A

MAJOR PROJECT REPORT ON

SMART BUS BOOKING SYSTEM

Submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY

SEETHA VARDHAN	218R1A0457
SIMMA SHRAVAN KUMAR	218R1A0458
SURAM ARAVIND	218R1A0459
THADAPALLY MOHAN KRINSHNA	218R1A0460

Under the Esteemed Guidance of

Mr. R. SRIDHAR
Assistant professor

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)

Kandlakoya(V), Medchal(M), Telangana – 501401

(2024-2025)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Approved by AICTE, Affiliated to JNTU Hyderabad, Accredited by NBA)
Kandlakoya(V), Medchal(M), Telangana – 501401

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that major project work entitled **"SMART BUS BOOKING SYSTEM"** is being submitted by **S. VARDHAN** bearing Roll No: **218R1A0457, S. SHARVAN KUMAR** bearing Roll No**: 218R1A0458, S. ARAVIND** bearing Roll No: **218R1A0459, T. MOHAN KRINSHNA** bearing Roll No: **218R1A0460** in B.Tech IV-I semester, Electronics and Communication Engineering is a record Bonafede work carried out by then during the academic year 2024-25.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
Mr. R. SRIDHAR	Dr. SUMAN MISHRA

EXTERNAL EXAMINER

ACKNOWLEDGEMENTS

We sincerely thank the management of our college **CMR Engineering College** for providing required facilities during our project work. We derive great pleasure in expressing our sincere gratitude to our Principal **Dr. A. S. Reddy** for his timely suggestions, which helped us to complete the project work successfully. It is the very auspicious moment we would like to express our gratitude to **Dr. SUMAN MISHRA,** Head of the Department, ECE for his consistent encouragement during the progress of this project.

We take it as a privilege to thank our project coordinator **Dr. T. SATYANARAYANA,** Associate Professor, Department of ECE for the ideas that led to complete the project work and we also thank him for his continuous guidance, support and unfailing patience, throughout the course of this work. We sincerely thank our project internal guide **Mr. R. SRIDHAR,** Assistant Professor of ECE for guidance and encouragement in carrying out this project work.

DECLARATION

We hereby declare that the major-project entitled **"SMART BUS BOOKING SYSTEM"** is the work done by us in campus at **CMR ENGINEERING COLLEGE,** Kandlakoya during the academic year 2024-2025 and is submitted as major project in partial fulfilment of the requirements for the award of degree of **BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGINEERING FROM JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD.**

S. VARDHAN	**(218R1A0457)**
S. SHARVAN KUMAR	**(218R1A0458)**
S. ARAVIND	**(218R1A0459)**
T. MOHAN KRINSHNA	**(218R1A0460)**

ABSTRACT

The Bus Booking System is a web-based application designed to simplify and streamline the process of booking bus tickets. It provides an efficient platform for users to search for available bus routes, select preferred seats, and make secure online payments. The system aims to reduce the dependency on traditional ticketing methods by offering real-time updates on bus schedules, seat availability, and fare details. It enhances user experience through features such as instant booking confirmation, user profile management, and trip cancellation options. Additionally, the system supports administrators in managing bus schedules, monitoring bookings, and generating sales reports. By automating the booking process, the Bus Booking System significantly improves operational efficiency, reduces human errors, and ensures a seamless travel experience for passengers.

A centralized dashboard enables real-time monitoring of bookings and occupancy levels, preventing overbooking and ensuring efficient fleet management. Advanced reporting tools generate valuable insights into sales trends, passenger behavior, and system performance, assisting in data-driven decision-making and business growth. By automating routine administrative tasks, the system reduces human errors, enhances operational accuracy, and optimizes resource allocation, ultimately leading to a more effective and streamlined transport service.

By integrating cutting-edge travel automation technologies, the Bus Booking System revolutionizes the ticketing process for both passengers and operators. It ensures a fast, secure, and convenient booking experience, significantly enhancing customer satisfaction and transport management efficiency. Additionally, by minimizing the use of paper-based ticketing, the system promotes eco-friendly and sustainable travel solutions, making it a future-ready and innovative approach to modern transportation.

Keyword: Bus Booking System, Online Ticketing, Seat Reservation, Bus Schedule Management, Real-time Booking, Online Payment, Passenger Management, Travel Automation.

CONTENTS

CHAPTERS **PAGE**

CHAPTER-1 INTRODUCTION

CHATER-2 LITERATURE SURVEY

CHAPTER-3 SYSTEM REQUIREMETS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In the modern era of rapid development and global connectivity, transportation systems are essential in enabling the smooth movement of people and goods. With continuous urban growth and rising population densities, the need for dependable, accessible, and efficient transport solutions has never been greater. Among the different transportation modes, buses remain a widely used and economical choice, particularly for both local and long-distance travel. They offer flexibility and affordability, making them an indispensable service for daily commuters, cross-city travelers, and tourists. Yet, as traveler expectations evolve with advancing technology, there is an increasing demand for a smarter, more streamlined, and user-oriented bus booking system

1.1 OVERVIEW OF THE PROJECT

In today's fast-paced and interconnected world, transportation systems play a pivotal role in facilitating the movement of people and goods. As cities expand and populations grow, the demand for reliable, accessible, and efficient transportation becomes increasingly vital. Among the various modes of transportation, buses stand out as one of the most popular and affordable means of travel, especially for intercity and intracity journeys. Buses provide a cost-effective and flexible option for passengers, serving as a lifeline for daily commuters, long-distance travelers, and tourists alike. However, to meet the evolving expectations of modern travelers, there is a growing need for an advanced, efficient, and user-friendly bus booking system.

Despite the widespread use of bus transportation, traditional methods of bus ticket booking remain outdated and inefficient. Manual reservations at ticket counters, reliance on third-party agents, and phone bookings are still common practices in many regions. These conventional methods are often time-consuming, requiring passengers to stand in long queues or rely on intermediaries who may charge additional fees. Moreover, manual processes are prone to human errors, such as double bookings, incorrect entries, and mismanagement of schedules. Such inefficiencies can lead to passenger dissatisfaction, operational losses, and increased workload for bus operators. The lack of real-time information on bus schedules, seat availability, and pricing further adds to the inconvenience, leaving passengers with limited options and flexibility when planning their journeys.

To address these challenges and modernize the bus ticketing process, the Bus Booking System has been developed as a comprehensive, web-based solution. This system is designed to automate and simplify the entire process of booking bus tickets, offering a seamless and hassle-free experience for both passengers and bus operators. The primary objective of the Bus Booking System is to eliminate the drawbacks associated with traditional booking methods by providing a digital platform where passengers can book tickets anytime and from anywhere.

The system’s intuitive user interface allows passengers to search for available buses based on their preferred source and destination, view detailed information on bus schedules, and select seats according to their preferences. Real-time updates ensure that users have access to the latest information regarding seat availability and pricing, enabling them to make informed decisions. The integration of secure online payment gateways ensures a smooth and safe transaction process, eliminating the need for cash payments and reducing the risk of financial discrepancies.Beyond passenger convenience, the Bus Booking System also brings significant advantages to bus operators and administrators. The system enables efficient management of bus schedules, routes, fares, and seat allocations, all from a centralized platform. Automated booking processes reduce the need for manual intervention, thereby minimizing human errors and operational costs. Additionally, the system provides valuable insights through comprehensive reports on booking trends, revenue generation, and customer preferences, empowering operators to make data-driven decisions for business growth.

Moreover, the Bus Booking System offers additional features such as booking history management, easy ticket cancellation, and customer support, further enhancing the user experience. The platform is designed to be scalable and adaptable, capable of supporting a growing user base and accommodating future enhancements such as mobile application integration and dynamic pricing models.

In conclusion, the Bus Booking System represents a significant step forward in revolutionizing the bus transportation industry. By leveraging modern technology to automate and streamline the ticket booking process, the system not only improves operational efficiency for bus operators but also delivers a convenient, transparent, and user-centric experience for passengers. This digital transformation ultimately contributes to a more connected and accessible transportation ecosystem, meeting the demands.

The Bus Booking System offers passengers the convenience of booking bus tickets from anywhere at any time, using an intuitive and interactive interface. Users can easily search for available buses based on source and destination, select seats according to preference, and make secure online payments. The system provides real-time updates on seat availability, bus schedules, and fare details, ensuring transparency and enhancing the overall user experience.

From an administrative perspective, the system facilitates effective management of bus schedules, routes, and fares, while providing insightful reports on bookings and revenue. It minimizes operational costs and reduces human errors through process automation.

Additionally, features such as booking history, cancellation options, and user profile management contribute to a seamless and personalized user journey. Overall, the Bus Booking System bridges the gap between passengers and bus operators by offering a reliable, efficient, and scalable platform that transforms the conventional ticketing process into a modern, digital experience.

1.2 OBJECTIVE OF THE PROJECT

The primary objective of the "Smart Bus Booking System" is to revolutionize the bus ticket booking process by creating a secure, efficient, and user-friendly digital solution. The project aims to address existing challenges in public transportation, including lengthy queues, ticket verification delays, and lack of real-time seat availability information. By integrating Aadhaar-based authentication, the system ensures accurate user identification and enhances security, making it easier for passengers to book tickets without the need for physical documents.

Another key objective is to improve the convenience of the ticket booking experience for passengers and the ticket validation process for conductors. For passengers, the app provides a seamless way to select their travel route, book seats, and make payments. With the QR code-based e-ticket system, passengers can access their tickets digitally and avoid the hassle of carrying paper tickets. For conductors, the app offers an efficient method to verify tickets by scanning QR codes, thereby streamlining the boarding process and minimizing human error. The project also aims to enhance the accessibility of public transportation by offering real-time information on bus schedules and seat availability. Through a user-friendly interface, passengers can easily check and book their desired seats without relying on traditional booking counters. This feature, coupled with dynamic fare calculation, empowers passengers to make informed choices about their travel plans and enables bus operators to manage seat occupancy efficiently.Promoting environmental sustainability is also a central goal. By eliminating paper tickets and reducing the need for physical ticket counters, the app reduces paper waste and contributes to an eco-friendlier public transportation system.

The app’s digital architecture supports ticket cancellations and refunds, making it adaptable to the needs of modern users while maintaining eco-conscious practices.

In summary, the "Smart Bus Booking System" aims to simplify and enhance the bus booking and ticketing experience through digital innovation. By focusing on security, convenience, and sustainability, the project seeks to transform public transportation, making it more accessible, secure, and environmentally friendly for all users.

1.3 MOTIVATION

The development of the "Smart Bus Booking System" was inspired by the need to modernize and simplify the bus ticketing process. In traditional bus booking systems, passengers face long queues, physical ticket requirements, and delayed boarding times, which create an overall unsatisfactory experience. With the rise of digital solutions, there is a growing demand for more efficient, accessible, and secure transportation options.

Additionally, the adoption of digital payments has seen a surge, especially with the emphasis on contactless transactions post-COVID-19. A mobile-based booking system meets the needs of today’s digitally literate population, allowing for seamless transactions and a more secure process compared to cash-based payments.

The push for sustainable solutions also serves as a key motivator. By eliminating paper-based tickets, this project aligns with eco-friendly practices, contributing to reduced waste and a greener approach to public transportation. The combination of these motivations drives the need for a "Smart Bus Booking System" that serves the public's evolving expectations for speed, security, and sustainability.

1.4 PROBLEM STATEMENT

The current bus ticketing system is fraught with issues that reduce operational efficiency and customer satisfaction. Common problems include:

- **Long Queues and Delays**: Passengers often wait in long queues to purchase tickets, especially during peak hours, leading to frustration and delayed boarding.
- **Dependence on Physical Tickets**: Physical tickets are easy to lose, can get damaged, and require manual verification, which can be slow and prone to errors.
- **Manual Verification Errors**: Conductors manually check each passenger's ticket, which can result in human error and slows down the boarding process.
- **Inefficient Data Tracking**: Traditional systems make it difficult to track real-time data on seat availability, booking history, and scheduling, which impacts both users and operators.

The "Smart Bus Booking System" seeks to address these issues by providing a secure, real-time digital solution that streamlines ticket booking, payment, and verification while reducing manual dependency and errors.

1.5 SCOPE OF THE PROJECT

The "Smart Bus Booking System" is designed to serve passengers and conductors by offering an easy-to-use mobile platform for booking and managing bus tickets. Key features include:

- **Aadhaar Authentication**: Using Aadhaar for identity verification ensures a secure and reliable user authentication process, minimizing fraud.
- **QR-based Ticketing**: Digital tickets with QR codes enable quick scanning and boarding, improving the efficiency and reliability of the ticket validation process.
- **Real-Time Data Handling**: The app provides real-time updates on seat availability, bus schedules, and booking statuses, allowing passengers to make informed booking decisions.
- **Conductor Support Interface**: A separate interface for conductors facilitates QR code scanning and ticket verification, reducing manual tasks and errors.

While the app is focused on improving the bus booking experience, it does not currently support other modes of transportation. Additionally, the scope does not include offline ticket booking, as the system relies on internet connectivity for real-time updates and transactions.

1.6 METHODOLOGY

The development of the "Smart Bus Booking System" follows a structured approach to ensure efficient design, implementation, and testing. Key steps in the methodology include:

1. **Research and Requirements Gathering**: Initial research focused on understanding current ticketing challenges, evaluating authentication and payment methods, and reviewing similar digital ticketing systems.
2. **Technology Stack Selection**: Based on the requirements, Java and Kotlin were chosen for the frontend development due to their compatibility with Android devices, while PHP with Laravel handles backend processes. SQL serves as the database to manage user data, booking records, and real-time updates.
3. **System Design**: A modular design approach was adopted to ensure each component (e.g., user interface, QR code generation, conductor interface) can be developed, tested, and refined independently.
4. **Development**: The application was developed in stages, with continuous testing and feedback loops to identify and resolve issues early. PHP and Laravel were used for API development to connect the frontend with backend services securely.
5. **Testing and Validation**: Extensive testing, including unit testing, integration testing, and user acceptance testing, was conducted to ensure functionality, security, and user-friendliness of the app.
6. **Deployment and Maintenance**: Once tested, the app will be deployed on app stores, with regular maintenance and updates to ensure compatibility, security, and performance improvements over time.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING SYSTEM

The existing bus booking systems are primarily manual or semi-automated, where users typically book tickets through physical ticket counters, travel agencies, or basic online platforms with limited functionalities. In many cases, users need to visit bus terminals to check bus availability, select seats, and make payments. Some online systems do exist but often lack real- time seat selection, secure payment gateways, and instant digital ticket generation. These systems may also rely heavily on paper tickets, which can be lost or damaged, causing inconvenience to travelers. Additionally, user authentication processes in many existing platforms are minimal, posing security risks during transactions.

Disadvantages

Lack of Real-Time Seat Availability:

Users cannot view or select available seats in real-time, leading to double bookings or last-minute seat unavailability.

Manual and Time-Consuming Process:

The manual booking process requires physical presence at ticket counters, which is time-consuming and inconvenient for users.

Limited Payment Options:

Many existing systems do not support secure online payments, forcing users to rely on cash transactions, which lack transparency and security.

No Digital Ticketing System:

The absence of digital tickets means users depend on physical tickets, which can be easily lost, damaged, or forgotten.

Minimal Security Measures:

Existing online systems may not have robust authentication methods, increasing the risk of fraudulent bookings and unauthorized access.

Inefficient Database Management:

Data related to bookings, users, and payments may not be stored efficiently, leading to difficulties in data retrieval, updates, and overall management.

Lack of Integration:

Current systems may not integrate essential features such as QR code generation, email confirmations, or automated billing, resulting in a fragmented user experience.

User Experience Issues:

Outdated user interfaces and lack of responsive design make navigation difficult, especially for mobile users.

Fig:2.1 Manual Tickets System

2.2 PROPOSED SYSTEM

The proposed **Bus Booking System** using the **Flask framework** is designed to provide a seamless and user-friendly platform for booking bus tickets online. The system begins with a **user registration** process where new users can sign up by providing their details.

To ensure security, the system incorporates an **OTP verification** step, where a one-time password is sent to the user’s email for authentication. Once verified, users can log in to the system. After logging in, users proceed to the **bus type selection** module, where they can choose from different types of buses.

