

**ONLINE CAB BOOKING SYSTEM**

**A PROJECT REPORT**

***Submitted by***

**DHARANEESHWARAN.V (8115U23AD303)**

***in partial fulfillment of requirements for the award of the course***

**CGB1201 - JAVA PROGRAMMING**

***in***

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**K. RAMAKRISHNAN COLLEGE OF ENGINEERING**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

# SAMAYAPURAM – 621 112

# 

**DECEMBER - 2024**

**K. RAMAKRISHNAN COLLEGE OF ENGINEERING**



**(AUTONOMOUS)**

**SAMAYAPURAM – 621 112**

# BONAFIDE CERTIFICATE

Certified that this project report on **“ ONLINE CAB BOOKING SYSTEM”** is the bonafide work of **DHARANEESHWARAN.V (8115U23AD303)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

|  |  |
| --- | --- |
|  |  |
| **SIGNATURE**  Dr.B.KIRAN BALA, B.Tech.,M.B.A.,M.E.,Ph.D,  **HEAD OF THE DEPARTMENT,**  ASSOCIATE PROFESSOR,  Department of Artificial Intelligence  and Data Science,  K.Ramakrishnan College of Engineering (Autonomous),  Samayapuram–621112. | **SIGNATURE**  Mr. M.KARTHIK, M.E.,  **SUPERVISOR,**  ASSISTANT PROFESSOR,  Department of Artificial Intelligence  and Data Science,  K.Ramakrishnan College of Engineering (Autonomous),  Samayapuram–621112. |

Submitted for the viva-voce examination held on …………….

|  |  |
| --- | --- |
|  |  |
| INTERNAL EXAMINER | EXTERNAL EXAMINER |

**DECLARATION**

I declare that the project report on **“ONLINE CAB BOOKING SYSTEM”** is the result of original work done by us and best of our knowledge, similar workhas not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

**Signature**

DHARANEESHWARAN.V

Place: Samayapuram

Date:

**ACKNOWLEDGEMENT**

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Engineering (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, M.B.A., Ph.D.,** for forwarding to our project and offering adequateduration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. B. KIRAN BALA, B.Tech.,M.B.A.,M.E., Ph.D,**

Head of the department, **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr. M. KARTHIK, M.E.,** Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE,** for his incalculable suggestions, creativity, assistance and patiencewhich motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND**

**DATA SCIENCE**

**VISION OF THE INSTITUTION**

To achieve a prominent position among the top technical institutions.

**MISSION OF THE INSTITUTION**

M1: To bestow standard technical education par excellence through state of the art

infrastructure, competent faculty and high ethical standards.

M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.

M3: To provide education for developing high-quality professionals to transform the society.

**VISION OF THE DEPARTMENT**

To prove excellence in Data Science research, education and innovation with AI tools.

**MISSION OF THE DEPARTMENT**

M1: To contribute for greater collaboration with academia and businesses.

M2: To impart quality and research based education to promote innovations providing smart solutions in multi-disciplinary area of Artificial Intelligence and Data Science.

M3: To provide eminent Data Scientists to serve humanity

**PROGRAM EDUCATIONAL OBJECTIVES (PEOS)**

Our graduates shall

PEO1: To create Graduates with successful career in the field of Data Science in all industries or pursue higher education and research or evolve as entrepreneur.

PEO2: To equip the Graduates with the ability and attitude to adapt to emerging technological changes in the field of expert systems.

PEO3: To excel the students as socially committed engineers with high ethical values, leadership qualities and openness for the needs of society.

**PROGRAM OUTCOMES**

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

* **PSO1:** To develop optimized Data Science Solutions, through analysis, design, implementation, and evaluation to give technological solutions for real-time societal issues.
* **PSO2:** To employ advanced analytic platforms in creating innovative career paths to become best data scientists.

**ABSTRACT**

The Online Cab Booking System is a comprehensive solution aimed at revolutionizing urban mobility by offering a streamlined and user-friendly platform for passengers and drivers. It simplifies the ride-booking process, ensuring convenience, transparency, and reliability. Passengers can book rides through a web or mobile application, access real-time cab tracking, estimate fares, and make secure payments, while drivers receive optimized ride requests and navigation assistance.

The system leverages advanced technologies such as GPS for precise location tracking, cloud-based databases for scalability and reliability, and machine learning algorithms for predictive analytics in demand forecasting, route optimization, and dynamic pricing. Administrators oversee operations by managing user accounts, resolving disputes, and monitoring system performance, ensuring a seamless and secure experience for all stakeholders.

By addressing common issues like inconsistent cab availability, inefficient dispatch systems, safety concerns, and a lack of fare transparency, this system provides a reliable and efficient alternative to traditional cab services. Additionally, the platform enhances safety through features such as trip monitoring, emergency support, and verified driver profiles. The Online Cab Booking System caters to the increasing demand for convenient, technology-driven transportation solutions, making it an indispensable tool for modern urban commuters.

