A

Project Report

On

Peer-to-Peer Ride Sharing Services using Blockchain

By

Divyansh Jaiswal (1901330100105)

Karan Kanojia (1901330100132)

Kartik Anand (1901330100133)

Abhishek Kumar (2001330109002)

Under the Supervision of

Ms. DeepShikha

(Assistant Professor CSE)

Submitted to the department of Computer Science and Engineering

For the partial fulfillment of the requirements

For award of Bachelor of Technology

In

Computer Science and Engineering

Noida Institute of Engineering & Technology Gr. Noida

Affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India

CERTIFICATE

This is to certify that the Minor Project report entitled "**Peer-to-Peer Ride Sharing Services using Block chain**" is a record of the work done by the following students:

Student Name:	**Roll Number:**
Divyansh Jaiswal	1901330100105
Karan Kanojia	1901330100132
Kartik Anand	1901330100133
Abhishek Kumar	2001330109002

This work is completed under our supervision and steering for the duration of the educational yr of 2022-2023. This record is submitted to the NOIDA INSTITUTE OF ENGINEERING & TECHNOLOGY, GREATER NOIDA, for partial fulfilment for the degree of **B.TECH** (Computer Science and Engineering) of **Dr. A.P.J. Abdul Kalam Technical University(AKTU), Lucknow, Uttar Pradesh, India.**

We desire him all the pleasant for all of the endeavors.

Signature of Guide:

Faculty Name: Ms. DeepShikha
Designation: Assistant Professor (CSE)

ACKNOWLEDGMENT

I am very grateful to **Ms**. **DeepShikha,** Associate Professor, Department of Computer Science & Engineering (CSE). **NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY, GREATER NOIDA**, Gautam Budha Nagar, Uttar Pradesh for her generous guidance, useful suggestions and all project related help.

I express my gratitude to **Prof. Kumud Saxen, HOD (CSE),** Noida Institute of Engineering and Technology, Greater Noida, for his encouraging guidance, encouragement and constant supervision during the present work.

Date:

Students Name & Roll No.

Divyansh Jaiswal (1901330100105)
Karan Kanojia (1901330100132)
Kartik Anand (1901330100133)
Abhishek Kumar (2001330109002)

Abstract

The emergence of peer-to-peer (P2P) journey-sharing offerings has revolutionized the way human beings commute, tough traditional transportation models and providing an progressive method to deal with the demanding situations of city mobility. This undertaking pursuits to expand and put into effect a P2P ride-sharing provider platform that connects individuals seeking transportation with fellow users who're willing to percentage their automobile and provide a journey.

The assignment makes a specialty of leveraging the strength of modern-day technology, particularly cell applications and cloud-primarily based platforms, to facilitate green and convenient ride-sharing experiences. Users can check in at the platform, create profiles, and specify their journey preferences, which include experience origin, vacation spot, and favored time of tour. The platform's algorithms in shape riders with suitable drivers, considering elements along with proximity, direction compatibility, and user scores.

To ensure protection and protection, the platform carries robust verification mechanisms, including motive force license verification, car registration tests, and person ratings and critiques. moreover, complete coverage coverage is furnished to guard each drivers and riders throughout the journey-sharing process.

The venture additionally explores the mixing of advanced features, which include actual-time GPS monitoring, payment structures, and person comments mechanisms, to enhance the general person revel in and promote believe and transparency within the network.

The development of this P2P experience-sharing carrier no longer best pursuits to provide users with a convenient and value-effective alternative to standard transportation strategies however also contributes to lowering site visitors congestion, carbon emissions, and overall environmental effect. by promoting collaborative consumption and optimizing the utilization of personal motors, the platform encourages a more sustainable and efficient use of current assets.

The success of this mission is predicated on thorough marketplace research, effective technological implementation, and strategic partnerships with applicable stakeholders, inclusive of transportation authorities, coverage providers, and network corporations. by using bridging the space among transportation deliver and demand via peer collaboration, this P2P journey-sharing provider has the capacity to convert urban mobility and create a extra connected and sustainable destiny for transportation.

TABLE OF CONTENTS

INDEX OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background:

Peer Pool is inspired by corporate giants like Uber and Ola and how it centralizes all public transactions between driver and passenger identities. Uber and Ola prevent pairing of users with drivers and collect 25% of the total amount paid by the passenger to the driver. We want to remove the middlemen (Uber and Ola) and create a new app that will have all the features of Uber and decentralize all operations of the ride-sharing platform. Peer Pool solves the problem by removing devices and decentralizing user data and keeping it inaccessible to anyone but themselves.

Today, taxi service units use a centralized methodology to run their day-to-day operations. Additionally, booking a taxi requires an intermediary or third-party business to complete the payment process. When more parties are involved, this lack of transparency becomes problematic.

1.2 Main Problem in previous design:

Challenges in the Ride-Sharing Industry

1. **Cost problems due to middlemen:** taxi booking requires payment processing from third party companies, vehicle tracking etc. demands. Each of these intermediaries will charge a large fee per transaction. This not only increases the cost of passengers but also reduces the wages of drivers.

2. **Limited user Base:** The previous design had a constrained consumer base, resulting in a lack of to be had drivers and riders at diverse times and places.
   Limited participation decreased the effectiveness of the provider and created longer wait instances for customers.
3. **Inefficient Matching Algorithms:** The matching algorithms used in the preceding layout were no longer optimized for green pairing of drivers and riders.

Lack of attention for elements inclusive of proximity, path compatibility, and consumer preferences led to suboptimal suits, growing travel time and inconvenience.

4. **Insufficient protection and security measures:** The previous layout lacked sturdy safety and security measures, posing risks for both drivers and riders.
   Inadequate driver verification techniques, automobile exams, and user remarks mechanisms undermined believe and compromised person safety.
5. **Restricted Integration of advanced capabilities:** The previous design did now not properly comprise advanced functions along with actual-time GPS tracking, fee structures, and person comments mechanisms.
   This obstacle led to a much less seamless user revel in, making it tough to music rides, manage bills, and provide comments.
6. **Insufficient Partnerships and Stakeholder Engagement:** The previous design did not set up strong partnerships with applicable stakeholders, consisting of transportation authorities and coverage vendors.
   Loss of collaboration hindered the platform's potential to deal with regulatory requirements, make certain right insurance coverage, and get right of entry to essential sources.
7. **Neglecting Sustainability and Environmental impact:**
   The preceding design did now not sufficiently deal with the environmental effect of transportation.
   Failing to sell sustainable practices and inspire shared vehicle utilization constrained the capacity for lowering site visitor’s congestion and carbon emissions.
8. **Constrained marketplace research:** The preceding design lacked comprehensive market studies to apprehend consumer wishes, preferences, and market dynamics.
   Insufficient market insights hindered the platform's capacity to tailor its offerings efficiently and attract a larger person base.
9. **Insufficient consumer enjoy:** The preceding layout did no longer prioritize improving the general consumer enjoy.
   Difficulties in navigation, lack of user-friendly interfaces, and lack of personalized features faded user pride and hindered platform adoption.

Addressing those essential troubles in the new layout will cause a extra effective and a hit peer-to-peer trip-sharing provider.

1.3 Project description:

Peer-to-Peer sharing is a project that aims to build a sharing service by decentralizing transactions in an application and ensuring that no single entity manages transactions or data related to those transactions. In addition, the project aims to eliminate third party services, guarantee customer privacy and ensure fair and accurate prices, and with the help of smart contracts, many tasks can be automated and practical crypto created. number that can be used to exchange shared services between the customer and the driver.

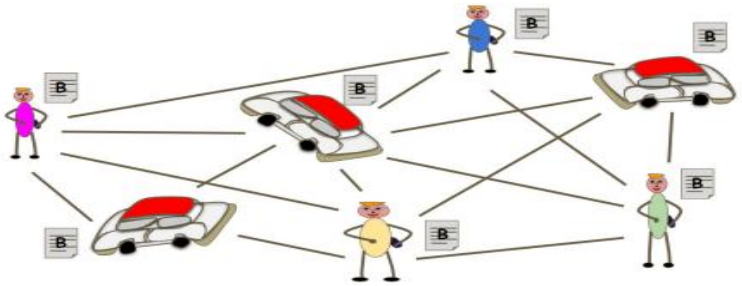


Figure 1.1: The system with decentralized trust

1.4 Motivation:

The goal of decentralization is what drives the search for blockchain in the first place, as blockchain allows for peer-to-peer validation of transactions to create a community image. Any malicious activity can be easily observed, and for a device to be malicious, at least 51% of the friends in the community must be malicious. This leads to a snapshot of all drivers and riders connected to their peers to view the network, and the rides recorded in the blockchain.

Motivation to put in force a Peer-to-Peer ride-Sharing carrier undertaking:

1. **Addressing Transportation challenges**: the motivation behind the assignment lies in addressing the urgent challenges confronted in traditional transportation systems, along with site visitors congestion, constrained parking, and environmental pollution. via promoting a peer-to-peer journey-sharing carrier, the mission aims to offer an progressive solution that optimizes present sources and reduces the general burden on transportation infrastructure.

2. **Fee-effective opportunity:** Peer-to-peer experience-sharing gives a price-effective alternative to traditional transportation methods. by means of connecting individuals with to be had seats of their cars to those in need of a journey, the venture allows value sharing and reduces person commuting prices. This thing is mainly attractive to finances-aware people and people seeking inexpensive transportation options.

3. **Network building and Collaboration:** The challenge fosters a sense of network and collaboration with the aid of connecting riders and drivers who proportion commonplace journey routes and destinations. It encourages social interplay, enables networking, and builds believe among people. Peer-to-peer journey-sharing promotes a way of life of sharing and cooperation, that may make stronger social bonds and enhance network concord.

4. **Environmental effect and Sustainability:** A key motivation for enforcing a peer-to-peer trip-sharing service is to reduce the environmental impact of transportation. via encouraging the sharing of motors, the project pursuits to lower the range of motors on the road, main to a discount in carbon emissions, visitors congestion, and fuel consumption. This attention on sustainability aligns with the developing worldwide recognition of the need for transportation alternatives.

5. **Technological improvements:** The challenge leverages advancements in mobile packages, GPS tracking, and cloud-based platforms to create an green and user-pleasant journey-sharing revel in. the motivation lies in harnessing the electricity of generation to overcome the restrictions of conventional transportation systems and provide customers with a seamless and convenient manner to connect and proportion rides.

6. **Improving Mobility and Accessibility:** Peer-to-peer trip-sharing improves mobility and accessibility, in particular in areas with restricted public transportation alternatives. by connecting riders and drivers in real-time, the challenge ambitions to bridge transportation gaps, facilitate final-mile connectivity, and decorate normal mobility alternatives for people who may additionally face demanding situations getting access to reliable transportation services.

7. **Tremendous Social effect:** implementing a peer-to-peer journey-sharing provider will have a nice social impact by using promoting inclusivity and lowering transportation disparities. The challenge objectives to offer transportation options for folks that may not have get admission to to personal cars or who face limitations in utilizing public transportation. It fosters a sense of independence and empowers people to fulfill their transportation desires greater effectively.

All in all, the motivation behind the peer-to-peer ride-sharing service project stems from the desire to create a more efficient, sustainable and community-oriented transportation system that solves the problems of traditional models while improving accessibility, reducing costs, and minimizing impact on the environment

.

CHAPTER 2
OBJECTIVE

The primary objective of a peer-to-peer experience-sharing gadget is to connect people who want a trip with those who can offer transportation offerings. This sort of machine targets to facilitate handy, green, and fee-powerful transportation options via leveraging the sharing financial system and utilizing to be had assets inside a community.

Right here are the important thing targets of a peer to peer ride journey sharing system:

1. **Efficient resource usage:** The device pursuits to optimize the usage of current transportation assets by means of connecting drivers who've available seats in their motors with passengers who want a journey in the equal course. This allows reduce empty seats in motors and maximizes the performance of transportation ability.

2. **Cost reduction:** via sharing rides, both drivers and passengers can doubtlessly lessen their transportation costs. Drivers can earn additional income by means of monetizing their empty seats, while passengers can find greater cheap transportation alternatives in comparison to standard taxis or vehicle offerings.

3. **Convenience and Accessibility:** The machine goals to offer convenient and reachable transportation options for users. Passengers can request rides via a cell app or internet site and get matched with to be had drivers in their vicinity. This gives extra flexibility and accessibility compared to traditional transportation techniques.

4. **Community constructing and Social interplay:** Peer-to-peer ride-sharing systems foster community engagement by way of connecting people within a neighborhood place. through sharing rides, customers have the possibility to fulfill and engage with new humans, selling social connections and a feel of belonging.

5. **Environmental Sustainability:** by using lowering the quantity of man or woman cars on the street via ride sharing, peer-to-peer experience-sharing systems make a

contribution to reducing site visitor's congestion and greenhouse gasoline emissions.

The principle objective of a peer-to-peer trip-sharing machine is to create a platform that facilitates the sharing of transportation assets, offers price-powerful options for customers, and promotes performance, comfort, and sustainability within the transportation sector.

