Online cab booking System

Overview:

* The Online Cab Booking System is a web-based platform that enables your clients to book taxis and executive taxis from the convenience of their own home or workplace. The platform should have an administrative interface via which the taxi business can control the content as well as access all reservations and customer data.

1.Introduction to Online Cab Booking System

In the rapidly evolving landscape of urban mobility, the need for efficient, reliable, and convenient transportation solutions has become more critical than ever. The online cab booking system has emerged as a transformative technology that meets this demand by leveraging the power of the internet and mobile devices to connect passengers with drivers seamlessly. This innovative platform not only revolutionizes the way people travel within cities but also enhances the overall transportation experience through real-time tracking, cashless payments, and user-centric features.

An online cab booking system simplifies the process of hailing a cab by allowing users to book rides through a smartphone application or a web interface. This system is designed to cater to the diverse needs of riders and drivers, offering a range of functionalities that ensure safety, efficiency, and transparency. For riders, the convenience of booking a cab from anywhere, at any time, with accurate fare estimates and route tracking, significantly enhances their travel experience. For drivers, the system provides a steady stream of ride requests, efficient navigation tools, and a platform to manage their earnings and feedback.

The backbone of an online cab booking system includes advanced technologies such as GPS for real-time location tracking, secure payment gateways for seamless transactions, and robust server-side infrastructure to handle a large volume of concurrent users. Additionally, the system incorporates features such as driver and passenger ratings, customer support, and emergency assistance to ensure a high level of service quality and safety.

The development and deployment of an online cab booking system involve multiple stages, from understanding user requirements and designing an intuitive interface to implementing robust backend services and performing rigorous testing. By integrating modern technologies and focusing on user experience, such a system not only meets the transportation needs of today’s urban dwellers but also sets the stage for future advancements in smart mobility solutions.

In essence, an online cab booking system is a pivotal innovation in the transportation sector, driving forward the concept of smart cities and connected services. It exemplifies how technology can be harnessed to improve everyday life, offering a glimpse into the future of urban transportation where efficiency, convenience, and sustainability are at the forefront.

**1.Functional Requirements:**

Users should be able to see all the cabs available with minimum price and ETA

Users should be able to book a cab for their destination

Users should be able to see the location of the driver

Users should be able to cancel their ride whenever they want

**2.Non-Functional requirements**

1. **Performance**:
   * **Response Time**: The system should respond to user actions within a predefined time limit to ensure a smooth user experience.
   * **Scalability**: The system should be able to handle a growing number of users and requests without significant degradation in performance.
   * **Throughput**: The system should be capable of processing a large number of concurrent transactions efficiently.
2. **Reliability**:
   * **Availability**: The system should be available for use 24/7 with minimal downtime for maintenance or unexpected failures.
   * **Fault Tolerance**: The system should be resilient to failures and should continue to function properly even if certain components fail.
   * **Data Integrity**: Ensure that data is accurate and consistent throughout the system.
3. **Security**:
   * **Authentication and Authorization**: Users should only have access to functionalities and data appropriate for their role.
   * **Data Encryption**: Sensitive information such as user details, payment information, and location data should be encrypted during transmission and storage.
   * **Protection against Attacks**: The system should be protected against common security threats such as SQL injection, cross-site scripting (XSS), and DDoS attacks.
4. **Usability**:
   * **User Interface Design**: The user interface should be intuitive and easy to navigate, allowing users to book a cab with minimal effort.
   * **Accessibility**: The system should be accessible to users with disabilities, following accessibility standards such as WCAG (Web Content Accessibility Guidelines).
5. **Scalability**:
   * **Vertical Scalability**: The system should be capable of handling increased load by adding more resources such as CPU, memory, or storage to a single server.
   * **Horizontal Scalability**: The system should be able to scale horizontally by adding more servers to distribute the load.
6. **Maintainability**:
   * **Modifiability**: The system should be designed in a modular way, allowing for easy updates and modifications to accommodate changing business requirements.
   * **Documentation**: The system should be well-documented to facilitate maintenance and troubleshooting by developers and administrators.
7. **Compliance**:
   * **Regulatory Compliance**: The system should comply with relevant laws and regulations governing data privacy, security, and transportation services.
   * **Industry Standards**: Adhere to industry standards and best practices in software development, security, and data handling.
8. **Performance**:
   * **Response Time**: The system should respond quickly to user requests, ensuring minimal waiting time for booking confirmations and updates.
   * **Resource Utilization**: The system should efficiently utilize hardware resources such as CPU, memory, and network bandwidth to maximize performance and minimize operating costs.
9. **Compatibility**:
   * **Cross-platform Compatibility**: The system should be compatible with a wide range of devices and operating systems, including desktops, laptops, smartphones, and tablets.
   * **Browser Compatibility**: The system should work seamlessly across different web browsers such as Chrome, Firefox, Safari, and Edge.

**System Architecture:**

Frontend :react js or Angular

Backend: Node.js with Express.js or Python with Django.

Database: PostgreSQL or MySQL.

**Gherkin features for Online cab booking**

**Feature: Online Cab Booking System**

Scenario 1: User Registration

Given the user is on the registration screen

When the user enters valid registration details

And clicks on the "Register" button

Then the system sends a verification email and OTP

When the user verifies the email and OTP

Then the system confirms registration and redirects to login screen

Scenario 2: User Login

Given the user is on the login screen

When the user enters valid login credentials

And clicks on the "Login" button

Then the system authenticates the user and redirects to home screen

Scenario 3: Booking a Ride

Given the user is logged into the passenger app

And is on the ride booking screen

When the user enters pickup and drop-off locations

And selects "Standard" ride type

And confirms the booking

Then the system shows estimated fare and driver details

And notifies the user of successful booking

Scenario 4: Making a Payment

Given the user has completed a ride

When the user selects "Credit Card" payment method

And enters credit card details

And clicks on the "Pay" button

Then the system processes the payment

And shows a payment confirmation with receipt

Scenario 5: Viewing Ride History

Given the user is logged into the passenger app

When the user navigates to the "Ride History" screen

Then the system displays a list of past rides with details

Scenario 6: User Logout

Given the user is logged into the passenger app

When the user clicks on the "Logout" button

Then the system logs the user out and redirects to login screen

Scenario 7: Driver Registration

Given the user has installed the driver app

And is on the driver registration screen

When the user enters valid registration details and uploads documents

And clicks on the "Register" button

Then the system sends a verification email and OTP

When the user verifies the email and OTP

Then the system confirms registration and redirects to login screen

Scenario 8: Driver Accepting a Ride Request

Given the driver is logged into the driver app

And is online

When a ride request is sent to the driver

Then the driver receives the request notification with details

When the driver accepts the request

Then the system notifies the passenger and shows the driver’s location

Scenario 9: Driver Completing a Ride

Given the driver has picked up the passenger

When the driver reaches the drop-off location

And marks the ride as completed

Then the system notifies the passenger of ride completion

And calculates the final fare

And prompts the passenger to make a payment

Scenario 10: Passenger Rating a Driver

Given the user has completed a ride

When the user rates the driver and submits a review

Then the system saves the rating and review

And updates the driver's overall rating

Scenario 11: Fare Estimation without Booking

Given the user is logged into the passenger app

And is on the ride booking screen

When the user enters pickup and drop-off locations

And requests a fare estimate

Then the system shows an estimated fare

And provides an option to confirm or cancel the booking

Scenario 12: Cancelling a Ride Before Confirmation

Given the user is logged into the passenger app

And is on the ride booking screen

When the user enters pickup and drop-off locations

And decides to cancel the booking

Then the system does not proceed with the booking

And returns the user to the home screen