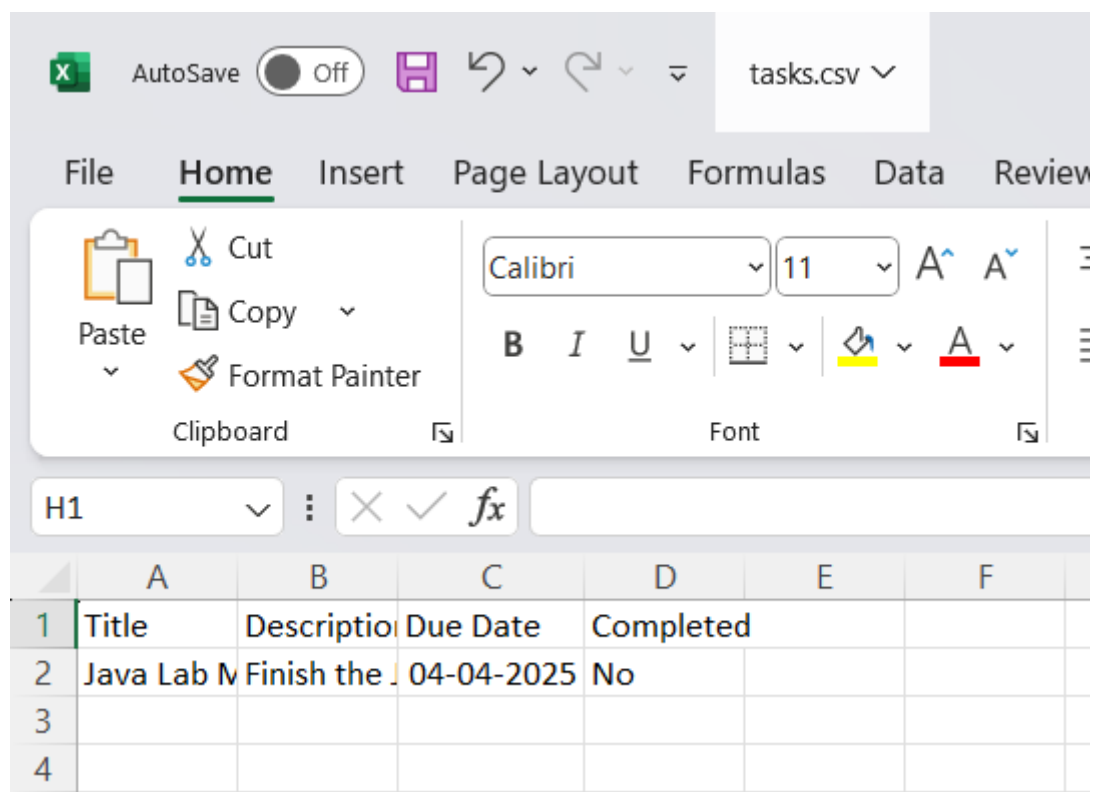


JAVA PROJECT – TASK MANAGER

Description:

The Task Manager is a simple console-based application built in Java for managing daily tasks efficiently. It provides users with an easy way to add, view, update, and delete tasks without requiring a database. Instead, all task data is stored in a file, ensuring persistence across multiple sessions. This lightweight application is ideal for individuals who need a quick and effective way to track their tasks without the complexity of database management.

CSV Output:



The screenshot shows the Microsoft Excel interface with a file named 'tasks.csv' open. The 'Home' tab is selected in the ribbon. The 'Clipboard' group shows 'Paste', 'Cut', 'Copy', and 'Format Painter' options. The 'Font' group shows 'Calibri' font, size '11', and various formatting options like bold, italic, underline, and text color. The formula bar shows 'H1'. The spreadsheet contains a table with 6 columns (A-F) and 4 rows (1-4). The data is as follows:

	A	B	C	D	E	F
1	Title	Description	Due Date	Completed		
2	Java Lab M	Finish the	04-04-2025	No		
3						
4						