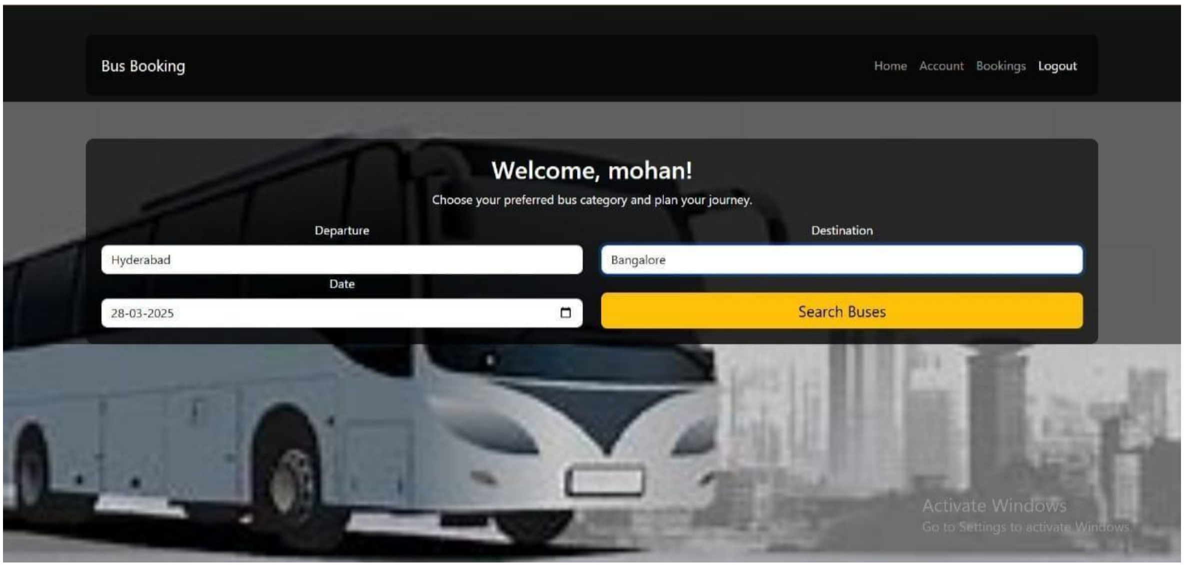
Following the bus type selection, the system allows users to select available seats through an interactive **seat selection** interface that updates seat availability in real time. After selecting seats, the system generates the **user information and bill**, detailing the booking summary and cost.

The user then moves to the **bill payment** section, where secure online payment options, such as credit/debit cards or digital wallets, are available. Upon successful payment, the system generates a **QR code** containing the ticket and booking details. This QR code serves as the digital ticket and can be scanned during boarding.

All user information, bookings, and payment records are securely stored in a **database**, ensuring data integrity and easy retrieval for future reference.

The entire system is structured using Flask's modular architecture, leveraging **Flask-SQL Alchemy** for database management, **Flask-Login** for user authentication, and the **qrcode** library for QR code generation. The user interface is built using **Jinja2 templates**, providing a dynamic and responsive design.

Overall, this proposed system ensures a secure, efficient, and convenient bus booking experience for users.



Welcome, mohan!

Choose your preferred bus category and plan your journey.

Departure: Hyderabad

Destination: Bangalore

Date: 28-03-2025

Fig:2.2 Home page

Advantages

Real-Time Seat Selection:

The system provides an interactive seat selection interface, allowing users to view and select available seats instantly, eliminating the risk of double bookings and enhancing the user experience.

Secure User Authentication:

The inclusion of **OTP verification** during registration and login ensures a secure authentication process, protecting user accounts from unauthorized access.

Digital Ticketing with QR Code Generation:

After successful payment, the system generates a **QR code** containing the booking details. This digital ticket can be easily scanned during boarding, reducing dependency on physical tickets and minimizing the risk of ticket loss or damage.

Multiple Secure Payment Options:

The system integrates reliable payment gateways, enabling users to complete transactions securely using credit/debit cards, net banking, or digital wallets, ensuring transparency and safety during payments.

Efficient Database Management:

Leveraging **Flask-SQL Alchemy**, the system securely stores user data, bookings, and payment history, allowing for efficient data retrieval, updates, and management, thereby improving overall system performance.

2.3 LITERATURE SURVEY

[1] Timetable coordination of first trains in urban railway network: A case study of Beijing

A model of timetable coordination of first trains in urban railway networks, based on the importance of lines and transfer stations, is proposed in this paper. A sub-network connection method is developed, and a mathematical programming solver is utilized to solve the suggested model. A simple test network and a real network of Beijing urban railway network are modeled to verify the effectiveness of our suggested model. Results demonstrate that the proposed model is effective in improving the transfer performance in that they reduce the connection time significantly.

[2] Predicting peak load of bus routes with supply optimization and scaled Shepard interpolation: A newsvendor model

The peak load of a bus route is essential to service frequency determination. From the supply side, there exist ineffective predicted errors of peak load for the optimal number of trips. Whilst many studies were undertaken to model demand prediction and supply optimization separately, little evidence is provided about how the predicted results of peak load affect supply optimization. We propose a prediction model for the peak load of bus routes built upon the idea of newsvendor model, which explicitly combines demand prediction with supply optimization.

A new cost-based indicator is devised built upon the practical implication of peak load on bus schedule. We further devise a scaled Shepard interpolation algorithm to resolve discontinuities in the probability distribution of prediction errors arising from the new indicator, while leveraging the potential efficacy of multi-source data by adding a novel quasi-attention mechanism (i.e., scaling feature space and parameter optimization). The real-world application showed that our method can achieve high stability and accuracy, and is more robust to predicted errors with higher capacity. Our method can also produce a larger number of better trip supply plans as compared to traditional methods, while presenting stronger explanatory power in prioritizing the relative contribution of influential factors to peak load prediction.

[3] Artificial Intelligence in Railway Transport: Taxonomy, Regulations and Applications

Artificial Intelligence (AI) is becoming pervasive in most engineering domains, and railway transport is no exception. However, due to the plethora of different new terms and meanings associated with them, there is a risk that railway practitioners, as several other categories, will get lost in those ambiguities and fuzzy boundaries, and hence fail to catch the real opportunities and potential of machine learning, artificial vision, and big data analytics, just to name a few of the most promising approaches connected to AI. The scope of this paper is to introduce the basic concepts and possible applications of AI to railway academics and practitioners. To that aim, this paper presents a structured taxonomy to guide researchers and practitioners to understand AI techniques, research fields, disciplines, and applications, both in general terms and in close connection with railway applications such as autonomous driving, maintenance, and traffic management. The important aspects of ethics and explainability of AI in railways are also introduced. The connection between AI concepts and railway subdomains has been supported by relevant research addressing existing and planned applications in order to provide some pointers to promising directions.

[4] A review on co-benefits of mass public transportation in climate change mitigation

The magnitude of co-benefits from policy targeting climate change mitigations has been widely promoted due to the desirable win-win results of such policies towards both local and global targets.

This review looks at studies on quantitative environmental and health co-benefits from various modal shifts to public transport scenarios. A systematic review was conducted to evaluate publications from 2004 to August 2015. A total of 153 articles were identified and 9 articles fulfilled all the criteria in this review. Many studies that have been done merely focused on the environmental benefits, especially on reduced air pollution from public transport in cities.

[5] An approach for two-dimensional convolutional neural networks for hourly passenger boarding demand prediction based on uneven smart-card data

An invaluable resource for understanding passenger boarding patterns and forecasting future travel demand is the tap-on smart-card data. Positive instances, on the other hand—

boarding at a given bus stop at a certain time—are less common than negative instances. when looking at the smart-card data (or instances) by boarding stops and by time of day. Machine learning algorithms that are used to estimate hourly boarding numbers at a certain location have been shown to be much less accurate when the data is imbalanced. Before using the smart-card data to forecast bus boarding demand, this research tackles the problem of data imbalance in the data. To create fake traveling instances to add into a synthetic training dataset containing more evenly distributed traveling and non-traveling examples, we suggest using deep generative adversarial networks (Deep-GAN). Next, a deep neural network, or DNN, is trained on the synthetic dataset to predict which instances from a given stop in a certain time frame will travel and which ones won't. According to the findings, resolving the data imbalance problem may greatly enhance the predictive model's functionality and make it more accurate in predicting ridership profiles. The suggested strategy may create a synthetic training set with a better similarity so diversity and, therefore, a stronger prediction capability, according to a comparison of the Deep-GAN's performance with other conventional resampling techniques. The study emphasizes the importance of the issue and offers helpful recommendations for enhancing the quality of the data and model performance for individual travel behavior analysis and travel behavior prediction.

[6] A real-time bus dispatching policy to minimize passenger wait on a high frequency route

One of the greatest problems facing transit agencies that operate high-frequency routes is maintaining stable headways and avoiding bus bunching.

In this work, a real-time holding mechanism is proposed to dispatch buses on a loop-shaped route using real-time information. Holds are applied at one or several control points to minimize passenger waiting time while maintaining the highest possible frequency, i.e. using no buffer time. The bus dispatching problem is formulated as a stochastic decision process. The optimality equations are derived and the optimal holding policy is found by backward induction. A control method that requires much less information and that closely approximates the optimal dispatching policy is found. A simulation assuming stochastic operating conditions and unstable headway dynamics is performed to assess the expected average waiting time of passengers at stations. The proposed control strategy is found to provide lower passenger waiting time and better resiliency than methods used in practice and recommended in the literature.

2.4 FUNCTIONAL REQUIREMENTS

The Functional requirements for a system describe the functionality or the services that the system is expected to provide. These are the statements of services the system should provide and how the system should react to particular inputs and how the system should behave in particular situation.

User Registration: User Register with their Registration details.

User Login: User Login their account using password

Live Inputs: Inputs Given By the User requirement.

Predict Output: Output will be predict based on parameters.

2.5 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements describe the system constraints.

Performance: The application should have better accuracy and should provide prediction in less time.

Scalability: The system must have the potential to be enlarged to accommodate the growth. Capability: The capability of the storage should be high so the large amount of data can be stored in order to train the model.

2.6 FEASIBILITY ANALYSIS

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates.

During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations are involved in the feasibility analysis are:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

2.6.1 economical feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited.

The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.6.2 technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.6.3 social feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

2.7 What is Python?

Below are some facts about Python.

- Python is currently the most widely used multi-purpose, high-level programming language.
- Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.
- Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
- Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python

Let’s see how Python dominates over other languages.

1. Extensive Libraries: Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.
2. Extensible: As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.
3. Embeddable: Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.
4. Improved Productivity: The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.
5. IOT Opportunities: Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.
6. Simple and Easy: When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.
7. Readable: Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory.

8. Object-Oriented: This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.
9. Free and Open-Source: Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.
10. Portable: When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.
11. Interpreted: Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section. Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support. The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

Processor	-	Pentium –IV
RAM	-	4 GB (min)
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

3.2 SOFTWARE REQUIREMENTS

Operating system	**:**	Windows 7 Ultimate.
Coding Language	**:**	Python.
Front-End	**:**	Python.
Back-End	**:**	Flask
Designing	**:**	Html, CSS, JavaScript.
Data Base	**:**	MySQL (WAMP Server).

Install Python Step-by-Step in Windows and Mac

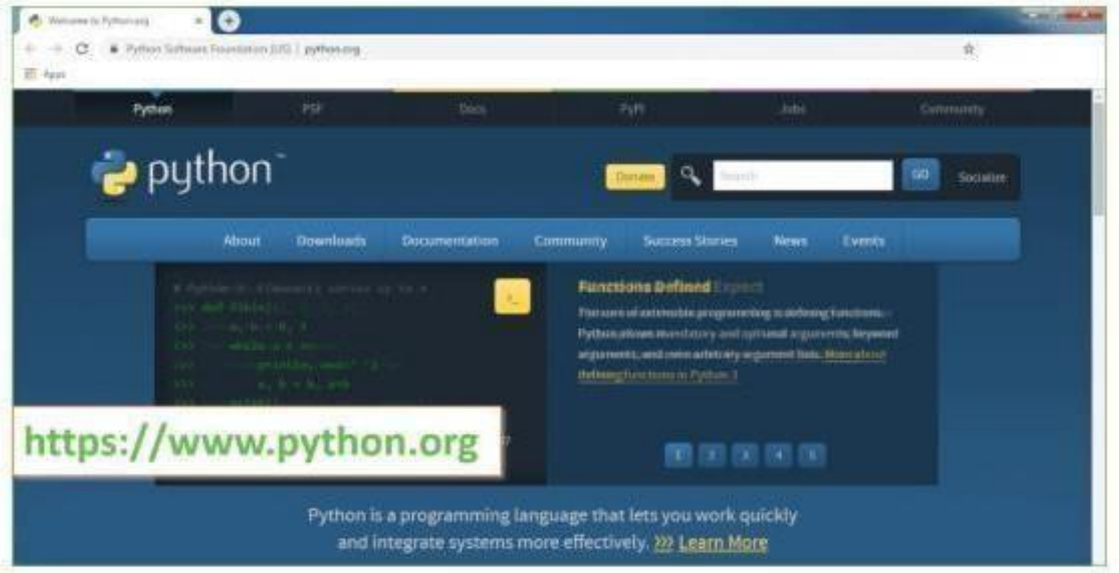
Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e., operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So, the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

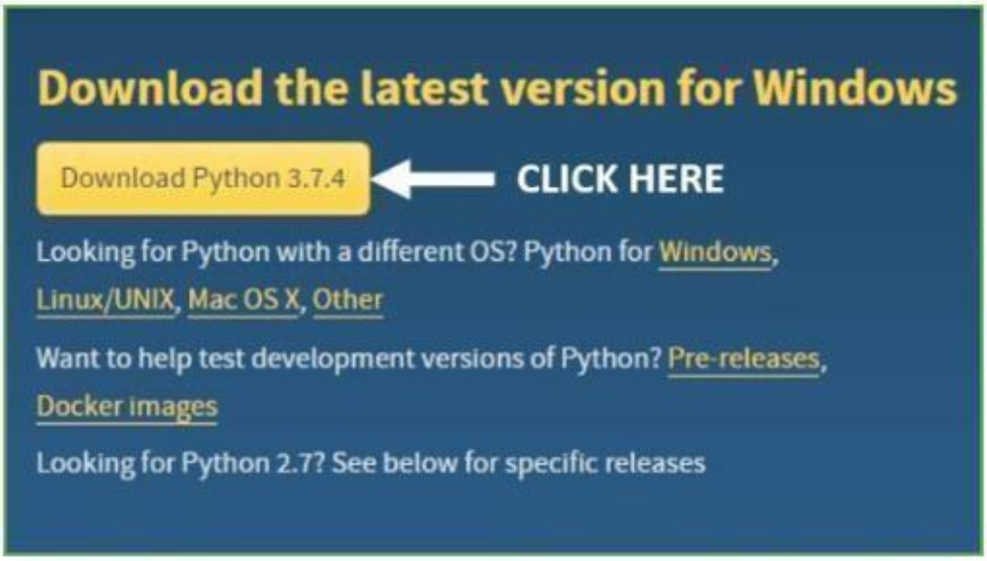
Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org



Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Download the latest version for Windows

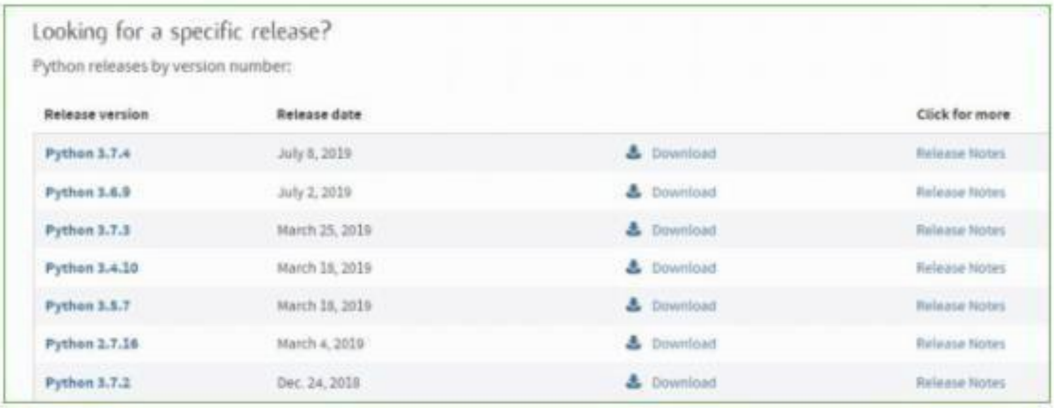
Download Python 3.7.4 ← CLICK HERE

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4.



