

# **V-Rides : The Collage Bicycle Rental Website**

## **A PROJECT REPORT**

*Submitted by*

Akshat Rastogi	22BAI10031
Karwa Tamsa Sandeep	22BAI10300
Sunnam Mehta	22BAI10068
Shubh Gupta	22BAI10084
Wrishita Paul	22BAI10226

*In partial fulfilment for the award of the degree  
of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**(ARTIFICIAL ENGINEERING AND MACHINE LEARNING)**



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRI KALAN, SEHORE**

**MADHYA PRADESH - 466114**

October 2023

**VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE  
MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**V-Rides : In Campus Rental Bicycle Website**” is the bonafide work of **Karwa Tamsa Sandeep (22BAI10300), Akshat Rastogi (22BAI10031), Shubh Gupta (22BAI10084), Sunnam Mehta (22BAI10068), Wrishita Paul (22BAI10226)** who carried out the project work under my supervision. Certified further that to the best of my knowledge, the work reported here does not form part of any other project/research work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROJECT SUPERVISOR**

Dr. Anjali Mathur , Assistant Professor  
School of Computing Science and Engineering  
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on \_\_\_\_\_

## **ACKNOWLEDGEMENT**

First and foremost we would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

We wish to express my heartfelt gratitude to Dr. Pradeep Kumar Mishra, Program Chair-Division of AIML, School of Computer Science and Engineering for much of his valuable support encouragement in carrying out this work.

We would like to thank my internal guide , Dr. Anjali Mathur , for continually guiding and actively participating in my project, and giving valuable suggestions to complete the project work.

We would like to thank all the technical and teaching staff of the School of Aeronautical Science, who extended directly or indirectly all support.

Last, but not the least, we are deeply indebted to our parents who have been the greatest support while we worked day and night for the project to make it a success.

# ABSTRACT

This comprehensive project report offers a detailed account of V-Rides, a college bike rental system, with a specific focus on the technical architecture and website development. The study delves into the intricate technical underpinnings, showcasing the extensive use of HTML and CSS for frontend design and SQL for robust database management. Central to the project's functionality, PHP is employed for server-side operations, ensuring the system's efficiency and reliability.

The report charts the evolution of V-Rides, commencing with its conceptualization and culminating in the integration of advanced features. Notable features include QR code scanning for streamlined bike access, an inbuilt wallet for user convenience, advanced booking options, and a distinctive logo incorporating a stylized "V" with an integrated bicycle. These features are dissected in the context of their technical implementation, demonstrating their pivotal role in enhancing user experience.

Key features of V-Rides include a QR code-based locking and unlocking mechanism, flexible rental durations, multiple payment options, real-time bicycle health monitoring, user profiles, an inbuilt wallet system for seamless transactions, pre-booking capabilities, and the ability to make multiple bicycle bookings. These features come together to offer a comprehensive solution that caters to the diverse needs of the student population.

V-Rides leverages modern web technologies, with a front-end developed using CSS, JavaScript, and HTML to ensure an intuitive and visually appealing user interface. The back-end is powered by PHP, providing robust functionality and server-side processing. SQL is used for database management, ensuring the safe and efficient storage of user data and rental information.

This project report serves as a testament to the intersection of cutting-edge web technologies and sustainable transportation solutions. It offers a blueprint for similar initiatives, showcasing the transformative power of technology in modern campus mobility. By delving into the intricacies of V-Rides, this report sheds light on the technical prowess that underpins an eco-friendly and user-centric transportation system, promising a brighter and more sustainable future for college campuses.

In summary, the "V-Rides" project represents an innovative and sustainable solution to address transportation issues within college campuses, promoting eco-friendly mobility options while enhancing the overall campus experience. This report serves as a comprehensive guide to the development and implementation of a college rental bicycle website and offers valuable insights for future projects in this domain.

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	List of Abbreviations List of Figures List of Tables	
1	<b>INTRODUCTION</b> 1.1 Introduction 1.2 Motivation for the work 1.3 Introduction to the project 1.4 Problem Statement 1.5 Objective of the work 1.6 Summary	
2	<b>LITERATURE REVIEW</b> 2.1 Introduction- Beginning of the era of online delivery 2.2 Online Platforms for bike 2.3 Rental Bike Platform 2.4 Benefits and Goals 2.5 Summary	

3	<b>SYSTEM ANALYSIS</b>  3.1 Introduction  3.2 Key features  3.3 System Component  3.4 Summary	
4	<b>SYSTEM DESIGN AND IMPLEMENTATION</b>  4.1 Introduction  4.2 Dataset design & implementation 4.2.1 Logo Design 4.2.2 User Registration and Authentication 4.2.3 User Account Dashboard and Management 4.2.4 QR code scanning 4.2.5 Bicycle management 4.2.6 Admin Dashboard Website 4.2.7 Feedback Page 4.2.8 Database management 4.2.9 Security and safety compliance  4.3 Summary	

5	<b>PERFORMANCE ANALYSIS</b>  5.1 Introduction 5.2 Performance Measures 5.3 Summary	
---	--	--

6	<b>FUTURE ENHANCEMENT AND CONCLUSION</b>  6.1 Introduction 6.2 Future Enhancements 6.3 Conclusion	
7	Appendix	
8	References	

## **LIST OF ABBREVIATIONS**

<b>HTML</b>	<b>Hypertext Markup Language</b>
<b>CSS</b>	<b>Cascading Style Sheets</b>
<b>SQL</b>	<b>Structured Query Language</b>
<b>PHP</b>	<b>Hypertext Preprocessor</b>



**LIST OF FIGURES**

<b>FIGURE NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
1	Logo	
2	Flowchart	
3	Database	

# INTRODUCTION

## 1.1 Introduction

In an era characterized by urbanization, environmental consciousness, and the pursuit of sustainable solutions, the need for innovative and eco-friendly transportation alternatives is more significant than ever. College campuses, often bustling hubs of activity and learning, are no exception to this trend. Recognizing the pressing need for an efficient and environmentally responsible mode of commuting within the campus, we have undertaken the development of a college rental bicycle website, hereafter referred to as "V-Rides."

## 1.2 Motivation for the Work

The motivation behind the creation of the V-Rides project is multifaceted. It stems from the growing environmental concerns, congestion issues on college campuses, and the desire to offer students and faculty a convenient, sustainable, and affordable mode of transportation. Bicycle-sharing systems have gained prominence globally as a means to reduce carbon emissions, and promote physical activity. Our project draws inspiration from these successes and seeks to implement a tailored solution designed for the unique needs of college communities.

## 1.3 Introduction to the Project

The "V-Rides" project, at its core, is a web-based platform that simplifies the process of renting and utilizing bicycles within a college campus. Through this platform, registered students and staff can seamlessly rent, track, and return bicycles, fostering a culture of sustainable transportation on their campus. By offering real-time bicycle availability information, route guidance, and user-friendly booking management, we aim to transform the way individuals move around their academic environment.

## 1.4 Problem Statement

The modern college campus faces several transportation-related challenges, including parking shortages, congestion, and an increased carbon footprint due to the reliance on motorized vehicles. Our project recognizes these issues as an opportunity for improvement. We aim to address these problems by providing an accessible and convenient means of transportation via a rental bicycle system tailored to the specific needs of college campuses.

## 1.5 Objective of the Work

The primary objective of this project is to develop and implement a user-friendly college rental bicycle website. This platform will not only simplify the process of renting bicycles but also promote sustainable transportation practices, reduce the environmental impact, and enhance the overall campus experience. Specific goals include creating a secure and efficient system, ensuring real-time bicycle tracking, and providing guidance to users on the best routes for their journeys.

## 1.6 Summary

In summary, the "V-Rides" project seeks to address the transportation challenges faced by modern college campuses by providing an innovative solution that promotes eco-friendly mobility options. This project report will delve into the development process, technologies employed, database structure, security measures, and user experience optimization. It also outlines the challenges faced during development and highlights potential future enhancements to the platform. The subsequent sections of this report will provide a detailed account of the design, development, and implementation of this sustainable campus transportation solution.

## **LITERATURE REVIEW**

### **2.1 Introduction :**

Beginning of the era of online delivery:

The emergence of Zomato and Swiggy in the early 2010s marked a transformative moment in the food and technology industries in India. These two startups, both founded in 2008, embarked on a journey that would redefine the way people in India ordered and experienced food.

Zomato, originally known as "Foodiebay," was established by Deepinder Goyal and Pankaj Chaddah in July 2008. The idea was born out of a personal frustration with the lack of comprehensive restaurant information available online. The founders saw an opportunity to create a platform that would offer detailed restaurant listings, menus, and user reviews, catering to the burgeoning urban dining culture in India. Zomato's journey began in Delhi, where the founders began by collecting menu cards and taking pictures of restaurant exteriors to build their database. Their dedication to providing accurate and up-to-date information quickly set them apart from traditional restaurant directories. They expanded their operations to other major cities, creating a nationwide network of restaurant listings. The company's first major turning point came in 2011 when it rebranded as "Zomato." The new name not only reflected the company's core mission, but it also marked the beginning of their international expansion. Zomato ventured into foreign markets, making its presence felt in countries across Asia, Europe, and Africa.

In 2014, Nandan Reddy, Rahul Jaimini, and Sriharsha Majety founded Swiggy in Bangalore, India. Swiggy aimed to tackle the challenge of food delivery, a space previously dominated by traditional phone orders and limited delivery services. The founders recognized the untapped potential in providing a seamless and technology-driven solution for ordering food from a wide range of restaurants. Swiggy's success was largely attributed to its unique approach to food delivery. The company built a robust delivery fleet and employed advanced technology to optimize delivery routes and reduce delivery times. This, combined with a diverse restaurant network, quickly made Swiggy a popular choice for those looking for convenient and timely food delivery. Swiggy's rapid growth caught the attention of investors, and the company secured significant funding rounds, allowing it to expand to multiple Indian cities and compete head-to-head with established players like Zomato.

The emergence of Zomato and Swiggy had profound implications for the Indian food industry and beyond. These startups leveraged technology to create a more efficient and convenient way

for consumers to discover and order food. They facilitated the growth of the restaurant industry by connecting them with a broader customer base and enabling them to offer delivery services without building their own infrastructure. The success of Zomato and Swiggy also inspired a wave of food-tech startups in India, highlighting the potential for innovation in a traditionally offline industry. It led to increased competition, further innovations in food delivery and online ordering, and ultimately improved services for consumers.

These platforms not only changed the way Indians dine and order food but also left a lasting impact on the global food delivery and technology landscape. Their stories serve as a valuable case study for understanding how startups can transform traditional sectors through innovation and customer-centric solutions.

## 2.2 Online Platforms for bikes :

The emergence of online platforms for buying and selling bikes in India has followed a similar trajectory to the global trend, albeit with some unique regional characteristics and market dynamics. In India, the online marketplace for motorcycles has witnessed significant growth over the past decade, fueled by a combination of factors, including the increasing popularity of two-wheelers as a mode of transportation and recreation, technological advancements, and changing consumer preferences.

The rise of online platforms for bike sales in India began with the proliferation of e-commerce and online classified websites in the mid-2000s. Websites like OLX and Quikr, which started as general classified platforms, saw a substantial number of motorcycle listings. These platforms provided a convenient way for individuals and dealers to advertise their bikes, reach a broader audience, and connect with potential buyers. The shift towards digital payment methods and the expansion of internet connectivity across the country further facilitated this trend.

In the late 2000s and early 2010s, India saw the emergence of dedicated online marketplaces that focused specifically on the sale of motorcycles. Platforms like BikeWale, BikeDekho, and Droom were established to cater to the unique needs of bike enthusiasts and buyers. These websites offered features such as detailed bike specifications, expert reviews, and price comparison tools, making it easier for consumers to research and choose the right motorcycle. Additionally, they provided a secure environment for transactions, with options for inspecting and test-riding bikes before purchase.

Mobile applications also played a pivotal role in the growth of online bike marketplaces in India. As smartphone penetration increased, mobile apps provided a user-friendly and on-the-go experience for both buyers and sellers. These apps often incorporated location-based services, helping buyers find motorcycles available in their vicinity.

Peer-to-peer platforms like Facebook Marketplace and classified listings on websites such as Quikr and OLX also gained prominence in India, allowing individuals to buy and sell bikes

directly. This approach provided an extensive range of choices, from budget-friendly commuter bikes to high-end sports motorcycles.

Furthermore, the popularity of electric scooters and motorcycles in India, driven by environmental concerns and government incentives, has given rise to online platforms specifically dedicated to electric two-wheelers. These platforms are at the forefront of promoting sustainable transportation alternatives.

In conclusion, the emergence of online platforms for buying and selling bikes in India reflects the broader trend of digitization and e-commerce adoption in the country. These platforms have made the process of purchasing and selling motorcycles more convenient, transparent, and accessible to a diverse range of buyers and sellers across India. As technology continues to advance and consumer preferences evolve, the online motorcycle marketplace in India is likely to continue growing and evolving to meet the unique demands of this dynamic market.

### 2.3 Rental Bike Platforms:

In India, the rental bike platform sector has witnessed significant growth and innovation in recent years, catering to the needs of both commuters and tourists. These platforms offer a convenient and flexible way to access bikes for various purposes, whether it's daily commuting, leisure rides, or exploring new destinations. Here are some of the notable rental bike platforms operating in India:

1. **Bounce:** Bounce, formerly known as Metro Bikes, is one of the largest bike rental platforms in India. They offer a diverse fleet of two-wheelers, including scooters and motorcycles. Users can locate and unlock bikes using their mobile app, making it a popular choice for short commutes in cities like Bangalore and Hyderabad. Bounce also introduced a unique "keyless" system, allowing users to start the rented bike using their smartphone.
2. **Vogo:** Vogo is another prominent bike rental service that primarily operates in cities like Bangalore and Hyderabad. Vogo offers a range of scooters that are well-suited for city commuting. Users can book scooters through their mobile app and access them at designated pickup and drop-off points. Vogo has gained popularity for its affordable pricing and ease of use.
3. **Revv:** While Revv is known for car rentals, they also offer two-wheeler rentals in select cities. They provide a range of bikes, including scooters and motorcycles. Revv's model allows users to rent bikes by the hour, day, week, or even on a monthly basis, making it suitable for various purposes, from short rides to long-term usage.
4. **Royal Brothers:** Catering to the growing demand for bike rentals, Royal Brothers operates in multiple cities across India, including Bangalore, Delhi, and Goa. Their platform offers a variety of motorcycles and scooters, appealing to both local commuters and tourists. Users can reserve bikes through their website or mobile app and opt for flexible rental durations.

5. ONN Bikes: ONN Bikes is a bike rental platform available in multiple cities across India, such as Bangalore, Hyderabad, and Jaipur. They provide a wide selection of two-wheelers, including scooters, sports bikes, and cruisers. ONN Bikes offers hourly, daily, and monthly rental options, making it suitable for different needs.

6. Drivezy: While primarily known for car rentals, Drivezy also offers bike rentals in select cities. Users can book a motorcycle or scooter through their platform and enjoy flexible rental periods. Drivezy's presence in several Indian cities makes it a convenient option for those seeking both cars and bikes.

7. Tazzo: Tazzo is a bike rental service primarily operating in cities like Bangalore and Hyderabad. They offer a variety of scooters and motorcycles for daily and monthly rentals. Tazzo's user-friendly mobile app simplifies the booking and pick-up process.

These rental bike platforms have transformed the way people in India access and use two-wheelers, providing a cost-effective and convenient alternative to ownership. With the increasing popularity of these services, it's likely that more regions will see the expansion of rental bike platforms, contributing to the overall growth of shared mobility in India.

#### 2.4 Limitations of the online bike rental systems:

While rental bike platforms in India have undoubtedly brought convenience and flexibility to users, they do come with certain limitations and challenges, including the absence of advanced features like QR code scanning. Here are some of the key limitations of these platforms:

1. No QR Code Scanning: One notable limitation is the absence of QR code scanning or other advanced technology features that could enhance user experience. QR code scanning can be a convenient way to unlock and access rented bikes, simplifying the rental process and reducing the need for physical keys or keyless start systems.

2. Security Concerns: Without advanced features like QR code scanning, rental bike platforms may face security concerns. Some platforms use less secure methods like physical keys, which could lead to theft or misuse of rented bikes. Enhanced security measures, such as biometric verification or more robust anti-theft systems, are needed.

3. Limited Geographic Availability: Rental bike platforms are primarily concentrated in major urban centers, leaving residents of smaller towns and rural areas with limited access to such services. Expanding to a wider geographic area is challenging due to logistical and infrastructural constraints.

4. Maintenance and Quality Control: Ensuring the quality and maintenance of the rental bikes can be a significant challenge. Regular maintenance and safety checks are crucial, but not all platforms may invest adequately in these areas, potentially leading to safety concerns for users.

5. Lack of Helmet Compliance: In many cases, rental bike platforms do not provide helmets or enforce helmet usage, which is a critical safety aspect, especially in a country like India. Promoting helmet compliance among users is essential for safer riding.

6. Limited Model Variety: While some platforms offer a range of bikes, many primarily focus on scooters, leaving motorcycle enthusiasts with fewer options. A broader variety of models and brands could cater to a more diverse audience.

7. Traffic and Safety Awareness: Rental bike users may not always be well-versed in local traffic rules and regulations, potentially leading to unsafe riding practices. More extensive user education on traffic and safety norms is needed.

8. Challenges with Parking: Finding safe and convenient parking spots for rental bikes can be challenging, particularly in crowded urban areas. Platforms need to work on solutions to address this issue, such as designated parking zones.

9. Pricing and Affordability: Rental bike platforms can be cost-effective for short rides, but long-term rentals may become expensive. More flexible and affordable pricing structures for extended usage would be beneficial.

10. Weather and Environmental Factors: Adverse weather conditions, particularly during the monsoon season, can limit the use of rental bikes. Platforms need to consider solutions for weather-related challenges, such as providing weatherproof gear.

Despite these limitations, the growth of rental bike platforms in India signifies a promising trend in shared mobility. Overcoming these challenges and incorporating advanced features like QR code scanning can lead to improved user experiences and greater adoption of such services, making them even more accessible and attractive for a broader range of users.

## 2.5 Summary:

In conclusion, while rental bike platforms in India have brought significant convenience and flexibility to users, they do face several limitations that need to be addressed to further enhance their appeal and safety. The absence of advanced features like QR code scanning, security concerns, and geographic limitations present challenges that the industry must tackle.

To overcome these limitations, the rental bike platforms can invest in technological advancements to improve security and user experience. Implementing features like QR code scanning and more secure locking systems can streamline the rental process and enhance bike security. Moreover, expanding their services to a wider geographic area and promoting safety awareness and compliance with traffic rules are essential steps toward ensuring a safer and more accessible experience for all users.

Additionally, addressing challenges related to maintenance, parking, affordability, and weather conditions can make rental bike platforms more practical and user-friendly. These platforms have the potential to significantly contribute to the shared mobility landscape in India,



providing an eco-friendly and cost-effective transportation alternative. By continuously innovating and overcoming these limitations, rental bike platforms can continue to play a vital role in the evolving transportation ecosystem in the country.

## **System Analysis for V-Rides - College Bike Rental System**

### **3.1 Introduction:**

V-Rides is a comprehensive bike rental system designed to cater to the specific needs of college students, offering convenient and cost-effective mobility solutions. This system leverages modern technology and innovative features to provide students with a hassle-free and eco-friendly mode of transportation on and around campus.

### **3.2 Key Features:**

#### **1. QR Code Scanning:**

V-Rides will feature QR code scanning capabilities, allowing users to unlock and access their rented bikes using their smartphones. This feature enhances the ease and speed of bike pickup and return, eliminating the need for physical keys or cumbersome processes.

#### **2. Birthday Offer:**

To encourage student engagement and loyalty, V-Rides can offer a special birthday promotion. On their birthdays, registered users could enjoy a discounted or free ride, making their day even more memorable.

#### **3. Hourly Charging:**

V-Rides will implement an hourly charging system, ensuring that students pay only for the time they use the bike. This flexible pricing model caters to different usage scenarios, from short trips around campus to longer rides.

#### **4. Inbuilt Wallet:**

The system will include an inbuilt wallet feature that allows users to load funds into their V-Rides account. This wallet will facilitate seamless and quick payments for bike rentals, eliminating the need for cash transactions.

## 5. Advanced Booking:

To provide greater convenience, V-Rides can offer an advanced booking feature. Students can reserve a bike for a specific time and date in advance. This ensures they have a bike available when needed, especially during peak hours.

## 3.3 System Components:

### 1. User Interface:

V-Rides will feature a user-friendly mobile application and a web interface. Users can register, log in, and manage their accounts through these platforms. The app will also allow users to scan QR codes, check bike availability, and make reservations.

### 2. Database Management:

The system will maintain a robust database to store user profiles, bike availability, transaction history, and wallet balances. This database ensures seamless data retrieval and storage.

### 3. QR Code Scanning Module:

The QR code scanning module will be an integral part of the system, allowing users to unlock and lock their rented bikes securely.

### 4. Payment Processing:

The system will integrate with various payment gateways to facilitate wallet top-ups and payment processing for bike rentals.

### 5. Bike Fleet Management:

V-Rides will include a fleet management module that tracks the condition, location, and maintenance schedule of bikes. This helps ensure that the bikes are in good working order and readily available to users.

## 3.4 Benefits and Goals:

1. Convenience: V-Rides aims to provide students with a convenient and affordable means of transportation within the college campus, reducing the need for personal vehicles.

2. Cost-Efficiency: The hourly charging system and wallet feature make bike rentals cost-effective, enabling students to pay only for the time they use.

3. Safety: Advanced booking and the inbuilt wallet reduce the need for cash transactions and help ensure a secure and safe rental experience.

4. Promotions and Loyalty: The birthday offer and other promotions can enhance user engagement and foster loyalty among college students.

