**Task 2 : Create a To-Do List Application (Console-based) :**

**To-Do List CLI App (Python)**

This is a simple command-line To-Do List application built in Python. It allows users to add, view, and remove tasks using a menu-driven interface. Tasks are saved in a file, so they remain even after restarting the program.

**Features**

* Add tasks
* View all tasks
* Remove tasks
* Tasks are saved in tasks.txt for persistence
* Input validation to avoid crashes
* Menu-driven CLI using a loop
* Modular code using functions

**How to Run**

1. Make sure you have Python installed.
2. Save the file as todo.py
3. Open your terminal or PowerShell.
4. Navigate to the folder where todo.py is saved:

cd path\to\your\folder

1. Run the program:

python todo.py

**Code Explanation:**

**1. Function Definitions**

def load\_tasks():

...

def save\_tasks(tasks):

...

def add\_task(task):

...

def remove\_task(index):

...

def view\_tasks():

...

* **load\_tasks()**: Reads tasks from tasks.txt into a list.
* **save\_tasks(tasks)**: Saves the current list of tasks to the file.
* **add\_task(task)**: Adds a new task to the list.
* **remove\_task(index)**: Deletes a task based on its number.
* **view\_tasks()**: Shows all the tasks with numbering.

These functions keep the code **modular, clean, and reusable**.

**2. Menu Interface**

def show\_menu():

print("1. View Tasks")

print("2. Add Task")

print("3. Remove Task")

print("4. Exit")

* Displays user options.
* Called in a loop so the menu keeps showing after each action.

**3. Main To-Do List Loop**

def main():

while True:

show\_menu()

choice = input("Choose an option (1-4): ")

if choice == '1':

view\_tasks()

elif choice == '2':

task = input("Enter the task: ")

add\_task(task)

elif choice == '3':

view\_tasks()

try:

task\_num = int(input("Enter task number to remove: ")) - 1

remove\_task(task\_num)

except ValueError:

print("Please enter a valid number.")

elif choice == '4':

print("Goodbye!")

break

else:

print("Invalid choice.")

* Uses a while True loop to keep the program running until the user chooses **Exit (4)**.
* Gets user input and calls the appropriate function.
* Handles errors for invalid choices and non-numeric input during task removal.

**4. Main Entry Point**

if \_\_name\_\_ == "\_\_main\_\_":

main()

* This line ensures the app runs only when you execute todo.py directly.
* It prevents the app from running if the file is imported somewhere else.

Sample output when run in terminal using powershell:



Navigate to the folder where task2.py is saved and run:

A screenshot of a computer program

AI-generated content may be incorrect.

The option displays for user to select operation. Choose either View Tasks, Add Task, Remove Task or Exit.

So, I chose option 1 to View Tasks. The output is displayed which are manreenyaseen and vamshi.

Next, I chose option 2 to Add Task. Add the task sheikh.

A black screen with white text

AI-generated content may be incorrect.

Next, I chose option 1 to view task The output is displayed which are manreenyaseen, vamshi, and sheikh.

A black screen with white text

AI-generated content may be incorrect.

Next, I chose option 3 to Remove Task. So remove 2 which is vamshi.

A black screen with white text

AI-generated content may be incorrect.

Then choose option 3 to View Tasks. It displays manreenyaseen and sheikh only because vamshi is removed.

A black screen with white text

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

Then choose option 4 to Exit. The output will display Goodbye!

INTERVIEW QUESTIONS - PYTHON DEVELOPER INTERNSHIP

1. What is normalization?

Organizing data to reduce redundancy and improve integrity.

2. Explain primary vs foreign key.

Primary key uniquely identifies records; foreign key links to a primary key in another table.

3. What are constraints?

Rules that enforce data integrity (e.g. NOT NULL, UNIQUE).

4. What is a surrogate key?

An artificially generated key used to uniquely identify a record.

5. How do you avoid data redundancy?

Normalize data, enforce foreign keys, and avoid data duplication.

6. What is ER diagram?

A visual model of entities and their relationships in a database.

7. What are the types of relationships in DBMS?

One-to-One, One-to-Many, Many-to-One, Many-to-Many.

8. Explain the purpose of AUTO\_INCREMENT.

Automatically generates a unique ID for new rows in a table.

9. What is the default storage engine in MySQL?

InnoDB.

10. What is a composite key?

A primary key made from multiple columns together.