

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AN EXPERIENTIAL LEARNING REPORT ON THE TOPIC :**

**FEE MANAGEMENT SYSTEM**

**UNDER THE SUPERVISION OF:**

**MAMTARANI DAS**

**ASSISTANT PROFESSOR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SUBMITTED BY:**

**GROUP: 10**

**SEMESTER: 4th**

**BRANCH: COMPUTER SCINENCE ENGINEERING**

|  |  |
| --- | --- |
| **NAME** | **REGD\_NO** |
| **ROUNAK RAJ** | **2201020925** |
| **VEDANAND** | **2201020848** |
| **ANISH KUMAR SHARMA** | **2201020902** |
| **LAL ANKIT NATH SHAHDEO** | **2201020938** |
| **ROHIT KUMAR** | **2201020333** |
| **ABHISHEK KUMAR** | **2201020320** |
| **MAYANK RAJ GAUTAM** | **2201020678** |
| **ROUNAK KUMAR** | **2201020334** |
| **AMARTYA ABHILASH** | **2201020939** |

**ACKNOWLEDGEMENT:**

We, the students of group 10 of the esteemed CV Raman Global University , Bhubaneshwar, branch of Computer Science Engineering , 4th Semester, do hereby acknowledge our greatfulness towards the people associated in the project of experiential learning.

First of all , We would like to pay our thanks to my respected associated Professor, Mamtarani Das Mam for giving us this opportunity. She helped us with her knowledge and sufficient information without which we would not have been able to complete the experiential learning project. Secondly , We are greatful for the library facility to gather information needed for the project. Above all , the grace of God of all creations led me to complete the project successfully.

**CONTENT:**

1. **Abstract**
2. **Introduction**
3. **Problem Statement**
4. **Algorithm**
5. **Concepts used in the Code**
6. **Model and Screenshots of Coding**
7. **Results**
8. **Conclusion**
9. **References**

**ABSTRACT:**

The “Fees Management System” is a desktop application with specialized majorities in the field of Fees management. Allow to admin to includes the information of Courses and Fees details. Allow customers to check, modify the information, and print data, fand verify the information.

It is a management system aiming to give the safe storing of Fees details and courses. The most useful and reliable functions which can not be found in another system. The project is built in Java platform with the help of IDE NetBeans and JDBC Derby database.

In order to make the best product possible, insight in languages like. Java, SQL for database, and another one is inevitable.

However, all of us final year students with limited knowledge and to be adept in all those languages is never an easy deal. Since then, although we do try our best, there are still some small malfunctions in our product.

ABSTRACT

FEES MANAGEMENT SYSTEM

iv

The <Fees Management System= is a desktop application with specialized

majorities in the field of Fees management. Allow to admin to includes the

information of Courses and Fees details. Allow customers to check, modify the

information, and print data, fand verify the information.



**INTRODUCTION:**

The main objective of the java project on Student Fees Management System is to manage the details of

Students. It manages all the information about Student fee Details Student, Course, Subject, Semester,

Fees. It manages all the information about Student, Payment, Fees, Student. The project is totally built at

administrative end and thus only the administrator is guaranteed the access. The purpose of the project is

to build an application program to reduce the manual work for managing the Student, Course, Payment,

Subject. It tracks all the details about the Subject, Semester, Fees.

The main objective of the java project on Student Fees Management System is to manage the details of

Students. It manages all the information about Student fee Details Student, Course, Subject, Semester,

Fees. It manages all the information about Student, Payment, Fees, Student. The project is totally built at

administrative end and thus only the administrator is guaranteed the access. The purpose of the project is

to build an application program to reduce the manual work for managing the Student, Course, Payment,

Subject. It tracks all the details about the Subject, Semester, Fees.

The main objective of the java project on Student Fees Management System is to manage the details of

Students. It manages all the information about Student fee Details Student, Course, Subject, Semester,

Fees. It manages all the information about Student, Payment, Fees, Student. The project is totally built at

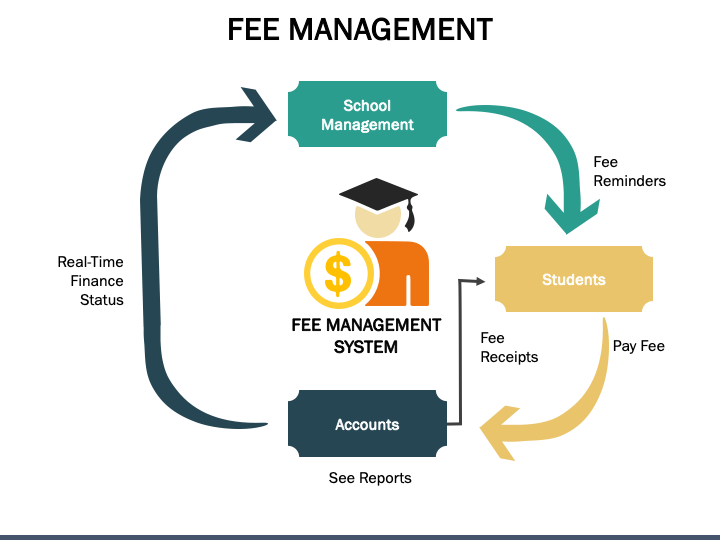
administrative end and thus only the administrator is guaranteed the access. The purpose of the project is

to build an application program to reduce the manual work for managing the Student, Course, Payment,

Subject. It tracks all the details about the Subject, Semester, Fees.

The main objective of the java project on Student Fees Management System is to manage the details of Students. It manages all the information about Student fee Details Student, Course, Subject, Semester, Fees. It manages all the information about Student, Payment, Fees, Student. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Student, Course, Payment, Subject. It tracks all the details about the Subject, Semester, Fees. The aim of the Project Is Managing collection of school fees, issuing fee receipts and fee register updation is done with the help of software resulting in highly accurate data. Software provides facility to print receipts automizing office work . This Software provides facility to generate due fees report easily and at any point of time.

A fee management system is used as task management software that automates collecting fees and generating fee receipts. It eliminates duplicate data entries and minimizes errors when entering entries into school accounts. The system supports all sizes of public and private institutes.



**PROBLEM STATEMENT**

The ABC Institute offers various academic programs and courses to students. The institute needs to develop a Fee Management System to streamline the process of fee collection, tracking, and reporting. The system should cater to the following requirements:

1. **Student Registration:** The system should allow new students to register and create their profiles. Each student profile should capture personal details (name, contact information, date of birth, etc.), program/course details, and other relevant information.

2. **Fee Structure Management:** The system should enable administrators to define and manage fee structures for different programs, courses, and academic years. This includes setting tuition fees, admission fees, hostel fees, and any other applicable charges.

