DATABASE MANAGEMENT SYSTEM

[UE19CS301]

5 TH SEM, Academic Year 2020-21

ASSIGNMENT – 4

Team 23:

Team Members:

1. Sadhvi Sushravya H S - PES1UG19CS410
2. Sanjana G - PES1UG19CS430
3. Sathvik K - PES1UG19CS435
4. Joshua Dsouza - PES1UG20CS811

OLYMPIC MANAGEMENT SYSTEM

FRONT END :

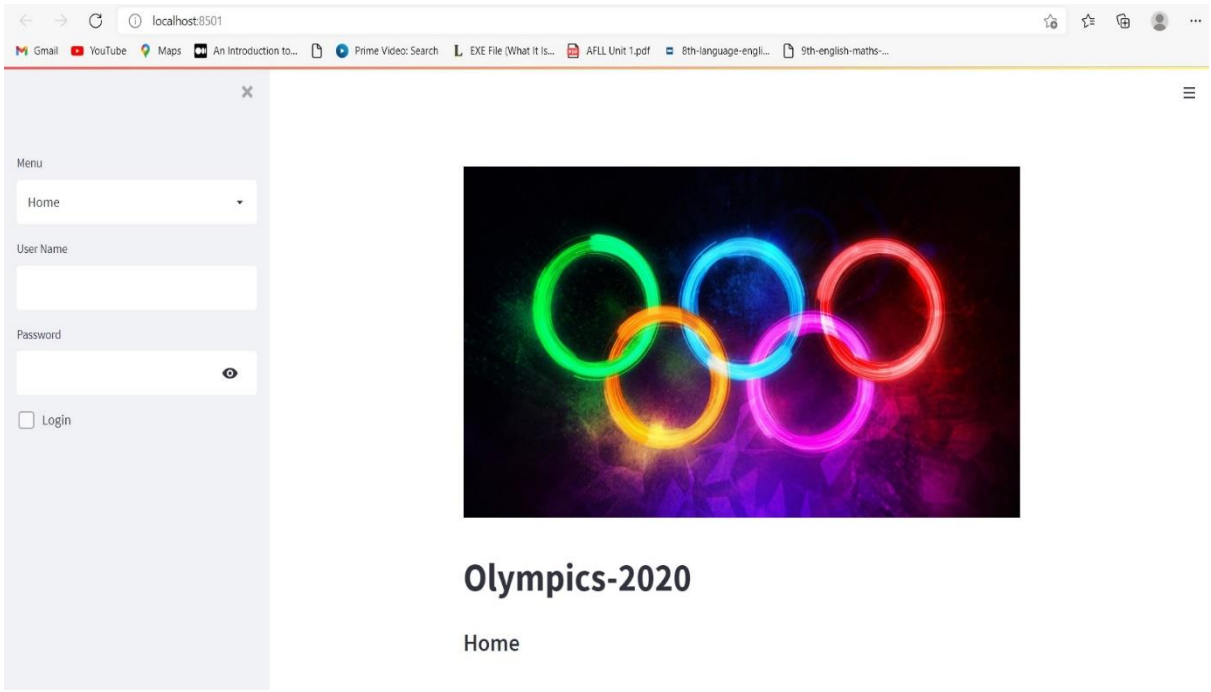
Streamlit (Open – Source Framework)

BACK END :

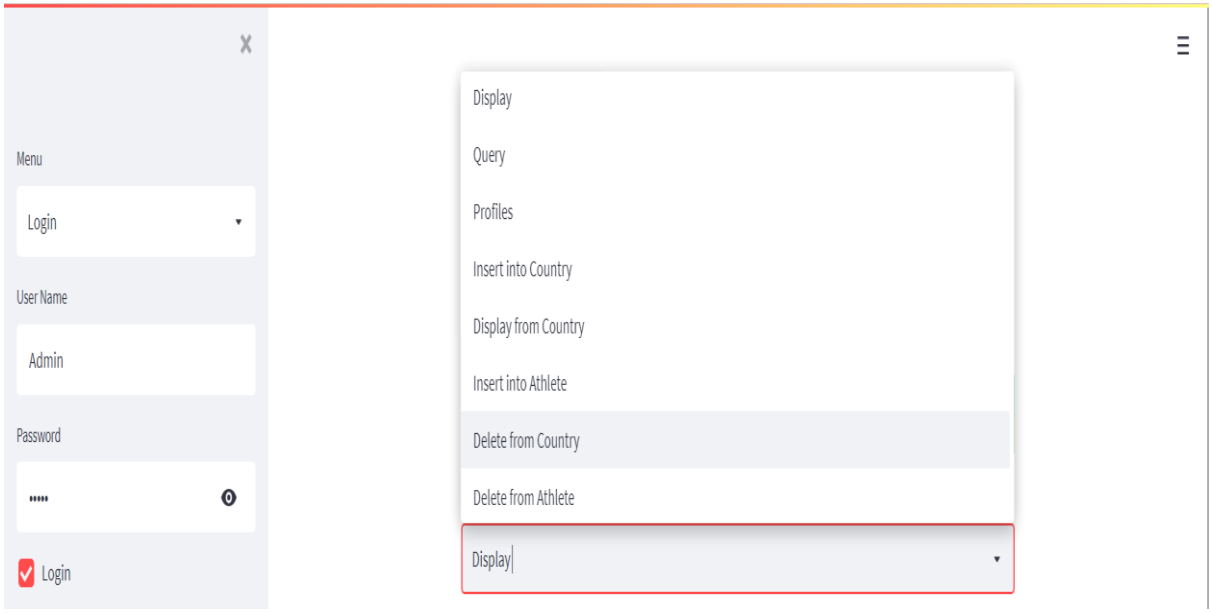
Postgresql Database

Screen – Shots :

The Home Page :

