Python Major Project – Personal TO-DO list

```
#Code starts here

import json

import os


# TaskEntry class representing a single task entry

class TaskEntry:

    def __init__(self, task_name, details, label):

        self.task_name = task_name

        self.details = details

        self.label = label

        self.is_done = False


    def set_done(self):

        self.is_done = True


    def to_dict(self):

        # Convert task entry to dictionary for saving as JSON

        return {

            'task_name': self.task_name,

            'details': self.details,

            'label': self.label,

            'is_done': self.is_done

        }


    @staticmethod
```

```
    def from_dict(data):
        # Create a task object from a dictionary (used when loading tasks)
        task = TaskEntry(data['task_name'], data['details'], data['label'])
        task.is_done = data['is_done']
        return task


# Function to store task entries into a JSON file
def save_entries(task_list, file_path='entries.json'):
    with open(file_path, 'w') as f:
        json.dump([task.to_dict() for task in task_list], f, indent=4)
    print("Task entries have been successfully saved.")


# Function to load task entries from a JSON file
def load_entries(file_path='entries.json'):
    if not os.path.exists(file_path):
        return []
    with open(file_path, 'r') as f:
        entries_data = json.load(f)
        return [TaskEntry.from_dict(entry) for entry in entries_data]


# Function to create a new task entry
def create_entry(task_list):
    task_name = input("Enter task name: ").strip()
    details = input("Enter task details: ").strip()
```

```
    label = input("Enter task label (e.g., Office, Home, Urgent): ").strip()
    task = TaskEntry(task_name, details, label)
    task_list.append(task)
    print(f"Task '{task_name}' has been added.")


# Function to show all task entries
def show_entries(task_list):
    if not task_list:
        print("No tasks available.")
        return
    print("\nTask List:")
    for idx, task in enumerate(task_list, 1):
        status = "□" if task.is_done else "□"
        print(f"{idx}. [{status}] {task.task_name} - {task.label}")
        print(f"   Details: {task.details}")
    print()


# Function to mark an entry as done
def mark_entry_done(task_list):
    show_entries(task_list)
    if not task_list:
        return
    try:
        choice = int(input("Enter the task number to mark as done: "))
        if 1 <= choice <= len(task_list):
```

```
            task_list[choice - 1].set_done()
            print(f"Task '{task_list[choice - 1].task_name}' marked as done.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Please enter a valid number.")


# Function to remove a task entry
def remove_entry(task_list):
    show_entries(task_list)
    if not task_list:
        return
    try:
        choice = int(input("Enter the task number to delete: "))
        if 1 <= choice <= len(task_list):
            removed_task = task_list.pop(choice - 1)
            print(f"Task '{removed_task.task_name}' removed.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Please enter a valid number.")


# Main function to handle user interaction
def task_manager():
    task_list = load_entries()  # Load task entries from the JSON file on startup
```

```
    while True:
        print("\n--- Task Manager ---")
        print("1. Add New Task")
        print("2. View All Tasks")
        print("3. Mark Task as Done")
        print("4. Remove Task")
        print("5. Exit")
        user_input = input("Choose an option (1-5): ").strip()

        if user_input == '1':
            create_entry(task_list)
        elif user_input == '2':
            show_entries(task_list)
        elif user_input == '3':
            mark_entry_done(task_list)
        elif user_input == '4':
            remove_entry(task_list)
        elif user_input == '5':
            save_entries(task_list)  # Save task entries before exiting
            print("Exiting Task Manager. Goodbye!")
            break
        else:
            print("Invalid option. Please select a valid number.")

if __name__ == "__main__":
    task_manager()
```