Python

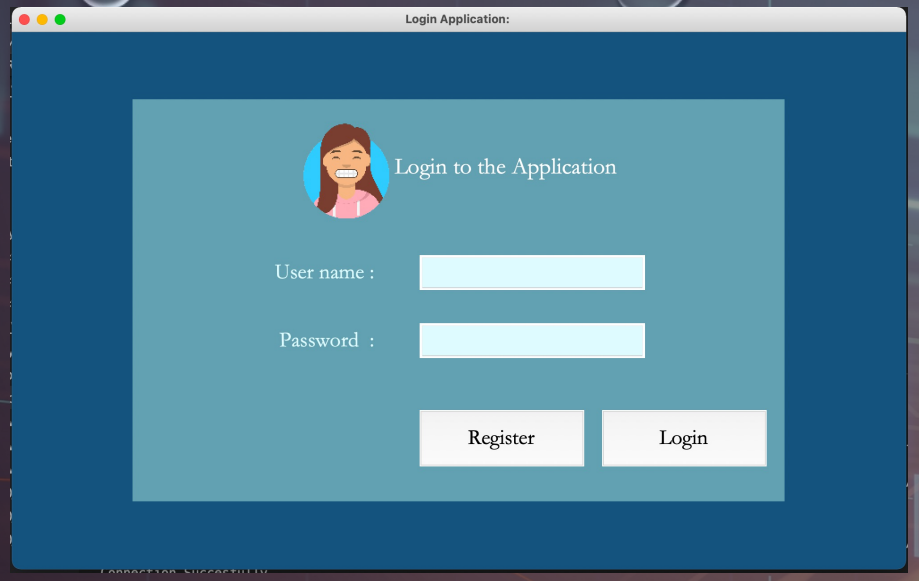
SQLite 3

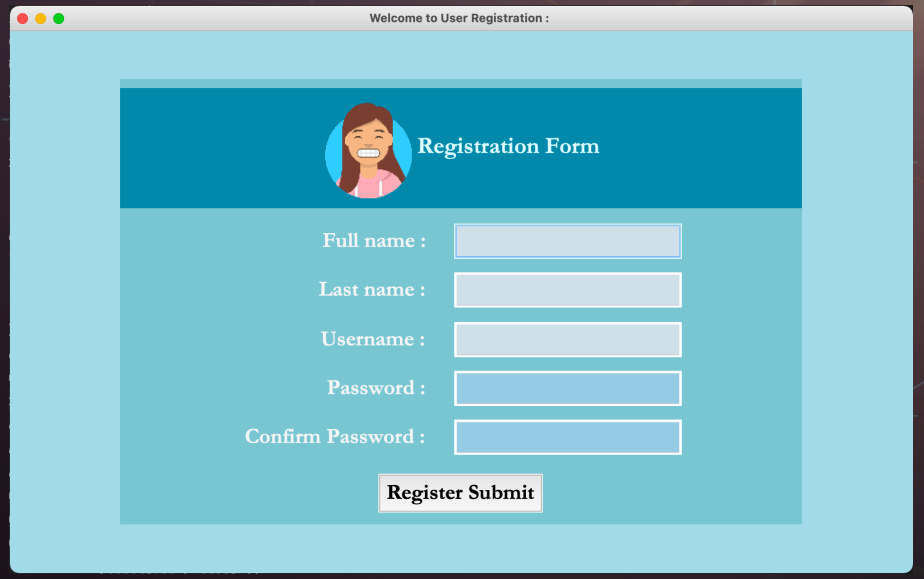
Database

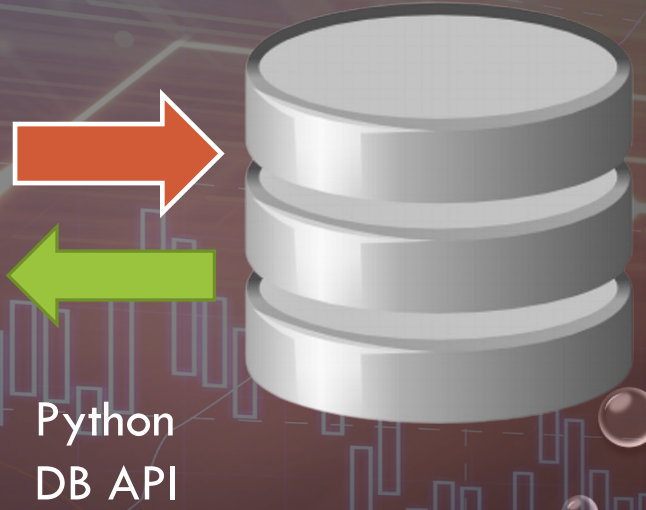
CS311: Computer Programming II

ผู้ช่วยศาสตราจารย์สิรินธร จียาศักดิ์

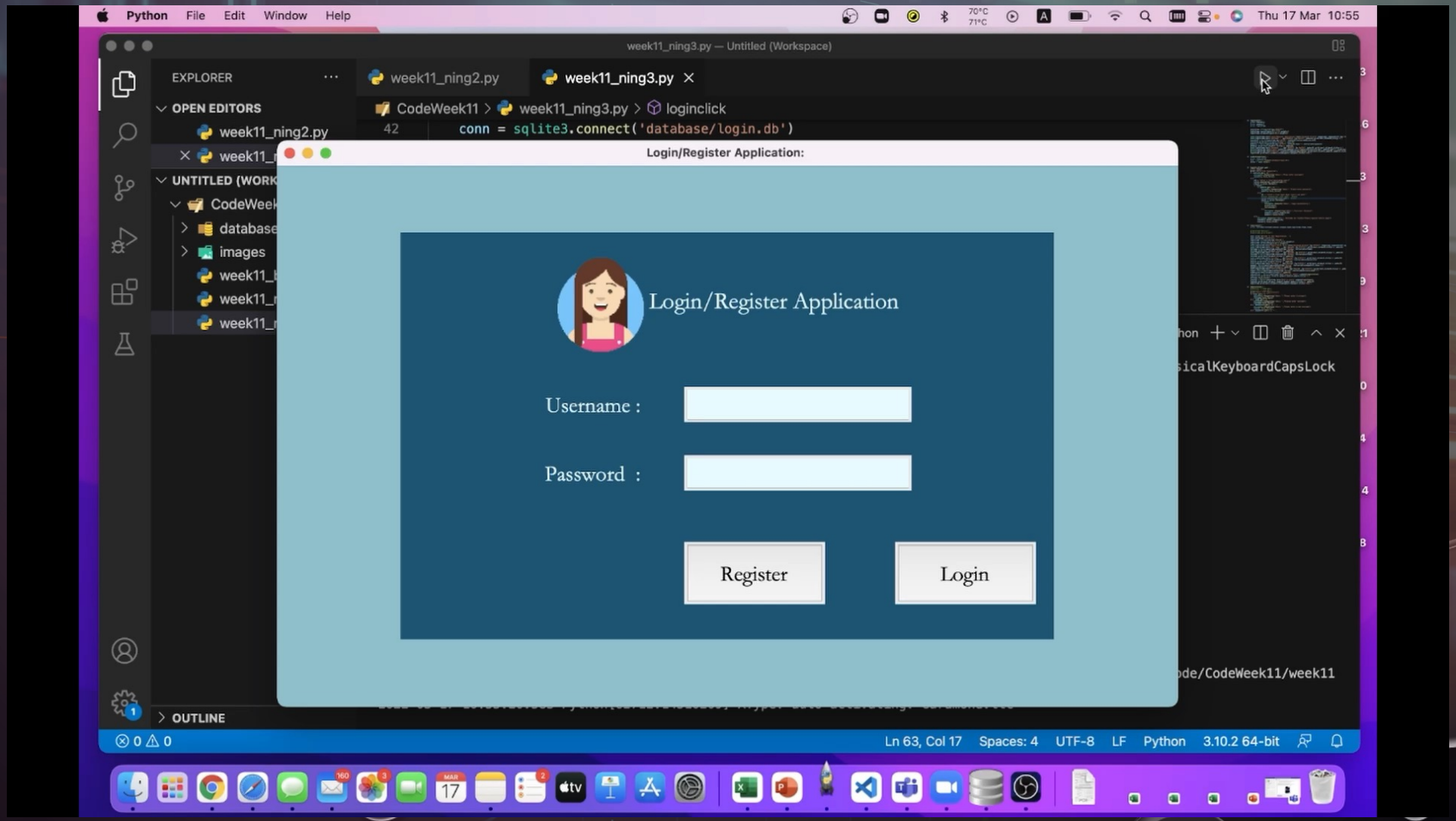
OUTCOME OF THIS WEEK







OUTCOME OF THIS WEEK

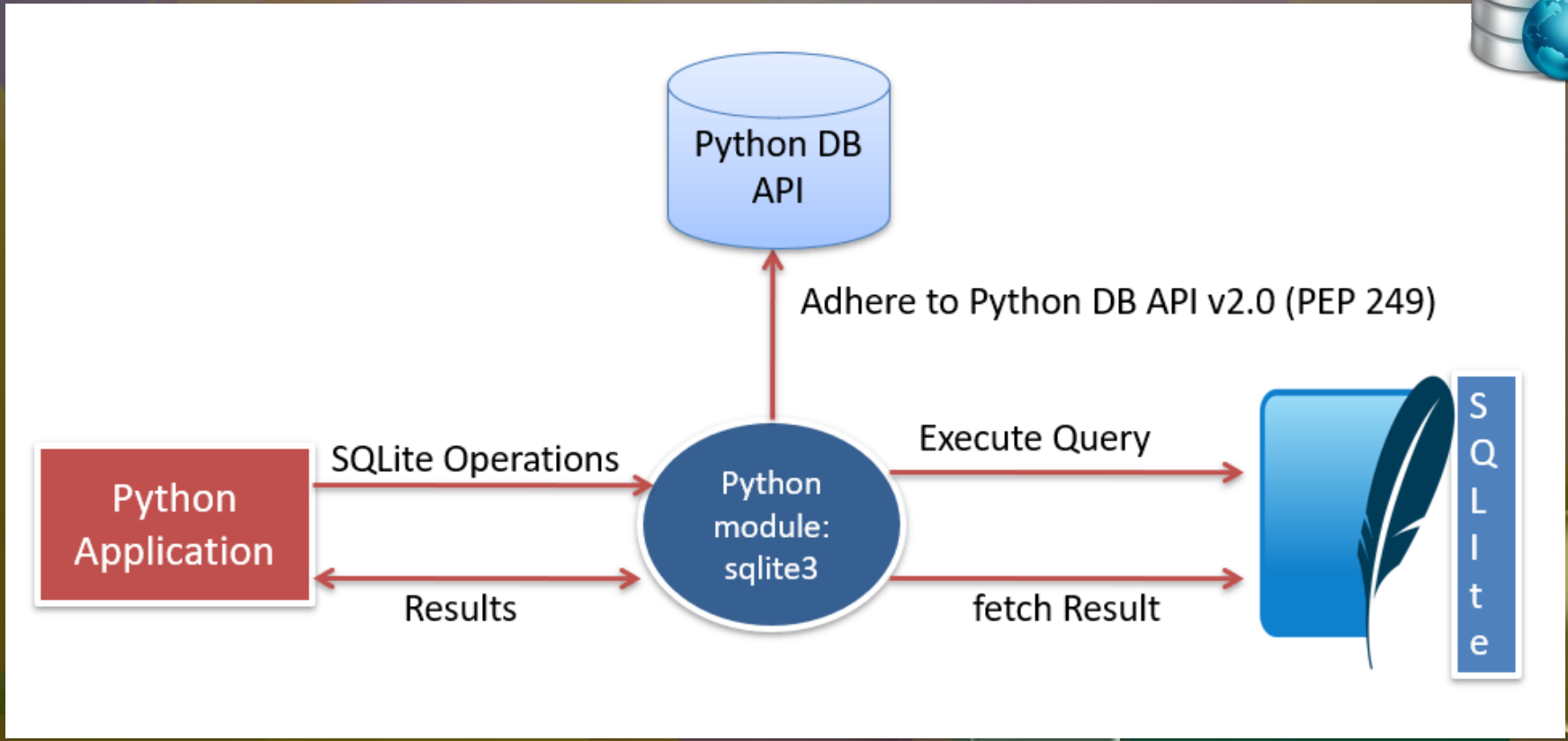


AGENDA

1. Steps to Creating a Connection
2. SQL for Insert a New Record
3. Steps of Insert into Table

1 Steps to Creating a Connection

PYTHON SQLITE CONNECTION ARCHITECTURE



https://pynative.com/python-sqlite/

STEPS OF CONNECTING SQLITE DB

1. **IMPORT SQLITE3 MODULE.** USING THE THE SQLITE3 MODULE TO COMMUNICATE WITH THE SQLITE DATABASE.
2. **USE THE CONNECT().** THIS METHOD RETURNS THE SQLITE CONNECTION OBJECT IF THE CONNECTION IS SUCCESSFUL.
3. **USE THE CURSOR() METHOD.** A CURSOR OBJECT TO EXECUTE SQLITE COMMAND/QUERIES FROM PYTHON.
4. **USE THE EXECUTE() METHOD.** THE EXECUTE() METHODS RUN THE SQL QUERY AND RETURN THE RESULT.
5. **EXTRACT RESULT USING FETCHALL()**
