Siliguri Institute Of Technology

PROJ-CS881

Ride Share hub

BY

CSE\_PROJ\_2024\_15

|  |  |
| --- | --- |
| Name of Students | Roll No. |
| 1. Rohan Jyoti Basu | 11900120053 |
| 1. Souvik Goswami | 11900120015 |
| 1. Subhangi Banerjee | 11900120034 |
| 1. Subhadeep Dey Sarkar | 11900120041 |

Under the Guidance

Of

Ast. Prof. Satadal Chakraborty

Submitted to the Department of Computer Science and Engineering in partial fulfilment of the requirements for the award of the degree Bachelor of Technology in Computer Science and Engineering.

Year of Submission: 2024



SILIGURI INSTITUTE OF TECHNOLOGY

P.O SUKNA, SILIGURI, DIST. DARJEELING, PIN: 734009

TEL. (0353)2778002/04, FAX: (0353)2778003­­­

**DECLARATION**

This is to certify that the Report entitled “Ride Share Hub” which is submitted by my team in partial fulfilment of the requirement for the degree of **Computer Science Engineering** at **Siliguri Institute Of Technology** under **Maulana Abul Kalam Azad University of Technology,** West Bengal. We took the help of other materials in our dissertation which have been properly acknowledged. This report has not been submitted to any other Institute for the award of any degree.

DATE:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Name of Student | Roll No | Signature |
| 1 | Rohan Jyoti Basu | 11900120053 |  |
| 2 | Souvik Goswami | 11900120015 |  |
| 3 | Subhangi Banerjee | 11900120034 |  |
| 4 | Subhadeep Dey Sarkar | 11900120041 |  |

**CERTIFICATE**

This is to certify that the project report entitled “Ride Share hub”, submitted to **Department of Computer Science and Engineering of Siliguri Institute Of Technology in** partial fulfilment of the requirement for the award of the degree of **Bachelor of technology in Computer Science & Engineering** during the academic year 2023-2024, is a bonafide record of the project work carried out by then under my guidance and supervision.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Group Number: 15** | | | |
| **SN** | **Name of Students** | **Registration No** | **Roll No** |
| 1 | Rohan Jyoti Basu | 110038 | 11900120053 |
| 2 | Souvik Goswami | 110076 | 11900120015 |
| 3 | Subhangi Banerjee | 110057 | 11900120034 |
| 4 | Subhadeep Dey Sarker | 110050 | 11900120041 |

………………………………… ……………………………….

Signature of Project Guide Signature of HOD

Name of the Guide: Department of Computer Science and Engineering

**ACKNOWLEDGEMENT**

The acknowledgement depicts the student’s gratitude, respect and thankfulness towards the people who helped them pursue the project successfully and ensured the successful completion and implementation of the project. On the page, the author express his gratitude and concern by praising and thanksgiving words.

Signature of all group members with date

1.

2.

3.

4.

**Table of Contents**

1. Introduction
   1. Project Overview
2. System Analysis
   1. Preliminary Investigation
   2. Feasibility Study
   3. Project Planning
   4. Project Scheduling
   5. Software Requirement Specifications(SRS)
   6. Software Engineering Paradigm Applied

.

1. System Design

3.1 Modularization

3.2 Data Integrity

3.3 Component Interaction

3.4 Security and Scalability

1. User Interface(UI)
2. CODING
3. TESTING

**Abstract**

The cab booking app is a cutting-edge solution designed to simplify ride booking and management for both passengers and drivers. Built with a robust technology stack including HTML, CSS, JavaScript, React.js, Next.js, Python, Flask, Node.js, and MySQL, the app delivers a seamless and efficient user experience through comprehensive integration of essential features.

Key functionalities of the app include secure user authentication, intuitive ride booking, precise price calculation, and real-time map integration. Users can easily register and log in, safeguarding their personal information. The ride booking process is straightforward, allowing users to enter pickup and drop-off locations and visualize their route on an interactive map powered by the Google Maps API.

Accurate fare estimation is provided based on distance, time, and factors like surge pricing, ensuring transparency for users. Additionally, the ride history feature allows users to review past rides, monitor expenses, and access receipts. For drivers, the app offers tools to manage ride requests, navigate to pickup locations, and track earnings, thereby enhancing their service delivery.

The backend, developed using Flask and Node.js, ensures scalability and efficiency. Flask handles core logic and API endpoints, while Node.js supports real-time communication and updates. MySQL securely stores user data, ride details, and transaction records.

The frontend, crafted with React.js and Next.js, offers a dynamic and responsive user interface. React.js enables the creation of reusable components, while Next.js provides server-side rendering for fast load times and improved SEO, enhancing the user experience.

Security is integral to the app's design, with JWT tokens used for secure authentication and integrated secure payment gateways for transactions. This ensures that user sessions and financial information are protected.

In summary, the cab booking app leverages modern web technologies to offer a reliable and efficient urban transportation solution. This documentation outlines the project's setup, architecture, and features, serving as a valuable guide for developers and users.

**Introduction**

* 1. Project Overview

The Ride Share Hub cab booking app is a comprehensive platform designed to facilitate easy and efficient transportation services for users. The app leverages modern web technologies and integrates various features to provide a seamless experience for both passengers and drivers.

**Brief Description:** The cab booking app allows users to book rides conveniently from their mobile or desktop devices. Users can register and log in, enter their pickup and drop-off locations, view available drivers on a map, and get an estimated fare for their ride. The app also includes driver functionalities, allowing drivers to manage ride requests and navigate to passengers' locations.

**Key Features:**

* **User Login:** Secure authentication system for users and drivers to log in and manage their profiles.
* **Price Calculation:** Dynamic fare estimation based on distance, time, and other factors such as surge pricing.
* **Maps Integration:** Real-time map display using Google Maps API for location selection, route visualization, and navigation.
* **Ride Booking:** Easy interface for users to book rides by entering pickup and drop-off points.
* **Driver Management:** Features for drivers to accept or decline ride requests and navigate to user locations.
* **Ride History:** Users can view their past rides and receipts.
* **Notifications:** Real-time notifications to keep users and drivers updated on ride status.
* **Payment Integration:** Secure payment options for users to pay for rides within the app.

This documentation will provide a detailed overview of the project's setup, architecture, and functionality, guiding you through the process of understanding and utilizing the app's features.

