09_Plagiarism_03 June 2025

by Jawahar CP

Submission date: 03-Jun-2025 12:50AM (UTC-0700)
Submission ID: 2691193309
File name: 09_blackbook.pdf (1.71M)
Word count: 5424
Character count: 32477

Abstract

With the rising global interest in sustainable and eco-friendly transportation, electric vehicles (EVs) are becoming increasingly popular. However, the availability and accessibility of charging infrastructure have not kept pace with the rapid adoption of EVs. One of the major issues faced by EV users is the lack of a centralized system to locate nearby charging stations and book charging slots in real time. This often leads to uncertainty, unnecessary delays, and inefficient use of resources at charging stations.

To address these challenges, we have developed a comprehensive Electric Vehicle Charging Slot Booking Application. This platform enables users to view a list of nearby charging stations, check real-time slot availability, and make reservations in advance. In addition to booking functionalities, the system also offers features such as payment integration, booking history, user authentication, and administrative access for charging station operators.

The application has been designed with a focus on simplicity, responsiveness, and operational efficiency. It provides a seamless user experience across multiple devices and supports both web and mobile platforms. By ensuring better time management, improved resource allocation, and increased transparency, our application contributes to the overall advancement of smart mobility systems. This project lays the foundation for a scalable and adaptive system capable of growing alongside the evolving needs of the electric vehicle ecosystem.

Indexed Terms- Electric Vehicles (EV), Charging Slot Booking, Smart Mobility, Real- Time Sc--heduling, Sustainable Transportation, Location-Based Services.

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

With the increasing popularity of electric vehicles, there is a parallel demand for an organized and reliable charging infrastructure. Despite the expansion of EV usage, users often face delays and uncertainty due to the lack of real-time information on charger availability. This project proposes a digital platform where EV owners can search for stations, view live slot status, and make reservations in advance. Our aim is to bridge the gap between user convenience and operational efficiency, contributing to the development of smart and sustainable transportation systems.

1.2 MOTIVATION

The rapid advancement in electric vehicle (EV) technology has been a major step toward reducing carbon emissions and achieving global sustainability goals. However, the growth of EV adoption has not been matched with an equally efficient charging infrastructure, especially in terms of accessibility and time management. One of the most common challenges faced by EV users is the unpredictability of slot availability at charging stations, which often leads to long waiting times, energy waste, and user dissatisfaction.

This gap in the EV ecosystem inspired us to develop a solution that bridges the disconnect between users and charging infrastructure through technology. We observed that while several EV charging stations are being installed, the lack of a digital system for slot management and pre-booking results in inefficient utilization of resources. By leveraging real-time data and user-friendly design, we were motivated to build a system that simplifies the process of locating charging stations and booking available lots in advance.

1.3 PROBLEM STATEMENT

Electric vehicle owners frequently experience inconvenience when searching for available charging stations. The absence of a centralized system for booking and managing charging slots leads to long queues and time wastage. Charging station operators also struggle with unorganized traffic and underutilization of resources. To resolve these issues, a structured platform is required that allows real-time booking, availability tracking, and administrative control. Our project addresses this problem by offering an automated, easy-to-use solution for both users and station managers

.

1.4 OBJECTIVE

- To develop a user-friendly web/mobile application that allows EV owners to locate near by charging stations.
- To implement a real-time slot booking system that enables users to check availability and reserve charging slots in advance.
- To reduce waiting times and improve the efficiency of EV charging station usage.
- To provide charging station administrators with an interface to manage slot bookings, monitor availability, and update station status.
- To enhance the user experience by integrating features like booking history, status notifications, and secure login functionality.

CHAPTER 2: LITERATURE SURVEY

2.1 INTRODUCTION

The surge in electric vehicle (EV) adoption has underscored the necessity for efficient charging infrastructure. A critical challenge faced by EV users is the lack of real-time information on charging station availability, leading to prolonged waiting times and inefficient resource utilization. To address these issues, various digital solutions have been proposed and implemented.

2.2 LITERATURE SURVEY

S.L. Sreedevi and B.T. Geetha (2023) introduced a priority-based EV slot booking system to address the inefficiencies in traditional first-come-first-serve charging setups. Their work emphasizes serving emergency vehicles by assigning charging priorities, thereby minimizing waiting time during high-demand periods. While our system draws inspiration from their queue management, it diverges by focusing on public usage with real-time updates and a user-first design, offering flexibility and equal accessibility to all users.

Ghegade Mayuri et al. (2023) presented a basic web-based slot booking application that allowed users to reserve charging slots ahead of arrival. Their work laid the groundwork for centralized EV slot management. Our system enhances this concept by adding mobile responsiveness, real-time station updates, and a secure payment gateway — offering a more complete and scalable user experience for widespread public adoption.

Rukmani Devi D. et al. (2024) emphasized emergency vehicle prioritization in EV slot booking, considering use cases like ambulances and public service vehicles. Our project, although acknowledging such use cases, is designed for everyday consumer utility. It ensures smooth, efficient booking for individual EV owners while maintaining station availability and administrative control.

Gowtham R. et al. (2023) developed an IoT-integrated EV charging architecture where smart devices updated the system with real-time availability. Their model focused on sensor-based hardware implementation. Our work aligns with this approach on the data sync level but is built as a software-driven solution, utilizing available APIs to display live availability without additional hardware constraints.

CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

This Software Requirements Specification (SRS) document outlines the functional and non- functional requirements for the Electric Vehicle Charging Slot Booking Application. This system will provide electric vehicle (EV) users with a platform to search, book, and manage charging slots in real time. The application will aim to streamline the booking process, reduce waiting times, and optimize the utilization of charging stations. This document serves the reference for developers and various component are volved in the project.

3.1.1 Project Scope

- Real-time availability checking of charging slots.
- Online booking system for EV charging stations.
- User authentication and authorization for both end-users and administrators.
- Payment gateway integration for charging session payments.

3.1.2 User Classes and Characteristics

User classes

Registration and Login For Users:

Our application allows registration for the new user by providing essential information such as their name, email address, phone number, and vehicle details. Users can set a password for secure access. After registration, they can log in using their credentials. Authentication ensures that only registered users can access the main features like searching for charging stations, booking slots, and managing their bookings. The system may also include options like email or SMS verification during registration for added security.

Search for Charging Stations:

Once user is logged in, users can easily search our provided charging stations based on their location or by entering the name of a city. The system uses location services to show nearby stations on a map or in a list format. It displays vital details such as station distance, available charger types (fast, slow, normal), operating hours

and user ratings. This helps EV owners quickly identify suitable stations according to their immediate needs

Book a Charging Slot:

The core functionality of the application is to enable users to book a charging slot at their chosen station. After selecting a station, users pick a specific date, time, and charger type based on availability. They confirm the booking through a simple checkout process, with the option to prepay online if the station requires it. A booking confirmation with details like booking ID, slot time, and station address is sent to the user via email or SMS.

Characteristics:

- Tech-Savviness: The users may range from tech-savvy individuals to those with limited technical knowledge, so the interface must be intuitive and easy to navigate. Features like location tracking, slot reservations, and payment gateways should be simple and user-friendly.
- Data Input: Users will input various parameters, such as their vehicle model, battery percentage, and preferred charging time. They may also provide their location to find nearby charging stations.
- Decision-Making: EV owners rely on the application to help them find the nearest and most convenient charging stations

Assumptions and Dependencies Data Availability:

Assumption: Sufficient and accurate data on charging station locations, availability, and user preferences is available for the system to function effectively. Dependency: The system's accuracy in slot availability, booking status, and optimal station recommendations depends on the real- time accuracy and completeness of the data provided by charging stations and users.

Model Training:

Assumption: Any machine learning models used for predictions, such as predicting charging slot demand or estimating user preferences, are trained on a diverse and representative dataset, including factors like location, time of day, and user behavior. Dependency: The effectiveness of the prediction depend on the our polished quality and diversity of the training data and the model's ability to organized to new data, such as changes in user patterns or station usage.

Market Conditions:

Assumption: The platform assumes that the demand for EV charging slots follows a reasonably predictable pattern based on user habits, geographic location,and vehicle usage.

Dependency: Changes in market conditions, such as an increase in the number of EV users, the introduction of new charging stations, or changes in user preferences, can significantly affect the accuracy of demand predictions, as well as the platform's overall performance and resource allocation.

3.2 FUNCTIONAL REQUIREMENTS

Charging Station Data Collection Data Processing Slot Filtering and Availability Slot Booking and Management Notifications and Confirmation Generation.

20

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 User Interfaces

The system will provide both a web-based interface and a mobile application to facilitate seamless interaction for users. The web interface will enable users to search for nearby charging stations, view available time slots, and make bookings easily through a clean and user-friendly design. It will include clear instructions and real-time feedback during the booking process to enhance the overall user experience.

