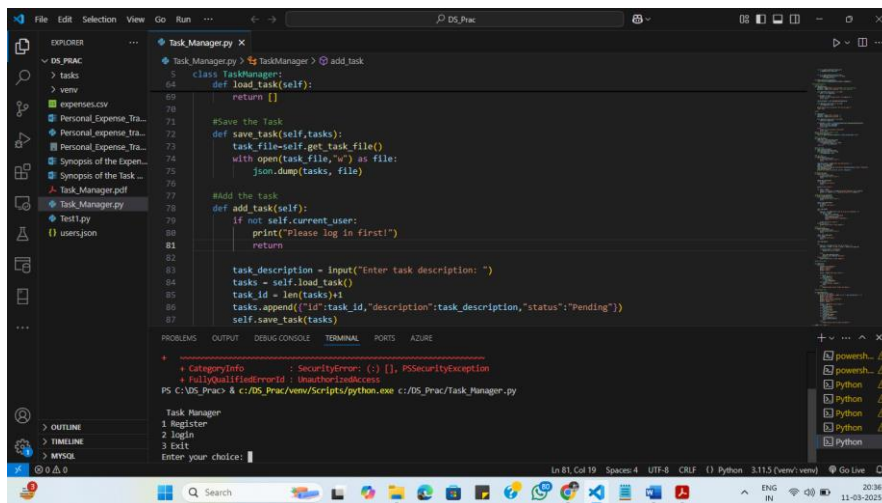
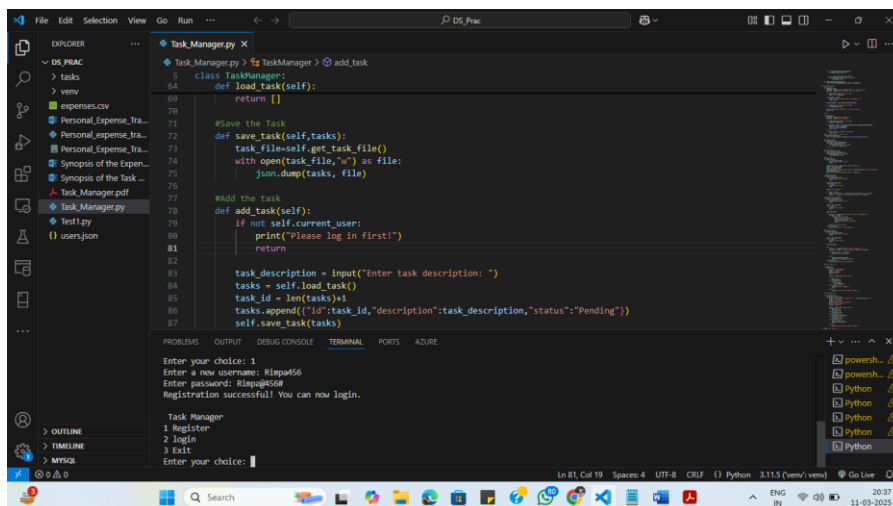


