CityWheels

Unlock Your city with CityWheels

A PROJECT SUBMITTED TO

Atmiya University

Department of Computer Science

RAJKOT

Submitted in partial fulfillment of the requirements for the degree of

"B.Sc. Information Technology"

Sem-5

(Year 2023-2024)

Submitted By:-

Isha Dave

Aasthaba Jadeja

Guided By:-

(Dr) Malay Solanki

(Dr) Madhuri Barchha

PROJECT PROFILE

Project Title	"CityWheels"
Organization	Atmiya University-Rajkot
Front-End Tools	Microsoft edge, Google crome, VS Code
Back-End Tools	Xampp
Language	HTML,PHP
Platform Used	Windows
Developed By	Isha Ashwinbhai Dave Aasthaba Nirmalsinh Jadeja
Project Guide	1) (Dr.)Malay Solanki 2) (Dr.)Madhuri Barchha

STUDENT PROFILE

Name	Isha Ashwinbhai Dave
Program	BSC IT
Enrolment No	210802027
Subject Code	21BITCC506
Class	F2
Batch	Y
Roll No	42
Project Title	CityWheels
Address	Atmiya University
Mobile	9925465144
Email Id	Ishadave2706@gmail.com

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all of the individuals who have helped me to successfully complete this college project.

First and foremost, I would like to thank my project advisor (Dr) Malay Solanki and (Dr) Madhuri Barchha for their guidance, support, and encouragement throughout the project. Their expertise and insights have been invaluable to me.

I would also like to thank my professors Divyesh Gohel and Priyank Doshi for sharing their knowledge and expertise in the subject matter, which helped me to shape my ideas and concepts.

I am very grateful for this golden opportunity to work on this wonderful project on the topic of Cab booking system. Our mentor support us and helped us out whenever we got stuck and find the difficulty. I came to know a lot about the system and what need to be done if we want to sustain in it.

DECLARATION

I, hereby declare that the project work entitled **“CityWheels”** is the original work done by me, and I further declare that it is never submitted anywhere else in part or in full.

Signature

“Isha Dave”

210802027

ABSTRACT

As we know that today's world is all about the digitalization. And still many people is struggling for catching the cab to travel from one city to another and it is hard to find the easy cab booking website. To solve this problem we came up with the online cab booking website which is very easy for the users to use and help the people to catch the cab. They can book their cab online anytime and save their time. Here they can also view their previous rides if any and they can cancel their ride before the time in any emergency.

To solve this problem we came up with the online cab booking system which help people not to struggle for catch the cab. They can book their cab online any time and save their time. Also they can view their booking history and also cancel booking if in any emergency.

Our system enables users to book cabs conveniently through a website. It incorporates various features like feedback mechanisms to ensure a seamless and satisfactory ride for passengers also have passenger ride history.

TABLE OF CONTENTS

Sr. No.	Chapter Title	Page no
CHAPTER 1 : INTRODUCTION		1-2
1.1	Problem Statement	2
1.2	Project Scope	2
1.3	Purpose	2
CHAPTER 2 : REQUIREMENTS AND ANALYSIS		3-8
2.1	System Analysis	4
2.2	Software and Hardware Requirements	7
CHAPTER 3: PROJECT PLANNING AND SCHEDULING		9-13
CHAPTER 4: SYSTEM DESIGN		14-26
4.1	Logical Design	
	4.1.1Used case	15
	4.1.2Data flow diagram (DFD)	18
	4.1.3 ER Diagram	21
	4.1.4 Module Design	23
	4.1.45Data Dictionary	24
CHAPTER 5 :SCREENLAYOUT AND TESTING		27-45
5.1	Screen Layout	
	5.1.1 Admin Side	28
	5.1.2 User Side	34
5.2	Testing Approach	40
CHAPTER 6 : SYSTEM SECURITY AND MEASURES		46-50
CHAPTER 7 : FUTURE SCOPE AND ENHANCEMENT		51-52
CHAPTER 8 : CONCLUSION AND LIMITATIONS		53-54
CHAPTER 9: BIBLOGRAPHY		55-56

Chapter 1: Introduction

1.1 Problem Statement:

In our community, booking reliable and convenient transportation, especially taxis, can often be a challenging and time-consuming task. Passengers face difficulties in finding available cabs, estimating fares, and ensuring their safety during rides. On the other hand, cab drivers may struggle with managing bookings efficiently.

Our cab booking website aims to address the difficulty that people face when trying to find and book a taxi quickly and conveniently. Many costumers struggle with long wait times, unreliable service. Our goal is to create a platform that simplifies the process of booking a cab, minimizes wait times, ensures the safety of passengers for a hassle-free transportation experience.

1.2 Project Scope:

The scope of our Cab Booking System project involves the creation of a simple and user-friendly website for booking cabs. This website will provide passengers with the ability to book a cab by their choice and by specifying pick-up and drop-off locations in the specific tier cities, receive basic fare estimates, and find contact information for inquiries and support.

1.3 Purpose:

The cab booking website is designed to make it easy and convenient for people to find, book, and use taxi services. It aims to save time, provide safety and reliability, and offer a hassle-free way to get from one place to another without the stress of hailing a cab on the street or waiting for one on the curb. In essence, the website is here to simplify and enhance the taxi booking experience for users.

Chapter 2: Requirements And Analysis

2.1 System Analysis:

The technology that we used for your project is PHP. PHP is a powerful and popular scripting language used for building dynamic websites and web applications. It's often embedded within HTML code and runs on the web server to generate dynamic web content. PHP allows you to interact with databases, process user input, and perform various tasks to create interactive and personalized web experiences. It's a fundamental tool for web developers to create websites that can do much more than just display static information, making the web more interactive and functional.

The PHP application codes can be written in any of the following languages:

- HTML
- CSS
- Bootstrap
- JavaScript

We have used Visual Studio 2010 IDE(Integrated Development Environment) to create our whole project. Its main purpose is to serve as a versatile and efficient code editor for software developers and Languages(You can code in C#, VB, F#, C++, HTML, JavaScript, Python, etc.) support.

We have used XAMPP database in your project. XAMPP is a software package that helps you set up a local web server with a database (usually MySQL) for web development and testing. It simplifies the process of creating, managing, and connecting to databases on your local machine.

Front-End Technology :

- HTML
- CSS
- JavaScript
- Bootstrap

Back-End Technology :

- PHP

Features of PHP :

- Easy to Learn: PHP is easy for beginners to pick up because its syntax is similar to other common programming languages.

- Open Source: PHP is free to use and has a large, active community of developers contributing to its improvement.

- Cross-Platform: It works on various operating systems like Windows, Linux, and macOS, making it versatile.

- Server-Side Scripting: PHP runs on the server, allowing you to create dynamic web pages and interact with databases.

- Database Connectivity: PHP can connect to a wide range of databases, making it suitable for data-driven web applications.

Features of CSS :

- Styling: CSS is primarily used to style web content. You can change the colors, fonts, sizes, and other visual aspects of your website elements.

- Selectors: You can select specific elements on a web page (like headings, paragraphs, or links) using CSS selectors. Once selected, you can apply styles to them.

- Cascading: The "C" in CSS stands for "Cascading." It means that styles can be inherited or overridden. You can define a style for a parent element, and it will affect its children, but children can also have their own styles that override the parent.

- Media Queries: You can use CSS to apply different styles based on the device's characteristics, such as screen width or device orientation.

- Animations and Transitions: CSS allows you to add animations and smooth transitions to elements, making your website more interactive and engaging.

- Flexibility: CSS is highly customizable. You can create complex layouts, position elements precisely, and even create complex shapes and effects.

- Compatibility: CSS is supported by all modern web browsers, ensuring that your styles will be displayed consistently to your users.

Features of HTML :

- Compatibility: HTML is supported by all web browsers, ensuring that your web pages can be viewed consistently by users.

- Accessibility: HTML provides features for making your web content accessible to people with disabilities, such as using alt attributes for images and providing semantic markup for screen readers.

- Comments: You can add comments in HTML to provide notes and explanations for yourself or other developers working on the code.

- Nesting: HTML elements can be nested within each other to create complex structures and layouts for your web page.

- Modularity: HTML can be divided into reusable components, allowing you to maintain and update your website more efficiently

- Structure: HTML provides the basic structure for web pages. It allows you to define the different parts of a webpage, such as headings, paragraphs, lists, and more.

- Text and Multimedia: You can use HTML to display text content and embed multimedia elements like images, videos, and audio on your web page.

2.2 Software And Hardware Requirements :

Hardware Requirement :

Processor	Intel core i3 Processor
Main Memory	2 GB RAM
Hard Disk	4 GB

Software Requirement :

- **Microsoft edge :** Microsoft Edge is a web browser created by Microsoft, similar to Google Chrome or Mozilla Firefox. It's the program you use to explore the internet, view websites, and search for information. It's designed to be fast, secure, and user-friendly. Microsoft Edge also comes with useful features like a built-in password manager and a dark mode for comfortable nighttime browsing. It's an excellent choice for people who use Windows computers because it works well with them**.**

- **VS Code :** Visual Studio Code (VS Code) is a popular, free, open-source code editor developed by Microsoft. It's known for its lightweight yet powerful features, making it a favorite among developers. With a vast extension ecosystem, integrated Git support, and a user-friendly interface, VS Code is an excellent choice for coding in multiple languages and frameworks.

- **Google Crome :** Google Chrome is a web browser created by Google. It's a computer program that you use to access and explore the internet. Chrome lets you visit websites, search for information, watch videos, and more. It's known for being fast, secure, and user-friendly. You can also add extensions to customize your browsing experience. Many people use Google Chrome as their preferred way to browse the web because of its speed and features. It's available for free, and you can download it to your computer or mobile device.

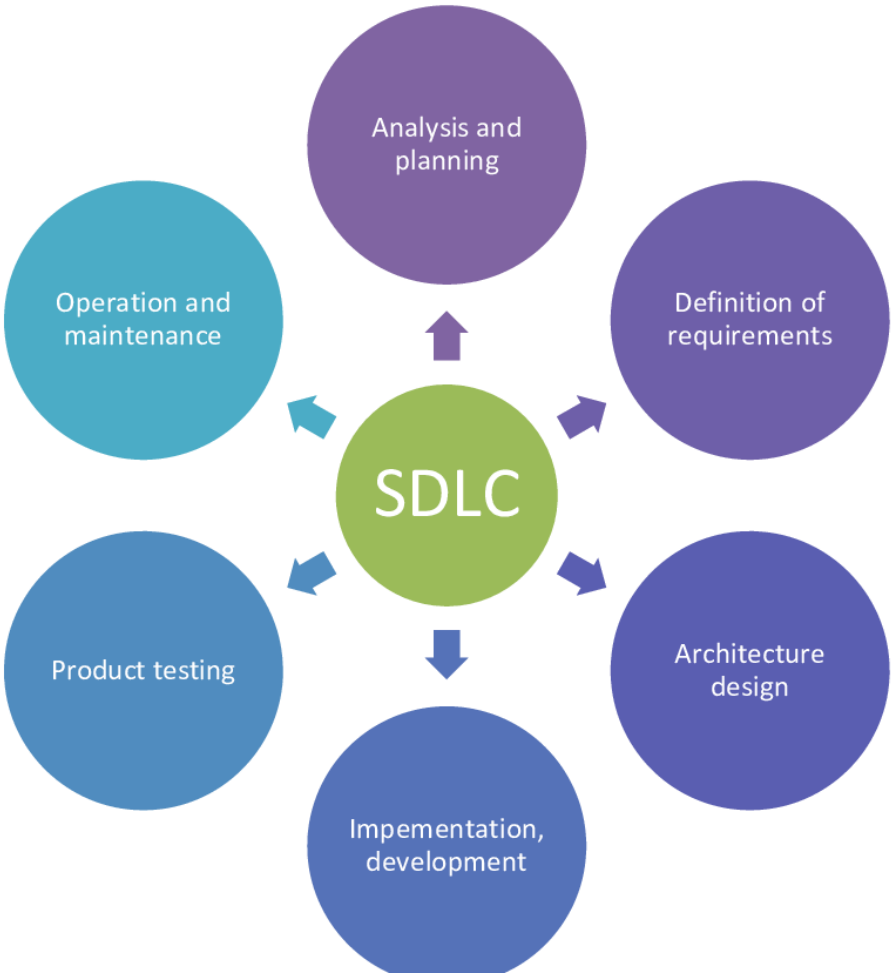
- **Xampp :** XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

Operating System	Windows 7
Front-End	Microsoft edge, Google crome
Back-End	XAMPP 3.3.0
Tools	Microsoft Visual Studio code 1.82.2

Chapter 3: Project Planning and Scheduling

Project Planning And Scheduling:

SDLC Life Cycle:



- **Requirements Gathering and Analysis**:
    - In this phase, the project team works closely with stakeholders to gather and document all the project requirements. These requirements serve as the foundation for the rest of the SDLC process.
- **System Design**:
    - Based on the gathered requirements, the system design phase involves creating a high-level architectural design of the software. This includes defining system components, interfaces, and data flow.
- **Implementation (Coding)**:
    - During the implementation phase, developers write the actual code for the software based on the design specifications. This is where the software is built.
- **Testing**:
    - In the testing phase, the software undergoes rigorous testing to identify and fix defects and ensure that it meets the specified requirements. Testing can include unit testing, integration testing, system testing, and user acceptance testing (UAT).
- **Deployment**:
    - Once the software is thoroughly tested and deemed ready for production use, it is deployed to the target environment, which could be a server, a cloud platform, or end-users' devices.
- **Maintenance and Support**:
    - After deployment, the software enters the maintenance phase. This phase involves ongoing support, bug fixes, updates, and enhancements to ensure the software remains functional and up-to-date.

Phase 1: Requirements Gathering and Analysis

- Duration: 2 week

Activities :

- Initiate project, gather initial requirements, and set up development environment.
- Research and collect sample templates for reference.
- Analyse the features and functionalities of existing websites or software related to our project.
- Compile a list of features and gather necessary information.
- Develop necessary diagrams (e.g., system architecture, database schema).
- Brainstorm and create a tagline for our project.
- create a logo for CityWheels project and design this logo with Canva.

Phase 2: Design and Development

- Design the system architecture and database schema.
- Create wireframes and prototypes for the user interface.
- Design the pages of our web application based on wireframes and prototypes.
- Review and refine the page designs.
- Continue working on page designs.
- Conduct user testing for page designs.
- Finalize and submit the page designs.

Phase 3: Implementation and Deployment

- Begin the development of our web application.
- Implement front-end and back-end components.
- Perform debugging and code optimization.
- Complete and submit the code.
- Start documenting the project, including user guides and technical manuals.
- Conduct internal reviews and revisions.

