

Step 1: Project Setup

1. **Create a new Laravel project** using Composer.
2. Use MySQL or PostgreSQL for your project and setup accordingly.
3. Run your project from localhost (**N.B: not from php artisan serve command**)
4. Create a public repository and push your first commit with proper message

Step 2: Login

1. **Set up authentication** using Laravel Breeze or Laravel Sanctum.
2. **Implement login & registration functionality** with hashed passwords.
3. **Validate user inputs** (email, password, etc.) using **Form Request Validation**.
4. Push your second commit with proper message

Step 3: Role Based Access Control(RBAC)

1. **Create roles and permissions tables** using migrations.
2. **Seed database with roles:** Admin and User.
3. **Assign roles to users** (use a pivot table or direct assignment).
4. **Implement middleware** to restrict access based on roles.
5. **Ensure only Admin can access a protected route** (e.g., /admin/dashboard).
6. The above procedure should be done by using **Spatie Laravel-Permission package**.
7. Push your third commit with proper message

Step 4: Simple Module

1. **Create a Product model, migration, and factory.**
2. **Implement CRUD operations** for products (Create, Read, Update, Delete) Using Eloquent ORM.
3. **Secure API using Laravel Sanctum.**
4. **Ensure only authenticated users** can create, update, and delete products.
5. **Implement pagination** in the GET /products API.
6. **Use Form Request Validation** to validate incoming data for product creation and updating (e.g., StoreProductRequest and UpdateProductRequest).
7. Push your fourth commit with proper message

React.js Frontend Assignment

Objective:

Develop a React.js frontend application that integrates with a Laravel backend, implementing authentication, role-based access control (RBAC), and product management functionalities.

Task Requirements:

Step 1: Project Setup

1. Initialize a new React.js project using **Vite** or **Create React App**.
2. Install required dependencies:
 - **React Router** (react-router-dom) for navigation.
 - **Axios** for API communication.
 - **Tailwind CSS** for styling.
3. Set up an .env file to store API base URL securely.
4. Push the initial commit to a **public GitHub repository** with a proper commit message.

Step 2: Authentication System

1. Implement **login and registration** pages with form validation.
2. Authenticate users by sending API requests to the Laravel backend.
3. Store the authentication token securely in **local storage** after successful login.
4. Redirect authenticated users to the **dashboard**.
5. Push your second commit with a proper message.

Step 3: Role-Based Access Control (RBAC)

1. Retrieve the **user role** from the backend after login.
2. Implement **protected routes**:
 - Only authenticated users should access the dashboard.
 - Only Admins should access the **Admin Panel**.
3. Conditionally display UI elements based on user roles (e.g., Admin options should only be visible to Admin users).
4. Use **React Context API** or state to manage authentication and roles.
5. Push your third commit with a proper message.

Step 4: Product Management Module

1. Implement a **Product Listing Page** that fetches and displays paginated products.
2. Allow authenticated users to:
 - **Create new products** using a form.
 - **Update existing products** (Only for authorized users).
 - **Delete products** (Only for authorized users).
3. Implement **form validation** for product creation and updating.
4. Display **error messages** for invalid inputs.
5. Push your fourth commit with a proper message.

Submission:

Submit your GitHub repository link and database(also attach login credential) within 48 hours from the starting time.