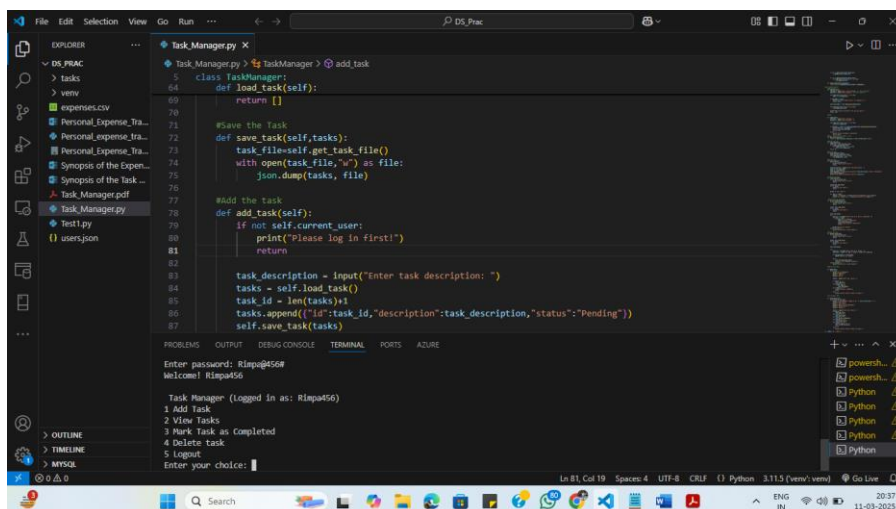
1. Run the Code



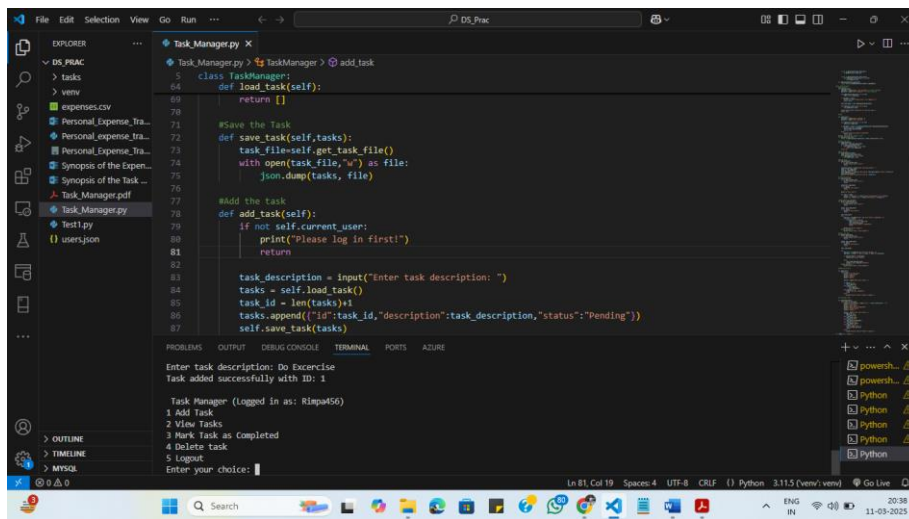
2. New User Registration



3. Login as user



4. Add a Task



```
5 class TaskManager:
6     def load_task(self):
7         return []
8
9     #Save the Task
10    def save_task(self, tasks):
11        task_file=self.get_task_file()
12        with open(task_file, "w") as file:
13            json.dump(tasks, file)
14
15    #Add the task
16    def add_task(self):
17        if not self.current_user:
18            print("Please log in first!")
19            return
20
21        task_description = input("Enter task description: ")
22        tasks = self.load_task()
23        task_id = len(tasks)+1
24        tasks.append({"id":task_id, "description":task_description, "status":"Pending"})
25        self.save_task(tasks)
```

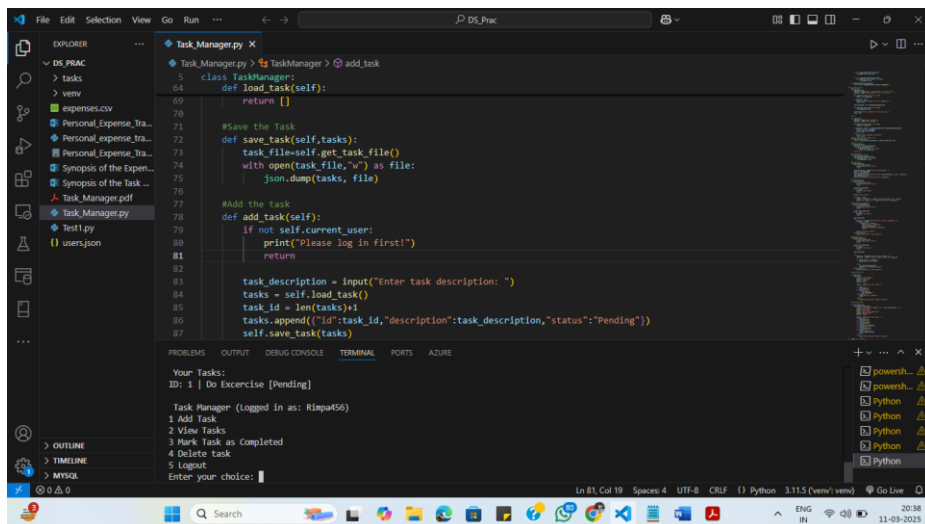
Enter task description: Do Exercise
Task added successfully with ID: 1

