EV Charging Stations Tracking APP

A PROJECT REPORT

submitted By

M MIDHUN

TVE21MCA2040

to

the APJ Abdul Kalam Technological University

in partial fullfilment of the requirements for the award of the degree

of

Master of Computer Applications

Department of Computer Applications

College of Engineering

Trivandrum-695016

DEC 2022

Declaration

I undersigned hereby declare that the project report titled **"EV Charging Stations Tracking APP"** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Smt. Pooja P, Asst.Professor. This submission represents my ideas in my words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum **M MIDHUN**

Date : 24/11/2022

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **EV Charging Stations Tracking APP** submitted by **M Midhun** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor External Supervisor

Head of the Dept

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr Suresh Babu V**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Dr. Deepa S S** , HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to **Smt. Pooja J P**, Asst. Professor, Department of Computer Applications, College of Engineering Trivandrum for her valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank other Asst. Professors in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

M Midhun

Abstract

The transition to electric mobility is a promising global strategy for decarbonizing the transport sector. India is among a handful of countries that support the global EV30@30 campaign, which targets to have at least 30percent new vehicle sales be electric by 2030. An accessible and robust network of electric vehicle (EV) charging infrastructure is an essential pre-requisite to achieving this ambitious transition.

Electric vehicles (EV) can be charged in a variety of ways, depending on location and requirement. Accordingly, charging infrastructure for EVs is of different types and designed for different applications. Specifications and standards for EV chargers, also known as electric vehicle supply equipment (EVSE), vary from one country to another, based on available EV models in the market and the characteristics of the electricity grid.

The primary reason why people don't prefer electric vehicles is because of the unavailability of charging stations. Charging stations, unlike petrol bunks, aren't available everywhere. There always exists a fear as to what might happen if the vehicle runs out of battery. People are worried about more straight forward and faster commuting methods in our country rather than saving the Earth from the ill effects caused by pollution. This project mainly deals with a simple solution to make charging stations more accessible.

The project aims to locate the available EV charging stations near you by accessing the device location. It details data of the selected charging stations, the charger type, electricity grid, price and also the direction to the point. A point to point trip guide with available charging stations en-route has also been facilitated as an additional feature.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As of now electric charging stations are limited in India due to which people can't find the right charging station which will save their time and money. EV charging stations requires space like parks, malls, societies. For private and semi-public charging stations, this space is available in the parking areas of the societies, apartment buildings, or of commercial or public or institutional areas. Due to this there is more difficulty for EV owners to find charging stations nearby them. The problem is not only to find the charging station but also to charge it quickly because of the time required to charge the EV's. This leads to inconvenience of EV users as it requires a lot of time, so need of slot booking is required in the charging of EV's. As electrical vehicle industry is growing in India and less charging stations are available India and also new registrations of charging station is growing so there is no availability of these growing charging station on virtual Maps. This leads to inconvenience of user for finding charging station virtually.

An Electric Vehicle Charging Station Finder App will show the nearby location of charging stations across our locality as well as nearby your destination. We will get various information about the stations such as how many ports are available and how variety of chargers available at that station. The app provides cost of the charging of car at the station, the type charging grid used and public rating. The users can contribute to app also by adding a new electric vehicle charging station as they discover.

Chapter 2

Problem Defnition and Motivation

When a customer buys an electric car, the maintenance of these cars is not like the ordinary cars. One have to seek some help such as an Electric vehicle charging station finder app to find charging stations. An electric vehicle charging station finder app can save our time to find these charging station rather than search independently. One cannot find charging stations like the petrol or diesel or CNG station which are available everywhere. Due to this problem we have to plan the refuelling (charging) of these cars, but with the help of this app which directly navigate us to nearby EV charging stations.

An Electric Vehicle Charging Station Finder App will show the nearby location of charging across stations across our locality as well as nearby your destination. We will get various information about the stations such as how many ports are available and how variety of chargers available at that station. The app provides, real-time availability of the stations, photos of the stations, and cost of the charging of car at the station.

Chapter 3

Requirement Analysis

3.1 Purpose

In this project I will design and develop an app which will find nearby charging stations of users locality. The app will show all nearby electric vehicle charging stations. The user can directly navigate to this charging stations. This app will provide a facility of booking slots for charging the Electric vehicle of the users based on the type and charging port of their vehicle in their convenient time slots. This app will save a lot of time of Electric vehicle owners.

