DBMS - Mini Project

Online Exam Management System

Submitted by

Akash Kumar
PES1UG20CS027
V Semester Section A

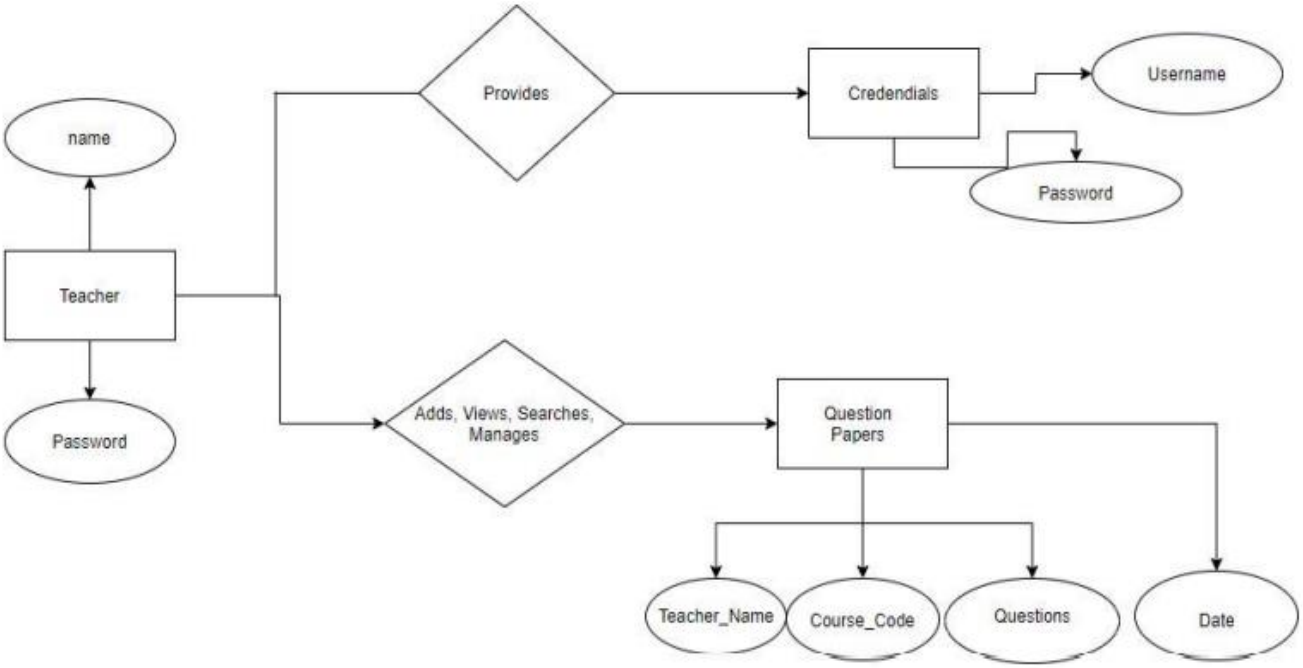
Short Description and Scope of the Project

The concept of traditional education has changed radically within the last couple of years.
Being physically present in a classroom isn't the only learning or making question papers and thus organizing them and also taking the exam offline option anymore — not with the rise of the internet and new technologies, at least.
Nowadays, we have access to a quality education whenever and wherever we want, as long as we have access to a computer. We are now entering a new era — the revolution of online education & Online Examination System is undoubtedly the ultimate backbone of this revolution. Is a web-based examination system where examinations are given online, either through the internet or intranet using computer system.

