DOCUMENTATION FOR PYTHON CODE

Name of the Intern : N.V.S.S HARSHAVARDHAN

Intern ID : SMI65278

Name of the Internship : Python

Level of the task done : Basic

Task Name : To-Do List

Software Used : Python IDLE 3.11 64 bit

To-Do List Application

This Python code implements a simple command-line to-do list application. It allows you to create, view, mark complete, and remove tasks.

Features:

- Stores tasks in a list
- Displays the current list of tasks
- Adds new tasks
- Marks tasks as complete and removes them from the list
- Removes tasks by index
- User-friendly menu for interaction

Code Structure:

1. **Task List:**
   - An empty list named tasks is created to store the to-do items.
2. **Functions:**
   - **show_tasks():**
     - Displays the current list of tasks. If the list is empty, it prints a message indicating that.
     - Iterates through the tasks list and prints each task with its corresponding index (starting from 1).
   - **add_task():**
     - Prompts the user to enter a new task using input().

- Appends the new task to the tasks list using append().
- Prints a success message.

- **complete_task():**
  - Calls show_tasks() to display the current list (optional for reference).
  - If there are tasks:
  - Calls show_tasks() again.
  - Prompts the user to enter the number of the task to mark complete.
  - Handles potential errors (invalid index or non-integer input) using a try-except block:
  - If the input is valid, removes the task at the specified index using pop().
  - Prints a success message.
  - If there's an error, prints an "Invalid task number!" message.
  - If there are no tasks, prints a message indicating that.
- **remove_task():**
  - Similar to complete_task(), it displays the current list, handles potential errors, and removes the task based on the user's input.

3. Main Loop:

- An infinite loop (while True) is used to continuously prompt the user for input.
- Inside the loop:
  - Prints a menu with options to view tasks, add tasks, mark tasks complete, remove tasks, and exit the application.
  - Prompts the user to enter their choice (1-5).
  - Uses an if-elif-else statement to handle the user's choice:
    - Choice 1: Calls show_tasks().
    - Choice 2: Calls add_task().
    - Choice 3: Calls complete_task().
    - Choice 4: Calls remove_task().
    - Choice 5: Prints a quit message, breaks out of the loop (ending the program).
    - Invalid choices: Prints an "Invalid choice" message.

Running the Application:

1. Save the code as a Python file (e.g., todo.py).
2. Open a terminal or command prompt and navigate to the directory where you saved the file.
3. Run the script using python todo.py.

Python Code:

```
# Define an empty list to store tasks
tasks = []
# Function to display the to-do list
def display_tasks():
    if not tasks:
        print("Your to-do list is empty.")
    else:
        print("To-Do List:")
        for i, task in enumerate(tasks, start=1):
            status = "Done" if task["completed"] else "Not Done"
            print(f"{i}. {task['task']} ({status})")

# Function to add a task to the to-do list
def add_task(task_name):
    task = {"task": task_name, "completed": False}
    tasks.append(task)
    print(f"Task '{task_name}' added to your to-do list.")

# Function to mark a task as completed
def mark_completed(task_number):
    try:
        task_number = int(task_number)  # Handle potential non-numeric input
        if 1 <= task_number <= len(tasks):
            tasks[task_number - 1]["completed"] = True
            print(f"Task {task_number} marked as completed.")
        else:
            print("Invalid task number. Please enter a valid task number.")
    except ValueError:
        print("Invalid task number. Please enter a number.")
# Function to remove a task from the to-do list
def remove_task(task_number):
    try:
        task_number = int(task_number)  # Ensure numeric input
        if 1 <= task_number <= len(tasks):
```

```
            removed_task = tasks.pop(task_number - 1)
            print(f"Task '{removed_task['task']}' removed from your to-do list.")
        else:
            print("Invalid task number. Please enter a valid task number.")
    except ValueError:
        print("Invalid task number. Please enter a number.")
# Main program loop
while True:
    print("\nOptions:")
    print("1. Display to-do list")
    print("2. Add a task")
    print("3. Mark a task as completed")
    print("4. Remove a task")
    print("5. Quit")
    choice = input("Enter your choice: ")
    if choice in ('1', '2', '3', '4', '5'):  # Check for valid choices
        if choice == '1':
            display_tasks()
        elif choice == '2':
            task_name = input("Enter the task: ")
            add_task(task_name)
        elif choice == '3':
            display_tasks()
            task_number = input("Enter the task number to mark as completed: ")
            mark_completed(task_number)
        elif choice == '4':
            display_tasks()
            task_number = input("Enter the task number to remove: ")
            remove_task(task_number)
        elif choice == '5':
            break
    else:
        print("Invalid choice. Please enter a valid option.")
```

Example Usage:

Options:

1. Display to-do list
2. Add a task
3. Mark a task as completed
4. Remove a task
5. Quit

Enter your choice : 1

To-Do List:

1. Buy groceries (Not Done)

Enter your choice : 3

To-Do List:

1. Buy groceries (Not Done)

Enter the task number to mark as completed: 1

Task 1 marked as completed.