

**Project Name : EV Charging Assistance**

**Batch PG-DAC Sep-2021**

Submitted By

**Group No: 6**

* 210943120086 Saurabh Phadnis
* 210943120079 Sanket Repale
* 210943120091 Nagesh Shinde
* 210943120057 Mugdha Shinde

**UNDER THE GUIDENCE OF:**

Mrs. Harshita Maheshwari

Faculty Member

**Infoway Technologies, Pune.**

**Table of Content**

**1. Introduction**           1.1. Problem Definition  
           1.2. Objective of Project  
**2. Feasibility Study  
3. Analysis**           3.1. Existing System                                    
           3.2. Proposed System                                    
           3.3. Software Requirement Specification                                    
**4. Design**           4.1. Data Flow diagram 4.2. ER diagram                      
**5. Implementation**           5.1. Modules  
           5.2. Module description  
           5.3. Introduction of technologies used  
**6. Test cases  
7. Screenshots of Webpages**

1. **Testing  
   9. Conclusion  
   10. Future Enhancement  
   11.  Bibliography**

1. **Introduction  
   1.1. Problem Definition**

As the world is facing an insufficiency in fossil fuels, every nation is moving towards sustainable, admissible, reliable and efficient green resources of energy. The technology supporting Electric Vehicles (EVs) is rapidly developing towards improvement as the cost of EV components are reducing. Electric Vehicles (EV) are gaining more popularity as the conventional vehicles are affecting the environment drastically. As EVs become more commercial, there will be a need to create an efficient slot booking system as the charging process can be time consuming and the need for more stations will be demanding. Unlike petrol, CNG and diesel driven automobiles, these electric vehicles requires specific charging cables, ports, speed and time. So users probably face lots of problems while travelling as they are not aware about the availability of nearby charging stations as well as the availability of Charging ports.

**1.2. Objective of Project**

Main objective of the application is that customer need not to wait in a long queue for charging their vehicles. Just on a single click they can book a slot and charge their EVs.

Charging station managers would be able to display their services on an online portal which will help them to reach out to the maximum customers.

1. **Feasibility Study**

The global EV market is predicted to increase at a rate of 26.8% per year from 2021 to 2030. In 2021, the number of EV units is more than 4,093k. This number will reach 34,756k by 2030. In 2020, there were more than 1.3 million publicly available EV chargers worldwide, of which 30% were fast chargers. Since EVs become more prevailing and widespread, the number of EV charging stations will grow significantly. The value of the EV market in India was 5 billion USD in 2020. It is expected to reach at a massive CAGR of 44% from 2021 to 2026, the market value will be more than 47 billion USD by 2026.

Investing in an EV charging assistance app development is beneficial in several ways. Because it is still a secure point in the marketplace with low competition. These market statistics show that the EV market is developing rapidly. The main reasons for this development are the requirement for zero carbon emission transportation and the government’s initiatives for supporting zero-emission vehicles through subsidies, tax rebates, etc.

The EV charging stations will grow with the increase in the number of EVs. Hence, by developing an EV charging station assistance app, one can take an excellent initiative for running a fruitful business in the future. An EV charging app displays ads through videos, sidebars, pop-ups, etc. Many applications don’t charge a fee from users for using them, but they show some ads, and their earning comes from ad fees. So It is the most standard money-making method. One can earn every time a user pays to subscribe to the app. If we want to provide a free membership opportunity, we can also charge them for booking a slot at the EV charging station and get payment from both the charging station and the user.

1. **Analysis**
   1. **Existing System**

* In current situation customer only get to know about location of nearby charging stations.
* Customer needs to stay in queue in order get their EV charged .
* Sometimes customer even don’t get to know about availability of specific charging cable according to model of their vehicle.
* Charging stations which are located away from prime areas get affected because customers are unaware about their existence.
  1. **Proposed System**
* Using this app customers can book appointment according to their time, vehicle model and convenience.
* This will significantly save time of customer surpassing one of the drawback of EV and will lead people to switch towards eco-friendly option of EV vehicles.
* Charging station managers would be able to display their services on an online portal which will help them to reach out to the maximum customers.
  1. **Software Requirement Specification**

**Hardware:**

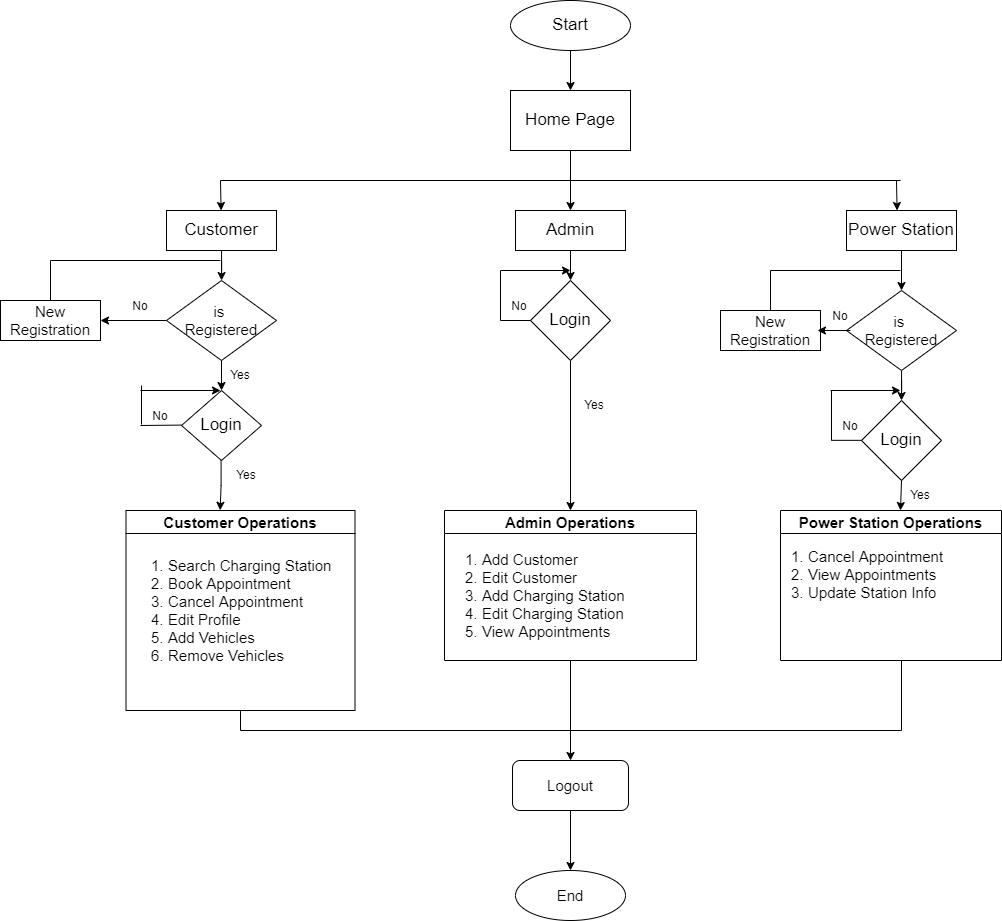
1. Intel i3 processor 3 rd generation or later / AMDRyzen2002ndgeneration or later
2. 2 GB ddr3 ram
3. Windows 7 Home edition or later
4. 200 GB Sata HDD Space
5. Data Connection 200 Kbps

