PEDALSTART ASSIGNMENT

TASK MANAGER USING PHP AND MYSQL

Why PHP and MYSQL??

- PHP is designed to be simple to learn and use, especially for those with basic programming knowledge. Its syntax is straightforward, and it integrates seamlessly with HTML, making it a popular choice for beginners and seasoned developers alike.

- PHP is highly versatile and can be embedded directly into HTML code. It supports various databases (besides MySQL), making it adaptable to different project needs.

- With its ease of learning and use, PHP allows developers to write and deploy code quickly. Frameworks like Laravel, Symfony, and CodeIgniter further enhance development speed by providing pre-built modules, tools, and best practices.

- MySQL is known for its user-friendly interface and easy setup. Its SQL syntax is clear and straightforward, making database management tasks simpler to execute.

- Both technologies can handle increasing workloads. PHP applications can be optimized and scaled horizontally with load balancers, while MySQL can be scaled using replication, clustering, and partitioning techniques. This scalability ensures that projects can grow over time without requiring a complete overhaul of the technology stack.

Index.php:

This PHP and MySQL script creates a task manager web application. It includes constants for configuration and connects to a MySQL database to retrieve and display lists and tasks. The menu dynamically lists task categories, while tasks are displayed in a table with options to update or delete each task. The script also handles session messages for adding, deleting, or updating tasks. The page layout is styled with an external CSS file.

List_Task.php:

This PHP script generates a web page to display tasks from a specific list in a task manager application. It includes a configuration file and retrieves the list ID from the URL. The page connects to a MySQL database to fetch and display tasks associated with the given list ID. The tasks are shown in a table with options to update or delete each task. The page also includes a dynamic menu listing all task categories and a link to manage lists, with the layout styled using an external CSS file.

Manage_Lists.php:

This PHP script creates a "Manage Lists" page for a task manager web application. It includes a configuration file and connects to a MySQL database to fetch and display all task lists in a table. The page provides session-based messages for feedback on adding, deleting, or updating lists. Each list entry includes links for updating or deleting the list. The page layout is styled using an external CSS file and includes navigation links for returning to the home page and adding new lists.

Add_list.php:

This PHP script generates an "Add List" page for a task manager web application. It includes a form to submit a new list's name and description, which are then processed by the script upon form submission. The script connects to a MySQL database to insert the new list details into the "`tbl_list`" table. Based on the success or failure of the insertion, it sets session messages and redirects the user to either the "Manage Lists" page or the "Add List" page. The layout is styled using an external CSS file and includes navigation links to the home and manage lists pages.

Add_Task.php:

This PHP script generates an "Add Task" page for a task manager web application. It includes a form for submitting a new task, with fields for task name, description, list selection, priority, and deadline. The form data is processed upon submission by connecting to a MySQL database and inserting the new task into the "`tbl_tasks`" table. Based on the success or failure of the insertion, session messages are set, and the user is redirected to either the home page or the add task page. The layout is styled using an external CSS file and includes navigation links.

Delete_List.php:

This PHP script handles the deletion of a task list in a task manager web application. It checks if a `list_id` is provided via the URL and, if so, connects to a MySQL database to delete the corresponding list from the `tbl_list` table. After attempting the deletion, it sets session messages indicating success or failure and redirects the user to the "Manage Lists" page. If no `list_id` is provided, it directly redirects to the "Manage Lists" page.

Delete_Task.php:

This PHP script deletes a task from a task manager web application. It checks if a `task_id` is provided via the URL and, if so, connects to a MySQL database to delete the task from the `tbl_tasks` table. After attempting the deletion, it sets session messages indicating success or failure and redirects the user to the home page. If no `task_id` is provided, it redirects to the "delete_task" page.

Update_List.php:

This PHP script allows updating a task list in a task manager application. Initially, it checks if a `list_id` is provided via the URL and retrieves the corresponding list details from the database. The form displays the current list name and description, allowing the user to update these details. Upon form submission, the script updates the list information in the database and sets session messages indicating success or failure, redirecting the user accordingly. If no `list_id` is provided, it redirects to the "Manage Lists" page.

Update_Task.php:

This PHP script facilitates the updating of a task in a task manager application. Initially, it fetches the task details from the database if a `task_id` is provided in the URL. It displays a form pre-filled with the task's current details, allowing the user to modify the task name, description, list, priority, and deadline. Upon form submission, it updates the task details in the database and sets session messages indicating success or failure, redirecting the user accordingly. If no `task_id` is provided, it redirects to the main task management page.

Constants.php:

This PHP script initializes a session and defines constants used for database connection and website URLs. `LOCALHOST` specifies the database server name, `DB_USERNAME` and `DB_PASSWORD` provide database access credentials, and `DB_NAME` specifies the database name used for the task manager application. `SITEURL` defines the base URL of the website. These constants ensure consistent database connectivity and URL references throughout the application, facilitating maintenance and updates.

Challenges Faced:

- **Error Handling**: Ensuring robust error handling throughout database interactions (`die(mysqli_error())`) to catch and handle errors gracefully.
- **Security**: Preventing SQL injection by properly sanitizing user inputs or using prepared statements in database queries (`mysqli_real_escape_string()` or prepared statements).
- **User Experience**: Designing user-friendly interfaces for tasks like adding, updating, and deleting tasks or lists, ensuring intuitive navigation and clear feedback messages.

Future Improvements:

- **Validation**: Adding client-side and server-side validation for form inputs to improve data integrity and user feedback.
- **Error Logging**: Implementing error logging to capture and analyze runtime errors or exceptions, facilitating troubleshooting and maintenance.
- **Performance Optimization**: Optimizing database queries and implementing caching strategies to improve application performance, especially as data scales.

Commands to run project:

Creating a project : XAMPP folder > htdocs > create folder > create files

Creating a Database : Localhost/phpMyAdmin > create a folder > create tables

How to run : localhost/project name/filename