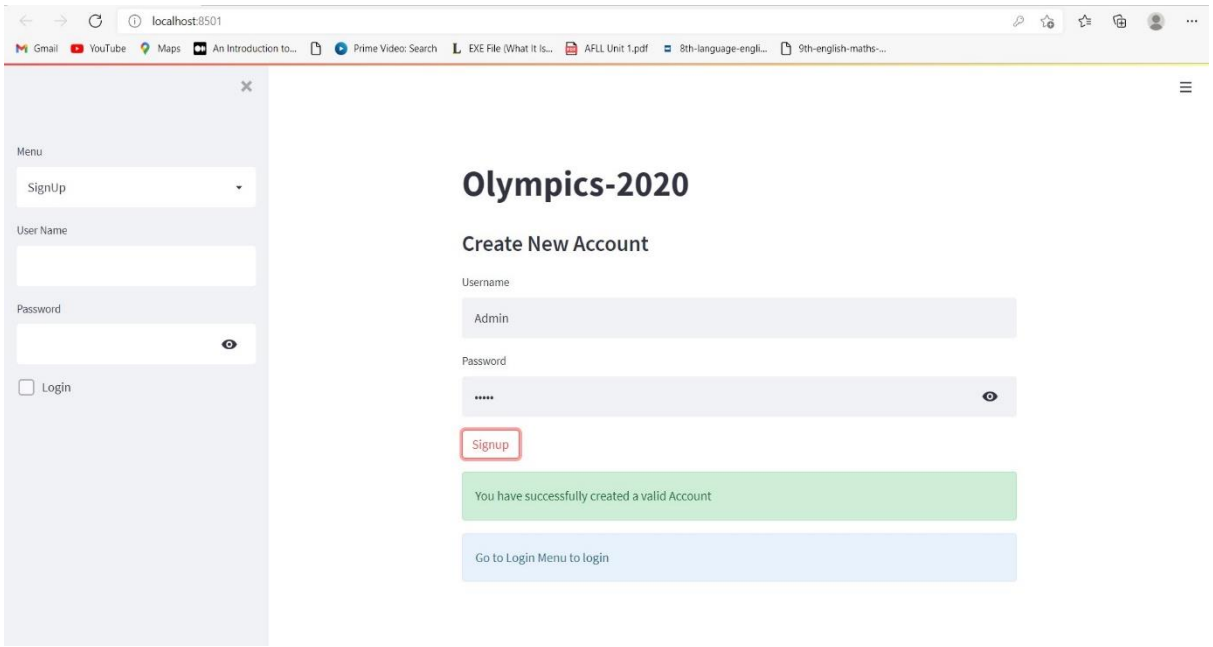


Fields :



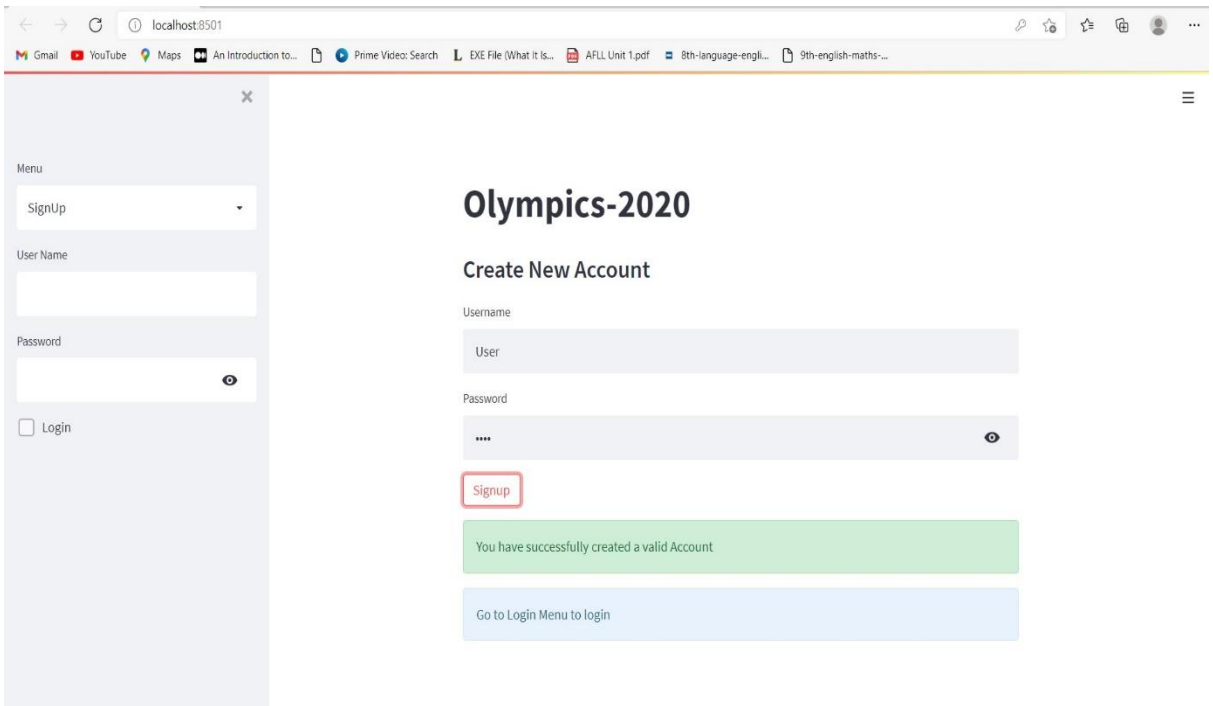
Admin SignUp:

Admin has access to modify the database (delete, update, insert).