**Software:**

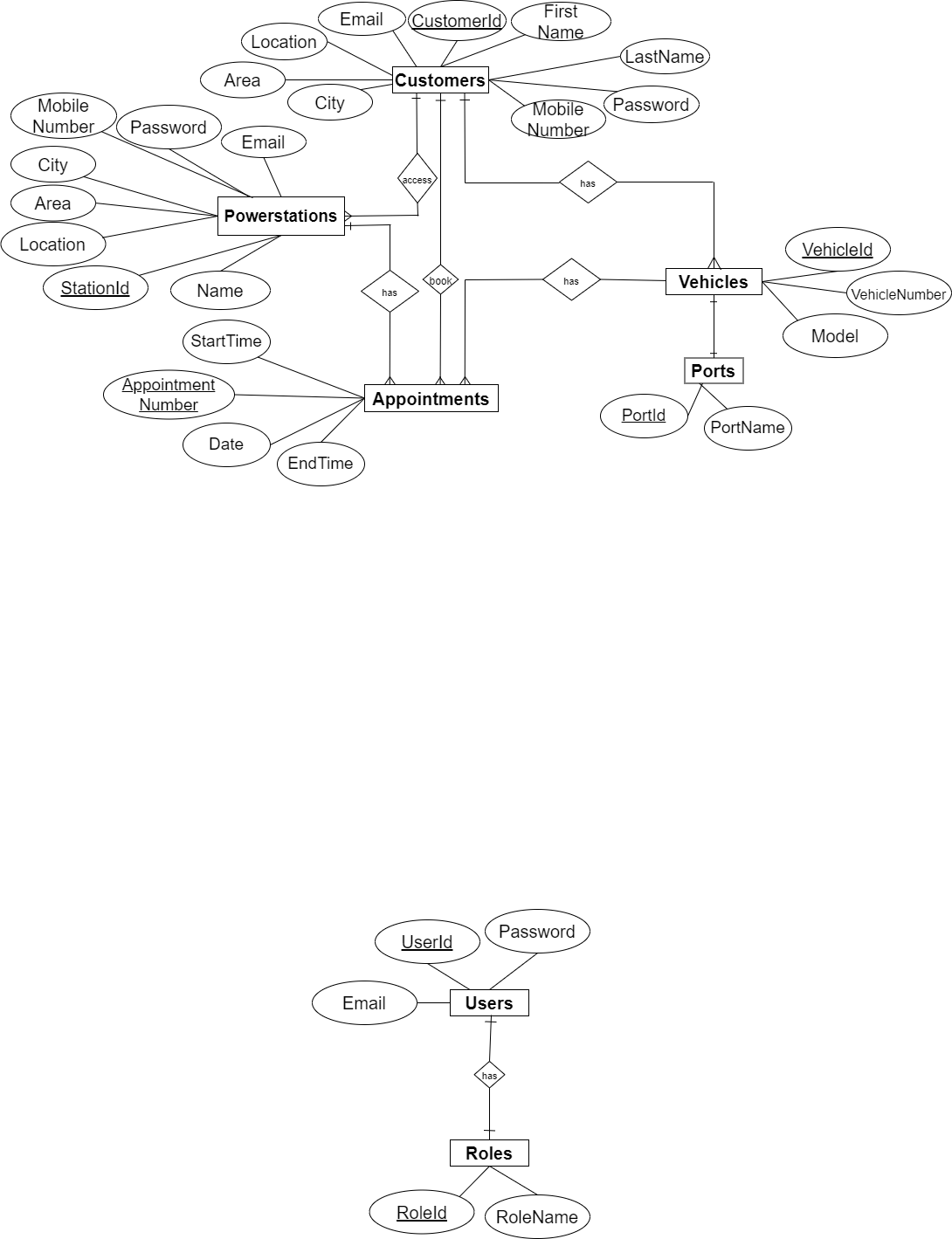
1. Eclipse EE 2020-21
2. MySQL 5.7 with Workbench 8.0
3. Google Chrome version 79.0
4. Apache Tomcat Server 8.5

5. Maven Dependencies

**4. Design                                    
4.1. Data Flow Diagram**



**4.2. ER Diagram**

****

**5. Implementation**

**5.1. Modules**

Customer

Admin

Charging Station Manager

**5.2. Module description**

**1) Customer**

If you are a user, you need to register on the app with your username, mobile number, email id, vehicle number(unique), vehicle details. User can add number of vehicles and login with username and mobile number.

Customer can book a time slot according to the availability. Also customer can cancel the booked appointment.

1. **Admin**

Admin manages customers and charging station Managers. Admin can login using username and password to operate functions of the app.

Admin adds new charging stations. Also admin can remove the charging station over a request. Booking history which consists of every appointment details of each customer is managed by admin only. In case of any failure at a charging station, its admin’s responsibility to notify customer.

**3) Charging Station Manager**

Charging station managers registers on the app by providing details like username, address of the charging station, types of ports. Charging station manager can login using username and password.

Charging station managers adds number and types of ports available at the charging station. It is charging station manager’s responsibility to book a time slot. Also charging station managers cancel a appointment over the request.

**5.3. Introduction of technologies used**

**5.3.1 Spring Boot Framework:**

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run. You can get started with minimum configurations without the need for an entire Spring configuration setup.

Spring enables you to build applications from “plain old Java objects” (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

**Features of Spring boot Framework:**

**Web Development**

It is well suited Spring module for web application development. We can easily create a self-contained HTTP server using embedded Tomcat, Jetty or Undertow. We can use the spring-boot- starter-web module to start and running application quickly.

**Spring Application**

It is a class which provides the convenient way to bootstrap a spring application which can be started from main method. You can call start your application just by calling a static run () method.

**Admin Support**

Spring Boot provides the facility to enable admin related features for the application. It is used to access and manage application remotely. We can enable it by simply using spring.application.admin. Enabled property.

**Logging**

Spring Boot uses Common logging for all internal logging. Logging dependencies are managed by default. We should not change logging dependencies, if there is no required customization is needed.

**Security**

Spring Boot applications are spring bases web applications. So, it is secure by default with basic authentication on all HTTP endpoints. A rich set of Endpoints are available for develop a secure Spring Boot application.

**Advantages of a Spring Boot application**

* Fast and easy development of Spring-based applications
* No need for the deployment of war files
* The ability to create standalone applications
* Helping to directly embed Tomcat, Jetty, or Undertow into an application
* No need for XML configuration
* Reduced amounts of source code

**The JDBC Template**

The central class of the Spring JDBC abstraction framework is the Jdbc Template class that includes the most common logic in using the JDBC API to access data, such as handling the creation of connection, statement creation, statement execution, and release of resource. The Jdbc - Template class can be found in the org.springframework.jdbc.core package.

The Jdbc Template class instances are thread-safe once configured. A single Jdbc Template can be configured and injected into multiple DAOs. We can use the Jdbc Template to execute the different types of SQL statements. Data Manipulation Language (DML) is used for inserting, retrieving, updating, and deleting the data in the database such as SELECT, INSERT, or UPDATE statements

**5.3.2 MySQL**

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**Features of MySQL:**

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 stand alone 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. 

**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. The MySQL Database Server is very fast, reliable, scalable, and easy to use. MySQL Server works in client/server or embedded systems. The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

**5.3.3 React JS**

React JS is JavaScript library used for building reusable UI components. According to React official documentation, following is the definition−React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boiler plate and is easier to reason about than traditional data binding.

**React Features**

* **JSX** − JSX is JavaScript syntax extension. It isn't necessary to use JSX in React development, but it is recommended. 
* **Components** − React is all about components. You need to think of everything as a component. This will help you maintain the code when working on larger scale projects. 
* **Unidirectional data flow and Flux** − React implements one-way data flow which makes it easy to reason about your app. Flux is a pattern that helps keeping your data unidirectional. 
* **License** − React is licensed under the Facebook Inc. Documentation is licensed under CC BY 4.0.

