BAHRIA UNIVERSITY ISLAMABAD

Employee Task Management System

GitHub Repository:

https://github.com/afnankhanniazi/task-management-system

1. Group Information

- **Group Member 1:** Afnan Abbas Khan 01-135231-006
- **Group Member 2:** Furqan Khalid Butt 01-135231-024
- **Group Member 1:** Bilal Azhar 01-135231-020

Table of Contents:

2. Project Title and Overview

Title: Employee Task Management System

Overview:

The Employee Task Management System is a web-based platform designed to help organizations manage and track employee tasks efficiently. It allows administrators to create, assign, and monitor tasks while providing employees with an interface to view, update, and complete their assigned work.

Organizations often struggle with poor task management, leading to inefficiencies, missed deadlines, and lack of accountability. This system aims to address these issues by providing a structured platform that organizes tasks, tracks progress, and ensures communication between team members. Employees will be able to see their assigned tasks, update statuses, and receive reminders for deadlines, while managers will have a clear view of work progress.

This project is crucial for improving workflow efficiency, ensuring task accountability, and enhancing team collaboration. It is relevant to the Web Systems and Technologies course as it integrates multiple technologies, including frontend design, backend logic, and database management.

Project Objectives:

- Automate task assignment and tracking to reduce manual workload.
- Improve communication between employees and administrators through real-time updates.
- Provide an intuitive and interactive interface for managing tasks.
- Ensure secure user authentication and role-based access to protect sensitive data.
- Generate reports to analyze employee productivity and overall task completion rates.

3. Functional and Non-Functional Requirements

Functional Requirements

These describe the system's capabilities and what it will do.

ID	Functional Requirement	Description
FR1	User Authentication	Employees and admins can register and log in securely.
FR2	Task Creation & Assignment	Admins can create tasks and assign them to employees.
FR3	Task Status Management	Employees can update task progress (Pending, In Progress, Completed).
FR4	Notification System	Users receive alerts for new tasks and approaching deadlines.
FR5	Task Filtering & Search	Employees can filter tasks based on status, priority, or deadline.

FR6	User Role Management	Admins have full control, while employees have restricted access.
FR7	Task Report Generation	Generates detailed reports on task completion and employee performance.

Non-Functional Requirements

These define the quality attributes of the system, such as security, performance, and usability.

ID	Non-Functional Requirement	Description
NFR1	Security	Data is protected using **hashed passwords** and **secure database queries**.
NFR2	Performance	The system should handle **at least 50 simultaneous users** without lag.
NFR3	Usability	The UI should be **user-friendly, responsive, and accessible** on different devices.
NFR4	Scalability	The database and backend should support future expansion and additional features.
NFR5	Availability	The system should have **99.9% uptime** with minimal downtime for maintenance.
NFR6	Maintainability	The system should allow easy updates and modifications.

4. Technical Requirements

The Employee Task Management System requires a combination of **frontend, backend, and database technologies** to function effectively. Below are the core technical components involved:

Frontend Technologies:

The front end of the system is responsible for the **user interface and user experience (UI/UX)**. It will be designed using:

- **HTML5**: Used to structure the web pages and create a well-defined layout.
- **CSS3 & Bootstrap**: Responsible for styling the website, making it responsive and visually appealing.
- **JavaScript**: To implement interactive elements such as dynamic updates, form validation, and search functionality.

Backend Technologies:

The backend is the logic that processes data and manages system functionality. The following technologies will be used:

- **PHP**: Handles all server-side operations, such as processing user requests, business logic, and database queries.

Database Management:

A robust database system is required to store and retrieve user and task-related data efficiently. We will use:

- **MySQL**: A relational database to store users, tasks, and system logs.
- **phpMyAdmin**: A tool to manage and manipulate database tables efficiently.

Development Tools:

To ensure smooth development, the following tools will be used:

- **VS Code**: An IDE for writing and debugging code.
- **XAMPP**: A local server for testing PHP and MySQL applications.

