



Project Report

CSE.Search Engine

Java-Based Student Directory System

Project by:

- **Ibad Ur Rahman (133-22-0004)**
- **Waqar Ahmed (133-22-0021)**
- **Zaghum Abbas (133-22-0034)**

Department of Computer Systems Engineering

1. Introduction

In academic institutions, efficiently managing and retrieving student records is crucial for various academic and administrative functions. Our project, 'CSE.Search Engine', is a Java-based desktop application designed for the Computer Systems Engineering (CSE) department. The software enables users to search for student information quickly and reliably using a graphical user interface (GUI) built with Java Swing and connected to an SQLite database. The project was inspired by the simplicity and effectiveness of the Google Chrome search engine, aiming to provide a similar experience for internal student data search within the Computer Systems Engineering (CSE) department. It mimics the intuitive search interface and delivers quick, relevant results from a local database.

2. Objectives

- Develop a functional search engine for student data.
- Store essential student records in a structured and secure database.
- To mimic the Google Chrome search engine's user-friendly interface.
- Provide a user-friendly Java GUI for interaction.
- Implement efficient and fast search algorithms for student lookup.
- Offer an expandable framework for future enhancements.

3. Scope of the Project

The project is intended for use by students, faculty, and administrative staff in the Computer Systems Engineering department. It allows users to search for students by their ID or name and view relevant details such as email and batch. This tool simplifies student identification and contact management within the department.

4. Tools & Technologies Used

Component	Details
Language	Java
GUI Framework	Java Swing
Database	SQLite
IDE	Blue j
JDBC Driver	sqlite-jdbc
Operating System	Windows / Cross-platform

5. System Architecture

The system follows a layered architecture:

- User Interface: Java Swing for interaction
- Logic Layer: Java application with event handling
- Data Layer: SQLite database accessed through JDBC

The system handles user input, performs database queries, and displays results accordingly.

6. Database Design

Database: Students_CSE.db

Table: students

Column Name	Data Type
Id	TEXT
Name	TEXT
Email	TEXT
Batch	TEXT

7. Functional Description

Users enter a student's name or ID into the search bar and press the Search button. The system looks up the SQLite database and retrieves matching records. If a student is found, the system displays their name, ID, email, and batch. If not, it displays a 'No Results Found' dialog.

8. GUI Elements

- Logo and branding header
- Main title displaying 'CSE.Search Engine'
- Rounded search bar with placeholder text
- Search button with click functionality
- Dynamic window for displaying student results

9. Challenges Faced

- Integrating SQLite using JDBC
- Managing GUI component layout
- Handling null and incorrect input cases
- Ensuring placeholder behavior works on focus events

10. Future Improvements

- Include profile pictures for students
- Add filters like year, batch, or department
- Create admin panel for record management
- Export results to PDF or Excel
- Convert to a web or mobile application

11. Conclusion

The CSE.Search Engine successfully demonstrates the utility of Java and SQLite for academic software development. It meets the departmental need for a quick and intuitive student lookup system and forms a solid base for further enhancement into a comprehensive academic portal.

The CSE.Search Engine is a user-centric application that fulfills its goal of quick and easy student lookup.

Inspired by the user experience of Google Chrome's search engine, this project demonstrates how simplicity and performance can be balanced even in academic tools. It also showcases practical implementation of Java development, GUI design, and database management.

12. Team Information

Ibad Ur Rahman – Lead Developer & Designer

Waqar Ahmed – Co-Designer

Zaghum Abbas – Database Assistant

13. Appendix

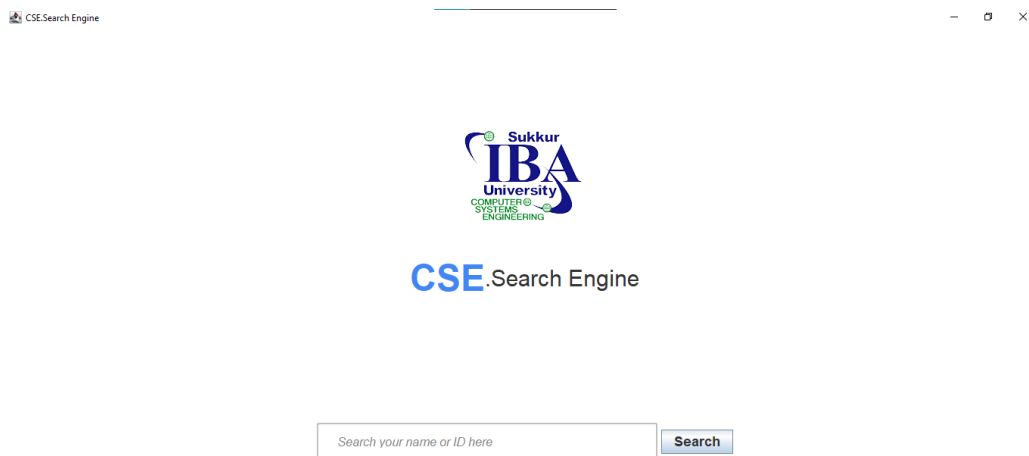
Sample SQL to Create Table:

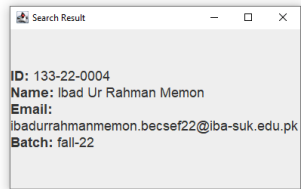
```
CREATE TABLE students (  
    Id TEXT PRIMARY KEY,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL,  
    batch TEXT NOT NULL  
);
```

Sample Data Entry:

```
INSERT INTO students (Id, name, email, batch)  
VALUES ('133-22-0004', 'Ibad Ur Rahman',  
'ibadurrahmanmemon.becsef22@iba-suk.edu.pk', 'Batch-22');
```

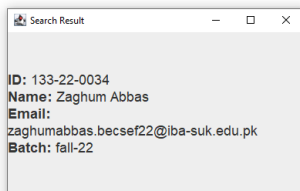
14. Results





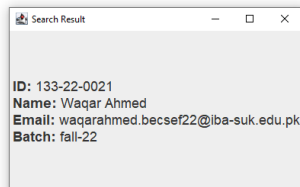
CSE.Search Engine

Search



CSE.Search Engine

Search



CSE.Search Engine

Search