

**NAMES: TUYIKUNDE Adeline**

**ID: 23389**

**COURSE: WEB TECHNOLOGY**

**DEPARTMENT: INFORMATION MANAGMENT**

**PROJECT NAME: CAR RENTAL MANAGEMENT SYSTEM**

1. **Description**

The Car Rental Management System is a comprehensive software solution designed to facilitate the efficient management of a vehicle rental business. It encompasses various functionalities to streamline the rental process, manage vehicles, handle customer interactions, and provide insightful analytics for business optimization. This system targets rental agencies, offering an end-to-end solution to handle bookings, vehicle availability, billing, and customer service.

1. **Purpose of the Project:**

Efficient Vehicle Management: The system should allow easy registration, categorization, and maintenance tracking of various vehicles available for rent.

**Customer Management:** Maintain a comprehensive database of customers, including their details, rental history, preferences, and payment information.

**Reservation System:** Implement a booking system that allows customers to reserve vehicles for specific dates, along with the ability to modify or cancel bookings.

**Inventory and Availability:** Real-time monitoring of available vehicles, their types, models, rental rates, and their status (e.g., rented, available, in maintenance).

**Billing and Payments:** Generate invoices, calculate rental charges based on duration, additional services, and manage payments (cash, card, online).

**Reporting and Analytics:** Provide insights through reports on revenue, popular vehicle types, peak booking times, etc., aiding in decision-making.

**Expected Outcome:**

* **User-friendly Interface**: Intuitive interfaces for both customers and staff to navigate the system easily.
* **Efficiency and Automation**: Automate processes wherever possible to reduce manual work, such as generating invoices, tracking vehicle maintenance schedules, etc.
* **Reliability and Security**: Ensure data security, backups, and reliability of the system to prevent any data loss or unauthorized access.
* **Enhanced Customer Experience**: Enable customers to easily find, book, and manage their rentals, leading to increased satisfaction and retention.
* **Improved Management**: Provide tools for management to track business performance, monitor inventory, and make informed decisions.

**Specific Constraints or Limitations:**

* **Security and Privacy**: Compliance with data protection laws and regulations to safeguard customer information.
* **Scalability**: Design the system to handle a growing number of vehicles, customers, and transactions without performance degradation.
* **Integration and Compatibility**: Compatibility with different platforms and devices, and possibly integration with external systems (e.g., payment gateways, accounting software).
* **Resource Constraints**: Consider budget, time, and technological limitations for development and implementation.
* **Maintenance and Support**: Plan for ongoing maintenance, updates, and user support to ensure the system's smooth operation post-implementation.

**Project Plan:**

1. Project Scope:

* **Objective**: Develop a comprehensive Car Rental Management System to streamline vehicle rental operations, enhance customer experience, and optimize business efficiency.
* **Key Features**: Vehicle management, customer database, reservation system, inventory monitoring, billing/payment, reporting/analytics.

2. Timeline:

* **Phase 1: Planning and Requirements Gathering** (1 weeks)
  + Define project scope, objectives, and requirements.
  + Identify potential risks and mitigation strategies.
* **Phase 2: System Design and Architecture** (12 days)
  + Design database schema, system architecture, and user interfaces.
  + Review and refine design based on feedback.
* **Phase 3: Development** (2 weeks)
  + Develop core functionalities iteratively.
  + Implement vehicle management, reservation system, billing, and reporting modules.
  + Conduct regular testing and quality assurance.
* **Phase 4: Implementation and Testing** (5 days)
  + Deploy the system in a controlled environment for testing.
  + Resolve bugs and refine system based on feedback.

3. Resources:

IN MY POJECT I USED THE Database named carrent\_db Which contain 4 tables (cardetails, useraccess, user, carrent)

2 non-functional (non-domain specific) features of my web-application.

✓ I was able to encrypt the user’s password (This is in case of security). It is possible that admin may view the user’s password so they can misuse their account which is why is decided to encrypt the password

✓ another non-functional feature is that it is easy to use Given the consumers' level of understanding, a basic yet high-quality user interface should be created to make it simple to comprehend and need minimal training

✓ Performance and Response Time: The system has a high-performance rate while executing user input and it is able to offer feedback or a response in a short amount of time, often 50 seconds for extremely difficult activities and 20 to 25 seconds for less sophisticated jobs.