   USE CURSOR.FETCHALL() OR FETCHONE() OR FETCHMANY() TO READ QUERY RESULT.
6. **CLOSE CURSOR AND CONNECTION OBJECTS**
   USE CURSOR.CLOSE() AND CONNECTION.CLOSE() TO CLOSE THE CURSOR AND SQLITE CONNECTIONS AFTER YOUR WORK COMPLETES.

2 Insert a New Record

HOW TO INSERTING DATA INTO A DATABASE

Insert a new record

SQL STATEMENT FOR INSERTING

- The **insert into statement** is used to insert new records in a table.

- Syntax :

```
INSERT INTO tablename
VALUES (value1, value2, value3, ...);
```

```
INSERT INTO tablename (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

INSERT INTO STATEMENT EXAMPLE

- INSERT INTO STATEMENT OF SQL LANGUAGE

```
insert into login values
('test1','test1','testman1','surnametest1')
```

```
insert into login values
('test2','test2','testman2','surnametest2')
```

```
insert into login (user,pwd,fname,lname)
values ('test3','test3','testman3','surnametest3')
```

user	pwd	fname	lname
Filter	Filter	Filter	Filter
admin1	admin1	admin1	surname1
admin2	admin2	admin2	surname2
ning1	ning1	Sirinthorn1	Cheyasak1
ning2	ning2	Sirinthorn2	Cheyasak2
ning3	ning3	Sirinthorn3	Cheyasak3
ning4	ning4	Sirinthorn4	Cheyasak4

WEEK11 : CLASS ACTIVITY#1

- CREATE A CUSTOMER TABLE WHICH THE OUTPUT AS SHOWN BELOW

Tables (1)		
customer		CREATE TABLE "customer" ("id"
id	INTEGER	"id" INTEGER NOT NULL
name	TEXT	"name" TEXT NOT NULL
age	INTEGER	"age" INTEGER NOT NULL
address	TEXT	"address" TEXT NOT NULL
salary	REAL	"salary" REAL NOT NULL

WEEK11 : CLASS ACTIVITY#1

- INSERT 5 RECORDS INTO THE CUSTOMER TABLE WHICH THE OUTPUT AS SHOWN BELOW (INSERT INTO STATEMENT)

Table: customer

	id	name	age	address	salary
	Filter	Filter	Filter	Filter	Filter
1	1	Sirinthorn1 Cheyasak1	22	Bangkok	35000.0
2	2	Sirinthorn2 Cheyasak2	22	Bangkok	25000.0
3	3	Sirinthorn3 Cheyasak3	23	Bangkok	27000.0
4	4	Sirinthorn4 Cheyasak4	22	Bangkok	25000.0
5	5	Sirinthorn5 Cheyasak5	25	Bangkok	45000.0

3 Python SQLite

Steps of insert into a table

STEPS OF INSERT INTO A TABLE

FOLLOW THE BELOW STEPS: –

1. **CONNECT TO SQLITE FROM PYTHON.** USING **SQLITE3 MODULE**.
