**Internship Project Summary: Weeks 1-3**

# Week 1: Banking System

## Purpose

The goal of this project was to develop a secure and functional **Banking System** that allows users to create accounts, deposit, withdraw, and transfer money while storing account details using file handling.

## Key Code Highlights

// File Handling for Saving Account Data BufferedWriter writer = new BufferedWriter(new FileWriter("accounts.txt", true));

writer.write(accountNumber + "," + name + "," + pin + "," + balance); writer.newLine();

writer.close();

## Explanation

1. Users can **create an account** with a unique account number and a PIN.
2. They can **deposit, withdraw, and check their balance** through a menu-driven interface.
3. The **transfer feature** enables sending money between accounts.
4. **File handling** ensures persistence of account data.

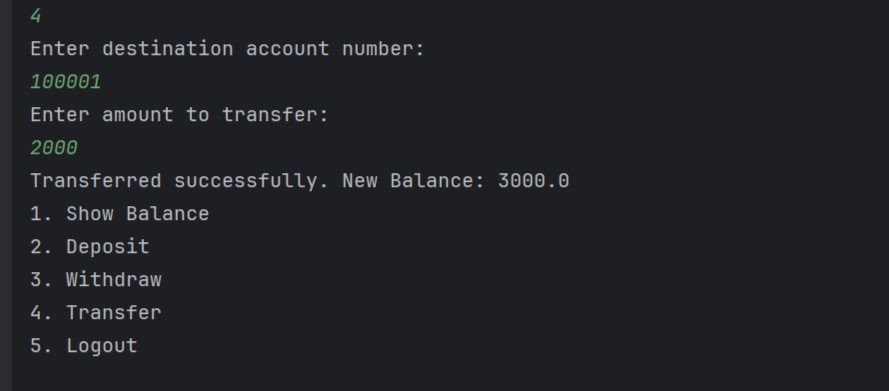
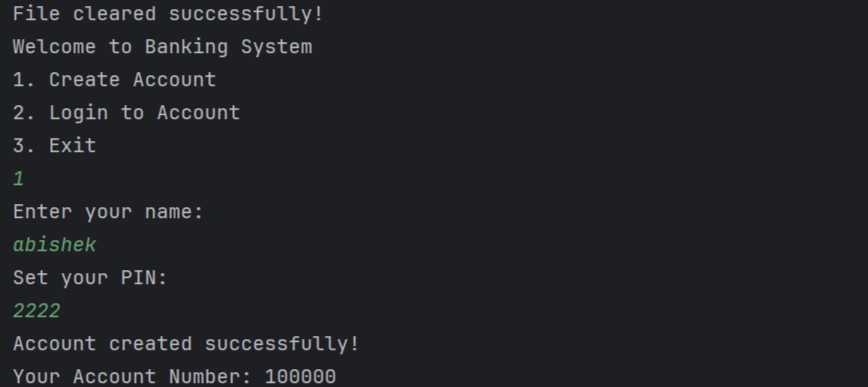
## Tools s Technologies Used

### Programming Language: Java

* + **IDE:** IntelliJ IDEA
  + **Technologies:** File Handling, Object-Oriented Programming (OOP)
  + **Version Control:** GitHub

## Output Screenshots

*(Includes images of the console interface, account creation, and transaction processing)*

**

# Week 2: Student Management System

## Purpose

This project aimed to create a **Student Management System** that allows adding, viewing, updating, and deleting student details while enabling sorting based on marks.

## Key Code Highlights

// Sorting students by marks students.sort(Comparator.comparingInt(Student::getMarks).reversed());

## Explanation

1. Users can **add student details** (ID, name, marks).
2. The system allows **editing and deleting student records**.
3. **Sorting functionality** helps arrange students based on marks.
4. Data is stored using **HashMap** for efficient retrieval.