5. Development Model

The Agile Development Model is an excellent choice for your project. It really does provide that perfect balance of structure and flexibility that most modern projects need. In today's fast-paced development environment, the ability to adapt quickly while maintaining progress is invaluable, and Agile delivers precisely this capability.

Your reasoning for choosing Agile makes perfect sense:

The incremental development approach will help you manage complexity by breaking the project into manageable pieces. This way, you'll have working modules throughout the development process rather than waiting until the end to see if everything works together. Each sprint delivers something tangible that stakeholders can review, providing clear evidence of progress and value creation.

The flexibility aspect is perhaps one of Agile's greatest strengths. Requirements often evolve during development, and Agile's iterative nature accommodates these changes much better than traditional waterfall methods. When new insights emerge or market conditions shift, your team won't be locked into an outdated plan but can pivot as needed while still maintaining overall project momentum.

Incorporating user feedback early and often is crucial for developing something that truly meets user needs. Those regular testing cycles will save you from discovering major usability issues late in development when they're much more expensive to fix. This continuous feedback loop ensures the final product will actually solve the problems it's meant to address, rather than just meeting technical specifications on paper.

The collaborative nature of Agile will keep your team aligned and working efficiently. The clear role distribution and regular communication that Agile emphasizes helps prevent silos and ensures everyone is moving in the same direction. Daily standups, sprint planning, and retrospectives create multiple touchpoints for coordination and problem-solving, fostering a true team environment.

Risk mitigation is another significant benefit of the Agile approach. By delivering working software in short iterations, you reduce the risk of project failure as problems are identified and addressed early in the development cycle. This creates opportunities to course-correct before small issues become major obstacles.

Finally, Agile provides better visibility into project progress through burndown charts, task boards, and other reporting mechanisms. This transparency helps all stakeholders understand exactly where things stand at any given moment, building trust and enabling better decision-making throughout the project lifecycle.

Have you determined which specific Agile framework you'll be implementing? Whether it's Scrum with its defined ceremonies and roles, Kanban with its visualization of workflow, or a hybrid approach tailored to your organization's needs, selecting the right methodology will help you maximize the benefits of your Agile implementation.

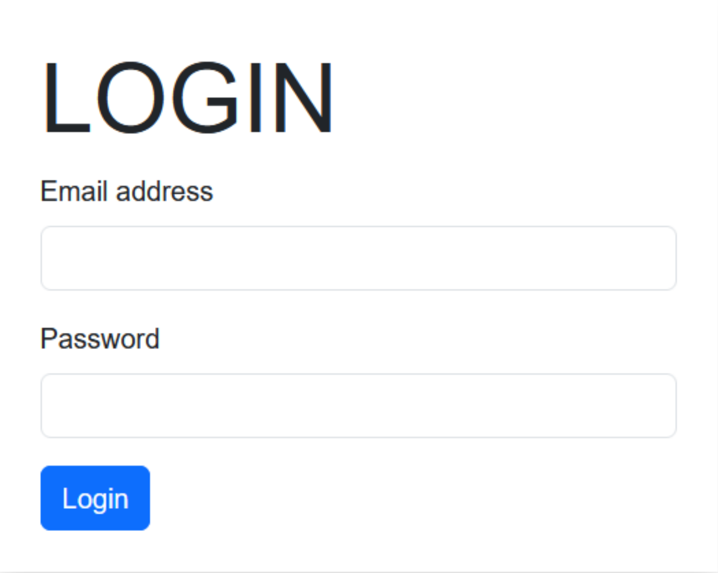
6. Project Timeline

The project will be completed in **5 weeks**, divided into structured phases to ensure efficient development.

