Cool things that we did for our project!

1. **Web scraping**
2. **Data Migration to MongoDB**
3. **Data Migration to AWS RDS Database**
4. **Github repo**
5. **Additional Features**

Web scraping

We scraped the **Blood Compatibility Table** from a website and stored it into our MySQL database using python program.

storing_website_data_into_mysql_table.ipynb contains the code.

The program:

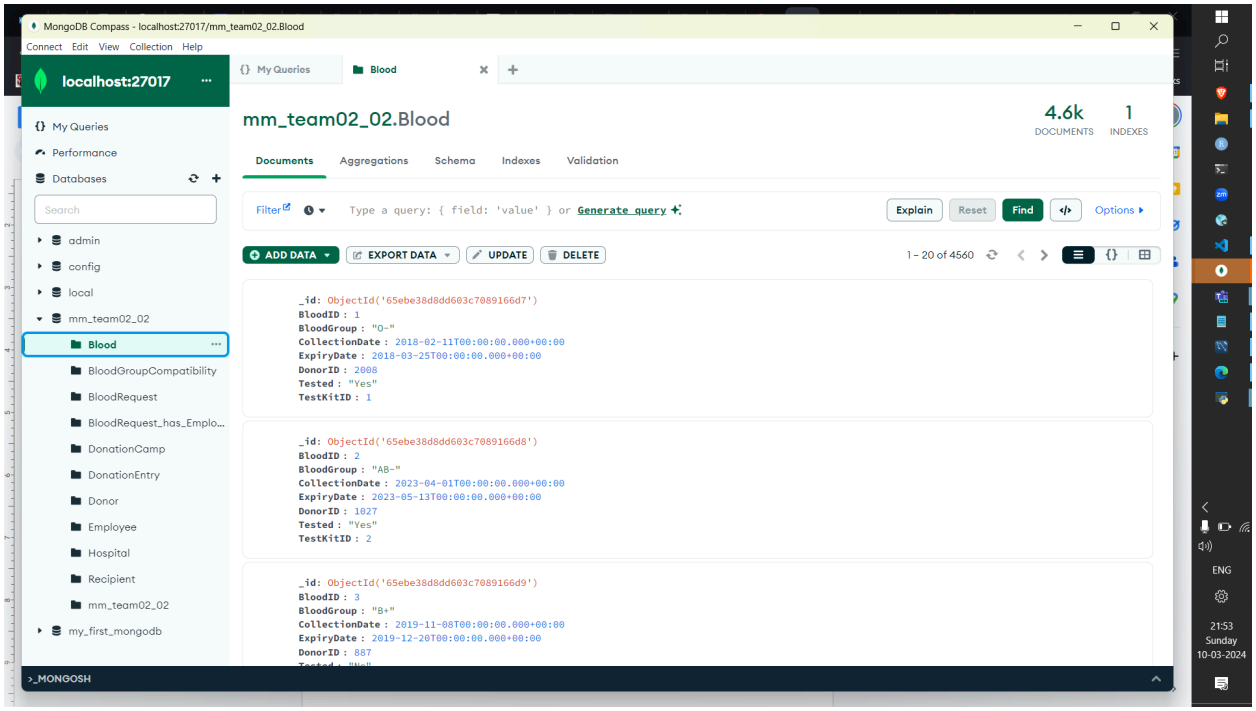
- Scrapes the table from the website and stores it into a csv file.
- Then connects to the database and drops the table if it exists.
- Creates the table and inserts all the rows from the csv file into the table.

Result Grid | Filter Rows: | Export:

Donor Blood Type	Can Donate to	Can Receive From
A+	A+, AB+	A+, A-, O+, O-
A-	A+, A-, AB+, AB-	A-, O-
B+	B+, AB+	B+, B-, O+, O-
B-	B+, B-, AB+, AB-	B-, O-
AB+	AB+	Everyone
AB-	AB+, AB-	AB-, A-, B-, O-
O+	A+, B+, AB+, O+	O+, O-
O-	Everyone	O-

Data Migration to MongoDB

We migrated all the tables from MySQL db to MongoDB using python.