3.3.2 Hardware Interfaces

The system will rely on appropriate hardware to ensure smooth operation and accessibility. Server hardware is essential to host the application and support backend operations, including data processing, booking management, and real- time communication. These servers must be capable of handling high volumes of concurrent requests and supporting database operations efficiently.

3.3.3 Software Interfaces

The backend of the application will be developed using robust web frameworks such as Django or Flask, which will handle server-side logic, user authentication, and data routing. For the frontend, technologies like HTML5, CSS3, and JavaScript will be employed, potentially enhanced by libraries such as React.js to deliver an interactive and dynamic user experience.

3.3.4 Communication Interfaces

Communication interfaces will ensure encrypted data transfer and secure access to booking and user information. Additionally, the platform may integrate external APIs to retrieve auxiliary data, such as location services or charger specifications, and to connect with other services as needed.

3.4 NONFUNCTIONAL REQUIREMENTS

3.4.1 Performance Requirements

- The system should deliver fast and seamless interactions to ensure a smooth user experience.
- Minimal Response Time.
- High Scalability: The system should handle a large number of concurrent usersand Optimal Resource Usages.

3.4.2 Safety Requirements

The safety of user data and system operations must be a top priority. The system should ensure data integrity, protecting all user and booking data from unauthorized access, accidental modifications, or deletions. In the event of any operational failure, the system should respond gracefully, providing users with informative error messages while internally logging these issues for further analysis and troubleshooting.

3.4.3 Security Requirements

Security is critical for protecting both user data and operational functionality. The system must include safe and secure authentication mechanisms that verify the identity of users and restrict access to sensitive features and data to authorized users only. All critical data, especially user credentials and booking information, should be encrypted during storage.

3.4.4 Software Quality Attributes

Maintainability: The system should be designed and implemented in a way that allows for easy maintenance and updates in the future.

Reliability: Our system must be reliable to ensure dependability of the application.

Usability: The system should be user-friendly, with a clear and intuitive interface that allows users to easily input property features and view prediction results.

Portability: The system should be portable, allowing it to run on different platforms and environments .

3.5 SYSTEM REQUIREMENTS

3.5.1 Software Requirements

Operating system: Windows, macOS, and Linux.

Programming Language: Python and JavaScript.

Tools: Visual Studio Code or PyCharm.

Frameworks and Libraries: Python libraries such as Flask for the backend.

3.5.2 Hardware Requirements

Processor: The system should run on a standard processor, such as Intel Core i5 or higher, to handle computations efficiently.

Memory: A min of 4 or 8 GB RAM is mendatory for smooth performance

.

Storage: The system should have sufficient storage capacity to store property data, user information, and prediction results. A minimum of 500 GB hard drive space is recommended.

Network: The system should have access to a stable internet connection for real-time data processing and update

3.6 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

The project plan provided delineates our project's execution process, following the Software Development Life Cycle (SDLC) phases. It will undergo continuous updates to accurately mirror the project's real progress and upcoming plans. In the initial stages of requirement gathering and analysis, our primary emphasis was on understanding the problem statement thoroughly and guaranteeing the system's dependability. This phase also encompassed gathering information about the essential software and hardware components required for the project.

The Waterfall model is a linear and sequential SDLC model. This model is suitable for projects with clear and well- defined requirements from the beginning. For your EV charging slot booking system, this model can be used if all features and functionalities (such as user registration, station listing, and booking system) are planned in advance. It ensures systematic development but lacks flexibility for mid-project changes.

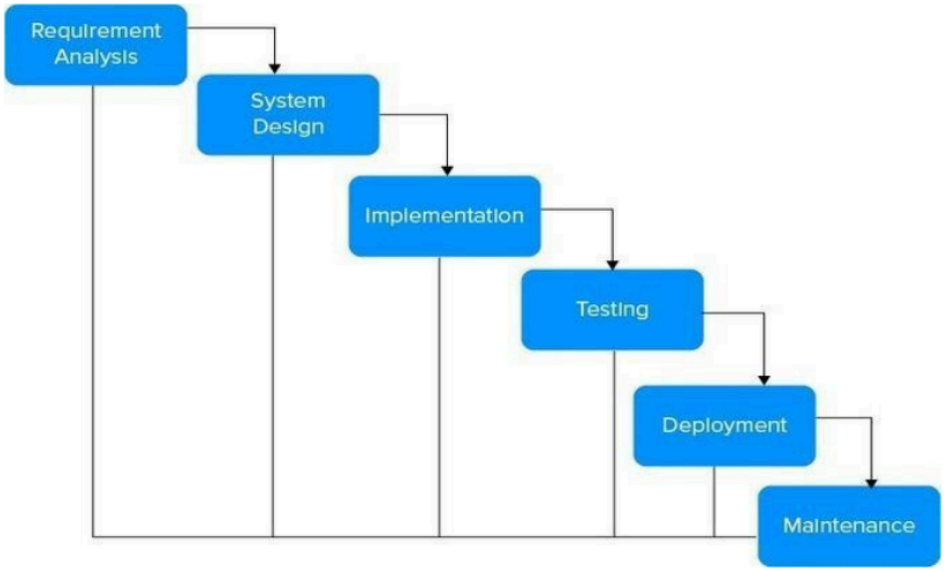


Figure 3.1: Waterfall Model

.

3.7 SYSTEM IMPLEMENTATION PLAN

Sr. NO	Task Name	Begin Date	End Date	Remarks
1	Selecting project domain	1 Aug 2024	10Aug 2024	Done
2	Understanding project need	11 Aug 2024	17 Aug 2024	Done
3	Understanding requisites	18 Aug 2024	19 Aug 2024	Done
4	Information Gathering	21 Aug 2024	31 Aug 2024	Done
5	Literature survey	1 Sept 2024	15 Sept 2024	Done
6	Refine project scope	16 Sept 2024	18 Sept 2024	Done
7	Concept development	20 Sept 2024	21 Sept 2024	Done
8	Planning and scheduling	22 Sept. 2024	23 Sept 2024	Done
9	Requirement Analysis, risk identification and monitoring	25 Sept 2024	26 Sept 2024	Done
10	Design review and refinement design and modeling	27 Sept 2024	28 Sept 2024	Done
11	Implementation	29 Sept 2024	16 Oct 2024	Done
12	Review and suggestion	17 Oct 2024	20 Oct 2024	Done
13	Outcome assessment	21 Oct 2024	20 Nov 2024	Done
14	Testing and QA assistance	21 Nov 2024	15 Feb 2025	Done
15	Review and suggestion QA	16 Feb 2025	15 Mar 2025	Done

CHAPTER 4:
SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture of our Electric Vehicle (EV) Charging Slot Booking Application is designed to ensure efficient interaction between users and charging infrastructure.The front-end (presentation layer) is used for the developing html, Css, Javascript, providing an intuitive interface for users to search, book, and manage charging slots. The application layer, built with Python and Flask, handles business logic, user authentication, and slot booking operations. MySQL is used as the backend database to store user data, station details, and booking information. RESTful APIs ensure smooth communication between the client and server. This architecture ensures scalability, security, and real-time data processing for a seamless user experience.

The system architecture of our Electric Vehicle (EV) Charging Slot Booking Application is structured to provide a reliable, scalable, and user-friendly platform for EV users to locate and reserve charging slots.

The **presentation layer** is responsible for user interaction and is developing the front end.

The **application layer** serves as the core of the system and is built using Python and the Flask framework. It manages business logic, user authentication, booking validation, and routing of user requests.

The **data layer** uses MySQL to handle all persistent data, including user profiles, station information, city data, slot availability, and booking history. It supports fast query processing and maintains data integrity.

