

A

PROJECT WORK REPORT

On

**“SMART MANAGEMENT OF EV CHARGING STATIONS
USING AI CHATBOT AND GMAPS API”**

Submitted to the Savitribai Phule University, Pune
In partial fulfillment of the requirements for the award of the degree

B.E. (INFORMATION TECHNOLOGY)

BY

Mr. DESHPANDE YASH NARENDRA

(SEAT NO. B150108521)

Mr. BHAKARE MAHESH SANTOSH

(SEAT NO. B150108506)

Mr. CHAUDHARI ABHIJIT AMBADAS

(SEAT NO. B150108511)

Ms. AMRUTKAR KALYANI KIRAN

(SEAT NO. B150108501)

Under the Guidance of

Dr. MARKAD A.V.



DEPARTMENT OF INFORMATION TECHNOLOGY

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

A/P: GHULEWADI, SANGAMNER, AHMEDNAGAR, PIN - 422608

ACADEMIC YEAR 2021-22

DEPARTMENT OF INFORMATION TECHNOLOGY
AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER
A/P: GHULEWADI, SANGAMNER, AHMEDNAGAR, PIN - 422608
ACADEMIC YEAR 2021-22



CERTIFICATE

This is to certify that Project Work Report entitled

**“SMART MANAGEMENT OF EV CHARGING STATIONS
USING AI CHATBOT AND GMAPS API”**

is submitted as partial fulfillment of curriculum of the B.E. of Information Technology

by

Mr. DESHPANDE YASH NARENDRA
Mr. BHAKARE MAHESH SANTOSH
Mr. CHAUDHARI ABHIJIT AMBADAS
Ms. AMRUTKAR KALYANI KIRAN

(SEAT NO. B150108521)
(SEAT NO. B150108506)
(SEAT NO. B150108511)
(SEAT NO. B150108501)

Dr. Markad A.V.
Project Guide

Dr. Markad A.V.
Project Coordinator

Dr. Gunjal B.L.
H.O.D. I.T.

Dr. M. A. Venkatesh
Principal

Savitribai Phule Pune University



CERTIFICATE

This is to certify that

Mr. DESHPANDE YASH NARENDRA

(SEAT NO. B150108521)

Mr. BHAKARE MAHESH SANTOSH

(SEAT NO. B150108506)

Mr. CHAUDHARI ABHIJIT AMBADAS

(SEAT NO. B150108511)

Ms. AMRUTKAR KALYANI KIRAN

(SEAT NO. B150108501)

Students of B.E. Information Technology were examined in Project Work entitled

**“SMART MANAGEMENT OF EV CHARGING STATIONS
USING AI CHATBOT AND GMAPS API”**

On

/ / 2022

At

DEPARTMENT OF INFORMATION TECHNOLOGY

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

YEAR 2021-22

(Internal Examiner)

(External Examiner)

Certificate by Guide

This is to certify that

Mr. DESHPANDE YASH NARENDRA

(SEAT NO. B150108521)

Mr. BHAKARE MAHESH SANTOSH

(SEAT NO. B150108506)

Mr. CHAUDHARI ABHIJIT AMBADAS

(SEAT NO. B150108511)

Ms. AMRUTKAR KALYANI KIRAN

(SEAT NO. B150108501)

has completed the Project work entitled “**SMART MANAGEMENT OF EV CHARGING STATIONS USING AI CHATBOT AND GMAPS API**” under my guidance and that, I have verified the work for its originality in documentation, problem statement, literature survey and conclusion presented in project work.

Date - / / 2022

Place - Sangamner

**Dr. Markad A. V.
Department of I.T.
AVCOE, Sangamner**

ACKNOWLEDGEMENT

We would like to take this opportunity to express our respect and deep gratitude to our guide **Dr. Markad A. V.** for giving me all necessary guidance required, for this project stage-I apart from being constant source of inspiration and motivation. It was our privilege to have worked under him.

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this project report and who influenced our thinking, behavior, and acts for this report.

We are thankful to **H.O.D. Dr. B. L. Gunjal** for her support, cooperation, and Motivation provided to us during the project for constant inspiration, guidance and blessings. I am highly obligated to my entire friends, whose contribution intellectually and deeds for preparation of this project stage-I.

Last but not the least I would like to thank **Dr. M. A. Venkatesh**, Principal and All those people who helped us in making this report.

Mr. DESHPANDE YASH NARENDRA
Mr. BHAKARE MAHESH SANTOSH
Mr. CHAUDHARI ABHIJIT AMBADAS
Ms. AMRUTKAR KALYANI KIRAN
AVCOE, Sangamner-422608

ABSTRACT

In Recent years car companies like TATA, TESLA introduced and launches new electric cars in the market. For charging these cars some of the stations are also set up. But considering the current situation, these cars take at least 15 minutes to half an hour to charge. If station is full and all the slots are filled previously then other customers have to wait for a long time. Our idea is to develop a system which will solve these kinds of issues.

We are developing a system in which we going to connect all the electric car charging stations together. By using our system user can find the station according to their choice and it will be useful for those who want to travel for long distance with their EV cars and it will be time saving. It will be very easy to use. If the given time slot is available then your place for the given slot will be booked. Otherwise system will ask to enter the new time schedule. In this system user has to pay some percent of amount online to confirm their booking. Our system will also provide shortest map route to reach at given station.

Our system will also provide interface for charging stations to view all available slots as well as booked slot lists and manage slot timing. We are going to develop this system for Android based devices. To develop this system, we are going to use time-slot allocation techniques as well as Google maps API for direction sensing. Our Chabot system will Control software via vocal commands. With the help of online payment gateway user can pay money quickly. By using the system peoples will save their so much time and they can view and book appropriate station easily.

Keywords: Smart management, charging slot, EV Cars, Map.

INDEX

	Page No.
Acknowledgement	i
Abstract	ii
Table of Contents	iii
List of Figures	v
List of tables	vi
List of Abbreviation	vii
1. Introduction	1
1.1 Background	1
1.2 Motivation	1
1.3 Aim	2
1.4 Objective	2
1.5 Organization Of project report	3
2. Literature Survey	4
3. Requirement and Analysis	6
3.1 Problem Statement	6
3.1.1 Description	6
3.2 Software Requirement Specification (SRS)	6
3.2.1 Introduction	6
3.2.2 Functional Requirements	6
3.2.3 External interface Requirements	7
3.2.4 Non-Functional Requirements	7
3.2.5 System Requirements	8

3.2.6 Hardware Requirements	9
3.2.7 Software Requirements	9
3.2.8 Setup	9
3.3 Software Engineering Methodology	10
3.3.1 Software Life-Cycle	10
3.3.2 Applications of Waterfall Model	10
3.3.3 System Implementation Plan	11
4. Design	12
4.1 System Architecture	13
4.2 Use Case Diagram	14
4.3 Class Diagram	16
4.4 Components Diagram	17
4.5 Sequence Diagram	18
4.6 Data Flow Diagram	19
4.7 Deployment Diagram	20
5. RESULTS AND DISSCUSSION	21
5.1 Testing	21
5.2 Types of Testing	21
5.3 Test strategy and approach	23
5.4 Test Cases	24
5.5 Application Results	26
6. Conclusion	31
6.1 Conclusion	31
References	32

LIST OF FIGURES

3.1	Waterfall Model	10
4.1	Booking System Architecture	13
4.2	Station System Architecture	13
4.3	Use case Diagram	15
4.4	Class Diagram	16
4.5	Components Diagram	17
4.6	Sequence Diagram	18
4.7	Data Flow Diagram	19
4.8	Deployment Diagram	20

LIST OF TABLES

3.1	System Implementation Plan	11
5.1	Test Case Results	24

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
EV	-	Electric Vehicle
NLP	-	Natural Language Processing
API	-	Application Programming Interface

CHAPTER 1

INTRODUCTION

1.1 Background

In recent years, global warming and the depletion of fossil fuels due to mass consumption of energy resource has become an increasingly recognized world problem. To control these problems, the installation of renewable energy systems, which do not depend on fossil fuels, is an effective countermeasure. In Japan, since the government has introduced Feed-in tariffs (Fit), the introduction of photovoltaic systems has been expanded rapidly. However, the output power from increased number of photovoltaic systems is extremely large and tends to have a bad effect on the system frequency and distribution voltage.

To address this problem, the Japanese government has begun reconsidering the Fit system. Adding to this problem, the cost of PV installation is decreasing year by year. Therefore, in the future, the price of PV power is expected to decrease greatly. In this study, EV charging stations that near-exclusively purchases power from PV systems on smart houses and sells power to electric vehicles (EV) and smart houses is proposed as an aggregator. The EV charging station has the need to utilize a fixed battery for electricity trading. A large number of public charging station networks may not come shortly in some urban areas due to many constraints such as financial, technological limitations etc. For instance, investing in fast charging infrastructure may be improbable in the places with lower EV adoption rates because it is not economically viable. In coming years, EVs and autonomous vehicles (AVs) will be imperative for the infrastructure and layout of cityscape in Smart cities.