Code:

```
import java.io.*;
import java.util.*;

class Task {
    String title;
    String description;
    String dueDate;
    boolean isCompleted;

    public Task(String title, String description, String dueDate, boolean
isCompleted) {
        this.title = title;
        this.description = description;
        this.dueDate = dueDate;
        this.isCompleted = isCompleted;
    }

    @Override
    public String toString() {
        return (isCompleted ? "[✓] " : "[ ] ") + title + " (Due: " +
dueDate + ") - " + description;
    }
}

public class TaskManager {
    private static final String FILE_NAME = "tasks.txt";
    private static final List<Task> tasks = new ArrayList<>();

    public static void main(String[] args) {
```

```
loadTasks();

Scanner scanner = new Scanner(System.in);

while (true) {
    System.out.println("\nTask Manager");
    System.out.println("1. Add Task");
    System.out.println("2. View Tasks");
    System.out.println("3. Mark Task as Completed");
    System.out.println("4. Delete Task");
    System.out.println("5. Save Tasks to CSV");
    System.out.println("6. Exit");
    System.out.print("Choose an option: ");

    int choice = scanner.nextInt();
    scanner.nextLine();

    switch (choice) {
        case 1 -> addTask(scanner);
        case 2 -> viewTasks();
        case 3 -> markTaskCompleted(scanner);
        case 4 -> deleteTask(scanner);
        case 5 -> saveTasksToCSV();
        case 6 -> {
            saveTasks();
            System.out.println("Exiting...");
            return;
        }
        default -> System.out.println("Invalid option!");
    }
}
}
```

```
private static void addTask(Scanner scanner) {
    System.out.print("Enter task title: ");
    String title = scanner.nextLine();
    System.out.print("Enter task description: ");
    String description = scanner.nextLine();
    System.out.print("Enter due date (YYYY-MM-DD): ");
    String dueDate = scanner.nextLine();

    tasks.add(new Task(title, description, dueDate, false));
    System.out.println("Task added successfully!");
}

private static void viewTasks() {
    if (tasks.isEmpty()) {
        System.out.println("No tasks available.");
        return;
    }
    for (int i = 0; i < tasks.size(); i++) {
        System.out.println((i + 1) + ". " + tasks.get(i));
    }
}

private static void markTaskCompleted(Scanner scanner) {
    viewTasks();
    if (tasks.isEmpty()) return;

    System.out.print("Enter task number to mark as completed: ");
    int index = scanner.nextInt() - 1;
    if (index >= 0 && index < tasks.size()) {
        tasks.get(index).isCompleted = true;
    }
}
```

```