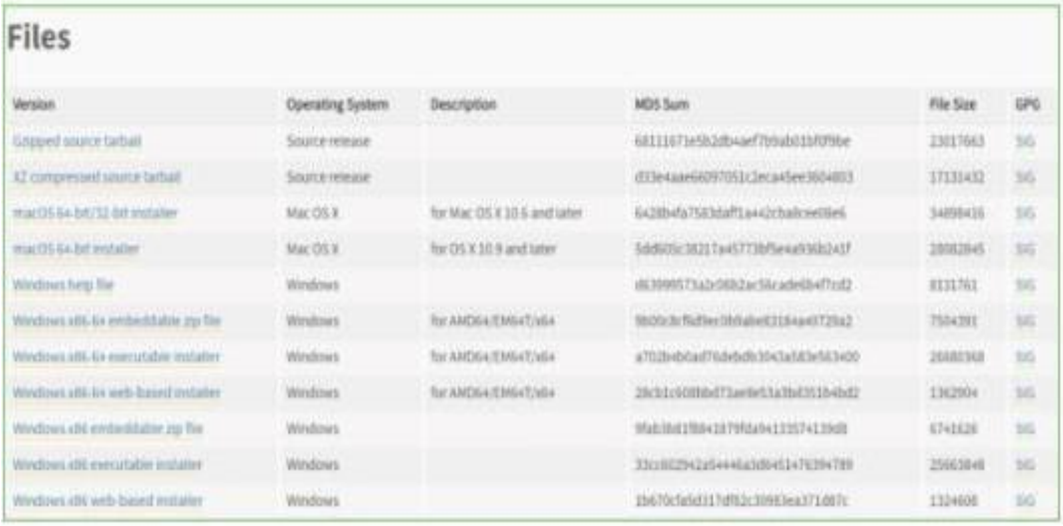
Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c06b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options:

Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows

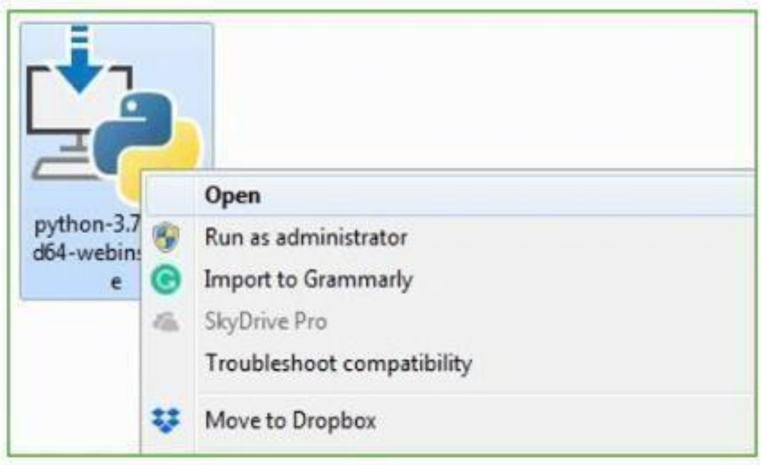
- x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation

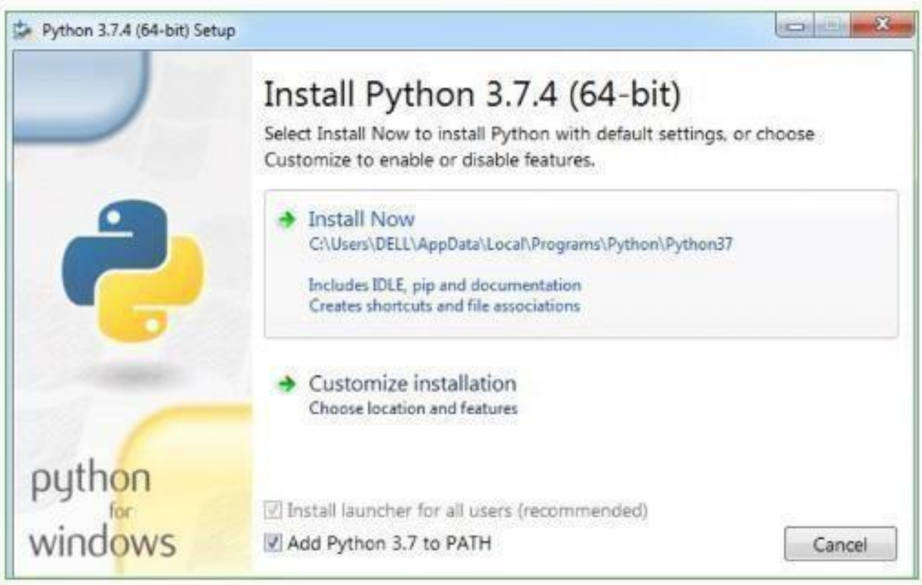
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

3.3 INSTALLATION OF PYTHON

Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



Step 2: Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

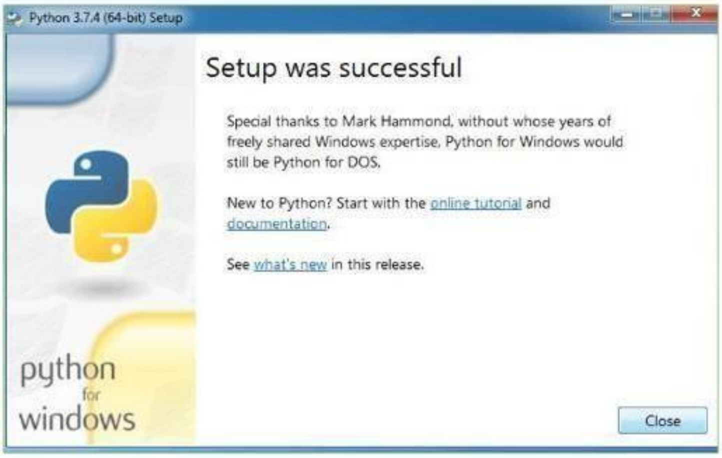
Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Step 3: Click on Install NOW After the installation is successful. Click on Close.



Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

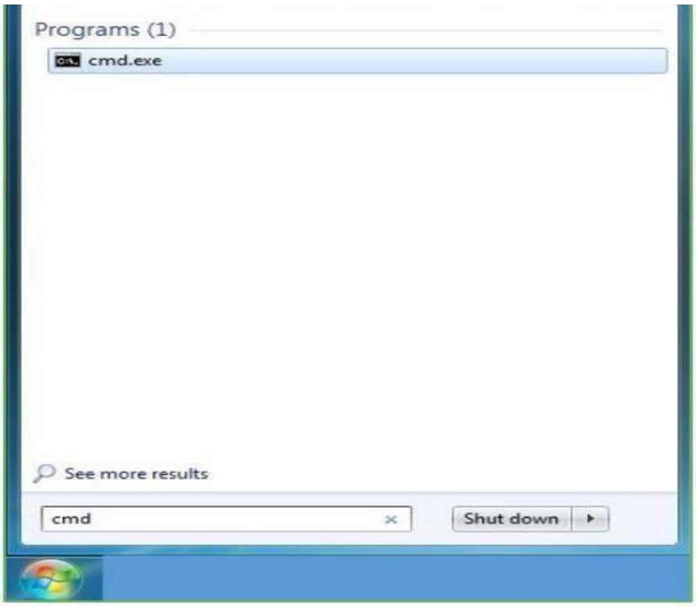
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the Python Installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type “cmd”.



Programs (1)

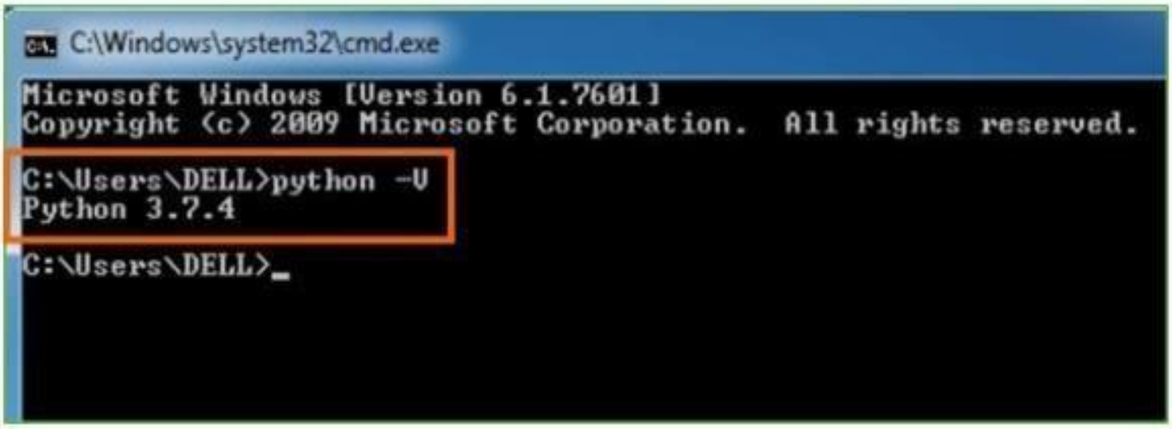
cmd.exe

See more results

cmd

Step 3: Open the Command prompt option

Step 4: Let us test whether the python is correctly installed. Type python –V and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

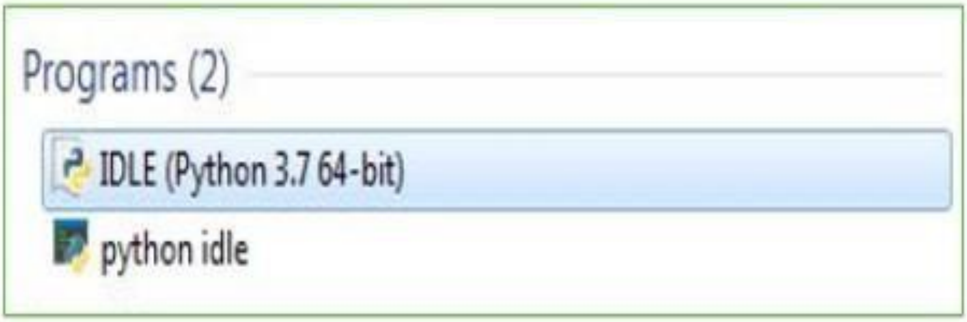
Step 5: You will get the answer as 3.7.

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

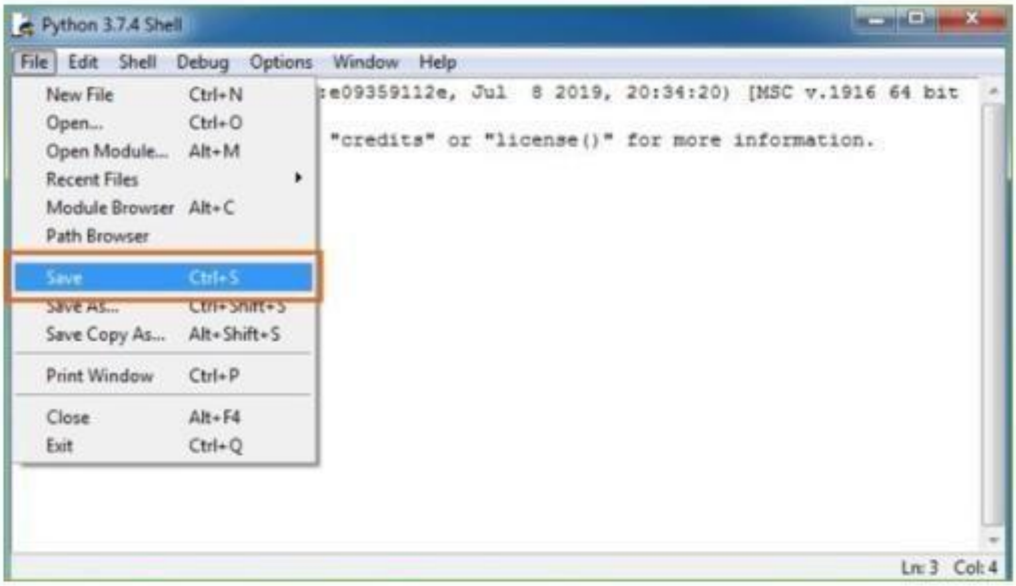
Step 1: Click on Start

Step 2: In the Windows Run command, type “python idle”.



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

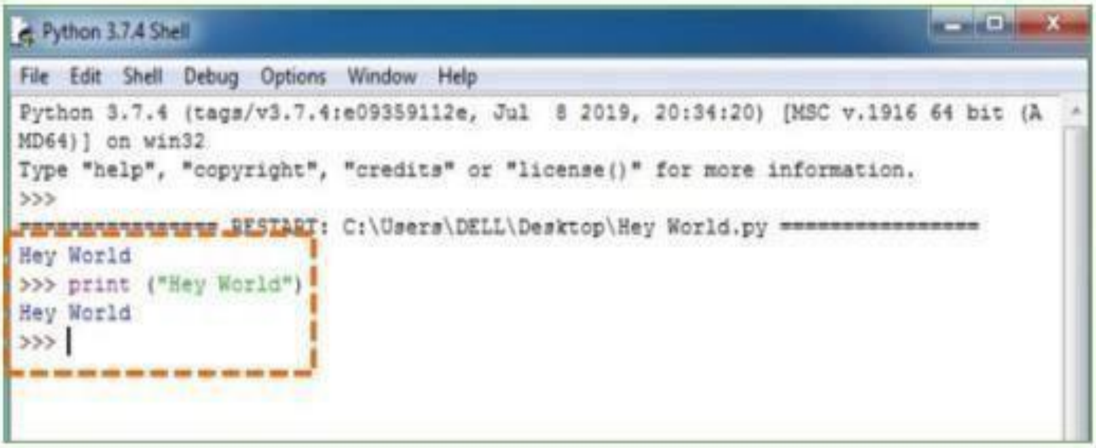
Step 4: To go ahead with working in IDLE you must first save the file. Click on File Click on Save.



```
e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
"credits" or "license()" for more information.
```

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g., enter print ("Hey World") and Press Enter.



```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit (A
MD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=============== RESTART: C:\Users\DELL\Desktop\Hey World.py ================
Hey World
>>> print ("Hey World")
Hey World
>>> |
```

You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python does not need semicolons at the end of the statements otherwise it won't work.

3.4 FLASK WEB FRAMEWORK

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Unlike the Django framework, Flask is very Pythonic. It's easy to get started with Flask, because it doesn't have a huge learning curve.

On top of that it's very explicit, which increases readability.

It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

3.5.1 RESEARCH

The development of the Smart Bus Booking System required extensive research to ensure efficiency, security, and user-friendliness. The research phase involved analyzing existing bus booking platforms, identifying common challenges faced by users, and exploring technologies that could enhance the booking experience.

Key areas of research included:

- **User Requirements Analysis**: Understanding the needs of passengers and conductors to design an intuitive and accessible interface.
- **Technology Stack Selection**: Evaluating different frontend and backend frameworks to ensure scalability and performance. React Native was chosen for cross-platform development, while Node.js and MongoDB were selected for backend and database management.
- **QR Code Integration**: Studying secure methods for generating and validating QR codes to facilitate seamless ticket verification by conductors.
- **Secure Payment Processing**: Investigating secure transaction methods, including Aadhaar-based authentication and bank integration, to enhance trust and compliance with financial regulations.

- **Seat Availability and Booking Optimization**: Exploring algorithms for real-time seat availability updates and efficient booking processes to minimize errors and double bookings.
- **Privacy and Security Measures**: Ensuring data encryption, user authentication, and compliance with privacy policies to protect user information.

This research phase played a crucial role in shaping the system's architecture, improving user experience, and ensuring the overall reliability of the Smart Bus Booking System.

3.5.2 researches on qr-code

QR -Code has been used widely nowadays since it improves the need of mobility of other materials such as business card, flyers and pamphlets. As the usage is spreading fast, the essentials of scanning QR-Code has been so much widely implemented in most of the devices used today especially mobile phones. Therefore, I choose to implement and use this technology in my project to increase the values and to keep up with latest techs used.

QR-Code

Information contains by their very nature, QR codes (and other data matrix codes) are meant to be read by machines, not humans, so there's only a certain amount we can tell just by looking at them. Although each code is different, they contain a few interesting, common features.

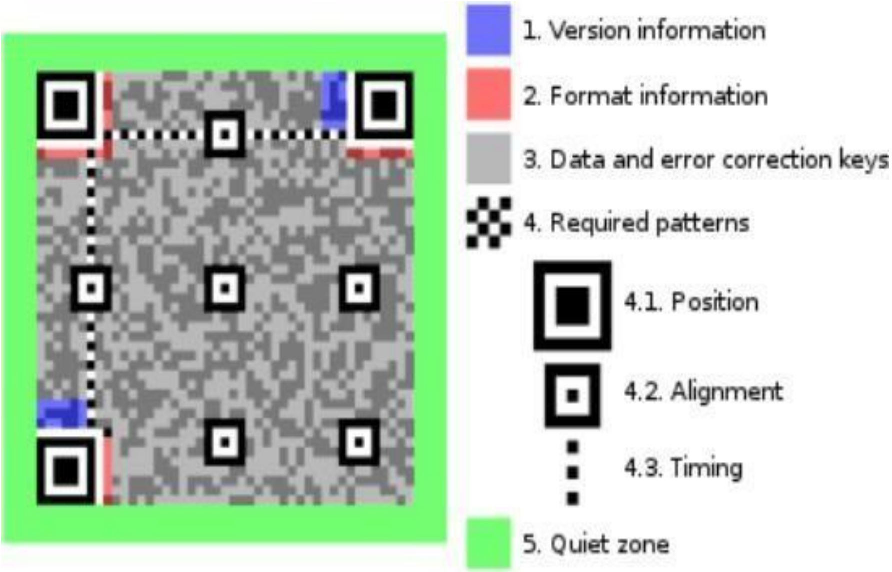


Fig: 3.1 QR code

Application for QR-Codes

While QR codes gained recognition due to their increasingly widespread use in marketing and consumer-facing applications, they can also be useful in industrial applications, such as:

1. Operational Instructions

QR codes can be used to convey operating instructions, procedures, and other information necessary for operating heavy equipment.