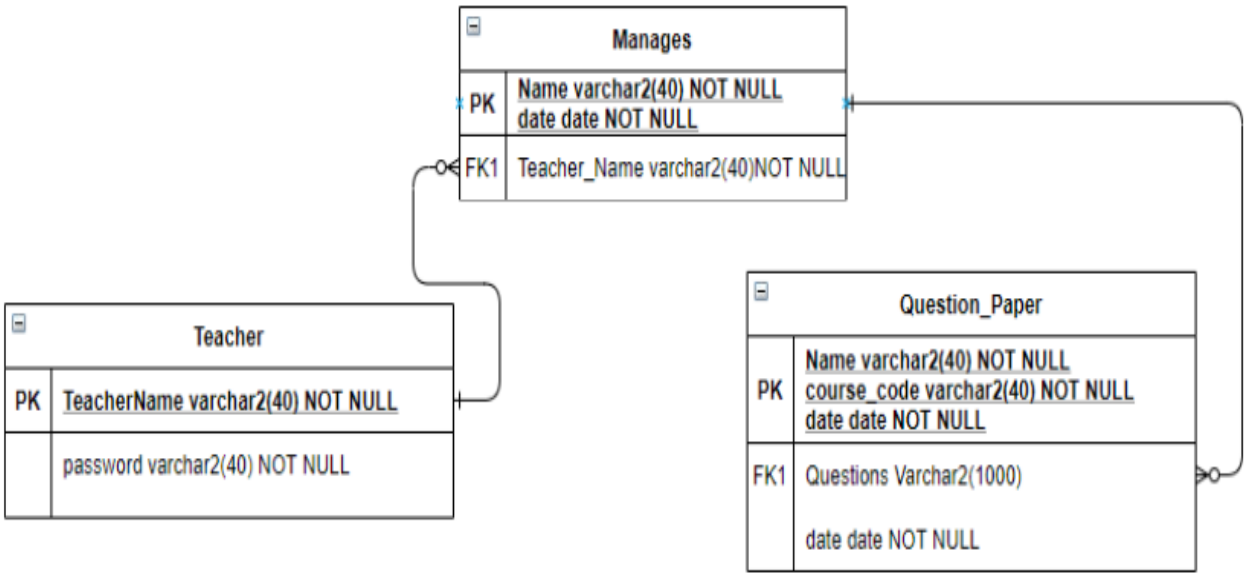
SCOPE

This System is developed or designed for educational institutes This system is designed to provide or facilitate Administrator and User. Complete and safe information is provided to the user. This system can be used anywhere anytime as it is a web-based application.

ER Diagram


name
Provides
Credendials
Username
Password
Teacher
Password
Adds, Views, Searches, Manages
Question Papers
Teacher_Name
Course_Code
Questions
Date

Relational Schema



DDL statements - Building the database

```
CREATE TABLE userstable (
  User varchar (50) NOT NULL,
  Pass varchar (32) NOT NULL
)
```

```
sqlite> select * from userstable
   ...> ;
qw|5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c173cacfc5
sm|5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c173cacfc5
Akashk12|00f51d088dc245360c6dd655f34daa81ddfa34f1f7a1277c78ede5aecee754e6
Mam1|a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3
```

```
CREATE TABLE IF NOT EXISTS qptable(teacher TEXT,subject TEXT,questions TEXT,date
DATE)
```

```
sqlite> .tables
numbers     qptable     tbl_admin   userstable
sqlite>
```

```
CREATE TABLE tbl_ans (
  `id` int(11) NOT NULL,
  `quesNo` int(11) NOT NULL,
  `rightAns` int(11) NOT NULL DEFAULT '0',
  `ans` text NOT NULL
)
```

Populating the Database

INSERT INTO userstable(username,password) VALUES (?,?)', (username, password))

```
sqlite> pragma table_info(userstable);
0|username|TEXT|0||0
1|password|TEXT|0||0
sqlite>
```

insert into qptable(teacher,subject,questions,date)
...> values("mam2" , "software", "What is SE ?", "28/11/22”);

```
mam2|software|What is SE ?|28/11/22
sqlite>
```

```
sqlite> select * from qptable
   ...> ;
Akashk12|UE20CS301|1. What is the full form of DBMS?
a) Data of Binary Management System
b) Database Management System
c) Database Management Service
d) Data Backup Management System

2. In which of the following formats data is stored in the database management system?
a) Image
b) Text
c) Table
d) Graph

3. Which of the following is not a type of database?
a) Hierarchical
b) Network
c) Distributed
d) Decentralized|2022-11-27
Mam1|QZ1| _____ operations do not preserve non-matched tuples.
a) Left outer join
b) Inner join
c) Natural join
d) Right outer join
```

INSERT INTO `tbl_ans` (`id`, `quesNo`, `rightAns`, `ans`)
...>VALUES
(128, 1, 0, 'Bev Littlewood'),
(129, 1, 1, 'Berry Bohem');

Join Queries

1.
NATURAL JOIN

```
sqlite> select username from userstable natural join tbl_admin;
qw
sm
```

2.
CROSS JOIN matches every row of the first table with every row of the second table. If the input tables have x and y row, respectively, the resulting table will have x*y row. Because CROSS JOINs have the potential to generate extremely large tables, care must be taken to only use them when appropriate.

```
sqlite> select adminId from tbl_admin CROSS JOIN numbers;
1
sqlite> select adminId from tbl_admin CROSS JOIN numbers;
```

