A

Project Report on

TO – DO LIST

Submitted in partial fulfillment of the requirements for

the degree of

Bachelor of Technology

in

Computer Science Engineering

BY

KONDAL SRINIDHI SAGAR

import os

import json

from datetime import datetime

class TodoList: def **init**(self): self.tasks = [] self.filename = "tasks.json" self.load_tasks()

```
def add_task(self, description, due_date=None):
    """Add a new task to the list"""
    task = {
        "id": len(self.tasks) + 1,
        "description": description,
        "completed": False,
        "created_at": datetime.now().strftime("%Y-%m-%d %H:%M:%S"),
        "due_date": due_date,
    }
    self.tasks.append(task)
    self.save_tasks()
    return task

def view_tasks(self, show_completed=True):
    """View all tasks or only incomplete tasks"""
    if not self.tasks:
        return "No tasks found."

    result = []
    for task in self.tasks:
        if show_completed or not task["completed"]:
            status = "✓" if task["completed"] else "□"
            due_info = f" (Due: {task['due_date']})" if
task["due_date"] else ""
            result.append(f"{task['id']}. [{status}]
{task['description']}{due_info}")

    return "\n".join(result)

def complete_task(self, task_id):
    """Mark a task as completed"""
    for task in self.tasks:
        if task["id"] == task_id:
            task["completed"] = True
            self.save_tasks()
            return f"Task '{task['description']}' marked as
completed."
    return f"Task with ID {task_id} not found."

def delete_task(self, task_id):
    """Delete a task from the list"""
    for i, task in enumerate(self.tasks):
        if task["id"] == task_id:
            deleted_task = self.tasks.pop(i)
            self.save_tasks()
            return f"Task '{deleted_task['description']}' deleted."
    return f"Task with ID {task_id} not found."

def save_tasks(self):
    """Save tasks to a JSON file"""
    with open(self.filename, "w") as f:
        json.dump(self.tasks, f, indent=2)

def load_tasks(self):
    """Load tasks from a JSON file"""
    if os.path.exists(self.filename):
        try:
            with open(self.filename, "r") as f:
                self.tasks = json.load(f)
        except json.JSONDecodeError:
            self.tasks = []
    else:
        self.tasks = []


def main(): todo = TodoList()

while True:
    print("\n===== TO-DO LIST APPLICATION =====")
    print("1. Add a task")
    print("2. View all tasks")
    print("3. View incomplete tasks")
    print("4. Mark a task as complete")
    print("5. Delete a task")
    print("6. Exit")

    choice = input("\nEnter your choice (1-6): ")

    if choice == "1":
        description = input("Enter task description: ")
        due_date = input("Enter due date (optional, format YYYY-MM-DD): ")
        due_date = due_date if due_date else None
        task = todo.add_task(description, due_date)
        print(f"Task added: {task['description']}")

    elif choice == "2":
        print("\n--- ALL TASKS ---")
        print(todo.view_tasks(show_completed=True))

    elif choice == "3":
        print("\n--- INCOMPLETE TASKS ---")
        print(todo.view_tasks(show_completed=False))

    elif choice == "4":
        task_id = input("Enter the ID of the task to mark as complete: ")
        try:
            task_id = int(task_id)
            print(todo.complete_task(task_id))
        except ValueError:
            print("Please enter a valid task ID (number).")

    elif choice == "5":
        task_id = input("Enter the ID of the task to delete: ")
        try:
            task_id = int(task_id)
            print(todo.delete_task(task_id))
        except ValueError:
            print("Please enter a valid task ID (number).")

    elif choice == "6":
        print("Thank you for using the To-Do List application!")
        break

    else:
        print("Invalid choice. Please try again.")
```

For demonstration purposes, let's create some sample tasks and show functionality

if **name** == "**main**": # Instead of running the interactive menu, let's demonstrate the functionality todo = TodoList()

```
# Add some tasks
print("Adding sample tasks...")
todo.add_task("Complete Python project", "2025-05-25")
todo.add_task("Buy groceries", "2025-05-20")
todo.add_task("Call mom")

# View all tasks
print("\nAll tasks:")
print(todo.view_tasks())

# Mark a task as complete
print("\nMarking task 2 as complete:")
print(todo.complete_task(2))

# View tasks again
print("\nAll tasks after completion:")
print(todo.view_tasks())

# View only incomplete tasks
print("\nIncomplete tasks:")
print(todo.view_tasks(show_completed=False))
```

```
# Delete a task
print("\nDeleting task 3:")
print(todo.delete_task(3))

# View final task list
print("\nFinal task list:")
print(todo.view_tasks())

print("\nTo use the interactive menu, run this script and follow the prompts!")
# Uncomment the line below to run the interactive menu
# main()
```