



**FACULTY OF ELECTRICAL ENGINEERING**

**SEEL 4213 – SOFTWARE ENGINEERING**

SESSIONS 2023/2024 - 01

SECTION 1 - GROUP 4

**GROUP ASSIGNMENT**

**CARENTAL SYSTEM**

**Lecturer: Dr. Ismahani Binti Ismail**

<b>No</b>	<b>NAME</b>	<b>MATRIC NO.</b>
1.	Anas Ahmed Fahmy	A19EE4063
2.	Mahmoud Mohamed Fathy	A19EE4068
3.	Aminu Ibrahim AbdulHafeez Mayowa	A19EE4062
	Mohamed Mamdouh	A20EE0299

# Title:

## CARENTAL SYSTEM

# Objectives:

1. To develop a system that can be used by users.
2. To develop a system that applies some calculations.
3. To develop a system that fetches data from a database.
4. To have an interesting GUI and user friendly.
5. To use Object Oriented Programming OOP.

# Introduction:

Our project aims to achieve multiple objectives in light of the changing car rental industry. We will focus on developing a user-friendly system that includes efficient calculations and seamless integration with a strong database. The primary aim is to redefine the car rental experience, eliminating traditional hassles and introducing a modern, digitalized approach that prioritizes user convenience and system efficiency.

# Requirements:

1. To install XAMPP as a local server.
2. To create the database on XAMPP server using MySQL.
3. To code the dashboard, signup and login pages with any language. In our case we use PHP and HTML.
4. To display the car rental amount on the dashboard after selecting the car and duration on the backend.
5. To develop the GUI of the system and have some friendly animation.

# Background:

The CARENTAL concept is born out of a fast-changing car rental market, where conventional models are finding it difficult to satisfy the ever-changing demands of contemporary customers. The project intends to address these difficulties by providing a technology-driven, user-friendly solution that smoothly blends sophisticated computations and extensive database operations, with an emphasis on reinventing the car rental experience. CARENTAL adopts a contemporary, digital approach to car rental transactions, prioritizing effectiveness, user-friendliness, and system dependability. With a focus on secure database administration, object-oriented programming (OOP) design, and user-friendly GUIs, the project aims to be a proactive solution that anticipates and addresses upcoming demands in the ever-changing vehicle rental market. CARENTAL seeks to set a new standard by providing a cutting-edge, user-focused platform for both service providers and clients, establishing itself as a benchmark in vehicle rental systems.

# Design Procedure:

This part outlines the design techniques that influenced the development of the "CARENTAL" project. The methods are categorized into four primary sections, each focusing on a crucial element of our growth process.

In the initial phase of the "CARENTAL" project, we focused on setting up the environment and database. This involved the use of XAMPP to create a robust development environment that integrates Apache, MySQL, PHP, and Perl. The database setup was pivotal, as we established a MySQL database to manage crucial data related to users, cars, and bookings. Points within this part include "Environment Setup" and "Database Setup," both of which laid the foundation for the project's development and data management.

The second phase focused on Security and user authentication were paramount in ensuring the trustworthiness of the "CARENTAL" system. To achieve this, we implemented a comprehensive set of procedures. "Signup and Login Setup" was instrumental, where a hashing function was introduced to encrypt user passwords and enhance security. Additionally, "User Data Hashing" was a crucial step in safeguarding sensitive information. These measures collectively ensured that user data remained secure and protected from unauthorized access.

Then we started with the user interface and overall system functionality that were key aspects of delivering good user experience. Within this part, "Dashboard Setup" played a central role, as we developed the primary interface using PHP. A visually appealing and user-friendly "GUI Implementation" was crucial to make the system easy to navigate. The "Implementation of OOP Classes" ensured a structured and maintainable codebase, allowing for efficient management of user interactions and data. The "Booking System" was another vital component in this phase, allowing users to reserve cars seamlessly.