3.
INNER JOIN creates a new result table by combining column values of two tables (table1 and table2) based upon the join-predicate. The query compares each row of table1 with each row of table2 to find all pairs of rows which satisfy the join-predicate. When the join-predicate is satisfied, the column values for each matched pair of rows of A and B are combined into a result row.

```
sqlite> select username from userstable inner join tbl_admin;
qw
sm
Akashk12
Mam1
sqlite>
```

4.
OUTER JOIN is an extension of INNER JOIN. Though SQL standard defines three types of OUTER JOINs: LEFT, RIGHT, and FULL, SQLite only supports the **LEFT OUTER JOIN**. OUTER JOINs have a condition that is identical to INNER JOINs, expressed using an ON, USING, or NATURAL keyword. The initial results table is calculated the same way. Once the primary JOIN is calculated, an OUTER JOIN will take any unjoined rows from one or both tables, pad them out with NULLs, and append them to the resulting table.

```
sqlite> select username from userstable left outer join tbl_admin;
qw
sm
Akashk12
Mam1
sqlite>
```

Aggregate Functions

Showcase at least 4 Aggregate function queries
Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the result

count(*)

The count(X) function returns a count of the number of times that *X* is not NULL in a group. The count(*) function (with no arguments) returns the total number of rows in the group.

```
sqlite> select count(username)
   ...> from userstable;
4
sqlite>
```

min(*X*)

The min() aggregate function returns the minimum non-NULL value of all values in the group. The minimum value is the first non-NULL value that would appear in an ORDER BY of the column. Aggregate min() returns NULL if and only if there are no non-NULL values in the group.

```
sqlite> select min(col1,col2)
   ...> from numbers;
d1
sqlite>
```

max(*X*)

The max() aggregate function returns the maximum value of all values in the group. The maximum value is the value that would be returned last in an ORDER BY on the same column. Aggregate max() returns NULL if and only if there are no non-NULL values in the group.

```
sqlite> select max(col1,col2)
   ...> from numbers;
d2
sqlite>
```

sum(*X*)

The sum() and total() aggregate functions return the sum of all non-NULL values in the group.

```
sqlite> select sum(username)
   ...> from userstable;
0.0
sqlite>
```

Set Operations

UNION

```
sqlite> select teacher from qptable
   ...> union
   ...> select username from userstable;
Akashk12
Mam1
mam2
qw
sm
sqlite>
```

UNION ALL

```
sqlite> select teacher,subject from qptable
   ...> union all
   ...> select teacher, script from tbl_ans;
Akashk12|UE20CS301
Mam1|QZ1
mam2|software
Akashk12|ML012
sqlite>
```

EXCEPT

```
sqlite> select teacher from qptable
   ...> except
   ...> select username from userstable;
mam2
sqlite>
```

INTERSECT

```
sqlite> select username from userstable
   ...> intersect
   ...> select teacher from qptable;
Akashk12
Mam1
sqlite>
```

Functions and Procedures

SQLite has had to sacrifice other characteristics that some people find useful, such as high concurrency, fine-grained access control, a rich set of built-in functions, ***stored procedures***, esoteric SQL language features, XML and/or Java extensions, tera- or peta-byte scalability, and so forth.

```
sqlite> create procedure specified procedure name ([specified statement]) as B;
```

```
# DB  Functions----------------------------------------------------------------------------------------------


def create_usertable():
    c.execute('CREATE TABLE IF NOT EXISTS userstable(username TEXT,password TEXT)')


def add_userdata(username, password):
    c.execute('INSERT INTO userstable(username,password) VALUES (?,?)', (username, password))
    conn.commit()


def login_user(username, password):
    c.execute('SELECT * FROM userstable WHERE username =? AND password = ?', (username, password))
    data = c.fetchall()
    return data


def view_all_users():
    c.execute('SELECT * FROM userstable')
    data = c.fetchall()
    return data
```

Triggers and Cursors

SQLite **Triggers** are database callback functions, which are automatically performed/invoked when a specified database event occurs

Syntax example:
CREATE TRIGGER trigger_name [BEFORE|AFTER] event_name
ON table_name
BEGIN
 -- Trigger logic goes here....
END;

