DBMS PROJECT REPORT

Student-Job Matching System

Name: Shreya Srinivasan	**Name: Varshini M**
SRN : PES2UG22CS536	**SRN : PES1UG22CS678**

Abstract

The Student-Job Matching System is designed to facilitate the recruitment process by connecting students with potential employers based on their skills, academic backgrounds. The system will provide a platform where students can manage their resumes and apply for job opportunities, while employers can filter through candidates and track applications effectively.

List of Softwares/Tools/Programming languages used

*Programming Languages**:***

Python

Web Framework:

Streamlit – Framework for building the interactive dashboard and front-end for the project.

Database:

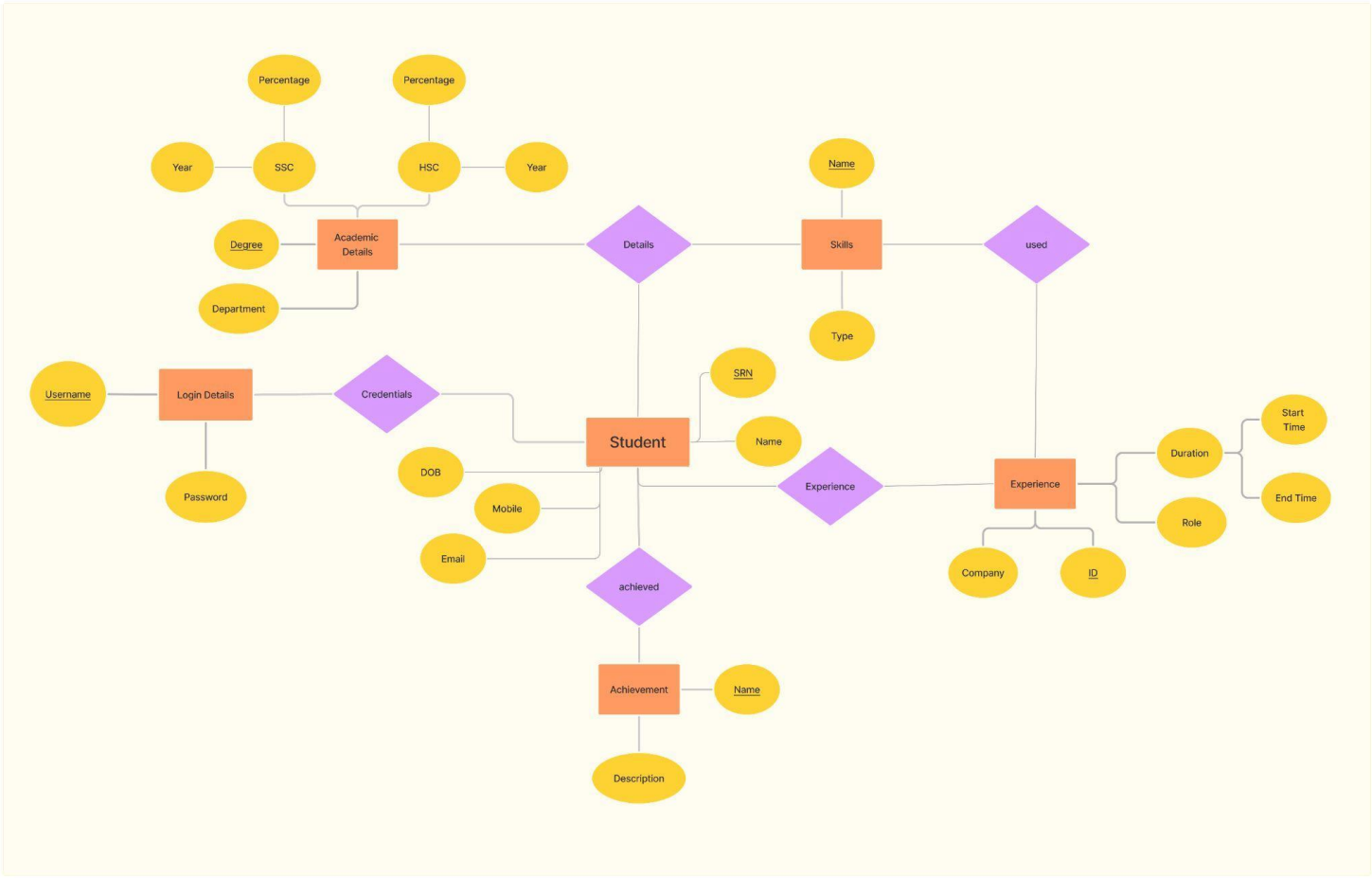
MySQL

Database Connectivity:

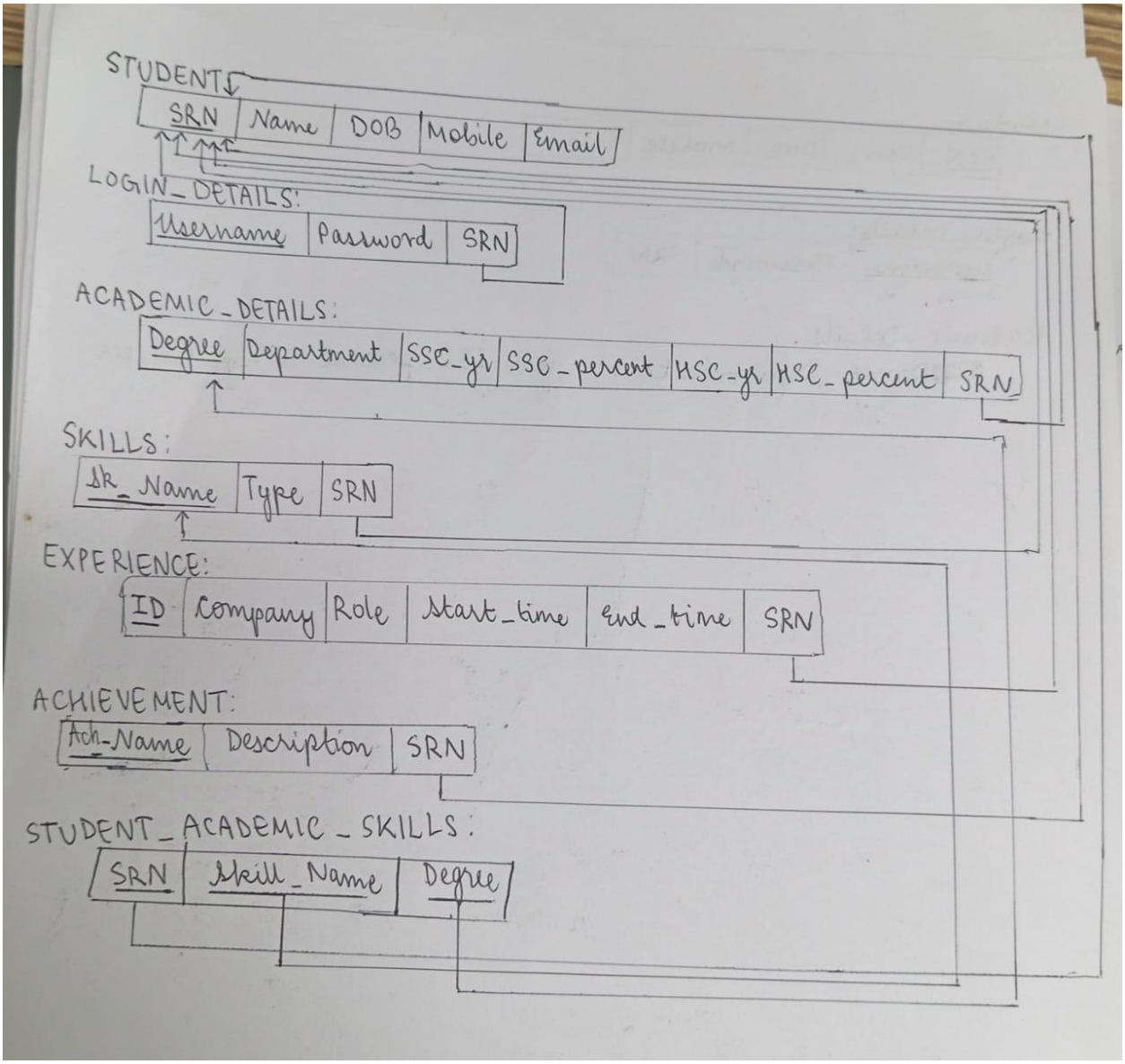
mysql-connector-python – Python library to connect and MySQL database

Other Tools: Github

ER Diagram

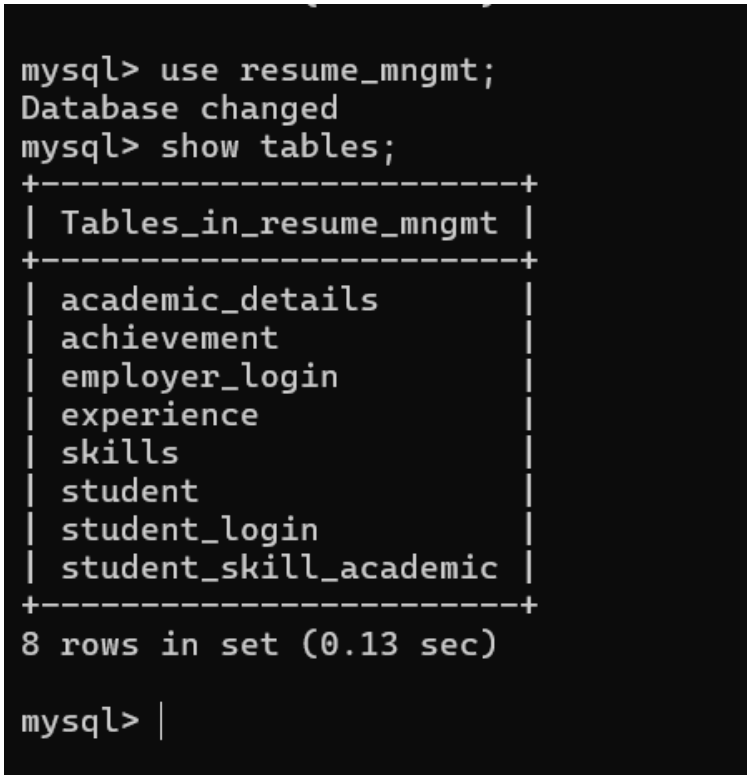

Percentage
Percentage
Year
SSC
HSC
Year
Name
Degree
Academic Details
Details
Skills
used
Department
Type
Username
Login Details
Credentials
SRN
Student
Name
DOB
Mobile
Email
Password
Experience
Experience
Duration
Start Time
End Time
Role
Company
ID
achieved
Achievement
Name
Description

Relational Schema



CRUD operation Screenshots

Tables



```
mysql> use resume_mngmt;
Database changed
mysql> show tables;
+------------------------+
| Tables_in_resume_mngmt |
+------------------------+
| academic_details       |
| achievement            |
| employer_login         |
| experience             |
| skills                 |
| student                |
| student_login          |
| student_skill_academic |
+------------------------+
8 rows in set (0.13 sec)

mysql>
```