Constructing public EV charging infrastructure in Smart cities would substantially encourage EV users to use EVs as well as promote commercial electric fleets, for instance, electric taxi and ride sharing. Thus, a higher EV adoption rate can be achieved if adequate such infrastructures are accessible.

- As we know EV Automobiles going to be future of the world but these machines need charging stations for charging.
- In this project work, system will provide the platform to book charging slots to available charging station according to need of customer.
- In this system user will get facilities like AI Chabot to book station via vocal commands, Maps features for direction sensing, Digital payment option, Notifications, Mails and SMS of each activity.
- 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, This chapter explains the technical concepts of electric vehicle charging infrastructure, and highlights the need for a contextual approach to local planning and implementation of EV charging networks.

1.2 Motivation -

The main purpose of carrying out of this project is to develop a hybrid web-based android application for the EV car owners to avail the charging station slot booking system to book slot of charging station prior of going to charge the vehicle. In addition, we propose an AI voice Chabot in the system to interact user to system via vocal commands. As well as system also provides shortest route to reach to destination through GMAPS API.

1.3 Aim-

To design and develop a web-based android application to book the charging slots to charge the electric vehicle. The system also proposes the slot booking according to charging socket type.

1.4 Objectives –

- To available a charging slots booking system for EV car owners.
- To design simple and adaptive interface which easily understandable to everyone.
- To connect multiple charging stations together via single system.
- To available a nearest stations details and route to reach at station via maps.
- To implement a system this will make easy to book and charge electric cars for car users.

1.5 Organization Of project report

Chapter 1: In this chapter, Introduction of the project i.e. what is need, relevance and what is actual project idea.

Chapter 2: In this chapter, we briefly review the literature. Literature review is a summary of studies related to a particular area of research

Chapter 3: Requirement Analysis describes the things require to develop the software, approach to be follow to develop software and requirements to run software after deployment

Chapter 4: In this chapter, we will design the different kinds of diagrams to understand the actual development flow of project.

Chapter 5: In this chapter, we have discussed about the different testing's that we have done on the system.

Chapter 6: In this chapter, we have concluded our project and we have provided last overview about the complete system.

CHAPTER 2

LITERATURE SURVEY

1. “Smart Electric Vehicle Charging Management for Smart Cities” Designed and implemented a smart EV charging management system utilizing charging strategy that includes effective reservation management and efficient slot allocation of charging stations. Considering composite cost that includes waiting time, estimated charging time, estimated charging cost, user discontent factor and charging station congestion impact in such a method, our scheduling scheme shall furnish a set of optimal solutions. [1]
2. “Voice Control Device using Raspberry Pi”. This voice control device uses Raspberry Pi. It uses speech-to-text engine to convert voice into a text. It uses Query Processing; with the help of NLP it extracts the required meaningful text from the command given by user. With the help of text to speech engine it gives the output to user in the form of voice. The limitations of this system are that, it requires the Raspberry Pi as a hardware device and we are implementing this algorithm directly on android device. [2]
3. Voice control is a major growing feature that change the way people can live. The voice assistant is commonly being used in smartphones and laptops. AI-based Voice assistants are the operating systems that can recognize human voice and respond via integrated voices. This voice assistant will gather the audio from the microphone and then convert that into text, later it is sent through GTTS (Google text to speech). GTTS engine will convert text into audio file in English language, then that audio is played using play sound package of python programming Language. [3]

4. “Online Payment for Access to Heterogeneous Mobile Networks”. It describes an architecture where access to heterogeneous mobile networks is granted on the basis of online payment methods. Access methods for GSM networks were designed for subscribed post-pay customers. With Intelligent Networks (IN) technology, operators can also offer services to prepay customers. We suggest an IP based architecture which uses a Diameter application for cost control and a charging component for online payment. [4]

5. “Shortest Route at Dynamic Location with Node Combination-Dijkstra Algorithm”. It is structured to modify the node combination algorithm to solve the problem of finding the shortest route at the dynamic location obtained from the fleet by displaying the node that have the shortest distance. [5]

CHAPTER 3

REQUIREMENT AND ANALYSIS

3.1 Problem Statement:

To build and implement Smart Management of EV Charging Stations using AI chatbot and GMAPS API.

Description –

We propose a Smart EV Charging station System where we used slot booking system to book EV vehicle charging station to charge vehicles. We further categories this slots according to charging socket type. We have also used AI voice assistant to communicate user with system via vocal commands. We also used GMAPS API to show shortest route to reach at destination. The system uses NLP for AI voice assistance as well as MySQL databases for storing a system logs as well as slot management.

3.2 Software Requirement Specifications (SRS) –

Introduction –

Purpose - To develop a web-based Hybrid Android Application for Smart management of EV charging station system to book charging slots to charge electric vehicles.

Scope - To available an easy and fastest way of charging station booking for customers to save their time.

Functional Requirement -

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail. The following are the key fields, which should be part of the functional requirements:

User: Execute the task

Usability: This relates to how easily people can use your app. A measure of usability could be the time it takes for end users to become familiar with your app's functions, without training or help.

Reliability: This is the percentage of time that your app works correctly to deliver the desired results, despite potential failures in its environment.

Performance: This is essentially how fast your app works. A performance requirement for the app could be start in less than 20 seconds.

Responsiveness: This requirement ensures that your app is ready to respond to a user's input or an external event no matter what it's doing currently

External Interface Requirement –

User Interfaces - The user interface is Android app in which he needs to register himself.

Hardware Interfaces- Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable, GSM, etc.

Software Interface Description- This system is a multi-user, multi-tasking environment. It enables the multiple users to interact with the same server. It uses Java and android as the back-end programming tool and MySQL as the back-end application tool.

Communication Interfaces - The system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

Non-Functional Requirements:

Performance Requirements –

- System can produce results faster on 4GB of RAM.
- It may take more time for peak loads at main node.
- The system will be available 100% of the time. Once there is a fatal error, the system will provide understandable feedback to the user.

Safety Requirements –

- The system is designed in modules where errors can be detected and fixed easily.

Software Quality Attributes –

Usability: This relates to how easily people can use your system. A measure of usability could be the time it takes for end users to become familiar with your system's functions, without training or help.

Reliability: This is the percentage of time that your app works correctly to deliver the desired results, despite potential failures in its environment.

Performance: This is essentially how fast your system works. A performance requirement for the system could be start in less than 20 seconds.

Security: Say that your system saves all the previous QR code and lets you reuse a saved QR code.

Responsiveness: This requirement ensures that your system is ready to respond to a user's input or an external event no matter what it's doing currently

System requirements –

Database Requirement

MySQL Database - MySQL is an open-source database which is mainly a RDBMS i.e., relational database management system. As a database server, primary function of this software is to store and retrieve data as requested by other from end software applications like Java which may or may not run either on the same computer or on different computer. This can be across the network either in internet or intranet.

Hardware Requirements:

- | | |
|--------------|-----------------------------|
| 1. Processor | - Intel i3/i5/i7 |
| 2. Speed | - 1.1 GHz |
| 3. RAM | - 2 GB (min) |
| 4. Hard Disk | - 40 GB |
| 5. Key Board | - Standard Windows Keyboard |
| 6. Mouse | - Two or Three Button Mouse |
| 7. Monitor | - SVGA |

Software Requirements:

- | | |
|-----------------------|---|
| 1. Operating System | - Windows 7/8/10 / Ubuntu18.04 LTS /20.04 LTS |
| 2. Application Server | - Apache Tomcat 7/8/9 |
| 3. Front End | - HTML, JDK 1.8, JSP |
| 4. Scripts | - JavaScript. |
| 5. Server-side Script | - Java Server Pages. |
| 6. Database | - My SQL 8.0 |
| 7. IDE | - Eclipse, Android Studio |

Setup –

Some of the important steps that I have to follow in order to set-up an environment for proposed system:

Task 1- Download and install JDK1.8 or higher

Task 2 - Set up database connectivity. Using JDBC and MySQL bridge drivers.

Task 3 - First go to project properties using right click on project in Eclipse IDE, select Run Tab, and set main page.

Software Engineering Methodology-

Software Life-cycle used in this Project – The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. In waterfall model phases do not overlap.

