# task-manager

October 27, 2024

#### 0.0.1 Task Manager System

## 0.1 Overview

The Task Manager System is designed to help users manage their tasks efficiently. It allows users to register, log in, add tasks, view tasks, mark tasks as completed, and delete tasks. The system ensures that each user's tasks are private and can only be accessed and modified by the respective user.

## 0.2 Components

1. **User Management**: Handles user registration and login.
2. **Task Management**: Allows users to add, view, mark as completed, and delete tasks.
3. **Interactive Menu**: Provides a user-friendly interface for interacting with the system.

## 0.3 Class Diagram

```
+-----------------+
|  TaskManager    |
+-----------------+
| - USER_DATA_FILE: str  |
| - TASK_DATA_FILE: str  |
| - logged_in_user: str  |
+-----------------+
| + __init__()            |
| + hash_password(password: str) -> str |
| + register_user()       |
| + login_user() -> bool  |
| + add_task()            |
| + view_tasks()          |
| + mark_task_completed() |
| + delete_task()         |
| + interactive_menu()    |
+-----------------+
```

## 0.4 Flowchart

```
+-------------------+
| Start             |
+-------------------+
    |
    v
+------------------+
| Display Main Menu|
+------------------+
    |
    v
+------------------+
| User Choice      |
+------------------+
    |
    v
+------------------+       +------------------+
| 1. Register      |----->| Register User    |
+------------------+       +------------------+
    |                        |
    v                        v
+------------------+       +------------------+
| 2. Login         |----->| Login User       |
+------------------+       +------------------+
    |                        |
    v                        v
+------------------+       +------------------+
| 3. Exit          |----->| Exit Program     |
+------------------+       +------------------+
    |
    v
+------------------+
| Interactive Menu |
+------------------+
    |
    v
+------------------+       +------------------+
| Task Choice      |----->| 1. Add Task       |
+------------------+       +------------------+
    |                        |
    v                        v
+------------------+       +------------------+
| 2. View Tasks    |----->| Display Tasks    |
+------------------+       +------------------+
    |                        |
    v                        v
+------------------+       +------------------+
| 3. Mark Completed|----->| Mark Task        |
+------------------+       +------------------+
    |                        |
    v                        v
+------------------+       +------------------+
| 4. Delete Task   |----->| Delete Task      |
+------------------+       +------------------+
    |                        |
    v                        v
+------------------+       +------------------+
| 5. Logout        |----->| Logout User      |
+------------------+       +------------------+
    |
    v
+------------------+
| End              |
+------------------+
```

## 0.5 User Management

- **Register User**: Prompts the user to enter a username and password, hashes the password, and stores the user data in a JSON file(user_data.json).
- **Login User**: Prompts the user to enter a username and password, verifies the credentials, and sets the logged-in user.

## 0.6 Task Management

- **Add Task**: Prompts the user to enter a task description and stores the task in a JSON file(task_data.json).
- **View Tasks**: Displays all tasks for the logged-in user.
- **Mark Task Completed**: Prompts the user to enter a task ID and marks the task as completed.
- **Delete Task**: Prompts the user to enter a task ID and deletes the task.

## 0.7 Interactive Menu

- Provides options for adding tasks, viewing tasks, marking tasks as completed, deleting tasks, and logging out.

