Passenger:

create_user(): This function takes a username and password as input and creates a new user in the database with the provided credentials. It establishes a connection to the MySQL database using the MySQL Connector/Python library, and uses SQL queries to insert the user data into the appropriate table in the database. This function is called when the user selects the option to create a new user in the menu-driven interface. create passenger(): This function takes passenger details such as name, phone number, and user ID as input and creates a new passenger in the database. It first checks if the provided user ID exists in the database by calling the check user() function. If the user ID is valid, it establishes a connection to the MySQL database and uses SQL queries to insert the passenger data into the appropriate table in the database. This function is called when the user selects the option to create a new passenger in the menu-driven interface. check user(): This function takes a username and password as input and checks if the provided credentials match a user in the database. It establishes a connection to the MySQL database, and uses SQL gueries to search for a matching user in the database. It returns a boolean value indicating whether the user exists or not. This function is called when the user selects the option to check user credentials in the menu-driven interface. check passenger(): This function takes a passenger ID as input and checks if the provided ID exists in the database. It establishes a connection to the MySQL database and uses SQL queries to search for a matching passenger in the database. It returns a boolean value indicating whether the passenger exists or not. This function is called when the user selects the option to check passenger details in the menu-driven interface. make_payment(): This function takes a passenger ID and booking ID as input and updates the payment status of the booking in the database. It first checks if the provided passenger ID and booking ID are valid by calling the check passenger() function. If the passenger and booking ID are valid, it establishes a connection to the MySQL database and uses SQL queries to update the payment status of the booking in the database. This function is called when the user selects the option to make a payment in the menu-driven interface. check booking status(): This function takes a passenger ID as input and retrieves the booking status from the database. It first checks if the provided passenger ID is valid by calling the check passenger() function. If the passenger ID is valid, it establishes a connection to the MySQL database and uses SQL queries to retrieve the booking status from the database. This function is called when the user selects the option to check booking status in the menu-driven interface. main(): This function serves as the entry point of the program and implements the menu-driven interface. It displays a menu to the user with options to create a new user, create a new passenger, check user credentials, check passenger details, make a payment, check booking status, or exit the program. It takes user input to select an option and calls the appropriate function based on the user's choice. It loops until the user chooses to exit the program.

Driver:

create_user(conn, user_id, password, category): This function creates a new user in the database with the given user ID, password, and category (in this case, "driver"). It inserts the user details into the "users" table in the MySQL database. create_car(conn, car_id, model): This function creates a new vehicle in the database with the given car ID and model. It inserts the vehicle details into the "vehicles" table in the MySQL database. get_passenger_details(conn,

passenger_id): This function retrieves passenger details from the database based on the given passenger ID. It selects the passenger details from the "passengers" table in the MySQL database. login(conn, user id, password): This function checks if a user with the given user ID and password exists in the database and retrieves driver details (such as driver ID, name, license number, and rating) from the database based on the user ID and password. It performs a JOIN operation on the "users" and "drivers" tables in the MySQL database. see bookings(conn): This function retrieves all bookings from the database and displays them with their details, including passenger name. It performs a JOIN operation on the "bookings" and "passengers" tables in the MySQL database. lessgo(conn, pass id, driver id): This function creates a new trip in the database with the given passenger ID, driver ID, and other details such as trip ID, status, date, drop location, and pickup location. It inserts the trip details into the "trips" table in the MySQL database and deletes the corresponding booking entry from the "bookings" table. tripdone(conn, driverid): This function updates the status of a completed trip in the database to "completed" for the given driver ID. It also inserts a payment entry into the "payments" table with random payment ID, payment type (cash), payment amount, payment status ("in progress"), and trip ID. create driver(Driver id, name, license, date of birth, rating, location, status, car id): This function creates a new driver in the database with the given driver details, such as driver ID, name, license number, date of birth, rating, location, status, and associated car ID. It inserts the driver details into the "drivers" table in the MySQL database. The code also includes a while loop that displays a menu for the driver terminal with options to sign up, log in, and exit. The user can choose the desired option and perform corresponding operations based on the input.