**System Analysis**

1. Preliminary Investigation

Purpose:

The preliminary investigation serves as the foundational step in determining the necessity and potential success of developing a cab booking app. The primary goals are to:

* **Assess the Need**: Evaluate the demand for a cab booking app that streamlines the process of booking taxis, ensuring it is easy and efficient for users.
* **Understand User Requirements**: Identify and understand the key needs and preferences of potential users, which include both passengers and drivers.
* **Identify Existing Solutions and Limitations**: Review existing cab booking services to identify gaps and areas for improvement that the new app can address.

Objectives:

To achieve the purpose outlined above, the preliminary investigation focuses on the following objectives:

1. Identify the Stakeholders:

* Passengers: Individuals who will use the app to book rides.
* Drivers: Individuals who will use the app to offer rides and manage their bookings.
* Administrators: Individuals who will manage the system, monitor performance, and handle any operational issues.

1. Gather Initial Requirements:

* Conduct surveys and interviews with potential users to collect their requirements and preferences.
* Perform market analysis to understand the competitive landscape and identify features that could differentiate the new app.

1. Determine Scope and Constraints:

* Define the boundaries of the project, including what will and will not be included in the initial version of the app.
* Identify any technical, financial, or legal constraints that could impact the project.

Outcome:

The preliminary investigation concludes with a comprehensive understanding of the project's goals and the problems it aims to solve. The key outcomes are:

* Project Goals: A clear definition of what the project aims to achieve, including the main features and functionalities of the app.
* Problem Statement: Identification of specific issues in the current cab booking market that the new app will address.
* Feasibility Report: A detailed report assessing whether the project is viable across various dimensions (technical, economic, operational, and legal). This report provides the basis for deciding whether to proceed to the next phase of the project.

**Emotional Analysis**

In the context of our cab booking app, the emotional analysis algorithm is a critical component designed to enhance user experience and improve service quality by interpreting the emotional tone conveyed in user feedback, reviews, and support interactions. This algorithm helps us understand passengers' and drivers' sentiments, enabling us to address concerns proactively, enhance satisfaction, and tailor our services more effectively.

The primary objectives of our emotional analysis algorithm include identifying specific emotions—such as satisfaction, frustration, happiness, and disappointment—expressed in user communications. By differentiating between positive, negative, and neutral sentiments, we can gain a deeper understanding of the user experience and identify areas for improvement. Additionally, the algorithm aims to accurately interpret the context of these emotions, handling nuances such as sarcasm, slang, and varying cultural expressions.

The algorithm's workflow begins with text preprocessing, which involves tokenizing user input, lemmatizing or stemming words to their base forms, and removing stop words that do not contribute significantly to emotional content. Feature extraction follows, utilizing lexical analysis (identifying emotional words from a dictionary), syntactic analysis (examining sentence structure), and semantic analysis (understanding the contextual meaning of words and phrases). These steps are crucial for accurately capturing the emotional tone of the feedback.

Model training is then conducted using supervised learning techniques on a labelled dataset of user reviews and feedback, where emotions are explicitly tagged. This is complemented by unsupervised learning to detect new patterns and deep learning approaches, such as recurrent neural networks (RNNs) or transformers, to handle complex language constructs. Once trained, the model classifies text into predefined emotional categories and assigns sentiment scores, indicating the intensity of the detected emotions.

Evaluation and validation are continuous processes to ensure the algorithm's effectiveness and accuracy. By implementing this emotional analysis algorithm, our cab booking app can better understand and respond to user needs, ultimately leading to improved service quality, enhanced customer satisfaction, and a more empathetic interaction between users and the app. This allows us to address user concerns promptly, celebrate positive experiences, and maintain a high standard of service in an increasingly competitive market.

**2.Feasibility Study**

**Economic Feasibility:**

Economic feasibility assesses the cost-effectiveness of developing and maintaining the cab booking app. It involves analyzing both the initial investment and ongoing operational costs against the potential financial benefits. Here's a breakdown:

Initial Costs:

* **Development Costs**: This includes expenses related to hiring developers, designers, and project managers. Using a combination of technologies like HTML, CSS, JavaScript, React.js, Next.js, Python, Flask, Node.js, and MySQL incurs development costs that might be higher due to the need for diverse expertise.
* **Infrastructure Costs**: Setting up servers, cloud services, and other necessary hardware and software infrastructure.
* **Licensing and Tools**: Purchasing any required licenses for development tools, APIs (like maps and authentication services), and other software dependencies.

Operational Costs:

* **Maintenance and Updates**: Regular updates, bug fixes, and adding new features will require ongoing investment.
* **Customer Support**: Implementing and maintaining a customer support system to handle user queries and issues.
* **Marketing and Advertising**: Costs associated with promoting the app to attract users, including digital marketing campaigns, social media promotions, and other advertising strategies.

Revenue Generation:

* **Commission from Rides**: A percentage cut from each ride booked through the app.
* **Advertising**: Revenue from advertisements displayed within the app.
* **Premium Services**: Offering premium features or subscription models to users.

Cost-Benefit Analysis:

If the projected revenue from commissions, advertising, and premium services outweighs the combined initial and operational costs, the project is economically feasible. Initial market research and user surveys can help estimate the potential user base and revenue, contributing to a positive return on investment (ROI).

**Schedule Feasibility:**

Schedule feasibility evaluates whether the cab booking app can be developed and launched within a reasonable timeframe. This involves detailed project planning and scheduling to ensure timely delivery. Here's a proposed timeline:

**Phase 1:** Planning and Requirements Gathering (1-2 months)

Conduct preliminary investigations, feasibility studies, and gather user requirements.

Identify stakeholders and define the project scope and constraints.

**Phase 2:** Design and Prototyping (2-3 months)

Develop wireframes, mockups, and prototypes for the app’s UI/UX.

Finalize the design and get approval from stakeholders.

**Phase 3:** Development (4-6 months)

Frontend Development: Implement the UI using HTML, CSS, JavaScript, React.js, and Next.js.

Backend Development: Set up the server-side using Python, Flask, and Node.js, and integrate the MySQL database.

API Integration: Integrate external APIs for maps, user authentication, and other necessary services.

**Phase 4:** Testing and Quality Assurance (2-3 months)

Conduct thorough testing, including unit tests, integration tests, and user acceptance testing (UAT).

Fix bugs and make necessary adjustments based on feedback.