**React Advantages**

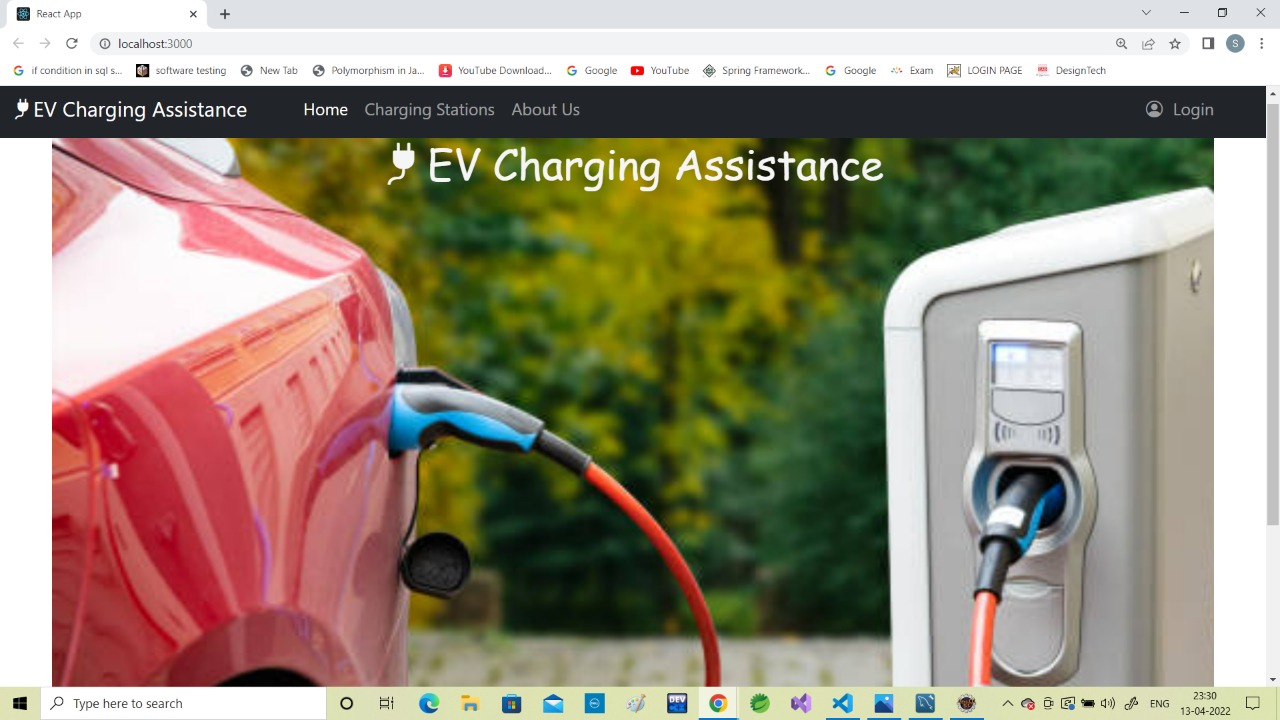
Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM. Can be used on client and server side as well as with other frameworks.  Component and data patterns improve readability, which helps to maintain larger apps.

**6. Test cases**

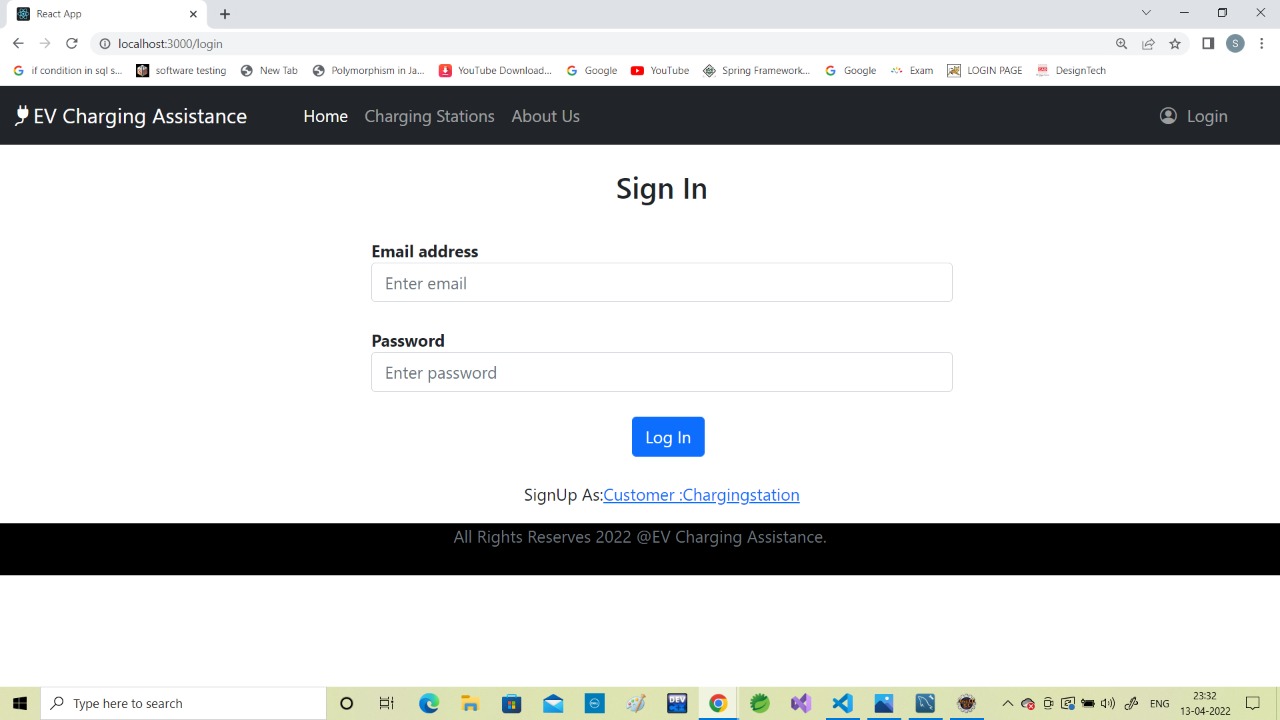
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test case Name** | **Test case description** | **Expected Result** | **Actual result** | **status** |
| 1 | Registration of customers | Customer can register by filling in required details | On successful registration, customer will be directed to the login page. | Customer is directed to the login page. | successful |
| 2 | Registration of charging station | Charging stations can register by filling in required details | On successful registration, charging station will be directed to the login page. | Charging station is directed to the login page. | successful |
| 3 | Admin Login | Admin needs to enter his email id and password for login | Admin can see his homepage | Admin is on his home page | successful |
| 4 | customer Login | Customer need to register first and then enter his customer id and password for login. | Customer can see his homepage | Customer is on his home page | successful |
| 5 | Charging station Login | Charging Station need to register first and then enter its id and password for login. | Charging Station can see its homepage | Charging Station is on his home page | successful |
| 6 | Appointment Booking | Customers will book appointment as per their requirements | After booking appointment, customer will see list of appointments. | Customer is on list of appointments page. | successful |