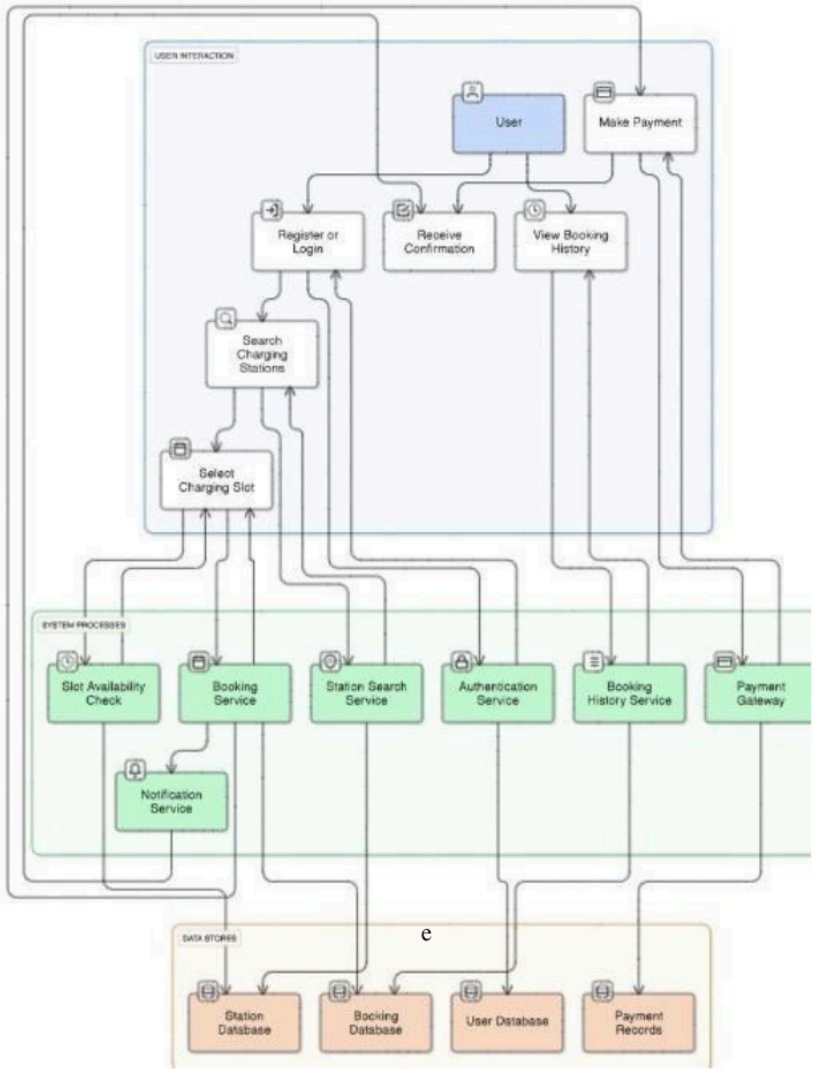


Figure 4.1: System Architecture

4.2 DATA FLOW (DFD) DIAGRAMS

A Data Flow Diagram (DFD) visually represents how information flows through a system and undergoes transformations as it moves from input to output. It illustrates the path of information and the processes or transformations that occur at each stage. This graphical technique is essential for understanding how data is processed within a system, from its initial input to the final output.

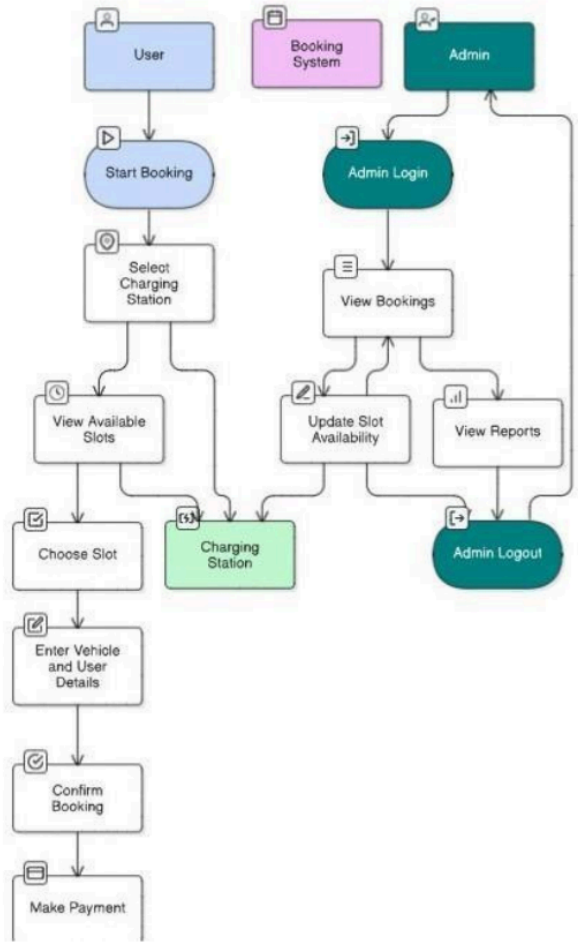


Figure 4.2: DFD Level 0 Diagram

DFD Level 1 Diagram

The data flow diagram (DFD) is a crucial modeling tool that helps represent various system components. These components typically include the system process itself, the data utilized by that process, external entities that interact with the system, and the pathways through which information flows within the system. DFDs provide a clear visual representation of how data moves through a system and how different elements interact with each other, aiding in system analysis and design.

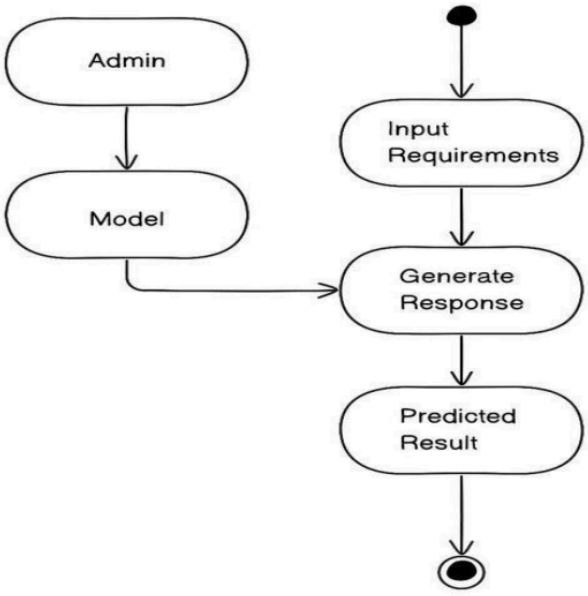


Figure 4.3: DFD Multi Level Diagram

4.3UML DIAGRAMS

4.3.1 Activity diagram

Activity diagrams are visual representations that depict workflows involving step-by-step activities and actions. They support various programming constructs which are help to the user. This diagrams are formed using a set of specific shapes connected to each other to illustrate the sequence and dependencies of activities and actions, making them a powerful tool for system analysis and design.

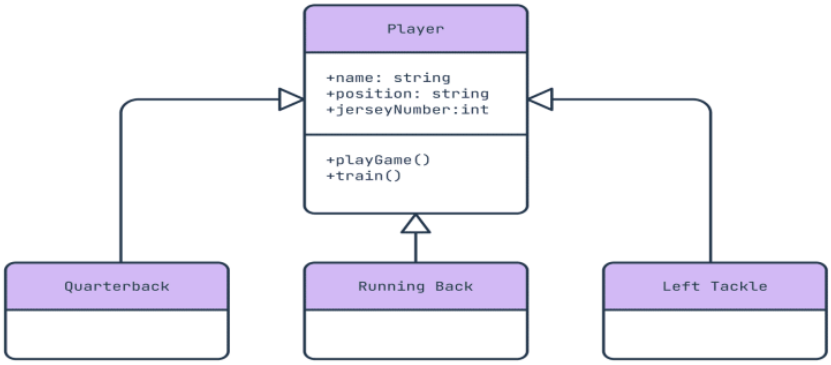


Figure 4.4: Activity Diagram

4.3.2 Sequence Diagram

A sequence diagram is an interaction diagram that illustrates the interactions and order of operations between processes.The diagram visually represents the sequence of events and object interactions over time. It specifically shows the objects and classes involved in a scenario and the messages exchanged between them to achieve the desired functionality within that scenario.