2. **DEFINE A SQL INSERT QUERY.** NEXT, PREPARE A SQL INSERT QUERY TO INSERT A ROW INTO A TABLE.
3. **GET CURSOR OBJECT FROM CONNECTION.** NEXT, USE A CONNECTION.**CURSOR() METHOD** TO CREATE A CURSOR OBJECT. USING CURSOR OBJECT WE CAN EXECUTE SQL QUERIES.
4. **EXECUTE THE INSERT QUERY USING EXECUTE() METHOD.** THE **CURSOR.EXECUTE(QUERY)** METHOD EXECUTES THE OPERATION STORED IN THE INSERT QUERY.
5. **COMMIT YOUR CHANGES.** AFTER SUCCESSFULLY EXECUTING AN INSERT OPERATION, MAKE CHANGES PERSISTENT INTO A DATABASE USING THE **COMMIT()** OF A CONNECTION CLASS.
6. **VERIFY RESULT USING THE SQL SELECT QUERY.** IF REQUIRED, EXECUTE SQLITE SELECT QUERY FROM PYTHON TO SEE THE NEW CHANGES.
7. **CLOSE THE CURSOR OBJECT AND DATABASE CONNECTION OBJECT.** USE CURSOR.CLOSE() AND CONNECTION.CLOSE() METHOD TO CLOSE THE CURSOR AND SQLITE CONNECTIONS AFTER YOUR WORK COMPLETES.

https://pynative.com/python-sqlite-insert-into-table/

RETRIEVING DATA FROM A DATABASE

- using parameter in SQL statement