# ABSTRACT WITH POs AND PSOs MAPPING

# CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS

|  |  |  |
| --- | --- | --- |
| **ABSTRACT** | **POs MAPPED** | **PSOs MAPPED** |
| The Online Cab Booking System is a digital platform that streamlines urban transportation by connecting passengers and drivers seamlessly. It offers features like real-time cab tracking, fare estimation, secure payments, and optimized routes. Leveraging GPS, cloud databases, and machine learning, the system ensures convenience, transparency, and efficiency. By addressing challenges such as unreliable availability and fare ambiguity, it provides a safe, user-friendly, and scalable solution for modern commuters. | PO1 - 2  PO2 - 2  PO3 - 3  PO4 - 3  PO7 - 2 | PSO1 - 1  PSO2 - 2 |

Note: 1- Low, 2-Medium, 3- High

**TABLE OF CONENTS**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| 2.2.1 | Block Diagram For Online Cab Booking system | 6 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATIONS** | **EXPANSION** |
| **EVS** | Electric Vehicles |
| **AVS** | Autonomous Vehicles |
| **AI** | Artificial Intelligence |
| **ML** | Machine Learning |
| **GPS** | Global Positioning System |
| **GUI** | Graphical User Interface |

# CHAPTER 1

# INTRODUCTION

# In today’s fast-paced world, efficient and reliable transportation systems are essential for urban mobility. The Online Cab Booking System is a transformative platform designed to address the challenges of traditional transportation methods by integrating technology into the booking process. This system allows passengers to book rides instantly through a mobile application or website, ensuring convenience, transparency, and accessibility.

# The platform caters to diverse user needs, offering features such as real-time location tracking, dynamic fare estimation, secure payment gateways, and trip history records. Passengers can enjoy a seamless booking experience with safety features like driver verification, trip monitoring, and emergency assistance, enhancing trust and reliability.

# For drivers, the system provides a structured approach to ride management, enabling them to receive ride requests, navigate optimized routes, and track their earnings efficiently. This integration reduces idle time and maximizes productivity. Administrators play a vital role by overseeing the platform’s operations, managing user accounts, resolving disputes, and ensuring smooth functionality.

# By leveraging technologies like GPS, cloud computing, and machine learning, the Online Cab Booking System adapts to user demands, predicts peak hours, and ensures optimal service quality. This innovative solution addresses common issues like unavailability of cabs, inefficient dispatch systems, and lack of fare transparency, making it a reliable alternative to traditional methods.

# The Online Cab Booking System not only enhances the transportation experience but also contributes to a more connected and technologically advanced urban infrastructure, meeting the needs of modern commuters while setting a benchmark for future mobility solutions.

# Objective

# The objective of the Online Cab Booking System is to provide a seamless and efficient platform for connecting passengers with cab services. This system aims to enhance user convenience by enabling passengers to book rides instantly through an intuitive mobile application or website. It seeks to improve transparency with real-time cab tracking, fare estimation, and detailed trip histories while ensuring secure and diverse payment options. For drivers, the platform optimizes ride management by offering tools for route navigation, earning tracking, and efficient communication with passengers. Additionally, the system prioritizes safety with features like driver verification, trip monitoring, and emergency assistance. By addressing common transportation challenges such as unavailability of cabs, inefficient dispatch systems, and fare ambiguity, the system aspires to deliver a reliable, scalable, and technologically advanced solution for modern urban mobility.

# Overview

# The Online Cab Booking System is a technology-driven platform designed to modernize the transportation sector by providing an efficient and convenient way to connect passengers with cab drivers. It serves as a one-stop solution for booking rides, offering features such as real-time location tracking, dynamic fare calculation, secure payment options, and trip management. The system caters to passengers by simplifying the process of finding and booking a ride while ensuring a safe and transparent experience and The Online Cab Booking System addresses common issues in traditional transportation methods, such as unreliable availability, lack of fare transparency, and safety concerns. By integrating modern technology and user-friendly interfaces, it provides a scalable, reliable, and secure solution to meet the demands of urban commuters while setting new standards for transportation systems.

**1.3 Java Programming Concepts**

**Basic concepts of OOPS**

1. **Classes and Objects**

A class is a blueprint for creating objects, which are instances of classes. In this project, the Item class acts as a blueprint, defining properties like name, price, and quantity. Objects like "Laptop," "Mouse," and "Smartphone" are created from the class to represent inventory items.

1. **Encapsulation**

Encapsulation is the bundling of data (fields) and methods into a single unit, restricting access to certain components. In this project, the Item class uses private fields to store data, and public getter and setter methods control access to these fields.

1. **Inheritance**

Inheritance allows a class to acquire properties and behaviors of another class. For example, if the project involves additional types of items, a subclass like ElectronicsItem can inherit common properties from the Item class, reducing redundancy.

