

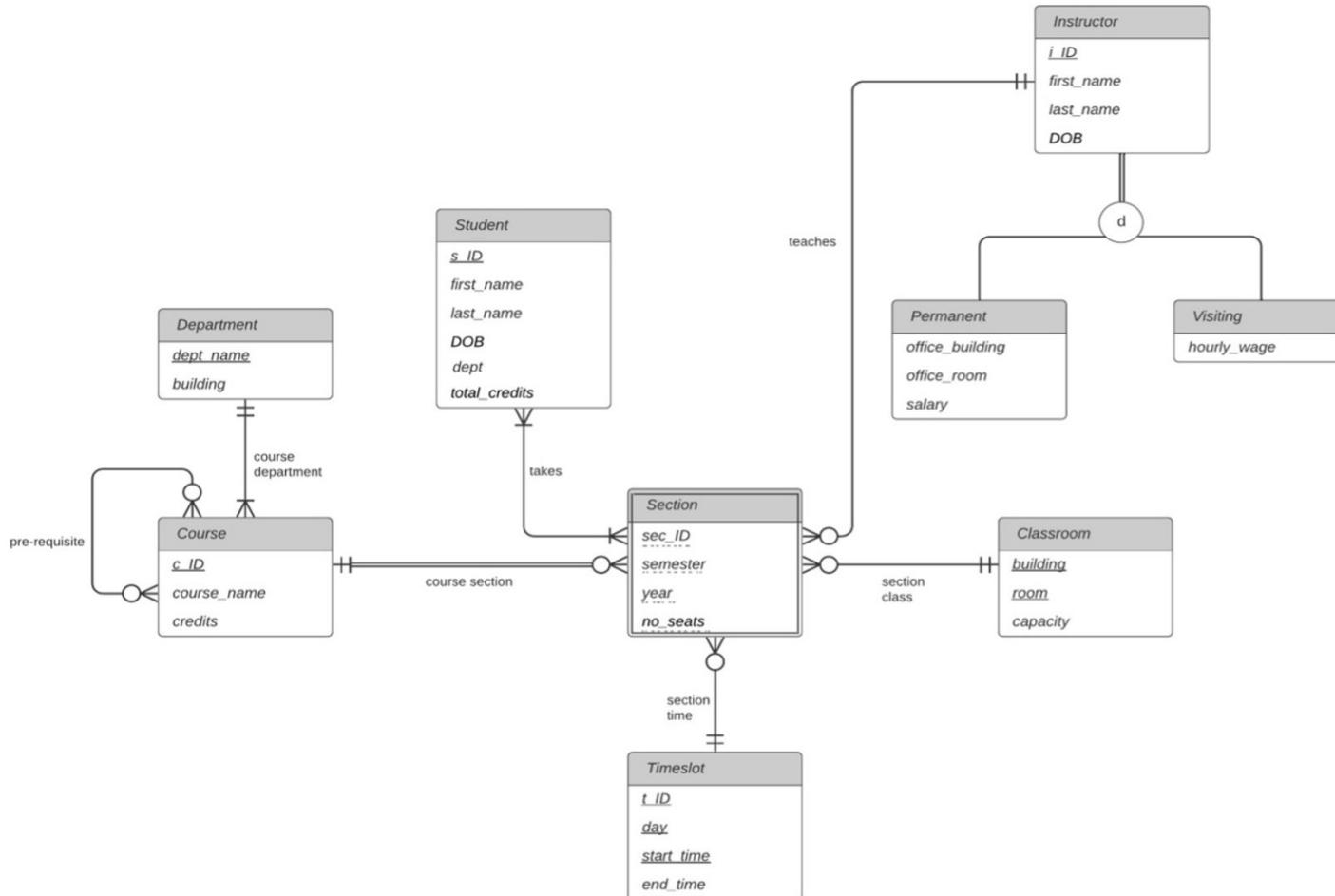
CS425 DATABASE ORGANIZATION PROJECT

Shahrukh Sohail

UNIVERSITY DB DESIGN

Uni Database ER diagram (crow's foot)

Shahrukh Sohail | March 9, 2022



PYTHON: WEB APPLICATION

- FLASK #Python library
- Psycopg2 #to connect to PostgreSQL DB
- Html files #web-based user interface

