Institute of Aeronautical Engineering

Department of Information Technology

MINI PROJECT REPORT

SMART TO-DO LIST APP

Submitted by:

Sai Vyshnavi Ronte

Roll No: 24951A1288

Department: Information Technology

Academic Year: 2024–2025

1. Title of the Project

SMART TO-DO LIST – CONSOLE BASED APPLICATION

2. Abstract

The Smart To-Do List is a **menu-driven console application** that helps users manage their daily tasks in a simple and organized way. Users can **add, view, update, mark as completed, and delete tasks** using a clean text-based interface. The project focuses on core programming concepts such as **loops, conditional statements, functions, data structures (arrays/lists)** and optional **file handling** to store tasks permanently. It is a simple yet effective mini project that improves logical thinking and demonstrates how real-life problems can be solved using basic programming.

3. Introduction

In day-to-day life, managing tasks and remembering pending work can be difficult, especially for students and working professionals. Writing tasks on paper can be messy and easy to lose.

This project aims to develop a simple and user-friendly To-Do List App that helps users manage their daily tasks digitally. The application allows users to create a list of tasks, categorize them (like work, study, personal), and track their completion status in a clear interface.

4. Objectives

- To design a **simple task management system** using a console interface.
- To practice **basic programming concepts** (loops, if-else, switch-case, functions).
- To manage tasks with operations like **add, view, update, delete, complete**.
- To optionally implement **file storage** so tasks are not lost after exiting the program.

5. Problem Statement

People often forget important tasks because they have no proper way to track them. Writing tasks in notebooks or random notes can be messy and difficult to maintain. There is a need for a **simple, lightweight tool** that runs even in low-end systems and allows the user to **quickly manage their tasks without any complex setup or installation**.

6. Scope of the Project

The scope of this project is to develop a simple **console-based Smart To-Do List application** that allows the user to add, view, mark as completed, and delete tasks using a menu-driven interface. It focuses only on **basic task management and core programming concepts** (loops, conditions, functions, arrays/lists, and optional file handling), and does not include any graphical user interface or online features.

