University Database

Database Systems and SQL

CSCI-2215-01/6622-01

Analytical Control

University of New Haven

TAGLIATELA COLLEGE OF ENGERNEERING, West Haven, CT

Submitted to:

Dr. Reza Sadeghi

Spring 2021

Analytical Control

Analytical Control

Hugh-John Dunkley: hdunk1@unh.newhaven.edu
Rogan Gopi: rgopi1@unh.newhaven.edu
Grace Mandada: gmand3@unh.newhaven.edu
Bhargavi Gottumukkula: gbhar1@unh.newhaven.edu
Samuel Mandada: smand17@unh.newhaven.edu

Team Roles

- Hugh-John Dunkley

  Team Leader, coordinator, researcher, designer

- Rogan Gopi

  Developer, Graphical User interface designer

- Grace Mandada

  Graphical user interface connector and researcher

- Samuel Mandada

  Graphical user interface connector and researcher

- Bhargavi Gottumukkula

  Database manager and researcher

Table of Contents

INTRODUCTION:

The main goal of our project is to create a university database which contains the details about the Students, Faculties, departments, Courses using MySQL and in order to access the details or search them efficiently, GUI was created using Tkinter (Python).

ENTITY RELATIONSHIP MODEL:

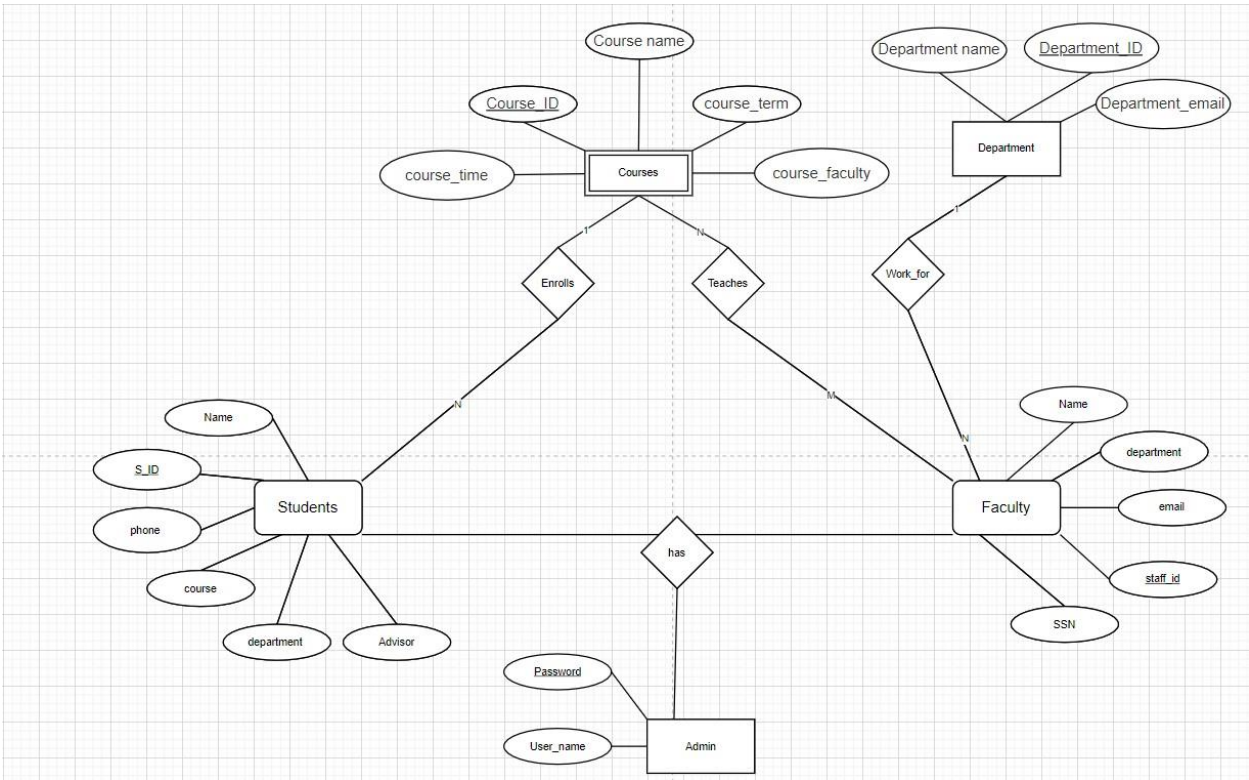


Fig:1 Entity relationship model for univ_database

The entity relationship model depicts how each and other entities were related to and it give visual idea how the entities in the database were connected with some specific details such as weak entity, strong entity, and so on.

CREATING DATABASE IN THE MYSQL AT A GLANCE:

The MySQL software was downloaded from MySQL community server with the specific operating system selected. As the next step is to install MySQL workbench to work with the database. In fact there are many ways to acces the SQL server like terminal, command prompt, we explicitly installed workbench to work hassle free and for more GUI experience.

Once all these were installed, we started creating the database named "univ-database" in our local instance. The command used to create the database is : "create database univ_database;"

But before that we have to double check whether any other database was created in that same name. To simply check type, "show database;"

CREATING STUDENT TABLE AND ADDING VALUES:

```
CREATE TABLE student(
id VARCHAR(20) NOT NULL,
name VARCHAR(20) NOT NULL,
phone VARCHAR(10) NOT NULL,
course VARCHAR(20) NOT NULL,
department VARCHAR(20) NOT NULL,
Advisor VARCHAR(20) NOT NULL);
```

INSERTING VALUES TO THE STUDENT TABLE:

```
insert into student values(001,'mike',2034556677,'database','computer
science','michael')
,(002,'steven',2034556678,'hacking','Cyber Security','mark')
,(003,'lewis',2034556679,'statistics','industrial','john')
,(004,'stuart',2034556680,'const_modeling','civil','george')
,(005,'stuart',2034556681,'cell biology','Bio Technology','greg');
```

CREATING FACULTY TABLE AND ADDING VALUES:

```
CREATE TABLE faculty(
staff_id VARCHAR(20) NOT NULL,
name VARCHAR(20) NOT NULL,
ssn VARCHAR(20) NOT NULL,
department VARCHAR(20) NOT NULL,
email VARCHAR(20) NOT NULL);
```

INSERTING VALUES TO THE FACULTY TABLE:

```
insert into faculty values(101,'michael',56455467,'computer
science','mic@gmail.edu')
,(102,'mark',99886545,'Cyber security','mark@gmail.edu')
,(103,'john',22324356,'industrial','john@gmail.edu')
,(104,'george',76568900,'civil','george@gmail.edu')
,(105,'greg',67586758,'Bio Technology','greg@gmail.edu');
```

CREATING DEPARTMENT TABLE AND ADDING VALUES:

```
CREATE TABLE department(
dept_id VARCHAR(20) NOT NULL,
dept_name VARCHAR(20) NOT NULL,
dept_email VARCHAR(20) NOT NULL);
```

INSERTING VALUES TO THE DEPARTMENT TABLE:

```
insert into department values(200,'computer science','cs@info.edu')
,(201,'cyber security','cyber@info.edu')
,(203,'industrial','indust@info.edu')
,(204,'civil','civil@info.edu')
,(205,'Bio Technology','bio@info.edu');
```

CREATING COURSE TABLE AND ADDING VALUES:

```
course_id VARCHAR(20) NOT NULL,
course_name VARCHAR(20) NOT NULL,
course_term VARCHAR(20) NOT NULL,
course_faculty VARCHAR(20) NOT NULL,
course_time VARCHAR(20)NOT NULL);
```

INSERTING VALUES TO THE COURSE TABLE:

```
insert into course values(601,'database','spring','michael','3:30-6:30')
,(602,'hacking','fall','mark','1:30-3:30')
,(603,'statistice','spring','john','2:30-5:30')
,(604,'const_modeling','spring','george','10:30-2:30')
,(605,'cell biology','fall','greg','8:30-10:30');
```

The step is to link the database with the python. The command used to link is :

"pip3 install mysql-connector-python"

A sample code was written to check the database was linked to the python.

```
import mysql.connector
db =mysql.connector.connect(
    host='localhost',
    user='root',
    password='rootroot',
    database='univ_database'
    )
```

```
mycursor = db.cursor()
mycursor.execute('select count(*) from students;')
for x in mycursor:
    print(x)
```

Upon success connection, we started writing the python program to implement the GUI.

PYTHON CODE EXPLANATION:

We have used programming language called “python” to build a database system. Python offers various utilities to design the GUI (Graphical User Interface), and one such utility is Tkinter. Using Tkinter, we have designed the GUI for the university database system.

The outline of university database management system will look like this

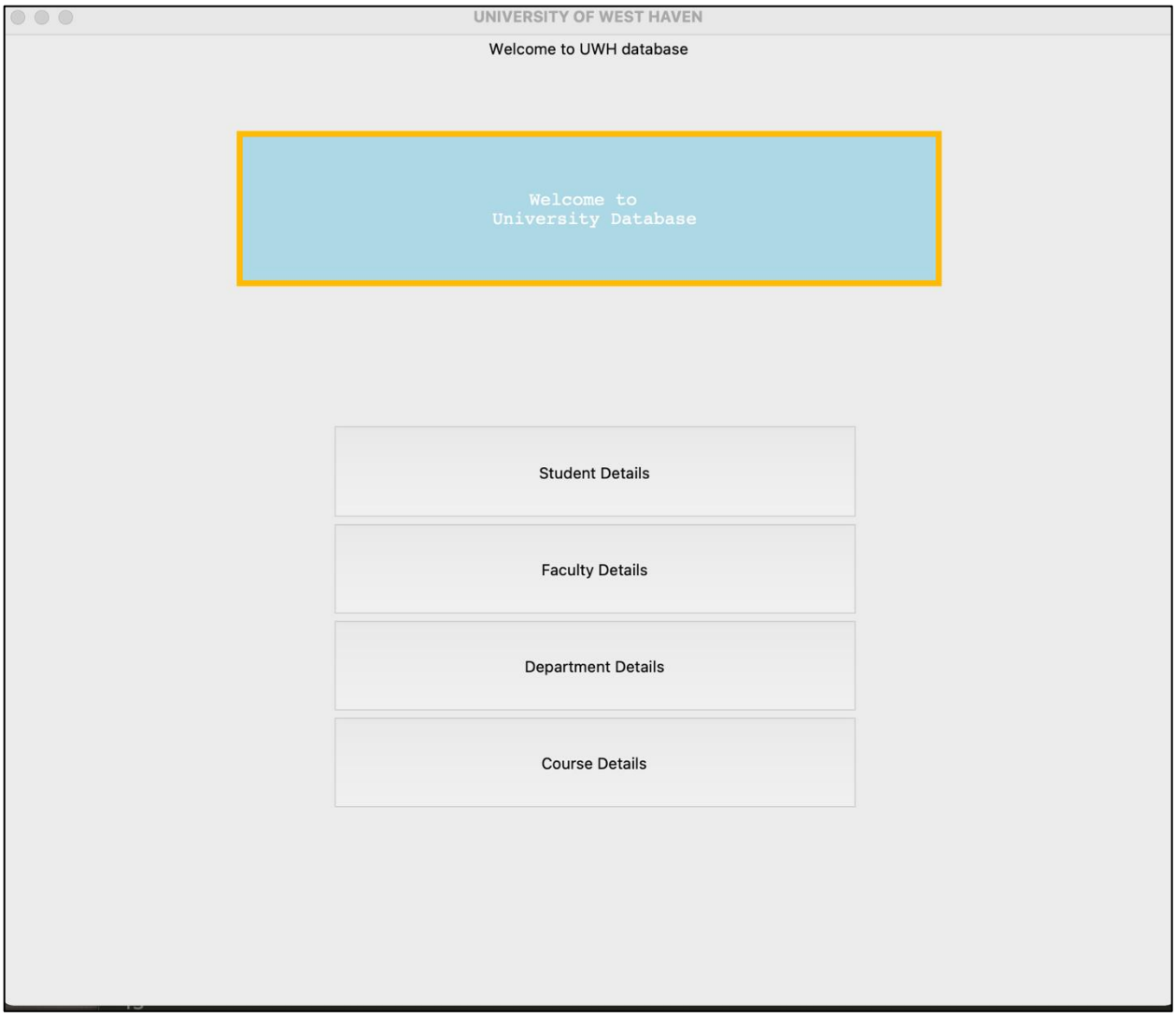


Fig: - University Database System

PROJECT PERQUISITES:

To use the tkinter utility in our python code, we have installed tkinter library in our system. Install Tkinter by using this command,

"Pip install Tkinter "

To communicate with our database system through python code, we have installed MySQL connector library. Install MySQL by using this command,

"Pip install mysql"

"pip3 install mysql-connector-python"

Database creation:

Before getting into the python code, we are required to install the MySQL Server database on our system to make mysql library work.

