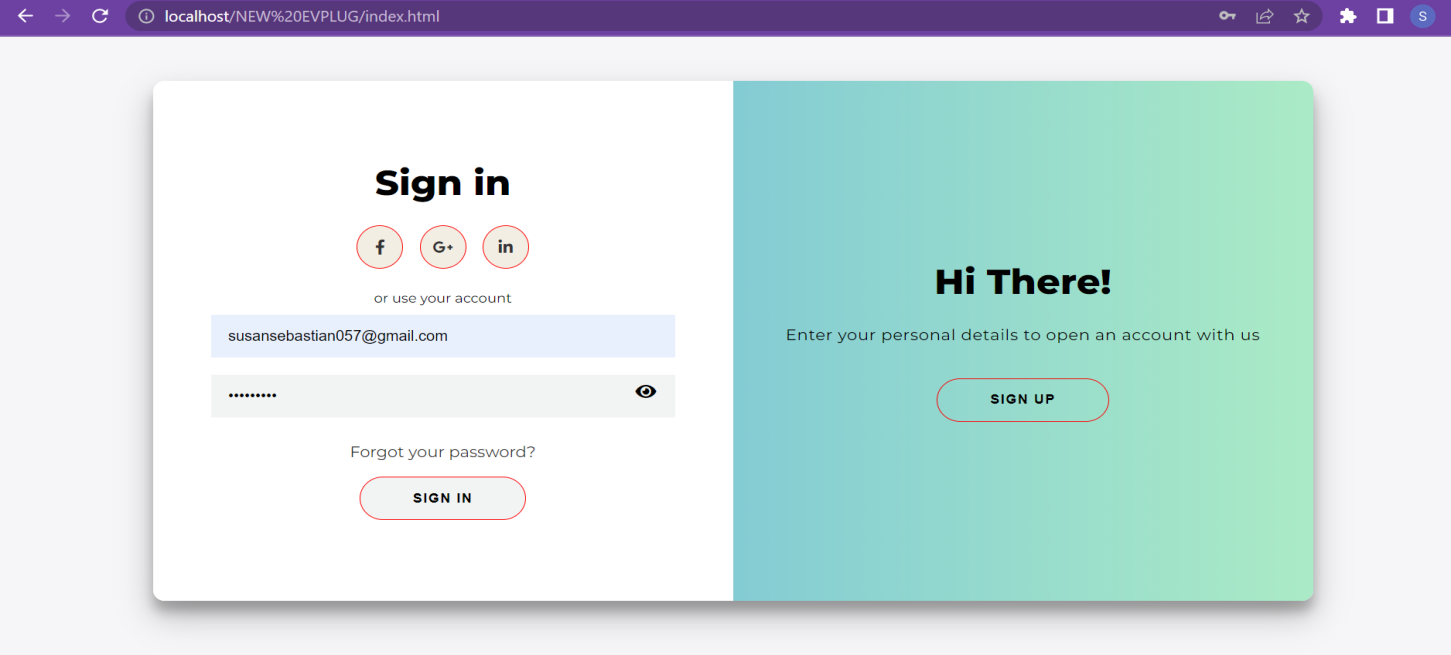
**Index Page**



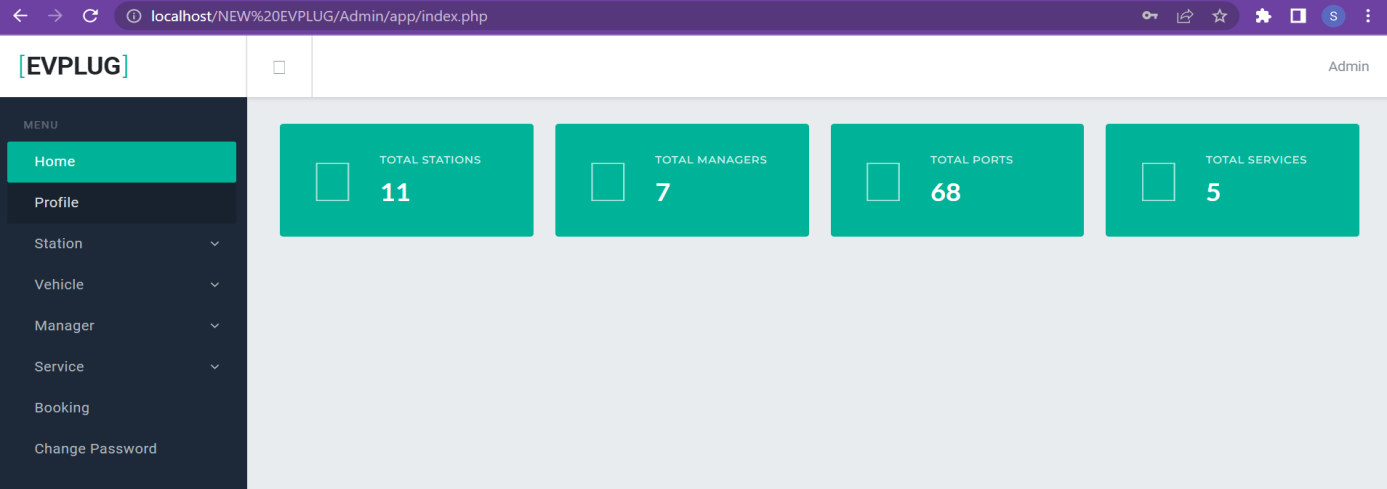
**Registration Page**



