TechInterex **INTERNS**

PYTHON DEVELOPMENT TASK 1

Given By: Divya Rajput (HR)

Simple To-Do List Application

Here’s a beginner-friendly Python development project that will help you get started with programming concepts, file handling, and user interaction.

Task for Interns: Simple To-Do List Application

Objective:

Create a console-based To-Do List application that allows users to add, view, update, and delete tasks. This project will help you understand basic Python concepts, including data structures, functions, and file handling.

Technologies:

- **Programming Language**: Python
- **Environment**: Any Python IDE (e.g., PyCharm, VSCode, or even a simple text editor)

Project Requirements:

1. **Project Setup**:
    - Create a new directory for your project.
    - Create a Python file named todo_list.py.
2. **Define a Data Structure**:
    - Use a list to store tasks. Each task can be a dictionary containing:
        - ➢ id: An integer to uniquely identify the task.
        - ➢ title: A string representing the task title.
        - ➢ completed: A boolean indicating whether the task is completed.
3. **Functions to Implement**:
    - **Add Task**:
        - ➢ Create a function to add a new task to the list.
        - ➢ Prompt the user for a task title.
        - ➢ Append the new task to the list with a unique ID and set completed to False.
    - **View Tasks**:
        - ➢ Create a function to display all tasks.
        - ➢ Show the task ID, title, and completion status.
    - **Update Task**:
        - ➢ Create a function to update a task's completion status.
        - ➢ Prompt the user for the task ID and toggle the completed status.
    - **Delete Task**:

        - Create a function to delete a task based on its ID.
        - Prompt the user for the task ID and remove it from the list.
    - **Save Tasks**:
        - Implement functionality to save tasks to a file (e.g., tasks.txt) when the application exits.
        - Load tasks from the file when the application starts.
4. **User Interface**:
    - Implement a simple text-based menu that allows users to choose an action:
        - 1: Add Task
        - 2: View Tasks
        - 3: Update Task
        - 4: Delete Task
        - 5: Exit
5. **Main Loop**:
    - Create a loop that displays the menu and prompts the user for their choice until they choose to exit.
    - Call the corresponding function based on the user's input.
6. **Testing**:
    - Test each functionality thoroughly to ensure that tasks can be added, viewed, updated, and deleted without issues.
    - Check that tasks are saved to and loaded from the file correctly.

Submission:

- Submit your completed Python script along with any additional documentation or notes you created during the project.

Summary:

This **To-Do List Application** will help beginners understand fundamental programming concepts in Python, including data handling, functions, user input, and file operations. It's a practical project that can be expanded with more features, such as deadlines for tasks or user authentication in the future. Enjoy coding!

Example Code Snippet:

Here’s a basic structure to get you started:

```
import json

tasks = []

def load_tasks():
    global tasks
    try:
        with open('tasks.json', 'r') as file:
            tasks = json.load(file)
    except FileNotFoundError:
        tasks = []

def save_tasks():
    with open('tasks.json', 'w') as file:
        json.dump(tasks, file)

def add_task(title):
    task = {
        'id': len(tasks) + 1,
        'title': title,
        'completed': False
    }
    tasks.append(task)

def view_tasks():
    for task in tasks:
        status = "✔" if task['completed'] else "✖"
        print(f"{task['id']}: {task['title']} [{status}]")

def update_task(task_id):
    for task in tasks:
        if task['id'] == task_id:
            task['completed'] = not task['completed']
            break

def delete_task(task_id):
    global tasks
    tasks = [task for task in tasks if task['id'] != task_id]

def main():
    load_tasks()
```

```
while True:
    print("\nTo-Do List")
    print("1. Add Task")
    print("2. View Tasks")
    print("3. Update Task")
    print("4. Delete Task")
    print("5. Exit")
    choice = input("Choose an option: ")

    if choice == '1':
        title = input("Enter task title: ")
        add_task(title)
    elif choice == '2':
        view_tasks()
    elif choice == '3':
        task_id = int(input("Enter task ID to update: "))
        update_task(task_id)
    elif choice == '4':
        task_id = int(input("Enter task ID to delete: "))
        delete_task(task_id)
    elif choice == '5':
        save_tasks()
```

```
            break
        else:
            print("Invalid choice, please try again.")

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