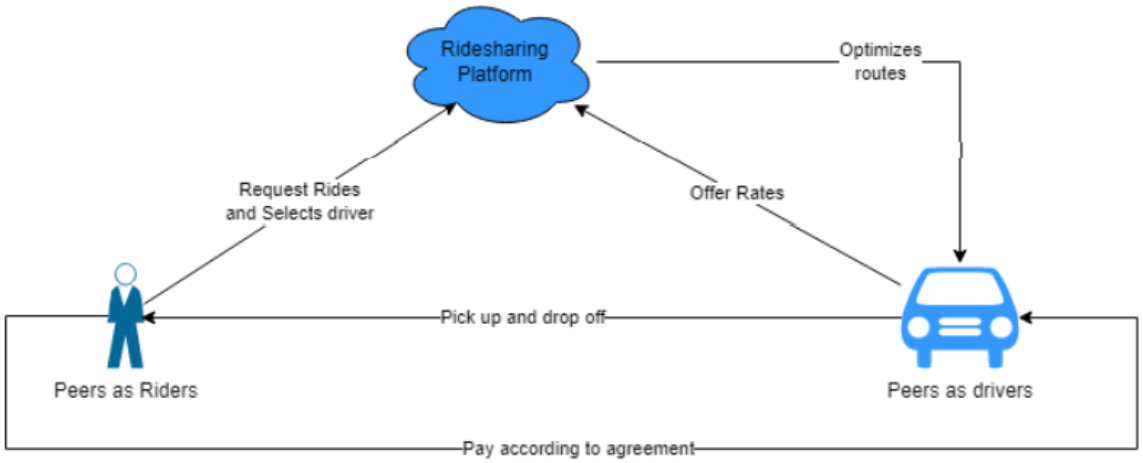


Fig: Structure of peer t peer ride sharing service

CHAPTER 3
LITERATURE REVIEW

3.1 Literature Review of Peer-to-Peer Ride-Sharing Service Project:

Here are the seven most popular Peer-to-Peer Ride Sharing Architecture proposals that are available for people.

1. **Block-5** uses an internal reputation gadget. To ensure the equity of the experience, the whole lot associated with the ride is permanently recorded inside the ledger, making it reachable to all and sundry in the peer-to-peer network. It prevents identification robbery with the aid of having a separate card for every account, and when a complaint is made, the government can affirm it through checking the ledger.

2. **Block-VN** distributed-transport-system-architecture that explores the transportation systems evolve and paradigms. The transport department sends the details to the cancellation office every time the vehicle registration is issued. Mining organizations then report all information about normal and mining nodes to the distributed blockchain.

3. **B-Ride** introduces a reputation model that rate the drivers based on past actions, allows drivers to make choices based on a collections of driver- interaction. Verification is done with zero proof of knowledge to protect the privacy of the driver / driver. To ensure fair payment, a fee-for-service philosophy is offered.

4. **Green Ride** promotes social responsibility through decentralization, which facilitates the reduction of carbon emissions. It consists of two structures; Centralized code to be hosted on Google Cloud App Engine and decentralized GRTs. This allows businesses, academies and government agencies to reduce their annual carbon footprint.

5. **PEBERS** - Ethereum based browsing service. Each user is assigned a unique ID that seeks to ride in the notebook using the fogging mechanism. Data is stored with a decentralized-distributed-ledger-technology. The protocol provides data consistency.

6. **O-Ride**, a privacy-preserving system, optimizes SHE to reduce bandwidth requirements and overhead through ciphertext wrapping and modified processing. Credit card payments include features such as contacting the driver if items are lost and tracking in case of criminal activity.

7. **Ride Matcher** is an architecture that creates online sharing groups instead of using a central database to find available rides for customers. In this program, one point solves and executes the task; find the point where the direction is right.

3.2 Challenges in the Ride-Sharing Industry:

The peer-to-peer experience-sharing industry faces numerous demanding situations that effect its operations, boom, and sustainability. those demanding situations include:

1. Consider and safety: believe and safety are great worries within the peer-to-peer journey-sharing industry. each drivers and passengers need to sense assured inside the reliability and safety of the platform. ensuring history assessments, identity verification, and implementing safety features are crucial to deal with those concerns.

2. Regulatory Compliance: Peer-to-peer experience-sharing frequently operates in a regulatory grey region, as it disrupts conventional transportation fashions. Compliance with neighborhood guidelines, lets in, coverage requirements, and licensing may be complex and vary across distinct jurisdictions. groups on this industry must navigate these guidelines to function legally and keep away from capacity felony issues.

3. Insurance coverage: coverage for peer-to-peer experience-sharing poses demanding situations. Conventional private automobile insurance rules might not cover injuries or incidents that arise at the same time as a automobile is being used for business functions. Ridesharing systems regularly provide insurance coverage, however there can be gaps in insurance, leaving drivers and passengers at chance.

4. Pricing and Profitability: attaining a sustainable pricing version and profitability is a challenge for peer-to-peer journey-sharing businesses. Balancing aggressive pricing for

passengers while making sure drivers earn a honest income is essential. organizations want to bear in mind elements along with motive force earnings, fee charges, incentives, and call for fluctuations to preserve a profitable enterprise version.

5. Driving force and Rider Acquisition and Retention: Attracting and preserving a enough quantity of drivers and riders is essential for the fulfillment of a peer-to-peer ride-sharing platform. making sure a consistent supply of to be had drivers to meet rider demand and maintaining riders engaged and dependable require powerful driving force and rider acquisition techniques, incentives, and customer service.

6. Scalability and Operations: Scaling operations to handle growing person demand can be challenging. Peer-to-peer journey-sharing platforms want to effectively manage and match motive force deliver with passenger demand, ensuring a stability to decrease wait times and maximize provider availability.

7. Competition and market Saturation: The peer-to-peer ride-sharing marketplace is distinctly competitive, with multiple players vying for market percentage. multiplied competition can lead to fee wars, reduced profitability, and demanding situations in differentiating offerings. moreover, in mature markets, saturation could make it tough for new entrants to advantage a foothold.

8. Exploitation of personnel by companies: Ridesharing groups are presently facing a labor dilemma, which stems from the truth that their courting with employees is exploitative. companies are dealing with severa proceedings over their labor practices. because the whole device is centralized, labor legal guidelines are enacted only through corporate executives, with little regard for real employees.

9. Social and moral problems: The peer-to-peer journey-sharing enterprise has raised issues about labor rights, worker classification, and income stability for drivers. Addressing these social and moral problems is important for lengthy-term sustainability and maintaining a nice public perception of the enterprise.

10. **Value issues due to intermediaries:** Require third-party companies to order taxis, process payments, track vehicles, and more. Each of those intermediaries will earn a large fee

by going through the transaction. This can not only increase the cost for passengers, but also reduce the profit of the driver.

Navigating those challenges requires modern answers, powerful law, and continuous model to evolving market dynamics. Peer-to-peer journey-sharing companies should prioritize safety, accept as true with, regulatory compliance, and fair repayment for drivers whilst delivering a seamless and dependable provider to passengers.

3.3 What PEER TO PEER RIDE SHARING actually is?

The peer-to-peer ride-sharing service allows customers to make first contact with neighboring homeowners through radio-based interaction and comprehensively search for rides. Simply put, Peer-to-peer transportation services allow drivers (peer-to-peer) to provide transportation to those who need a ride (peer-to-peer) using private vehicles.

Peer-to-peer networks have different hosts, such as private cars and public transport. These various actions affect contracts and ultimately travel opportunities. in addition to the environmental advantages of reducing the wide variety of vehicles and the amount of fuel wished, this type of system opens up new financial opportunities for automobile owners.

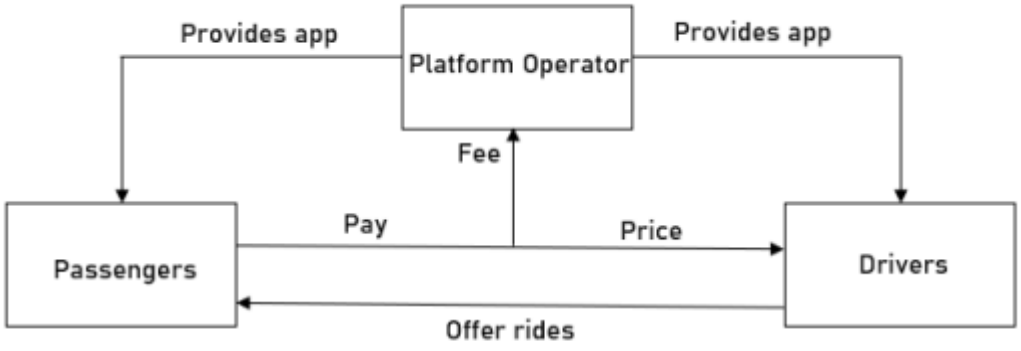


Figure 1- Peer-to-Peer Ride-Sharing Process

3.4 MERITS AND DEMERITS OF BLOCKCHAIN BASED SYSTEM

The future of blockchain is very good and shiny . Although it works as a method to solve the problems faced by contemporary ride-sharing devices, it also has some risks.

3.4.1 Merits

Peer-to-peer trip-sharing structures provide numerous merits and advantages for both drivers and passengers. Here are some of the key benefits of peer to peer journey sharing:

1. Price savings: Peer-to-peer trip-sharing can offer value-effective transportation options for passengers. It often offers lower fares compared to traditional taxis or vehicle services, making it an lower priced alternative for ordinary commuting or occasional rides.

2. Comfort and Accessibility: Peer-to-peer journey-sharing structures are commonly accessible thru mobile apps or web sites, presenting a handy and person-friendly interface for passengers to request rides. This ease of access and the capability to fast book a ride make it a convenient transportation choice, mainly in urban areas.

3. Extended Transportation alternatives: Peer-to-peer trip-sharing expands the variety of transportation alternatives to be had to passengers. It gives flexibility in phrases of car kinds, such as fashionable sedans, luxurious cars, or shared rides. Passengers can pick out the option that first-class suits their wishes and possibilities.

4. Decreased Congestion and Emissions: by way of encouraging journey-sharing and lowering the variety of man or woman cars on the road, peer-to-peer experience-sharing structures contribute to reducing site visitors congestion and reducing carbon emissions. this could have tremendous environmental impacts and sell a more sustainable transportation version.

5. Improved usage of assets: Peer-to-peer trip-sharing optimizes the utilization of current transportation sources. by using filling empty seats in automobiles, it maximizes the

performance of car ability and decreases inefficiencies related to man or woman vehicle ownership.

6. Community constructing and Social interaction: Peer-to-peer experience-sharing systems provide possibilities for social interplay and network engagement. Passengers and drivers have the hazard to meet new human beings, share experiences, and foster connections inside their nearby groups.

7. Additional earnings for Drivers: Peer-to-peer journey-sharing lets in individuals with a automobile to earn extra income by means of offering rides to passengers. this could be mainly useful for drivers who've spare time and need to monetize their available resources.

8. Remarks and rating systems: Peer-to-peer trip-sharing systems often incorporate feedback and score structures, allowing passengers to price drivers and offer remarks based totally on their experience. This enables hold best standards, incentivizes properly provider, and builds accept as true with within the network.

9. A consensus-based totally control tool provides a permanent and recognizable blockchain document that may make sure against double-spending with out entrusting itself to middlemen in a decentralized layout. This element creates a framework for economic specialties in the ITS surroundings.

10. It guarantees that a malicious rider can't make numerous proposals to riders or offer valid confirmation to gain an area that guarantees the rider's repayment or unfair waiver via the time-limited deposit contract, proof of understanding and reputation system.

Basic, peer-to-peer journey-sharing structures offer affordability, convenience, flexibility, and social benefits. They offer a platform for green useful resource usage, reduce congestion, and provide a viable opportunity to standard transportation strategies.

3.4.2 Demerits

Peer-to-peer ride-sharing structures have a few ability drawbacks. Here are some demerits associated with peer-to-peer ride-sharing structures:

1. **Protection issues:** safety is a huge challenge in peer-to-peer ride-sharing. As drivers aren't professional drivers and may not undergo the same stage of training and background assessments as taxi or professional chauffeurs, there may be a higher danger of safety incidents. although systems typically enforce safety measures, including motive force heritage checks and user rankings, there may be still a degree of uncertainty as compared to traditional transportation offerings.
2. **Lack of regulation:** Peer-to-peer ride-sharing operates in a regulatory grey area in many areas. The absence of clear guidelines and oversight can result in troubles such as insufficient coverage insurance, non-compliance with licensing requirements, and challenges in imposing protection standards. This could create dangers for both drivers and passengers.
3. **Insurance insurance Gaps:** non-public vehicle insurance regulations might not provide insurance during p2p ride sharing sports. at the same time as ride sharing systems commonly provide insurance coverage, there may be gaps in coverage or disputes concerning legal responsibility. this can result in capability headaches for drivers and passengers inside the event of accidents or other incidents.
4. **Inconsistent carrier fine:** not like professional transportation services, peer-to-peer journey-sharing does not assure steady carrier fine. The enjoy can vary depending on the driving force, their automobile, and different factors. Passengers might also come across problems which includes unclean or poorly maintained automobiles, green drivers, or insufficient customer support.
5. **Limited Availability:** Peer-to-peer journey-sharing services won't be as readily to be had in all locations compared to traditional transportation options. In regions with decrease driver density or at some stage in off-top hours, finding available drivers can be hard. this will cause longer wait instances or unavailability of rides while wished.
6. **Dependence on user evaluations:** person ratings and evaluations are a number one mechanism for feedback and responsibility in peer-to-peer trip-sharing. however, those structures may be subject to biases, unfair scores, or fake comments, that could negatively impact both drivers and passengers. depending entirely on person

evaluations can also result in unfair remedy or erroneous tests.