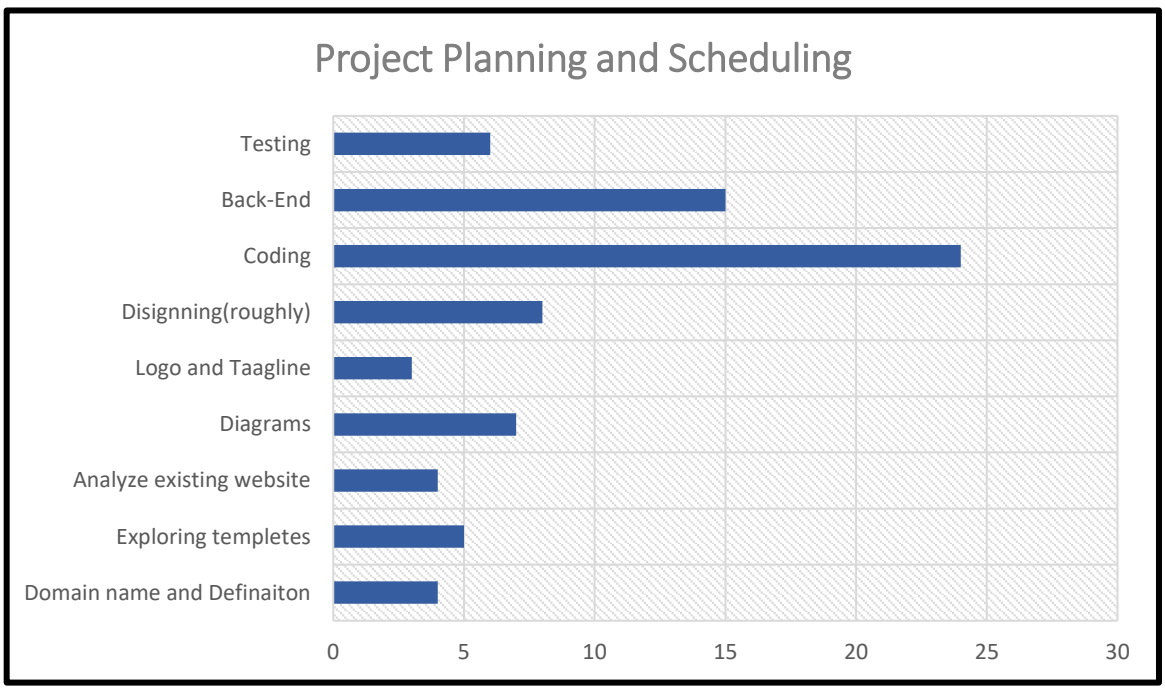
Phase 4: Maintenance and Support

- Duration: Ongoing

  Activities:

  - Monitor system performance and resolve issues.
  - Provide ongoing user support and training.
  - Update system features and functionalities as needed.

- **Bar chart of Planning and scheduling :**



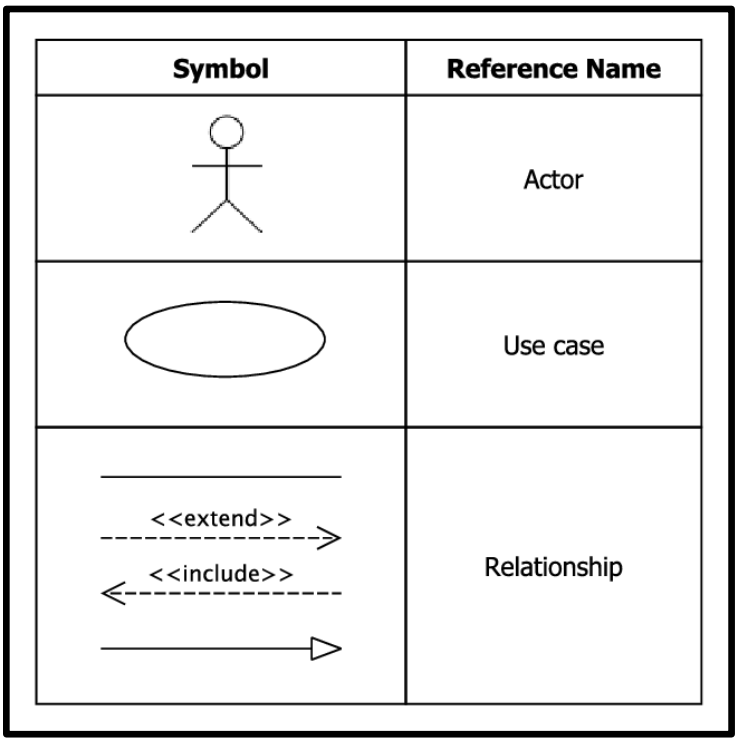
Chapter 4:
System Design

4.1 Logical Design:

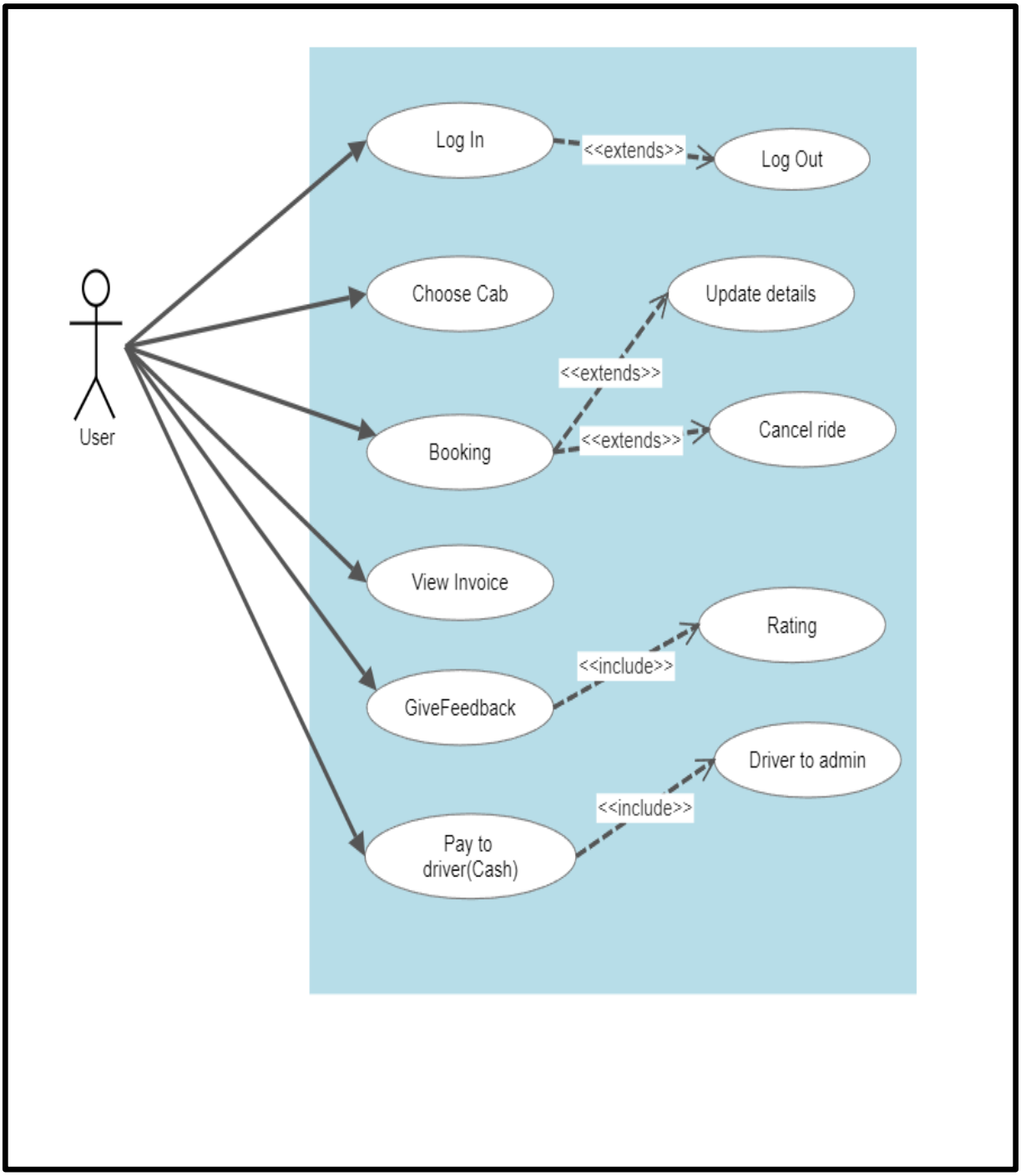
4.1.1 Use Cases:

Use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use case and actors. An actor is represents a user or another system that will interact with the system you are modelling. A use case is an external view of system that represents some action the user might perform in order to complete a task. Use case are helpful in enlightening requirements and planning the project.

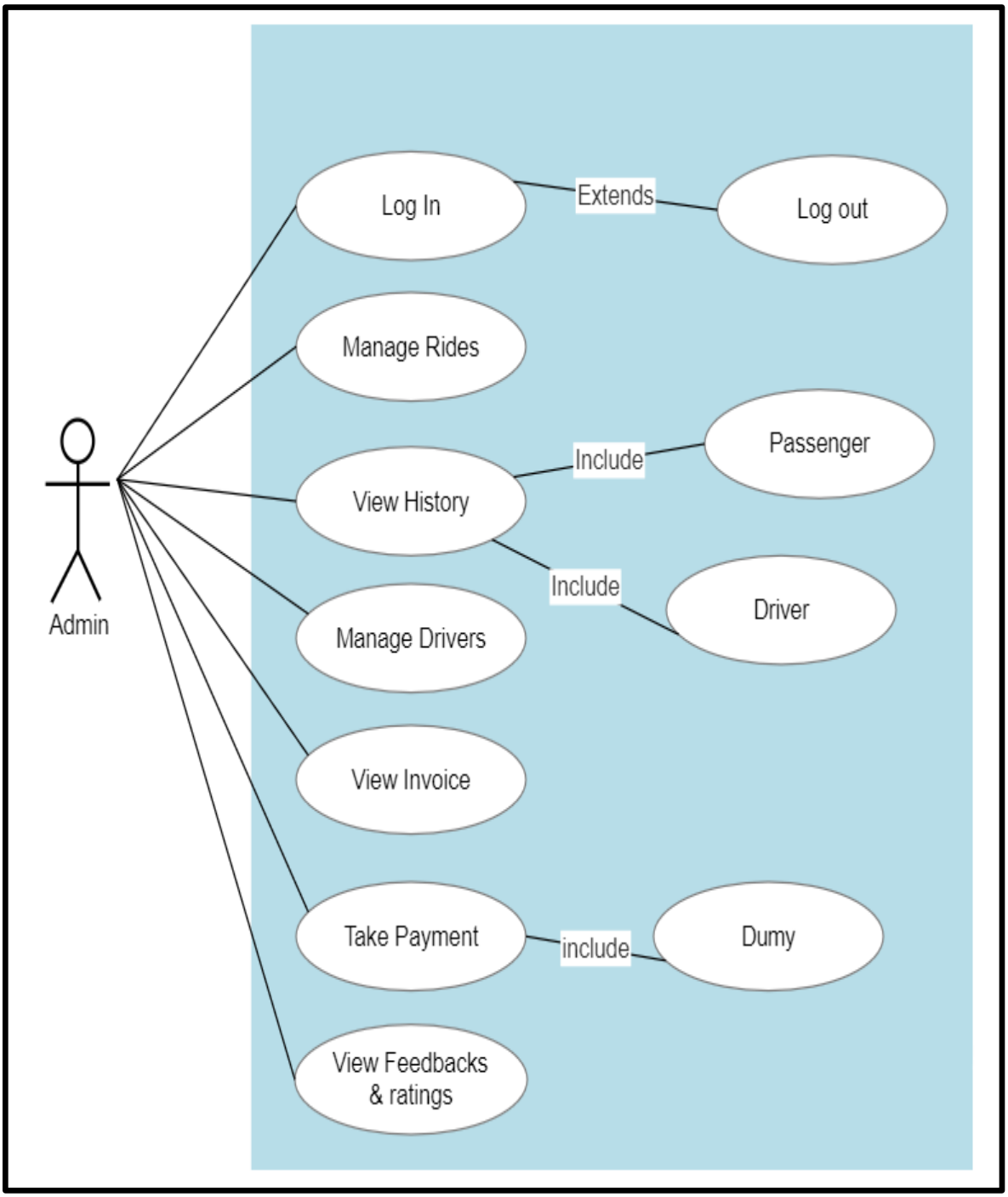
A *use case* is a list of actions or event steps typically defining the interactions between a role (known in the Unifies Modeling Language (UML) as an actor) and a system to achieve a goal. The actor can be a human or another external system. In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.



User:

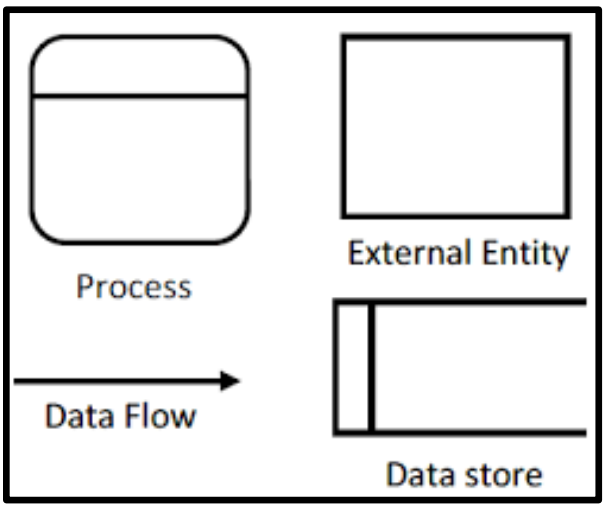

Log In
<<extends>>
Log Out
Choose Cab
Update details
<<extends>>
User
Booking
<<extends>>
Cancel ride
View Invoice
Rating
<<include>>
GiveFeedback
Driver to admin
<<include>>
Pay to
driver(Cash)

Admin:

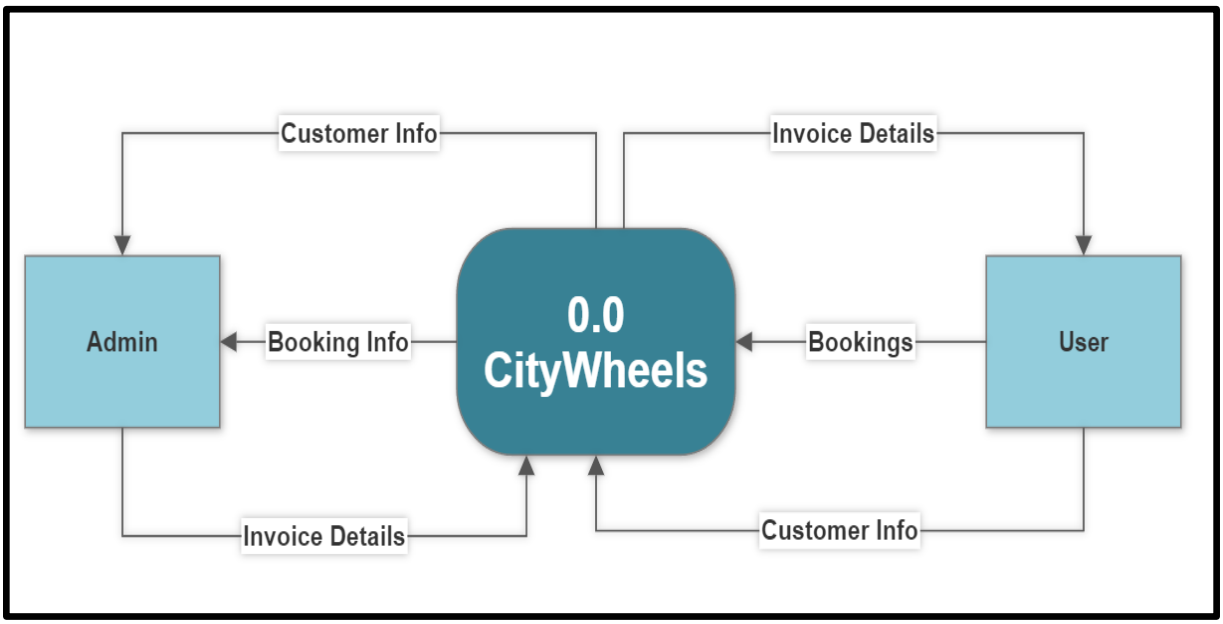

Log In
Extends
Log out
Manage Rides
Passenger
Include
View History
Include
Driver
Manage Drivers
View Invoice
Take Payment
include
Dumy
View Feedbacks
& ratings
Admin