5. Environmental Impact: V-Rides promotes eco-friendly transportation by encouraging bike usage, reducing the carbon footprint on campus.

### 3.5 Summary:

In summary, V-Rides, the college bike rental system, promises to be a game-changer in student transportation. By incorporating advanced features like QR code scanning, exclusive birthday offers, flexible hourly charging, an inbuilt wallet, and advanced booking options, V-Rides offers a comprehensive and student-centric solution. This system not only simplifies access to bikes but also aligns with the larger trend of eco-friendly and cost-effective mobility solutions.

V-Rides seeks to enhance the college experience by providing a seamless, secure, and engaging mode of transportation on campus, reducing the need for personal vehicles and contributing to a more sustainable environment. The system's innovative features, alongside its promotions and emphasis on student safety, are poised to create a dynamic and connected campus community.

In essence, V-Rides is not just a bike rental system; it's a transformative approach to college transportation, designed to offer students greater convenience and promote a greener, more affordable, and efficient campus lifestyle while fostering a strong sense of community among its users.

## SYSTEM DESIGN AND IMPLEMENTATION

### 4.1 Introduction

The system design and implementation phase is the heart of the "V-Rides" project. In this section, we will outline the design and implementation of key modules, each contributing to the successful development of our bicycle rental platform.

### 4.2. Dataset Design and Implementation:

#### 4.2.1 Logo Design:

The V-Rides logo is a striking representation of the system, featuring a prominent "V" with a bicycle incorporated within it. This design communicates the essence of V-Rides - offering a sustainable, efficient mode of transportation within a college community. The letter "V" symbolizes victory, vitality, and versatility, aligning with the energetic nature of college life. The bicycle within the logo emphasizes the eco-friendly aspect of V-Rides.



Figure : Logo

#### 4.2.2 User Registration and Authentication:

Design: User registration and authentication pages are designed for user-friendliness and security. Users are prompted to provide essential information, such as name, contact details, and a secure password. Email id of the university is a part of the process for identity confirmation.

Implementation: The registration process includes email verification of the university to ensure the authenticity of user accounts. Stringent password policies and encryption measures are implemented for enhanced security. Users can also choose the convenience of staying logged in or can easily log out.

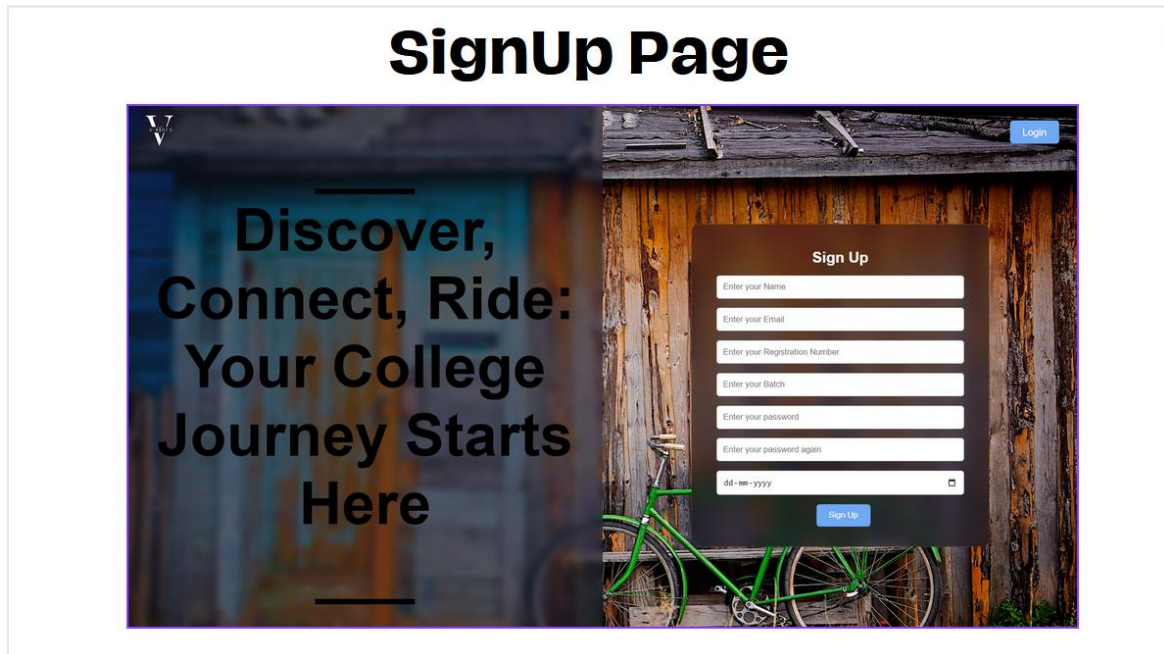


Figure : Signup Page

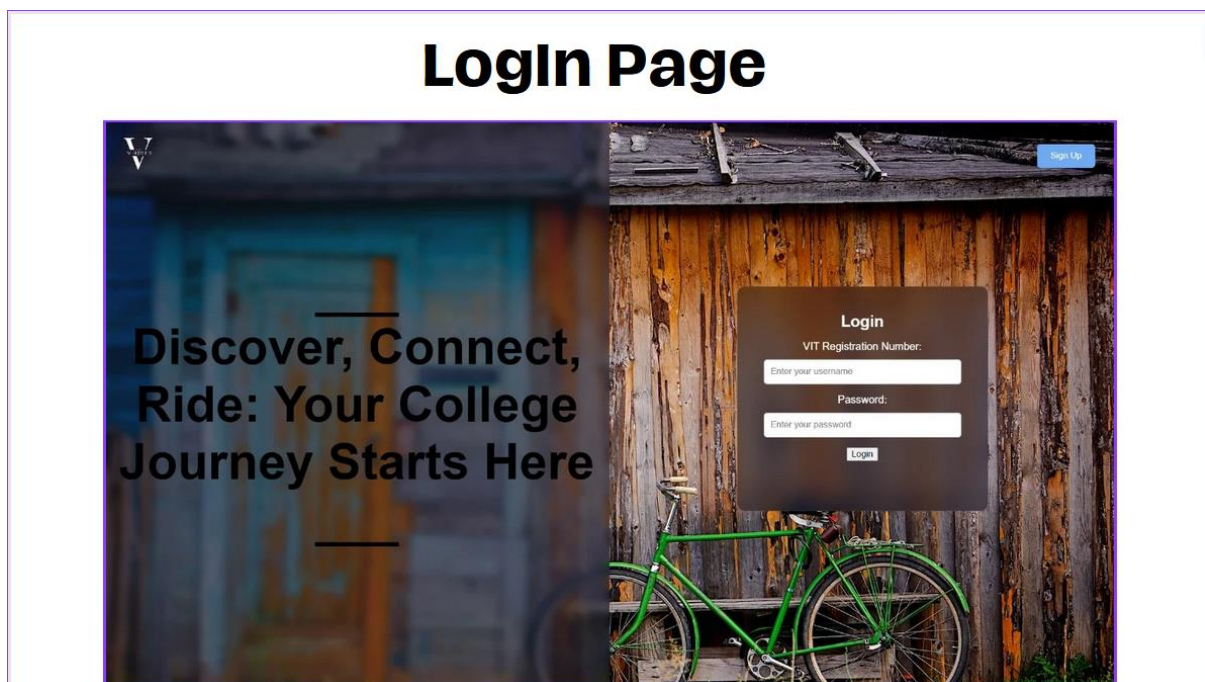


Figure : Login Page

### 4.2.3 User Account Dashboard and Management:

**Design:** The user account dashboard is intuitively designed, displaying essential account details, wallet balance, ride history, and booking status. User profiles, payment methods, and ride records are easily accessible through clear tabs.

**Implementation:** User account data is securely stored, and the wallet transaction history is logged for transparency. Users can conveniently recharge their accounts and review past ride details in a centralized location.

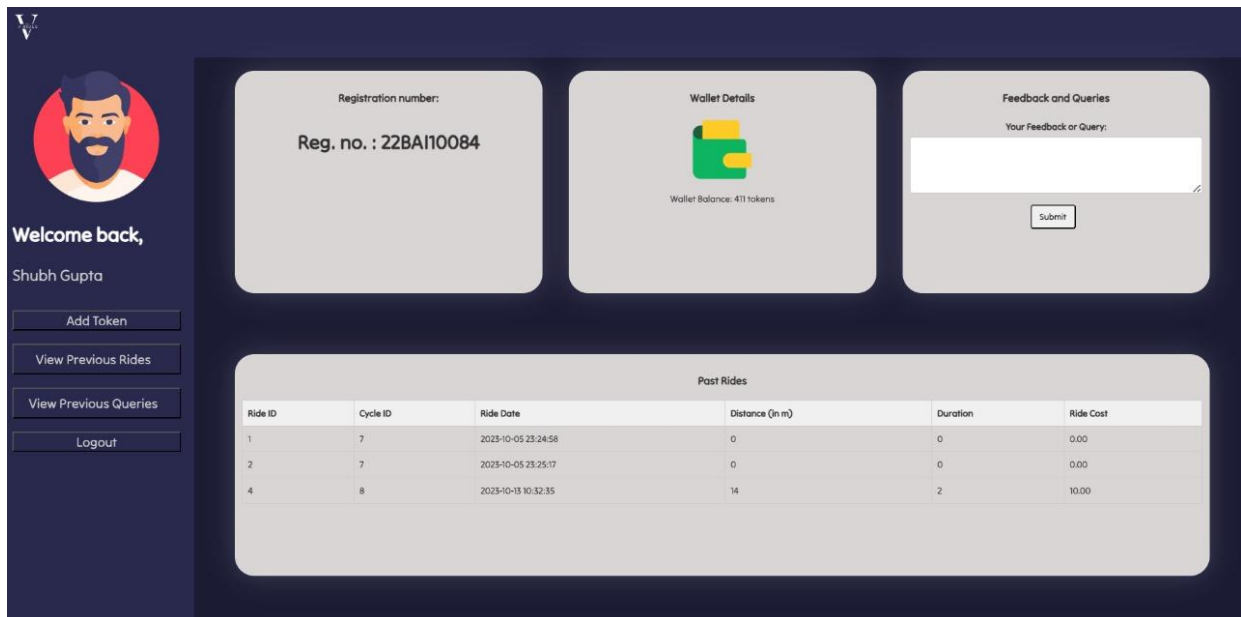


Figure : UserDashboard

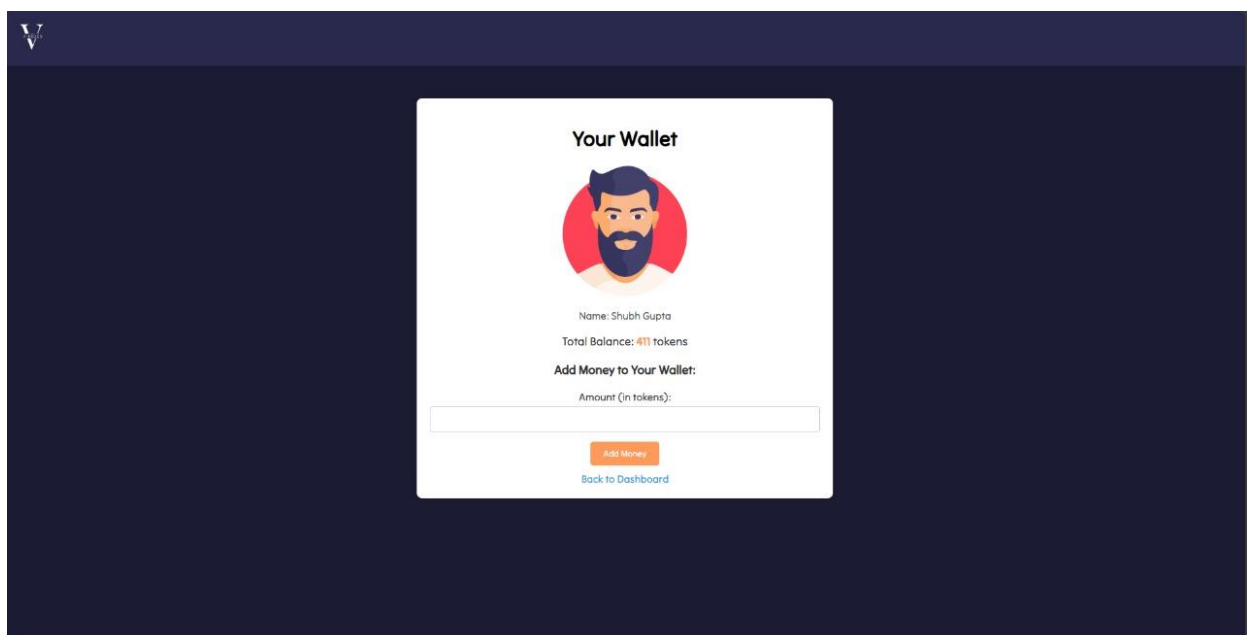


Figure : E-Wallet

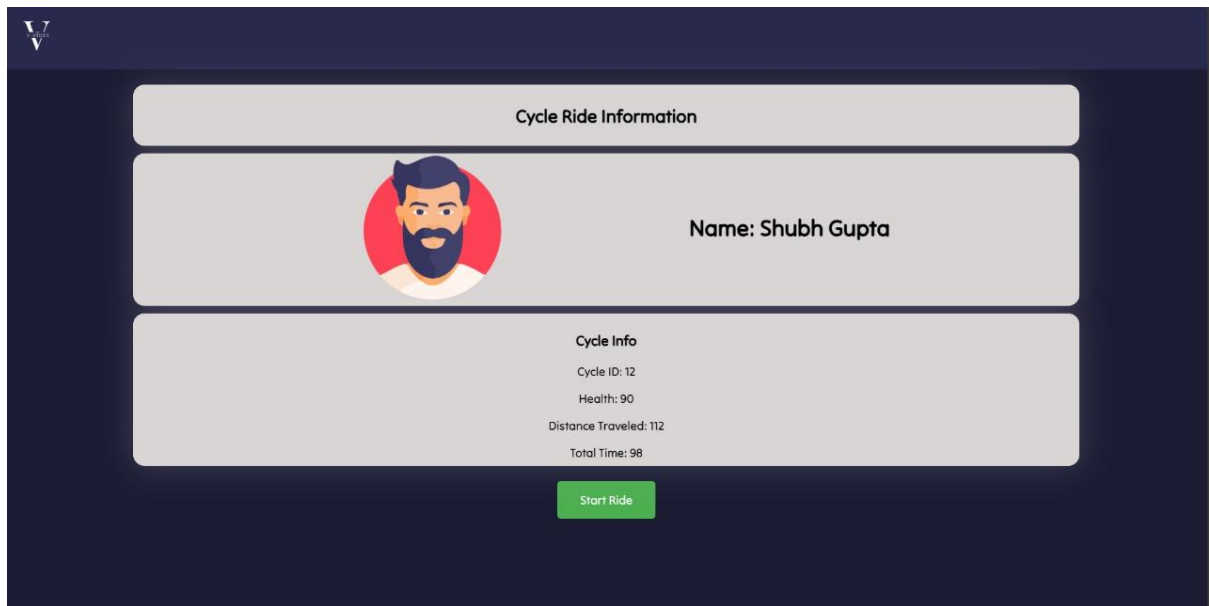


Figure : Cycle Ride Information

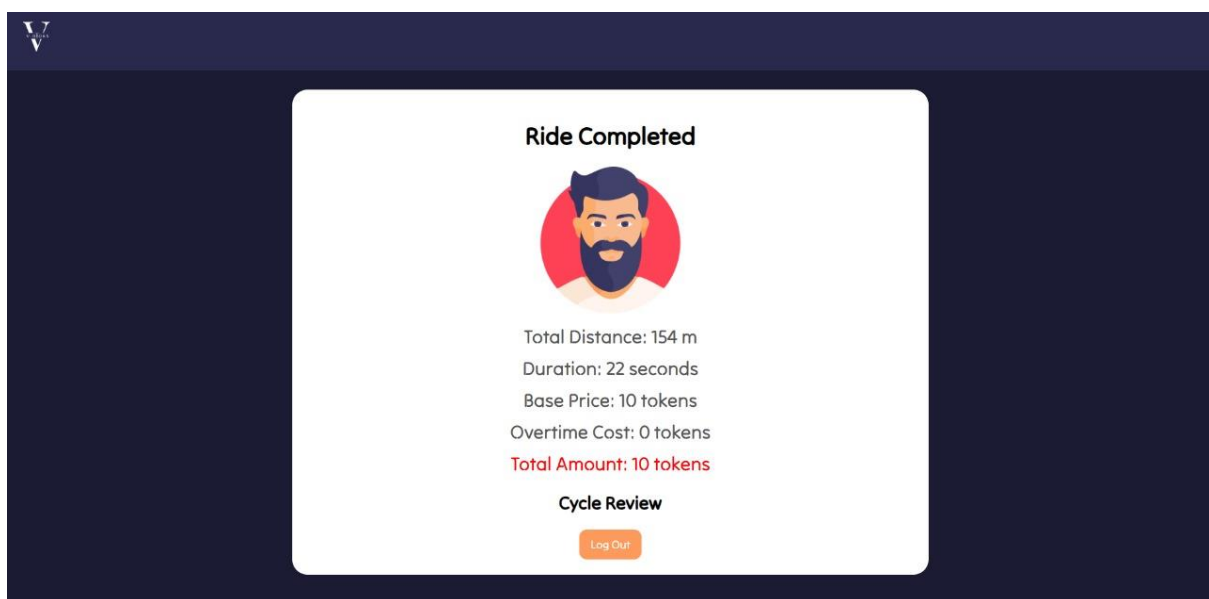


Figure : Ride Completion Information

#### 4.2.4 QR Code Scanning:

Design: The QR code scanning feature is easily accessible from the main dashboard. It provides clear instructions for users to initiate the scanning process through a built-in camera interface.

Implementation: Smartphone cameras are used to scan QR codes on the bikes. The scanned data is verified against the database for authentication, ensuring a quick and secure bike access process.

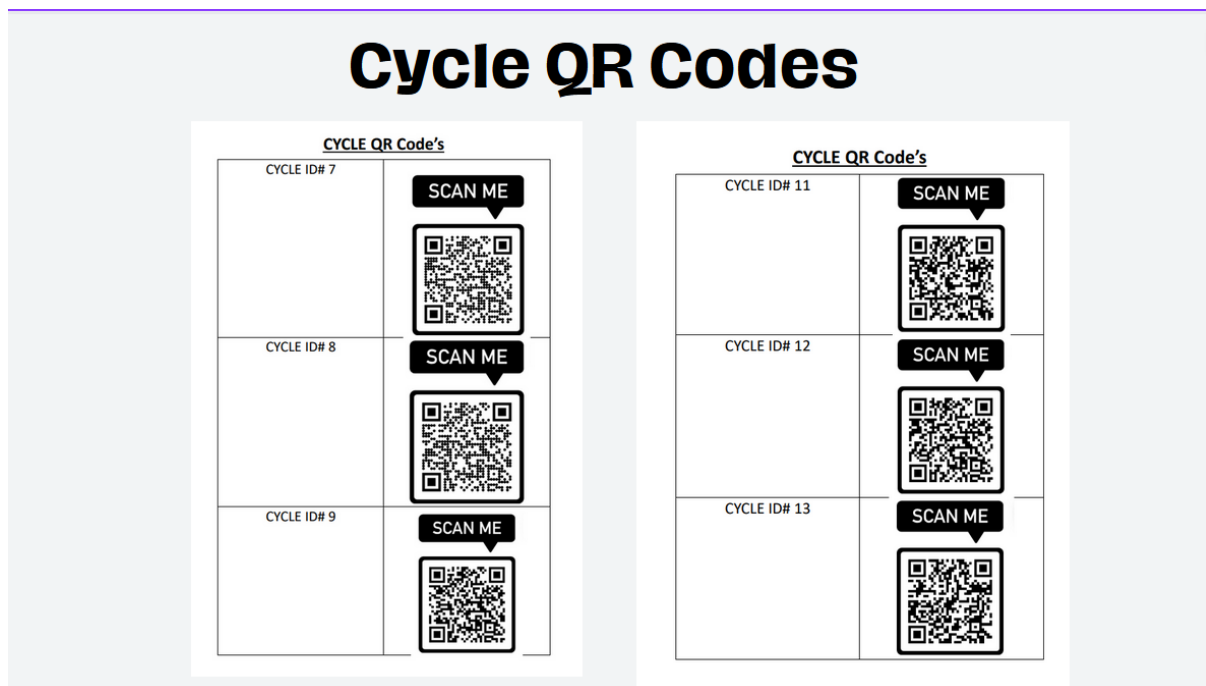



Figure : QR Scanner

#### 4.2.5 Bicycle Management:

Design: Bicycle management is seamlessly integrated into the admin dashboard, which is accessible only to authorized personnel. It offers a comprehensive overview of bike status, including location, maintenance records, and availability.

Implementation: Real-time GPS tracking is used to monitor the location of bikes. Scheduled maintenance and maintenance alerts are implemented to ensure bikes are in optimal condition. Inventory management guarantees the availability of bikes on the platform.





