**Project Title:** To-Do List Application

**Developed by:** Harshitt Singhrowa [23FE10CSE00838]

**Date:** 22/11/2024

**1. Introduction**

The To-Do List application is a command-line and GUI-based tool that allows users to manage their tasks efficiently. The application is built using Python and employs file handling to store tasks persistently. The GUI is developed using the PySimpleGUI library for a simple and user-friendly interface.

**2. Objective**

The primary objective of the project is to create a functional and interactive to-do list application that can add, show, edit, complete, and remove tasks. The project also demonstrates essential Python concepts, such as file handling, exception handling, and GUI design.

**3. Features**

1. **Add a Task**: Users can add new tasks to their to-do list.
2. **Show Tasks**: Users can view all added tasks, displayed in a numbered format for easy reference.
3. **Edit a Task**: Users can modify an existing task.
4. **Complete a Task**: Users can mark a task as completed, which removes it from the list.
5. **Exit**: Users can safely exit the application.

**4. Technical Overview**

**Programming Language and Libraries**

* **Python**: The main programming language used.
* **PySimpleGUI**: Used to create a graphical interface for the application.
* **Time**: For displaying current time information in the application.

**File Handling**

* The tasks are saved in a file named todos.txt, which allows tasks to persist across different sessions.

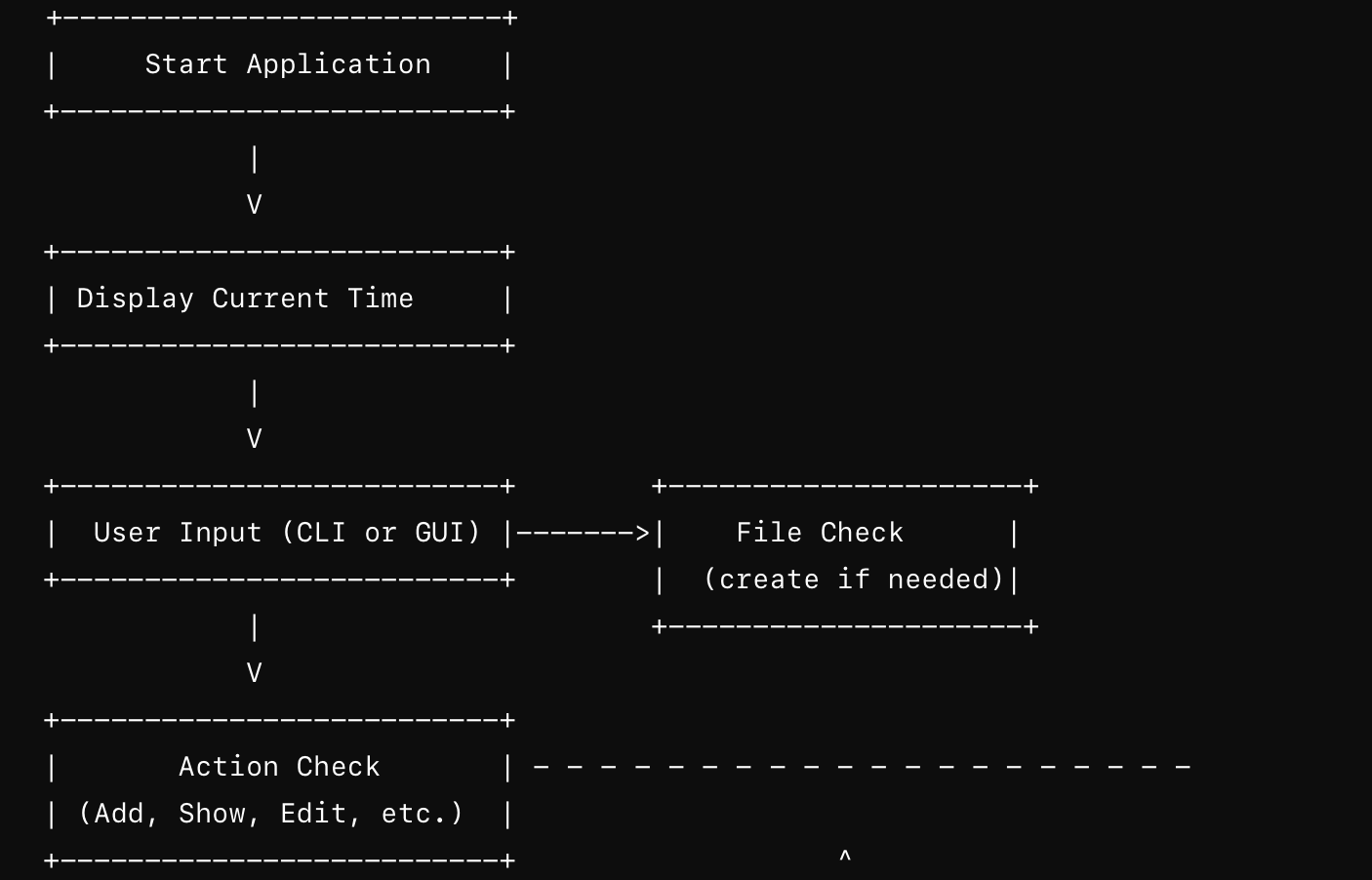
**5. Code Execution Flow**

**Code Structure:**

* **get\_todos()**: Reads tasks from todos.txt and returns them as a list.
* **write\_todos()**: Writes the list of tasks to todos.txt.
* **Main Program Loop (CLI)**: Handles user commands (add, show, edit, complete, exit) and calls appropriate functions.
* **GUI Components (PySimpleGUI)**: Implements buttons and input fields to perform similar actions as the CLI version but in a graphical format.