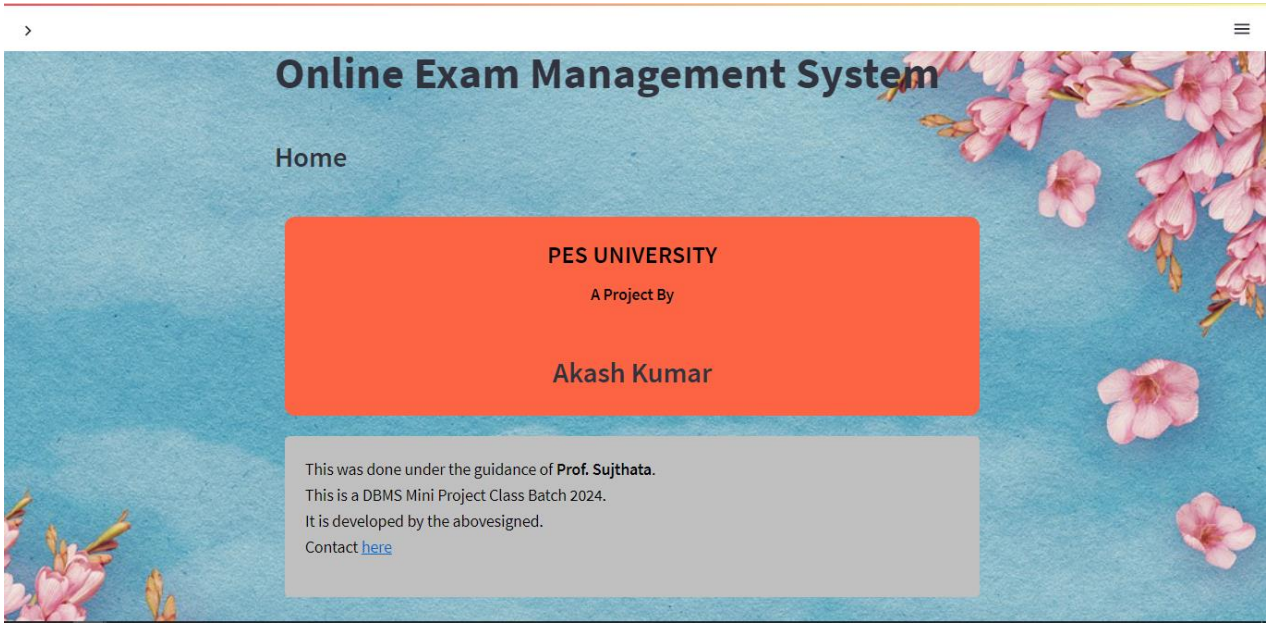
event_name could be *INSERT, DELETE,* and *UPDATE* database operation on the mentioned table **table_name**. You can optionally specify FOR EACH ROW after table name.