### All Cycle Information


Total Number of Cycles: 9

[Add New Cycle](#)
[Maintenance](#)

Cycle ID	Health	Distance Traveled	Total Time	Cycle Link	Total Rides	Action
7	100	0	529	<a href="#">28ec37416406</a>	5	<a href="#">Delete</a>
8	98	84	218	<a href="#">dc9b3d514d67</a>	6	<a href="#">Delete</a>
11	98	91	28	<a href="#">31f59caf0d0d</a>	3	<a href="#">Delete</a>
12	92	84	94	<a href="#">7bfe773fe31</a>	8	<a href="#">Delete</a>
13	84	203	125	<a href="#">76fd7e1d7737</a>	37	<a href="#">Delete</a>
14	100	0	57	<a href="#">c51e9a148d3a</a>	2	<a href="#">Delete</a>
15	100	0	2	<a href="#">fc8cd72e4f25</a>	1	<a href="#">Delete</a>
16	100	0	37	<a href="#">c4954cde6a30</a>	1	<a href="#">Delete</a>
17	100	0	0	<a href="#">5a9ed952ffa4</a>	0	<a href="#">Delete</a>

[Back to Dashboard](#)

Figure : All Cycles Information



### Cycle Maintenance

Cycles Requiring Maintenance


Cycle ID	Health	Action
13	84	<a href="#">Send to Maintenance</a>

Healthy Cycles

Cycle ID	Health
7	100
14	100
15	100
16	100
17	100
18	100
8	98
11	98
12	90

[Back to Dashboard](#)

Figure : Cycle Maintenance




Ride Information						
Ride ID	User ID	Cycle ID	Ride Date	Distance (in m)	Duration	Ride Cost
1	22BAI10084	7	2023-10-05 23:24:58	0	0	0.00
2	22BAI10084	7	2023-10-05 23:25:17	0	0	0.00
4	22BAI10084	8	2023-10-13 10:32:35	14	2	10.00
5	22BAI10084	8	2023-10-13 10:34:06	7	1	10.00
6	22BAI10084	9	2023-10-13 10:34:57	14	2	10.00
7	22BAI10084	9	2023-10-13 10:42:07	7	1	10.00
8	22BAI10084	8	2023-10-13 10:48:10	14	2	10.00
9	22BAI10084	9	2023-10-13 10:49:44	21	3	10.00
10	22BAI10084	9	2023-10-13 10:54:56	294	42	10.00
11	22BAI10084	8	2023-10-13 10:56:30	14	2	10.00
12	22BAI10084	10	2023-10-13 10:57:55	266	38	10.00
13	22BAI10084	9	2023-10-13 11:06:56	105	15	10.00
14	22BAI10084	10	2023-10-13 11:08:45	14	2	10.00
15	22BAI10084	7	2023-10-13 13:47:05	14	2	10.00
16	22BAI10084	14	2023-10-13 13:49:13	371	53	10.00

Figure : Ride Information

#### 4.2.6 Admin Dashboard:

**Design:** The admin dashboard offers a clear overview of system operations. It includes sections for user management, bike status, financial reports, and feedback monitoring, providing efficient central administration.

**Implementation:** Admin privileges encompass user account management, bike status updates, and financial report generation. The admin dashboard streamlines the system's administration, ensuring bikes are well-maintained and the user experience remains smooth.



Developer Dashboard					
Total Cycles 9	Total Users 8	Total Amount 857	Query Handling Total Pending: 0 Total Answered: 16	Total Rides 78	Cycles Requiring Maintenance 0

Figure : Developer Dashboard

#### 4.2.7 Website:

**Design:** The website is designed with a responsive and user-friendly layout, ensuring ease of use. It contains informative sections on bike rental guidelines, safety tips, and contact information. The website's color scheme is consistent with the logo, maintaining a unified brand identity.

**Implementation:** The website is built using responsive web design to accommodate various devices. It features dynamic content updates and secure payment gateways, facilitating seamless booking and information access for users.

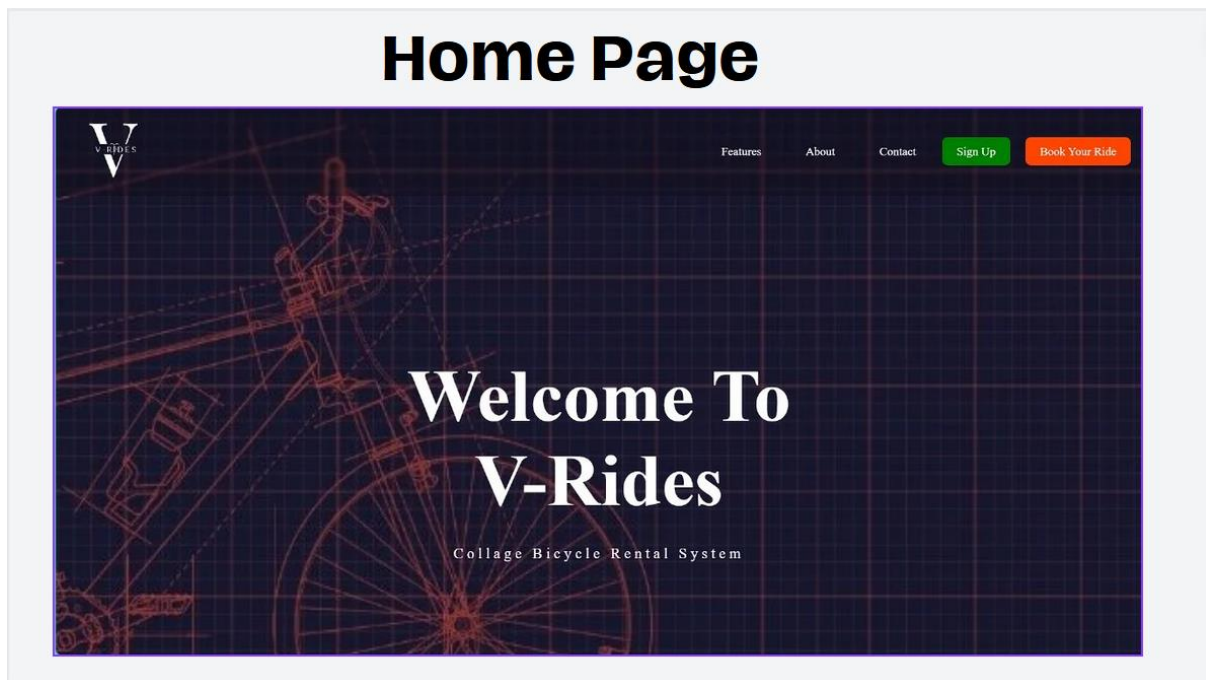


Figure : Home Page of the website

#### 4.2.8 Feedback Page:

**Design:** The feedback page is accessible from the user dashboard and prominently featured on the website's main menu. Users can provide feedback on their rides, report issues, and share suggestions.

**Implementation:** Feedback collection is integrated into user accounts and analyzed for continuous system improvement. User comments and suggestions are highly valued and contribute to enhancing the overall V-Rides experience.

#### 4.2.9 Database Management:

**Design:** A robust database management system is implemented to handle user data, bike information, and transaction history. It ensures efficient data retrieval and storage.

**Implementation:** The database securely stores user accounts, transaction history, bike details, and maintenance schedules. It employs encryption measures and regular backups to safeguard data integrity.

**This table contains all the information of user**

Field	Type	Null	Key	Default	Extra
id	int(6) unsigned	NO	PRI	NULL	auto_increment
vit_registration_number	varchar(255)	YES	UNI	NULL	
name	varchar(255)	NO		NULL	
batch	varchar(50)	NO		NULL	
email_id	varchar(255)	NO	UNI	NULL	
passcode	varchar(255)	NO		NULL	
birthday	date	NO		NULL	

**This table contains all the information of cycle**

Field	Type	Null	Key	Default	Extra
cycle_id	int(11)	NO	PRI	NULL	auto_increment
health	int(11)	NO		100	
distance_travelled	int(11)	NO		0	
total_time	int(11)	NO		0	
cycle_link	varchar(255)	YES	MUL	NULL	
status	varchar(255)	NO		available	
total_rides	int(11)	YES		0	

## This Table stores all Ride information

Field	Type	Null	Key	Default	Extra
ride_id	int(11)	NO	PRI	NULL	auto_increment
cycle_id	varchar(255)	NO		NULL	
ride_date	timestamp	NO		CURRENT_TIMESTAMP	
distance	float	NO		0	
duration	int(11)	NO		0	
start_time	int(11)	YES		NULL	
status	varchar(20)	NO		ongoing	
vit_registration_number	varchar(255)	NO	MUL	NULL	
ride_cost	decimal(10,2)	NO		0.00	

## This table stores information regarding Money in Users wallet

Field	Type	Null	Key	Default	Extra
vit_registration_number	varchar(255)	NO	PRI	NULL	
wallet_balance	int(11)	NO		50	

Figure : Database tables

### 4.2.10 Security and Safety Compliance:

**Design:** Security and safety are paramount in V-Rides. The system is designed to incorporate security features like encrypted user data, secure payment gateways, and maintenance alerts for bike safety.

**Implementation:** The system adheres to best practices for data security, with robust encryption measures in place. Bikes are regularly maintained and equipped with safety features such as lights and reflectors for rider protection.

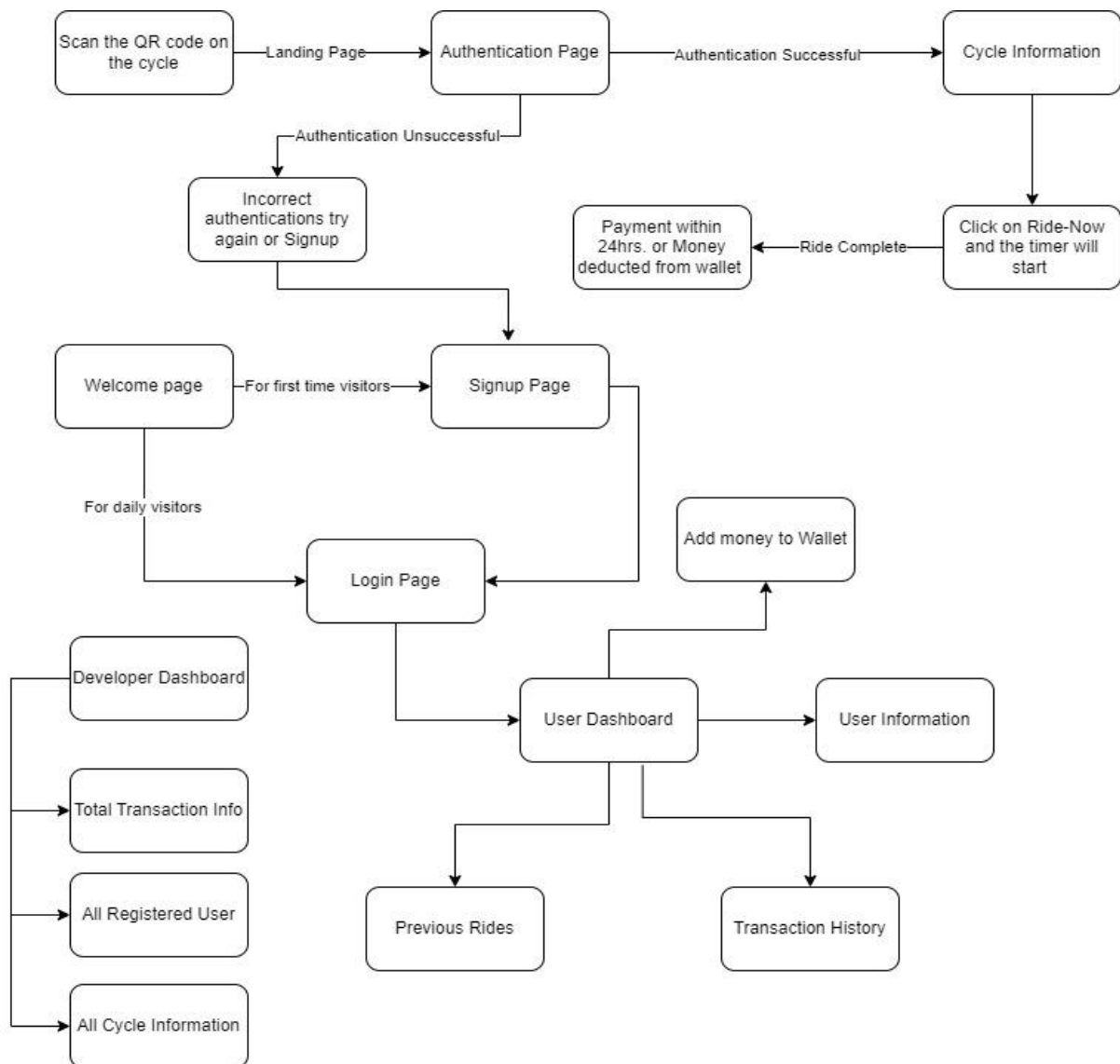


Figure : Flowchart

#### 4.3 Summary:

In summary, V-Rides demonstrates a harmonious fusion of innovative design and technically sound implementation. The system's logo, with its striking "V" and integrated bicycle, symbolizes both vitality and eco-consciousness, capturing the essence of college life on two wheels. User registration and authentication, fortified by robust encryption and multi-factor verification, underscore security and reliability.

The user account dashboard, designed for seamless navigation, aligns user-centricity with transaction transparency and history. QR code scanning, utilizing smartphone cameras,

enhances the access process, ensuring swift and secure bicycle retrieval. Bicycle management, featuring GPS tracking and maintenance schedules, epitomizes operational efficiency.

The admin dashboard, central to system administration, empowers administrators with tools for user management, bike status updates, and financial insight. The responsive website, in concert with secure payment gateways, offers intuitive access and payment convenience. User feedback, incorporated into user accounts, fuels iterative enhancement.

Complementing these components, a robust database management system ensures the integrity of user data, transaction history, and bike information. The system's commitment to security, underscored by encryption and regular backups, fortifies data protection.

Lastly, V-Rides pays meticulous attention to security and safety compliance. Data security is fortified through encryption, while regular bike maintenance and the incorporation of safety features on the bicycles underscore the commitment to rider well-being.

In summation, V-Rides offers an integrated and secure ecosystem for college students, championing sustainability, security, and user-centric design, providing a technologically advanced solution for campus transportation.

# PERFORMANCE ANALYSIS

## 5.1 Introduction

Performance analysis is a critical step in evaluating the effectiveness and efficiency of the "V-Rides" system. In this section, we will assess the system's performance using various measures and present the findings through performance measures and analysis in the form of tables, text, graphs, and charts.

## 5.2 Performance Measures

To comprehensively evaluate the performance of the "V-Rides" system, we have considered the following key performance measures:

Table 1: Performance Measures

Measure	Description
User Registration Time	The time taken for users to complete registration.
Login Time	Time required for users to log into the system.
Bicycle Reservation Time	Time needed to book a bicycle through the system.
Real-time Tracking Accuracy	Accuracy of the system in displaying bicycle locations.
System Response Time	The time taken by the system to respond to user requests.
Data Synchronization Time	Time for data to be updated across the system.

## 5.3 Summary

The performance analysis of the "V-Rides" system is crucial in assessing its efficiency and effectiveness. By examining user registration times, system response times, real-time tracking accuracy, and other performance measures, we gain valuable insights into the system's performance. The results, as presented in tables, text, graphs, and charts, help us understand the strengths and weaknesses of our system. This analysis informs future optimization and enhancements to ensure the delivery of a seamless and user-friendly bicycle rental experience within the campus



## **FUTURE ENHANCEMENT AND CONCLUSION**

### **6.1 Introduction**

The "V-Rides" project represents a significant step towards providing a sustainable and efficient solution for on-campus transportation. In this final section, we will discuss the limitations and constraints of the current system, outline potential future enhancements, and provide a concluding summary of the project.

### **6.3 Future Enhancements**

In future to continuously improve the "V-Rides" system, we propose the following enhancements:

- Scalability Solutions: Implementing strategies for system scalability to accommodate a growing user base and bicycle fleet. This may involve cloud-based solutions and load balancing.
- Enhanced Coverage: Exploring alternative technologies or hybrid solutions for tracking in areas with limited GPS coverage, such as Bluetooth or Wi-Fi-based location tracking.
- Budget Allocation: Seeking additional funding sources or partnerships to invest in advanced tracking technology and expand the bicycle fleet to meet the growing demand.
- App Launching: Instead of the web page, with more integration and new features an app could be developed for the same purpose.
- User Engagement: Developing marketing campaigns and user engagement strategies to encourage more students and staff to use the bicycle rental system, thereby fostering a culture of sustainable commuting.
- User-friendly Environment: The goal is to make the website more user- friendly and easy to handle.
- Third Party Integration with the payment app: Connecting a third party for the method of payment in the website itself.

### **6.4 Conclusion**

The "V-Rides" project embodies our commitment to addressing the transportation challenges within college campuses by providing an innovative and sustainable solution. This project report has detailed the design, implementation, and performance analysis of our system.

While the system has made significant strides in enhancing campus transportation, we acknowledge the challenges and constraints that need to be overcome. Through continuous improvement and the incorporation of future enhancements, we aspire to offer an even more efficient, scalable, and user-friendly bicycle rental platform.

In conclusion, the "V-Rides" project underscores the importance of sustainable transportation solutions in the modern world. It aligns with the broader goal of reducing carbon emissions, alleviating congestion, and promoting a greener and more accessible future. We look forward to the ongoing development and evolution of this project, working toward a more sustainable and efficient college campus experience.

## APPENDIX

Index.html(home page):

```
<!-- welcome page of our website -->
<!DOCTYPE html>
<html lang="en">

<head>
  <title>V-Rides</title>
  <link rel="stylesheet" href="css/index.css">
</head>

<body background="assets/bg_index.png">
  <nav class="navbar background">
    <ul class="nav-list">
      <div class="logo">
        
      </div>
    </ul>

    <div class="rightNav">
      <ul class="nav-list">
        <li><a href="#program">Features</a></li>
        <li><a href="#program2">About</a></li>
        <li><a href="#program3">Contact</a></li>
      </ul>
      <a href="signup.php" class="signup-btn">Sign Up</a>
      <a href="#" class="btn btn-primary">Book Your Ride</a>
    </div>
  </nav>

  <section class="firstsection">
    <div class="box-main">
      <div class="firstHalf">
        <h1 class="text-big" id="web">Welcome To <br> V-Rides </h1>
        <p class="text-littlesmall">
          Collage Bicycle Rental System
        </p>
      </div>
    </div>
  </section>

  <section class="secondsection" id="program">
```

```

<div class="box-main">
  <div class="firstHalf">
    <h1 class="text-midium" id="program">
      Our Features
    </h1>
    <div class="box-main">
      <div class="section-content">
        <div class="image-container">
          
        </div>
      </div>
      <button class="rent-button">
        <a href="login.php" class="rent-button">Rent a
bicycle</a>
      </button>
    </div>
  </div>
</div>
</section>

<section class="thirdsection" id="program2">
  <div class="box-main">
    <div class="section-content">
      <div class="image-container">
        
      </div>
      <div class="text-content">
        <h1 class="text-midium" id="web"> About V-Rides </h1>
        <p class="text-small">
          v-ride is a college cycle rental system designed to
provide flexible rental duration,
          payment option, bicycle health monitoring, user
profile, pre-booking, and multiple bicycle
          booking. Our system uses QR code-based locking and
unlocking to ensure security and ease of
          use for our customers. Our mission is to provide
convenient and affordable bicycles for
          college students and staff. Rent a bicycle from V-
Rides and enjoy a smooth ride to your
          destination.
        </p>
      </div>
    </div>
  </div>
</div>
</section>

<section class="section">
  <div class="paras">

```

```

        <h1 class="sectionTag text-littlesmall">
            Timings
        </h1>
        <p class="sectionSubTag text-verysmall">
            Monday - Sunday
        </p>
        <p class="sectionSubTag text-verysmall">
            7AM TO 8PM
        </p>
    </div>
</section>

<footer>
    <div class="contact-links">
        <h2>Contact Us</h2>
        <a href="https://www.instagram.com/your_instagram_link"
target="_blank">
            
        </a>
        <a href="https://www.youtube.com/your_youtube_link"
target="_blank">
            
        </a>
        <a href="https://www.facebook.com/your_facebook_link"
target="_blank">
            
        </a>
    </div>
    <div class="feedback-form" id="program3">
        <h2>Feedback</h2>
        <form action="submit_feedback.php" method="post">
            <input type="text" id="first-name" name="first_name"
placeholder="First Name" required>
            <input type="email" id="email" name="email"
placeholder="Email" required>
            <input type="text" id="type-your-problem"
name="type_your_problem" placeholder="Type Your Problem"
required>
            <button type="submit">Submit</button>
        </form>
        <!-- <p>Thanks for submitting!</p> -->
    </div>
</footer>
</body>

</html>