**7. Screenshots of Webpages**

**Home Page**

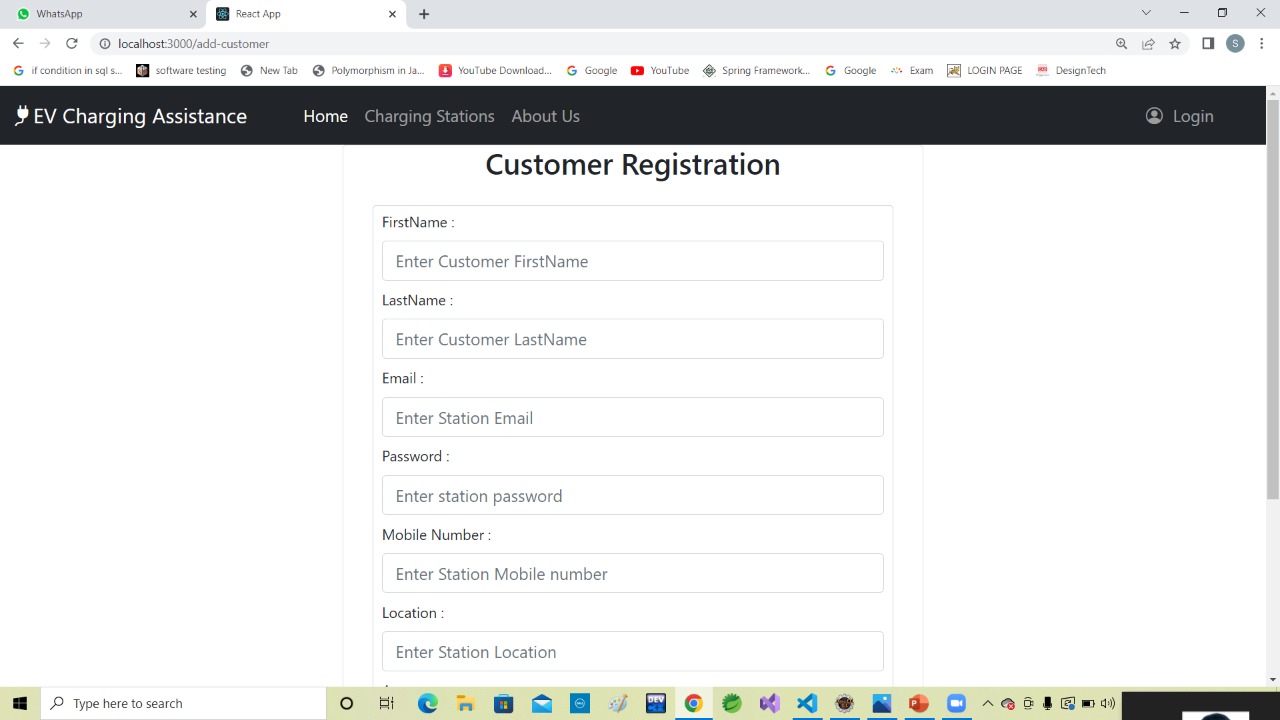


**Login Page**

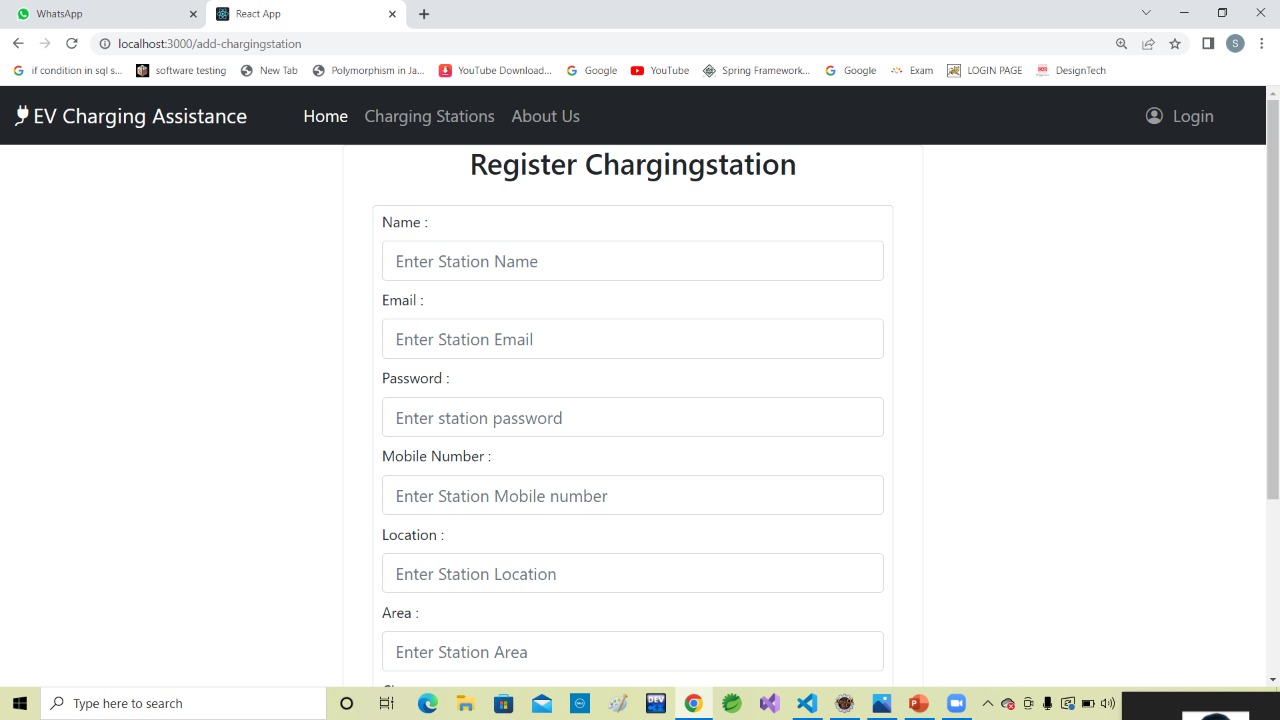


**7. Screenshots of Webpages**

**Customer Registration Form**

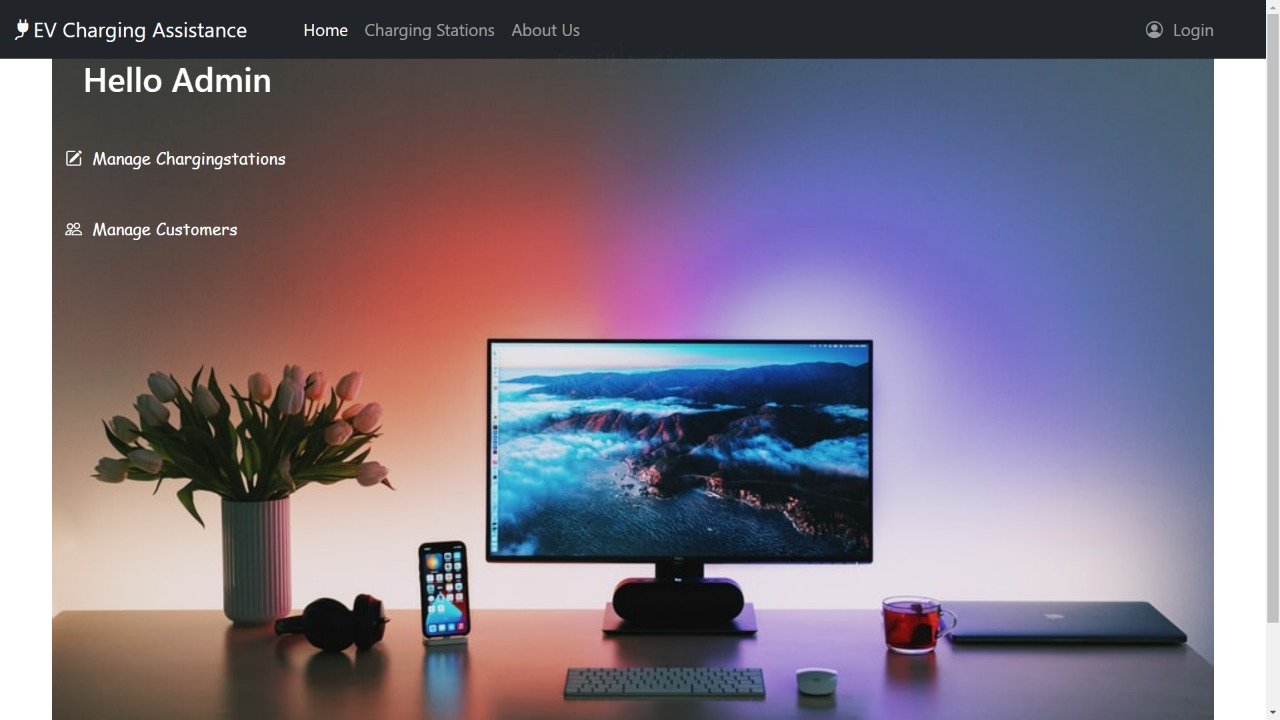


**Charging Station Registration Form**

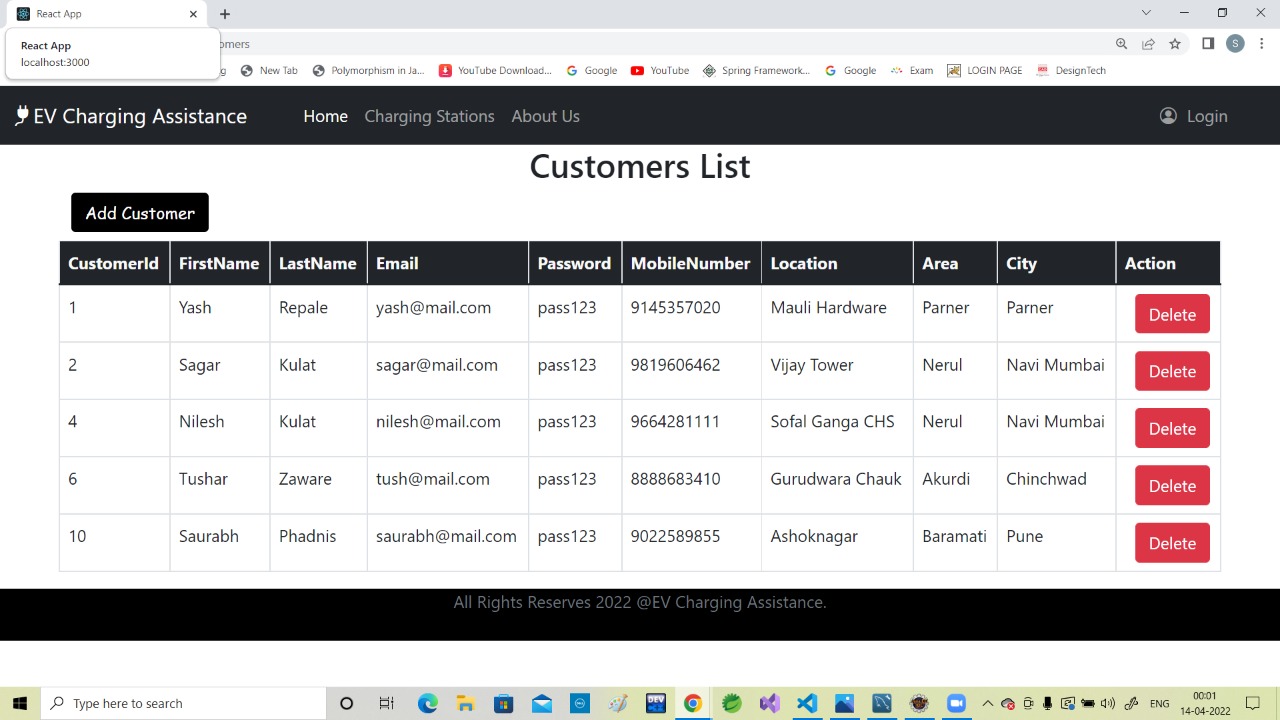


**7. Screenshots of Webpages**

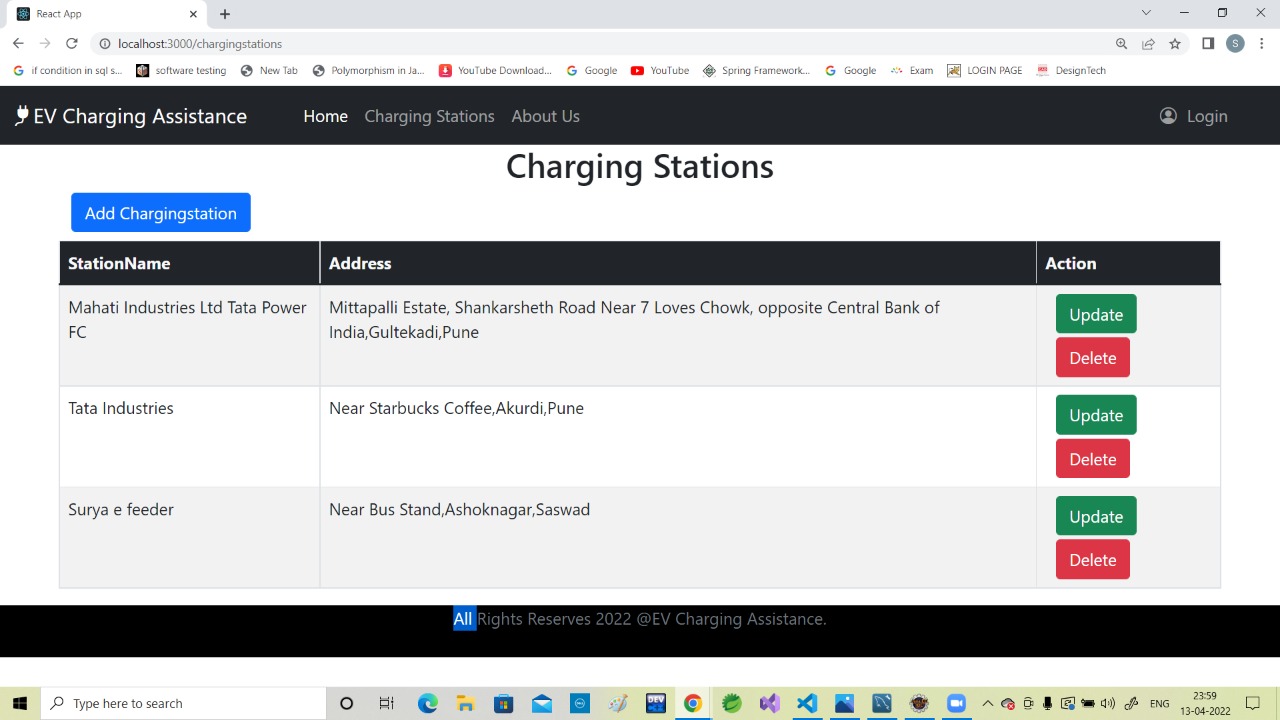
**Admin Page**



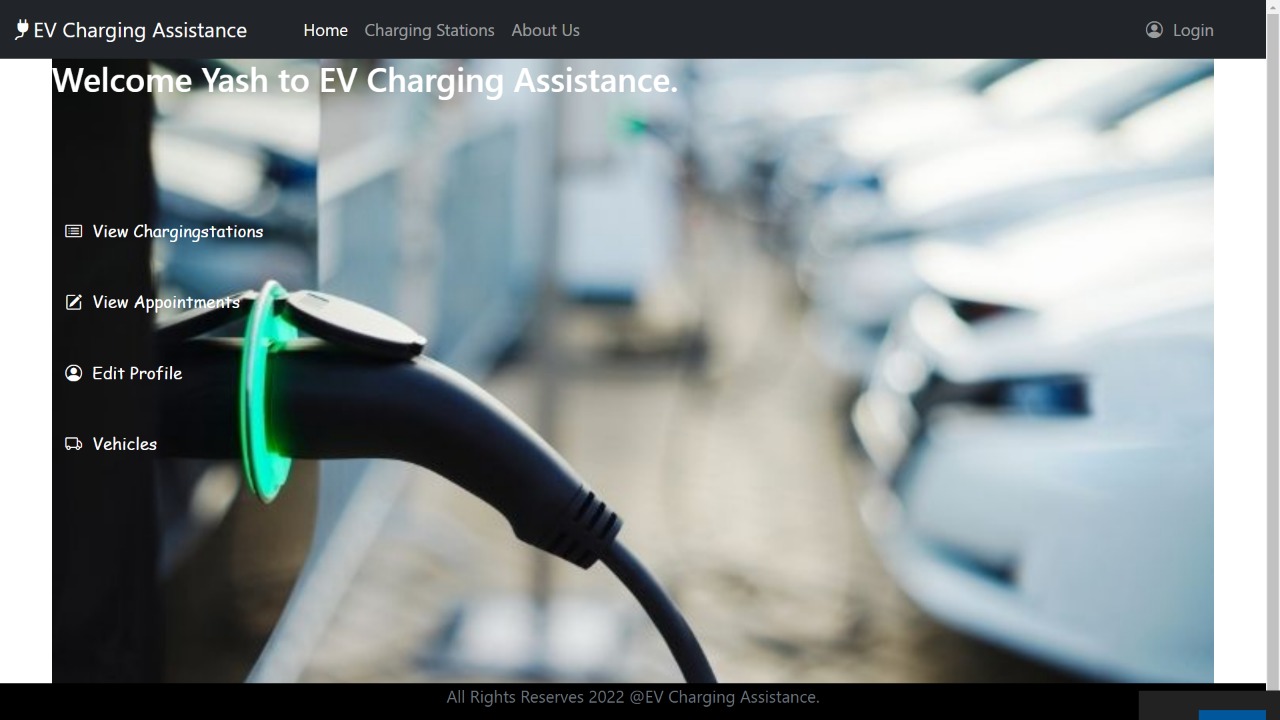
**Manage Customers Page**



**7. Screenshots of Webpages  
Manage Charging Station Page**

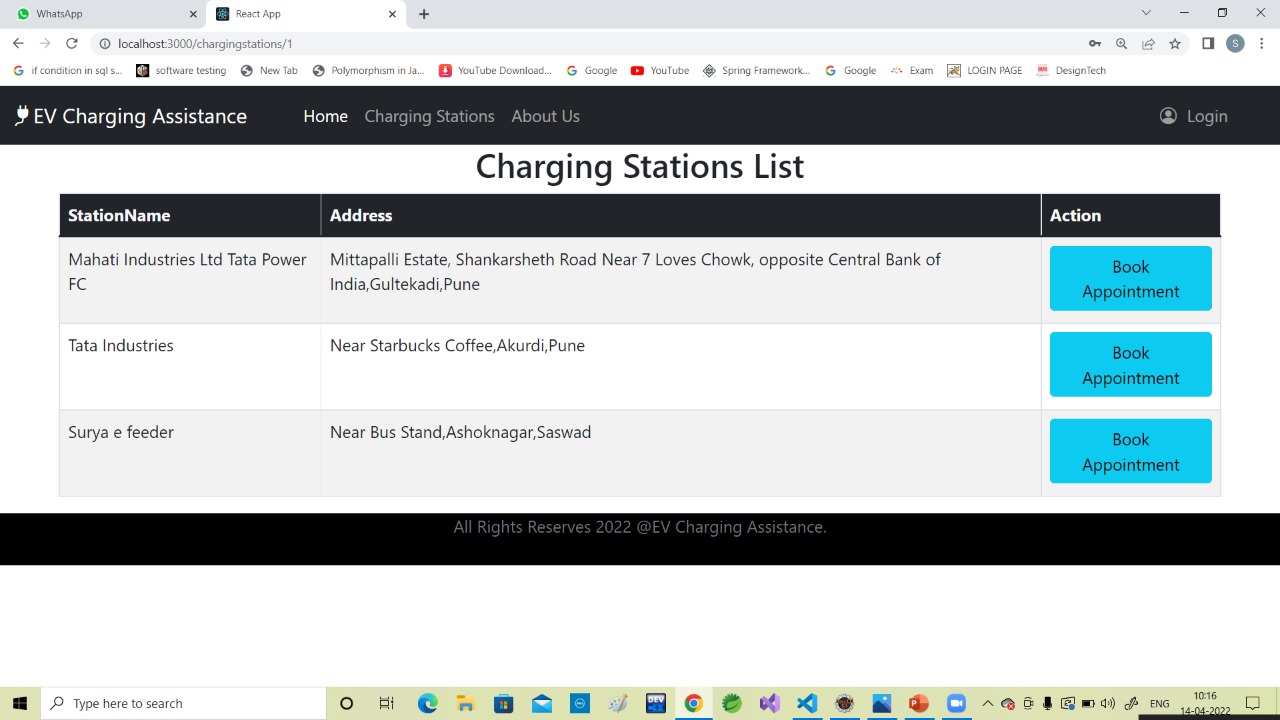


**Customer Page**

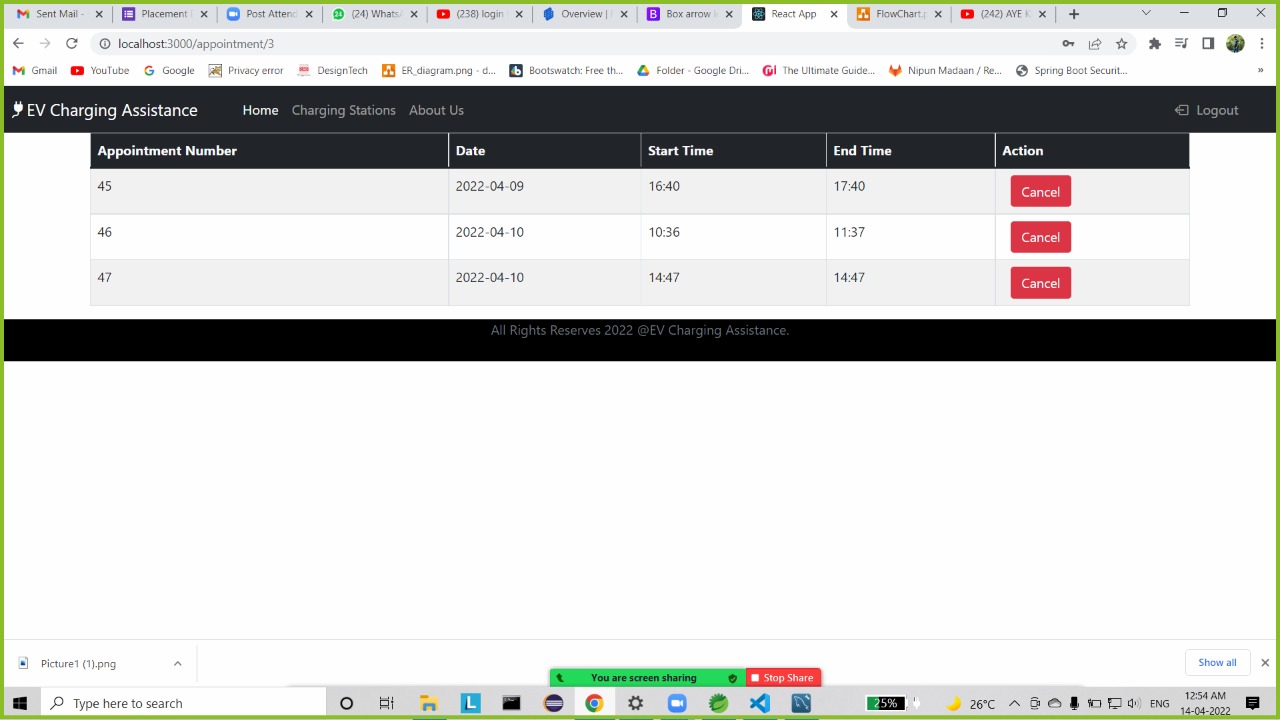


**7. Screenshots of Webpages**

**List of Charging Stations**

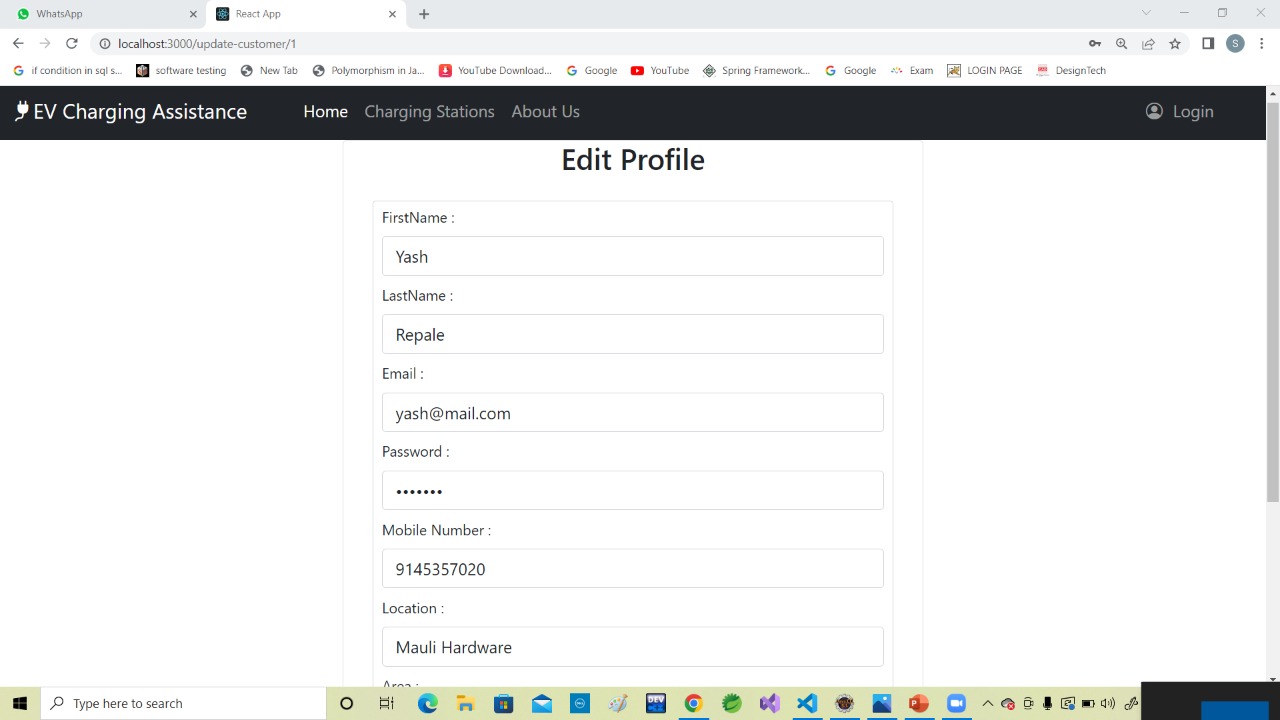