1. **Polymorphism**

Polymorphism enables a single function to behave differently based on the context. In this project, method overloading could allow multiple ways to initialize inventory items (e.g., providing name and price only or providing name, price, and quantity).

1. **Abstraction**

Abstraction hides implementation details and shows only the functionality. In this project, the GUI hides the underlying code and logic for managing the inventory system, presenting only the essential actions to the user (e.g., Add, Update, or Delete)

**Project related Concepts**

1. **Real-Time Location Tracking**

This concept involves using technologies like GPS and Google Maps API to track the real-time location of cabs and passengers. This allows passengers to monitor their ride’s progress and ensures that drivers are directed to the most efficient routes. Real-time data streaming and handling are key components of this feature.

1. **Dynamic Pricing and Surge Pricing**

This refers to adjusting ride prices based on factors like demand, traffic, weather conditions, and time of day. Machine learning algorithms can be implemented to predict demand spikes and optimize pricing. Dynamic pricing ensures that prices are fair and competitive while also benefiting drivers during peak times.

1. **Payment Gateway Integration**

The system must incorporate secure and reliable methods for payment processing, including options like credit/debit cards, wallets, and online banking. Integration with third-party payment gateways (e.g., PayPal, Stripe) is essential for enabling smooth, secure transactions within the app.

1. **Ride Matching and Dispatch System**

This system matches available drivers with passengers based on proximity, driver availability, and ride preferences. Algorithms for ride matching optimize both passenger wait time and driver efficiency. The dispatch system is crucial for managing the flow of requests and ensuring that passengers are paired with the closest available driver.

1. **User Authentication and Authorization**

To ensure the security of both drivers and passengers, user authentication (through OAuth, email verification, etc.) is implemented. Role-based access control (RBAC) is used to differentiate between the Passenger, Driver, and Admin user roles, ensuring the relevant features and data.