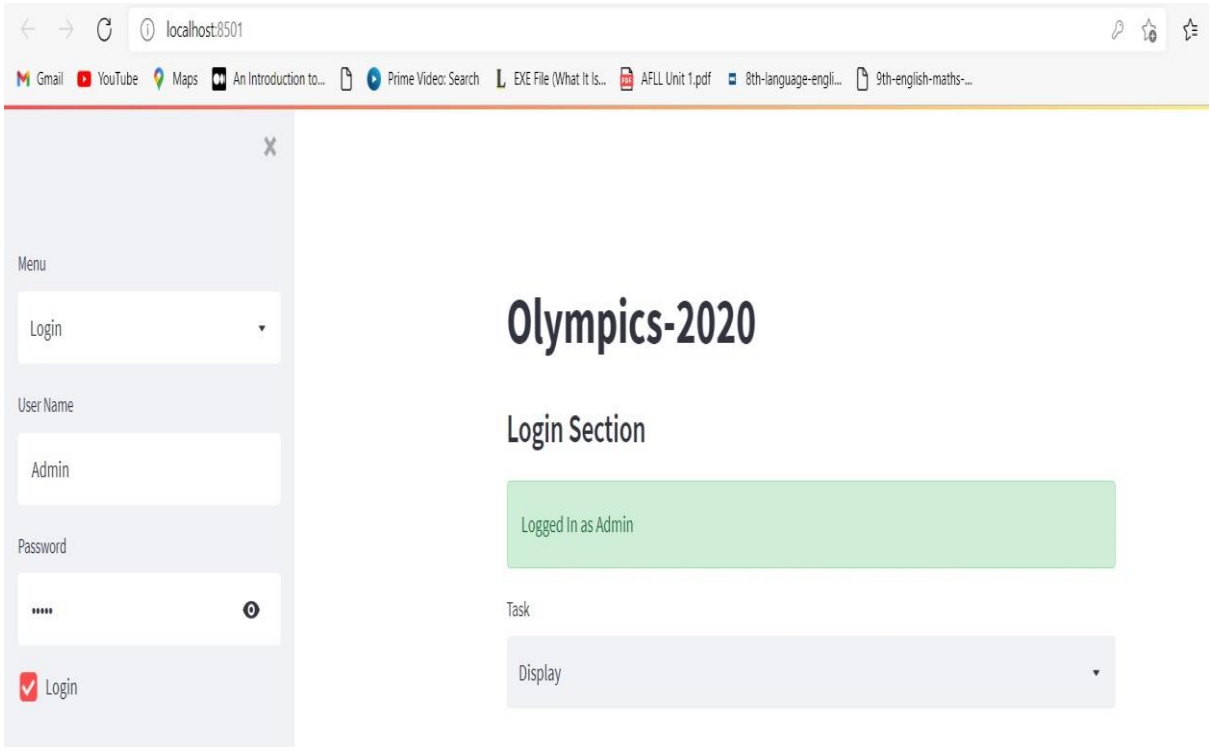


User SignUp:

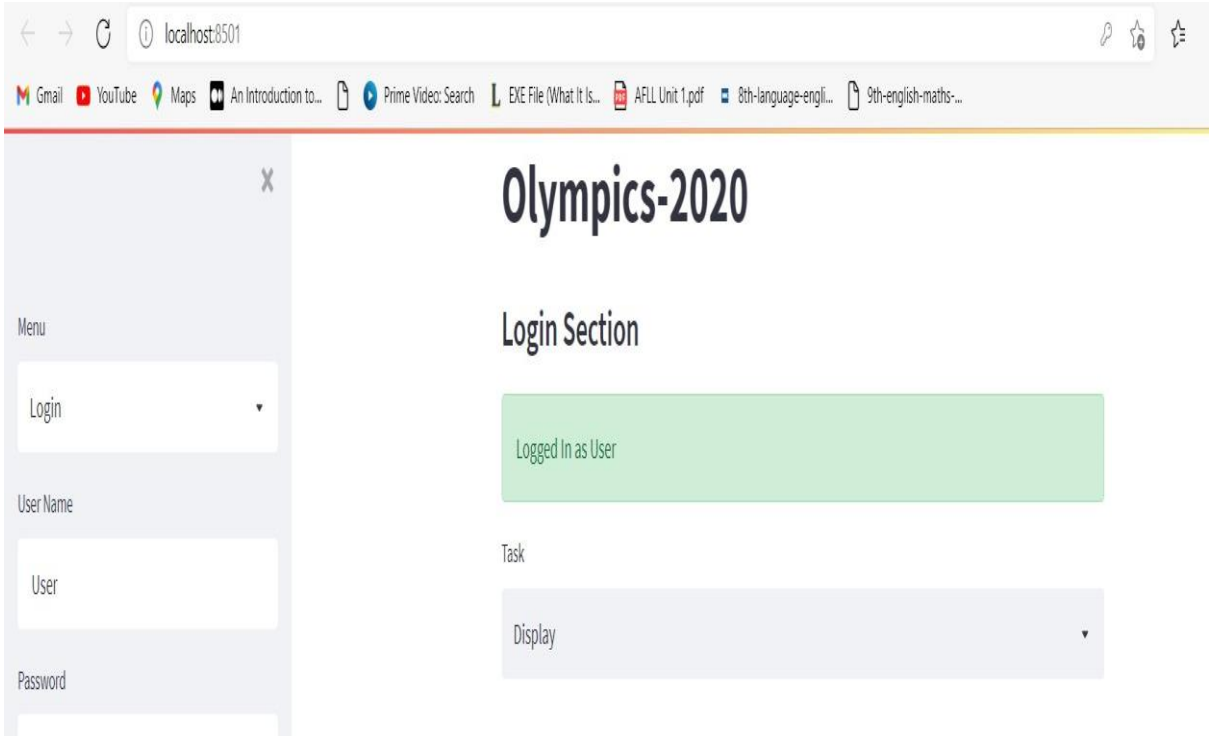
User only has the access to view the changes made by the admin.