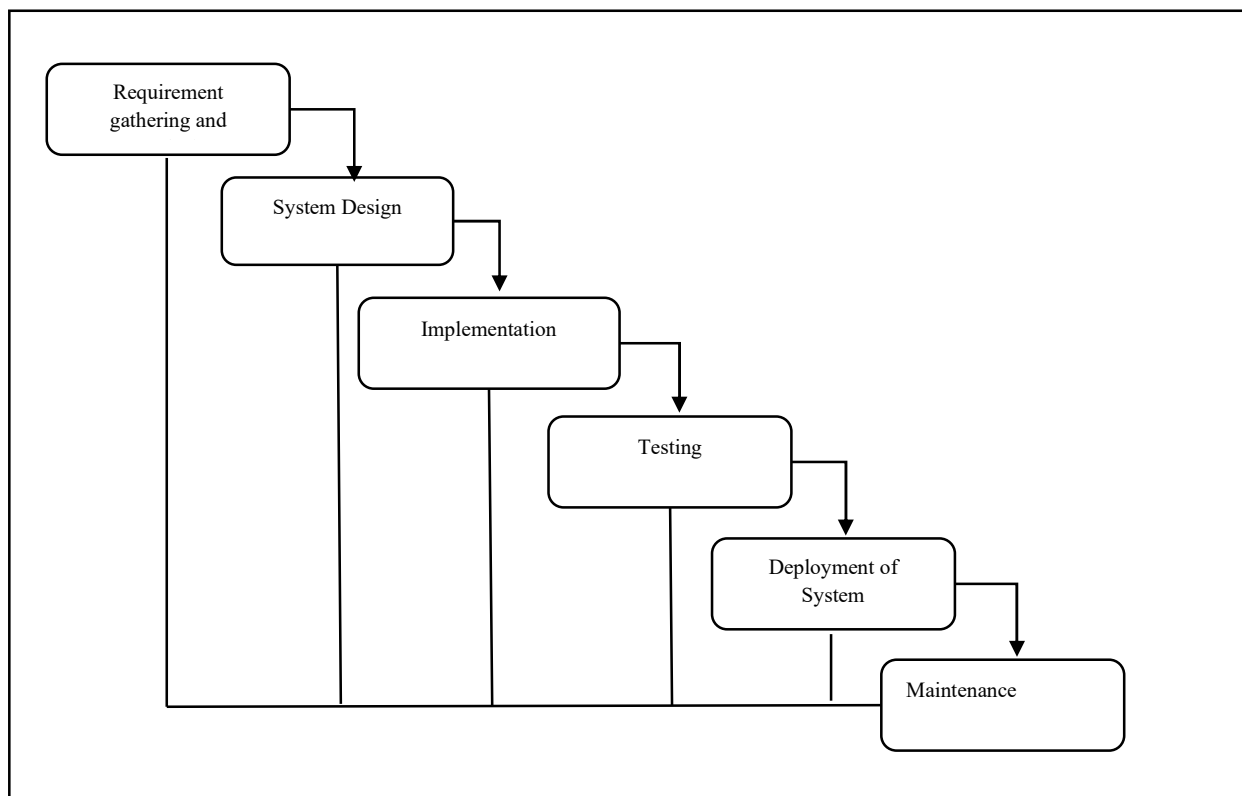


Fig 3.1 Waterfall Model

Applications of Waterfall model –

1. This model is simple and easy to understand and use.
2. It is easy to manage due to the rigidity of the model - each phase has specific deliverables and a review process.

System Implementation Plan –

Activity	I	II	III	IV	V	VI	VII	VIII	IX
	Week	Week	week	week	Week	week	week	week	week
	Aug 4	Aug 11	Aug 18	Aug 25	Sept 1	Sept 8	Sept 15	Sept 22	Sept 29
Initiate the project									
Communication									
Literature survey									
Define scope									
Develop SRS									
Plan the project									
Design mathematical model									
Feasibility Analysis									
Develop work breakdown structure									
Planning project schedule									
Design UML and other diagrams									
Design test plan									
Design risk management plan									

Activity	XI week	XII week	XII I Week	XI V week	XV week	XV I week	XV II week	XVI II week	XIX week	XX week	XXI week	XXII week
	Jan 5	Jan 15	Jan 19	Jan 26	Feb 2	Feb 9	Feb 16	Feb 23	Mar 2	Mar 9	Mar 16	April 25
Execute the project												
Build and test basic functional unit												
Build and test database with login and session maintenance facility												
Designing of 1 st and 2 nd module												
Testing of 1 st and 2 nd module												
Designing of 3 rd and 4 th module												
Testing of 3 rd and 4 th module												
Integration of all module												
Final Report and Presentation												

Table 3.1 System Implementation Plan

CHAPTER 4

DESIGN

4.1 System Architecture:

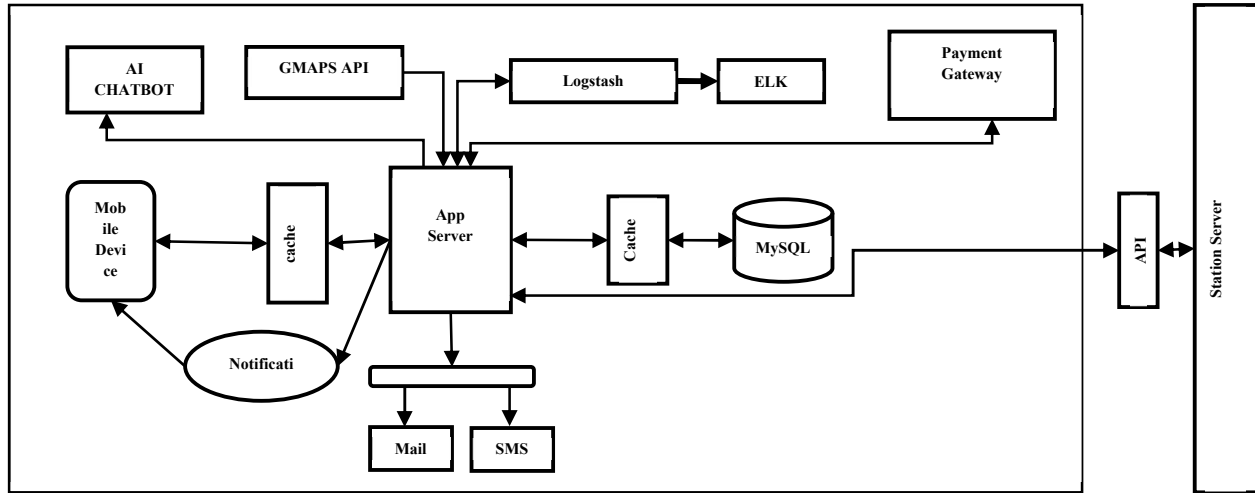


Fig 4.1 Booking System Architecture

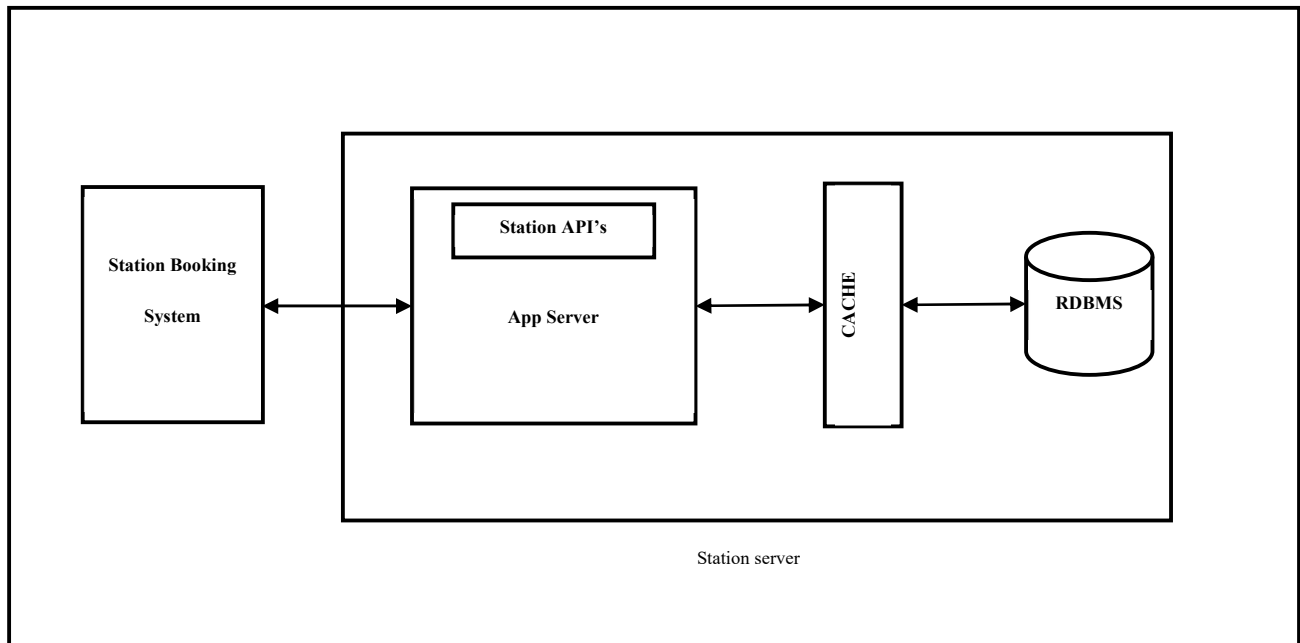


Fig. 4.2 Station System Architecturwe

System Architecture is divided into two parts –

1. Booking System Architecture –

In booking System architecture, it will completely work at user side. In this architecture initially user connects to the app server and login into the system.