        System.out.println("Task marked as completed!");
    } else {
        System.out.println("Invalid task number.");
    }
}

private static void deleteTask(Scanner scanner) {
    viewTasks();
    if (tasks.isEmpty()) return;

    System.out.print("Enter task number to delete: ");
    int index = scanner.nextInt() - 1;
    if (index >= 0 && index < tasks.size()) {
        tasks.remove(index);
        System.out.println("Task deleted!");
    } else {
        System.out.println("Invalid task number.");
    }
}

private static void saveTasksToCSV() {
    try (FileWriter writer = new FileWriter("tasks.csv")) {
        writer.write("Title,Description,Due Date,Completed\n");
        for (Task task : tasks) {
            writer.write(task.title + "," + task.description + "," +
task.dueDate + "," + (task.isCompleted ? "Yes" : "No") + "\n");
        }
        System.out.println("Tasks saved to tasks.csv");
    } catch (IOException e) {
        System.out.println("Error saving tasks: " + e.getMessage());
    }
}
```

```
    }

    private static void saveTasks() {
        try (ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(FILE_NAME))) {
            out.writeObject(tasks);
        } catch (IOException e) {
            System.out.println("Error saving tasks: " + e.getMessage());
        }
    }

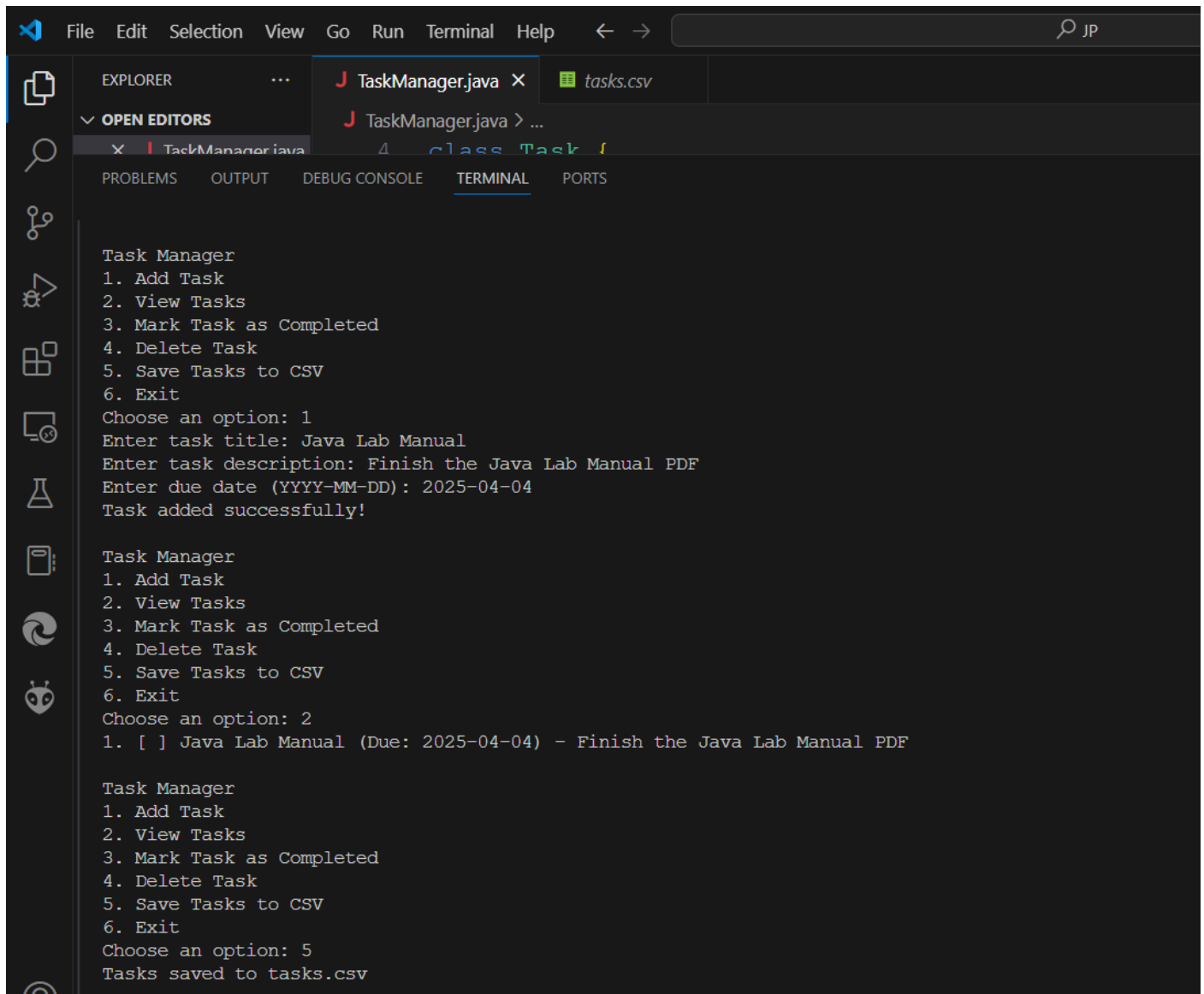
    @SuppressWarnings("unchecked")
    private static void loadTasks() {
        try (ObjectInputStream in = new ObjectInputStream(new
FileInputStream(FILE_NAME))) {
            Object obj = in.readObject();
            if (obj instanceof List<?>) {
                tasks.addAll((List<Task>) obj);
            }
        } catch (IOException | ClassNotFoundException e) {
            // Ignore if file doesn't exist yet
        }
    }
}
```

GITHUB: [GITHUB-SARVAN-2187](https://github.com/SarvanKumar2187)

FUTURE IMPROVEMENTS:

- Add priority levels to tasks
- Implement sorting/filtering options
- Enhance file encryption for security

TERMINAL OUTPUT:



```
Task Manager
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Delete Task
5. Save Tasks to CSV
6. Exit
Choose an option: 1
Enter task title: Java Lab Manual
Enter task description: Finish the Java Lab Manual PDF
Enter due date (YYYY-MM-DD): 2025-04-04
Task added successfully!

Task Manager
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Delete Task
5. Save Tasks to CSV
6. Exit
Choose an option: 2
1. [ ] Java Lab Manual (Due: 2025-04-04) - Finish the Java Lab Manual PDF

Task Manager
1. Add Task
2. View Tasks
3. Mark Task as Completed
4. Delete Task
5. Save Tasks to CSV
6. Exit
Choose an option: 5
Tasks saved to tasks.csv
```