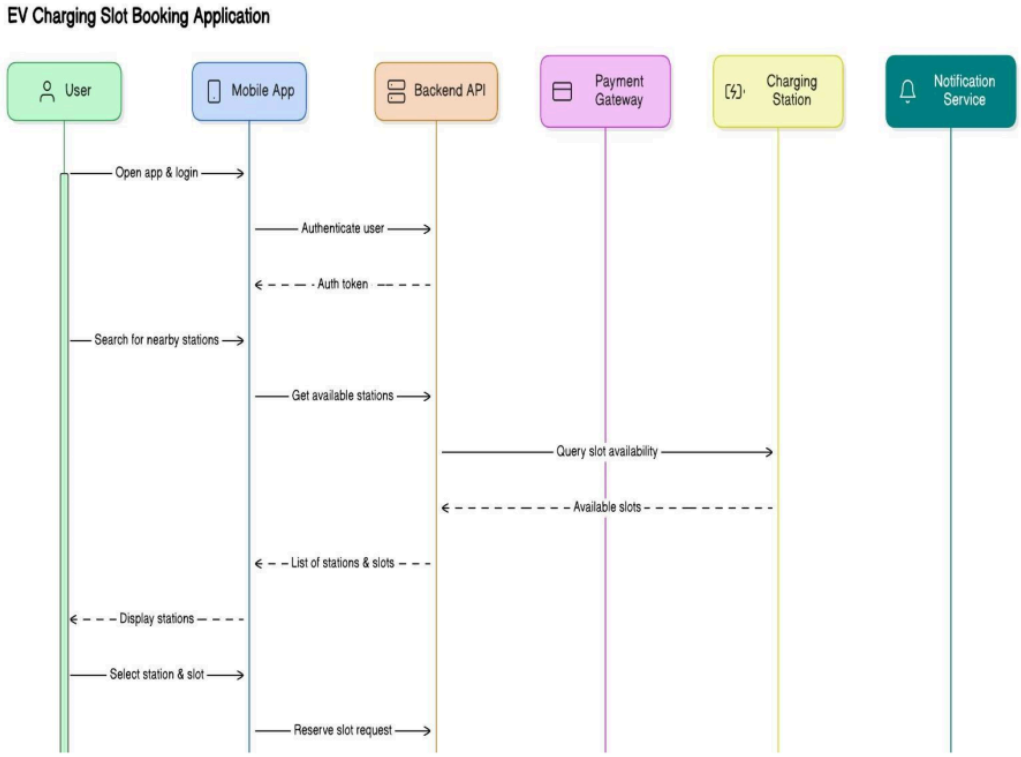


Figure 4.5: Sequence Diagram

4.3.3 Use case Diagram

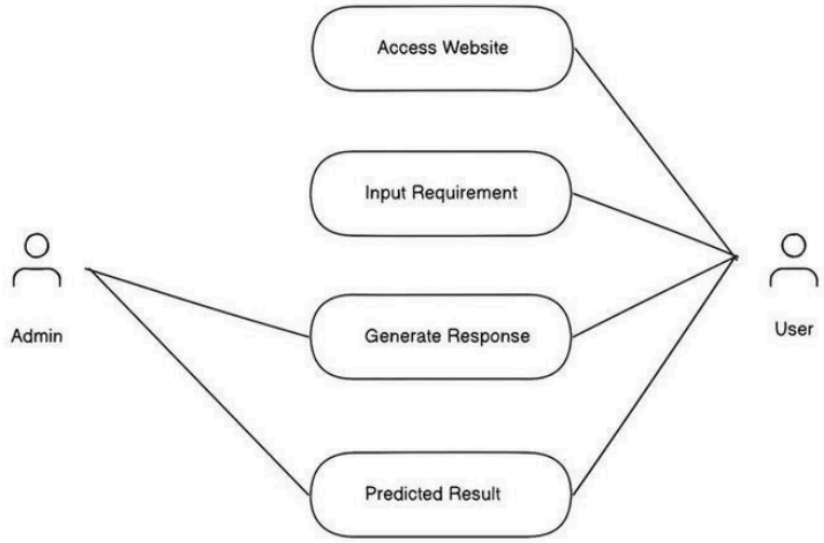


Figure 4.6: Use case Diagram

4.3.4 Class Diagram

The class diagram is a static representation in Unified Modeling Language (UML), providing a snapshot of an application's structure. It offers a visual depiction of the system's static view, focusing on the classes, their attributes, operations, and the relationships between classes. While primarily used for visualizing and documenting system aspects, the class diagram also plays a crucial role in software development.

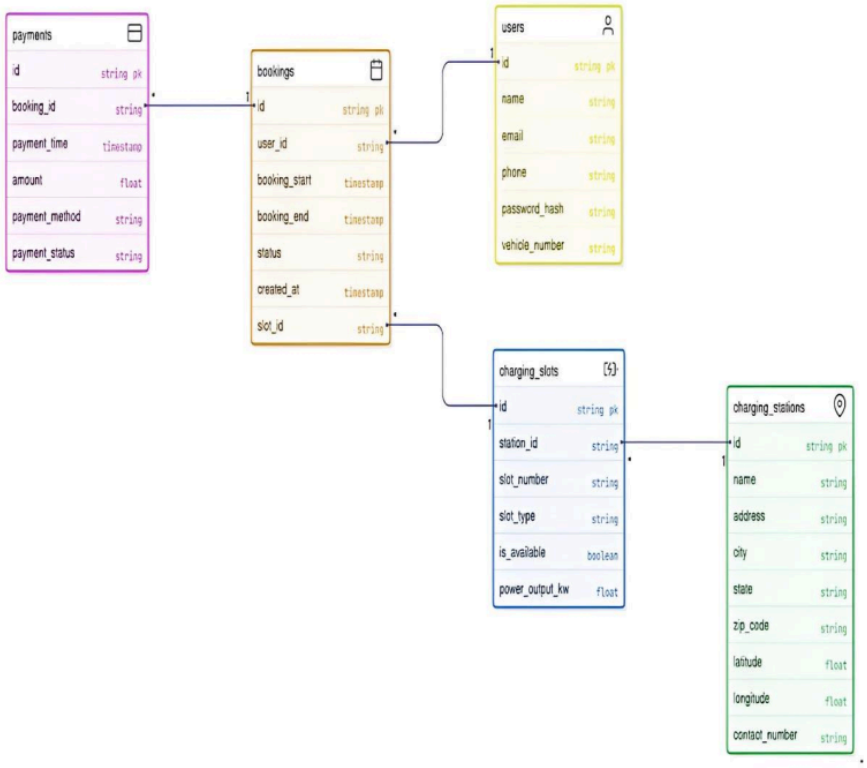


Figure 4.7: Class Diagram

4.4 ENTITY RELATIONSHIP DIAGRAMS

Entity relationship diagram (ERD) represents the relationship sets of the entities. It represents the all relationship of the project. It shows the relationships of set of entity which are stored in a database. Set of entities are similar to the all entities.

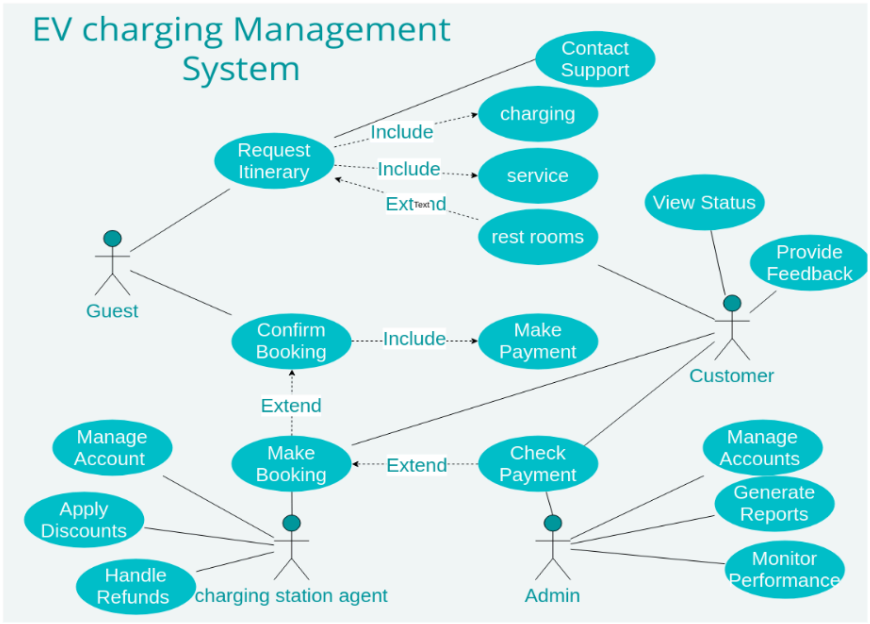
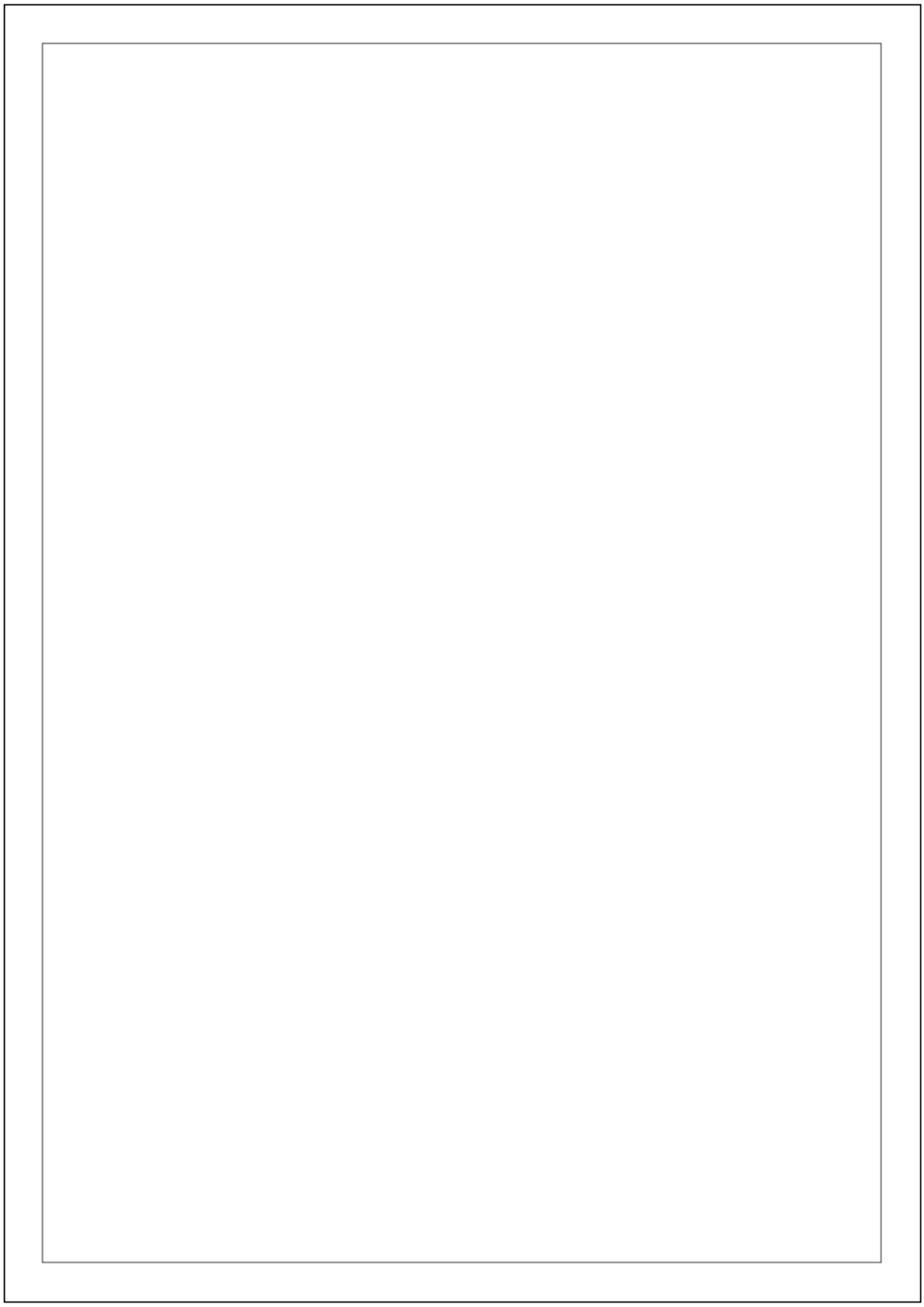