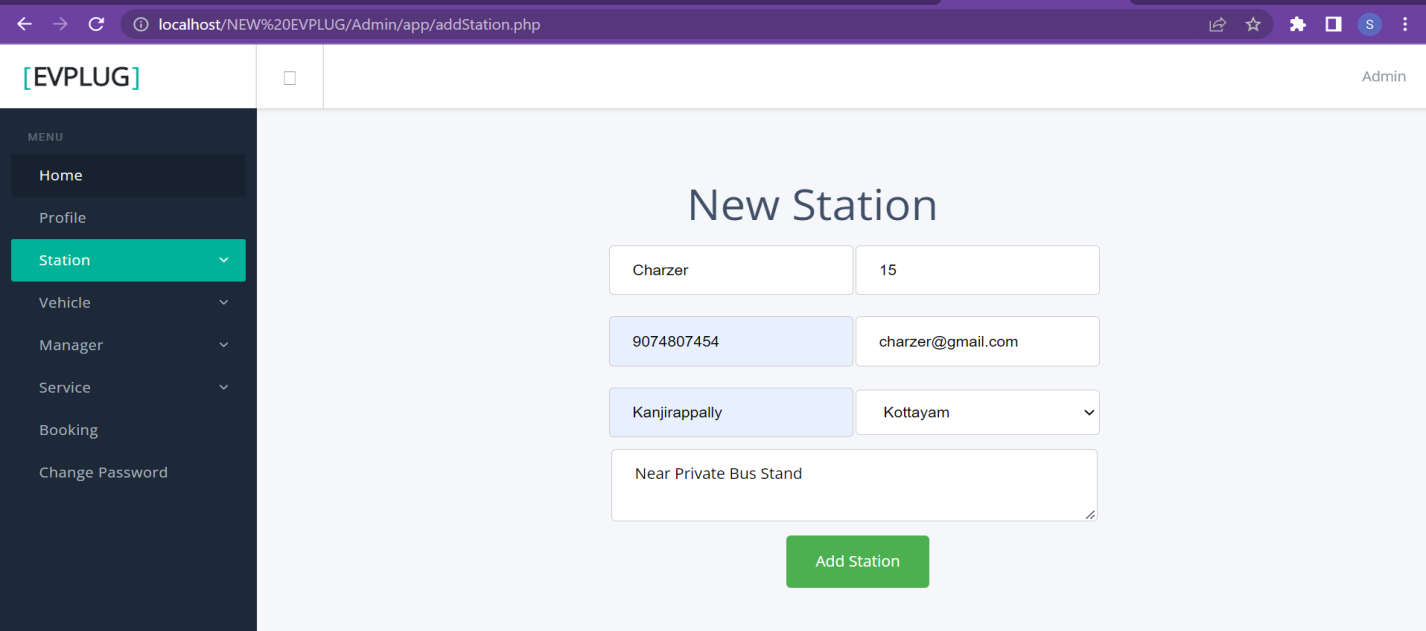
**Login Page**



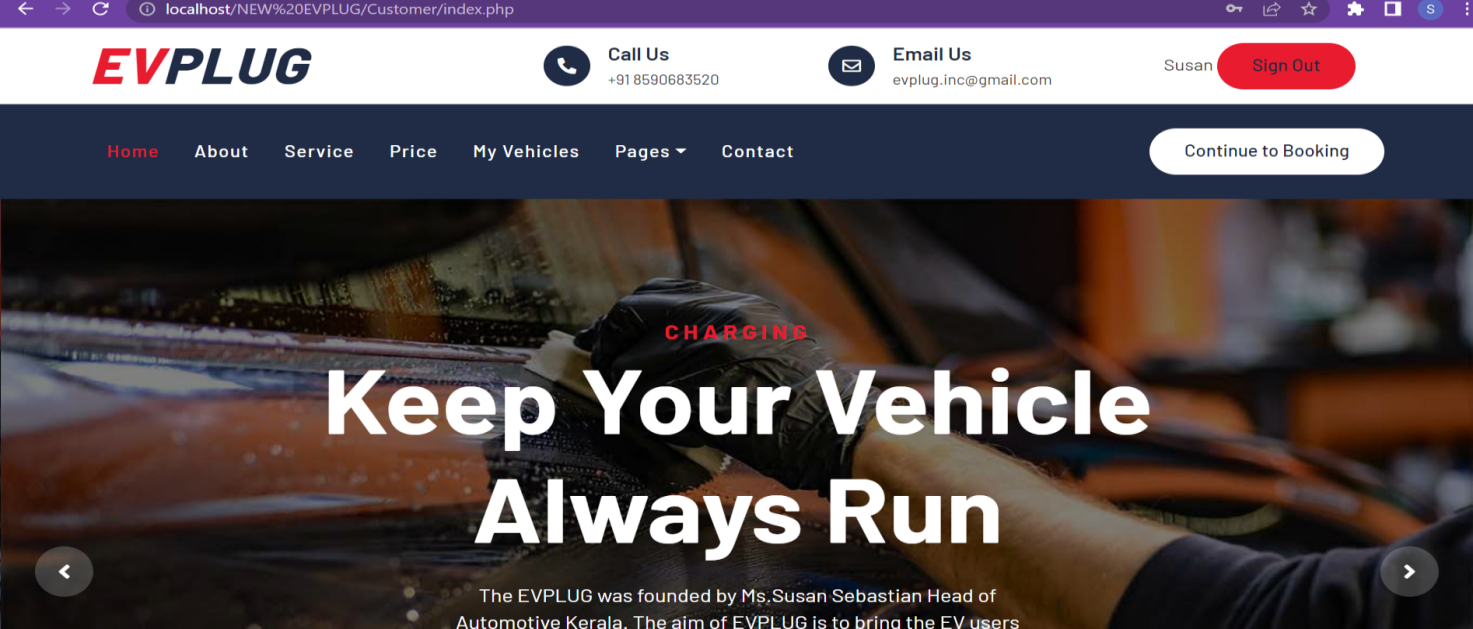
**Admin Dashboard page**