A) Create Tables

```
mysql> show create table academic_details;
+------------------+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
| Table            | Create Table



                                                                                |
+------------------+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
| academic_details | CREATE TABLE `academic_details` (
  `Degree` varchar(50) NOT NULL,
  `Department` varchar(100) DEFAULT NULL,
  `SSC_Year` int DEFAULT NULL,
  `SSC_Percent` decimal(5,2) DEFAULT NULL,
  `HSC_Year` int DEFAULT NULL,
  `HSC_Percent` decimal(5,2) DEFAULT NULL,
  `SRN` varchar(10) NOT NULL,
  `CGPA` decimal(3,1) NOT NULL,
  PRIMARY KEY (`SRN`,`Degree`),
  CONSTRAINT `academic_details_ibfk_1` FOREIGN KEY (`SRN`) REFERENCES `student` (`SRN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+------------------+--------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
1 row in set (0.04 sec)

mysql> desc academic_details;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| Degree      | varchar(50)  | NO   | PRI | NULL    |       |
| Department  | varchar(100) | YES  |     | NULL    |       |
| SSC_Year    | int          | YES  |     | NULL    |       |
| SSC_Percent | decimal(5,2) | YES  |     | NULL    |       |
| HSC_Year    | int          | YES  |     | NULL    |       |
| HSC_Percent | decimal(5,2) | YES  |     | NULL    |       |
| SRN         | varchar(10)  | NO   | PRI | NULL    |       |
| CGPA        | decimal(3,1) | NO   |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
8 rows in set (0.01 sec)
```

```
mysql> show create table achievement;
+-------------+---------------------------------------------------------------------
---------------------------------------------------------------------------------------
| Table       | Create Table

+-------------+---------------------------------------------------------------------
---------------------------------------------------------------------------------------
| achievement | CREATE TABLE `achievement` (
  `Achievement_Name` varchar(100) NOT NULL,
  `Description` text,
  `SRN` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`Achievement_Name`),
  KEY `SRN` (`SRN`),
  CONSTRAINT `achievement_ibfk_1` FOREIGN KEY (`SRN`) REFERENCES `student` (`SRN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-------------+---------------------------------------------------------------------
---------------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> desc achievement;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| Achievement_Name | varchar(100) | NO   | PRI | NULL    |       |
| Description      | text         | YES  |     | NULL    |       |
| SRN              | varchar(10)  | YES  | MUL | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> show create table employer_login;
+----------------+------------------------------------------------------------------
---------------------------------------------------------------------------------------
| Table          | Create Table

+----------------+------------------------------------------------------------------
---------------------------------------------------------------------------------------
| employer_login | CREATE TABLE `employer_login` (
  `Employer_ID` int NOT NULL,
  `Email` varchar(100) NOT NULL,
  `Password` varchar(100) NOT NULL,
  PRIMARY KEY (`Employer_ID`),
  UNIQUE KEY `Email` (`Email`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+----------------+------------------------------------------------------------------
---------------------------------------------------------------------------------------
1 row in set (0.01 sec)

mysql> desc employer_login;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| Employer_ID | int          | NO   | PRI | NULL    |       |
| Email       | varchar(100) | NO   | UNI | NULL    |       |
| Password    | varchar(100) | NO   |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> show create table experience;
+------------+---------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--------------------------------------------+
| Table      | Create Table


                                            |
+------------+---------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--------------------------------------------+
| experience | CREATE TABLE `experience` (
  `Experience_ID` varchar(15) NOT NULL,
  `Company` varchar(100) DEFAULT NULL,
  `Role` varchar(100) DEFAULT NULL,
  `Duration` varchar(50) DEFAULT NULL,
  `SRN` varchar(10) DEFAULT NULL,
  `Skill_Name` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`Experience_ID`),
  KEY `SRN` (`SRN`),
  KEY `Skill_Name` (`Skill_Name`),
  CONSTRAINT `experience_ibfk_1` FOREIGN KEY (`SRN`) REFERENCES `student` (`SRN`),
  CONSTRAINT `experience_ibfk_2` FOREIGN KEY (`Skill_Name`) REFERENCES `skills` (`Skill_Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+------------+---------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------------------
--------------------------------------------+
1 row in set (0.08 sec)

mysql> desc experience;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Experience_ID | varchar(15)  | NO   | PRI | NULL    |       |
| Company       | varchar(100) | YES  |     | NULL    |       |
| Role          | varchar(100) | YES  |     | NULL    |       |
| Duration      | varchar(50)  | YES  |     | NULL    |       |
| SRN           | varchar(10)  | YES  | MUL | NULL    |       |
| Skill_Name    | varchar(100) | YES  | MUL | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> show create table skills;
+--------+-----------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
| Table  | Create Table

+--------+-----------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
| skills | CREATE TABLE `skills` (
  `Skill_Name` varchar(100) NOT NULL,
  `Type` enum('Technical','Non-technical') DEFAULT NULL,
  `SRN` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`Skill_Name`),
  KEY `fk_srn` (`SRN`),
  CONSTRAINT `fk_srn` FOREIGN KEY (`SRN`) REFERENCES `student` (`SRN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+--------+-----------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> desc skills;
+------------+-----------------------------------+------+-----+---------+-------+
| Field      | Type                              | Null | Key | Default | Extra |
+------------+-----------------------------------+------+-----+---------+-------+
| Skill_Name | varchar(100)                      | NO   | PRI | NULL    |       |
| Type       | enum('Technical','Non-technical') | YES  |     | NULL    |       |
| SRN        | varchar(10)                       | YES  | MUL | NULL    |       |
+------------+-----------------------------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

```
mysql> show create table student;
+---------+----------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
| Table   | Create Table

+---------+----------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
| student | CREATE TABLE `student` (
  `SRN` varchar(10) NOT NULL,
  `Name` varchar(100) DEFAULT NULL,
  `DOB` date DEFAULT NULL,
  `Mobile` varchar(15) DEFAULT NULL,
  `Email` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`SRN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+---------+----------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> desc student;
+--------+--------------+------+-----+---------+-------+
| Field  | Type         | Null | Key | Default | Extra |
+--------+--------------+------+-----+---------+-------+
| SRN    | varchar(10)  | NO   | PRI | NULL    |       |
| Name   | varchar(100) | YES  |     | NULL    |       |
| DOB    | date         | YES  |     | NULL    |       |
| Mobile | varchar(15)  | YES  |     | NULL    |       |
| Email  | varchar(100) | YES  |     | NULL    |       |
+--------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> show create table student_login;
+---------------+--------------------------------------------------------------------
| Table         | Create Table
+---------------+--------------------------------------------------------------------
| student_login | CREATE TABLE `student_login` (
  `Username` varchar(50) NOT NULL,
  `Password` varchar(100) NOT NULL,
  `SRN` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`Username`),
  KEY `SRN` (`SRN`),
  CONSTRAINT `student_login_ibfk_1` FOREIGN KEY (`SRN`) REFERENCES `student` (`SRN`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+---------------+--------------------------------------------------------------------
1 row in set (0.00 sec)

mysql> desc student_login;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| Username | varchar(50)  | NO   | PRI | NULL    |       |
| Password | varchar(100) | NO   |     | NULL    |       |
| SRN      | varchar(20)  | YES  | MUL | NULL    |       |
+----------+--------------+------+-----+---------+-------+
3 rows in set (0.01 sec)
```

```
mysql> show create table student_skill_academic;
+------------------------+--------------------------------------------------------------------
| Table                  | Create Table
                                                                                             |
+------------------------+--------------------------------------------------------------------
| student_skill_academic | CREATE TABLE `student_skill_academic` (
  `Student_SRN` varchar(10) NOT NULL,
  `Academic_Degree` varchar(50) NOT NULL,
  `Skill_Name` varchar(100) NOT NULL,
  PRIMARY KEY (`Student_SRN`,`Academic_Degree`,`Skill_Name`),
  KEY `Skill_Name` (`Skill_Name`),
  CONSTRAINT `student_skill_academic_ibfk_1` FOREIGN KEY (`Student_SRN`) REFERENCES `student` (`SRN`),
  CONSTRAINT `student_skill_academic_ibfk_2` FOREIGN KEY (`Student_SRN`, `Academic_Degree`) REFERENCES `academic_details` (`SRN`, `Degree`),
  CONSTRAINT `student_skill_academic_ibfk_3` FOREIGN KEY (`Skill_Name`) REFERENCES `skills` (`Skill_Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+------------------------+--------------------------------------------------------------------
1 row in set (0.01 sec)

mysql> desc student_skill_academic;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| Student_SRN     | varchar(10)  | NO   | PRI | NULL    |       |
| Academic_Degree | varchar(50)  | NO   | PRI | NULL    |       |
| Skill_Name      | varchar(100) | NO   | PRI | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

