MINI PROJECT:

Develop a basic to-do -list application using function and data structures.

Project Overview:

Objective: Develop a simple to-do list application using Java with an emphasis on functions and data structures.

Key Components:

1. Functions (Methods): In Java, functions are referred to as methods. You'll be implementing various methods to handle different aspects of the to-do list application. Methods are modular blocks of code that perform specific tasks, making your code more organized and easier to understand.
Method to add a task

Method to delete a task

Method to display the list of tasks

Method to mark a task as complete

2. Data Structures: Utilize appropriate data structures to store and manage the to-do list. Common choices in Java include ArrayList, LinkedList, or HashMap, but you can explore other options based on your creativity and understanding.

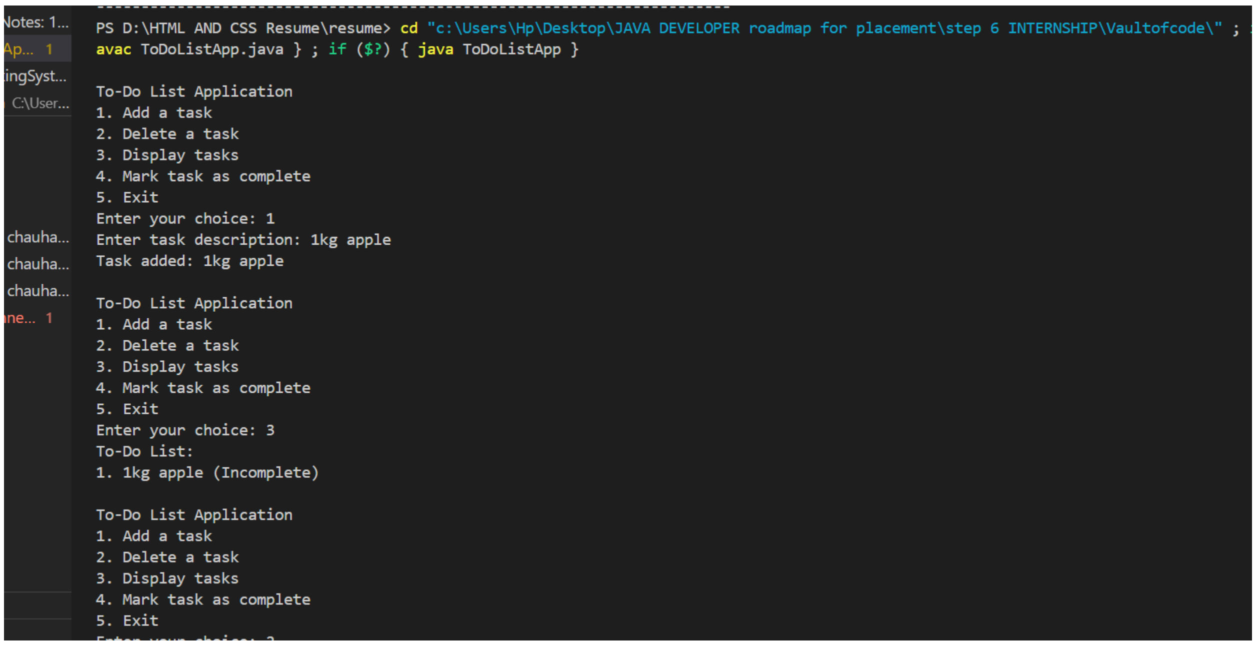
CODE:

```java
import java.util.ArrayList;
import java.util.Scanner;

public class ToDoListApp {
    // Data structure to store tasks
    private ArrayList<Task> tasks;

    // Constructor
    public ToDoListApp() {
        tasks = new ArrayList<>();
    }
    // Method to add a task
    public void addTask(String description) {
        Task newTask = new Task(description);
        tasks.add(newTask);
        System.out.println("Task added: " + description);
    }

    // Method to delete a task
    public void deleteTask(int taskId) {
        if (taskId < 1 || taskId > tasks.size()) {
            System.out.println("Invalid task ID.");
        } else {
            Task removedTask = tasks.remove(taskId - 1);
            System.out.println("Task deleted: " + removedTask.getDescription());
        }
    }

    // Method to display the list of tasks
    public void displayTasks() {
        if (tasks.isEmpty()) {
            System.out.println("No tasks in the list.");
        } else {
            System.out.println("To-Do List:");
            for (int i = 0; i < tasks.size(); i++) {
                Task task = tasks.get(i);
                System.out.println((i + 1) + ". " + task);
            }
        }
    }

    // Method to mark a task as complete
    public void markTaskAsComplete(int taskId) {
        if (taskId < 1 || taskId > tasks.size()) {
            System.out.println("Invalid task ID.");
        } else {
            Task task = tasks.get(taskId - 1);
            task.setComplete(true);
            System.out.println("Task marked as complete: " +
task.getDescription());
        }
    }

    // Main method to run the application
    public static void main(String[] args) {
```

```java
        ToDoListApp toDoListApp = new ToDoListApp();
        Scanner scanner = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\nTo-Do List Application");
            System.out.println("1. Add a task");
            System.out.println("2. Delete a task");
            System.out.println("3. Display tasks");
            System.out.println("4. Mark task as complete");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    System.out.print("Enter task description: ");
                    String description = scanner.nextLine();
                    toDoListApp.addTask(description);
                    break;
                case 2:
                    System.out.print("Enter task ID to delete: ");
                    int deleteId = scanner.nextInt();
                    toDoListApp.deleteTask(deleteId);
                    break;
                case 3:
                    toDoListApp.displayTasks();
                    break;
                case 4:
                    System.out.print("Enter task ID to mark as complete: ");
                    int completeId = scanner.nextInt();
                    toDoListApp.markTaskAsComplete(completeId);
                    break;
                case 5:
                    System.out.println("Exiting...");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 5);

        scanner.close();
    }
}
```

```java
// Task class to represent a task in the to-do list
class Task {
    private String description;
    private boolean isComplete;

    public Task(String description) {
        this.description = description;
        this.isComplete = false;
    }

    public String getDescription() {
        return description;
    }

    public boolean isComplete() {
        return isComplete;
    }

    public void setComplete(boolean isComplete) {
        this.isComplete = isComplete;
    }

    @Override
    public String toString() {
        return description + (isComplete ? " (Complete)" : " (Incomplete)");
    }
}
```

OUTPUT:



```
Notes: 1...
Ap...  1
ingSyst...
 C:\User...

chauha...
chauha...
chauha...
ane...  1
```

```
PS D:\HTML AND CSS Resume\resume> cd "c:\Users\Hp\Desktop\JAVA DEVELOPER roadmap for placement\step 6 INTERNSHIP\Vaultofcode\" ;
avac ToDoListApp.java } ; if ($?) { java ToDoListApp }

To-Do List Application
1. Add a task
2. Delete a task
3. Display tasks
4. Mark task as complete
5. Exit
Enter your choice: 1
Enter task description: 1kg apple
Task added: 1kg apple

To-Do List Application
1. Add a task
2. Delete a task
3. Display tasks
4. Mark task as complete
5. Exit
Enter your choice: 3
To-Do List:
1. 1kg apple (Incomplete)

To-Do List Application
1. Add a task
2. Delete a task
3. Display tasks
4. Mark task as complete
5. Exit
```