2. Facilities management

They can be used to document schematics and other instructions for plumbing, wiring systems, and alarm systems, providing an easy way to communicate these details to contractors or maintenance workers.

4. Maintenance and repairs

QR codes may be used to submit requests for maintenance service or as a way to easily document that routine maintenance has been performed, creating a complete audit trail of service and repair records.

5. Regulatory compliance

In industrial applications, equipment and machinery often requires periodic inspection, regular maintenance, and permits or licenses to comply with regulatory requirements. QR codes can be utilized to store this information and make it readily accessible

Types of QR-Code

1. QR Code Model 1 and Model 2

i. QR Code Model 1:

The original QR Code, a code capable of coding 1,167 numerals with its maximum version being 14 (73 x 73 modules).

QR Code Model 1

Fig: 3.2 QR code model 1

ii. QR Code Model 2:

QR Code created by improving Model 1 so that this code can be read smoothly even if it is distorted in some way. QR Codes that are printed on a curved surface or whose reading images are distorted due to the reading angle can be read efficiently by referring to an alignment pattern embedded in them. This code can encode upto7,089 numerals with its maximum version being 40(177 x 177 modules).

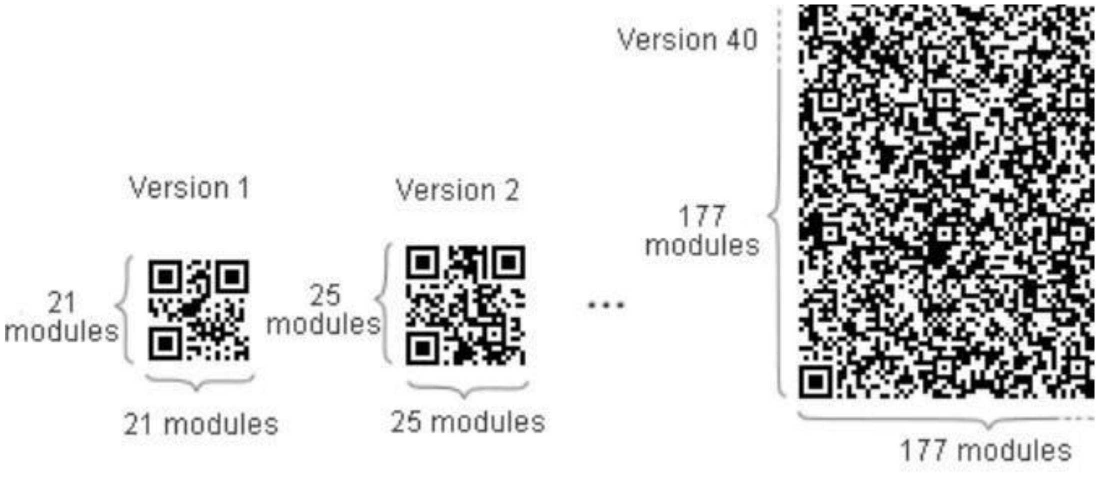


Fig: 3.3 QR code model 2

2. Micro QR Code

A major feature of Micro QR Code is it has only one position detection pattern, compared with regular 21 QR Code that require a certain amount of area because position detection patterns are located at the three corners of a symbol. Furthermore, QR Code requires at least a four-module wide margin around a symbol, whereas a two-module wide margin is enough for Micro QR Code. This configuration of Micro QR Code allows printing in areas even smaller than QR Code.

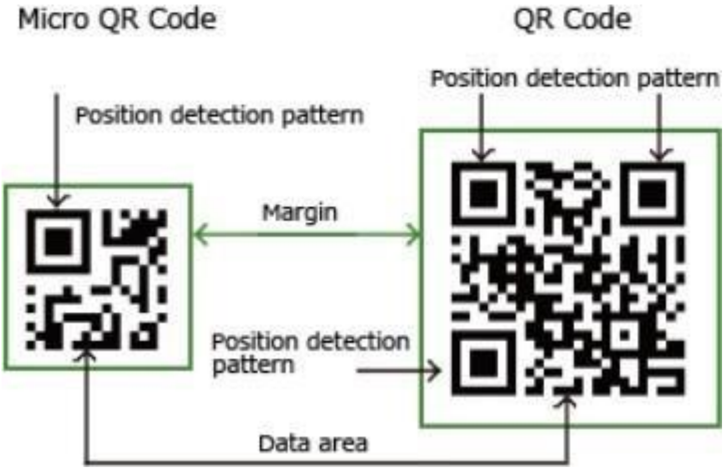

Micro QR Code
QR Code
Position detection pattern
Position detection pattern
Margin
Position detection pattern
Data area

Fig: 3.4 Micro QR code

3. iQR Code

iQR Code is a matrix-type 2D code allowing easy reading of its position and size. This code allows a wide size range of codes from ones smaller than the traditional QR Code and Micro QR Code to large ones that can store more data than these. This code can be printed as a rectangular code, turned-over code, black-and-white inversion code or dot pattern code (direct part marking) as well, allowing a wide range of applications in various areas.

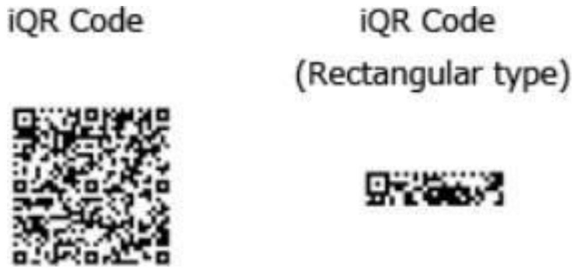


Fig: 3.5 iQR code

4. sQR Code

A single QR Code can carry public data and private data. The private data can be read only with a dedicated reader having the cryptographic key, which provides data protection. Since SQRC looks exactly the same as the regular QR Code, it can prevent forgery and tampering.

Fig: 3.6 sQR code

5. Frame QR Code

This code has an area, or a frame, for holding an image. Since the shape and color of the frame can be changed flexibly, the code has a variety of applications

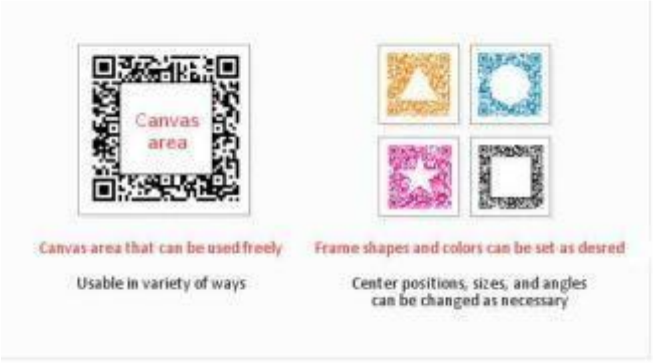

Canvas
area
Canvas area that can be used freely
Usable in variety of ways
Frame shapes and colors can be set as desred
Center positions, sizes, and angles
can be changed as necessary

Fig: 3.7 Frame QR code

Advantages of QR-Code

The QR code has many advantages over a conventional barcode, however. Them a in advantage is that you can store up to a hundred times more information on a QR code than on a conventional horizontal barcode. In addition, QR codes can be scanned from any direction for 360 degrees. This makes them easier for your device to read and lessens the possibility of background interference.

The third main advantage is that from a marketing point of view, the code's appearance is unique and interesting, increasing the likelihood of engaging the customer in any campaign where it might be deployed. 15 A QR code reader can be downloaded onto a smartphone by anyone, and they are mostly free of charge. This means that any customer can walk into your business with his or her smartphone, and scan a QR code which you have generated. An Android user can use something like QR Code Reader, and an iPhone user can download the Quick Scan app. Both are free of charge.

CHAPTER 4

SOFTWARE DESIGN

4.1 UNIFIED MODELLING LANGUAGE DIAGRAMS

UML is a method for describing the system architecture in detail using the blue print. UML represents a collection of best engineering practice that has proven successful in the modelling of large and complex systems. The UML is very important parts of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the helps UML helps project teams communicate explore potential designs and validate the architectural design of the software.

4.1.1 use case diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

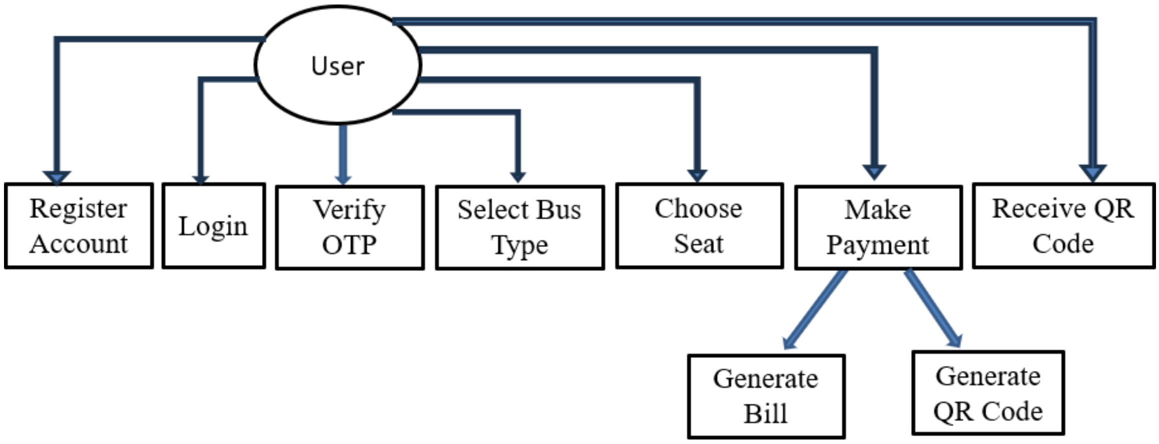


Fig: 4.1 Use Case Diagram

4.1.2 flow chart

Activity diagrams are graphical representations of work flows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step work flows of components in a system. An activity diagram shows the overall flow of control.

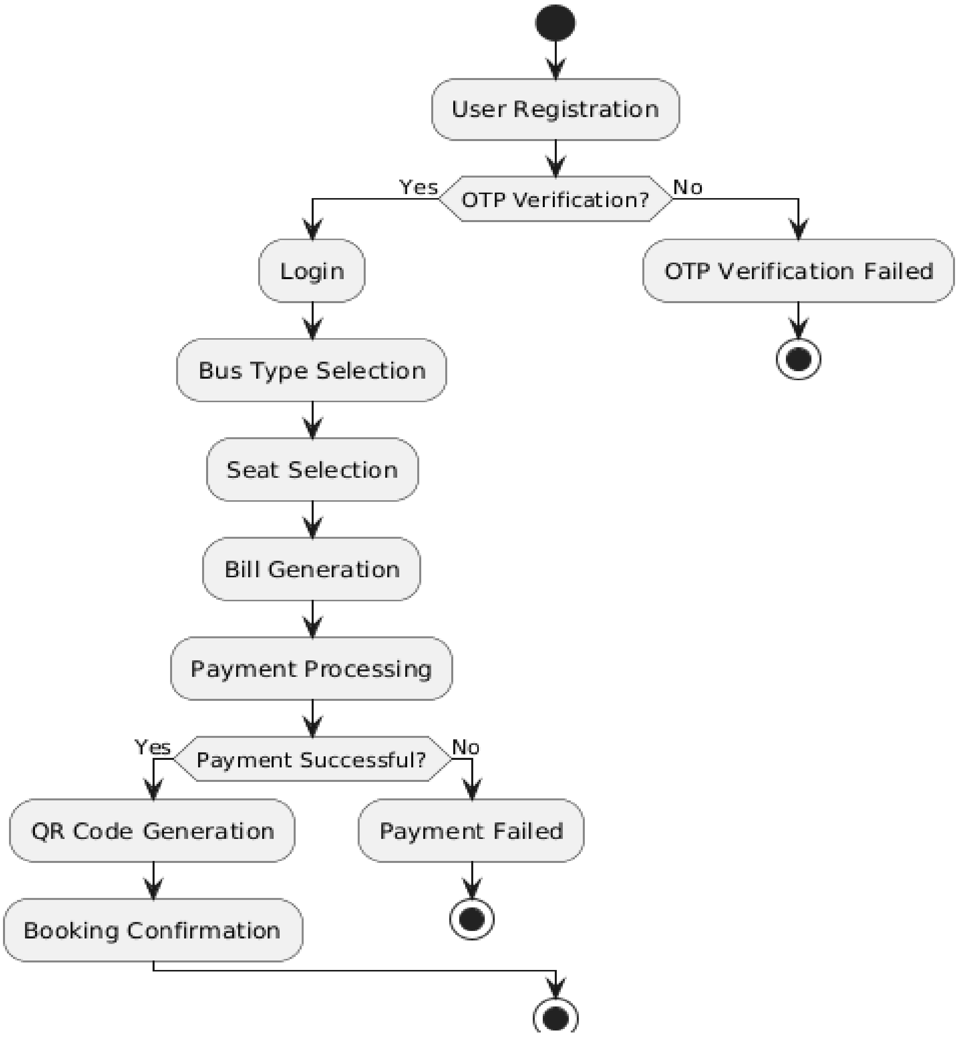


Fig: 4.2 Flowchart

4.1.3 sequence diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagram

The proposed Bus Booking System using Flask offers a secure, efficient, and user-centric platform for online bus ticket booking. By following this step-by-step methodology, the system ensures real-time seat booking, secure payments, OTP-based user verification, and digital QR code ticketing, addressing the limitations of existing systems and providing a scalable solution for future enhancements.

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

A class diagram provides a clear representation of the system's objects, their attributes, behaviors, and relationships, making it an essential tool for object-oriented design and development. By visualizing the interactions between different classes, developers can better understand the data flow, dependencies, and constraints within the system. This structured approach facilitates efficient coding, debugging, and system expansion while ensuring that design principles such as encapsulation, inheritance, and polymorphism are effectively implemented.

4.1.4 class diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

A class diagram provides a clear representation of the system's objects, their attributes, behaviors, and relationships, making it an essential tool for object-oriented design and development. By visualizing the interactions between different classes, developers can better understand the data flow, dependencies, and constraints within the system. This structured approach facilitates efficient coding, debugging, and system expansion while ensuring that design principles such as encapsulation, inheritance, and polymorphism are effectively implemented.

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

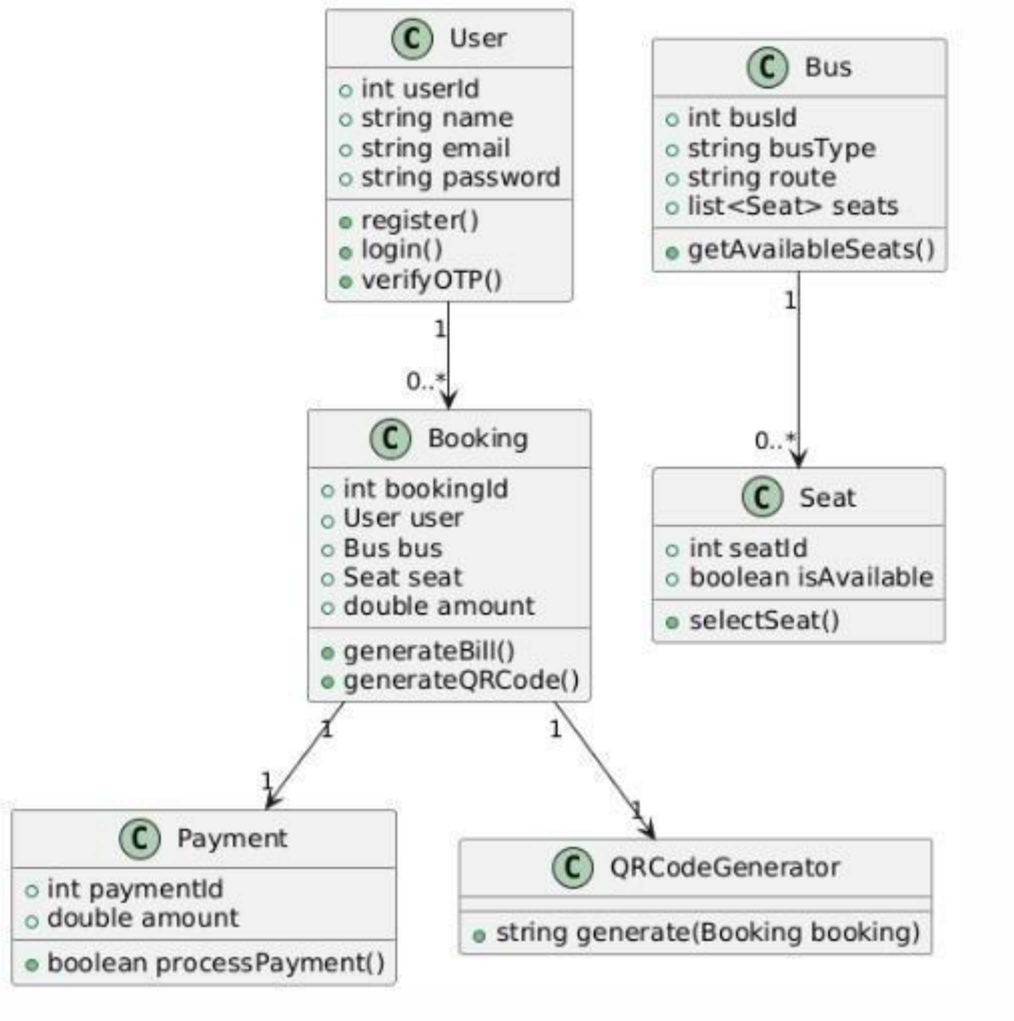


Fig: 4.4 Class diagram

4.2 Methodology

The methodology for developing the Bus Booking System using the Flask framework involves a structured step-by-step approach. This approach ensures that all functional requirements are met while maintaining scalability, security, and user-friendliness. The methodology covers the planning, design, implementation, testing, and deployment phases.