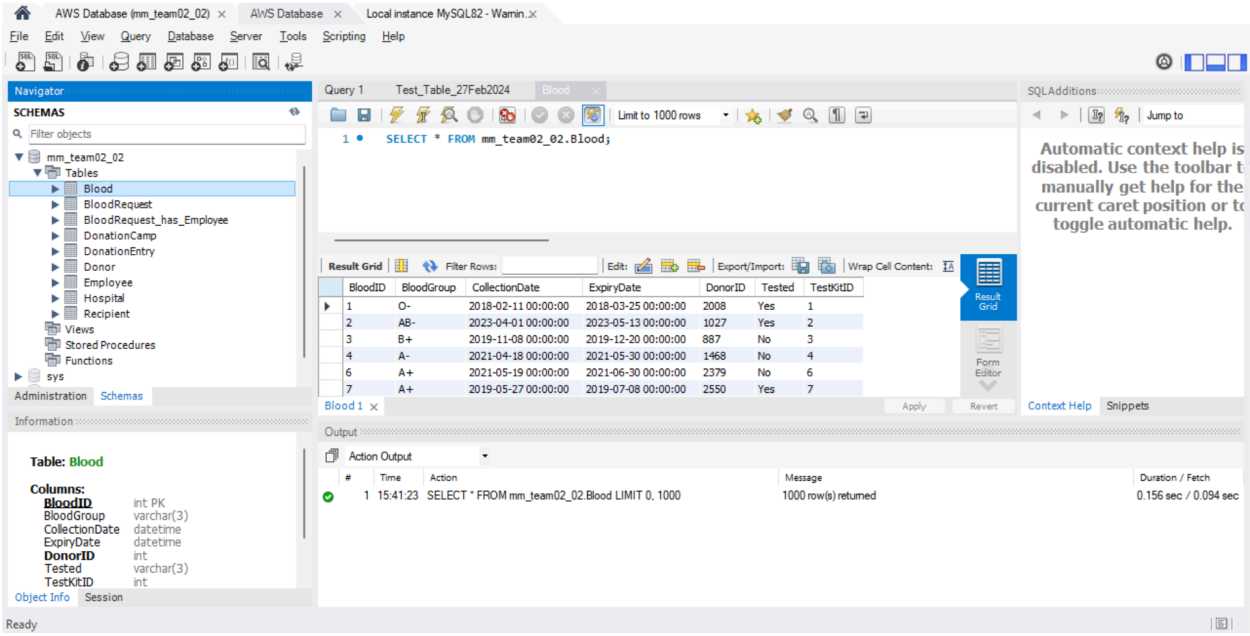
mysql_to_mongodb_migration.py contains the code.

The program:

- Connects to MySQL db and MongoDB localhost.
- Get a list of all tables in the MySQL db.
- Drops the collection in the MongoDB database and creates a new one.
- Iterates over all the tables in the list and converts the data to JSON format.
- Uploads all the tables to MongoDB collection.

Data Migration to AWS RDS Database

We migrated all the data from MySQL db to RDS database.



Steps:

- We connected to the AWS RDS server using the AWS Academy.
- Started the Learner Lab session
- Logged into AWS RDS using the below credentials.
- Migrated data using the dump file through the console.

Credentials to verify:
UserName : admin
Password : Seattle1234
Endpoint : database-1.cd4ick4wgvbm.us-east-1.rds.amazonaws.com
Port : 3306

Github repo

We created a [github repo] for our project which makes it easy to run the application by following the steps below.

1. Clone the repository to your local machine using

   `git clone https://github.com/Ruqhaiya/Blood-Bank-Management-System.git'