3.**Fee Calculation and Generation of Fee Receipts:** Based on the student's enrolled program/course and the defined fee structure, the system should calculate the applicable fees for each student. It should generate fee receipts/invoices for students, including details such as the fee breakdown, due dates, and payment methods.

4. **Payment Management:** The system should facilitate various payment methods, such as online payments, cash payments, bank transfers, or installment options. It should record and update the payment status for each student's fee receipt.

5. **Fee Reminders and Notifications:** The system should send reminders and notifications to students regarding upcoming fee due dates, outstanding balances, and late payment penalties (if applicable).

6. **Fee Concessions and Scholarships:** The system should allow administrators to define and apply fee concessions, discounts, or scholarships based on predefined criteria (e.g., merit-based, need-based, or other eligibility rules).

7. **Reporting and Analytics:** The system should provide comprehensive reporting and analytics capabilities, including revenue reports, student fee status reports, outstanding fee reports, and other relevant reports for decision-making and financial planning purposes.

8. **User Management and Access Control:** The system should have different user roles and access levels, such as administrators, staff, and students. Each user role should have appropriate permissions and access rights to perform their respective tasks.

9. **Data Security and Backup:** The system should ensure data security and provide backup and restore mechanisms to protect sensitive student and financial data.

10. **Integration with Other Systems:** If required, the Fee Management System should have the capability to integrate with other systems within the institute, such as the Student Information System, Accounting System, or Learning Management System.

The Fee Management System should be user-friendly, efficient, and scalable to accommodate the growing needs of the institute. It should comply with relevant laws, regulations, and policies related to fee collection and financial management in educational institutions.

**ALGORITHM**

The provided code is a Java program for a Fee Management System. Here's the algorithm explaining the program's flow:

1**. Initialize the GUI components**

- Create labels, text fields, text areas, radio buttons, combo boxes, lists, and buttons for the user interface.

- Set the properties (bounds, font, color, etc.) of each component and add them to the frame.

**2. Handle user input**

- Add action listeners to the buttons for handling user interactions.

3. **Process user input**

- When the "Show" button is clicked:

- Get the selected values from the lists (courses and hostel facilities).

- Display the selected options in a label.

**4. Generate fee receipt**

- When the "Generate Receipt" button is clicked:

- Retrieve the entered data from the text fields and selected options from radio buttons and combo boxes.

- Construct the fee receipt text by combining the user information and selected options.

- Calculate and include the total fee amount based on the selected hostel facility.

- Display the fee receipt text in the text area.

- Save the fee receipt text to a file named "FEE\_Reciept.txt" in the specified directory.

- Show a success message dialog.

**5. Reset form**

- When the "Reset" button is clicked:

- Clear the text fields and text areas.

6. **Print receipt**

- When the "Print" button is clicked:

- Print the contents of the text area containing the fee receipt.

7. Close the program

- Add a window listener to handle the window closing event.

- When the window is closed, exit the program.

8. Set up the frame

- Set the frame's title, icon, size, layout, and visibility.

- Set the background color of the frame.

9. Main method

- Create an instance of the `fee` class (the main GUI class).

- Print the current working directory.

The algorithm covers the main functionalities of the Fee Management System, including displaying the user interface, handling user input, generating fee receipts, saving and printing receipts, and managing the program's lifecycle.

Note that the code also includes some file handling operations (saving the fee receipt to a file) and GUI-related tasks (setting up labels, text fields, buttons, etc.).

**CONCEPT USED**

The Java code for the Fee Management System incorporates several concepts and features. Here are the key concepts used in the code:

**1. Swing GUI:**  The program utilizes the Swing library for creating the graphical user interface (GUI). It makes use of various Swing components such as `JFrame`, `JLabel`, `JTextField`, `JTextArea`, `JRadioButton`, `JComboBox`, `JList`, `JButton`, and `JOptionPane`.

**2. Event Handling:** The code implements event handling using `ActionListener` and `WindowListener`. Action listeners are used to handle button clicks and other user interactions, while a window listener is used to handle the window closing event.

**3. Layout Management:** The code uses the `null` layout, where components are positioned explicitly using setBounds() method calls. While this approach is not recommended for complex UIs, it is used in this code for simplicity.

**4. Input Validation:** The code does not include explicit input validation for the text fields, which is a good practice for robust applications.

**5. File I/O:** The program uses `FileWriter` to write the generated fee receipt to a text file named "FEE\_Reciept.txt" in a specified directory.

**6. Exception Handling:** The code includes try-catch blocks to handle exceptions that may occur during file writing and printing operations.

**7. Printing:** The program utilizes `print()` method from the `JTextArea` class to print the contents of the text area containing the fee receipt.

8. **Collections and Data Structures:** The code makes use of the `DefaultListModel` and `JList` components to display lists of options (courses and hostel facilities). It also uses `ButtonGroup` to group together radio buttons for mutually exclusive selections.

9. **Conditional Statements:** The program employs extensive use of `if` statements and logical operators to handle different user selections and generate the appropriate fee receipt content.

10. **String Manipulation:** The code performs string concatenation and manipulation to construct the fee receipt text based on user input.

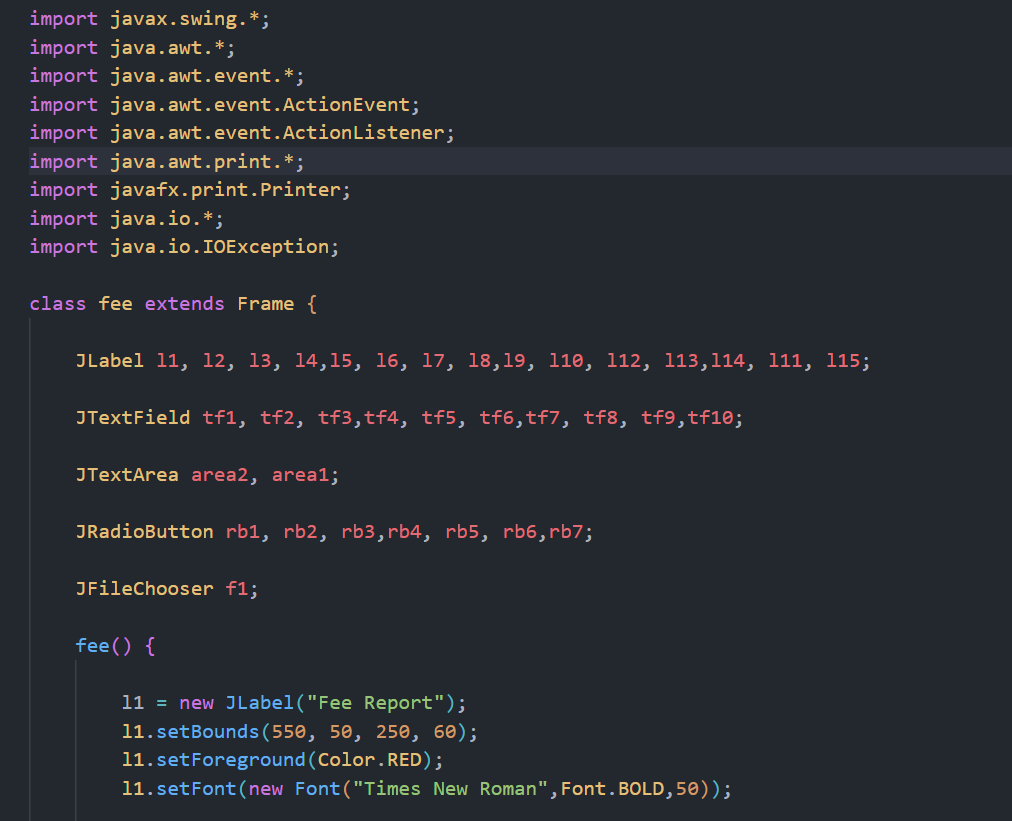
**11**. GUI Customization: The program customizes the appearance of GUI components by setting properties such as fonts, colors, borders, and icons.