Admin Login:

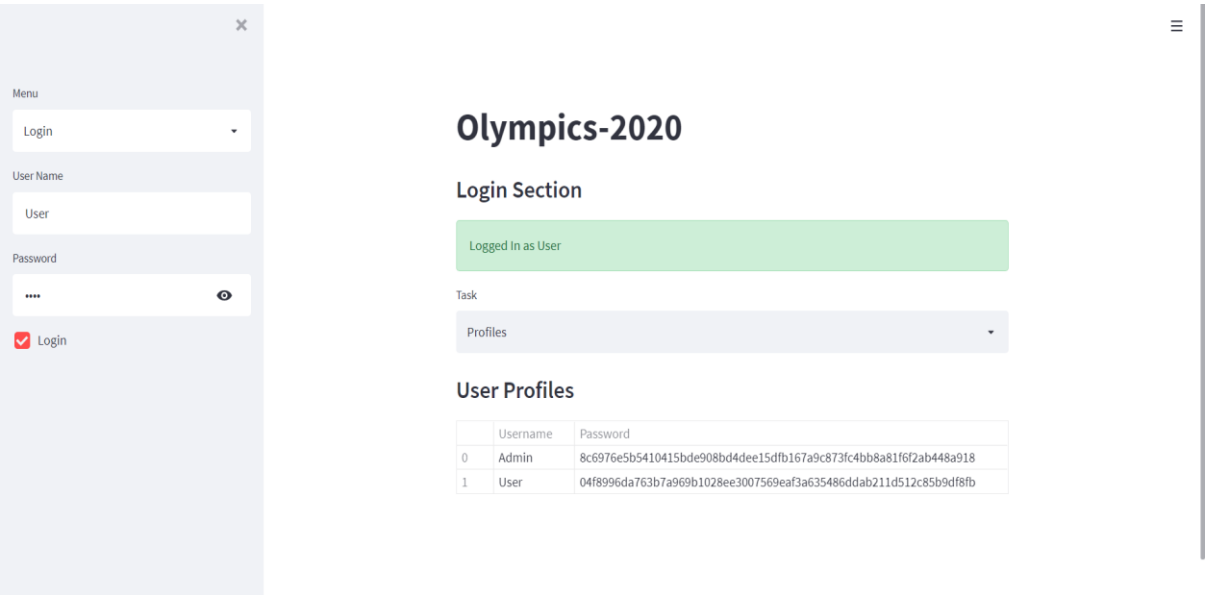


User Login:



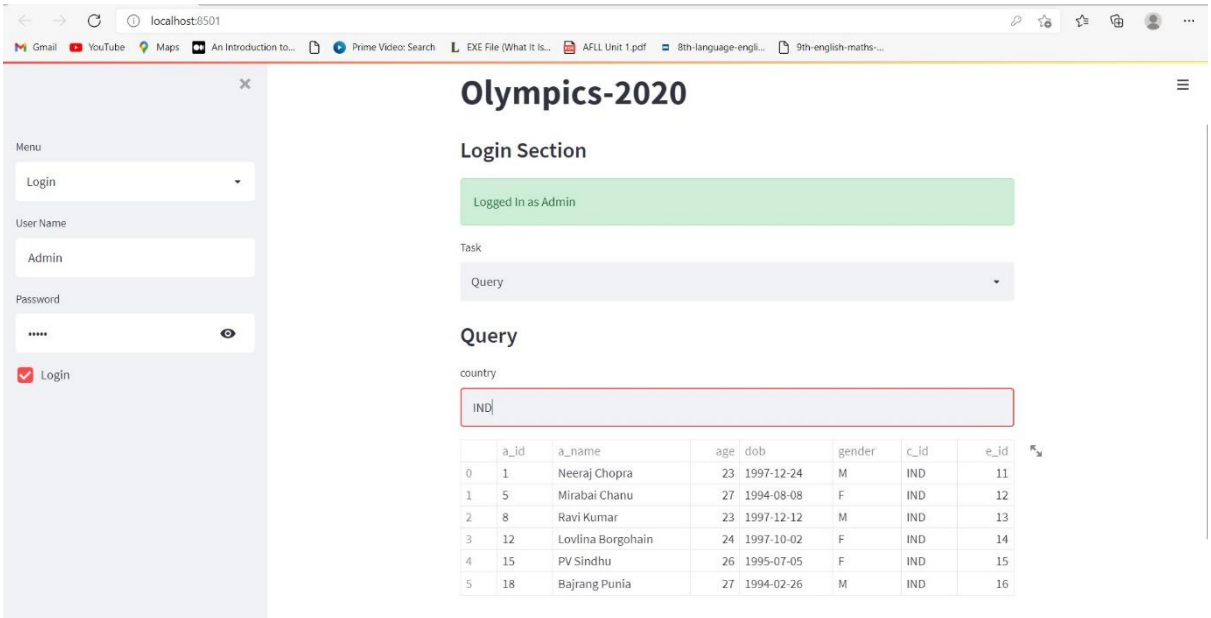
User Profiles:

This section shows the number of users logged in with their respective usernames and encrypted password.



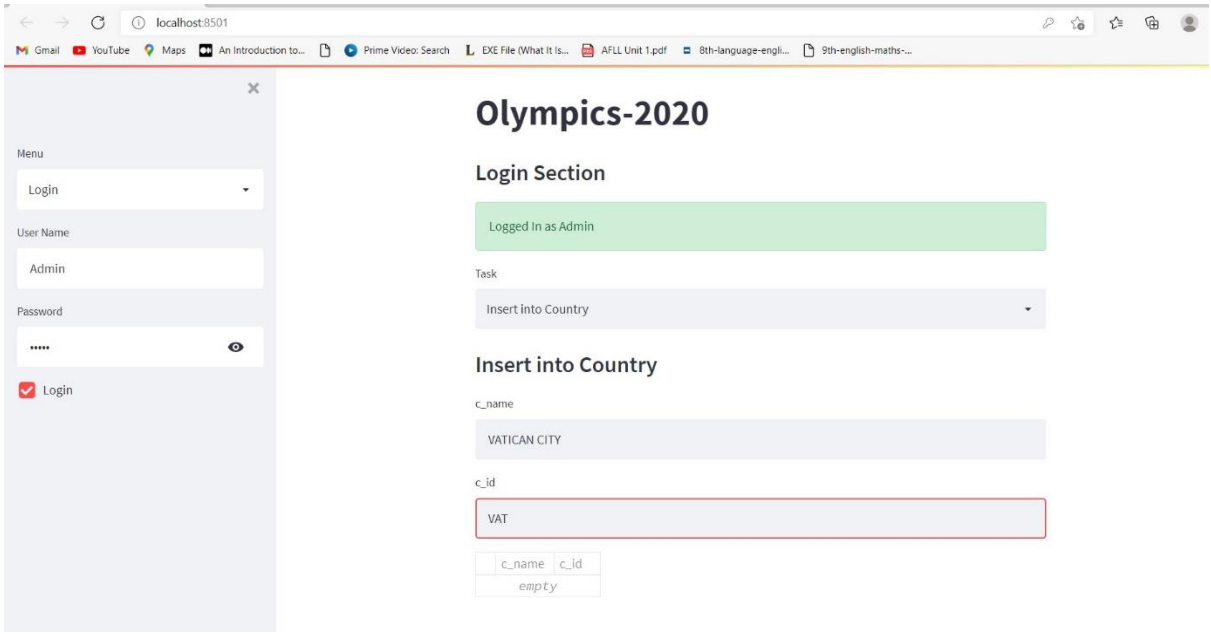
Query on athlete:

This query displays the all the athletes who belong to a particular country based on user input. In our case it displays the athletes belonging to INDIA.



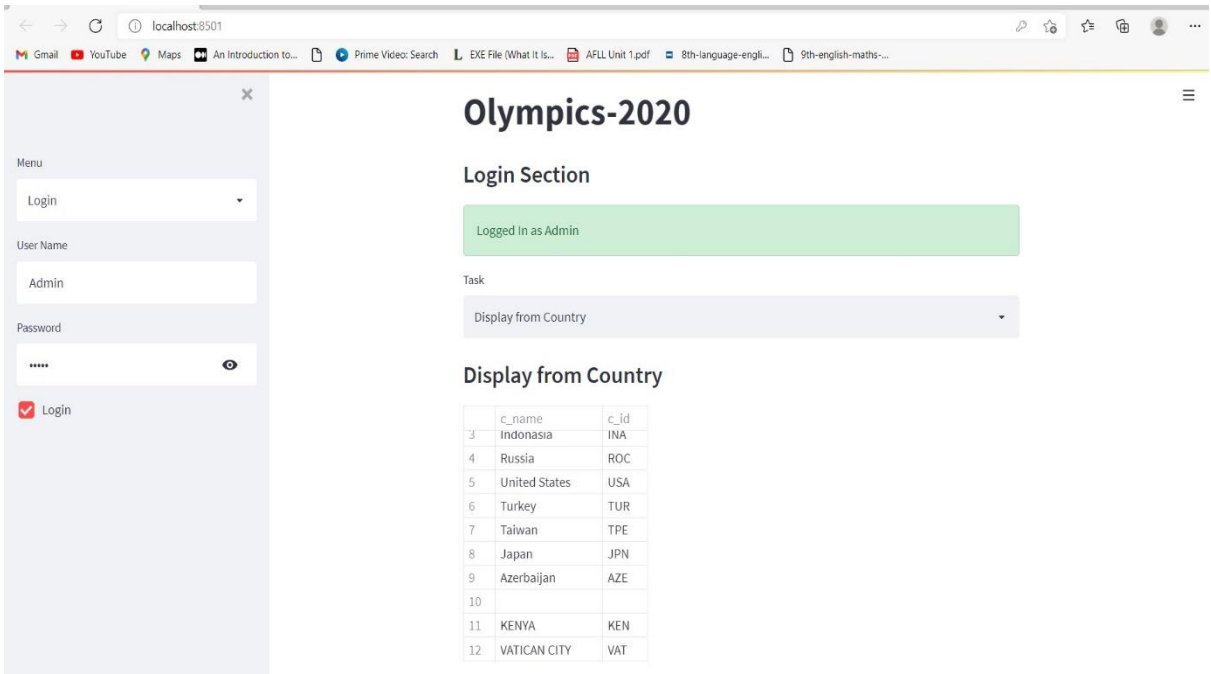
Inserting into Country Table:

It is seen that Country named Vatican City is inserted into the table. It must be noted that this can only be done by Admin and not User.



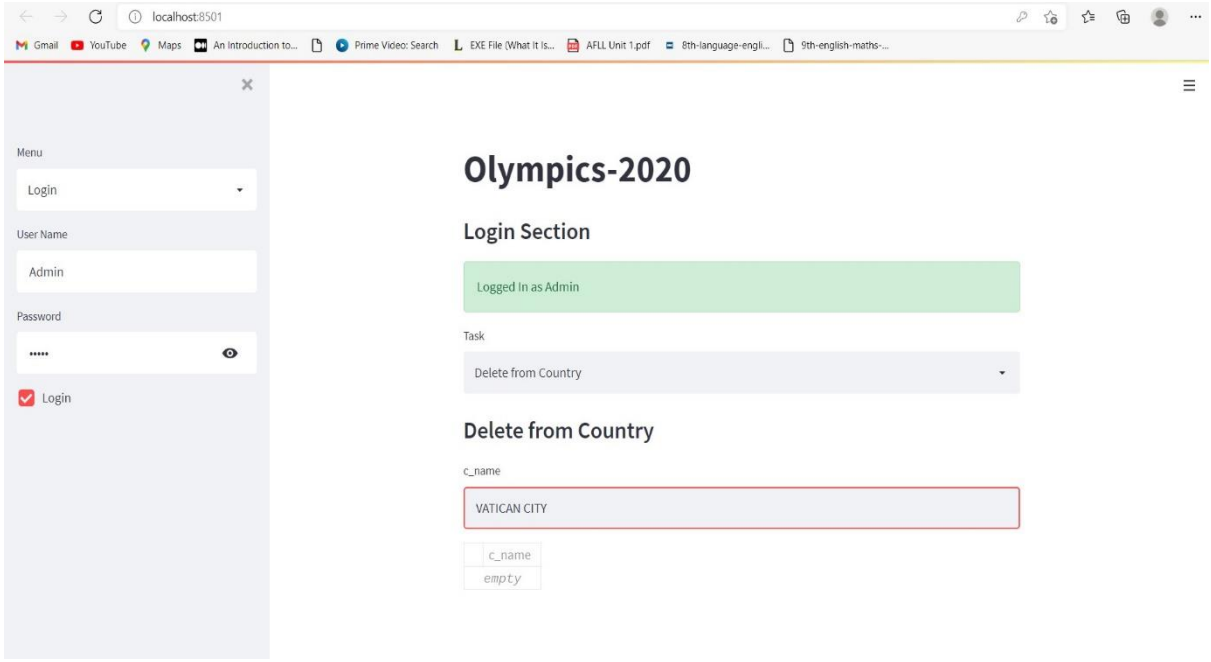
Display from Country Table:

Vatican City is displayed in the Country table.