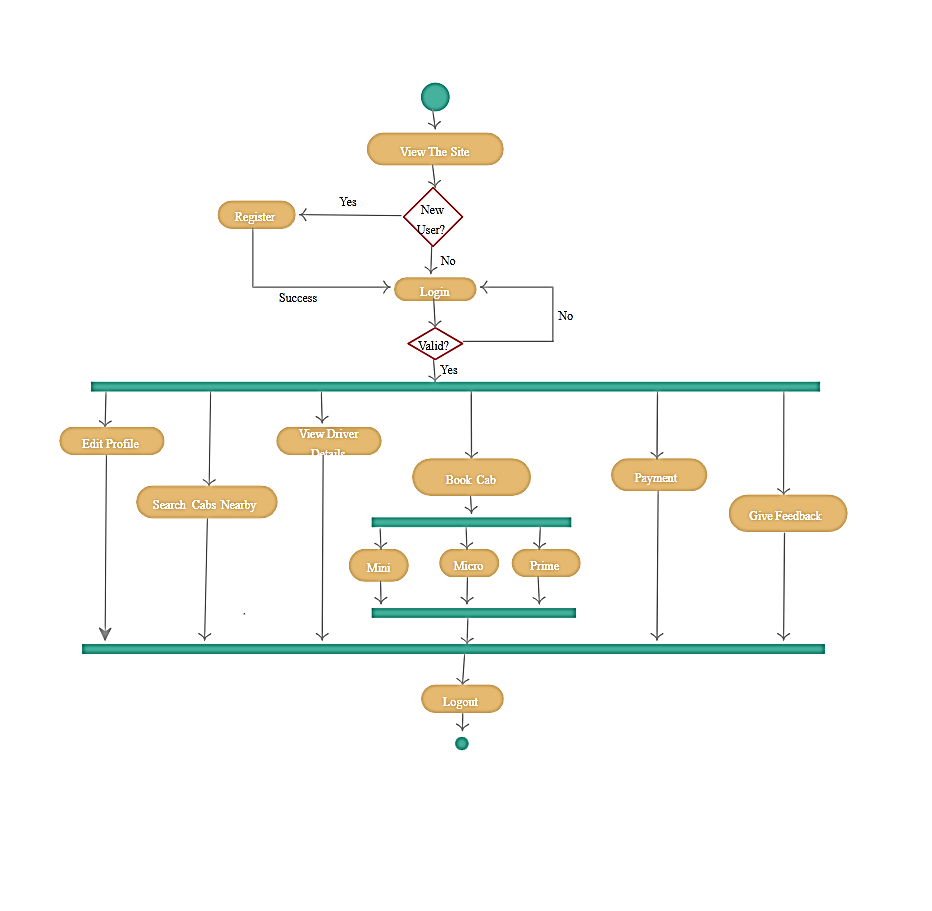
**CHAPTER 2**

**PROJECT METHODOLOGY**

**2.1 Proposed Work**

The proposed work for the Online Cab Booking System focuses on developing a comprehensive, efficient, and scalable platform that connects passengers and drivers seamlessly. The first step involves designing the system architecture to ensure scalability, reliability, and efficiency, utilizing cloud infrastructure and creating a user-friendly interface. Real-time location tracking and route optimization will be implemented using GPS to allow passengers to monitor rides and drivers to follow the most efficient routes. The system will incorporate intelligent ride matching algorithms and dynamic pricing to adjust fares based on demand, traffic, and other factors. Secure payment gateway integration will facilitate various payment options, ensuring data privacy and financial security. User authentication mechanisms, including multi-factor authentication, will be employed to ensure the security of passengers and drivers, with role-based access control for customized user experiences. An admin dashboard will be created for platform management, enabling administrators to monitor ride requests, handle disputes, and track system performance. Safety features, such as driver background checks, real-time monitoring, and an SOS button, will prioritize user security. A mobile application will be developed for both iOS and Android, providing an intuitive interface for booking rides, managing payments, and tracking drivers. The system will use data analytics for reporting and improving service quality, helping optimize ride patterns and user experience. Lastly, scalability and performance optimization will be a priority, ensuring the platform can handle increasing demand through cloud technologies and performance management tools. The overall aim of the project is to provide a secure, reliable, and efficient urban transportation solution, enhancing the daily commute for passengers while supporting drivers' productivity.

**2.2 Block Diagram**

****

**Fig 2.2.1 Block Diagram For Online Cab Booking System**

**CHAPTER 3**

**MODULE DESCRIPTION**

## 3.1 User Management Module

## The User Management Module is essential for maintaining a structured and secure platform for all users. It allows passengers and drivers to register by providing basic details such as name, contact information, and identity verification documents. Drivers can also upload details about their vehicles, including license numbers and insurance information. This module implements secure authentication mechanisms, such as email and phone verification or multi-factor authentication, to protect user accounts. It also manages user profiles, enabling passengers to view their trip history and preferences, and allowing drivers to track their earnings and schedules. Role-based access control ensures that each user type—passenger, driver, or admin—has access only to features relevant to their role. For instance, passengers can book rides and make payments, while drivers can accept ride requests and navigate optimized routes. This module plays a critical role in ensuring a seamless and secure user experience.

## 3.2 Ride Management Module

## The Ride Management Module is the backbone of the Online Cab Booking System, handling all aspects of booking, tracking, and completing rides. Passengers can request rides by specifying their pick-up and drop-off locations, which are processed in real-time by the system. The module then matches passengers with the nearest available driver using an intelligent ride-matching algorithm that considers factors like proximity, driver availability, and user preferences. GPS-enabled tracking ensures passengers can monitor the real-time location of their cab and provides drivers with optimized routes to the destination, reducing travel time and fuel consumption. The module also maintains a detailed log of all rides, including timestamps, trip distances, and fare calculations, accessible to both passengers and drivers.

## 3.3 Payment and Billing Module

## The Payment and Billing Module ensures that financial transactions within the system are seamless, secure, and transparent. It calculates fares dynamically, considering factors like distance, travel time, and demand fluctuations (surge pricing). This module integrates with multiple secure payment gateways to support various payment methods, including credit/debit cards, digital wallets, and bank transfers. Payment processing is encrypted to ensure data security and protect sensitive user information. It also provides passengers with detailed billing information, including fare breakdowns and electronic receipts, which can be accessed via the app or email. Drivers can use this module to track their earnings, receive payouts, and view their payment history. The module incorporates refund mechanisms and support for resolving billing disputes, enhancing user trust and satisfaction.

## 3.4 Admin Dashboard Module

## The Admin Dashboard Module empowers system administrators to oversee and manage the platform effectively. It provides a centralized interface for monitoring user activity, managing accounts, and handling ride disputes. Administrators can review driver and passenger profiles, approve or reject driver registrations, and monitor ongoing trips for quality assurance. The dashboard includes analytics and reporting tools to provide insights into system performance, ride statistics, and revenue generation. Administrators can also use this module to configure pricing strategies, manage promotional campaigns, and handle system notifications. By offering full control over the platform’s operations, the Admin Dashboard Module ensures that the system runs smoothly and remains user-centric and the module allows configuration of pricing strategies, promotions, and discounts. It provides comprehensive reports on rides, revenue, and user activity, helping administrators make data-driven decisions to improve the platform's efficiency and user satisfaction.

## 3.5 Safety and Notification Module

## The Safety and Notification Module is a crucial aspect of the Online Cab Booking System, ensuring user security and providing real-time communication updates. To enhance safety, the module includes features such as driver background verification, where drivers are thoroughly vetted through identity and criminal record checks before being approved to operate on the platform.

## Real-time trip monitoring allows passengers to share their live trip details with trusted contacts, ensuring they feel secure during their rides. An emergency SOS button is integrated into the app, enabling passengers or drivers to immediately alert authorities or the platform’s support team in case of emergencies. This module also includes features like panic alerts and automated notifications to admins for rapid response in critical situations.