After that the find outs the required slot from the database system and books the slot. Once the slot gets booked user will get Mail Notification with unique booking ID. Th the system caching is also implemented to achieve high performance.

There is also chatbot API is implemented in System which will help to solve user queries.

2. Station System Architecture –

At station side admin has the access to at charging slots to the database. Admin can also see the booking details as well as payment Details. From admin side to user side data transfer, APIs are implemented.

4.2 Use Case Diagram:

A use case diagram is a graphical representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can show the different types of users of a system and the various ways in which they interact with the system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionality use cases are prepared and actors are identified. The purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interaction among the actors.

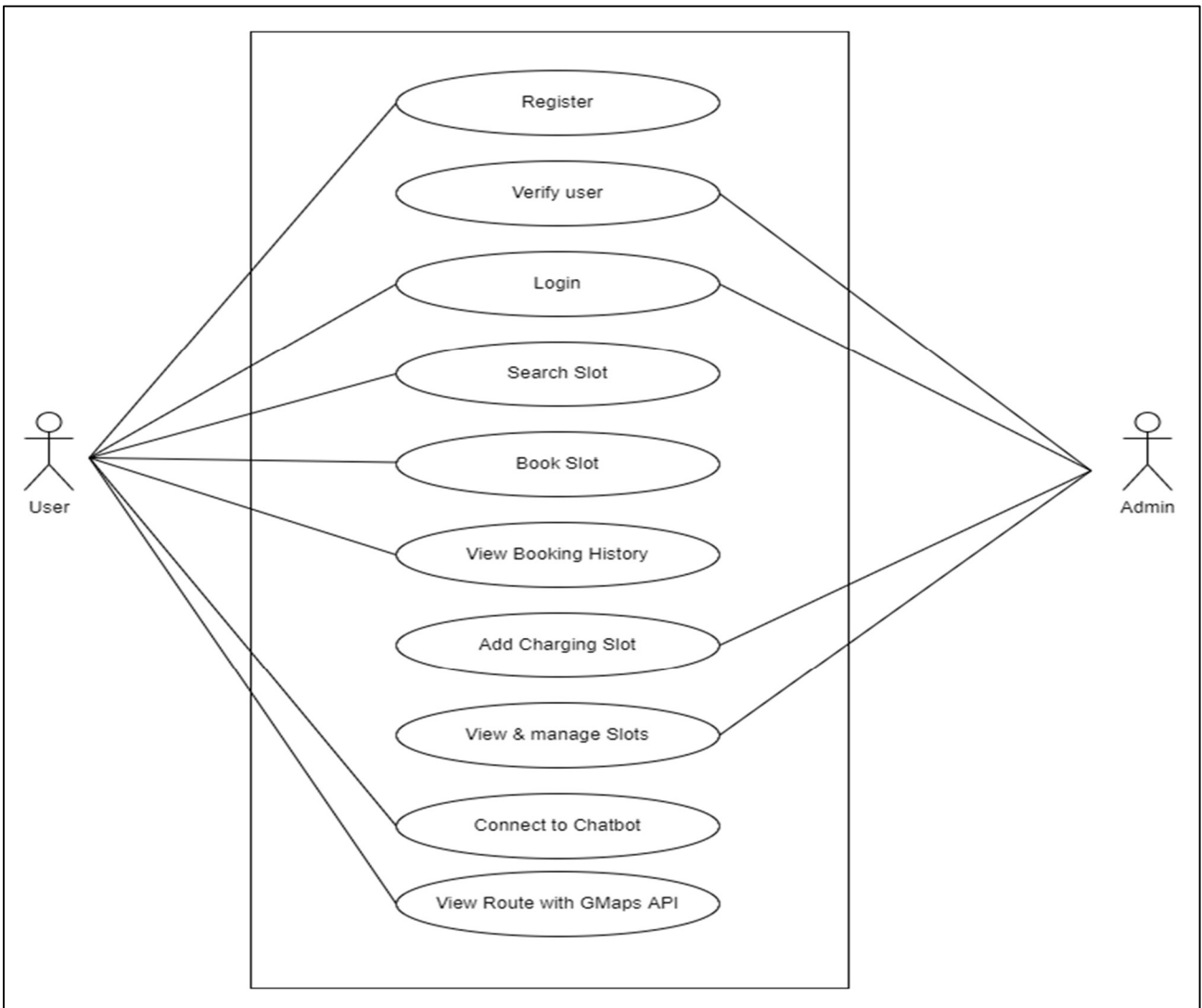


Fig 4.3 Use case Diagram

4.3 Class Diagram

The 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. The 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. The 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 is to model the static view of an application