3.2 Overall Description

An electric vehicle charging station finder app can save our time to find these charging station rather than search independently. One cannot find charging stations like the petrol or diesel or CNG station which are available everywhere. Due to this problem we have to plan the fuelling (charging) of these cars, but with the help of this app we can directly navigate to nearby EV charging stations. An Electric Vehicle Charging Station Finder App will show the nearby location of charging across stations across our locality as well as nearby your destination. We will get various information about the stations such as how many ports are available and how variety of chargers available at that station. The app provides, real-time availability of the stations, photos of the stations, and cost of the charging of car at the station. The users can contribute to app also by adding a new electric vehicle charging station as they discover.

3.2.1 Hardware Requirements

- Processor : Intel Core Pentium
- Storage : 512 GB Hard Disk space
- Memory : 4 GB RAM

3.2.2 Software Requirements

- Operating System : Linux/Windows
- Platform : Node JS
- Technology used : React , nodejs , MongoDB, Express and MapBox

3.3 Functional Requirements

The functional requirements includes all the activities or processes that should be achieved by the proposed system. It includes

- **React:** React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.
- **Node JS:** Node.js is an open-source server environment.It is cross-platform and runs on Windows, Linux, Unix, Mac OS, etc. It is a back-end JavaScript runtime environment. Node.js runs on a JavaScript Engine (i.e. V8 engine) and executes JavaScript code outside a web browser.
- **Express:** Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js

- **MongoDB:**MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.JavaScript can be used in queries, aggregation functions (such as MapReduce), and sent directly to the database to be executed.

- **Mapbox:**Mapbox is an American provider of custom online maps for websites and applications.It has rapidly expanded the niche of custom maps, as a response to the limited choice offered by map providers such as Google Maps.

3.4 Non Functional Requirements

3.4.1 Performance Requirements

- Accuracy : Accuracy in functioning and the nature of user-friendly should be maintained by the system.

- Speed : The system must be capable of offering speed.

- Low cost: This system is very cheap to implement and is also user-friendly.

- Less Time consuming: It uses very less time comparing to the existing sysytem .

- User Friendly: This proposed system is highly user friendly they enables to create a good environment.

3.4.2 Quality Requirements

- Scalability : The software will meet all of the functional requirements.

- Maintainability : The system should be maintainable. It should keep backups to atone for system failures, and should log its activities periodically.

- Reliability : The acceptable threshold for down-time should be large as possible. i.e. mean time between failures should be large as possible. And if the system is broken, time required to get the system backup again should be minimum.

- Availability: This system is easily available as the core equiments in building the sofware is easily obtained.

- High- Functionality: This system is highly functional in all environment since, They are highly adaptable.

Chapter 4

Design And Implementation

The proposed system is used to guide users from their source point to destination displaying the available EV Charging stations en-route and to track the nearby charging stations

4.1 Overall Design

The proposed system follows client server architecture. That is the EV charging tracking system has a client part and a server part as well. The client part is used by the user to track the nearby charging stations. The input is passed to server and the evaluated result is given back to the client. The server side is developed in Nodejs and the client side is built using React.

4.1.1 System Design

The system is web based. The input is taken from the user through a web page and the input is passed to the nodejs program running in the server side. The server program perform tasks such as query processing and gathering data from the database. The results of these processes are shown to the users through an interactive Mapbox.

The Admin can use this same web app to login and add the new charging stations which will be updated in the database.

4.1.2 Methodology

This project has two modules. One is the user module and other for the admin.This project is a progressive web app which is a single page application developed using MERN stack.

The first page is the user interface which is an interactive portfolio web page with four different components navigation bar, intro section, details and descriptions and footer respectively. It is from the intro section where the user is taken to the map area where the nearby ev charging charging stations are displayed as markers and on clicking the markers a popup is displayed showing the details of the charging stations which include the title, address, charger type, number of ports , price of charging and review.