MySQL Server Installation

Download Link - https://dev.mysql.com/downloads/mysql/

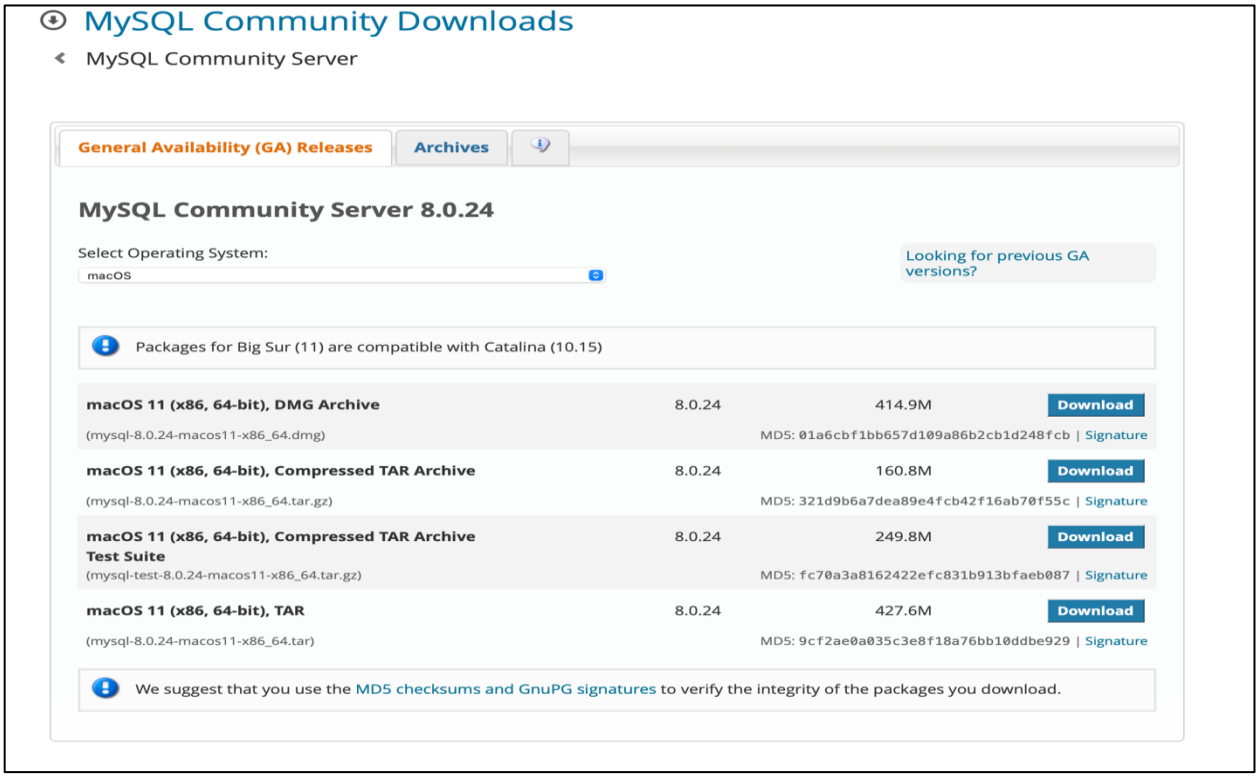


Fig: - MySQL Community server

MYSQL WORKBENCH INSTALLATION:

Download Link - https://dev.mysql.com/downloads/workbench/

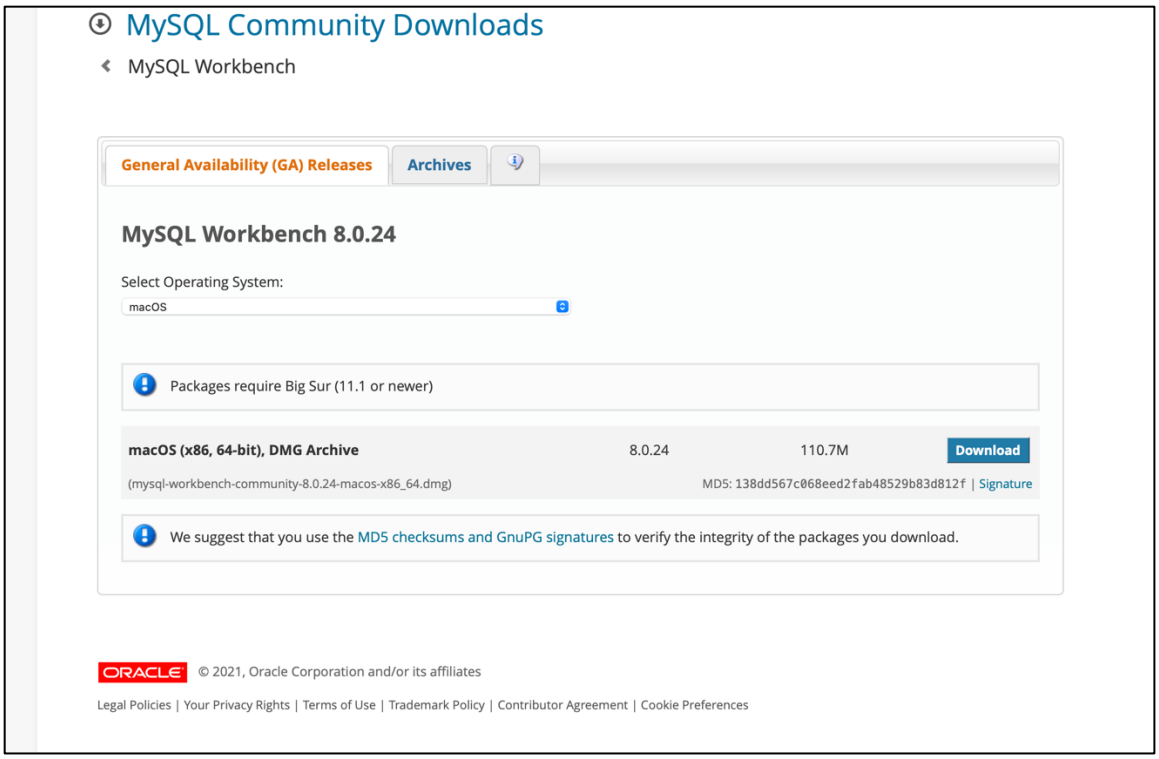
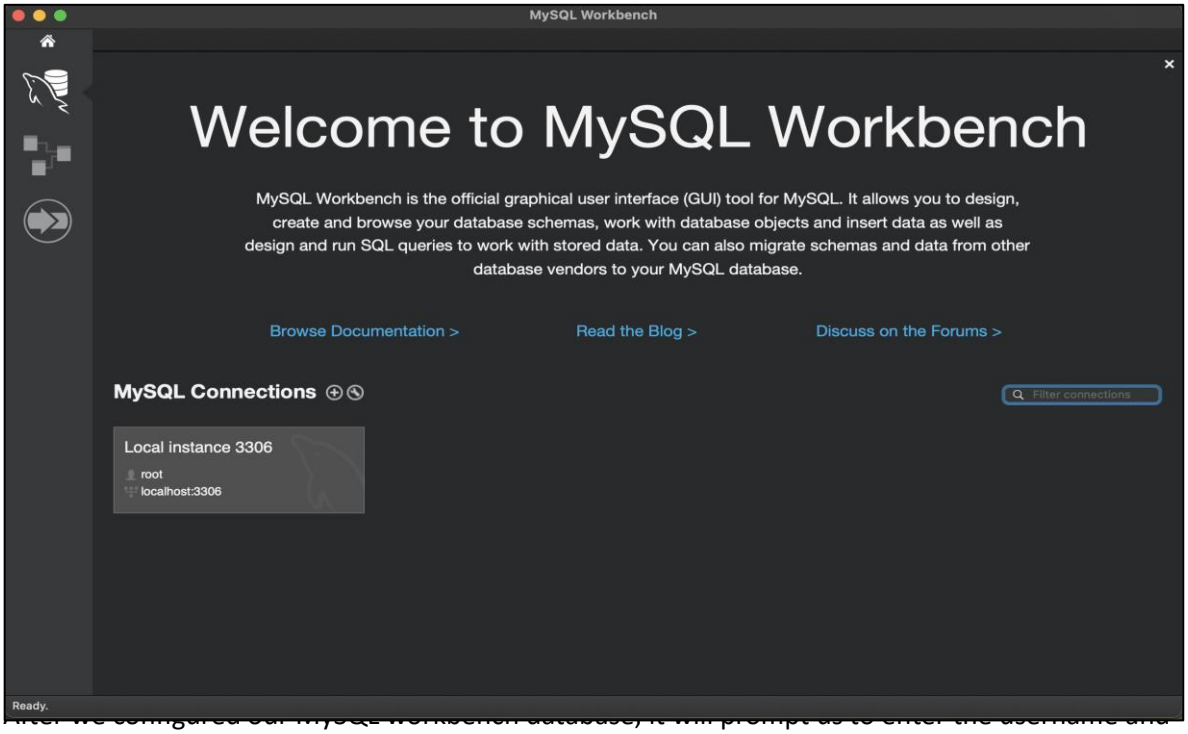
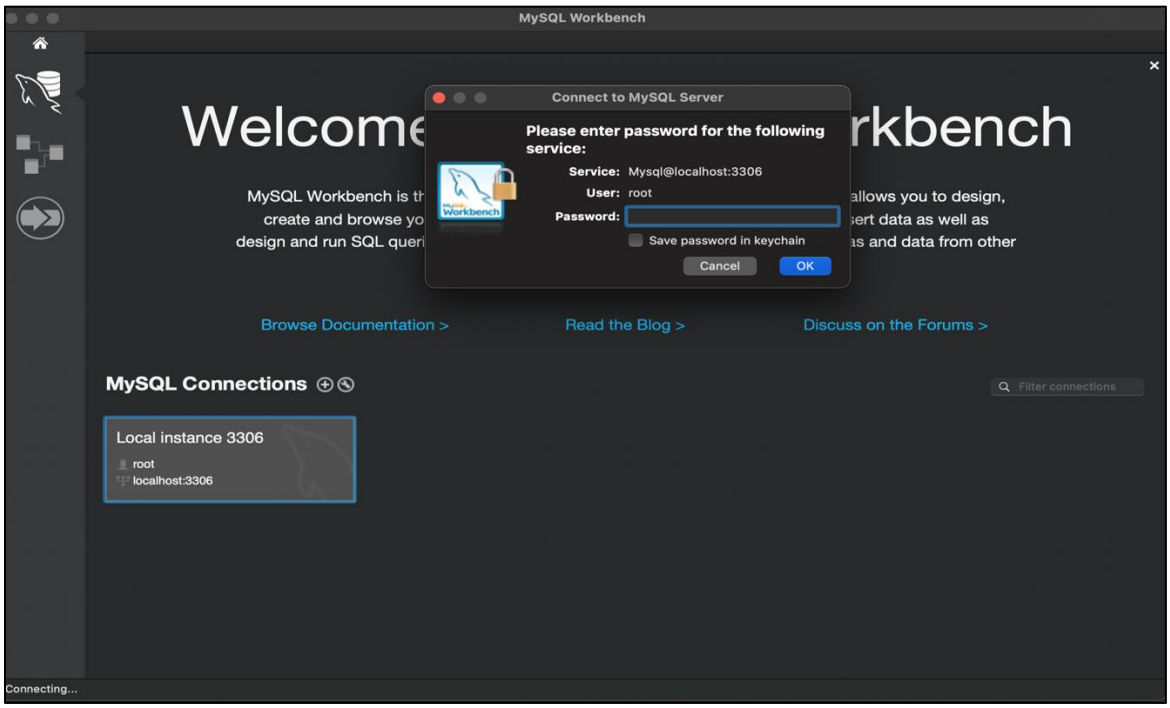
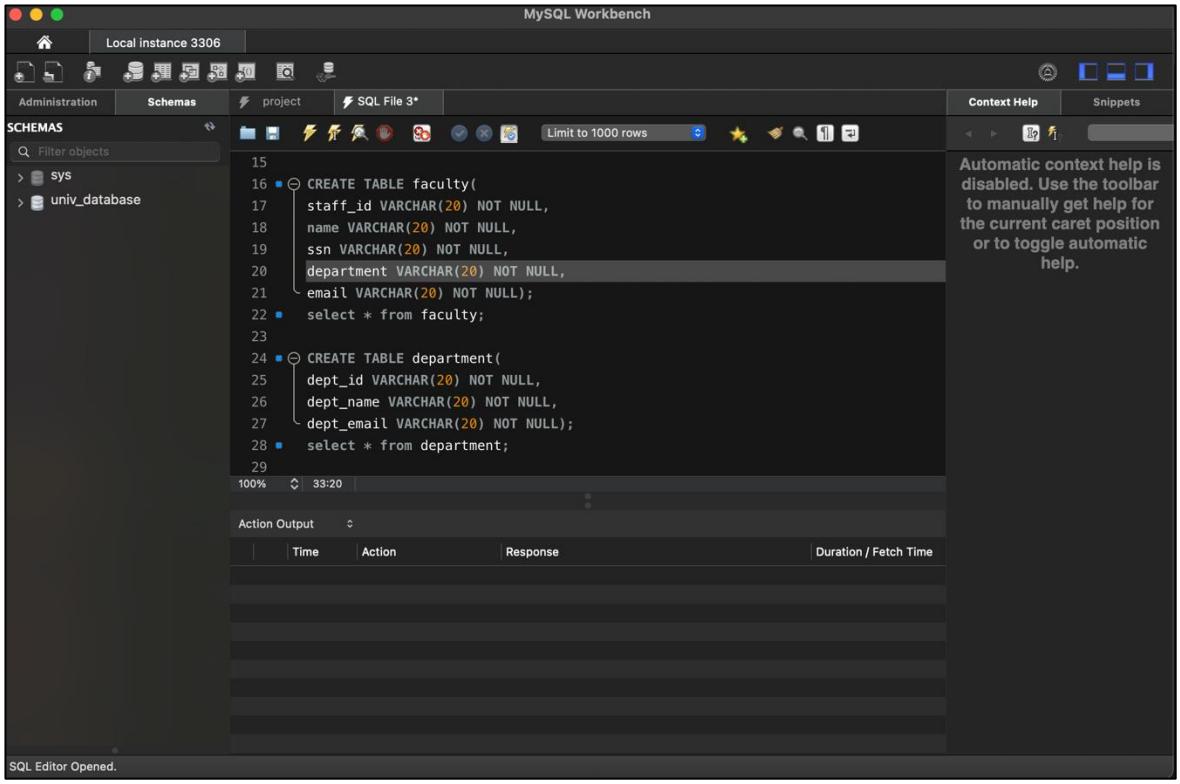