4.1.2 Data Flow Diagram:

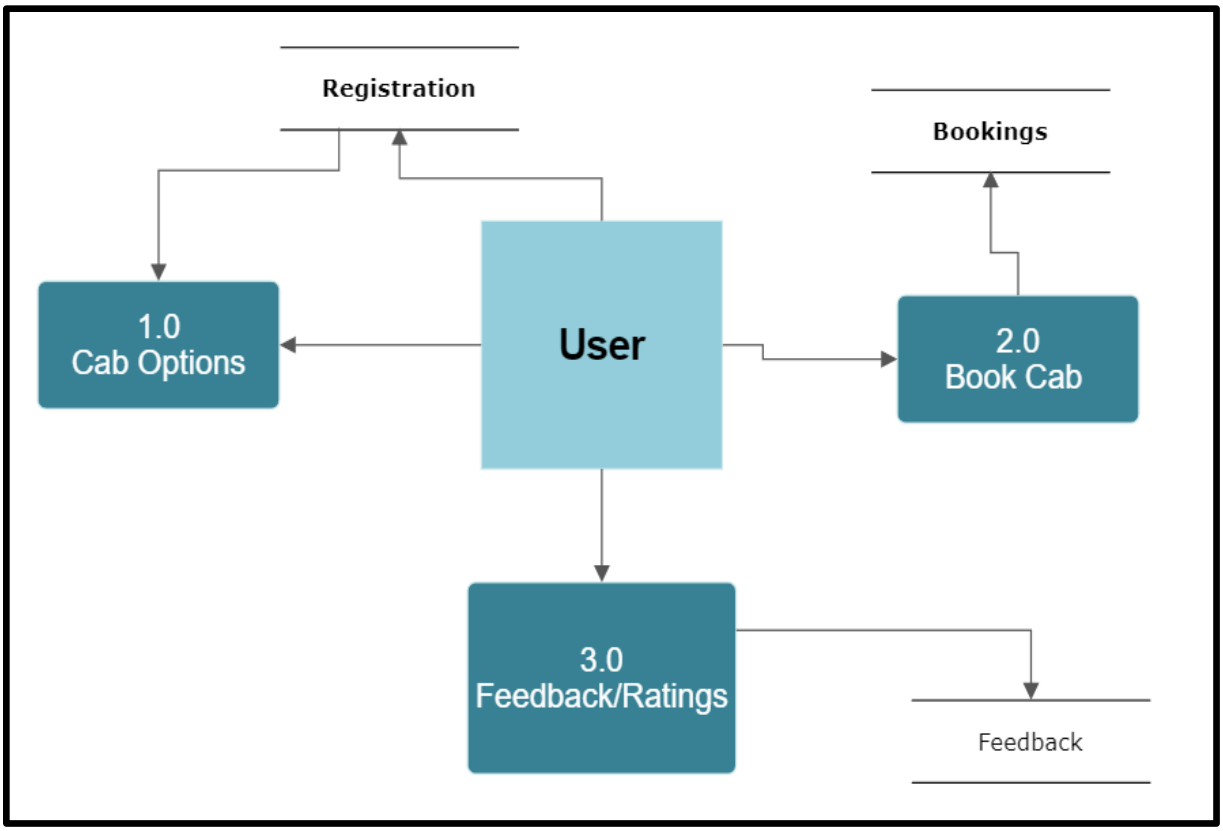
Diagram of flow of data in system and its processing which converts data into valuable information is known as data flow diagram. A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself.



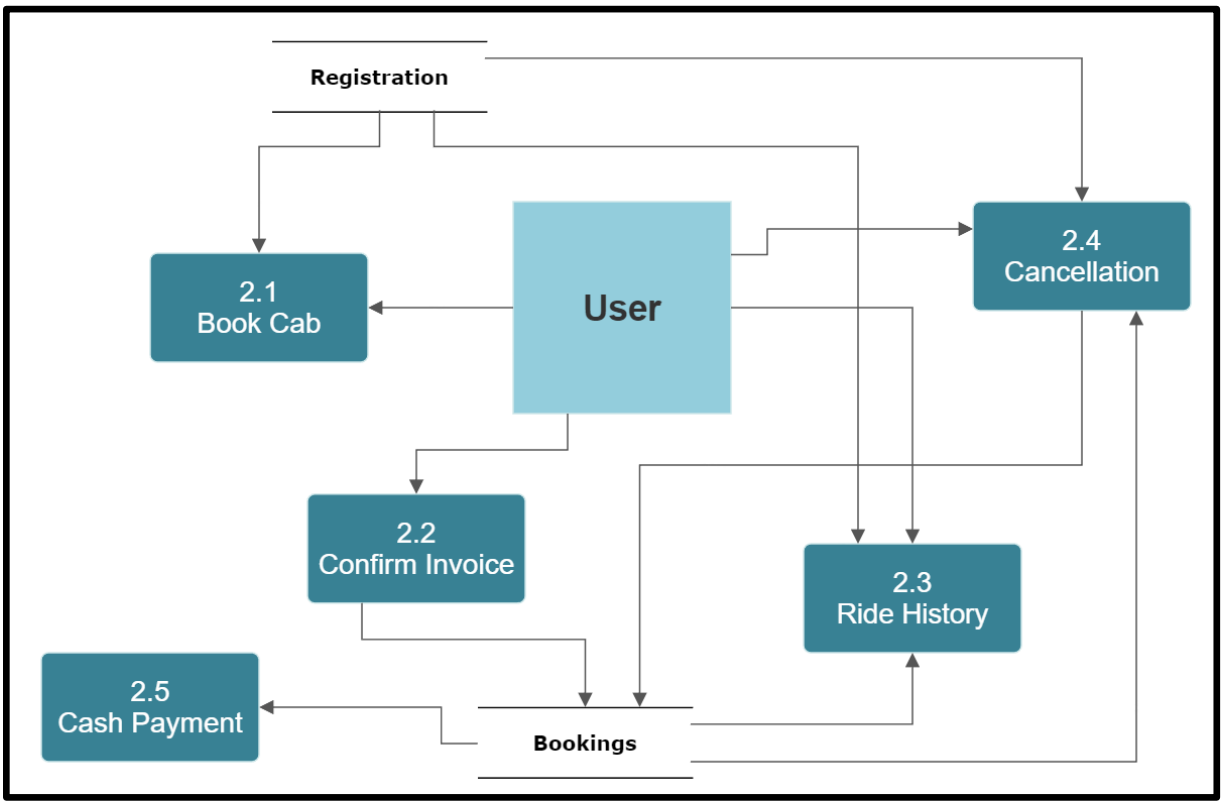
- **0 level :**



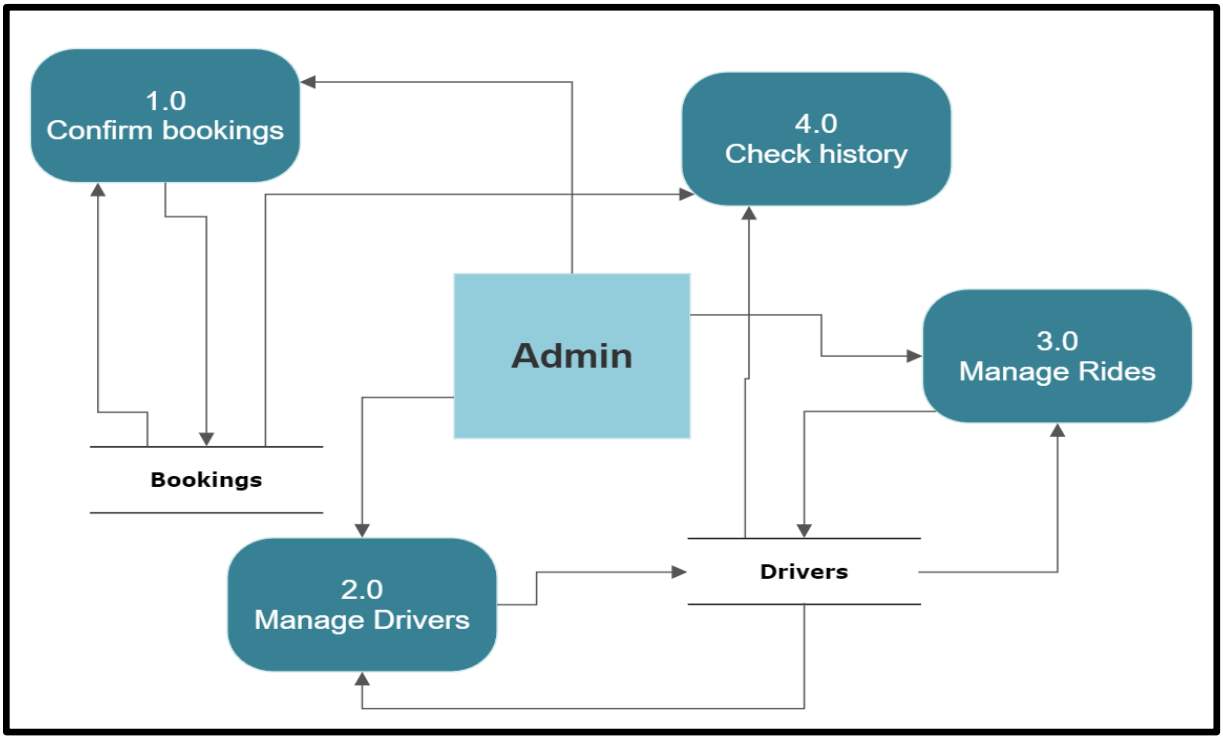
- **User level 1:**



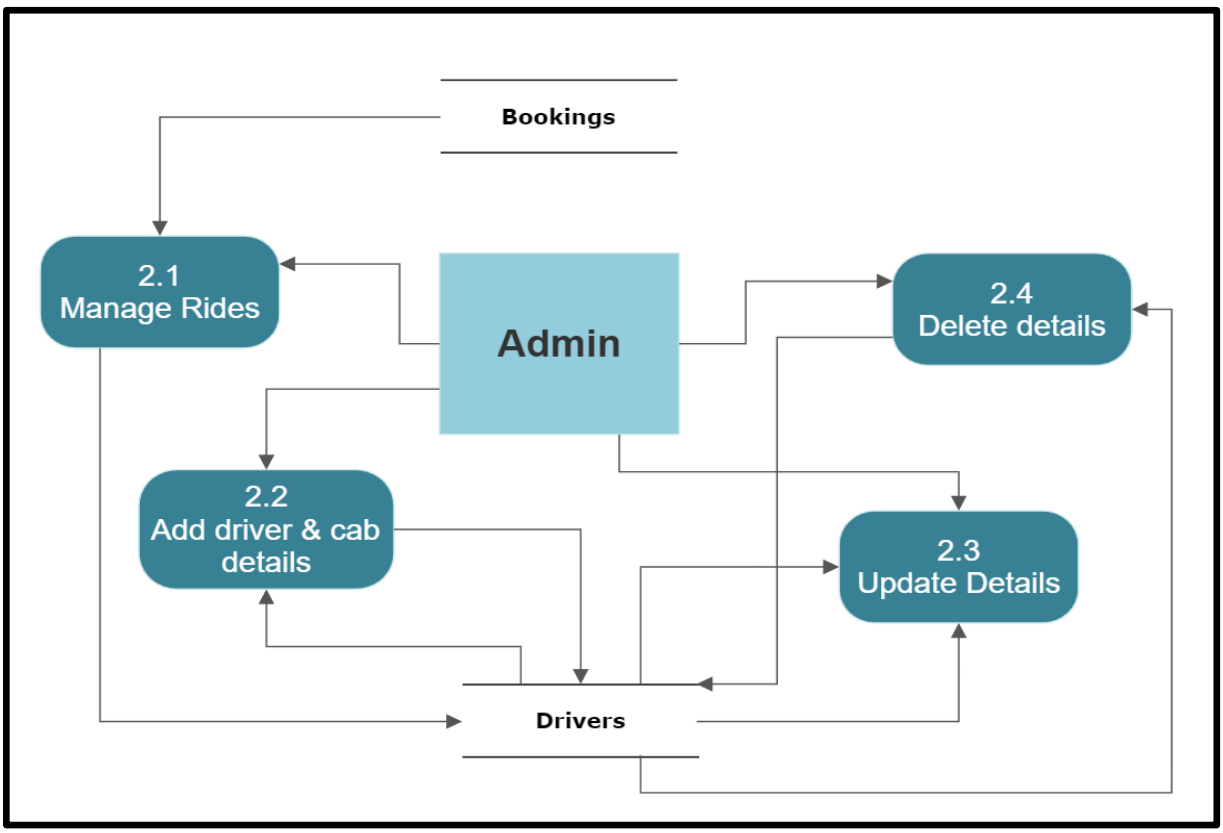
- **User Level 2:**



- **Admin level 1:**



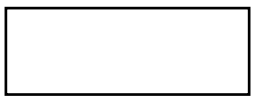
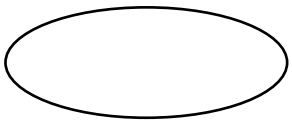
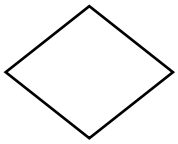
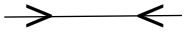
- **Admin level 2 :**

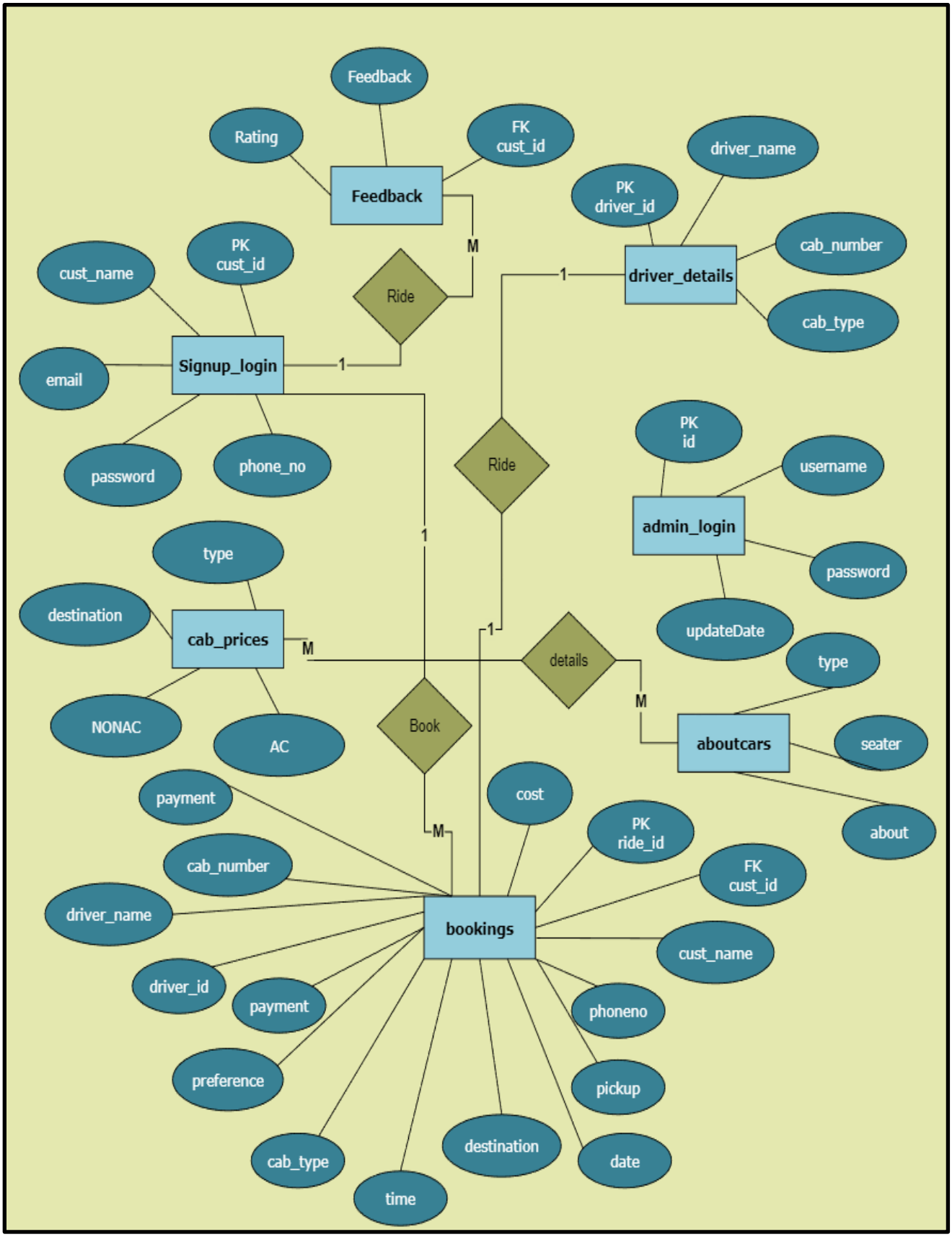


4.1.3 ER Diagram:

The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.

