SURMOUNT
INTERNATIONAL SCHOOL
GORAKHPUR

# PROJECT REPORT

## ONLINE EXAMINATION SYSTEM

COMPUTER SCIENCE

SESSION 2021-22

SUBMITTED BY- ANANT KUMAR SINGH

CLASS- XII B

SURMOUNT INTERNATIONAL SCHOOL

GORAKHPUR

EXCELLENCE DILIGENCE

# CERTIFICATE

This is to certify that the project entitled “Online Examination management” is a record of Bona fide work carried out by

**Anant Kumar Singh XII B .**

| | | |
|---|---|---|
| Principal | External Examiner | Internal Examiner |

# DECLARATION

I hereby declare that the project work entitled "Online examination Management", submitted to the Department of Computer Science, Surmount International School, Siddharthpuram, Gorakhpur is completely prepared by me. All the codings are result of my personal efforts.

I further declare that this project record or any of its part has not been submitted elsewhere for any other class.

Anant Kumar Singh

XII, B

# Acknowledgement

I would like to express a deep sense of gratitude to my project guide **teacher** who guided me immensely through the course of the project.  His constructive advice and efficient teaching have been responsible for the successful completion of this project.

My sincere thanks to Our Principal, for her coordination in extending every possible support for the completion of this project.

I must thank my classmates for their overwhelming support and timely help which made this project possible. I would also like to extend my hearty gratitude to my friend devash, XII B who made a similar project on the same theme and helped me throughout the process.

Last but not the least; I would like to thank all those people who had helped directly or indirectly towards the completion of this project.

Anant Kumar Singh
XII, B

# Examination Management System

Examination management system report in python.Examination is a core activity of any educational institution. As the examination arrives there exists a lot of work like consolidating the time table, seating arrangement and invigilation allotment which will be done manually and it takes lot of time and requires man power. Thus, an automated system would solve the above stated problem in just few clicks of work. The purpose of developing Examination Management Automation System is to computerize the traditional way of conducting exams. It is a web and android application that can be used by students and exam cell coordinator using their smart phones or PCs. The project keeps track of various details in modules such as Students Details, Staff Details, and Hall Details with proper descriptions. It also has some features to generate reports for bundle handovers, absentee's statement and roll lists. Python project Report on Examination management system.

The proposed system is an android and web based application that is designed to manage and handle the operations in an educational institute during the time examinations. It is an application that can be used by all the students and staff in an educational institute in order to facilitate the communication between them. The application is easily adaptable as it is used on a desktop systems and mobile device. Since the developed application is used on a mobile devices, it improves connectivity between the users, thus helping the institution to provide a more transparent system altogether. The Examination Management

Automation System was developed for the educational institute to simplify the allocation of halls, seating arrangement of students and allocating staff to the examination halls.

Allocation of faculty to corresponding rooms will be done by the exam cell coordinator in the form of word documents and excel sheets and also allocation of students to their corresponding rooms is a frantic work which will be done manually and it takes lot of time and requires man power. The project keeps track of various details in modules such as Students Details, Staff Details, and Hall Details with proper descriptions. Most of the important processes in an educational institute are carried out manually such as teaching and nonteaching staff details, student information and number of halls available for the examination as all these process is done manually it increase the work load and easily prone to errors. The current systems are traditional systems which support manual processes leading to an immense time consumption and pile of hard copies. Existing system is inefficient, ineffective and less accurate, in such a situations report generation is not an simple task also if report is generated calculations has to be done manually which will surely results in errors.

# PURPOSE

The purpose of developing the Examination Management Automation System is to automate the regular way of organizing the examinations in an educational institute and generating reports according to the examination type and time.

Disadvantages of current system:

- Takes a lot of time.
- Resembles like a complex problem while allocating faculty to different rooms.
- Less Accurate.
- Requires more manual work.
- Paper work required.
- Previous records not stored.

Advantages over current system:

- Easy to handle and operate.
- Friendly interface.
- Fast and convenient.
- Easy to update.
- Smart way of communication
- No paper work required.

# Problem statement

Since the traditional have many drawbacks such as time consuming, Difficulty of analysing the test manually, More observers are required to take exam of many students, Results are not accurate since calculations is done manually, The chance of losing exam's result is higher in current systems, Checking of result is time consuming since it done manually, Limitation of no of student can give examination at a time. with the development of information technology and use it in an orderly and properly helps to overcome the existing error in the manual system . Online examination system saves the exams information in a database, and this make it an easier way to give exam teachers can add theirs exams rules , and student can give exam in a totally automated system

# Proposed Method

the high end architecture of the application and shows that it takes the input from the user either from the web page through Mozilla Firefox, Google Chrome, Internet Explorer or any other web browsers or from the android application. The request is sent to the apache server where the data is processed by the server and sends a query to the database if any records needs to be fetched and fetches the information from the database.If the requested data is present in the database then the results are retrieved by the server and the reply is sent to the user. The High end architecture consists of three different tiers, which are used to receive and display the result according to the user's requirement. The requested query from the user is fetched from the database, if requested data is present then it is retrieved back to the user according to his requirement. Examination management automation system consists of following tiers.

- Presentation tier.
- Business tier.
- Data tier.

## Presentation Tier

The presentation tier is the logical group of components that provide a user interface. These components can include server pages, forms and reports, the components allow the users to interact with the application, but they do not process data, handle business rules, directly access databases or other storage media. The Front end of the Examination Cell Automation System is called as Presentation tier was designed using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Bootstrap and JavaScript. Presentation tier is the GUI, from where the user can request the server and view the results that were displayed on the screen. The presentation tier consists of four different user interfaces based on the type of users.

- Admin UI
- Student UI

# Business Tier

The business tier is a heavyweight processing, validation, business rules, workflow and interfaces to external systems. Business tier components run on the middle tier and thus have historically been referred to as middleware. These components run within an application server, which provides the container for the components. The business tier consists of components that provide the business logic for an application. The business tier is the intermediate between the presentation tier and the data tier. All the business logics are performed in this tier, it takes the input from the presentation tier execute the query and fetches the data from database. It uses Apache server and PHP Scripting for doing the operations.