Efficient data management and catering to user preferences were addressed in this part. We initiated "Database Initialization on XAMPP Server" to ensure the smooth storage and retrieval of user data. To manage user and car data effectively, "Creation of 'users' and 'cars' Tables" was implemented, providing structured data organization. Lastly, "Offering Various Car Categories" was integrated into the system to offer users a diverse selection of vehicles to choose from, enhancing their overall experience.

## Results:

### Back-End:

The backbone of CARENTAL lies in its backend. This section delves into the core functionalities that drive the system, ensuring a seamless and secure car rental experience. We will focus on three main parts managing data, processing requests, and ensuring the functionality of the car rental system.

**Managing Data and Security:** The backend of the CARENTAL project plays a critical role in managing data and ensuring the security of user information. It utilizes MySQL for efficient database management, where data related to users, cars, and bookings are stored and retrieved securely. To protect user data, robust security measures are implemented, including user authentication with secure password hashing and data encryption practices. This ensures that sensitive information remains confidential and protected from unauthorized access.

**Processing Requests and Server-Side Functionality:** Processing user requests and maintaining the functionality of the car rental system are key responsibilities of the backend. It handles user inputs, car bookings, and user account management, ensuring a seamless user experience. Server-side scripting, implemented in PHP, acts as the bridge between the frontend and the database, facilitating smooth communication. Additionally, the system is designed to integrate with external APIs, potentially enhancing functionalities such as GPS tracking and payment processing, making the rental process more efficient and convenient.

**Error Handling and Logging:** To maintain the reliability and integrity of the car rental system, robust error handling and logging mechanisms are in place. These procedures are vital for ensuring smooth operation and troubleshooting potential issues. The backend keeps track of system errors and maintains detailed logs, allowing for the identification and resolution of any technical challenges that may arise during operation. This proactive approach to error management ensures that the CARENTAL system operates flawlessly and provides users with a hassle-free car rental experience.

## Front-End: Graphical User Interface (GUI):

For our graphical user interface, we use CSS as our backend with some animation and images to make it look nice. We have several pages for our website. In any website there is a signup page for new users, where they can create their new account. Figure 1 below shows the signup page.

As we can see there is a button to create the account and another button to navigate user to login page in case user already have an account and exists in our database.

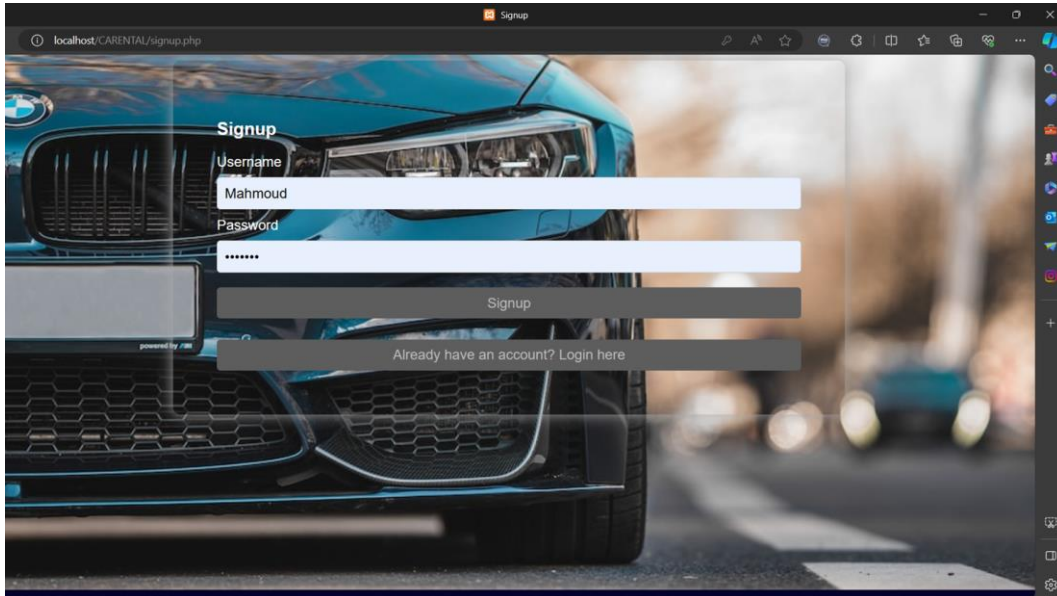


Figure 1: Signup page

For the next page, we have the login page and, on these users, can access their accounts to browse the cars available and to make a reservation.

