MAJOR PROJECT

Name: Rashmi

Email: rashmirashmi63244@gmail.com

Domain: Python Programming Internship

Number:8867625992

Deployed link: (Facing authentication error so I unable to copy the link)

Github Repo: https://github.com/RashmiAcharya24/To_do_application

LinkedIn post link: https://www.linkedin.com/posts/rashmi-acharya-1b878230b

CODE:

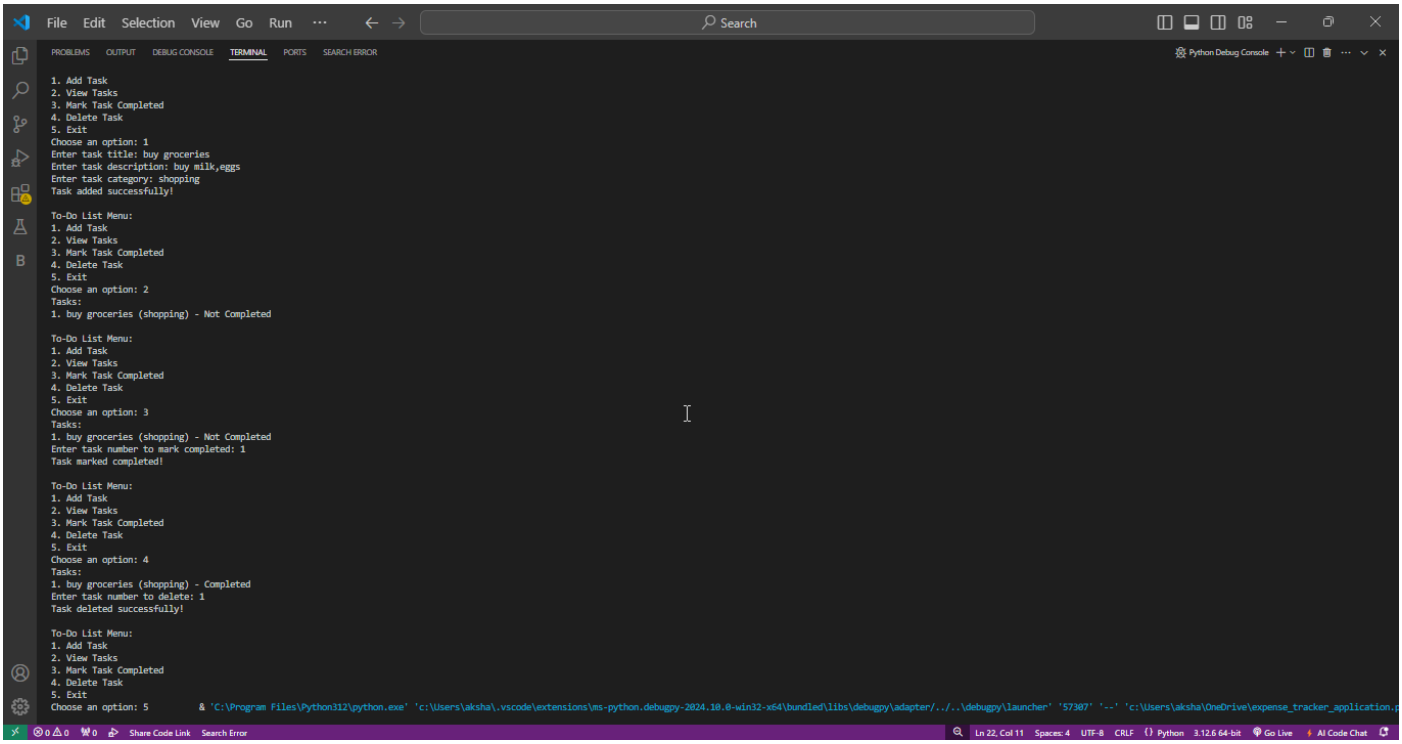
```
import json    # Importing JSON module to handle file saving/loading

# Class representing a Task
class Task:
    def __init_ _(self, title, description, category):
        # Constructor to initialize the task's title, description, and category
        self.title = title
        self.description = description
        self.category = category
        self.completed = False    # Default task status is not completed

    # Method to mark the task as completed
    def mark_completed(self):
        self.completed = True

# Function to save the list of tasks to a file (tasks.json)
def save_tasks(tasks):
    with open('tasks.json', 'w') as f:
```

```
        # Convert the list of Task objects to dictionaries and save as JSON
        json.dump([task.__dict__ for task in tasks], f)

# Function to load tasks from the file (tasks.json)
def load_tasks():
    try:
        with open('tasks.json', 'r') as f:
            # Convert the JSON data back into Task objects
            return [Task(**data) for data in json.load(f)]
    except FileNotFoundError:
        # If file does not exist, return an empty list
        return []

# Function to add a new task
def add_task(tasks):
    title = input("Enter task title: ")  # Get task title from user
    description = input("Enter task description: ")    # Get task description from user
    category = input("Enter task category: ")     # Get task category from user
    task = Task(title, description, category)      # Create a new Task object
    tasks.append(task)     # Add the new task to the list
    save_tasks(tasks)     # Save the updated task list
    print("Task added successfully!")

# Function to display all tasks
def view_tasks(tasks):
    print("Tasks:")
    # Loop through the tasks and display their details (index, title, category, and
completion status)
    for i, task in enumerate(tasks, start=1):
```

```
        print(f"{i}. {task.title} ({task.category}) - {'Completed' if task.completed else 'Not Completed'}")


# Function to mark a task as completed
def mark_task_completed(tasks):
    view_tasks(tasks)     # Show the tasks so user can choose one
    task_index = int(input("Enter task number to mark completed: ")) - 1     # Get task number
    if task_index < len(tasks):     # Ensure the task number is valid
        tasks[task_index].mark_completed()     # Mark the task as completed
        save_tasks(tasks)      # Save the updated task list
        print("Task marked completed!")
    else:
        print("Invalid task number.")


# Function to delete a task
def delete_task(tasks):
    view_tasks(tasks)    # Show the tasks so user can choose one
    task_index = int(input("Enter task number to delete: ")) - 1     # Get task number
    if task_index < len(tasks):     # Ensure the task number is valid
        del tasks[task_index]    # Delete the selected task
        save_tasks(tasks)      # Save the updated task list
        print("Task deleted successfully!")
    else:
        print("Invalid task number.")


# Main function to run the to-do list application
def main():
    tasks = load_tasks()     # Load existing tasks (if any)
    while True:
```

```
        # Display menu options to the user
        print("\nTo-Do List Menu:")
        print("1. Add Task")
        print("2. View Tasks")
        print("3. Mark Task Completed")
        print("4. Delete Task")
        print("5. Exit")
        choice = input("Choose an option: ")     # Get user input for menu option

        # Perform action based on the user's choice
        if choice == '1':
            add_task(tasks)     # Add a new task
        elif choice == '2':
            view_tasks(tasks)    # View all tasks
        elif choice == '3':
            mark_task_completed(tasks)      # Mark a task as completed
        elif choice == '4':
            delete_task(tasks)     # Delete a task
        elif choice == '5':
            save_tasks(tasks)      # Save tasks before exiting
            break    # Exit the loop and end the program
        else:
            print("Invalid choice. Please try again.")    # Handle invalid input

# Entry point of the program
if __name__ == "__main__":
    main()  # Call the main function to start the program
```

