Create a Taxi Service Application

This doc contains the requirements. Commit your progress throughout the test and keep commit messages readable. Test duration is **11:00-16:00**. Last commit before 16:00 will be evaluated. We will need a working application with as many requirements satisfied as possible and json inputs for each API. Use any local sql database of your choice and integrate Swagger. Follow best coding practices and all Dotnet conventions.

Evaluation Criteria

- Coding standards (folder structure, code quality, naming, formatting)
- AuthN and AuthZ
- Validations
- Error handling
- Data normalization
- High level design/solution
- Demo

Stretch/Bonus goals

- All APIs implementations.
- Unit Tests
- Realistic assumptions

For any confusion in the requirements assume the best possible case.

Taxi Service Actions -

- Rider
    - login/signup
    - can request a ride
    - Get the current ride
    - can cancel a ride
    - can rate a driver

- Driver
    - signup/login
    - availability
    - Get the current ride
    - can cancel ride
    - Can start a ride
    - Can end a ride
    - Can rate a rider

Task Definitions

1. **Rider - Sign Up**
   Create an API where a rider can sign up using Name, Email, Phone Number, and Password.

   Acceptance Criteria:
   - User should be created in the database.
   - If a user with the same email or phone number exists, return a "User already exists" error message.
   - The API should have proper validations for all fields.

2. **Rider/Driver - Login**
   Create an API where a rider or driver can sign in using Phone Number/Email, Password, and User Type (rider or driver) and receive a JWT token.

   Acceptance Criteria:
   - Only signed-up users should be able to log in.
   - Return a proper error message for "Invalid Username or Password."
   - Return a JWT token for further operations.
   - The API should have proper validations for all fields.

3. **Rider - Request a Ride**
   Create an API where a rider can request a ride by providing "Pickup Location", "Drop Location," and "Type of Ride."
   - Assume location is just a string and not latitude/longitude.
   - Assume any 3 types of ride like bike, car, auto etc

   Acceptance Criteria:
   - Only a logged-in rider should be able to access the API.
   - An available driver should be assigned automatically.
   - If a driver is available, the API should respond with RideId, Driver's Details, an OTP, and ride status.
   - If no driver is available, the API should respond with "No available driver found."
   - The API should have proper validations for all fields.

4. **Rider - Get Current Ride**
   Create an API where a rider can get the details of the current ride.

   Acceptance Criteria:
   - If there is an ongoing ride, the API should respond with the driver's details, type of ride, OTP (if the ride has not started), and ride status.
   - If there is no ongoing ride, the API should respond with "No ongoing ride available."

5. **Rider/Driver - Cancel Ride**

Create an API where a user can cancel the ride by passing the rideId.

Acceptance Criteria:

- The ride should only be canceled if the rideId belongs to the logged-in user and the ride has not started yet.
- If the ride has already started, respond with "Cannot cancel a ride that has already started."
- If the ride does not belong to the user or is not present in the database, respond with "Ride not found."

6. **Driver/Rider - End Ride**
   Create an API where a user can end the ride by passing the rideId.

   Acceptance Criteria:

   - The ride should only be ended if the rideId belongs to the logged-in user and the ride has started.
   - If the ride has not started yet, respond with "Cannot end a ride that has not started."
   - If the ride does not belong to the user or is not present in the database, respond with "Ride not found."

7. **Rider - Rate Driver**
   Create an API where a rider can rate the driver by passing the rideId and rating (1 to 5).

   Acceptance Criteria:

   - The rating should only be allowed if the rideId belongs to the logged-in user and the ride has ended.
   - If the ride has not ended, respond with "Cannot rate a ride that has not ended."
   - If the ride does not belong to the rider or is not present in the database, respond with "Ride not found."
   - The API should have proper validations for all fields.

8. **Driver - Sign Up**
   Create an API where a driver can sign up by passing the Name, Phone Number, Email, Password, and Vehicle Details (Vehicle Plate Number and Vehicle Type).

   Acceptance Criteria:

   - The driver should be saved in the database.
   - If a driver with the same email or phone number exists, return a "Driver already exists" error message.
   - There can be a rider and a driver with the same email and/or phone number.
   - The API should have proper validations for all fields.
   - A driver can be signed up with only one vehicle.

9. **Driver - Toggle Availability**
   Create an API where a driver’s availability can be toggled.

   Acceptance Criteria:
   - If the driver's availability is changed to "Available," they should be available to assign rides.
   - If the driver's availability is changed to "Unavailable," they should not be available for assigning rides.

10. **Driver - Get Current Ride**
    Create an API where a driver can get the details of the current ride.

    Acceptance Criteria:
    - If there is an ongoing ride, the API should respond with the rider’s details, pickup location, drop location, and ride status.
    - If there is no ongoing ride, the API should respond with "No ongoing ride available."

11. **Driver - Start Ride**
    Create an API where a driver can start a ride by passing the rideId and OTP.

    Acceptance Criteria:
    - The ride should only be started if:
        - The ride belongs to the driver.
        - The OTP is correct.
        - The ride status is "yet to start."
    - The API should fail in all other cases with a proper error message.

12. **Driver - Rate Rider**
    Create an API where a driver can rate the rider by passing the rideId and rating (1 to 5).

    Acceptance Criteria:
    - The rating should only be allowed if:
        - The ride belongs to the driver.
        - The ride has ended.
    - The API should fail in all other cases with a proper error message.
    - The API should have proper validations for all fields.