```

Styling index.css:

```
body {
  background-size: cover;
  background-repeat: no-repeat;
  background-attachment: fixed;
}

* {
  margin: 0;
  padding: 0;
}

.background {
  background: rgba(0, 0, 0, 0.1);
  backdrop-filter: blur(2px);
}

.navbar {
  position: fixed;
  display: flex;
  align-items: center;
  justify-content: center;
  position: sticky;
  top: 0;
  cursor: pointer;
  height: 100px;
  left: 0;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

.background {
  background-blend-mode: darken;
  background-size: cover;
  background-color: rgba(0, 0, 0, 0.2);
}

.nav-list {
  width: 100%;
  display: flex;
  align-items: center;
}

.logo {
  display: flex;
  justify-content: left;
  align-items: left;
```

```
}

.logo img {
  width: 150px;
  border-radius: 20px;
}

.nav-space {
  margin-right: 30px;
}

.nav-list li {
  list-style: none;
  padding: 15px 25px;
  margin-right: 5px;
}

.nav-list li a {
  text-decoration: none;
  color: white;
}

.nav-list li a:hover {
  color: grey;
}

.rightNav {
  display: flex;
  align-items: center;
  margin-right: 10px;
}

.rightNav .nav-list {
  list-style: none;
  display: flex;
  margin-right: 15px;
}

.rightNav .nav-list li {
  margin-right: 10px;
}

.rightNav .nav-list li a {
  text-decoration: none;
  color: white;
}

.rightNav .nav-list li a:hover {
```

```
    color: grey;
}

.rightNav .btn.btn-primary {
    padding: 10px 20px;
    background-color: orangered;
    color: white;
    border-radius: 8px;
    text-decoration: none;
    font-size: 16px;
    margin-left: 20px;
    transition: background-color 0.3s ease-in-out;
    white-space: nowrap;
}

.rightNav .btn.btn-primary:hover {
    background-color: rgba(255, 68, 0, 0.386);
}

.signup-btn {
    padding: 10px 20px;
    background-color: green;
    color: white;
    border-radius: 8px;
    text-decoration: none;
    font-size: 16px;
    margin-left: 0px;
    transition: background-color 0.3s ease-in-out;
    white-space: nowrap;
}

.rent-button {
    display: inline-block;
    padding: 10px 20px;
    background-color: orangered;
    color: white;
    border: none;
    border-radius: 8px;
    text-decoration: none;
    font-size: 16px;
    transition: background-color 0.3s ease-in-out;
    cursor: pointer;
}

.rent-button:hover {
    background-color: rgba(255, 68, 0, 0.8);
}
```



```
.signup-btn:hover {
  background-color: #32CD32;
}

.firstsection {
  height: 100vh;
  background-size: cover;
  background-repeat: no-repeat;
  background-position: center center;
  color: white;
  text-align: center;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}

.firstsection .box-main {
  display: flex;
  justify-content: center;
  align-items: center;
  max-width: 100%;
  margin: auto;
  height: 100%;
  color: white;
  margin-top: -50px;
}

.secondsection {
  background-color: rgb(255, 255, 255);
  height: 600px;
  display: flex;
  justify-content: center;
  align-items: center;
}

.secondsection .box-main {
  max-width: 100%;
  text-align: center;
  margin-top: 50px;
}

.section-content {
  text-align: center;
}

.centered-container {
```

```
display: flex;
flex-direction: column;
align-items: center;
justify-content: center;
margin: auto;
}

.feature-image {
  max-width: 100%;
  height: auto;
}

.secondsection .btn {
  padding: 10px 20px;
  background-color: orangered;
  color: white;
  border-radius: 8px;
  text-decoration: none;
  font-size: 16px;
  transition: background-color 0.3s ease-in-out;
}

.secondsection .btn:hover {
  background-color: #ff7f50;
}

.thirdsection {
  background-color: white;
  height: 800px;
  padding: 20px;
  display: flex;
  align-items: center;
  justify-content: center;
}

.thirdsection .box-main {
  max-width: 100%;
  margin: auto;
  color: #8e1515;
  margin-top: 80px;
}

.section-content {
  display: flex;
  align-items: center;
  justify-content: center;
}
```

```
.image-container {
  flex: 1;
  text-align: Left;
  padding: 20px;
}

.image-container img {
  max-width: 100%;
  height: auto;
}

.text-content {
  flex: 1;
  padding: 30px;
}

.firsthalf {
  width: 100%;
  display: flex;
  flex-direction: column;
  justify-content: center;
}

.secondhalf {
  width: 30%;
}

.secondhalf img {
  width: 50%;
  border: 4px solid black;
  border-radius: 100px;
  display: block;
  margin: auto;
}

.text-big {
  font-family: 'Piazzolla', serif;
  font-weight: bold;
  font-size: 100px;
}

.text-midium {
  font-size: 70px;
}

.text-small {
  font-size: 25px;
}
```

```
.text-verysmall {
  font-size: 18px;
  letter-spacing: 5px;
}

.text-littlesmall {
  font-size: 20px;
  letter-spacing: 5px;
}

.btn {
  padding: 8px 20px;
  margin: 20px 0;
  border-radius: 8px;
  background: none;
  color: white;
  cursor: pointer;
}

.btn-sm {
  padding: 30px 45px;
  vertical-align: middle;
}

.section {
  height: 200px;
  display: flex;
  align-items: center;
  justify-content: center;
  max-width: 100%;
  margin: auto;
  /* background-color: #f4f4f4; */
}

.section-Left {
  flex-direction: row-reverse;
}

.paras {
  text-align: center;
}

.sectionTag {
  font-size: 24px;
  color: white; /* Adjust the color as needed */
  margin-bottom: 10px;
}
```

```
.sectionSubTag {
    font-size: 18px;
    color: white; /* Adjust the color as needed */
    margin-bottom: 5px;
}

.thumbnail img {
    width: 250px;
    border: 2px solid black;
    border-radius: 26px;
    margin-top: 19px;
}

.center {
    text-align: center;
}

.text-footer {
    text-align: center;
    padding: 30px 0;
    font-family: 'Ubuntu', sans-serif;
    display: flex;
    justify-content: center;
    color: white;
}

footer {
    background-color: rgba(0, 0, 0, 0.5);
    padding: 20px;
    color: rgb(0, 0, 0);
    display: flex;
    align-items: center;
    justify-content: space-between;
}

.contact-links {
    text-align: left;
}

.contact-links h2 {
    margin: 0;
    font-size: 20px;
    color: white;
    margin-bottom: 20px;
}

.contact-links a {
```

```
    color: white;
    text-decoration: none;
    display: block;
    margin-bottom: 10px;
}

.contact-links img {
    width: 30px;
    height: 30px;
    margin-right: 5px;
    vertical-align: middle;
}

.feedback-form {
    text-align: center;
    color: white;
    margin-bottom: 20px;
    margin-right: 20px;
}

.feedback-form h2 {
    margin-bottom: 20px;
    font-size: 24px;
}

.feedback-form form {
    max-width: 400px;
    margin: 0 auto;
    text-align: left;
}

.feedback-form label {
    display: block;
    margin-bottom: 8px;
    color: white;
    font-size: 16px;
}

.feedback-form input {
    width: 100%;
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
}

.feedback-form button {
```

```

background-color: #007bff;
color: white;
border: none;
padding: 10px 20px;
cursor: pointer;
font-size: 16px;
border-radius: 5px;
transition: background-color 0.3s ease-in-out;
display: block;
margin: 0 auto;
}

.feedback-form button:hover {
    background-color: #0056b3;
}

.feedback-form p {
    margin-top: 10px;
    font-size: 18px;
}

```

## Login.php

```

<!-- this page is used for login -->
<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $servername = "localhost";
    $username = "root";
    $password = "123456";
    $dbname = "v_rides";

    $conn = new mysqli($servername, $username, $password, $dbname);

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $vit_registration_number = $_POST["vit_registration_number"];
    $passcode = $_POST["password"]; // Update to match the input field name in
your HTML form

    $sql = "SELECT * FROM user_info WHERE vit_registration_number =
'$vit_registration_number'";

```

```

$result = $conn->query($sql);

if ($result->num_rows == 1) {
    $row = $result->fetch_assoc();
    if (password_verify($passcode, $row["passcode"])) { // Update to match
the column name in your database
        $_SESSION["vit_registration_number"] = $vit_registration_number;
        $_SESSION["name"] = $row["name"];
        header("Location: user_dashboard.php");
        exit;
    } else {
        $login_error = "Invalid passcode. Please try again.";
    }
} else {
    $login_error = "User not found. Please check your VIT registration
number.";
}

$conn->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/signup.css">
    <title>Login Page</title>
</head>
<body>
    <div class="main">

        <div class="overlay">
            <div class="param">
                <h1>
                    _____
                    <br>Discover, Connect, Ride.<br> _____
                </h1>
            </div>
        </div>

    </div>

    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>

```



```

        <div class="buttons">
            <a href="signup.php" class="button">Signup</a>
        </div>
    </div>
</nav>

<div class="login-container">
    <h2>Login</h2>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
        <div class="form-group">
            <label class="input_head"
for="vit_registration_number">VIT Registration Number:</label>
            <input type="text" id="vit_registration_number"
name="vit_registration_number" placeholder="Enter your VIT registration
number" required>
        </div>
        <div class="form-group">
            <label class="input_head" for="password">Password:</label>
            <input type="password" id="password" name="password"
placeholder="Enter your password" required>
        </div>

        <?php
        if (isset($login_error)) {
            echo '<p style="color: red;">' . $login_error . '</p>';
        }
        ?>

        <div class="lg_button_cont">
            <button class="lg_button">Login</button>
        </div>
    </form>

</div>
</div>
</body>
</html>

```

```

Login.css :
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-image: url("../assets/bg.jpg"); /* Background image */
    background-size: cover; /* Cover the entire background */
    background-repeat: no-repeat; /* Do not repeat the background image */
}

```

```
.main{
display: flex;

}

.overlay{
backdrop-filter: blur(6px) saturate(200%);
-webkit-backdrop-filter: blur(6px) saturate(200%);
background-color: rgba(17, 25, 40, 0.59); /* rgba(17, 25, 40, 0.75); */
height: 100vh;
width: 50vw;
position: relative;
}

.navbar {
  position: absolute;
  width: 100%;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 98%;
}

.menu ul {
  list-style: none;
  padding: 0;
  margin: 0;
  display: flex;
}

.menu li {
  margin-left: 20px; /* Adjust spacing between menu items */
}

.menu li:first-child {
  margin-left: 0;
}

.menu a {
  color: white; /* Text color */
  text-decoration: none;
}

.logo img {
  height: 100px; /* Adjust the height of the Logo */
}
```

```
}

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
}

.logo {
  color: white;
}

.buttons a {
  background-color: rgb(112 170 243); /* Orange button color */
  color: white; /* White text color */
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  margin-left: 10px;
  text-decoration: none;
}

.login-container {
  margin: 10px auto; /* Center the login container */
  padding: 20px 40px;
  border: 1px solid #ccc;
  border-radius: 10px;
  width: fit-content;
  height: fit-content;
  color: white;
  margin-top: 100px;
  text-align: center; /* Center-align text within the container */
  backdrop-filter: blur(16px) saturate(180%);
  /* -webkit-backdrop-filter: blur(16px) saturate(180%);
  background-color: rgba(17, 25, 40, 0.75); */
  -webkit-backdrop-filter: blur(6px) saturate(200%);
  background-color: rgba(17, 25, 40, 0.59); /* rgba(17, 25, 40, 0.75); */
  border-radius: 12px;
  border: 1px solid rgba(255, 255, 255, 0.125);
}

.login-container h2 {
  margin-bottom: 15px;
}

.form-group {
  display: flex;
```

```

    flex-direction: column;
    margin-bottom: 15px;
}

.form-group label {
    display: block;
}

.form-group input {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
}

.input_head {
    margin-bottom: 10px;
}

.lg_button{
    background-color: rgb(112 170 243); /* Orange button color */
    color: white; /* White text color */
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    margin-left: 10px;
    text-decoration: none;
}

.param {
display: flex;
justify-content: center; /* Horizontally center */
align-items: center; /* Vertically center */
text-align: center; /* This affects the text inside the element */
font-size: 3em;
width: 100%; /* Define a width */
height: 100vh; /* Define a height (100vh means 100% of the viewport height) */
}

/* Styles for smaller screens (820px and below) */
@media screen and (max-width: 820px) {
    .overlay {
        height: auto;
        width: 100%;
        position: static;
    }
}

.main{

```

```
flex-direction: column;
}
}
```

Logout.php:

```
<!-- this page is used for logout -->
<?php
session_start();
session_destroy();
header("Location: login.php");
exit;
?>
```

Signup.php:

```
<!-- This is code for sign up page -->
<?php
// Establish a database connection (replace with your database credentials)
$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check the connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Initialize variables to store form data
$vit_registration_number = $name = $batch = $email = $password =
$confirm_password = $birthday = "";
$password_match_error = "";
$registration_result = "";

// Handle user registration when the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $vit_registration_number = $_POST["vit_registration_number"];
    $name = $_POST["name"];
    $batch = $_POST["batch"];
    $email = $_POST["email"];
```

```

$password = $_POST["password"];
$confirm_password = $_POST["confirm_password"];
$birthday = $_POST["birthday"];

// Check if passwords match
if ($password === $confirm_password) {
    // Hash the password for security (you should use a stronger hashing
method)
    $hashed_password = password_hash($password, PASSWORD_DEFAULT);

    // Check if the user already exists
    $check_user_sql = "SELECT * FROM user_info WHERE
vit_registration_number = ?";
    $stmt_check_user = $conn->prepare($check_user_sql);
    $stmt_check_user->bind_param("s", $vit_registration_number);
    $stmt_check_user->execute();
    $result_check_user = $stmt_check_user->get_result();

    if ($result_check_user->num_rows > 0) {
        // User already exists, display an appropriate message
        $registration_result = "User with VIT registration number
$vit_registration_number already exists.";
    } else {
        // User does not exist, proceed with the registration

        // SQL query to insert user data into the database using prepared
statement
        $sql = "INSERT INTO user_info (vit_registration_number, name,
batch, email_id, passcode, birthday)
VALUES (?, ?, ?, ?, ?, ?)";

        // Use prepared statement to prevent SQL injection
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("ssssss", $vit_registration_number, $name,
$batch, $email, $hashed_password, $birthday);

        if ($stmt->execute()) {
            // Insert initial wallet balance (50 tokens) for the user
            $initial_balance = 50;
            $wallet_sql = "INSERT INTO wallet (vit_registration_number,
wallet_balance)
VALUES (?, ?)";

            // Use prepared statement to prevent SQL injection
            $wallet_stmt = $conn->prepare($wallet_sql);
            $wallet_stmt->bind_param("si", $vit_registration_number,
$initial_balance);

```

```

        if ($wallet_stmt->execute()) {
            $registration_result = "Registration successful!";
        } else {
            $registration_result = "Error: " . $wallet_stmt->error;
        }

        $wallet_stmt->close();
    } else {
        $registration_result = "Error: " . $stmt->error;
    }

    // Close the prepared statement
    $stmt->close();
}
} else {
    $password_match_error = "Passwords do not match. Please try again.";
}
}

// Close the database connection
$conn->close();
?>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/signup.css">
    <title>Signup Page</title>
</head>
<body>
    <div class="main">

        <div class="overlay">
            <div class="param">
                <h1>
                    _____
                    <br>Discover, Connect, Ride.<br> _____
                </h1>
            </div>
        </div>
    </div>

    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>

```

```

        <div class="buttons">
            <a href="login.php" class="button">Login</a>
        </div>
    </div>
</nav>

<div class="login-container">
    <h2>Sign Up</h2>
    <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">

        <div class="form-group">

            <!-- <label class="input_head" for="name">Name:</label> -->
            <input type="text" name="name" placeholder="Enter your Name"
required>

        </div>

        <div class="form-group">

            <!-- <label class="input_head" for="password">Email:</label> -->
            <input type="email" name="email" placeholder="Enter your Email"
required>

        </div>

        <div class="form-group">

            <!-- <label class="input_head" for="username">Vit Registration
Number :</label> -->
            <input type="text" name="vit_registration_number"
placeholder="Enter your Registration Number" required>

        </div>

        <div class="form-group">

            <!-- <label class="input_head" for="batch">Batch:</label> -->
            <input type="number" name="batch" placeholder="Enter your Batch"
required>

        </div>

        <div class="form-group">

```



```

        <!-- <Label class="input_head" for="password">Password:</Label> --
    >
        <input type="password" name="password" placeholder="Enter your
password" required>

    </div>

    <div class="form-group">

        <!-- <Label class="input_head" for="password">Confirm
Password:</Label> -->
        <input type="password" name="confirm_password" placeholder="Enter
your password again" required>

    </div>

    <span style="color: red;"><?php echo $password_match_error; ?></span>

    <div class="form-group">

        <!-- <Label for="birthday" class="input_head">Birthday:</Label> --
    >
        <input type="date" name="birthday" required>

    </div>
    <div class="lg_button_cont">

        <input type="submit" value="Sign Up" class="lg_button">

    </div>
</form>
<p><?php echo $registration_result; ?></p>
</div>
</div>
</body>
</html>

```

Signup.css

```

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-image: url("../assets/bg.jpg"); /* Background image */
    background-size: cover; /* Cover the entire background */
    background-repeat: no-repeat; /* Do not repeat the background image */
}

```

```
}

.main{
display: flex;

}

.overlay{
backdrop-filter: blur(6px) saturate(200%);
-webkit-backdrop-filter: blur(6px) saturate(200%);
background-color: rgba(17, 25, 40, 0.59); /* rgba(17, 25, 40, 0.75); */
height: 100vh;
width: 50vw;
position: relative;
}

.navbar {
  position: absolute;
  width: 100%;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 98%;
}

.menu ul {
  list-style: none;
  padding: 0;
  margin: 0;
  display: flex;
}

.menu li {
  margin-left: 20px; /* Adjust spacing between menu items */
}

.menu li:first-child {
  margin-left: 0;
}

.menu a {
  color: white; /* Text color */
  text-decoration: none;
}

.logo img {
```

```

    height: 100px; /* Adjust the height of the Logo */
}

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.logo {
    color: white;
}

.buttons a {
    background-color: rgb(112 170 243); /* Orange button color */
    color: white; /* White text color */
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    margin-left: 10px;
    text-decoration: none;
}

.login-container {
    margin: 10px auto; /* Center the login container */
    padding: 20px 40px;
    border: 1px solid #ccc;
    border-radius: 10px;
    width: 400px;
    height: 480px;
    color: white;
    margin-top: 100px;
    text-align: center; /* Center-align text within the container */
    backdrop-filter: blur(16px) saturate(180%);
    /* -webkit-backdrop-filter: blur(16px) saturate(180%);
    background-color: rgba(17, 25, 40, 0.75); */
    -webkit-backdrop-filter: blur(6px) saturate(200%);
    background-color: rgba(17, 25, 40, 0.59); /* rgba(17, 25, 40, 0.75); */
    border-radius: 12px;
    border: 1px solid rgba(255, 255, 255, 0.125);
}

.login-container h2 {
    margin-bottom: 15px;
}

.form-group {

```

```

    display: flex;
    flex-direction: column;
    margin-bottom: 15px;
}

.form-group label {
    display: block;
}

.form-group input {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-sizing: border-box;
}

.input_head {
    margin-bottom: 10px;
}

.lg_button{
    background-color: rgb(112 170 243); /* Orange button color */
    color: white; /* White text color */
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    margin-left: 10px;
    text-decoration: none;
}

.param {
    display: flex;
    justify-content: center; /* Horizontally center */
    align-items: center; /* Vertically center */
    text-align: center; /* This affects the text inside the element */
    font-size: 3em;
    width: 100%; /* Define a width */
    height: 100vh; /* Define a height (100vh means 100% of the viewport height)
*/
}

/* Styles for smaller screens (820px and below) */
@media screen and (max-width: 820px) {
    .overlay {
        height: auto;
        width: 100%;
        position: static;
    }
}

```

```

    .main{
        flex-direction: column;
    }
}

```

Cycle\_info.php

```

<!-- <?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Handle the form submission for adding a new cycle or deleting a cycle
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST["add_new_cycle"])) {
        // Generate a random link for the new cycle
        $cycle_link = bin2hex(random_bytes(6)); // Adjust length as needed

        // Insert a new cycle with default values and the generated link
        $insertSql = "INSERT INTO cycle_info (health, distance_travelled,
total_time, cycle_link)
                    VALUES (100, 0, 0, '$cycle_link')";

        if ($conn->query($insertSql) === TRUE) {
            // Redirect to refresh the page after adding a new cycle
            header("Location: " . $_SERVER['PHP_SELF']);
            exit;
        } else {
            $error_message = "Error adding a new cycle: " . $conn->error;
        }
    } elseif (isset($_POST["delete_cycle"])) {
        // Handle the form submission for deleting a cycle
        $cycle_id_to_delete = $_POST["cycle_id_to_delete"];

        // Delete the selected cycle
    }
}

```

```

        $deleteSql = "DELETE FROM cycle_info WHERE cycle_id =
$cycle_id_to_delete";
        if ($conn->query($deleteSql) === TRUE) {
            // Redirect to refresh the page after deleting the cycle
            header("Location: " . $_SERVER['PHP_SELF']);
            exit;
        } else {
            $error_message = "Error deleting the cycle: " . $conn->error;
        }
    }
}

// Query to fetch all cycle information including total rides
$sql = "SELECT *, (SELECT COUNT(*) FROM rides WHERE cycle_id =
cycle_info.cycle_id) AS total_rides
FROM cycle_info";
$result = $conn->query($sql);

// Get the total number of cycles
$total_cycles = $result->num_rows;
?> -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>All Cycle Information</title>
    <link rel="stylesheet" href="css/cycle_info.css">
</head>
<body>
    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>
    <div class="container">
        <h2>All Cycle Information</h2>

        <p id="total-cycles">Total Number of Cycles: <?php echo $total_cycles;
?></p>

        <?php
        if (isset($error_message)) {
            echo '<p class="error-message">' . $error_message . '</p>';
        }
        ?>

```

```

        <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
            <input type="submit" name="add_new_cycle" value="Add New Cycle"
class="button">
            <a href="cycle_maintenance.php" class="button">Maintenance</a>
        </form>

        <table border="1" class="cycle-table">
            <tr>
                <th>Cycle ID</th>
                <th>Health</th>
                <th>Distance Traveled</th>
                <th>Total Time</th>
                <th>Cycle Link</th>
                <th>Total Rides</th>
                <th>Action</th>
            </tr>
            <?php
            if ($result->num_rows > 0) {
                while ($row = $result->fetch_assoc()) {
                    echo "<tr>";
                    echo "<td>" . $row["cycle_id"] . "</td>";
                    echo "<td>" . $row["health"] . "</td>";
                    echo "<td>" . $row["distance_travelled"] . "</td>";
                    echo "<td>" . $row["total_time"] . "</td>";
                    echo "<td><a href='user_authentication.php?cycle_id=" .
$row["cycle_id"] . "' target='_blank'>" . $row["cycle_link"] . "</a></td>";
                    echo "<td>" . $row["total_rides"] . "</td>";
                    echo "<td>";
                    echo "<form method='post' action='" . $_SERVER['PHP_SELF']
. "'>";
                    echo "<input type='hidden' name='cycle_id_to_delete'
value='" . $row['cycle_id'] . "'>";
                    echo "<input type='submit' name='delete_cycle'
value='Delete' class='delete-button'>";
                    echo "</form>";
                    echo "</td>";
                    echo "</tr>";
                }
            } else {
                echo "<tr><td colspan='7'>No cycle information
available</td></tr>";
            }
            ?>
        </table>

        <br>
        <a href="dev_dashboard.php" class="back-link">Back to Dashboard</a>
    </div>