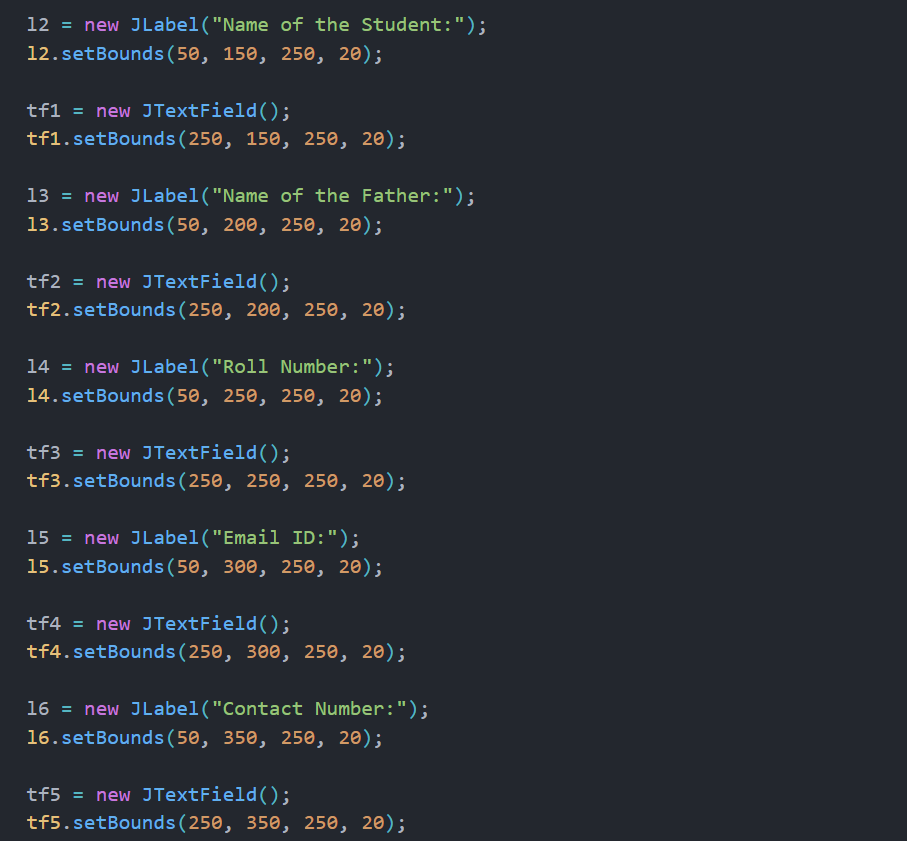
12. Object-Oriented Programming (OOP): The code follows an object-oriented approach by creating a custom class `fee` that extends the `JFrame` class and encapsulates the GUI components and program logic.

13. Main Method: The program has a `main` method, which is the entry point for the application and creates an instance of the `fee` class.

While the code demonstrates the usage of various Java concepts, it could benefit from improvements in areas such as code organization, naming conventions, and adherence to best practices. Additionally, incorporating input validation, better error handling, and modular design would enhance the robustness and maintainability of the application.