PYTHON FLASK: WEB APPLICATION

How to Run

1. Place all files in one folder (**static & templates** folder + **app.py**)
2. Open folder in Visual Studio Code (VS Code)
3. Run app.py
4. Open <http://127.0.0.1:8080/> in a web browser
5. In browser settings disable cache (**Recommended**)

PYTHON FLASK: WEB APPLICATION

Connect to PostgreSQL DB

- Psycopg2 is used to connect to PostgreSQL DB

```
DB_HOST = "localhost"
DB_NAME = "University"
DB_USER = "postgres"
DB_PASS = "postgre"

conn = psycopg2.connect(dbname=DB_NAME, user=DB_USER, password=DB_PASS, host=DB_HOST)
```

PYTHON FLASK: WEB APPLICATION

Search data in Database

- Psycopg2 is used to Query results from PostgreSQL DB

```
cursor = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
cursor.execute("SELECT * FROM Student_T")
employeeelist = cursor.fetchall()
return render_template("StudentOutput.html", value=employeeelist)
```

Search data in Database

1. Run app.py
2. Open <http://127.0.0.1:8080/> in a web browser.
3. Use the options and filters to navigate through the Web Application.

PYTHON FLASK: WEB APPLICATION

Database Project

• UNIVERSITY DATABASE •

Select Table

Student

Department

Instructor

Permanent

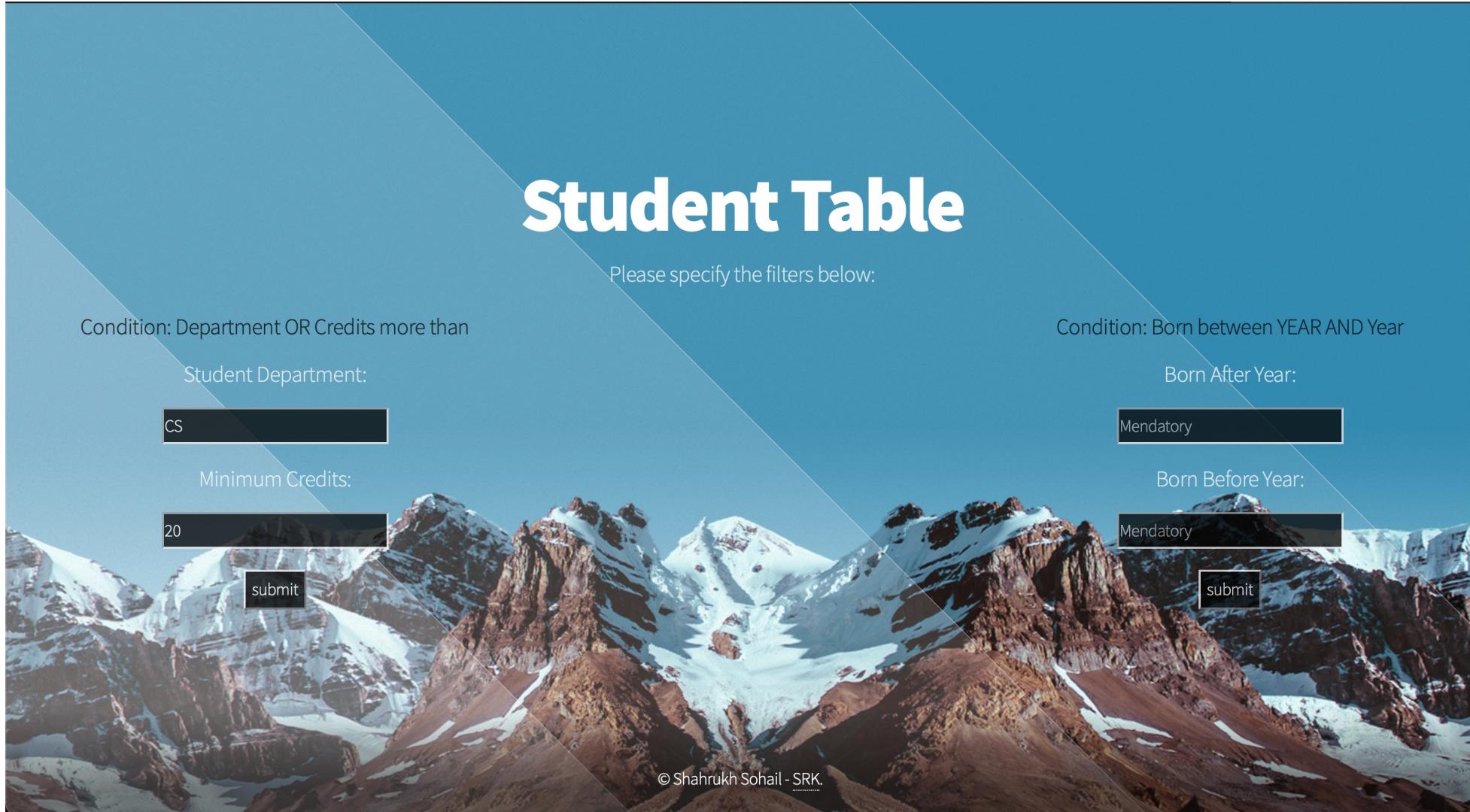
Visiting

Complex

© Shahrukh Sohail - SRK.

PYTHON FLASK: WEB APPLICATION