B) Read

```
mysql> select * from academic_details;
+--------+------------------------+----------+-------------+----------+-------------+----------+------+
| Degree | Department             | SSC_Year | SSC_Percent | HSC_Year | HSC_Percent | SRN      | CGPA |
+--------+------------------------+----------+-------------+----------+-------------+----------+------+
| B.Tech | Computer Science       |     2016 |       85.00 |     2018 |       90.00 | SRN00001 |  6.9 |
| B.Sc   | Physics                |     2017 |       90.50 |     2019 |       88.50 | SRN00002 |  9.8 |
| B.Com  | Commerce               |     2015 |       78.00 |     2017 |       74.00 | SRN00003 |  8.7 |
| M.Tech | Information Technology |     2018 |       88.00 |     2020 |       92.00 | SRN00004 |  8.0 |
| MBA    | Business Administration|     2017 |       75.00 |     2019 |       80.00 | SRN00005 |  5.8 |
| BBA    | Management             |     2016 |       80.00 |     2018 |       86.00 | SRN00006 |  7.8 |
| BCA    | Computer Applications  |     2019 |       95.00 |     2021 |       94.00 | SRN00007 |  5.3 |
| M.Sc   | Biology                |     2015 |       82.00 |     2017 |       77.00 | SRN00008 |  9.3 |
| M.Com  | Commerce               |     2018 |       89.00 |     2020 |       89.00 | SRN00009 |  8.0 |
| Ph.D   | Engineering            |     2014 |       76.00 |     2016 |       81.00 | SRN00010 |  8.5 |
+--------+------------------------+----------+-------------+----------+-------------+----------+------+
10 rows in set (0.01 sec)
```

```
mysql> select * from achievement;
+----------------------+-------------------------------------------------------------+----------+
| Achievement_Name     | Description                                                 | SRN      |
+----------------------+-------------------------------------------------------------+----------+
| Best Intern          | Recognized as the best intern of the year                   | SRN00003 |
| Best Presentation    | Gave the best presentation in a class seminar               | SRN00010 |
| Coding Competition   | Secured a top spot in a coding competition                  | SRN00007 |
| Conference Speaker   | Delivered a talk at an international conference             | SRN00004 |
| Dean's List          | Included in the Dean's list for academic excellence         | SRN00008 |
| Hackathon Winner     | Won first place in a national hackathon                     | SRN00001 |
| Leadership Award     | Awarded for leadership in organizing university events      | SRN00009 |
| Project Excellence   | Completed an outstanding project with distinction           | SRN00005 |
| Research Publication | Published a research paper in a reputed journal             | SRN00006 |
| Top Scorer           | Achieved the highest grade in semester exams                | SRN00002 |
+----------------------+-------------------------------------------------------------+----------+
10 rows in set (0.00 sec)
```

```
mysql> select * from employer_login;
+-------------+------------------------+----------------+
| Employer_ID | Email                  | Password       |
+-------------+------------------------+----------------+
|           1 | employer1@example.com  | password123    |
|           2 | employer2@example.com  | securePass456  |
|           3 | employer3@example.com  | welcome789     |
|           4 | employer4@example.com  | adminAccess1   |
|           5 | employer5@example.com  | safePassword2  |
|           6 | employer6@example.com  | loginPassword3 |
|           7 | employer7@example.com  | employerPass4  |
|           8 | employer8@example.com  | myPassword5    |
|           9 | employer9@example.com  | companyLogin6  |
|          10 | employer10@example.com | entrySecure7   |
+-------------+------------------------+----------------+
10 rows in set (0.00 sec)
```

```
mysql> select * from experience;
+--------------+---------------+--------------------------+----------+----------+--------------------+
| Experience_ID | Company       | Role                     | Duration | SRN      | Skill_Name         |
+--------------+---------------+--------------------------+----------+----------+--------------------+
| EXP00001      | Tech Solutions | Software Engineer        | 2 years  | SRN00001 | Python             |
| EXP00002      | Innovatech     | Data Analyst             | 1 year   | SRN00002 | Java               |
| EXP00003      | DataCorp       | Project Manager          | 3 years  | SRN00003 | Data Analysis      |
| EXP00004      | ProjectWorks   | Intern                   | 6 months | SRN00004 | Project Management |
| EXP00005      | AI Labs        | Machine Learning Engineer | 1 year   | SRN00005 | Machine Learning   |
| EXP00006      | Web Solutions  | Web Developer            | 8 months | SRN00006 | Web Development    |
| EXP00007      | Cloud Services | Cloud Engineer           | 1.5 years | SRN00007 | Cloud Computing    |
| EXP00008      | SecureNet      | Cybersecurity Analyst    | 2 years  | SRN00008 | Cybersecurity      |
| EXP00009      | Design Studio  | Graphic Designer         | 9 months | SRN00009 | Graphic Design     |
| EXP00010      | CommWorks      | Communications Manager   | 1 year   | SRN00010 | Communication      |
+--------------+---------------+--------------------------+----------+----------+--------------------+
10 rows in set (0.03 sec)
```

```
mysql> select * from skills;
+--------------------+---------------+----------+
| Skill_Name         | Type          | SRN      |
+--------------------+---------------+----------+
| c                  | Technical     | SRN00003 |
| Cloud Computing    | Technical     | SRN00007 |
| Communication      | Non-technical | SRN00010 |
| Cybersecurity      | Non-technical | SRN00008 |
| Data Analysis      | Technical     | SRN00003 |
| Graphic Design     | Non-technical | SRN00009 |
| Java               | Technical     | SRN00002 |
| Machine Learning   | Technical     | SRN00005 |
| Project Management | Non-technical | SRN00004 |
| Python             | Technical     | SRN00001 |
| Web Development    | Technical     | SRN00006 |
+--------------------+---------------+----------+
11 rows in set (0.00 sec)
```

```
mysql> select * from student;
+----------+----------------+------------+------------+--------------------------+
| SRN      | Name           | DOB        | Mobile     | Email                    |
+----------+----------------+------------+------------+--------------------------+
| SRN00001 | John Doe       | 2000-01-01 | 1234567890 | johndoe@example.com      |
| SRN00002 | Jane Smith     | 1999-05-15 | 0987654321 | janesmith@example.com    |
| SRN00003 | Alice Johnson  | 1998-03-20 | 1231231234 | alicej@example.com       |
| SRN00004 | Bob Brown      | 2001-12-05 | 9879879876 | bobbrown@example.com     |
| SRN00005 | Charlie Davis  | 1997-07-30 | 4564564567 | charliedavis@example.com |
| SRN00006 | Diana Prince   | 2000-09-10 | 3213213210 | dianaprince@example.com  |
| SRN00007 | Ethan Hunt     | 1995-11-25 | 6546546543 | ethanhunt@example.com    |
| SRN00008 | Fiona Glenanne | 2002-04-17 | 7897897890 | fionag@example.com       |
| SRN00009 | George Weasley | 1996-08-08 | 1112223333 | georgeweasley@example.com |
| SRN00010 | Hannah Baker   | 1999-10-22 | 4445556666 | hannahbaker@example.com  |
+----------+----------------+------------+------------+--------------------------+
10 rows in set (0.01 sec)
```

```
mysql> select * from student_login;
+---------------+------------+----------+
| Username      | Password   | SRN      |
+---------------+------------+----------+
| alicej        | password1  | SRN00003 |
| bobbrown      | password4  | SRN00004 |
| charlied      | password5  | SRN00005 |
| dianaprince   | password6  | SRN00006 |
| ethanhunt     | password7  | SRN00007 |
| fionag        | password8  | SRN00008 |
| georgeweasley | password9  | SRN00009 |
| hannahbaker   | password10 | SRN00010 |
| janesmith     | password2  | SRN00002 |
| johndoe       | password1  | SRN00001 |
+---------------+------------+----------+
10 rows in set (0.00 sec)
```

C) Update

```
mysql> UPDATE achievement SET Description = 'Top spot' WHERE Achievement_Name = 'Coding Competition' AND SRN = 'SRN00007';
Query OK, 1 row affected (0.06 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Before:

```
mysql> select * from achievement;
+---------------------+-----------------------------------------------------------+----------+
| Achievement_Name    | Description                                               | SRN      |
+---------------------+-----------------------------------------------------------+----------+
| Best Intern         | Recognized as the best intern of the year                 | SRN00003 |
| Best Presentation   | Gave the best presentation in a class seminar             | SRN00010 |
| Coding Competition  | Secured a top spot in a coding competition                | SRN00007 |
| Conference Speaker  | Delivered a talk at an international conference           | SRN00004 |
| Dean's List         | Included in the Dean's list for academic excellence       | SRN00008 |
| Hackathon Winner    | Won first place in a national hackathon                   | SRN00001 |
| Leadership Award    | Awarded for leadership in organizing university events    | SRN00009 |
| Project Excellence  | Completed an outstanding project with distinction         | SRN00005 |
| Research Publication| Published a research paper in a reputed journal           | SRN00006 |
| Top Scorer          | Achieved the highest grade in semester exams              | SRN00002 |
+---------------------+-----------------------------------------------------------+----------+
10 rows in set (0.00 sec)
```

After:

```
mysql> select * from achievement;
+----------------------+-----------------------------------------------------------+----------+
| Achievement_Name     | Description                                               | SRN      |
+----------------------+-----------------------------------------------------------+----------+
| Best Intern          | Recognized as the best intern of the year                 | SRN00003 |
| Best Presentation    | Gave the best presentation in a class seminar             | SRN00010 |
| Coding Competition   | Top spot                                                  | SRN00007 |
| Conference Speaker   | Delivered a talk at an international conference           | SRN00004 |
| Dean's List          | Included in the Dean's list for academic excellence       | SRN00008 |
| Hackathon Winner     | Won first place in a national hackathon                   | SRN00001 |
| Leadership Award     | Awarded for leadership in organizing university events    | SRN00009 |
| Project Excellence   | Completed an outstanding project with distinction         | SRN00005 |
| Research Publication | Published a research paper in a reputed journal           | SRN00006 |
| Top Scorer           | Achieved the highest grade in semester exams              | SRN00002 |
+----------------------+-----------------------------------------------------------+----------+
10 rows in set (0.01 sec)
```

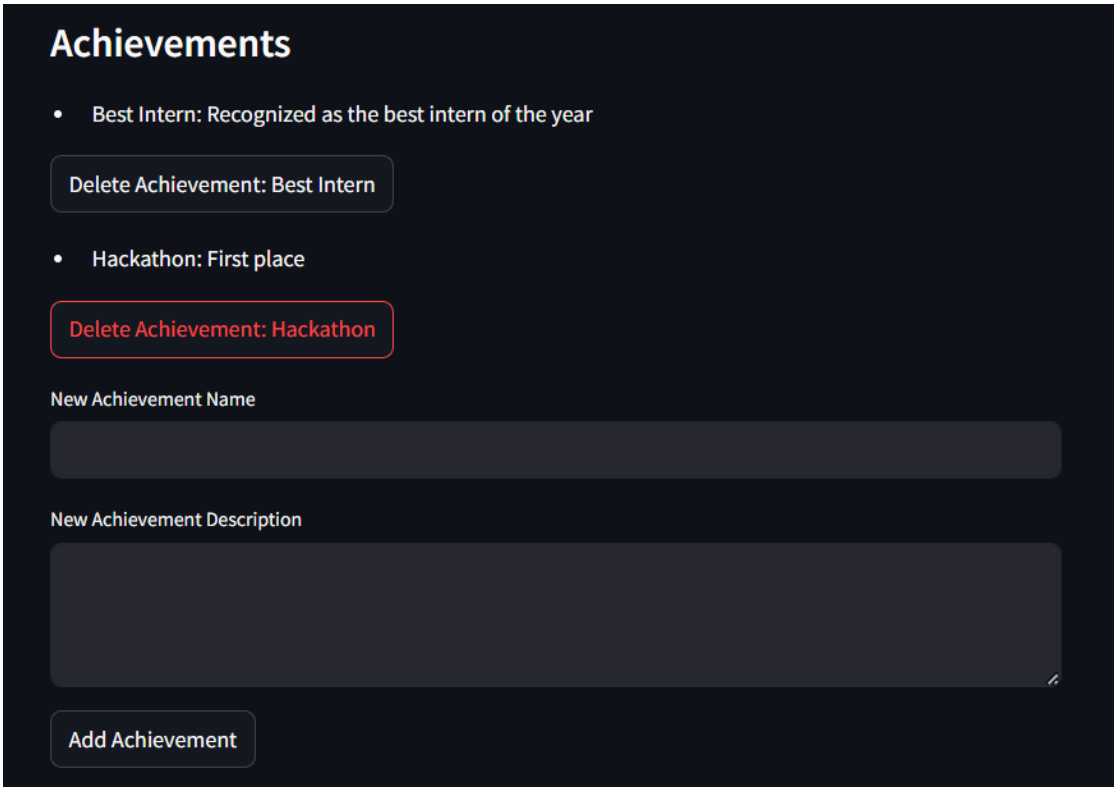
D) Delete

1) Delete Achievement

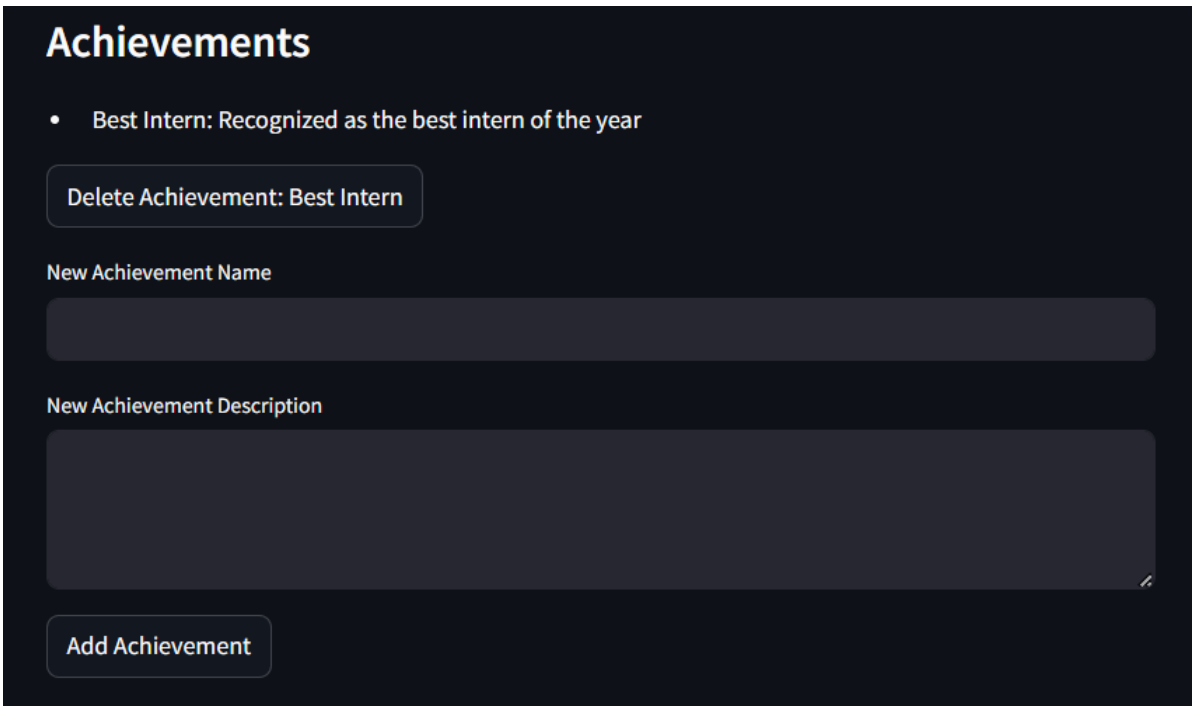
```python
# Delete a skill for the student
def delete_skill(srn, skill_name):
    conn = create_connection()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM skills WHERE SRN=%s AND Skill_Name=%s", (srn, skill_name))
    conn.commit()
    conn.close()
```