# Data Tier

A data-tier application is a body that contains all databases and instance objects which were used in an application. A data tier provides a single unit for authoring, deploying, and managing the data-tier objects instead of having to manage them separately. A data tier allows tighter integration of data-tier development with the development of the associated application code. It also gives administrators an application level view of resource usage in their systems. The end tier of the architecture is data tier in which all the data is stored in its RDBMS. RDBMS have been a common choice for the storage of information in new databases used for financial records, manufacturing and logistical information and personnel data.

# Application flow graph

## Admin UI

the admin has to log on to the web page using his/her credentials like username and password then redirected to new page showing different modules like Staff, Student, Rooms. It is responsibility of the admin to upload all the student details like registered number, name, department etc., availability staff in the institute and the examination halls details like the capacity of the hall. The admin module consists of the following features/functions.

- Staff In this panel the admin can add the staff into the database by clicking on the add staff icon, details required for adding the staff are staff id, staff name, designation, department, email, contact and status of the staff. If the staff is on leave then the status is updated as not available. The admin can delete, view and finally edit and update the staff details and acknowledgement can be displayed.
- Student In this panel the admin enters the students into the database by clicking on the add student icon. Unlike staff the admin can directly upload the excel sheet of student details into the database. The admin can deletes entire students list or can be done on individual. There is also a feature to view and finally edit and update the student details by giving the inputs like academic year, year of study and semester.
- Rooms In this panel the admin allocate the rooms to the students by clicking on the add room icon, details required are room number, type, capacity and status of the room. If

the room is not available, then the status is updated as not available so that the room will not comes under invigilation duties. Admin can also delete, edit, update and view rooms. Staff can view the rooms that are allotted to them during the invigilation.

### Student UI

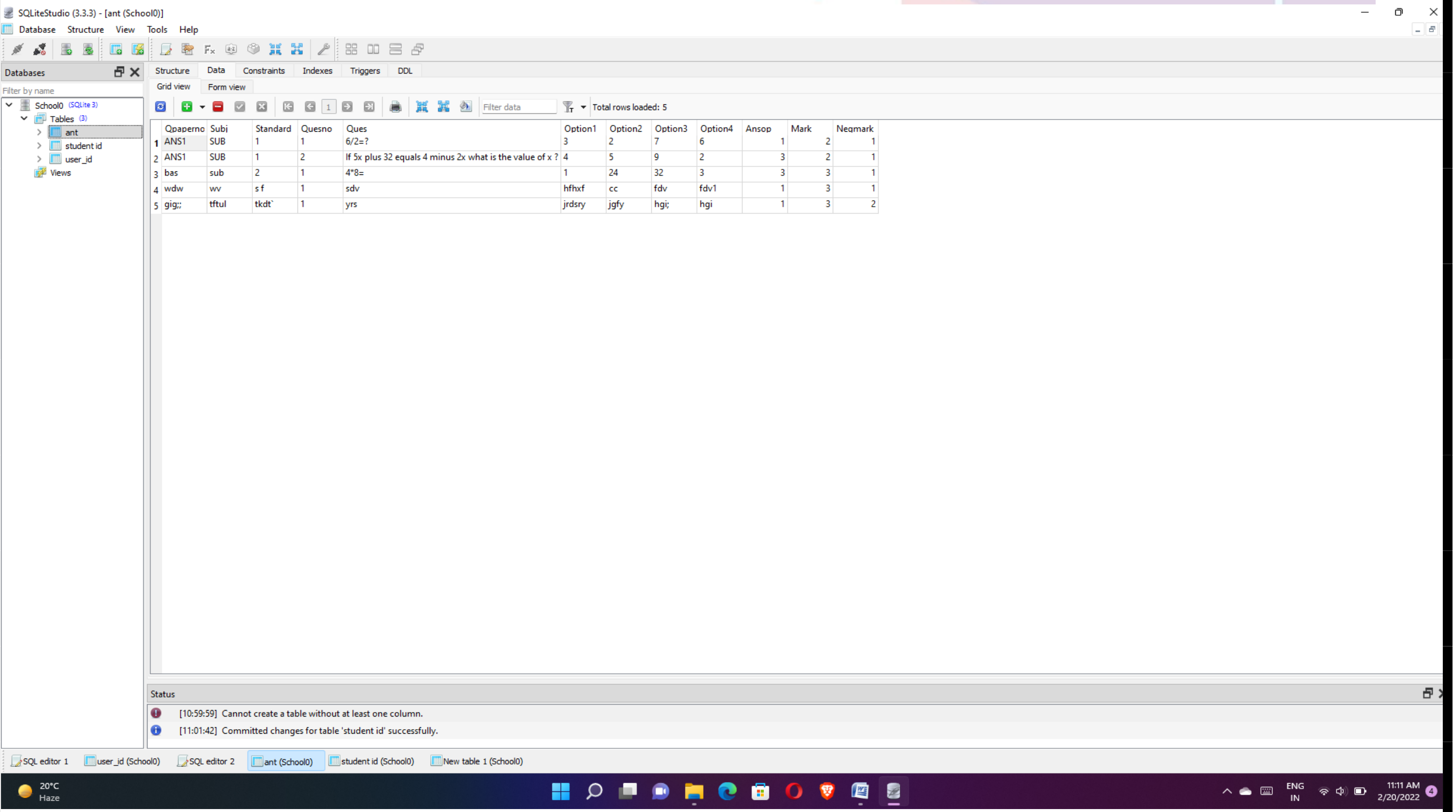
As soon as the moderator updates any information related to the examination cell, the students are immediately notified if they are connected to the application. If any file or documents are placed in the notice board along with the message students can access the file /document .Examination Management System Report. The Student UI consists of the notice board and the description of the examination cell.

## Conclusion

Examination management system report in python.Ultimately the output of the project reduces the manpower, workload on students as well as staff. It benefits all the educational institutes by reducing the complexity involved while allocating the exam duty for the staff, examination rooms for the students. All these data is stored in a centralized database which can be accessed whenever needed. Python project on Examination management system Report.

