

**Name : Abdul Rehman Raza**

**Section : BSDS:3A**

**Subject : Artificial Intelligence (Theory)**

**Roll Number “ 052 “**

**GitHub Link : “**[**https://github.com/AbdulRehmanRaza03/AI-Lab-**](https://github.com/AbdulRehmanRaza03/AI-Lab-)**”**

**Linkedin Profile Link : “**[**https://www.linkedin.com/in/abdul-rehman-raza-7a125b332/?trk=public-profile-join-page**](https://www.linkedin.com/in/abdul-rehman-raza-7a125b332/?trk=public-profile-join-page)**”**

**Project : Short explanation**

This project is a simple **console-based To-Do List Maker** built in Python. It first asks the user how many tasks they want to add and stores them in a list. A menu-driven loop is then displayed where the user can **add new tasks, update existing ones, view all tasks, or delete tasks**. The list is managed using basic Python operations such as **lists, loops, conditional statements, indexing, and deletion**. The program continues running until the user selects the exit option. The main concept used in this project is the **list data structure** along with **control flow (if-else, while loop)** to create an interactive menu system.

**Project Explanation step by step :**

**Step 1: Start of the program**

* The program begins by showing a welcome message: *“Welcome to My To\_Do\_List Maker”*.
* It then asks the user: *“How many tasks do you want to add?”*

**Step 2: Add initial tasks**

* The user enters a number (for example, 5).
* The program then asks for that many tasks one by one.
* Each entered task is stored inside a list called tasks.
* Finally, it prints the complete list of tasks that the user entered.

**Step 3: Menu loop begins**

* After showing the initial tasks, the program enters an infinite loop (while True).
* Inside this loop, the program shows a menu with **5 options**:
  1. Add new task
  2. Update an existing task
  3. View all tasks
  4. Delete a task
  5. Exit the program
* The user types a number between 1 and 5 to choose an operation.

**Step 4: If user chooses 1 (Add task)**

* The program asks: *“Enter task you want to add”*.
* Whatever the user types gets added to the list of tasks.
* A success message is displayed confirming the addition.

**Step 5: If user chooses 2 (Update task)**

* The program asks: *“Enter the task name you want to update”*.
* If that task exists in the list, it then asks for the new task name.
* The old task is replaced with the new one.
* A message is shown: *“updated\_task …”*.
* If the task is not found, nothing happens.

**Step 6: If user chooses 4 (Delete task)**

* The program asks: *“Which task do you want to delete?”*
* If the entered task exists in the list, it gets removed.
* A message confirms that the task has been deleted.
* If the task is not found, nothing happens.

**Step 7: If user chooses 3 (View tasks)**

* The program simply prints all tasks currently stored in the list.

**Step 8: If user chooses 5 (Exit)**

* The program prints *“You exit the project.”*
* It breaks the loop and the program ends.

**Step 9: If user types any other number**

* The program prints: *“Invalid input”* and shows the menu again.

👉 In short:

* **Start:** Ask how many tasks and store them.
* **Menu loop:** Repeatedly let the user add, update, view, delete, or exit.
* **Exit:** When option 5 is chosen, the program ends.