```
def insert_table2(conn,param):
    #Creating a cursor object using the cursor() method
    cursor = conn.cursor()
    sql = "insert into login values (? , ? , ? , ?)"
    cursor.execute(sql,param)            #1: using param
    cursor.execute(sql,[v1,v2,v3,v4])    #2: using [] all case
    cursor.execute(sql,(v1,v2,v3,v4))    #3: using () > 1 value
```

RETRIEVING DATA FROM A DATABASE

- using parameter in SQL statement

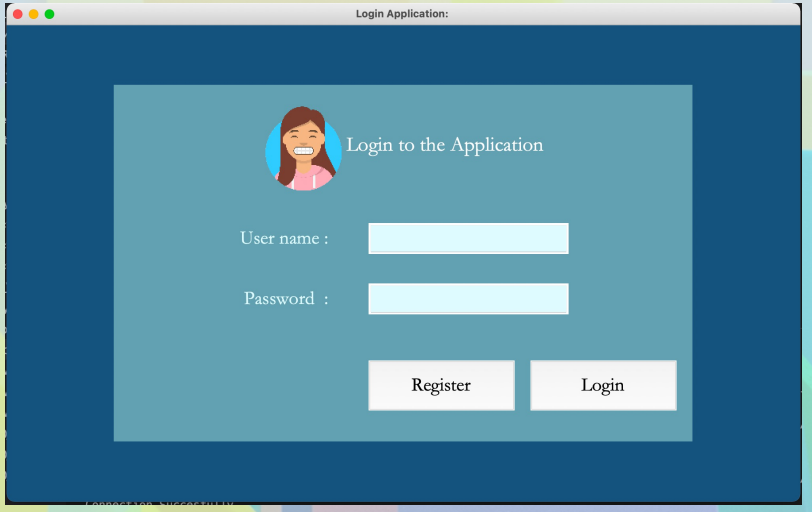
```
def insert_table2(conn,param):
    #Creating a cursor object using the cursor() method
    cursor = conn.cursor()
    sql = '''
        insert into login values (? , ? , ? , ?)


    '''
    cursor.execute(sql,param)
```

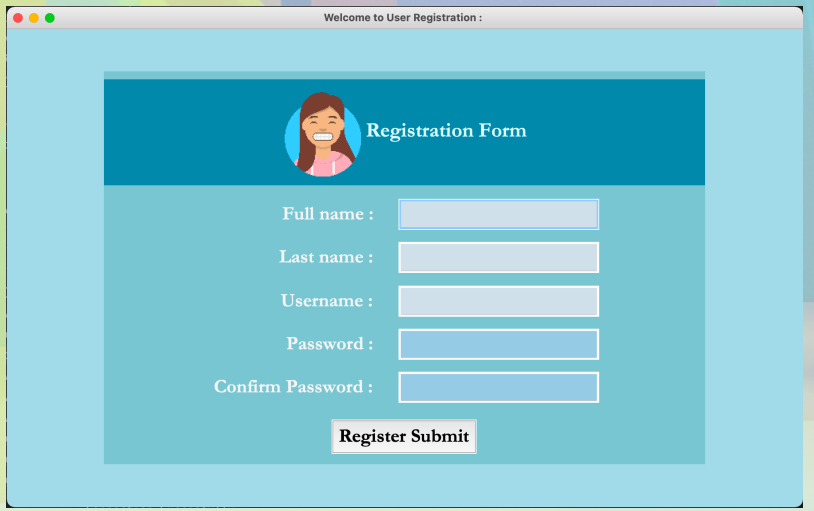
COMMIT() METHOD

CONNECTION.COMMIT()

- THIS METHOD COMMITS THE CURRENT TRANSACTION. IF YOU DON'T CALL THIS METHOD, ANYTHING YOU DID SINCE THE LAST CALL TO COMMIT() IS NOT VISIBLE FROM OTHER DATABASE CONNECTIONS.

Class Activity of Week11





Register Login/ Application

See you next week