Once you click login, it will navigate you to the main page of our system. If happen that a new user access this page, the user can click on signup here and it navigates user to signup page to create an account.

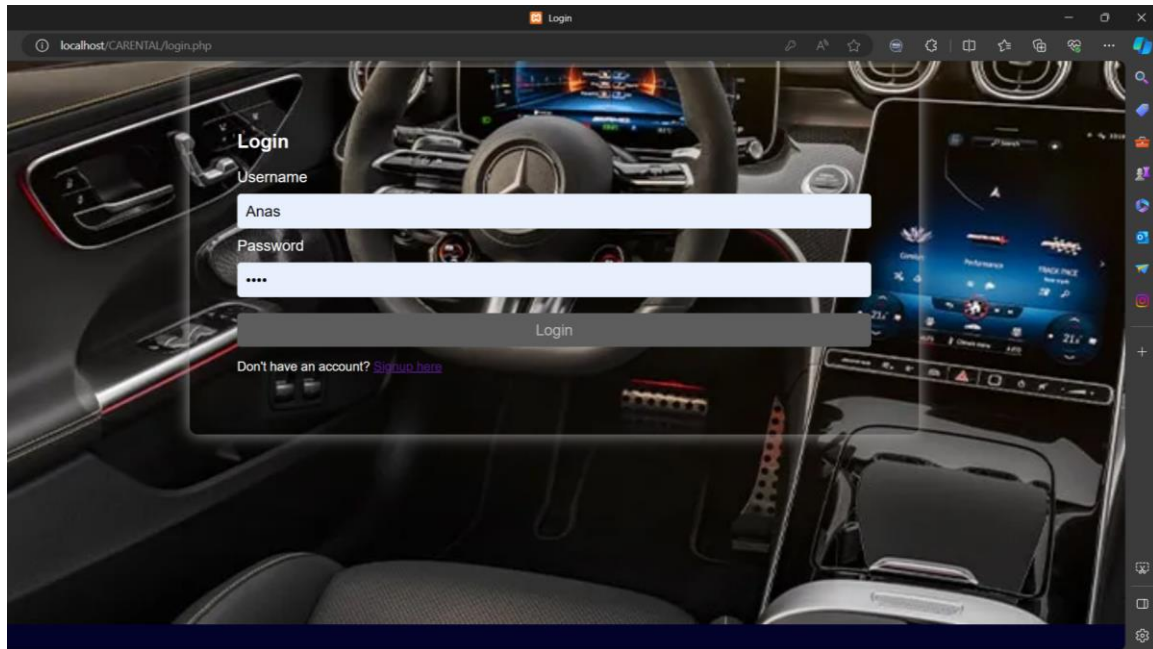


Figure 2: Login page

After the login page, the main page of the system will appear. The dashboard pages. The dashboard page contains all the cars data and its status whether it's available or not and also price for renting per hour. It also contains the reservation time and date as well. The total amount of the rent will appear here on the system.