**Phase 5:** Deployment and Launch (1 month)

Deploy the app on the chosen platforms (web, iOS, Android).

Monitor initial performance and user feedback, and make quick fixes if necessary.

**Phase 6:** Post-Launch Support and Maintenance (Ongoing)

Regular updates, new feature additions, and continuous monitoring for any issues.

**Overall Timeline:**

The total estimated timeline is approximately 10-15 months from planning to launch. This schedule can be adjusted based on the team's size, expertise, and any unforeseen challenges. Regular reviews and agile project management practices can help ensure the project stays on track and within the proposed timeline.

**Conclusion:**

With a detailed project plan and an efficient development team, the cab booking app is both economically and schedule feasible. The potential revenue generation opportunities and the structured timeline ensure that the project can be completed within a reasonable timeframe and budget, leading to a successful launch and sustainable operation.

Project Planning

Project Goals and Objectives

Goals:

* Develop a user-friendly cab booking app that facilitates easy and efficient taxi bookings.
* Enhance user experience through seamless navigation, real-time tracking, and secure payment options.
* Provide a reliable platform for drivers to connect with passengers and manage their rides effectively.
* Achieve a competitive edge in the market by offering unique features and superior customer service.

Objectives:

* Implement a robust user authentication system for both passengers and drivers.
* Integrate a dynamic pricing algorithm to calculate ride costs based on distance, time, and demand.
* Embed interactive maps for real-time tracking and route planning.
* Ensure the app is scalable and secure, protecting user data and ensuring smooth performance.
* Launch the app within the proposed timeline and budget, achieving key performance metrics in user acquisition and retention.

Project Scope

In-Scope:

* Development of frontend and backend components using technologies like HTML, CSS, JavaScript, React.js, Next.js, Python, Flask, Node.js, and MySQL.
* Integration of external APIs for maps, user authentication, and payment processing.
* Creation of user interfaces for login, registration, ride booking, and ride management.
* Implementation of features like price calculation, ride tracking, user reviews, and notifications.
* Testing phases including unit testing, integration testing, and user acceptance testing (UAT).

Out-of-Scope:

* Development of features for markets or regions not initially targeted.
* Implementation of advanced AI features beyond the initial scope, such as predictive analytics for ride demand.
* Support for legacy devices or outdated operating systems.

Project Phases and Milestones

**Phase 1:** Planning and Requirements Gathering (1-2 months)

Milestones:

* Completion of preliminary investigation and feasibility study.
* Identification of stakeholders and gathering of initial requirements.
* Finalization of project scope and constraints.

**Phase 2:** Design and Prototyping (2-3 months)

Milestones:

* Development of wireframes and mockups.
* Approval of UI/UX design.
* Creation of prototypes for key functionalities.

**Phase 3:** Development (4-6 months)

Milestones:

* Completion of frontend development using HTML, CSS, JavaScript, React.js, and Next.js.
* Completion of backend development using Python, Flask, Node.js, and MySQL.
* Successful integration of external APIs for maps and user authentication.

**Phase 4:** Testing and Quality Assurance (2-3 months)

Milestones:

* Execution of unit tests and integration tests.
* Completion of user acceptance testing (UAT).
* Resolution of identified bugs and issues.

**Phase 5:** Deployment and Launch (1 month)

Milestones:

* Deployment of the app on web, iOS, and Android platforms.
* Initial user feedback collection and quick fixes.
* Formal launch of the app.

**Phase 6:** Post-Launch Support and Maintenance (Ongoing)

Milestones:

* Regular updates and new feature releases.
* Continuous monitoring and performance optimization.
* Ongoing customer support and issue resolution.

Overall Milestone Timeline:

Month 1-2: Completion of planning and requirements gathering.

Month 3-5: Completion of design and prototyping.

Month 6-11: Completion of development phase.

Month 12-14: Completion of testing and quality assurance.

Month 15: Deployment and initial launch.

Ongoing: Post-launch support and maintenance.

By adhering to this detailed project plan, the development team can ensure the successful creation and launch of a high-quality cab booking app, meeting the defined goals and objectives within the stipulated timeline and budget.

**Project Scheduling:**

Project scheduling involves defining the timeline for completing various tasks and milestones throughout the development of the cab booking app. Here is a brief overview of the project schedule, including key phases and their corresponding timelines.

**Phase 1:** Planning and Requirements Gathering (1-2 months)

Tasks:

* Conduct preliminary investigation and feasibility study.
* Identify stakeholders (passengers, drivers, administrators).
* Gather initial requirements through surveys, interviews, and market analysis.
* Finalize project scope and constraints.

Milestones:

* Completion of feasibility report.
* Approval of project scope and initial requirements.

Timeline:

* Month 1-2

**Phase 2:** Design and Prototyping (2-3 months)

Tasks:

* Develop wireframes and mockups for the app's user interface.
* Design the overall architecture and data flow.
* Create prototypes for key functionalities such as user login, ride booking, and price calculation.

Milestones:

* Approval of UI/UX design.
* Completion of initial prototypes.

Timeline:

* Month 3-5

**Phase 3:** Development (4-6 months)

Tasks:

Frontend Development:

* HTML, CSS, and JavaScript coding for static pages.
* Development using React.js and Next.js for dynamic content.

Backend Development:

* Set up Python, Flask, and Node.js environments.
* Develop APIs and integrate with MySQL database.

API Integration:

* Integrate external APIs for maps, user authentication, and payment processing.

Milestones:

* Completion of frontend development.
* Completion of backend development and database integration.
* Successful integration of external APIs.

Timeline:

* Month 6-11

**Phase 4:** Testing and Quality Assurance (2-3 months)

Tasks:

Unit Testing:

* Test individual components for correct functionality.

Integration Testing:

* Ensure that different components work together seamlessly.

User Acceptance Testing (UAT):

* Gather feedback from a small group of end-users.
* Identify and fix bugs and issues.

Milestones:

* Successful completion of unit and integration testing.
* Positive feedback from UAT and resolution of identified issues.

Timeline:

* Month 12-14

**Phase 5:** Deployment and Launch (1 month)

Tasks:

* Deploy the app on web, iOS, and Android platforms.
* Perform final pre-launch checks and optimizations.
* Launch the app to the public.

Milestones:

* Successful deployment of the app.
* Formal launch and initial user acquisition.