```python
# Display achievements section
st.subheader("Achievements")
for achievement in achievements:
    st.write(f"- {achievement['Achievement_Name']}: {achievement['Description']}")
    if st.button(f"Delete Achievement: {achievement['Achievement_Name']}"):
        delete_achievement(srn, achievement['Achievement_Name'])
        st.success("Achievement deleted successfully.")
        st.rerun()
```

Before:



After:

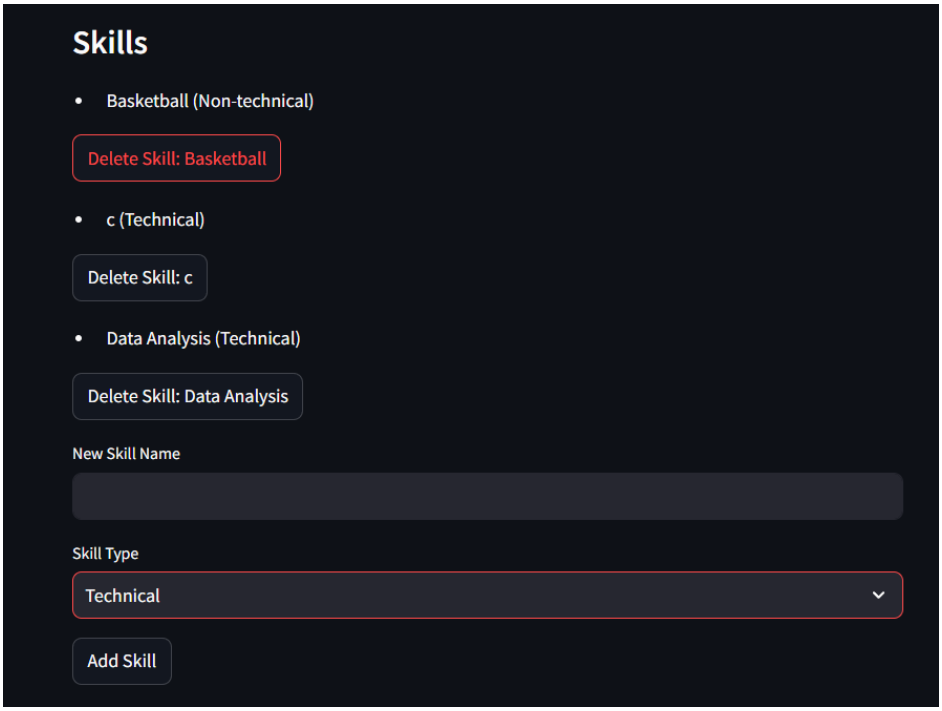


2) Delete Skill

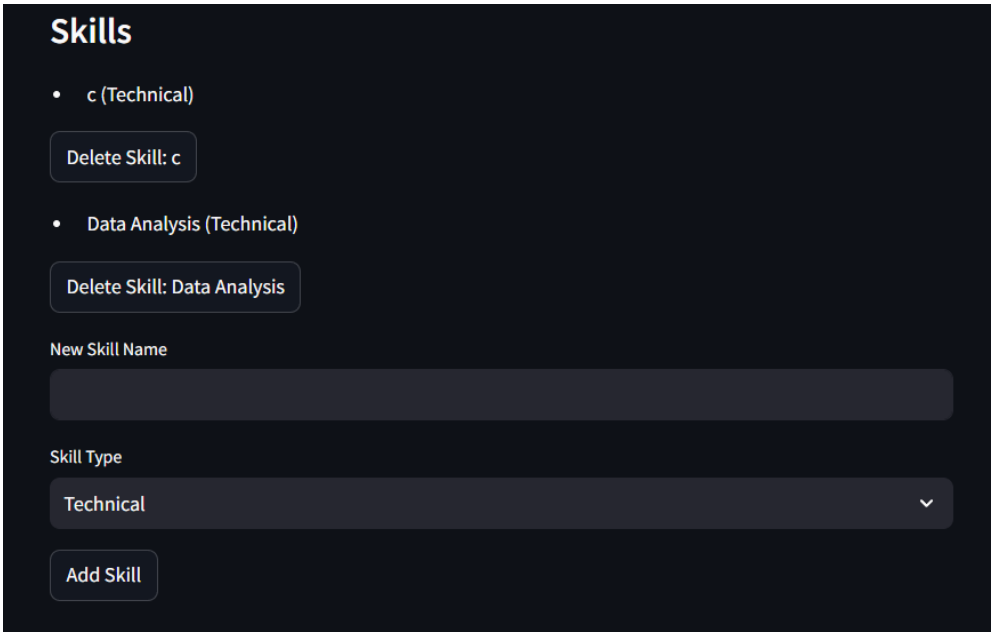
```python
# Delete an achievement for the student
def delete_achievement(srn, achievement_name):
    conn = create_connection()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM achievement WHERE SRN=%s AND Achievement_Name=%s", (srn, achievement_name))
    conn.commit()
    conn.close()
```

```python
# Display skills section
st.subheader("Skills")
for skill in skills:
    st.write(f"- {skill['Skill_Name']} ({skill['Type']})")
    if st.button(f"Delete Skill: {skill['Skill_Name']}"):
        delete_skill(srn, skill['Skill_Name'])
        st.success("Skill deleted successfully.")
        st.rerun()
```

Before:



After:



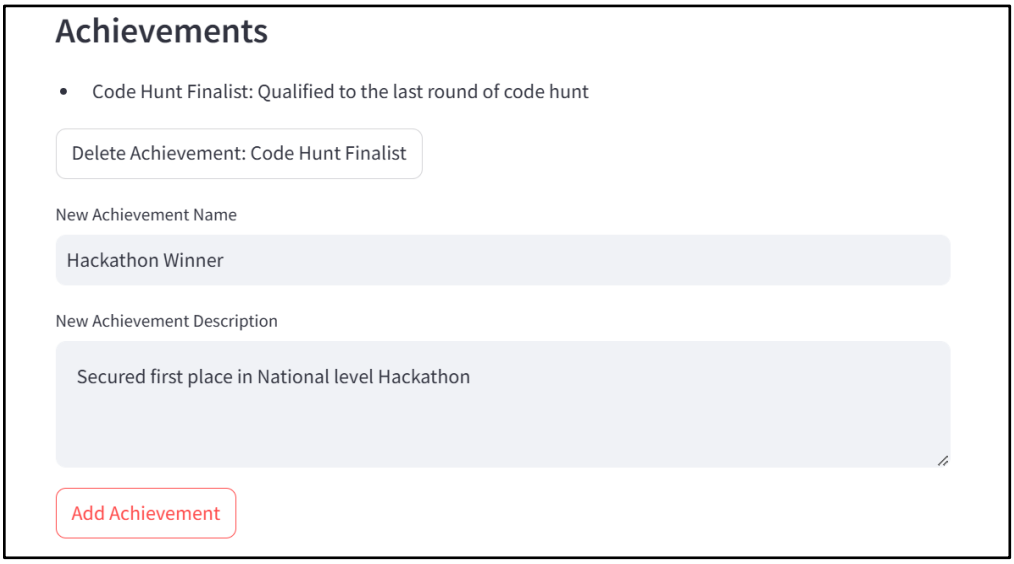
DDL Command Screenshots
Add Feature:

1) Adding Achievement

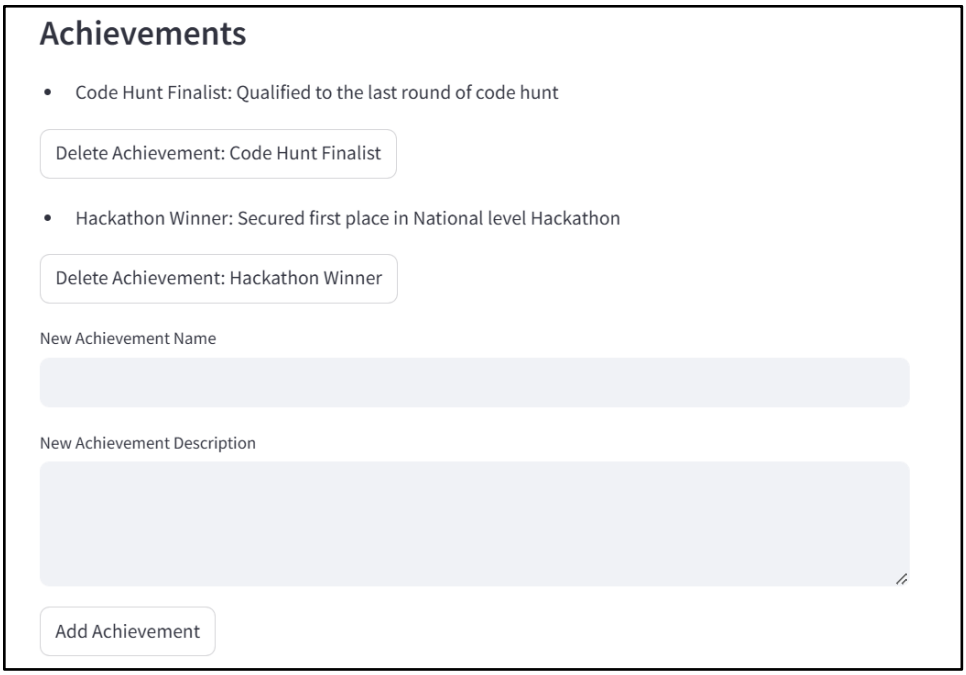
```python
# Add an achievement for the student
def add_achievement(srn, achievement_name, description):
    conn = create_connection()
    cursor = conn.cursor()
    cursor.execute("INSERT INTO achievement (Achievement_Name, Description, SRN) VALUES (%s, %s, %s)",
                   (achievement_name, description, srn))
    conn.commit()
    conn.close()