2. Navigate to the cloned directory and install the required dependencies by running below command in your terminal.

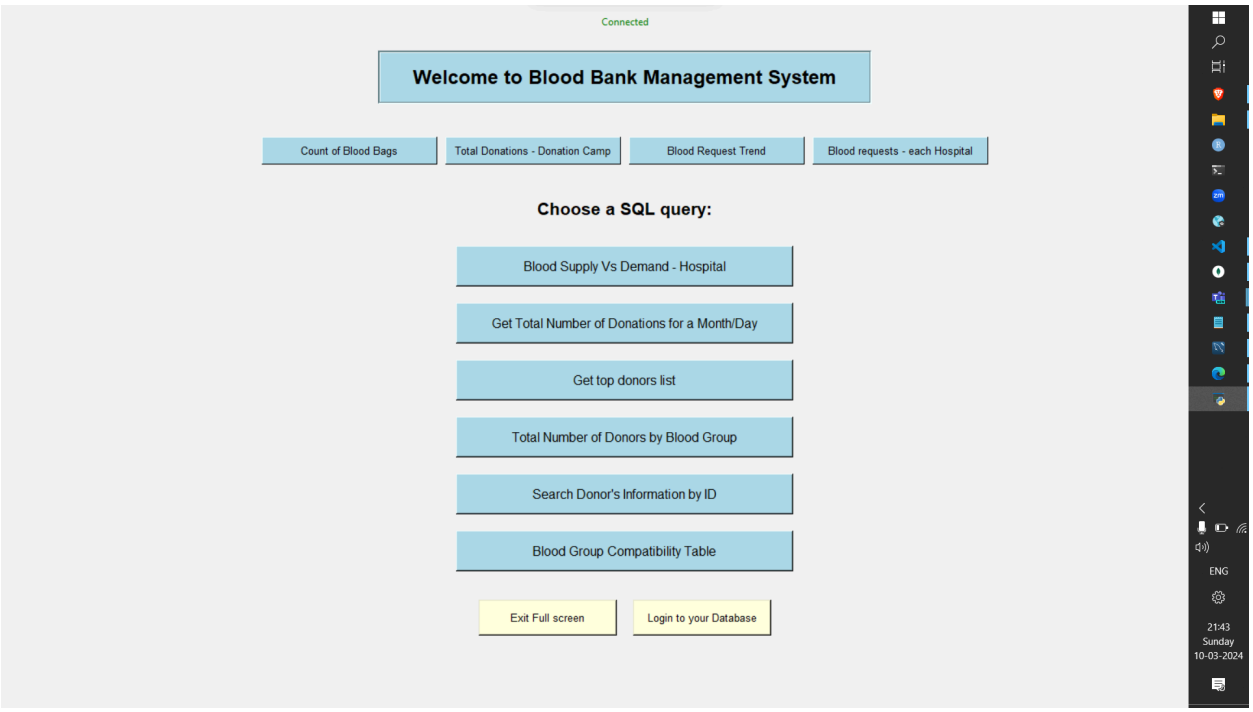
   pip install -r requirements.txt

3. Start the application by running **python