# Database

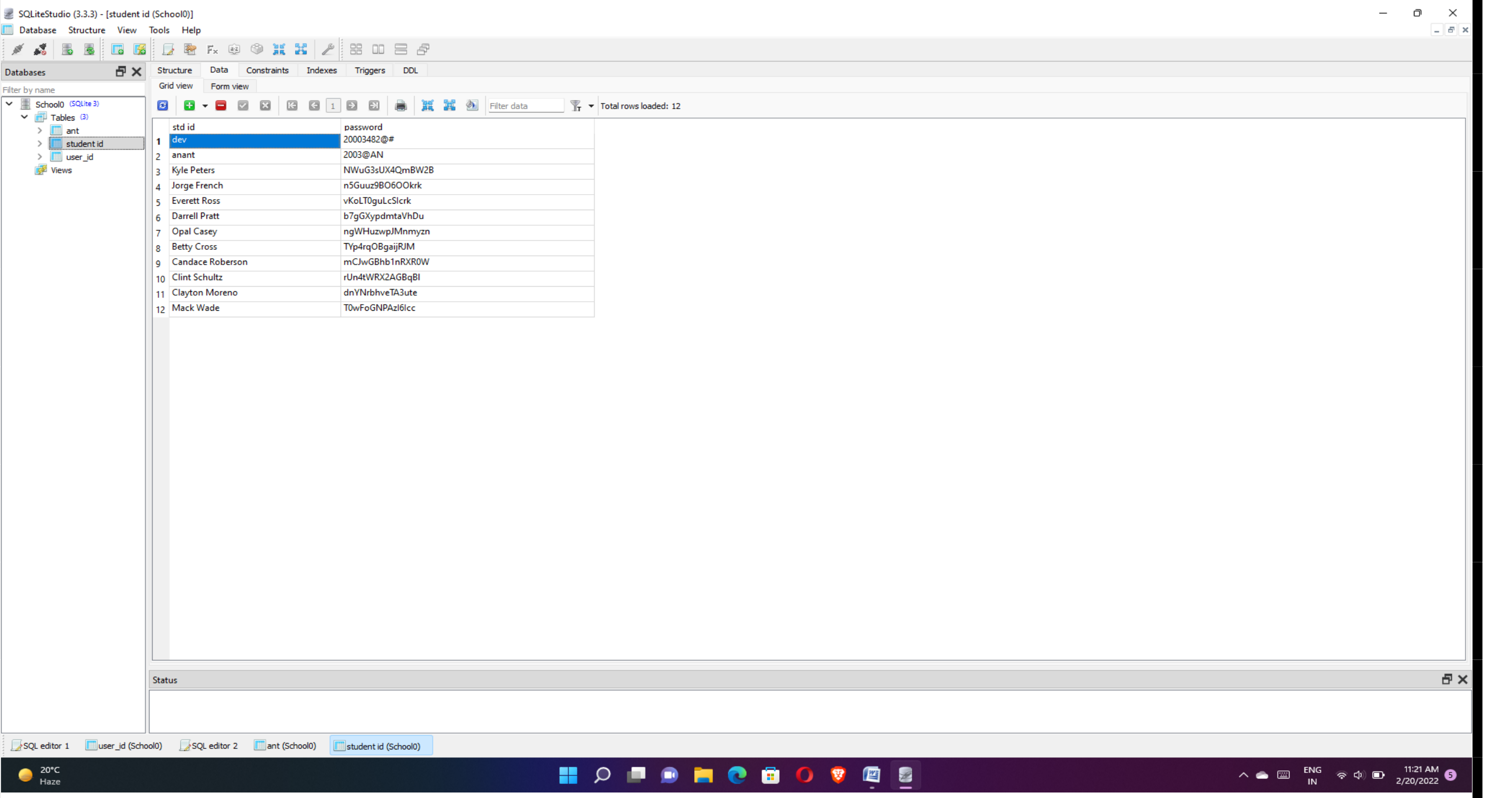
## >>Question paper database



| | Qpaperno | Subj | Standard | Quesno | Ques | Option1 | Option2 | Option3 | Option4 | Ansop | Mark | Negmark |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ANS1 | SUB | 1 | 1 | 6/2=? | 3 | 2 | 7 | 6 | 1 | 2 | 1 |
| 2 | ANS1 | SUB | 1 | 2 | If 5x plus 32 equals 4 minus 2x what is the value of x ? | 4 | 5 | 9 | 2 | 3 | 2 | 1 |
| 3 | bas | sub | 2 | 1 | 4*8= | 1 | 24 | 32 | 3 | 3 | 3 | 1 |
| 4 | wdw | wv | s f | 1 | sdv | hfhxf | cc | fdv | fdv1 | 1 | 3 | 1 |
| 5 | gig;; | tftul | tkdt` | 1 | yrs | jrdsry | jgfy | hgi; | hgi | 1 | 3 | 2 |