7. **Loss of Accessibility**: Peer-to-peer trip-sharing won't be equally accessible to every person. people without smart phones or dependable net get admission to might also face problems in having access to and the usage of the provider. moreover, people with disabilities may also come across demanding situations in locating handy motors or drivers educated to house their particular needs.

CHAPTER 4

METHODOLOGY

4.1 SDLC LIFE CYCLE

Software program improvement life Cycle is a system used to design, broaden and check high great software by the software program enterprise. The SDLC targets to supply a excessive- first-class software that meets or exceeds client expectancies, reaches of entirety-inside-instance and fee estimate. SDLC is the acronym of software program development existence Cycle. It's also referred to as software program development system. SDLC is a framework defining responsibilities done at each step within the software improvement system.

The stages of SDLC are as follows:

1. Planning & requirements analysis:

- Become aware of and accumulate the unique necessities for the peer-to-peer ride-sharing carrier.
- engage with stakeholders, consisting of users, drivers, and transportation authorities, to understand their desires and expectations.
- define the useful and non-practical requirements, consisting of user registration, experience matching algorithms, payment structures, and safety features.

2. System designing:

- design the device structure and components based on the diagnosed requirements.
- Create wireframes, person interface designs, and information fashions.
- define the algorithms for matching riders with drivers, implementing real-time GPS tracking, and ensuring records safety.

3. Development/Coding:

- put into effect the system components and functionalities in line with the layout specifications.
- develop the the front-quit interfaces, returned-stop server structures, and cell applications.

- comprise vital functions such as person registration, trip request submission, motive force verification, and charge processing.

4. **Testing**:

- behavior diverse testing sports to ensure the gadget's functionality, performance, and safety.
- carry out unit testing to affirm the correctness of man or woman components.
- conduct integration checking out to check the interactions between distinct gadget modules.
- carry out system testing to validate the device's compliance with necessities.

5. **Deployment**:

- prepare the device for deployment to the production surroundings.
- installation the vital infrastructure, such as servers, databases, and community configurations.
- set up the device components, making sure proper integration and functionality.
- behavior compatibility trying out with diverse devices, running structures, and browsers.

6. Operations and maintenance:

- screen the system's performance, availability, and safety within the production surroundings.
- deal with any stated issues or bugs right away via bug fixes and patches.
- constantly improve and optimize the device based on user feedback and converting necessities.
- frequently replace the device to comprise new capabilities, security enhancements, and overall performance upgrades.

4.2 SDLC Diagram:

The Software Development Life Cycle (SDLC) for a Peer-to-Peer Ride-Sharing Service typically involves the following phases:

CHAPTER 5

MODULES

1. Ganache

Ganache is a personal blockchain for Ethereum development and testing. It creates a local Ethereum network on your machine, allowing you to deploy contracts, run tests, and simulate interactions with the blockchain. It provides predefined accounts with test Ether, has deterministic behavior, and integrates with popular development tools. Ganache is widely used by Ethereum developers for local testing and rapid development iterations.

2. Block chain Architectures used for ride

- The choice of blockchain architecture for a ride-sharing service depends on factors such as the desired level of transparency, regulatory compliance requirements, data privacy concerns, scalability needs, and the level of trust among participants. It's important to carefully evaluate the specific use case and requirements before deciding on the appropriate blockchain architecture.
- **Public Blockchain**: A public blockchain, such as the Ethereum blockchain, allows for transparency, immutability, and decentralized control. In a ride-sharing context, the blockchain can be used to record and verify ride transactions, driver and passenger identities, and payment settlements.
- **Permission/Private Blockchain:** A permission or private blockchain restricts access to a specific group of participants, such as ride-sharing service providers, drivers, and passengers. This type of blockchain architecture offers greater privacy and control over the network compared to public blockchain.

3. Sign Up/Login/Logout of the Ride sharing app

- **Sign Up:**
  - Open the ride-sharing app.
  - Tap on "Sign Up" or "Create an Account."
  - Provide required information accurately.

- Submit the form and complete any additional verification steps.

- **Login:**
  - Open the ride-sharing app.
  - Tap on "Log In" or "Sign In."
  - Enter your login credentials (email/username and password).
  - Complete any additional verification steps if prompted.

- **Logout:**
  - Open the ride-sharing app (ensure you are logged in).
  - Access your account settings or menu.
  - Find "Logout" or "Sign Out."
  - Confirm the logout action if asked.
  - You will be logged out and returned to the login screen or app's home screen

4. Front-end

- Front-end design that the user will use to interact with the application. The design may be generated depending on whether the DApp is designed for the mobile platform or for the web.

5. Google Maps API for geolocation for driver/customer tracking

- The Geolocation API returns a location and accuracy radius primarily based on information about cellular towers and a nodes that the cell client can come across. This record describes the protocol used to send these facts to the server and to go back a response to the consumer.

● Conversation is carried out over HTTPS using post. both request and response are formatted as JSON, and the content form of each is software/json.

CHAPTER 6

TOOLS AND TECHNIQUES

The system requirement can be classified into two categories:

1. Hardware Requirements
2. Software Requirements

6.1 Hardware Requirements:

1. CPU 4 processor or upper version.

2. Memory 2 GB or higher.

3. Hard disk 160 GB or higher.

4. Network card.

5. Windows operating system.

6. Working Camera.

7. Working Mic.

6.2 Software Requirements:

1. Visual Studio or Android Studio

Visual Studio Code (VS Code) is a popular code editor developed by Microsoft. It is a lightweight, versatile tool that provides extensive functionality for editing, deb ugging and managing code in multiple programming languages.

2. Browser (Chrome, edge, firefox)

A browser is a piece of software that enables users to access and see information on the World Wide Web. It serves as a bridge between a user and the internet, allowing the user to explore websites, read web pages, interacts with web information, and engage in other online activities.

3. Block Chain

Blockchain is a decentralized and distributed digital ledger technology that records transactions across multiple computers. It allows for secure and transparent recording of data and is most commonly associated with crypto currencies like Bit coin.

4. Google Maps/Directions API

The Google Maps/Directions API is a service provided by Google that allows developers to integrate mapping and routing functionalities into their applications. It provides a set of APIs that enable developers to access various features and data related to maps, directions, and geolocation.

Create better experiences and improve operations with rich, detailed geospatial data, helpful mapping tools, and industry-leading reliability.

5. Rich geospatial data

Use Autocomplete to simplify address entry, and avoid missed pickups by sending drivers to customers' locations. You can also use that same geolocation information to quickly provide customers with rich Place information like phone numbers and reviews.

Google Maps/Directions API allows you to display geospatial data on maps. You can plot markers, polygons, and polylines to represent various points of interest, areas, and routes on the map.

6. Ganache

Ganache is a personal blockchain for Ethereum development and testing. It creates a local Ethereum network on your machine, allowing you to deploy contracts, run tests, and simulate interactions with the blockchain

7. **Payment interface:** Payment Integration: The integration of secure and reliable payment gateways is essential to process transactions between riders and drivers. Popular options include Stripe, Braintree, or PayPal.

CHAPTER 7

PERFORMANCE EVOLUTION

Performance evaluation of a peer-to-peer ride-sharing service typically involves evaluating various aspects of service effectiveness and efficiency. Here are some key areas to consider for performance appraisal:

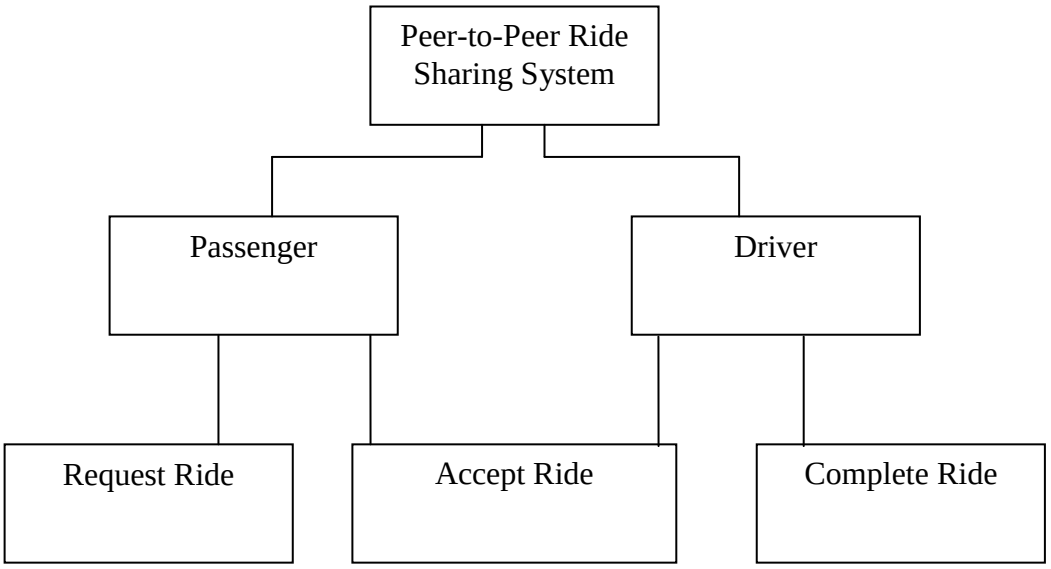
1. **User Satisfaction:** To measure user satisfaction through surveys, ratings and feedback to measure their overall experience with the Service, including app usability, driver professionalism and ride quality.
2. **Service Reliability:** Assess service reliability by monitoring uptime, response time, and system availability. Ensure that the platform is accessible to users without significant downtime or technical issues.
3. **Ride Matching Efficiency**: Evaluate the efficiency of the ride matching algorithm to ensure that it successfully connects riders with available drivers in a timely manner, minimizes wait times, and maximizes service utilization.
4. **Driver Performance:** Monitor driver performance metrics such as acceptance rate, cancellation rate and average rating to ensure drivers are meeting service standards and providing a positive rider experience.
5. **Payment Processing**: Assess the accuracy and timeliness of payment processing for both riders and drivers. Ensure transactions are processed securely and efficiently with minimum discrepancies or delays.
6. **Safety and Security:** Review safety protocols, driver background checks and incident reporting mechanisms to ensure the service maintains a safe environment for users. Monitor and promptly address any reported safety concerns.

CHAPTER 8

DESIGN PHASE

8.1 Use Case Diagram:

Use case diagram depicts the explanation from gadget architecture and indicates the relation between driving force and rider. here after a hit login the person selects either to be a rider or a driving force. If motive force is selected then the motive force will post the ride while if rider is selected then the rider will seek the journey, get statistics and subsequently e book the experience. Now if the whole thing suits they'll proportion the experience.



In this diagram, the principle elements are represented as follows:

- **Peer-to-Peer trip sharing gadget:** The relevant device that facilitates the experience-sharing technique between passengers and drivers.
- **Passenger:** An actor representing a person who requests a trip.
- **Driver:** An actor representing a person who accepts journey requests and gives transportation offerings.

- **Request experience:** A use case that represents the movement of a passenger inquiring for a ride thru the device.
- **Take delivery of ride:** A use case that represents the movement of a driver accepting a journey request from a passenger.
- **Whole ride:** A use case that represents the motion of a driver marking a journey as whole after the adventure is finished.