```

```
</body>
</html>
```

Cycle\_info.css

```
@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{
  font-family: 'Tilt Neon', sans-serif;
}

body {
  margin: 0;
  padding: 0;
  background: #1b1b33;
  background-size: cover;
  background-repeat: no-repeat;
}

/* Navigation Bar */
.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding-right: 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  margin: 0 auto;
}

.navbar .logo img{
  width: 80px;
}

.container {
  margin: 20px auto;
  padding: 20px;
```



```
border: 1px solid #ccc;
border-radius: 10px;
background-color: #fff;
width: 80%;
}

h2 {
  color: #333;
  text-align: center;
}

.maintenance-list, .healthy-list {
  margin-top: 20px;
  padding: 10px;
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.1);
  background-color: #fff;
  border-radius: 4px;
}

.maintenance-list h3, .healthy-list h3 {
  color: #555;
}

table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 10px;
}

th, td {
  padding: 10px;
  border: 1px solid #ddd;
  text-align: left;
}

th {
  background-color: #f2f2f2;
}

.send-to-maintenance {
  background-color: #4caf50;
  color: #fff;
  border: none;
  padding: 8px 16px;
  text-decoration: none;
  cursor: pointer;
  border-radius: 4px;
  font-size: 14px;
}
```

```

}

.send-to-maintenance:hover {
    background-color: #45a049;
}

.back-link {
    display: block;
    margin-top: 20px;
    color: #ccc;
    text-decoration: none;
    font-weight: bold;
    text-align: center;
}

.back-link h3:hover {
    color: #45a049;
}

```

Cycle\_maintenance.php

```

<!-- This page control Cycle maintenance -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch cycles that require maintenance (health less than 70)
$fetch_cycles_sql = "SELECT * FROM cycle_info WHERE health < 90 ORDER BY
health ASC";
$result = $conn->query($fetch_cycles_sql);

// Fetch cycles that don't require maintenance (health 70 or above)
$healthy_cycles_sql = "SELECT * FROM cycle_info WHERE health >= 90 ORDER BY
health DESC";
$healthy_result = $conn->query($healthy_cycles_sql);

```

```

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cycle Maintenance</title>
    <link rel="stylesheet" href="css/cycle_info.css">
    <style>

    </style>
</head>
<body>

    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>

<div class="container">
    <h2>Cycle Maintenance</h2>

    <!-- Display cycles requiring maintenance -->
    <div class="maintenance-list">
        <h3>Cycles Requiring Maintenance</h3>
        <table class="cycle-table">
            <thead>
                <tr>
                    <th>Cycle ID</th>
                    <th>Health</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                <?php
                if ($result && $result->num_rows > 0) {
                    while ($row = $result->fetch_assoc()) {
                        echo "<tr>";
                        echo "<td>" . $row["cycle_id"] . "</td>";
                        echo "<td>" . $row["health"] . "</td>";
                        echo "<td><button class='send-to-maintenance'
onclick='sendToMaintenance(" . $row["cycle_id"] . ")'>Send to
Maintenance</button></td>";
                        echo "</tr>";
                    }
                }
            </tbody>
        </table>
    </div>

```

```

        } else {
            echo "<tr><td colspan='3'>No cycles require
maintenance.</td></tr>";
        }
    ?>
</tbody>
</table>
</div>

<!-- Display healthy cycles -->
<div class="healthy-list">
    <h3>Healthy Cycles</h3>
    <table class="cycle-table">
        <thead>
            <tr>
                <th>Cycle ID</th>
                <th>Health</th>
            </tr>
        </thead>
        <tbody>
            <?php
            if ($healthy_result && $healthy_result->num_rows > 0) {
                while ($row = $healthy_result->fetch_assoc()) {
                    echo "<tr>";
                    echo "<td>" . $row["cycle_id"] . "</td>";
                    echo "<td>" . $row["health"] . "</td>";
                    echo "</tr>";
                }
            } else {
                echo "<tr><td colspan='2'>No healthy cycles
available</td></tr>";
            }
            ?>
        </tbody>
    </table>
</div>
</div>

<a href="dev_dashboard.php" class="back-link"><h3>Back to Dashboard</h3></a>
<br>
<script>
    function sendToMaintenance(cycleId) {
        // Implement the logic to send the cycle with the given ID to
maintenance
        // You can use AJAX to send an asynchronous request to the server
        // and update the database accordingly.
        alert("Sending Cycle ID " + cycleId + " to maintenance.");
    }

```

```
</script>
```

```
</body>
```

```
</html>
```

Cycle\_ride.php

```
<!-- this page show information of particular cycle to start ride -->
```

```
<?php
```

```
session_start();
```

```
$servername = "localhost";
```

```
$username = "root";
```

```
$password = "123456";
```

```
$dbname = "v_rides";
```

```
$conn = new mysqli($servername, $username, $password, $dbname);
```

```
if ($conn->connect_error) {
```

```
    die("Connection failed: " . $conn->connect_error);
```

```
}
```

```
// Check if the user is Logged in; if not, redirect to the Login page
```

```
if (!isset($_SESSION["vit_registration_number"])) {
```

```
    header("Location: ride_login.php");
```

```
    exit;
```

```
}
```

```
// Check if the cycle_id is set in the session; if not, redirect to the  
cycle_info page
```

```
if (!isset($_SESSION["cycle_id"])) {
```

```
    header("Location: cycle_info.php");
```

```
    exit;
```

```
}
```

```
$vit_registration_number = $_SESSION["vit_registration_number"];
```

```
$cycle_id = $_SESSION["cycle_id"];
```

```
// Fetch user information
```

```
$user_info_sql = "
```

```
    SELECT u.name, w.wallet_balance
```

```
    FROM user_info u
```

```
    JOIN wallet w ON u.vit_registration_number = w.vit_registration_number
```

```
    WHERE u.vit_registration_number = '$vit_registration_number'
```

```

";

$user_info_result = $conn->query($user_info_sql);

if ($user_info_result->num_rows == 1) {
    $user_info = $user_info_result->fetch_assoc();
}

// Fetch cycle information
$cycle_info_sql = "SELECT * FROM cycle_info WHERE cycle_id = $cycle_id";
$cycle_info_result = $conn->query($cycle_info_sql);

if ($cycle_info_result->num_rows == 1) {
    $cycle_info = $cycle_info_result->fetch_assoc();
} else {
    // Handle the case where cycle information is not found
    echo "Cycle information not found.";
    exit;
}

// Check if the user is starting the ride and has enough money
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["start_ride"])) {
    $requiredBalance = 10; // Adjust this value based on your ride cost

    // Check if the user has enough money in the wallet
    if ($user_info["wallet_balance"] >= $requiredBalance) {
        // Set the start time and duration in the session
        $_SESSION["ride_start_time"] = time();
        $_SESSION["ride_duration"] = 15 * 60; // 15 minutes in seconds

        // Redirect to the ongoing ride page
        header("Location: ongoing_ride.php");
        exit;
    } else {
        $insufficientBalanceMessage = "Not enough money. <a
href='wallet.php'>Recharge your wallet now.</a>";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cycle Ride Details</title>
    <link rel="stylesheet" type="text/css" href="css/cycle_ride.css">

```

```

</head>
<body>
    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>

    <section class="main-content">
        <div class="flex-item flex-head">
            <h1>Cycle Ride Information</h1>
        </div>
        <div class="flex-item flex-col">

            <div class="review">
                
            </div>

            <div>
                <p class="text_">Name: <?php echo $user_info["name"]; ?></p>
            </div>

        </div>
        <div class="flex-item">
            <h3>Cycle Info</h3>
            <p>Cycle ID: <?php echo $cycle_info["cycle_id"]; ?></p>
            <p>Health: <?php echo $cycle_info["health"]; ?></p>
            <p>Distance Traveled: <?php echo
$cycle_info["distance_travelled"]; ?></p>
            <p>Total Time: <?php echo $cycle_info["total_time"]; ?></p>
        </div>

        <div class="start-ride-button">
            <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
                <input type="submit" name="start_ride" value="Start Ride">
            </form>
        </div>

        <!-- <?php
if (isset($insufficientBalanceMessage)) {
    echo '<p style="color: red;">' . $insufficientBalanceMessage .
'</p>';
}
?> -->

    </section>

```

```
</body>
</html>
```

Cycle\_ride.css

```
/* css/cycle_ride.css */
@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{
  font-family: 'Tilt Neon', sans-serif;
}

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-size: cover;
  background-repeat: no-repeat;
  background: #1b1b33;
}

.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding-right: 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  margin: 0 auto;
}

.navbar .logo img{
  width: 80px;
}

.main-content{
  flex-wrap: wrap;
```



```

width: 80vw;
margin: 0px auto;
align-items: center;
justify-content: center;
}

.flex-item {
  /* display: flex; */
  flex: 1 0 50%;
  padding: 5px;
  box-sizing: border-box;
  text-align: center;
  height: 25vh;
  backdrop-filter: blur(16px) saturate(200%);
  -webkit-backdrop-filter: blur(16px) saturate(200%);
  background-color: rgb(216, 213, 213);
  border-radius: 15px;
  border: 1px solid rgba(255, 255, 255, 0.125);
  margin: 10px;
  box-shadow: 0px 0px 40px 2px rgb(244 240 240 / 17%);
  transition: transform 0.5s ease-in-out;
  justify-content: space-around;
}

.flex-head{
  height: 10vh;
  margin-top: 20px;
}

.review{
  display: flex;
  align-items: center;
  justify-content: left;
  margin-left: 50px;
}

.review img{
  width: 190px;
}

.flex-col{
  display: flex;
  justify-content: space-evenly;
  /* padding-right: 500px; */
  align-items: center;
  margin-top: 0px;
  padding-top: 0px;
}

```

```

}

.text_{
    font-size: 30px;
    font-weight: 600;
}

.container {
    text-align: center;
    background-color: rgba(255, 255, 255, 0.9);
    padding: 20px;
    border-radius: 10px;
    width: 80%;
}

.info-box {
    border: 1px solid #ccc;
    padding: 20px;
    margin-bottom: 20px;
    border-radius: 10px;
}

/* Updated styles for displaying user info and cycle info side by side */
.info-box {
    display: inline-block;
    width: 45%; /* Adjust the width as needed */
    margin-right: 20px;
}

.start-ride-button {
    margin-top: 20px;
    display: flex;
    justify-content: center;
    align-items: center;
}

/* Enlarged button styles */
input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 15px 30px; /* Enlarged padding */
    font-size: 16px; /* Enlarged font size */
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

input[type="submit"]:hover {

```

```

        background-color: #45a049;
    }

    a {
        color: orange;
        text-decoration: none;
    }

    a:hover {
        text-decoration: underline;
    }

```

Dev\_dashboard.php

```

<!-- admin dashboard page -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch and display the total number of pending queries
$pendingQueriesSql = "SELECT COUNT(*) as total_pending_queries FROM feedback
WHERE solved = 0";
$pendingQueriesResult = $conn->query($pendingQueriesSql);
$totalPendingQueries = ($pendingQueriesResult->num_rows == 1) ?
$pendingQueriesResult->fetch_assoc()["total_pending_queries"] : 0;

// Fetch and display the total number of answered queries
$answeredQueriesSql = "SELECT COUNT(*) as total_answered_queries FROM feedback
WHERE solved = 1";
$answeredQueriesResult = $conn->query($answeredQueriesSql);
$totalAnsweredQueries = ($answeredQueriesResult->num_rows == 1) ?
$answeredQueriesResult->fetch_assoc()["total_answered_queries"] : 0;

// Fetch and display the total number of rides

```

```

$totalRidesSql = "SELECT COUNT(*) as total_rides FROM rides";
$totalRidesResult = $conn->query($totalRidesSql);
$totalRides = ($totalRidesResult->num_rows == 1) ? $totalRidesResult-
>fetch_assoc()["total_rides"] : 0;
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Developer Dashboard</title>
    <!-- Add your CSS Link here -->
    <link rel="stylesheet" href="css/dev_dashboard.css">
</head>
<body>
    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>

    <div class="container">
        <h2>Developer Dashboard</h2>

        <div class="dashboard">
            <div class="dashboard-box"
onclick="location.href='cycle_info.php';">
                <h3>Total Cycles</h3>
                <?php
                // Database connection code here (replace this with your
actual database connection code)
                $servername = "localhost";
                $username = "root";
                $password = "123456";
                $dbname = "v_rides";

                $conn = new mysqli($servername, $username, $password,
$dbname);

                if ($conn->connect_error) {
                    die("Connection failed: " . $conn->connect_error);
                }

                // Fetch and display the total number of cycles from your
database

```

```

        $totalCyclesSql = "SELECT COUNT(*) as total_cycles FROM
cycle_info";
        $totalCyclesResult = $conn->query($totalCyclesSql);
        $totalCycles = ($totalCyclesResult->num_rows == 1) ?
$totalCyclesResult->fetch_assoc()["total_cycles"] : 0;
        echo "<p>$totalCycles</p>";
        ?>
    </div>

    <div class="dashboard-box"
onclick="location.href='user_info.php';">
        <h3>Total Users</h3>
        <?php
// Fetch and display the total number of users from your
database
        $totalUsersSql = "SELECT COUNT(*) as total_users FROM
user_info";
        $totalUsersResult = $conn->query($totalUsersSql);
        $totalUsers = ($totalUsersResult->num_rows == 1) ?
$totalUsersResult->fetch_assoc()["total_users"] : 0;
        echo "<p>$totalUsers</p>";
        ?>
    </div>

    <div class="dashboard-box" onclick="location.href='#';">
        <h3>Total Amount</h3>
        <?php
// Fetch and display the sum of the total amount from the
wallet
        $totalAmountSql = "SELECT SUM(wallet_balance) as total_amount
FROM wallet";
        $totalAmountResult = $conn->query($totalAmountSql);
        $totalAmount = ($totalAmountResult->num_rows == 1) ?
$totalAmountResult->fetch_assoc()["total_amount"] : 0;
        echo "<p>$totalAmount</p>";
        ?>
    </div>

    <div class="dashboard-box"
onclick="location.href='handle_query.php';">
        <h3>Query Handling</h3>
        <p>Total Pending: <?php echo $totalPendingQueries; ?></p>
        <p>Total Answered: <?php echo $totalAnsweredQueries; ?></p>
    </div>
    <div class="dashboard-box"
onclick="location.href='ride_info.php';">
        <h3>Total Rides</h3>
        <p><?php echo $totalRides; ?></p>

```

```

        </div>

        <div class="dashboard-box"
onclick="location.href='cycle_maintenance.php';">
            <h3>Cycles Requiring Maintenance</h3>
            <?php
                // Fetch and display the total number of cycles requiring
maintenance (health less than 70)
                $cyclesRequiringMaintenanceSql = "SELECT COUNT(*) as
cycles_requiring_maintenance FROM cycle_info WHERE health < 70";
                $cyclesRequiringMaintenanceResult = $conn-
>query($cyclesRequiringMaintenanceSql);
                $cyclesRequiringMaintenance =
($cyclesRequiringMaintenanceResult->num_rows == 1) ?
$cyclesRequiringMaintenanceResult-
>fetch_assoc()["cycles_requiring_maintenance"] : 0;
                echo "<p>$cyclesRequiringMaintenance</p>";
            ?>
        </div>

    </div>
</div>
</body>
</html>

```

Dev\_dashboard.css

```

/* user_dashboard.css */

@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{
    font-family: 'Tilt Neon', sans-serif;
}

body {
    margin: 0;
    padding: 0;
    background: #1b1b33;
}

```

```
    background-size: cover;
    background-repeat: no-repeat;
}

.navbar {
    display: flex;
    background-color: #29294d;
    justify-content: space-between;
    align-items: center;
    padding-right: 20px;
}

.navbar-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 100%;
    margin: 0 auto;
}

.navbar .logo img{
    width: 80px;
}

main{
    height: 70vh;
    display: flex;
    justify-content: center;
    align-items: center;
}

.container {
    width: 80vw;
    margin: 50px auto;
    background-color: #fff;
    border-radius: 10px;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
    color: #333;
}

.dashboard {
    display: flex;
    justify-content: space-around;
```

```

    margin-top: 30px;
}

.dashboard-box {
    flex: 1;
    background-color: #fff;
    border: 1px solid #ddd;
    border-radius: 20px;
    padding: 20px;
    text-align: center;
    cursor: pointer;
    transition: transform 0.4s ease-in;
}

.dashboard-box:hover {
    background-color: #f0f0f0;
    transform: scale(1.1);
}

h3 {
    color: #333;
}

p {
    color: #666;
    font-size: 20px;
    margin-top: 10px;
    font-weight: bold;
}

```

Handle\_query.php

```

<!-- this page will handle queries send by users -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```



```

}

// Fetch all queries from the feedback table initially
$fetchAllQueriesSql = "SELECT * FROM feedback";
$allQueriesResult = $conn->query($fetchAllQueriesSql);

$unsolvedQueries = [];
$solvedQueries = [];

// Separate queries into unsolved and solved
while ($row = $allQueriesResult->fetch_assoc()) {
    if ($row["solved"] == 0) {
        $unsolvedQueries[] = $row;
    } else {
        $solvedQueries[] = $row;
    }
}

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["submit_answer"])) {
    $queryId = $_POST["query_id"];
    $answer = mysqli_real_escape_string($conn, $_POST["answer"]);

    // Update the feedback table to mark the query as solved
    $updateQuerySql = "UPDATE feedback SET solved = 1 WHERE query_id = $queryId";

    if ($conn->query($updateQuerySql) === TRUE) {
        // Insert the solution into the solutions table
        $insertSolutionSql = "INSERT INTO solutions (query_id, solution)
VALUES ($queryId, '$answer')";

        if ($conn->query($insertSolutionSql) === FALSE) {
            echo '<p style="color: red;">Error submitting answer: ' . $conn->error . '</p>';
        }
    } else {
        echo '<p style="color: red;">Error updating query status: ' . $conn->error . '</p>';
    }
}

// Fetch all queries again after the modification
$allQueriesResult = $conn->query($fetchAllQueriesSql);
$unsolvedQueries = [];
$solvedQueries = [];

// Separate queries into unsolved and solved
while ($row = $allQueriesResult->fetch_assoc()) {

```

```

        if ($row["solved"] == 0) {
            $unsolvedQueries[] = $row;
        } else {
            $solvedQueries[] = $row;
        }
    }
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Answer Query</title>
    <link rel="stylesheet" href="css/handle_query.css">
</head>
<body>
    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>

    <main>
        <div class="container">
            <div class="flex-container">
                <!-- Unsolved Queries on Left -->
                <div class="flex-item">
                    <h3>Unsolved Queries</h3>
                    <?php if (!empty($unsolvedQueries)) { ?>
                        <?php foreach ($unsolvedQueries as $row) { ?>
                            <div class="query-box">
                                <p><strong>Query ID:</strong> <?php echo
$row["query_id"]; ?></p>
                                <p><strong>Name:</strong> <?php echo
$row["name"]; ?></p>
                                <p><strong>Query:</strong> <?php echo
$row["query"]; ?></p>

                                <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
                                    <input type="hidden" name="query_id"
value="<?php echo $row["query_id"]; ?>">
                                    <label for="answer"><strong>Your
Answer:</strong></label>
                                    <textarea id="answer" name="answer"
rows="4" required></textarea>

```

```

        <input type="submit" name="submit_answer"
value="Submit Answer">
    </form>
</div>
    <?php } ?>
    <?php } else { ?>
        <p>No unsolved queries left.</p>
    <?php } ?>
</div>

<!-- Solved Queries on Right with Scroll -->
<div class="flex-item scroll-right">
    <h3>Solved Queries</h3>
    <?php if (!empty($solvedQueries)) { ?>
        <?php foreach ($solvedQueries as $row) { ?>
            <div class="query-box">
                <p><strong>Query ID:</strong> <?php echo
$row["query_id"]; ?></p>
                <p><strong>Name:</strong> <?php echo
$row["name"]; ?></p>
                <p><strong>Query:</strong> <?php echo
$row["query"]; ?></p>
                <p class="solved-message">This query has
already been solved.</p>
            </div>
        <?php } ?>
    <?php } ?>
</div>
</div>
</div>
</main>

    <a class="back-to-dashboard" href="dev_dashboard.php">Back to
Dashboard</a>
    <br>
</body>
</html>

```

Handle\_query.css

```

/* handle_query.css */

@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

```

```
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: 'Tilt Neon', sans-serif;
  background: #1b1b33;
  background-size: cover;
  background-repeat: no-repeat;
  margin: 0;
  color: #fff;
}

/* Navigation Bar */
.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding: 10px 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  margin: 0 auto;
}

.navbar .logo img {
  width: 80px;
}

/* Main Content */
.container {
  margin: 20px;
}

.flex-container {
  display: flex;
  justify-content: space-between;
}

.flex-item {
  flex: 1;
}
```

```
padding: 20px;
box-sizing: border-box;
margin: 10px;
border-radius: 10px;
/* background-color: rgba(255, 255, 255, 0.9); */
}

.scroll-right {
  overflow-y: auto;
  max-height: 80vh; /* Adjust as needed */
}

/* Query Box */
.query-box {
  margin-bottom: 20px;
}

h2, h3 {
  color: #fff;
}

/* Query Box Styling */
.query-box {
  background-color: #29294d;
  border: 2px solid #fff;
  border-radius: 10px;
  padding: 15px;
  margin-bottom: 15px;
}

textarea {
  width: 100%;
  padding: 8px;
  margin-bottom: 10px;
  box-sizing: border-box;
  border: 1px solid #ccc;
  border-radius: 4px;
}

input[type="submit"] {
  background-color: #007bff;
  color: #fff;
  padding: 8px 15px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}
```