## On the communication front, the module incorporates a robust notification system to keep users informed at every stage of their journey. Push notifications are used to update passengers about ride confirmations, driver arrival, trip progress, and payment status. Drivers receive updates regarding ride requests, route changes, and earnings. Additionally, in-app messaging facilitates direct communication between passengers and drivers for ride coordination, minimizing misunderstandings and delays. Notifications are designed to be non-intrusive yet effective, ensuring users stay updated without feeling overwhelmed. By combining advanced safety protocols with reliable communication tools, the Safety and Notification Module builds trust and enhances the overall experience for all users of the platform.

## For emergency situations, an SOS button is available, allowing passengers and drivers to instantly alert emergency services or the platform’s support team. This feature ensures quick assistance during critical incidents and in the terms of communication, the module uses push notifications to keep users informed about their ride status, including updates on driver arrival, trip progress, and payment confirmation. Drivers are also notified of ride requests, navigation updates, and earnings summaries.

**CHAPTER 4**

**CONCLUSION & FUTURE SCOPE**

**4.1 CONCLUSION**

The Online Cab Booking System represents a significant leap forward in urban transportation, offering a seamless, user-friendly, and secure platform for both passengers and drivers. The system addresses the complexities of traditional taxi services by leveraging technology to optimize processes, improve user experience, and ensure safety. Throughout this project, the primary goal has been to create a system that not only meets but exceeds the expectations of users in terms of ease of use, security, and reliability.

One of the standout features of the system is its user-friendly interface, which ensures that both passengers and drivers can easily interact with the platform. From booking a ride to making payments, every aspect of the system has been designed with simplicity in mind, ensuring that users of all technical backgrounds can navigate the platform effortlessly. The system’s intelligent ride matching and route optimization algorithms ensure that passengers are paired with nearby drivers quickly and efficiently, minimizing wait times and maximizing convenience. Furthermore, by using GPS and real-time data, the system enhances both the speed and accuracy of rides, allowing users to track their journeys and adjust routes as needed, further improving the overall experience.

Another critical aspect of the system is its focus on security and safety. The Safety and Notification Module ensures that users feel secure at all times during their rides. By incorporating features such as driver background checks, real-time trip monitoring, an emergency SOS button, and the ability to share live trip details with trusted contacts, the system ensures that both passengers and drivers are protected. The integration of these features helps to build trust, which is essential for the success of any service-oriented platform. Moreover, the notification system plays a vital role in ensuring effective communication between passengers, drivers, and administrators.

The push notifications and in-app messaging system help maintain smooth coordination and keep users informed about important updates such as ride status, driver arrival, and payment confirmation.

In addition to enhancing safety, the Payment and Billing Module ensures a smooth and secure financial experience for users. With its dynamic pricing system, the platform adjusts fares based on real-time demand, traffic conditions, and other factors, ensuring fair pricing for both passengers and drivers. Integration with multiple payment gateways allows for flexibility in payment options, including credit/debit cards, mobile wallets, and bank transfers. This, combined with encrypted payment processing, ensures that all transactions are secure and hassle-free. For drivers, the ability to track earnings and receive timely payouts further increases the appeal of the system, making it a valuable tool for both passenger and driver communities.

The Admin Dashboard Module offers a comprehensive view of the system’s operations, allowing administrators to manage users, monitor ride requests, handle disputes, and optimize performance. The dashboard’s analytics capabilities provide valuable insights into ride trends, user behavior, and overall platform performance. By utilizing these insights, administrators can make informed decisions about pricing strategies, promotional campaigns, and user engagement tactics. This helps ensure that the system remains responsive to changing demands and maintains a high level of service quality.

Furthermore, the system’s scalability and performance optimization ensure that it can handle an increasing number of users without sacrificing speed or reliability. By leveraging cloud-based infrastructure and advanced backend technologies, the platform can scale up efficiently as demand grows, making it future-proof. The modular design of the system allows for easy updates and improvements, ensuring that the platform remains competitive in the ever-evolving transportation industry.

**4.2 FUTURE SCOPE**

The future scope of the Online Cab Booking System offers vast opportunities for further enhancements and expansion. One of the most exciting developments is the integration of electric vehicles (EVs) and autonomous vehicles (AVs), which could revolutionize the platform by providing eco-friendly and self-driving ride options, enhancing efficiency and reducing operational costs. Additionally, the use of Artificial Intelligence (AI) and Machine Learning (ML) can further personalize user experiences, optimize route matching, and adjust pricing dynamically based on real-time conditions, ultimately improving both passenger and driver satisfaction. As security remains a top priority, future improvements could include real-time facial recognition and smart wearables to enhance safety features, while in-ride video surveillance could provide added protection.