Figure 4.8: ER Diagram



CHAPTER 5:
IMPLEMENTATION DETAILS

5.1 DESCRIPTION OF TOOLS

VS Code

Visual Studio Code is a free and open-source code editor popular among developers. It offers a user-friendly interface with features like syntax highlighting, code completion, and debugging tools, making it suitable for various programming languages. VS Code goes beyond basic editing by allowing customization through themes and extensions, letting developers tailor the experience to their specific needs. This flexibility, coupled with its lightweight design, makes VS Code a powerful and versatile tool for building modern software applications.

Jupyter Notebook

Jupyter Notebook is a web-based application designed for interactive data analysis. It combines code execution, rich text explanations, and visualizations into a single document. Imagine a notebook where you can write Python code (or other languages), see the results right away, and add explanations and visualizations alongside your code. This makes it ideal for data exploration, prototyping machine learning models, and creating data science reports that are easy to understand and share.

5.2 PROGRAMMING LANGUAGE DESCRIPTION

Mysql:

MySQL is a 'relational database management system (RDBMS)' that uses Structured Query Language (SQL) to manipulate and manage data. It is widely used for storing data in website applications and supports features like transactions, indexing, and data security. MySQL is known for its computational rate, credibility and simplicity, making it a favoure choice for both small and large-scale applications.

Django:

Django is a high-level, open-source web framework written in Python that enables swift development of secure and safe websites. It sticks to the Model- View- Template (MVT) architectural pattern and includes built-in features like authentication, admin interface, and ORM for database operations.

Python:

Python is a versatile and beginner-friendly programming language known for its easy- to-read syntax. This makes it ideal for beginners but powerful enough for tasks like web development, data analysis, and machine learning. As an interpreted language, Python simplifies the development process by eliminating the need for compilation before running the code.

5.3 IMPLEMENTATION DETAILS

The implementation steps of the project :

1. Data Collection: Real estate data is collected from various sources containing information about houses including location, square footage, number of bathrooms, bedrooms, balconies, area type, and availability.
2. Data Preprocessing:

   The collected data is cleaned and preprocessed to handle missing values, encode categorical variables, and scale numerical features.
3. Model Training:

   A machine learning model is trained using the preprocessed data. Various algorithms like Linear Regression, Random Forest, or Gradient Boosting are used.
4. Web Application Development:

   A web application is developed using Python for the frontend, and Django for the backend. The application a offers to the users to input the features of a slot and get an estimated price prediction for this slot.
5. Deployment:

   The trained model is deployed on the Django server, and the web application is deployed on a hosting service like Heroku or AWS.

5.4 ALGORITHM DETAILS

The Electric Vehicle Charging Slot Booking Application follows a structured algorithm to manage user activities efficiently. First, the system handles user registration and login. During registration, the application collects user information such as name, email, phone number, and password.

After validating the input fields to ensure correct data entry, the user information is stored securely in the MySQL database. For login, the system requires the user's email and password, which are verified against stored records. If the credentials match, access is granted; otherwise, an appropriate error message is displayed.

Once logged in, the user can find available charging stations. The user selects or enters the desired city or location, upon which the application queries the database for all active charging stations within that area. The system then displays a list of matching stations along with relevant details like station name, address, and number of available slots.

To book a charging slot, the user selects a station from the displayed list. The application then shows the available time slots for that station. The user selects a preferred date and time slot. Before confirming the booking, the system checks the database to ensure the slot is still available. If the slot is free, the booking is processed and stored; otherwise, the user is asked to select another available slot. After a successful booking, the database is updated to reflect that the slot is no longer available for others.

5.5 OUTPUT OF IMPLEMENTATION

The integration of the EV charging station and slot booking system heads to the creation of a accessible web-based platform that allows users to simply find nearby charging stations, check the availability of slots, and make reservations for their required dates and times. The system comprises intuitive interfaces that present station information, including name, location, charging type, and current slot availability. Users can satisfactory reserve a charging station by choosing their required options and receiving a confirmation along with a booking ID.

