



Instructions

Implement the solutions for each problem in their own files within a GitHub repository.

Create a GitHub repository and organize the solutions as separate folders, with clear folder names.

Include a README.md file in the repository that provides an overview of the test, explains the purpose of each problem, and describes the solutions.

Commit and push your changes to the GitHub repository.

Share the link to the GitHub repository as part of your submission.

Additionally, as part of your submission, please provide a sample project that showcases your understanding of software engineering principles.

Backend (Laravel) Assignment:

You are tasked with creating a simple RESTful API using Laravel. The API should allow users to perform CRUD operations on a resource called "clients." Follow the steps below:

1. Set up a new Laravel project and create a "Client" model with fields like name, gender, dob, marital_status, and approval_status.
2. Implement the necessary routes and controller methods to handle the following operations:
 - Create a new client
 - Retrieve a specific client by ID
 - Update an existing client
 - Delete a client
 - Approve and withdraw client approval_status
3. Use middleware to handle authentication for the API. The API should only allow authenticated users to access the CRUD operations.
4. Implement validation middleware to ensure that the incoming data meets the required criteria (e.g., name should be a string, dob needs to be a date, etc.).

Frontend (Angular) Assignment:

You are tasked with creating a frontend interface using Angular to interact with the RESTful API created in the backend assignment. Follow the steps below:

1. Create an Angular project and set up necessary dependencies.
2. Develop a component to display a table of clients.
3. Fetch the list of clients from the RESTful API created using Laravel.
4. Display the client details in a table format with columns for name, gender, dob, marital status, and approval status.
5. Include a button or action column in the table to approve a client.
6. Implement functionality in Angular to send a request to the Laravel API to update the approval status of a client to true when the approve button is clicked.
7. Handle validation of data received from the Laravel API on the frontend to ensure consistency and prevent errors in displaying client details.