Simple Task Management System

This project is a simple Task Management System built using PHP and MySQL. It allows users to manage tasks by adding, viewing, and assigning them to different users. Here is an overview of how the project works:

Files and Their Roles

config.php: Contains database configuration constants such as SERVER_NAME, DB_USER, DB_PASSWORD, and DB_NAME.

db_manager.php: Contains functions to manage the database, including creating the database and tables, adding users and tasks, and retrieving users and tasks.

index.php: The main interface of the application. It displays a form to add tasks and a table to list all tasks.

Key Functions in *db_manager.php*

createDB(): Creates a connection to the MySQL server and Creates the database if it doesn't exist.

connectToDB(): Establishes a connection to the database and returns the connection object.

createTables(): Creates the users and tasks tables if they don't exist.

addUsersFromJson($jsonFilePath): Deletes all existing users and Reads users from a JSON file and adds them to the users table.

getUsers(): Retrieves all users from the users table.

addUser($name, $email): Adds a new user to the users table.

addTask($taskName, $deadline, $status, $assignedTo): Adds a new task to the tasks table.

getTasks(): Retrieves all tasks along with the assigned user's name from the tasks table.

How *index.php* Works

1. Includes Required Files: Includes **app.php** and **db_manager.php**.
2. Retrieves Data: Calls getUsers() to retrieve the list of users.
3. Calls getTasks() to retrieve the list of tasks.

4. Displays the Form: Displays a form to add a new task with fields for task name, deadline, status, and assigned user.
5. Displays the Task List: Displays a table listing all tasks with columns for task name, deadline, status, and assigned user.

Example Usage

Adding Users: Users can be added by calling the ***addUsersFromJson*** function with the path to a JSON file containing user data.

Adding Tasks: Users can add tasks using the form on the index.php page. The form submits data to **app.php** using javascript (using **axios** library for AJAX request) BY issuing **POST** request to the php script on the backend which calls ***addTask*** function to add new task to the database.

The **index.php** uses some bootstrap classes and a main css file to display all tasks in a table, showing the task name, deadline, status, and assigned user.