

Assignment 3

Building Dynamic Web Application with Laravel PHP

(Practical Group Task)

Description:

Each group will work on creating a dynamic web application using the Laravel PHP framework. The application must include features such as MVC architecture, routing, Blade templates, middleware, and Eloquent ORM for database interaction.

This assignment builds upon your previous work but now requires you to implement the case study in Laravel, showcasing its core capabilities.

Assignment Tasks:

Task 1: Routing, Controllers, and Middleware	(25 Marks)	[CLO 4, 5]
---	-------------------	-------------------

- **Routes:** Define routes for key features of your case study (e.g., displaying services, user profiles, or products).
- **Controllers:**
 1. Create controllers for handling the business logic of your application.
 2. Organize controller methods for different tasks (e.g., `index`, `store`, `update`).

Example: Use a `ProductController` with methods to handle CRUD operations for products.
- **Middleware:** Create and apply middleware to check user authentication and restrict access to certain routes.

Task 2: Blade Templates and Layouts	(25 Marks)	[CLO 3, 5]
--	-------------------	-------------------

- **Master Layout:**
 - Create a master layout using Blade templates.
 - Include sections for a header, footer, and sidebar.
- **Blade Directives:**
 - Use Blade directives (`@yield`, `@section`, `@include`) for organizing content.
- **Dynamic Pages:**
 - Create pages for your case study (e.g., “About Us”, “Services”, or “Products”) and extend the master layout.

Example: A page listing services dynamically fetched from the database.

Task 3: Eloquent ORM and Database Operations	(25 Marks)	[CLO 4, 5]
---	-------------------	-------------------

- **Database Interaction:**
 1. Use Eloquent models to interact with the database.
 2. Create a migration for a table related to your case study (e.g., “users”, “products”, or “services”).
- **CRUD Operations:**

Implement the following operations:

 1. **Create:** Insert a new record into the database.
 2. **Read:** Display records dynamically on a Blade view.
 3. **Update:** Allow editing of existing records.
 4. **Delete:** Implement record deletion with a confirmation.

Example: A CRUD system for managing a list of services/products.

Task 4*:Additional Task for Groups with 4 Members: Forms and Validation (25 Marks)
[CLO 3, 5]

- Forms:
 1. Design a form for user input using Blade and Laravel's form handling capabilities.
 2. Include fields relevant to your case study (e.g., user registration, product creation).
- Validation:
 1. Validate the form input on the server side using Laravel's validation rules.

Example: Ensure a name field is required, and an email field is correctly formatted. Display validation error messages dynamically.

Mapping to CLOs:

CLO 3: Build Static and Dynamic Websites and Applications

- Tasks focus on building dynamic websites by using JavaScript for interactive features like DOM manipulation, dynamic styling, and form validation.

CLO 4: Produce Solutions using Web-Oriented Programming Constructs

- Students solve programming challenges using JavaScript constructs, creating dynamic content, managing user inputs, and enhancing website functionality.

CLO 5: Perform Effectively as a Member of a Team

- Team members collaborate on designing, coding, and testing interactive features, ensuring effective teamwork in building and debugging the website.

Submission Guidelines:

- Ensure that:
 - All routes are properly defined and functional.
 - Blade templates are clean and organized.
 - Database operations are thoroughly tested.
- Add the name/roll numbers of all group members and link to GitHub Repository in a README file.
- Zip your assignment folder, name it `Assignment_3_GroupNumber.zip`, and submit via LMS.
- Only one submission per group is required.

Marking Rubrics (For groups of 3 members, Total Marks: 75)

Sr. No.	Criteria	Excellent (25 Marks)	Good (20 Marks)	Satisfactory (15 Marks)	Needs Improvement (10 Marks)	Marks
1.	Routing, Controllers, & Middleware	Fully functional	Mostly functional	Basic routing or issues	Incorrect implementation	25
2.	Blade Templates	Organized and reusable	Mostly functional	Basic layout, minor issues	Poor organization	25
3.	Eloquent ORM (CRUD)	All operations work	Minor issues	Basic CRUD, partial bugs	Incomplete implementation	25
Additional Marks (For groups of 4 members, Total Marks: 100)						
4.	Forms & Validation	Complete and error-free	Mostly functional	Basic validation, some bugs	Minimal validation	25

Good Luck (☺)