SD Task Manager

Sergey Dmitriev

12/15/24

Period 2

Documentation

**Purpose**

The purpose of this project is to create a program that manages tasks in an user-friendly way. This program lets users add new tasks with details like description, due date, and priority, view their ongoing tasks, mark tasks as complete, and review completed tasks in a separate section. By providing these functions, the program is made to solve the common problem of keeping track of daily activities and ensuring that important tasks are not missed. It simplifies task management through an intuitive graphical interface, making it accessible and convenient for users.

**Procedures and Explanation**

**Step-by-Step Explanation**

1. **Initial Setup:**
   * The program starts by importing needed libraries like tkinter for the graphical user interface (GUI) and messagebox for pop-up notifications.
   * Two lists, tasks and completed\_tasks, are initialized to store tasks that are active or completed.
   * The main application window is created using Tk(), which serves as the primary interface.

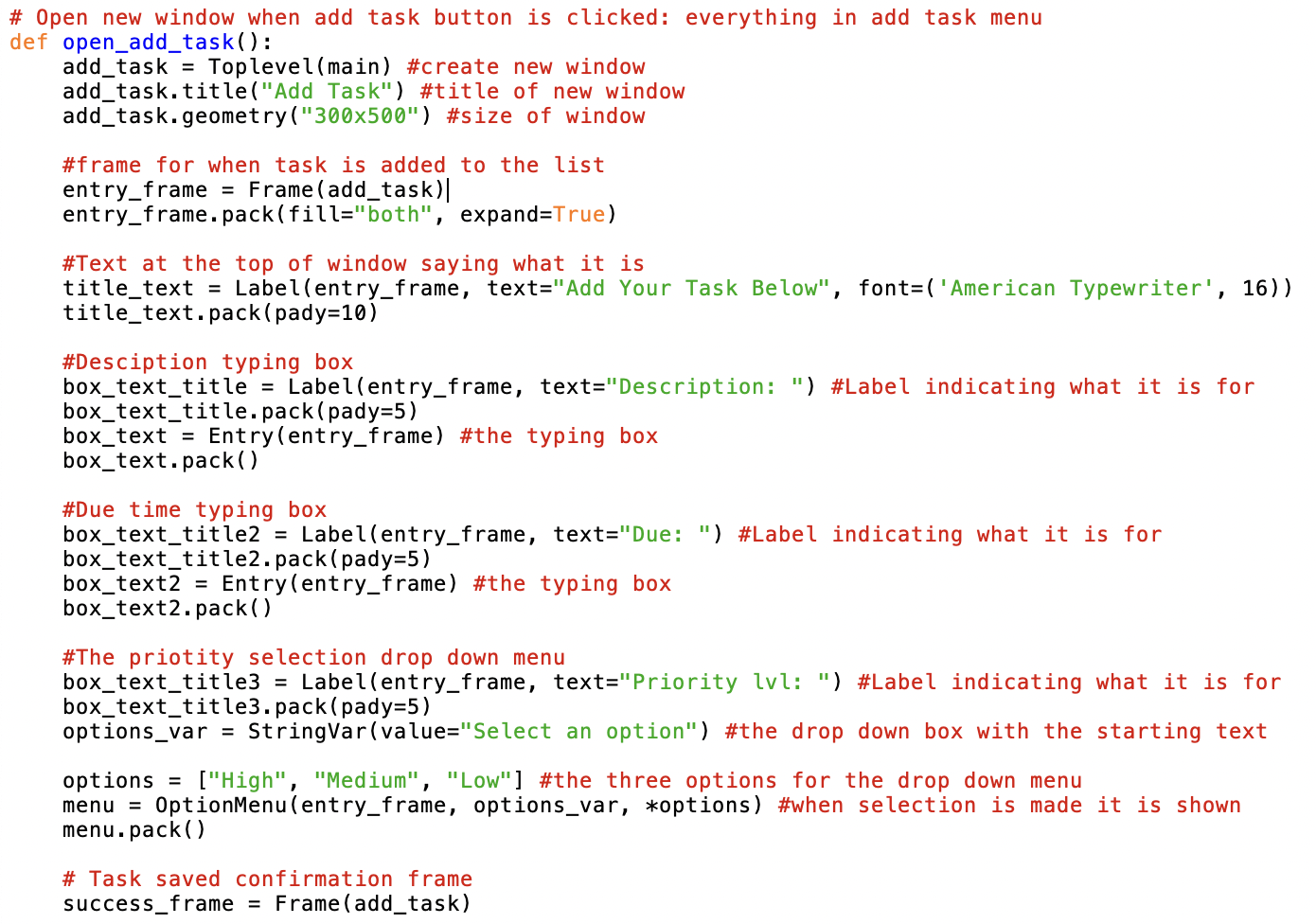
**Code Snippet**

A white background with red text

Description automatically generated

1. **Core Functionalities:**
   * **Add Task Feature:**
     + Clicking the "Add Task" button opens a new window for entering task details, such as description, due date, and priority level.
     + Input fields are validated to ensure all required information is provided.
     + When the "Save Task" button is clicked, the task is added to the tasks list and displayed in the main window.

