

Student Management System

**Bachelor of Computer Applications  
Semester - II  
Lab - Object Oriented Programming in Java**

**Project Title:  
Student Management System**

Submitted By:  
1. Prem Panchal – Reg no: 2411021240011

2.Yashwant Chandawat – Reg no: 2411021240005

3.Aryan Sovit – Reg no: 2411021240074

Submitted To:  
Faculty Name: ***Veera Raghav Reddy Kathula***

**---------------- -------------------**

***Faculty Sign Hod Sign***

Department of Computer Application  
Alliance University  
Chandapura-Anekal Main Road, Anekal  
Bengaluru – 562 106  
April 2025

# Acknowledgement

I take this opportunity to sincerely thank and express my deep sense of gratitude to my mentor ***Veera Raghav Reddy Kathula*** for their invaluable support, technical guidance, and continuous encouragement throughout the duration of this project.  
  
I would also like to extend heartfelt thanks to the Department of Computer Applications at Alliance University, for promoting a learning environment focused on hands-on and project-based education.  
  
I am thankful to my family and peers for their support and motivation during this journey.  
  
Additionally, I acknowledge the use of open-source documentation, online learning platforms, and virtual programming assistants, which supported the technical planning, system design, and clarity in developing this project.  
  
This project has been a valuable experience in blending academic learning with practical implementation.

Contents

[Chapter 1: Introduction to Java 3](#_Toc196154158)

[Chapter 2: Object-Oriented Programming in Java 3](#_Toc196154159)

[Chapter 3: Problem Statement & Objective 4](#_Toc196154160)

[Chapter 4: Tools and Technologies Used 4](#_Toc196154161)

[Chapter 5: Project Overview 4](#_Toc196154162)

[Chapter 6: System Requirements 5](#_Toc196154163)

[Chapter 7: System Design 5](#_Toc196154164)

[Chapter 8: Project Implementation 6](#_Toc196154165)

[Chapter 10: Screenshots of Project 7](#_Toc196154166)

[Chapter 11: Challenges Faced 9](#_Toc196154167)

[Chapter 12: Future Scope 9](#_Toc196154168)

[Chapter 13: Conclusion 9](#_Toc196154169)

[Chapter 14: References 9](#_Toc196154170)

# Chapter 1: Introduction to Java

Java is a platform-independent, object-oriented programming language widely used for software development. With its principle of "Write Once, Run Anywhere," Java programs are executed via the Java Virtual Machine (JVM). Java provides robust features like automatic memory management, exception handling, multi-threading, and a rich set of APIs.  
  
Java Editions:  
- Java SE: Used in this project for core functionality.  
- Java EE: For enterprise applications.  
- Java ME: For embedded and mobile devices.

# Chapter 2: Object-Oriented Programming in Java

Java is inherently object-oriented. OOP concepts used in this project include:  
  
- Encapsulation: Student data is managed through a class with private variables and public getters/setters.  
- Abstraction: The user interacts with forms, abstracting server logic.  
- Polymorphism & Inheritance: While not heavily used, the structure allows future scalability.

# Chapter 3: Problem Statement & Objective

Problem Statement:  
Colleges require a system to manage student records — including their name, roll number, and course — along with functionality to add, list, remove, and export data to PDF.  
  
Objective:  
To develop a Java-based web application using embedded HTTP server that allows:  
- Student login authentication  
- Add, remove, and list students  
- Export the student list to PDF  
- HTML/CSS UI to interact with the backend

# Chapter 4: Tools and Technologies Used

- Language: Java (JDK 17)  
- Server: com.sun.net.httpserver.HttpServer  
- PDF Generation: iText library  
- Frontend: HTML5, CSS3, JavaScript  
- IDE: Visual Studio Code  
- OS: Windows 10

# Chapter 5: Project Overview

The Student Management System is a Java-based application that offers core features like:  
- Login authentication  
- Add student (name, roll, course)  
- List all students  
- Remove student by roll number  
- Download a PDF report of all students  
  
Frontend pages:  
- login.html  
- home.html  
- add.html  
- remove.html  
- list.html  
  