# Delete an achievement for the student
```

Before:



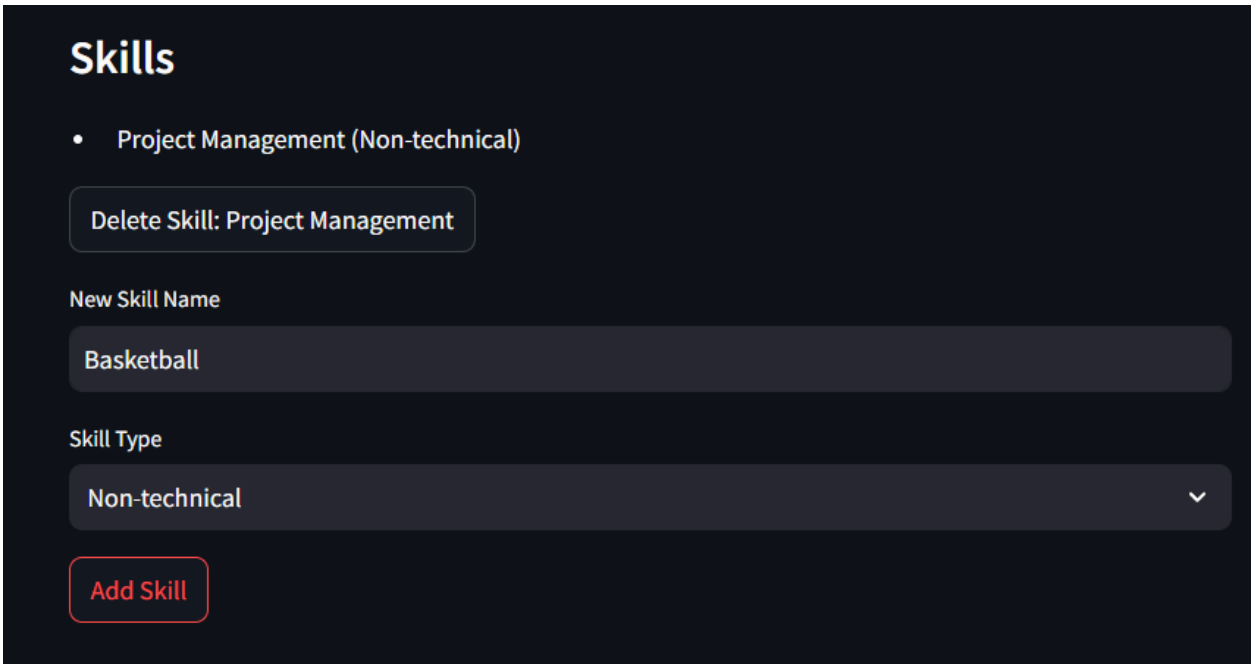
After:



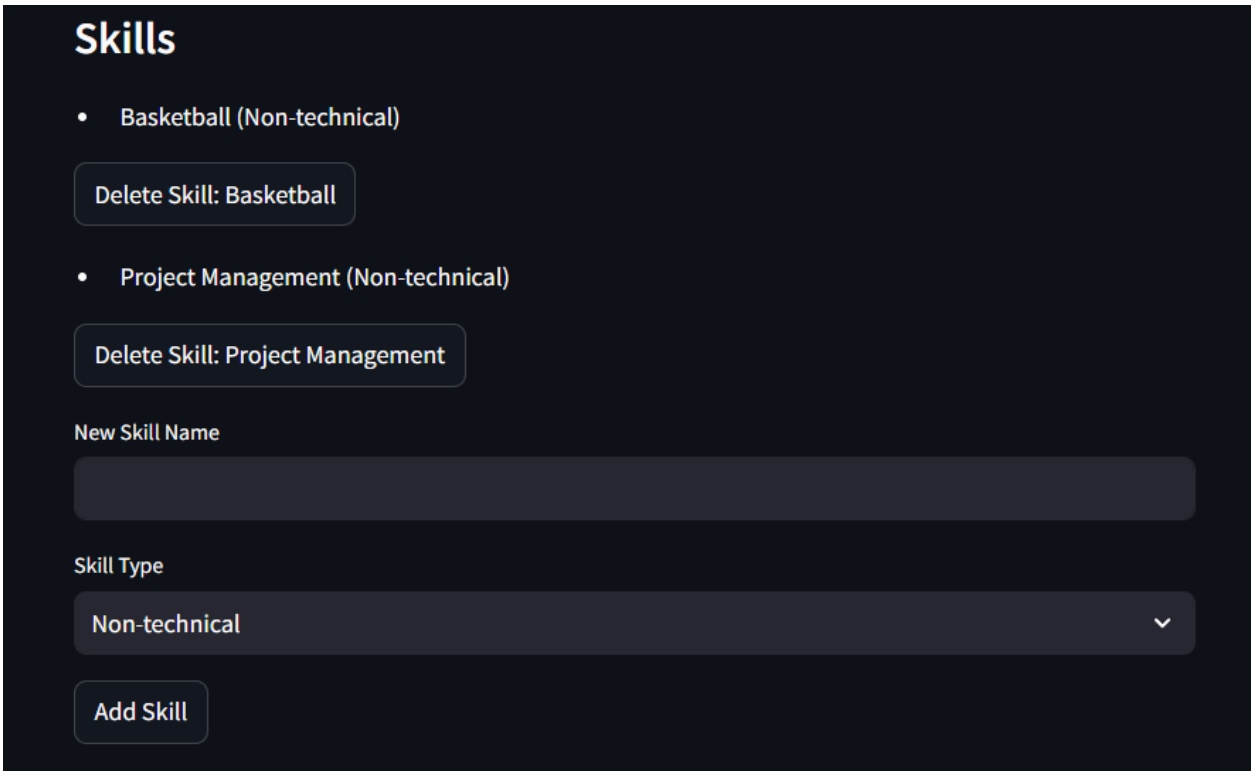
2) Adding Skill

```python
# Add a skill for the student
def add_skill(srn, skill_name, skill_type):
    conn = create_connection()
    cursor = conn.cursor()
    try:
        cursor.execute("INSERT INTO skills (Skill_Name, Type, SRN) VALUES (%s, %s, %s)",
                       (skill_name, skill_type, srn))
        conn.commit()
        st.success(f"Skill added successfully.")
    except mysql.connector.Error as err:
        if err.errno == 1644:
            st.write("<div style='color:red;'>This skill has already been added for the student.</div>", unsafe_allow_html=True)
        else:
            st.write(f"<div style='color:red;'>An unexpected error occurred: {err}</div>", unsafe_allow_html=True)
        time.sleep(20)
        st.empty()
```

Before:



After:



TRIGGERS

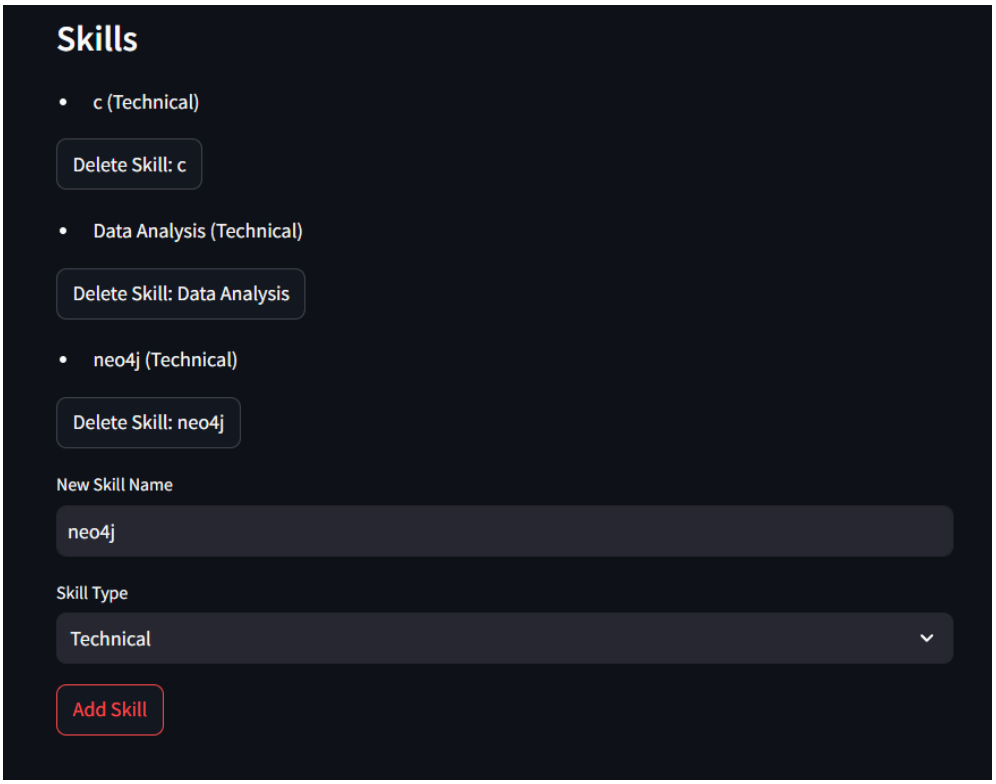
1) Trigger for duplicate Skill:

Checks if a skill for the same student (SRN and Skill_Name pair) already exists in the table.
If a duplicate is detected, it prevents the insertion by raising an error with a specific message, thus helps maintain data integrity.

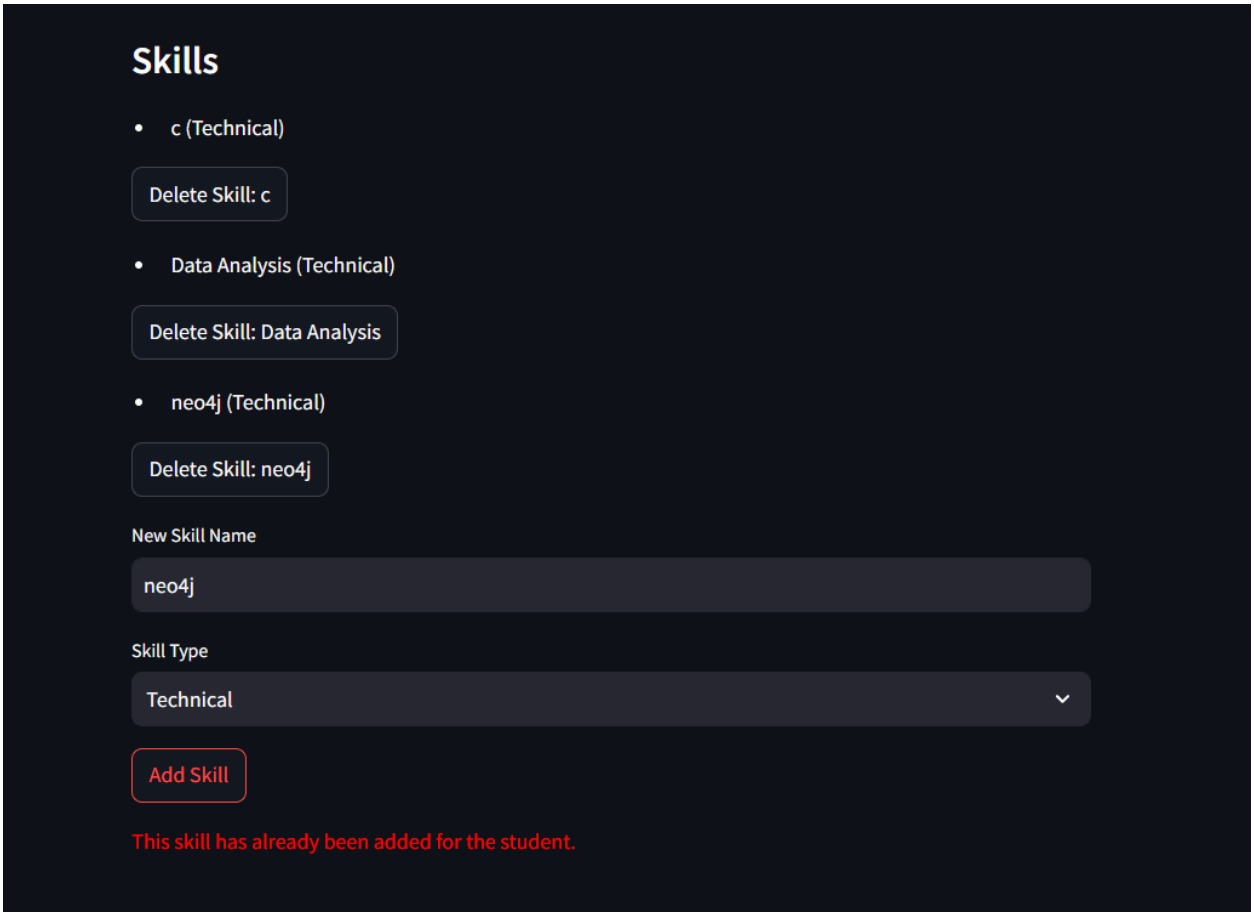
Code:

```
CREATE DEFINER=`root`@`localhost` TRIGGER `skills_BEFORE_INSERT` BEFORE INSERT ON `skills` FOR EACH ROW BEGIN
    IF EXISTS (SELECT 1 FROM `skills` WHERE `SRN` = NEW.SRN AND `Skill_Name` = NEW.Skill_Name) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Duplicate skill for the same student is not allowed';
    END IF;
END
```

Before Trigger:



After Trigger:

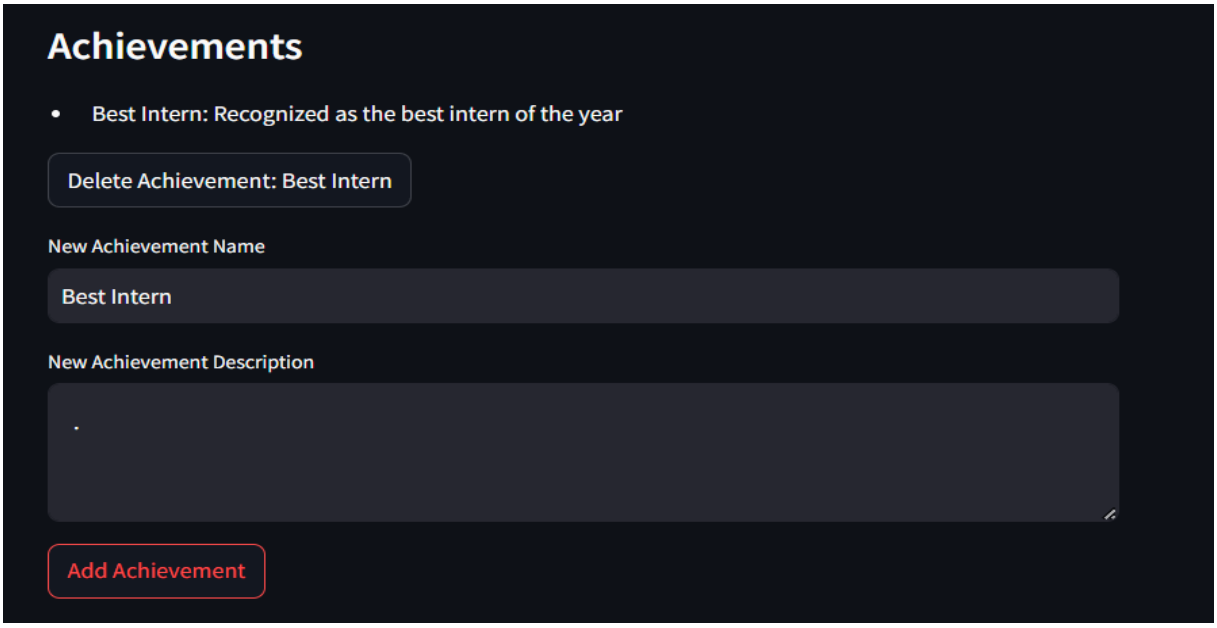


2)Trigger for duplicate Achievement

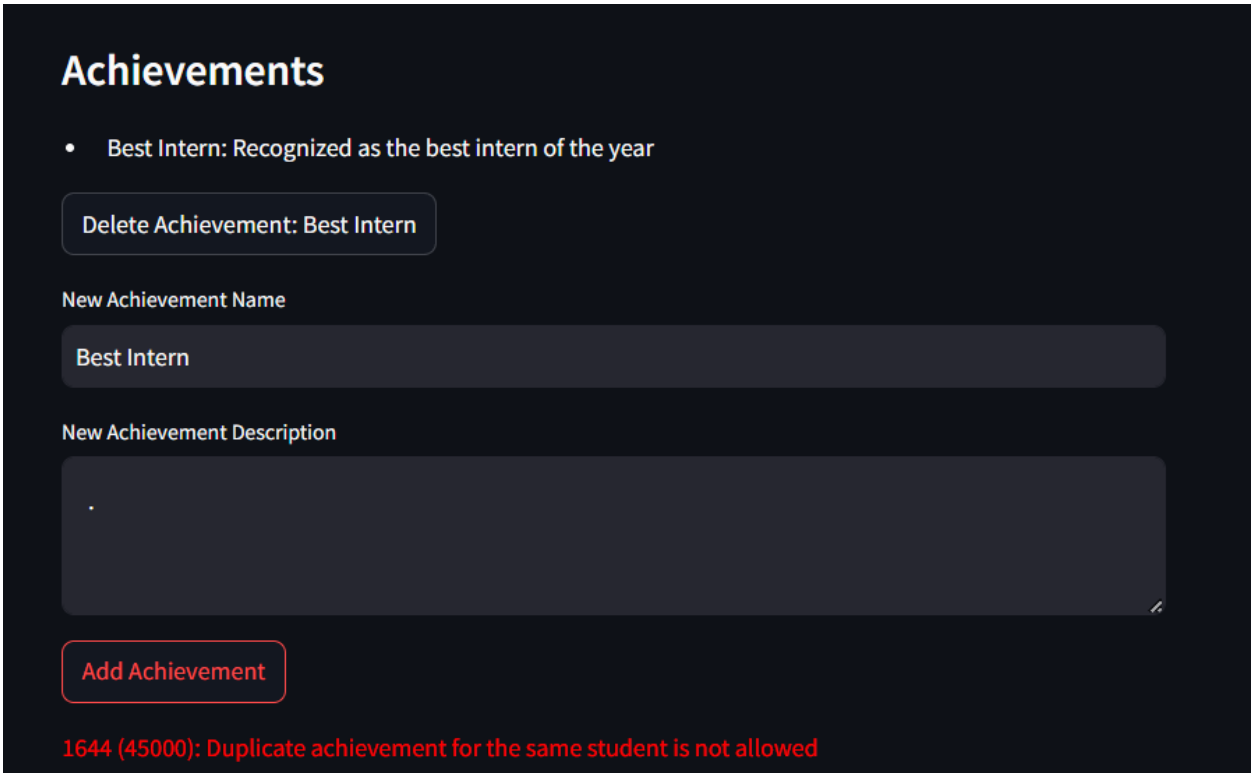
Checks if an achievement for the same student (SRN and Achievement_Name pair) already exists.
If a duplicate is detected, it prevents the insertion by raising an error with a specific message, thus helps maintain data integrity.

