Advanced Web Programming

Course Project – Spring 2025

Dr. Amjad AbuHassan

Tasks Management System (SPA)

Application Description:

The Task Management System is a web-based application that facilitates the organization, tracking, and management of tasks and projects. It is designed to enhance productivity and collaboration among users, including students and administrators. The project will be developed in three phases, each focusing on different aspects of the application.

Key Features:

1. **User Authentication:** Users can sign up and sign in to the application, with options to stay signed in for convenience.
2. **User Roles:** The application supports different user roles, such as students and administrators, allowing for tailored functionalities based on user type.
3. **Task Management:** Users can create, view, edit, and delete tasks. Tasks can be assigned to specific users, and their status can be tracked (e.g., Pending, In Progress, .. etc).
4. **Student Information:** For users who are students, there is an option to input their university ID, which can be used for identification and task assignment.
5. **Chat Functionality:** The application includes a chat feature that allows administrators to communicate with students, facilitating support and collaboration.
6. **Responsive Design:** The user interface is designed to be user-friendly and responsive, ensuring accessibility across various devices.
7. **Date and Time Display:** The application displays the current date and time, enhancing user awareness of deadlines and schedules.

8. **Dark Theme:** The application features a dark theme for a modern look and improved usability in low-light environments.
9. **SPA:** The application must be single page application

Target Users:

1. **Students:** To manage their tasks, communicate with administrators, and track their academic responsibilities.
2. **Administrators:** To oversee task assignments, manage student information, and facilitate communication.

Project Delivery Milestones

Phase 1: Front-End Development with Fake Data (use local storage)

Deadline: 14/3/2025

Objectives:

- Develop the front-end interface using HTML, CSS (LESS), and JavaScript.
- Create a responsive layout for the dashboard
- Implement fake data to simulate application.
- Focus on minimal CSS styling to ensure a clean and functional UI.

Phase 2: React + Tailwind CSS

Objectives:

- Refactor the front-end code to use React for building the user interface.
- Integrate Tailwind CSS for styling to enhance the visual appeal and responsiveness of the application.
- Implement state management to handle village data dynamically.
- Create reusable components for projects cards, modals, forms, etc.
- Ensure that the application is fully functional with the new React architecture, maintaining the features from Phase 1.

Phase 3: Node.js Backend and Integration (use mysql or mongodb)

Deadline: 16/5/2025

Objectives:

- Develop a Node.js backend to handle data storage and retrieval.
- Set up a **GraphQl** to manage village data.
- Integrate the front-end application with the backend to fetch and display real data.
- Implement user authentication and authorization for admin functionalities.
- Ensure proper error handling and validation for all API endpoints.
- Conduct testing to ensure the application works seamlessly across the FE and BE.

Final Deliverables

- A fully functional application with a user-friendly interface.
- Documentation detailing the project structure, setup instructions, and usage.
- A presentation demonstrating the features and functionalities of the application.

Notes

- Students are encouraged to collaborate and communicate effectively throughout the project phases.
- Regular check-ins and progress updates should be scheduled to ensure milestones are met on time.