Project Title: Chrive

1. **Project Overview:**

   Chrive is a luxury ride ordering website. Users use the website to make a reservation for a luxury ride. They choose a date and a time to be picked up in a luxury vehicle. It will help companies and VIPs make a reservation in advance without the worry if the ride will be late as they are making a reservation some time in advance. It also helps those who visit NYC to book a ride before their arrival.

2. **Project Timeline:**
   - Day 1: Homepage, Landing Page, User Authentication, and Database Setup
     - Create the homepage and landing page.
     - Set up the basic HTML structure.
     - Add necessary content and images.
     - Apply initial styling using CSS.
     - Develop the login, signup, and logout functionality.
     - Implement user authentication using PHP and MySQL.
     - Design and create the database structure.
     - Create a database table 'user' with relevant columns.
     - Create a database table trips_history' with appropriate columns.
     - Set up secure password storage using password hashing.
     - Establish a foreign key relationship with the 'user' table.
     - Ensure data integrity and normalization.
   - Day 2: Frontend-Backend Connection, Trip Booking, Trip History, and User Profile
     - Implement communication between the user interface and server.

    - Test and validate data transmission between the two.
    - Create the 'New Trip' page.
    - Design a user-friendly form for booking a new trip.
    - Implement form validation on both client and server sides using JavaScript and php to ensure all required fields are filled.
    - Develop the 'Trips History' page.
    - Retrieve and display a user's past trip records from the database.
    - Format and present the trip data in a visually appealing manner.
    - Create the 'Profile' page.
    - Allow users to view and edit their profile information.
    - Implement functionality to update user details and save changes to the database.

1. **Project Architecture:**
    - Frontend:
        - Homepage and Landing Page: Users will land on the homepage, which will provide an introduction to the luxury car service and navigation options.
        - User Authentication: Users can access the login and signup pages to create an account or log in with existing credentials.
        - New Trip Booking: After logging in, users can navigate to the 'New Trip' page to fill out a form for booking a new trip.
        - Trips History: The 'Trips History' page will display a list of past trips for the logged-in user.

    - User Profile: Logged-in users can view and edit their profile information on the 'Profile' page.

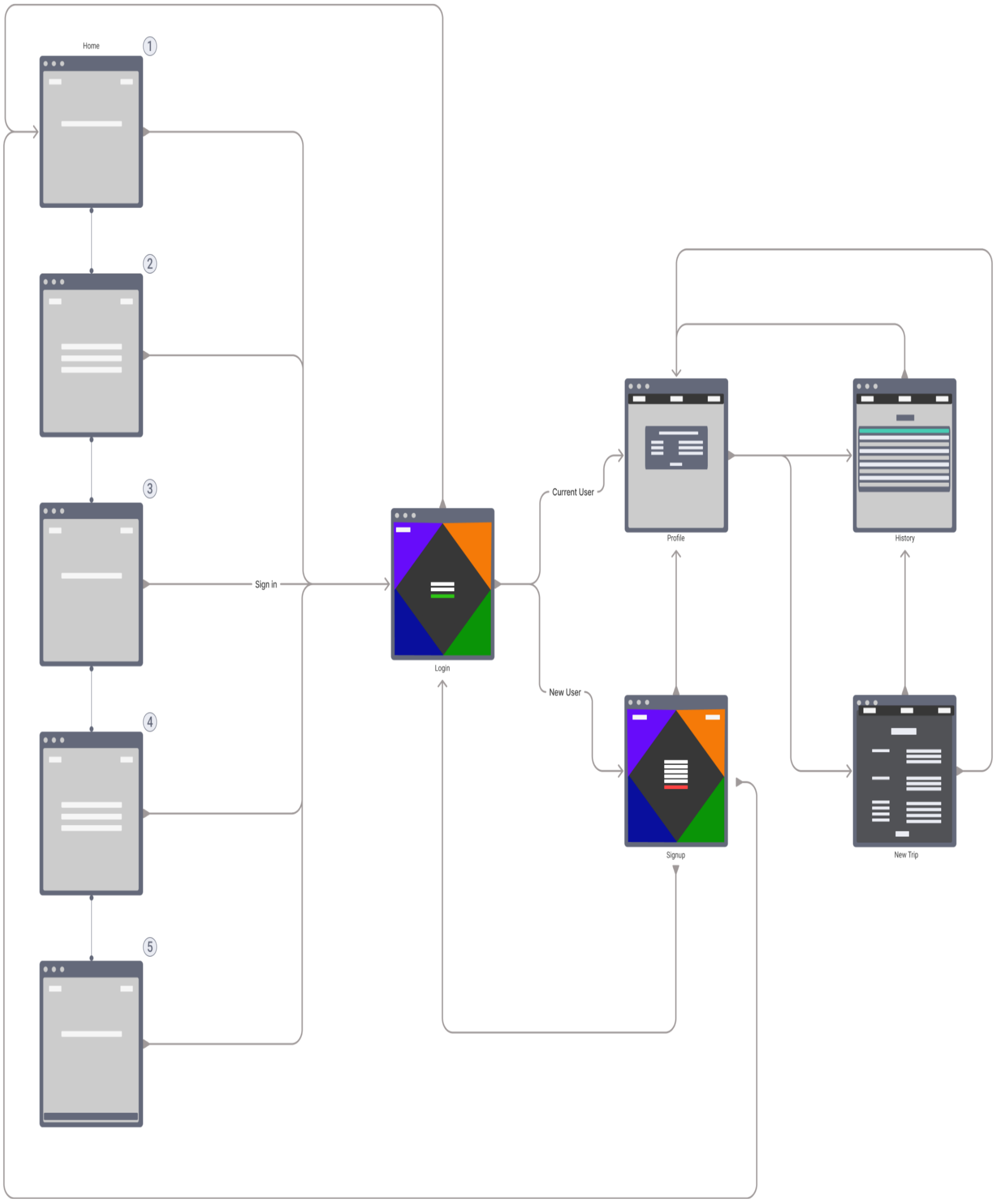
- Backend:
    - User Authentication: The backend will handle user authentication using PHP and MySQL. It will verify user credentials and provide access tokens upon successful login.
    - Database Interaction: PHP scripts will manage interactions with the MySQL database. They will retrieve, update, and store user data and trip history.
    - Form Validation: JavaScript will be used for client-side form validation to ensure data consistency before submitting to the server.

