CS432 - DATABASE

Assignment 3

Library Management and system

Team Members

Jayesh Bhadange 20110082

Pranav Rathod 20110143

Naval Jaggi 20110118

Harendra Khatik 20110072

Sparsh Dawra 20110203

Manpreet Singh 20110109

Sidharth Joshi 19110169

Gajanan Donge 20110061

Manish Jangir 20110107

Sankarshan 20110184

Riya Dhantoliya 20110168

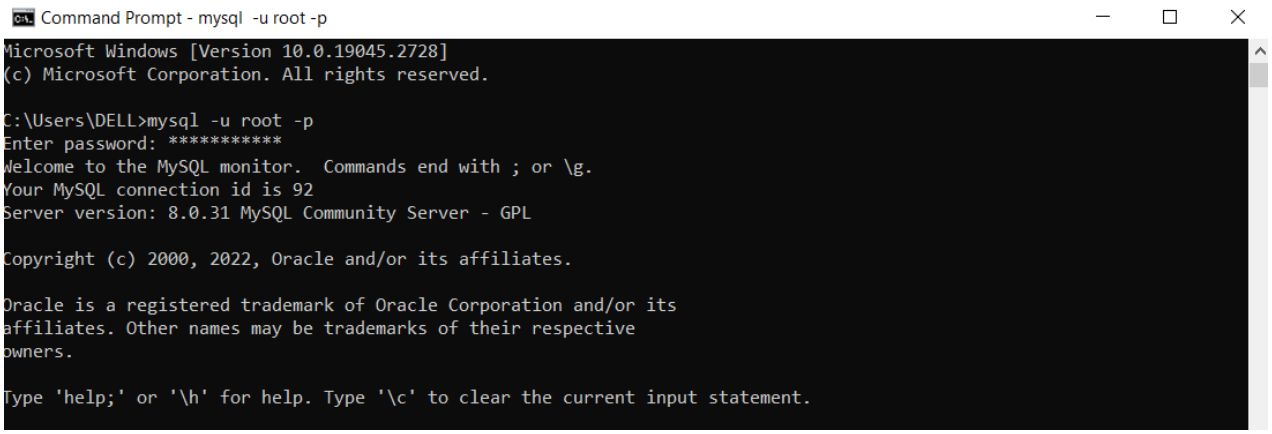
Pintu Kumar Meena 19110193

Installation Requirements:

- Pip3 install pyyaml
- Pip3 intall flask
- Pip3 install flask-mysqldb
- Python3 -m venv env-flask
- MySQL Server and workbench

❖ First create database of library management system using cmd

1) Open mysql from cmd (mysql -u root -p)
   Enter password



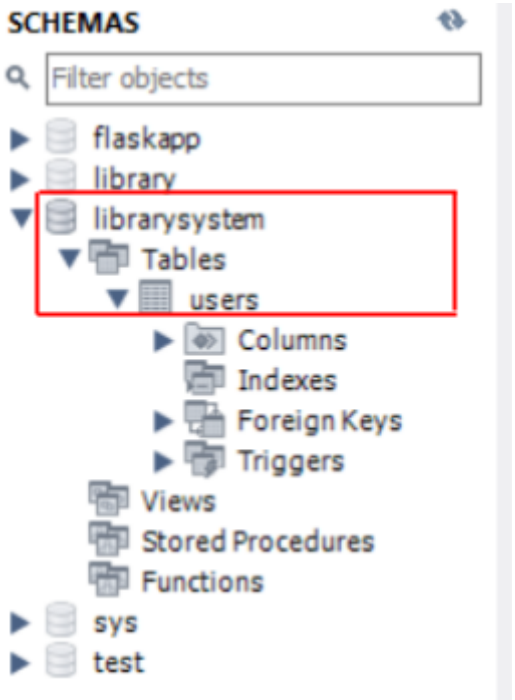
2) Then show databases.

3) If database is not created, create database librarysystem,

4) So now database do not have any table, so we can create table from MySQL workbench or from cmd

```
mysql> CREATE TABLE users(name VARCHAR(20), email VARCHAR(40));
Query OK, 0 rows affected (0.03 sec)
```

We can see schema is updated in mysql workbench also.