bbms.py** or by running the code in jupyter notebook.

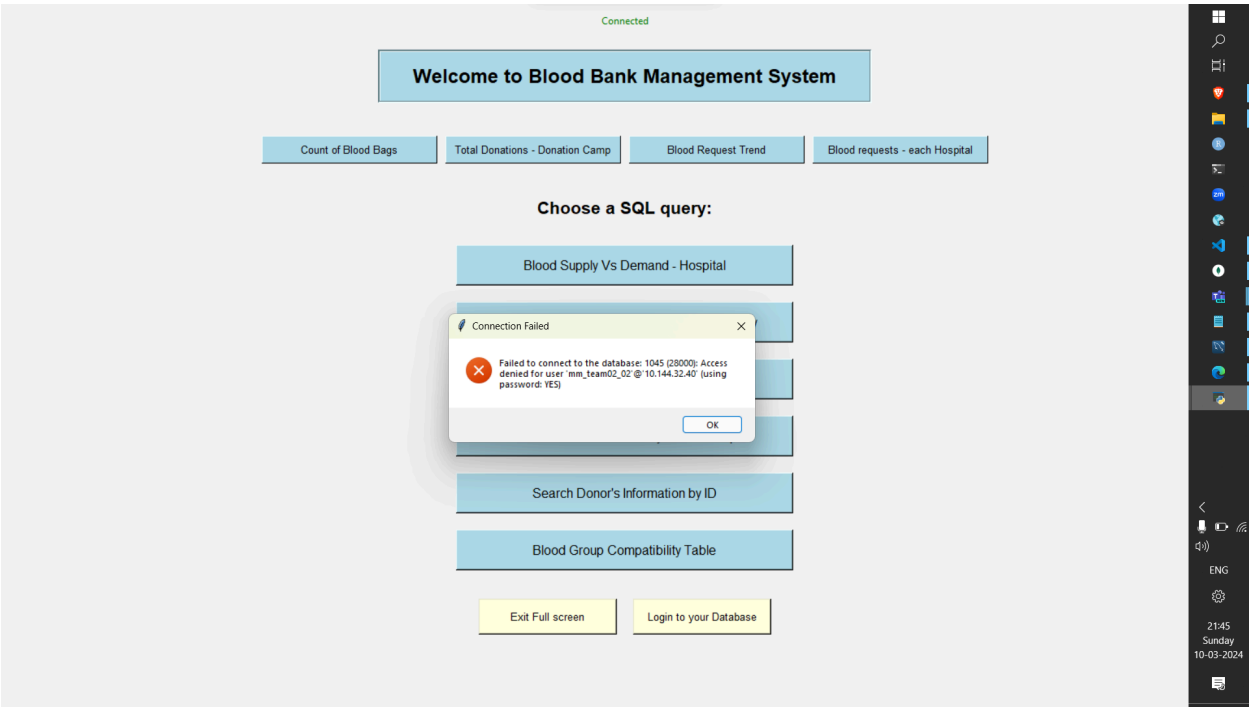
Additional Features

1. Python

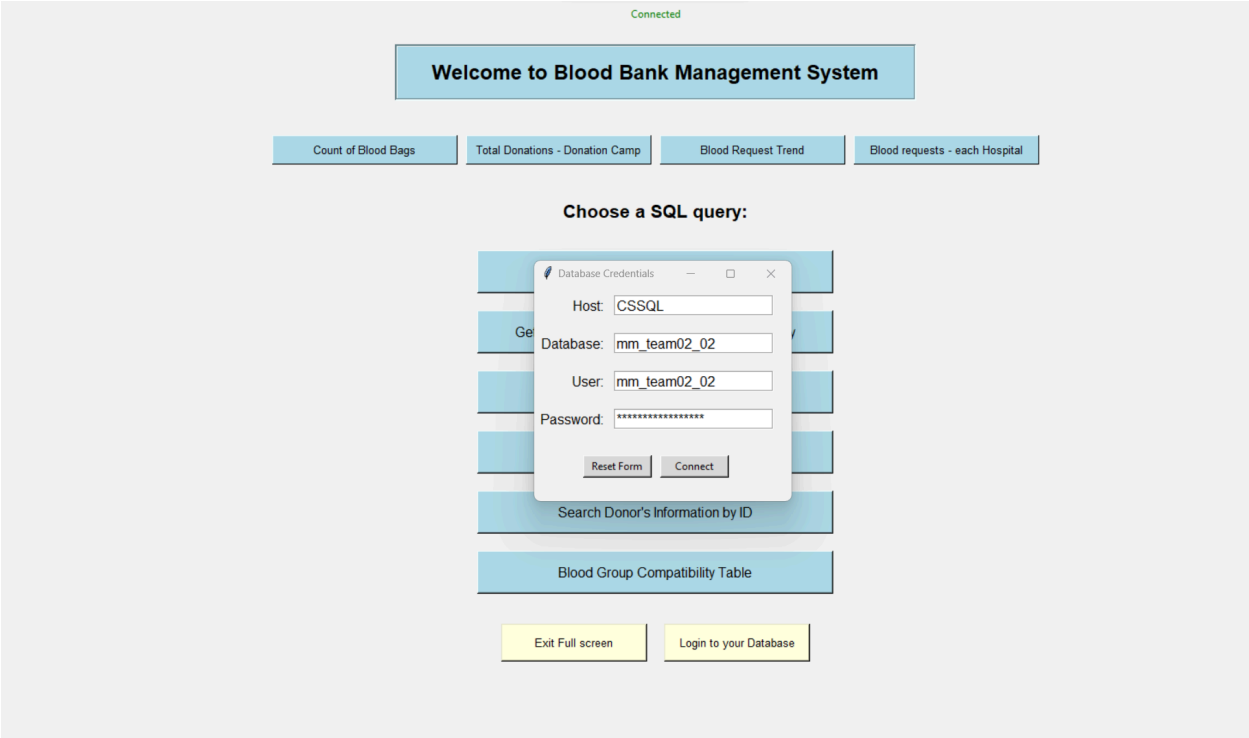
- Object Oriented code
- Seamless UI design using Tkinter GUI