Task Manager (Logged in as: Rimp456)

- 1 Add Task
- 2 View Tasks
- 3 Mark Task as Completed
- 4 Delete task
- 5 Logout

Enter your choice: 1

5. View Task



```
5 class TaskManager:
6     def load_task(self):
7         return []
8
9     #Save the Task
10    def save_task(self, tasks):
11        task_file=self.get_task_file()
12        with open(task_file, "w") as file:
13            json.dump(tasks, file)
14
15    #Add the task
16    def add_task(self):
17        if not self.current_user:
18            print("Please log in first!")
19            return
20
21        task_description = input("Enter task description: ")
22        tasks = self.load_task()
23        task_id = len(tasks)+1
24        tasks.append({"id":task_id, "description":task_description, "status":"Pending"})
25        self.save_task(tasks)
```

Your Tasks:

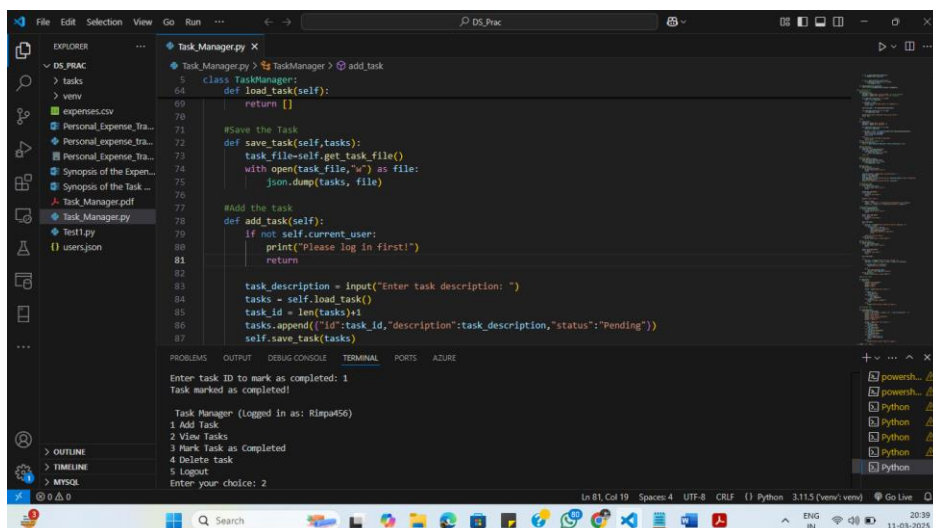
ID: 1 | Do Exercise [Pending]

Task Manager (Logged in as: Rimp456)

- 1 Add Task
- 2 View Tasks
- 3 Mark Task as Completed
- 4 Delete Task
- 5 Logout

Enter your choice: 1

6. Mark Task as Completed



```
5 class TaskManager:
6     def load_task(self):
7         return []
8
9     #Save the Task
10    def save_task(self, tasks):
11        task_file=self.get_task_file()
12        with open(task_file, "w") as file:
13            json.dump(tasks, file)
14
15    #Add the task
16    def add_task(self):
17        if not self.current_user:
18            print("Please log in first!")
19            return
20
21        task_description = input("Enter task description: ")
22        tasks = self.load_task()
23        task_id = len(tasks)+1
24        tasks.append({"id":task_id, "description":task_description, "status":"Pending"})
25        self.save_task(tasks)
```