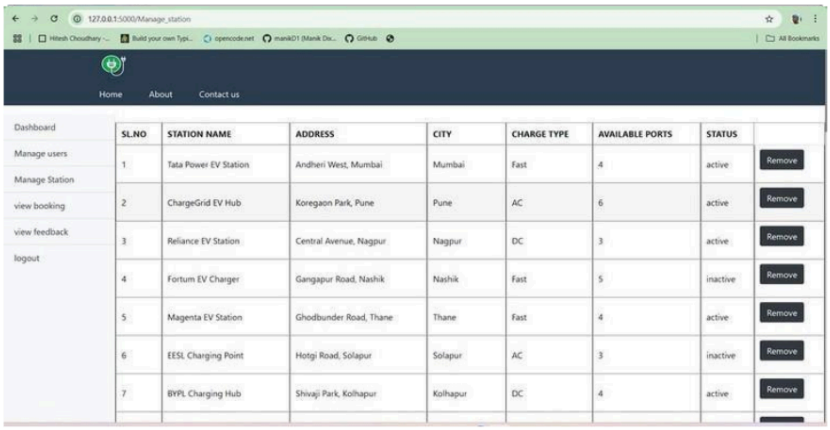


Figure 5.1: Slot allocation

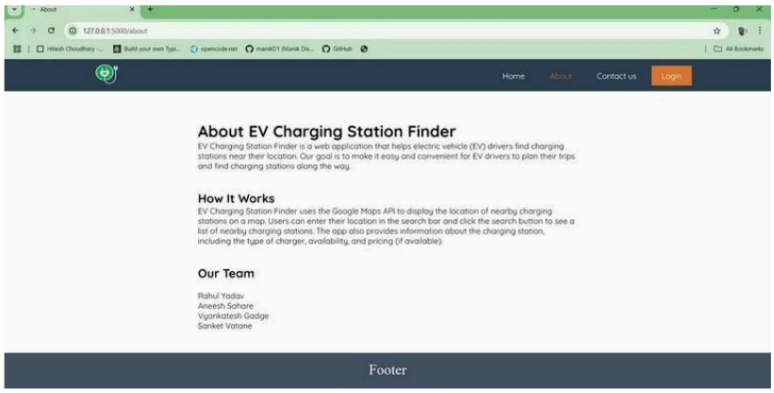


Figure 5.2: About us

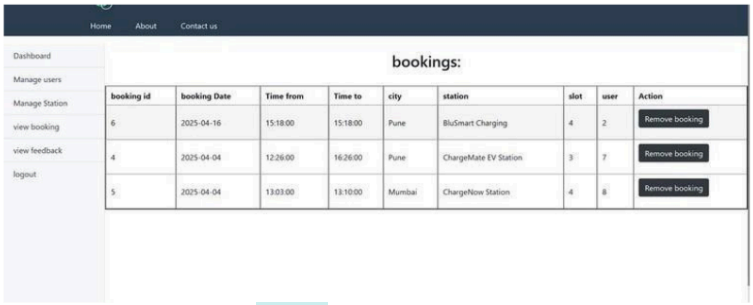


Figure 5.3: Database

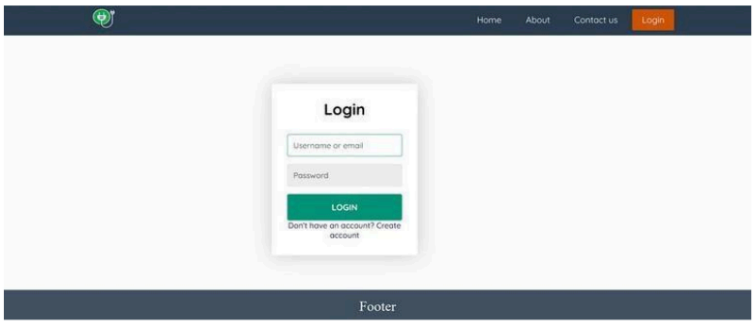


Figure 5.4: Login

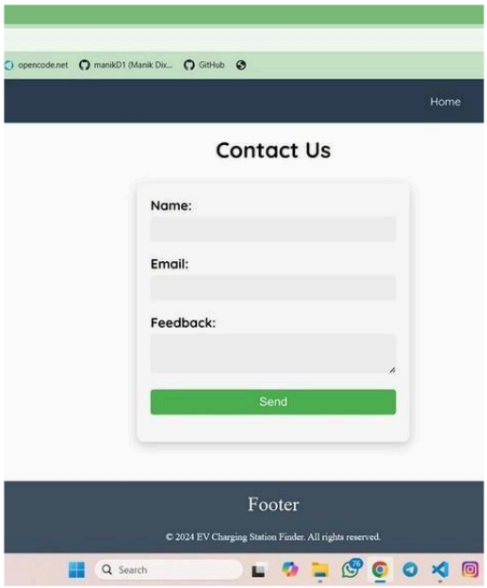


Figure 5.5: Contact page

CHAPTER 6 :
TESTING

6.1 INTRODUCTION

Testing is very important phase of the software development life cycle. The purpose of this phase is to validate the lifetime of the system. This is compulsory Period of time. Data Information given in this section provides required for the testing operations that should be approved for the "opinion mining" of scheme. Tester must estimate test of every component along with it, it should write down test cases which belongs to user requirement and system structure.

Principle of Testing:

- To know the system performance.
- To recognize the functionality of each and every individual module.
- To verify whether system is functioning as per the user requirements.

6.2 TYPE OF TESTING

1 ' White-box Testing':

The White box testing is executed by the tester which possesses valuable, vast knowledge of the programming languages. White box testing applied on algorithm or source code of the following project. It is the procedure technique of applying the input to the given project and verifying the system process input to produce output. The interior information is required to known to the tester. This technique is also called as Transparent Testing. These tests require the code to check so it is necessary for the tester to have the knowledge coding.

Given Below are the methods of White Box testing:

- 'Programming style'
- 'Control method'
- 'Source language'
- 'Database design'

This type of a test is useful to beat defects at structural level. This test goes lower the top or functional layer to expose defects.

2 'Black box Testing':

In this technique, we check requirements with actual results. Tester does not require prior knowledge the internal reasoning of the given project. Functional Testing is performed in black box testing. The knowledge about the internal execution of the program or is not required. In Accordance with the testing plan, Followed below are the required functionality which cover under the black box testing.

Server connection

- Data generation
- Data extraction
- Machine learning algorithm process
- Prediction
- Results

3. 'Unit Testing':

Unit testing refers to the minor particles of a system to check it might be as methods, tasks, classes of code, GUI of the given system. Tester will test each unit of the system to investigate whether the module is suitable for the system. This technique covers a few advantages for example, the exceptions, bugs and harmful errors found at early stage.

4'Integration Testing':

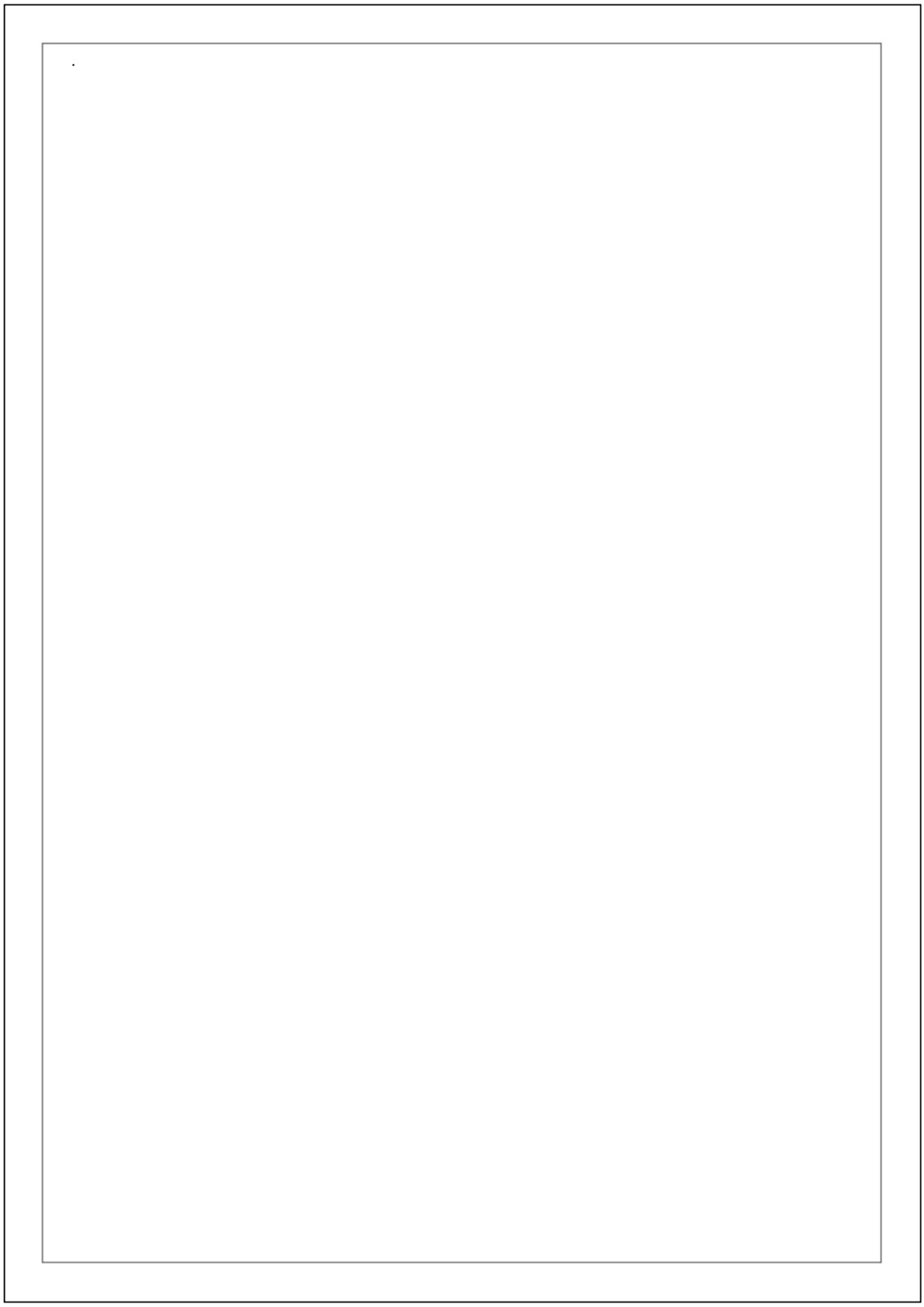
In the Integration testing, different modules are combined together and tested. To Swap information among various modules of the system. When whole testing work is concluded, the deployment of software on customer system is carried out

5'Validation testing':

Initially Validates each and every model which is created in the upcoming testing phase making sure that the required software is accurately positioned. Also Validates whether system is working precisely as it is supposed to be.

- Justification of The Given system and Execution of the system behavior.
- probable data given as required input and captures projected result.

6 'System Testing': Next up is the output testing, The required format is essential for the productivity of the application. The processed output is summarized to the user. The favorable format for the user is predominately PPT. System Testing is the final operation performed in order to complete the step. After System testing , all the given features are available to the user



6.3 TEST CASES AND RESULTS

After System testing, it detects the faults and disorder in the code. The detected faults and disorders are resolved by few important methods in less time. Each module is tested and prediction about the result are conducted. We later check it the predicted output matches the actual output and also check if the obtained output resolves the required needs of the user. Below given table indicates and checks test cases which might be vital for project.

Below Table shows the suite of test cases which are executed and passed.