Fig: - MySQL Workbench



After we configured our MySQL workbench database, it will prompt us to enter the username and password to login into our DB instance.


MySQL Workbench
Welcome
rkbench
MySQL Workbench is th
create and browse yo
design and run SQL queri
allows you to design,
ert data as well as
as and data from other
Connect to MySQL Server
Please enter password for the following service:
Service: Mysql@localhost:3306
User: root
Password:
Save password in keychain
Cancel
OK
Browse Documentation >
Read the Blog >
Discuss on the Forums >
MySQL Connections
Filter connections
Local instance 3306
root
localhost:3306
Connecting...


MySQL Workbench
Local instance 3306
Administration
Schemas
project
SQL File 3*
Context Help
Snippets
SCHEMAS
Filter objects
sys
univ_database
Limit to 1000 rows
15
16 CREATE TABLE faculty(
17 staff_id VARCHAR(20) NOT NULL,
18 name VARCHAR(20) NOT NULL,
19 ssn VARCHAR(20) NOT NULL,
20 department VARCHAR(20) NOT NULL,
21 email VARCHAR(20) NOT NULL);
22 select * from faculty;
23
24 CREATE TABLE department(
25 dept_id VARCHAR(20) NOT NULL,
26 dept_name VARCHAR(20) NOT NULL,
27 dept_email VARCHAR(20) NOT NULL);
28 select * from department;
29
100%
33:20
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Action Output
Time
Action
Response
Duration / Fetch Time
SQL Editor Opened.

We have created the database called “univ_database”. We also created 4 tables in our database for each section i.e.

1. Student
2. Faculty
3. Department
4. Course

Creating database and tables using the following MySQL Command

```
show databases;
use univ_database;
show tables;
drop table student;
CREATE TABLE student(
id VARCHAR(20) NOT NULL,
name VARCHAR(20) NOT NULL,
phone VARCHAR(10) NOT NULL,
course VARCHAR(20) NOT NULL,
department VARCHAR(20) NOT NULL,
Advisor VARCHAR(20) NOT NULL);
desc student;
select * from student;
drop table faculty;

CREATE TABLE faculty(
staff_id VARCHAR(20) NOT NULL,
name VARCHAR(20) NOT NULL,
ssn VARCHAR(20) NOT NULL,
department VARCHAR(20) NOT NULL,
email VARCHAR(20) NOT NULL);
select * from faculty;

CREATE TABLE department(
dept_id VARCHAR(20) NOT NULL,
```

```
dept_name VARCHAR(20) NOT NULL,

dept_email VARCHAR(20) NOT NULL);

select * from department;


CREATE TABLE course(

course_id VARCHAR(20) NOT NULL,

course_name VARCHAR(20) NOT NULL,

course_term VARCHAR(20) NOT NULL,

course_faculty VARCHAR(20) NOT NULL,

course_time VARCHAR(20)NOT NULL);

select * from course;
```

Description of Project Files

Below are the project files you will get once you download and extract the project file:

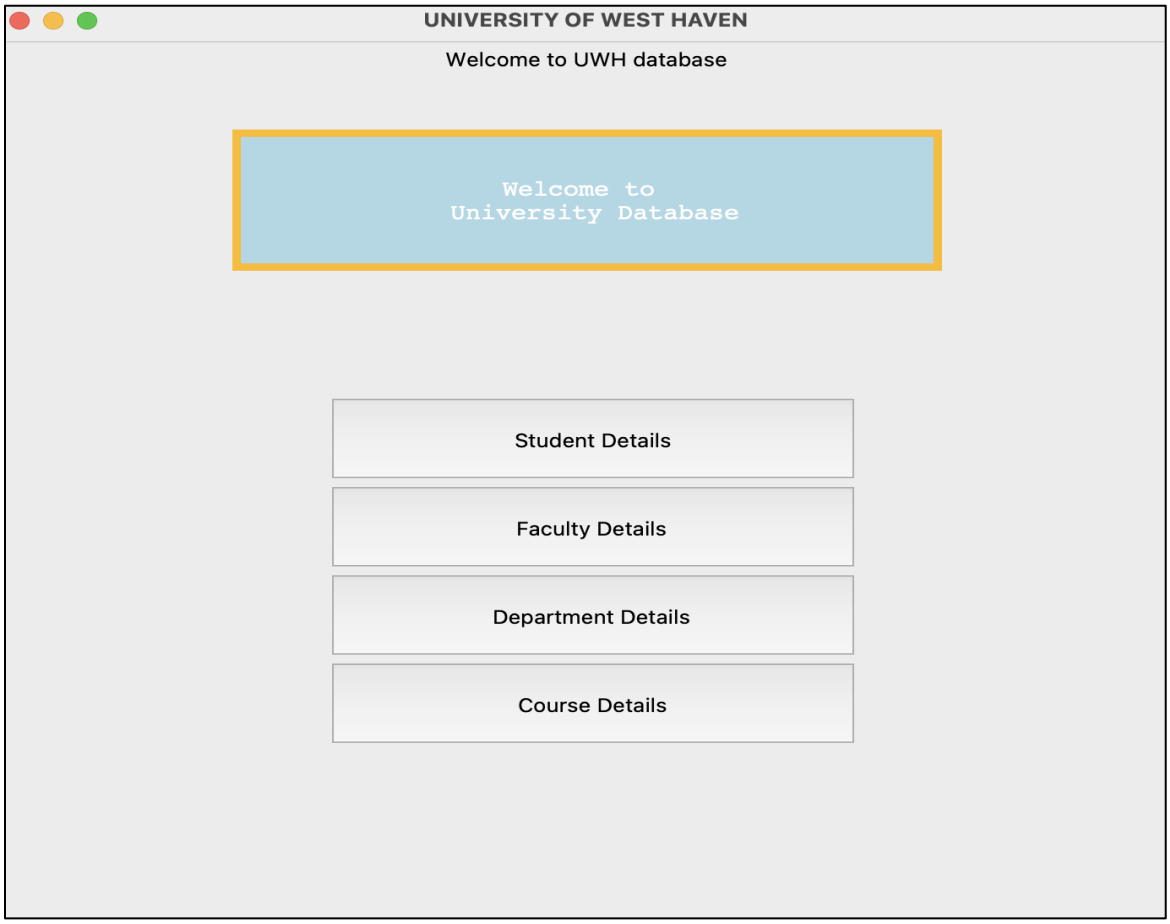
- **test.py** – which does function call to all other python files
- **student.py** – CRUD operation for the students
- **faculty.py** – CRUD operation for the faculty
- **course.py** – CRUD operation for the course
- **department.py** – CRUD operation for the department

test.py

```
from tkinter import *
from PIL import ImageTk,Image
from tkinter import messagebox
import mysql.connector as mysql
from login import *
from faculty import *
from department import *
from course import *

mypass = "rootroot"
mydatabase="univ_database"

con =
mysql.connect(host="localhost",user="root",password=mypass,database=mydatabas
e)
cur = con.cursor()

root = Tk()
root.title("UNIVERSITY OF WEST HAVEN")
```

```
root.minsize(width=500,height=500)
root.geometry("700x600")

label = Label(root, text = "Welcome to UWH database",fg = "black").pack()


#icon = PhotoImage(file = "/Users/gokulsd/Desktop/Rogan/Gokul/image.jpeg")
#label = Label(root, image = icon)
#xqlabel.pack()


headingFrame1 = Frame(root,bg="#FFBB04",bd=5)
headingFrame1.place(relx=0.2,rely=0.1,relwidth=0.6,relheight=0.16)

headingLabel = Label(headingFrame1, text="Welcome to \n University Database",
bg='light blue', fg='white', font=('Courier',15))
headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)

btn1 = Button(root,text="Student Details",bg='red', fg='black',command=stud)
btn1.place(relx=0.28,rely=0.4, relwidth=0.45,relheight=0.1)

btn2 = Button(root,text="Faculty Details",bg='black',
fg='black',command=fac_login)
btn2.place(relx=0.28,rely=0.5, relwidth=0.45,relheight=0.1)

btn3 = Button(root,text="Department Details",bg='black',
fg='black',command=dep_login)
btn3.place(relx=0.28,rely=0.6, relwidth=0.45,relheight=0.1)

btn4 = Button(root,text="Course Details",bg='black',
fg='black',command=course_login)
btn4.place(relx=0.28,rely=0.7, relwidth=0.45,relheight=0.1)



root.mainloop()
```

Output:



Student.py

```
from tkinter import *
from tkinter import messagebox
import mysql.connector as mysql

def success():
        uname= e1.get()
        password=e2.get()

        if(uname == "" and password == ""):
                messagebox.showinfo("", "Please fill all fields")

        elif(uname == "Admin" and password == "12345"):
                messagebox.showinfo("", "Login Success")
                details();

        else:
                messagebox.showinfo("", "Incorrect username and password")
```

```
def details():

        global e_id,e_name,e_phone,e_course,e_dept,e_advisor

        root = Tk()
        root.title("UNIVERSITY OF NEW HAVEN")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        ID = Label(root,text="Enter ID").place(x=20,y=30)

        name = Label(root,text="Enter Name").place(x=20,y=60)

        phone = Label(root,text="Enter Mobile No.").place(x=20,y=90)

        course = Label(root,text="Enter Course").place(x=20,y=120)

        department = Label(root,text="Enter Department").place(x=20,y=150)

        Advisor = Label(root,text="Enter Advisor Name").place(x=20,y=180)

        e_id = Entry(root)
        e_id.place(x=150, y=30)

        e_name = Entry(root)
        e_name.place(x=150, y=60)

        e_phone = Entry(root)
        e_phone.place(x=150, y=90)

        e_course = Entry(root)
        e_course.place(x=150, y=120)

        e_dept = Entry(root)
        e_dept.place(x=150, y=150)

        e_advisor = Entry(root)
        e_advisor.place(x=150, y=180)


        Button (root,text= "insert",command=insert).place(x=10, y=240)

        Button (root,text= "delete",command=delete).place(x=80, y=240)

        Button (root,text= "update",command=update).place(x=150, y=240)

        Button (root,text= "get",command=get).place(x=230, y=240)

        #list= Listbox(root)
        #list.place(x=390,y=30)
        #show();

        root.mainloop()


def insert():
```