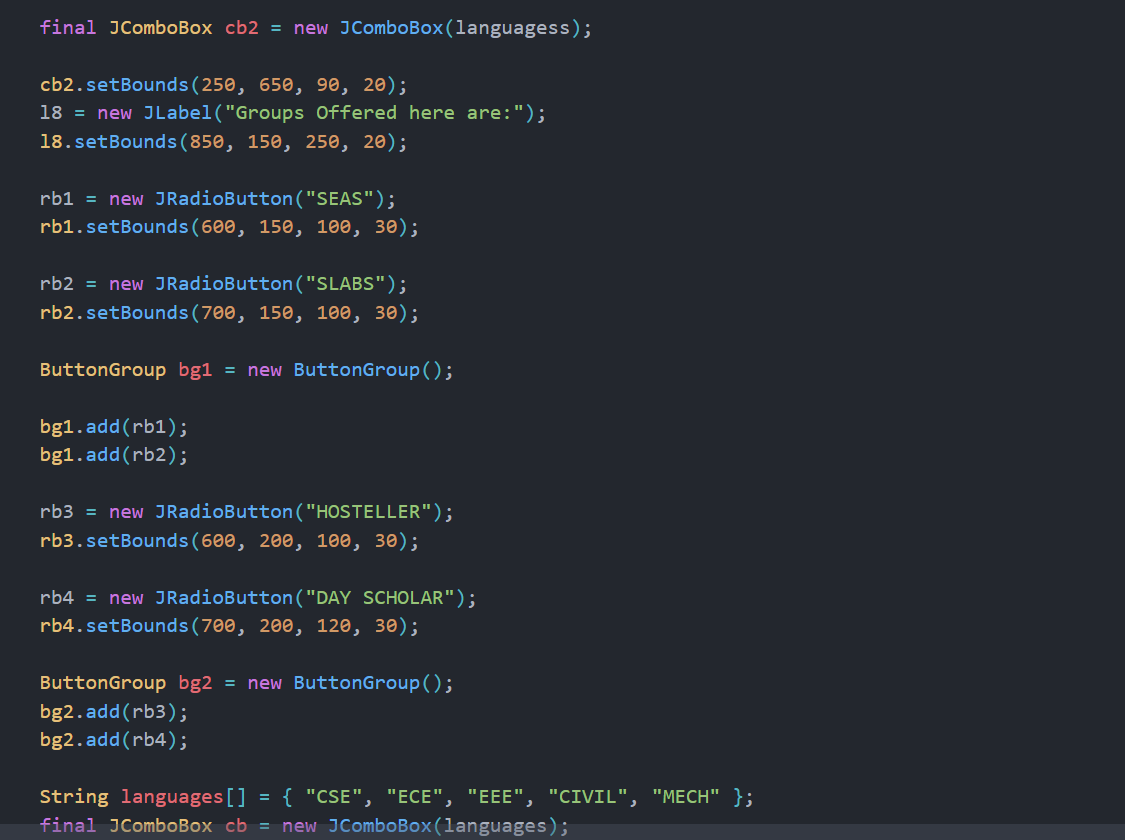
**MODEL AND SCREENSHORTS OF CODING**

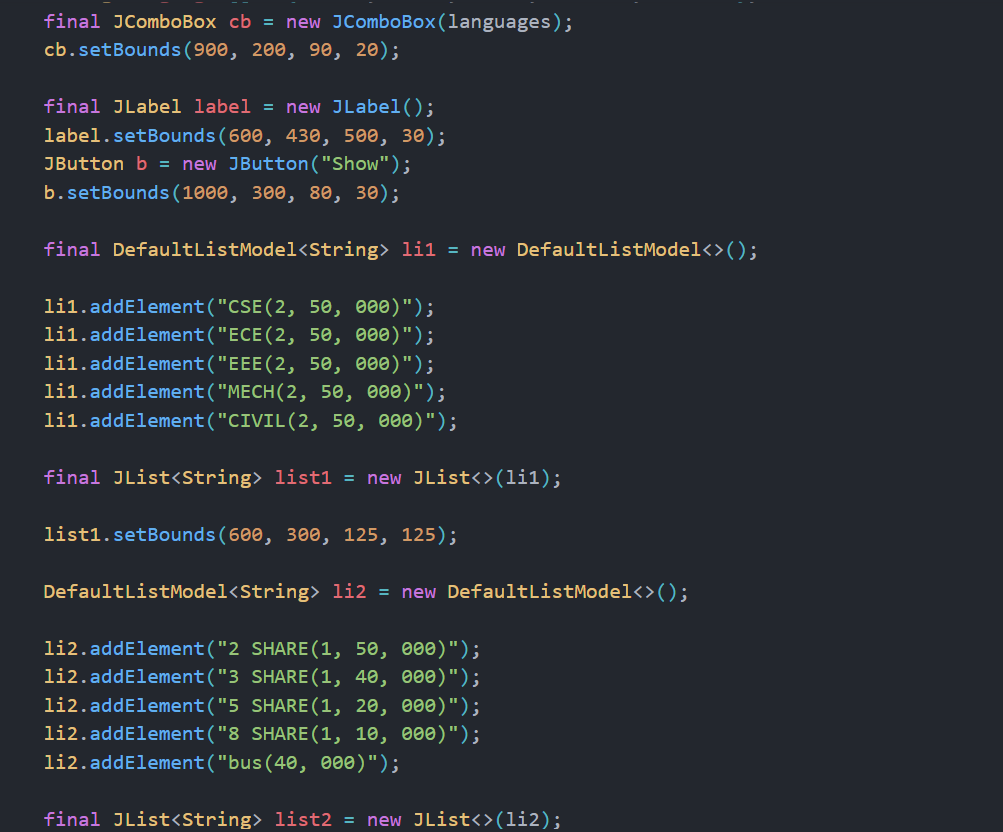