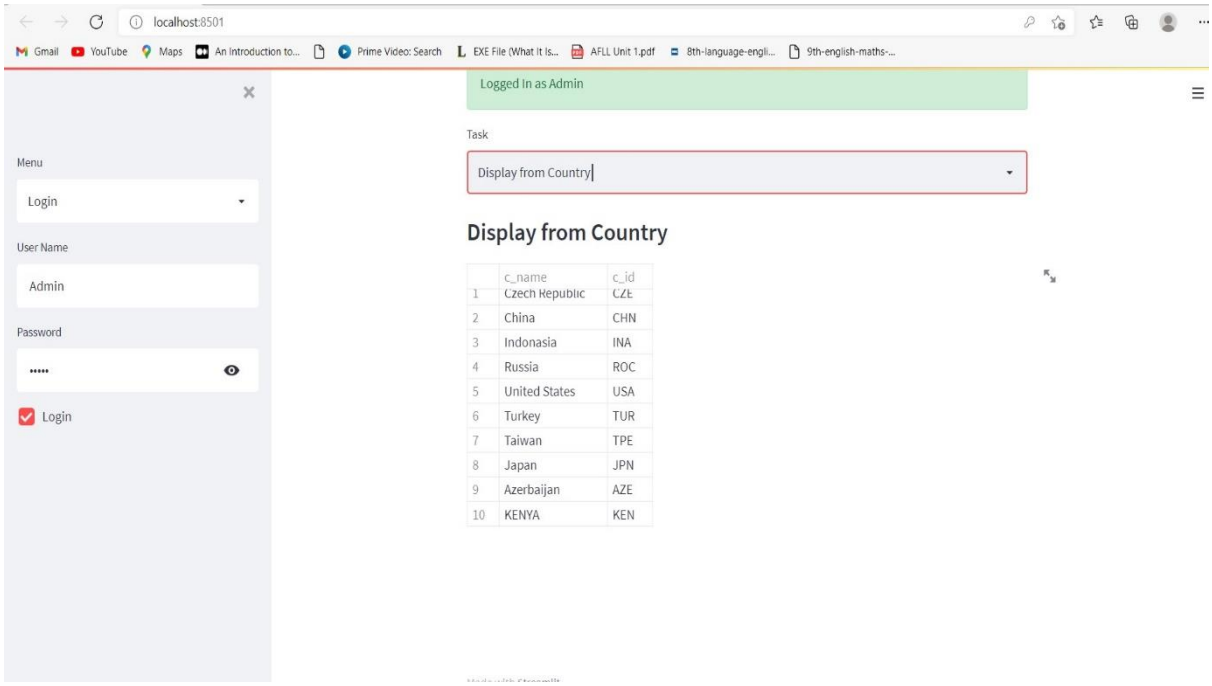
Deleting from Country Table:

The country Vatican City being inserted above is now being deleted from the database.



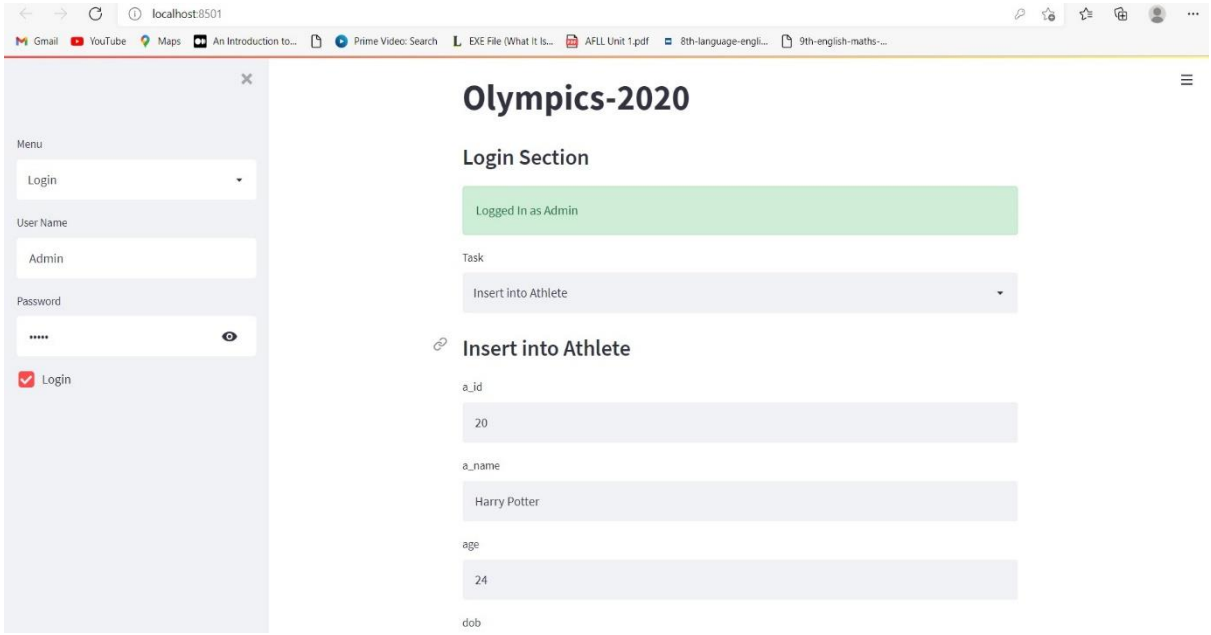
Display from Country Table:

The display field below does not display Vatican City now because it is been deleted as listed above.