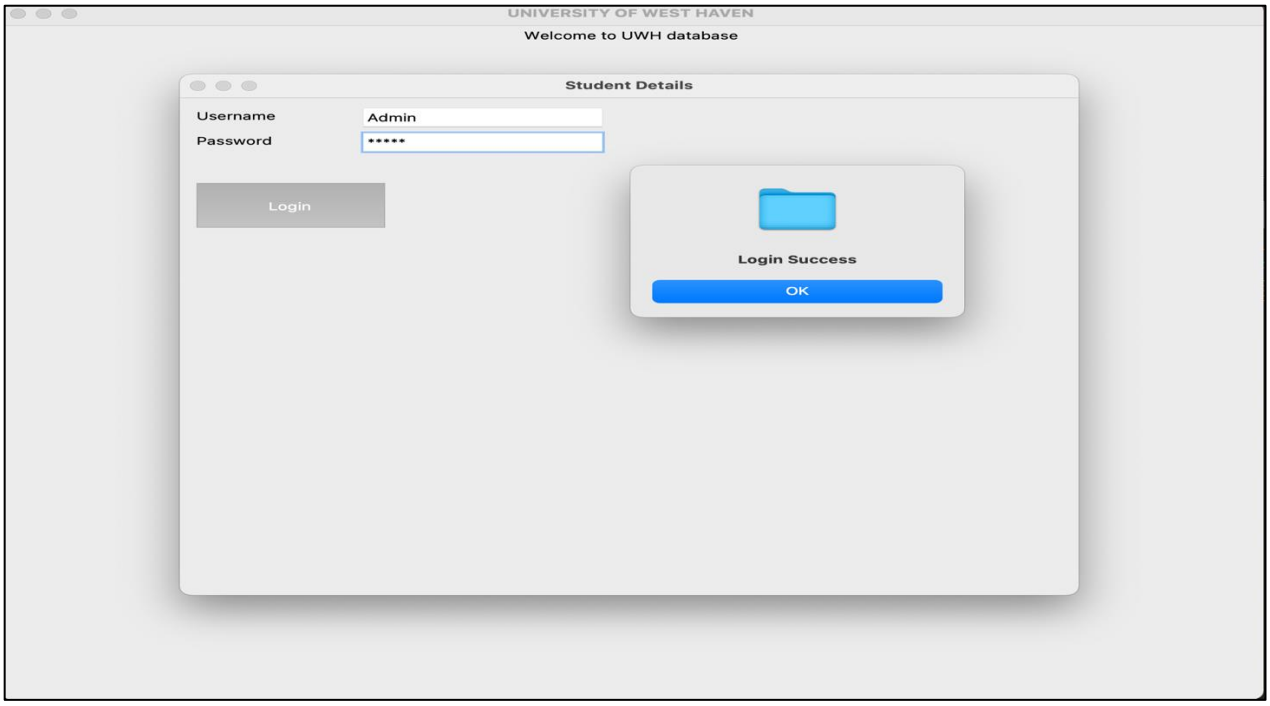
```
                                ID =e_id.get()
                                name=e_name.get()
                                phone=e_phone.get()
                                course=e_course.get()
                                department=e_dept.get()
                                Advisor=e_advisor.get()

                                if(ID =="" or name=="" or phone=="" or
course==""or department==""or Advisor==""):

        messagebox.showinfo("Insert Status","All fields are required");
                                else:
                                                                con =
mysql.connect(host="localhost", user="root", password="rootroot",
database="univ_database")
                                                                cursor=
con.cursor()

        cursor.execute("insert into student values('"+ ID +"','"+ name +"',
'"+ phone +"','"+ course +"','"+department +"','"+ Advisor +"')")

        cursor.execute("commit")
                                                                con.close()

def update():

        ID =e_id.get()
        name=e_name.get()
        phone=e_phone.get()
        course=e_course.get()
        department=e_dept.get()
        Advisor=e_advisor.get()


        if(ID =="" or name=="" or phone==""):
                messagebox.showinfo("Update Status","All fields are
required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("update student set
name='"+name+"',phone='"+phone+"',course='"+course+"',department='"+departmen
t+"',Advisor='"+Advisor+"' where id = '"+ ID +"'")
                cursor.execute("commit")

                e_id.delete(0,'end')
                e_name.delete(0,'end')
                e_phone.delete(0,'end')
                e_course.delete(0,'end')
                e_dept.delete(0,'end')
                e_advisor.delete(0,'end')
                messagebox.showinfo("Update status","Update Successfully")
                con.close()
```

```
def get():
        if(e_id.get()==""):
                messagebox.showinfo("Fetch status", "ID is compulsory to fetch
details")
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("select * from student where id = '"+
e_id.get() +"'")
                rows = cursor.fetchall()

                for row in rows:
                        e_name.insert(0, row[1])
                        e_phone.insert(0, row[2])
                        e_course.insert(0, row[3])
                        e_dept.insert(0, row[4])
                        e_advisor.insert(0, row[5])

                con.close()


def delete():
                                ID = e_id.get()
                                #name=e_name.get()
                                #phone=e_phone.get()

                                if(ID== ""):

        messagebox.showinfo("Delete Status","ID is required");
                                else:
                                                                        con =
mysql.connect(host="localhost", user="root", password="rootroot",
database="univ_database")
                                                                        cursor=
con.cursor()

        cursor.execute("delete from student where id = '"+ ID +"'")

        cursor.execute("commit")
                                                                        con.close()

def stud():

        root = Tk()
        root.title("Student Details")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        global e1,e2

        uname = Label(root,text="Username").place(x=10,y=10)

        password = Label(root,text="Password").place(x=10,y=40)


        e1 = Entry(root)
```

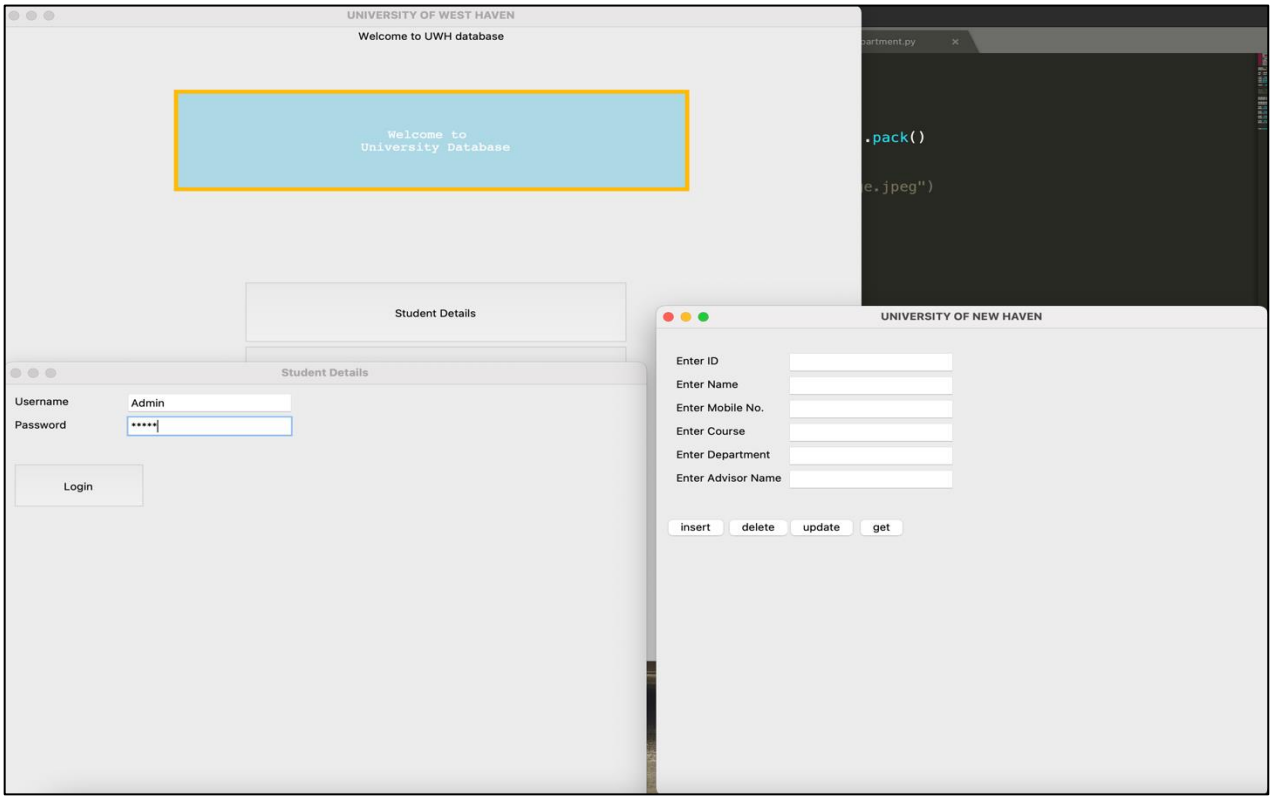
```
        e1.place(x=140, y=10)

        e2 = Entry(root)
        e2.place(x=140, y=40)
        e2.config(show="*")

        Button(root, text="Login", command=success, height =3, width =
13).place(x=10,y=100)


        root.mainloop()
```