Student Table



Student Table

Please specify the filters below:

Condition: Department OR Credits more than

Student Department:

CS

Minimum Credits:

20

submit

Condition: Born between YEAR AND Year

Born After Year:

Mandatory

Born Before Year:

Mandatory

submit

© Shahrukh Sohail - SRK.

PYTHON FLASK: WEB APPLICATION

Student Table

Student Data Search Result

Student ID, First Name, Department and Total Credits of Students in CS Department OR Total Credits less than 20.

Student ID	First Name	Department	Total Credits
111	Shahrukh	CS	15
555	Steven	CS	7
777	Victor	Physics	33
105	Oliver	Business	28
103	Walter	Music	25
101	Amy	CS	18

PYTHON FLASK: WEB APPLICATION

Student Table

QUERY:

Find Student ID, First Name, Department and Total Credits of Students in CS department OR total credits less than 20.

```
7  SELECT S_ID, first_name, dept, total_credits  
8  FROM Student_T  
9  WHERE dept = 'CS' OR total_credits > 20
```

Data Output Explain Messages Notifications					
	s_id [PK] character (3)	first_name character varying (20)	dept character varying (15)	total_credits integer	
1	111	Shahrukh	CS	15	
2	555	Steven	CS	7	
3	777	Victor	Physics	33	
4	105	Oliver	Business	28	
5	103	Walter	Music	25	
6	101	Amy	CS	18	

PYTHON FLASK: WEB APPLICATION

Student Table

QUERY:

Find Student ID, First Name, Department and Total Credits of Students in department in `input1` OR total credits less than `input2`.

```
@app.route("/studentquery1",methods=["POST","GET"])
def Squery1():
    try:
        if request.method == "POST":
            cursor = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
            user = request.form['depname']
            user2 = request.form['credits']
            title2 = 'Student ID, First Name, Department and Total Credits of Students in {} Department OR Total Credits less than {}'.format(user, user2)

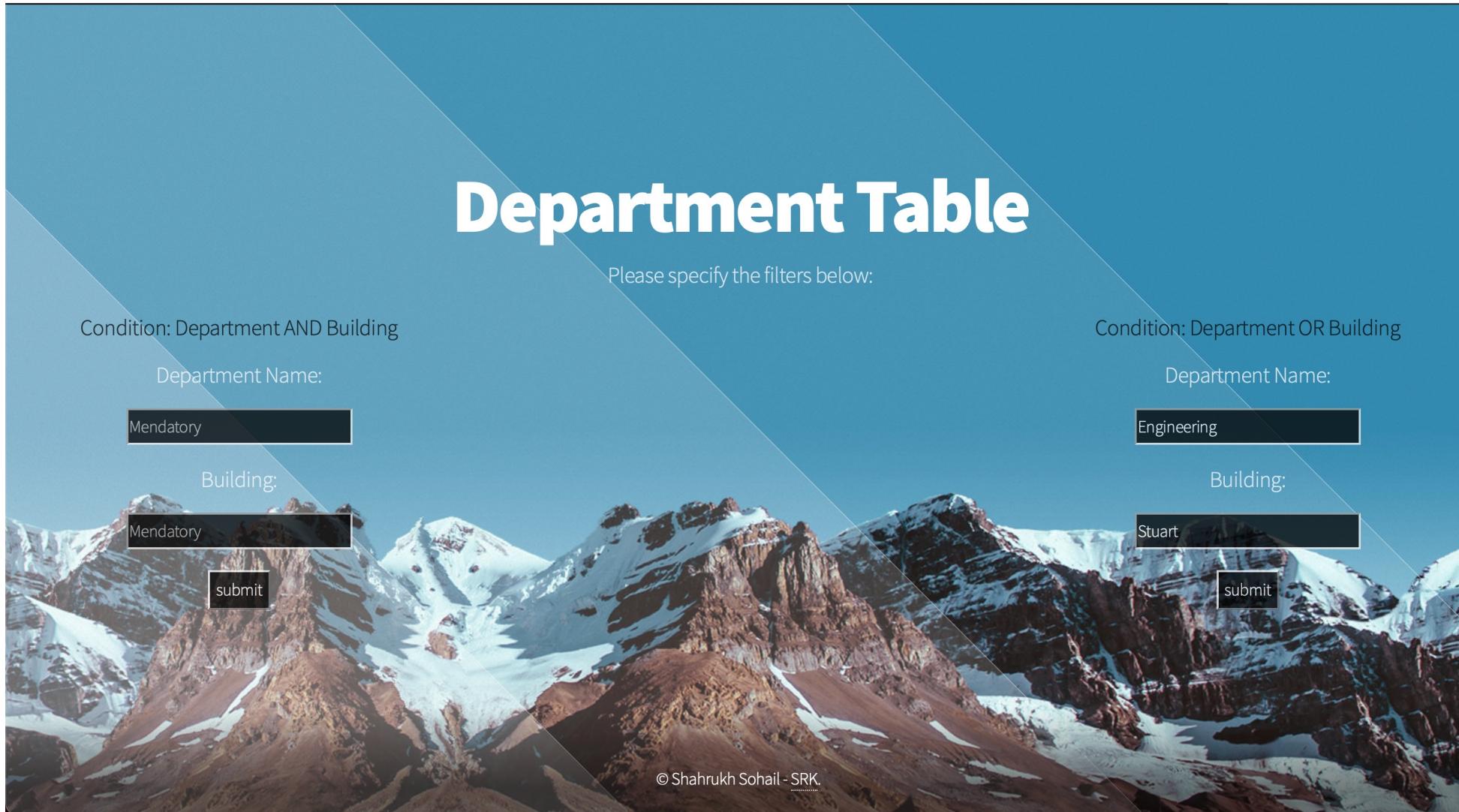
            cursor.execute("""SELECT s_ID,first_name,dept,total_credits FROM Student_T
                WHERE dept = %(value1)s OR total_credits > %(value2)s""", {"value1": user, "value2": user2})

            employeelist = cursor.fetchall()
            return render_template("StudentOutput.html", value=employeelist, title2=title2, column3='Department', column4='Total Credits')

    except Exception as e:
        print(e)
    finally:
        cursor.close()
```

PYTHON FLASK: WEB APPLICATION

Department Table



Department Table

Please specify the filters below:

Condition: Department AND Building

Department Name:

Building:

Condition: Department OR Building

Department Name:

Building:

© Shahrukh Sohail - SRK.

PYTHON FLASK: WEB APPLICATION

Department Table

Department Data Search Result

Department and Building name of Engineering department OR Stuart building.

Department

CS

Engineering

Business

Building

Stuart

Rettaliata

Stuart

PYTHON FLASK: WEB APPLICATION

Department Table

QUERY:

Find Department and Building name of Engineering department OR Stuart building.

```
7  SELECT *
8  FROM Department_T
9  WHERE dept_name = 'Engineering' OR building ='Stuart'
10
```

Data Output Explain Messages Notifications

	dept_name [PK] character varying (15)	building character varying (20)	
1	CS	Stuart	
2	Engineering	Rettaliata	
3	Business	Stuart	

PYTHON FLASK: WEB APPLICATION

Department Table

QUERY:

Find Department and Building name where department is input1 OR building is input2.

```
@app.route("/departmentquery1",methods=["POST","GET"])
def Dquery1():
    try:
        if request.method == "POST":
            cursor = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
            user = request.form['depname']
            user2 = request.form['buildname']
            title2 = 'Department and Building name of {} department AND {} building.'.format(user, user2)

            cursor.execute("""SELECT * FROM Department_T WHERE dept_name = %(value1)s AND building = %(value2)s""", {"value1": user, "value2": user2})

            employeelist = cursor.fetchall()
            return render_template("DepartmentOutput.html",title2=title2, value=employeelist)

    except Exception as e:
        print(e)
```

PYTHON FLASK: WEB APPLICATION

Instructor Table

Instructor Table

Please specify the filters below:

Condition: Last Name OR Born Before Year

Last Name:

Einstein

Born Before Year:

1955

submit

Condition: Born between YEAR AND Year

Born After Year:

Mandatory

Born Before Year:

Mandatory

submit

© Shahrukh Sohail - SRK.

PYTHON FLASK: WEB APPLICATION

Instructor Table

Instructor Data Search Result

Instructor ID, First Name, Last Name and Date of Birth of Instructors with Last name Einstein OR born before year 1955.

Instructor ID	First Name	Last Name	Date of Birth
123	Issac	Newton	1951-08-09
456	Albert	Einstein	1958-12-17
286	Michael	Faraday	1953-12-07

PYTHON FLASK: WEB APPLICATION

Instructor Table

QUERY:

Find Instructors where last name is input1 OR born before year input2.

```
@app.route("/instructorquery1",methods=["POST","GET"])
def Iquery1():
    try:
        if request.method == "POST":
            cursor = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
            user = request.form['lname']
            user2 = request.form['dob']
            title2 = 'Instructor ID, First Name, Last Name and Date of Birth of Instructors with Last name {} OR born before year {}'.format(user, user2)

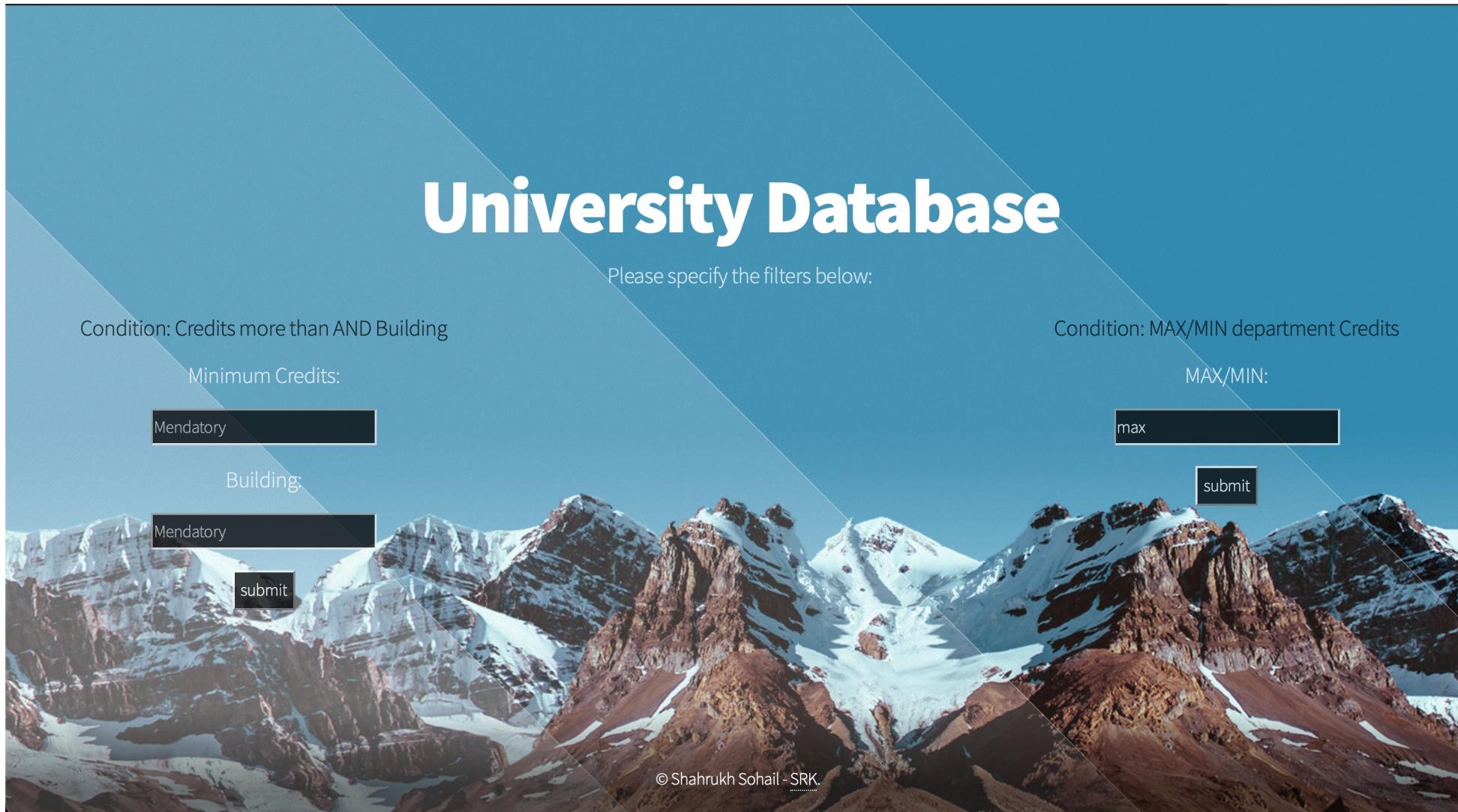
            cursor.execute("""SELECT * FROM Instructor_T WHERE last_name = %(value1)s OR
EXTRACT(YEAR FROM DOB) < %(value2)s""", {"value1": user, "value2": user2})

            employeeelist = cursor.fetchall()
            return render_template("InstructorOutput.html", title2=title2, value=employeeelist)

    except Exception as e:
        print(e)
```