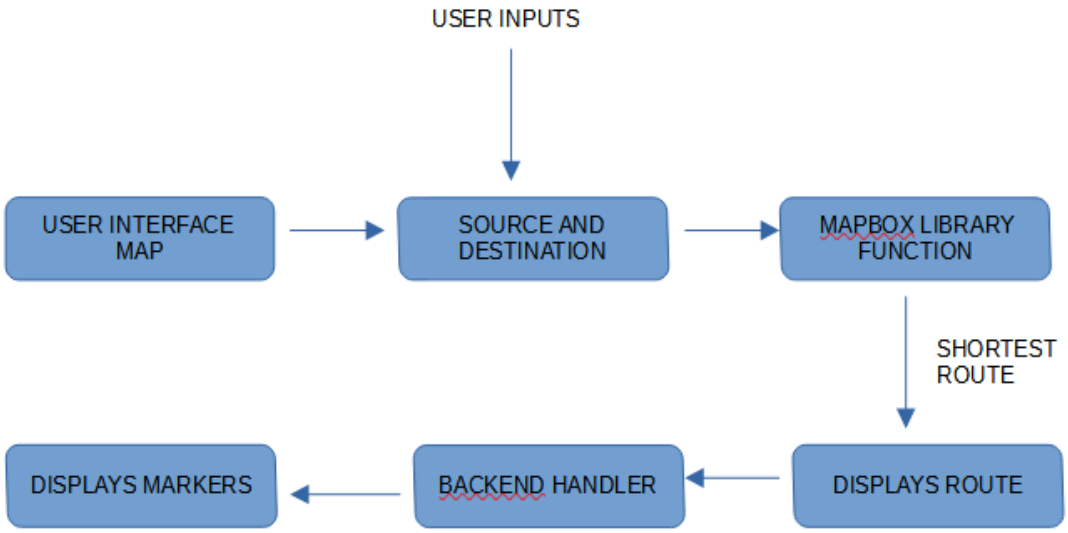

Figure 4.1: Working of the project

- **Directions:**A Direction bar is available on the map where the user enters the source and destination upon which the routing is displayed along with the nearby charging stations en-route.

- **Backend:**The backend server is connected to the frontend via nodejs and its framework express. The server used is of mongoDB where the details and descriptions of the charging stations are stored and retrieved.

- **Testing:** In testing phase we test the project by adding data to database and try to retrieve it by showing the markers . The user enters source and destination and the corresponding route is successfully displayed with charging stations en-route.

The second part of the project is the admin module. The admin has a login portal where he logs in using the log in credentials which is compared to the details in the database.on successfull

log in a map area is displayed which contains all the available charging stations along with its descriptions on clicking any marker.It is the admins job to add new charging stations which is done through the admin module. On double clicking on the map a marker along with a form appears which isto be filled an submitted . This is the details of the new charging stations which is stored in the database. The updated map is visible at the same time by the user.

4.2 Data Flow Diagram

DFD is one of the graphical representation techniques used in a project to show the flow of the data through a project. DFD helps us to obtain an idea about the input, output, and process involved. The things absent in a DFD are control flow, decision rules, and loops. It can be described as a representation of functions, processes that capture, manipulate, store, and distribute data between a system and the surrounding and between the components of the system. The visual representation helps for good communication.

It shows the journey of the data and how will it be stored in the last. It does not provide details about the process timings or if the process shall have a parallel or sequential operation. It is very different from a traditional flow chart or a UML that shows the control flow or the data flow.

In level 0 the basic data flow of the application is showcased. It does not show the flow of data much deeper. It will be evaluated in the higher levels of Data Flow Diagram. The Data Flow Diagram of EV Charging Stations Tracking App is shown below.

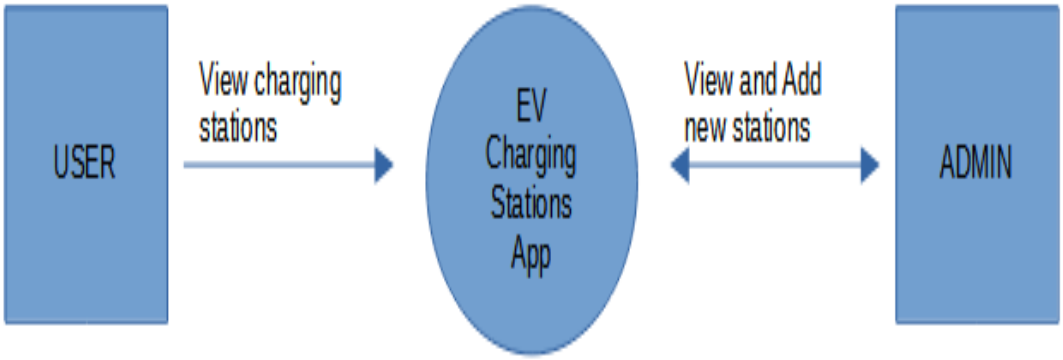

Figure 4.2: Level 0 DFD

The diagram shows Level 0 Data flow diagram of the EV Charging Stations Tracking App. As the diagram indicates there is a user part and an admin one. The user is introduced to a web site where he/she is taken to the map which displays the available charging stations near the location. The user can also navigate from source to destination using mapbox directions API which also displays the nearby charging stations en-route. The Admin has a portal where he can add new charging stations by just doubleclicking on the map area and filling the details which is uploaded to the database.