8.2 DFD Diagram

A records go with the flow diagram (DFD) is a graphical illustration that illustrates the waft of facts within a device. It shows how records is input, processed, and outputted within various components of a device. Within the context of a peer-to-peer journey-sharing device, a DFD can illustrate how statistics flows among exclusive additives, together with users, databases, and offerings. here's an instance of a excessive-stage records glide diagram for a peer-to-peer ride-sharing device:

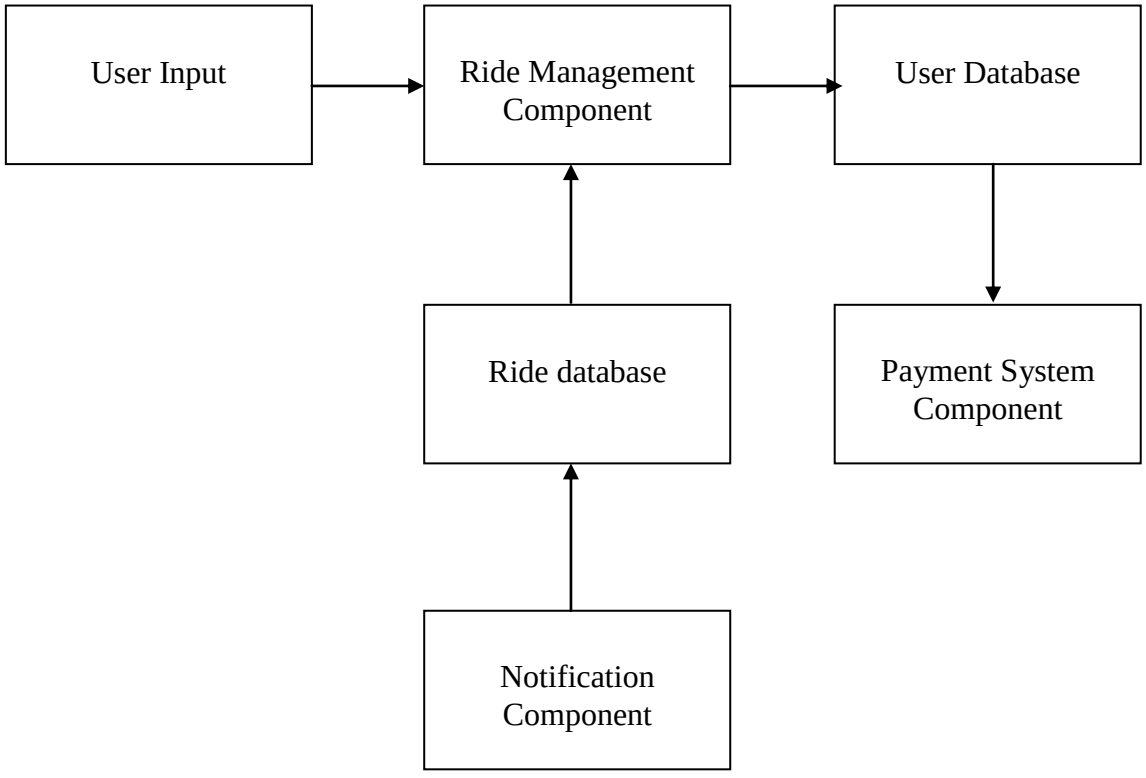


Fig: DFD of peer to peer ride sharing system

In diagram, arrows constitute the drift of information. here's a breakdown of the additives and the data go with the flow:

- **User input:** Represents the facts entered with the aid of the users, consisting of trip requests, charge records, and preferences.
- **Experience control factor:** tactics the user input and handles numerous ride control functions, inclusive of matching drivers with passengers, updating experience reputation, and producing notifications.
- **Experience Database:** shops records related to rides, consisting of trip info, driving force and passenger records, and journey fame.
- **User Database:** stores consumer-related statistics, together with user profiles, choices, and experience records.
- **Payment device issue:** Manages the payment-related functionalities, which includes processing payments from passengers, calculating fares, and updating charge transactions.
- **Notification factor:** Generates and sends notifications to users regarding trip updates, charge confirmations, and other applicable records.

The facts go with the flow diagram presents a excessive-degree view of the way data movements between distinctive components inside the device. It facilitates visualize the go with the flow of facts and apprehend the interactions among components in a peer-to-peer experience-sharing system.

8.3 Component diagram

A component diagram shows the structural dating of components of a software device. Those are mostly used whilst operating with complex systems which have many additives. Additives talk with each different the usage of interfaces. The interfaces are linked using connectors. under snap shots shows a factor diagram.

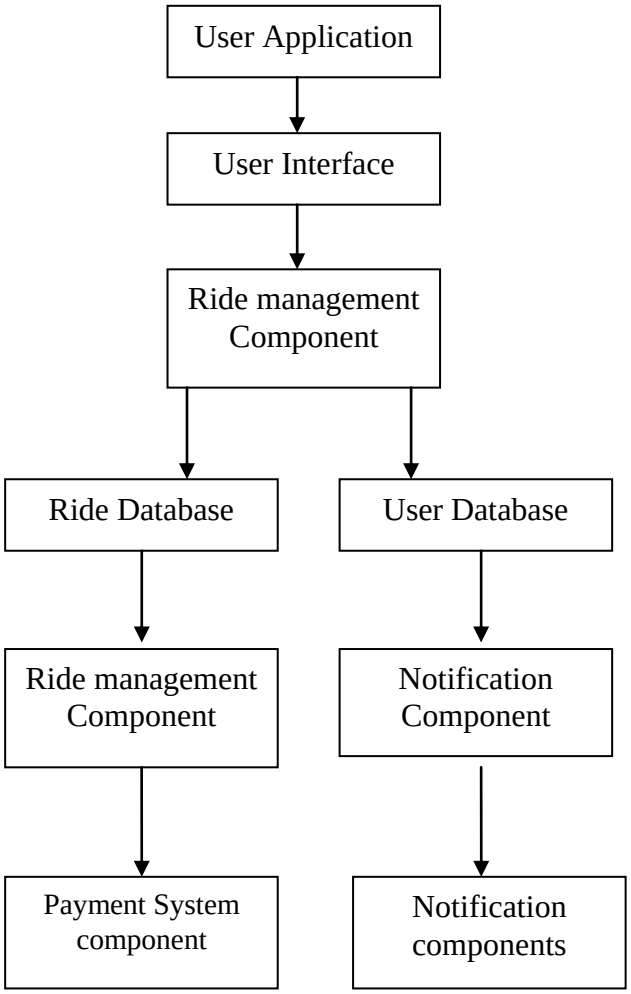


Fig: Component diagram

In this Component diagram, the main components of the peer-to-peer journey-sharing gadget are represented through boxes. Here's a quick description of every aspect:

- **User application**: The software established on users' devices (which include a cell app) that allows them to engage with the trip-sharing device.
- **User Interface**: The issue accountable for supplying the person interface factors to the customers and taking pictures their enter.

- **Journey management aspect:** Handles the middle functionalities associated with experience control, which includes matching drivers with passengers, tracking experience status, and coping with journey requests.
- **Ride Database:** stores information about rides, which includes ride details, motive force and passenger records, and trip repute.
- **User Database:** stores person-related data, along with consumer profiles, preferences, and experience records.
- **Price gadget factor:** Handles the payment-related functionalities, consisting of processing payments from passengers, calculating fares, and managing fee transactions.
- **Notification issue:** chargeable for sending notifications to users regarding ride updates, payment confirmations, and different applicable information.

8.4 Flow Chart

A flowchart diagram represents the drift of activities and decisions inside a machine or technique. Within the context of a peer-to-peer ride-sharing system, the flowchart can illustrate the series of steps concerned in requesting and finishing a trip. here's an example of a flowchart diagram for a peer-to-peer journey-sharing device:

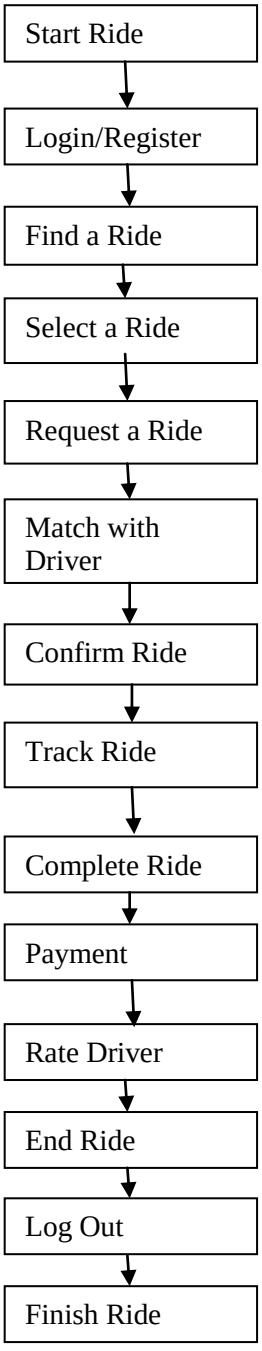


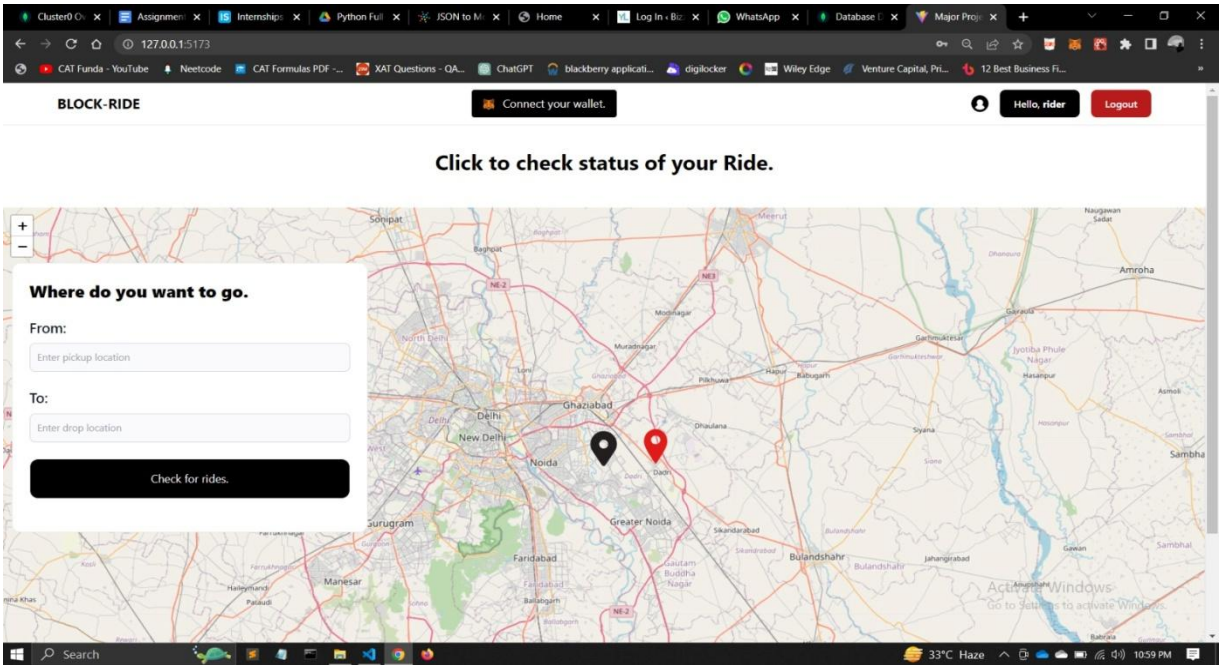
Fig: Flow chart diagram

CHAPTER 9

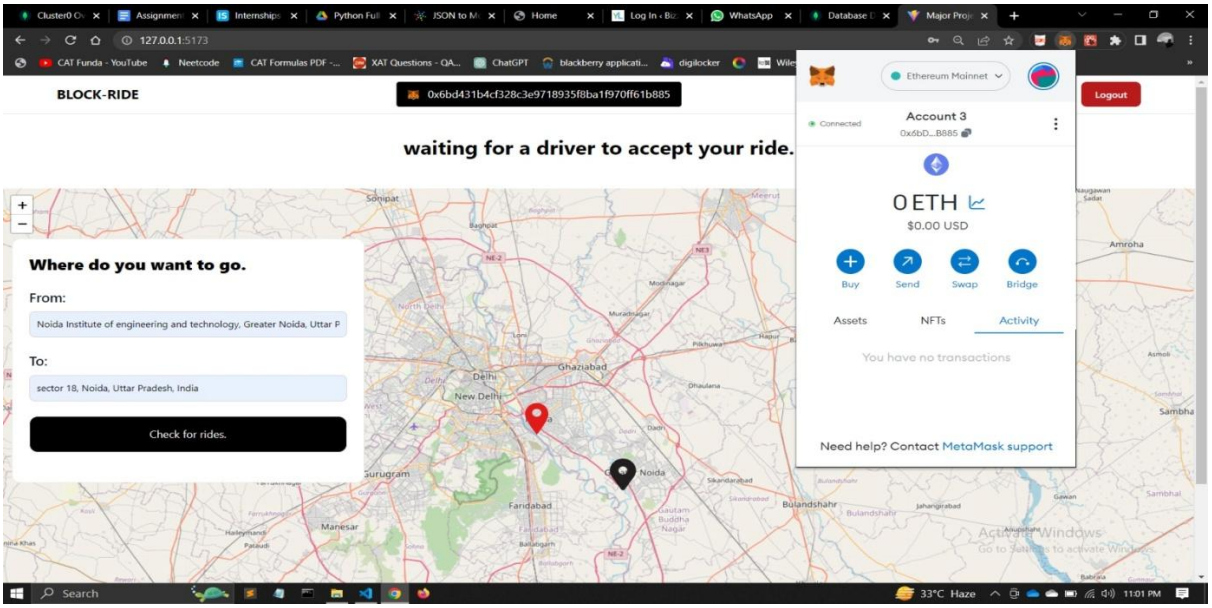
PROTOTYPE DESIGN OF THE USER INTERFACE

9.1 Home page :

This is the home page of our web-app where user will visit to book a ride.

