TO-DO LIST APPLICATION - COMMAND-LINE AND GUI BASED

OPTION 1: Command-Line Based To-Do App

Technologies:

- Python 3
- File Handling (.txt or .json) or SQLite (optional)
- Libraries: datetime, json

Sample Code:

```
import json
import os

FILENAME = "todo.json"

def load_tasks():
    if os.path.exists(FILENAME):
        with open(FILENAME, "r") as f:
            return json.load(f)
    return []

def save_tasks(tasks):
    with open(FILENAME, "w") as f:
        json.dump(tasks, f, indent=4)

def show_tasks(tasks):
    print("\nYour To-Do List:")
    if not tasks:
        print("No tasks yet!")
    for idx, task in enumerate(tasks, 1):
        status = "Done" if task["done"] else "Not Done"
        print(f"{idx}. [{status}] {task['task']}")

def add_task(tasks):
    task_text = input("Enter a new task: ")
    tasks.append({"task": task_text, "done": False})

def mark_done(tasks):
    show_tasks(tasks)
    index = int(input("Enter task number to mark as done: ")) - 1
    if 0 <= index < len(tasks):
        tasks[index]["done"] = True
    else:
        print("Invalid index!")

def delete_task(tasks):
    show_tasks(tasks)
    index = int(input("Enter task number to delete: ")) - 1
    if 0 <= index < len(tasks):
        del tasks[index]
    else:
        print("Invalid index!")

def main():
    tasks = load_tasks()
    while True:
        print("\n--- TO-DO MENU ---")
        print("1. Show tasks")
        print("2. Add task")
        print("3. Mark task as done")
        print("4. Delete task")
        print("5. Exit")

        choice = input("Choose an option: ")

        if choice == "1":
            show_tasks(tasks)
        elif choice == "2":
            add_task(tasks)
        elif choice == "3":
            mark_done(tasks)
        elif choice == "4":
            delete_task(tasks)
        elif choice == "5":
            save_tasks(tasks)
            print("Tasks saved. Goodbye!")
            break
        else:
            print("Invalid choice! Try again.")

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

OPTION 2: GUI-Based Version (Optional)

Technologies:

- Python 3
- Tkinter (built-in GUI library)
- JSON or SQLite for saving tasks

Features:

- GUI for adding, deleting, marking tasks
- Scrollable task list
- Persistent storage

Let me know if you'd like the GUI version code using Tkinter.