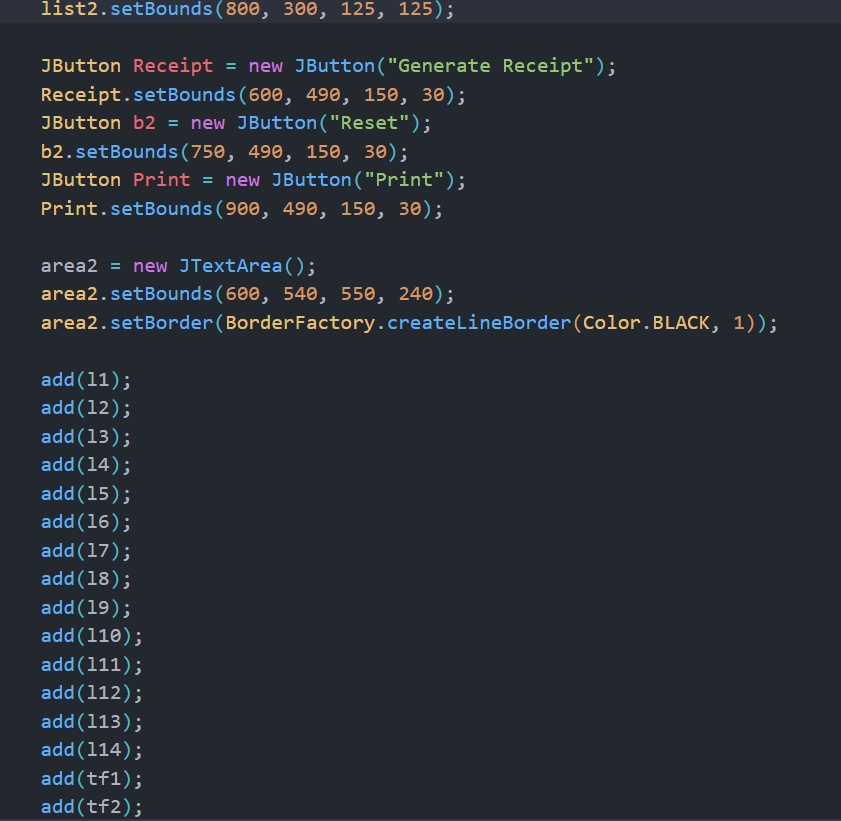




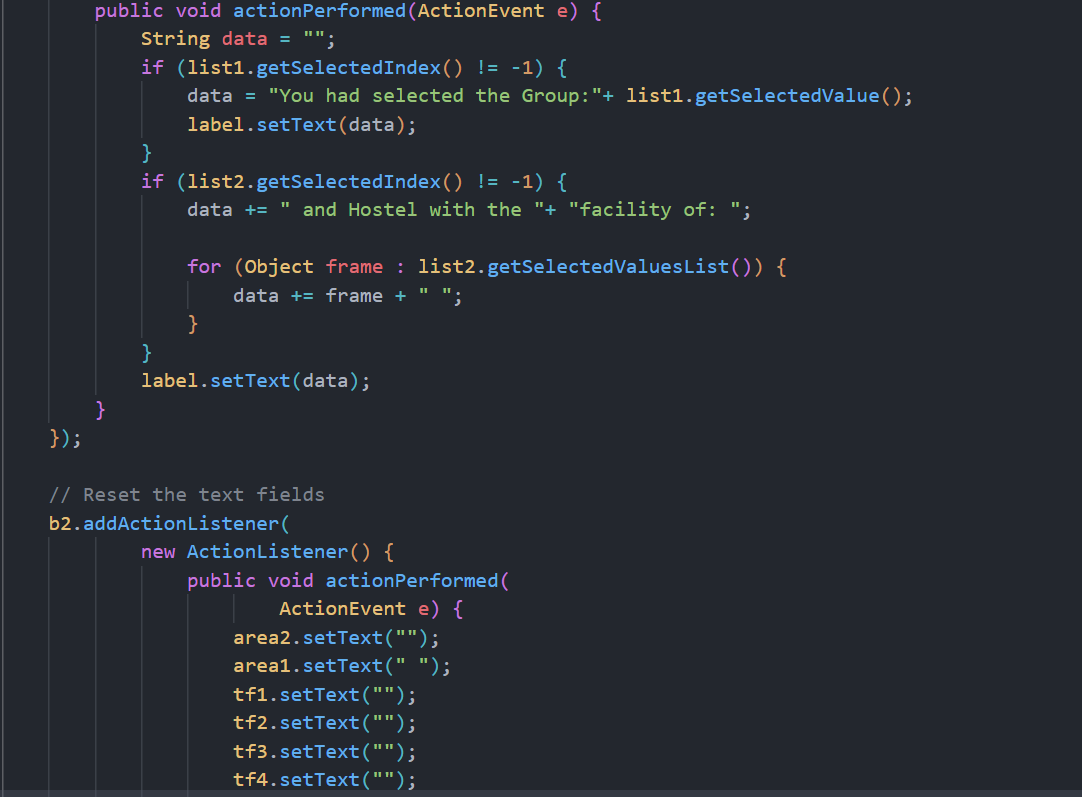


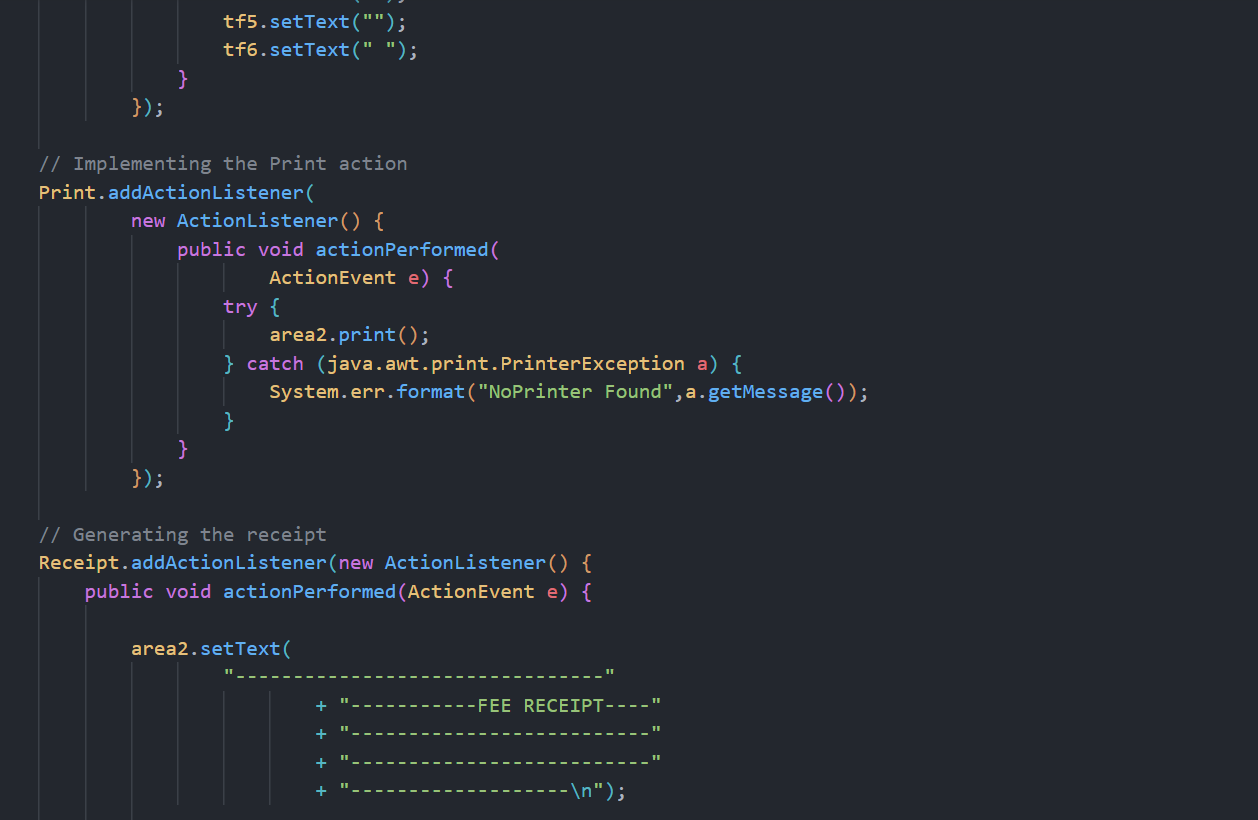