4.3 Screenshots of user interface

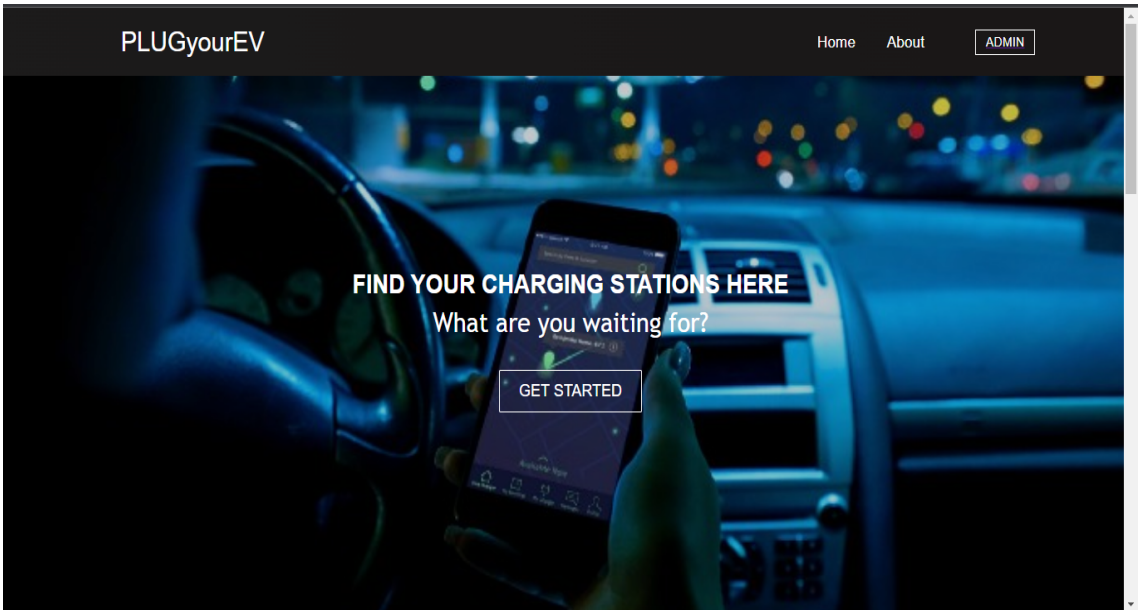

Figure 4.3: Home page

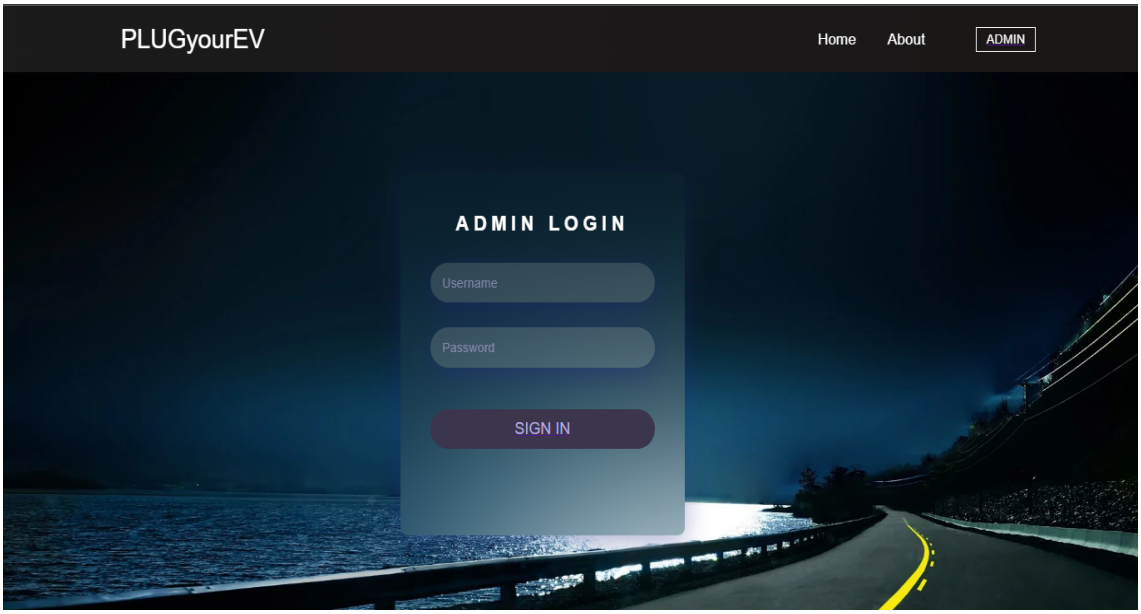

Figure 4.4: Admin login

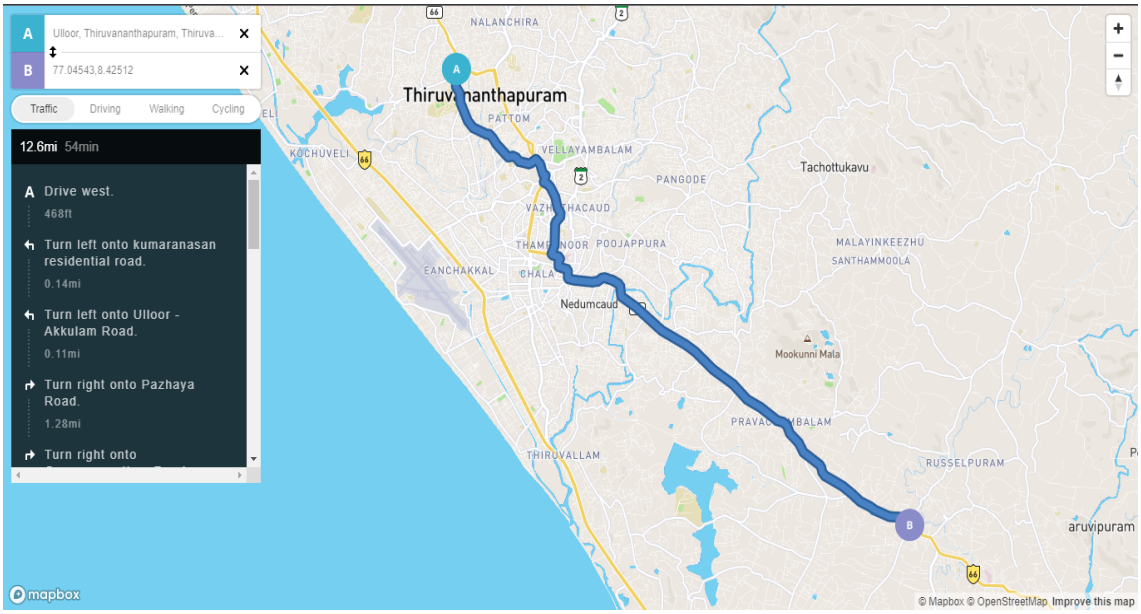

Figure 4.5: Map area

Chapter 5

Coding

```
function Maparea() {
  const [pins, setPins] = useState([]);
  const [currentPlaceId, setCurrentPlaceId] = useState(null);

  const [viewport, setViewport] = React.useState({});
  useEffect(() => {
    const getPins = async () => {
      try {
        const allPins = await axios.get("/pins");
        setPins(allPins.data);
      } catch (err) {
        console.log(err);
      }
    };
    getPins();
  }, []);


  const handleMarkerClick =(id)=>{
      setCurrentPlaceId(id);
  };

  return (
    <div>
      <Navbar />
      <Map
        mapboxAccessToken="pk.eyJ1IjoibW1pZGh1biIsImEiOiJjbDlweHZ3eHAwM2kyM29sNzhpb3g0cjJoIn0.k7tqrkmM20x27lx3bXZ4Jw"
        initialViewState={{
          longitude: 76.93792505109217,
          latitude: 8.525794306303943,
          zoom: 12
        }}
        style={{width: "100%", height: "89vh"}}
        mapStyle="mapbox://styles/mapbox/streets-v9"
      >
      <NavigationControl position='bottom-right' />
```

Figure 5.1: Map Area Code

```
import React from 'react';
import '../App.css';
import { Button } from './Button';
import './HeroSection.css';

function HeroSection() {
  return (
    <div className='hero-container'>
      <h1>FIND YOUR CHARGING STATIONS HERE</h1>
      <p>What are you waiting for?</p>
      <div className='hero-btns'>
        <Button
          className='btns'
          buttonStyle='btn--outline'
          buttonSize='btn--large'
        >
          GET STARTED
        </Button>

      </div>
    </div>
  );
}

export default HeroSection;
```