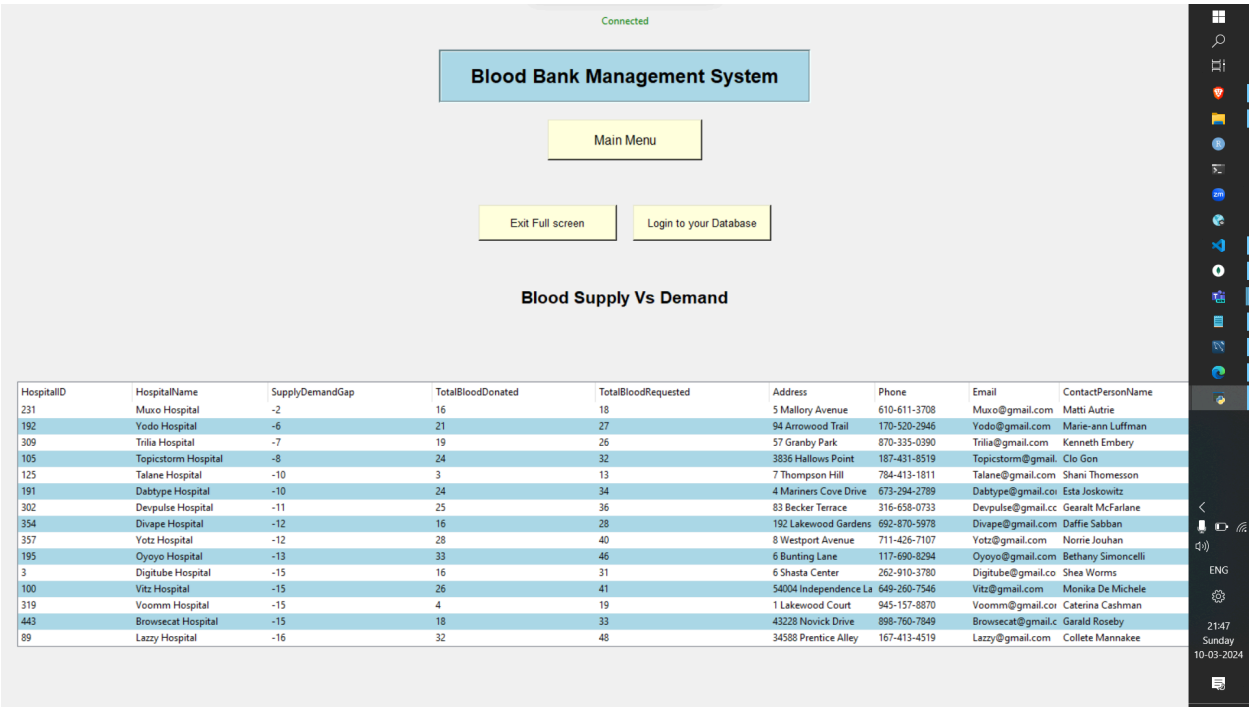
- Error Handling for database connectivity and running queries