❖ Steps for website:

1) Crate file **db.yaml** file ,

```
! db.yaml > ...
1   mysql_host: 'localhost'
2   mysql_user: 'root'
3   mysql_password: '*********'
4   mysql_db: 'LibrarySystem'
5
```

The db.yaml file typically contains information about the database connection, such as the database server's hostname, port number, username, and password, as well as other options such as the name of the database to be used.
By storing this information in a separate configuration file, it becomes easier to manage and modify the database connection settings without having to modify the code of the Flask application itself. This also makes it easier to share the application with others who may need to use different database connection settings.

2) **App.py** file - In the case of a Flask web application that uses a database, the app.py file typically includes code to create and manage the database connection, as well as code to define and implement the various routes and views that the application provides.

3) Create a **template folder** where we will store all our .html files which defines structure and layout of weboages

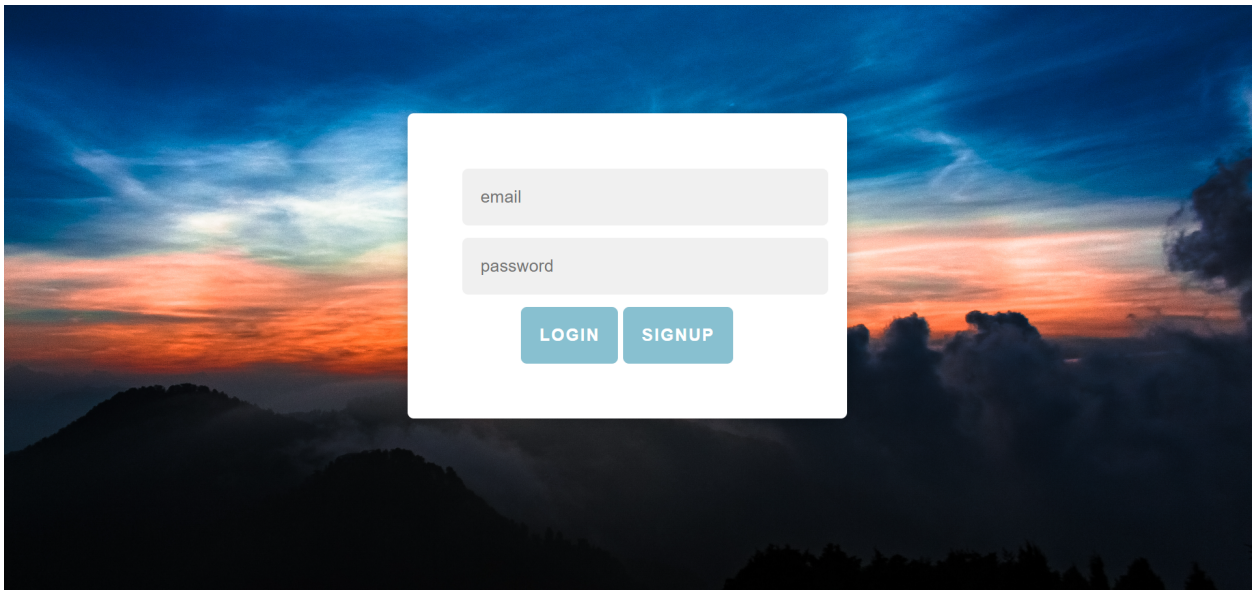
4) Create **static folder** which contains static files such as images, CSS files, and JavaScript files that are served directly to the user's web browser without any processing by the server.

5) Create **DBMS.html** file - contains HTML template that defines the structure and layout of the main page or landing page of the web application. This file typically includes placeholders for dynamic content that will be filled in by Flask at runtime.