**6. Block Diagram of Code Execution**

**Below is a block diagram that explains the flow of the program, from user input to file storage:**

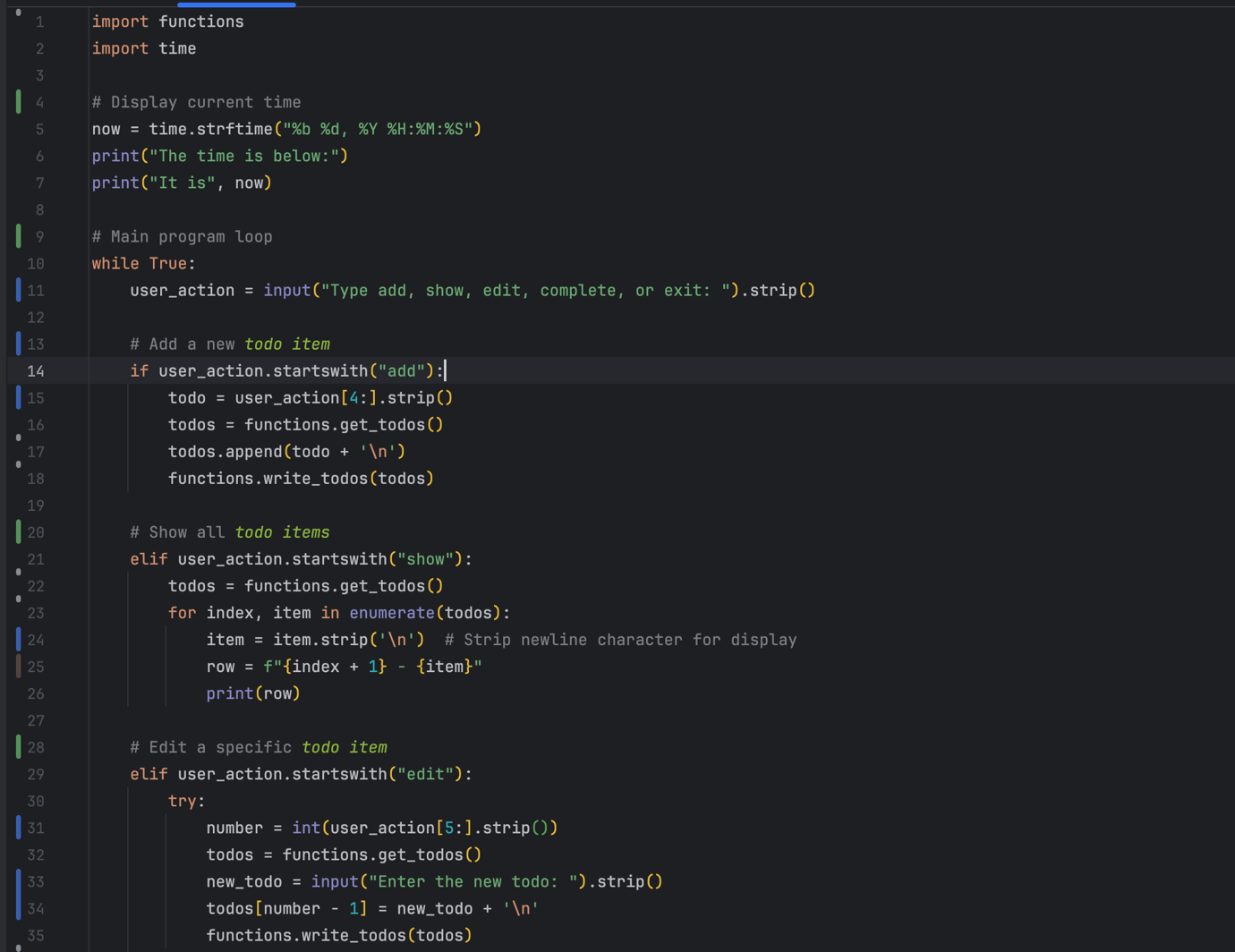


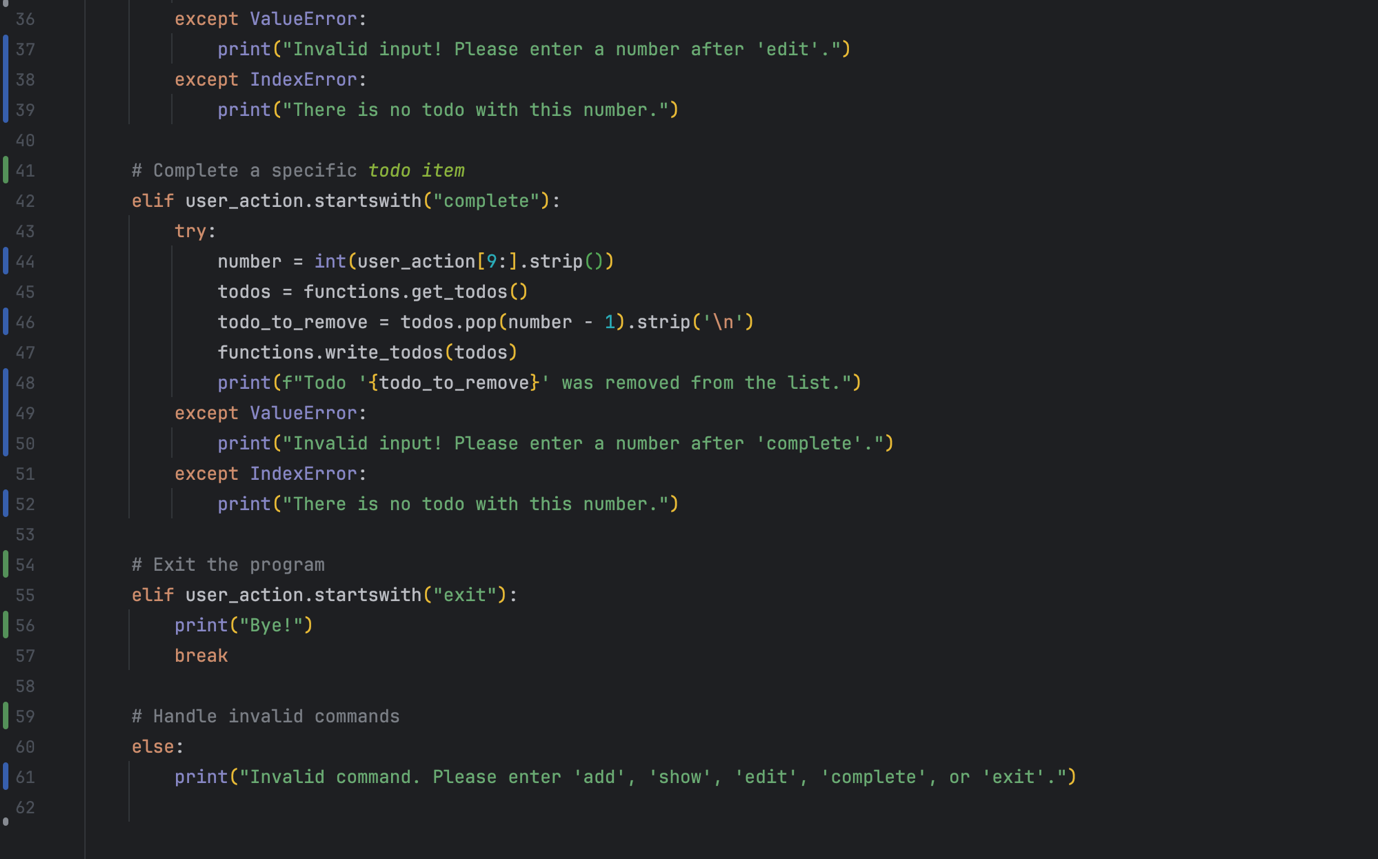
A computer screen with white text

Description automatically generated

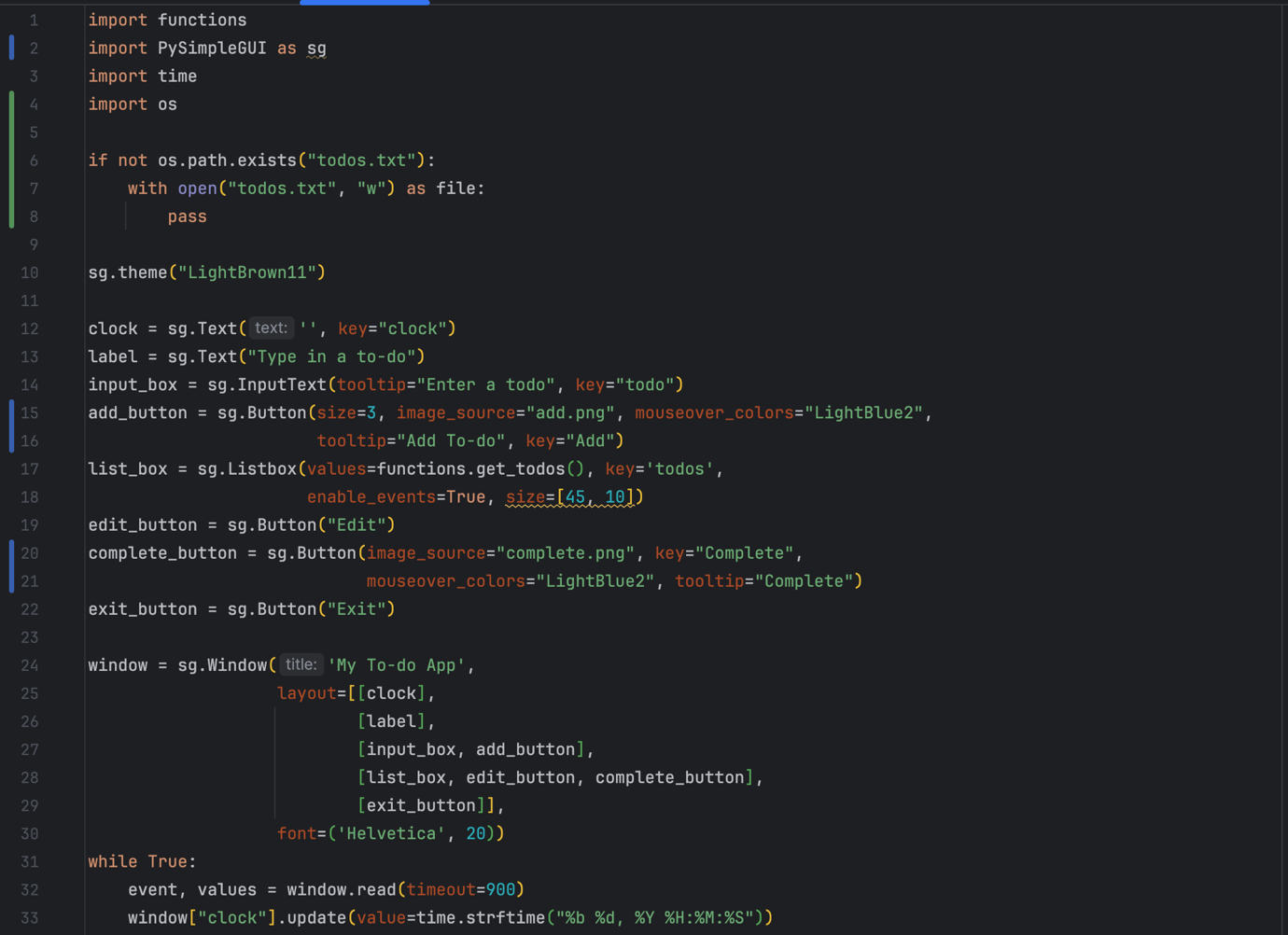