**List of Appointments Page**

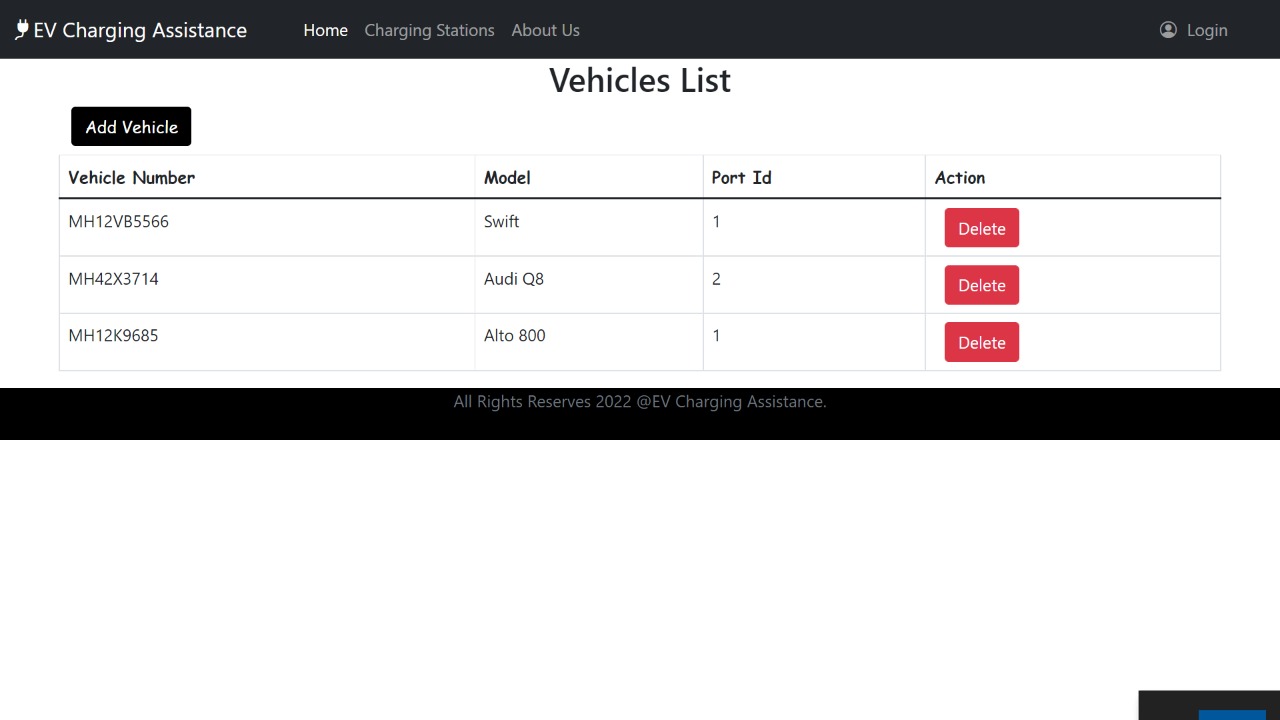


**7. Screenshots of Webpages**

**Edit Customer Profile Page**

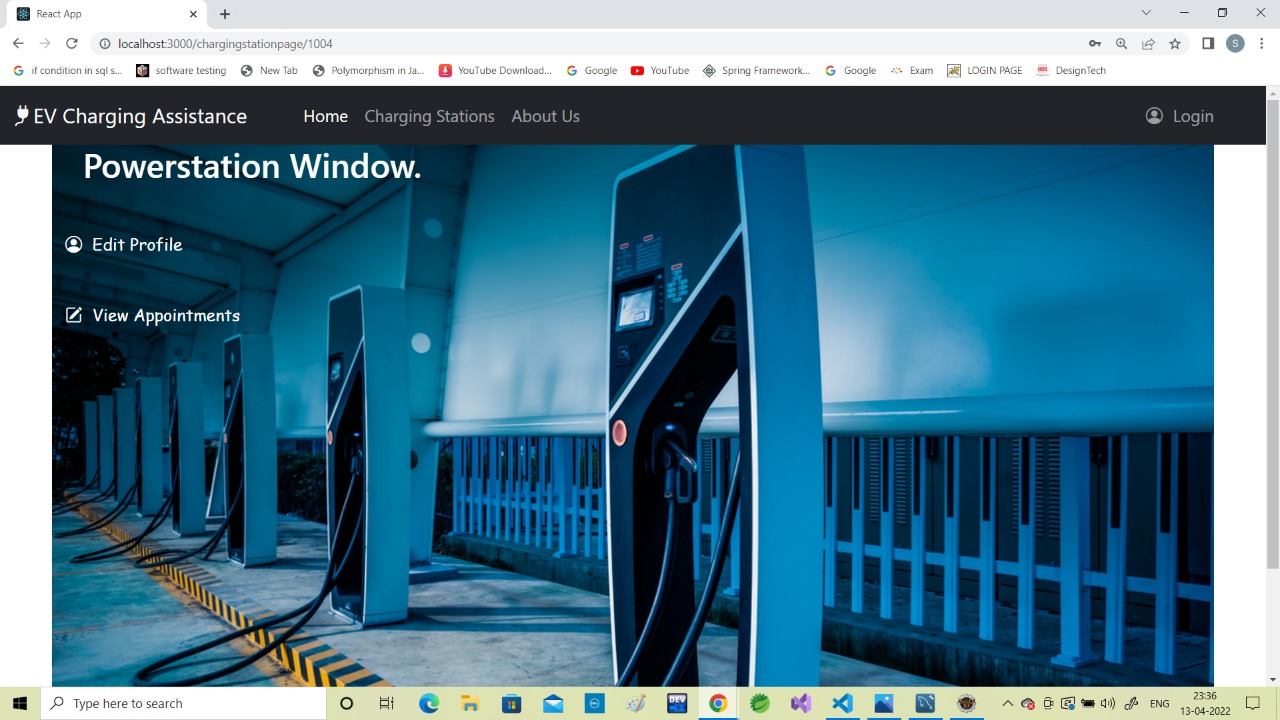


**Vehicle List Page**

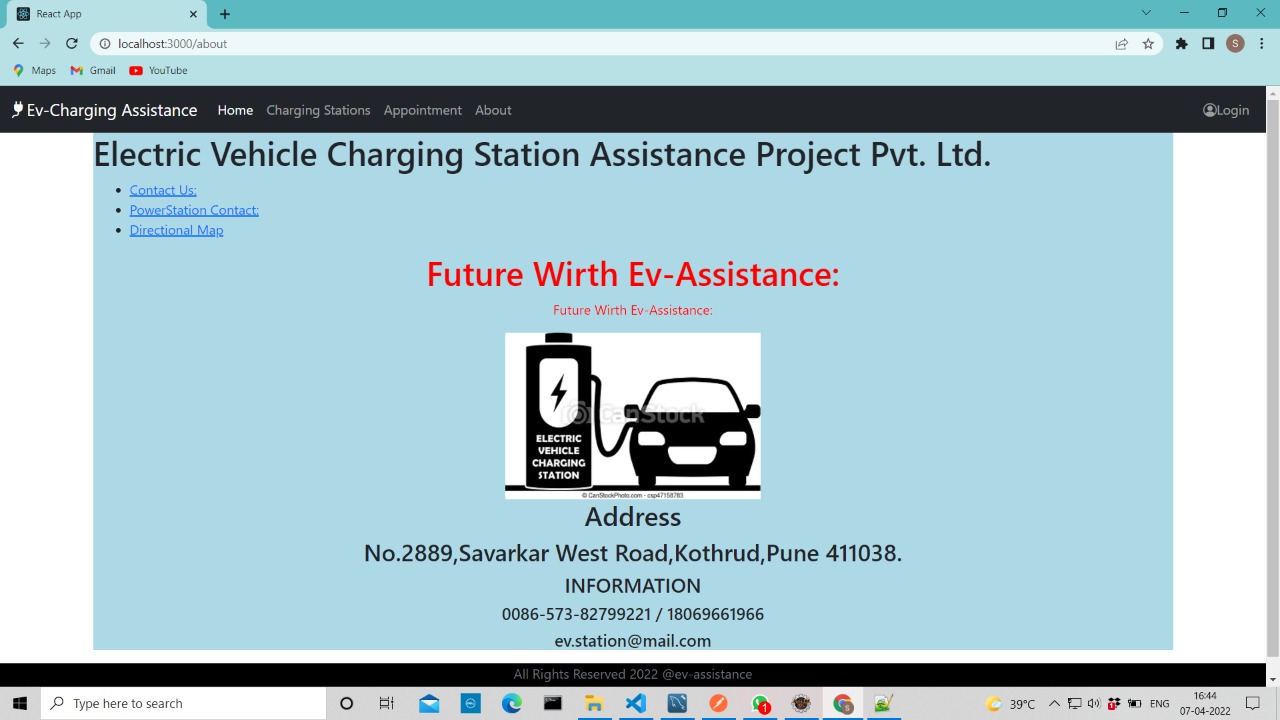


**7. Screenshots of Webpages**

**Powerstation Page**



**About Us Page**



1. **Testing**

To build up our project we used software testing process for executing a program with the intent of finding error that is uncovering errors in a program makes it a feasible task and also typing to find the errors (whose presence is assumed) in a program. As it is a destructive process.

Here we just mentioned that how the testing is related to this software and in which way we have test the software. In our project we have used five types of testing this are listed below;

UNIT TESTING –