Step 1: Requirement Analysis

The first step is understanding and gathering the complete requirements of the system.

Key Requirements:

- User registration and secure authentication (with OTP verification).
- Selection of bus types Rajadhani or delxue.
- Real-time seat selection and availability checking.
- Bill generation based on user selections.
- Secure payment gateway integration.

- QR code generation for digital ticketing.
- Efficient database management for user, booking, and payment records.
- User-friendly and responsive interface.

Step 2: System Design

2.1 Architecture Design:

- Use Flask as the backend framework for its lightweight nature and modular structure.
- MVC Pattern (Model-View-Controller) for separation of concerns.
    - Model: Handles database operations using Flask-SQLAlchemy.
    - View: Renders HTML templates using Jinja2.
    - Controller: Manages the application logic and routing.

Step 3: Development Process

3.1 Setting up the Flask Application

Create a virtual environment and install dependencies (Flask, Flask-SQLAlchemy, Flask-WTF, Flask-Login, qrcode, smtplib).

Structure the Flask app into modules:

```
/bus_booking_system/
├── app.py
├── /templates/
├── /static/
├── /models/
├── /routes/
├── /utils/
└── requirements.txt
```

3.2 User Authentication & OTP Verification

- User Registration: Secure registration with hashed passwords.
- OTP Verification: Use smtplib to send OTP to the user's email.
- Login Management: Use Flask-Login for user session management.

3.3 Bus and Seat Selection

- Bus Type Selection: Fetch available bus types and schedules from the database.
- Seat Selection: Display an interactive seat map, updating real-time seat availability.

3.4 Bill Generation and Payment Integration

- QR Code Generation
- Display and email the QR code to the user as a digital ticket.

Deployment

Deployment to Production Server

- Deploy the application using a reliable web server

Key Technologies Used:

- Backend: Python, Flask
- Frontend: HTML5, CSS3, JavaScript (with Jinja2 for templating)
- Database: SQLite/MySQL (using Flask-SQLAlchemy ORM)
- Authentication: Flask-Login, OTP via smtplib
- QR Generation: Python qrcode library

The proposed Bus Booking System using Flask offers a secure, efficient, and user-centric platform for online bus ticket booking. By following this step-by-step methodology, the system ensures real-time seat booking, secure payments, OTP-based user verification, and digital QR code ticketing, addressing the limitations of existing systems and providing a scalable solution for future enhancements.

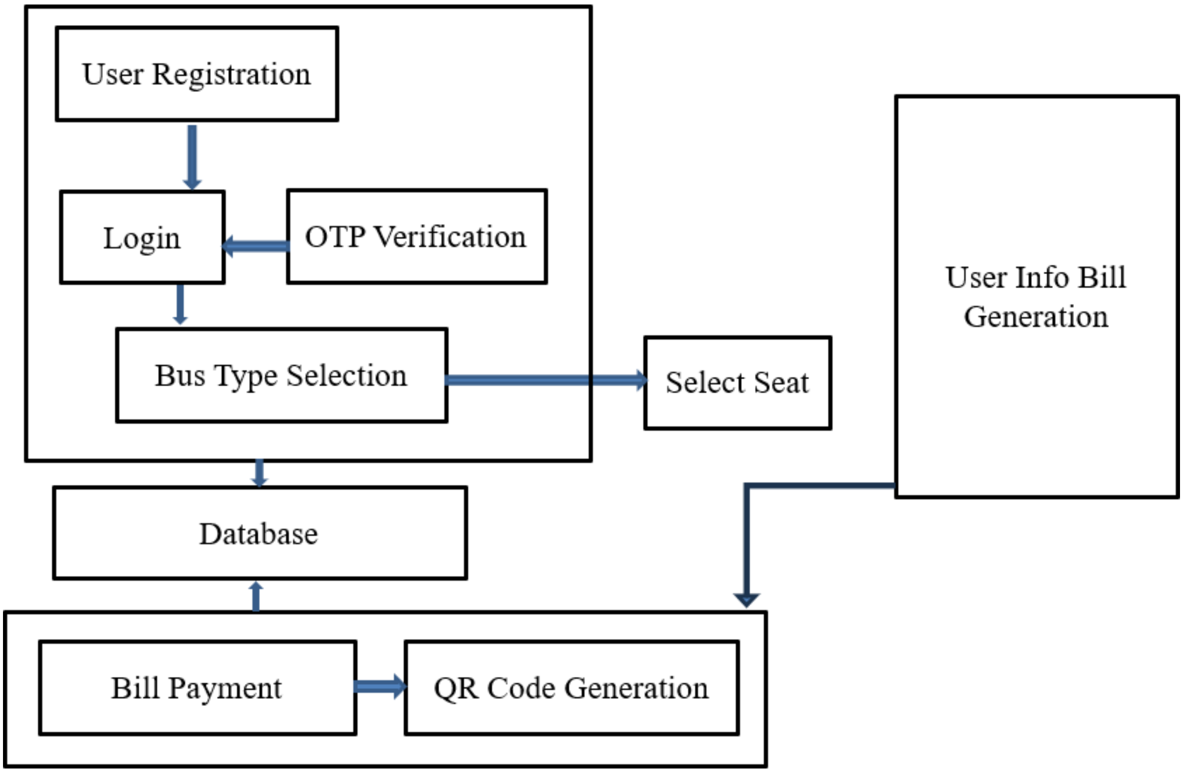


Fig: 4.5 Architecture Diagram

4.3 Module

The **Bus Booking System** is divided into various functional modules, each responsible for handling specific tasks. This modular approach ensures easier development, testing, maintenance, and future scalability. Below is a detailed explanation of each core module:

1. User Management Module

- **User Registration:**
    - Users can create an account by providing details like name, email, and password.
    - Passwords are securely stored using hashing algorithms (e.g., **bcrypt**).
- **OTP Verification:**
    - Sends an OTP to the registered email for user verification using smtplib.
    - Verifies OTP input before allowing login access.
- **Login and Authentication:**
    - Uses **Flask-Login** to manage user sessions.
    - Provides login/logout functionalities with proper authentication flow.

2. Bus Management Module

- **Bus Listing:**
    - Displays all available buses with filters based on bus type (AC, Non-AC, Sleeper), timing, and routes.
- **Bus Details:**
    - Shows detailed information like departure time, arrival time, and available seats.
- **Dynamic Schedule Management:**
    - Allows admins to add, update, or remove bus schedules.

3. Seat Management Module

- **Interactive Seat Map:**
    - Displays seat layouts with availability status.
- **Real-Time Seat Selection:**
    - Users can select multiple seats.
    - Once selected, the seat status updates instantly to prevent double bookings.
- **Seat Reservation Logic:**
    - Holds seat selection temporarily during the payment process to avoid conflicts.

4. Booking Management Module

- **Booking Confirmation:**
    - Creates a booking record after successful seat selection and payment.

- **Fare Calculation:**
    - Calculates the total fare based on the selected bus type, route, and number of seats.
- **Ticket Details:**
    - Displays complete booking information along with a unique booking ID.

5. QR Code Generation Module

- **QR Code Creation:**
    - Uses the Python **qrcode** library to generate codes with embedded booking details.
- **Digital Ticket:**
    - The QR code can be scanned at boarding points for quick verification.
- **Email Delivery:**
    - The generated QR code is emailed to the user along with the ticket details.

6. Billing and Invoice Module

- **Automated Billing:**
    - Generates detailed bills based on fare, taxes, and other charges.
- **Invoice Download:**
    - Users can download or print the invoice from their dashboard.
- **Email Integration:**
    - Sends billing details automatically after successful booking.

7. Admin Management Module

- **Bus Schedule Management:**
    - Admin can add or remove bus routes, modify timings, and update seat capacities.
- **User Management:**
    - View user activity and manage registered users.
- **Booking Oversight:**
    - Access and review all booking details.
- **Report Generation:**
    - Generate reports related to revenue, user engagement, and seat occupancy trends.

8. Notification and Communication Module

- **Email Notifications:**
    - Sends confirmation emails for registration, booking, payment, and ticket details.
- **OTP Delivery:**
    - Sends OTPs during user authentication for secure access.

- **Alerts and Reminders:**
    - Sends reminders for upcoming trips and promotional offers.

9. **Billing and Invoice Module**

- **Automated Billing:**
    - Generates detailed bills based on fare, taxes, and other charges.
        - **Invoice Download:**
    - Users can download or print the invoice from their dashboard.
        - **Email Integration:**
    - Sends billing details automatically after successful booking.

CHAPTER 5
WORKING AND TESTING

5.1 WORKING

The Bus Booking System is a robust, user-friendly application designed to facilitate seamless bus ticket reservations. It leverages Aadhaar-based authentication to ensure secure login and user verification. This authentication method reduces the risk of fraudulent activities and ensures that only genuine users access the platform.

Once logged in, users are greeted with an intuitive interface where they can search for buses by entering the source, destination, and travel date. The system displays a list of available buses with their schedules, fares, and seat availability. Real-time data management ensures that seat information remains accurate, minimizing instances of double bookings.

User Journey

1. **Registration and Login:**
    - Users register using their Aadhaar number for verification.
    - Secure OTP-based verification ensures user authenticity.
    - Registered users can log in using their Aadhaar credentials.
2. **Bus Search and Selection:**
    - Users input their preferred source, destination, and date of travel.
    - The system displays available buses along with details like departure time, fare, and available seats.
    - Users can apply filters based on bus type, operator, and ticket price.
3. **Seat Selection:**
    - A seat layout is provided, allowing users to select available seats.
    - The fare is dynamically calculated based on the number of seats selected and applicable service fees.
4. **Payment Process:**
    - Users can proceed to the payment gateway for secure transactions.
    - Various payment methods, including linked bank accounts, credit/debit cards, and UPI, are supported.
    - The system verifies the payment and updates the seat status in real-time.

5. **QR Code Generation:**
    - Upon successful booking, a unique QR code is generated for the user.
    - The QR code serves as a digital ticket and is accessible through the user's profile.

6. **Conductor Verification:**
    - Conductors use a dedicated mobile application for QR code scanning.
    - The app validates the QR code against the central database, ensuring the ticket's authenticity.
    - Upon successful verification, passengers are allowed to board the bus. The Bus Booking System efficiently streamlines the ticket booking process while enhancing the overall user experience with its secure, reliable, and user-friendly functionality.
    - From seamless seat selection and secure payments to QR code-based ticket verification, the system covers every aspect of the bus reservation lifecycle.
    - Real-time information on bus availability, dynamic fare calculations, and transparent booking management ensure users can make informed decisions.
    - Additionally, Aadhaar-based authentication enhances security, providing a safe and trustworthy platform for users.
    - Conductors benefit from the streamlined ticket validation process, reducing the chances of fraud.

    - Upon successful verification, passengers are allowed to board the bus. The Bus Booking System efficiently streamlines the ticket booking process while enhancing the overall user experience with its secure, reliable, and user-friendly functionality.
    - From seamless seat selection and secure payments to QR code-based ticket verification, the system covers every aspect of the bus reservation lifecycle.
    - Real-time information on bus availability, dynamic fare calculations, and transparent booking management ensure users can make informed decisions.
    - Additionally, Aadhaar-based authentication enhances security, providing a safe and trustworthy platform for users.

Additional Features

- **Booking History Management:** Users can view their booking history, including past and upcoming trips. The system provides detailed information like bus details, payment confirmation, and QR codes.
- **Cancellation and Refunds:** Users can cancel their bookings through the app, and refunds are processed as per the cancellation policy.
- **Notifications and Alerts:** Users receive real-time notifications for booking confirmations, cancellations, and payment updates.
- **Support and Feedback:** The system includes a support section for users to raise complaints or provide feedback.

Backend and Scalability

The backend is powered by **Flask** to ensure seamless API management. Data is stored securely using **MongoDB**, which offers efficient data retrieval and management. The system is designed with scalability in mind, capable of handling a large number of concurrent users without performance degradation. Real-time updates ensure that seat availability and payment status are reflected accurately across all users.

Security Features

- **Aadhaar-based Authentication:** Ensures secure user verification.
- **Data Encryption:** User data, including payment information, is encrypted using AES encryption standards.
- **Payment Gateway Integration:** Secure transactions are facilitated using trusted third-party payment gateways.
- **Role-based Access Control:** Separate access for users and conductors ensures data privacy and system integrity.

The Bus Booking System not only streamlines the ticket booking process but also enhances user experience through its secure, reliable, and efficient functionality. From seamless seat selection and secure payments to QR code-based verification, the system covers all aspects of the bus reservation lifecycle. Future improvements may include predictive analytics for better bus recommendations, expanded payment options, and multilingual support for broader accessibility.

selection and secure payments to QR code-based verification, the system covers all aspects of the bus reservation lifecycle. Future improvements may include predictive analytics for better bus recommendations, expanded payment options, and multilingual support for broader accessibility.

The Bus Booking System efficiently streamlines the ticket booking process while enhancing the overall user experience with its secure, reliable, and user-friendly functionality. From seamless seat selection and secure payments to QR code-based ticket verification, the system covers every aspect of the bus reservation lifecycle. Real-time information on bus availability, dynamic fare calculations, and transparent booking management ensure users can make informed decisions. Additionally, Aadhaar-based authentication enhances security, providing a safe and trustworthy platform for users. Conductors benefit from the streamlined ticket validation process, reducing the chances of fraud.

Future improvements for the system may include integrating predictive analytics to recommend buses based on user preferences, previous bookings, and demand patterns. Expanding payment options to include digital wallets, UPI, and international cards would further increase accessibility. Multilingual support can make the platform inclusive, catering to users from diverse backgrounds. Additional features like real-time bus tracking, automated notifications, and a robust feedback system could further optimize the user experience. With these enhancements, the Bus Booking System can continue to revolutionize the public transportation sector, making bus travel more convenient, secure, and efficient.

5.2 TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

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 cantered 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. Itis 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.

CHAPER 6

RESULTS

Step1: Home Page

- On the homepage of the website, we have a user login and an admin login.