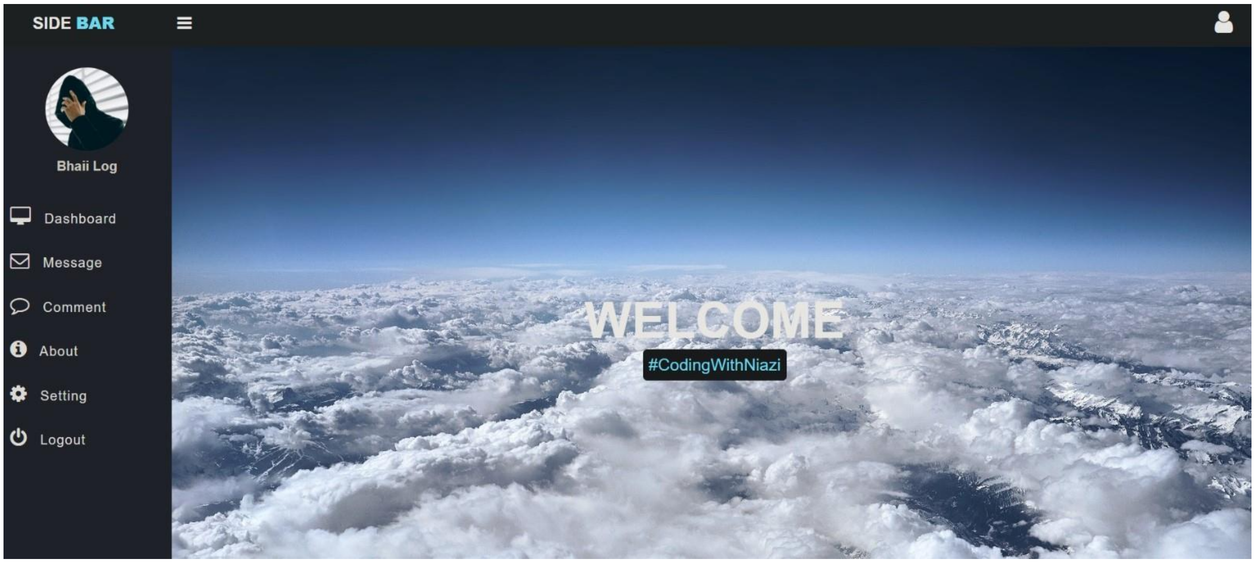
Week	**Task**	**Description**
Week 1	Frontend Development	Design UI with HTML, CSS, and Bootstrap. Implement JavaScript for interactive elements. Ensure the website is responsive. Begin initial prototyping of the UI.
Week 2	Database & Backend Setup	Create MySQL database and tables. Develop PHP backend for handling user authentication and CRUD operations. Ensure database connectivity with PHP. Conduct preliminary testing on database connections.
Week 3	Core Functionalities Implementation	Implement user authentication, task creation, task status updates, and notification system. Add search and filtering options for better task management. Conduct real-time testing to ensure smooth user interactions.
Week 4	Testing, Debugging & Deployment	Perform thorough testing to identify and fix bugs. Deploy the system using a free hosting platform (InfinityFree or 000WebHost). Document system architecture and final report submission. Provide a training guide for users to understand system functionality.