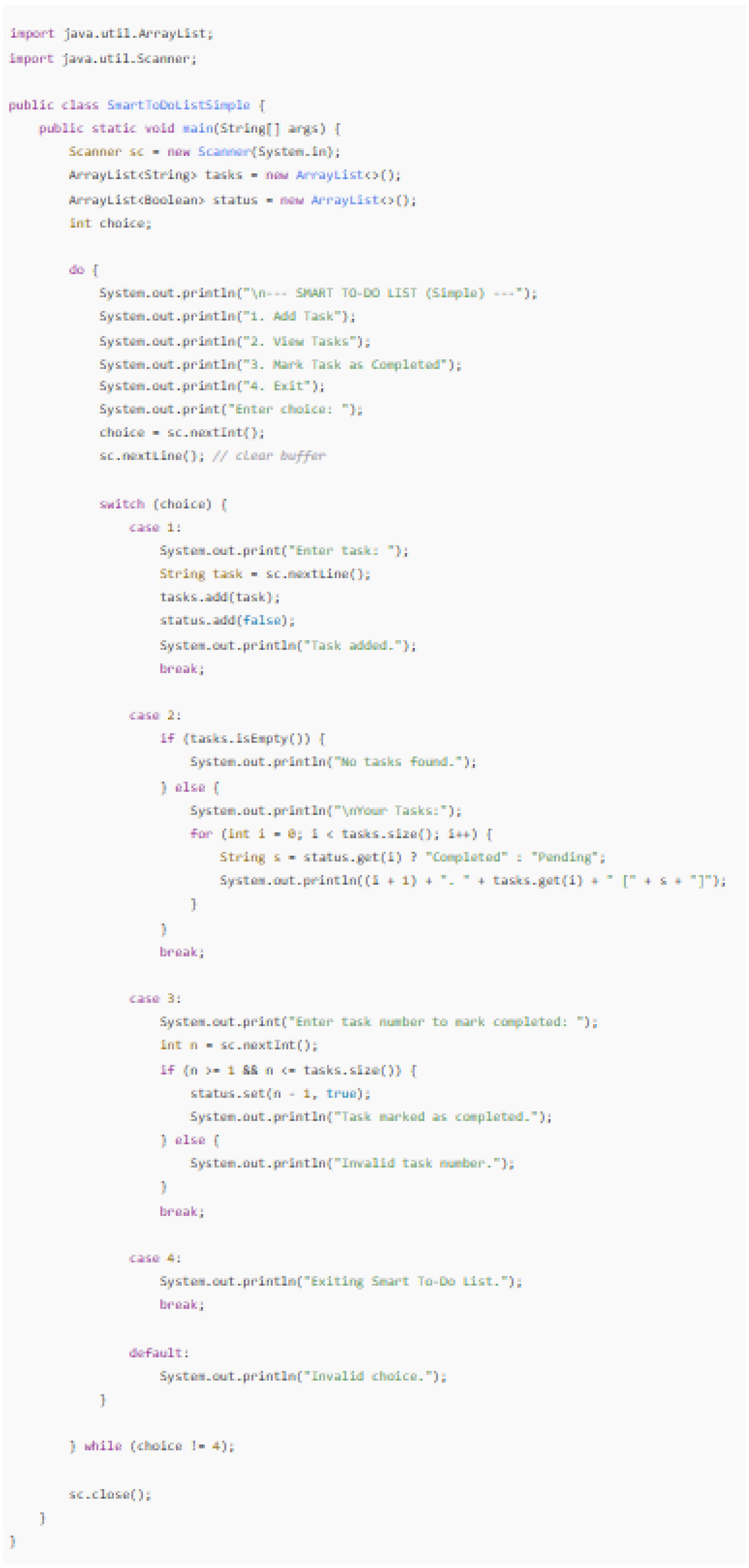
7. Methodology

1. The program displays a **menu** with options like:
    - Add Task
    - View Tasks
    - Mark Task as Completed
    - Delete Task
    - Exit
2. The user enters their **choice** from the keyboard.
3. According to the choice, corresponding **functions** are called (e.g., addTask(), viewTasks() etc.).
4. Tasks are stored in an **array/list** with details like **task ID, task name, and status (Pending/Completed)**.
5. A **loop** keeps showing the menu until the user selects **Exit**.
6. (If used) **File handling** is done to save and load tasks between program runs.

8. System Requirements

- **Programming Language:** Java
- **Platform:** Console / Command Prompt / Terminal
- **Concepts Used:**
    - Variables and Data Types
    - Loops (for, while)
    - Conditional Statements (if-else, switch-case)
    - Functions / Methods
    - Arrays / Lists / Structures / Classes
    - File Handling (optional)

9. Implementation



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

public class SmartToDoListSimple {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<String> tasks = new ArrayList<>();
        ArrayList<Boolean> status = new ArrayList<>();
        int choice;

        do {
            System.out.println("\n--- SMART TO-DO LIST (Simple) ---");
            System.out.println("1. Add Task");
            System.out.println("2. View Tasks");
            System.out.println("3. Mark Task as Completed");
            System.out.println("4. Exit");
            System.out.print("Enter choice: ");
            choice = sc.nextInt();
            sc.nextLine(); // clear buffer

            switch (choice) {
                case 1:
                    System.out.print("Enter task: ");
                    String task = sc.nextLine();
                    tasks.add(task);
                    status.add(false);
                    System.out.println("Task added.");
                    break;

                case 2:
                    if (tasks.isEmpty()) {
                        System.out.println("No tasks found.");
                    } else {
                        System.out.println("\nYour Tasks:");
                        for (int i = 0; i < tasks.size(); i++) {
                            String s = status.get(i) ? "Completed" : "Pending";
                            System.out.println((i + 1) + ". " + tasks.get(i) + " [" + s + "]");
                        }
                    }
                    break;

                case 3:
                    System.out.print("Enter task number to mark completed: ");
                    int n = sc.nextInt();
                    if (n >= 1 && n <= tasks.size()) {
                        status.set(n - 1, true);
                        System.out.println("Task marked as completed.");
                    } else {
                        System.out.println("Invalid task number.");
                    }
                    break;

                case 4:
                    System.out.println("Exiting Smart To-Do List.");
                    break;

                default:
                    System.out.println("Invalid choice.");
            }

        } while (choice != 4);

        sc.close();
    }
}
```

10. Results

- Add new tasks with unique identification.
- View the complete list of **pending and completed** tasks.
- Mark any task as **Completed**.
- Delete tasks that are no longer needed.
- Simple **menu-driven navigation**.
- Optional) Tasks remain saved using **file storage**.

11. Advantages

- Very **easy to use** and understand.
- Runs on **any basic system**—no installation or internet needed.
- Excellent mini project to **show programming logic skills**.
- Can be easily extended to more advanced versions (GUI / web).

12. Applications

- Personal daily task manager for students and professionals.
- Example mini project for **lab submissions and viva**.
- Teaching aid for **beginner programming concepts**.

13. Future Enhancements

- Add **login system** so each user can have their own tasks.
- Include **due dates and priority levels** for tasks.
- Convert the console app into a **GUI or web-based** To-Do List.
- Use a **database** instead of text files for storing tasks.
- Add **search and filter** options (e.g., show only pending or high-priority tasks).

14. Conclusion

The Smart To-Do List Console Application successfully demonstrates how a simple program can solve a real-life problem like task management. Using only basic programming constructs, it provides a practical, working solution and helps students strengthen their **logic building, problem-solving, and coding skills**. This project can also act as a base for future enhancements like GUI or web-based to-do list systems.

15. References

Official Java Documentation

- Oracle Java Documentation – Core Java Classes and APIs

Java Programming Tutorials

- *w3schools.com* – Java Basics and Control Statements
- *GeeksforGeeks.org* – Java OOPs, Collections, File Handling
- *TutorialsPoint.com* – Java Programming and Console Programs

Additional Learning

- YouTube – Java Console Mini Project Tutorials
- Java books like *"Head First Java"* for core concepts