```

input[type="submit"]:hover {
    background-color: #0056b3;
}

/* Solved Message Styling */
.solved-message {
    color: #28a745;
    font-weight: bold;
}

/* Back to Dashboard Styling */
.back-link {
    display: block;
    margin-top: 20px;
    text-align: center;
    color: #007bff;
    text-decoration: none;
    cursor: pointer;
}

/* Back to Dashboard Styling */
.back-to-dashboard {
    position: fixed;
    bottom: 20px;
    left: 50%;
    transform: translateX(-50%);
    background-color: #007bff;
    color: #fff;
    padding: 10px 20px;
    border-radius: 5px;
    text-decoration: none;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.back-to-dashboard:hover {
    background-color: #0056b3;
}

```

Ongoing\_ride.php

```

<!-- this code is for ongoing ride, having current ride details -->
<?php
session_start();

```

```

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

// Check if the cycle_id is set in the session; if not, redirect to the
cycle_info page
if (!isset($_SESSION["cycle_id"])) {
    header("Location: cycle_info.php");
    exit;
}

$vit_registration_number = $_SESSION["vit_registration_number"];
$cycle_id = $_SESSION["cycle_id"];

// Fetch user information
$user_info_sql = "SELECT * FROM user_info WHERE vit_registration_number =
'$vit_registration_number'";
$user_info_result = $conn->query($user_info_sql);

if ($user_info_result->num_rows == 1) {
    $user_info = $user_info_result->fetch_assoc();
}

// Fetch cycle information
$cycle_info_sql = "SELECT * FROM cycle_info WHERE cycle_id = $cycle_id";
$cycle_info_result = $conn->query($cycle_info_sql);

if ($cycle_info_result->num_rows == 1) {
    $cycle_info = $cycle_info_result->fetch_assoc();
} else {
    // Handle the case where cycle information is not found
    echo "Cycle information not found.";
    exit;
}

```

```

// Check if the ride is ongoing
if (isset($_SESSION["ride_start_time"]) && isset($_SESSION["ride_duration"]))
{
    $start_time = $_SESSION["ride_start_time"];
    $duration = $_SESSION["ride_duration"];

    // Calculate elapsed time
    $current_time = time();
    $elapsed_time = $current_time - $start_time;

    // Check if the ride duration has passed
    if ($elapsed_time >= $duration) {
        // Calculate the base cost
        $base_cost = 10; // You can adjust this value

        // Calculate the overtime cost
        $overtime_minutes = ceil(($elapsed_time - $duration) / 60);
        $overtime_cost = $overtime_minutes * 0.5; // Adjust the overtime rate
as needed

        // Calculate the total cost
        $total_cost = $base_cost + $overtime_cost;

        // Deduct the cost from the user's wallet (assuming you have a wallet
system)
        $wallet_sql = "UPDATE wallet SET balance = balance - $total_cost WHERE
vit_registration_number = '$vit_registration_number'";
        $conn->query($wallet_sql);

        // Update the ride completion details in the database (you may need to
adjust this based on your database schema)
        $ride_id = $_SESSION["ride_id"];
        $completion_time = date("Y-m-d H:i:s");
        $update_ride_sql = "UPDATE rides SET status = 'completed',
completion_time = '$completion_time', total_cost = $total_cost WHERE ride_id =
$ride_id";
        $conn->query($update_ride_sql);

        // Clear the ride-related session variables
        unset($_SESSION["ride_id"]);
        unset($_SESSION["ride_start_time"]);
        unset($_SESSION["ride_duration"]);

        // Redirect to a page indicating ride completion
        header("Location: ride_completed.php");
        exit;
    }
}

```



```

}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ongoing Ride Details</title>
    <link rel="stylesheet" type="text/css" href="css/ongoing_ride.css">
    <script>
        var initialDuration = 90; // Initial ride duration in seconds
        var extraTimeRate = 30; // Cost per minute for overtime in tokens

        function updateTimer() {
            // var startTime = <?php echo $_SESSION["ride_start_time"];
        };

        var elapsedTime = Math.max(0, time() - startTime);
        var remainingTime = Math.max(initialDuration - elapsedTime,
0);

        if (remainingTime <= 0) {
            // Stop updating timer if initial time is up
            document.getElementById("timer").innerHTML = "Initial Time
Expired";

            startExtraTime(startTime);
            return;
        }

        var minutes = Math.floor(remainingTime / 60);
        var seconds = remainingTime % 60;

        var formattedMinutes = (minutes < 10) ? '0' + minutes :
minutes;
        var formattedSeconds = (seconds < 10) ? '0' + seconds :
seconds;

        var timerDisplay = "Timer: " + formattedMinutes + "m " +
formattedSeconds + "s";
        document.getElementById("timer").innerHTML = timerDisplay;

        // Schedule the next update in 1 second
        setTimeout(updateTimer, 1000);
    }

    function startExtraTime(startTime) {
        var extraTime = 0;

        // Start counting extra time
    }

```

```

        var extraTimeInterval = setInterval(function () {
            extraTime++;

            // Display the extra time
            displayExtraTime(extraTime);

            // Calculate and display extra cost
            var extraCost = Math.ceil(extraTime / 60) * extraTimeRate;
            displayExtraCost(extraCost);
        }, 1000);

        // Store the extra time interval ID in a session variable for
later use
        sessionStorage.setItem('extraTimeInterval',
extraTimeInterval);
    }

    function displayExtraTime(extraTime) {
        var extraMinutes = Math.floor(extraTime / 60);
        var extraSeconds = extraTime % 60;
        var formattedExtraMinutes = (extraMinutes < 10) ? '0' +
extraMinutes : extraMinutes;
        var formattedExtraSeconds = (extraSeconds < 10) ? '0' +
extraSeconds : extraSeconds;

        var extraTimeDisplay = "Extra Time: " + formattedExtraMinutes
+ "m " + formattedExtraSeconds + "s";
        document.getElementById("extraTime").innerHTML =
extraTimeDisplay;
    }

    function displayExtraCost(extraCost) {
        var extraCostDisplay = "Extra Cost: " + extraCost + " tokens";
        document.getElementById("extraCost").innerHTML =
extraCostDisplay;
    }

    function time() {
        return Math.floor(new Date().getTime() / 1000);
    }

    // Initial call to start the timer
    updateTimer();

</script>
</head>

<body>

```

```

<nav class="navbar">
  <div class="navbar-container">
    <div class="logo">
      
    </div>
  </div>
</nav>

<section class="main-content">
  <div class="flex-item flex-head">
    <h1>Ongoing Ride</h1>
  </div>
  <div class="flex-item flex-col">

    <div class="review">
      
    </div>

    <div>
      <p class="text_">Name: <?php echo $user_info["name"]; ?></p>
    </div>

  </div>
  <div class="flex-item flex-col">
    <p class="text_">Cycle ID: <?php echo $cycle_info["cycle_id"];
?></p>
  </div>

  <div class="flex-item flex-head">
    <h1>Actions</h1>
  </div>

  <form method="post" action="ride_completed.php">
    <input type="submit" name="complete_ride" value="Complete Ride">
  </form>

</section>

```

Ongoing\_ride.css

```

@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{

```

```
    font-family: 'Tilt Neon', sans-serif;
}

body {
    background-color: #1b1b33;
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

/* Navigation Bar */
.navbar {
    display: flex;
    background-color: #29294d;
    justify-content: space-between;
    align-items: center;
    padding-right: 20px;
}

.navbar-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 100%;
    margin: 0 auto;
}

.navbar .logo img{
    width: 80px;
}

.main-content{
    display: flex;
    flex-wrap: wrap;
    height: 50vh;
    width: 80vw;
    margin: 0px auto;
    align-items: center;
    justify-content: center;
}

.flex-item {
    /* display: flex; */
    flex: 1 0 35%;
    padding: 5px;
    box-sizing: border-box;
    text-align: center;
    height: 25vh;
}
```

```

    backdrop-filter: blur(16px) saturate(200%);
    -webkit-backdrop-filter: blur(16px) saturate(200%);
    background-color: rgb(216, 213, 213);
    border-radius: 15px;
    border: 1px solid rgba(255, 255, 255, 0.125);
    margin: 10px;
    box-shadow: 0px 0px 40px 2px rgb(244 240 240 / 17%);
    transition: transform 0.5s ease-in-out;
    justify-content: space-around;
}

.flex-head{
    height: 10vh;
    margin-top: 20px;
    flex: 1 0 100%;
}

.review{
    display: flex;
    align-items: center;
    justify-content: left;
    margin-left: 50px;
}

.review img{
    width: 190px;
}

.flex-col{
    display: flex;
    justify-content: space-evenly;
    align-items: center;
    margin-top: 0px;
    padding-top: 0px;
}

.text_{
    font-size: 30px;
    font-weight: 600;
}

.container {
    text-align: center;
    background-color: white;
    padding: 20px;
    border-radius: 10px;
    width: 80%; /* Adjusted width for better display on larger screens */

```

```

margin: 20px auto;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
display: flex; /* Use flexbox for layout */
flex-wrap: wrap; /* Allow items to wrap to the next line if there's not
enough space */
justify-content: space-around; /* Add space between items */
}

h2, h3 {
color: #333;
width: 100%; /* Make headings take full width to start on a new line */
margin-top: 0; /* Remove default margin for consistency */
}

.info-box {
flex: 0 1 45%; /* Adjust flex properties to control item size */
box-sizing: border-box; /* Include padding and border in the width
calculation */
border: 1px solid #ddd; /* Add a border for separation */
padding: 20px;
margin: 10px;
border-radius: 10px;
text-align: left;
}

#timer {
font-size: 1.5em;
font-weight: bold;
color: #4CAF50;
margin-bottom: 20px;
}

#extraTime, #extraCost {
font-size: 1.2em;
color: #555;
}

form {
margin-top: 20px;
}

input[type="submit"] {
background-color: #4CAF50;
color: white;
font-size: 15px;
padding: 10px 20px;
border: none;
border-radius: 5px;
}

```

```

        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }

    a {
        color: #3498db;
        text-decoration: none;
    }

    a:hover {
        text-decoration: underline;
    }

```

Previous\_ride.php

```

<!-- this page stores history of previous rides -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

$vit_registration_number = $_SESSION["vit_registration_number"];

// fetching past rides
$past_rides_sql = "SELECT * FROM rides WHERE vit_registration_number = '$vit_registration_number'";

```

```
$past_rides_result = $conn->query($past_rides_sql);

?>
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Previous Rides</title>
  <link rel="stylesheet" href="css/user_dashboard.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      margin: 0;
    }

    h2 {
      text-align: center;
      color: #ccc;
    }

    .past-rides-table {
      width: 80%;
      margin: 20px auto;
      border-collapse: collapse;
    }

    .past-rides-table th,
    .past-rides-table td {
      border: 1px solid #ccc;
      padding: 10px;
      text-align: left;
    }

    .past-rides-table td{
      color: #ccc;
    }

    .past-rides-table th {
      background-color: #f0f0f0;
    }
  </style>
</head>

<body>
  <div>
    <h2>Previous Rides</h2>
    <table>
      <thead>
        <tr>
          <th>Ride ID</th>
          <th>Ride Name</th>
          <th>Start Date</th>
          <th>End Date</th>
          <th>Status</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>1</td>
          <td>Ride 1</td>
          <td>2023-01-01</td>
          <td>2023-01-02</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>2</td>
          <td>Ride 2</td>
          <td>2023-01-03</td>
          <td>2023-01-04</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>3</td>
          <td>Ride 3</td>
          <td>2023-01-05</td>
          <td>2023-01-06</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>4</td>
          <td>Ride 4</td>
          <td>2023-01-07</td>
          <td>2023-01-08</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>5</td>
          <td>Ride 5</td>
          <td>2023-01-09</td>
          <td>2023-01-10</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>6</td>
          <td>Ride 6</td>
          <td>2023-01-11</td>
          <td>2023-01-12</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>7</td>
          <td>Ride 7</td>
          <td>2023-01-13</td>
          <td>2023-01-14</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>8</td>
          <td>Ride 8</td>
          <td>2023-01-15</td>
          <td>2023-01-16</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>9</td>
          <td>Ride 9</td>
          <td>2023-01-17</td>
          <td>2023-01-18</td>
          <td>Completed</td>
        </tr>
        <tr>
          <td>10</td>
          <td>Ride 10</td>
          <td>2023-01-19</td>
          <td>2023-01-20</td>
          <td>Completed</td>
        </tr>
      </tbody>
    </table>
  </div>
</body>
</html>
```



```

        .button {
            background-color: orange;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            text-decoration: none;
            display: inline-block;
            margin-top: 20px;
        }

        .button:hover {
            background-color: darkorange;
        }
    </style>
</head>

<body>
    <h2>Previous Rides</h2>

    <?php if ($past_rides_result->num_rows > 0) { ?>
        <table border="1" class="past-rides-table">
            <tr>
                <th>Ride ID</th>
                <th>Cycle ID</th>
                <th>Ride Date</th>
                <th>Distance</th>
                <th>Duration</th>
                <th>Ride Cost</th>
            </tr>
            <?php while ($row = $past_rides_result->fetch_assoc()) { ?>
                <tr>
                    <td><?php echo $row["ride_id"]; ?></td>
                    <td><?php echo $row["cycle_id"]; ?></td>
                    <td><?php echo $row["ride_date"]; ?></td>
                    <td><?php echo $row["distance"]; ?></td>
                    <td><?php echo $row["duration"]; ?></td>
                    <td><?php echo $row["ride_cost"]; ?></td>
                </tr>
            <?php } ?>
        </table>
    <?php } else { ?>
        <p>No past rides found.</p>
    <?php } ?>

    <a href="user_dashboard.php" class="button">Back to User Dashboard</a>
</body>

```

```
</html>
```

Ride\_completed\_content.php

```
<!-- this page shows information of a completed ride -->
<?php
// session_start();

// Initialize variables
$thankYouMessage = "";

// Database connection
$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the cycle_id is set in the session
// if (!isset($_SESSION['cycle_id'])) {
//     // Redirect to select_cycle.php if cycle_id is not set
//     header("Location: select_cycle.php");
//     exit;
// }

// $cycleId = $_SESSION["cycle_id"];

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["submitReview"])) {
    // Reduce cycle health by 2 points
    $updateCycleHealthSql = "UPDATE cycle_info SET health = health - 2 WHERE
cycle_id = $cycleId";

    if ($conn->query($updateCycleHealthSql) === TRUE) {
        $thankYouMessage = '<p style="color: green;">Cycle health reduced by 2
points. Thank you for submitting the form!</p>';
    } else {
        $thankYouMessage = '<p style="color: red;">Error updating health: ' .
$conn->error . '</p>';
    }
}
```

```

}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Ride Completed</title>
    <link rel="stylesheet" href="css/ride_completed.css">
</head>
<body>

    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>

    <main>
        <div class="container">
            <h1>Ride Completed</h1>

            
            <!-- Your ride completion details -->
            <p>Total Distance: <?php echo $total_distance; ?> m</p>
            <p>Duration: <?php echo $ride_duration; ?> seconds</p>
            <p>Base Price: <?php echo $base_price; ?> tokens</p>
            <p>Overtime Cost: <?php echo $overtime_cost; ?> tokens</p>
            <p style="color: red;">Total Amount: <?php echo $total_amount; ?>
tokens</p>

            <!-- Display the thank you message -->
            <div id="thankYouMessage">
                <?php echo $thankYouMessage; ?>
            </div>

            <div>
                <h2>Cycle Review</h2>

                <!-- Display the review form -->
                <form id="reviewForm" method="post" action="logout.php">
                    <input type="submit" name="submitReview" value="Log Out">
                </form>
            </div>
        </div>
    </main>

```

```
</body>
</html>
```

Ride\_completed.css

```
@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{
  font-family: 'Tilt Neon', sans-serif;
}

body {
  margin: 0;
  padding: 0;
  background: #1b1b33;
  background-size: cover;
  background-repeat: no-repeat;
  text-align: center;
}

/* Navigation Bar */
.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding-right: 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  margin: 0 auto;
}

.navbar .logo img{
  width: 80px;
}

h1 {
  color: black;
```

```
    margin-top: 20px;
}

p {
    color: #555;
    margin: 10px 0;
    font-size: 25px;
}

.button {
    background-color: #ff6600;
    color: black;
    padding: 10px 20px;
    border: none;
    cursor: pointer;
    margin-top: 20px;
}

.button:hover {
    background-color: #cc5500;
}

main{
    height: 85vh;
    display: flex;
    justify-content: center;
    align-items: center;
}

.container {
    width: 50vw;
    margin: 20px auto;
    background-color: #fff;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 20px;
}

.container img{
    width: 12em;
}

input[type="submit"] {
    background-color: rgb(255, 154, 93);
    color: #fff;
    border: none;
    padding: 10px 15px;
    cursor: pointer;
}
```

```
    font-size: 15px;
    border-radius: 10px;
}

input[type="submit"]:hover {
    background-color: rgb(255, 160, 93);
}
```

Ride\_completed.php

```
<!-- this page make all the updation of ride in database -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

// Check if the cycle_id and ride_start_time are set in the session; if not,
redirect to the ongoing ride page
if (!isset($_SESSION["cycle_id"]) || !isset($_SESSION["ride_start_time"])) {
    header("Location: ongoing_ride.php");
    exit;
}

// Check if the cycle_id is set in the session; if not, redirect to the
cycle_info page
if (!isset($_SESSION["cycle_id"])) {
    header("Location: cycle_info.php");
    exit;
}
```

```

}

$vit_registration_number = $_SESSION["vit_registration_number"];
$cycle_id = $_SESSION["cycle_id"];
$ride_start_time = $_SESSION["ride_start_time"];

// Fetch user wallet information using prepared statements
$wallet_info_sql = "SELECT * FROM wallet WHERE vit_registration_number = ?";
$stmt = $conn->prepare($wallet_info_sql);
$stmt->bind_param("s", $vit_registration_number);
$stmt->execute();
$result = $stmt->get_result();

if ($result) {
    $wallet_info = $result->fetch_assoc();

    // Check if the wallet information is retrieved successfully
    if ($wallet_info) {
        // Check if the "wallet_balance" key exists in the array
        if (isset($wallet_info["wallet_balance"])) {
            // Calculate the ride duration
            $current_time = time();
            $ride_duration = $current_time - $ride_start_time;

            // Calculate tokens to be deducted (base price + overtime)
            $base_price = 10;
            $overtime_price = max(0, ceil($ride_duration / 60) - 15) * 0.5;
            $total_price = $base_price + $overtime_price;

            // Check if the user has enough tokens in the wallet
            if ($wallet_info["wallet_balance"] >= $total_price) {
                // Deduct tokens from the wallet using prepared statements
                $new_balance = $wallet_info["wallet_balance"] - $total_price;
                $update_wallet_sql = "UPDATE wallet SET wallet_balance = ?
WHERE vit_registration_number = ?";
                $stmt = $conn->prepare($update_wallet_sql);
                $stmt->bind_param("ds", $new_balance,
$vit_registration_number);
                $stmt->execute();

                // Update the ride duration in the database
                $update_duration_sql = "UPDATE cycle_info SET total_time =
total_time + ? WHERE cycle_id = ?";
                $stmt = $conn->prepare($update_duration_sql);
                $stmt->bind_param("ii", $ride_duration, $cycle_id);
                $stmt->execute();

                // Update total_rides in cycle_info

```

```

        $update_cycle_sql = "UPDATE cycle_info SET total_rides =
total_rides + 1 WHERE cycle_id = ?";
        $stmt = $conn->prepare($update_cycle_sql);
        $stmt->bind_param("i", $cycle_id);
        $stmt->execute();

        // Reduce the health of the cycle by 2
        $update_health_sql = "UPDATE cycle_info SET health =
GREATEST(health - 2, 0) WHERE cycle_id = ?";
        $stmt = $conn->prepare($update_health_sql);
        $stmt->bind_param("i", $cycle_id);
        $stmt->execute();

        // Clear the ride-related session variables
        unset($_SESSION["cycle_id"]);
        unset($_SESSION["ride_start_time"]);

        // Calculate distance based on average speed (7 meters per
second)
        $average_speed = 7;
        $calculated_distance = $average_speed * $ride_duration;

        // Update the total distance traveled in the database
        $update_distance_sql = "UPDATE cycle_info SET
distance_travelled = distance_travelled + ? WHERE cycle_id = ?";
        $stmt = $conn->prepare($update_distance_sql);
        $stmt->bind_param("di", $calculated_distance, $cycle_id);
        $stmt->execute();

        // Insert ride details into rides table
        $insert_ride_sql = "INSERT INTO rides (cycle_id, ride_date,
distance, duration, start_time, status, vit_registration_number, ride_cost)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
        $status = 'completed'; // Assuming 'completed' is the status
for a completed ride

        $current_time = time(); // Store the result of the time()
function in a variable
        $ride_date = date('Y-m-d H:i:s'); // Store the result of the
date() function in a variable

        $stmt = $conn->prepare($insert_ride_sql);
        $stmt->bind_param("ssddissd", $cycle_id, $ride_date,
$calculated_distance, $ride_duration, $ride_start_time, $status,
$vit_registration_number, $total_price);
        $stmt->execute();

        // Additional information for displaying

```



```

        $total_distance = $average_speed * $ride_duration;
        $overtime_cost = $overtime_price;
        $total_amount = $total_price;

        // Show the HTML content
        include('ride_completed_content.php');
        exit;
    } else {
        echo "Not enough tokens in the wallet.";
    }
} else {
    echo "Error: Missing 'wallet_balance' key in wallet information.";
}
} else {
    echo "Error: Wallet information not found.";
}
} else {
    echo "Error fetching wallet information: " . $conn->error;
}
?>

```

Ride\_info.php