Java Files:  
- StudentManagementServer.java  
- Student.java

# Chapter 6: System Requirements

Software:  
- JDK 17 or higher  
- VS Code with Java extensions  
- Web browser (Chrome/Edge)  
  
Hardware:  
- Processor: Intel i3+  
- RAM: 4GB+  
- Disk Space: 100MB+

# Chapter 7: System Design

Design Layers:  
- UI: HTML forms  
- Server logic: Java (HttpHandlers)  
- Data model: Java Map<String, String> for student info  
  
Folder Structure:  
- web/  
 |- login.html  
 |- home.html  
 |- add.html  
 |- remove.html  
 |- list.html  
- StudentManagementServer.java  
- Student.java  
- styles.css  
- login.css  
- images/screenshots/

# Chapter 8: Project Implementation

Java Code Snippet (Add Student):

static class AddStudentHandler implements HttpHandler {  
 @Override  
 public void handle(HttpExchange exchange) throws IOException {  
 if ("POST".equals(exchange.getRequestMethod())) {  
 ... // read form  
 students.add(student);  
 }  
 }  
}

Frontend Sample (add.html):

<form method="POST" action="/add-student">  
 <input type="text" name="name" />  
 <input type="text" name="roll" />  
 <input type="text" name="course" />  
 <button type="submit">Add Student</button>  
</form>

PDF Generation Sample:

Document document = new Document();  
PdfWriter.getInstance(document, new FileOutputStream("students.pdf"));  
document.open();  
document.add(new Paragraph("Student List"));

Chapter 9: Sample Snippets

public class Student {  
 private String name;  
 private String age;  
 private String grade;  
  
 public Student(String name, String age, String grade) {  
 this.name = name;  
 this.age = age;  
 this.grade = grade;  
 }  
}