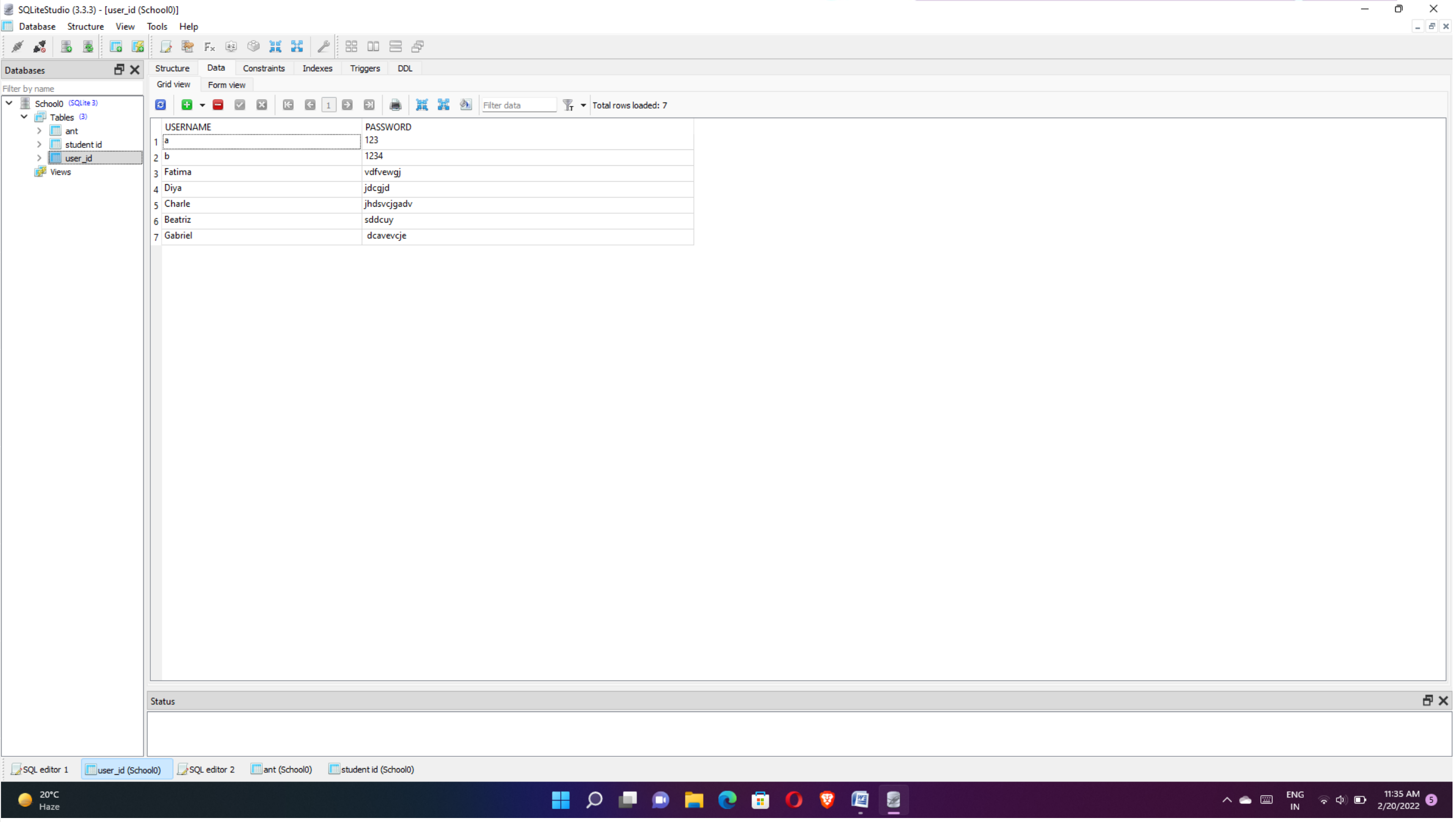
```
[10:59:59]  Cannot create a table without at least one column.
[11:01:42]  Committed changes for table 'student id' successfully.
```

## >>Student database



| | std id | password |
|---|---|---|
| 1 | dev | 20003482@# |
| 2 | anant | 2003@AN |
| 3 | Kyle Peters | NWuG3sUX4QmBW2B |
| 4 | Jorge French | n5Guuz9BO6OOkrk |
| 5 | Everett Ross | vKoLT0guLcSlcrk |
| 6 | Darrell Pratt | b7gGXypdmtaVhDu |
| 7 | Opal Casey | ngWHuzwpJMnmyzn |
| 8 | Betty Cross | TYp4rqOBgaijRJM |
| 9 | Candace Roberson | mCJwGBhb1nRXR0W |
| 10 | Clint Schultz | rUn4tWRX2AGBqBl |
| 11 | Clayton Moreno | dnYNrbhveTA3ute |
| 12 | Mack Wade | T0wFoGNPAzl6lcc |