- Database:
    - User Table: The 'user' table will store user information, including username, password (hashed), and email.
    - Trip History Table: The 'trip_history' table will store trip records, including user ID, pick up location, drop off location, trip date, trip time, number of passengers, and phone number.

- Navigation:
    - Users will navigate through the application by clicking on links or buttons to move between different pages.
    - After logging in, users can access the 'New Trip,' 'Trips History,' and 'Profile' pages and logout from the navigation menu.
    - Form submissions will trigger backend API requests to store data in the database.

    - Error messages will be displayed to users based on the outcome of backend interactions.

- Data Flow:
    - User interacts with the frontend, filling out forms or clicking on links.
    - Frontend validates user inputs and sends requests to backend APIs.
    - Backend processes requests, communicates with the database, and sends back responses to the frontend.
    - Frontend updates the UI based on the response received from the backend.
    - Database stores user details and trip records, which can be retrieved by the backend for display in the frontend.



Home
1
2
3
4
5
Sign in
Login
Current User
New User
Profile
History
Signup
New Trip

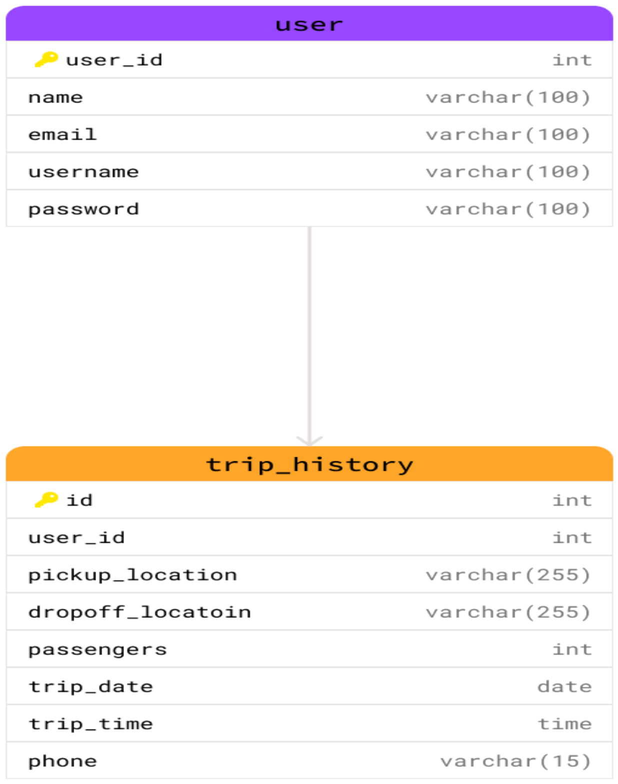
2. **User Interface (UI) Design**

The website will feature a clean and user-friendly UI design to provide a seamless experience. The layout will be responsive, adapting to different screen sizes. The color scheme will combine soothing colors such as yellow, green, and blue to evoke a sense of trust and relaxation. The typography will use the "Oswald" font for headings and "Raleway" for content, ensuring readability. Interactive elements include hover effects on buttons, smooth transitions, animations, and form validation.

3. **Database Design**

- Users Table:
    - user_id (Primary Key)
    - username
    - password (hashed)
    - email

- Trips History Table:
    - trip_id (Primary Key)
    - user_id (Foreign Key referencing User Table)
    - pickup_location
    - dropoff_location
    - trip_date
    - trip_time
    - passengers
    - Phone



4. Functionality & Features

- Functionality & Features:
  - User Registration and Login:
  - Purpose: Allow users to create accounts and log in securely.
  - Implementation: PHP for user authentication, password hashing, and session management.
- New Trip Booking:
  - Purpose: Enable users to book new trips by filling out a form.
  - Implementation: HTML forms, PHP for form handling, JavaScript for client-side validation.

- Trips History:
    - Purpose: Display a list of past trips for each user.
    - Implementation: PHP for querying the database based on user ID.
- User Profile:
    - Purpose: Allow users to view and edit their profile information.
    - Implementation: PHP for retrieving and updating user details.

5. **Technology Stack**
    - Frontend: HTML, CSS, JavaScript
    - Backend: PHP 8.2.8
    - Database: MySQL 11.0.2
    - Additional Tools: Visual Studio Code, Terminal
6. **Challenges & Mitigation**
    - Security: Ensuring secure user authentication and data handling.
    - Mitigation: Using password hashing, prepared statements, and session management in PHP.
    - Responsive Design: Creating a consistent UI across different devices.
    - Mitigation: Utilizing CSS media queries for responsive layouts and testing on various screen sizes.
    - Data Validation: Preventing invalid or malicious data from being stored.
    - Mitigation: Implementing client-side and server-side validation for forms.
7. **References**
    - W3Schools (https://www.w3schools.com/).
    - Stack Overflow (https://stackoverflow.com/) for troubleshooting.