Test Cases for Electrical Vehicle Charging Slot Booking Application

ID	Test Case	Expected Result	Obtained Result	P/F
TC1	Search Charging Station	List of stations displayed	Charging stations shown	P
TC2	Book Charging Slot	Slot booked successfully	Slot confirmed and booked	P
TC3	Payment Process	Payment completed successfully	Payment processed without error	P
TC4	User Login	User logged in successfully	Login successful, dashboard opened	P
TC5	Admin Login and Station Update	Station details updated properly	Station updated as expected	P
TC6	Real-Time Slot Update	Slot availability shown live	Slot status updated instantly	P
TC7	Cancel Booking	Booking cancelled and slot free	Slot released after cancellation	P
TC8	View Booking History	Past bookings shown correctly	Booking history displayed properly	P
TC9	Station Filtering	Filter displays relevant stations	Filtered results shown correctly	P
TC10	Mobile Responsiveness	Mobile-friendly interface shown	UI responsive on phone	P

CHAPTER 7: CONCLUSION

7.1 CONCLUSION

The successful development of the Electric Vehicle Charging Slot Booking Application marks a significant step toward solving one of the key challenges in the electric mobility sector: inefficient and unpredictable access to charging infrastructure. By providing users with a convenient way to search, locate, and reserve charging stations based on real-time data, the application not only saves time but also enhances the reliability of EV usage for daily commuters.

From a user perspective, the platform ensures greater transparency, less waiting time, and the ability to make informed decisions regarding where and when to charge. For charging station operators, the system provides tools to streamline station traffic, manage bookings efficiently, and maintain updated status records. The backend architecture is designed for scalability and real-time synchronization, ensuring seamless interaction between users and administrators.

Overall, this system is a meaningful contribution to the growing ecosystem of electric mobility and sets a foundation for future enhancements and real-world deployment.

7.2 FUTURE SCOPE

As the adoption of electric vehicles accelerates, there is immense potential to expand and enhance the capabilities of our charging slot booking system. In the future, the application can be integrated with **renewable energy-powered charging stations**, such as those using solar or wind energy. This would reduce carbon dependency and further promote sustainable transportation.

Another important area of development is the **use of artificial intelligence** to predict user demand patterns. AI algorithms can be implemented to analyze user behavior, station traffic, and booking trends to suggest optimal time slots and balance peak load periods more effectively.

The system could also incorporate **dynamic pricing models**, where rates change based on demand, time of day, or energy source availability. This would not only optimize the utilization of charging stations but also encourage users to charge during off-peak hours.

CHAPTER 8: APPENDIX A

8.1 JOURNAL PAPER

The title of the project is “Electric Vehicle Charging Station and Slot Booking Web Application”. Journal Papers Submitted to ‘Iconic Research And Engineering Journals’.

IRE JOURNALS

ICONIC RESEARCH AND ENGINEERING JOURNALS

[An International Open Access, Peer-reviewed, Refereed Journal]