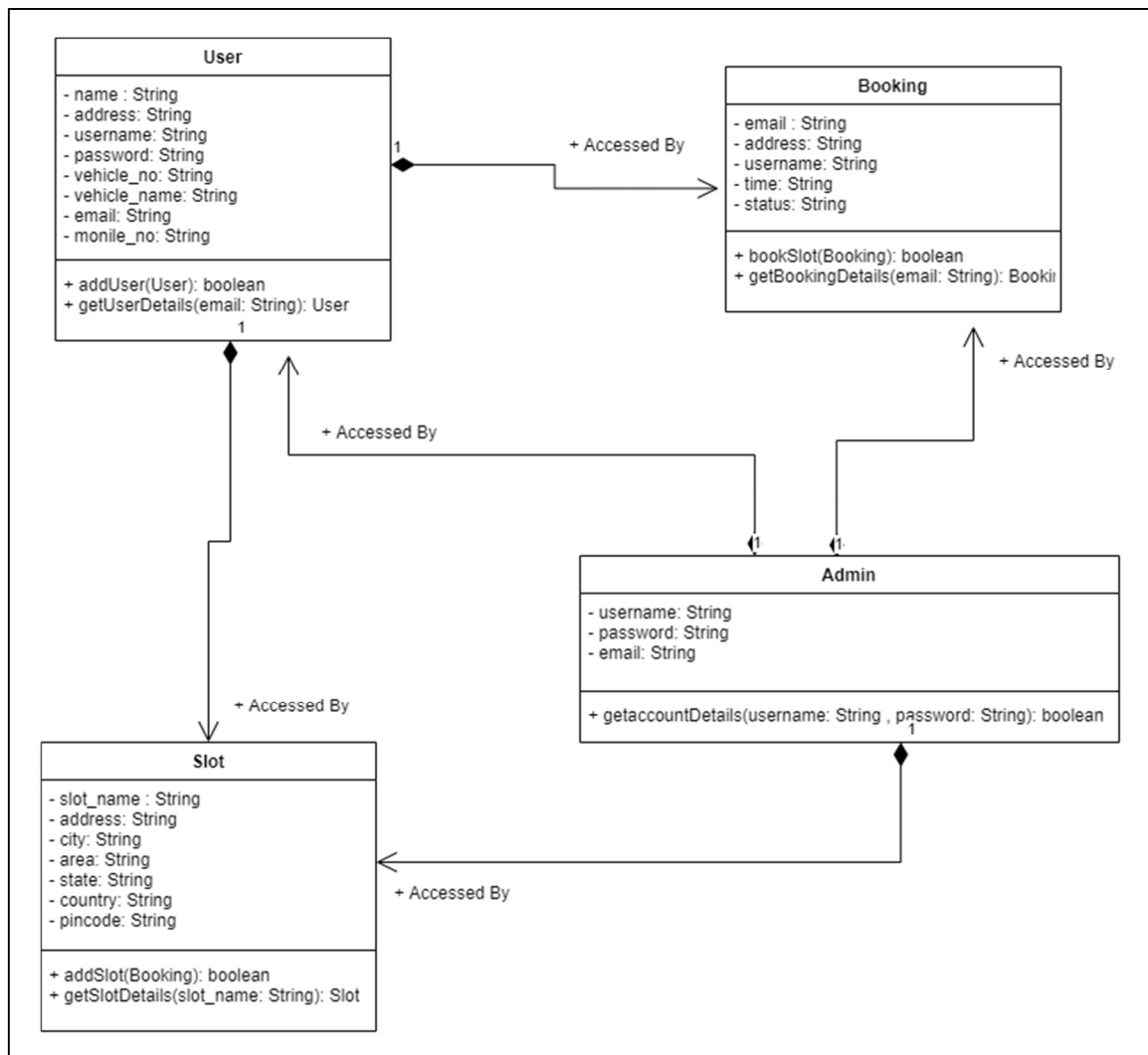


Fig 4.4 Class Diagram

4.4 Component Diagram

A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

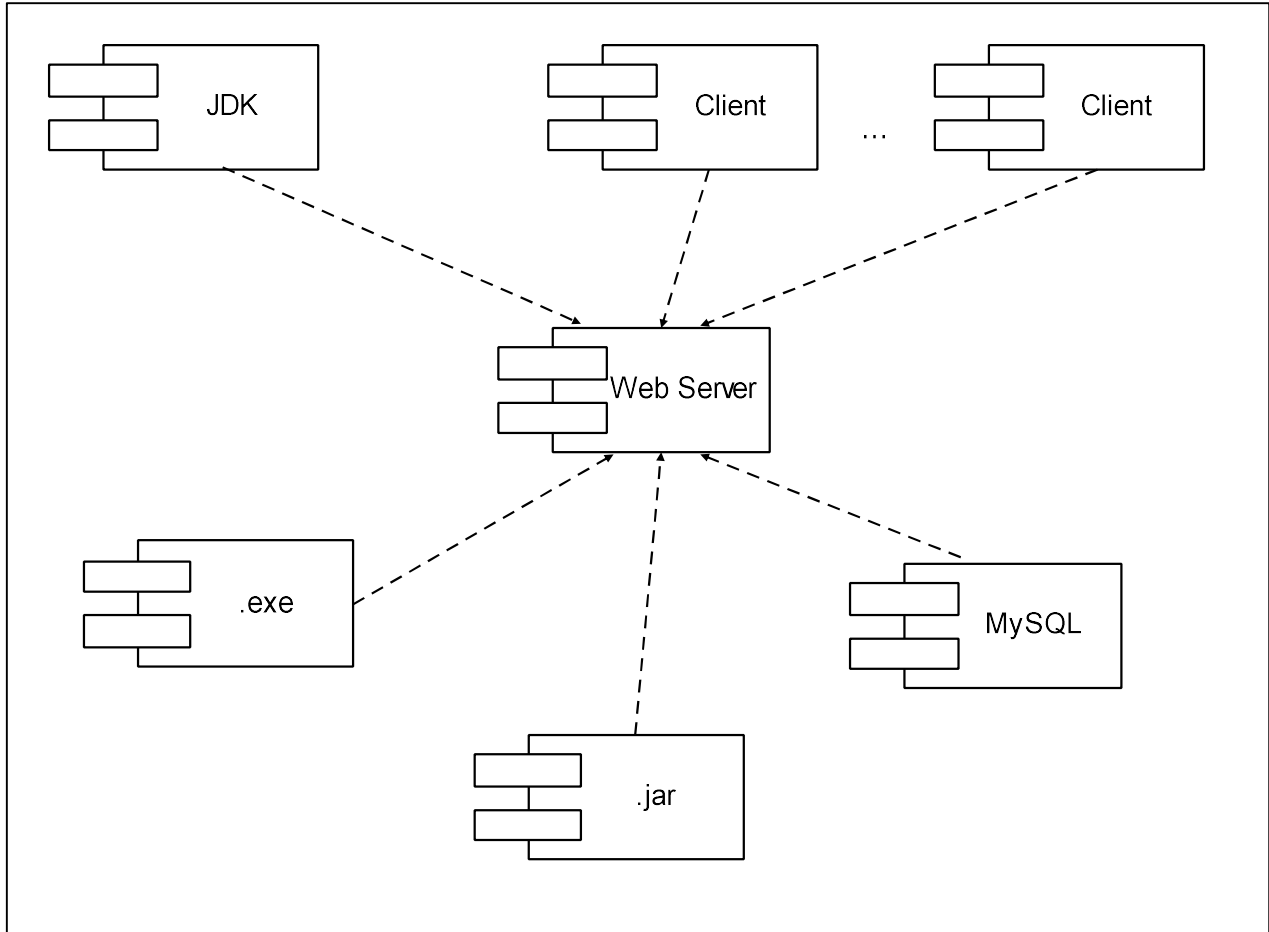


Fig 4.5 Components Diagram

4.5 Sequence Diagram

In this project, sequence diagram shows that User and Database are object fields. when user login to the system he gets the slot details which is already stored on database. When he received slot history or details then he book the slot properly and other information for slot booking filled. Then according to near station or availability user booked the slots and his booking for charge his EV Charging will be done. Successfully booked the slots. Admin has rights to add new slots or view booking history or change slots .It represents that when Admin login he has rights to add new slots, he will checking that no overlapping slots or not. Availability of slots or view slots. Then his work is to solve customers' problems or any queries or enquiries. Then finally Admin also logout from the system.