# Chapter 10: Screenshots of Project

Fig 1 – Login Page

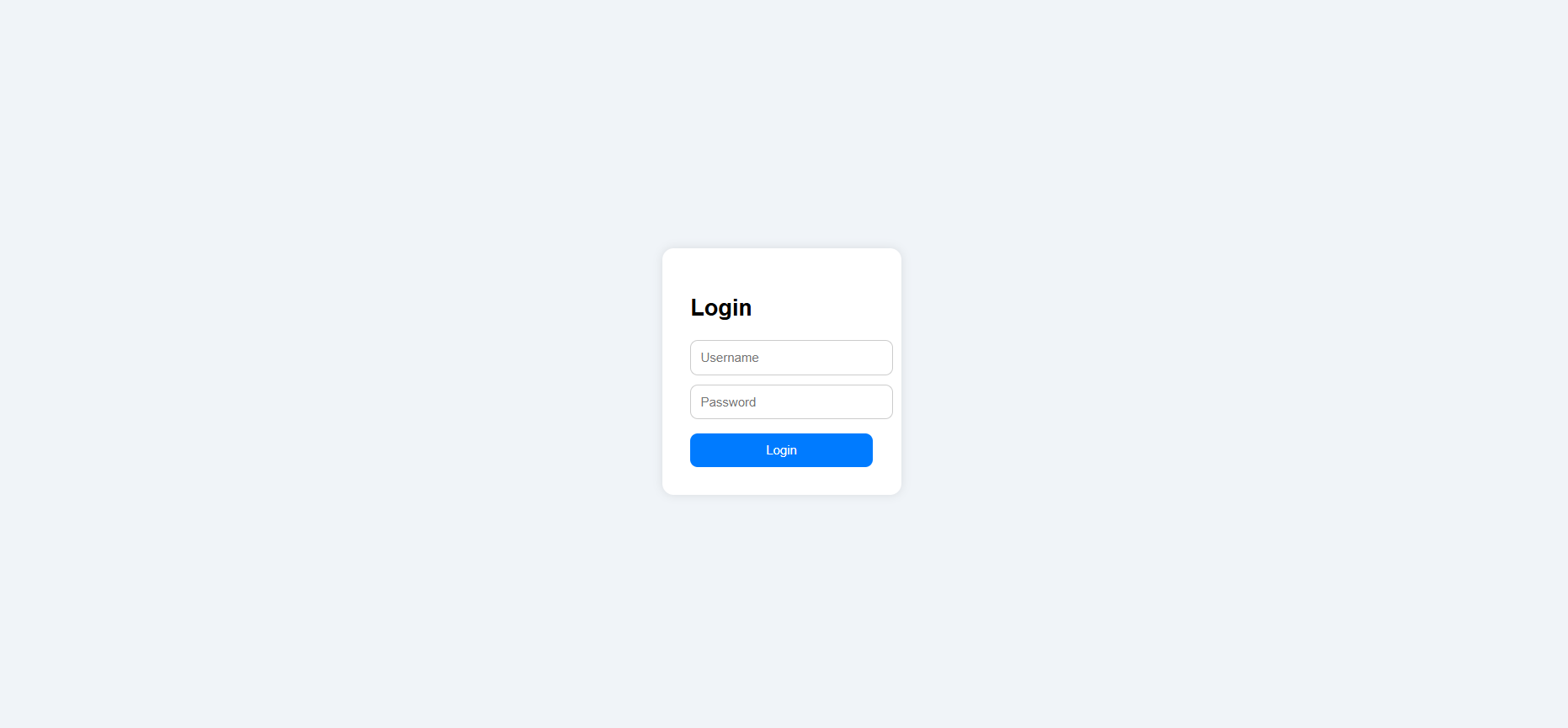


Fig 2 – Home Dashboard

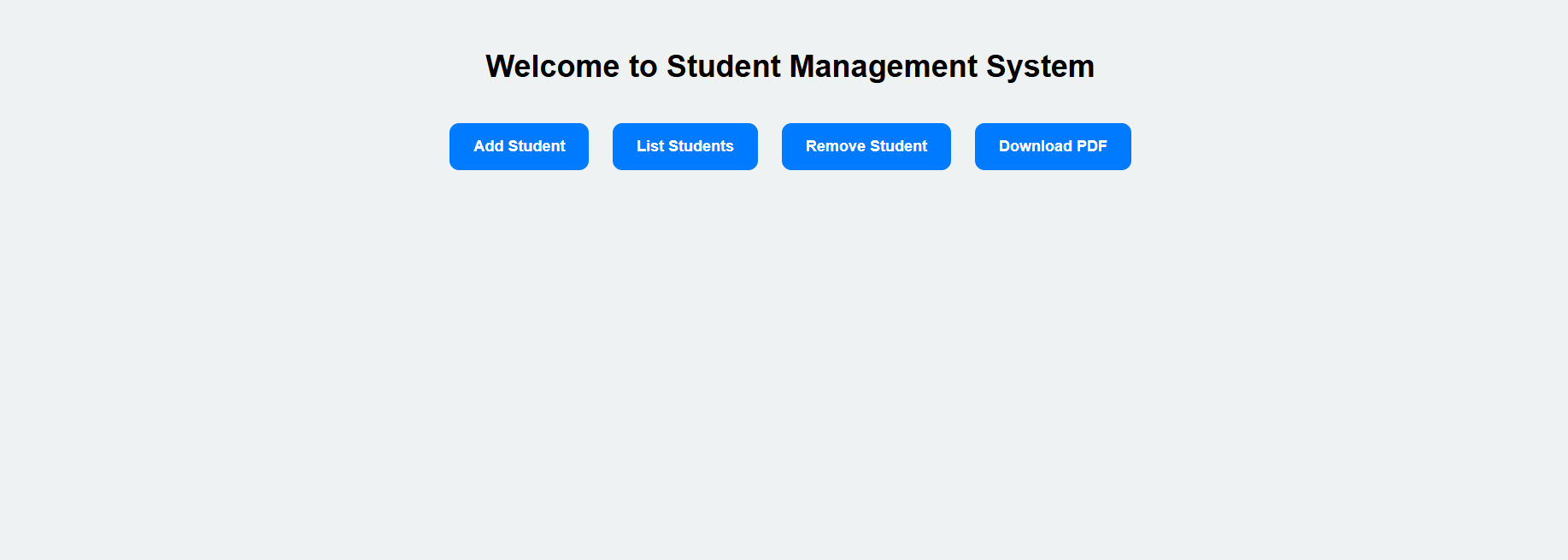


Fig 3 – Add Student Page

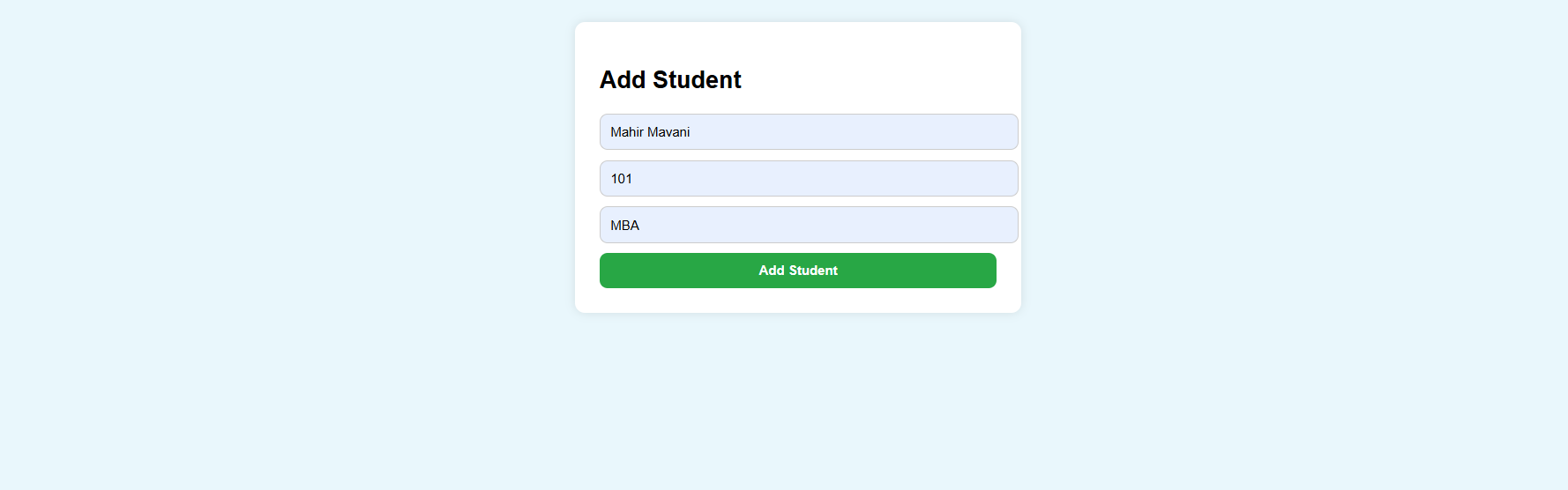


Fig 4 – Student List

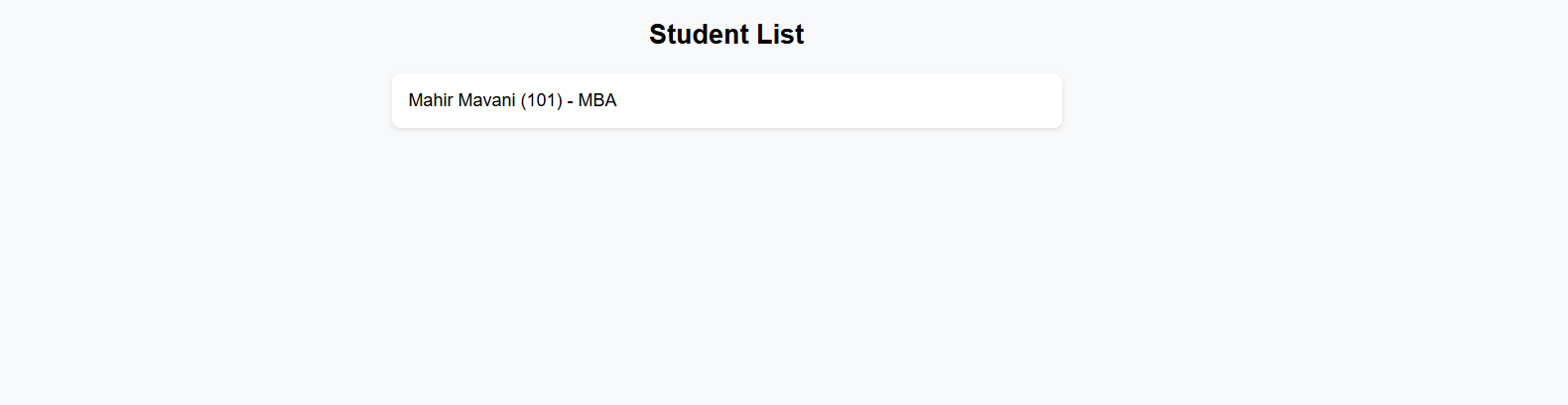


Fig 5 – Remove Student

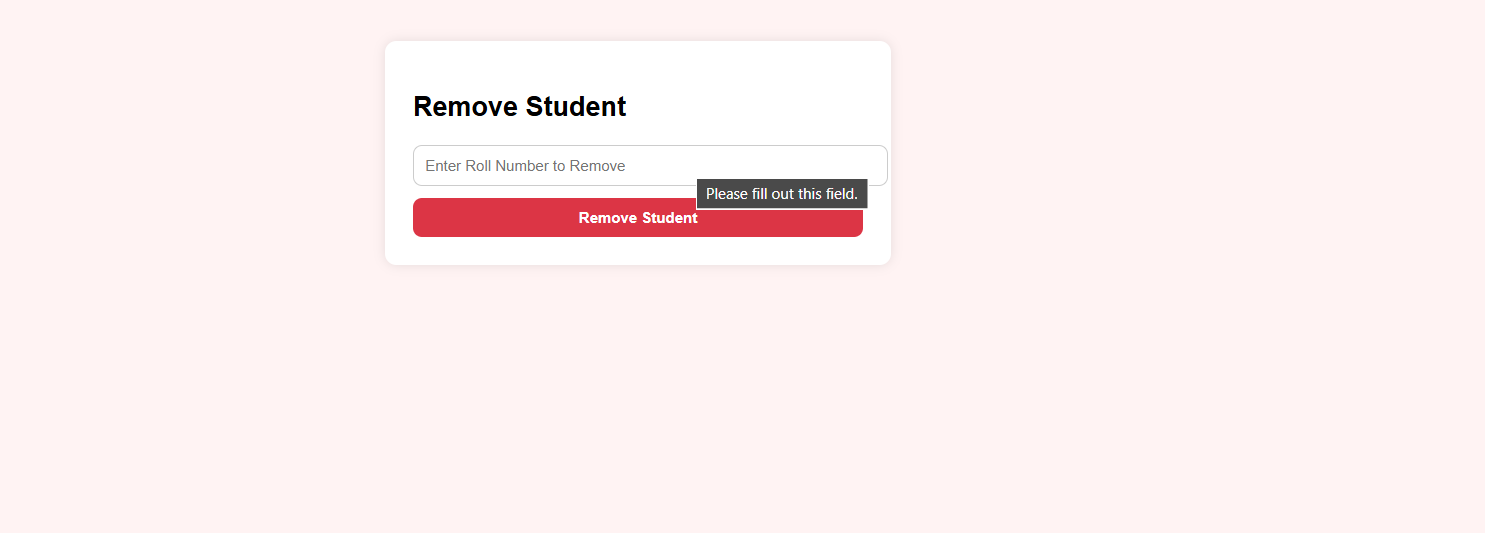
