

Project Title: **CLI-BASED TAXI BOOKING SYSTEM**

**GROUP MEMBERS:**

Uzair Ibrahim Roll#: 24k-0867

Mohsin Hassan Roll#: 24k-1000

Ali Khurram Roll#: 24k-1005

**1. Project Overview**

The CLI-Based Taxi Booking System is a command-line interface (CLI) application designed to allow users to book taxis from various predefined pickup points. The system enables user registration, login, and booking, while also offering functionality to update user information and view booking history. The system stores user data, booking records, and driver details in text files for persistent storage.

**2. Key Features**

* **Signup and Login:** Allows users to create an account and log in to the system using a phone number and password.
* **Fare Calculation:** Fare rates are based on the selected vehicle type.
* **Pickup Points:** Users can select from 16 different pickup points.
* **Driver Assignment:** A driver is assigned to each booking from a list of available drivers.
* **Booking Details:** After booking, details such as pickup point, drop-off point, distance, fare, and assigned driver are saved in a booking record.
* **Update Phone Number:** Users can update their phone number, which also updates the records in the booking history.
* **View Booking History:** Users can view details of their past bookings.

**3. System Design**

The system consists of several key components:

1. **User Management:**
   * Users can sign up with their name, phone number, and password.
   * On login, the phone number and password are checked against stored data to validate the user.
2. **Taxi Booking:**
   * Users can select a vehicle type and pickup point, then enter a drop-off point.
   * Based on these selections, the system calculates the distance between the pickup and drop-off points and computes the fare.
   * A driver is assigned to each booking from a list of available drivers.
3. **File Handling:**
   * User data, booking records, and driver information are stored in text files (users.txt, booking\_records.txt, drivers.txt).
   * The system handles file reading and writing operations to store and retrieve data.
4. **Fare Calculation:**
   * The fare is based on the selected vehicle type. A fare rate is associated with each vehicle type, with the rates defined in the program.
5. **Distance Calculation:**
   * The distance between pickup and drop-off points is stored in a predefined distance matrix. This matrix is used to calculate the distance and determine the fare.

**4. Technical Specifications**

* **Programming Language:** C
* **Libraries Used:**
  + stdio.h for input and output operations.
  + stdlib.h for system functions and memory allocation.
  + string.h for string manipulation functions.
  + time.h for time-related functions (optional in the current code, but could be used for timestamping).
* **Data Storage:**
  + **Users:** Stored in users.txt file, where each line contains the user's name, phone number, and password.
  + **Bookings**: Stored in booking\_records.txt file, containing details of each booking.
  + **Drivers:** Stored in drivers.txt file, with each line containing the driver's name.

**5. User Flow**

1. **Signup:**
   * The user enters their name, phone number (validated for length and numeric characters), and password.
   * The system checks if the phone number is already registered. If it is, the user is prompted to choose a different one.
   * If registration is successful, the user can proceed to login.
2. **Login:**
   * The user enters their phone number and password.
   * The system validates the credentials by comparing them with stored user data.
   * On successful login, the user is taken to the main menu for booking or managing bookings.
3. **Booking a Taxi:**
   * The user selects the vehicle type and the pickup and drop-off points.
   * The system calculates the distance based on the points selected and computes the fare.
   * A driver is assigned from the available list.
   * The booking record is saved, including the user's phone number, selected vehicle, pickup and drop-off points, fare, and driver information.
4. **Viewing Booking History:**
   * Users can view their previous bookings by querying their phone number. The system searches the booking records and displays details of past bookings.
5. **Updating Phone Number:**
   * Users can update their phone number, which also updates all associated booking records to reflect the new phone number.

**6. Challenges Encountered**

* **File Handling:** Handling file I/O operations correctly to ensure that records are saved and retrieved accurately without corruption was a challenge. Careful use of temporary files was necessary to handle updates (such as phone number changes).
* **Input Validation:** Ensuring correct format and length for phone numbers, passwords, and vehicle types was handled using loops and conditionals. Validating user input consistently was an essential part of the project.
* **User Experience:** Since the system operates via CLI, ensuring that the interface is user-friendly and the prompts are clear and informative was essential. This was managed by providing clear instructions and error messages.

**7. Future Enhancements**

* **Driver Availability:** Implement logic to manage and assign drivers based on availability, including a system to mark drivers as busy or free.
* **Pricing Models:** Introduce dynamic pricing models based on time of day or demand.
* **User Interface:** Enhance the CLI with more interactive features, such as color coding for different types of messages (error, success, etc.).
* **Authentication:** Use hashed passwords for enhanced security instead of plain-text storage.

**8. Conclusion**

This project demonstrates how a simple CLI-based system can be developed to handle basic taxi booking operations, including user authentication, booking management, and file-based storage. While there are areas for improvement, such as advanced features and a more robust interface, the system is functional and provides a clear structure for taxi booking in a real-world scenario.