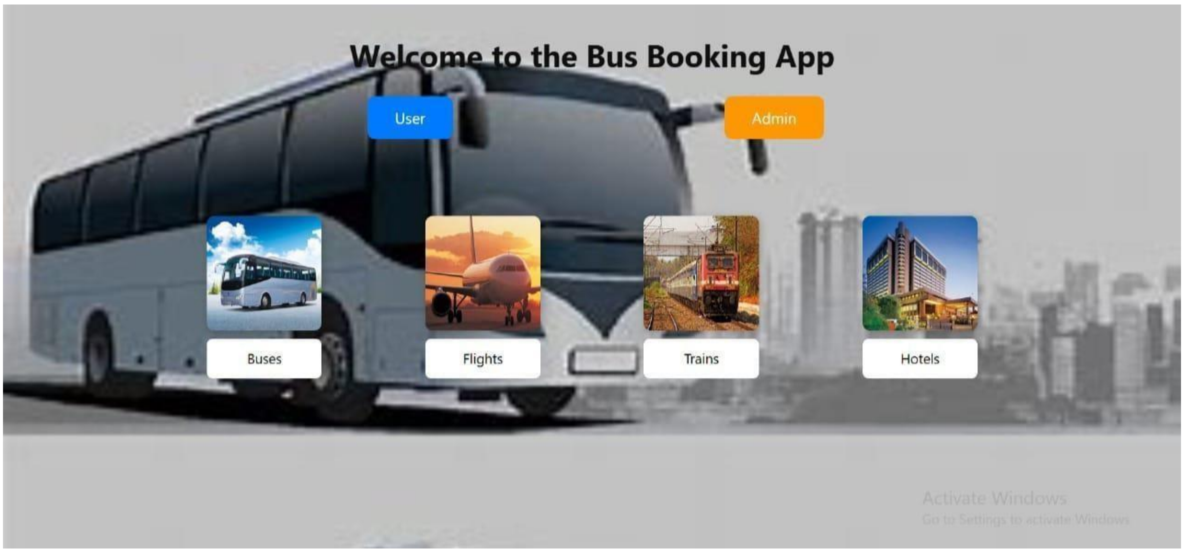
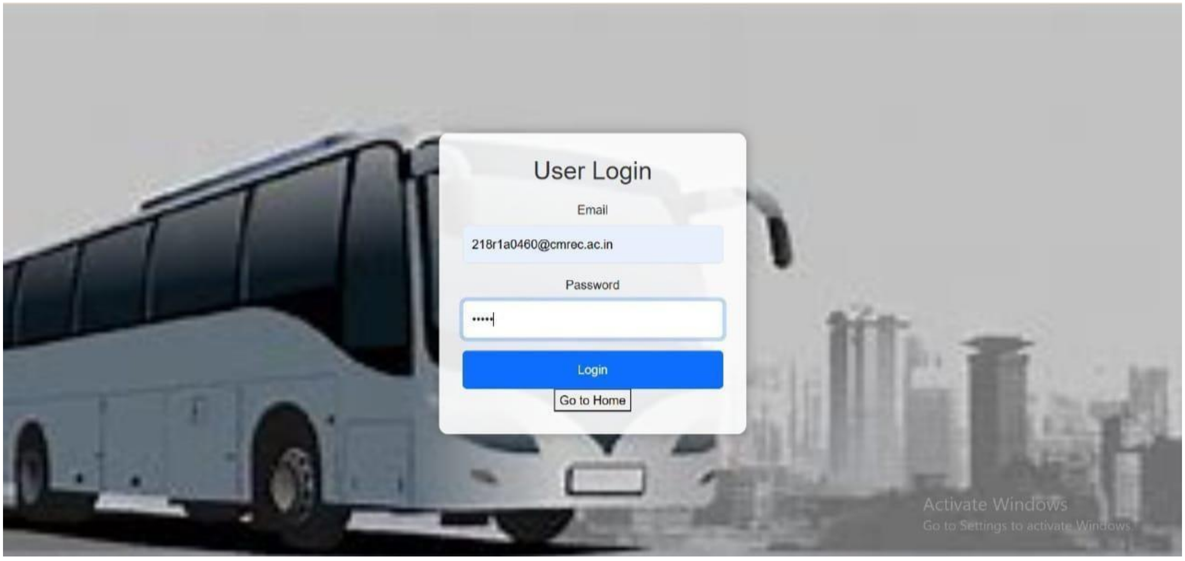

Welcome to the Bus Booking App
User
Admin
Buses
Flights
Trains
Hotels
Activate Windows
Go to Settings to activate Windows

Fig: 6.1 Home Page

Step2: user login

- This is the login page of the website.
- Here, we have fields for email and password.



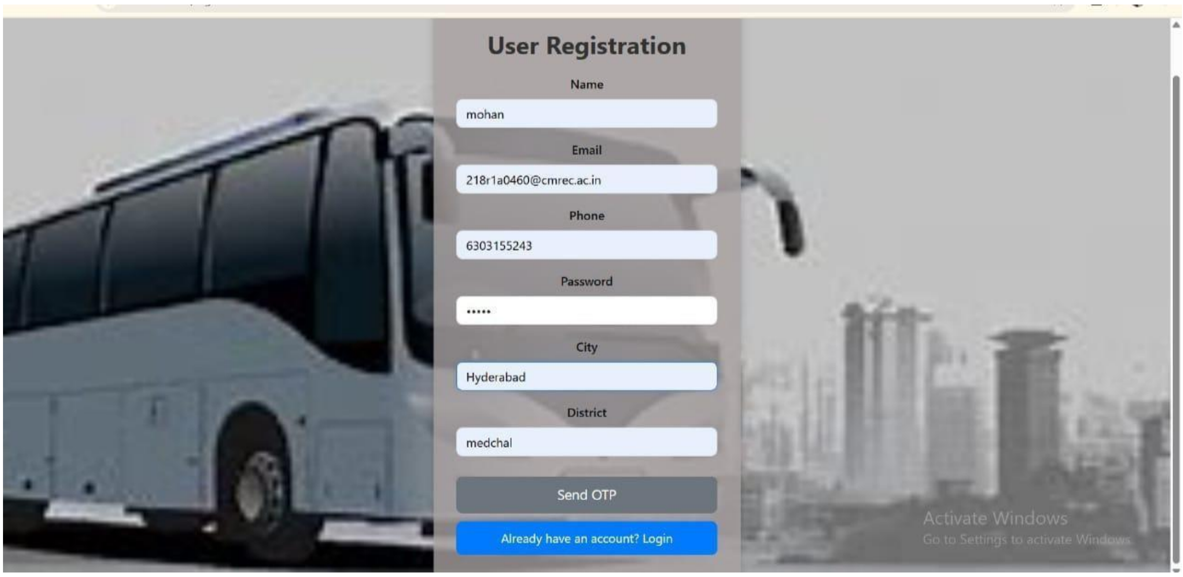
Email: 218r1a0460@cmrec.ac.in

Password: •••••

Fig: 6.2 Login page

Step3: Register as a new user

- Register as a new user: Name, Email, Phone, Password, City, District.



User Registration

Name: mohan

Email: 218r1a0460@cmrec.ac.in

Phone: 6303155243

Password: •••••

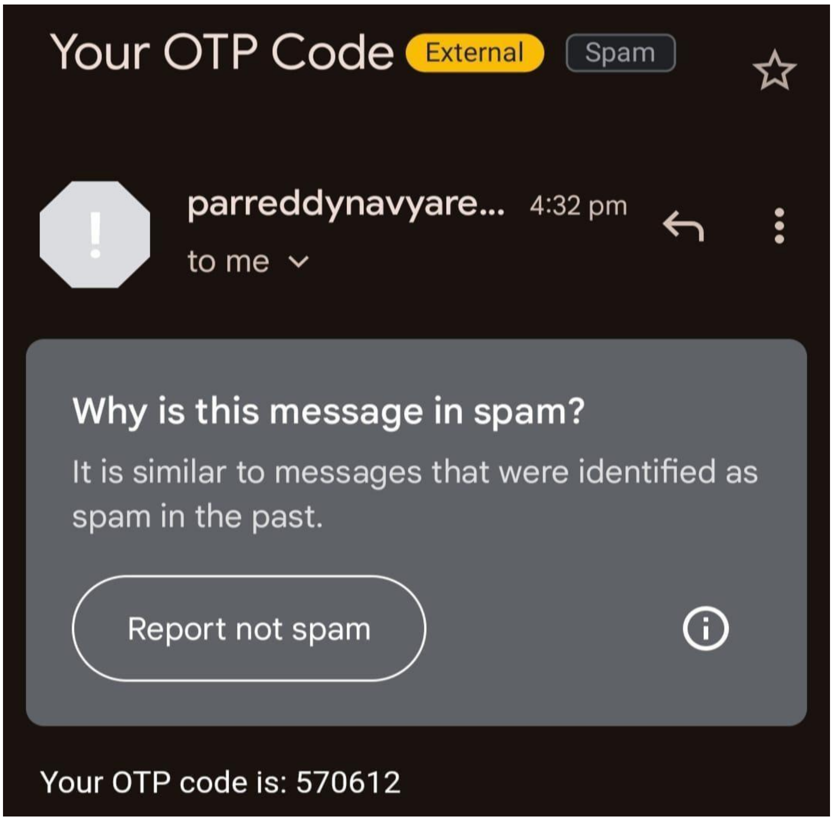
City: Hyderabad

District: medchal

Send OTP
Already have an account? Login
Activate Windows
Go to Settings to activate Windows.

Fig: 6.3 Registration page

- OTP verification through email



Your OTP Code

parreddynavyare... 4:32 pm

to me

Why is this message in spam?

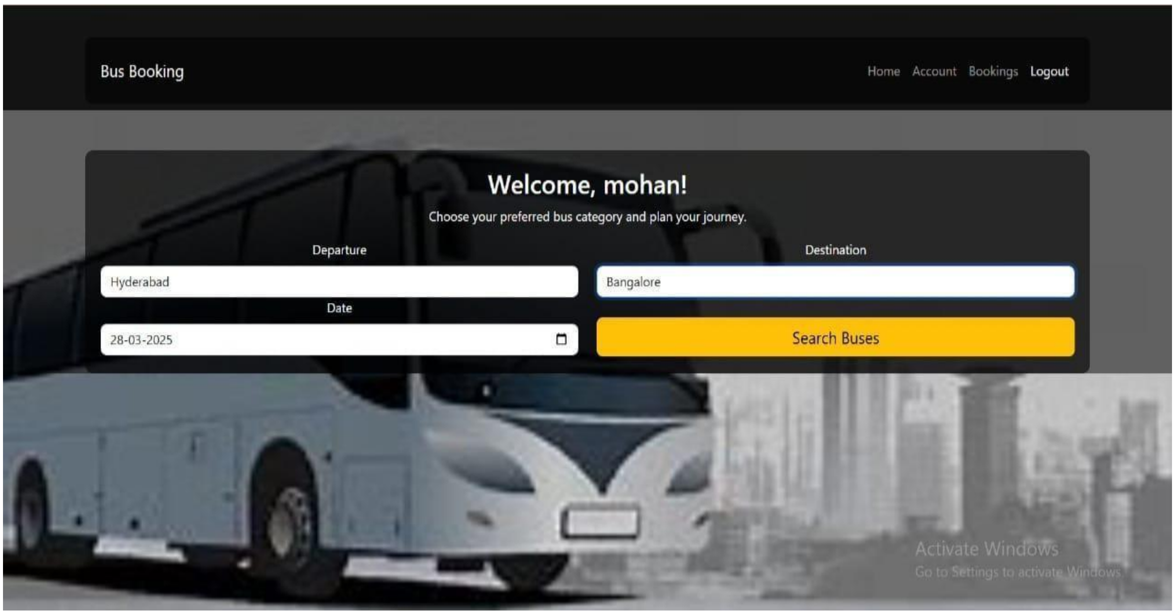
It is similar to messages that were identified as spam in the past.

Your OTP code is: 570612

Fig: 6.4 OTP email

Step4: Home of booking app

- Home page of the booking app: Departure, Destination, and Date.
- After clicking on **Search Buses**.



Welcome, mohan!

Choose your preferred bus category and plan your journey.

Departure: Hyderabad

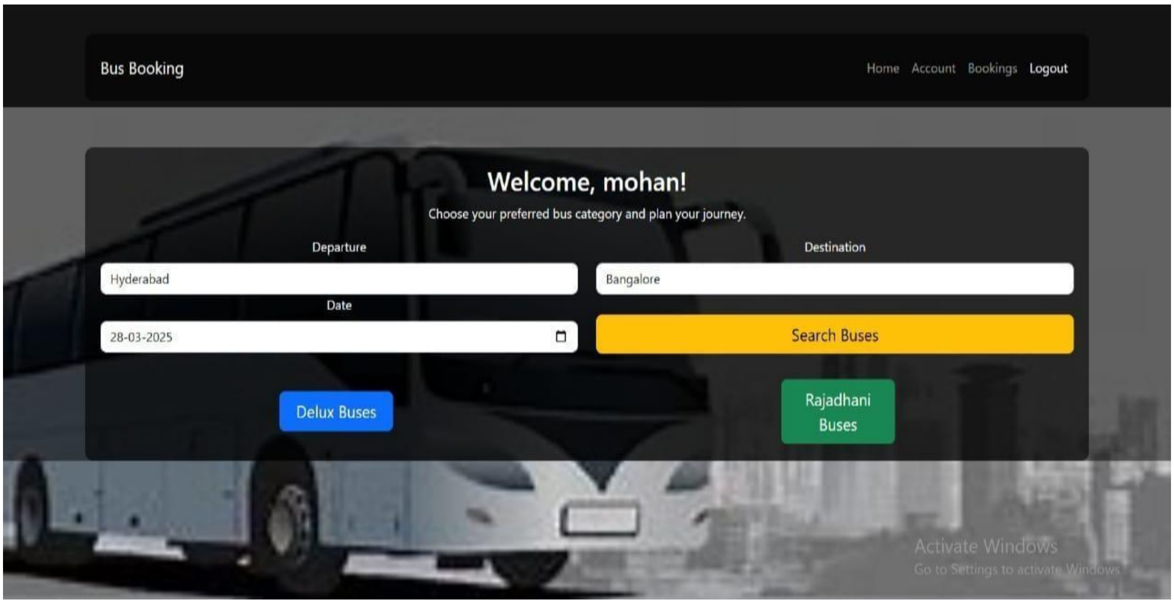
Destination: Bangalore

Date: 28-03-2025

Fig: 6.5 Home page

Step5: Select Departure and Destination

- We have two types of buses: Deluxe and Rajadhani.
- Users can select any type of bus they prefer by clicking on it.



Welcome, mohan!

Choose your preferred bus category and plan your journey.

Departure: Hyderabad

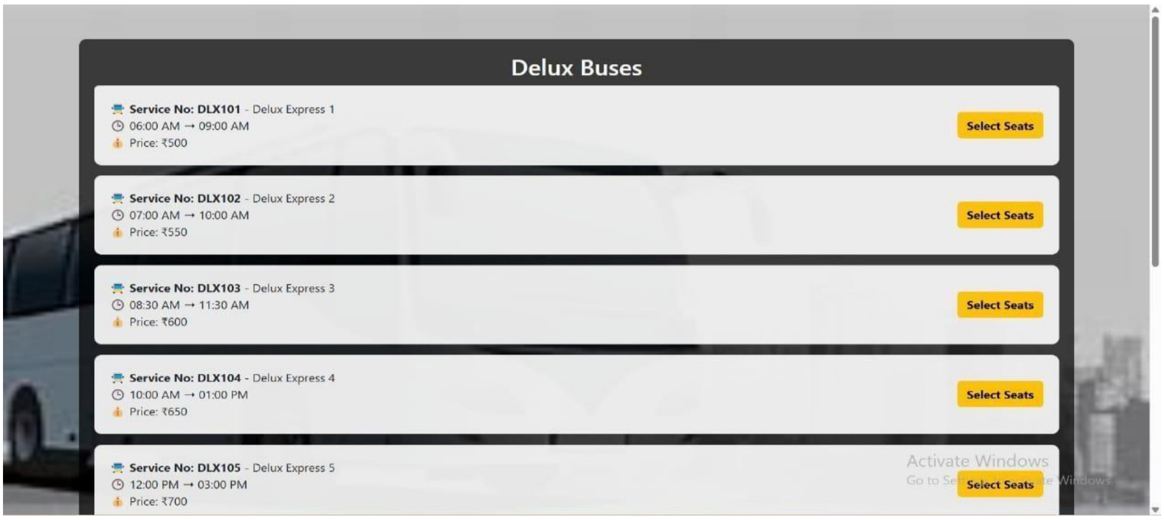
Destination: Bangalore

Date: 28-03-2025

Fig: 6.6 Select departure and destination

Step6: Buses list

- A list of available buses will be displayed based on the selected route and date.



Delux Buses

Service No: DLX101 - Delux Express 1
06:00 AM → 09:00 AM
Price: ₹500

Select Seats

Service No: DLX102 - Delux Express 2
07:00 AM → 10:00 AM
Price: ₹550

Select Seats

Service No: DLX103 - Delux Express 3
08:30 AM → 11:30 AM
Price: ₹600

Select Seats

Service No: DLX104 - Delux Express 4
10:00 AM → 01:00 PM
Price: ₹650

Select Seats

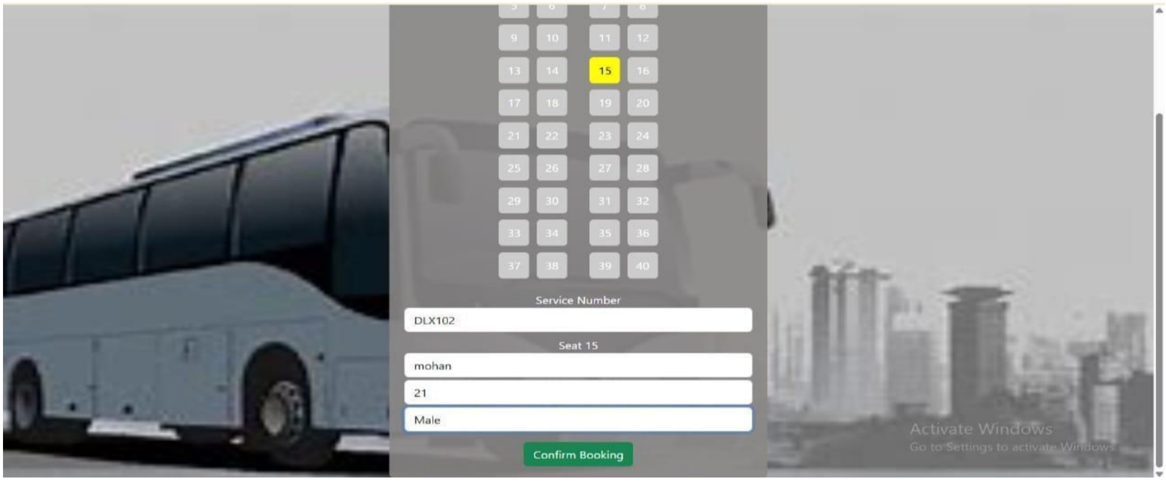
Service No: DLX105 - Delux Express 5
12:00 PM → 03:00 PM
Price: ₹700

Select Seats

Fig: 6.7 Buses list

Step7: select seats

- After selecting a bus, we proceed to choose a seat and provide the passenger's details such as name, age, and gender.