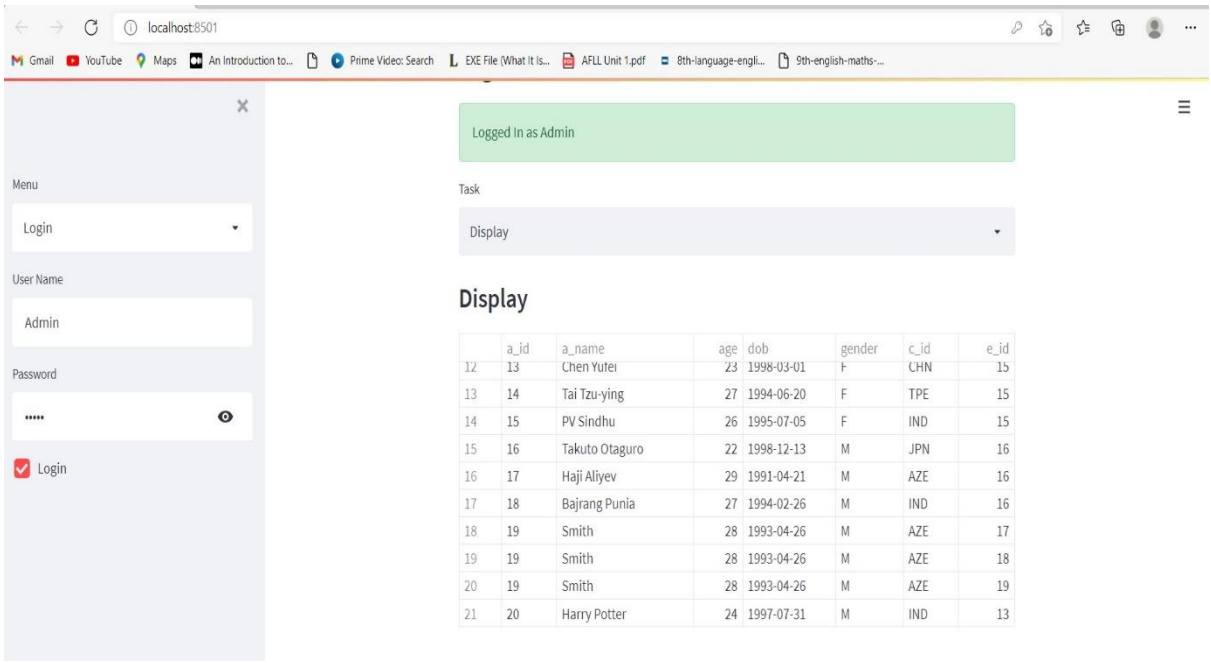
Inserting into Athlete Table:

An athlete named Harry Potter with athlete id 20, age 24, dob 1997-07-31, gender Male, country id IND and event id 13 is inserted into the Athlete table.



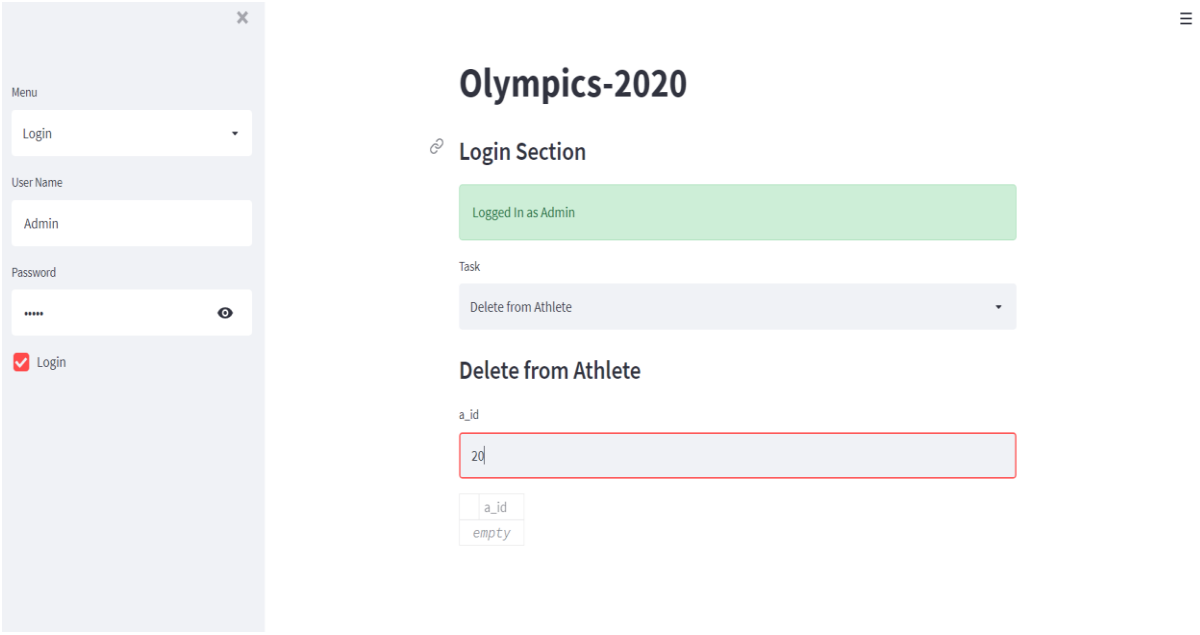
Display from Athlete Table:

Athlete Harry Potter is being displayed.