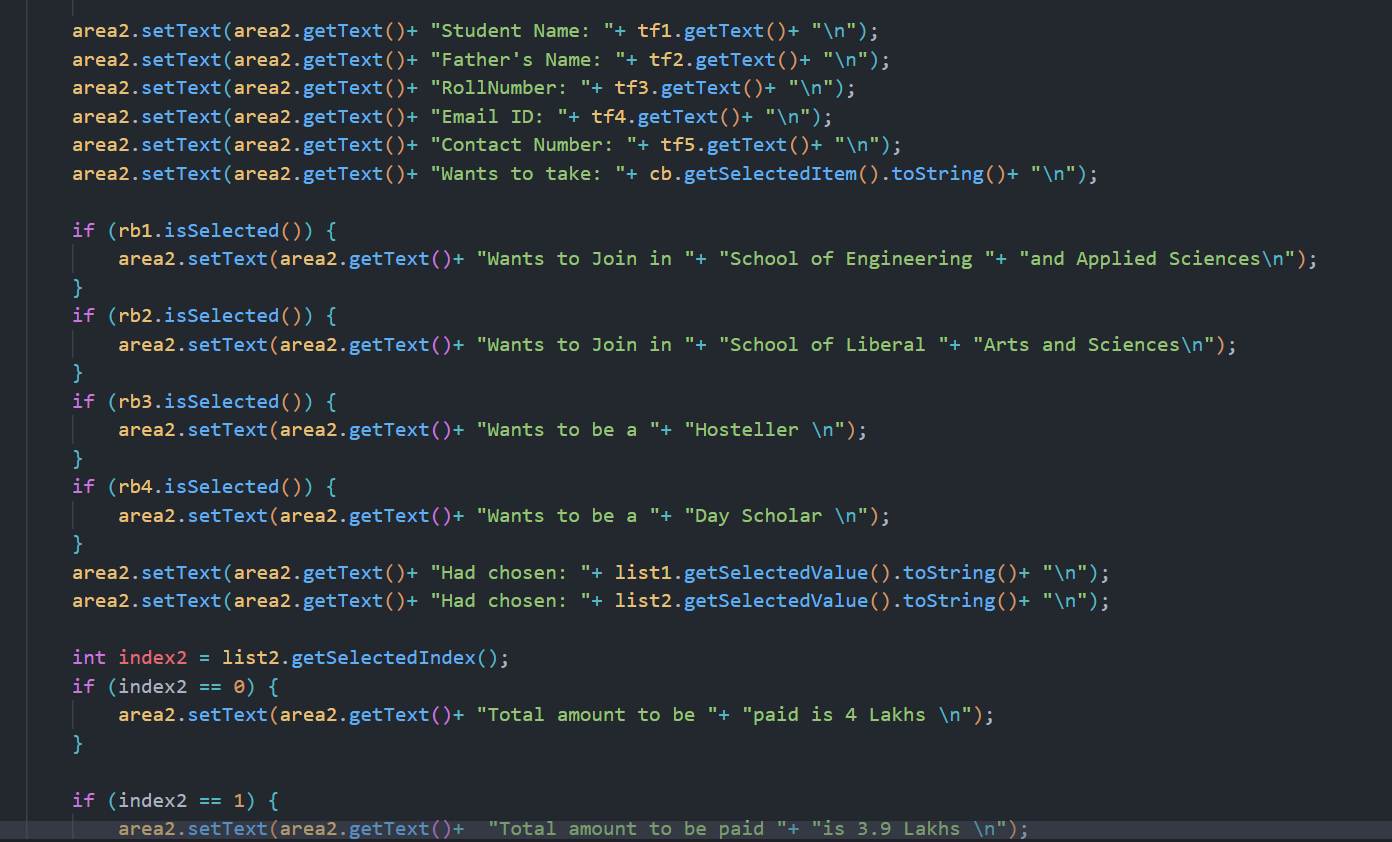
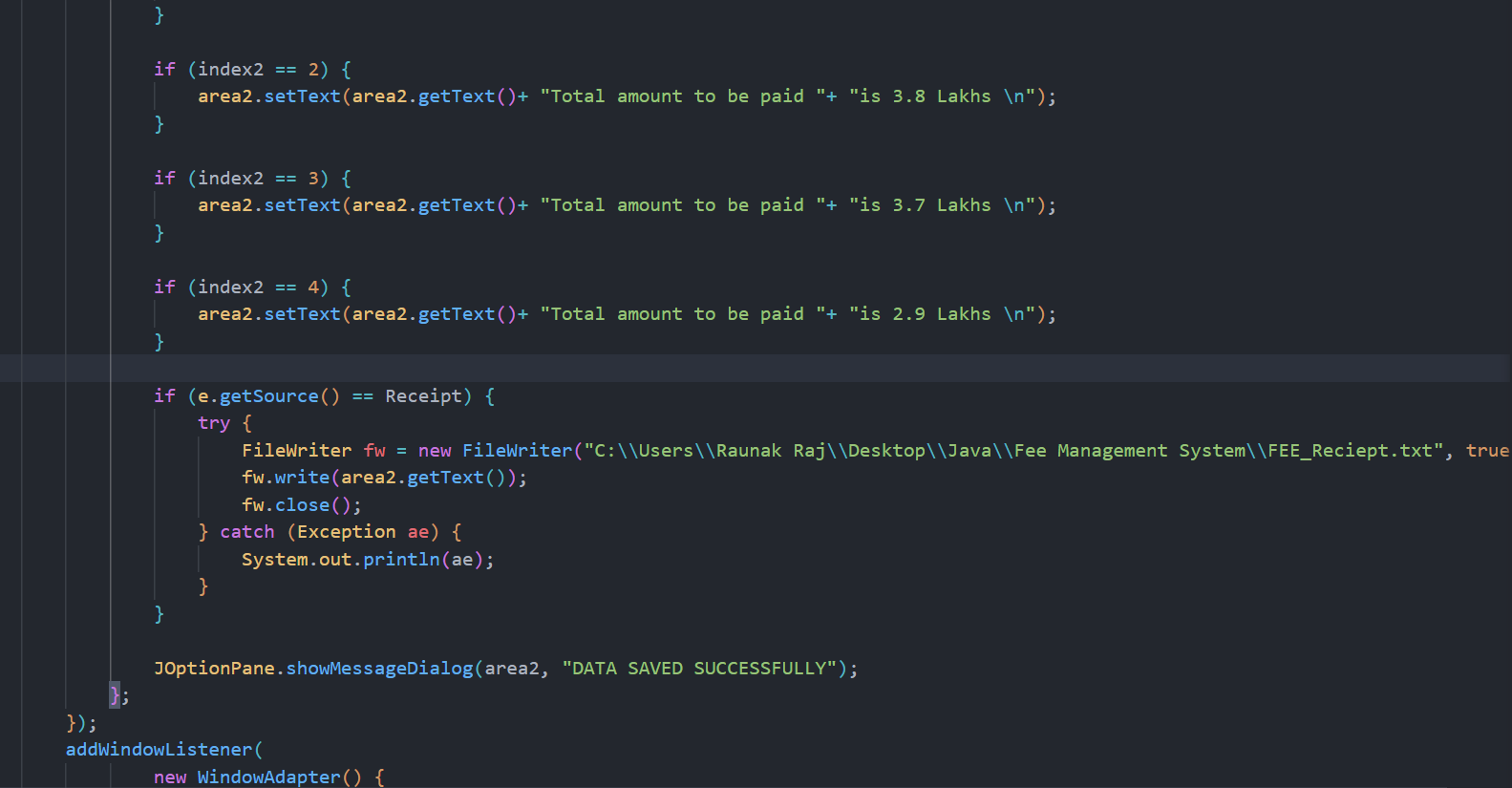
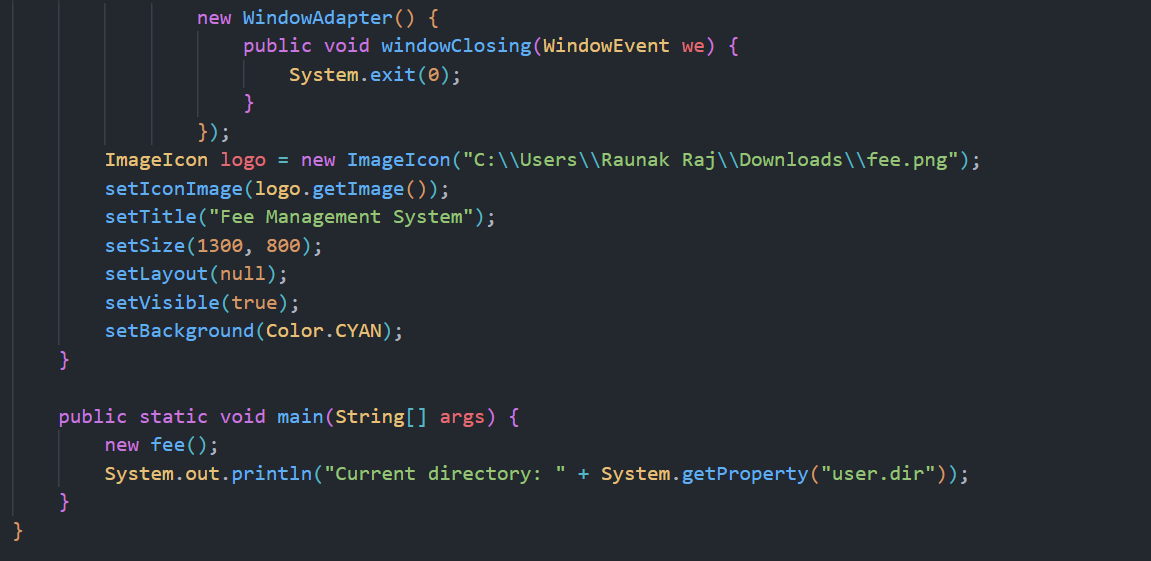