# >>Administrator database

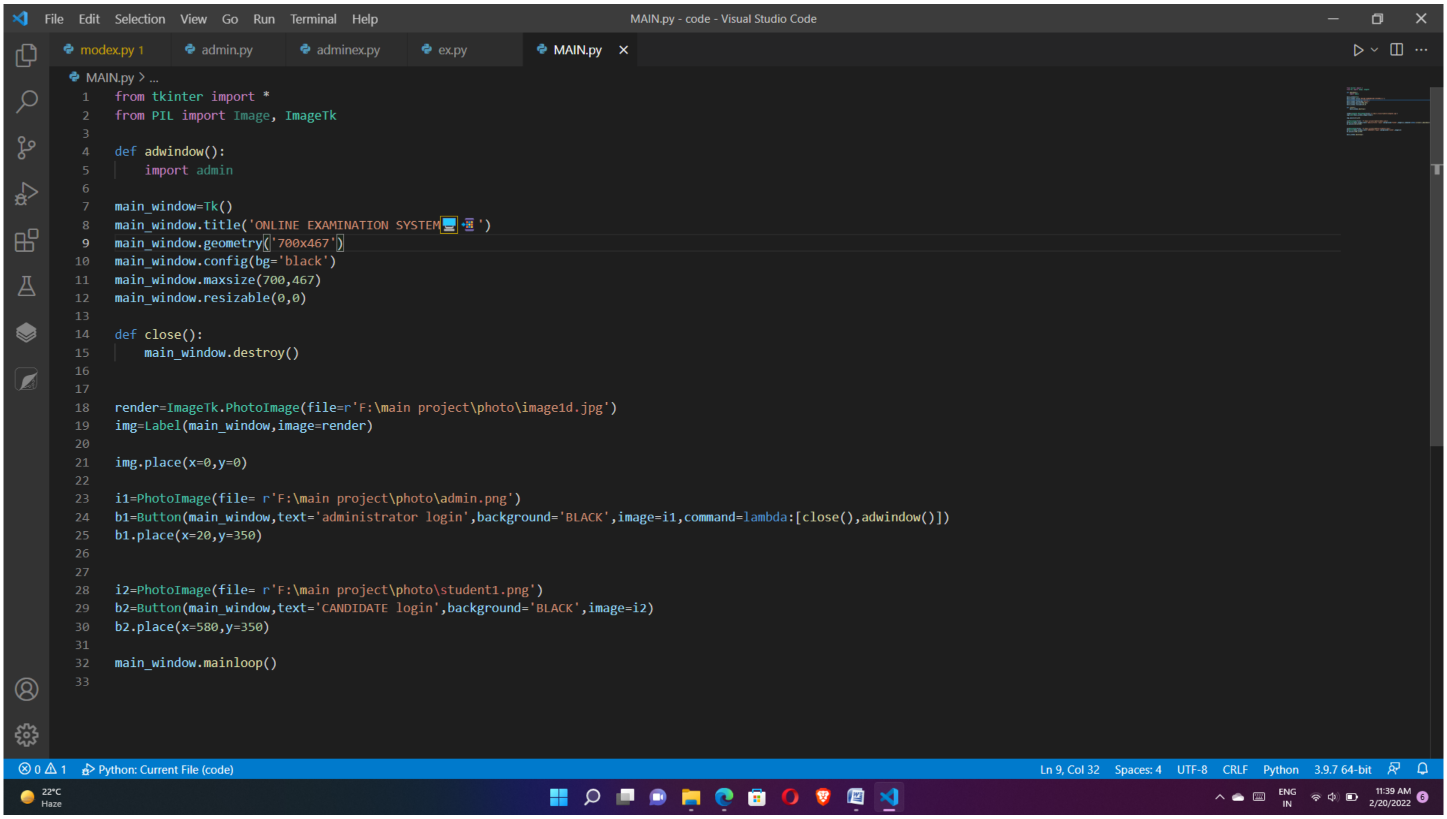


SQLiteStudio (3.3.3) - [user_id (School0)]

Total rows loaded: 7

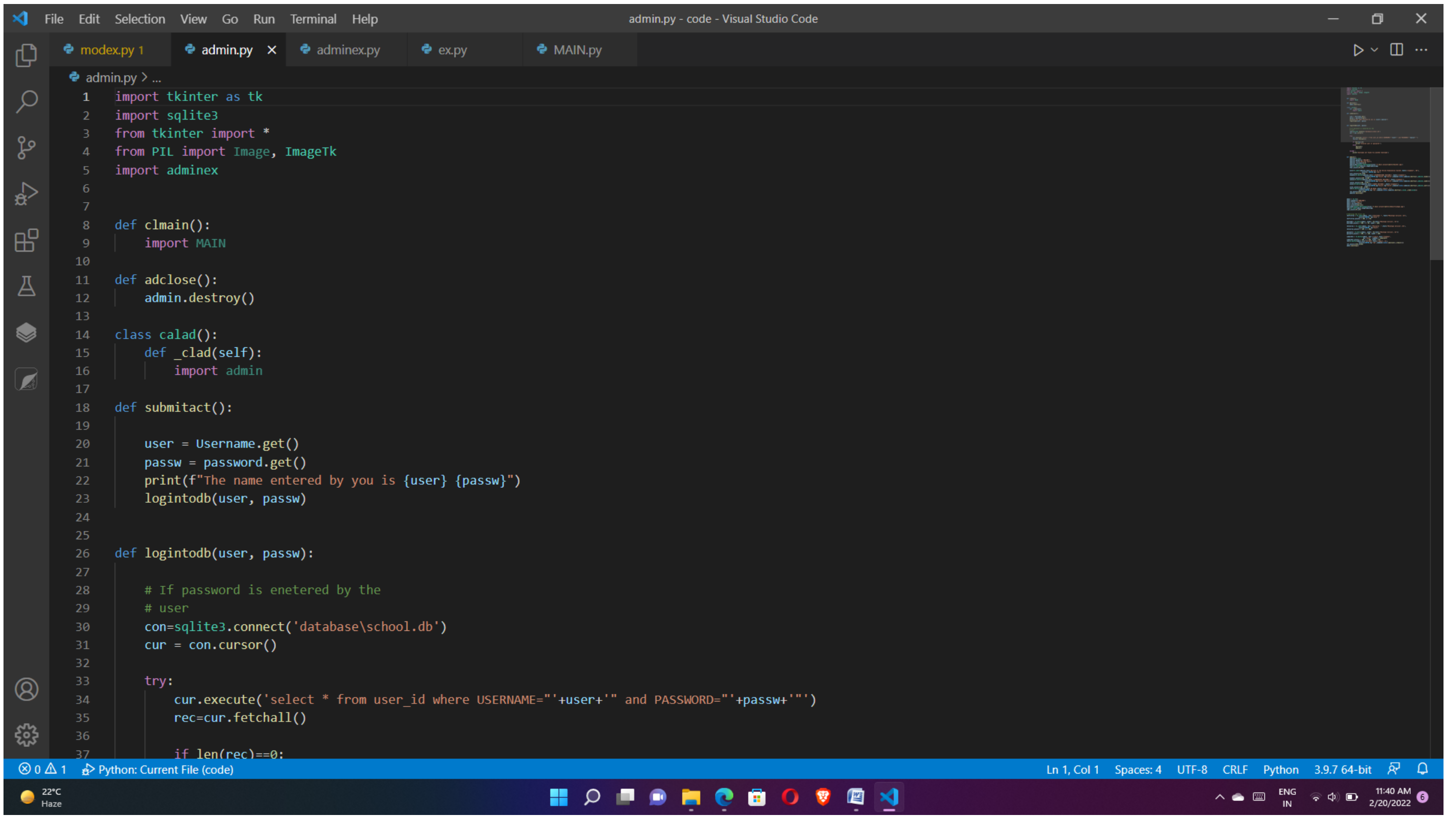
|   | USERNAME | PASSWORD |
|---|---|---|
| 1 | a | 123 |
| 2 | b | 1234 |
| 3 | Fatima | vdfvewgj |
| 4 | Diya | jdcgjd |
| 5 | Charle | jhdsvcjgadv |
| 6 | Beatriz | sddcuy |
| 7 | Gabriel | dcavevcje |