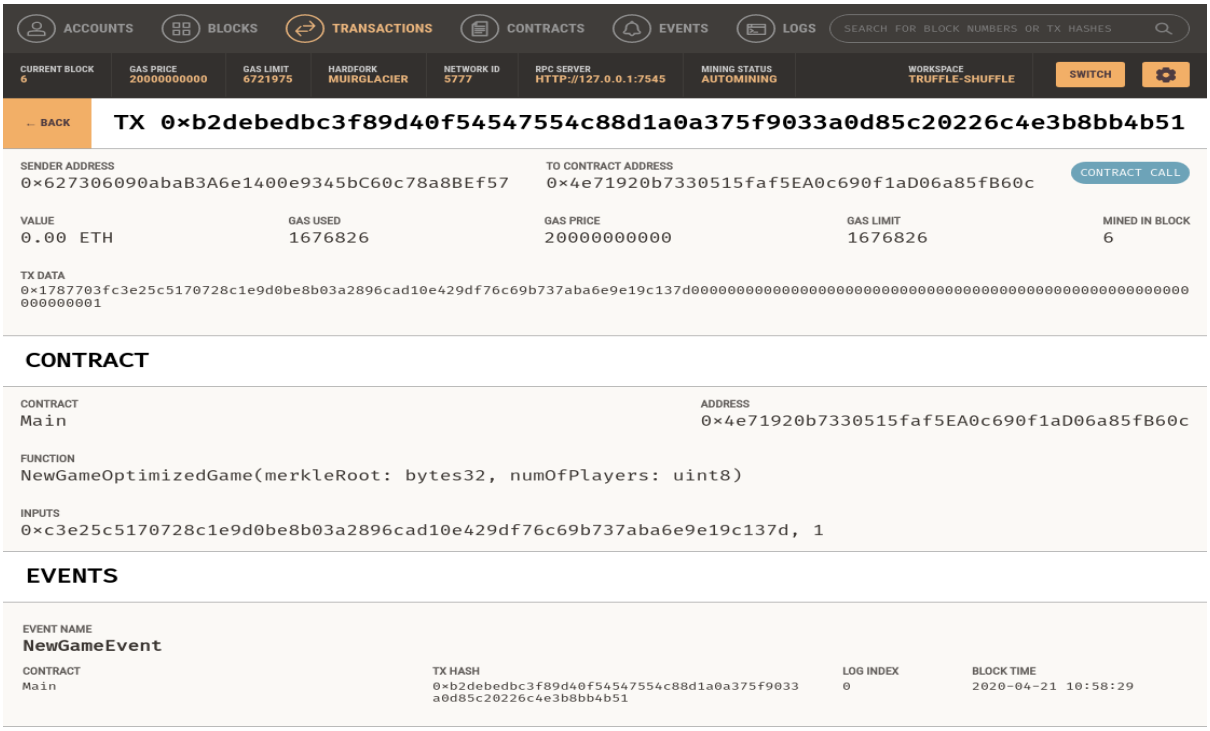


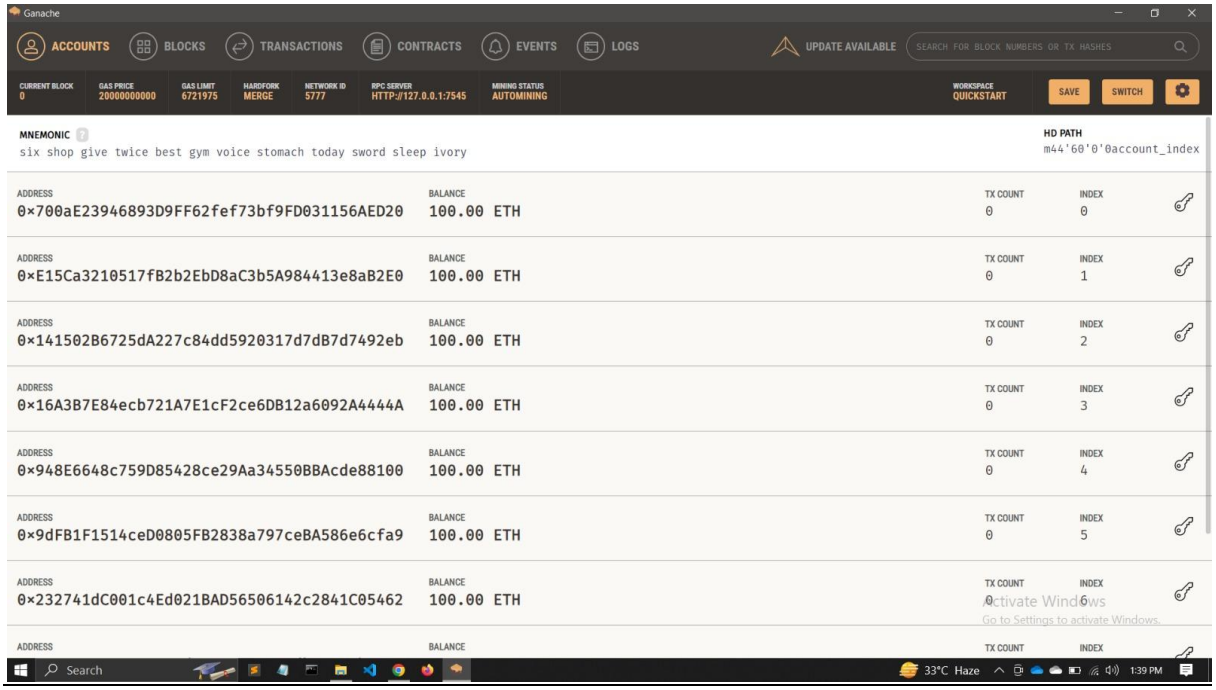
9.2 Booking a ride as shown:



9.3 Ganache Serve Screenshot

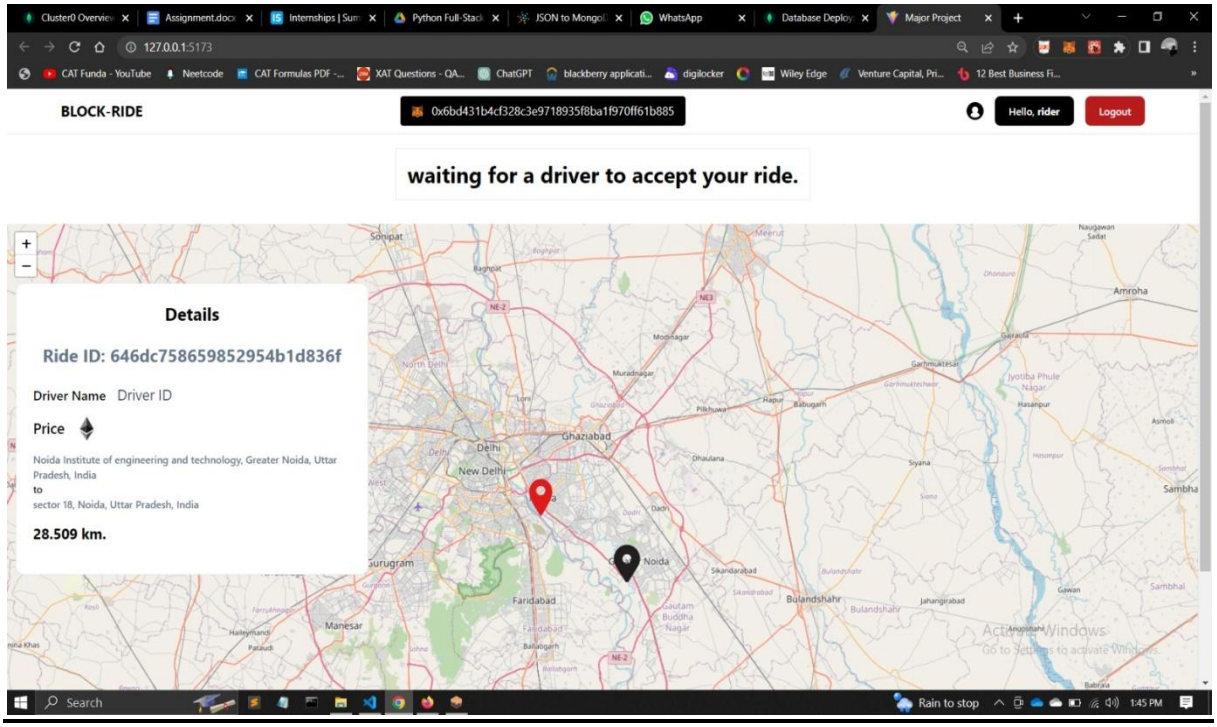
Ganache will now try and decode transactions that are contract calls. similarly to list the events (encoded or now not) for the transaction, Ganache will show the feature signature of the transaction in addition to the values of the arguments.