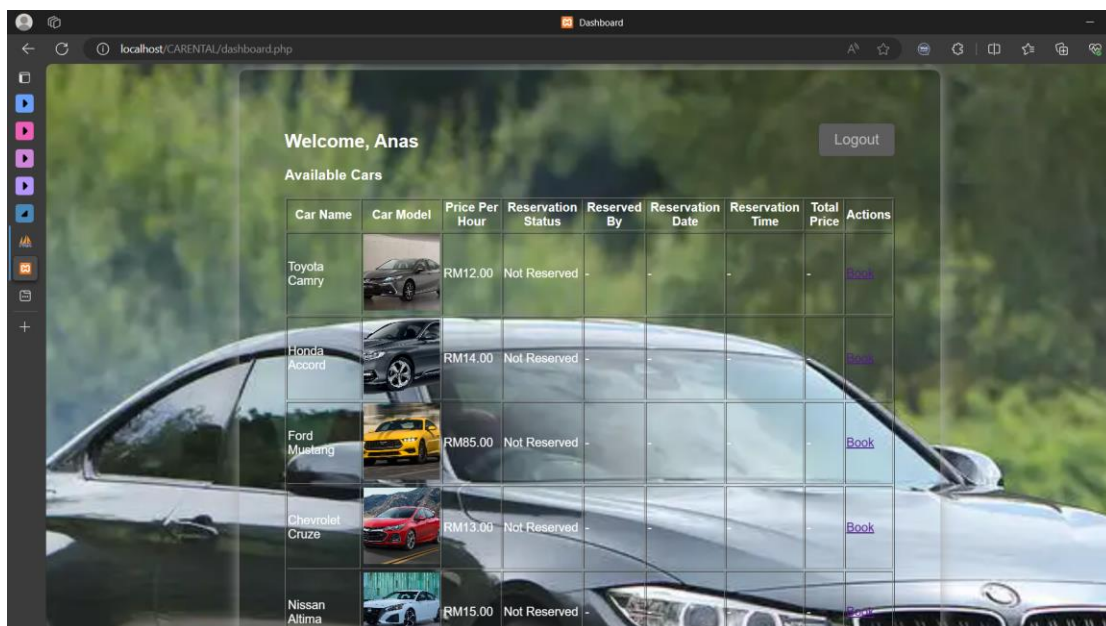
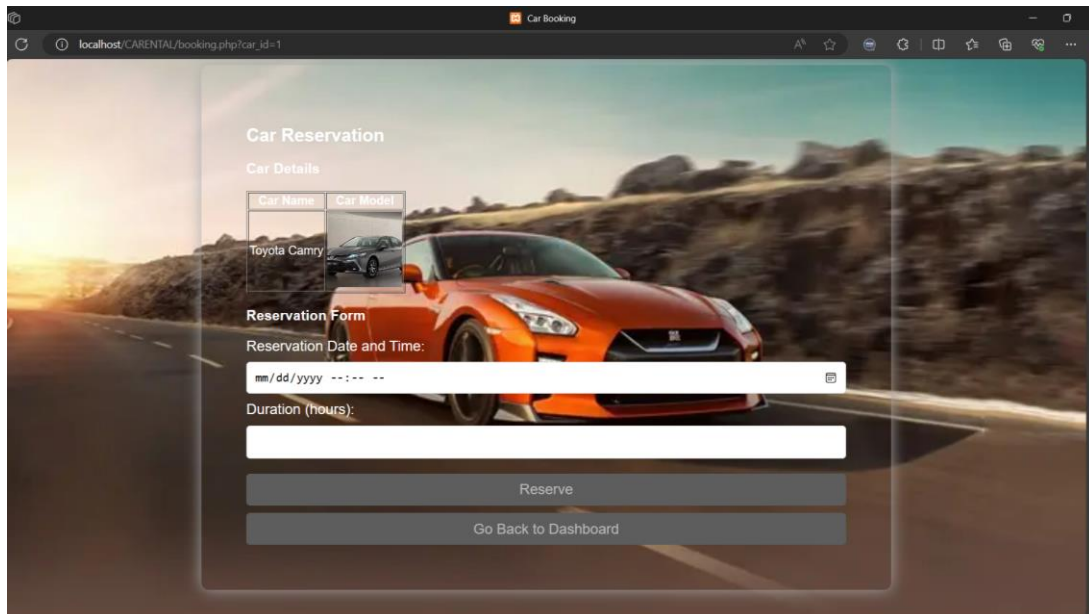


Figure 3: Dashboard page




The last page for our system is the booking page. As we can see on the dashboard page, once you click on the book button, it will navigate you to the booking page and on this page, you can choose the data, time and how long you need this car.