```

<!-- this page track all ride history -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Fetch ride information
$rideInfoSql = "SELECT * FROM rides";
$rideInfoResult = $conn->query($rideInfoSql);
?>

<!DOCTYPE html>
<html lang="en">

```

```

<head>
    <meta charset="UTF-8">
    <title>Ride Information</title>
    <!-- Add your CSS Link here -->
    <link rel="stylesheet" href="css/ride_info.css">
</head>
<body>
    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>
    <div class="container">
        <h1>Ride Information</h1>
        <?php if ($rideInfoResult->num_rows > 0) { ?>
            <table border='1'>
                <thead>
                    <tr>
                        <th>Ride ID</th>
                        <th>User ID</th>
                        <th>Cycle ID</th>
                        <th>Ride Date</th>
                        <th>Distance (in m)</th>
                        <th>Duration</th>
                        <th>Ride Cost</th>
                    </tr>
                </thead>
                <tbody>
                    <?php while ($row = $rideInfoResult->fetch_assoc()) { ?>
                        <tr>
                            <td><?php echo $row["ride_id"]; ?></td>
                            <td><?php echo $row["vit_registration_number"];
?></td>

                            <td><?php echo $row["cycle_id"]; ?></td>
                            <td><?php echo $row["ride_date"]; ?></td>
                            <td><?php echo $row["distance"]; ?></td>
                            <td><?php echo $row["duration"]; ?></td>
                            <td><?php echo $row["ride_cost"]; ?></td>
                        </tr>
                    <?php } ?>
                </tbody>
            </table>
            <?php } else { ?>
                <p>No past rides found.</p>
            <?php } ?>
        </div>
    </body>

```

```

        <br>

        <a class="back-to-dashboard" href="dev_dashboard.php">Back to
Dashboard</a>
    </div>
</body>
</html>

```

Ride\_info.css

```

/* ride_info.css */

@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

body {
    font-family: 'Tilt Neon', sans-serif;
    background: #1b1b33;
    background-size: cover;
    background-repeat: no-repeat;
    margin: 0;
}

/* Navigation Bar */
.navbar {
    display: flex;
    background-color: #29294d;
    justify-content: space-between;
    align-items: center;
    padding: 10px 20px;
}

.navbar-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    width: 100%;
    margin: 0 auto;
}

```

```
.navbar .logo img {
    width: 80px;
}

/* Container Styling */
.container {
    margin: 10px auto;
    padding: 15px;
    border: 2px solid #ccc;
    border-radius: 10px;
    background-color: rgba(255, 255, 255, 0.9);
    width: 90%;
    overflow: auto; /* Add a scroll bar */
    text-align: center; /* Center-align content */
}

.container h1 {
    margin-bottom: 15px;
    color: #333;
}

/* Table Styling */
table {
    width: 100%;
    border-collapse: collapse;
    margin: 15px auto;
    border: 2px solid #ddd; /* Improved border color */
}

th, td {
    border: 2px solid #ddd; /* Improved border color */
    padding: 10px;
    text-align: center; /* Center-align content */
}

th {
    background-color: #f0f0f0;
}

/* Link Styling */
a {
    color: #007bff; /* Improved link color */
    text-decoration: none;
    cursor: pointer;
}

/* Back to Dashboard Styling */
```

```

.back-to-dashboard {
    position: fixed;
    bottom: 20px;
    left: 50%;
    transform: translateX(-50%);
    background-color: #007bff;
    color: #fff;
    padding: 10px 20px;
    border-radius: 5px;
    text-decoration: none;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.back-to-dashboard:hover {
    background-color: #0056b3;
}

/* Responsive Design */
@media only screen and (max-width: 768px) {
    .navbar-container {
        width: 90%;
    }

    .container {
        width: 95%;
    }
}

```

Ride\_login.php

```

<?php
session_start(); // Start a new session or resume the existing session

// Handle login when the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Establish a database connection (replace with your database
    credentials)
    $servername = "localhost";
    $username = "root";
    $password = "123456";
    $dbname = "v_rides";

    $conn = new mysqli($servername, $username, $password, $dbname);

```

```

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$vit_registration_number = $_POST["vit_registration_number"];
$passcode = $_POST["passcode"];

// Query to fetch user data by VIT registration number
$sql = "SELECT * FROM user_info WHERE vit_registration_number = '$vit_registration_number'";
$result = $conn->query($sql);

if ($result->num_rows == 1) {
    $row = $result->fetch_assoc();
    // Verify passcode using password_verify for securely hashed passcodes
    if (password_verify($passcode, $row["passcode"])) {
        // Authentication successful, set session variables
        $_SESSION["vit_registration_number"] = $vit_registration_number;
        $_SESSION["name"] = $row["name"];
        header("Location: cycle_ride.php"); // Redirect to the cycle ride
        exit;
    } else {
        $login_error = "Invalid passcode. Please try again.";
    }
} else {
    $login_error = "User not found. Please check your VIT registration
number.";
}

$conn->close();
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h2>Login</h2>
    <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
        <label>VIT Registration Number:</label>
        <input type="text" name="vit_registration_number" required><br>

        <label>Passcode:</label>
        <input type="password" name="passcode" required><br>

        <?php

```

```

        if (isset($login_error)) {
            echo '<p style="color: red;">' . $login_error . '</p>';
        }
    ?>

    <input type="submit" value="Login">
</form>
</body>
</html>

```

Select\_cycle.php

```

<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

// Fetch available cycles
$cycles_sql = "SELECT * FROM cycle_info WHERE status = 'available'";
$cycles_result = $conn->query($cycles_sql);

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Select Cycle</title>
    <link rel="stylesheet" type="text/css" href="css/select_cycle.css">
</head>

```

```

<body>
  <header>
    
    <h1>V-Rides</h1>
  </header>

  <div class="content">
    <h2>Select a Cycle</h2>

    <?php if ($cycles_result->num_rows > 0) { ?>
      <ul>
        <?php while ($cycle = $cycles_result->fetch_assoc()) { ?>
          <li>
            <!-- Display the cycle link with an identifier -->
            <strong>Cycle Link for Cycle ID <?php echo
$cycle['cycle_id']; ?>:</strong>
            <a href="user_authentication.php?cycle_id=<?php echo
$cycle['cycle_id']; ?>" >
              <?php echo $cycle['cycle_link']; ?>
            </a>
          </li>
        <?php } ?>
      </ul>
    <?php } else { ?>
      <p>No cycles available at the moment.</p>
    <?php } ?>

    <br>
    <a href="user_dashboard.php">Back to Dashboard</a>
  </div>
</body>
</html>

```

Select\_cycle.css

```

/* Reset some default styles */
body, h1, h2, h3, p, ul {
  margin: 0;
  padding: 0;
}

body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
}

```



```
header {
  background-color: #333;
  color: white;
  text-align: center;
  padding: 10px 0;
}

header img {
  height: 40px; /* Adjust the height based on your Logo size */
  margin-right: 10px;
}

header h1 {
  display: inline-block;
  font-size: 1.5em;
}

.content {
  max-width: 800px;
  margin: 20px auto;
}

h2 {
  color: #333;
  margin-bottom: 20px;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  margin-bottom: 20px;
  background-color: #fff;
  border: 1px solid #ddd;
  padding: 15px;
  border-radius: 8px;
  transition: background-color 0.3s, box-shadow 0.3s;
  cursor: pointer;
}

li:hover {
  background-color: #f0f0f0;
}

.selected {
  background-color: #3498db;
```

```

    color: #fff;
    box-shadow: 0 0 10px rgba(52, 152, 219, 0.8);
}

a {
    color: #3498db;
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

```

User\_authentication.php

```

<!-- this page authenticate user after scanning QR -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if vit_registration_number and cycle_id are provided in the
GET request
if ($_SERVER["REQUEST_METHOD"] == "GET" && isset($_GET['cycle_id'])) {
    $cycle_id = $_GET['cycle_id'];

    // You might want to add more authentication logic here
    // For now, we'll assume a simple check

    // Check if the cycle exists
    $cycle_check_sql = "SELECT * FROM cycle_info WHERE cycle_id =
$cycle_id";
    $cycle_check_result = $conn->query($cycle_check_sql);

```

```

    if ($cycle_check_result->num_rows == 1) {
        // Cycle exists, set session variable

        // Check if the user's account is suspended
        $user_registration_number =
$_SESSION['vit_registration_number'] ?? null;
        $user_suspended_sql = "SELECT suspended FROM user_info WHERE
vit_registration_number = '$user_registration_number'";
        $user_suspended_result = $conn->query($user_suspended_sql);

        if ($user_suspended_result->num_rows == 1) {
            $user_suspended = $user_suspended_result-
>fetch_assoc()["suspended"];

            if ($user_suspended == 1) {
                // User's account is suspended, display an error
message
                $error_message = "Your account is suspended. You can't
proceed with the ride.";
            } else {
                // User's account is not suspended, redirect to the
cycle ride page
                $_SESSION['cycle_id'] = $cycle_id;
                header("Location: cycle_ride.php");
                exit;
            }
        } else {
            // Unable to check account status, display an error message
            $error_message = "Unable to check your account status.
Please try again.";
        }
    } else {
        // Cycle not found, display an error message
        $error_message = "Invalid Cycle ID. Please try again.";
    }
}

// Close the database connection
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>User Authentication</title>

```

```

</head>
<body>
    <h2>User Authentication</h2>

    <?php if (isset($error_message)): ?>
        <p style="color: red;"><?php echo $error_message; ?></p>
    <?php endif; ?>

    <!-- Form not needed since we are using GET parameters -->
</body>
</html>

```

User\_dashboard.php

```

<!-- this is code for user dashboard where user will control all
activites -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

$vit_registration_number = $conn-
>real_escape_string($_SESSION["vit_registration_number"]);

// Fetch user information
$user_info_sql = "SELECT * FROM user_info WHERE vit_registration_number
= '$vit_registration_number'";
$user_info_result = $conn->query($user_info_sql);

```

```

if ($user_info_result->num_rows == 1) {
    $user_info = $user_info_result->fetch_assoc();
} else {
    // Handle the case where user information is not found
    echo "User information not found.";
    exit;
}

// Fetch past rides
$past_rides_sql = "SELECT * FROM rides WHERE vit_registration_number = '$vit_registration_number'";
$past_rides_result = $conn->query($past_rides_sql);

// Check if there are past rides
if ($past_rides_result->num_rows > 0) {
    $rides = $past_rides_result->fetch_all(MYSQLI_ASSOC);
    $displayedRides = array_slice($rides, 0, 3);
} else {
    $displayedRides = array(); // Set $displayedRides to an empty array
if there are no past rides
}

// Fetch wallet details
$wallet_sql = "SELECT * FROM wallet WHERE vit_registration_number = '$vit_registration_number'";
$wallet_result = $conn->query($wallet_sql);

if ($wallet_result->num_rows == 1) {
    $wallet_info = $wallet_result->fetch_assoc();
} else {
    // Handle the case where wallet information is not found
    echo "Wallet information not found.";
    exit;
}

// Submit feedback
if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST["submit_feedback"])) {
    $user_email = $user_info["email_id"]; // Assuming 'email' is a
field in the 'user_info' table

    // Sanitize the input to prevent SQL injection
    $feedback = $conn->real_escape_string($_POST["feedback"]);

    // Use prepared statement to prevent SQL injection

```

```

        $insertFeedbackSql = $conn->prepare("INSERT INTO feedback (name,
vit_registration_number, email, query, solved) VALUES (?, ?, ?, ?,
0)");
        $insertFeedbackSql->bind_param("ssss", $user_info["name"],
$user_info["vit_registration_number"], $user_email, $feedback);

        if ($insertFeedbackSql->execute()) {
            echo '<p style="color: green;">Feedback submitted
successfully!</p>';
        } else {
            echo '<p style="color: red;">Error submitting feedback: ' .
$conn->error . '</p>';
        }

        $insertFeedbackSql->close();
    }

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>User Dashboard</title>
    <link rel="stylesheet" href="css/user_dashboard.css">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <style>
        /* Additional CSS for improved styling */
        .feedback-form {
            margin-top: 20px;
        }
    </style>
</head>

<body>
    <nav class="navbar">
        <div class="navbar-container">
            <div class="logo">
                
            </div>
        </div>
    </nav>

```

```

<section class="flex">
    <div class="left-nav">
        <div class="review">
            
        </div>
        <h1>Welcome back,</h1>
        <p><?php echo htmlspecialchars($user_info["name"]); ?></p>
        <?php
        // Assuming $user_info is the user information array
        $isSuspended = $user_info["suspended"];

        // Check if the account is not suspended before showing the
"Add Token" button
        if (!$isSuspended) {
            echo '<form method="post" action="wallet.php">';
            echo '<button type="submit" class="active">Add
Token</button>';
            echo '</form>';
        }
        ?>

        <button class="active">
            <a href="previous_rides.php" class="button">View
Previous Rides</a>
        </button>
        <button class="active">
            <a href="user_queries.php" class="button">View Previous
Queries</a>
        </button>
        <form method="post" action="logout.php"> <!-- Add a form
for Logout -->
            <button class="active" type="submit"
name="logout">Logout</button>
        </form>
    </div>

    <div class="main-content">
        <div class="flex-item <?php echo ($isSuspended) ? 'account-
suspended' : 'text-warnings'; ?>">
            <h3>Registration number:</h3>
            <p class="text_">Reg. no. : <?php echo
htmlspecialchars($user_info["vit_registration_number"]); ?></p>

            <?php
            // Check if there are warnings

```

```

        $warningCount = $user_info["warnings"]; // Assuming
warnings field in the table
        if ($warningCount > 0) {
            echo '<h3>Warning</h3>';
            echo '<p>Number of warnings given: ' .
$warningCount . '</p>';
        }

        // Check if the account is suspended
        if ($isSuspended) {
            echo '<h3>Your account is suspended</h3>';
        }
    ?>
</div>

<div class="flex-item">
    <h3>Wallet Details</h3>
    <div class="wallet">
        
    </div>
    <p>Wallet Balance: <?php echo
htmlspecialchars($wallet_info["wallet_balance"]); ?> tokens</p>
</div>

<div class="flex-item feedback">
    <h3>Feedback and Queries</h3>
    <form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>" class="feedback-form">
        <label for="feedback">Your Feedback or
Query:</label>
        <textarea name="feedback" id="feedback" rows="4"
cols="50" required></textarea>
        <button type="submit" class="button"
name="submit_feedback">Submit</button>
    </form>

</div>

<div class="flex-item">
    <h3>Past Rides</h3>
    <?php if (!empty($displayedRides)) { ?>
        <table border="1" class="past-rides-table">
            <tr>
                <th>Ride ID</th>
                <th>Cycle ID</th>
                <th>Ride Date</th>

```



```

                <th>Distance (in m)</th>
                <th>Duration</th>
                <th>Ride Cost</th>
            </tr>
            <?php foreach ($displayedRides as $row) { ?>
                <tr>
                    <td><?php echo
htmlspecialchars($row["ride_id"]); ?></td>
                    <td><?php echo
htmlspecialchars($row["cycle_id"]); ?></td>
                    <td><?php echo
htmlspecialchars($row["ride_date"]); ?></td>
                    <td><?php echo
htmlspecialchars($row["distance"]); ?></td>
                    <td><?php echo
htmlspecialchars($row["duration"]); ?></td>
                    <td><?php echo
htmlspecialchars($row["ride_cost"]); ?></td>
                </tr>
            <?php } ?>
        </table>

        <?php } else { ?>
            <p>No past rides found.</p>
        <?php } ?>
    </div>
</div>
</section>

</body>

</html>
<?php

?>

```

User\_dashboard.css

```

/* user_dashboard.css */

@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

```

```
* {
  font-family: 'Tilt Neon', sans-serif;
}

body {
  margin: 0;
  padding: 0;
  background: #1b1b33;
  background-size: cover;
  background-repeat: no-repeat;
}

/* Navigation Bar */
.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding-right: 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  margin: 0 auto;
}

.navbar .logo img {
  width: 80px;
}

.left-nav {
  display: flex;
  position: relative;
  flex-direction: column;
  background-color: #29294d;
  color: white;
  width: fit-content;
  padding: 20px;
  height: 85vh;
}

.left-nav h1 {
```

```
        font-size: 30px;
    }

    .left-nav h2 {
        font-size: 20px;
        color: #ccc;
    }

    .left-nav p {
        color: #ccc;
        font-size: 24px;
        margin-top: -5px;
    }

    .review {
        display: flex;
        align-items: center;
        justify-content: center;
    }

    .review img {
        width: 200px;
    }

    .active {
        width: 100%;
        font-size: 20px;
        margin: 10px;
        margin-left: 0px;
        color: #ccc;
        background-color: #29294d;
    }

    .text_ {
        font-size: 30px;
        font-weight: 600;
    }

    .flex {
        display: flex;
        flex-direction: row;
    }

    .main-content {
        display: flex;
        flex-wrap: wrap;
    }
```

```

    width: 80vw;
    margin: 0px auto;
    justify-content: space-around;
}

.flex-item {
    flex: 1 0 30%;
    /* Adjust this value as per your preference */
    padding: 10px;
    box-sizing: border-box;
    text-align: center;
    height: 35vh;
    backdrop-filter: blur(16px) saturate(200%);
    -webkit-backdrop-filter: blur(16px) saturate(200%);
    background-color: rgb(216, 213, 213);
    border-radius: 30px;
    border: 1px solid rgba(255, 255, 255, 0.125);
    margin: 20px;
    box-shadow: 0px 0px 40px 2px rgb(244 240 240 / 17%);
    transition: transform 0.5s ease-in-out;
    justify-content: space-around;
}

.text-warning {
    background-color: #f1f18e; /* Light yellow */
}

.account-suspended {
    background-color: #ffcccc;
}

.wallet {
    display: flex;
    align-items: center;
    justify-content: center;
}

.wallet img {
    width: 100px;
    height: 100px;
}

.buttons button {
    background-color: #ff8c00;
    color: white;
    padding: 8px 15px;
}

```

```
    border: none;
    border-radius: 5px;
    margin-left: 10px;
    transition: background-color 0.3s ease;
}

.buttons button:hover {
    background-color: #e07b00;
}

.outerr {
    height: 90vh;
    display: flex;
    align-items: center;
    justify-content: center;
}

.feedback-form {
    margin-top: 15px;
}

label {
    display: block;
    margin-bottom: 5px;
    font-weight: bold;
}

textarea {
    width: 100%;
    padding: 8px;
    margin-bottom: 10px;
    box-sizing: border-box;
    border: 1px solid #ccc;
    border-radius: 4px;
}

.button {
    display: inline-block;
    padding: 8px 12px;
    margin-right: 10px;

    /* color: white; */
    text-decoration: none;
    border-radius: 4px;
}
```

```
        cursor: pointer;
    }

    /* User Information and Past Rides Container */
    .container {
        margin: 10px auto;
        padding: 15px;
        border: 1px solid #ccc;
        border-radius: 10px;
        background-color: rgba(255, 255, 255, 0.9);
        width: 90%;
    }

    .container h2 {
        margin-bottom: 15px;
    }

    /* Past Rides Table */
    .past-rides-table {
        width: 100%;
        border-collapse: collapse;
        margin-bottom: 15px;
    }

    .past-rides-table th,
    .past-rides-table td {
        border: 1px solid #ccc;
        padding: 8px;
        text-align: left;
    }

    .past-rides-table th {
        background-color: #f0f0f0;
    }

    /* Link Styling */
    a {
        color: #ccc;
        text-decoration: none;
        cursor: default;
    }

    /* Additional Suggestions */

    /* Responsive Design */
```

```

@media only screen and (max-width: 768px) {
    .navbar-container {
        width: 90%;
    }

    .container {
        width: 95%;
    }
}

/* Improved Styling */
body {
    color: #333;
}

/* Navigation Bar Styling */
.buttons button {
    cursor: pointer;
    transition: background-color 0.3s ease, transform 0.2s ease;
}

.buttons button:hover {
    background-color: #e07b00;
    transform: scale(1.05);
}

/* Typography */
body {
    font-size: 14px;
    line-height: 1.6;
}

h2 {
    color: #333;
}

/* Icons */
.action-buttons button::after {
    content: "\2022"; /* Unicode character for a bullet (can be
replaced with an icon font) */
    margin-left: 5px;
    font-size: 8px;
    vertical-align: middle;
}

/* Animations */

```

```
.buttons button,
.action-buttons button {
    transition: background-color 0.3s ease, transform 0.2s ease;
}

/* Whitespace */
.container {
    margin: 10px auto;
    padding: 15px;
}

/* Action Buttons Styling */
.action-buttons {
    display: flex;
    justify-content: space-between;
    margin-top: 20px;
}

.action-buttons form {
    margin-right: 10px; /* Adjust the spacing between the buttons
*/
}

.action-buttons button {
    background-color: orange; /* Orange button color */
    color: white; /* White text color */
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.action-buttons button:hover {
    background-color: darkorange; /* Darker orange on hover */
}

.back-link {
    display: block;
    margin-top: 20px;
    text-align: center;
    color: #007bff;
    text-decoration: none;
}
```



## User\_info.php

```
<!-- this page contains all user informations -->
<?php
// Database connection code here (replace this with your actual
database connection code)
$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Handle user removal
if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST["remove_user"])) {
    $removeRegistrationNumber =
$_POST["registration_number_to_remove"];
    $removeUserSql = "DELETE FROM user_info WHERE
vit_registration_number = '$removeRegistrationNumber'";

    if ($conn->query($removeUserSql) === TRUE) {
        // Redirect to refresh the page after removing the user
        header("Location: " . $_SERVER['PHP_SELF']);
        exit;
    } else {
        $error_message = "Error removing user: " . $conn->error;
    }
}

