**CIS 476 Term Project**

**Document/Program and Demo Due Dates**

**Monday 04/07/25**

This project is required for all students. You may finish the project all by yourself or team with others. 3 team members maximum.

In this project, you are required to design DriveShare - Peer-to-Peer Car Rental Platform. DriveShare is a standalone or web or mobile platform designed to connect car owners with individuals seeking short-term car rentals. The platform aims to provide a convenient and efficient solution for users to rent personal vehicles directly from their owners. DriveShare facilitates listing, searching, transactions and enhances user communication to ensure a seamless car-sharing experience. Use Turo.com as an example.

**Functions/Features**

* User Registration and Authentication:
  + Users can register on DriveShare using email and password authentication.
  + Include 3 security questions when registering.
* Car Listing and Management:
  + Car owners can list their vehicles for short-term rental, providing details such as car model, year, mileage, availability calendar, pick up location and rental pricing.
  + Owners have the ability to manage their car listings, including updating availability and price
  + The system should prevent the same car from being rented by more than once at the same time.
* Search and Booking:
  + Renters can search for available cars based on location, date, and other preferences.
  + Implement a booking system allowing renters to reserve a car for a specific period.
* Messaging and Communication:
  + DriveShare includes a messaging system to facilitate communication between car owners and renters.
  + Send notifications for booking requests, confirmations, and important updates via email or in-app messages.
* Payment:
  + Allow renters to make payment (A real payment is not required. A payment button with amount is enough, clicking on the button will change the balance and send notification to both owner and renter)
* Rental History (optional for working on the project alone)
  + Both car owners and renters can view their rental history.
* Reviews: (optional for working on the project alone)
  + Implement a review and rating system for both owner and renter to provide feedback on the rental experience.

**Implementation**

* User Authentication and Encryption:
  + Implement the Singleton pattern to manage the user's session securely.
* Observer Pattern for Booking Notifications:
  + Apply the Observer pattern to notify users about important booking-related updates and changes and reviews
* Mediator Pattern for UI Components:
  + Implement the Mediator pattern to manage communication between different UI components, creating a cohesive and user-friendly interface.
* Builder Pattern for Car Listing Creation:
  + Utilize the Builder pattern to create car listing objects with various attributes, allowing for flexibility in car listing creation and customization.
* Proxy Pattern for Payment Integration:
  + Implement the Proxy pattern to handle secure communication and transactions between the application and the payment system.
* Chain of Responsibility for Password Recovery:
  + Apply the Chain of Responsibility pattern to create a secure process (using three security questions to build the chain) for recovering a forgotten password.
* **Notes:** 
  + **Provide detailed design of patterns used in the solution via class diagram with mapping of pattern classes to the actual application classes.**
  + **The developed code must be thoroughly commented and synchronized with the model.**
* **Submission**
  + **Submit One .zip file that contains the followings**
    - **Source code**
    - **A report that includes class diagrams and their descriptions, database schema and descriptions, user-interface screen shots and descriptions, references.**