I used:

**NetBeans server: Glassfish 4.1.1**

**Language: JSF (MVC framework)**

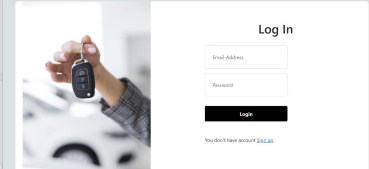
**Library: Hibernate**

**Connection: MySQL**

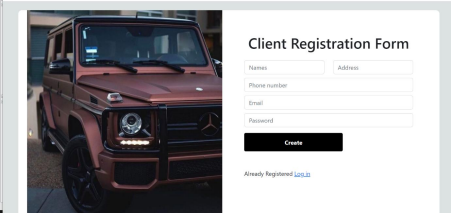
==> By running my project, it will immediately take you to my home page. which has two main menus

**- HOME**

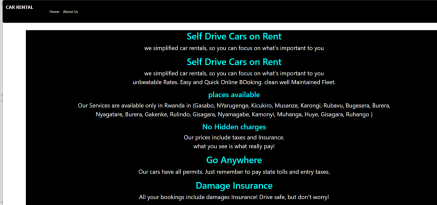
**-LOGIN** (It will take you to login page for the user who is already registered in the system)



**N.B:** if you are new to the system, by clicking on login menu instead of you entering credentials you will click on signup so that you will register yourself in the system. after that you hit login again then you will be able to see some of the services our system offers.



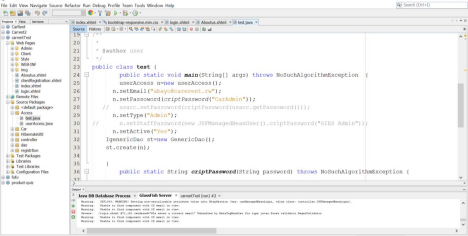
-**ABOUTUS:** this menu directs you to the page which explains more about this system (Car Rent System)

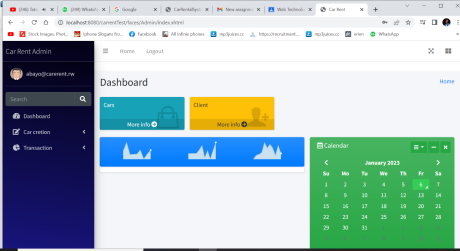
**==> *in this project there are two types of users***

• Admin

• Client

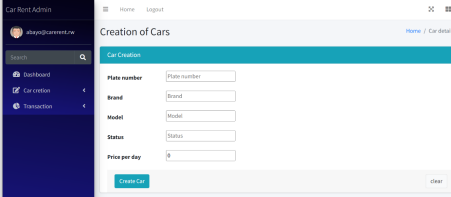
**Admin** is one user who has been added in the backend (source codes. )

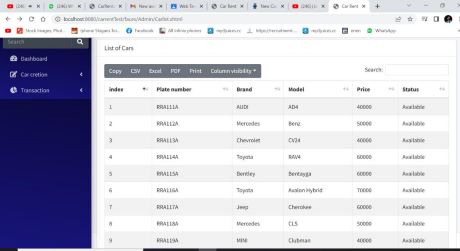
-he will be able to log in and directly headed to the page which has all the services he/she will be managing

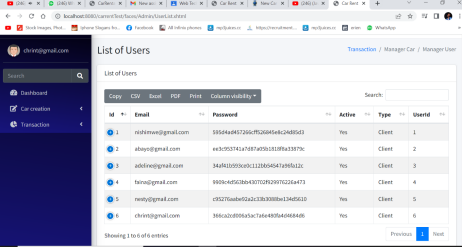


**Admin will be able to**:

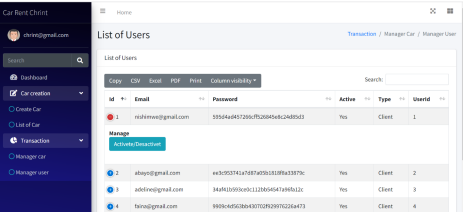
-register a car (under the dashboard by clicking on car creation, it will direct him to the page of creating the car or registering new car)

-view the car lists (both available and booked ones. Under the dashboard, car creation there is list of cars.)

-view users' lists.

figure

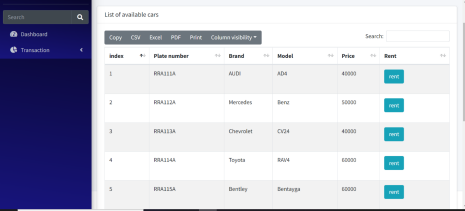
-activate and disactivate the users

-he will be able to receive returned car

**Client:** is one who is coming for our services.

(he will login and will be directed to the page of services we offer.)