The system could also expand beyond taxis into a multi-modal transportation platform, where users can book shared rides, electric scooters, or even access public transportation options all through one app, providing more flexible and cost-effective mobility solutions. Moreover, the platform has the potential to reach rural areas and global markets, catering to underserved regions and offering localized solutions in international locations. On the sustainability front, the platform could integrate carbon footprint tracking, encourage carpooling, and enable eco-friendly ride options to reduce environmental impact.

Additionally, partnerships with smart city infrastructure could enable the system to integrate with real-time traffic data, providing more efficient and reliable service to users. Subscription models and loyalty programs could be introduced to reward frequent riders and drivers, fostering long-term engagement. As urban mobility continues to evolve, the future of the Online Cab Booking System is filled with possibilities for innovation, increased convenience, and improved user experience, positioning it as a leader in the future of transportation.

**APPENDIX A**

**(SOURCE CODE)**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List; // Explicit import for java.util.List

import java.util.Map;

class Cab {

String id;

String type;

double farePerKm;

boolean available;

public Cab(String id, String type, double farePerKm, boolean available) {

this.id = id;

this.type = type;

this.farePerKm = farePerKm;

this.available = available;

}

}

public class CabBookingSystemSwing {

private static Map<String, String> users = new HashMap<>();

private static Map<String, Cab> cabs = new HashMap<>();

private static List<String> bookings = new ArrayList<>(); // Corrected List declaration

private static JFrame frame;

private static String loggedInUser = null;

public static void main(String[] args) {

initializeCabs();

showMainMenu();

}

private static void initializeCabs() {

cabs.put("Cab1", new Cab("Cab1", "Sedan", 10.0, true));

cabs.put("Cab2", new Cab("Cab2", "SUV", 15.0, true));

cabs.put("Cab3", new Cab("Cab3", "Hatchback", 8.0, true));

}

private static void showMainMenu() {

frame = new JFrame("Online Cab Booking System");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(400, 300);

JLabel label = new JLabel("Welcome to Cab Booking System", SwingConstants.CENTER);

label.setFont(new Font("Arial", Font.BOLD, 16));

JButton registerButton = new JButton("Register");

JButton loginButton = new JButton("Login");

registerButton.addActionListener(e -> showRegisterScreen());

loginButton.addActionListener(e -> showLoginScreen());

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(3, 1, 10, 10));

panel.add(label);

panel.add(registerButton);

panel.add(loginButton);

frame.add(panel);

frame.setVisible(true);

}

private static void showRegisterScreen() {

frame.getContentPane().removeAll();

frame.repaint();

JLabel usernameLabel = new JLabel("Username:");

JTextField usernameField = new JTextField(20);

JLabel passwordLabel = new JLabel("Password:");

JPasswordField passwordField = new JPasswordField(20);

JButton registerButton = new JButton("Register");

JButton backButton = new JButton("Back");

registerButton.addActionListener(e -> {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

if (users.containsKey(username)) {

JOptionPane.showMessageDialog(frame, "Username already exists!", "Error", JOptionPane.ERROR\_MESSAGE);

} else {

users.put(username, password);

JOptionPane.showMessageDialog(frame, "Registration successful!");

showMainMenu();

}

});

backButton.addActionListener(e -> showMainMenu());

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 2, 10, 10));

panel.add(usernameLabel);

panel.add(usernameField);

panel.add(passwordLabel);

panel.add(passwordField);

panel.add(registerButton);

panel.add(backButton);

frame.add(panel);

frame.revalidate();

}

private static void showLoginScreen() {

frame.getContentPane().removeAll();

frame.repaint();

JLabel usernameLabel = new JLabel("Username:");

JTextField usernameField = new JTextField(20);

JLabel passwordLabel = new JLabel("Password:");

JPasswordField passwordField = new JPasswordField(20);

JButton loginButton = new JButton("Login");

JButton backButton = new JButton("Back");

loginButton.addActionListener(e -> {

String username = usernameField.getText();

String password = new String(passwordField.getPassword());

if (users.containsKey(username) && users.get(username).equals(password)) {

loggedInUser = username;

JOptionPane.showMessageDialog(frame, "Login successful!");

showUserMenu();

} else {

JOptionPane.showMessageDialog(frame, "Invalid credentials!", "Error", JOptionPane.ERROR\_MESSAGE);

}

});

backButton.addActionListener(e -> showMainMenu());

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 2, 10, 10));

panel.add(usernameLabel);

panel.add(usernameField);

panel.add(passwordLabel);

panel.add(passwordField);

panel.add(loginButton);

panel.add(backButton);

frame.add(panel);

frame.revalidate();

}

private static void showUserMenu() {

frame.getContentPane().removeAll();

frame.repaint();

JLabel label = new JLabel("Welcome, " + loggedInUser, SwingConstants.CENTER);

label.setFont(new Font("Arial", Font.BOLD, 16));

JButton viewCabsButton = new JButton("View Available Cabs");

JButton bookCabButton = new JButton("Book a Cab");