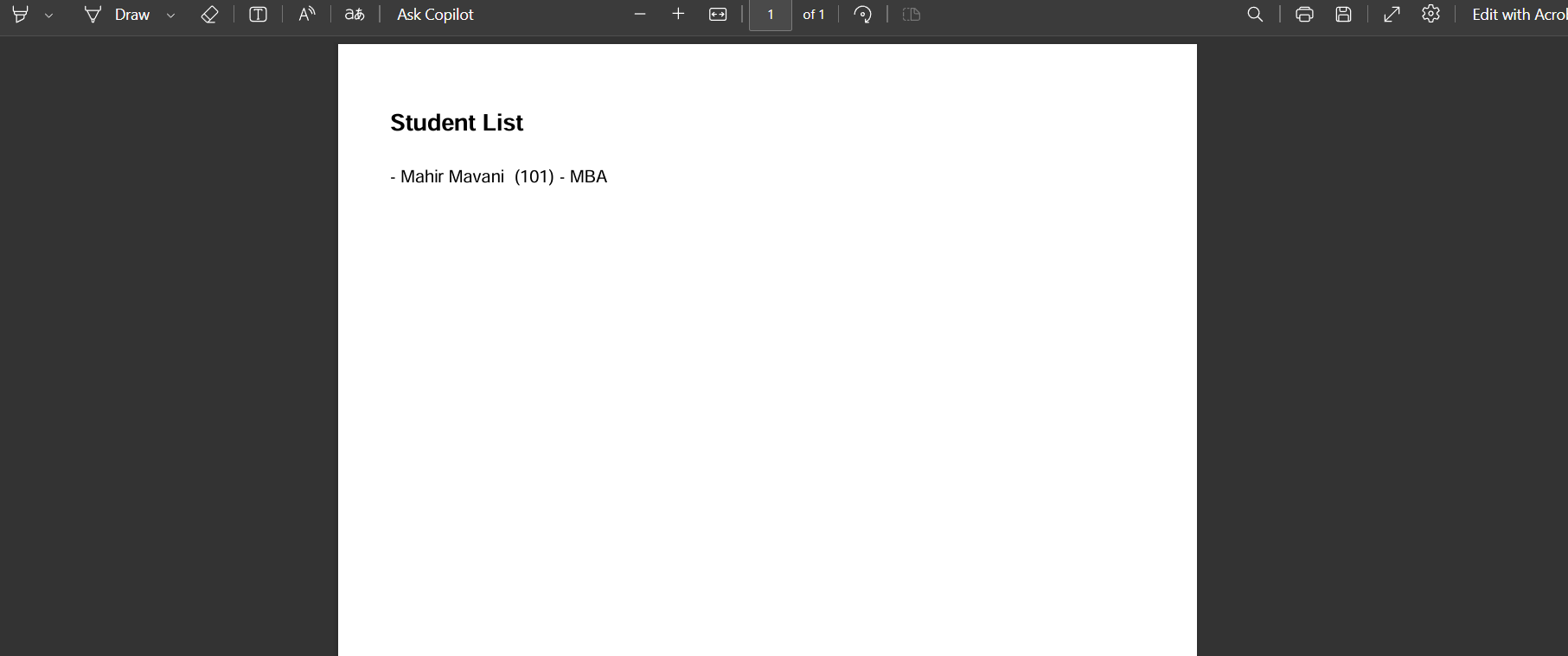


Fig 6 – Generated PDF



# Chapter 11: Challenges Faced

- Managing list and map data across multiple requests  
- iText naming conflict with java.util.List  
- HTML and server integration without a framework  
- PDF formatting with ListItem

# Chapter 12: Future Scope

- Connect to real database (MySQL)  
- Add login sessions  
- Export to CSV, Excel  
- Use JavaFX for a GUI  
- Deploy online using Spring Boot

# Chapter 13: Conclusion

This project helped understand:  
- Java networking and embedded HTTP servers  
- File handling and PDF generation  
- Modular programming and clean code  
- HTML/Java integration for practical apps  
  
It simulated a real-world backend with a minimal tech stack and strong OOP practices.

# Chapter 14: References

- Oracle Java Docs – https://docs.oracle.com/javase  
- iText PDF Docs – https://itextpdf.com  
- W3Schools – https://www.w3schools.com  
- GeeksforGeeks – https://www.geeksforgeeks.org  
- Stack Overflow  
- Intellije Ideas

GitHub: [JAVA-PROJECT/ at main · Prem2309/JAVA-PROJECT](https://github.com/Prem2309/JAVA-PROJECT/tree/main)