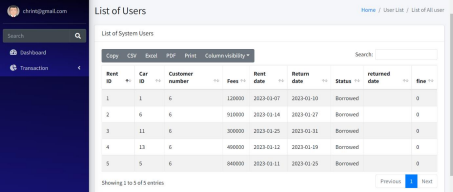
-he will be able view the lists of available cars

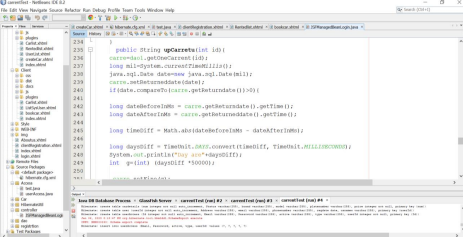


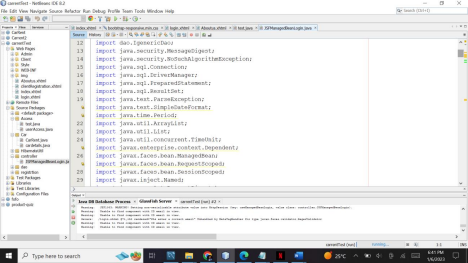
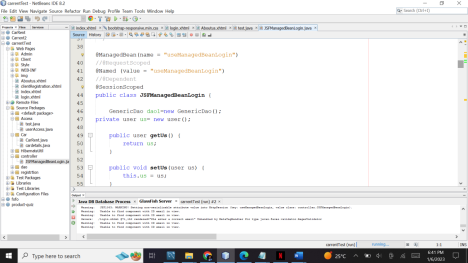
there is a button (Rent) in case a client found the car matching his desires he will click on that button and it will direct him to the renting page where he will have to enter the rent date and return date.