7. SCREENSHOTS OF KEY PAGES:

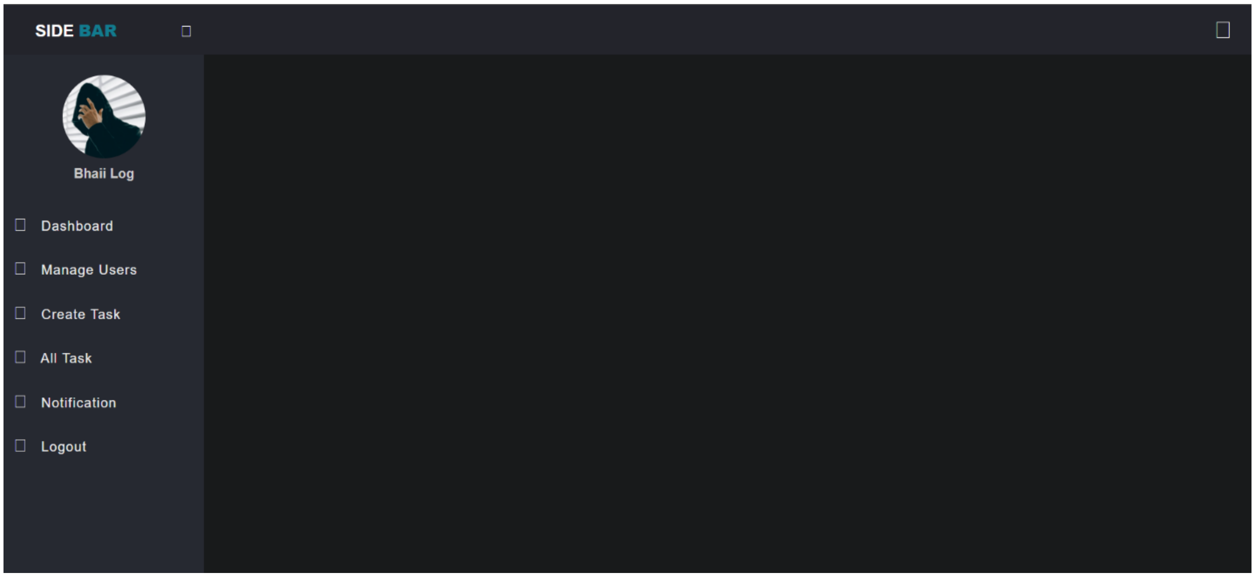
LOGIN PAGE:



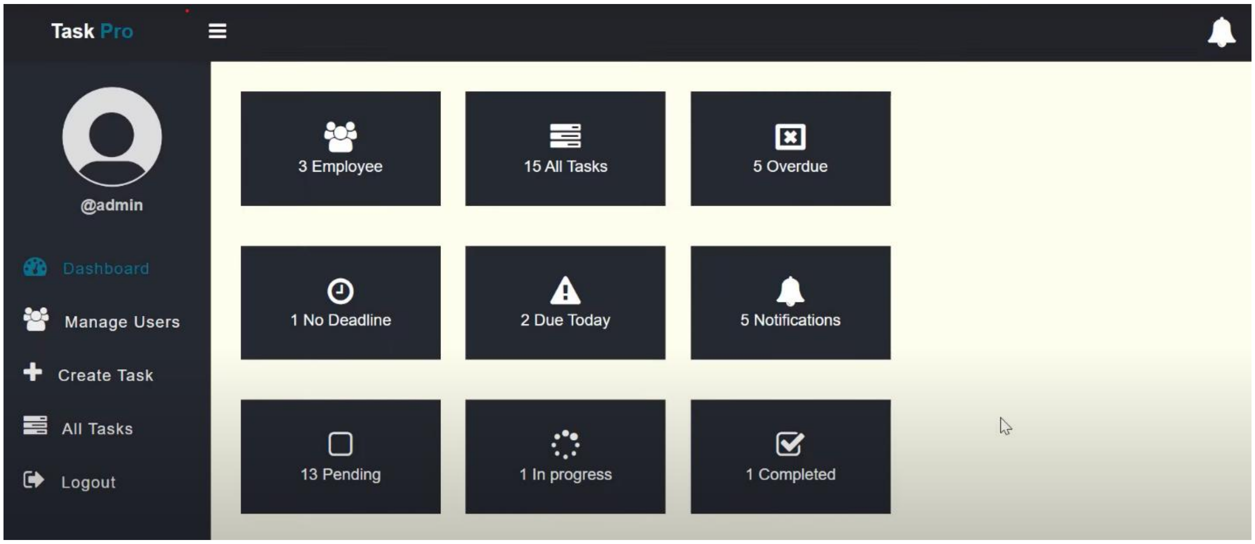
DASHBOARD:



SIDE BAR:



ADD TASK:



8. Brief Explanation of Front-End Structure:

The front-end of the Employee Task Management System is responsible for how the user interacts with the system visually and functionally. This includes everything that the user sees on the screen and how the browser responds to user actions. Below is a detailed explanation divided into seven comprehensive parts.