Output –

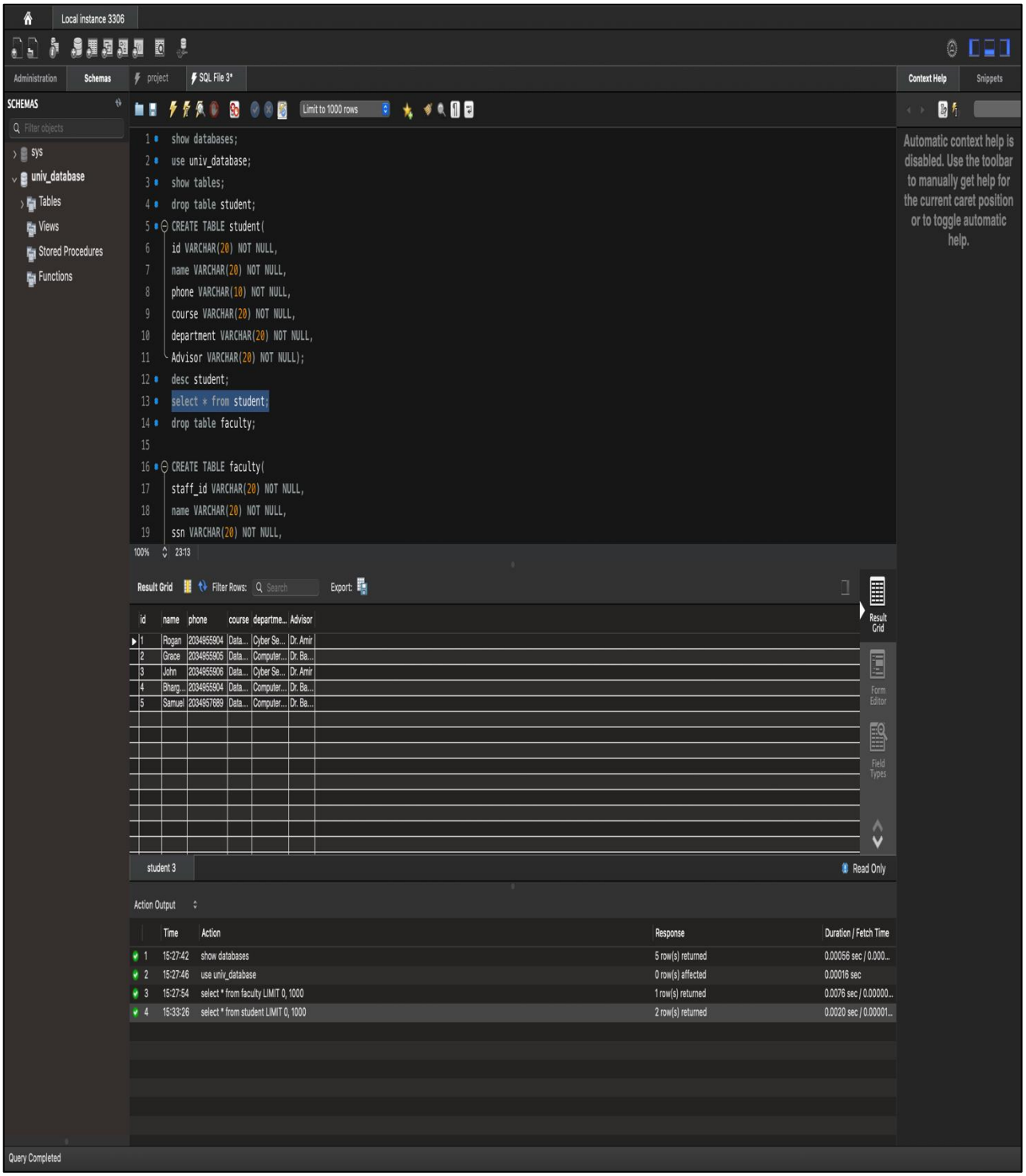
Login Page



CRUD Operation in Student Section



DB Result


Local instance 3306
Administration
Schemas
project
SQL File 3*
Context Help
Snippets
SCHEMAS
Filter objects
sys
univ_database
Tables
Views
Stored Procedures
Functions
Limit to 1000 rows
1 show databases;
2 use univ_database;
3 show tables;
4 drop table student;
5 CREATE TABLE student(
6 id VARCHAR(20) NOT NULL,
7 name VARCHAR(20) NOT NULL,
8 phone VARCHAR(10) NOT NULL,
9 course VARCHAR(20) NOT NULL,
10 department VARCHAR(20) NOT NULL,
11 Advisor VARCHAR(20) NOT NULL);
12 desc student;
13 select * from student;
14 drop table faculty;
15
16 CREATE TABLE faculty(
17 staff_id VARCHAR(20) NOT NULL,
18 name VARCHAR(20) NOT NULL,
19 ssn VARCHAR(20) NOT NULL,
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
100% 23:13
Result Grid
Filter Rows:
Search
Export:
id name phone course departme... Advisor
1 Rogan 2034955904 Data... Cyber Se... Dr. Amir
2 Grace 2034955905 Data... Computer... Dr. Ba...
3 John 2034955906 Data... Cyber Se... Dr. Amir
4 Bharg... 2034955904 Data... Computer... Dr. Ba...
5 Samuel 2034957689 Data... Computer... Dr. Ba...
Result Grid
Form Editor
Field Types
student 3
Read Only
Action Output
Time Action Response Duration / Fetch Time
1 15:27:42 show databases 5 row(s) returned 0.00056 sec / 0.000...
2 15:27:46 use univ_database 0 row(s) affected 0.00016 sec
3 15:27:54 select * from faculty LIMIT 0, 1000 1 row(s) returned 0.0076 sec / 0.00000...
4 15:33:26 select * from student LIMIT 0, 1000 2 row(s) returned 0.0020 sec / 0.00001...
Query Completed

Faculty.py

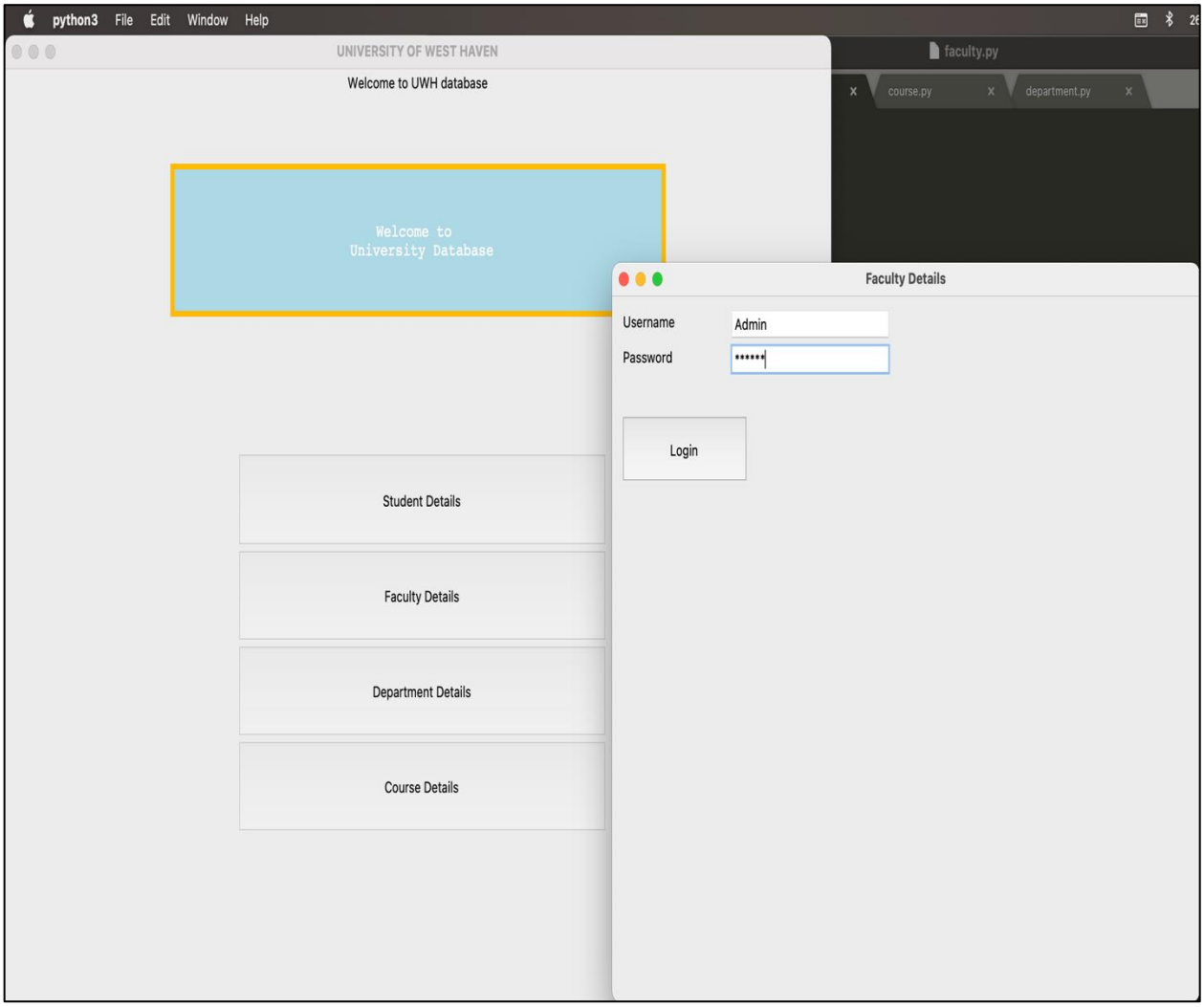
```
from tkinter import *
from tkinter import messagebox
import mysql.connector as mysql

def identity():
        uname= e1.get()
        password=e2.get()

        if(uname == "" and password == ""):
                messagebox.showinfo("", "Please fill all fields")

        elif(uname == "Admin" and password == "123456"):
                messagebox.showinfo("", "Login Success")
                details();

        else:
                messagebox.showinfo("", "Incorrect username and password")


def fac_login():

        root = Tk()
        root.title("Faculty Details")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        global e1,e2

        uname = Label(root,text="Username").place(x=10,y=10)

        password = Label(root,text="Password").place(x=10,y=40)


        e1 = Entry(root)
        e1.place(x=140, y=10)

        e2 = Entry(root)
        e2.place(x=140, y=40)
        e2.config(show="*")

        Button(root, text="Login", command=identity, height =3, width =
13).place(x=10,y=100)


        root.mainloop()



def details():

        global f_staffid,f_name,f_ssn,f_dept,f_email

        root = Tk()
```

```
        root.title("UNIVERSITY OF NEW HAVEN")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        staff_id = Label(root,text="Enter Staff ID").place(x=20,y=30)

        name = Label(root,text="Enter Name").place(x=20,y=60)

        ssn = Label(root,text="Enter SSN ").place(x=20,y=90)

        department = Label(root,text="Enter Department").place(x=20,y=120)

        email = Label(root,text="Enter Email").place(x=20,y=150)

        f_staffid = Entry(root)
        f_staffid.place(x=150, y=30)

        f_name = Entry(root)
        f_name.place(x=150, y=60)

        f_ssn = Entry(root)
        f_ssn.place(x=150, y=90)
        f_ssn.config(show="*")

        f_dept = Entry(root)
        f_dept.place(x=150, y=120)

        f_email = Entry(root)
        f_email.place(x=150, y=150)


        Button (root,text= "insert",command=insert).place(x=10, y=240)

        Button (root,text= "delete",command=delete).place(x=80, y=240)

        Button (root,text= "update",command=update).place(x=150, y=240)

        Button (root,text= "get",command=get).place(x=230, y=240)

        #list= Listbox(root)
        #list.place(x=390,y=30)
        #show();

        root.mainloop()


def insert():
        staff_id =f_staffid.get()
        name=f_name.get()
        ssn=f_ssn.get()
        department=f_dept.get()
        email=f_email.get()

        if(staff_id =="" or name=="" or ssn==""or department==""or email==""):
                messagebox.showinfo("Insert Status","All fields are
required");
        else:
```

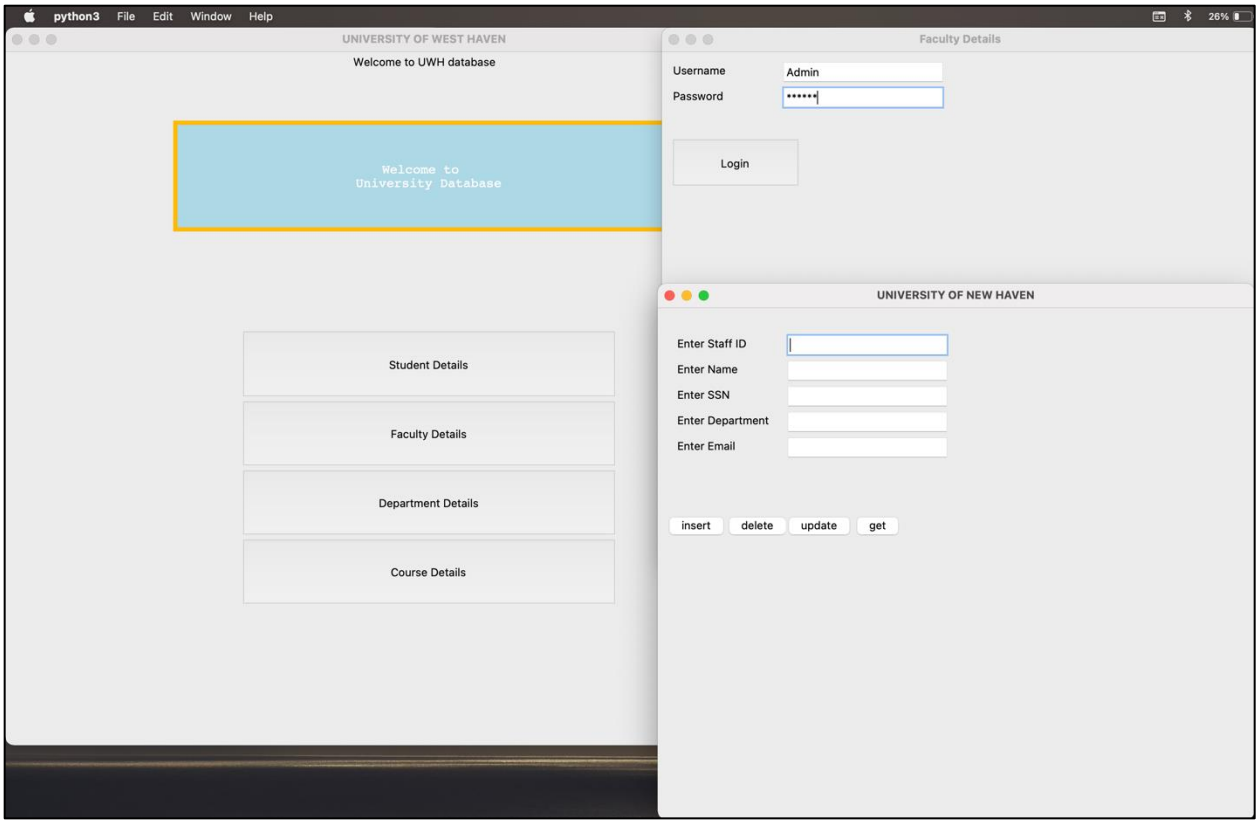
```
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("insert into faculty values('"+ staff_id
+"','"+ name +"', '"+ssn +"','"+ department +"','"+email +"')")
                cursor.execute("commit")
                con.close()

def update():

        staff_id =f_staffid.get()
        name=f_name.get()
        ssn=f_ssn.get()
        department=f_dept.get()
        email=f_email.get()


        if(staff_id =="" or name=="" or ssn=="" or department=""or
email==""):
                messagebox.showinfo("Update Status","All fields are
required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("update faculty set
staff_id='"+staff_id+"',name='"+name+"',ssn='"+ssn+"',department='"+departmen
t+"',email='"+email+"' where staff_id = '"+ staff_id+"'")
                cursor.execute("commit")

                f_staffid.delete(0,'end')
                f_name.delete(0,'end')
                f_ssn.delete(0,'end')
                f_dept.delete(0,'end')
                f_email.delete(0,'end')
                messagebox.showinfo("Update status","Update Successfully")
                con.close()


def delete():
        staff_id = f_staffid.get()
        if(staff_id== ""):
                messagebox.showinfo("Delete Status","ID is required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("delete from faculty where staff_id = '"+
staff_id +"'")
                cursor.execute("commit")
                con.close()



def get():
        if(f_staffid.get()==""):
```

```
                messagebox.showinfo("Fetch status", "ID is compulsory to fetch
details")
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("select * from faculty where staff_id = '"+
f_staffid.get() +"'")
                rows = cursor.fetchall()

                for row in rows:
                        f_name.insert(0, row[1])
                        f_ssn.insert(0, row[2])
                        f_dept.insert(0, row[3])
                        f_email.insert(0, row[4])

                con.close()
```