-he will be able to book a car

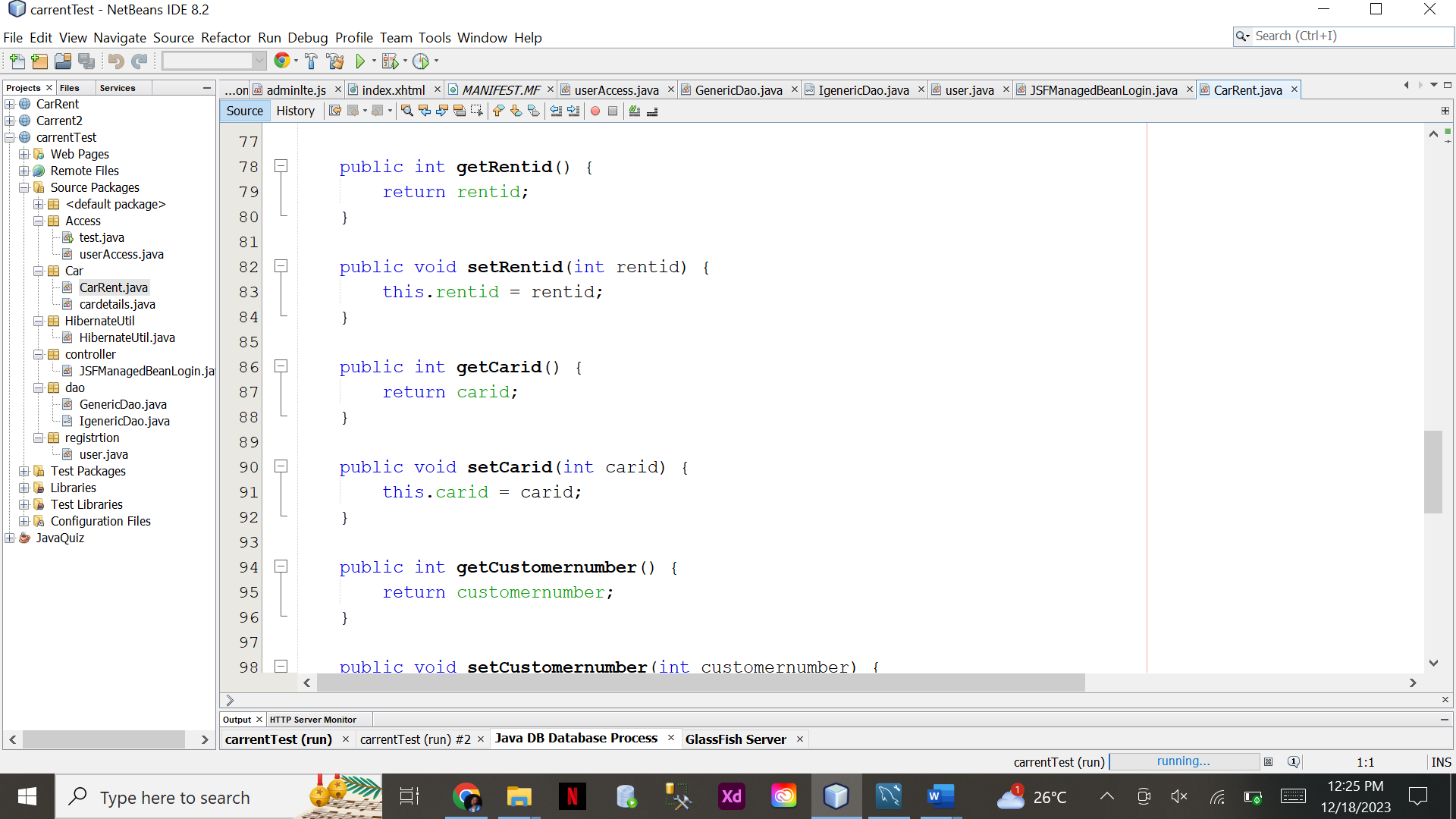
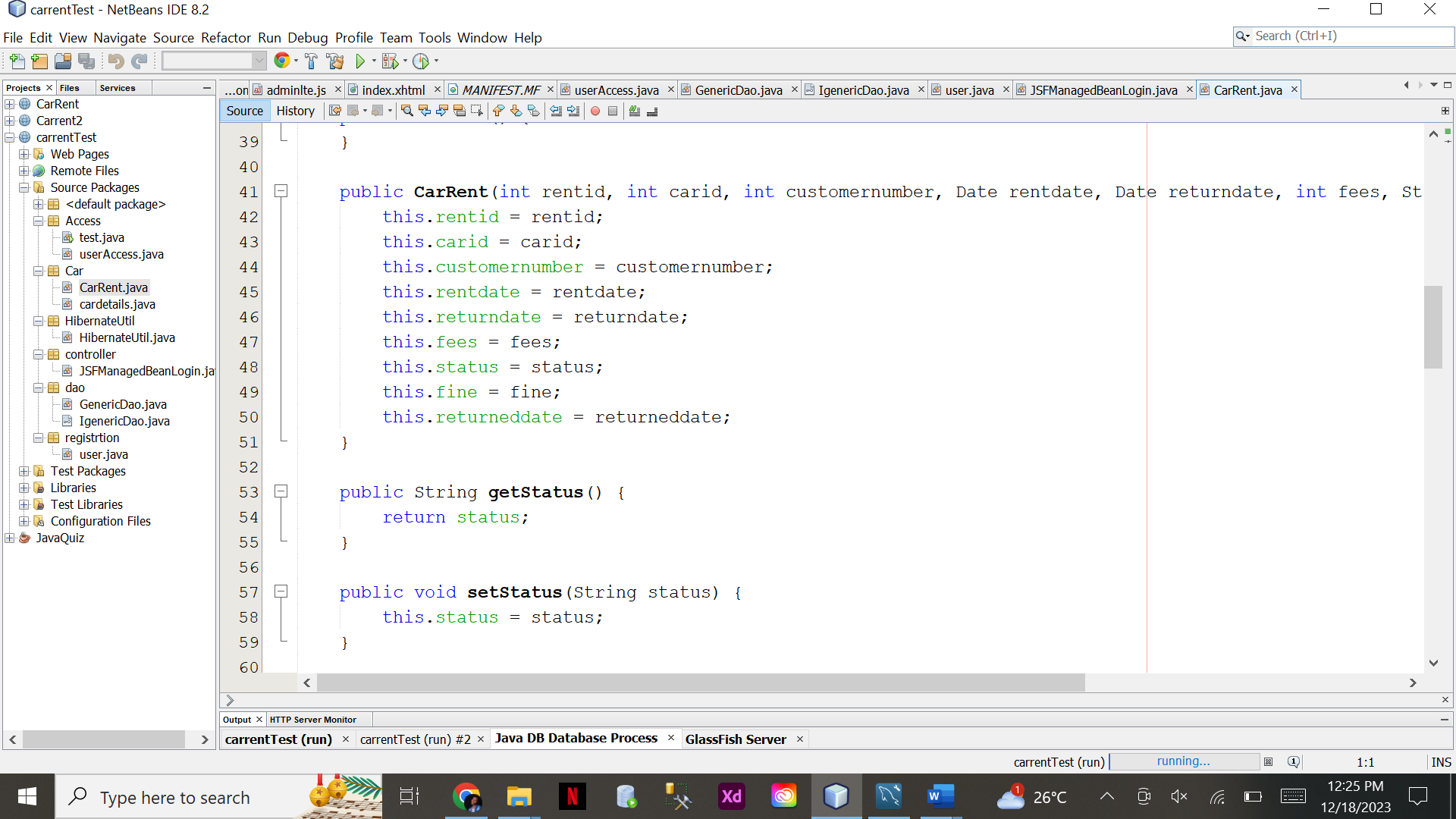
-he will be able view the list of his rents(car) which will be having fines in case he delayed in returning a car.