Certificate of Publication

The board of IRE Journals
is hereby awarding this certificate

YADAV RAHUL

In recognition of the publication of the paper entitled

ELECTRICAL VEHICLE CHARGING SLOT BOOKING APPLICATION USING PYTHON, FLASK, AND MYSQL

Publication In e-Journal
VOLUME 8 ISSUE 10 APR 2025

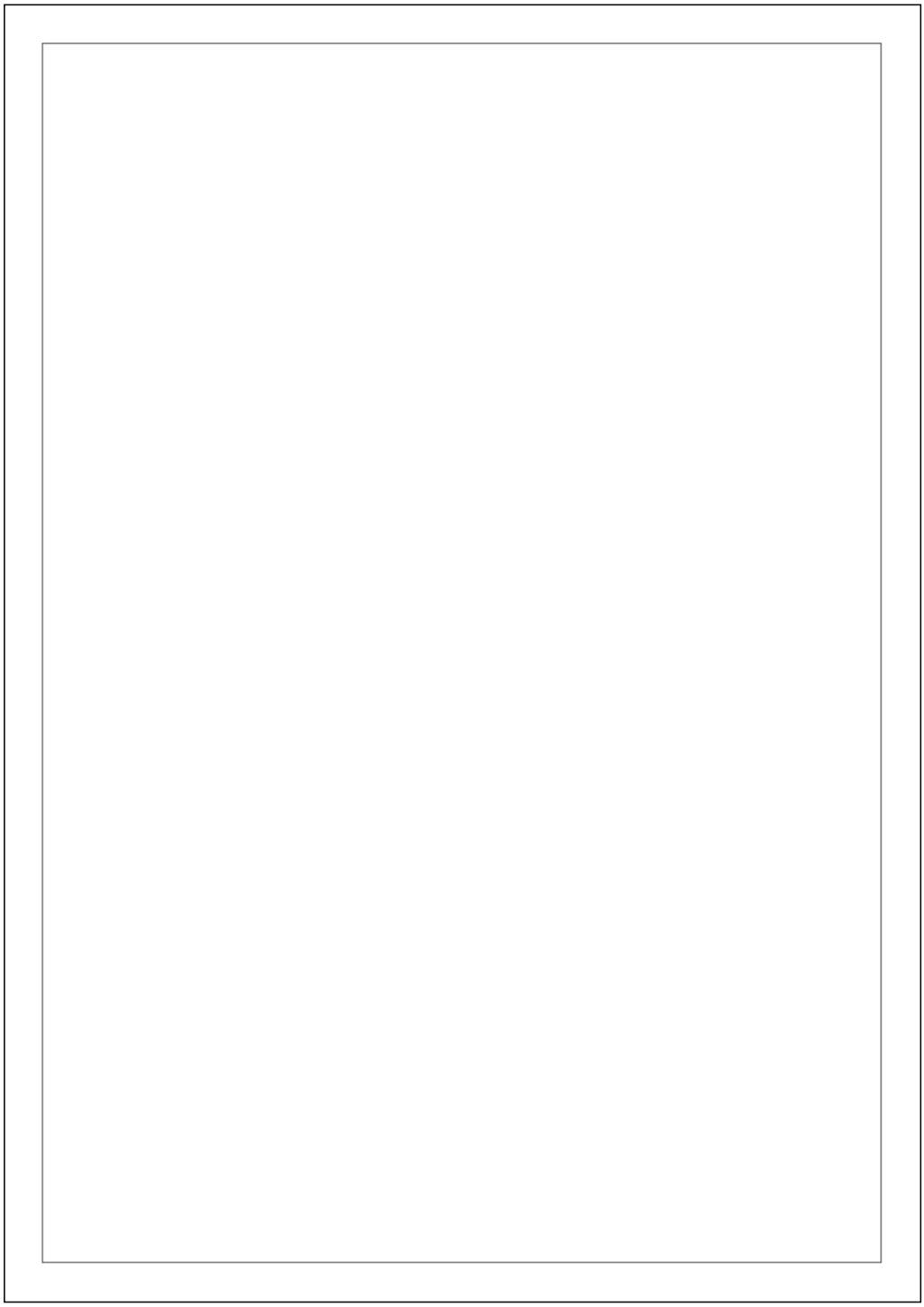
PAPER ID :- 1707841

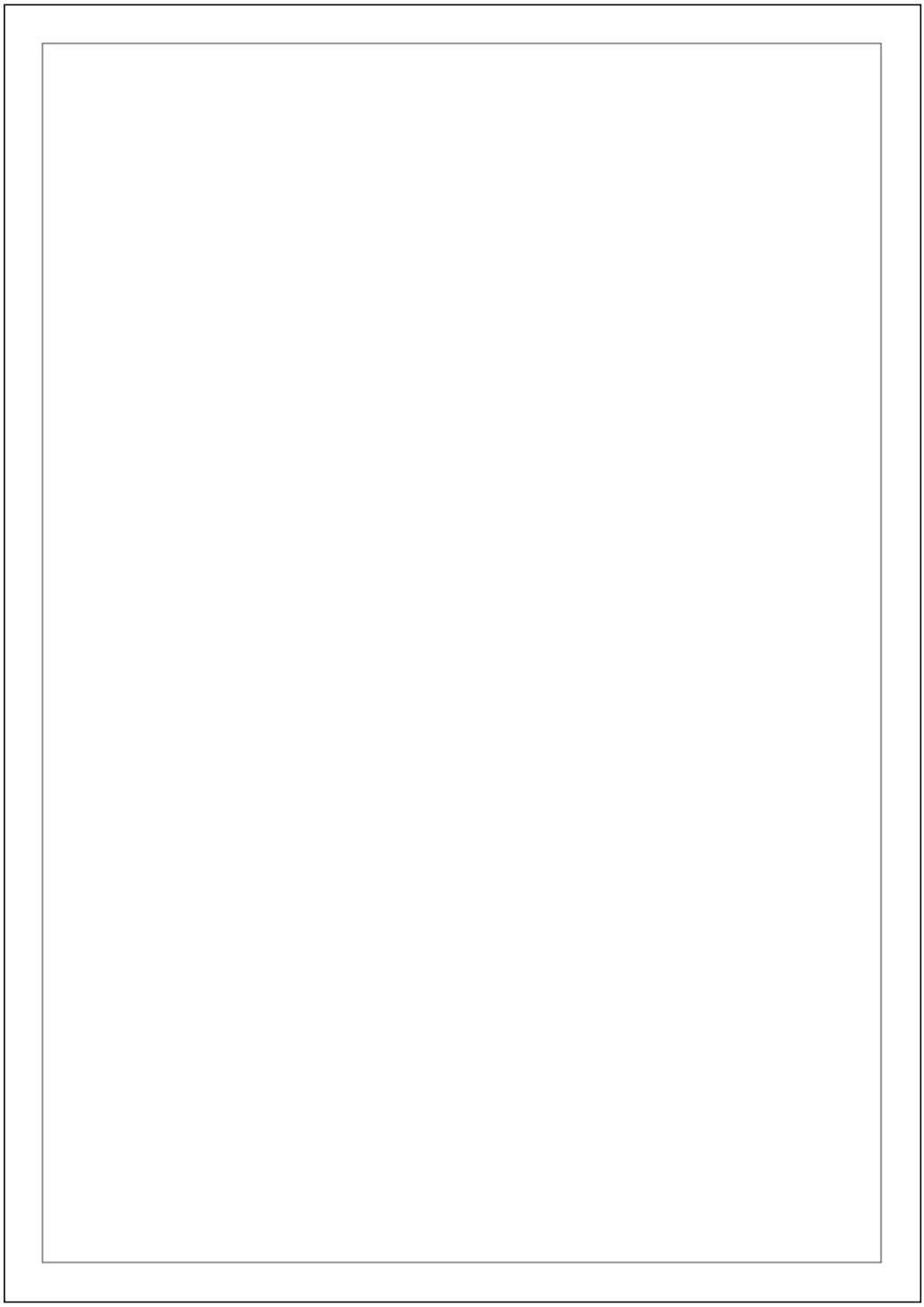
Sanket Patel
EDITOR IN CHIEF

ICONIC RESEARCH AND ENGINEERING JOURNALS

Website : www.irejournals.com | Email ID : irejournals@gmail.com | ISSN : 2456 - 8880

Figure 8.1: Certificate of publication





CHAPTER 9: APPENDIX B

9.1 PLAGIARISM REPORT OF PROJECT REPORT

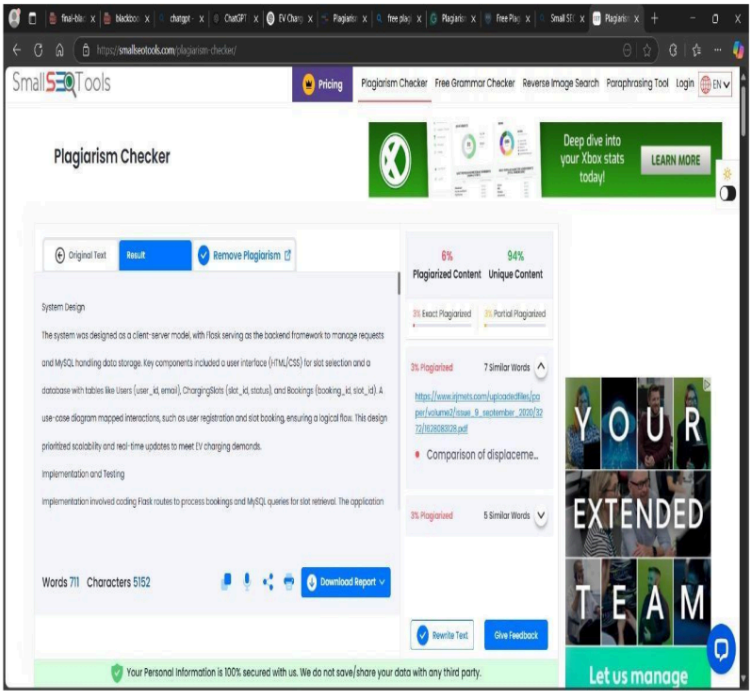


Figure 9.1: Plagiarism Report

CHAPTER 10: APPENDIX C

10.1 COPYRIGHT FORM

IRE JOURNALS

Copyright Transfer and Declaration

1. I hereby transfer the Copyright of the title: _____

By (Authors): Electrical vehicle charging slot booking application

Full Postal address of Principal Author (With phone no. and mail id):

vyankateshgadge8@gmail.com

2. I understand that the Editor-in-Chief-IRE may transfer the Copyright to a publisher at his discretion.
3. The author(s) reserve(s) all proprietary rights such as patent rights and the right to use all or part of the article in future works of their own such as lectures, press releases, and reviews of textbooks. In the case of republication of the whole, part, or parts thereof, in periodicals or reprint publications by a third party, written permission must be obtained from the Editor-in-Chief IRE.
4. I hereby declare that the material being presented by me in this paper is our original work, and does not contain or include material taken from other copyrighted sources. Wherever such material has been included, it has been indented or/and identified by quotation marks and due and proper acknowledgments given by citing the source at appropriate places.
5. The paper, the final version of which I enclose, is not substantially the same as any that I/we have already published elsewhere.
6. I/we have not sent the paper or any paper substantially the same as the enclosed one, for publication anywhere else.
7. Furthermore, the author may only post his/her version provided acknowledgment is given to the original source of publication and a link is inserted to the published article on the IRE website.
8. The submitted/enclosed camera-ready paper is thoroughly proofread by me and in conformity with the instructions for authors communicated to me.
9. If any plagiarism is found in my camera-ready paper after Publication, I am the whole responsible not IRE or IRE Board members.

Author Signature	Sujata	Vyankatesh	Sanket	Aneesh	Rahul
Author's Name (in Block letters)	Prof. Sujata Jogdand Gaikwad	Gadge Vyankatesh Narsing	Vatane Sanket Namdev	Sahare Aneesh Vijay	Yadav Rahul Indradev

DATE: 17/04/25

PLACE: Pune

• Kindly send a scanned copyof the completed and duly signed form by email to the Editor-in-Chief at irejournals@gmail.com.

CHAPTER 10:
REFRENCES

1. C. Zhang, Y. Wu, Y. Li, and M. Zhang, "Smart Electric Vehicle Charging System Based on Internet of Things," IEEE Access, vol. 7, pp. 134654–134666, October 2019.

2. H. Shareef, M. S. Ahmed, A. Mohamed, and E. Al Hassan, "Review on Home Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers," IEEE Access, vol. 6, pp. 24498– 24509, April 2018.

3. H. Shareef, M. S. Ahmed, A. Mohamed, and E. Al Hassan, "Review on Home

4. Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers," IEEE Access, vol. 6, pp. 24498–24509, April 2018.

5. Ghosh, S. Sen, A. Sinha, and B. Mukherjee, "IoT-Enabled Smart Charging Station for Electric Vehicle in a Smart City," IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3375–3384, May 2021.

6. Y. Wang, M. Peng, Y. Lu, and Y. Liang, "Optimal Scheduling Strategy for Electric Vehicle Charging Stations Based on User Priority and Renewable Energy," IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3306–3316, July 2020.

7. L. Chen, Y. He, and X. Huang, "An Intelligent Charging Strategy for Electric

8. Vehicle Stations Considering Dynamic Pricing," IEEE Transactions on Industrial Electronics, vol. 67, no. 8, pp. 6500–6509, August 2020.

9. H. Li, Y. Xu, and Z. Liu, "Optimization of Electric Vehicle Charging Stations in Smart Grid Using Machine Learning Algorithms," IEEE Transactions on Power Systems, vol. 35, no. 2, pp. 1241–1252, February 2020.

09_Plagiarism_03 June 2025

ORIGINALITY REPORT

9%	7%	2%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Bharati Vidyapeeth Deemed University College Of Engineering Student Paper	2%
2	www.ijtsrd.com Internet Source	1%
3	bvmengineering.ac.in Internet Source	1%
4	Submitted to Manipal University Student Paper	<1%
5	programmingtrick.com Internet Source	<1%
6	Submitted to AlHussein Technical University Student Paper	<1%
7	philarchive.org Internet Source	<1%
8	www.ssgmce.ac.in Internet Source	<1%
9	Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical and Computer Technologies", CRC Press, 2025 Publication	<1%
10	Submitted to University of Stirling Student Paper	<1%
11	eprints.utar.edu.my Internet Source	<1%

12	eitca.org Internet Source	<1%
13	Submitted to Kingston University Student Paper	<1%
14	Submitted to Swinburne University of Technology Student Paper	<1%
15	Submitted to University of Georgia Student Paper	<1%
16	Submitted to University of Greenwich Student Paper	<1%
17	softradar.com Internet Source	<1%
18	arxiv.org Internet Source	<1%
19	moldstud.com Internet Source	<1%
20	www.coursehero.com Internet Source	<1%

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		