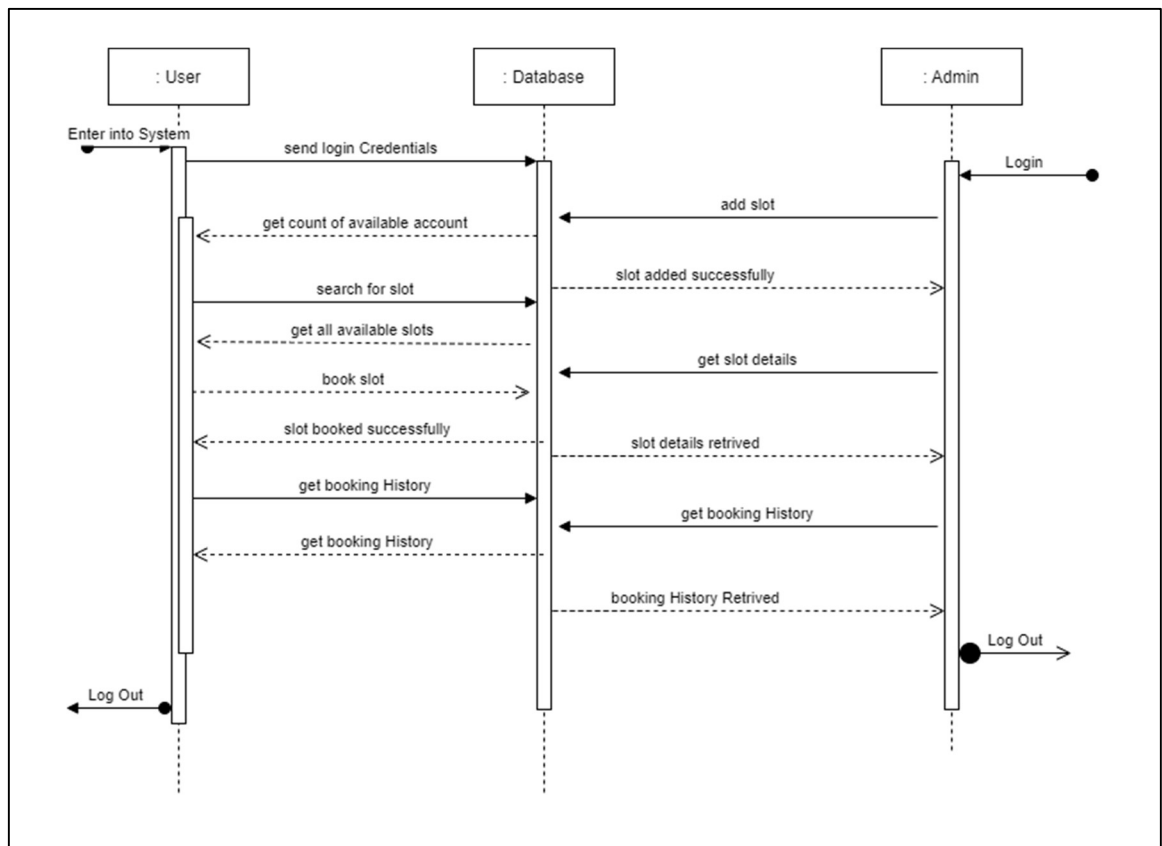


Fig. 4.6 Sequence Diagram

4.6 Data Flow Diagram

A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

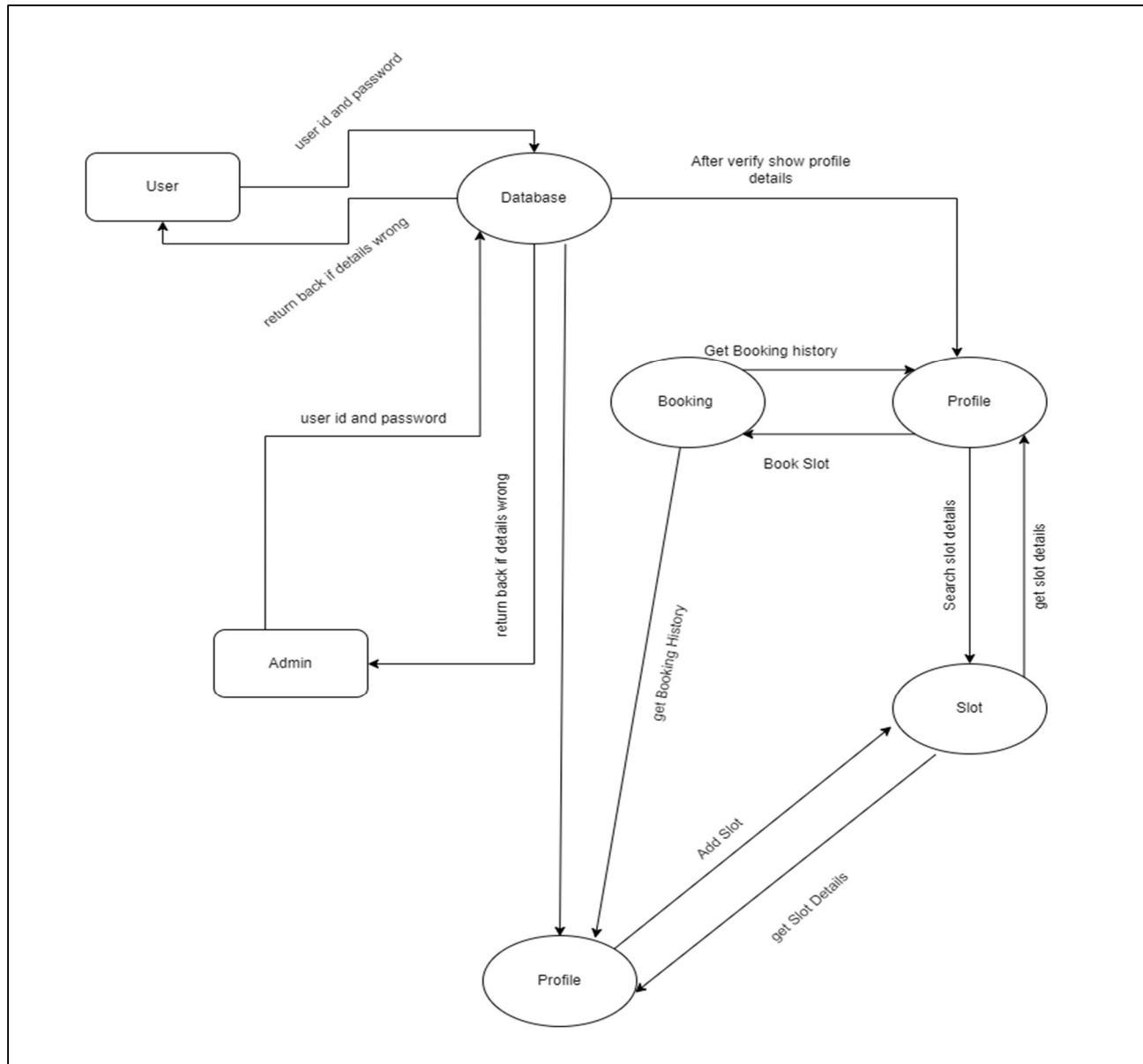


Fig 4.7 Data flow Diagram

4.7 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

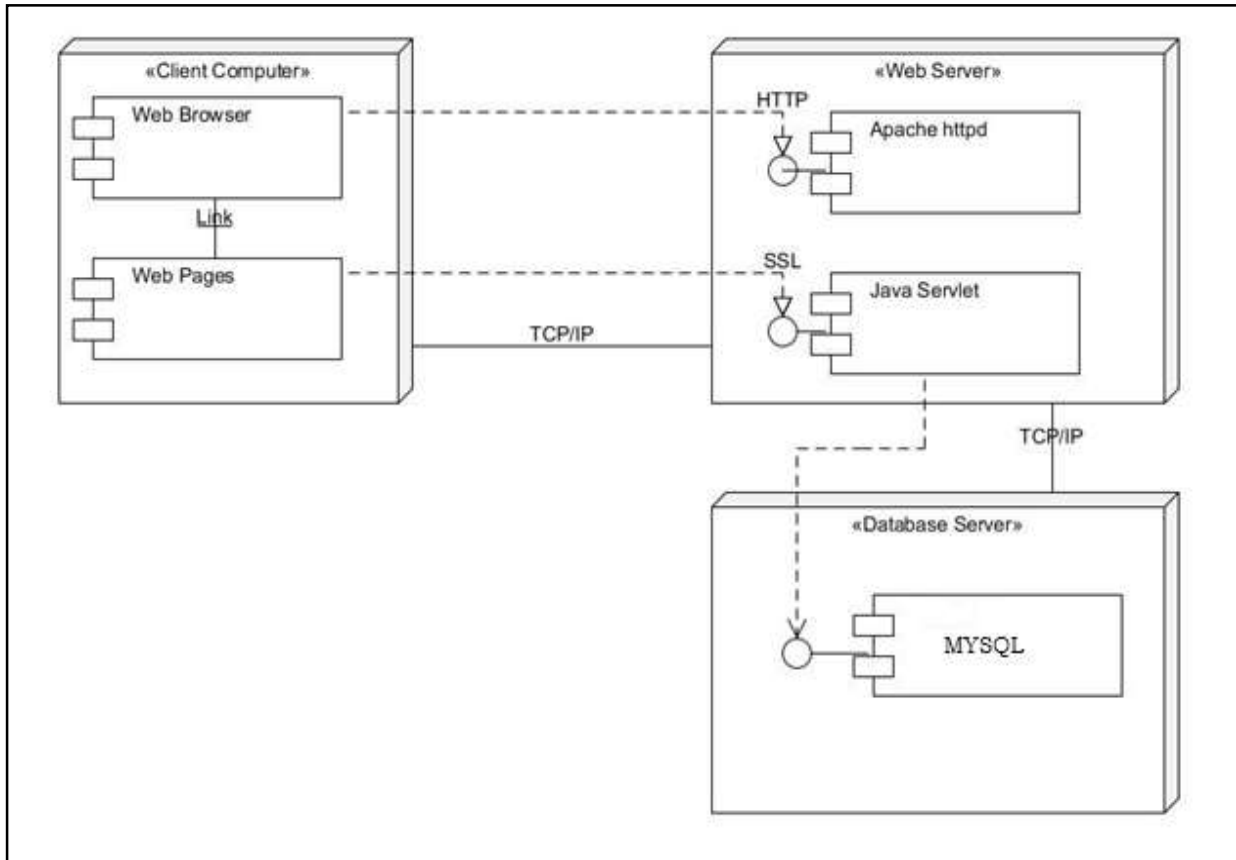


Fig 4.8 Deployment Diagram

CHAPTER 5

RESULTS AND DISSCUSSION

5.1. Testing -

Software testing is the act of examining the artifacts and the behavior of the software under test by validation and verification. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

Types of Testing -

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach -

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing -

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results –

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Test cases:

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Test Case ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
101	Enter the case insensitive Username click on Submit button.	Username	Error comes	Error Should come	Pass
102	Enter the case sensitive Username click on Submit button.	Username	Accept	Accept Username	Pass
201	Enter the case insensitive Password click on Submit button.	Password	Error comes	Error Should come	Pass
202	Enter the case sensitive Password click on Submit button	Password	Accept	Accept	Pass
301	Enter the case insensitive Mobile Number click on Submit button	Mobile Number	Error comes	Error Should come	Pass

Pass 302	Enter the case sensitive Mobile Number click on Submit button.	Mobile Number	Accept	Accept	Pass
303	Enter slot name, slot address, slot area, city, state, country	Add	Add Slot	Accept	Pass
304	Search Charging Slot, if it is added then	Search	Search Slot	Search the slot	Pass
305	View Charging Slot	View	View Slot	Show	Pass
306	Booking slot	Book	Show Booked Slot	Show	Pass
307	Get location of slot	Source and Destination	Show map view	show	Pass
308	Status of user	OPEN/CLOSED	Get Booking History	Booking History	Pass
309	Search Charging Slot, if it is not added then	Slot Not Found	Error Comes	Error Should come	Pass

Table 5.1 Test Case results

- **Application Results-**

Home Page -

This is the homepage it gives the idea of the overall system the user can navigate through it provide menus where user can go deeper into various areas of the site.

Like “About”, “Contact us”, “Login page”. In addition, the home page often serves to users by providing titles, headlines and images and visuals that show what the website is about.