Once the customer is already confirmed and satisfied with the car, he can click on reserve and car will be reserved for him.



**Car Reservation**

**Car Details**

Car Name	Car Model
Toyota Camry	

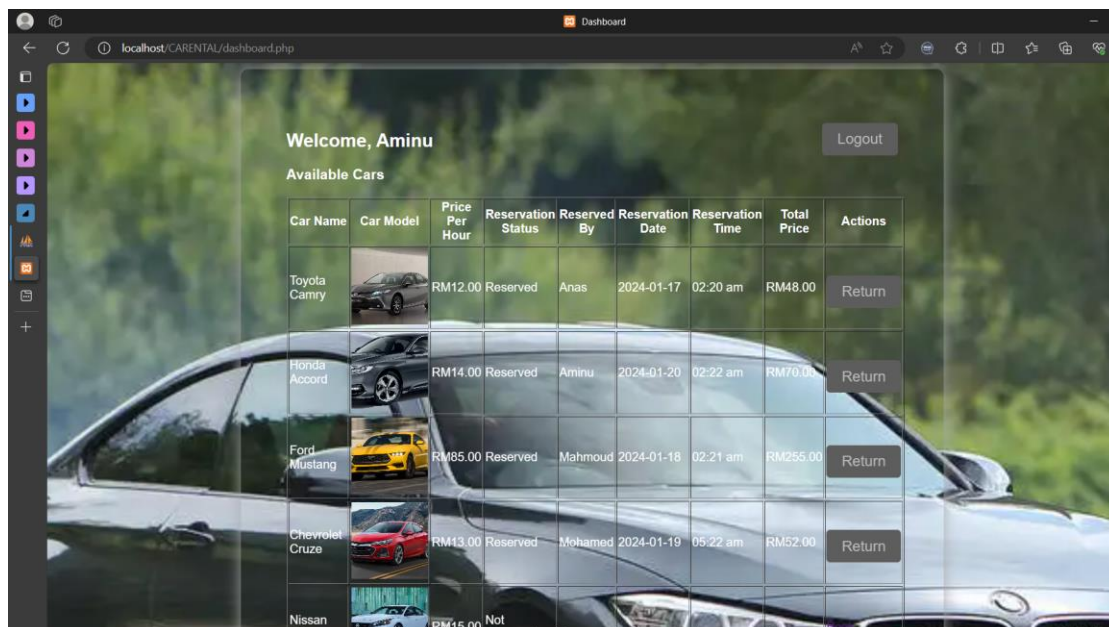
**Reservation Form**

Reservation Date and Time:

mm/dd/yyyy --:--

Duration (hours):

Figure 4: Booking page.



**Welcome, Aminu**

**Available Cars**






Car Name	Car Model	Price Per Hour	Reservation Status	Reserved By	Reservation Date	Reservation Time	Total Price	Actions
Toyota Camry		RM12.00	Reserved	Anas	2024-01-17	02:20 am	RM48.00	<input type="button" value="Return"/>
Honda Accord		RM14.00	Reserved	Aminu	2024-01-20	02:22 am	RM70.00	<input type="button" value="Return"/>
Ford Mustang		RM85.00	Reserved	Mahmoud	2024-01-18	02:21 am	RM255.00	<input type="button" value="Return"/>
Chevrolet Cruze		RM13.00	Reserved	Mohamed	2024-01-19	05:22 am	RM52.00	<input type="button" value="Return"/>
Nissan		RM15.00	Not					

Figure 5: Dashboard after customer reservation.

# Database:

The 'carental' database has been designed to facilitate the management of a car rental system. This report outlines the database schema, including the structure of the 'users' and 'cars' tables, along with the initial insertion of data into the 'cars' table.

## Database Structure

### 1. Users Table

The 'users' table is responsible for storing user information. It includes the following columns:

- id (INT): Auto-incremented primary key for unique user identification.
- username (VARCHAR(255)): Unique username for each user, not nullable.
- password (VARCHAR(255)): Password associated with the user's account, not nullable.