JButton logoutButton = new JButton("Logout");

viewCabsButton.addActionListener(e -> showAvailableCabs());

bookCabButton.addActionListener(e -> showCabBookingScreen());

logoutButton.addActionListener(e -> {

loggedInUser = null;

showMainMenu();

});

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(4, 1, 10, 10));

panel.add(label);

panel.add(viewCabsButton);

panel.add(bookCabButton);

panel.add(logoutButton);

frame.add(panel);

frame.revalidate();

}

private static void showAvailableCabs() {

StringBuilder availableCabs = new StringBuilder("Available Cabs:\n");

for (Cab cab : cabs.values()) {

if (cab.available) {

availableCabs.append(String.format("%s: %s - ₹%.2f per km\n", cab.id, cab.type, cab.farePerKm));

}

}

JOptionPane.showMessageDialog(frame, availableCabs.toString(), "Available Cabs", JOptionPane.INFORMATION\_MESSAGE);

}

private static void showCabBookingScreen() {

JTextField cabIdField = new JTextField(10);

JTextField distanceField = new JTextField(10);

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(3, 2, 10, 10));

panel.add(new JLabel("Cab ID:"));

panel.add(cabIdField);

panel.add(new JLabel("Distance (km):"));

panel.add(distanceField);

int result = JOptionPane.showConfirmDialog(frame, panel, "Book a Cab", JOptionPane.OK\_CANCEL\_OPTION);

if (result == JOptionPane.OK\_OPTION) {

String cabId = cabIdField.getText();

String distanceStr = distanceField.getText();

if (cabs.containsKey(cabId) && cabs.get(cabId).available) {

double distance = Double.parseDouble(distanceStr);

Cab cab = cabs.get(cabId);

double fare = distance \* cab.farePerKm;

cab.available = false;

bookings.add(String.format("User: %s, Cab: %s, Fare: ₹%.2f", loggedInUser, cab.type, fare));

JOptionPane.showMessageDialog(frame, "Booking successful! Total fare: ₹" + fare);

} else {

JOptionPane.showMessageDialog(frame, "Invalid Cab ID or Cab is not available!", "Error", JOptionPane.ERROR\_MESSAGE);

