APPLICANT CAN PICK ANY OF THE TWO SCENARIOS BELOW

Scenario 1: Project Management System

A company ABC wants to develop a web-based project management system using Laravel. The application should allow project managers to track tasks, assign them to team members, and monitor progress.

Task 1: Project and Task Management

- **Project Creation:** Implement a form to create a new project with fields for project name, description, start date, and end date. Add server-side validation to ensure that the start date is before the end date.

- **Task Management:** Implement CRUD (Create, Read, Update, Delete) functionality for tasks within a project. Each task should have a name, description, start date, end date, and assignee.

- **Task Assignment:** Implement functionality to assign tasks to users. Ensure that only users with the role of "Project Manager" can assign tasks.

Task 2: Dashboard and Reporting

- **Dashboard Creation:** Create a dashboard that displays an overview of all projects, showing the number of tasks completed and the percentage of progress for each project.

- **Overdue Tasks:** Implement a feature that highlights overdue tasks on the dashboard. Provide a way to filter tasks by their status (e.g., completed, in progress, overdue).

- **Report Generation:** Develop a feature to generate PDF reports summarizing the status of each project, including completed tasks, pending tasks, and overdue tasks

Task 3: Notifications and Task Dependencies

- **Email Notifications:** Set up email notifications to alert users when they are assigned a task, when a task is nearing its deadline, or when a task is overdue. Use Laravel's built-in mail capabilities.

- **Task Dependencies:** Implement a system to manage task dependencies where one task cannot start until another is completed. Automatically update the status of dependent tasks based on the completion of their preceding tasks.

Task 4: Access Control

- **Role-Based Access Control:** Implement role-based access control using Laravel's authorization features. Ensure that different roles (e.g., Admin, Project Manager, Team Member) have appropriate permissions to view, edit, and delete projects and tasks.

- **Middleware Implementation:** Create middleware to enforce that only authorized users can access specific routes and functionalities in the application.

Scenario 2: Employee Task Reporting System

A company XYZ wants to develop a Laravel-based web application for employees to submit and manage their daily task reports. The application should provide a secure environment for data entry and reporting.

Task 1: Task Submission

- **Form Creation:** Develop a form for employees to submit their daily task reports with fields for Date, Employee name, Department, Task details, and Hours worked. Implement server-side validation to ensure all fields are filled out correctly.

- **Task Editing:** Allow employees to edit their submitted task reports within 24 hours of submission. After 24 hours, the report should be locked for editing.

Task 2: Dashboard and Filtering

- **Task Dashboard:** Create a dashboard that displays a summary of all tasks submitted for the day, showing details such as Employee name, Department, and Hours worked.

- **Filtering:** Implement a filtering feature that allows users to filter tasks by department, employee name, or date range.

Task 3: Data Export

- **Export to PDF and Excel:** Implement a feature that allows users to export the daily task reports to PDF and Excel formats.

- **Scheduled Reports:** Set up a feature where daily reports are automatically generated and emailed to department heads at the end of each day.

Task 4: User Authentication and Role Management

- **Authentication:** Implement user authentication using Laravel's built-in authentication scaffolding. Ensure that users must be logged in to submit or view task reports.

- **Role Management:** Develop a simple role management system where users can be assigned roles such as "Employee," "Manager," or "Admin." Use Laravel's authorization features to control access to different parts of the application based on roles.

Task 5: Approval Workflow

- **Task Approval:** Implement a simple approval workflow where submitted tasks must be approved by a manager before being finalized. Include the ability for managers to approve or reject tasks with comments.