Figure 5.2: Home page code

Chapter 6

Testing and Implementation

6.1 Testing and various types of testing used.

Once a software is developed, the major activity is to test whether the actual results match with the experimental results. This process is called testing. It's used to make sure that the developed system is defect free. The main aim of testing is to find the errors and missing operations by executing the program. It also ensure that all of the objectivs of the project are met by the developer. The objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiency, usability and accuracy of it. It aims to measure the functionality, specification and performance of a software program. Tests are performed on the created software and their results are compared with the expected documentation. When there are too much errors occurred, debugging is performed. And the result after debugging is tested again to make sure that the software is error free. The major testing processes applied to this project are unit testing, integration testing and system testing. In unit testing, our aim is to test all individual units of the software. It makes sure that all of the units of the software works as it intended. In integration testing, the combined individual units are tested to check whether it met the intended function or not. It helps us to find out the faults that may arise when the units are combined. In system testing the entire software is tested to make sure that it satisfies all of the requirements. The tables shown below describes the testing process occurred during the development of this project "EV Charging Stations Tracking App". This defines the various steps took to create the project error free.

6.1.1 Unit Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	create the user interface	To load the web page with required fields	Same as expected	Pass
2	node server program	mongoDB connection established	Same as expected	Pass
3	display stations	display markers with details of charging stations	same as expected.	Pass
4	add stations	admin adding stations and details from map area	data added to database	Pass
5	retrieving data from database	user enters source and destination and corresponding route is displayed	markers displayed along the route	Pass

Table 6.1: Unit test cases and results

6.1.2 Integration Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	load the user interface from node	the user interface is loaded when we run the MERN stack program	Same as expected	Pass
2	display markers	user enters source and destination and corresponding route is displayed and markers displayed along the route	Same as expected	Pass
3	add stations	admin adding stations and details from map area	data added to database successfully	Pass
4	display markers	the updated map is viewed by user	data extracted successfully	Pass

Table 6.2: Integration cases and result

6.1.3 System Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	to run node server	Server program executed successfully, hence the backend program worked without any crash	Same as expected	Pass
2	to run react server	Server program executed successfully, hence the frontend program worked without any crash	Same as expected	Pass

Table 6.3: System test cases and results

Chapter 7

Results and Discussion

The main aim of the project was to display the electric vehicle charging stations available near our location . And it is observed that the system performs all the functionalities as expected. By using this web application users can track the nearest charging stations when they plan for a trip from any given source to destination

7.1 Advantages and Limitations

The proposed system is a web application which shows the Electric Vehicle charging stations available and displays the details of the charging stations. The proposed system posses more advantages over the existing system. The proposed system save a huge amount of time as it is specifically aimed for EV customers. Like every other system, this system also have it's own disadvantages. But they are negligible while comparing with the advantages and they will be overcome in future.

7.1.1 Advantages

- Can save the time needed in looking for charging stations as they are not available everywhere .
- The Human resource needed for finding the location can be saved.
- The admin can just login to the app and add any new charging stations rather than working in the backend.

- The users can view the charging stations available en-route which is not available in most of the other apps.
- Being a progressive web app it can be used in any device irrespective of the platform.

7.1.2 Limitations

- The current dataset is comparatively small as it is limited to only Kerala. It can be improved by adding more charging stations into the database.
- The availability of the charging stations whether it is free ,occupied or booked is unknown.

Chapter 8

Conclusion and Future Scope

This App will show the nearby location of charging stations across our locality as well as nearby your destination. We will get various information about the stations such as how many ports are available and how variety of chargers available at that station. The app provides cost of the charging of car at the station, the type charging grid used and public rating.

The additional features added to this is that the users can track the stations when they are on a trip. . The majority of the project was built in React.The backend database used is mongoDB. It uses nodejs to connect the backend to the user interface built using Reactjs.

The feature scope of this project can be extended to multiple dimensions. Future development on this topic may lead to the addition of a booking system through which the users can reserve the charging stations much before arrival. The payment option can also be included using which the users need not download the service equipment's app separately each time to pay. Another possible addition to the app would be a user log in through which the users can keep track of their EV usage and spending.

Bibliography

[1] https://docs.mapbox.com/api/navigation/directions/

[2] https://visgl.github.io/react-map-gl/docs/api-reference/map-provider

[3] https://docs.mapbox.com/mapbox-gl-js/example/mapbox-gl-directions/

[4] Full-Stack React Projects - Second Edition by Shama Hoque ,Released April 2020

[5] Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker ,by Frank Zammetti (Author)