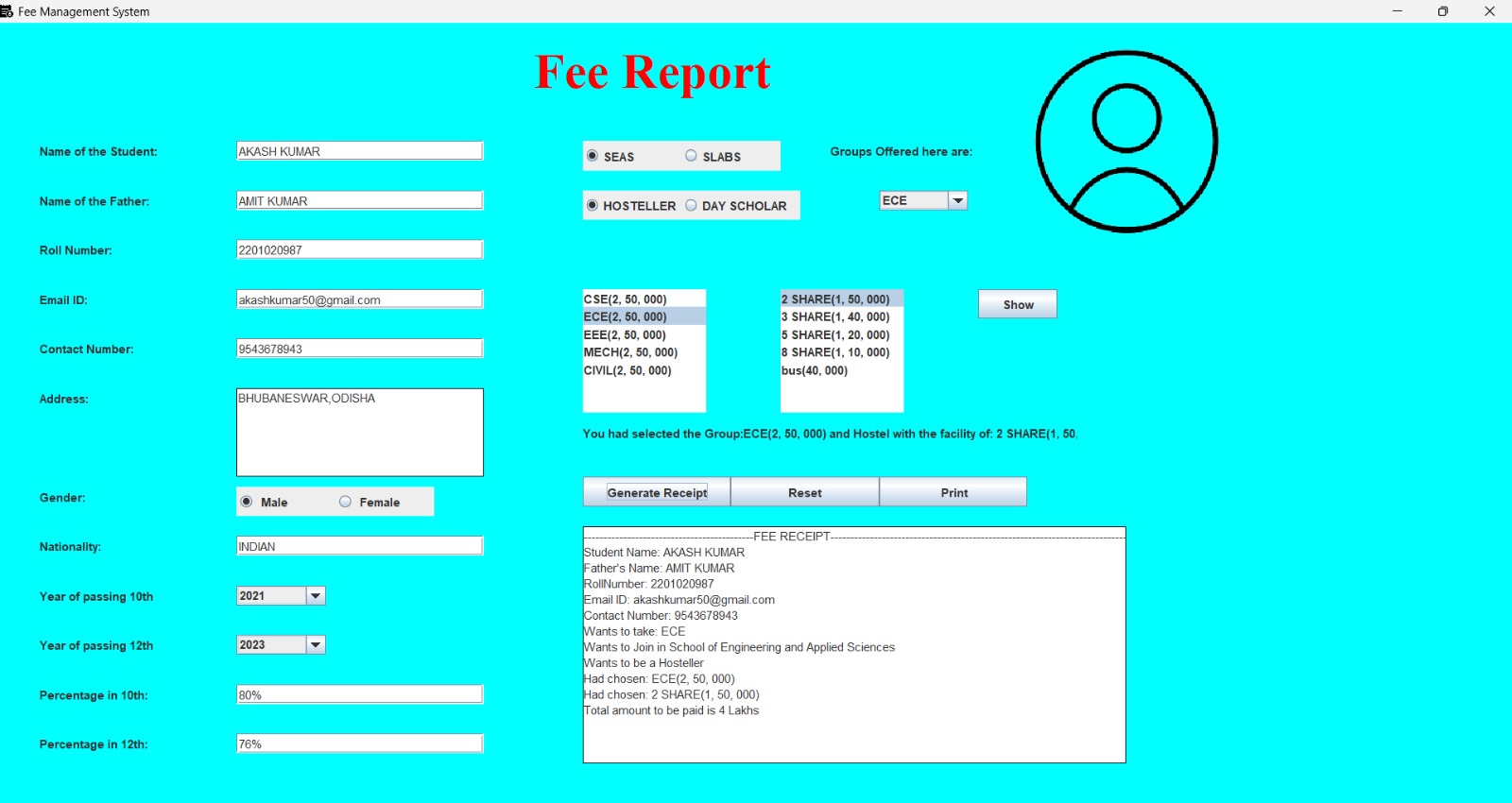


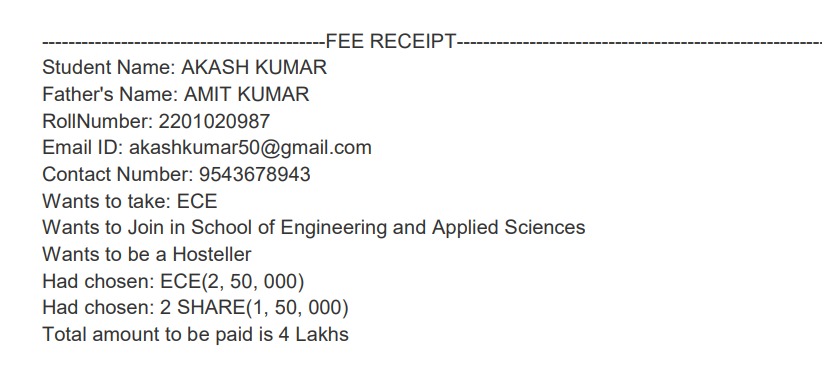
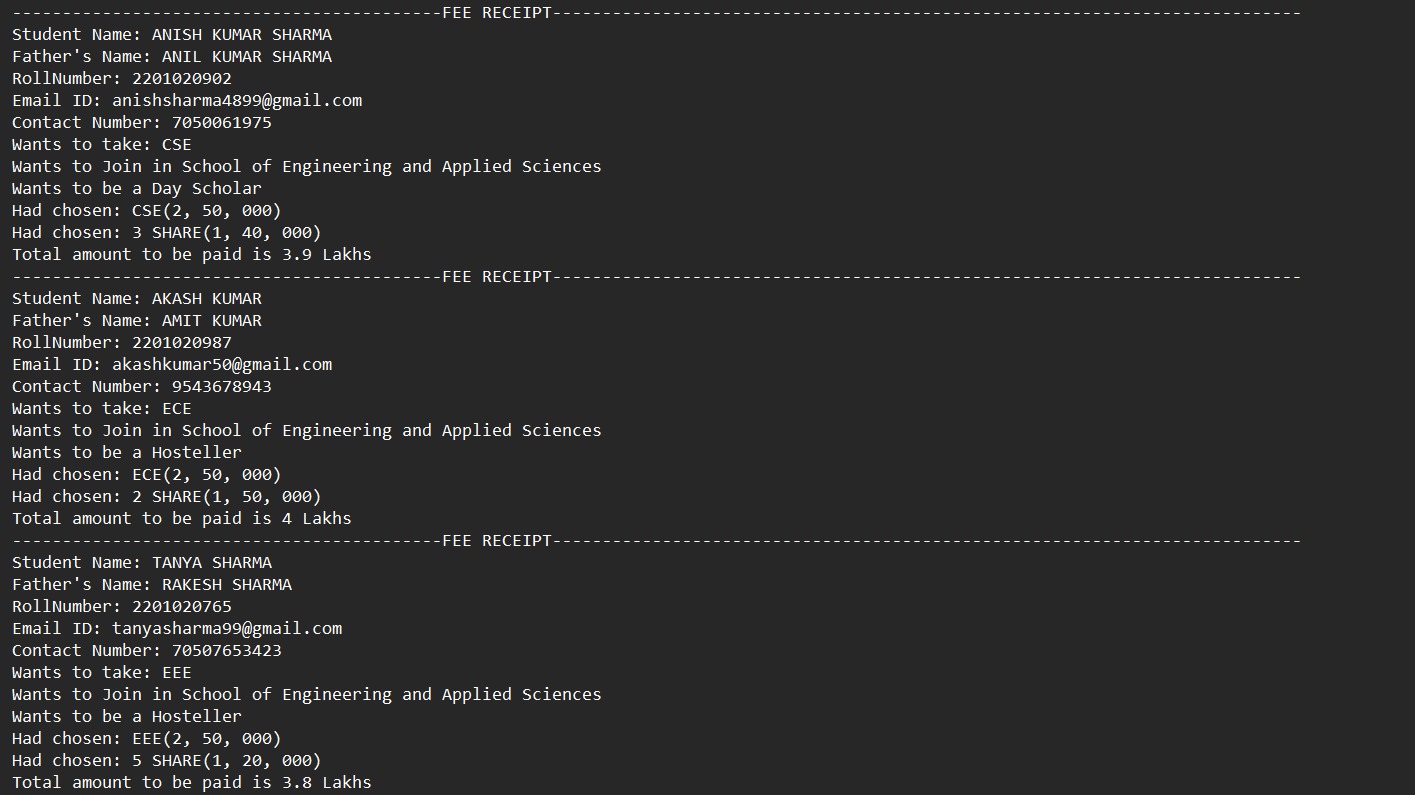
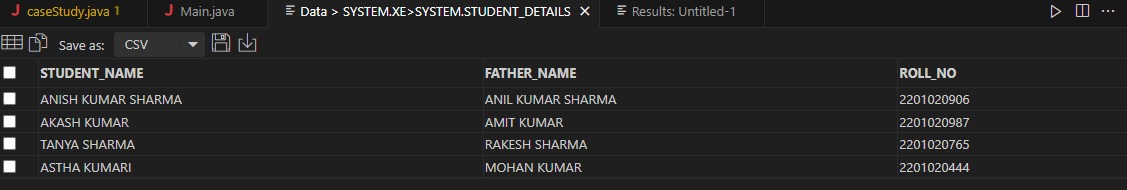


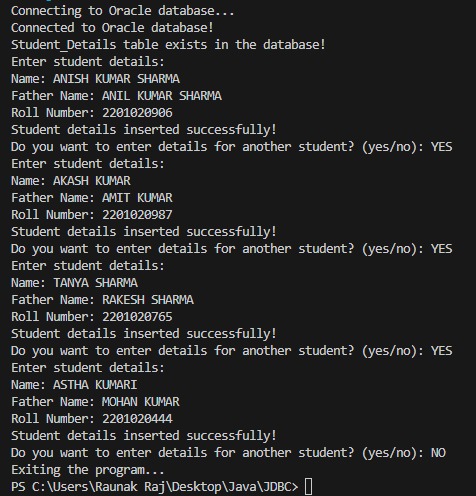






**RESULT**



**CONCLUSION**

**In conclusion, the implementation of a fee management system represents a pivotal step towards modernizing and optimizing financial operations within organizations. By automating fee collection, tracking payments, and managing financial records, these systems enhance efficiency, transparency, and accountability. The convenience of online payment options improves user experience, while robust security measures ensure the safety of transactions and data. Furthermore, fee management systems offer valuable insights through analytics, enabling institutions to make informed decisions and forecast revenue more accurately. With scalability and integration capabilities, these systems can adapt to the evolving needs of institutions and seamlessly integrate with existing infrastructure. Overall, the adoption of fee management systems not only streamlines administrative processes but also contributes to cost savings, user satisfaction, and improved financial management**

**REFERENCES**

**1: Lemke and Lins, Regulation of Investment Advisers, §2:9 (Thomson West, 2013 ed.)**

**2: SEC.gov. They are also sometimes called maintenance fees.**

**3: Lemke, Lins and Smith, Regulation of Investment Companies, §7.11 and §9.09[2] (Matthew Bender, 2013 ed.).**

**4: Private equity industry dictionary Archived 2008-05-05 at the Wayback Machine. CalPERS Alternative Investment Program**