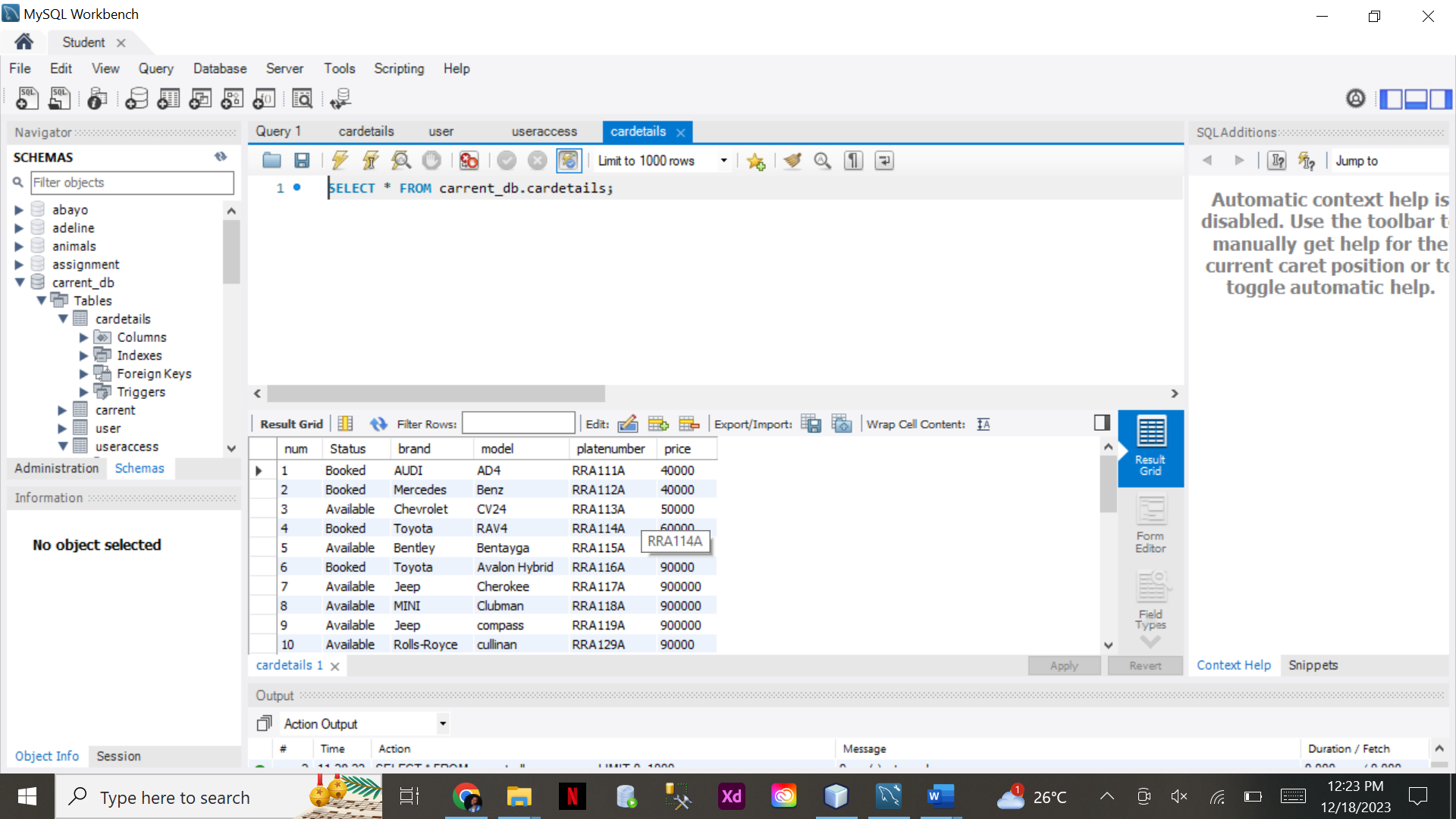
THIS IS HOW I CALCULATED THE DIFFERENCES BETWEEN DATES THESE ARE THE SCREENSCHOOTS FOR MY CONTROLLER