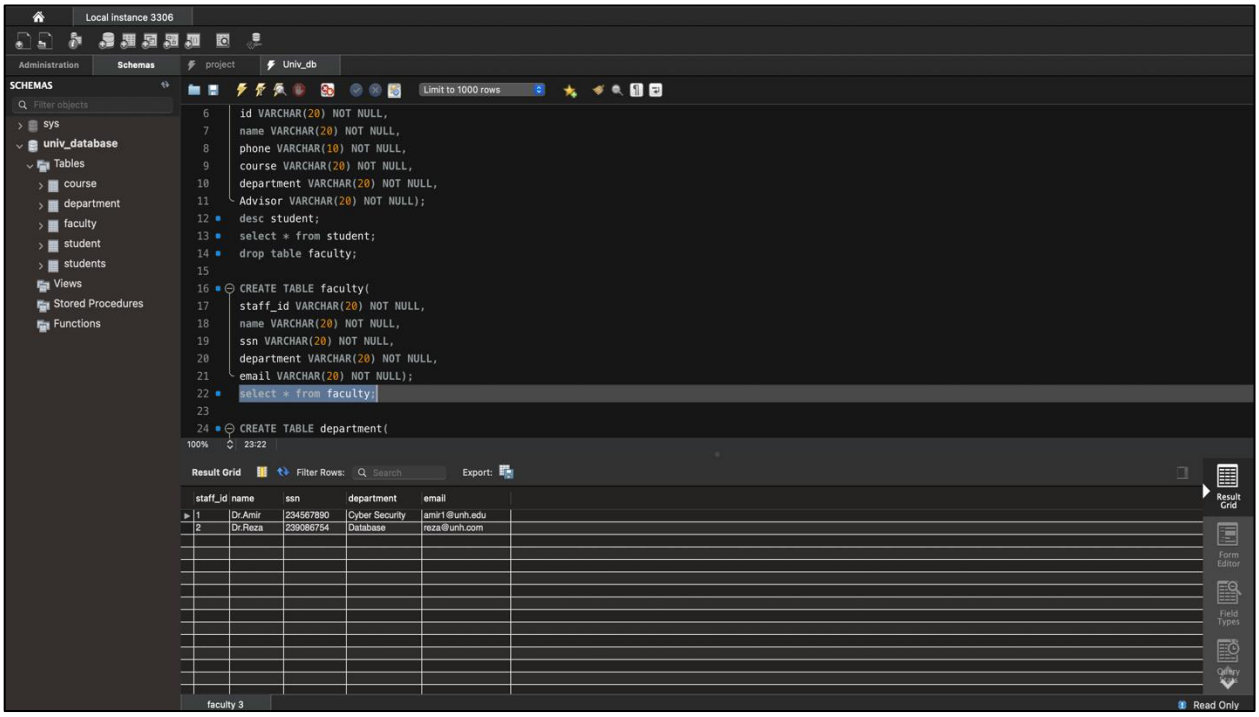
Output



CRUD Operation in Faculty Section:



DB Output



Department.py

```
from tkinter import *
from tkinter import messagebox
import mysql.connector as mysql



def identity():
        uname= e1.get()
        password=e2.get()

        if(uname == "" and password == ""):
                messagebox.showinfo("", "Please fill all fields")

        elif(uname == "Admin" and password == "1234"):
                messagebox.showinfo("", "Login Success")
                details();

        else:
                messagebox.showinfo("", "Incorrect username and password")



def dep_login():

        root = Tk()
        root.title("Department Details")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        global e1,e2

        uname = Label(root,text="Username").place(x=10,y=10)
```

```
        password = Label(root,text="Password").place(x=10,y=40)



        e1 = Entry(root)
        e1.place(x=140, y=10)

        e2 = Entry(root)
        e2.place(x=140, y=40)
        e2.config(show="*")

        Button(root, text="Login", command=identity, height =3, width =
13).place(x=10,y=100)



        root.mainloop()



def details():

        global d_deptid, d_name, d_email

        root = Tk()
        root.title("UNIVERSITY OF NEW HAVEN")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        d_deptid = Label(root,text="Enter Department ID").place(x=20,y=30)

        d_name = Label(root,text="Enter Department Name").place(x=20,y=60)

        d_email = Label(root,text="Enter Department Email").place(x=20,y=90)
```

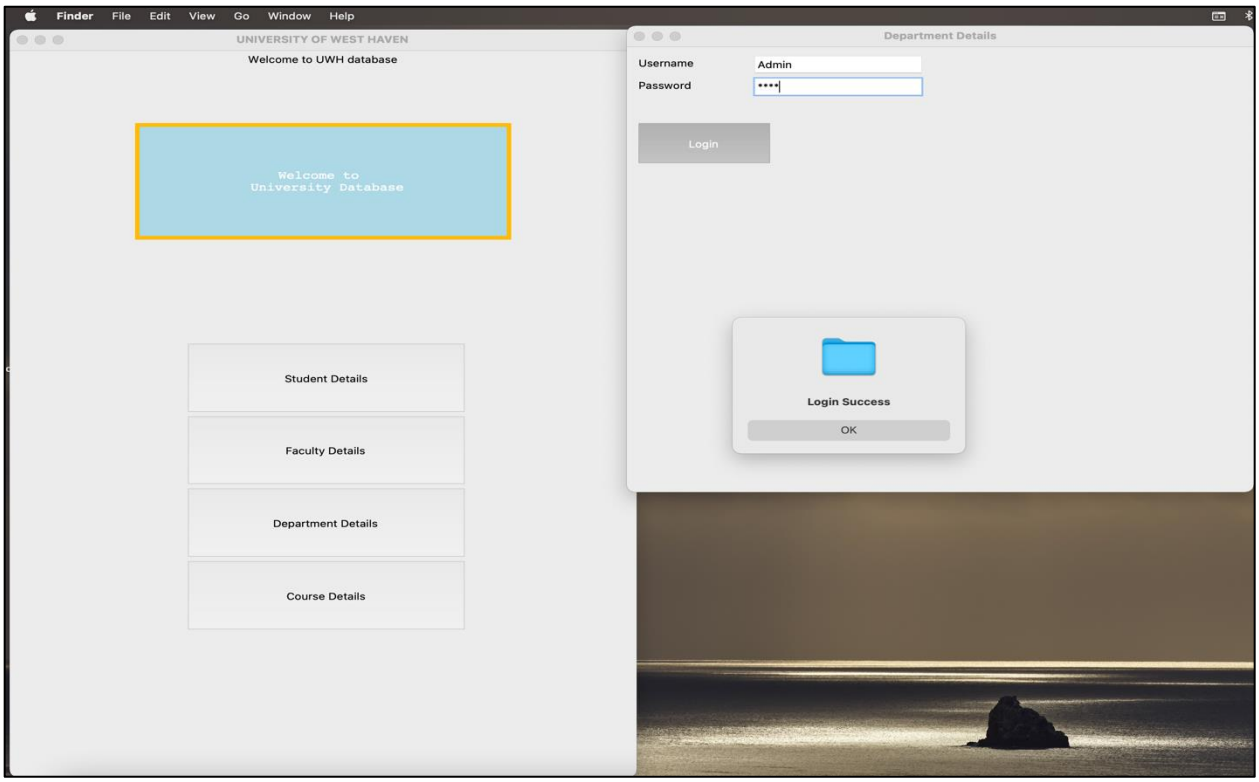
```
        d_deptid = Entry(root)
        d_deptid.place(x=150, y=30)

        d_name = Entry(root)
        d_name.place(x=180, y=60)

        d_email = Entry(root)
        d_email.place(x=180, y=90)



        Button (root,text= "insert",command=insert).place(x=10, y=240)

        Button (root,text= "delete",command=delete).place(x=80, y=240)

        Button (root,text= "update",command=update).place(x=150, y=240)

        Button (root,text= "get",command=get).place(x=230, y=240)

        #list= Listbox(root)
        #list.place(x=390,y=30)
        #show();

        root.mainloop()

def insert():
        dept_id = d_deptid.get()
        dept_name = d_name.get()
        dept_email = d_email.get()

        if(dept_id == "" or dept_name == "" or dept_email==""):
```

```
                messagebox.showinfo("Insert Status","All fields are required")
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("insert into department values('"+ dept_id
+"','"+ dept_name +"','"+dept_email +"')")
                cursor.execute("commit")
                con.close()


def update():


        dept_id = d_deptid.get()
        dept_name = d_name.get()
        dept_email = d_email.get()



        if(dept_id =="" or dept_name =="" or dept_email==""):
                messagebox.showinfo("Update Status","All fields are
required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("update department set
dept_id='"+dept_id+"',dept_name='"+dept_name+"',dept_email='"+dept_email+"'
where dept_id = '"+ dept_id+"'")
                cursor.execute("commit")


                dept_id.delete(0,'end')
                dept_name.delete(0,'end')
                dept_email.delete(0,'end')


                messagebox.showinfo("Update status","Update Successfully")
```

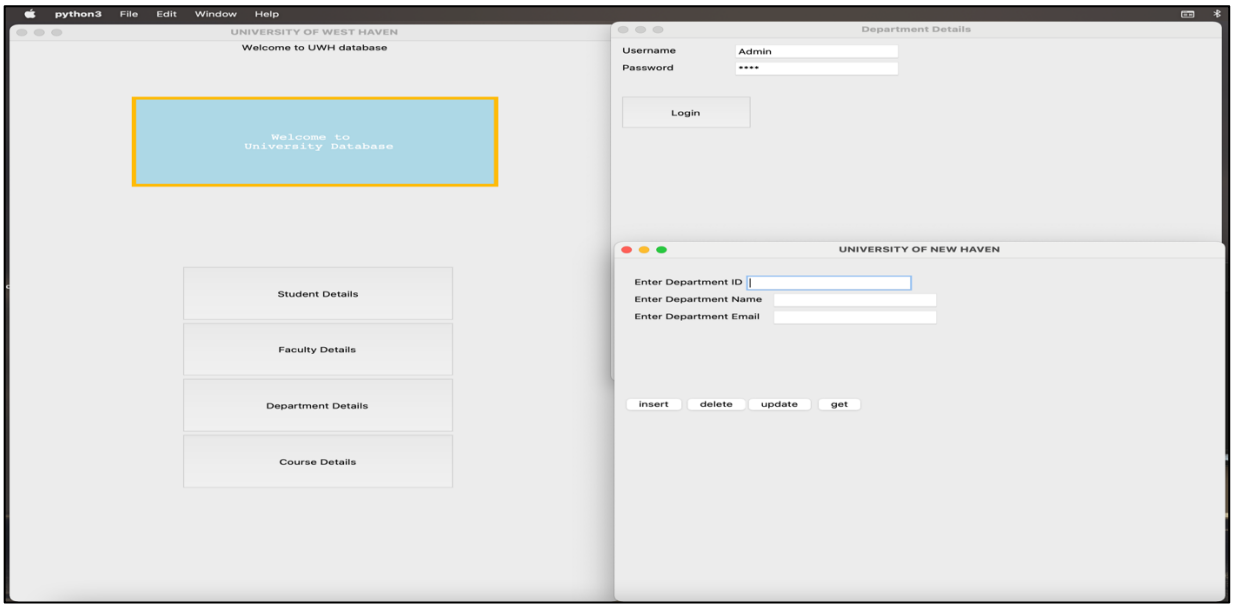
```
                con.close()


def delete():
        dept_id = d_deptid.get()
        if(dept_id== ""):
                messagebox.showinfo("Delete Status","ID is required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("delete from department where dept_id = '"+
dept_id +"'")
                cursor.execute("commit")
                con.close()



def get():
        if(d_deptid.get()==""):
                messagebox.showinfo("Fetch status", "ID is compulsory to fetch
details")
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("select * from department where dept_id = '"+
d_deptid.get() +"'")
                rows = cursor.fetchall()


                for row in rows:
                        d_name.insert(0, row[1])
                        d_email.insert(0, row[2])


                con.close()
```