# SOURCE CODE

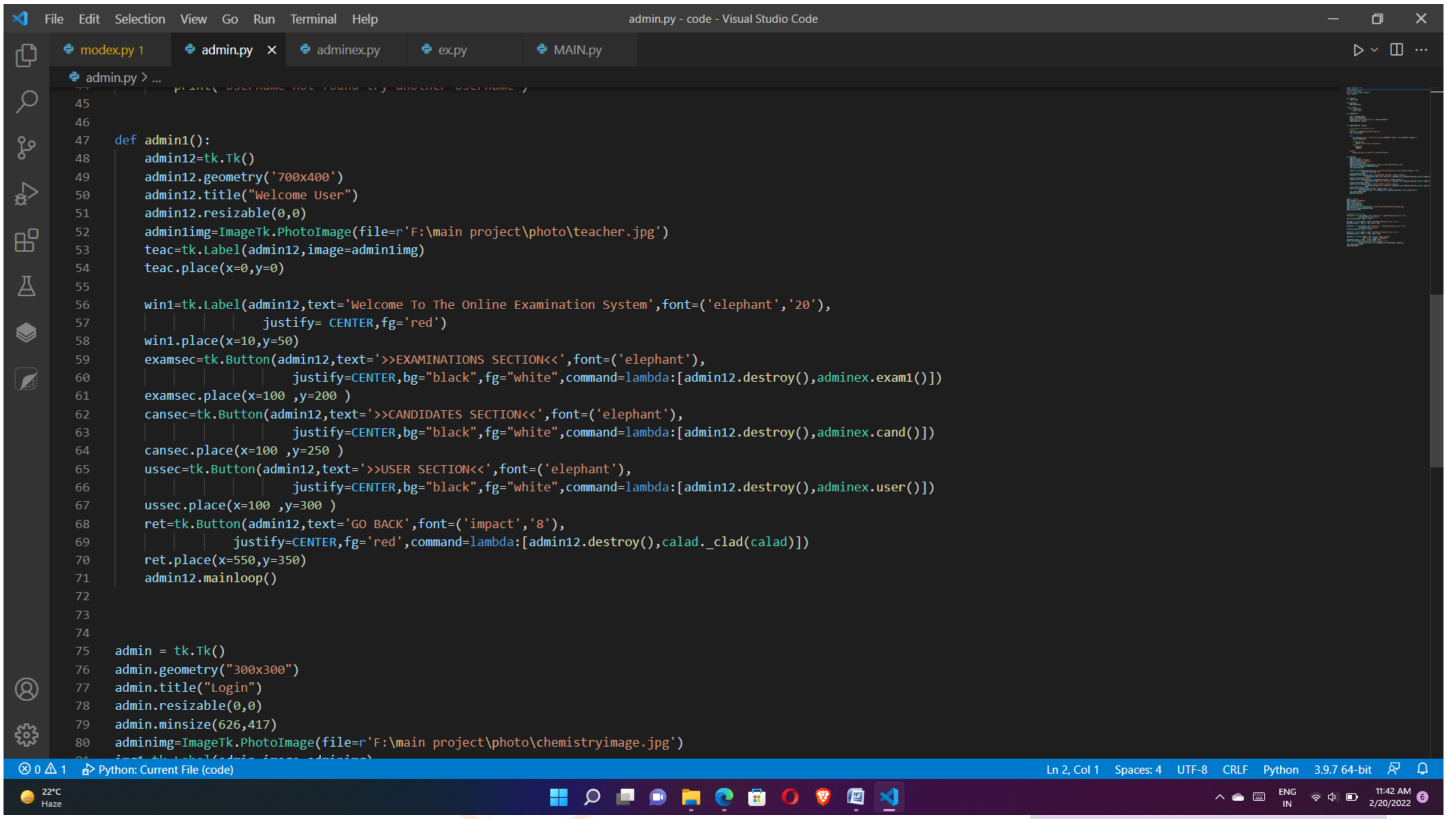


MAIN.py - code - Visual Studio Code

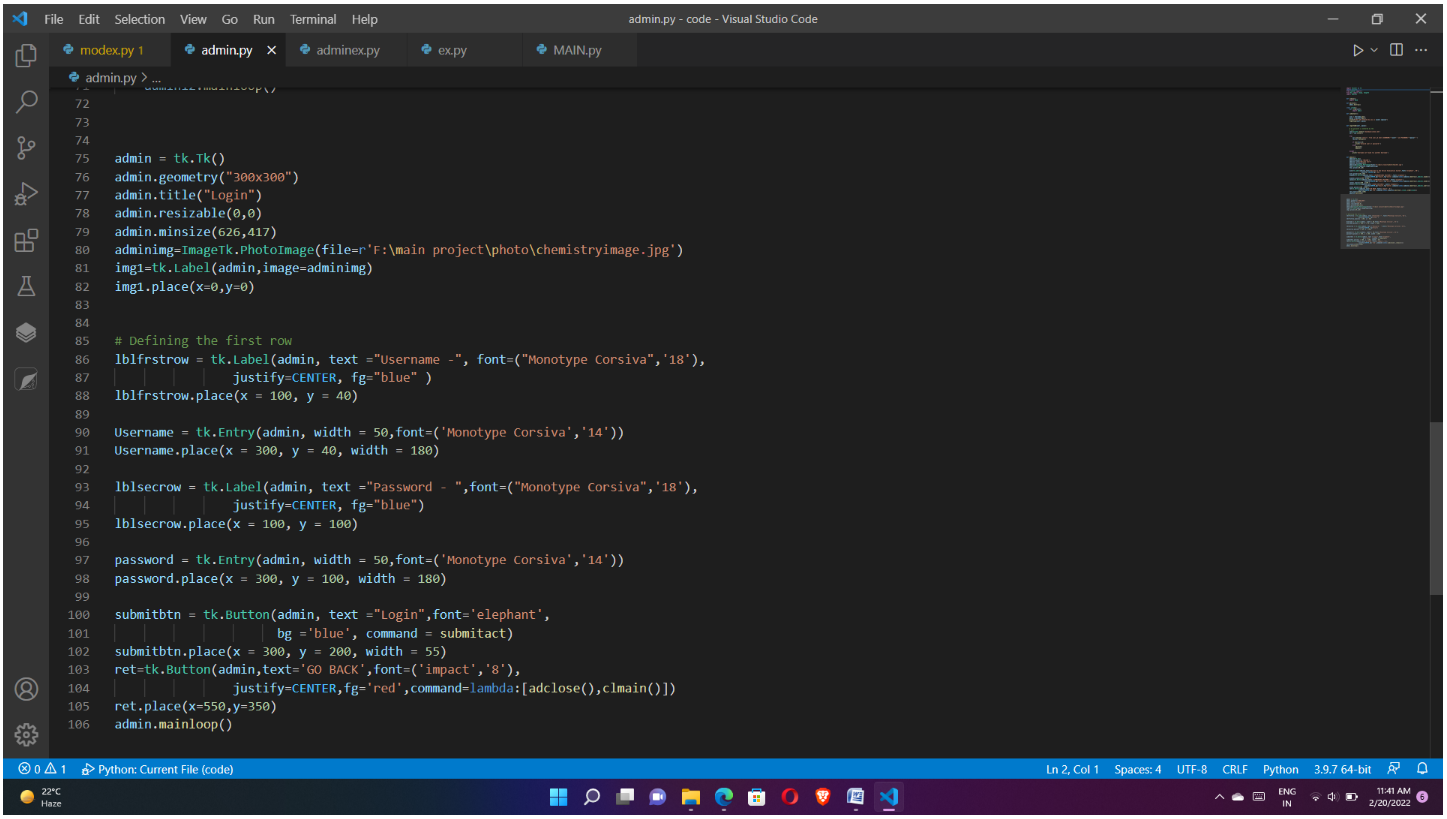
```python
from tkinter import *
from PIL import Image, ImageTk

def adwindow():
    import admin

main_window=Tk()
main_window.title('ONLINE EXAMINATION SYSTEM💻🧮')
main_window.geometry('700x467')
main_window.config(bg='black')
main_window.maxsize(700,467)
main_window.resizable(0,0)

def close():
    main_window.destroy()


render=ImageTk.PhotoImage(file=r'F:\main project\photo\image1d.jpg')
img=Label(main_window,image=render)

img.place(x=0,y=0)

i1=PhotoImage(file= r'F:\main project\photo\admin.png')
b1=Button(main_window,text='administrator login',background='BLACK',image=i1,command=lambda:[close(),adwindow()])
b1.place(x=20,y=350)


i2=PhotoImage(file= r'F:\main project\photo\student1.png')
b2=Button(main_window,text='CANDIDATE login',background='BLACK',image=i2)
b2.place(x=580,y=350)

main_window.mainloop()
```