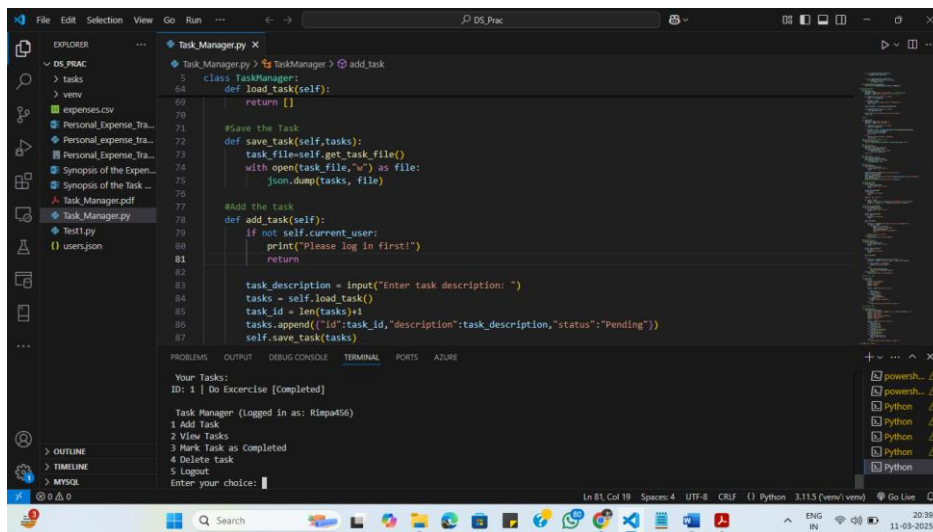
Enter task ID to mark as completed: 1
Task marked as completed!

Task Manager (Logged in as: Rimp456)

- 1 Add Task
- 2 View Tasks
- 3 Mark Task as Completed
- 4 Delete task
- 5 Logout

Enter your choice: 2

7. View Task as Completed

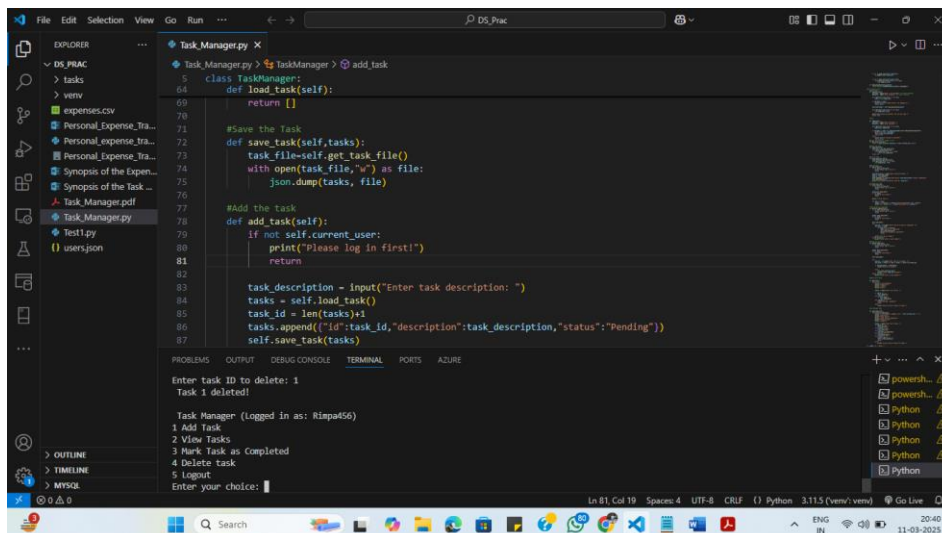


The screenshot shows the VS Code editor with the `TaskManager.py` file open. The code defines a `TaskManager` class with methods `load_task`, `save_task`, `add_task`, and `delete_task`. The terminal output shows the following sequence of commands and responses:

```
Your Tasks:
ID: 1 | Do Exercise [Completed]

Task Manager (Logged in as: Riya456)
1 Add Task
2 View Tasks
3 Mark Task as Completed
4 Delete task
5 Logout
Enter your choice: 3
```