```
1  CREATE DEFINER=`root`@`localhost` TRIGGER `achievement_BEFORE_INSERT` BEFORE INSERT ON `achievement` FOR EACH ROW BEGIN
2      IF EXISTS (
3          SELECT 1
4          FROM achievement
5          WHERE SRN = NEW.SRN
6            AND Achievement_Name = NEW.Achievement_Name
7      ) THEN
8          SIGNAL SQLSTATE '45000'
9          SET MESSAGE_TEXT = 'Duplicate achievement for the same student is not allowed';
10     END IF;
11 END
```

Before:



After:

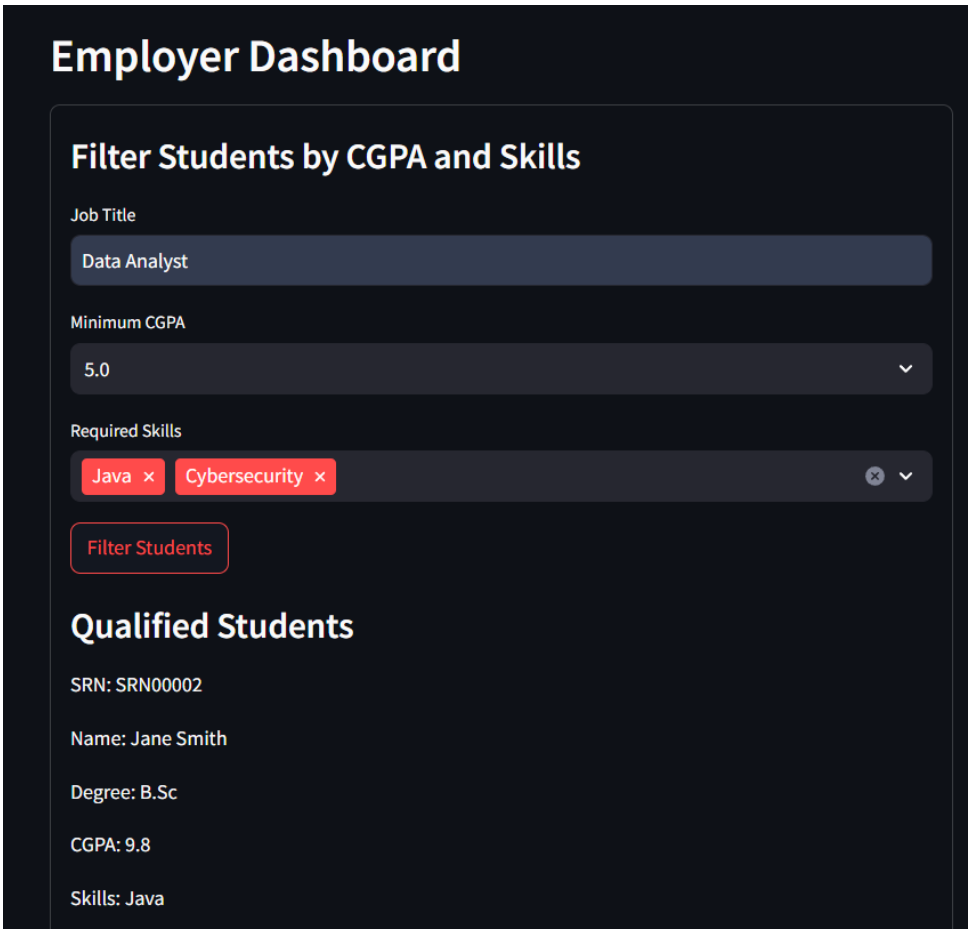


PROCEDURES / FUNCTIONS

1) filter_students_by_cgpa_and_skills

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `filter_students_by_cgpa_and_skills`(IN min_cgpa FLOAT, IN selected_skills TEXT)
BEGIN
    SET @skills_placeholder = selected_skills;
    SELECT s.SRN, s.Name, a.Degree, a.CGPA, GROUP_CONCAT(DISTINCT sk.Skill_Name) AS Skills
    FROM academic_details a
    JOIN student s ON a.SRN = s.SRN
    JOIN skills sk ON sk.SRN = s.SRN
    WHERE a.CGPA >= min_cgpa
    AND FIND_IN_SET(sk.Skill_Name, @skills_placeholder) > 0
    GROUP BY s.SRN, a.Degree, a.CGPA;
END
```

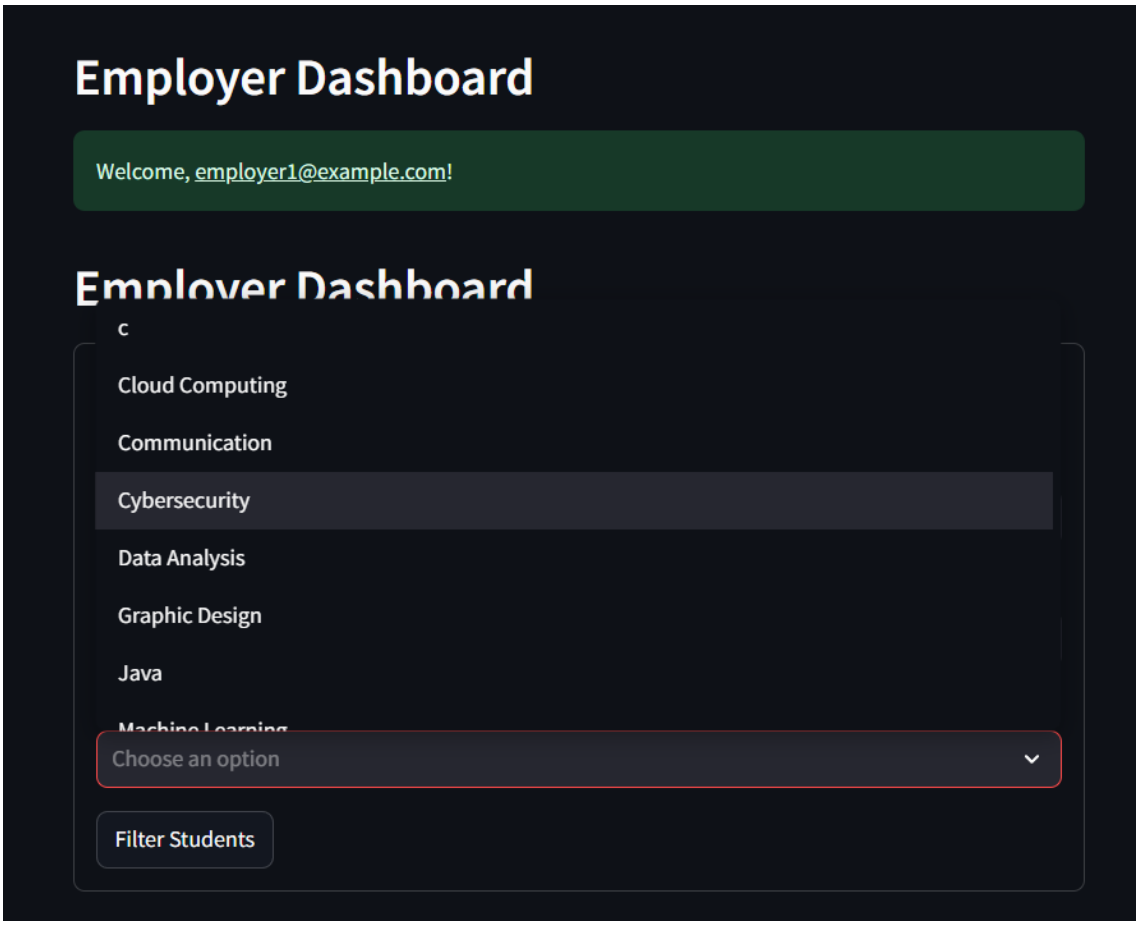
filter_students_by_cgpa_and_skills() is a stored procedure designed to retrieve students who meet specific CGPA and skill requirements



2) get_all_skills

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `get_all_skills`()
BEGIN
    SELECT DISTINCT Skill_Name
    FROM skills
    ORDER BY Skill_Name;
END
```

get_all_skills() procedure retrieves all distinct skills from the skills table and orders them alphabetically.

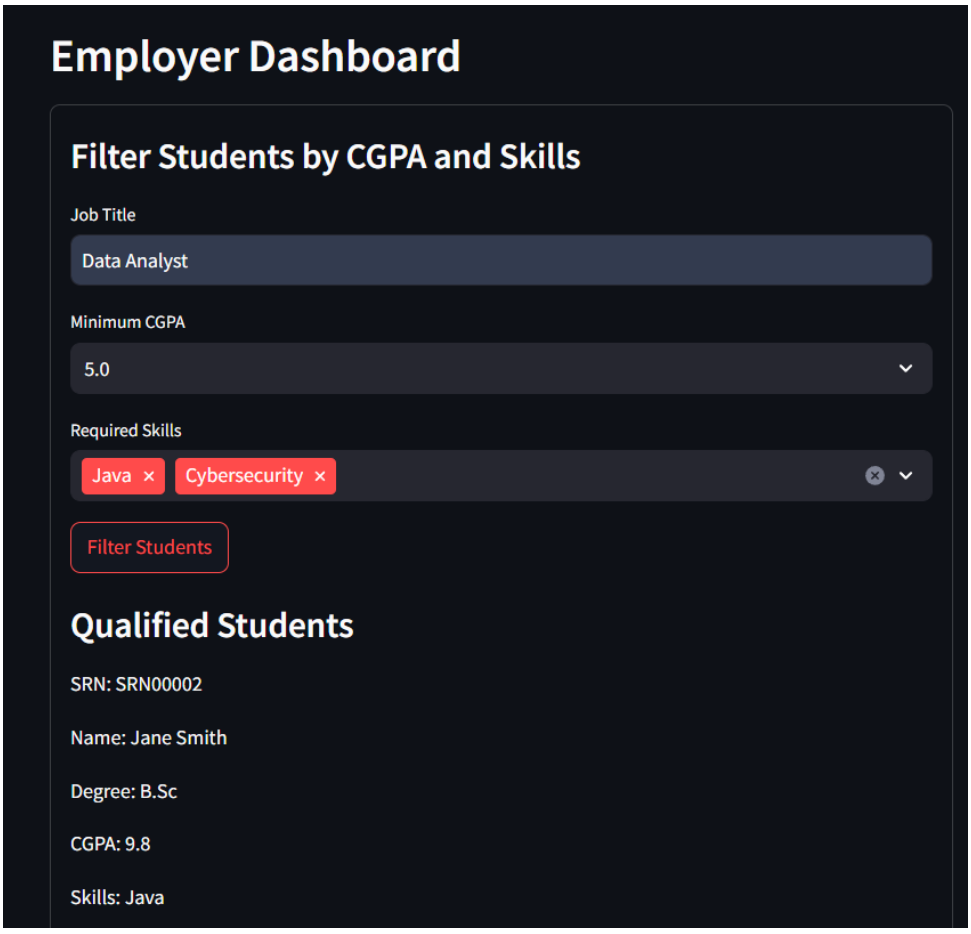


Join Command:

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `filter_students_by_cgpa_and_skills`(IN min_cgpa FLOAT, IN selected_skills TEXT)
BEGIN
    SET @skills_placeholder = selected_skills;
    SELECT s.SRN, s.Name, a.Degree, a.CGPA, GROUP_CONCAT(DISTINCT sk.Skill_Name) AS Skills
    FROM academic_details a
    JOIN student s ON a.SRN = s.SRN
    JOIN skills sk ON sk.SRN = s.SRN
    WHERE a.CGPA >= min_cgpa
    AND FIND_IN_SET(sk.Skill_Name, @skills_placeholder) > 0
    GROUP BY s.SRN, a.Degree, a.CGPA;
END
```

The query joins three tables:

- academic_details (aliased as a)
- student (aliased as s)
- skills (aliased as sk)



Nested query

```python
def get_avg_ssc_hsc_for_student_department(srn):
    conn = create_connection()
    cursor = conn.cursor(dictionary=True)
    try:
        # Nested query to get average SSC and HSC percentages for the logged-in student's department
        query = """
        SELECT Department,
               AVG((SSC_Percent + HSC_Percent) / 2) AS Average_Percent
        FROM academic_details
        WHERE Department = (SELECT Department FROM academic_details WHERE SRN = %s)
        GROUP BY Department;
        """
        # Execute the query with the student's SRN
        cursor.execute(query, (srn,))
        result = cursor.fetchall()

    except mysql.connector.Error as err:
        # Print the exact error for debugging
        st.write(f"Error: {err}")
        result = None

    # Close connection
    conn.close()
    return result
```

- The subquery (SELECT Department FROM academic_details WHERE SRN = %s) retrieves the Department for the given student SRN.
- The outer query then uses this department to filter results, allowing it to compute average scores (SSC and HSC percentages) for all students in the same department as the specified student.