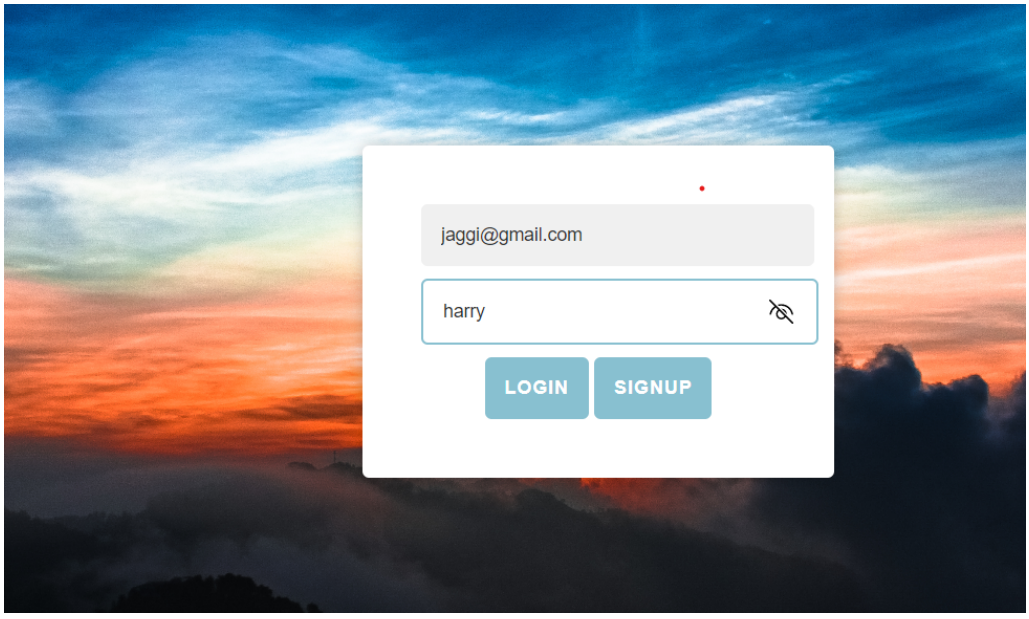
❖ After running code,

```
* Running on http://127.0.0.1:5000
```

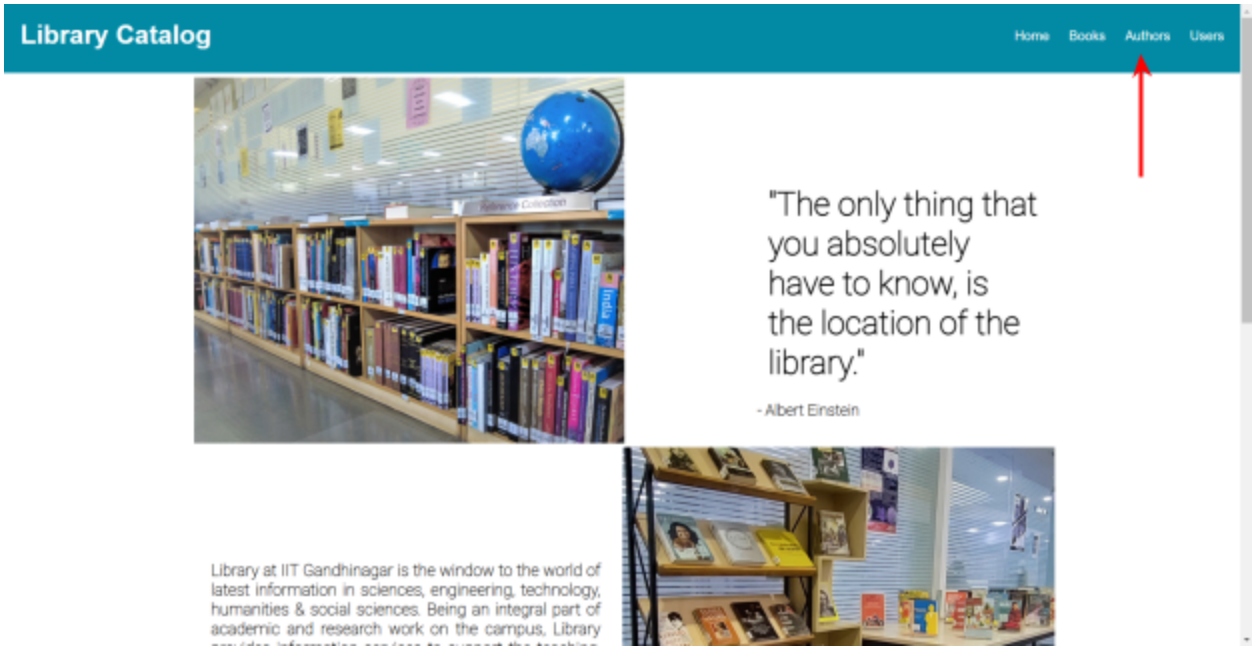
In the URL localhost:5000, 5000 refers to the port number that the Flask application is running on.



Admin view :
Username: **jaggi@gmail.com**
Password: **harry**



After logging in, main page of library will be open.