Output

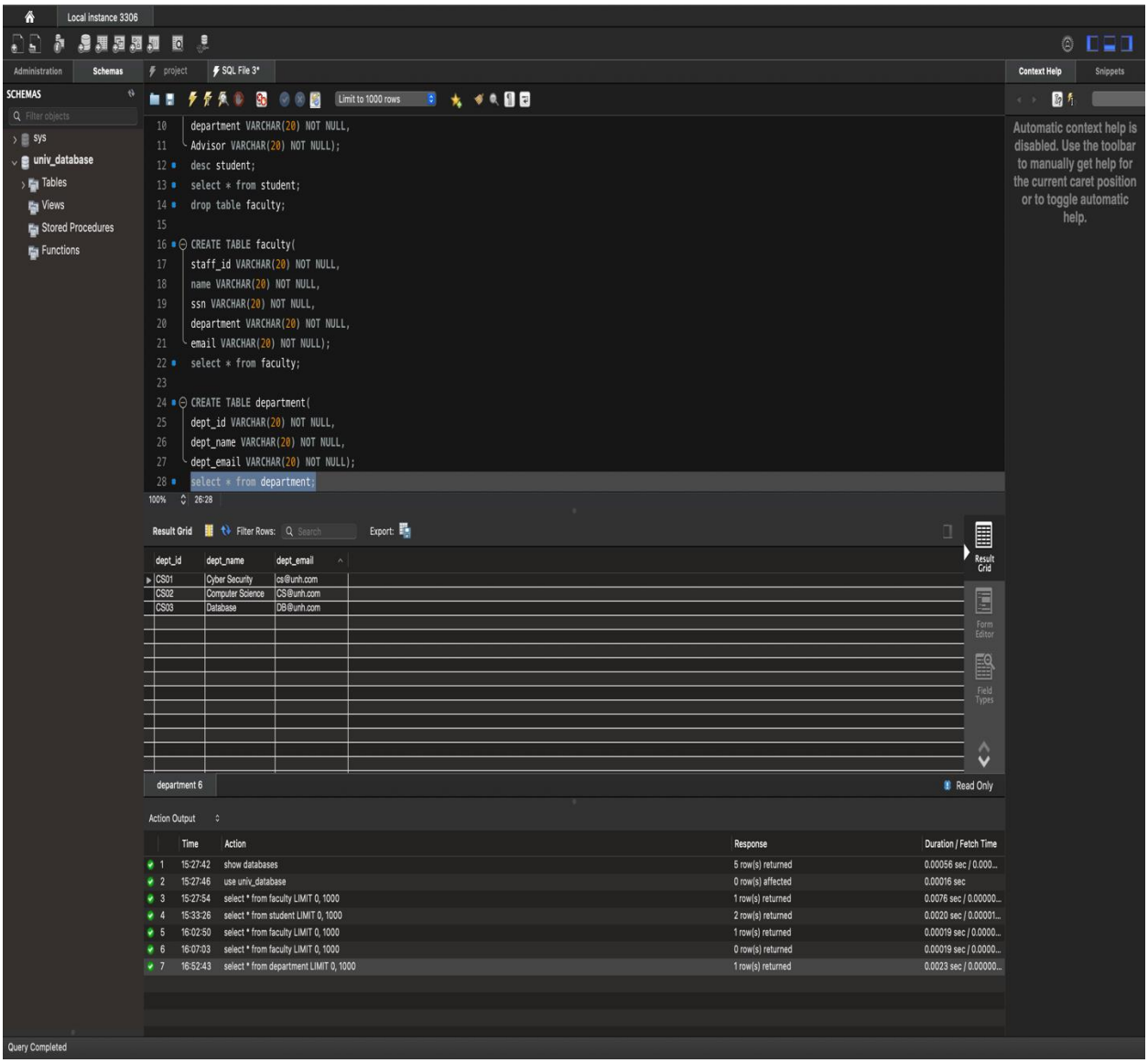
Login for Faculty



CRUD for Faculty Section



DB Output


Local instance 3306
Administration
Schemas
project
SQL File 3*
Context Help
Snippets
SCHEMAS
Filter objects
sys
univ_database
Tables
Views
Stored Procedures
Functions
Limit to 1000 rows
10 department VARCHAR(20) NOT NULL,
11 Advisor VARCHAR(20) NOT NULL);
12 desc student;
13 select * from student;
14 drop table faculty;
15
16 CREATE TABLE faculty(
17 staff_id VARCHAR(20) NOT NULL,
18 name VARCHAR(20) NOT NULL,
19 ssn VARCHAR(20) NOT NULL,
20 department VARCHAR(20) NOT NULL,
21 email VARCHAR(20) NOT NULL);
22 select * from faculty;
23
24 CREATE TABLE department(
25 dept_id VARCHAR(20) NOT NULL,
26 dept_name VARCHAR(20) NOT NULL,
27 dept_email VARCHAR(20) NOT NULL);
28 select * from department;
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Result Grid
Filter Rows:
Export:
dept_id dept_name dept_email
CS01 Cyber Security cs@unh.com
CS02 Computer Science CS@unh.com
CS03 Database DB@unh.com
department 6
Read Only
Action Output
Time Action Response Duration / Fetch Time
1 15:27:42 show databases 5 row(s) returned 0.00056 sec / 0.000...
2 15:27:46 use univ_database 0 row(s) affected 0.00016 sec
3 15:27:54 select * from faculty LIMIT 0, 1000 1 row(s) returned 0.0076 sec / 0.00000...
4 15:33:26 select * from student LIMIT 0, 1000 2 row(s) returned 0.0020 sec / 0.00001...
5 16:02:50 select * from faculty LIMIT 0, 1000 1 row(s) returned 0.00019 sec / 0.0000...
6 16:07:03 select * from faculty LIMIT 0, 1000 0 row(s) returned 0.00019 sec / 0.0000...
7 16:52:43 select * from department LIMIT 0, 1000 1 row(s) returned 0.0023 sec / 0.00000...
Query Completed

Course.py

```
from tkinter import *
from tkinter import messagebox
import mysql.connector as mysql

def identity():
        uname= e1.get()
        password=e2.get()

        if(uname == "" and password == ""):
                messagebox.showinfo("", "Please fill all fields")

        elif(uname == "Admin" and password == "1243"):
                messagebox.showinfo("", "Login Success")
                details();

        else:
                messagebox.showinfo("", "Incorrect username and password")


def course_login():

        root = Tk()
        root.title("Course Details")
        root.minsize(width=500,height=500)
        root.geometry("700x600")

        global e1,e2

        uname = Label(root,text="Username").place(x=10,y=10)

        password = Label(root,text="Password").place(x=10,y=40)


        e1 = Entry(root)
        e1.place(x=140, y=10)

        e2 = Entry(root)
        e2.place(x=140, y=40)
        e2.config(show="*")

        Button(root, text="Login", command=identity, height =3, width =
13).place(x=10,y=100)


        root.mainloop()


def details():

        global c_cid, c_cname, c_cterm, c_cfaculty, c_ctime

        root = Tk()
        root.title("UNIVERSITY OF NEW HAVEN")
        root.minsize(width=500,height=500)
        root.geometry("700x600")
```

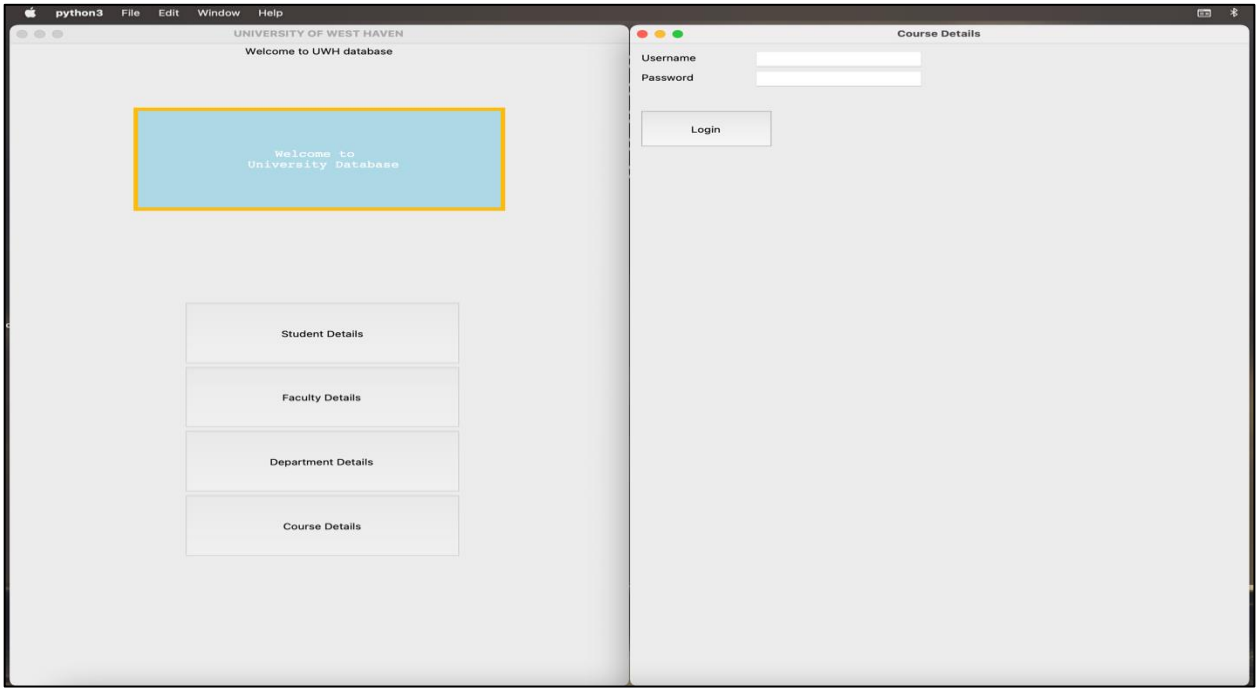
```
        c_cid = Label(root,text="Enter Course ID").place(x=20,y=30)

        c_cname = Label(root,text="Enter Course Name").place(x=20,y=60)

        c_cterm = Label(root,text="Enter Course Term ").place(x=20,y=90)

        c_cfaculty = Label(root,text="Enter Faculty Name").place(x=20,y=120)

        c_ctime = Label(root,text="Enter Course Time").place(x=20,y=150)


        c_cid= Entry(root)
        c_cid.place(x=150, y=30)

        c_cname = Entry(root)
        c_cname.place(x=180, y=60)

        c_cterm = Entry(root)
        c_cterm.place(x=180, y=90)

        c_cfaculty = Entry(root)
        c_cfaculty.place(x=180, y=120)

        c_ctime = Entry(root)
        c_ctime.place(x=180, y=150)


        Button (root,text= "insert",command=insert).place(x=10, y=240)

        Button (root,text= "delete",command=delete).place(x=80, y=240)

        Button (root,text= "update",command=update).place(x=150, y=240)

        Button (root,text= "get",command=get).place(x=230, y=240)

        #list= Listbox(root)
        #list.place(x=390,y=30)
        #show();

        root.mainloop()


def insert():
        course_id = c_cid.get()
        course_name = c_cname.get()
        course_term = c_cterm .get()
        course_faculty = c_cfaculty.get()
        course_time = c_ctime.get()

        if(course_id =="" or course_name =="" or course_term =="or
course_faculty =="or course_time==""):
                messagebox.showinfo("Insert Status","All fields are
required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
```