5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40

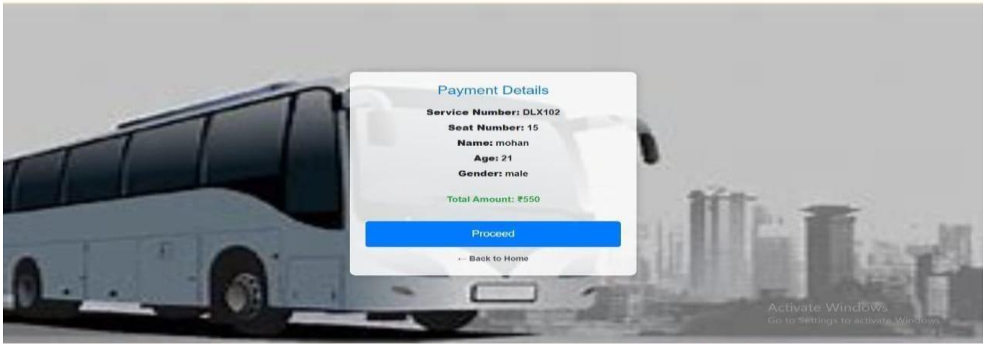
Service Number
DLX102

Seat 15
mohan
21
Male

Confirm Booking

Fig: 6.8 Select seat

Step8: Payment page



Payment Details

Service Number: DLX102
Seat Number: 15
Name: mohan
Age: 21
Gender: male

Total Amount: ₹550

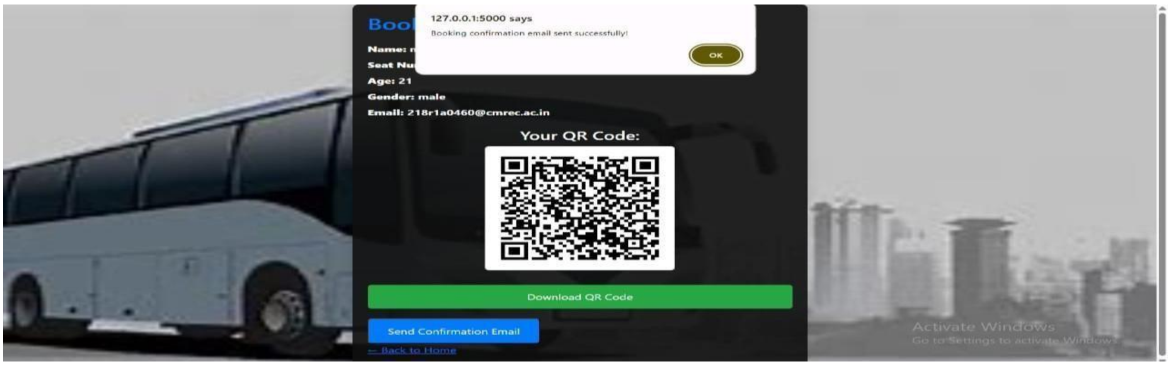
Proceed

← Back to Home

Fig: 6.9 Payment page

Step9: Payment QR page

- The passenger details and a QR code containing the ticket information will be generated.



127.0.0.1:5000 says

Booking confirmation email sent successfully!

Boo

Name: r

Seat Nu

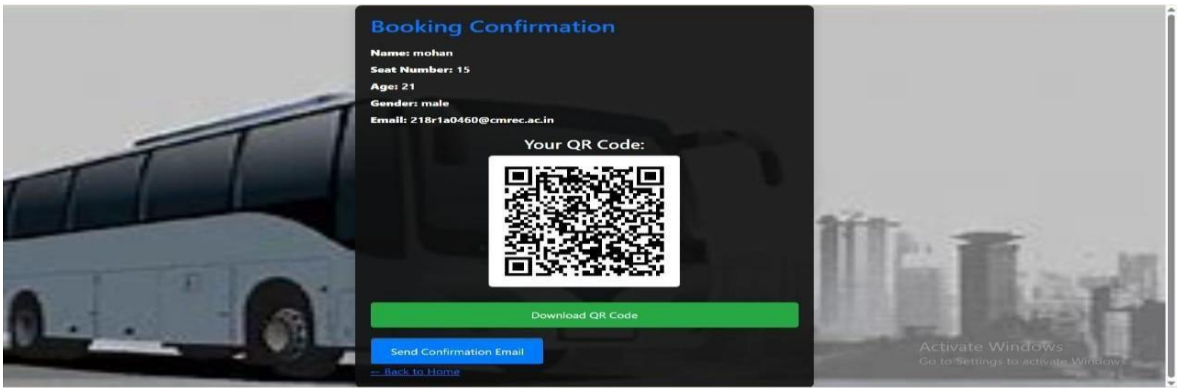
Age: 21

Gender: male

Email: 218r1a0460@cmrec.ac.in

Fig: 6.10 Payment QR page

Step10: Ticket confirmation



Booking Confirmation

Name: mohan

Seat Number: 15

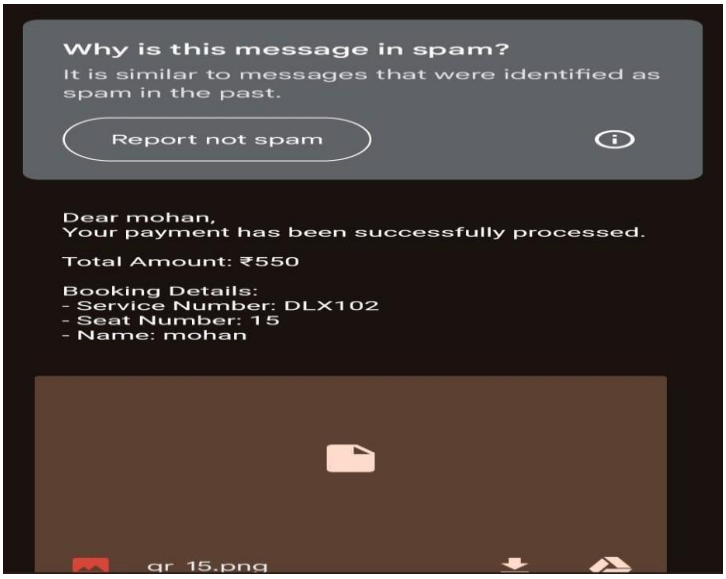
Age: 21

Gender: male

Email: 218r1a0460@cmrec.ac.in

Fig: 6.11 Ticket confirmation

- Ticket conformation through mail



Why is this message in spam?

It is similar to messages that were identified as spam in the past.

Dear mohan,
Your payment has been successfully processed.

Total Amount: ₹550

Booking Details:
- Service Number: DLX102
- Seat Number: 15
- Name: mohan

Fig: 6.12 Ticket conformation through mail

Step11: Admin Login Page

- This is the login page of the admin.
- Here, we have fields for email and password.

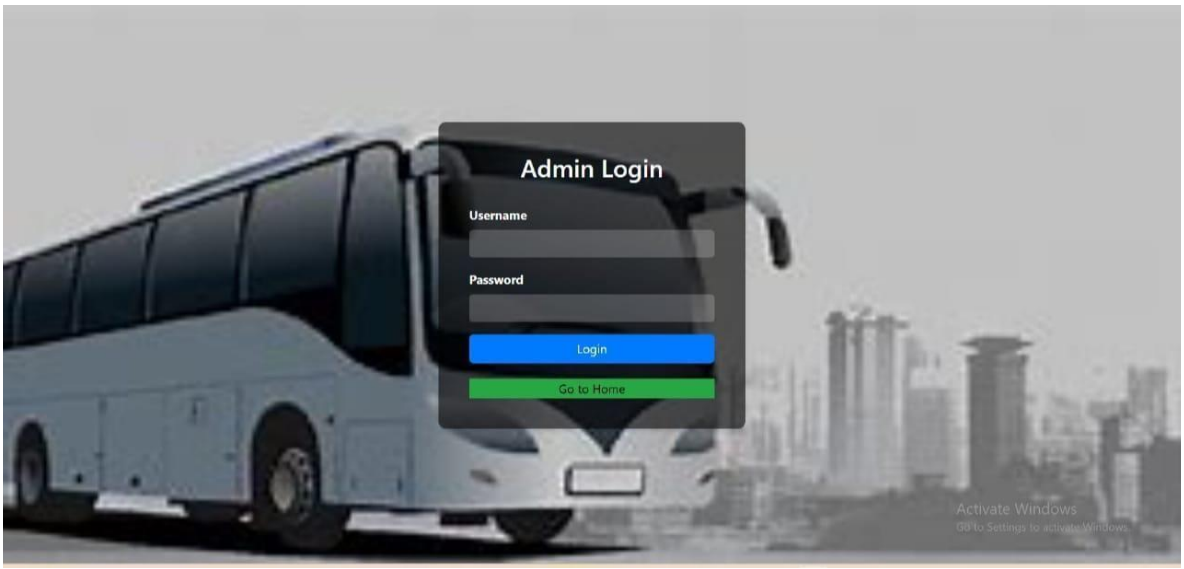
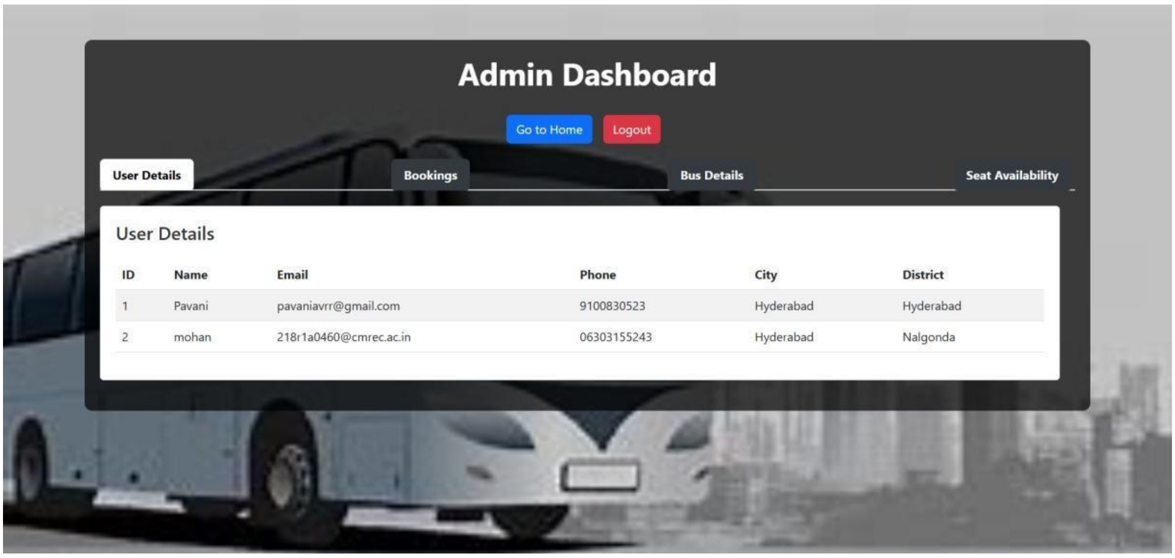


Fig: 6.13 Admin login

Step12: User Details

- This section displays the details of all registered users.



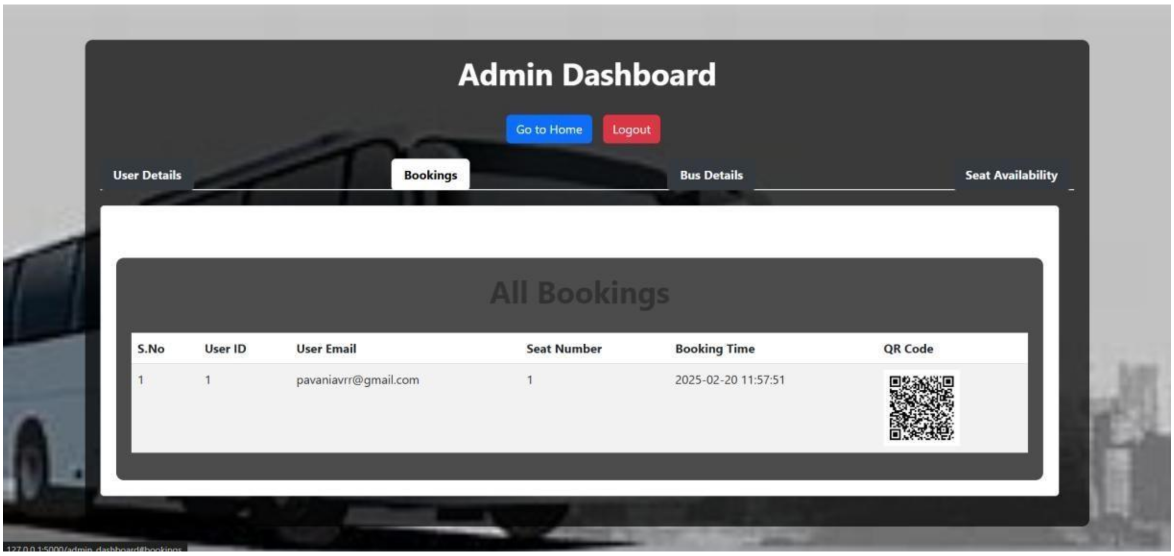
User Details

ID	Name	Email	Phone	City	District
1	Pavani	pavaniavrr@gmail.com	9100830523	Hyderabad	Hyderabad
2	mohan	218r1a0460@cmrec.ac.in	06303155243	Hyderabad	Nalgonda

Fig: 6.14 User Details

Step13: All Bookings

- This section displays all booking details for all users.

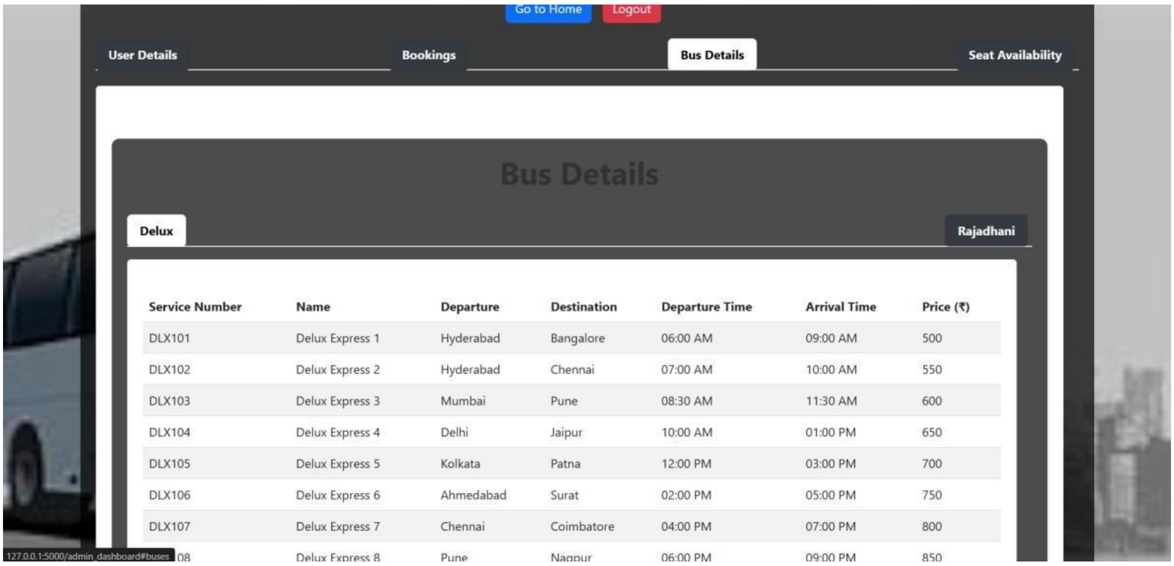


S.No	User ID	User Email	Seat Number	Booking Time	QR Code
1	1	pavaniavrr@gmail.com	1	2025-02-20 11:57:51	