8. Delete the Task

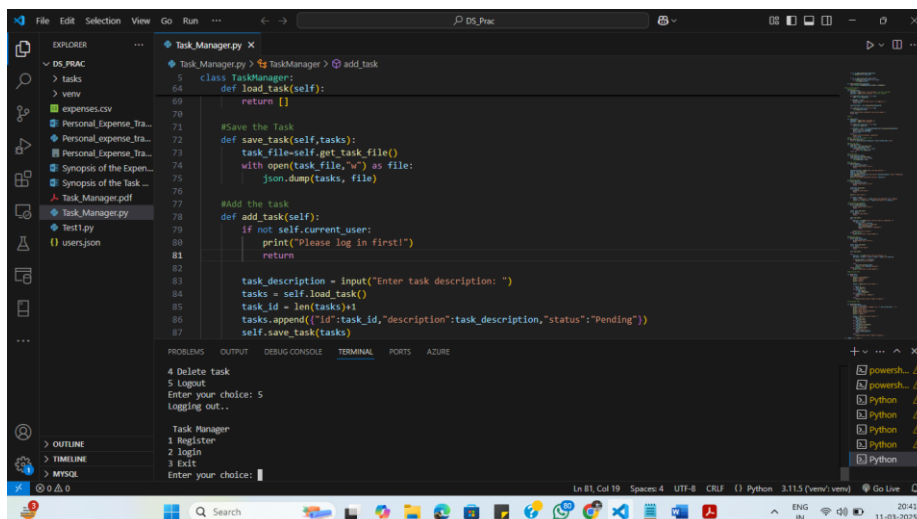


The screenshot shows the VS Code editor with the `TaskManager.py` file open. The code defines a `TaskManager` class with methods `load_task`, `save_task`, `add_task`, and `delete_task`. The terminal output shows the following sequence of commands and responses:

```
Enter task ID to delete: 1
Task 1 deleted!

Task Manager (Logged in as: Riya456)
1 Add Task
2 View Tasks
3 Mark Task as Completed
4 Delete task
5 Logout
Enter your choice: 4
```

9. Logging Out

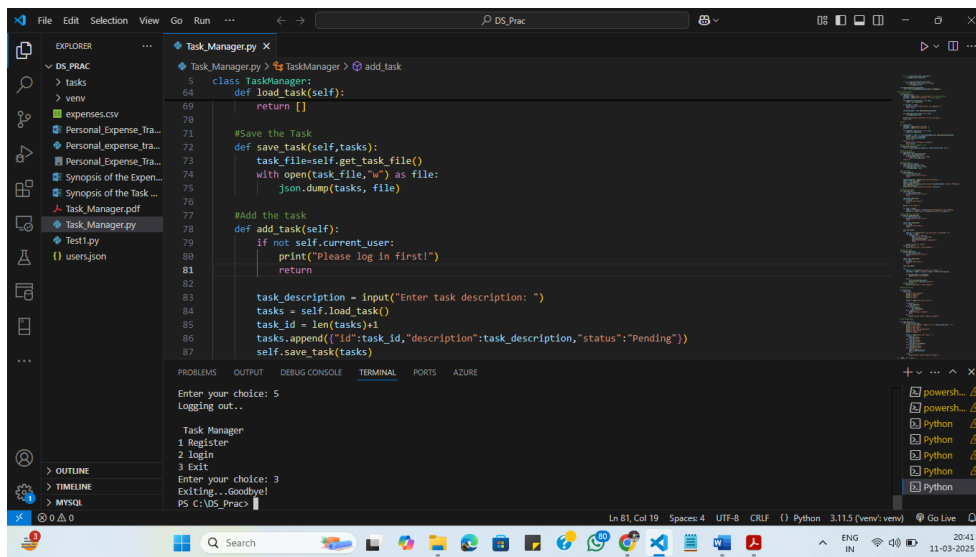


The screenshot shows the VS Code editor with the `TaskManager.py` file open. The code defines a `TaskManager` class with methods `load_task`, `save_task`, `add_task`, and `delete_task`. The terminal output shows the following sequence of commands and responses:

```
4 Delete task
5 Logout
Enter your choice: 5
Logging out...

Task Manager
1 Register
2 login
3 Exit
Enter your choice: 1
```

10. Exiting from Menu



The screenshot shows a Visual Studio Code editor with a file explorer on the left. The file explorer shows a project named 'DS_Prac' with a folder 'tasks' containing 'Task_Manager.py'. The main editor window displays the code for 'Task_Manager.py'. The code defines a 'TaskManager' class with methods 'load_task', 'save_task', and 'add_task'. The 'add_task' method prompts the user for a task description and appends it to a list of tasks. The terminal window at the bottom shows the output of the program, including the prompt 'Enter your choice: 5' and the response 'Logging out..'. The status bar at the bottom indicates the file is in UTF-8 encoding and the Python interpreter is set to 'venv'.