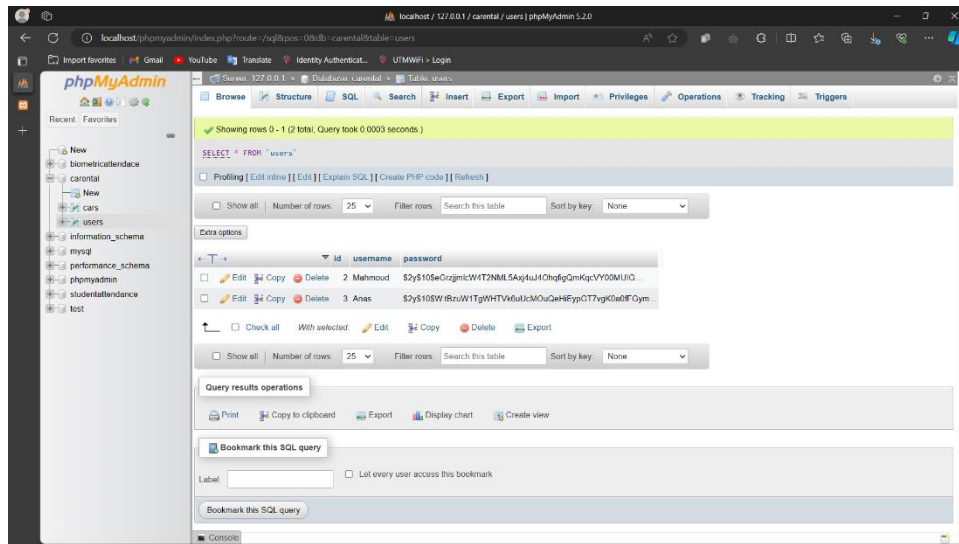


Figure 6: Users Table with Hashed Passwords

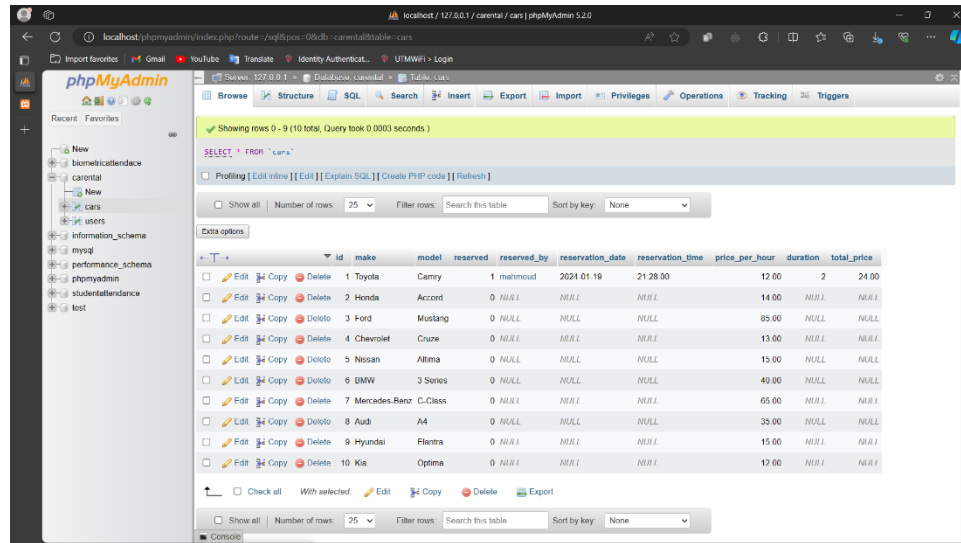
### 2. Cars Table

The 'cars' table stores information about available cars for rental. The table structure is as follows:

- id (INT): Auto-incremented primary key for unique car identification.
- make (VARCHAR(255)): Make of the car, not nullable.
- model (VARCHAR(255)): Model of the car, not nullable.
- reserved (TINYINT(1)): Indicator of whether the car is reserved (default is 0 for not reserved).