s

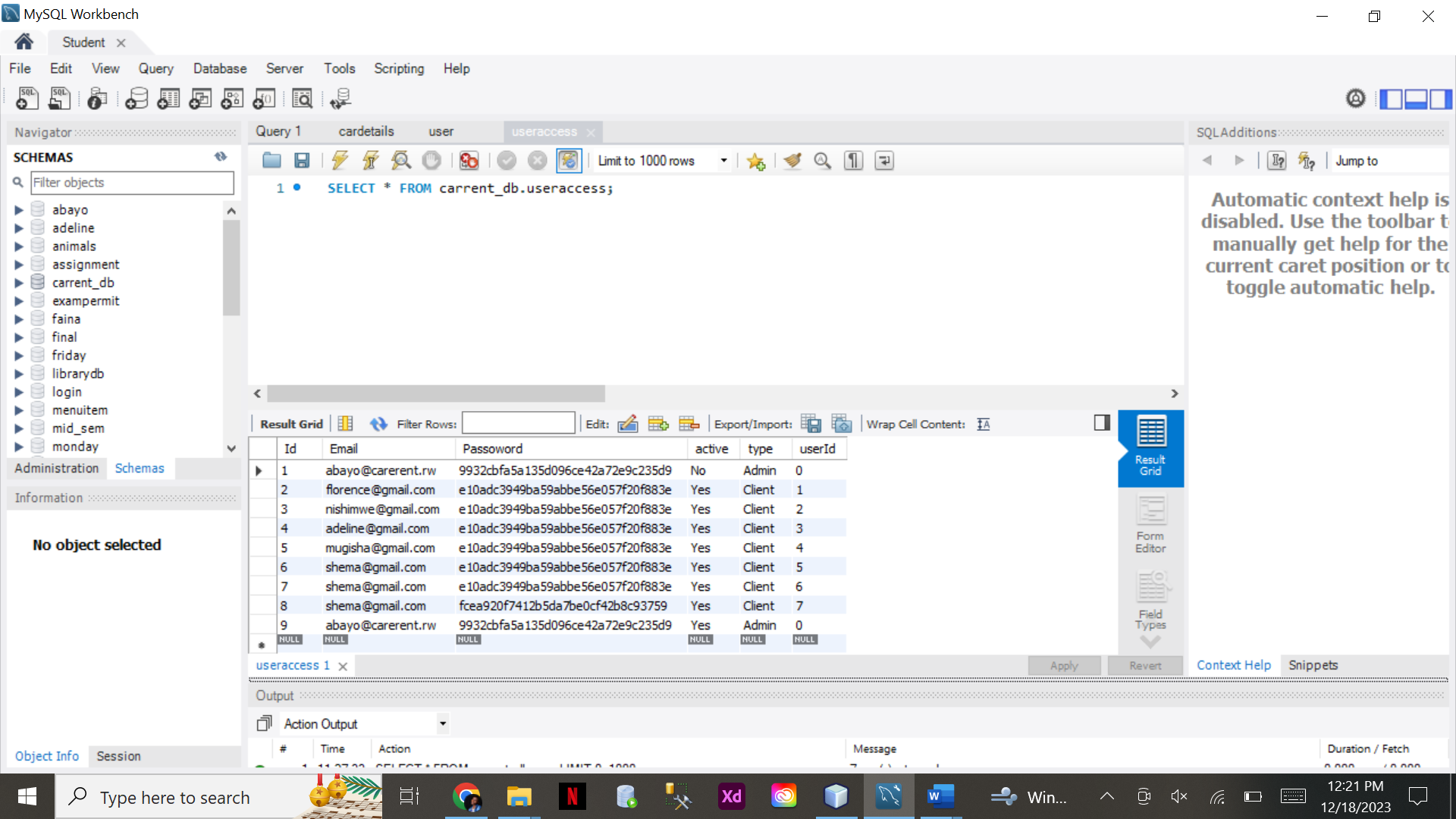
My CARRENT Entity



MY CARDETAILS TABLE



USER ACCESS TABLE



**This is the end of my project.**