**7. Codes**

**1) Cli.py**

****

****

**2) gui.py**

****

****

**A computer screen shot of a computer code

Description automatically generated**

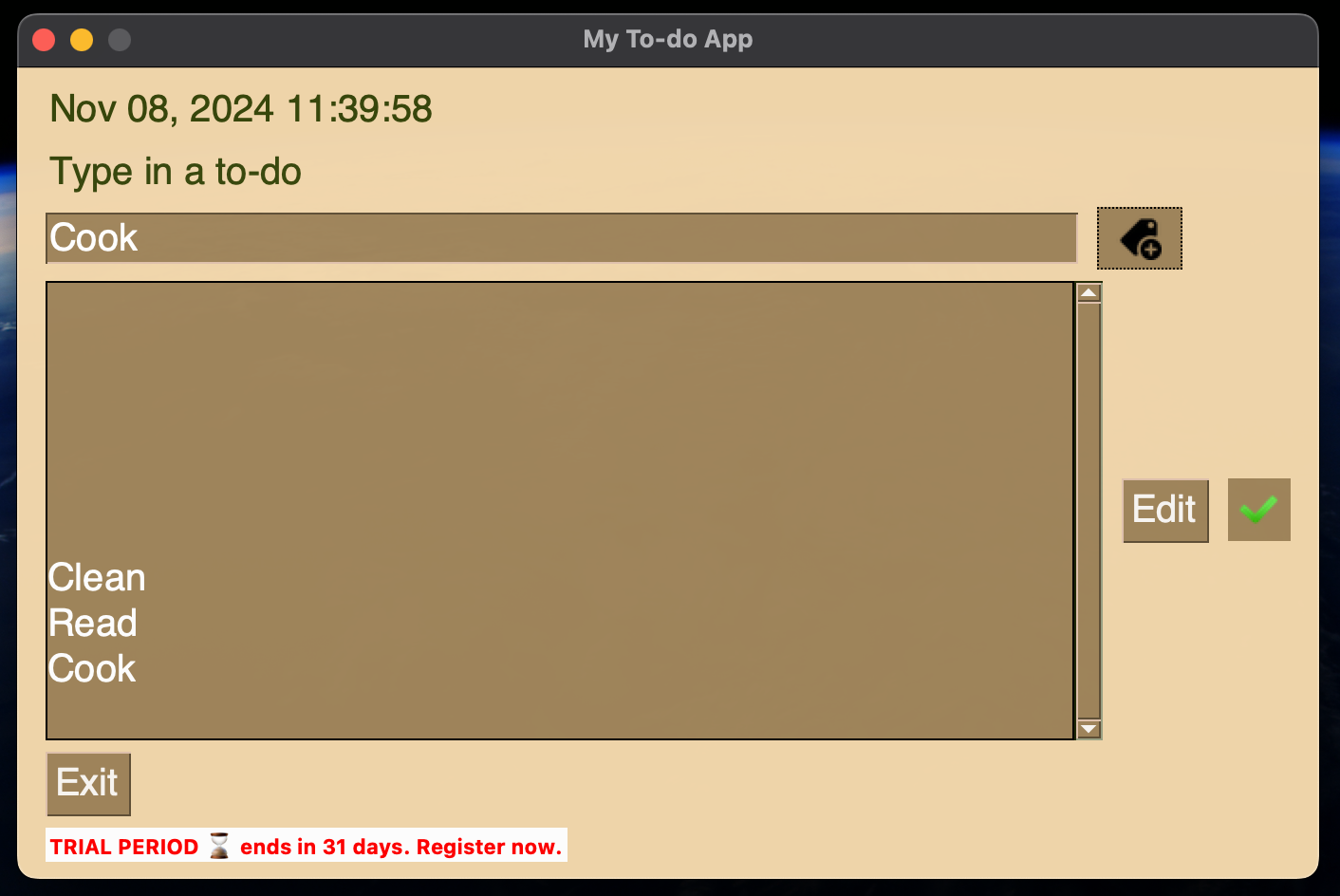
**3) Functions.py**

****

**To-Do List Application Code Execution Flow:**

1. **User Input  
    ↓**
2. **Main Program**
   * **Checks command type (e.g., add, show, edit, complete, exit)**
   * **Routes to the corresponding function or GUI component  
     ↓**
3. **Functions**
   * **get\_todos(): Reads tasks from todos.txt**
   * **write\_todos(): Writes updated tasks to todos.txt  
     ↓**
4. **File Storage**
   * **Saves tasks to todos.txt for persistence across sessions**

**8. Output:**

****

**9. Conclusion**

The To-Do List Application is a helpful productivity tool that allows users to manage tasks efficiently. Through this project, I gained experience in using Python's file handling capabilities and developed a functional user interface using PySimpleGUI.