[4]:

```python
import json
import hashlib
import os

class TaskManager:
    USER_DATA_FILE = 'user_data.json'
    TASK_DATA_FILE = 'task_data.json'

    def __init__(self):
        self.logged_in_user = None

    def hash_password(self, password):
        return hashlib.sha256(password.encode()).hexdigest()

    def register_user(self):
        username = input("Enter a username: ")
        password = input("Enter a password: ")

        if os.path.exists(self.USER_DATA_FILE):
            with open(self.USER_DATA_FILE, 'r') as file:
                users = json.load(file)
        else:
            users = {}

        if username in users:
            print("Username already exists. Please choose a different username.")
            return

        # Add user to the users dictionary
        users[username] = self.hash_password(password)
        # Write the updated users dictionary to the user_data.json file
        with open(self.USER_DATA_FILE, 'w') as file:
            json.dump(users, file)

        print(f"{username} registered successfully!")

    def login_user(self):
        username = input("Enter your username: ")
        password = input("Enter your password: ")

        if os.path.exists(self.USER_DATA_FILE):
            with open(self.USER_DATA_FILE, 'r') as file:
                users = json.load(file)
        else:
            print("No users registered. Please register first.")
            return False

        if username in users and users[username] == self.hash_password(password):
            print(f"{username} Login successful!")
            self.logged_in_user = username
            return True
        else:
            print("Invalid username or password.")
            return False

    def add_task(self):
        if not self.logged_in_user:
            print("Please login first.")
            return
        task_description = input("Enter task description: ")

        if os.path.exists(self.TASK_DATA_FILE):
            with open(self.TASK_DATA_FILE, 'r') as file:
                tasks = json.load(file)
        else:
            tasks = {}

        # Generate a new task ID
        task_id = len(tasks) + 1
        # Add the new task to the tasks dictionary
        tasks[task_id] = {'username': self.logged_in_user, 'description': task_description, 'status': 'Pending'}

        # Write the updated tasks dictionary to the task_data.json file
        with open(self.TASK_DATA_FILE, 'w') as file:
            json.dump(tasks, file)

        print("Task added successfully!")

    def view_tasks(self):
        if not self.logged_in_user:
            print("Please login first.")
            return

        if os.path.exists(self.TASK_DATA_FILE):
            with open(self.TASK_DATA_FILE, 'r') as file:
                tasks = json.load(file)
        else:
            tasks = {}

        # Print all tasks for the logged in user
        for task_id, task in tasks.items():
            if task['username'] == self.logged_in_user:
                print(f"Task ID: {task_id}, Description: {task['description']}, Status: {task['status']}")

    def mark_task_completed(self):
        if not self.logged_in_user:
            print("Please login first.")
            return

        task_id = input("Enter task ID to mark as completed: ")

        if os.path.exists(self.TASK_DATA_FILE):
            with open(self.TASK_DATA_FILE, 'r') as file:
                tasks = json.load(file)
        else:
            tasks = {}

        # Check if the task exists and the logged in user is the owner of the␣
↪task
        if task_id in tasks and tasks[task_id]['username'] == self.
↪logged_in_user:
            # Update the status of the task to 'Completed'
            tasks[task_id]['status'] = 'Completed'
            with open(self.TASK_DATA_FILE, 'w') as file:
                json.dump(tasks, file)
            print("Task marked as completed!")
        else:
            print("Task not found or you do not have permission to modify this␣
↪task.")

    def delete_task(self):
        if not self.logged_in_user:
            print("Please login first.")
            return

        task_id = input("Enter task ID to delete: ")

        if os.path.exists(self.TASK_DATA_FILE):
            with open(self.TASK_DATA_FILE, 'r') as file:
                tasks = json.load(file)
        else:
            tasks = {}

        if task_id in tasks and tasks[task_id]['username'] == self.
↪logged_in_user:
            # Delete the task
            del tasks[task_id]
            with open(self.TASK_DATA_FILE, 'w') as file:
                json.dump(tasks, file)
            print("Task deleted successfully!")
        else:
            print("Task not found or you do not have permission to delete this␣
↪task.")

    def interactive_menu(self):
        while True:
            print("\nMenu:")
            print("1. Add a Task")
            print("2. View Tasks")
            print("3. Mark a Task as Completed")
            print("4. Delete a Task")
            print("5. Logout")

            choice = input("Interactive Menu - Enter your choice: ")

            if choice == '1':
                self.add_task()
            elif choice == '2':
                self.view_tasks()
            elif choice == '3':
                self.mark_task_completed()
            elif choice == '4':
                self.delete_task()
            elif choice == '5':
                print("Logging out...")
                self.logged_in_user = None
                break
            else:
                print("Invalid choice. Please try again.")
```

[5]:

```
if __name__ == "__main__":
    manager = TaskManager()
    while True:
        print("\n1. Register")
        print("2. Login")
        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            manager.register_user()
        elif choice == '2':
            if manager.login_user():
                manager.interactive_menu()
        elif choice == '3':
            break
        else:
            print("Invalid choice. Please try again.")
```

```
1. Register
2. Login
3. Exit

Enter your choice:  1
Enter a username:  Jennifer
Enter a password:  daliXu123

Jennifer registered successfully!
```

```
1. Register
2. Login
3. Exit

Enter your choice:  2
Enter your username:  Jennifer
Enter your password:  daliXu123

Jennifer Login successful!

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  1
Enter task description:  task manager project

Task added successfully!

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  1
Enter task description:  online study

Task added successfully!

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  2

Task ID: 1, Description: task manager project, Status: Pending
Task ID: 2, Description: online study, Status: Pending

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
```

```
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  3
Enter task ID to mark as completed:  1

Task marked as completed!

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  2

Task ID: 1, Description: task manager project, Status: Completed
Task ID: 2, Description: online study, Status: Pending

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  4
Enter task ID to delete:  2

Task deleted successfully!

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  2

Task ID: 1, Description: task manager project, Status: Completed

Menu:
1. Add a Task
2. View Tasks
3. Mark a Task as Completed
4. Delete a Task
5. Logout

Interactive Menu - Enter your choice:  5
```

```
Logging out…

1. Register
2. Login
3. Exit

Enter your choice:  3
```

```
[ ]:
```