**TASK 1 TO-DO LIST**

A To-Do List application is a useful project that helps users manage

and organize their tasks efficiently. This project aims to create a

command-line or GUI-based application using Python, allowing

users to create, update, and track their to-do lists

Ans:-

Command-Line To-Do List Application

User Interface: Design a command-line interface (CLI) using libraries like argparse or custom input/output handling to take user input and display tasks.

Task Data Structure: Define a data structure for tasks. Each task should have attributes like a unique ID, description, due date, status (completed or not), and any other relevant information.

Task Management Functions:

Add a new task.

List all tasks.

Update a task (mark as completed, change description, change due date, etc.).

Delete a task.

File Storage: Implement a way to save and load tasks from a file (e.g., a JSON or CSV file) so that the tasks persist between sessions.

User Input Handling: Create functions to handle user input, parse command-line arguments, and execute the appropriate task management functions.

User Feedback: Provide clear and informative feedback to the user after each operation.

Date Handling: Implement date handling, including parsing and formatting dates for tasks with due dates.

Error Handling: Add error handling to manage unexpected situations gracefully.

GUI-Based To-Do List Application

For a GUI-based To-Do List application, you can use a GUI library like Tkinter, PyQt, or Kivy. Here's an outline for a Tkinter-based application:

User Interface: Create a GUI with a window that allows users to interact with the application.

Task List Display: Add a widget (e.g., a listbox or a treeview) to display the list of tasks with their details.

Add Task Form: Provide a form for users to enter new task details, such as description, due date, and any other relevant information.

Update and Delete Functionality: Implement buttons or context menus to update or delete tasks. You can use pop-up windows for editing tasks.

Status Management: Allow users to mark tasks as completed or uncompleted.

File I/O: Save and load tasks to/from a file so that tasks persist between sessions.

Date Picker: Implement a date picker widget to facilitate date selection for due dates.

Error Handling: Handle errors, such as invalid input or file I/O issues.

User Feedback: Provide feedback through message boxes or labels to inform users about the status of their actions.

Keyboard Shortcuts: Consider adding keyboard shortcuts for common actions to enhance usability.