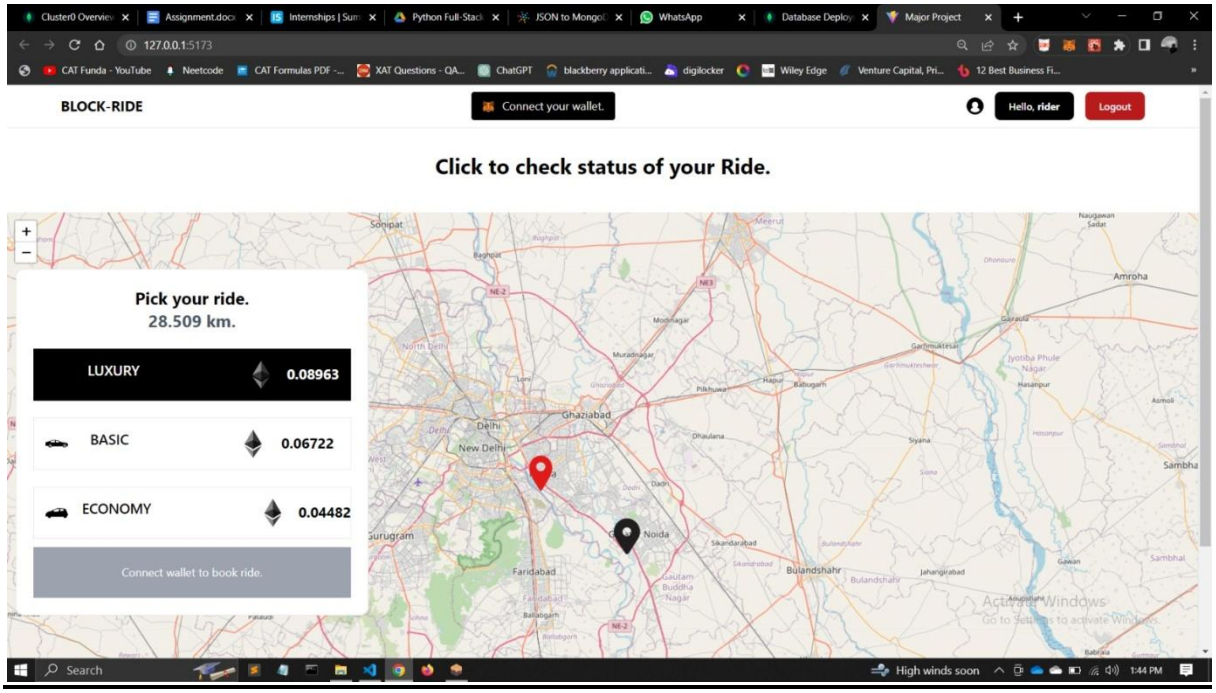


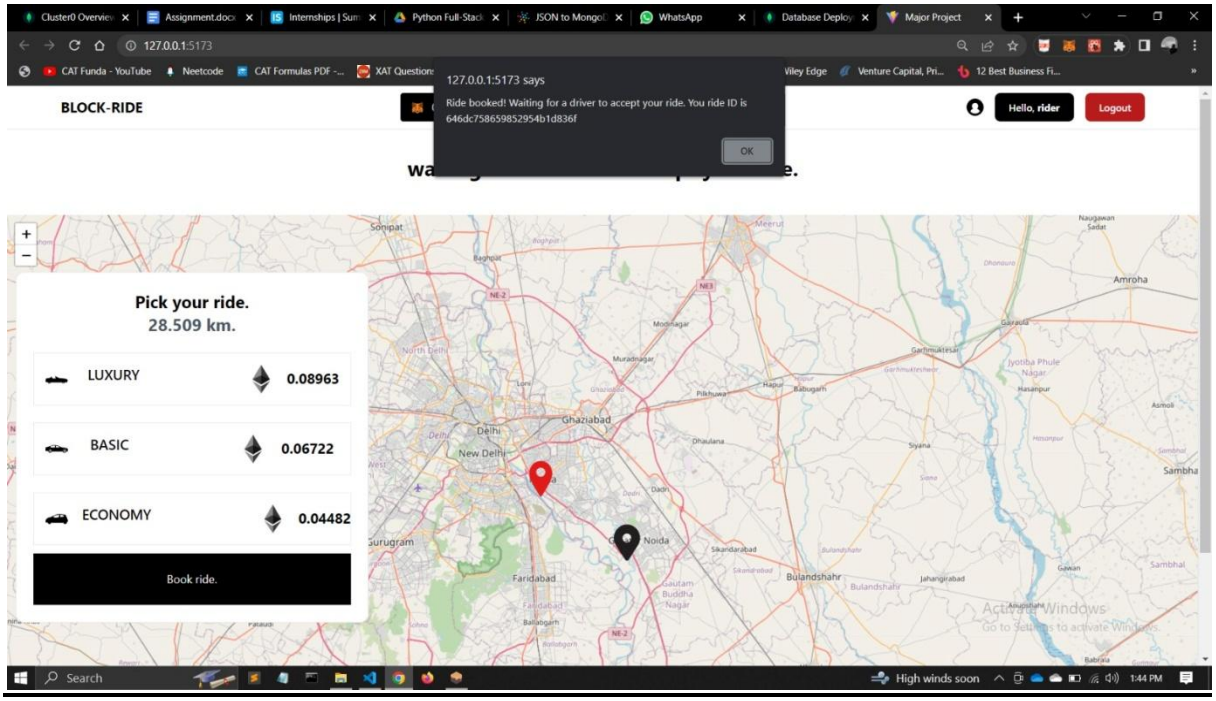


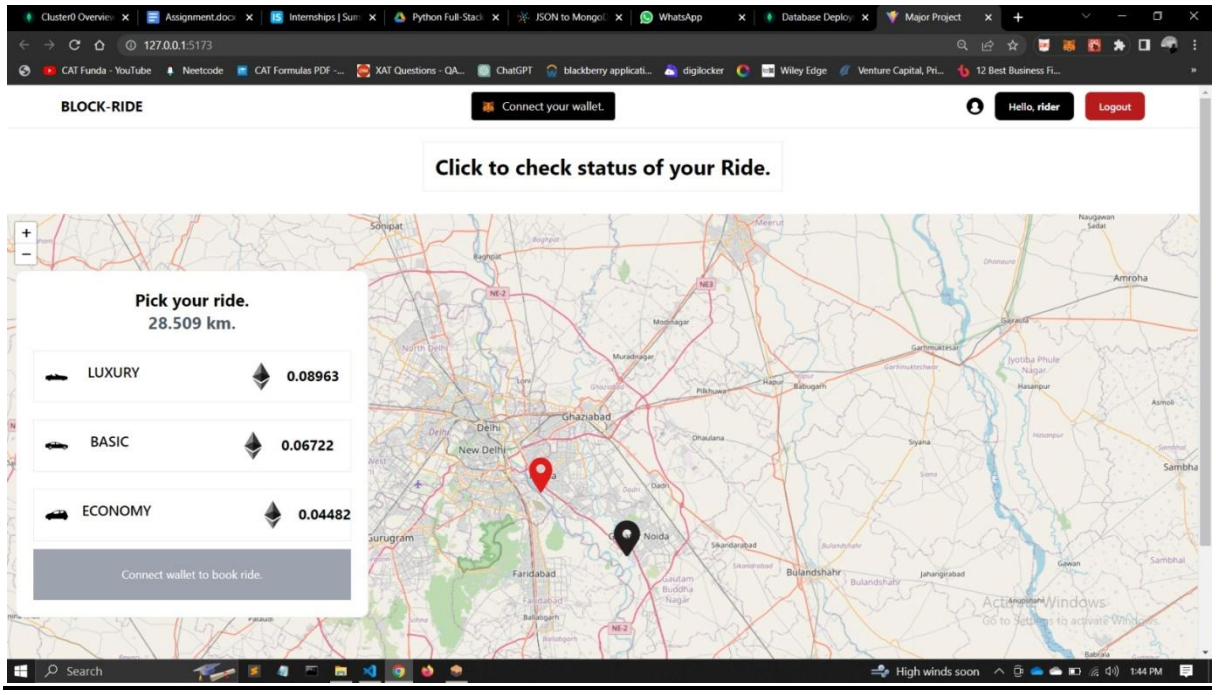
Ganache screenshot

9.4 Ride booking Screenshot

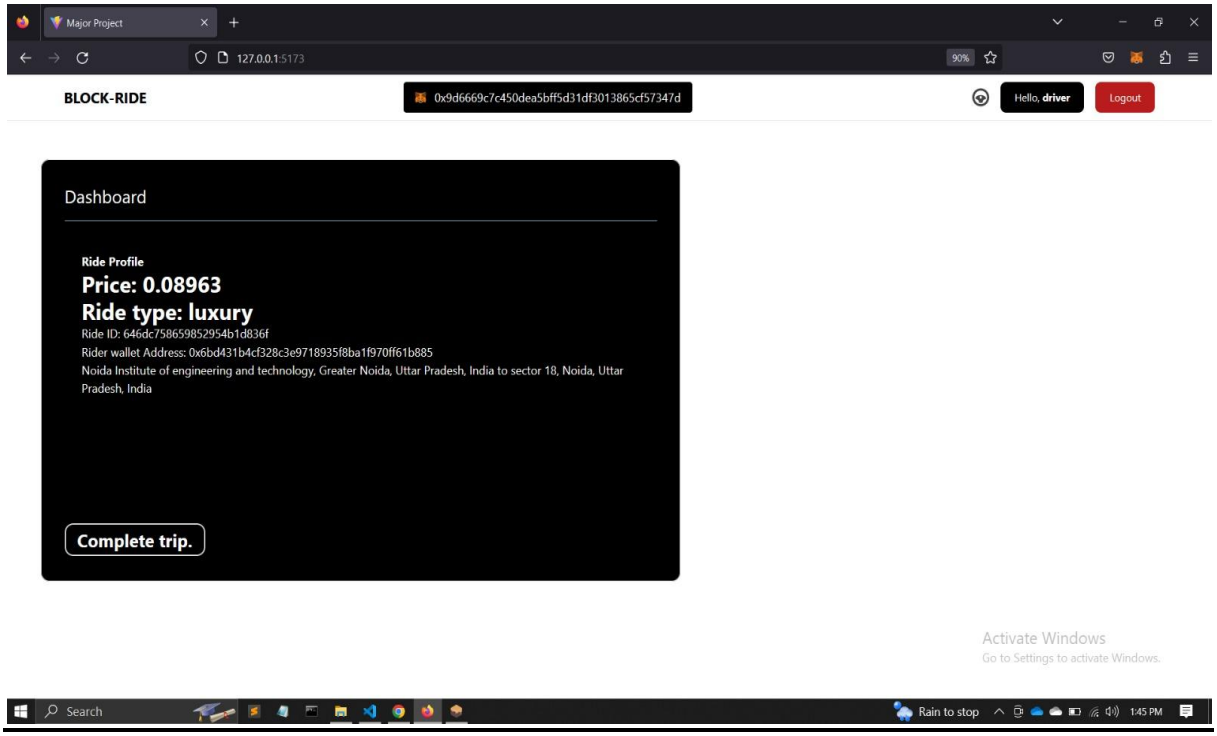


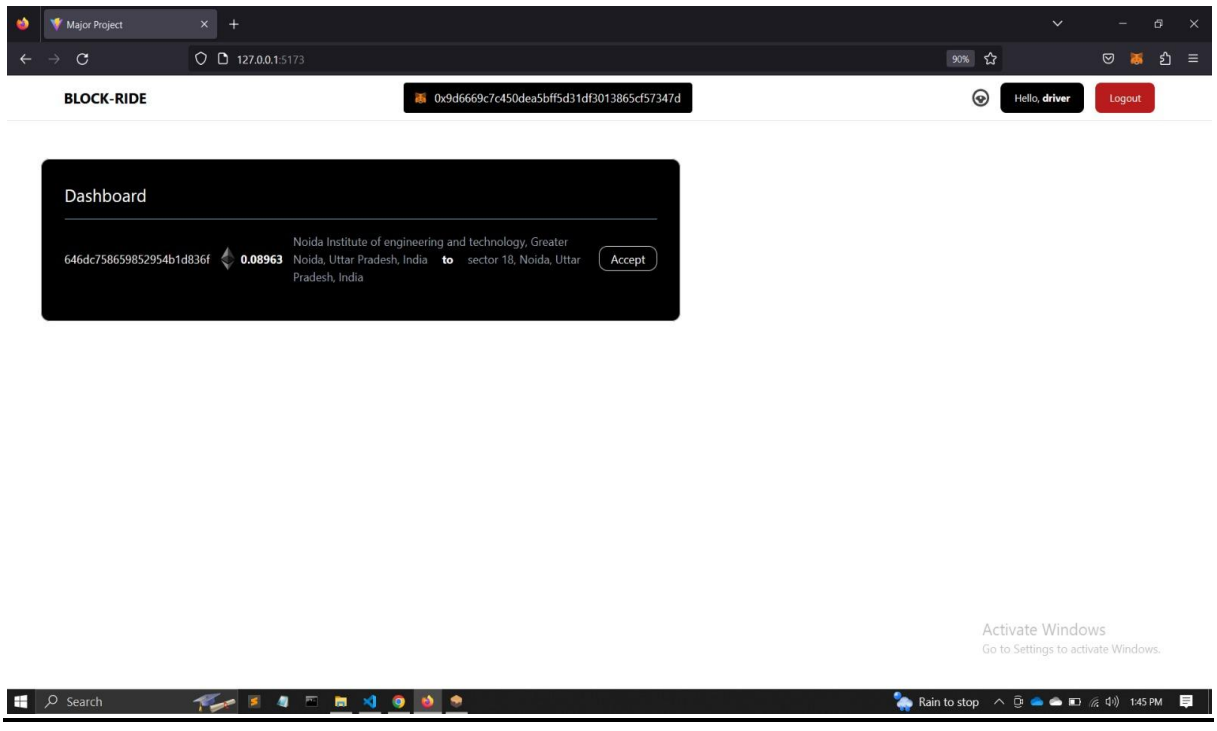



127.0.0.1:5173 says
Ride booked! Waiting for a driver to accept your ride. You ride ID is 646dc758659852954b1d836f
OK
BLOCK-RIDE
Hello, rider
Logout
Pick your ride.
28.509 km.
LUXURY
0.08963
BASIC
0.06722
ECONOMY
0.04482
Book ride.


BLOCK-RIDE
Connect your wallet.
Hello, rider
Logout
Click to check status of your Ride.
Pick your ride.
28.509 km.
LUXURY
0.08963
BASIC
0.06722
ECONOMY
0.04482
Connect wallet to book ride.

9.5 Payment Screenshot





CHAPTER 10

CODING PHASE

10.1 Ride.js file

```
const mongoose = require("mongoose");
const { Schema, model } = mongoose;

const RidesSchema = new Schema({
  tier: { type: String, required: true },
  from: { type: String, required: true },
  to: { type: String, required: true },
  riderWalletId: { type: String, required: true },
  driverWalletId: { type: String },
  price: { type: Number },
  status: { type: String },
});

const ridesModel = model("Rides", RidesSchema);

module.exports = ridesModel;
```

10.2 User.js file

```
const mongoose = require("mongoose");
const { Schema, model } = mongoose;

const UserSchema = new Schema({
  email: { type: String, required: true, min: 4, unique: true },
  password: { type: String, required: true },
  firstName: { type: String, required: true },
  lastName: { type: String, required: true },
  phoneNumber: { type: String, required: true },
  typeOfUser: { type: String, required: true },
});

const UserModel = model("User", UserSchema);

module.exports = UserModel;
```

10.3 index.js file

```
const express = require("express");
const cors = require("cors");
const mongoose = require("mongoose");
const User = require("./models/User");
```

```
const Ride = require("./models/Ride");
const bcrypt = require("bcrypt");
const app = express();
const jwt = require("jsonwebtoken");
const cookieParser = require("cookie-parser");

const salt = bcrypt.genSaltSync(10);
const secret = "asdadaaxaxaaxaxa213123121qd";

app.use(cors({ credentials: true, origin: "http://127.0.0.1:5173" }));
app.use(express.json());
app.use(cookieParser());

const port = 5000;
const mongooseConnectionString =
  "mongodb+srv://kartikanand:Dr%40gonball2821@cluster0.paoeug0.mongodb.net/?retryWrites=true&w=majority";

mongoose.connect(mongooseConnectionString);

app.post("/register", async (req, res) => {
  const { email, password, firstName, lastName, phoneNumber, typeOfUser } =
    req.body;
  try {
    const userDoc = await User.create({
      email,
      firstName,
      lastName,
      phoneNumber,
      typeOfUser,
      password: bcrypt.hashSync(password, salt),
    });
    console.log(userDoc);
    res.json(userDoc);
  } catch (error) {
    console.log(error);
    res.status(400).json(error);
  }
});

app.post("/book-ride", async (req, res) => {
  const { price, tier, from, to, riderWalletId } = req.body;
  try {
    const rideDoc = await Ride.create({
      from,
      to,
      price,
      riderWalletId,
      tier,
    });
```

```
54          res.json(rideDoc);
55        } catch (error) {
56          console.log(error);
57          res.status(400).json(error);
58        }
59      });
60
61      app.get("/get-rides", async (req, res) => {
62        try {
63          const rides = await Ride.find({ status: null });
64          res.json(rides);
65        } catch (error) {
66          console.log(error);
67        }
68      });
69
70      app.post("/get-ride-details", async (req, res) => {
71        const { id } = req.body;
72        try {
73          const rideDoc = await Ride.findOne({ _id: id });
74          res.json(rideDoc);
75        } catch (error) {
76          res.json(error);
77        }
78      });
79
80      app.post("/update-driver-wallet-id", async (req, res) => {
81        const { id, driverWalletId } = req.body;
82        try {
83          const rideDoc = await Ride.findOneAndUpdate(
84            { _id: id },
85            { driverWalletId: driverWalletId },
86            { new: true }
87          );
88          res.json(rideDoc);
89        } catch (error) {
90          console.log(error);
91        }
92      });
93
94      app.post("/complete-ride", async (req, res) => {
95        const { id, status } = req.body;
96        try {
97          const rideDoc = await Ride.findOneAndUpdate(
98            { _id: id },
99            { status: status },
100           { new: true }
101         );
102         res.json(rideDoc);
103       } catch (error) {
```

```
104         res.json(error);
105       }
106     });
107
108     app.post("/login", async (req, res) => {
109       const { email, password } = req.body;
110       const userDoc = await User.findOne({ email: email });
111       console.log(userDoc);
112       const passOk = bcrypt.compareSync(password, userDoc.password);
113       if (passOk) {
114         // logged in
115         jwt.sign(
116           {
117             id: userDoc._id,
118             firstName: userDoc.firstName,
119             typeOfUser: userDoc.typeOfUser,
120           },
121           secret,
122           {},
123           (err, token) => {
124             if (err) throw err;
125             res.cookie("token", token, { sameSite: "none", secure: true }).json({
126               id: userDoc._id,
127               firstName: userDoc.firstName,
128               typeOfUser: userDoc.typeOfUser,
129             });
130           }
131         );
132       } else {
133         res.status(400).json("Invalid Credentials.");
134       }
135     });
136
137     app.get("/profile", (req, res) => {
138       const { token } = req.cookies;
139       jwt.verify(token, secret, {}, (err, info) => {
140         if (err) throw err;
141         console.log(info);
142         res.json(info);
143       });
144     });
145
146     app.post("/logout", (req, res) => {
147       res.cookie("token", "").json("ok");
148     });
149
150     app.listen(port, () => {
151       console.log(`App listening on port ${port}`);
152     });
```