File Edit Selection View Go Run Terminal Help
admin.py - code - Visual Studio Code
modex.py 1
admin.py
adminex.py
ex.py
MAIN.py
admin.py > ...
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.7 64-bit
Python: Current File (code)

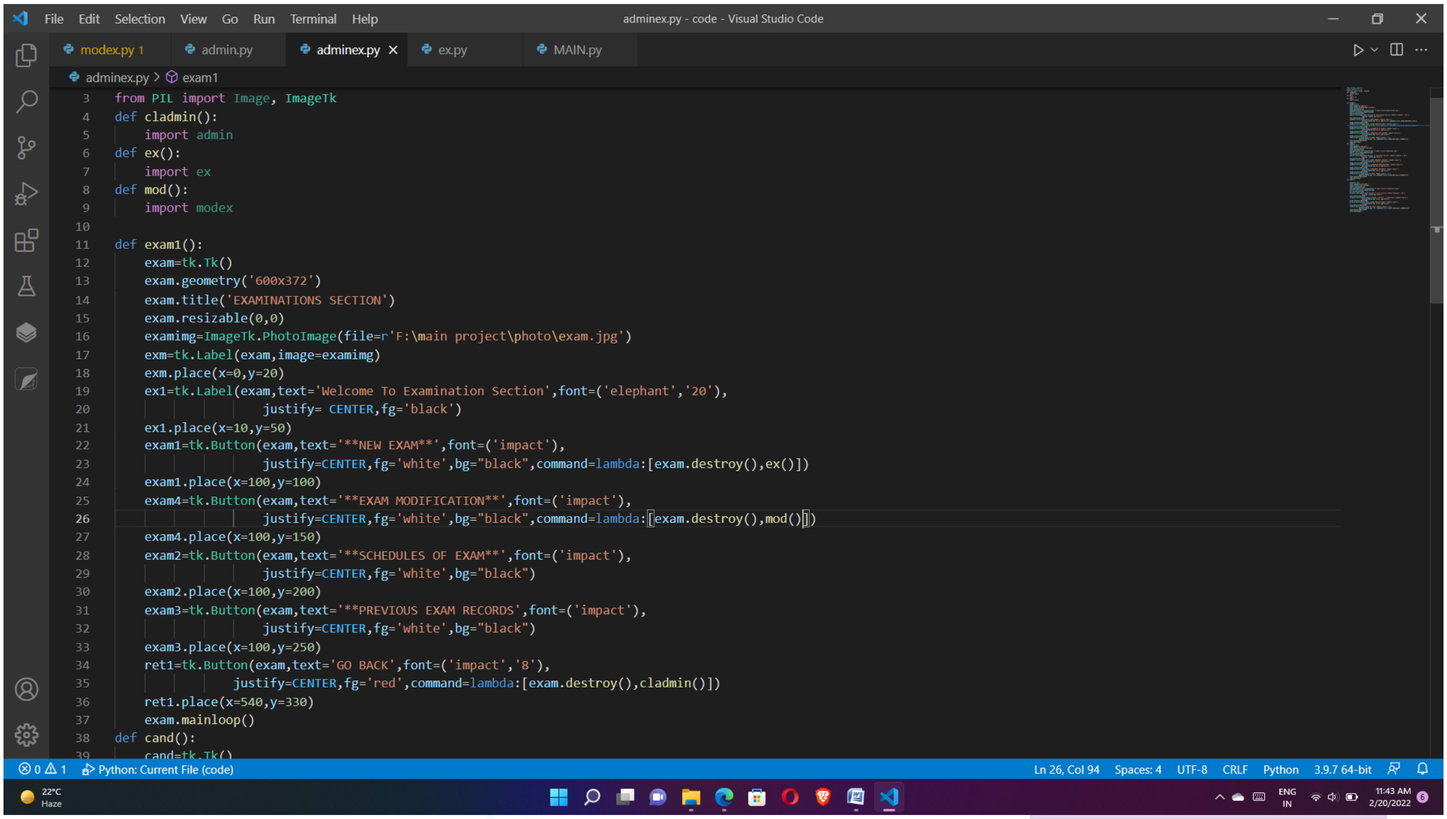
```python
import tkinter as tk
import sqlite3
from tkinter import *
from PIL import Image, ImageTk
import adminex


def clmain():
    import MAIN

def adclose():
    admin.destroy()

class calad():
    def _clad(self):
        import admin

def submitact():

    user = Username.get()
    passw = password.get()
    print(f"The name entered by you is {user} {passw}")
    logintodb(user, passw)


def logintodb(user, passw):

    # If password is enetered by the
    # user
    con=sqlite3.connect('database\school.db')
    cur = con.cursor()

    try:
        cur.execute('select * from user_id where USERNAME="'+user+'" and PASSWORD="'+passw+'"')
        rec=cur.fetchall()

        if len(rec)==0:
```