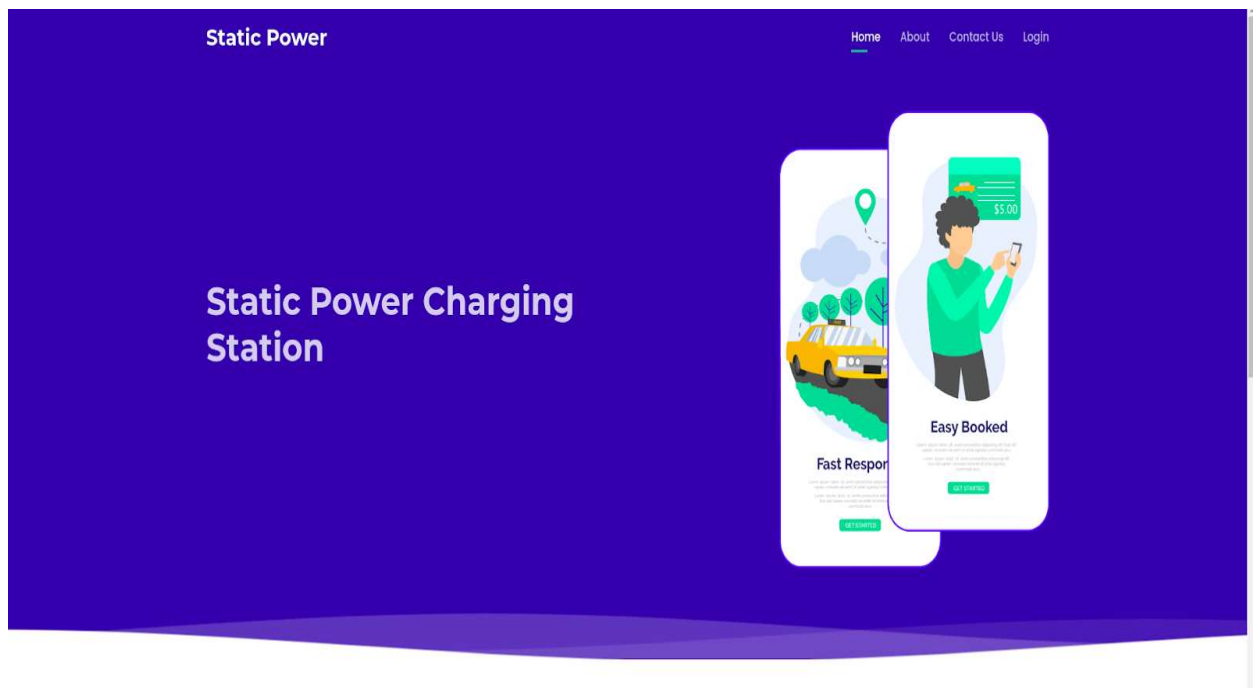
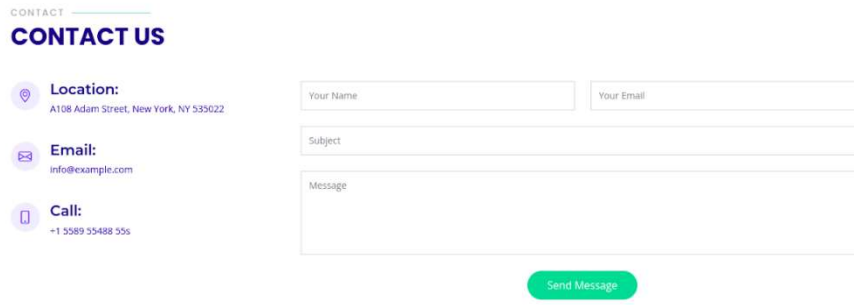


Fig. Home Page

Contact us -

The contact page describes how your system can help solve the user's problems. Include an email and phone number so users can get in touch with you on their first attempt. Include a short form using fields that'll help to understand who's contacting them.



CONTACT

CONTACT US

Location:
A108 Adam Street, New York, NY 535022

Email:
info@example.com

Call:
+1 5589 55488 555

Your Name

Your Email

Subject

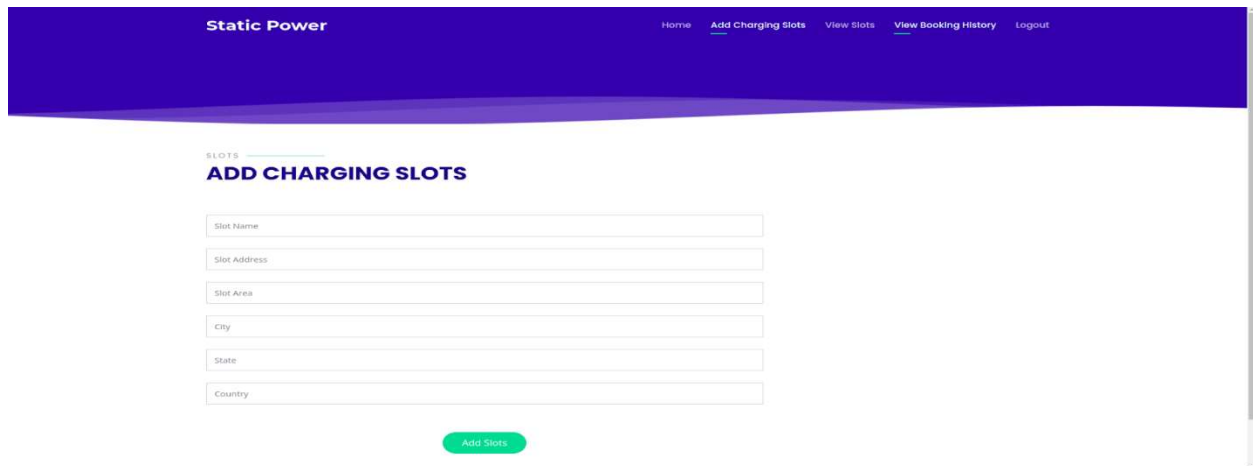
Message

[Send Message](#)

Fig. Contact us

Add Charging Slot

Admin can add charging slots by filling the information like the name of the slot, address of slot, area where the slot is located, city, state, country in the fields.



Static Power

Home [Add Charging Slots](#) [View Slots](#) [View Booking History](#) [Logout](#)

SLOTS

ADD CHARGING SLOTS

Slot Name

Slot Address

Slot Area

City

State

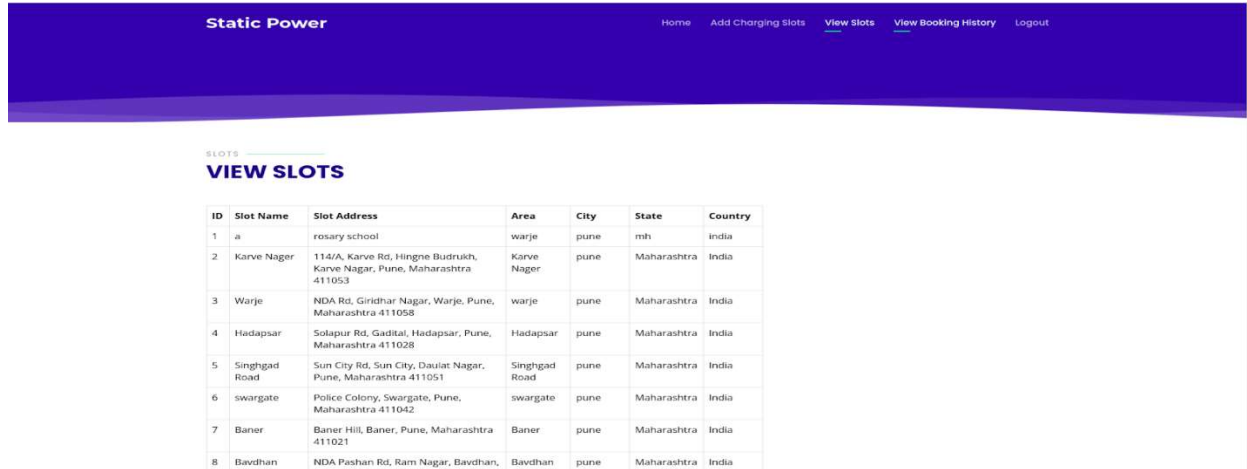
Country

[Add Slots](#)

Fig. Add charging slot

View Charging Slots –

After adding the charging slot in the add charging slot section, the user can view the slots that are added in the view slots page.

