

## Overview

The candidate will develop a Task Management System within approximately 1-2 weeks. The development of the demo application is expected to be completed within 16-24 hours. The goal is to evaluate the candidate's skills in Laravel, including CRUD operations, user authentication, role-based access control, image upload functionality, and AJAX for dynamic interactions.

## Examination Criteria

### 1. Technical Skill:

- Ability to design the application structure and codebase in a maintainable and modular way, suitable for team collaboration.
- Develop a functional UI that adapts to different display ratios.
- Implement AJAX for dynamic interactions such as loading and updating tasks without full page reloads.

### 2. Service Direction Skill:

- Ability to implement features that add value from a user perspective, ensuring usability and practicality.

### 3. Project Management Skill:

- Ability to prioritize tasks and manage development effectively within the given timeframe.
- Note: The visual appeal of the design is secondary; however, the ability to implement a functional UI and features is critical.

## Service Concept (For This Assignment Purpose Only)

Develop a Task Management Application with the following features:

### 1. Role Permissions:

- **Admin:**
  - Manage all tasks: view, create, edit, and delete any tasks.
  - Manage users: view, create, edit, and delete user accounts.
  - **Admin Dashboard:** Display the highest solver and count each solving time.
- **User:**
  - Manage own tasks: view, create, edit, and delete tasks assigned to them or created by them. count each solving time.

## 2. Features:

### ○ **CRUD Operations:**

- Implement Create, Read, Update, and Delete functionalities for tasks.
- Each task should include:
  - `title` (string)
  - `description` (text)
  - `status` (enum: 'Pending', 'In Progress', 'Completed')
  - `due_date` (date)
  - `assigned_to` (foreign key to `users` table)
  - `image` (nullable file, URL to the uploaded image)
  - Add more columns if you need

- **Task List:** Display tasks in a list format with filtering options (e.g., by status, due date).
- **Task Details:** View detailed information about a task, including the uploaded image.

## 3. Image Upload:

- Implement functionality to upload an image when creating or updating a task.
  - Store the uploaded image in the `storage/app/public/images` directory.
- Validate the image file (e.g., size, type) and ensure it is correctly displayed in task details.

## 4. User Authentication:

- Implement user registration and login functionality using Laravel's built-in authentication system.
- Ensure users can create, view, and manage their tasks.

## 5. AJAX Integration:

- Use AJAX to handle dynamic interactions such as:
  - Loading task lists without refreshing the page.
  - Updating task details or status without a full page reload.
  - Handling image uploads and form submissions asynchronously.

## 6. Admin Dashboard Enhancements:

- **Highest Solver:**
  - Display the user with the highest number of completed tasks.
- **Solving Time:**
  - Track and display the time taken to complete each task.
  - Aggregate and show statistics on average solving time per user.

## 7. Database Seeder:

- Create seeders to populate the `tasks` and `users` tables with sample data for testing purposes.

## Platform

- Laravel PHP

## UI Framework

- Bootstrap
- Any other framework you are familiar with

## Deployment

- Make the code available on Git and provide a public Git repository link.

## Application Requirements

### Technical Elements:

- **User Management:** Implement new registration and login functionalities.
- **Database Access:** Ensure the application can save and load data from the database.
- **UI Creation:** Develop the user interface according to the role-based functionality. ●  
**AJAX:** Implement dynamic interactions for better user experience.