Timeline:

* Month 15

Phase 6: Post-Launch Support and Maintenance (Ongoing)

Tasks:

* Monitor app performance and user feedback.
* Release regular updates and new features.
* Provide continuous customer support and issue resolution.

Milestones:

* Regular updates and improvements based on user feedback.
* High user satisfaction and retention rates.

Timeline:

* Ongoing

**Summary Schedule**

* Month 1-2: Planning and Requirements Gathering
* Month 3-5: Design and Prototyping
* Month 6-11: Development
* Month 12-14: Testing and Quality Assurance
* Month 15: Deployment and Launch
* Ongoing: Post-Launch Support and Maintenance

This project schedule ensures a structured approach to developing the cab booking app, with clear timelines and milestones for each phase. By adhering to this schedule, the development team can manage time effectively, ensuring timely delivery of a high-quality product.

**Software Requirement Specification(SRS)**

The Software Requirement Specifications (SRS) document serves as a detailed guide outlining the functional and non-functional requirements for the Ride-Share-Hub cab booking application. This document ensures that all stakeholders have a clear understanding of the system's functionalities and constraints.

1. Introduction

1.1 Purpose:

The purpose of this SRS document is to provide a detailed description of the functionalities, performance, and constraints of the Ride-Share-Hub cab booking application. It serves as a contract between the project stakeholders and the development team.

1.2 Scope:

The Ride-Share-Hub app facilitates easy and efficient booking of taxis, targeting passengers and drivers. Key features include user login, ride price calculation, maps integration for route planning, and secure payment processing.

1.3 Definitions, Acronyms, and Abbreviations:

* SRS: Software Requirement Specifications
* API: Application Programming Interface
* UI: User Interface
* DB: Database

2. Overall Description

2.1 Product Perspective:

The Ride-Share-Hub app is a standalone system developed to provide a reliable cab booking service. It integrates with external APIs for maps and payment processing and interfaces with a MySQL database for data storage.

2.2 Product Functions:

* User Registration and Login: Allows users to create accounts and log in securely.
* Ride Booking: Enables users to book rides, select pick-up and drop-off locations.
* Price Calculation: Calculates ride prices based on distance, time, and traffic conditions.
* Maps Integration: Provides route planning and real-time tracking of rides.
* Payment Processing: Facilitates secure payments via various methods.
* Driver Management: Manages driver profiles, availability, and ratings.
* Customer Support: Offers support features like contact forms and feedback systems.

2.3 User Characteristics:

* Passengers: Individuals seeking transportation services.
* Drivers: Individuals offering transportation services.
* Administrators: Staff managing the system and handling support requests.

2.4 Constraints:

* The system must ensure data privacy and security.
* The app should be compatible with major web browsers and mobile platforms (iOS and Android).
* Integration with reliable third-party APIs for maps and payment processing.

2.5 Assumptions and Dependencies:

* Users have access to the internet.
* External APIs for maps and payments are available and operational.
* The MySQL database is accessible and operational.

3. Functional Requirements

3.1 User Registration and Login:

* Users can register with their username, email, password, and other personal details.
* Users can log in using their credentials.
* Password recovery and reset functionality.

3.2 Ride Booking:

* Users can select pick-up and drop-off locations on a map.
* Users can view estimated prices before confirming a booking.
* Real-time availability of drivers is displayed.

3.3 Price Calculation:

* Dynamic pricing based on distance, time, and current traffic conditions.
* Display of estimated fare breakdown.

3.4 Maps Integration:

* Route planning from pick-up to drop-off location.
* Real-time tracking of the ride.

3.5 Payment Processing:

* Integration with payment gateways for secure transactions.
* Support for multiple payment methods (credit/debit cards, digital wallets).

3.6 Driver Management:

* Drivers can register and manage their profiles.
* Availability status management for drivers.
* Rating and feedback system for drivers.

3.7 Customer Support:

* Contact forms for queries and complaints.
* FAQ section and help guides.

4. Non-Functional Requirements

4.1 Performance:

* The app should load within 3 seconds on standard internet connections.
* The system should handle concurrent users without performance degradation.

4.2 Security:

* User data must be encrypted during transmission and storage.
* Secure authentication mechanisms to prevent unauthorized access.

4.3 Usability:

* The user interface should be intuitive and easy to navigate.
* Consistent user experience across different devices and platforms.

4.4 Reliability:

* The system should have an uptime of 99.9%.
* Regular backups of the database to prevent data loss.

4.5 Maintainability:

* The codebase should be modular to facilitate easy updates and maintenance.
* Comprehensive documentation for developers and administrators.

5. Software Engineering Paradigm Applied

The development of the Ride-Share-Hub app will follow the Agile software development methodology. This approach allows for iterative progress through continuous feedback and improvements. Agile facilitates adaptive planning and fosters a flexible response to changes, ensuring the delivery of a high-quality product.

This SRS document provides a comprehensive overview of the requirements for developing the Ride-Share-Hub cab booking application, ensuring that all stakeholders have a clear understanding of the system's functionalities and constraints.

**Software Engineering Paradigm Applied**

**Agile Software Engineering Paradigm**

The Agile software engineering paradigm is a popular and effective approach for developing the Ride-Share-Hub cab booking app. Agile focuses on iterative development, flexibility, and collaboration, which are crucial for addressing the dynamic requirements of modern software projects.

Key Aspects of Agile Applied to Ride-Share-Hub:

**Iterative Development:**

* Frequent Releases: The project is divided into small, manageable iterations (sprints), each lasting 2-4 weeks. Each sprint delivers a potentially shippable product increment.
* Continuous Improvement: Feedback from stakeholders (passengers, drivers, administrators) is continuously integrated into the development process, allowing for quick adjustments and improvements.

**Customer Collaboration:**

* Stakeholder Involvement: Regular meetings with stakeholders ensure that their needs and expectations are clearly understood and addressed. This includes gathering feedback from passengers and drivers to enhance user experience.
* User Stories: Requirements are captured as user stories, which are concise descriptions of a feature from the perspective of the end user.

**Flexibility and Adaptability:**

* Responsive to Change: Agile embraces changes in requirements, even late in development. This is particularly important for the cab booking app, where market demands and user needs can evolve rapidly.
* Prioritization: Features are prioritized based on their value to users and the business, ensuring that the most critical functionalities are developed first.