1. User Interface (UI) Design and Structure of UI Components:

The UI is built using:

- **HTML5**: Defines the structure of each web page, including layout, headings, buttons, forms, and containers.
- **CSS3:** Provides styling for spacing, fonts, color schemes, hover effects, and mobile responsiveness.
- **Bootstrap (v4 or v5):** A front-end framework used to design a responsive layout with minimal effort using pre-built components such as navbars, modals, cards, buttons, and forms.
- **Font Awesome:** Adds high-quality vector icons (e.g., trash, edit, dashboard icons) for intuitive user interaction.
- **Header Section:** Displays the top navigation bar for branding or logout.
- **Sidebar:** Fixed navigation menu on the left, used to switch between pages like Dashboard, Add Task, Manage Tasks, and User Management.
- **Main Content Area:** Central display area where all data, forms, and dashboards are dynamically loaded.
- This ensures a modern, clean, and consistent interface across all devices.

Navigation and User Flow

- **Sidebar Menu (Consistent across all pages):**
- Home/Dashboard
- Add Task
- Manage Tasks
- Register Employee and Logout

- **Routing:**

Each page like dashboard.php, add-task.php, and register-user.php is accessed via this sidebar.

- **Navigation Logic:**

Using PHP includes (header.php, footer.php) to prevent code repetition.

Highlighted active link to show user's current location.

Smooth user flow with a simple back and forth between task management, employee registration, and dashboard overview.

This ensures users do not get lost or confused in the application's structure.

Interactive Components

The project includes dynamic interactive elements to improve usability:

- **Forms:**

Used on pages like Add Task, Edit Task, and Register User.

All forms include required field validation (client-side and server-side).

Bootstrap styling used for input fields, labels, and validation messages.

- **Buttons**:

Submit, Cancel, Edit, Delete buttons styled with Bootstrap classes and icons.

- **Tables:**

Display lists of tasks or users in a tabular format.

Includes sortable columns and optional search or filter fields.

- **Modals (Optional):**

For confirmation dialogs or alerts (delete confirmation).

- **JavaScript**:

Minor JS used for form validation, sidebar toggle, or interactive alert messages (e.g., successful task creation).

Core Front-End Functionalities

Although the back-end handles logic and database, the front-end supports essential functions:

- **Form Validation:**

All forms (e.g., Add Task, Register Employee) use Bootstrap’s client-side validation to ensure fields are filled properly.

- **Data Rendering:**

PHP is used to fetch and render data inside HTML tables dynamically.

- **Responsive Cards:**

On the dashboard, task summaries/statistics are shown using responsive Bootstrap cards.

- **Conditional Buttons:**

Buttons like "Edit" or "Delete" only appear to authorized users based on session conditions (handled in PHP but controlled visually on the front-end).

- **Alert Messages:**

Success or error alerts appear when actions like task creation or deletion are performed.

6. Front-End Page-by-Page Breakdown

These pages will be made along with back-end and database one by one.

Page Name	Role & Description
index.php	Login screen with form, password validation, and error messages
dashboard.php	Displays cards/statistics of tasks (Pending, Completed)
add-task.php	Form to input task details such as title, deadline, priority
manage-task.php	Table listing all tasks with edit/delete actions
register-user.php	Form to add a new user or employee
header.php, sidebar.php, footer.php	Included across all pages for consistent layout
style.css	Contains custom CSS that overrides default Bootstrap styling assets/ Holds icons, fonts, or other media files
includes/db.php	Handles database connection for rendering dynamic data