10.4 DriverCard.jsx file

```jsx
import React, { useContext, useState } from "react";
import { RideContext } from "../context/RideContext";
import axios from "axios";

function DriverCard({ distance }) {
  const eth_logo = "https://cryptologos.cc/logos/ethereum-eth-logo.svg?v=024";
  const {
    acceptedRideId,
    loc,
    dest,
    rideInfo,
    setRideInfo,
    isPaymentComplete,
    setIsPaymentComplete,
  } = useContext(RideContext);
  return (
    <div className="p-6">
      <h1 className="text-2xl text-center font-bold mb-6">Details</h1>
      {/* <button
        className="w-full text-white bg-black mb-4 py-2 rounded-xl"
        onClick={checkStatus}
      >
        refresh
      </button> */}

      <h1 className="text-2xl text-center font-bold mb-6 text-slate-500">
        Ride ID: {acceptedRideId}
      </h1>
      <div className="flex gap-4 justify-start mb-4 overflow-hidden">
        <p className="text-lg font-semibold">Driver Name</p>
        <h1 className="text-xl text-slate-700">
          {/* {rideInfo.driverWalletId
            ? rideInfo.driverWalletId
            : "driver's wallet ID"} */}
          Driver ID
        </h1>
      </div>
      <div className="flex gap-4 justify-start items-center mb-4">
        <p className="text-xl font-semibold">Price</p>
        <img className="h-8 w-8" src={eth_logo} alt="eth logo" />
      </div>

      <div className="flex gap-4 justify-start mb-4">
        <p className="text-sm font-semibold">
          | <p className="text-slate-500">{loc}</p> to{" "}
          <p className="text-slate-500">{dest}</p>
        </p>
      </div>

      <div className="flex gap-4 justify-start mb-4">
        <h1 className="text-xl font-bold">{distance} km.</h1>
      </div>
    </div>
  );
}

export default DriverCard;
```

10.5 DriverDashboard.jsx

```jsx
import React, { useState, useEffect, useContext } from "react";
import axios from "axios";
import { WalletContext } from "../context/WalletContext";
import { RideContext } from "../context/RideContext";
import RiderCard from "./RiderCard";

function DriverDashboard() {
  const { walletConnected, walletId } = useContext(WalletContext);
  const [driverLocationCoords, setDriverLocationCoords] = useState([]);
  const [driverLocationName, setDriverLocationName] = useState("");
  const [allRides, setAllRides] = useState(null);
  const { acceptedRideId, setAcceptedRideId, rideInfo, setRideInfo } =
    useContext(RideContext);
  const eth_logo = "https://cryptologos.cc/logos/ethereum-eth-logo.svg?v=024";

  const testLocation = [28.461, 77.4969];

  //   useEffect(() => {
  //     navigator.geolocation.getCurrentPosition((position) => {
  //       setDriverLocationCoords([
  //         position.coords.latitude,
  //         position.coords.longitude,
  //       ]);
  //     });
  //   }, []);

  useEffect(() => {
    axios
      .get("http://localhost:5000/get-rides")
      .then((response) => console.log(setAllRides(response.data)));
  }, []);

  async function acceptRide() {
    const response = await axios.post(
      "http://localhost:5000/update-driver-wallet-id",
      { id: acceptedRideId, driverWalletId: walletId }
    );
    console.log(response.data);
    setRideInfo(response.data);
  }

  async function completeTrip() {
    const response = await axios.post("http://localhost:5000/complete-ride", {
      id: acceptedRideId,
      status: "completed",
    });
    console.log(response.data);
    setRideInfo(response.data);
    window.location.reload(false);
  }
```

```
59      return (
60        <div className="mt-16 justify-center item-center p-12 w-full h-full">
61          <div className="border rounded-xl h-content p-8 bg-black text-white w-[900px]">
62            <h1 className="font-sans text-2xl mb-4"> Dashboard </h1>
63            <div className="border-b-2 border-slate-600 mb-4"></div>
64            <div>
65              {/* <button onClick={getLocationOfDriver}> Click </button> */}
66            </div>
67            {rideInfo?.status == "completed" || !rideInfo ? (
68              <div>
69                {allRides?.map((ride) => {
70                  return (
71                    <div key={ride._id} className="flex gap-4 mb-4 items-center ">
72                      <div>{ride._id}</div>
73                      <div className="flex gap-2 items-center">
74                        <img className="h-8 w-8" src={eth_logo} alt="eth_logo" />
75                        <h1 className="font-2xl font-bold">{ride.price}</h1>
76                      </div>
77                      <div className="border border-white"></div>
78                      <div className="text-gray-400">
79                        <p>
80                          {ride.from} <b className="text-white ml-4 mr-4 "> to </b>
81                          {ride.to}
82                        </p>
83                      </div>
84                      <div>
85                        <button
86                          className="border rounded-xl px-4 py-1 hover:bg-white hover:text-black"
87                          disabled={walletConnected ? false : true}
88                          onClick={() => {
89                            setAcceptedRideId(ride._id);
90                            acceptRide();
91                          }}
92                        >
93                          {walletConnected
94                            ? "Accept"
95                            : "Please connect wallet to accept ride."}
96                        </button>
97                      </div>
98                    </div>
99                  );
100               })}
101             </div>
102           ) : (
103             <div>
104               <div>
105                 <RiderCard rideInfo={rideInfo} />
106                 <button
107                   className="rounded-xl text-2xl font-bold border-2 px-4 py-1 hover:bg-white hover:text-black"
108                   onClick={completeTrip}
109                 >
110                   Complete trip.
```

```
                  </button>
                </div>
              </div>
            )}
          </div>
        </div>
      );
    }

    export default DriverDashboard;
```

10.6 RideCard.jsx

```
import React from "react";

function RiderCard({ rideInfo }) {
  return (
    <div className="h-96 w-full p-6">
      <h1 className="font-2xl font-bold">Ride Profile</h1>
      <h1 className="text-3xl font-bold">Price: {rideInfo.price}</h1>
      <h1 className="text-3xl font-bold">Ride type: {rideInfo.tier}</h1>

      <div>Ride ID: {rideInfo._id}</div>
      <div>Rider wallet Address: {rideInfo.riderWalletId}</div>
      <div>
        {rideInfo.from} to {rideInfo.to}
      </div>
    </div>
  );
}

export default RiderCard;
```

10.7 RiderDashboard.jsx

```
import React, { useState, useContext, useEffect } from "react";
import { RideContext } from "../context/RideContext";
import { ethers } from "ethers";
import RenderMap from "./RenderMap";
import Pricing from "./Pricing";
import axios from "axios";
import { WalletContext } from "../context/WalletContext";
import DriverCard from "./DriverCard";
```

10.8 RenderMap.jsx

```
import React from "react";
import { MapContainer, TileLayer, useMap, Marker, Popup } from "react-leaflet";
import L from "leaflet";
```

```
export default function RenderMap(props) {
  const locationCoordinates = [
    props.lat ? props.lat : 28.5942,
    props.lon ? props.lon : 77.4431,
  ];
  const destinationCoordinates = [
    props.destLat ? props.destLat : 28.598,
    props.destLon ? props.destLon : 77.5431,
  ];

  const locationMarker = new L.icon({
    iconUrl: "https://www.svgrepo.com/show/513450/location-pin.svg",
    iconSize: [70, 50],
  });

  const destinationMarker = new L.icon({
    iconUrl: "https://www.svgrepo.com/show/292182/pointer-pin.svg",
    iconSize: [70, 50],
  });

  return (
    <div className="h-full w-full">
      <MapContainer
        className="h-[700px]"
        center={locationCoordinates}
        zoom={10}
        scrollWheelZoom={true}
        bounds={(locationCoordinates, destinationCoordinates)}
      >
        <TileLayer
          attribution='&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
          url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
        />
        <Marker position={locationCoordinates} icon={locationMarker}>
          <Popup>You are here.</Popup>
        </Marker>

        <Marker position={destinationCoordinates} icon={destinationMarker}>
          <Popup>Here's where you want to go.</Popup>
        </Marker>
      </MapContainer>
    </div>
  );
}
```

10.9 Home.jsx

```
import React, { useContext, useEffect } from "react";
import { RideContext } from "../context/RideContext";
import { UserContext } from "../context/UserContext";
import DriverDashboard from "./DriverDashboard";
import RiderDashboard from "./RiderDashboard";
```

```
6
7   function Home() {
8     const { userInfo } = useContext(UserContext);
9
10    if (userInfo?.typeOfUser === "rider") {
11      return <RiderDashboard />;
12    }
13    if (userInfo?.typeOfUser === "driver") {
14      return <DriverDashboard />;
15    }
16  }
17
18  export default Home;
```

10.10 Login.jsx

```
1   import React, { useState, useContext } from "react";
2   import { Link, Navigate } from "react-router-dom";
3   import { UserContext } from "../context/UserContext";
4   import axios from "axios";
5
6   function Login() {
7     const [loginFormData, setLoginFormData] = useState({
8       email: "",
9       password: "",
10    });
11
12    const { setUserInfo } = useContext(UserContext);
13
14    const [redirect, setRedirect] = useState(false);
15
16    function handleChange(e) {
17      const name = e.currentTarget.name;
18      const value = e.currentTarget.value;
19
20      setLoginFormData({ ...loginFormData, [name]: value });
21    }
22
23    async function handleSubmit(e) {
24      e.preventDefault();
25
26        const response = await axios.post(
27          "http://localhost:5000/login",
28          loginFormData,
29          { withCredentials: true }
30        );
31
```

```
32          if (response.status === 200) {
33            alert("Login Successful.");
34            const userInfo = response.data;
35            setRedirect(true);
36            setUserInfo(userInfo);
37          } else {
38            alert("Invalid Credentials");
39          }
40        }
41        if (redirect) return <Navigate to={"/"} />;
42
43        return (
44          <div className="p-16 mt-12">
45            <h1 className="text-center text-2xl font-bold  mb-10">Login.</h1>
46            <form onSubmit={handleSubmit}>
47              <div class="mb-6">
48                <label
49                  for="email"
50                  class="block mb-2 text-sm font-medium text-gray-900 "
51                >
52                  Your email
53                </label>
54                <input
55                  type="email"
56                  name="email"
57                  id="email"
58                class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500
59                placeholder="johndoe@email.com"
60                value={loginFormData.email}
61                onChange={handleChange}
62                required
63              />
64            </div>
65            <div class="mb-6">
66              <label
67                for="password"
68                class="block mb-2 text-sm font-medium text-gray-900 "
69              >
70                Your password
71              </label>
72              <input
73                type="password"
74                name="password"
75                id="password"
76                class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500
77                value={loginFormData.password}
78                onChange={handleChange}
79                required
80              />
81            </div>
82            <button
83              type="submit"
```

```
            class="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus
          >
            Submit
          </button>
        </form>

        <p className="text-center font-bold mt-6">
          Not a user?{" "}
          <Link to="/register" className="text-blue-600">
            Register Here.
          </Link>
        </p>
      </div>
    );
  }

  export default Login;
```