- Login form



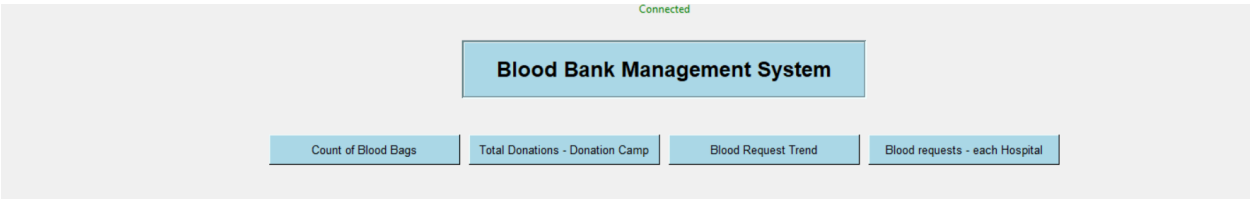
- Result table design



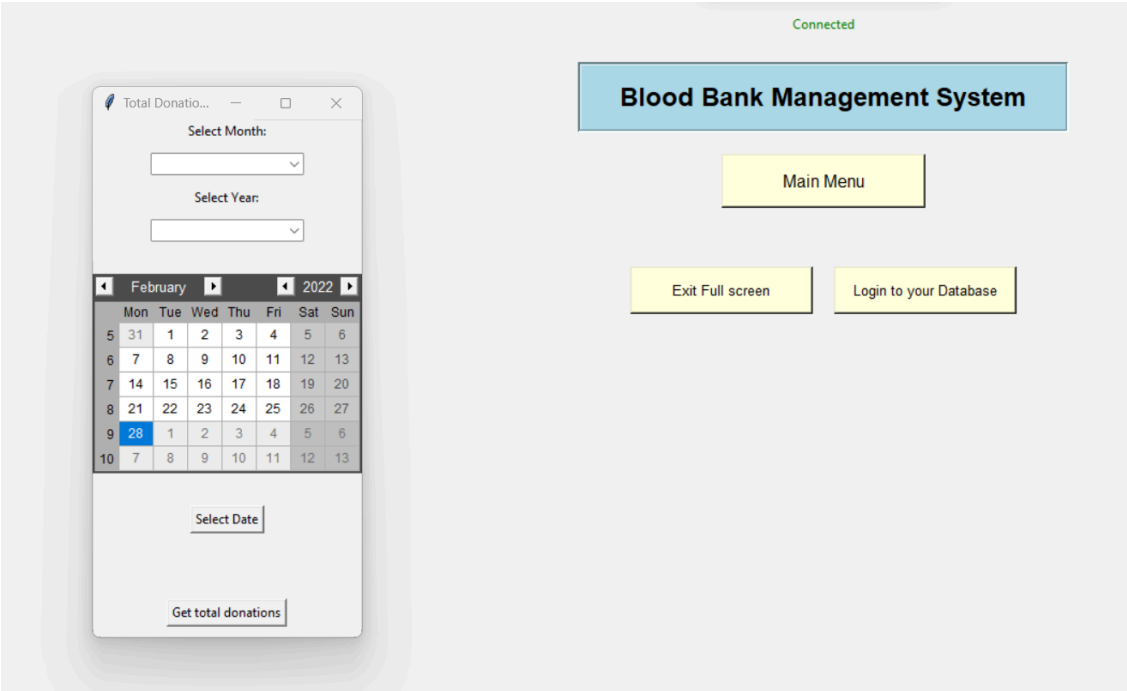
HospitalID	HospitalName	SupplyDemandGap	TotalBloodDonated	TotalBloodRequested	Address	Phone	Email	ContactPersonName
231	Muxo Hospital	-2	16	18	5 Mallory Avenue	610-611-3708	Muxo@gmail.com	Matti Autrie
192	Yodo Hospital	-6	21	27	94 Arrowood Trail	170-520-2946	Yodo@gmail.com	Marie-ann Luffman
309	Trilia Hospital	-7	19	26	57 Granby Park	870-335-0390	Trilia@gmail.com	Kenneth Embery
105	Topicstorm Hospital	-8	24	32	3836 Hallows Point	187-431-8519	Topicstorm@gmail.	Clo Gon
125	Talane Hospital	-10	3	13	7 Thompson Hill	784-413-1811	Talane@gmail.com	Shani Thomesson
191	Dabtype Hospital	-10	24	34	4 Mariners Cove Drive	673-294-2789	Dabtype@gmail.co	Esta Joskowitz
302	Devpulse Hospital	-11	25	36	83 Becker Terrace	316-658-0733	Devpulse@gmail.cc	Gearalt McFarlane
354	Divape Hospital	-12	16	28	192 Lakewood Gardens	692-870-5978	Divape@gmail.com	Daffie Sabban
357	Yotz Hospital	-12	28	40	8 Westport Avenue	711-426-7107	Yotz@gmail.com	Norrie Jouhan
195	Oyoyo Hospital	-13	33	46	6 Bunting Lane	117-690-8294	Oyoyo@gmail.com	Bethany Simoncelli
3	Digitube Hospital	-15	16	31	6 Shasta Center	262-910-3780	Digitube@gmail.co	Shea Worms
100	Vitz Hospital	-15	26	41	54004 Independence La	649-260-7546	Vitz@gmail.com	Monika De Michele
319	Voomm Hospital	-15	4	19	1 Lakewood Court	945-157-8870	Voomm@gmail.co	Caterina Cashman
443	Browsecat Hospital	-15	18	33	43228 Novick Drive	898-760-7849	Browsecat@gmail.c	Garald Roseby
89	Lazzy Hospital	-16	32	48	34588 Prentice Alley	167-413-4519	Lazzy@gmail.com	Collete Mannakee