OUTPUT


File Edit Selection View Go Run
Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
Python Debug Console
1. Add Task
2. View Tasks
3. Mark Task Completed
4. Delete Task
5. Exit
Choose an option: 1
Enter task title: buy groceries
Enter task description: buy milk,eggs
Enter task category: shopping
Task added successfully!
To-Do List Menu:
1. Add Task
2. View Tasks
3. Mark Task Completed
4. Delete Task
5. Exit
Choose an option: 2
Tasks:
1. buy groceries (shopping) - Not Completed
To-Do List Menu:
1. Add Task
2. View Tasks
3. Mark Task Completed
4. Delete Task
5. Exit
Choose an option: 3
Tasks:
1. buy groceries (shopping) - Not Completed
Enter task number to mark completed: 1
Task marked completed!
To-Do List Menu:
1. Add Task
2. View Tasks
3. Mark Task Completed
4. Delete Task
5. Exit
Choose an option: 4
Tasks:
1. buy groceries (shopping) - Completed
Enter task number to delete: 1
Task deleted successfully!
To-Do List Menu:
1. Add Task
2. View Tasks
3. Mark Task Completed
4. Delete Task
5. Exit
Choose an option: 5
& 'C:\Program Files\Python312\python.exe' 'c:\Users\aksha\.vscode\extensions\ms-python.debugpy-2024.10.0-win32-x64\bundled\libs\debugpy\adapter/../..\debugpy\launcher' '57307' '--' 'c:\Users\aksha\OneDrive\expense_tracker_application.p
Share Code Link Search Error
Ln 22, Col 11 Spaces: 4 UTF-8 CRLF Python 3.12.6 64-bit Go Live AI Code Chat