Symbol	Name
	Entity
	Attribute
	Relationship
	One – to One
	Many - to - many
	One - to - many


Feedback
Rating
FK
cust_id
Feedback
M
driver_name
PK
driver_id
driver_details
cab_number
cab_type
1
cust_name
PK
cust_id
Ride
Signup_login
email
1
password
phone_no
Ride
PK
id
username
admin_login
password
updateDate
1
type
destination
cab_prices
M
1
details
type
M
NONAC
AC
Book
aboutcars
seater
about
payment
cost
M
PK
ride_id
cab_number
FK
cust_id
driver_name
bookings
cust_name
driver_id
payment
phoneno
preference
pickup
cab_type
destination
date
time

4.1.4 Module Design :

- User side before login :
  - Home :
    - Cab option
    - About :
      - About us
      - Contact us
      - FAQs
- User side after login :
    - Choose cab
      - Book cab
      - Invoice
      - Conformation
    - History :
      - User ride history
      - cancellation
    - Feedback
    - Logout

- Admin side after login:
- Dashboard:
    - No. of users
    - No of bookings
    - No. of drivers
    - No. of cabs
    - Car details:
    - History:
      - Passenger History
      - Driver History
      - Invoice History
    - Feedbacks
    - Driver:
      - Driver details
      - Add driver
      - Update
      - Delete

4.1.5 Data Dictionary:

sign up_login:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	cust_id	int(4)			No	*None*		AUTO_INCREMENT
☐	2	cust_name	varchar(30)	utf8mb4_general_ci		No	*None*		
☐	3	email	varchar(30)	utf8mb4_general_ci		No	*None*		
☐	4	password	varchar(15)	utf8mb4_general_ci		No	*None*		
☐	5	phone_no	varchar(10)	utf8mb4_general_ci		No	*None*		

feedback:

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	cust_id	int(20)			No	*None*
☐	2	feedback	varchar(100)	utf8mb4_general_ci		No	*None*
☐	3	rating	int(6)			No	*None*

bookings:

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	ride_id	int(15)			No	*None*		AUTO_INCREMENT
☐	2	cust_id	int(15)			No	*None*		
☐	3	cust_name	varchar(30)	utf8mb4_general_ci		No	*None*		
☐	4	phoneno	varchar(10)	utf8mb4_general_ci		No	*None*		
☐	5	date	date			No	*None*		
☐	6	pickup	varchar(100)	utf8mb4_general_ci		No	Rajkot		
☐	7	destination	varchar(100)	utf8mb4_general_ci		No	*None*		
☐	8	time	time			No	*None*		
☐	9	cab_type	varchar(15)	utf8mb4_general_ci		No	*None*		
☐	10	preference	varchar(10)	utf8mb4_general_ci		No	*None*		
☐	11	payment	varchar(10)	utf8mb4_general_ci		No	cash		
☐	12	driver_id	int(20)			No	*None*		
☐	13	driver_name	varchar(50)	utf8mb4_general_ci		No	*None*		
☐	14	cab_number	varchar(50)	utf8mb4_general_ci		No	*None*		
☐	15	cost	int(15)			No	*None*		

Cab_prices :

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	destination	varchar(30)	utf8mb4_general_ci		No	*None*		
☐	2	type	varchar(30)	utf8mb4_general_ci		No	*None*		
☐	3	AC	int(30)			No	*None*		
☐	4	NONAC	int(30)			No	*None*		

admin_login:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(5)			No	*None*		AUTO_INCREMENT
2	username	varchar(15)	utf8mb4_general_ci		No	*None*		
3	password	varchar(10)	utf8mb4_general_ci		No	*None*		
4	updateDate	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()

driver_details:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	driver_id	int(15)			No	*None*		AUTO_INCREMENT
2	driver_name	varchar(30)	utf8mb4_general_ci		No	*None*		
3	cab_number	varchar(15)	utf8mb4_general_ci		No	*None*		
4	cab_type	varchar(30)	utf8mb4_general_ci		No	*None*		

aboutcars :

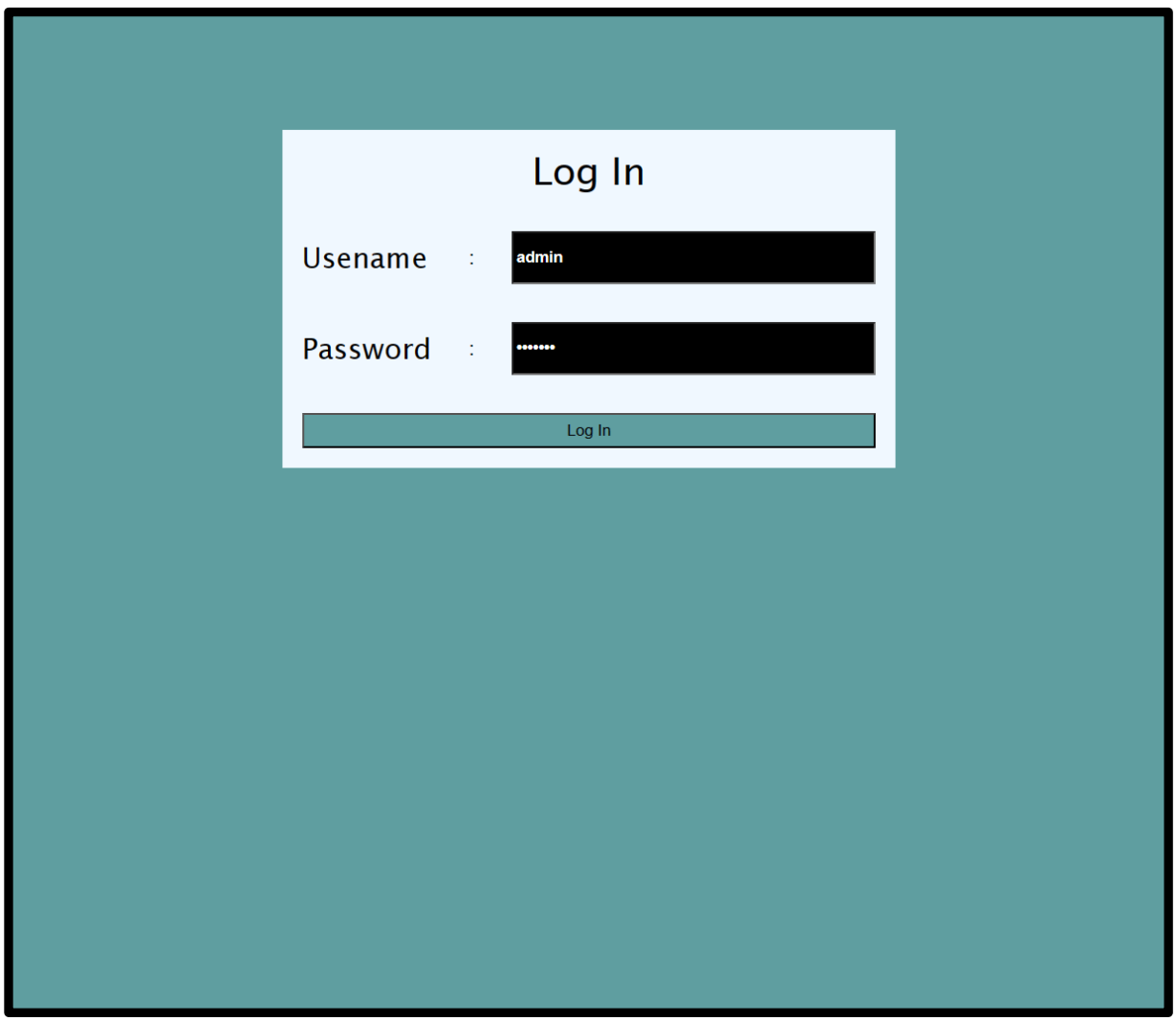
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	type	varchar(30)	utf8mb4_general_ci		No	*None*		
2	seater	int(10)			No	*None*		
3	about	varchar(100)	utf8mb4_general_ci		No	*None*		

Chapter 5: Screen Layout And Testing

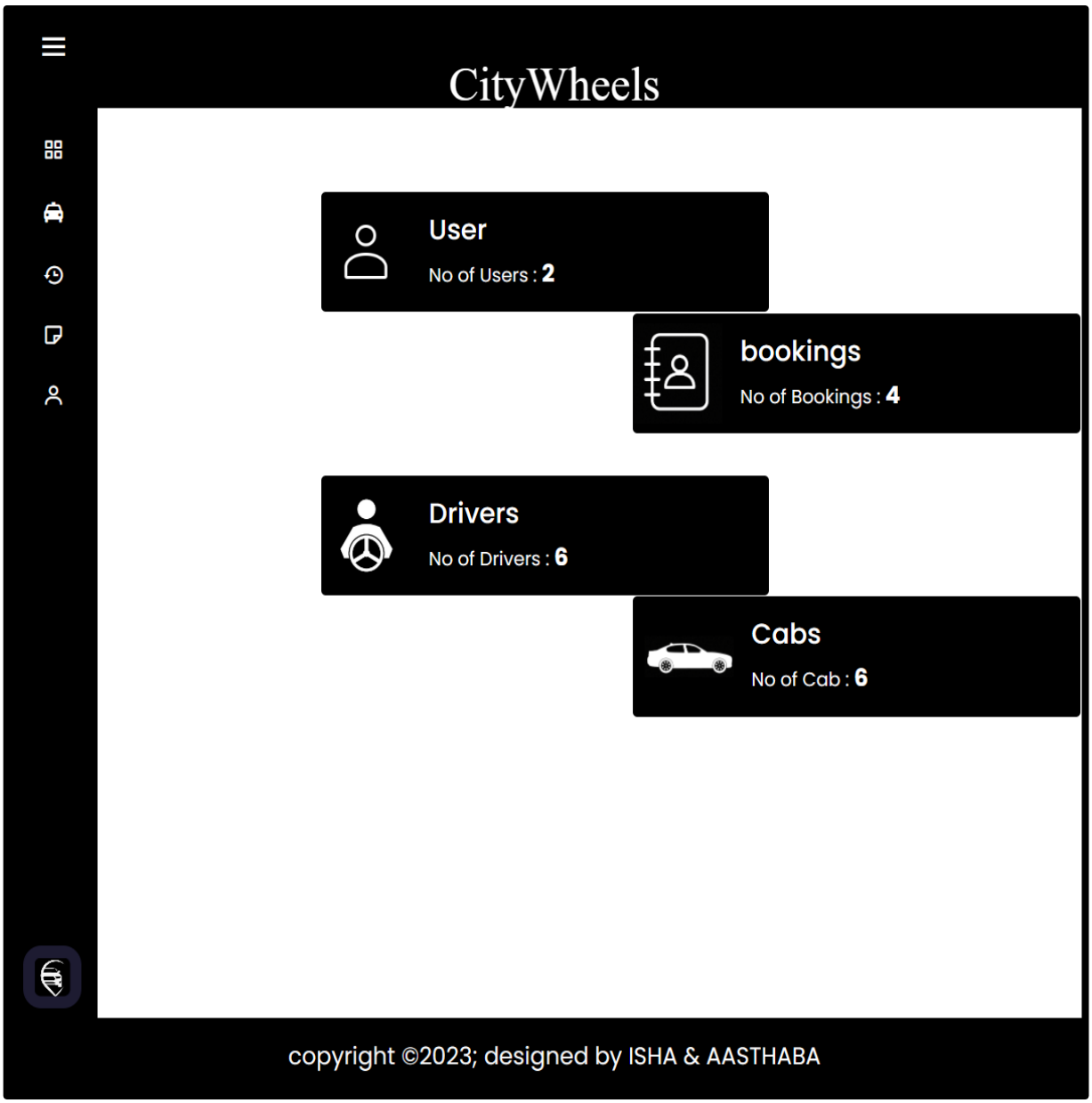
5.1 Screen layout :

5.1.1 Admin Side :

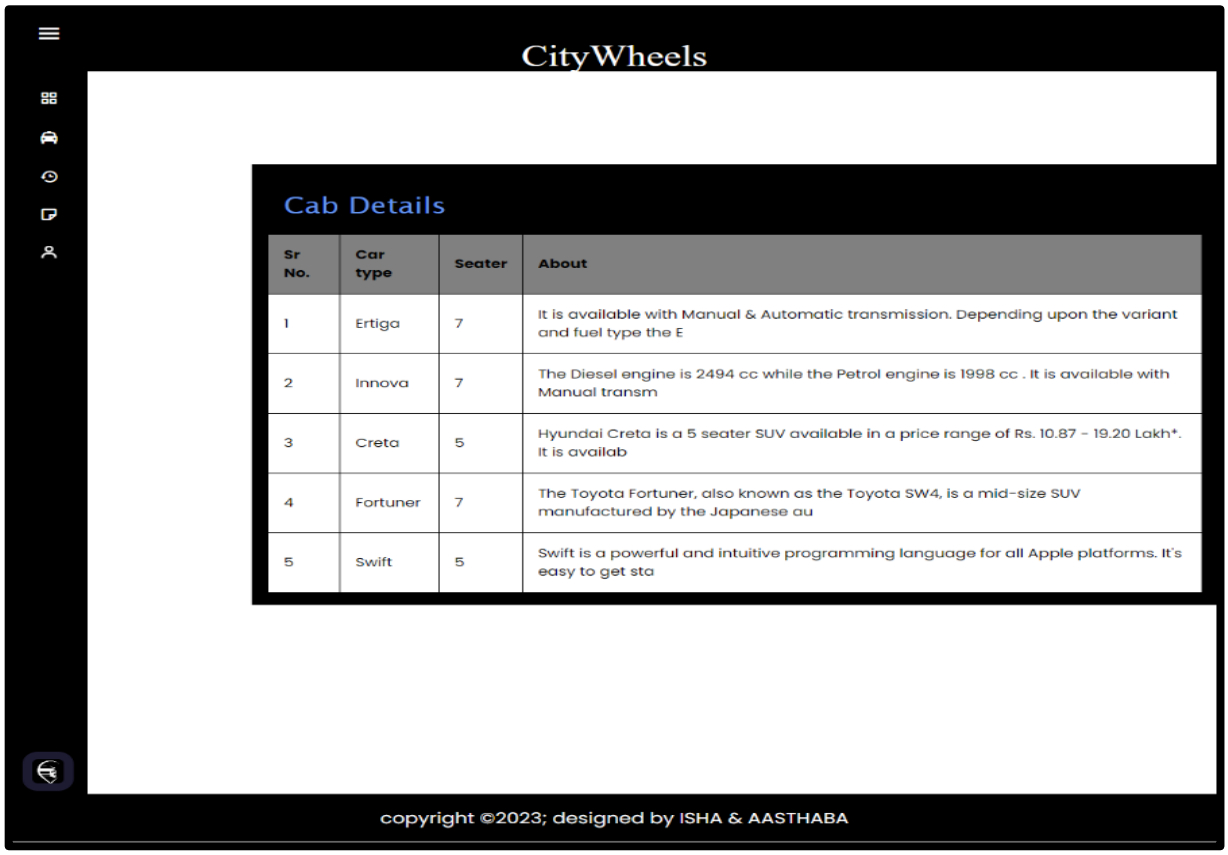
Login Page :