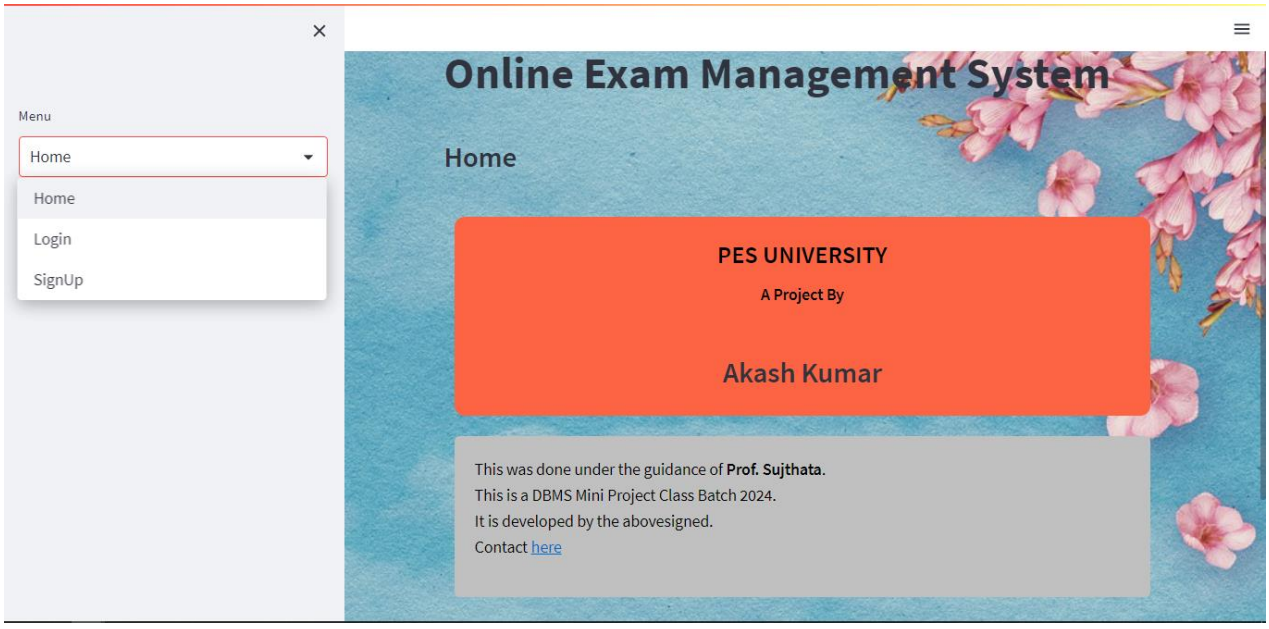
```
sqlite> CREATE TABLE COMPANY(
   ...>    ID INT PRIMARY KEY     NOT NULL,
   ...>    NAME           TEXT    NOT NULL,
   ...>    AGE            INT     NOT NULL,
   ...>    ADDRESS        CHAR(50),
   ...>    SALARY         REAL
   ...> );
sqlite> CREATE TABLE AUDIT(
   ...>    EMP_ID INT NOT NULL,
   ...>    ENTRY_DATE TEXT NOT NULL
   ...> );
sqlite> CREATE TRIGGER audit_log AFTER INSERT
   ...> ON COMPANY
   ...> BEGIN
   ...>    INSERT INTO AUDIT(EMP_ID, ENTRY_DATE) VALUES (new.ID, datetime('now'));
   ...> END;
sqlite>  INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
   ...> VALUES (1, 'Paul', 32, 'California', 20000.00 );
sqlite>  SELECT name FROM sqlite_master
   ...> WHERE type = 'trigger';
audit_log
sqlite>
```

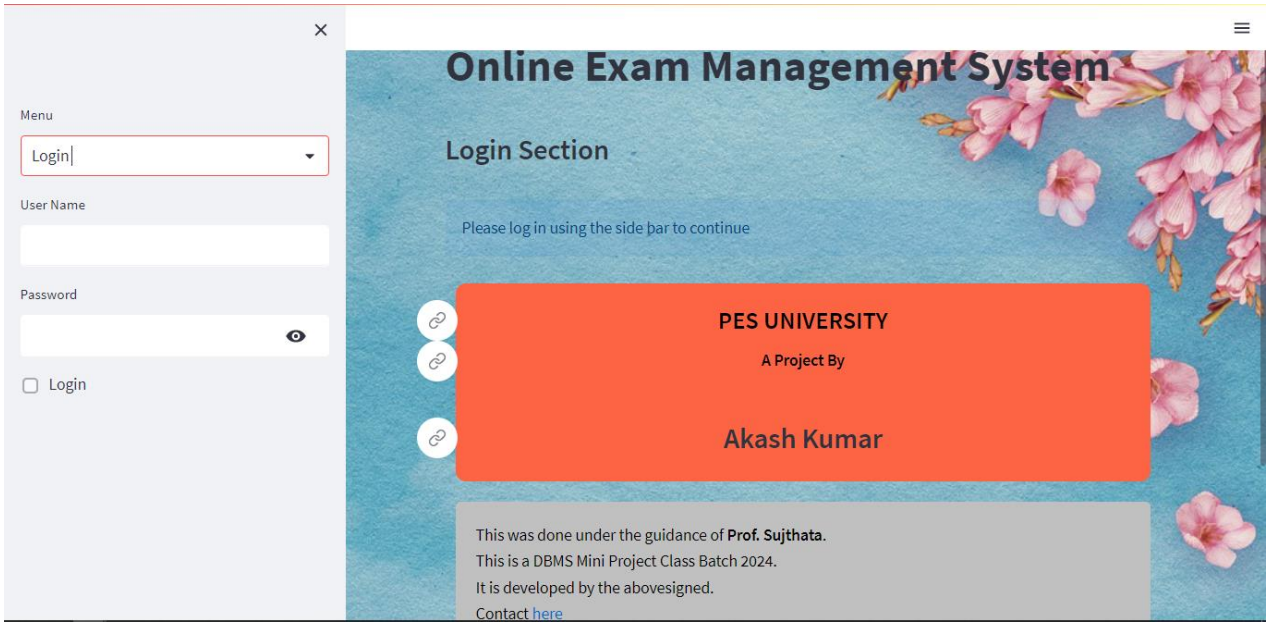
The sqlite3.Cursor class is an instance using which you can invoke methods that execute SQLite statements, fetch data from the result sets of the queries. You can create **Cursor** object using the cursor() method of the Connection object/class.