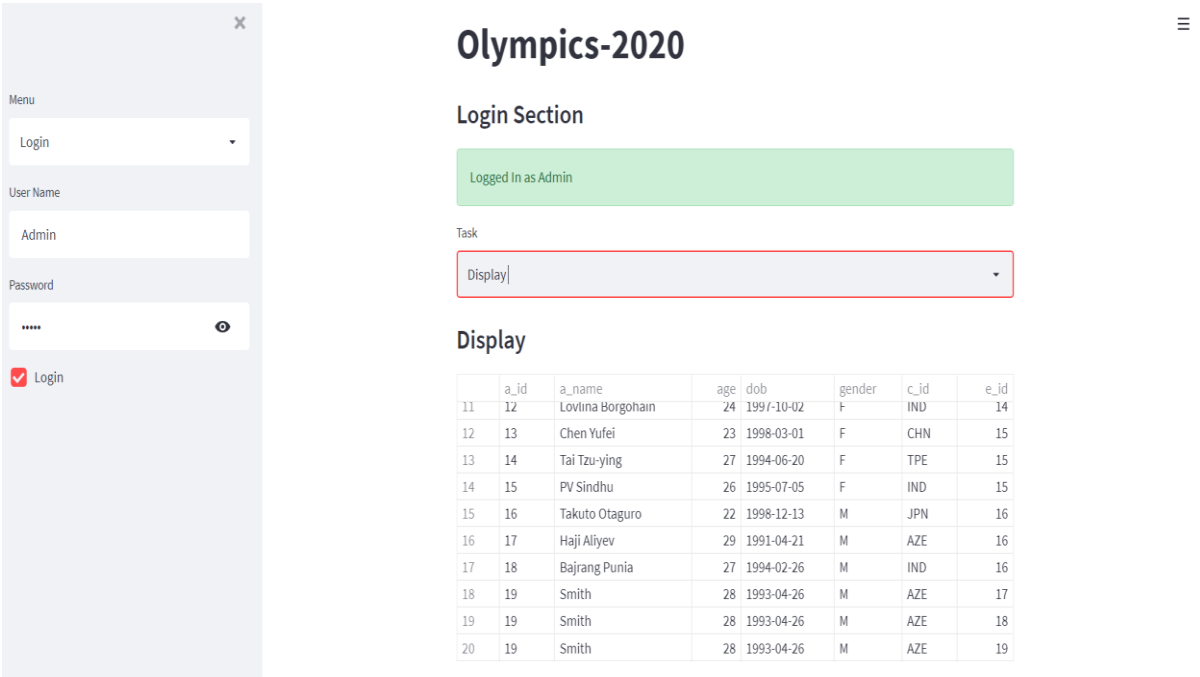
Delete from Athlete Table:

Athlete named Harry Potter has now been deleted from the database .



Display from Athlete Table:

Athlete Harry Potter is now not displayed in the Display table below because its been deleted.



	a_id	a_name	age	dob	gender	c_id	e_id
11	12	Lovlina Borgohain	24	1997-10-02	F	IND	14
12	13	Chen Yufei	23	1998-03-01	F	CHN	15
13	14	Tai Tzu-ying	27	1994-06-20	F	TPE	15
14	15	PV Sindhu	26	1995-07-05	F	IND	15
15	16	Takuto Otaguro	22	1998-12-13	M	JPN	16
16	17	Haji Aliyev	29	1991-04-21	M	AZE	16
17	18	Bajrang Punia	27	1994-02-26	M	IND	16
18	19	Smith	28	1993-04-26	M	AZE	17
19	19	Smith	28	1993-04-26	M	AZE	18
20	19	Smith	28	1993-04-26	M	AZE	19

Write up about the changes in Business/Application changes/expansion - that might lead to:

- **Schema Changes:**

In the scenario of the entry of a new player into the Olympics or even exit of a player due to injury, we can expect changes in the schema.

Under some drastic circumstances, like exit of a player due to illegal accusitions, we can expect huge changes in the schema for the entire database as requirements with respect to how data consistency and maintenance needs to be done can change drastically.

- **Constraint changes:**

We can expect to see changes in constraints per table in the scenario that a new entity/table is introduced. In such a case, we would require existing tables to reference the primary key of the new table and vice versa introducing constraint changes.

- **DBMS migration:**

We can expect DBMS migration from RDBMS to a NoSQL system happening in the scenario where the Olympic Management System would like to expand and want to scale their existing DBMS system. NoSQL gives us the benefit of scalability, ex - as seen with MongoDB and its storing data in documents which is easy to add/delete.

If you have to migrate to any No-SQL variety, then which one will be your choice? Why? Out of the 4 major varieties, you have to pick one and justify your choice. If you can give comparative features across different flavours, that would be great.

The 4 Varieties of NoSQL databases are:

- key-value store,
- document store,
- column-oriented database, and - graph database.

The common databases are MySql, MongoDB, NoSQL and Oracle. The most common features they have are schema flexibility, database performances, relationships and security.

To start with why we could have chosen MongoDB is that document stores in MongoDB are created and stored in BSON files which are, in fact, a little-modified version of JSON files and hence all JS are supported.

Because of this, it is frequently used for Node.js projects. Moreover, JSON facilitates the exchange of data between web apps and servers in a human-readable format. It offers greater efficiency and reliability which in turn can meet data storage capacity and speed demands.

Real-Time Data Integration

There is a lot of value to data if it is consolidated and aggregated into one single view, and MongoDB plays a vital role in doing that. Due to the query capabilities and flexibility of MongoDB, now it is easier to aggregate data and create tools that will make an organization more efficient.

Contribution:

Commonly done: Report and brainstorming on additional queries

SADHVI SUSHRAVYA H S - PES1UG19CS410 (5 hours) - Worked on Front End, ER Diagram, Relational Table and Report

SANJANA G - PES1UG19CS410 (5 hours) - Worked on Front End, ER Diagram, Relational Table and Report

SATHVIK K - PES1UG19CS435 (5 hours) - Worked on queries, refined the front end and worked on ER Diagram

JOSHUA D'SOUZA – PES1UG20CS811 (5 hours) - Worked on simple and complex queries, refined the front end.

Github Link:

https://github.com/Sanjanagujjar09/DataBaseManagementSystem