- Connection status on the top after successful login

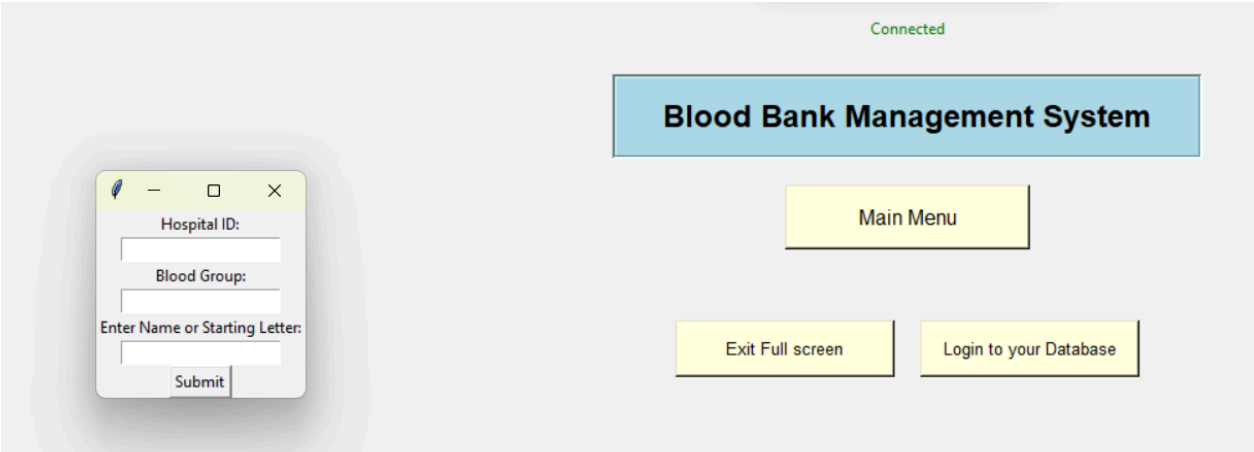
- Buttons for Tableau Visualisations that take you to Tableau public cloud.



- TkCalendar to efficiently fetch user input.



- Optional parameters so that the user can still see the result without providing any input.



- We created 2 stored procedures

1. Get Eligible Donors By BloodGroup

Name: GetEligibleDonorsByBloodGroup

The name of the routine is parsed automatically from th statement. The DDL is parsed automatically while you t

DDL:

```
CREATE DEFINER=`mm_team02_02`@`%` PROCEDURE `GetEligibleDonorsByBloodGroup`(IN input_bloodgroup VARCHAR(3))
BEGIN
    SELECT DonorID, Concat(FirstName," ",LastName) AS Name, BloodGroup, LastDonationDate, Phone
    FROM Donor
    WHERE LastDonationDate <= DATE_SUB(NOW(), INTERVAL 3 MONTH)
    AND BloodGroup = input_bloodgroup
    ORDER BY LastDonationDate DESC;
END
```

```
CALL GetEligibleDonorsByBloodGroup('O+');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DonorID	Name	BloodGroup	LastDonationDate	Phone
2477	Walsh Cruikshank	O+	2023-11-04 00:00:00	456-241-8805
1221	Patton Rudram	O+	2023-10-30 00:00:00	897-127-3681
1116	Ida McGinlay	O+	2023-10-12 00:00:00	789-908-0817
2863	Dorie Mandre	O+	2023-10-11 00:00:00	840-487-2145
1121	Kip Burgis	O+	2023-10-09 00:00:00	102-267-3765
1483	Carolann Sisselot	O+	2023-10-08 00:00:00	239-273-1609
1856	Robers Gain	O+	2023-09-27 00:00:00	560-472-5272
1159	Kalinda Warrender	O+	2023-09-25 00:00:00	619-159-1247
2670	Aveline Searle	O+	2023-09-25 00:00:00	723-454-9722
2320	Jerrold Redwin	O+	2023-09-17 00:00:00	769-171-2836
445	Gale Preto	O+	2023-09-16 00:00:00	218-950-2467
928	Merrel Cheke	O+	2023-09-14 00:00:00	808-941-9423
216	Bogey Siddell	O+	2023-08-24 00:00:00	256-478-2678
2393	Ardith Kieff	O+	2023-08-14 00:00:00	361-908-7687

Output:

2. Get Total Donations by Month or Year

Name: GetTotalDonationsByMonthYear

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
CREATE DEFINER=`mm_team02_02`@`%` PROCEDURE `GetTotalDonationsByMonthYear`(IN inputMonth INT, IN inputYear INT, OUT totalDonations INT)
BEGIN
    SELECT COUNT(*) INTO totalDonations
    FROM DonationEntry
    WHERE MONTH(DonationTS) = inputMonth AND YEAR(DonationTS) = inputYear;
END
```

```
use mm_team02_02;

CALL GetTotalDonationsByMonthYear(5, 2023, @totalDonations);
SELECT @totalDonations;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	@totalDonations
▶	55

Databases where we have our data

- MySQL db localhost
- Seattle University's public cloud - CSSQL
- AWS RDS
- MongoDB