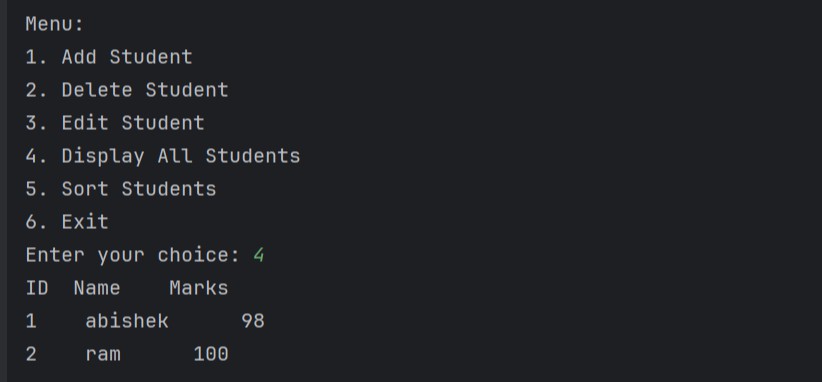
## Tools s Technologies Used

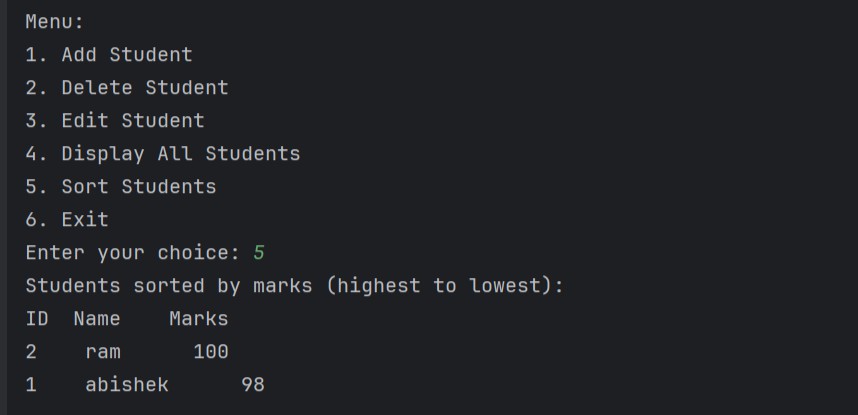
### Programming Language: Java

* + **IDE:** IntelliJ IDEA
  + **Data Structures:** ArrayList, HashMap
  + **Version Control:** GitHub

## Output Screenshots

*(Includes images of the console showing student records and sorting by marks)*

**



# Week 3: To-Do List Application

## Purpose

This project involved building a **To-Do List Application** that allows users to manage tasks efficiently by adding, editing, deleting, prioritizing, and saving/loading tasks using file handling.

## Key Code Highlights

// Saving tasks to a file

BufferedWriter writer = new BufferedWriter(new FileWriter("tasks.txt"));

for (Task task : tasks) {

writer.write(task.getTaskName() + "," + task.isDone() + "," + task.getTaskPriority());

writer.newLine();

}

writer.close();

## Explanation

### Users can add, edit, delete, and mark tasks as complete.

1. **Task priority (High, Medium, Low)** helps in organizing tasks.
2. **File handling ensures data persistence**, allowing users to save and reload tasks.
3. A **menu-driven interface** simplifies user interaction.

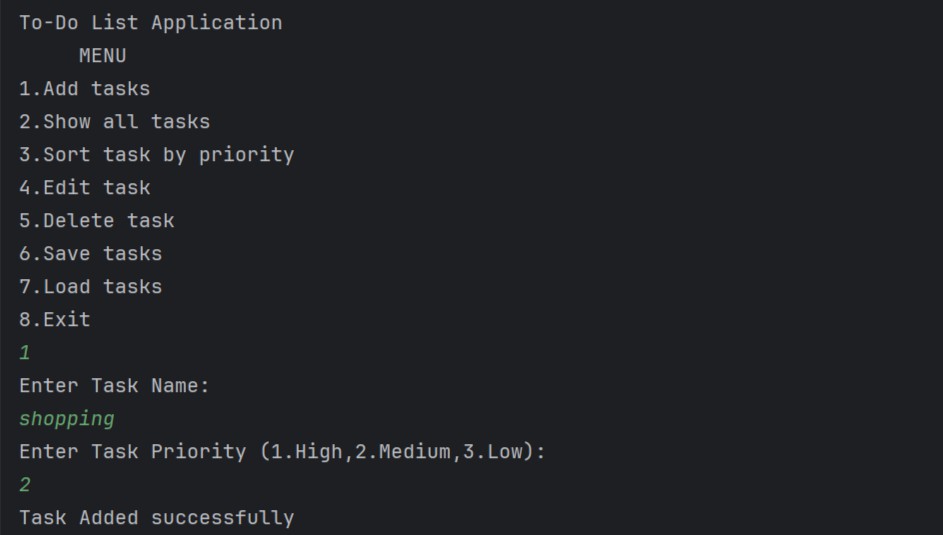
## Tools s Technologies Used

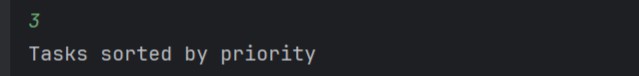
### Programming Language: Java

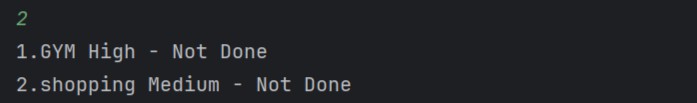
* + **IDE:** IntelliJ IDEA
  + **Technologies:** File Handling, Object-Oriented Programming (OOP)
  + **Data Structures:** ArrayList
  + **Version Control:** GitHub

## Output Screenshots

*(Includes images of the console displaying task lists, completed tasks, and file persistence results)*

**



**

# Conclusion

Over the past three weeks, these projects helped me strengthen my Java skills, gain experience with **file handling, data structures, and OOP principles**, and apply **real-world problem-solving techniques**. I also improved my ability to manage projects using GitHub for version control.

.s7'¸•˙ Looking forward to more learning and challenges ahead!