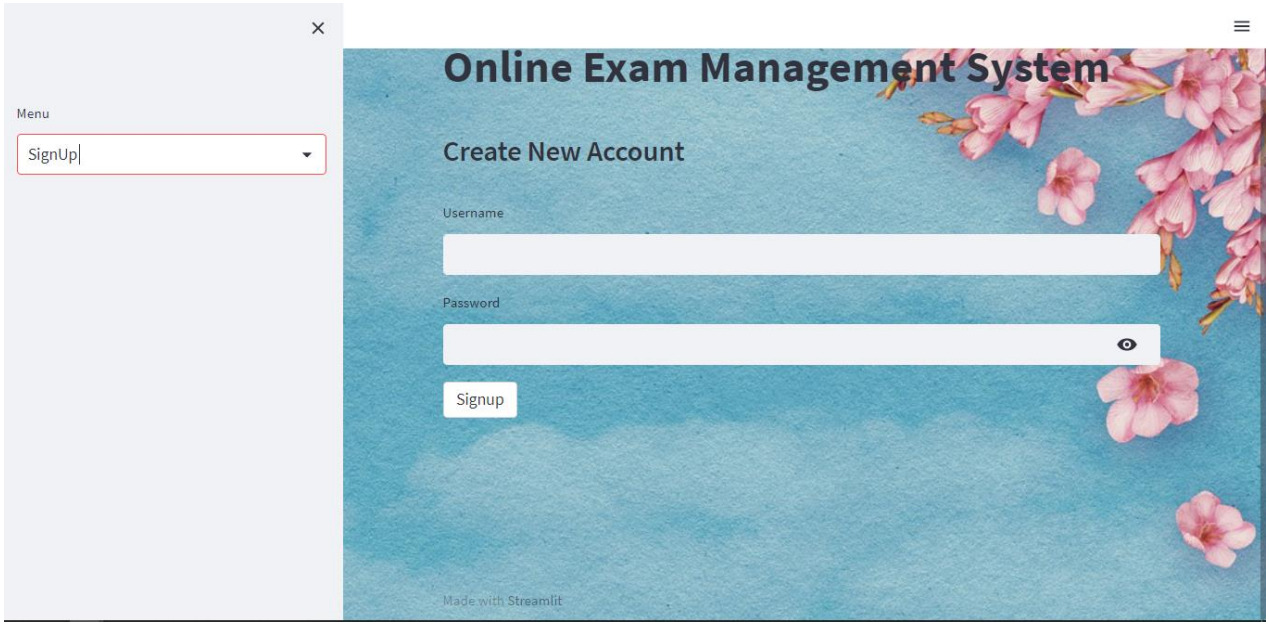
```
# DB Management----------------------------------------------------------------------------------------
import sqlite3
conn = sqlite3.connect('QPTABLE.db')
c = conn.cursor()
```