**Cross-functional Teams:**

* Collaborative Environment: Development teams consist of members with diverse skills (developers, testers, designers) working collaboratively. This promotes knowledge sharing and faster problem-solving.
* Daily Stand-ups: Short daily meetings help the team stay aligned on progress and quickly address any obstacles.

**Test-Driven Development (TDD):**

* Quality Assurance: Tests are written before code, ensuring that the code meets the required functionality from the start. Continuous integration tools are used to run automated tests, maintaining high code quality.

**Continuous Integration and Deployment (CI/CD):**

* Automated Workflows: Code changes are automatically integrated, tested, and deployed to staging or production environments. This reduces the risk of integration issues and accelerates the release cycle.

**Application to Ride-Share-Hub:**

* Project Planning: Initial planning focuses on defining the project scope, user stories, and prioritizing features.
* Sprints: Each sprint aims to develop specific functionalities such as user registration, cab booking, payment integration, and feedback systems.
* Review and Retrospective: At the end of each sprint, the team reviews the completed work with stakeholders and conducts a retrospective to identify areas for improvement.
* Product Backlog: A dynamic list of features and enhancements that are continuously updated based on stakeholder feedback and market trends.

By applying the Agile software engineering paradigm, the Ride-Share-Hub project can ensure timely delivery of high-quality software that meets user needs and adapts quickly to changing requirements. This approach fosters collaboration, enhances flexibility, and focuses on delivering value throughout the development process.

**System Design**

The system design of the Ride-Share-Hub cab booking app focuses on creating a reliable, scalable, and user-friendly platform with a modular approach and strong data integrity measures.

High-Level Architecture

1. Frontend (Client-Side):

* Technologies: HTML, CSS, JavaScript, React.js, Next.js
* Components:
  + User interface for login, registration, booking, and payments.
  + Interactive maps for displaying cabs and routes.
  + Real-time notifications for updates.

1. Backend (Server-Side):

* Technologies: Python, Flask, Node.js
* Components:
  + Authentication service for user login and registration.
  + Booking management for ride requests and assignments.
  + Price calculation service.
  + Integration with external APIs for maps and payments.

1. Database:

* Technology: MySQL
* Schemas: User information, ride details, driver data, feedback, and support tickets.

**Modularization**

The system is divided into distinct modules to ensure maintainability, scalability, and ease of development:

1. User Management Module:

* Handles user registration, authentication, and profile management.
* Ensures secure storage of user credentials.

1. Booking Management Module:

* Manages the lifecycle of ride bookings from request to completion.
* Interfaces with the maps module to provide real-time location tracking.

1. Pricing Module:

* Calculates ride prices based on distance, time, and dynamic factors like demand.

1. Maps Integration Module:

* Integrates with external APIs to fetch map data, display routes, and track driver locations.

1. Payment Processing Module:

* Handles secure transactions through integrated payment gateways.
* Supports multiple payment methods.

1. Feedback and Support Module:

* Allows users to submit feedback and support requests. Provides administrative interfaces for managing feedback and support tickets.

**Data Integrity**

To ensure data integrity, the following measures are implemented:

1. Database Constraints:

* Use of primary keys, foreign keys, and unique constraints to maintain relational integrity.
* Implementation of proper indexing to enhance query performance.

1. Transaction Management:

* Use of database transactions to ensure atomicity, consistency, isolation, and durability (ACID properties).
* Ensures that all operations within a transaction are completed successfully before committing changes.

1. Data Validation:

* Both client-side and server-side validation to ensure data accuracy.
* Sanitization of user inputs to prevent SQL injection and other attacks.

1. Regular Backups:

* Scheduled database backups to prevent data loss.
* Implementation of recovery procedures in case of data corruption or hardware failure.

1. Audit Logs:

* Maintaining logs of critical operations to track changes and detect anomalies.
* Ensures accountability and traceability of data modifications.

**Security and Scalability**

* Security: Secure data transmission (HTTPS), encryption for sensitive data, regular security audits.
* Scalability: Modular design for easy scaling, load balancing, distributed servers, and optimized database indexing.

This system design ensures Ride-Share-Hub delivers a robust, secure, and efficient cab booking experience for its users.

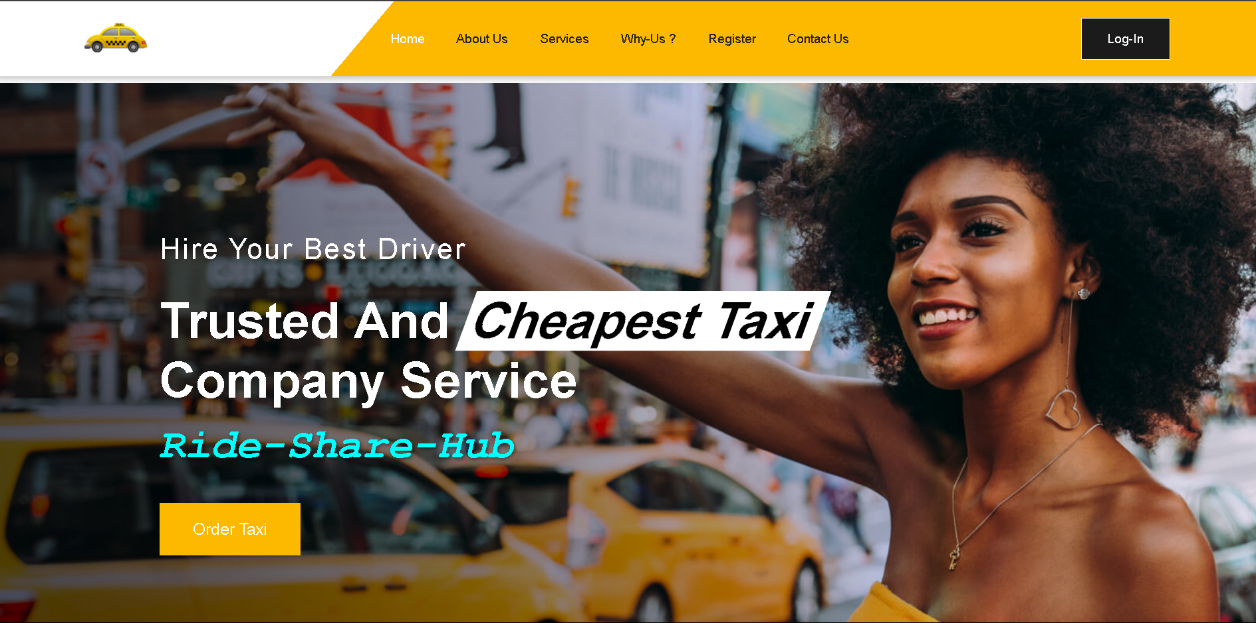
**User Interface(UI)**

Overview

The User Interface (UI) of the Ride-Share-Hub cab booking application is designed to provide an intuitive and seamless experience for users. This document outlines the key components, layout, and functionality of the UI, ensuring that all stakeholders have a clear understanding of how users will interact with the application.