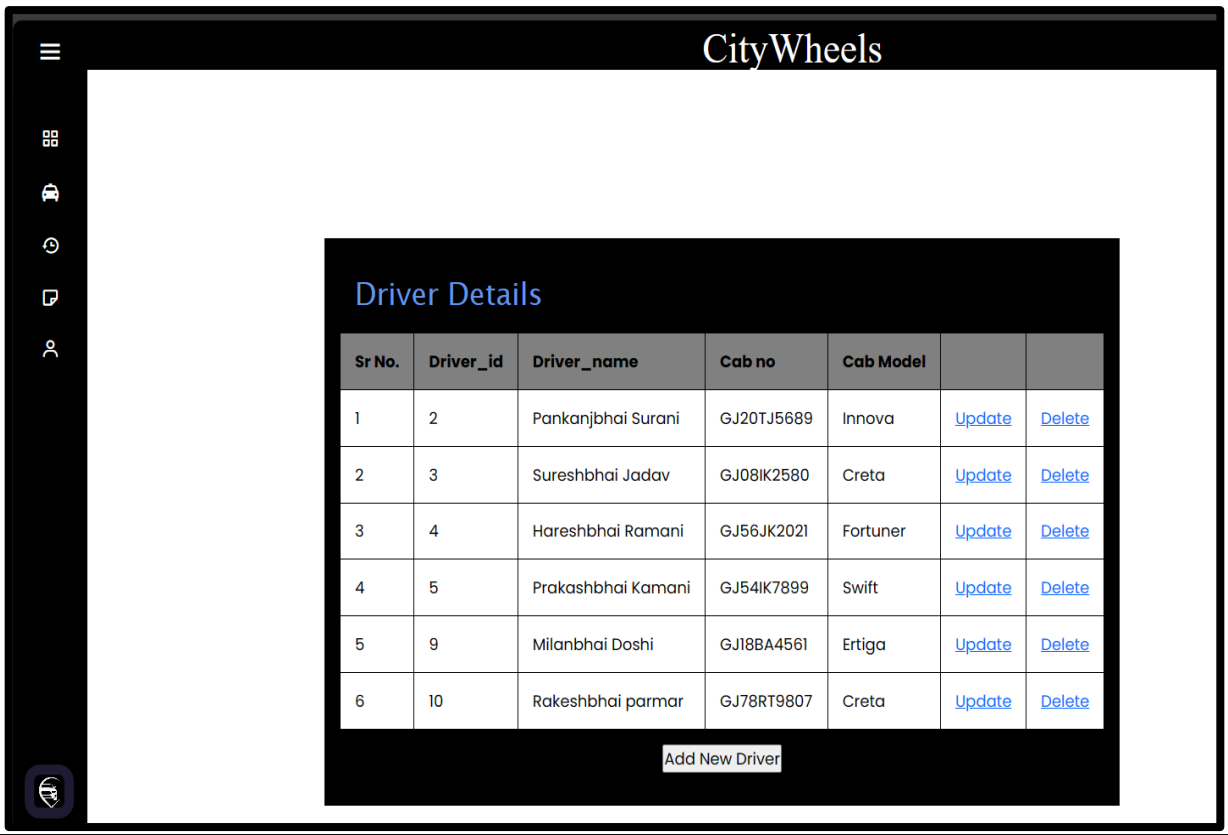
Dashboard :


CityWheels
User
No of Users : 2
bookings
No of Bookings : 4
Drivers
No of Drivers : 6
Cabs
No of Cab : 6
copyright ©2023; designed by ISHA & AASTHABA

Cab details :



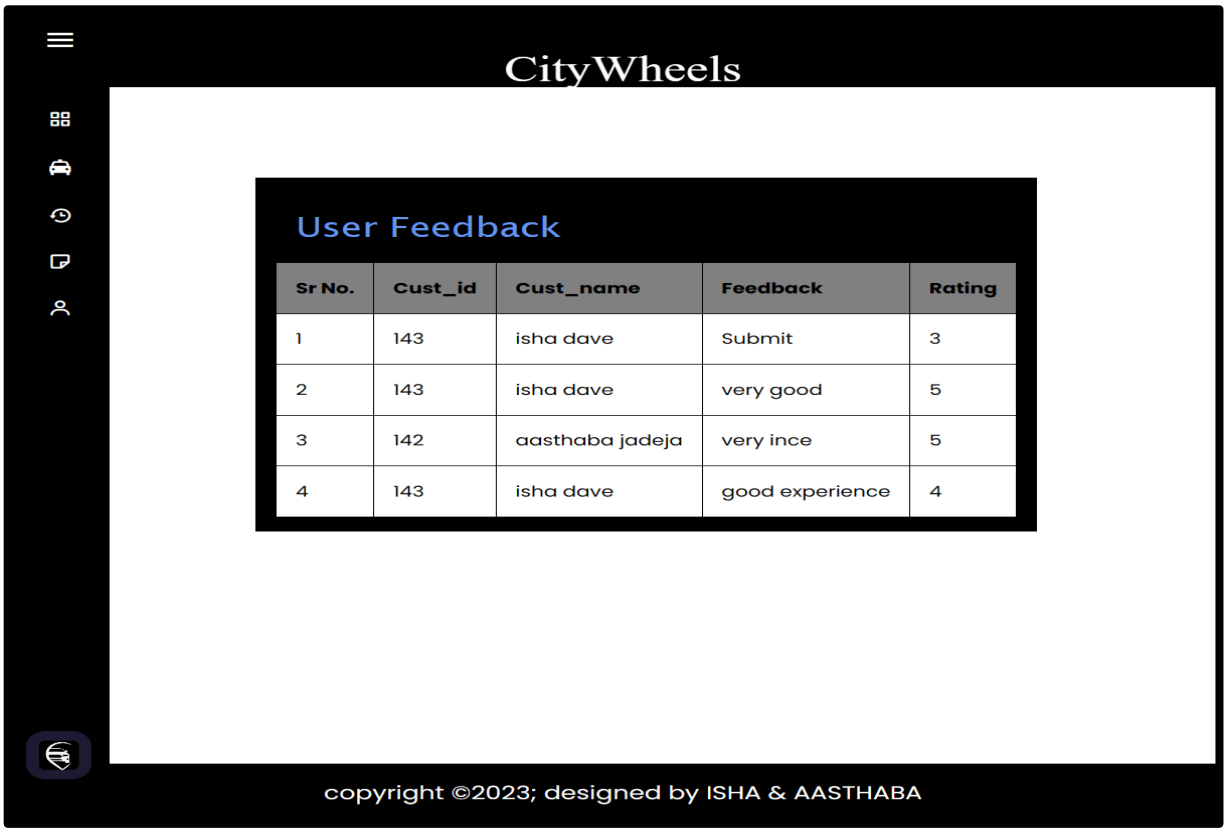
Driver details :



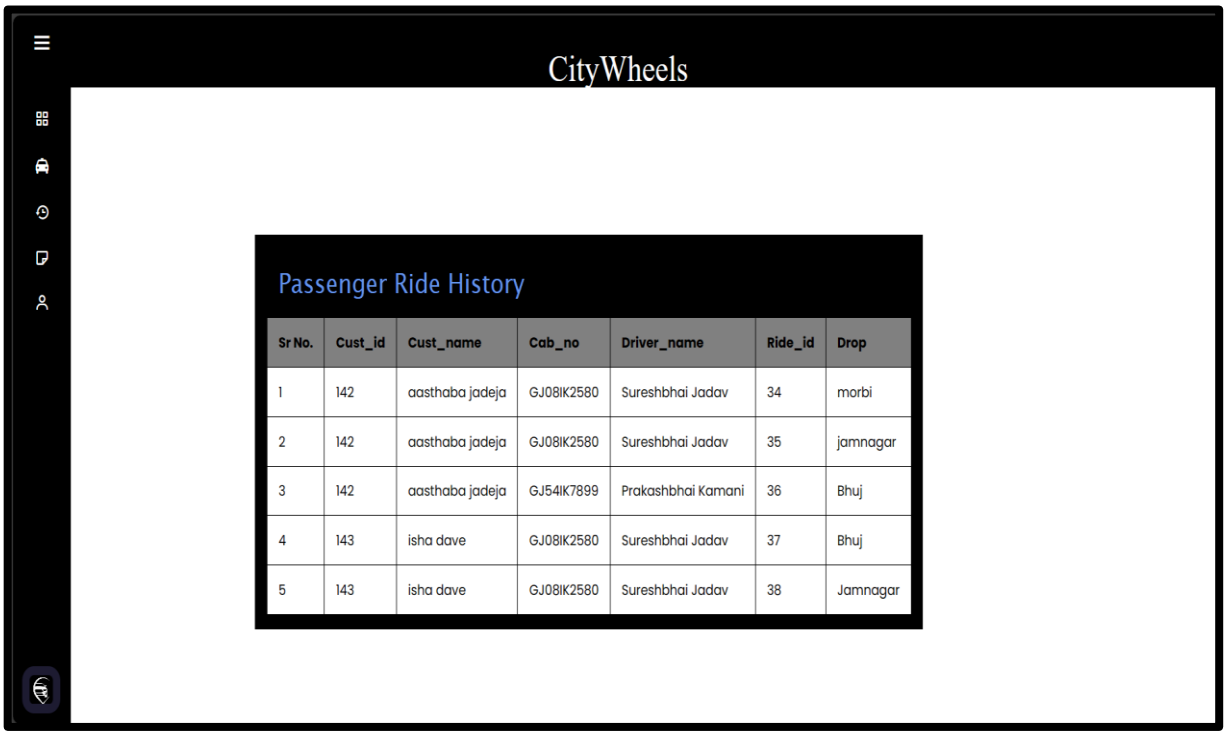
Driver Update:

Add driver :

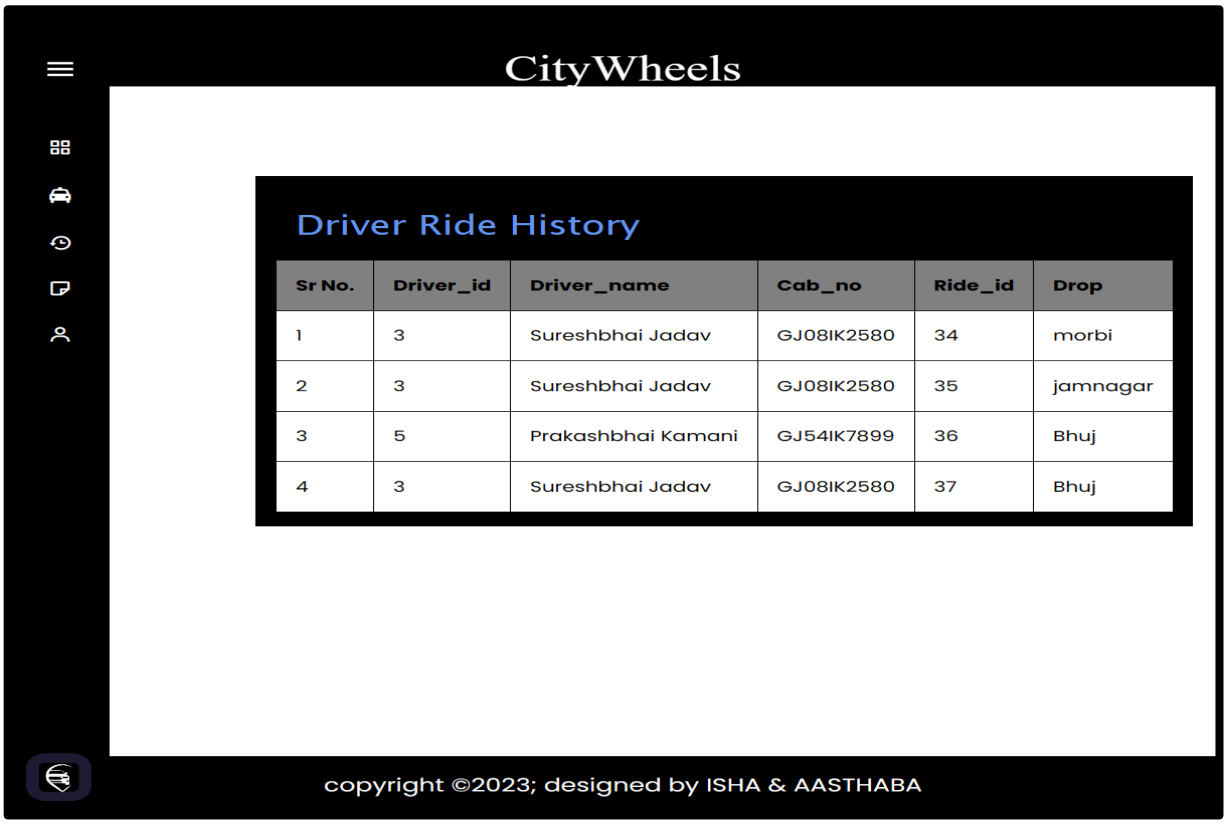
Feedback :



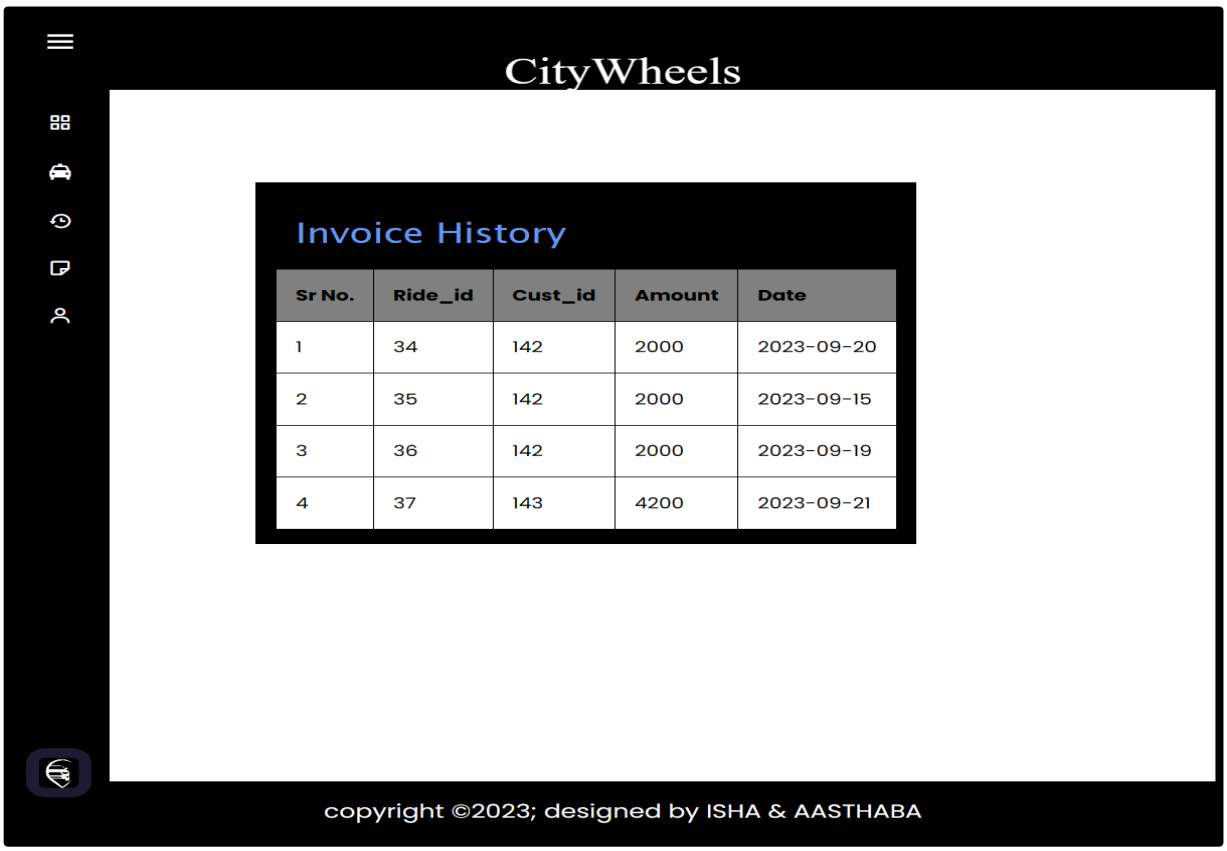
Passenger History:



Driver History:

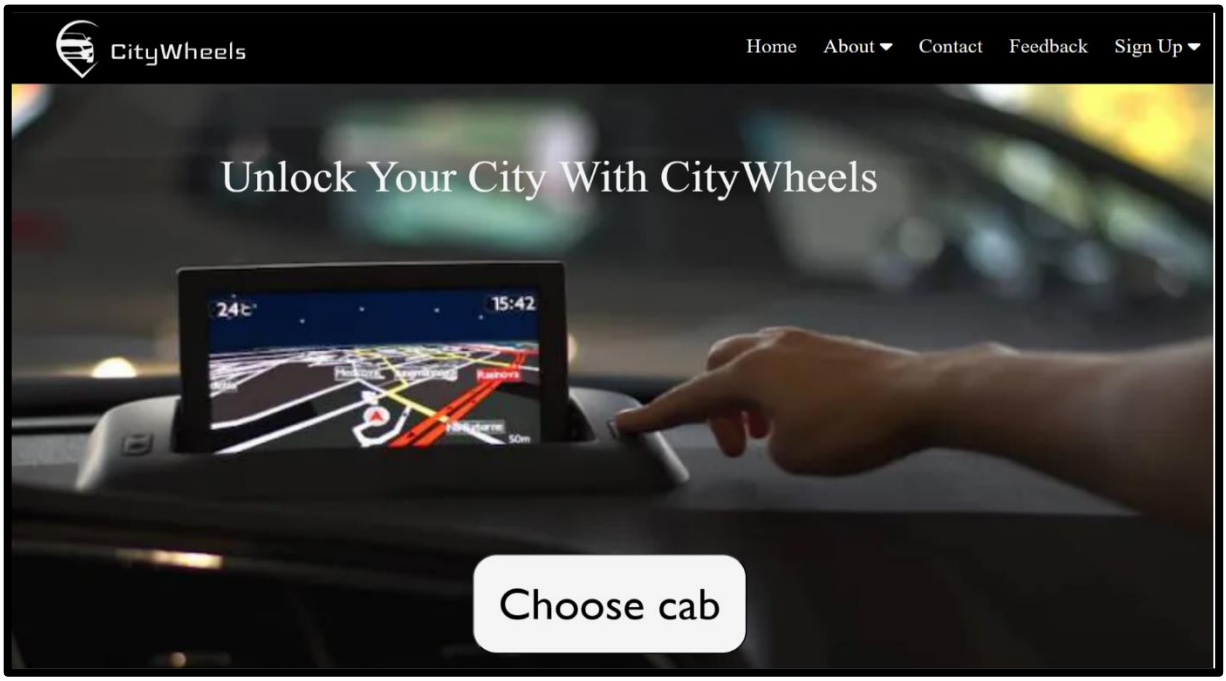


Invoice history:

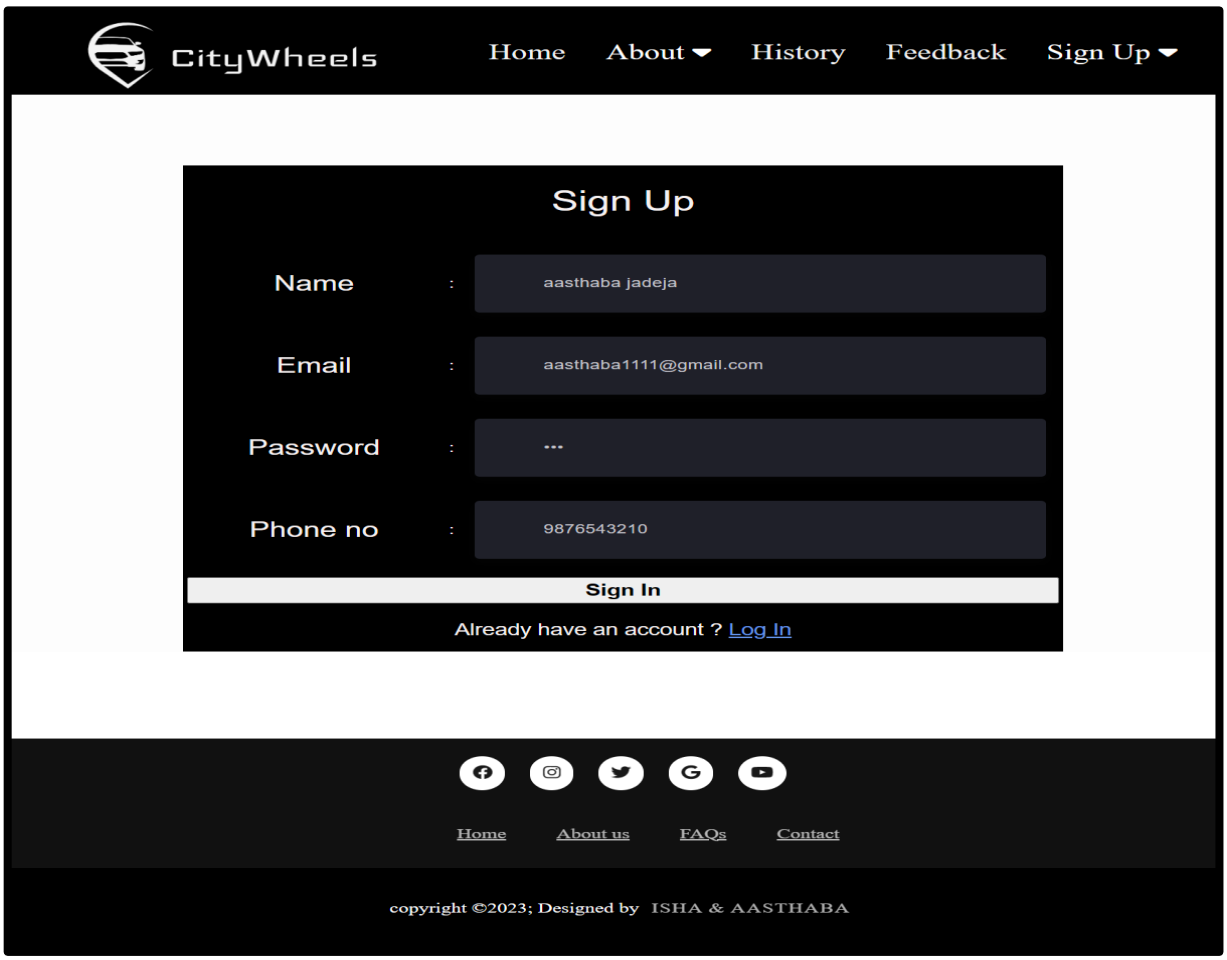


5.1.2 User Side :

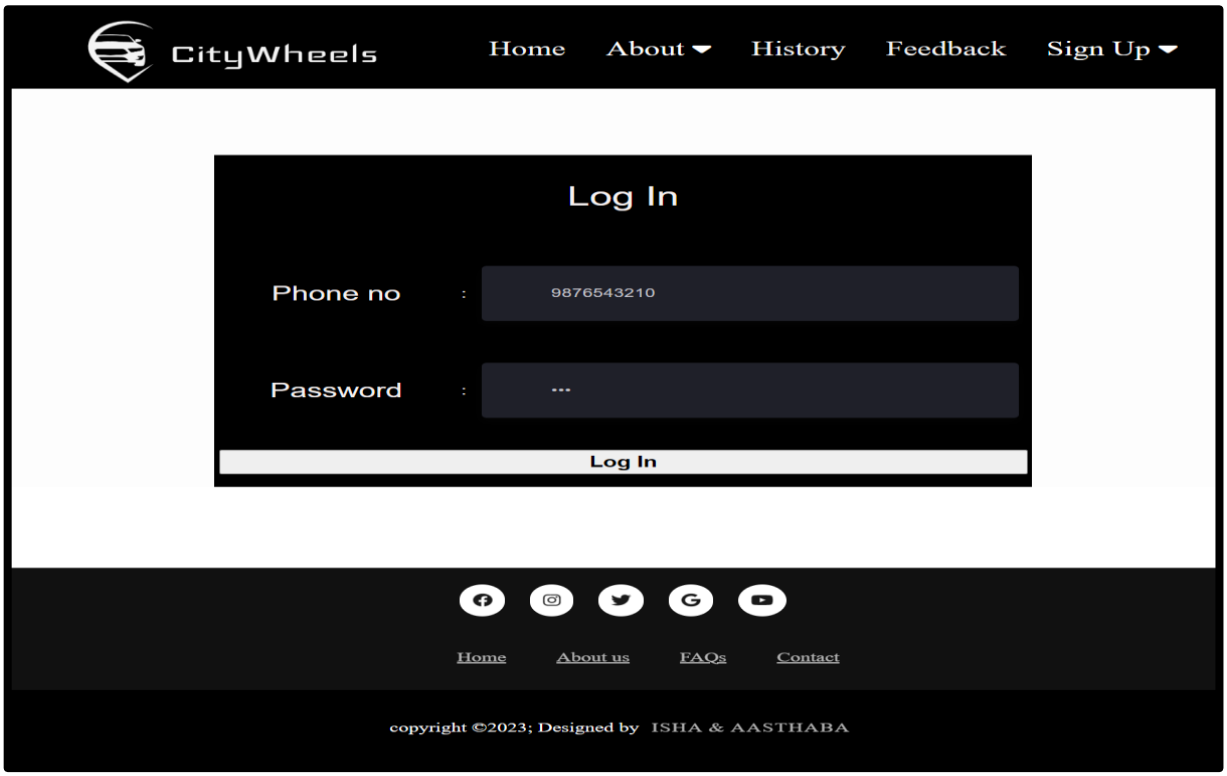
Home:



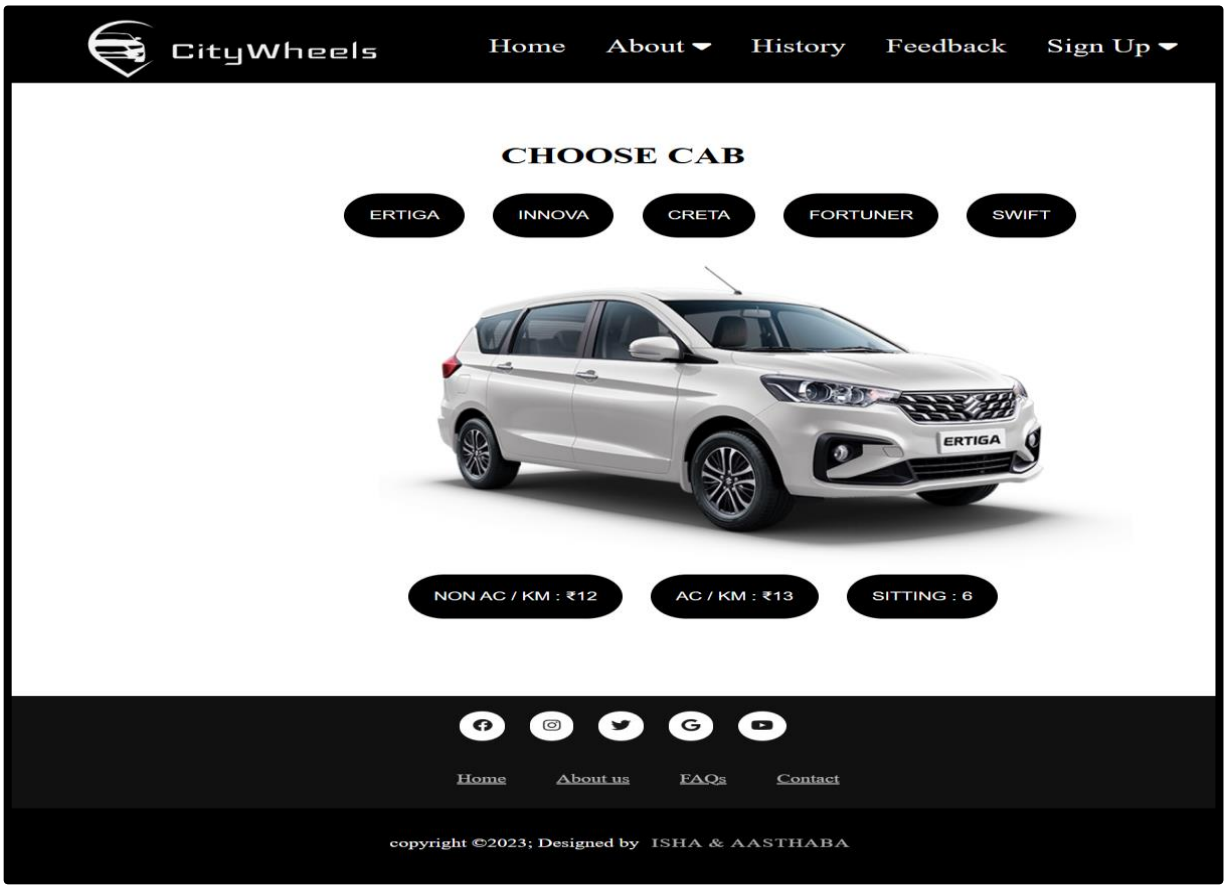
Sign up :



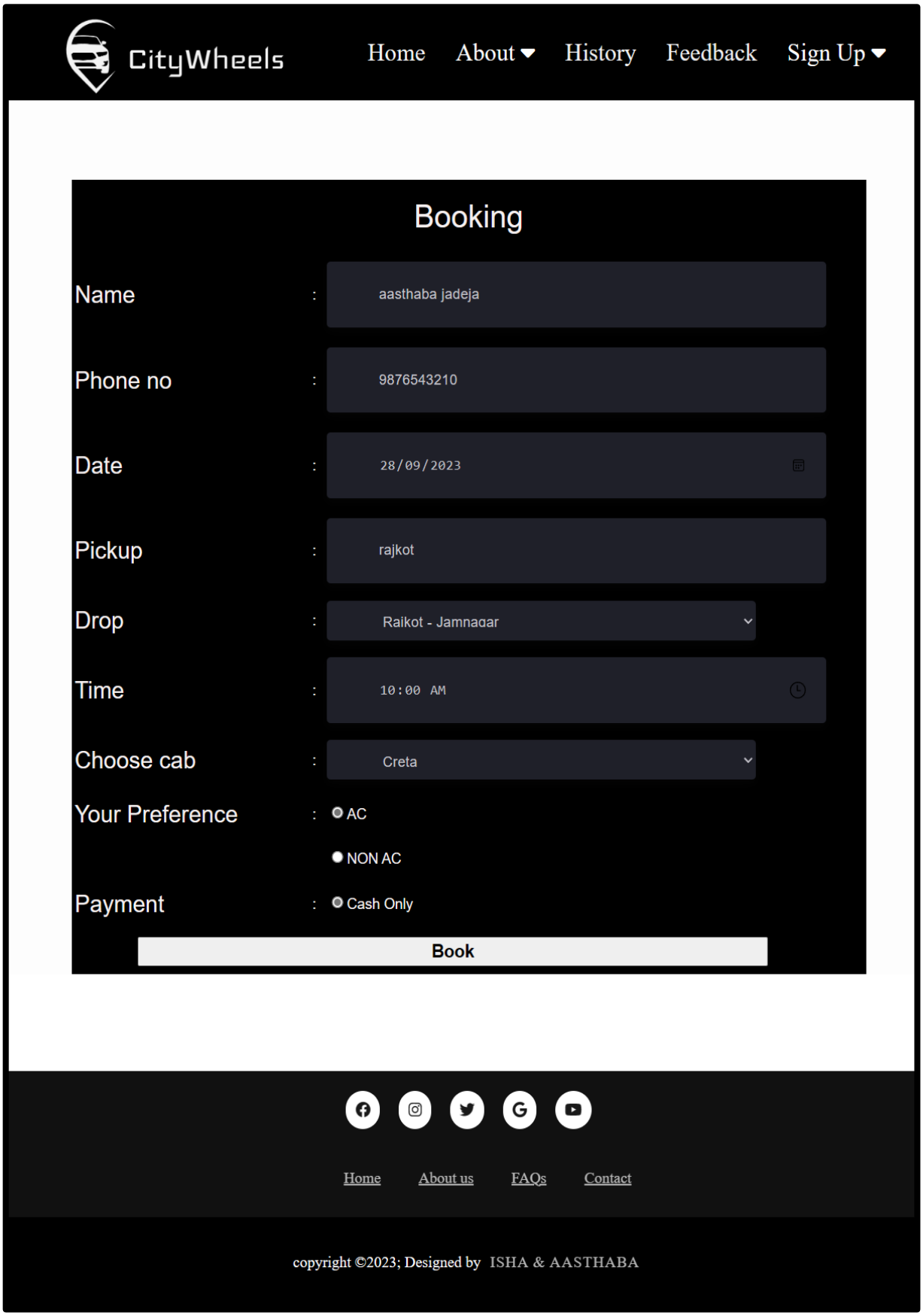
Login:



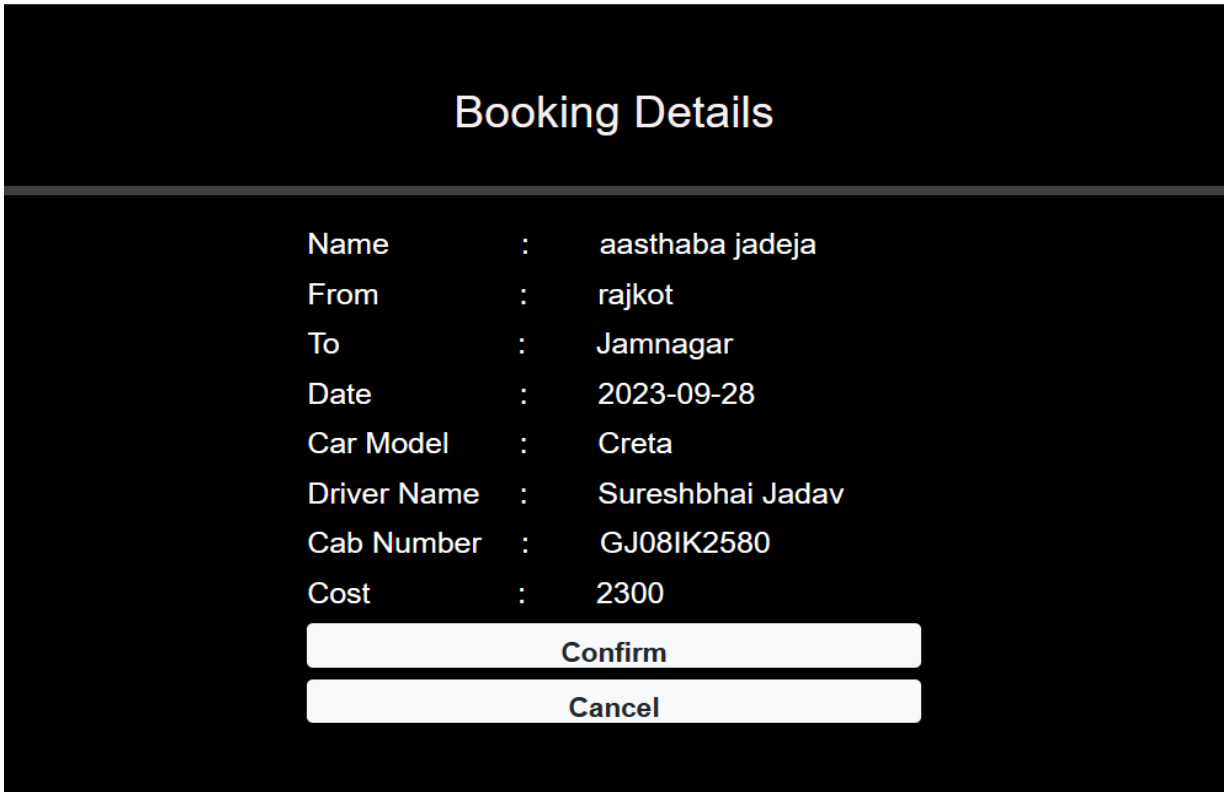
Choose cab:



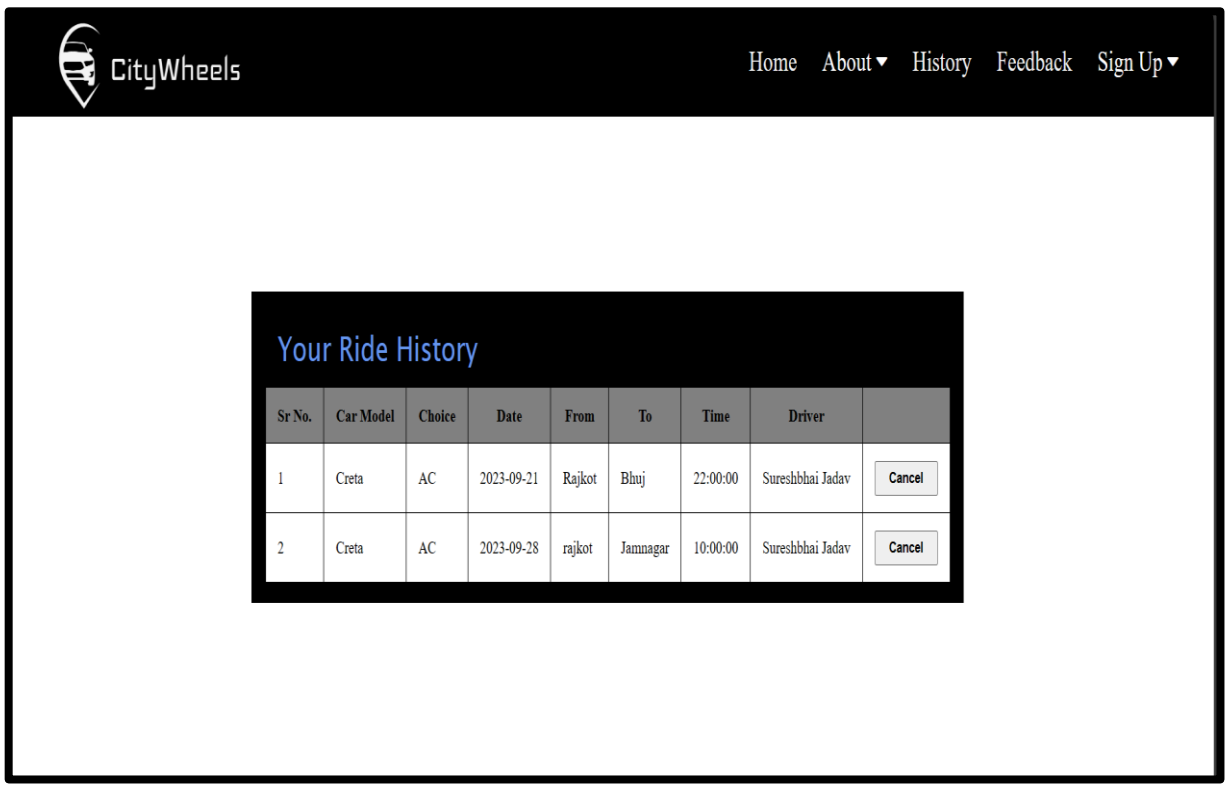
Booking:


CityWheels
Home
About
History
Feedback
Sign Up
Booking
Name : aasthaba jadeja
Phone no : 9876543210
Date : 28/09/2023
Pickup : rajkot
Drop : Raikot - Jamnagar
Time : 10:00 AM
Choose cab : Creta
Your Preference : AC
NON AC
Payment : Cash Only
Book
Home
About us
FAQs
Contact
copyright ©2023; Designed by ISHA & AASTHABA

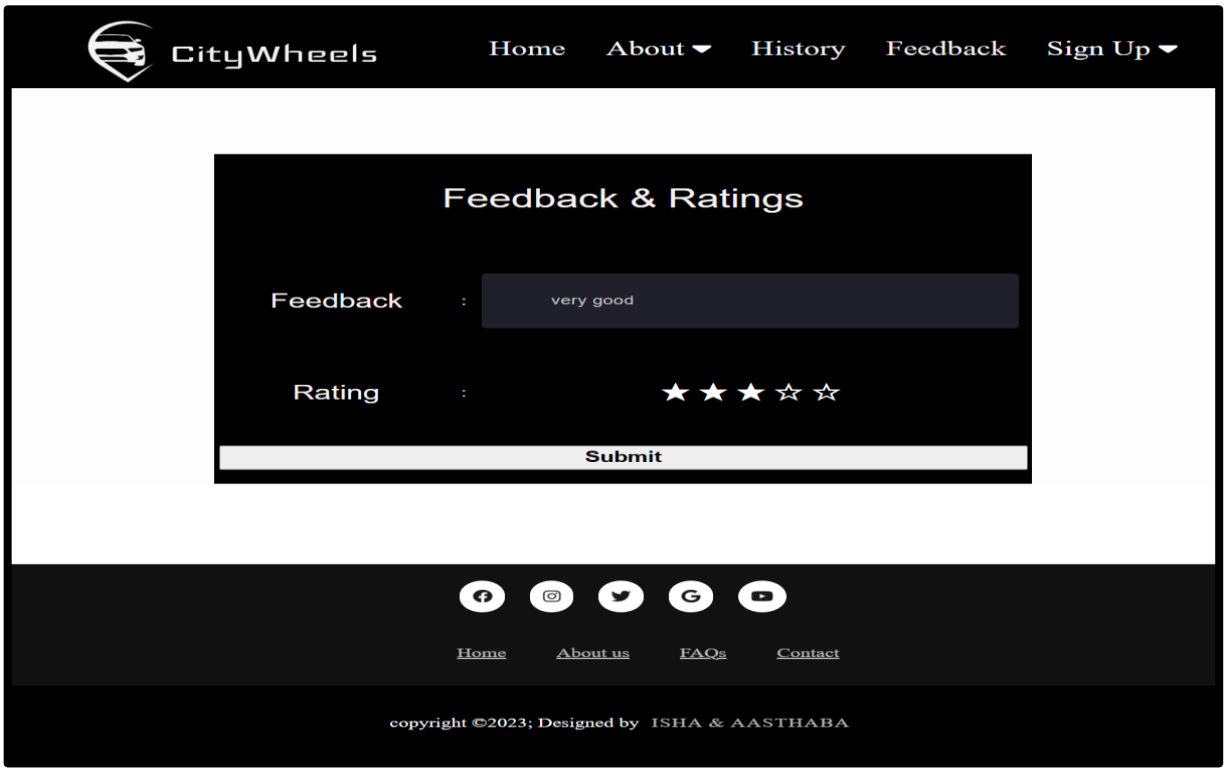
Invoice:



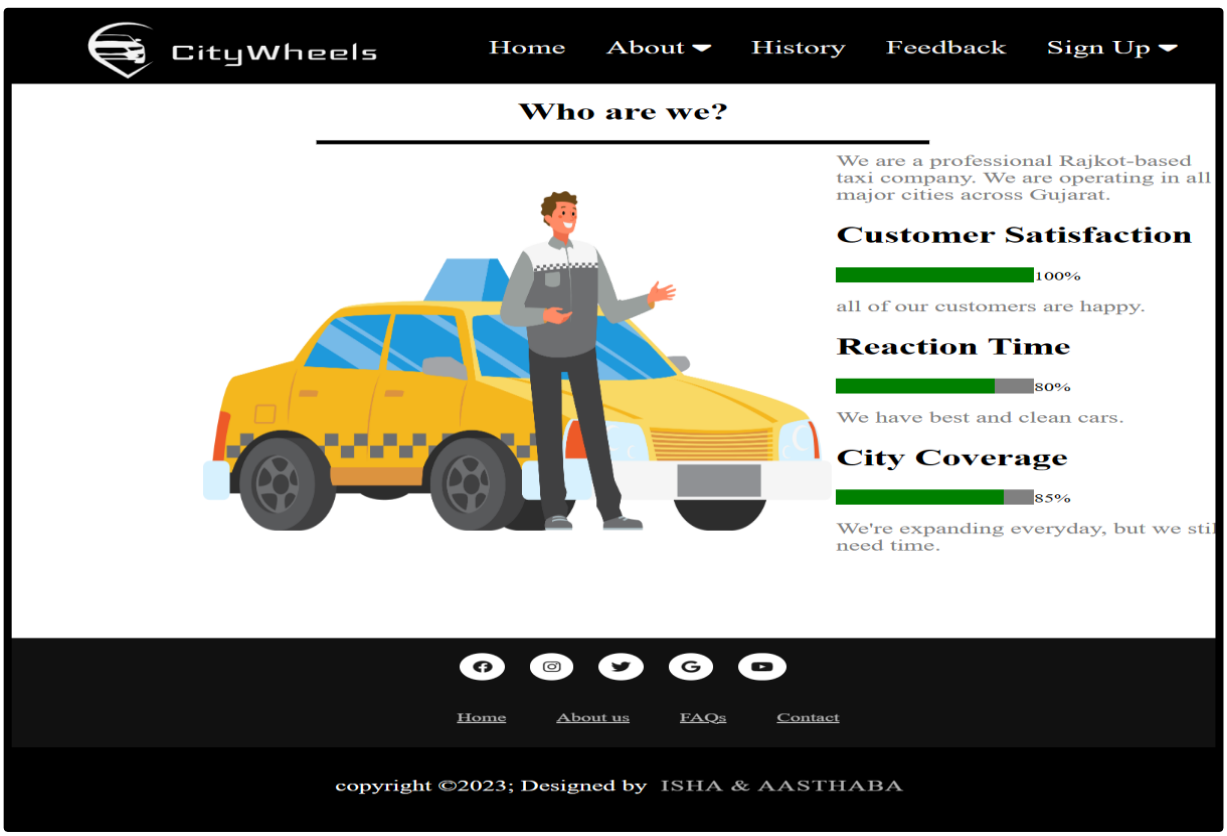
History:



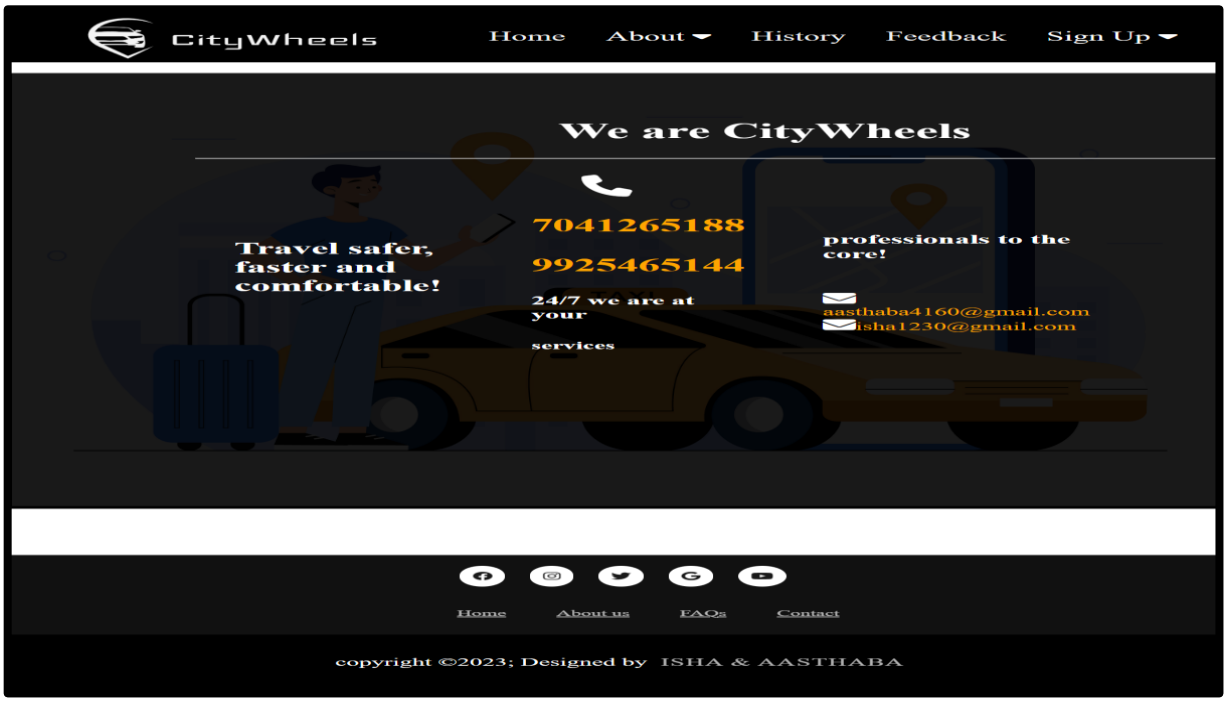
Feedback:



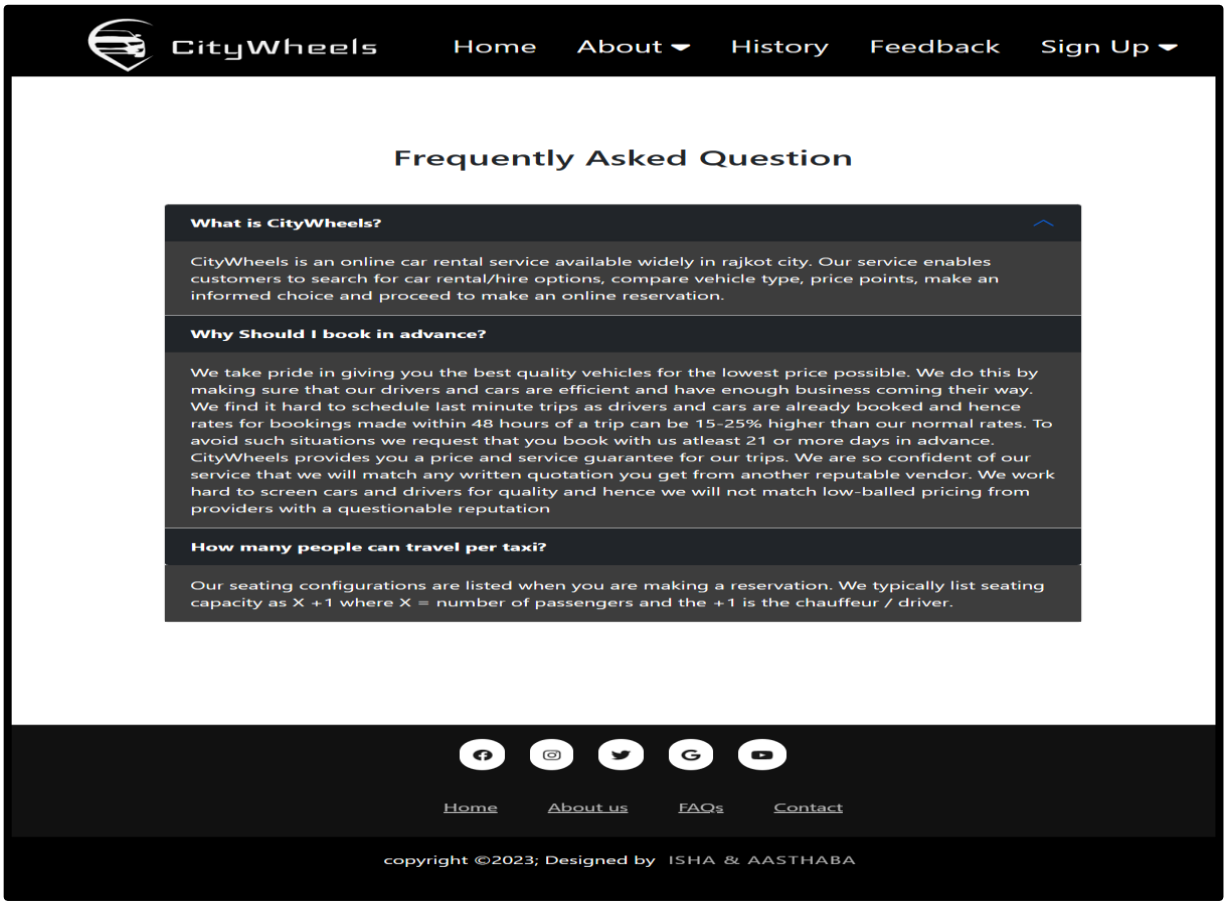
About us:



Contact us:



FAQS :



5.2 Testing Approach :

- **Black Box Testing :**

Black box testing is a software testing technique that focuses on examining the functionality of a software application without considering its internal code structure, logic, or implementation details. In other words, testers conduct black box testing without any knowledge of the software's internal workings; they treat the software as a "black box" where inputs are provided, and outputs are observed and analyzed.

- Common types of black box testing include:

Functional Testing: Ensures that the software functions according to its specifications, including testing individual functions, features, and user interactions.

Integration Testing: Validates the interactions between different components or modules of the software to ensure they work together as expected.

System Testing: Examines the complete software system to ensure it meets the specified requirements and performs well as a whole.

Acceptance Testing: Performed by users or stakeholders to determine whether the software meets their acceptance criteria and is ready for deployment.

Black box testing is valuable for uncovering defects, verifying that the software meets user requirements, and ensuring that it functions correctly from an end-user perspective. It complements other testing techniques like white box testing, which examines the internal code structure and logic of the software. Together, these testing methods provide comprehensive coverage of software quality assurance.

- **White Box Testing :**

White box testing, also known as clear box testing, structural testing, or glass box testing, is a software testing technique that focuses on examining the internal structure, logic, and code of a software application. Unlike black box testing, which tests the functionality of the software without considering its internal details, white box testing involves inspecting and testing the code, data structures, algorithms, and other implementation-specific aspects of the software.

- Common techniques and approaches used in white box testing include:

Statement Coverage: Ensuring that every line of code in the program is executed at least once during testing.

Branch Coverage: Ensuring that all possible branches (if statements, loops, and conditional statements) within the code are tested.

Path Coverage: Testing all possible paths through the code, considering various combinations of conditions and control flow.

Code Reviews: Manual or automated reviews of the source code to identify potential issues.

Static Analysis: Using tools to analyze the source code without executing it, looking for coding issues, security vulnerabilities, and adherence to coding standards.

White box testing is valuable for uncovering defects related to the internal workings of the software, ensuring code quality, and optimizing performance. It is commonly used in conjunction with black box testing to provide comprehensive test coverage and improve software quality.

- **Gray Box Testing :**

Gray box testing is a software testing technique that combines elements of both black box testing and white box testing. In gray box testing, the tester has limited knowledge of the internal workings of the software, somewhere between the complete knowledge of white box testing and the lack of knowledge in black box testing. This means that the tester may have access to some information about the code, architecture, or design of the software, but not a full understanding of its implementation details.

Gray box testing is particularly useful when complete transparency of the software's internals is not feasible or practical but some knowledge is necessary to design effective test cases. It provides a middle ground between the black box approach (which relies solely on external behavior) and the white box approach (which requires full knowledge of the code) and can be a valuable addition to a comprehensive testing strategy.

- **TOP-DOWN Approach :**

It starts with Main Program which is calling program and others are called routine. Initially there may be very few modules called in main but gradually other modules are completed, they can be called in the main.

All working modules are leaf nodes in structure chart so until all lower modules are finished main is not able to accomplish the job. So it is sometimes difficult to test upper component in absence of lower modules.

Full system is tested repeatedly after calling routines in main so effect of adding a single modules on other modules is also recognized step by step so this becomes more manageable.

This is a text about top-down programming, which is a software development methodology that breaks down a program into smaller and smaller subroutines or modules. The main program is at the top of the hierarchy and calls the other modules. The other modules can call other modules, and so on.

- **Top-down programming has several advantages, including:**

  - It makes the program easier to understand and maintain.
  - It allows for the development of modules in parallel.
  - It makes it easier to test the program.

However, top-down programming can also be challenging, especially for large and complex programs. One of the challenges is that it can be difficult to design the modules correctly at the beginning of the development process. Another challenge is that it can be difficult to test the upper modules of the program until the lower modules are completed.

- **BOTTOM -UP Approach :**

Each sub system tested in discrete way and integrated gradually.Modules are integrated after unit testing and component has been formed.Individual components tested separately and sub system generated.All subsystem tested and gathers to form system.In system having big size and several levels this approach gives best result.
All team can accomplish their individual testing parallel.Disadvantage is that we can run complete system at the end of each sub system.

- **Bottom-up programming has several advantages, including:**

  - It is easier to test individual components than it is to test a complete system.
  - It allows for the development of components in parallel.

General Test Cases:

Sr. No.	Test Case	Expected Result	Passed / Failed
1	Type the URL of the Home page in the address bar of the browser and try to access the application by passing the Login page	The Login page of the application page should be displayed and the user should not be allowed to access the application directly	Failed
2	Login into the application by giving valid Username and Password. Then logout from the application. Then click on the Back button of the browser	The Login page should be displayed and user should not be allowed to view the previous page	Passed
3	Check whether the password field is stored in the database	The password should be stored in the.	Passed
4	Check the title of the internet explorer window	The name should be meaningful and should reflect the action done in window	Passed
5	Check the size of buttons in all windows	All the buttons should be of same size	Passed
6	Check the size of pop up screen	All the pop up screens should be of appropriate size according to the action it is doing	Passed
7	Check the placement of all the controls	All controls should be properly placed	Passed
8	Check the color in all window	The color should be uniform	Passed

9	Check the font and size of font in different pages	The font and size of font should be same	Passed
10	Login to the site and click on 'Refresh' button on the browser	The page should get refreshed	Passed
11	Click on 'Back' button on the browser (other than Home Page)	It should go back to the previous page	Passed
12	Click on 'Forward' button on the browser	It should go to the next page (only for those pages for which are browsed already)	Passed
13	While doing any of the functions with regards to application click on browser 'Stop' button	It should stop that process	Passed
14	In the site, check if there any link appears, if there is any, right click on that link and select 'Open in a New Window'	The corresponding link should be displayed in new window	Passed
15	In the site check if there any link appears, if there is any, right click on that link and select 'Open Link'	The corresponding page for the link should be displayed.	Passed
16	Check for the expiration of the session.	An opened session should be kept as open for pre-defined time, the session should not get expired in between some operations are being performed	Passed

17	Check the functionalities in all the browsers like different versions of IE, different versions of Netscape etc…	In all the versions of the browsers the functionalities, fonts and images should be same.	Passed
18	Click on the link provided for logout	It should take to the login screen	Passed
19	After logging out click on back button in the browser	It should not go to the previous page	Passed
20	Check session is destroyed after logout, Try direct URL	It should not allow you to move directly without login to application. And login page should be displayed	Passed
21	Without sign up user can't login	It shows pop up box that shows "sign up first"	Passed
22	Without sign up and login user can't book cab	It shows pop up box that shows "please login first"	Passed
23	Without login user can't see their ride history	It shows pop up box that shows "please login first"	Passed
24	Without login user can't give feedback	It shows pop up box that shows "please login first"	Passed
25	User can't see other passenger's ride history	The ride history is based on the user login so user can only see their rides in the history	Passed

Chapter 6: System Security And Measures

System security :

User side:

This PHP code manages user login for a website. It checks if a login form was submitted, verifies the provided phone number's validity, and attempts to match it with a stored password in a database. If successful, it starts a session, records the phone number, and redirects to a page for choosing a cab. If the login fails, it redirects to the signup page with an error message. Additionally, it includes a configuration file for database connection.

Here is the code of login that saw security feature describe above.

```
<?php
session_start();
include "includes/config.php";


if(isset($_POST['login']))
{
   $password=$_POST['logpass'];
   $phone_no=$_POST['logphone'];


   $check = "SELECT `password`, `phone_no` FROM `signup_login` WHERE phone_no='$phone_no' AND password='$password'";
   $query = mysqli_query($con,$check);


  if(!is_numeric($phone_no) || !preg_match('/^[0-9]{10}+$/',$phone_no) )
   {
      echo "<script>
```

```
                alert('Invalid Phone number');
                window.location.href = 'login.php';
            </script>";
        }
        else
        {
            if($query->num_rows>0)
            {
                $_SESSION['phono']=$_POST['logphone'];
                echo "<script>
                    alert('You have login succesfully');
                    window.location.href = 'choosecab.php';
                </script>";
            }
            else
            {
                echo "<script>
                    alert('You are not signup yet. Please signup first');
                    window.location.href = 'signup.php';
                </script>";
            }
        }
    }
    ?>
```

Admin side:

This PHP code is for handling user login functionality. It starts a session and includes a configuration file. When a user submits a login form, it retrieves the entered username and password. Then, it checks if these credentials match any record in the "admin_login" table of a database. If a match is found, it sets a session variable to indicate that the user is logged in as an admin and redirects them to a dashboard page. If no match is found, it displays an error message stating "Invalid Details."

```
<?php
session_start();
include('includes/config.php');
if(isset($_POST['login']))
{
   $username = $_POST['username'];
   $password = $_POST['password'];
   // Check if the username and password are valid
   $sql = "SELECT username,password FROM `admin_login` WHERE username='$username' AND password='$password'";
   $result =mysqli_query($con,$sql);
   if ($result->num_rows>0)
   {
    // The username and password are valid, so log the user in
    $_SESSION['alogin']='admin';
    header("location:dashboard.php");
   }
```

```
    else {
            // The username and password are not valid, so display an
        error message
        echo "<script>alert('Invalid Details');</script>";
      }
}


?>
```

Chapter 7: Future Scope And Enhancement

I want to implement this features in future :

- Multiple Stop Option: This feature allows passengers to add more than one destination to their trip, making it easier to pick up or drop off friends or run errands along the way.
- No. of Users 2-Way Intercity Ride Emergency SOS: It tells you how many people can ride together for a round-trip journey between cities, and it includes an emergency SOS button for safety.
- No. of Booking Airport Pick Up/Drop Augmented Reality: You can book rides to and from the airport, and the app uses augmented reality to provide helpful information.
- Cab Sharing In-App Messenger: Passengers can communicate with each other or their driver through an in-app messaging system when they're sharing a cab.
- Passenger Ratio: It shows the number of passengers compared to the number of drivers available.
- Driver Ride History Online Pay with UPI: Drivers can see a history of their past rides, and passengers can pay for their rides online using UPI (a digital payment method.
- Cars Availability Tracking Panel: A tracking panel shows the availability of cars in real-time.
- Share a Review Call Help: Passengers can share their thoughts and call for assistance if needed.
- Invoice Generation: The app automatically creates receipts for completed rides.
- City Profile Digital Card: It provides information about the city you're in and offers a digital card for access to local services and discounts

Chapter 8: Conclusion And Limitation

Conclusion :

To conclude we would like to say that the main purpose of our project was to booking cab online & we would like to say that we have achived that task successfully.

There are many other things which can be done in this project like, can create an applications for this, live cab tracking can be added to this project, live chat can be added …. And many other things which can make take this project to another level.

Limitation :

Our limitation in this project are that there is no live tracking cab service ,only have cash payments ,no live chat to communicate directly…,etc .If the some how the cab that user choosed for ride is not available then the user can not get the instance message for that the admin should call the user and request to book another cab. Currently we don't have the online payment method in the system. which we can surely cover in future development.

Chapter 9: Bibliography

Bibliography :

Front-End:

1.W3school.com . HTML tutorial (https://www.w3schools.com/html/html_forms_attributes.asp)

2.W3school.com . SQL tutorial (https://www.w3schools.com/sql/default.asp)

3.FineGap. (https://youtu.be/IcyXS9aL4bs?si=JU-W-oPDYKldnSyN)

4.Vicode Media. (https://youtu.be/jlkHoJPSyfU?si=IbOvjHcNAaC0u4If)

Back-End:

5. Coder_mnv . (https://youtu.be/jlkHoJPSyfU?si=IbOvjHcNAaC0u4If)

6.Cyber Worriors. (https://youtu.be/XbnP7PL7CL0?si=tvmd4Ax6Ncck2UlD)

7.Thapa technical . (https://youtu.be/jU_KvKFwxJI?si=oFE5ac9z5HzMG3E2)

8.Andiran Twarog . (https://youtu.be/qNifU_aQRio?si=XfgpqlaKGCAUZD9j)

9.Keep Coding . (https://youtu.be/G3tZUO2fAEU?si=LieH3nvanHHxWA07)

10. Academind . (https://youtu.be/23bpce-5s8I?si=tdMRQpP8mcd9wdwq)