// Handle adding money to wallet
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["add_money"]))
{
    $amount_to_add = $_POST["amount"];
    $vit_registration_number = $_POST["vit_registration_number"];

    // Update wallet balance
    $update_wallet_sql = "UPDATE wallet SET wallet_balance =
wallet_balance + '$amount_to_add' WHERE vit_registration_number =
'$vit_registration_number'";
```

```

$conn->query($update_wallet_sql);

// Redirect to user_info.php to refresh the page
header("Location: user_info.php");
exit;
}

// Fetch user count
$userCountSql = "SELECT COUNT(*) as total_users FROM user_info";
$userCountResult = $conn->query($userCountSql);
$total_users = ($userCountResult->num_rows == 1) ? $userCountResult->fetch_assoc()["total_users"] : 0;

// Fetch user information
$userInfoSql = "SELECT id, vit_registration_number, name, warnings, suspended FROM user_info";
$userInfoResult = $conn->query($userInfoSql);

// Fetch additional details like wallet balance, total rides, warnings, and suspension status
$userDetails = array();
while ($row = $userInfoResult->fetch_assoc()) {
    $registrationNumber = $row["vit_registration_number"];

    $walletSql = "SELECT wallet_balance FROM wallet WHERE vit_registration_number = '$registrationNumber'";
    $walletResult = $conn->query($walletSql);
    $walletBalance = ($walletResult->num_rows == 1) ? $walletResult->fetch_assoc()["wallet_balance"] : 0;

    $totalRidesSql = "SELECT COUNT(*) AS total_rides FROM rides WHERE vit_registration_number = '$registrationNumber'";
    $totalRidesResult = $conn->query($totalRidesSql);
    $totalRides = ($totalRidesResult->num_rows == 1) ? $totalRidesResult->fetch_assoc()["total_rides"] : 0;

    $userDetails[] = array(
        "id" => $row["id"],
        "registration_number" => $registrationNumber,
        "name" => $row["name"],
        "batch" => isset($row["batch"]) ? $row["batch"] : "Not Available",
        "email_id" => isset($row["email_id"]) ? $row["email_id"] : "Not Available",
        "birthday" => isset($row["birthday"]) ? $row["birthday"] : "Not Available",
    );
}

```

```

        "wallet_balance" => $walletBalance,
        "total_rides" => $totalRides,
        "warnings" => $row["warnings"],
        "suspended" => $row["suspended"]
    );
}

// Handle suspending user account
if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST["suspend_user"])) {
    $suspendRegistrationNumber =
$_POST["registration_number_to_suspend"];
    $suspendUserSql = "UPDATE user_info SET suspended = 1 WHERE
vit_registration_number = '$suspendRegistrationNumber'";

    if ($conn->query($suspendUserSql) === TRUE) {
        // Redirect to refresh the page after suspending the user
        header("Location: " . $_SERVER['PHP_SELF']);
        exit;
    } else {
        $error_message = "Error suspending user: " . $conn->error;
    }
}

// Handle unsuspending user account
if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST["unsuspend_user"])) {
    $unsuspendRegistrationNumber =
$_POST["registration_number_to_unsuspend"];
    $unsuspendUserSql = "UPDATE user_info SET suspended = 0 WHERE
vit_registration_number = '$unsuspendRegistrationNumber'";

    if ($conn->query($unsuspendUserSql) === TRUE) {
        // Redirect to refresh the page after unsuspending the user
        header("Location: " . $_SERVER['PHP_SELF']);
        exit;
    } else {
        $error_message = "Error unsuspending user: " . $conn->error;
    }
}

// Handle giving a warning to a user
if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST["give_warning"])) {
    $warningRegistrationNumber = $_POST["registration_number_to_warn"];

```

```

    $giveWarningSql = "UPDATE user_info SET warnings = warnings + 1
WHERE vit_registration_number = '$warningRegistrationNumber'";

    if ($conn->query($giveWarningSql) === TRUE) {
        // Redirect to refresh the page after giving a warning
        header("Location: " . $_SERVER['PHP_SELF']);
        exit;
    } else {
        $error_message = "Error giving a warning to the user: " .
$conn->error;
    }
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>User Information</title>
    <link rel="stylesheet" href="css/user_info.css">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min
.css">
    <script src="https://code.jquery.com/jquery-
3.2.1.slim.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper
.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.j
s"></script>

    <!-- Include your CSS file or add styles here -->
</head>

<body>
    <div class="container">

        <h2>User Information</h2>

        <?php
        if (isset($error_message)) {
            echo '<p style="color: red;">' . $error_message . '</p>';

```

```

}
?>

<p>Total Number of Users: <?php echo $total_users; ?></p>

<table border="1" class="cycle-table">
    <tr>
        <th>Name</th>
        <th>Registration Number</th>
        <th>Wallet Balance</th>
        <th>Total Rides</th>
        <th>Warnings</th>

        <th>Action</th>

    </tr>
    <?php foreach ($userDetails as $user) { ?>
        <tr <?php echo ($user['suspended'] == 1) ?
'<div class="suspended">' : (($user['warnings'] >= 1) ? '<div class="warned">' :
''); ?>>

            <td><?php echo $user["name"]; ?></td>
            <td><?php echo $user["registration_number"];
?></td>

            <td><?php echo $user["wallet_balance"]; ?></td>
            <td><?php echo $user["total_rides"]; ?></td>
            <td><?php echo $user["warnings"]; ?></td>

            <td>
                <div class="button-container">
                    <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
                        <input type="hidden"
name="registration_number_to_remove" value="<?php echo
$user['registration_number']; ?>">
                        <input type="submit" name="remove_user"
value="Remove" class='delete-button' onclick="return
confirmUserRemoval();">
                    </form>

                    <?php if ($user['suspended'] == 1) { ?>
                        <!-- If user is suspended, show
Unsuspend button -->

                        <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">

```

```

                <input type="hidden"
name="registration_number_to_unsuspend" value="<?php echo
$user['registration_number']; ?>">
                <input type="submit"
name="unsuspend_user" value="Unsuspend" class='unsuspend-button'
onclick="return confirm('Are you sure you want to unsuspend this
user?');">
            </form>
            <?php } else { ?>
            <!-- If user is not suspended, show
Suspend and Give Warning buttons -->
            <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
                <input type="hidden"
name="registration_number_to_suspend" value="<?php echo
$user['registration_number']; ?>">
                <input type="submit"
name="suspend_user" value="Suspend" class='suspend-button'
onclick="return confirm('Are you sure you want to suspend this
user?');">
            </form>
            <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
                <input type="hidden"
name="registration_number_to_warn" value="<?php echo
$user['registration_number']; ?>">
                <input type="submit"
name="give_warning" value="Give Warning" class='warning-button'
onclick="return confirm('Are you sure you want to give a warning to
this user?');">
            </form>
            <button
onclick="showAddMoneyOverlay('<?php echo $user['registration_number'];
?>')" class="add-money-button">Add Money</button>

            <?php } ?>
        </div>
    </td>

    </tr>
    <?php } ?>
</table>

```

```

        <br>
        <a href="dev_dashboard.php" class="back-link">Back to
Dashboard</a>

        <div id="overlay">
            <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
                <label>Amount (in tokens):</label>
                <input type="number" name="amount" required>
                <input type="hidden" name="vit_registration_number"
id="overlay-vit-registration-number" value="">
                <input type="submit" name="add_money" value="Add
Money">
                <button type="button"
onclick="hideAddMoneyOverlay()">Cancel</button>
            </form>
        </div>

        <!-- Script for confirmation on user removal -->
        <script>
            function confirmUserRemoval() {
                return confirm('Are you sure you want to remove this
user?');
            }
        </script>

    </div>

    <script>
        function showAddMoneyOverlay(registrationNumber) {
            document.getElementById('overlay-vit-registration-
number').value = registrationNumber;
            document.getElementById('overlay').style.display = 'block';
        }

        function hideAddMoneyOverlay() {
            document.getElementById('overlay').style.display = 'none';
        }
    </script>

</body>

</html>

```

User\_info.css

```
/* cycle_info.css */

body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #1b1b33;
}

.container {
  margin: 20px auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
  background-color: #fff;
  width: 80%;
}

h2 {
  text-align: center;
}

#total-cycles {
  text-align: center;
}

.error-message {
  color: red;
}

.button {
  background-color: #007bff;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

/* Add this to your existing CSS file or create a new one */

.delete-button {
  background-color: #dc3545;
```



```
    color: #fff;
    padding: 7px 10px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.delete-button:hover {
    background-color: #c82333;
}

/* Additional styles for better visibility */
.delete-button:active {
    transform: translateY(1px);
    box-shadow: 0 1px 2px rgba(0, 0, 0, 0.1);
}

.cycle-table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}

.cycle-table th,
.cycle-table td {
    border: 1px solid #ccc;
    padding: 10px;
    text-align: left;
}

.cycle-table th {
    background-color: #f0f0f0;
}

.back-link {
    display: block;
    margin-top: 20px;
    text-align: center;
    color: #007bff;
    text-decoration: none;
}

.back-link:hover {
    text-decoration: underline;
}
```

```
}

.cycle-table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
}

.cycle-table th, .cycle-table td {
  padding: 8px;
  text-align: left;
}

.delete-button {
  background-color: #ff5c5c;
  color: white;
  padding: 5px 10px;
  border: none;
  border-radius: 3px;
  cursor: pointer;
}

#overlay {
  display: none;
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.7);
  justify-content: center;
  align-items: center;
}

#overlay form {
  background: white;
  padding: 20px;
  border-radius: 5px;
  text-align: center;
}

tr.suspended {
```

```
        background-color: #FFD6D6; /* Light red */
    }

    tr.warned {
        background-color: #FFFFCC; /* Light yellow */
    }

    /* Add these styles to your existing CSS or create a new CSS file */
    .button-container {
        display: flex;
    }

    .delete-button,
    .suspend-button,
    .unsuspend-button,
    .warning-button,
    .add-money-button {
        margin-right: 10px;
        cursor: pointer;
        border: none;
        border-radius: 4px;
        text-align: center;
        text-decoration: none;
        font-size: 10px;
        transition-duration: 0.4s;
        padding: 8px 12px;
    }

    .delete-button {
        background-color: #ff3333; /* Red */
        color: white;
    }

    .suspend-button {
        background-color: #ffcc00; /* Yellow */
        color: black;
    }

    .unsuspend-button {
        background-color: #66ff66; /* Light green */
        color: black;
    }

    .warning-button {
        background-color: #ff9900; /* Orange */
    }
}
```

```

        color: black;
    }

    .add-money-button {
        background-color: #4CAF50; /* Green */
        color: white;
    }

```

User\_queries.php

```

<!-- this contains all the query -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

$vit_registration_number = $_SESSION["vit_registration_number"];

// Fetch user information
$user_info_sql = "SELECT * FROM user_info WHERE vit_registration_number
= '$vit_registration_number'";
$user_info_result = $conn->query($user_info_sql);

if ($user_info_result->num_rows == 1) {
    $user_info = $user_info_result->fetch_assoc();
} else {
    // Handle the case where user information is not found
    echo "User information not found.";
}

```

```

        exit;
    }

    // Fetch feedback and queries along with solutions using a JOIN
    operation
    $user_queries_sql = "SELECT feedback.query_id, feedback.query,
                           feedback.solved, solutions.solution
                           FROM feedback
                           LEFT JOIN solutions ON feedback.query_id =
solutions.query_id
                           WHERE feedback.vit_registration_number =
'$vit_registration_number'";

    $user_queries_result = $conn->query($user_queries_sql);
    ?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>User Queries</title>
    <link rel="stylesheet" href="css/user_queries.css">
</head>

<nav class="navbar">
    <div class="navbar-container">
        <div class="logo">
            
        </div>
        <h1>User Queries</h1>
    </div>
</nav>

<body>
    <main>

        <div class="container">

            <div class="flex-items height">
                
            </div>

            <div class="flex-items height">

```

```

        <h3>User Information</h3>
        <p>Name: <?php echo $user_info["name"]; ?></p>
        <p>Reg. no.: <?php echo
$user_info["vit_registration_number"]; ?></p>
    </div>

    <div class="flex-items">

        <h3>Previous Queries</h3>

        <?php if ($user_queries_result->num_rows > 0) { ?>
            <table border="1" class="user-queries-table">
                <tr>
                    <th>Query ID</th>
                    <th>Query</th>
                    <th>Status</th>
                    <th>Response</th>
                </tr>
                <?php while ($row = $user_queries_result-
>fetch_assoc()) { ?>
                    <tr>
                        <td><?php echo $row["query_id"];
?></td>
                        <td><?php echo $row["query"]; ?></td>
                        <td><?php echo $row["solved"] ?
'Answered' : 'Pending'; ?></td>
                        <td><?php echo $row["solution"] ?? '';
?></td>
                    </tr>
                <?php } ?>
            </table>
            <?php } else { ?>
                <p>No previous queries found.</p>
            <?php } ?>
        </div>

        <br>

    </div>

</main>
<footer>
    <a href="user_dashboard.php">Back to Dashboard</a>
</footer>
</body>

```

```
</html>
```

User\_queries.css

```
@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{
  font-family: 'Tilt Neon', sans-serif;
}

body {
  margin: 0;
  padding: 0;
  background: #1b1b33;
  background-size: cover;
  background-repeat: no-repeat;
}

/* Navigation Bar */
.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding-right: 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  text-align: center; /* Add this line */
}

.navbar-container.h1 {
  color: #fff; /* Adjust the color as needed */
  margin: 0;
  display: inline-block; /* Add this line */
}

.navbar .logo img{
```

```

    width: 80px;
}

main{
    display: flex;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
    width: 80vw;
    margin: 20px auto;
    background-color: #fff;
    padding: 20px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    border-radius: 10px;
}

.flex-items {
    flex: 1 0 40%; /* Adjust this value as per your preference */
    padding: 10px;
    box-sizing: border-box;
    text-align: center;
    height: 35vh;
    backdrop-filter: blur(16px) saturate(200%);
    -webkit-backdrop-filter: blur(16px) saturate(200%);
    background-color: rgb(216, 213, 213);
    border-radius: 30px;
    border: 1px solid rgba(255, 255, 255, 0.125);
    margin: 20px;
    box-shadow: 0px 0px 40px 2px rgb(244 240 240 / 17%);
    transition: transform 0.5s ease-in-out;
    justify-content: space-around;
    overflow: hidden;
}

.height{
    height: 25vh;
}

.flex-items img{
    width: 20vh;
    height: 20vh;
}

/* Add this style to make the height dynamic with a scrollbar */
.flex-items {
    overflow: auto; /* Enable vertical scrolling */

```



```
    max-height: 35vh; /* Set the maximum height */
}

/* Customize the scrollbar */
.flex-items::-webkit-scrollbar {
    width: 8px;
}

.flex-items::-webkit-scrollbar-thumb {
    background-color: #888;
}

.flex-items::-webkit-scrollbar-thumb:hover {
    background-color: #555;
}

/* Rest of your existing CSS */

.container {
    display: flex;
    flex-wrap: wrap;
    width: 80vw;
    margin: 0px auto;
}

/* Rest of your existing CSS */

h1 {
    color: #333;
    text-align: center;
    margin-top: 0px;
}

h3 {
    color: #555;
}

p {
    margin: 0;
}

.user-queries-table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
}
```

```

}

.user-queries-table th, .user-queries-table td {
    border: 1px solid #ddd;
    padding: 8px;
    text-align: left;
}

.user-queries-table th {
    background-color: #f2f2f2;
}

a {
    color: #007bff;
    text-decoration: none;
    display: inline-block;
    margin-top: 10px;
}

a:hover {
    text-decoration: underline;
}

footer {
    position: fixed;
    bottom: 0;
    width: 100%;
    background: linear-gradient(to top, rgba(255, 255, 255, 0.9) 10%,
    rgba(255, 255, 255, 0) 100%);
    text-align: center;
    padding: 10px 0;
    height: 10vh;
    box-shadow: 0px -5px 15px rgba(0, 0, 0, 0.1); /* Optional: Add a
subtle shadow */
}

/* Adjust the link style as needed */
footer a {
    color: #007bff;
    text-decoration: none;
    font-size: 2vh;
}

```

## Wallet.php

```
<!-- this page controlls money related transactions -->
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "123456";
$dbname = "v_rides";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Check if the user is logged in; if not, redirect to the login page
if (!isset($_SESSION["vit_registration_number"])) {
    header("Location: login.php");
    exit;
}

// Fetch user information along with wallet balance
$vit_registration_number = $_SESSION["vit_registration_number"];
$user_info_sql = "SELECT *, w.wallet_balance FROM user_info u
    LEFT JOIN wallet w ON u.vit_registration_number =
w.vit_registration_number
    WHERE u.vit_registration_number =
'$vit_registration_number'";
$user_info_result = $conn->query($user_info_sql);

if ($user_info_result->num_rows == 1) {
    $user_info = $user_info_result->fetch_assoc();
} else {
    echo "User information not found.";
    exit;
}

// Fetch user's wallet balance
$wallet_balance_sql = "SELECT wallet_balance FROM wallet WHERE
vit_registration_number = '$vit_registration_number'";
$wallet_balance_result = $conn->query($wallet_balance_sql);
```

```

if ($wallet_balance_result->num_rows == 1) {
    $wallet_balance_row = $wallet_balance_result->fetch_assoc();
    $wallet_balance = $wallet_balance_row["wallet_balance"];
} else {
    // Set default wallet balance if not found and insert a new wallet
    entry
    $wallet_balance = 50;

    $insert_wallet_sql = "INSERT INTO wallet (vit_registration_number,
wallet_balance) VALUES ('$vit_registration_number',
'$wallet_balance')";
    if ($conn->query($insert_wallet_sql) !== TRUE) {
        echo "Error inserting wallet entry: " . $conn->error;
        exit;
    }
}

// Handle adding money to wallet
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $amount_to_add = $_POST["amount"];

    // Update wallet balance
    $new_balance = $wallet_balance + $amount_to_add;

    // Update the wallet_balance in the wallet table
    $update_wallet_sql = "UPDATE wallet SET wallet_balance =
'$new_balance' WHERE vit_registration_number =
'$vit_registration_number'";
    if ($conn->query($update_wallet_sql) === TRUE) {
        // Redirect to wallet page with updated balance
        header("Location: wallet.php");
        exit;
    } else {
        echo "Error updating wallet balance: " . $conn->error;
        exit;
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Wallet</title>
    <link rel="stylesheet" type="text/css" href="css/wallet.css">
    <!-- Additional styles or external CSS can be linked here -->

```

```
</head>
<body>
  <nav class="navbar">
    <div class="navbar-container">
      <div class="logo">
        
      </div>
    </div>
  </nav>
  <main>
    <div class="container">
      <h1>Your Wallet</h1>
      
      <div class="user-info flex">
        <p>Name: <?php echo $user_info["name"]; ?></p>
      </div>

      <div class="wallet-balance flex">
        <p>Total Balance: <span><?php echo $wallet_balance;
?></span> tokens</p>
      </div>

      <div class="add-money-section flex">
        <h3>Add Money to Your Wallet:</h3>
        <form method="post" action="<?php echo
$_SERVER['PHP_SELF']; ?>">
          <label>Amount (in tokens):</label>
          <input type="number" name="amount" required>
          <input type="submit" value="Add Money">
        </form>
      </div>

      <div class="back-to-dashboard flex">
        <a href="user_dashboard.php">Back to Dashboard</a>
      </div>
    </div>
  </main>
</body>
</html>
```

## Wallet.css

```
/* Wallet CSS */
@import
url('https://fonts.googleapis.com/css2?family=Tilt+Neon&display=swap');

*{
  font-family: 'Tilt Neon', sans-serif;
}

body {
  margin: 0;
  padding: 0;
  background: #1b1b33;
  background-size: cover;
  background-repeat: no-repeat;
}

/* Navigation Bar */
.navbar {
  display: flex;
  background-color: #29294d;
  justify-content: space-between;
  align-items: center;
  padding-right: 20px;
}

.navbar-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  width: 100%;
  margin: 0 auto;
}

.navbar .logo img{
  width: 80px;
}

.container {
  max-width: 600px;
  margin: 50px auto;
  background-color: #fff;
  border-radius: 8px;
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
  padding: 20px;
}
```

```
    text-align: center;
}

.container img{
    width: 200px;
}

h2, h3 {
    color: #333;
}

.user-info p {
    color: #555;
    margin-bottom: 10px;
}

.wallet-balance p {
    color: #333;
    font-size: 18px;
    margin-bottom: 20px;
}

.wallet-balance span {
    font-weight: bold;
    color: rgb(255, 154, 93);
}

.add-money-section h3 {
    color: #333;
    margin-bottom: 15px;
}

form {
    margin-top: 20px;
}

label {
    display: block;
    margin-bottom: 5px;
    color: #333;
}

input[type="number"] {
    padding: 10px;
    width: 100%;
    box-sizing: border-box;
```

```
margin-bottom: 15px;
border: 1px solid #ccc;
border-radius: 4px;
}

input[type="submit"] {
  background-color: rgb(255, 154, 93);
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  margin-bottom: 10px;
}

input[type="submit"]:hover {
  background-color: rgb(255, 170, 99);
}

.back-to-dashboard a {
  color: #3498db;
  text-decoration: none;
}

.back-to-dashboard a:hover {
  text-decoration: underline;
}
```



## REFERENCES

Websites :

1. <https://www.zomato.com/india>
2. <https://www.swiggy.com/>
3. <https://driveeasy.com/>
4. <https://www.emerald.com/insight/content/doi/10.1108/EEMCS-06-2021-0208/full/html>
5. <https://vogo.in/>

Books :

1. HTML and CSS: Design and Build Websites, by Jon Duckett
2. The Joy of PHP Programming: By Alan Forbes
3. PHP & MySQL: The Missing Manual