Main Components

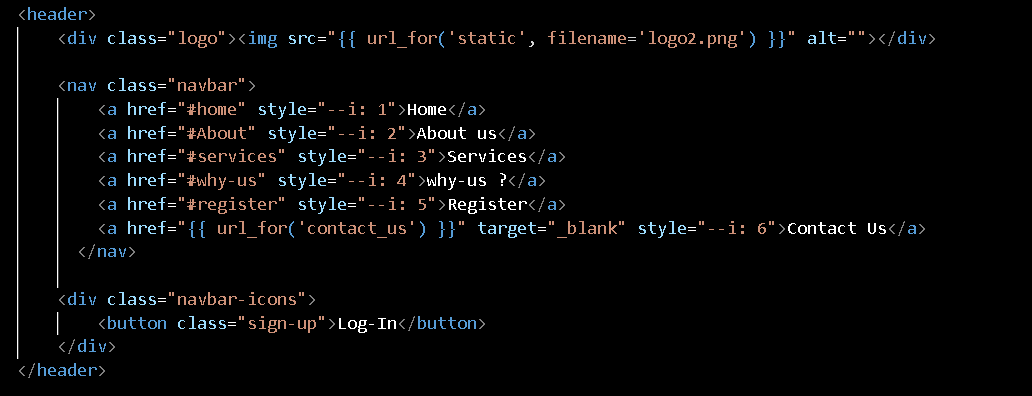
* Header
* Home Section
* About Us Section
* Services Section
* Why Choose Us Section
* Register Section
* Footer



Header

Components:

* **Logo:** Positioned at the top-left, displaying the company logo.
* **Navigation Bar:** Contains links to different sections of the website, including Home, About Us, Services, Why Us, Register, and Contact Us.
* **Login Button:** A "Log-In" button for user authentication.



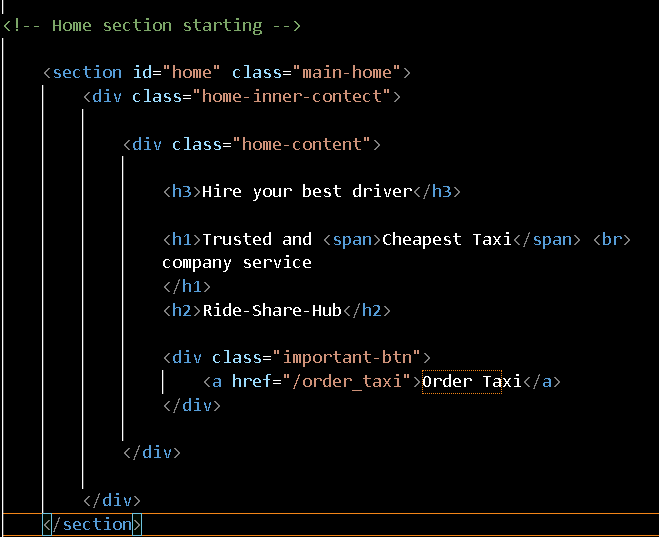
Home Section

Purpose:

* Introduces users to the service.
* Provides a quick link to order a taxi.

Components:

* Main Banner: Contains a welcoming message and a brief introduction.
* Order Taxi Button: A prominent button directing users to book a taxi.