10.11 Pricing.jsx

```
import React, { useState, useContext } from "react";
import axios from "axios";
import { WalletContext } from "../context/WalletContext";
import { RideContext } from "../context/RideContext";

function Pricing(props) {
  const [ride, setRide] = useState("");
  const distance = props.distance;
  const basePrice = 1572;
  const eth_logo = "https://cryptologos.cc/logos/ethereum-eth-logo.svg?v=024";
  const [finalPrice, setFinalPrice] = useState(null);

  const { walletConnected, walletId } = useContext(WalletContext);

  const cabs = {
    luxury: {
      name: "LUXURY",
      imgUrl: "src/assets/luxury.svg",
      multiplier: (distance / 10) * 2,
    },

    basic: {
      name: "BASIC",
      imgUrl: "src/assets/basic.svg",
      multiplier: (distance / 10) * 1.5,
      },

      economy: {
        name: "ECONOMY",
        imgUrl: "src/assets/economy.svg",
        multiplier: (distance / 10) * 1,
      },
    };

    const {
      loc,
```

```
    dest,
    acceptedRideId,
    setAcceptedRideId,
    driverCardVisibility,
    setDriverCardVisibility,
  } = useContext(RideContext);
  const rideData = {
    tier: ride,
    from: loc,
    to: dest,
    riderWalletId: walletId,
    price: finalPrice,
  };

  async function bookRide() {
    const response = await axios.post(
      "http://localhost:5000/book-ride",
      rideData
    );
    const data = response.data;
    console.log(data._id);
    setAcceptedRideId(data._id);
    alert(
      `Ride booked! Waiting for a driver to accept your ride. You ride ID is ${data._id}`
    );
    setDriverCardVisibility(true);
  }
  return (
    <div className="p-6">
       <h1 className="text-2xl text-center font-bold">Pick your ride.</h1>
       <h1 className="text-2xl text-center font-bold text-gray-600">
         {distance} km.
       </h1>

       <div className="mt-6">
         <button
           className="flex border hover:border-black focus:border-black focus:bg-black focus:
           onClick={() => {
             setRide("luxury");
             setFinalPrice(
               ((basePrice / 10 ** 5) * cabs.luxury.multiplier).toFixed(5)
             );
           }}
         >
           <div className="items-center">
             <img
               className="h-10 w-8 mr-8"
               src={cabs.luxury.imgUrl}
               alt={cabs.luxury.name}
             />
           </div>
           <div className="mr-40">
```

```
              <h3 className="text-xl text-center font-semibold">
                {cabs.luxury.name}
              </h3>
            </div>
            <div className="flex items-center gap-6">
              <img src={eth_logo} alt="eth-logo" className="h-10 w-8" />
              <h3 className="text-xl font-bold">
                {((basePrice / 10 ** 5) * cabs.luxury.multiplier).toFixed(5)}
              </h3>
            </div>
          </button>
        </div>

        <div className="mt-6 w-full">
          <button
            className="flex border hover:border-black focus:border-black focus:bg-black focus:
            onClick={() => {
              setRide("basic");
              setFinalPrice(
                ((basePrice / 10 ** 5) * cabs.basic.multiplier).toFixed(5)
              );
            }}
          >
            <div className="items-center">
              <img
                className="h-10 w-8 mr-8"
                src={cabs.basic.imgUrl}
                alt={cabs.basic.name}
                />
            </div>
            <div className="mr-40">
              <h3 className="text-xl text-center font-semibold">
                {cabs.basic.name}
              </h3>
            </div>
            <div className="flex items-center gap-6 justify-end">
              <img src={eth_logo} alt="eth-logo" className="h-10 w-8" />
              <h3 className="text-xl font-bold">
                {((basePrice / 10 ** 5) * cabs.basic.multiplier).toFixed(5)}
              </h3>
            </div>
          </button>
        </div>

        <div className="mt-6 w-full">
          <button
            className="flex border hover:border-black focus:border-black focus:bg-black
            onClick={() => {
              setRide("economy");
              setFinalPrice(
                ((basePrice / 10 ** 5) * cabs.economy.multiplier).toFixed(5)
              );
            }}
          >
```

```
143             <div className="items-center">
144               <img
145                 className="h-10 w-8 mr-8"
146                 src={cabs.economy.imgUrl}
147                 alt={cabs.economy.name}
148               />
149             </div>
150             <div className="mr-40">
151               <h3 className="text-xl text-center font-semibold">
152                 {cabs.economy.name}
153               </h3>
154             </div>
155             <div className="flex items-center gap-6">
156               <img src={eth_logo} alt="eth-logo" className="h-10 w-8" />
157               <h3 className="text-xl font-bold">
158                 {((basePrice / 10 ** 5) * cabs.economy.multiplier).toFixed(5)}
159               </h3>
160             </div>
161           </button>
162         </div>
163
164         <button
165           className="w-full border-black bg-black text-white p-6 disabled:bg-gray-400"
166           disabled={walletConnected ? (ride === "" ? true : false) : true}
167           onClick={bookRide}
168         >
169           {walletConnected
170             ? ride === ""
171               ? "Pick an option to book ride."
172               : "Book ride."
173             : "Connect wallet to book ride."}
174         </button>
175       </div>
176     );
177   }
178
179   export default Pricing;
```

CHAPTER 11

Testing

In a peer-to-peer trip-sharing machine, various trying out stages are generally carried out to make sure the device's capability, security, and value. Right here are the main checking out phases concerned, at the side of a few sub-testing activities which might be typically completed inside each segment:

1. **Unit Testing:**
   - Check person additives or modules of the trip-sharing device.
   - Sub-trying out sports may additionally include checking out person functions, strategies, or instructions to verify their correctness and expected behavior.
2. **Integration testing:**
   - Check the interaction and compatibility between extraordinary additives of the machine.
   - Sub-trying out activities might also consist of testing the verbal exchange between the driver and passenger apps, the integration of charge gateways, or the interaction with outside offerings which includes mapping and navigation APIs.
3. **Gadget testing:**
   - Take a look at the general behavior and performance of the journey-sharing gadget as a whole.
   - Sub-testing activities may also consist of quit-to-stop checking out of the journey booking method, verifying the accuracy of fare calculation, testing system scalability beneath exclusive load conditions, and assessing system reaction time.
4. **Security testing**:
   - Take a look at the system for vulnerabilities and make certain the protection of sensitive records.
   - Sub-checking out activities may encompass penetration checking out to perceive capacity safety breaches, statistics encryption and decryption trying out, and get right of entry to manage checking out.
5. **Usability testing:**
   - Compare the gadget's person-friendliness, intuitiveness, and ease of use.

- Sub-checking out sports may also include assessing the readability of consumer interfaces, testing navigation and flow within the app, and carrying out person feedback periods to acquire insights on person experience.

6. **Overall performance testing:**
   - Assess the machine's performance underneath special load conditions and pressure stages.
   - Sub-checking out activities may additionally include load checking out to decide the device's maximum capability, strain trying out to assess gadget stability underneath heavy hundreds, and benchmarking against defined performance metrics.
7. **Acceptance testing:**
   - Validate whether or not the trip-sharing machine meets the required requirements and is ready for deployment.
   - Sub-checking out activities may also include consumer popularity checking out (UAT) where real users simulate various situations, verifying if the gadget satisfies their needs and meets their expectancies.

CHAPTER 12

FUTURE SCOPE

While certain aspects are still unclear, blockchain technology could have a transformative impact on how existing ride-sharing companies operate. Using Ethereum-based crypto assets, we understand the ongoing employment problems these companies face, characterized by numerous lawsuits regarding exploitative relationships and working practices with their employees. One of the most profound challenges is achieving transparency in the payment process. But by combining the principles and concepts of Bit coin and digital currencies. As part of our future plans, we are considering developing a mobile app to provide users with a seamless driving experience. It is compatible with all platforms and alleviates the common problems of web applications.

The Future scope of peer-to-peer experience-sharing structures is promising, with several ability trends and improvements at the horizon. Here are a few key areas that might shape the future of peer-to-peer journey sharing:

1. **Self sustaining cars:** The emergence of self-riding automobiles ought to revolutionize the peer-to-peer journey-sharing enterprise. As self sufficient generation improves and guidelines evolve, we are able to expect to peer self-driving motors being deployed for experience-sharing services. this would get rid of the want for human drivers and probably lessen costs, boom performance, and enhance safety.

2. **Electric and Sustainable vehicles:** The adoption of electric cars (EVs) in journey-sharing structures is gaining traction due to their environmental benefits and reducing expenses. within the destiny, we will expect to see a much wider deployment of EVs in peer-to-peer ride sharing, leading to decreased emissions and a greater sustainable transportation system.

3. **Integration with Public Transit:** Peer-to-peer experience-sharing structures can combine with public transit networks to offer seamless first-mile and last-mile transportation solutions. This integration can allow customers to combine ridesharing

with public transportation, making commuting greater convenient and reducing the reliance on personal vehicles.

4. **Multi-Modal alternatives:** The future of peer-to-peer trip-sharing might also contain integrating various transportation modes, inclusive of bikes, scooters, and public transportation, right into a single platform. customers may want to have access to a variety of transportation options within the equal app, bearing in mind greater flexible and efficient journey making plans.

5. **Stronger protection features:** destiny ride-sharing systems may additionally incorporate superior protection functions, including actual-time tracking, driver conduct analysis, and progressed identification verification. these measures can in addition enhance passenger protection and construct consider inside the peer-to-peer experience-sharing community.

6. **Blockchain and smart Contracts:** the use of block chain can deliver transparency, protection, and accountability to look-to-peer ride-sharing structures. those technologies can facilitate computerized price settlements, identity verification, and comfy sharing of records between participants.

7. **Personalization and AI assistance:** AI can play a giant role in improving the user experience of peer-to-peer ride-sharing. AI algorithms can examine user preferences, tour styles, and ancient facts to provide personalised guidelines, optimize routing, and improve the general efficiency of the system.

8. **Environmental and Social effect:** As sustainability and social obligation come to be more crucial, future peer-to-peer experience-sharing systems may additionally prioritize motors with decrease emissions, incentivize carpooling, and aid community-driven tasks targeted on lowering congestion and selling shared mobility.

CHAPTER 13

CONCLUSION

13.1 Conclusion:

While current rides sharing platforms are being widely used, there are areas that can be enhanced, such as pricing models, user safety, transaction transparency, privacy, and data security. Blockchain-based systems present an opportunity to address these concerns and introduce innovative features that are more user-friendly and manageable. By eliminating intermediaries, blockchain applications can ensure precise and equitable pricing, privacy-conscious applications, secure data handling, and transparent transactions, while also automating intermediary tasks like payment processing through smart contracts.

Furthermore, blockchain technology offers decentralized control, which reduces the reliance on a single entity for managing and governing the ride-sharing platform. This decentralized nature increases trust among users by providing a distributed ledger that records all transactional data in a transparent and immutable manner. Additionally, smart contracts executed on the blockchain can streamline processes such as driver verification, ride payments, and resolving disputes, thereby enhancing the overall efficiency and reliability of the ride-sharing experience. By leveraging blockchain-enabled systems, the ride-sharing industry can strive towards a safer, more transparent, and user-centric ecosystem.

13.2 Result

The very last result of a peer-to-peer ride-sharing device challenge can range relying at the specific dreams and requirements of the challenge. however, here are a few key elements which are typically anticipated within the very last end result:

1. **Practical ride-Sharing Platform:** The task must supply a fully useful experience-sharing platform that connects drivers and passengers. The platform ought to permit passengers to request rides, drivers to accept ride requests, and facilitate the entire journey reserving procedure.

2. **User-friendly cell programs:** The venture need to consist of properly-designed and user-friendly cellular packages for both drivers and passengers. these packages ought to provide intuitive interfaces, easy navigation, and seamless conversation between customers.

3. **Ggreen Matching and Routing:** The trip-sharing machine need to put in force efficient algorithms for matching passengers with appropriate drivers based totally on factors consisting of region, vacation spot, and availability. additionally, the device must provide most beneficial routing and navigation guidelines to make certain efficient journey.

4. **Charge and Fare Calculation:** The undertaking should include a comfy and dependable price system that permits passengers to pay for rides and drivers to receive charge. The fare calculation need to be correct, obvious, and don't forget factors like distance, time, and any extra fees.

5. **Score and feedback gadget:** A strong score and comments device must be applied to permit passengers to rate drivers and provide remarks on their ride revel in. This facilitates keep best requirements, enhance accountability, and construct agree with within the ride-sharing network.

6. **Ssafety and privateness Measures:** The experience-sharing device should prioritize the security and privateness of person information. It need to put into effect measures

together with at ease user authentication, records encryption, and compliance with relevant statistics protection regulations.

7. **Administrative Dashboard:** An administrative dashboard or panel have to be protected to allow directors to manipulate the journey-sharing machine efficiently. This dashboard provides tools for tracking device hobby, handling consumer debts, resolving disputes, and gaining access to essential analytics and reports.

8. **Ttrying out and high-quality assurance:** The final end result should encompass thorough checking out and pleasant warranty approaches to make certain that the experience-sharing system functions as supposed, is reliable, and provides a great person revel in. trying out should cowl regions consisting of functionality, overall performance, safety, and usability.

9. **Documentation and assist:** The task need to provide comprehensive documentation that consists of user manuals, technical guides, and any necessary documentation for system directors. Additionally, there ought to be ongoing support channels to help users and address any troubles or questions that may get up.

REFERENCES

1. https://www.researchgate.net/publication/335241161_Blockchain_Based_Car-Sharing_Platform
2. https://www.ijert.org/a-survey-of-peer-to-peer-ride-sharing-services-using-blockchain
3. https://devpost.com/software/decentralized-uber
4. https://www.ijsr.net/archive/v11i6/SR22608100338.pdf
5. https://www.academia.edu/download/69306330/a_survey_of_peer_to_peer_ride_sharing_services_IJERTV10IS080172.pdf

plag report

ORIGINALITY REPORT

7%	6%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	yournameonmyskin.skyrock.com Internet Source	3%
2	www.coursehero.com Internet Source	2%
3	Submitted to University of Greenwich Student Paper	<1%
4	Submitted to University of Ulster Student Paper	<1%
5	Submitted to Kwame Nkrumah University of Science and Technology Student Paper	<1%
6	"Data Intelligence and Cognitive Informatics", Springer Science and Business Media LLC, 2023 Publication	<1%
7	www.tib.eu Internet Source	<1%
8	"Web and Wireless Geographical Information Systems", Springer Science and Business	<1%