Average SSC and HSC Percentages by Department

	Department	Average_Percent
0	Commerce	82.5

Student Dashboard

Student Information

	SRN	Name	DOB	Mobile	Email
0	SRN00003	Alice Johnson	1998-03-20	1231231234	alicej@example.com

Average SSC and HSC Percentages by Department

	Department	Average_Percent
0	Commerce	82.5

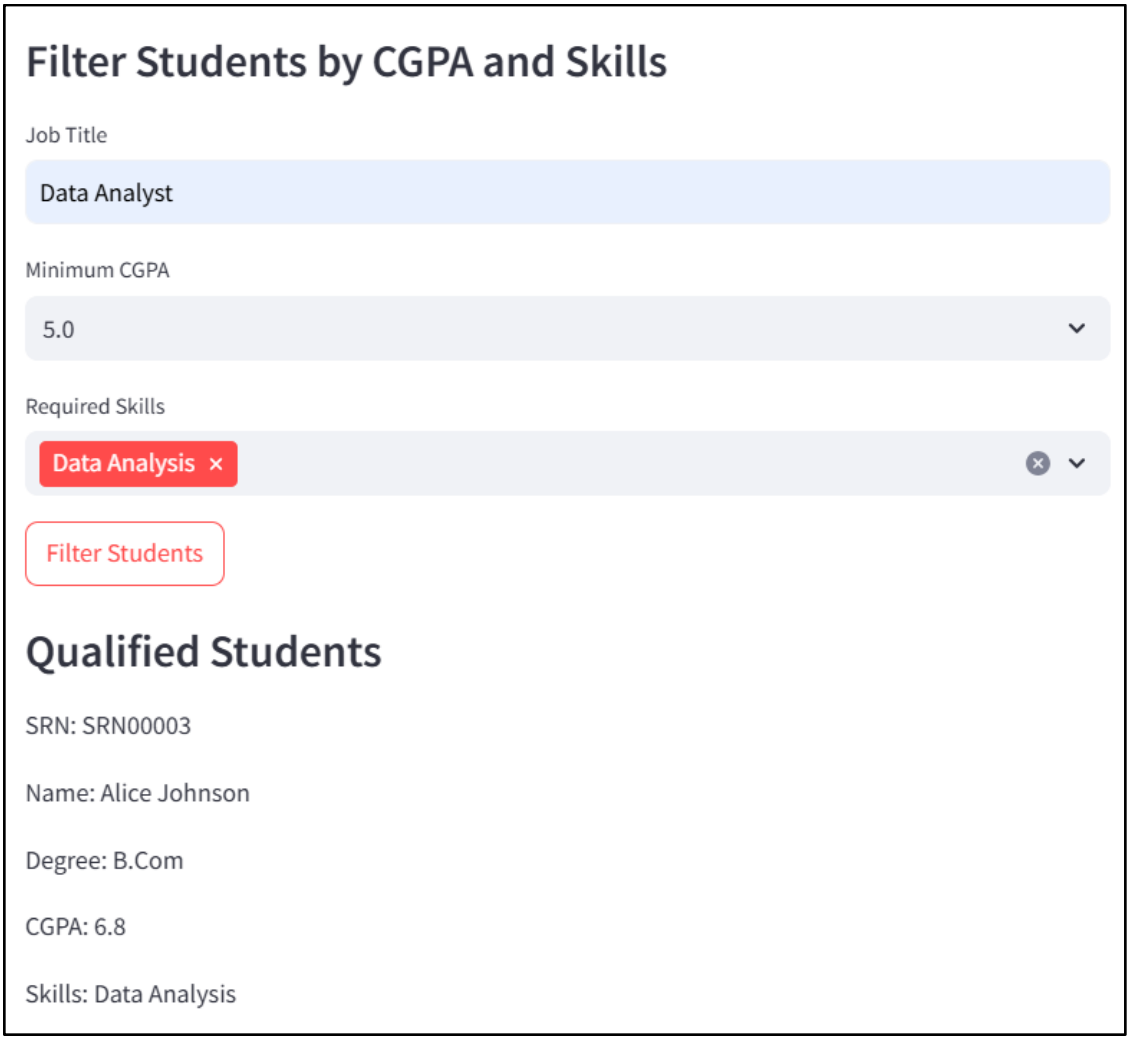
Academic Details

	Degree	Department	SSC_Year	SSC_Percent	HSC_Year	HSC_Percent	SRN	CGPA
0	B.Com	Commerce	2015	78	2017	74	SRN00003	8.7

Aggregate Query:

```sql
CREATE DEFINER=`root`@`localhost` PROCEDURE `filter_students_by_cgpa_and_skills`(IN min_cgpa FLOAT, IN selected_skills TEXT)
BEGIN
    SET @skills_placeholder = selected_skills;
    SELECT s.SRN, s.Name, a.Degree, a.CGPA, GROUP_CONCAT(DISTINCT sk.Skill_Name) AS Skills
    FROM academic_details a
    JOIN student s ON a.SRN = s.SRN
    JOIN skills sk ON sk.SRN = s.SRN
    WHERE a.CGPA >= min_cgpa
    AND FIND_IN_SET(sk.Skill_Name, @skills_placeholder) > 0
    GROUP BY s.SRN, a.Degree, a.CGPA;
END
```

Groups data by student and lists all skills for each student in a single row using GROUP_CONCAT and GROUP BY



Other Functionalities:

1) Login

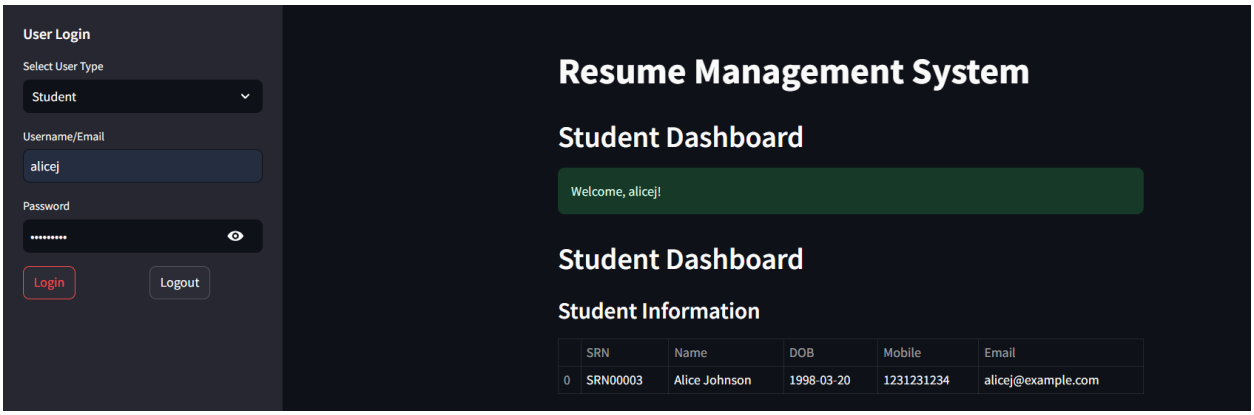
```python
# Login functionality for Student and Employer
def login_user(username, password, user_type="student"):
    conn = create_connection()
    cursor = conn.cursor(dictionary=True)

    if user_type == "student":
        cursor.execute("SELECT * FROM student_login WHERE Username=%s AND Password=%s", (username, password))
    else:
        cursor.execute("SELECT * FROM employer_login WHERE Email=%s AND Password=%s", (username, password))

    user = cursor.fetchone()
    conn.close()
    return user
```

```python
with st.sidebar:
    st.subheader("User Login")
    user_type = st.selectbox("Select User Type", ["Student", "Employer"])
    username = st.text_input("Username/Email", key="username")
    password = st.text_input("Password", type="password", key="password")
    col1, col2 = st.columns(2)
    if "login_error" not in st.session_state:
        st.session_state.login_error = None
    with col1:
        if st.button("Login"):
            user = login_user(username, password, user_type=user_type.lower())
            if user:
                st.session_state.logged_in = True
                st.session_state.user_info = user
                st.session_state.user_type = user_type
            else:
                st.session_state.login_error = "Invalid credentials. Please try again."
    with col2:
```


User Login
Select User Type
Student
Username/Email
alicej
Password
Press Enter to apply
Login
Logout
Resume Management System


User Login
Select User Type
Student
Username/Email
alicej
Password
Login
Logout
Resume Management System
Student Dashboard
Welcome, alicej!
Student Dashboard
Student Information
SRN
Name
DOB
Mobile
Email
0
SRN00003
Alice Johnson
1998-03-20
1231231234
alicej@example.com

2) Logout

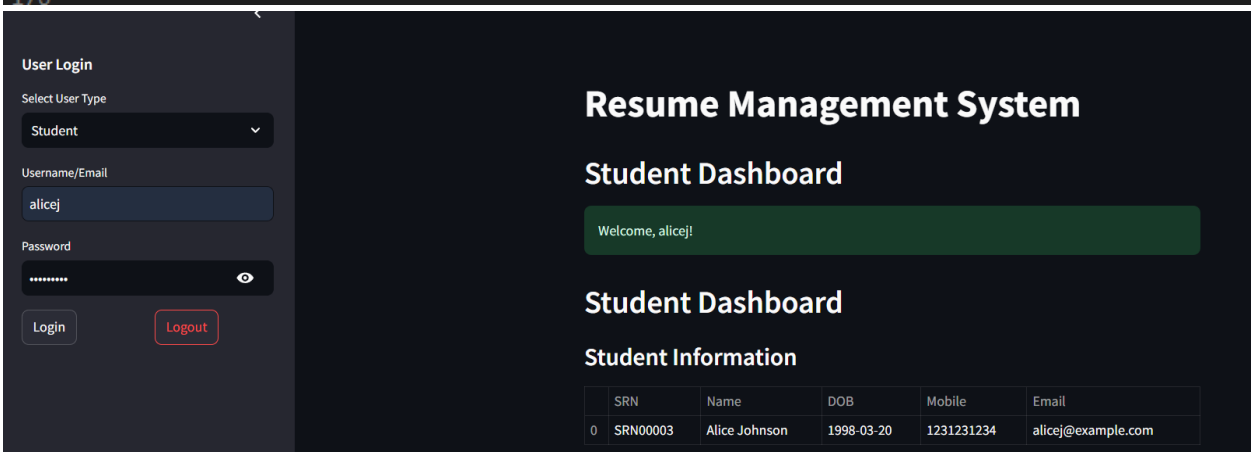
```python
# Logout function
def logout():
    st.session_state.logged_in = False
    st.session_state.user_info = None
    st.session_state.user_type = None

# Streamlit UI
st.title("Resume Management System")

# Initialize session state variables
if "logged_in" not in st.session_state:
    st.session_state.logged_in = False
    st.session_state.user_info = None
    st.session_state.user_type = None

with st.sidebar:
    st.subheader("User Login")
    user_type = st.selectbox("Select User Type", ["Student", "Employer"])
    username = st.text_input("Username/Email", key="username")
    password = st.text_input("Password", type="password", key="password")
    col1, col2 = st.columns(2)
    if "login_error" not in st.session_state:
        st.session_state.login_error = None
    with col1:
        if st.button("Login"):
            user = login_user(username, password, user_type=user_type.lower())
            if user:
                st.session_state.logged_in = True
                st.session_state.user_info = user
                st.session_state.user_type = user_type
            else:
                st.session_state.login_error = "Invalid credentials. Please try again."
    with col2:
        if st.button("Logout"):
            logout()
            st.session_state.login_error = "Logged out successfully."

```



Github repo link:

[Github Repository]