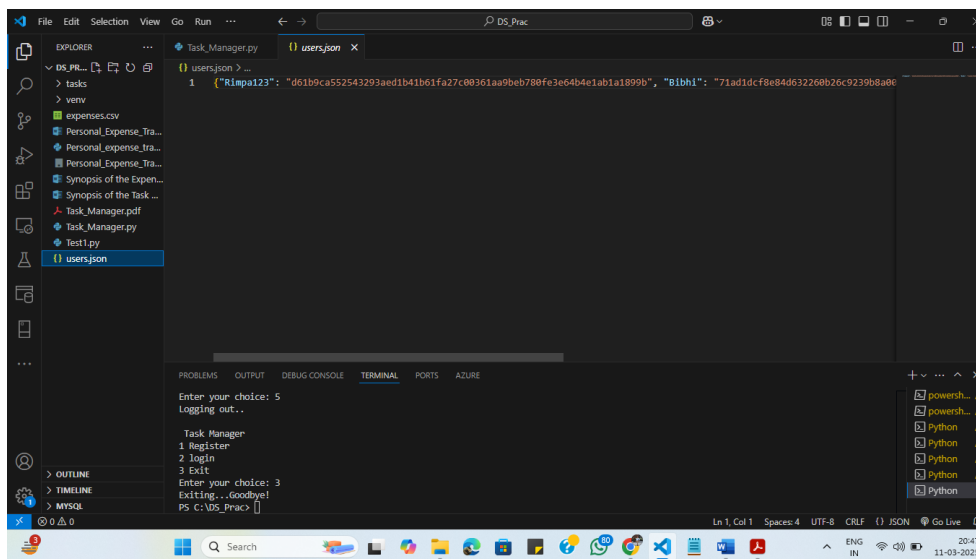
```
Task_Manager.py
5 class TaskManager:
6     def load_task(self):
7         return []
8
9     #Save the Task
10    def save_task(self, tasks):
11        task_file=self.get_task_file()
12        with open(task_file,"w") as file:
13            json.dump(tasks, file)
14
15    #Add the task
16    def add_task(self):
17        if not self.current_user:
18            print("Please log in first!")
19            return
20
21        task_description = input("Enter task description: ")
22        tasks = self.load_task()
23        task_id = len(tasks)+1
24        tasks.append({"id":task_id,"description":task_description,"status":"Pending"})
25        self.save_task(tasks)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Enter your choice: 5
Logging out..

Task Manager
1 Register
2 login
3 Exit
Enter your choice: 3
Exiting...Goodbye!
PS C:\DS_Prac>

11. Users.json file with hashed password Security:



The screenshot shows a Visual Studio Code editor with a file explorer on the left. The file explorer shows a project named 'DS_Prac' with a folder 'tasks' containing 'Task_Manager.py'. The main editor window displays the code for 'users.json'. The code is a JSON array containing two user objects. Each object has a 'username' field and a 'password' field. The 'password' field contains a hashed password. The terminal window at the bottom shows the output of the program, including the prompt 'Enter your choice: 5' and the response 'Logging out..'. The status bar at the bottom indicates the file is in UTF-8 encoding and the JSON interpreter is set to 'JSON'.

```
users.json
1 [
2   {
3     "username": "Rimp123",
4     "password": "d61b9ca552543293aed1b41b61fa27c0e361aa9beb780fe3e64b4e1ab1a1899b",
5     "BIbhl": "71ad1dcf8e84d632260b26c9239b8a0e"
6   }
7 ]
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AZURE

Enter your choice: 5
Logging out..

Task Manager
1 Register
2 login
3 Exit
Enter your choice: 3
Exiting...Goodbye!
PS C:\DS_Prac>