- reserved\_by (VARCHAR(255)): Username of the user who reserved the car, with a foreign key constraint referencing the 'username' column in the 'users' table.
- reservation\_date (DATE): Date when the car is reserved.
- reservation\_time (TIME): Time when the car is reserved.
- price\_per\_hour (DECIMAL(10, 2)): Price per hour in RM (Ringgit Malaysia), defaulting to 0.00.
- duration (INT): Duration of the reservation.
- total\_price (DECIMAL(10, 2)): Total price for the reservation.



id	make	model	reserved	reserved_by	reservation_date	reservation_time	price_per_hour	duration	total_price
1	Toyota	Camry	1	mahmoud	2024-01-19	21:28:00	12.00	2	24.00
2	Honda	Accord	0	NULL	NULL	NULL	14.00	NULL	NULL
3	Ford	Mustang	0	NULL	NULL	NULL	85.00	NULL	NULL
4	Chevrolet	Cruze	0	NULL	NULL	NULL	13.00	NULL	NULL
5	Nissan	Altima	0	NULL	NULL	NULL	15.00	NULL	NULL
6	BMW	3 Series	0	NULL	NULL	NULL	40.00	NULL	NULL
7	Mercedes-Benz	C-Class	0	NULL	NULL	NULL	65.00	NULL	NULL
8	Audi	A4	0	NULL	NULL	NULL	25.00	NULL	NULL
9	Hyundai	Elantra	0	NULL	NULL	NULL	15.00	NULL	NULL
10	Kia	Optima	0	NULL	NULL	NULL	12.00	NULL	NULL

Figure 7: Cars Table with all parameter columns.

## Initial Data

To populate the 'cars' table, 10 different cars have been inserted with their respective make, model, and initial price per hour in RM. The cars are as follows:

- Toyota Camry - RM 50.00 per hour
- Honda Accord - RM 60.00 per hour
- Ford Mustang - RM 12.00 per hour
- Chevrolet Cruze - RM 14.00 per hour
- Nissan Altima - RM 85.00 per hour
- BMW 3 Series - RM 13.00 per hour
- Mercedes-Benz C-Class - RM 15.00 per hour
- Audi A4 - RM 40.00 per hour
- Hyundai Elantra - RM 65.00 per hour

The 'carental' database provides a solid foundation for managing a car rental system, offering tables to store user information and details about available cars. The initial insertion of data into the 'cars' table allows for testing and further development of the system.

OOP:

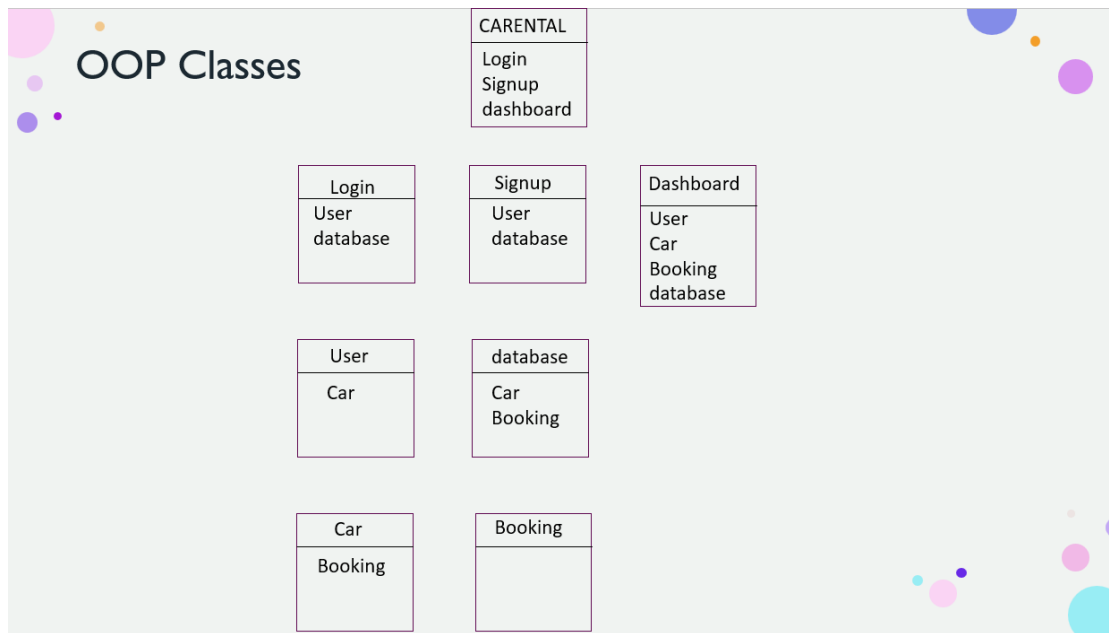


Figure 8: OOP classes

GitHub:

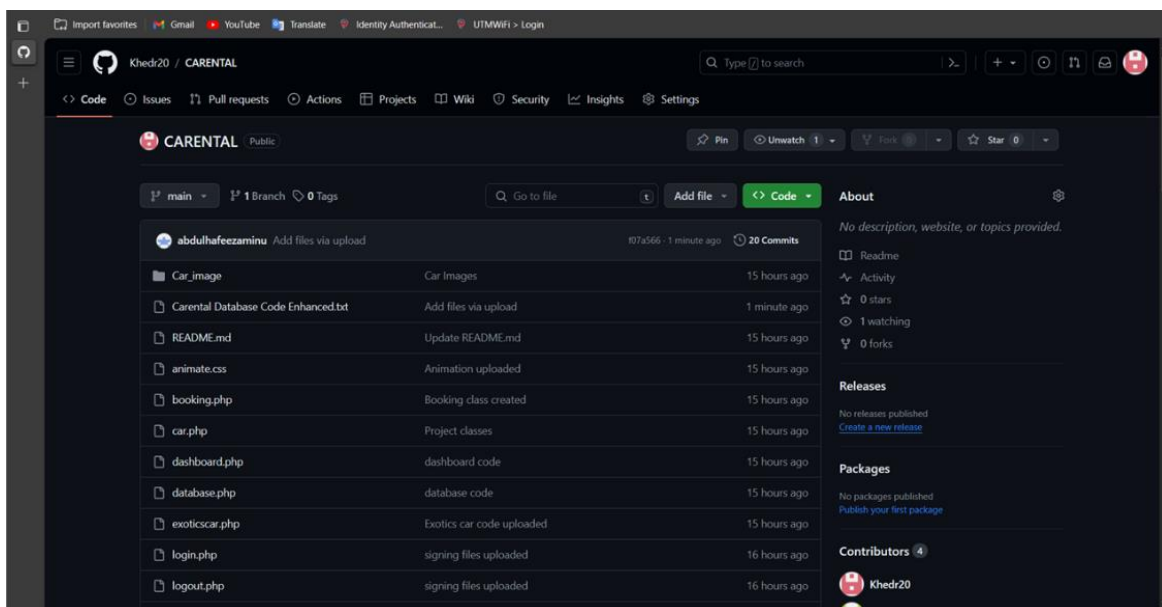


Figure 9: GitHub repository

Link: <https://github.com/Khedr20/CARENTAL>

# Conclusion:

The CARENTAL project lays the groundwork for a dependable and highly efficient system with its strong OOP design. This architectural strategy guarantees smooth performance and functionality, which adds to the project's overall success.

With a focus on protecting user privacy, CARENTAL has implemented a secure database and advanced data hashing algorithms to ensure the highest level of protection for user data. By placing a strong emphasis on data security, CARENTAL is positioned as a reliable and responsible solution in line with industry standards.

With an innovative price calculating capability and an easy graphical user interface, CARENTAL prioritizes user-friendliness over technical prowess. The project is an exceptional example of software engineering innovation in the cutthroat automobile rental industry since it not only demonstrates technological prowess but also strives to improve the whole customer experience.