**Code Snippet**

****

A white background with black and red text

Description automatically generated

* **Display Task:**
  + Tasks are displayed in the main window along with a "Complete" button. Each task is displayed in its own frame for clarity.
  + Clicking "Complete" removes the task from the tasks list and adds it to the completed\_tasks list.

**Code Snippet**

**A computer code with black text

Description automatically generated**

**A white background with black and red text

Description automatically generated**

* **View Completed Tasks:**
  + A separate window shows all tasks marked as completed.
  + The window refreshes every second to update the list in real time.

**Code Snippet**

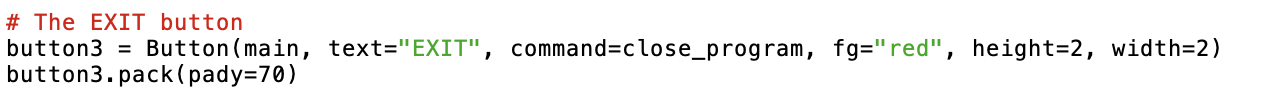
A computer screen shot of a program

Description automatically generated

1. **Exit Functionality:**
   * An "EXIT" button is included to close the program, ensuring an easy way to quit the application.

**Code Snippet**

A close-up of a white background

Description automatically generated

1. **Programming Concepts Used:**
   * **Data Types and Variables:**
     + Used strings for task details and lists (tasks, completed\_tasks) to store and manage tasks.
   * **Operators:**
     + Used conditional operators to validate inputs before saving tasks.
   * **Functions:**
     + Created functions such as open\_add\_task and complete\_task for better organization.
   * **Modules:**
     + Used tkinter to create the GUI and messagebox for notifications.
   * **Control Flow:**
     + Made conditionals to check if input fields are filled.
     + Used loops for auto-refreshing the completed tasks window.
   * **Strings and String Manipulation:**
     + Concatenated strings to display task details like "Task (Due: Date, Priority: Level)".
   * **Lists:**
     + Stored tasks in lists for easy addition, removal, and iteration.

**Results and Outputs Including Screenshots**

**Screenshots**

1. **Main Window:**

A screenshot of a phone

Description automatically generated

1. **Add Task Window:**

A screenshot of a task

Description automatically generated

1. **Task List:**

A screenshot of a computer

Description automatically generated

1. **Completed Tasks Window:**

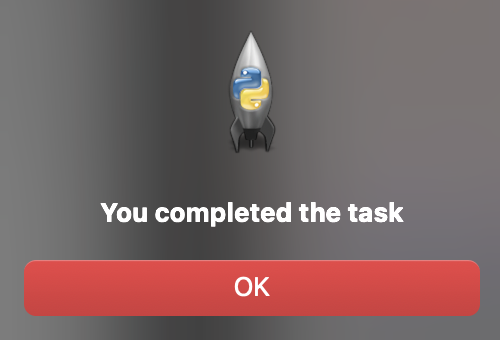
A screenshot of a computer

Description automatically generated

**Outputs**

* Tasks added are displayed in the main window.
* Completed tasks move to the completed tasks list and are shown in a separate window.
* Pop-ups confirm successful actions like marking tasks as complete.

A screenshot of a computer

Description automatically generated

**Conclusion**

**Results**

The SD Task Manager successfully allows users to manage tasks by adding, completing, and viewing them in an organized interface. All features work as expected.

**Error Analysis**

* **Error:** Tasks were not saving properly when input fields were left empty.
  + **Resolution:** Added checks to ensure all fields are filled before saving.
* **Error:** When wanting to make a frame saying “Task Successfully saved” my other code wouldn’t work
  + **Resolution:** Had to make my add task window a frame in the window so they swap.

**Improvements**

* Include an "Edit Task" feature to modify task details.
* Add sorting options (e.g., by priority or due date).
* Improve the user interface with color-coded priorities.
* Add transition effects

**Reflection**

This project helped me understand GUI programming and how to manage user input effectively. I also improved my ability to structure programs using modular functions. I also made it my muscle memory to add comments to what is what code.