After clicking on authors navigation bar, we can see author table which contains author id and author name.

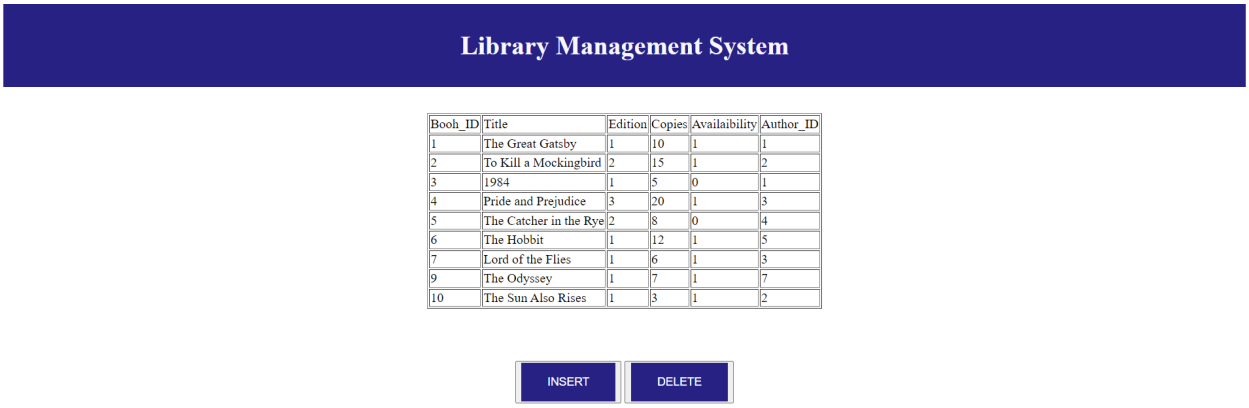
Authors Table view:-

Library Management System

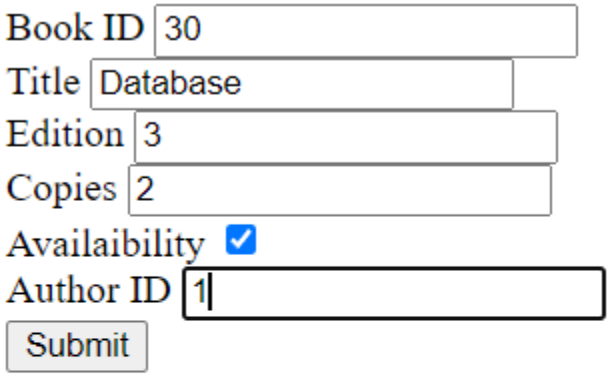
author_ID	author_Name
1	J K Rowling
2	Stephen King
3	Margaret Atwood
4	James Baldwin
5	Jane Austen
6	Gabriel Garcia Marquez
7	Chimamanda Ngozi Adichie
8	Toni Morrison
9	Haruki Murakami
10	Salman Rushdie

After clicking on book navigation bar, we can see book table contains columns as Book_ID, Title, Edition, Copies, Availability and foreign key as author id