PYTHON FLASK: WEB APPLICATION

Complex Queries



PYTHON FLASK: WEB APPLICATION

Complex Queries

University Database Search Result

Student ID, First Name, Department and Total Credits of Students with MAX credits in each department

Student ID	First Name	Department	Credits
222	Peter	Biology	9
777	Victor	Physics	33
999	Clark	Science	10
109	Bruce	Engineering	8
105	Oliver	Business	28
103	Walter	Music	25
101	Amy	CS	18

PYTHON FLASK: WEB APPLICATION

Complex Queries

QUERY:

Find Student ID, First Name, Department and Total Credits of Students with maximum credits in each department.

```
12  SELECT s_ID, first_name, dept, total_credits AS max_credit_student
13  FROM Student_T
14  WHERE total_credits IN (SELECT MAX(total_credits)
15          FROM Student_T
16          GROUP BY dept)
17
```

Data Output Explain Messages Notifications

	s_id [PK] character (3)	first_name character varying (20)	dept character varying (15)	max_credit_student integer	
1	222	Peter	Biology	9	
2	777	Victor	Physics	33	
3	999	Clark	Science	10	
4	109	Bruce	Engineering	8	
5	105	Oliver	Business	28	
6	103	Walter	Music	25	
7	101	Amy	CS	18	

PYTHON FLASK: WEB APPLICATION

Complex Queries

QUERY:

Find Student ID, First Name, Department and Total Credits of Students with maximum credits in each department.

```
@app.route("/complexquery2",methods=["POST","GET"])
def Cquery2():
    try:
        if request.method == "POST":
            cursor = conn.cursor(cursor_factory=psycopg2.extras.DictCursor)
            user = request.form['credits']
            if user=='max':
                title2 = 'Student ID, First Name, Department and Total Credits of Students with MAX credits in each department'

                cursor.execute("""SELECT S_ID, first_name, dept, total_credits AS max_credit_students FROM Student_T WHERE total_credits
IN (SELECT MAX(total_credits) FROM Student_T GROUP BY dept)""", {"value1": user})

            employeeelist = cursor.fetchall()
            return render_template("ComplexOutput.html", value=employeeelist, title2=title2, column1 = 'Student ID',
column2 = 'First Name', column3 = 'Department', column4 = 'Credits')

    except Exception as e:
        print(e)
```

PYTHON FLASK: WEB APPLICATION

THE END