}

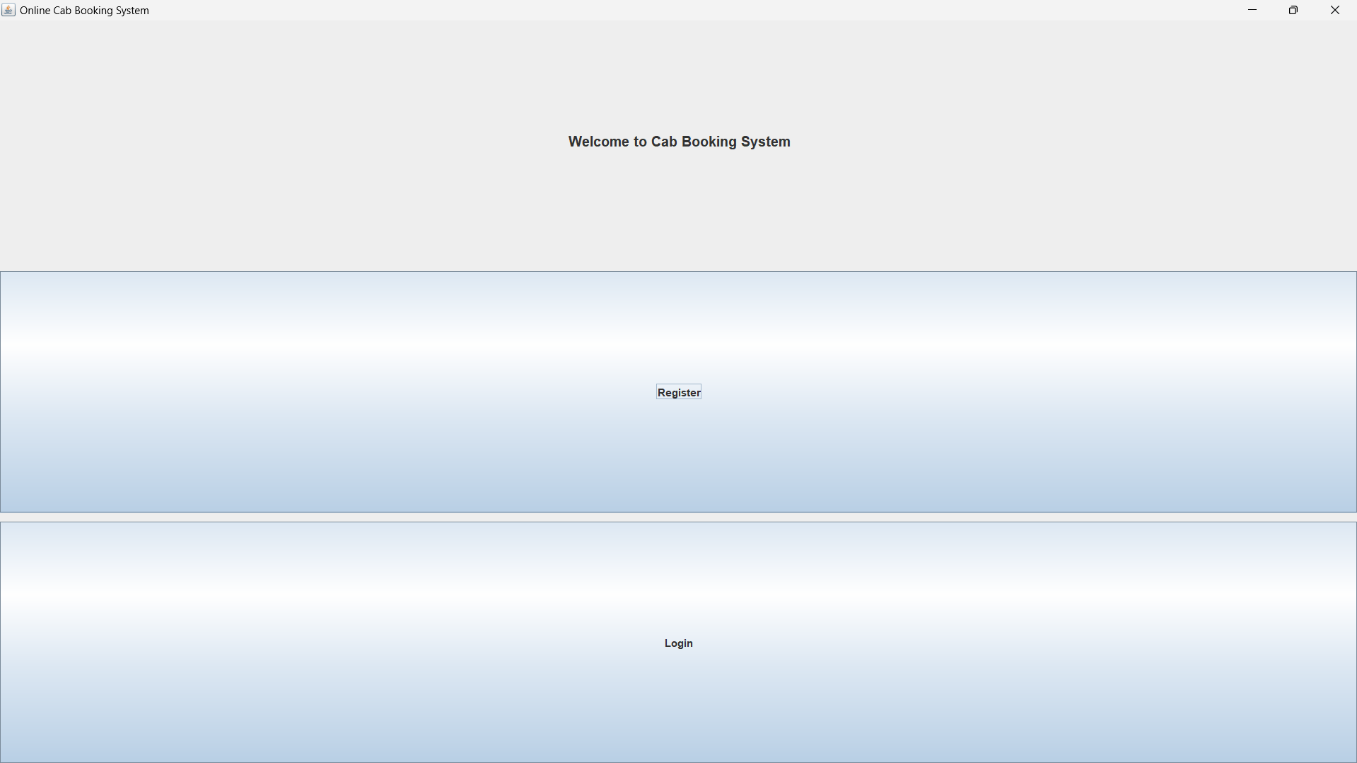
}

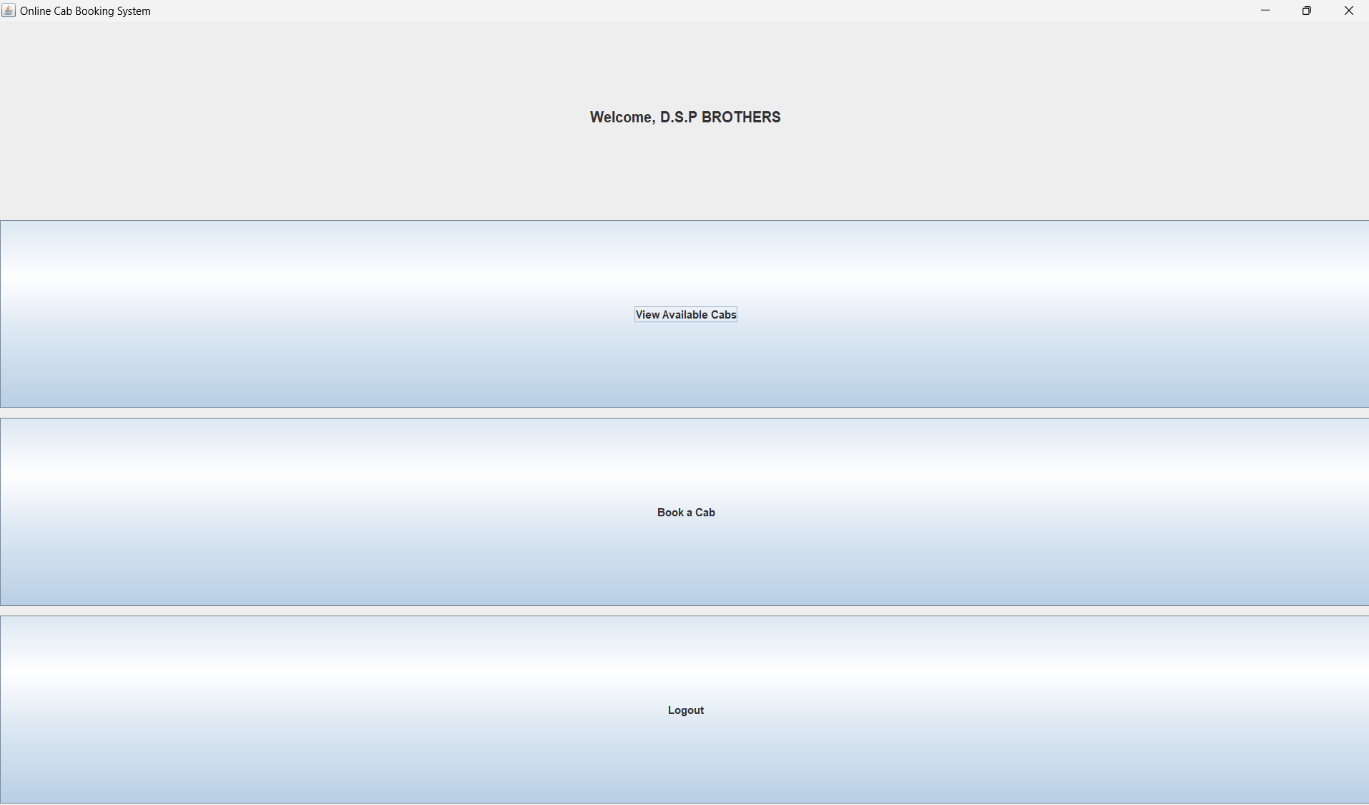
}

}

**APPENDIX B**

**(SCREENSHOTS)**

****Online Cab Booking System Interface

Online Cab Booking System Homepage

**REFERENCES**

1. Oracle Java Documentation: <https://docs.oracle.com/en/java/>
2. W3Schools - Java Tutorial: https://www.w3schools.com/java/
3. GeeksforGeeks Java Programming: https://www.geeksforgeeks.org/java/
4. Stack Overflow Java Community: <https://stackoverflow.com/questions/tagged/java>
5. Telusko YouTube Channel: <https://www.youtube.com/user/javaboynavin>