**TASK NAME: TASK MANAGER**

**PREPARED BY: KIPCHIRCHIR RAPHAEL**

**NEPTUNE CODE: LGL7CS**

**Project Overview**

The Task Manager Application is a command-line tool for organizing and tracking tasks. It allows users to manage both personal and work-related tasks by adding, listing, deleting, and saving tasks. Users can also load saved tasks from a CSV file and view pending or overdue tasks. The application supports two task types:

* **Personal Tasks**: Simple tasks with a description and priority level.
* **Work Tasks**: Tasks that allow for team collaboration by assigning team members.

**Features**

1. **Create Tasks**: Add new personal or work tasks, setting attributes like title, due date, description, and team members.
2. **View Tasks**: List all tasks, including task details and status.
3. **Delete Tasks**: Remove tasks by task ID.
4. **Save to CSV**: Save all tasks to a CSV file.
5. **Load from CSV**: Load tasks from a previously saved CSV file.
6. **View Pending and Overdue Tasks**: Display tasks that are either pending or overdue.

**Project Structure**

The project consists of three main files:

1. **interface.py**: Handles user interaction and presents menu options for managing tasks.
2. **task\_manager.py**: Manages the task list, including adding, listing, deleting, saving, and loading tasks.
3. **task.py**: Defines the Task, PersonalTask, and WorkTask classes with attributes like title, due date, and status.

**Code Overview**

**interface.py**

* **main\_menu(task\_manager)**: Displays the main menu and directs users to the appropriate functions.
* **switch\_menu(option, task\_manager)**: Processes user input to create, view, delete, save, or load tasks.
* **Task Creation**: Prompts users for task type (Personal/Work), title, and due date. Allows setting descriptions for personal tasks and team members for work tasks.

**task\_manager.py**

* **TaskManager**: Manages a list of tasks.
  + **add\_task(task)**: Adds a task to the list.
  + **list\_tasks()**: Lists all tasks.
  + **delete\_task(task\_id)**: Deletes a task by ID.
  + **save\_task()**: Saves all tasks to a CSV file.
  + **load\_task()**: Loads tasks from a CSV file.
  + **get\_pending\_tasks()** and **get\_overdue\_tasks()**: Returns tasks that are pending or overdue based on the current date.

**task.py**

* **Task**: Base class representing a general task, with attributes like title, due date, and status.
* **PersonalTask**: Inherits from Task and includes a priority attribute.
* **WorkTask**: Inherits from Task and supports adding multiple team members.

**Setup and Installation**

1. Clone the repository:

**Copy code**

git clone https://github.com/username/task-manager.git

cd task-manager

1. Install dependencies:
   * Ensure Python 3.x is installed.
   * No additional libraries are required.
2. Run the application:

**Copy code**

python interface.py

**Usage Instructions**

1. **Starting the Application**: Run interface.py, which will display the main menu.
2. **Menu Options**:
   * Select options (1-7) to create, view, delete, save, load tasks, or view pending/overdue tasks.
3. **Saving and Loading**: Use options 4 and 5 to save tasks to or load tasks from task\_list.csv.
4. **Exit**: Choose option 7 to exit the application.

**CSV Format**

The tasks are saved in task\_list.csv with the following columns:

* task\_id: Unique identifier for each task.
* title: Task title.
* due\_date: Due date in YYYY/MM/DD format.
* status: Current task status (e.g., "pending", "completed").
* description: Brief description for personal tasks.

**Example Usage**

1. **Creating a Task**:
   * Select option 1.
   * Enter "Personal" or "Work" for the task type.
   * Provide the title, due date, and description/team members as prompted.
2. **Viewing Tasks**:
   * Select option 2 to view all tasks.
3. **Deleting a Task**:
   * Select option 3, then provide the task ID to delete.

**Error Handling and Validation**

* **Invalid Task Type**: If an invalid type is entered, an error message is displayed.
* **Description Length**: Descriptions are limited to 15 characters for personal tasks.
* **Team Member Validation**: Only non-empty names can be added as team members.

**Future Enhancements**

* **UI Improvement**: Implement a GUI for better user experience.
* **Recurring Tasks**: Support for setting recurring tasks.
* **Task Prioritization**: Add advanced filtering based on task priority.
* **Notification System**: Send reminders for pending and overdue tasks.