Books view:-

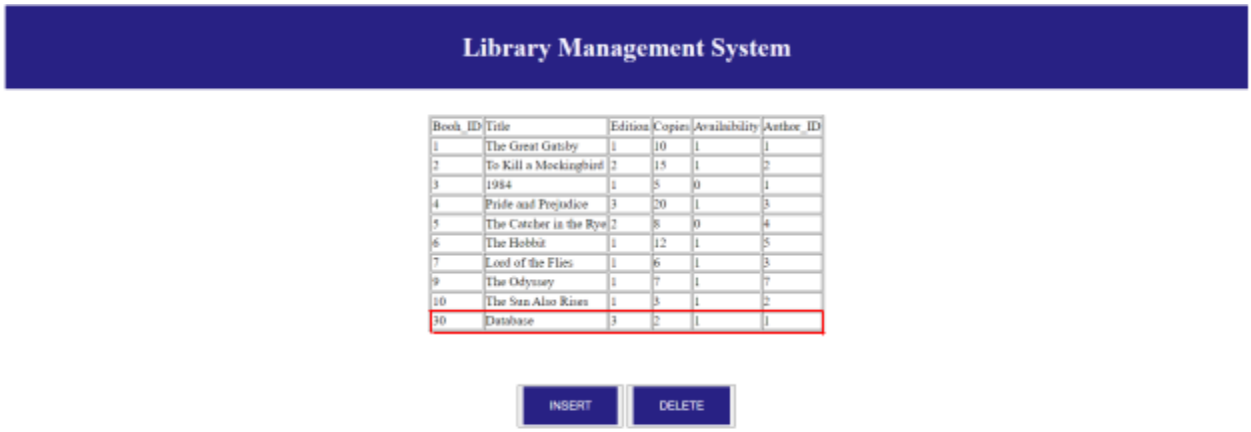


Insert Operation :- on book table we can perform operation Insert to add new book in library.
After clicking Insert tab, a new window will be open as shown below.



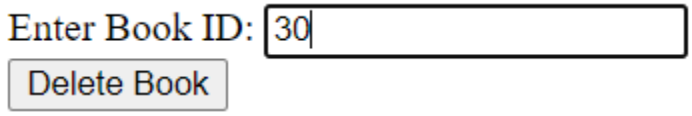
After submitting it, it will be added into database and book table as shown below.

Updated database:-



Again on same book table we can perform delete operation to remove book from database. After clicking delete option in book table a window will appear as shown in below. For deleting book we just have to enter book id and that row will be deleted.

delete:-



Updated Book table:-

Library Management System

Booh_ID	Title	Edition	Copies	Availaibility	Author_ID
1	The Great Gatsby	1	10	1	1
2	To Kill a Mockingbird	2	15	1	2
3	1984	1	5	0	1
4	Pride and Prejudice	3	20	1	3
5	The Catcher in the Rye	2	8	0	4
6	The Hobbit	1	12	1	5
7	Lord of the Flies	1	6	1	3
9	The Odyssey	1	7	1	7
10	The Sun Also Rises	1	3	1	2

INSERT DELETE

Here we have created a user table contains columns as UserID, First name, middle name, end name and contact as email id.

User table:-

Initial Table:-

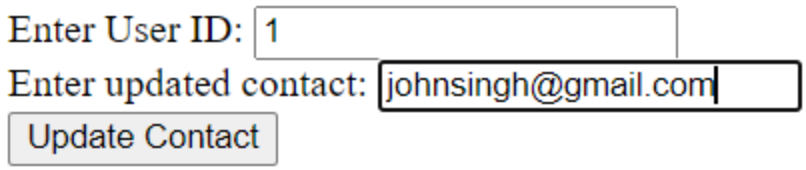
Library Management System

UserID	first_name	middle_name	end_name	Contact
1	John	David	Doe	john.doe@example.com
2	Jane	Elizabeth	Smith	jane.smith@example.com
3	Michael	Patrick	Scott	michael.scott@example.com
4	Pamela	Morgan	Halpert	pamela.halpert@example.com
5	Jim	Duncan	Halpert	jim.halpert@example.com
6	Angela	Noelle	Martin	angela.martin@example.com
7	Kevin	Jay	Malone	kevin.malone@example.com
8	Oscar	Gutierrez	Martinez	oscar.martinez@example.com
9	Toby	Wyatt	Flenderson	toby.flenderson@example.com
10	Stanley	James	Hudson	stanley.hudson@example.com

Update Contact DELETE RENAME

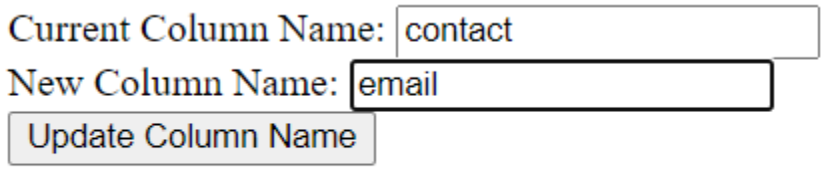
Here on this table we can perform 3 operations, update contact, delete contact and rename contact. As shown below.

Update:- after clicking update contact a window will appear as shown below.



Updated users table:-

Rename:- This operation is used to change column name



Rename updated:-

Contribution :

- G1 - Frontend
  1) Sankarshan
  2) Gajanan Donge
  3) Riya dhantoliya
  4) Jayesh Bhadange
  5) Manpreet singh

- G2 - Backend
  1) Harendra khatik
  2) Pranav rathod
  3) Naval Jaggi
  4) Siddarth joshi
  5) Sparsh dawra