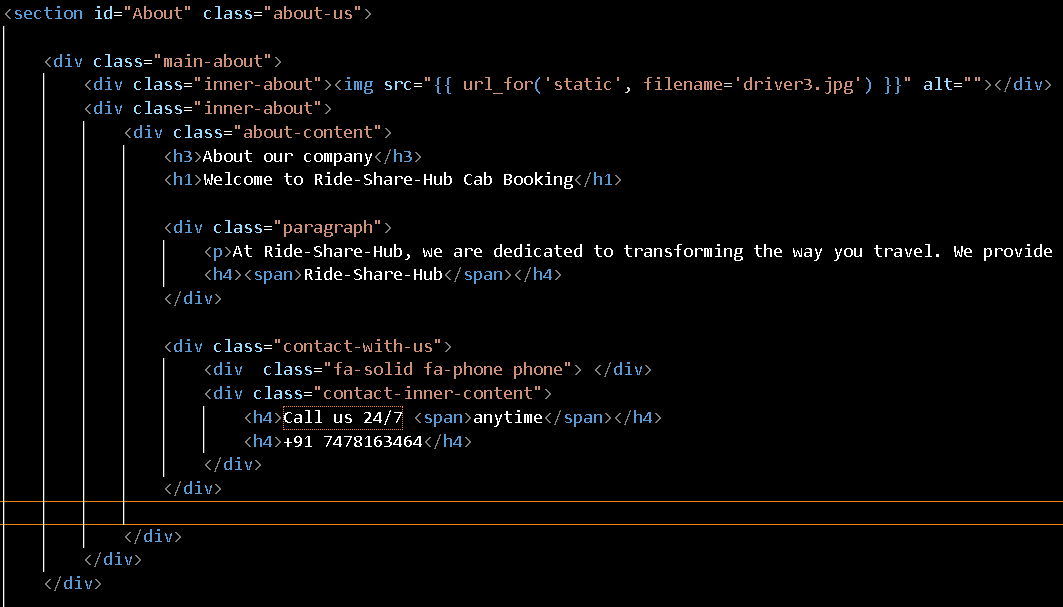
About Us Section

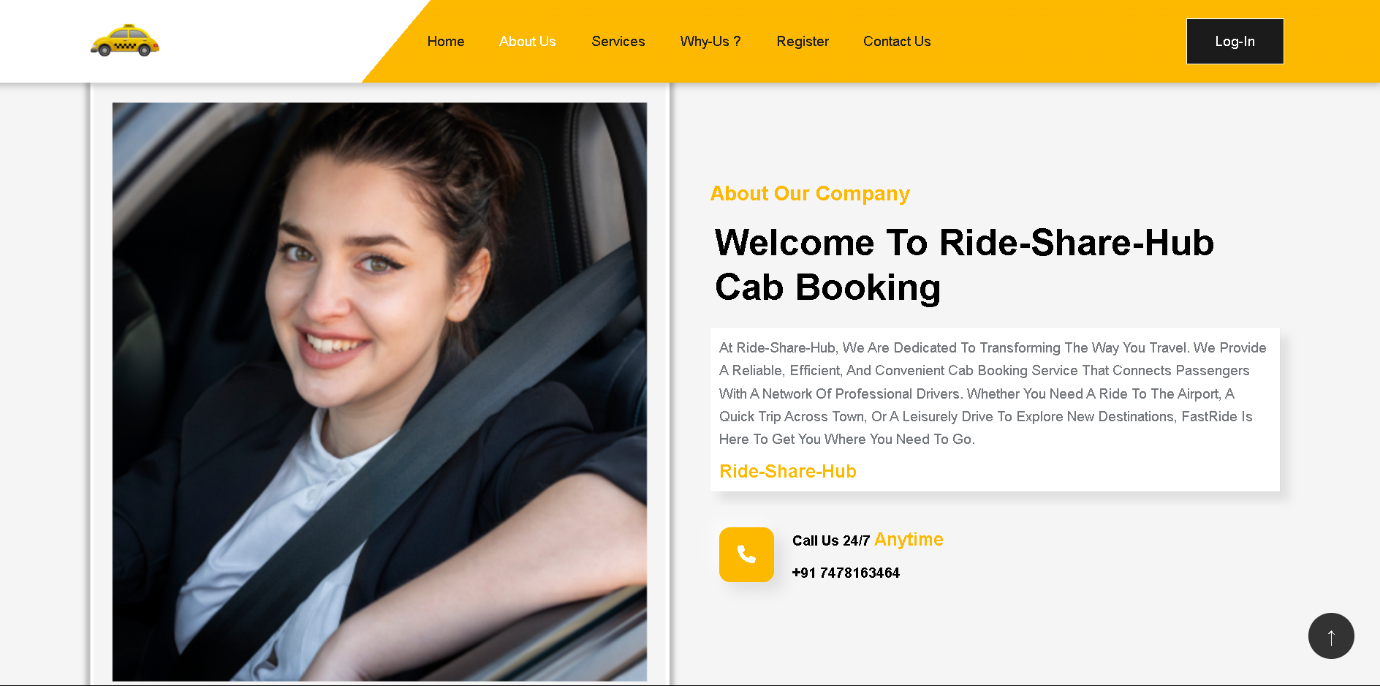
Purpose:

* Provides an overview of the company.
* Highlights the company's mission and values.

Components:

* Image: Visual representation of the service.
* Text Content: Describes the company's goals and customer service commitment.





Register Section

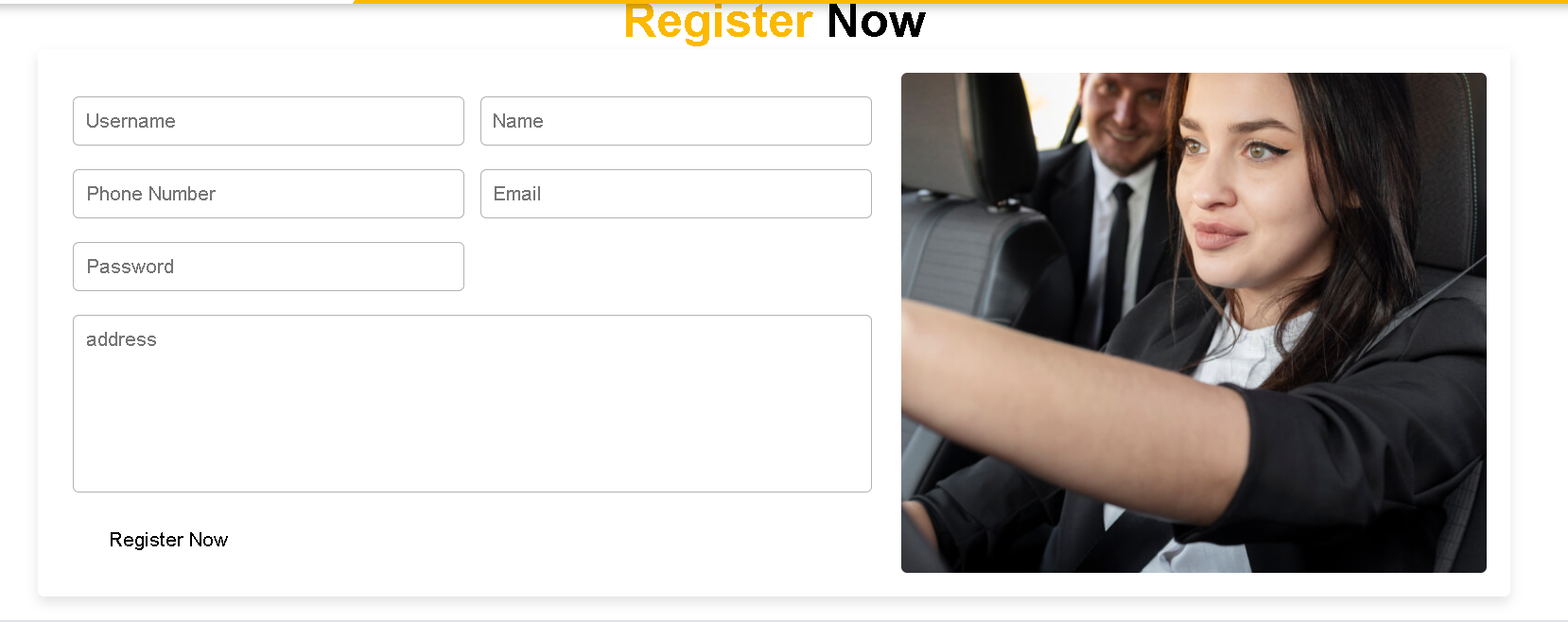
Purpose:

* Allows new users to register for the service.

Components:

* Registration Form: Includes fields for username, full name, phone number, email, password, and address.