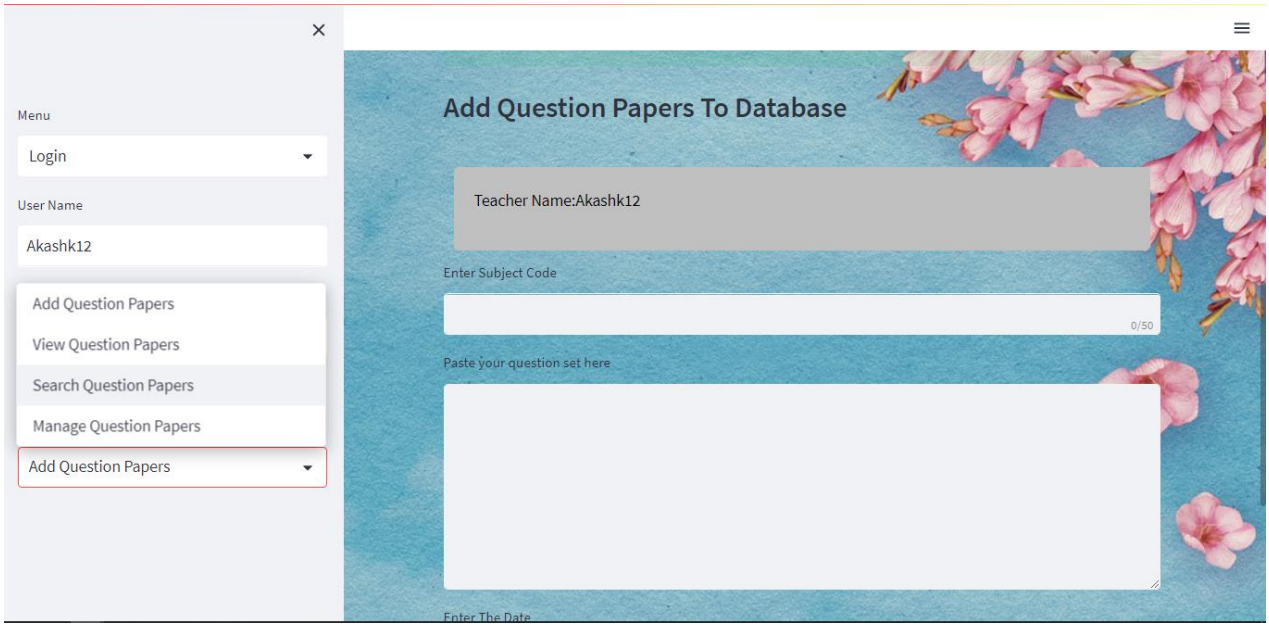
Developing a Frontend

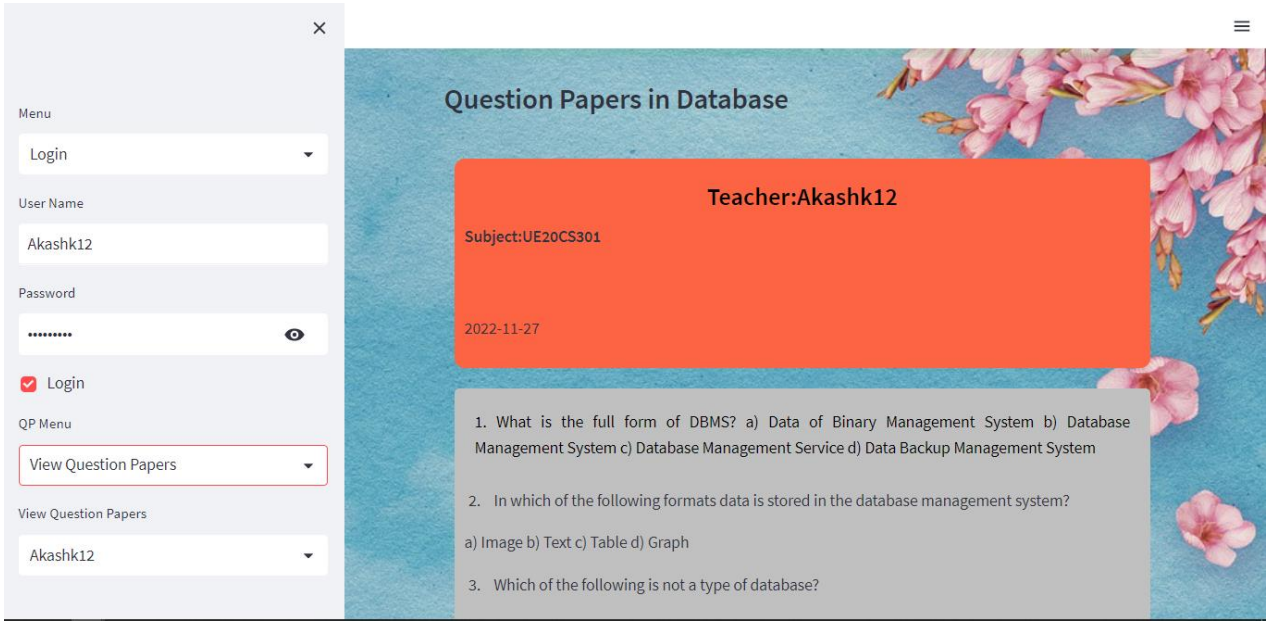


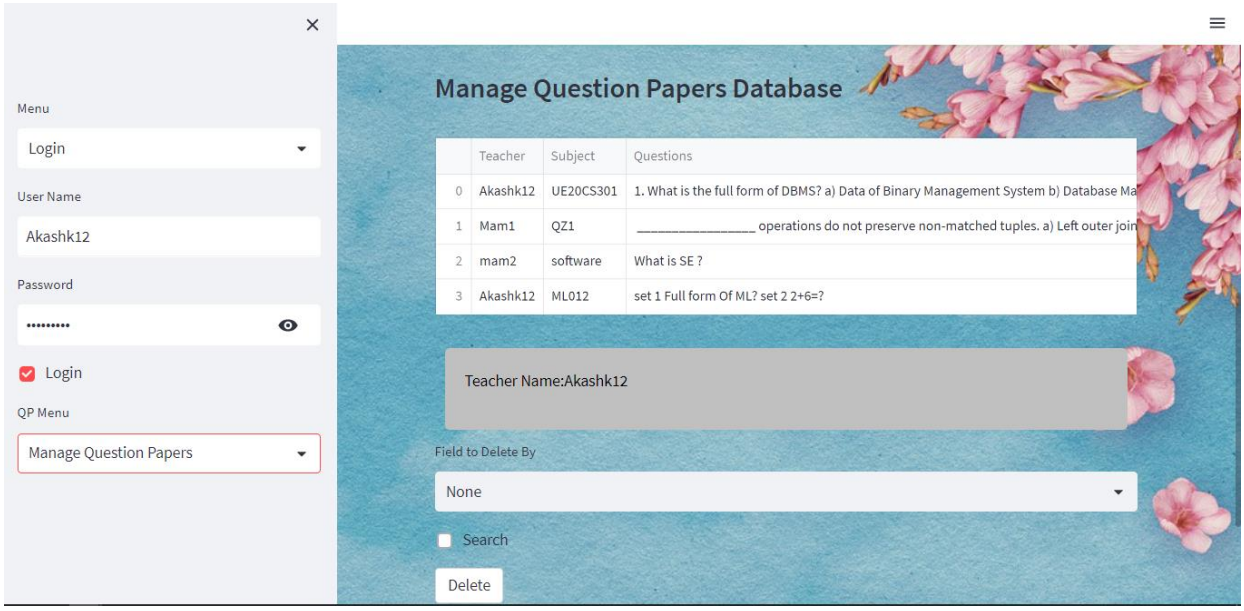



Menu
Login
User Name
Password
Login
Online Exam Management System
Login Section
Please log in using the side bar to continue
PES UNIVERSITY
A Project By
Akash Kumar
This was done under the guidance of Prof. Sujthata.
This is a DBMS Mini Project Class Batch 2024.
It is developed by the abovesigned.
Contact here


Menu
SignUp
Online Exam Management System
Create New Account
Username
Password
Signup
Made with Streamlit


Menu
Login
User Name
Akashk12
Add Question Papers
View Question Papers
Search Question Papers
Manage Question Papers
Add Question Papers
Add Question Papers To Database
Teacher Name:Akashk12
Enter Subject Code
0/50
Paste your question set here
Enter The Date


Menu
Login
User Name
Akashk12
Password
Login
QP Menu
View Question Papers
View Question Papers
Akashk12
Question Papers in Database
Teacher:Akashk12
Subject:UE20CS301
2022-11-27
1. What is the full form of DBMS? a) Data of Binary Management System b) Database Management System c) Database Management Service d) Data Backup Management System
2. In which of the following formats data is stored in the database management system?
a) Image b) Text c) Table d) Graph
3. Which of the following is not a type of database?


Menu
Login
User Name
Akashk12
Password
Login
QP Menu
Manage Question Papers
Manage Question Papers Database
Teacher
Subject
Questions
0
Akashk12
UE20CS301
1. What is the full form of DBMS? a) Data of Binary Management System b) Database Ma
1
Mam1
QZ1
_____ operations do not preserve non-matched tuples. a) Left outer join
2
mam2
software
What is SE ?
3
Akashk12
ML012
set 1 Full form Of ML? set 2 2+6=?
Teacher Name:Akashk12
Field to Delete By
None
Search
Delete


thank you