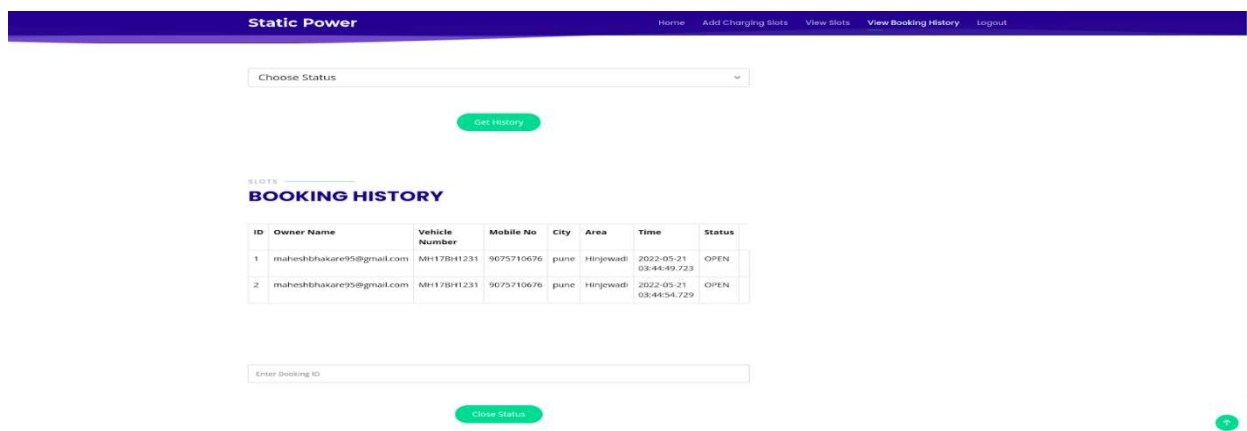


ID	Slot Name	Slot Address	Area	City	State	Country
1	a	rosary school	warje	pune	mh	india
2	Karve Nager	114/A, Karve Rd, Hingne Budrukhi, Karve Nagar, Pune, Maharashtra 411053	Karve Nager	pune	Maharashtra	India
3	Warje	NDA Rd, Giridhar Nagar, Warje, Pune, Maharashtra 411058	warje	pune	Maharashtra	India
4	Hadapsar	Solapur Rd, Gadital, Hadapsar, Pune, Maharashtra 411028	Hadapsar	pune	Maharashtra	India
5	Singhad Road	Sun City Rd, Sun City, Daulat Nagar, Pune, Maharashtra 411051	Singhad Road	pune	Maharashtra	India
6	swargate	Police Colony, Swargate, Pune, Maharashtra 411042	swargate	pune	Maharashtra	India
7	Baner	Baner Hill, Baner, Pune, Maharashtra 411021	Baner	pune	Maharashtra	India
8	Bavdhan	NDA Pashan Rd, Ram Nagar, Bavdhan, Pune, Maharashtra 411021	Bavdhan	pune	Maharashtra	India

Fig. View charging slot

Booking history –

The booking history page describes the booking status of the user through user name vehicle number, mobile number, area and the time at which the slot is booked by the user, it also gives the status whether the slot are free or engaged to use.

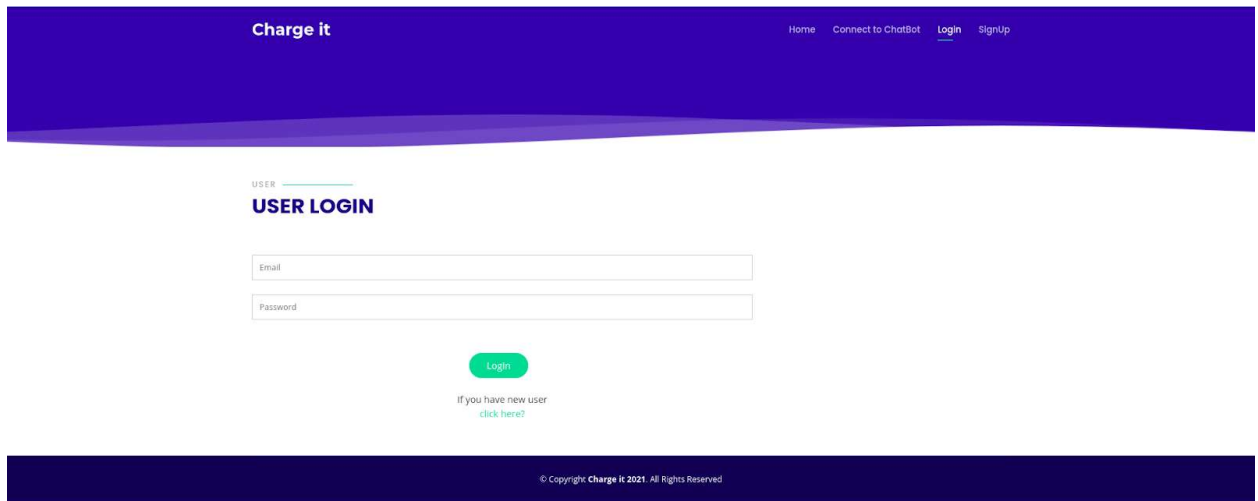


ID	Owner Name	Vehicle Number	Mobile No	City	Area	Time	Status
1	maheshbhakare95@gmail.com	MH17BH1231	9075710676	pune	Hinjewadi	2022-05-21 03:44:59.723	OPEN
2	maheshbhakare95@gmail.com	MH17BH1231	9075710676	pune	Hinjewadi	2022-05-21 03:44:54.729	OPEN

Fig. Booking History

User Login -

A login page is an entry page to a website that requires user identification and authentication, and it is performed by entering email id and password combination.



The screenshot shows a web application titled "Charge it" with a dark blue header. Navigation links include "Home", "Connect to ChatBot", "Login" (highlighted), and "SignUp". Below the header, the page is titled "USER LOGIN". It features two input fields: "Email" and "Password". A green "Login" button is positioned below the fields. A link "If you have new user click here?" is located below the button. The footer contains the copyright notice: "© Copyright Charge it 2021. All Rights Reserved".

Fig. User Login

Chat Bot -

Chabot is used to conduct a conversation with human using textual or auditory methods.

Chabot are used for various purposes including customer service, request routing, or information gathering.

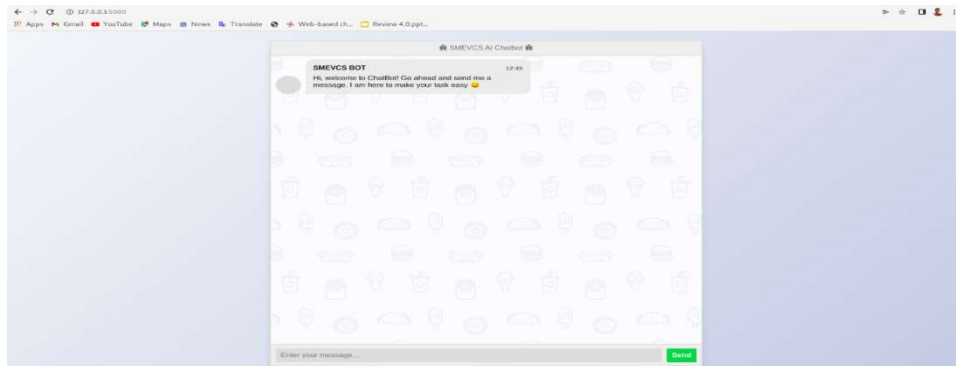


Fig. chatbot

Slot Information -

This page gives the information about the slots. Like the name of the slot, address of the slot, area where the slot are located, city, state, country. It also has a option of view map it gives the location of station through the map.

Charge it Home Connect to ChatBot Search Slot Booking History Logout

SLOT **SLOT INFORMATION**

Slot Name	Hinjewadi
Address	Hinjewadi Phase 1 Rd, Phase 1, Hinjewadi Rajiv Gandhi Infotech Park, Hinjewadi, Pimpri-Chinchwad, Maharashtra 411057
Area	Hinjewadi
City	pune
State	Maharashtra
Country	India
View Map	View Map

[Book Slot](#)

© Copyright Charge it 2021. All Rights Reserved

Fig. Slot Information

CHAPTER 6

CONCLUSION

6.1 Conclusion

We have designed and implemented a “Smart Management of EV Charging Stations” with hybrid approach of android application development. charging management system utilizing charging strategies that include effective reservation management and efficient allocation of time slots of charging stations. Such a system is developed in consideration of the urban environment. This system is also contains the AI Chabot for query solving as well as GMAPS API for direction sensing. The system having good user-friendly UI, which will help for user to navigate throughout the system easily.

REFERENCES

1. Binod Vaidya¹, Hussein T. Mouftah: Smart Electric Vehicle Charging Management for Smart cities: IET Research Journals, The Institution of Engineering and Technology 2015
2. Pooja Singh, Pinki Nayak, Arpita Datta, Depanshu Sani, Garima Raghav, Rahul Tejpal: Voice Control Device using Raspberry Pi: 2019 Amity International Conference on Artificial Intelligence (AICAI)
3. Subhash S, Prajwal N Srivatsa, Siddesh S: Artificial Intelligence Based Voice Assistant: 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4).
4. Heiko Knospe, Scarlet Schwiderski-Grosche: Online Payment for Access to Heterogeneous Mobile Networks: IST Programme under Contract IST-2000-25350.
5. Achmad fitro: Shortest Route at Dynamic Location with Node Combination-Dijkstra Algorithm: 978-1-5386-8402-3/18/\$31.00 ©2018 IEEE