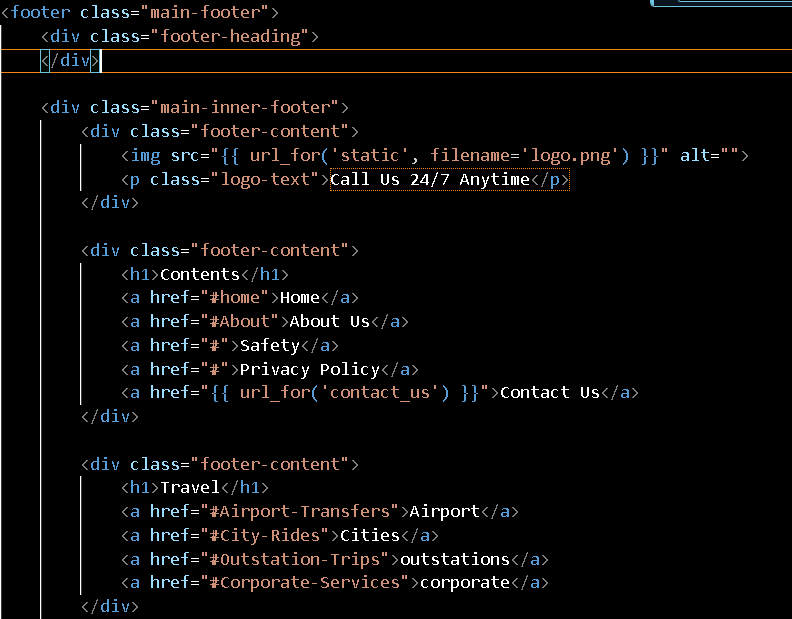
Footer

Purpose:

* Provides additional navigation and contact information.

Components:

* Company Information: Includes logo and contact details.
* Links: Quick links to important sections of the website.
* Social Media: Icons linking to social media profiles.

**CODING**

**Flask Application**

This document provides an overview of a Flask-based web application for a ride-sharing service called "Ride-Share-Hub". The application includes functionality for user registration, login, contacting customer support, and ordering a taxi. The backend is powered by MySQL.

**Application Structure**

1. Configuration and Initialization

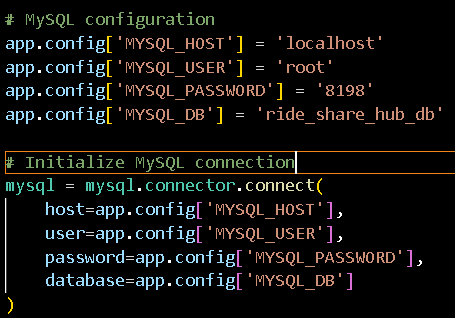
The application is initialized and configured to connect to a MySQL database using the mysql.connector library. Configuration parameters for the database connection are stored in the application's configuration settings.

2. Routes

The application defines several routes to handle different user interactions, including rendering templates, handling form submissions, and inserting data into the database.

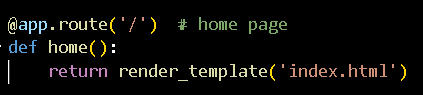
3. Database Connection

A connection to the MySQL database is established using the credentials provided in the configuration. The mysql.connector.connect function is used to create this connection.

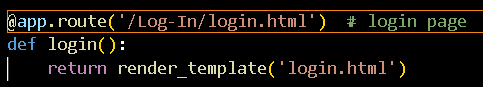


**Routes and Their Functions**

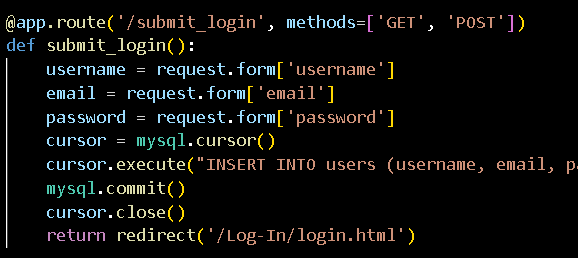
Home Page Route



Login Page Route



Submit Login Route



This route handles the submission of the login form. It extracts the username, email, and password from the form, inserts them into the users table in the database, commits the transaction, and redirects the user back to the login page.

**Conclusion**

This Flask application provides a robust framework for a ride-sharing service with features such as user registration, login, contact form submissions, and ordering taxis. The application leverages MySQL for data storage and retrieval, ensuring a reliable and scalable backend.

**TESTING**

This document outlines the testing procedures for the Ride-Share-Hub web application. It includes test plans for different components of the application to ensure functionality, usability, and performance.

Introduction

The purpose of this document is to provide a comprehensive guide for testing the Ride-Share-Hub application. The goal is to identify and fix any defects or issues before the application goes live.

Test Environment

* Operating System: Windows 10 / macOS / Linux
* Browsers: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
* Devices: Desktop, Mobile, Tablet
* Database: MySQL
* Framework: Flask

Testing Strategy

The testing strategy includes:

* Unit Testing: Testing individual components and functions.
* Integration Testing: Ensuring that different components work together.
* Functional Testing: Validating the application against the functional requirements.
* Usability Testing: Ensuring the application is user-friendly.
* Performance Testing: Assessing the application's performance under different conditions.
* Security Testing: Identifying and fixing security vulnerabilities.

Performance Testing

Test Case 10: Load Testing

* Objective: Assess the application's performance under load.
* Steps:
  + Simulate multiple users accessing the application simultaneously.
  + Monitor response times and resource usage.
  + Expected Result: The application maintains acceptable performance levels under load.

Security Testing

Test Case 11: SQL Injection

* Objective: Ensure that the application is protected against SQL injection attacks.
* Steps:
  + Attempt to enter SQL injection code into form fields.
  + Submit the form.

Expected Result: The application should sanitize input and prevent SQL injection.

Test Case 12: Cross-Site Scripting (XSS)

* Objective: Ensure that the application is protected against XSS attacks.
* Steps:
  + Attempt to enter malicious scripts into form fields.
  + Submit the form.
  + Expected Result: The application should sanitize input and prevent XSS.

Conclusion

This testing documentation provides a structured approach to verify the functionality, usability, performance, and security of the Ride-Share-Hub application. Following this plan will help ensure the application meets its requirements and provides a reliable and secure experience for users.