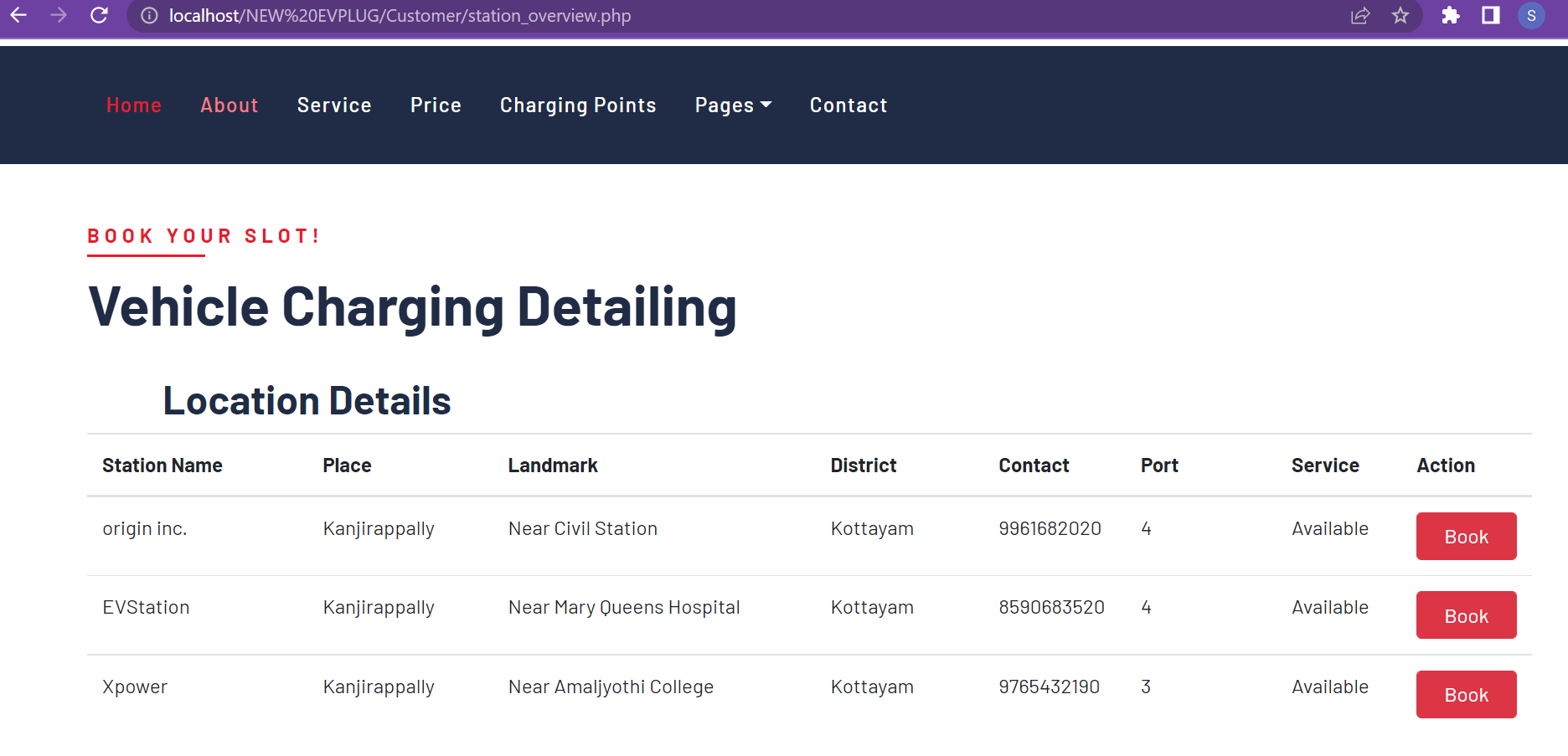
**Admin add Station**



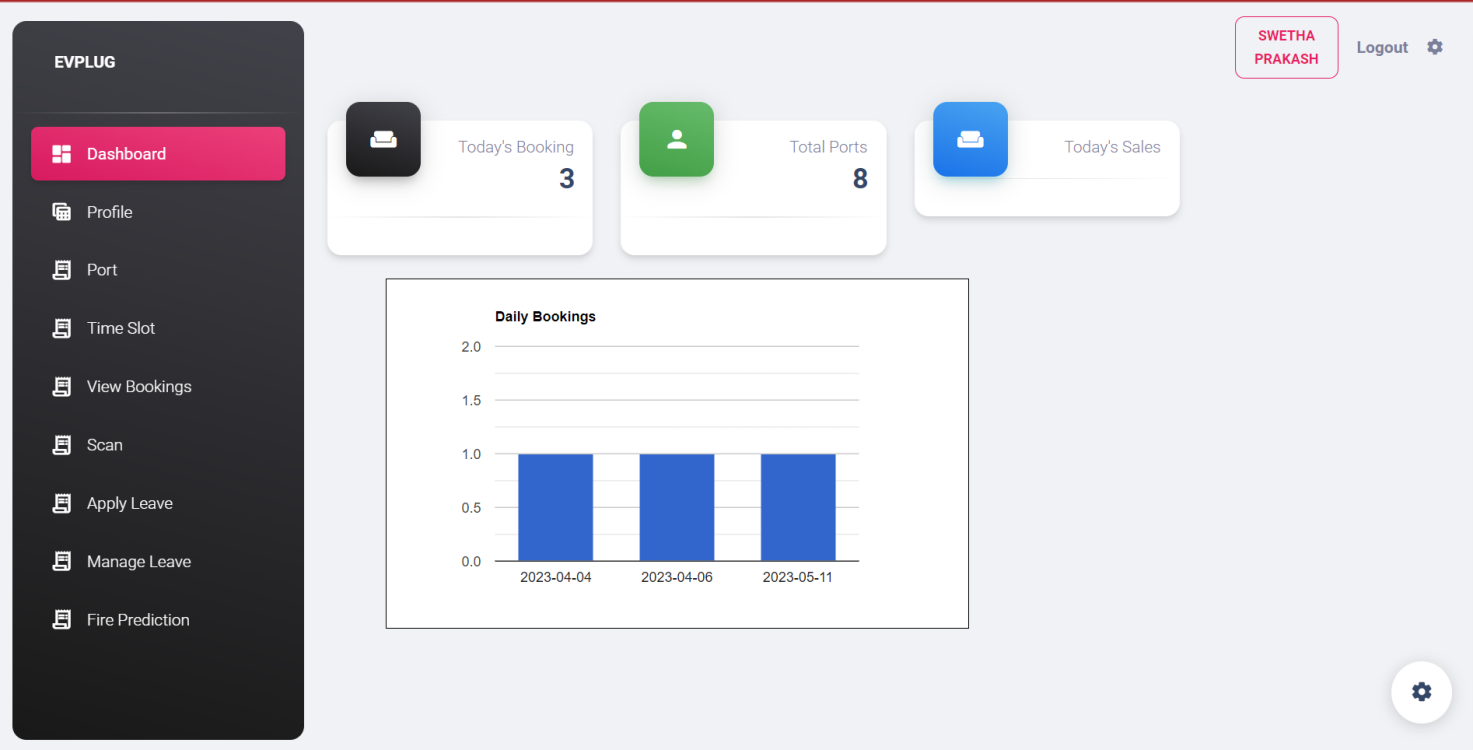
**Customer Index Page**



**Customer - Station & Location details Page**



**Manager Dashboard - dashboard.php**



**Manager ViewPort - port.php**