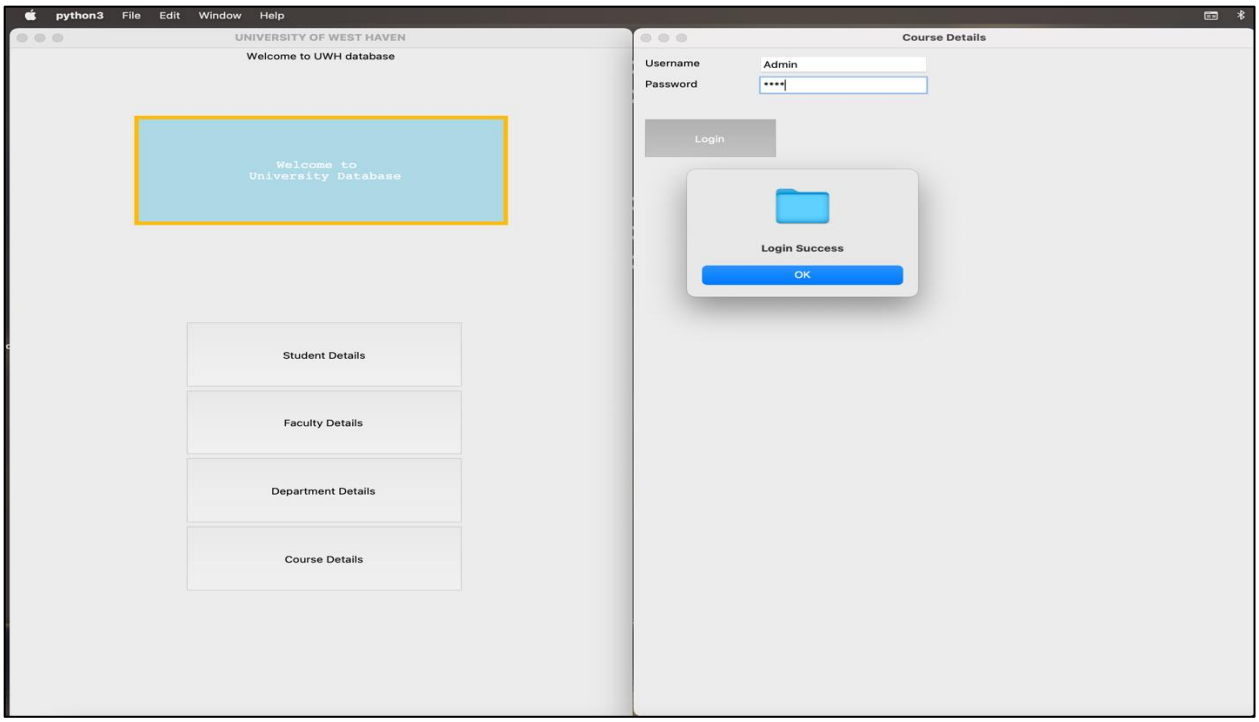
```
                cursor= con.cursor()
                cursor.execute("insert into course values('"+ course_id+"','"+
course_name +"', '"+ course_term +"','"+ course_faculty
+"','"+course_time+"')")
                cursor.execute("commit")
                con.close()

def update():

        course_id = c_cid.get()
        course_name = c_cname.get()
        course_term = c_cterm .get()
        course_faculty = c_cfaculty.get()
        course_time = c_ctime.get()

        if(course_id =="" or course_name =="" or course_term =""or
course_faculty =""or course_time==""):
                messagebox.showinfo("Update Status","All fields are
required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("update course set
course_id='"+course_id+"',course_name='"+course_name+"',course_term
='"+course_term+"',course_faculty ='"+course_faculty+"',course_time
='"+course_time+"' where course_id = '"+ course_id+"'")
                cursor.execute("commit")

                course_id.delete(0,'end')
                course_name.delete(0,'end')
                course_term.delete(0,'end')
                course_faculty.delete(0,'end')
                course_time.delete(0,'end')

                messagebox.showinfo("Update status","Update Successfully")
                con.close()

def delete():
        course_id = c_cid.get()
        if(course_id== ""):
                messagebox.showinfo("Delete Status","ID is required");
        else:
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("delete from course where course_id = '"+
course_id +"'")
                cursor.execute("commit")
                con.close()

def get():
        if(c_cid.get()==""):
                messagebox.showinfo("Fetch status", "ID is compulsory to fetch
details")
        else:
```

```
                con = mysql.connect(host="localhost", user="root",
password="rootroot", database="univ_database")
                cursor= con.cursor()
                cursor.execute("select * from course where course_id = '"+
c_cid.get() +"'")
                rows = cursor.fetchall()

                for row in rows:
                        c_cname.insert(0, row[1])
                        c_cterm.insert(0, row[2])
                        c_cfaculty.insert(0, row[3])
                        c_ctime.insert(0, row[4])


                con.close()
```

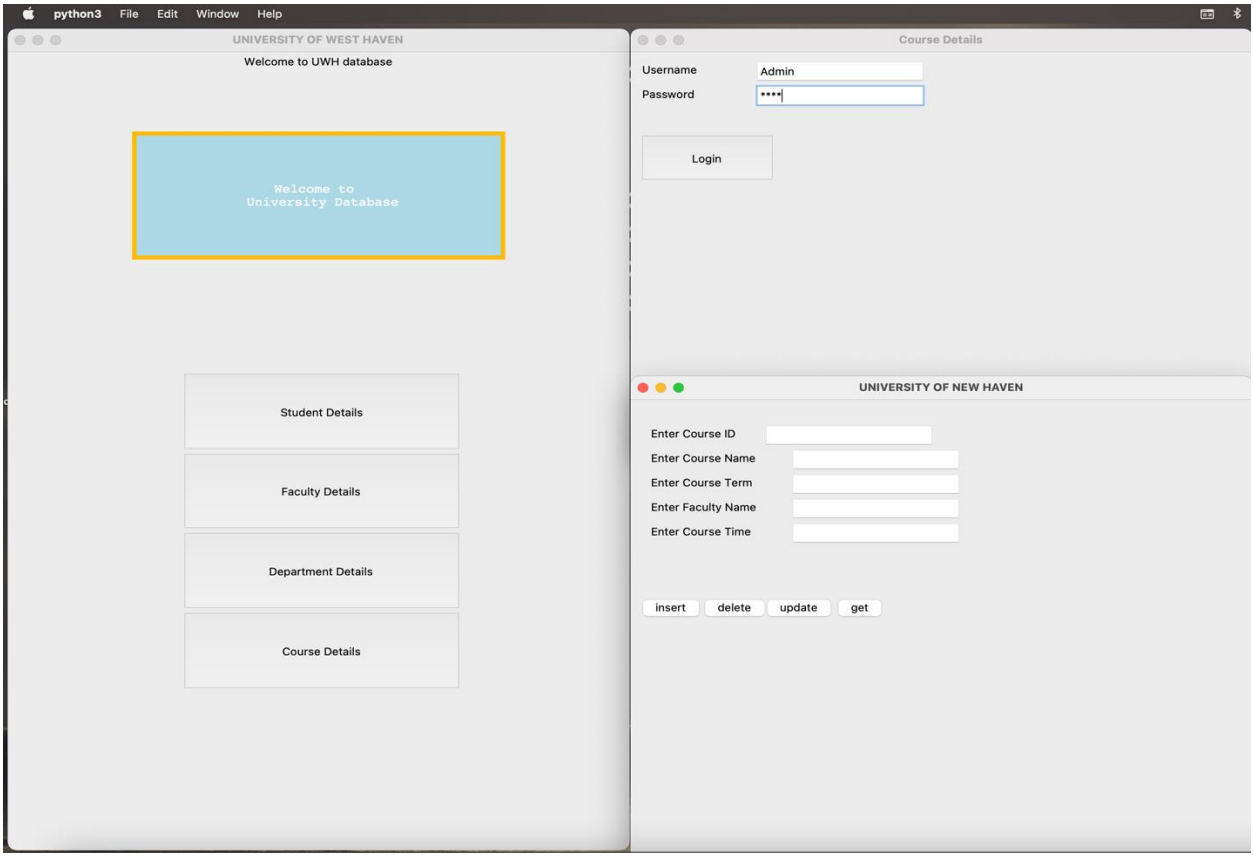
Output

Login page for Course Details

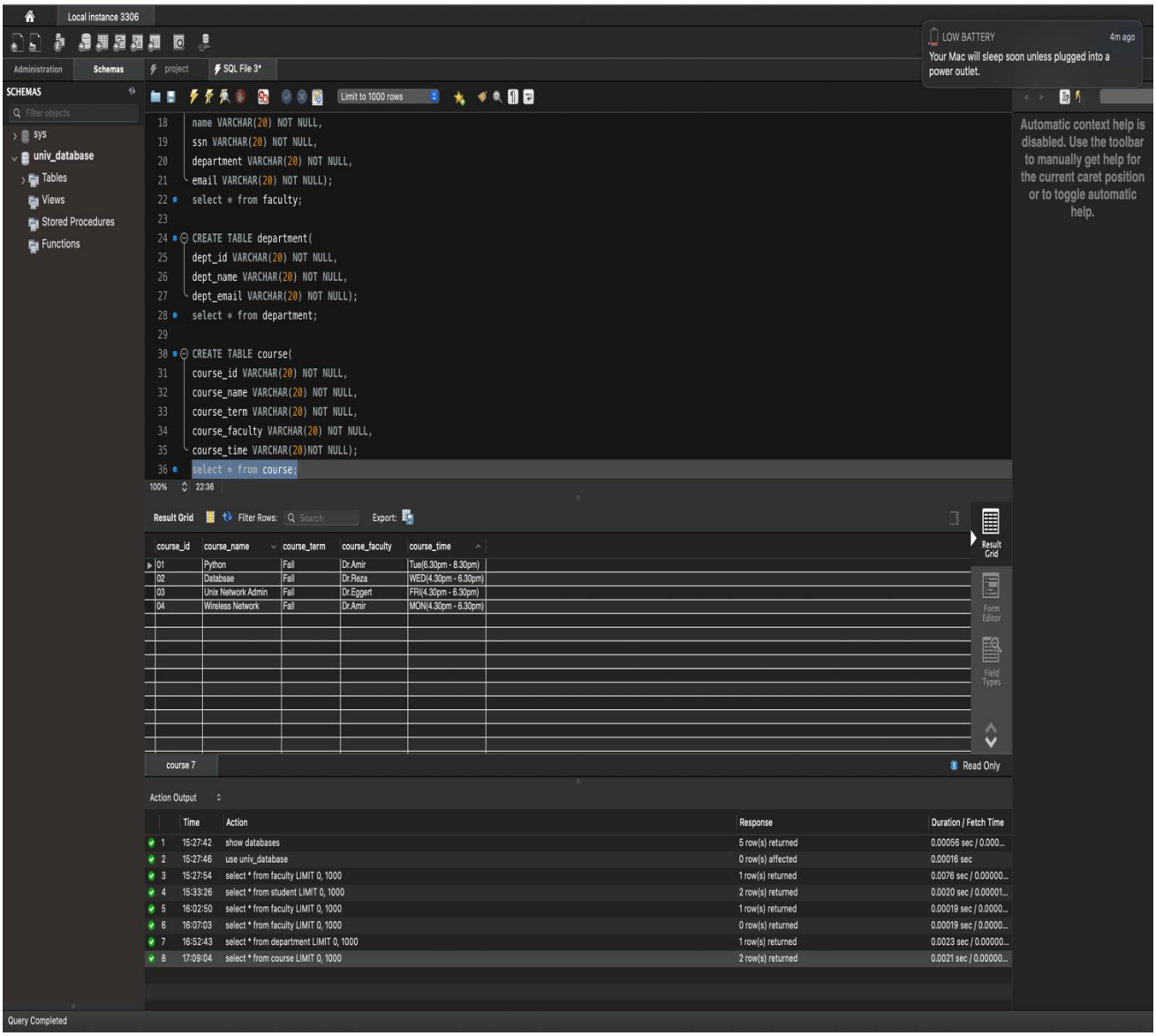




CRUD Operation



DB Result



Conclusion:

The university database was successfully created and implemented using the GUI. In future, this database will be implemented with some more details and will be hosted in the internet to be accessed by everyone with added security features.