Fig: 6.15 All Bookings

Step14: All Bus Details

- This section provides complete bus details for all available routes

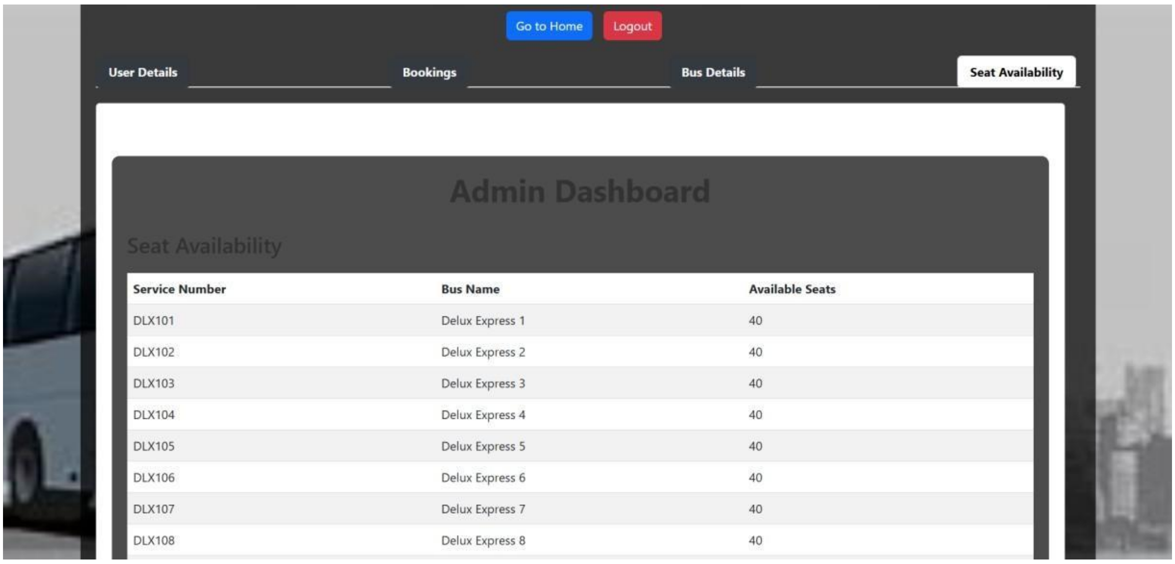


Service Number	Name	Departure	Destination	Departure Time	Arrival Time	Price (₹)
DLX101	Delux Express 1	Hyderabad	Bangalore	06:00 AM	09:00 AM	500
DLX102	Delux Express 2	Hyderabad	Chennai	07:00 AM	10:00 AM	550
DLX103	Delux Express 3	Mumbai	Pune	08:30 AM	11:30 AM	600
DLX104	Delux Express 4	Delhi	Jaipur	10:00 AM	01:00 PM	650
DLX105	Delux Express 5	Kolkata	Patna	12:00 PM	03:00 PM	700
DLX106	Delux Express 6	Ahmedabad	Surat	02:00 PM	05:00 PM	750
DLX107	Delux Express 7	Chennai	Coimbatore	04:00 PM	07:00 PM	800
08	Delux Express 8	Pune	Nagpur	06:00 PM	09:00 PM	850

Fig: 6.16 All Bus Bookings

Step15: Seat Availability

- This section shows the seat availability for each bus.



Admin Dashboard

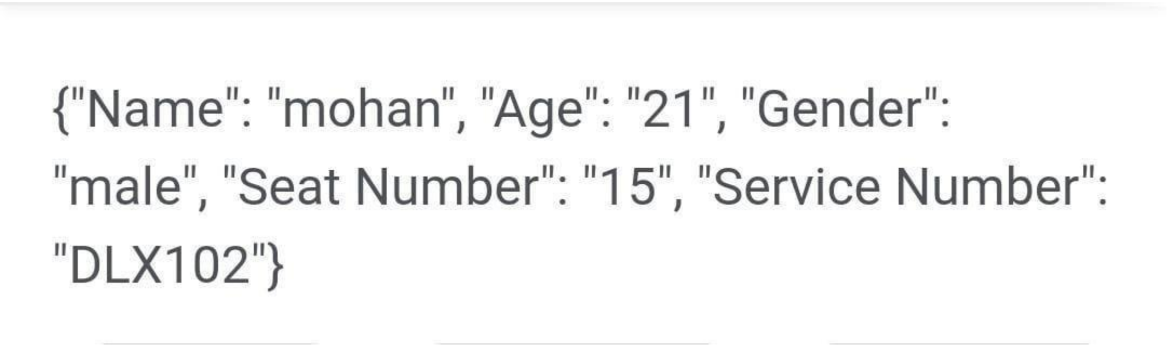
Seat Availability

Service Number	Bus Name	Available Seats
DLX101	Delux Express 1	40
DLX102	Delux Express 2	40
DLX103	Delux Express 3	40
DLX104	Delux Express 4	40
DLX105	Delux Express 5	40
DLX106	Delux Express 6	40
DLX107	Delux Express 7	40
DLX108	Delux Express 8	40

Fig: 6.17 Seat Availability

Step16: Conductor scan page

- On the Conductor Scan Page, once the QR code is scanned, the passenger's details will be shown.



```
{"Name": "mohan", "Age": "21", "Gender":
"male", "Seat Number": "15", "Service Number":
"DLX102"}
```

Fig: 6.18 Conductor scan page

ADVANTAGES

The Bus Booking System offers numerous advantages that benefit both users and bus operators, making transportation management more efficient and user-friendly.

1. **Convenience and Accessibility:**
   - Users can book bus tickets anytime, anywhere using their smartphones or computers.
   - Real-time seat availability and instant booking confirmation reduce the hassle of standing in queues.
2. **Secure and Reliable Payments:**
   - The integration of secure payment gateways ensures safe transactions through linked bank accounts or UPI.
   - Aadhaar-based authentication enhances security and prevents unauthorized access.
3. **Real-Time Updates:**
   - The system provides real-time updates on seat availability, fare changes, and route modifications.
   - Conductors and operators can manage seat bookings and cancellations efficiently.
4. **QR Code-Based E-Tickets:**
   - Environment-friendly QR codes eliminate the need for physical tickets.
   - Conductors can quickly verify tickets using the scanning feature, ensuring smooth boarding.
5. **User-Friendly Interface:**
   - The application has an intuitive design, making it easy for both tech-savvy and non-tech-savvy users to navigate.
   - Multilingual support further enhances usability for diverse users.
6. **Cost-Effective Solution:**
   - Digital ticketing reduces paper usage and administrative costs.
   - Automated management of bookings minimizes the need for manual intervention, reducing labor costs.

7. **Efficient Cancellations and Refunds:**
    - Users can cancel bookings and initiate refund requests directly from the app.
    - The automated refund system ensures timely processing and transparency.
8. **Data Management and Insights:**
    - Bus operators gain access to valuable data on passenger trends and route popularity.
    - Data analytics help in optimizing routes and pricing strategies.
9. **Reduced Fraudulent Activities:**
    - QR code authentication minimizes the chances of ticket duplication or fraud.
    - Conductors can verify tickets instantly through the system.
10. **Enhanced User Satisfaction:**
    - The overall experience of secure, seamless, and hassle-free booking improves customer satisfaction

APPLICATIONS

The Bus Booking System is versatile and can be applied across various scenarios, including:

1. **Public Transport Management:**
    - State and city transport authorities can implement the system to enhance public transport efficiency.
2. **Private Bus Operators:**
    - Private bus companies can use the system to manage routes, pricing, and passenger data effectively.
3. **Tourism and Travel Agencies:**
    - Travel agencies can integrate the system into their platforms to offer comprehensive travel packages.
4. **Corporate Transport Solutions:**
    - Companies can use the system to manage employee transportation through scheduled bus services.
5. **Event Transportation Management:**

Organizers of large events can implement the system to manage guest transportation

CHAPTER-7
CONCLUSION AND FUTURESCOPE

CONCLUSION

The **Bus Booking System** developed using the **Flask framework** offers a comprehensive, secure, and user-friendly platform for managing online bus ticket reservations. The system addresses the limitations of traditional bus booking processes by providing real-time seat availability, secure online payments, OTP-based user authentication, and QR code-enabled digital ticketing. The modular design ensures that each component—such as user management, bus and seat selection, payment processing, and ticket generation—works seamlessly together while remaining scalable for future enhancements. The integration of essential features like automated billing, robust security measures, and real-time notifications enhances the overall user experience and operational efficiency. By leveraging **Flask's lightweight architecture**, the system achieves high performance, flexibility, and maintainability, making it ideal for both small and large-scale implementations. Additionally, features like QR code generation for digital tickets reduce paper usage, contributing to an eco-friendly solution. In conclusion, this **Bus Booking System** not only streamlines the ticket booking process but also provides a secure, scalable, and user-centric solution that can be easily adapted and expanded to meet evolving user needs and technological advancements. The project successfully delivers a modern, efficient, and reliable bus reservation experience, making travel planning more accessible and convenient for all users.

FUTURE SCOPE

The Bus Booking System has significant potential for growth and improvement. Some areas of future development include:

1. **AI-Powered Predictive Analysis:**
   - Implementing AI algorithms can predict peak booking times, suggest optimal routes, and help operators manage fleet schedules more effectively.
2. **Multimodal Transport Integration:**
   - Expanding the system to include other modes of transport, such as trains, metro, and ferries, can provide users with a seamless end-to-end travel experience.
3. **Voice Assistance and Chatbots:**
   - Integrating voice-enabled assistance and AI chatbots can enhance customer support and improve user experience.
4. **Personalized Recommendations:**
   - Using machine learning, the system can offer personalized bus recommendations based on user preferences and travel history.
5. **Offline Booking Support:**
   - Developing offline booking capabilities can cater to users in remote areas with limited internet access.
6. **Loyalty Programs and Discounts:**
   - Implementing a loyalty rewards system can encourage frequent users by offering discounts and special promotions.
7. **Sustainability Initiatives:**
   - Promoting the use of eco-friendly transport by providing carbon footprint data and incentives for users choosing sustainable travel options.
8. **Enhanced Security Features:**
   - Advanced biometric authentication and facial recognition can further strengthen user verification.
9. **Partnership with Local Services:**
   - Collaborating with local businesses for exclusive deals and partnerships can improve the user experience and drive revenue growth.

REFERENCE

[1] X. Guo, J. Wu, H. Sun, R. Liu, and Z. Gao, “Timetable coordination of first trains in urban railway network: A case study of beijing,” Applied Mathematical Modelling, vol. 40, no. 17, pp. 8048–8066, 2016.

[2] W. Wu, P. Li, R. Liu, W. Jin, B. Yao, Y. Xie, and C. Ma, “Predicting peak load of bus routes with supply optimization and scaled shepard interpolation: A newsvendor model,” Transportation Research Part E: Logistics and Transportation Review, vol. 142, p. 102041, 2020.

[3] N. Besinovi ˇ c, L. De Donato, F. Flammini, R. M. Goverde, Z. Lin, R. Liu, ´ S. Marrone, R. Nardone, T. Tang, and V. Vittorini, “Artificial intelligence in railway transport: Taxonomy, regulations and applications,” IEEE Transactions on Intelligent Transportation Systems, 2021.

[4] S. C. Kwan and J. H. Hashim, “A review on co-benefits of mass public transportation in climate change mitigation,” Sustainable Cities and Society, vol. 22, pp. 11–18, 2016.

[5] Y. Wang, W. Zhang, T. Tang, D. Wang, and Z. Liu, “Bus od matrix reconstruction based on clustering wi-fi probe data,” Transportmetrica B: Transport Dynamics, pp. 1–16, 2021, doi: 10.1080/21680566.2021.1956388.

[6] S. J. Berrebi, K. E. Watkins, and J. A. Laval, “A real-time bus dispatching policy to minimize passenger wait on a high frequency route,” Transportation Research Part B: Methodological, vol. 81, pp. 377–389, 2015.

[7] A. Fonzone, J.-D. Schmocker, and R. Liu, “A model of bus bunching ¨ under reliability-based passenger arrival patterns,” Transportation Research Part C: Emerging Technologies, vol. 59, pp. 164–182, 2015.

[8] J. D. Schmocker, W. Sun, A. Fonzone, and R. Liu, “Bus bunching ¨ along a corridor served by two lines,” Transportation Research Part B: Methodological, vol. 93, pp. 300–317, 2016.

[9] D. Chen, Q. Shao, Z. Liu, W. Yu, and C. L. P. Chen, “Ridesourcing behavior analysis and prediction: A network perspective,” IEEE Transactions on Intelligent Transportation Systems, pp. 1–10, 2020.

[10] E. Nelson and N. Sadowsky, “Estimating the impact of ride-hailing app company entry on public transportation use in major us urban areas,” The B.E. Journal of Economic Analysis & Policy, vol. 19, no. 1, p. 20180151, 2019.

[11] Z. Chen, K. Liu, J. Wang, and T. Yamamoto, “H-convlstm-based bagging learning approach for ride-hailing demand prediction considering imbalance problems and sparse

uncertainty," Transportation Research Part C: Emerging Technologies, vol. 140, p. 103709, 2022.

[12] R. Liu and S. Sinha, "Modelling urban bus service and passenger reliability," 2007.

[13] J. A. Sorratini, R. Liu, and S. Sinha, "Assessing bus transport teliability using micro-simulation," Transportation Planning and Technology, vol. 31, no. 3, pp. 303–324, 2008.

[14] Y. Wang, W. Zhang, T. Tang, D. Wang, and Z. Liu, "Bus od matrix reconstruction based on clustering wi-fi probe data," Transportmetrica B: Transport Dynamics, pp. 1–16, 2021.

[15] Y. Hollander and R. Liu, "Estimation of the distribution of travel times by repeated simulation," Transportation Research Part C: Emerging Technologies, vol. 16, no. 2, pp. 212–231, 2008.

[16] W. Wu, R. Liu, and W. Jin, "Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour," Transportation Research Part B: Methodological, vol. 104, pp. 175–197, 2017.

[17] W. Wu, R. Liu, W. Jin, and C. Ma, "Stochastic bus schedule coordination considering demand assignment and rerouting of passengers," Transportation Research Part B: Methodological, vol. 121, pp. 275–303, 2019.

[18] W. Wu, R. Liu, and W. Jin, "Designing robust schedule coordination scheme for transit networks with safety control margins," Transportation Research Part B: Methodological, vol. 93, pp. 495–519, 2016.

[19] S. Zhong and D. J. Sun, A Spatio-temporal Distribution Model for Determining Origin–Destination Demand from Multisource Data. Springer, Singapore, 2022, pp. 33–52.

[20] M. Bordagaray, L. dell'Olio, A. Fonzone, and Ibeas, "Capturing the conditions that introduce systematic variation in bike-sharing travel behavior using data mining techniques," Transportation Research Part C: Emerging Technologies, vol. 71, pp. 231–248, 2016.

[21] B. Chidlovskii, "Mining smart card data for travellers' mini activities," IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 11, pp. 3676–3685, 2018.

[22] T. Tang, R. Liu, and C. Choudhury, "Incorporating weather conditions and travel history in estimating the alighting bus stops from smart card data," Sustainable Cities and Society, vol. 53, p. 101927, 2020.

[23] X. Zhang, Q. Zhang, T. Sun, Y. Zou, and H. Chen, "Evaluation of urban public transport priority performance based on the improved topsis method: A case study of wuhan," .

APPENDIX

Appendix-1: Gather Components

Before beginning the project, ensure you have all necessary components:

1. Development Device (PC/Laptop with suitable IDE)
2. Internet Connectivity for real-time booking and updates
3. Android Smartphone or Web Browser (for user interface)
4. Server/Database System (e.g., MySQL or Firebase)
5. GPS Module or API (for bus tracking, optional)
6. Payment Gateway Integration (optional for e-ticketing)

Appendix-2: System Architecture & Flow

Component Overview

1. **User Interface:** Android/Web app for booking, viewing buses, and payment.
2. **Backend Server:** Handles logic for booking, user data, and bus schedules.
3. **Database:** Stores user credentials, bus routes, seat availability, etc.
4. **Admin Panel:** Manages buses, timings, and booking logs.
5. **Notification System:** Sends booking confirmations or alerts to users.
6. **Optional GPS Integration:** Tracks bus location for real-time updates.

Appendix-3: Setting Up the Development Environment

1. Install required IDE (e.g., Android Studio / VS Code)
2. Set up development SDKs (e.g., Java/Kotlin for Android or Node.js/Python for web)
3. Configure Database (MySQL, Firebase, or MongoDB)
4. Integrate Firebase or Twilio for notifications (optional)
5. Set up APIs for real-time data (e.g., location, payment gateway)

Appendix-4: Writing the Code for Core Modules

1. User Login/Signup
2. Search and Browse Bus Routes
3. Seat Selection and Booking Confirmation
4. Database Connectivity
5. (Optional) Real-time Bus Tracking Module

Appendix-5: Testing the System

1. Test user login and signup with valid/invalid inputs
2. Verify bus listing and seat selection functionality
3. Test booking process and database updates
4. Check responsiveness and UI on different devices

Appendix-6: Deployment & Installation

1. Host backend on cloud (Heroku, Firebase, etc.)
2. Deploy front-end app or APK for users
3. Connect front-end to backend/database
4. Distribute to test users or publish on app store

Appendix-7: Monitoring and Maintenance

1. Monitor booking logs and server performance
2. Regularly update bus schedules and fare info
3. Handle feedback and fix bugs
4. Plan for version updates with new features