Unit testing where individual program units or object class are tested here by using this testing, we have focus on testing functionality of the methods.

MODULE TESTING–

Where this is the combination of unit program is called module. Here we tested unit program is where the module program have dependency.

SUB SYSTEM TESTING –

Then we combined some module for the preliminary system testing in out project.

SYSTEM TESTING –

Where it is combination of two or more sub system and then it is tested here we tested the entire system a per requirement.

ACEEPTANCE TESTING –

Normally this type of testing is done to verify if system meets the customer specified requirements. After submitting this project to the user then they tested and to determine whether to accept the application. It is the system of testing performed by the customer to determine where they should accept the delivery of system.

1. **Conclusion**

With some great features and functionalities, we expect to get many electric vehicles on the market in the future. The rising demand for EVs increases the need for charging stations and charging station assistance apps also. So to create such an app to meet users’ expectations and win the competition, we should look forward to integrate basic functionalities used in this app with user friendly interface and much more features which are discussed within future scope of this documentation.

1. **Future Enhancement**

* T[he payment gateway](https://www.mindinventory.com/blog/integrating-payment-gateway-in-mobile-app/" \t "https://mail.google.com/mail/u/0/" \l "sent/_blank) can be added for user's convenience. For instance, one can integrate to pay with e wallets, UPI, PayPal, debit or credit card, etc.
* After charging electric vehicle, users can give ratings about the charging station. It’s an essential feature for a brand as it helps enhance the service quality.
* AI-based chat-bots can respond efficiently to support and answer users’ queries whenever needed.
* Regular use of an EV charging app giving users some reward points in the forms of promotional coupons, discount vouchers, free one-month membership is considered to create loyalty among users.

1. **Bibliography**

* <https://www.javatpoint.com/spring-boot-tutorial>
* <https://www.w3schools.com/REACT/DEFAULT.ASP>
* https://www.javatpoint.com/mysql-tutorial
* https://www.youtube.com/watch?v=36WoQ1anwM0

**Thank You**