File Edit Selection View Go Run Terminal Help
admin.py - code - Visual Studio Code
modex.py 1
admin.py
adminex.py
ex.py
MAIN.py
admin.py > ...
Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Python 3.9.7 64-bit
Python: Current File (code)

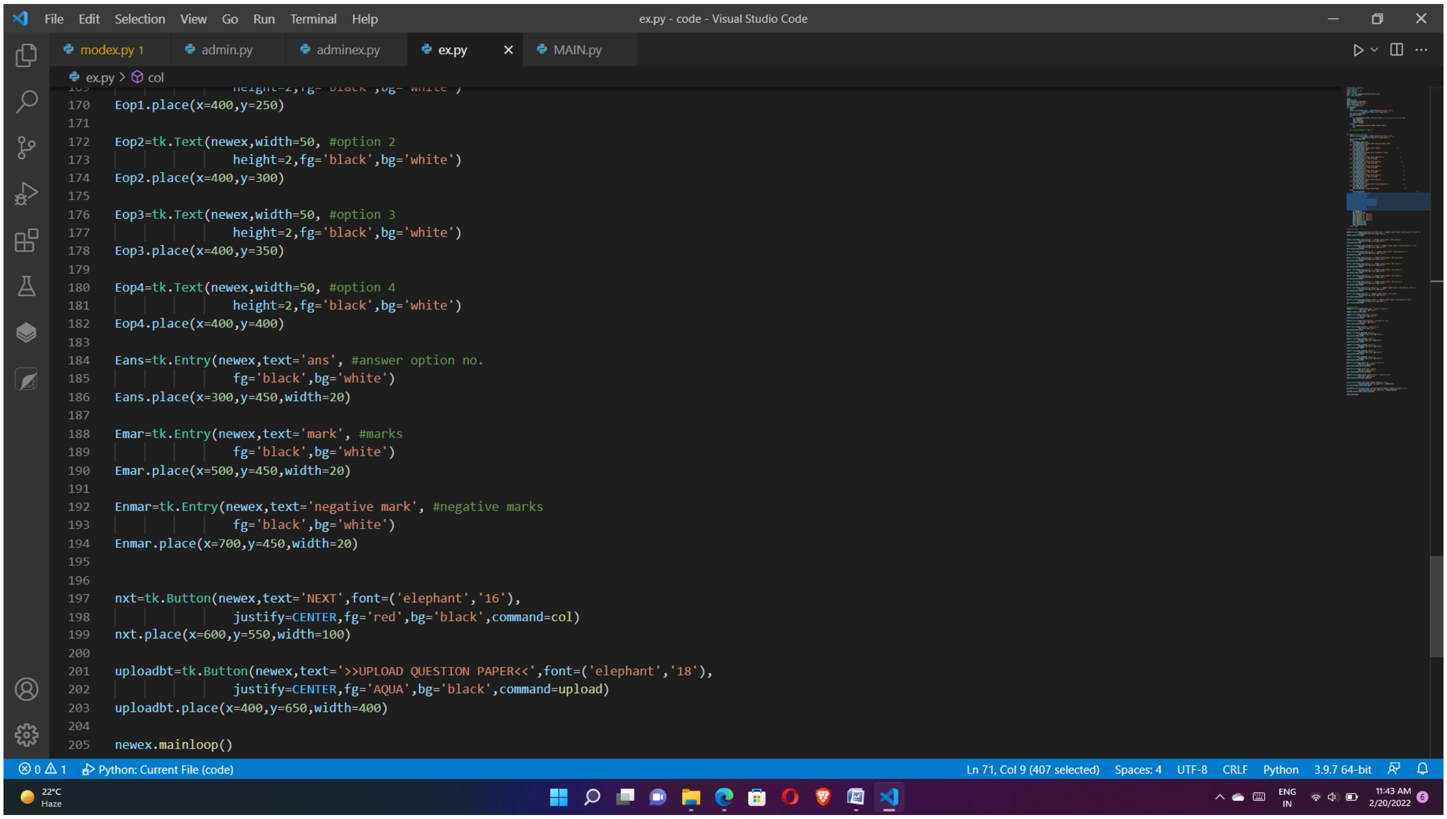
```python
            print('Username not found try another Username')


def admin1():
    admin12=tk.Tk()
    admin12.geometry('700x400')
    admin12.title("Welcome User")
    admin12.resizable(0,0)
    admin1img=ImageTk.PhotoImage(file=r'F:\main project\photo\teacher.jpg')
    teac=tk.Label(admin12,image=admin1img)
    teac.place(x=0,y=0)

    win1=tk.Label(admin12,text='Welcome To The Online Examination System',font=('elephant','20'),
                    justify= CENTER,fg='red')
    win1.place(x=10,y=50)
    examsec=tk.Button(admin12,text='>>EXAMINATIONS SECTION<<',font=('elephant'),
                        justify=CENTER,bg="black",fg="white",command=lambda:[admin12.destroy(),adminex.exam1()])
    examsec.place(x=100 ,y=200 )
    cansec=tk.Button(admin12,text='>>CANDIDATES SECTION<<',font=('elephant'),
                        justify=CENTER,bg="black",fg="white",command=lambda:[admin12.destroy(),adminex.cand()])
    cansec.place(x=100 ,y=250 )
    ussec=tk.Button(admin12,text='>>USER SECTION<<',font=('elephant'),
                        justify=CENTER,bg="black",fg="white",command=lambda:[admin12.destroy(),adminex.user()])
    ussec.place(x=100 ,y=300 )
    ret=tk.Button(admin12,text='GO BACK',font=('impact','8'),
                justify=CENTER,fg='red',command=lambda:[admin12.destroy(),calad._clad(calad)])
    ret.place(x=550,y=350)
    admin12.mainloop()



admin = tk.Tk()
admin.geometry("300x300")
admin.title("Login")
admin.resizable(0,0)
admin.minsize(626,417)
adminimg=ImageTk.PhotoImage(file=r'F:\main project\photo\chemistryimage.jpg')
```



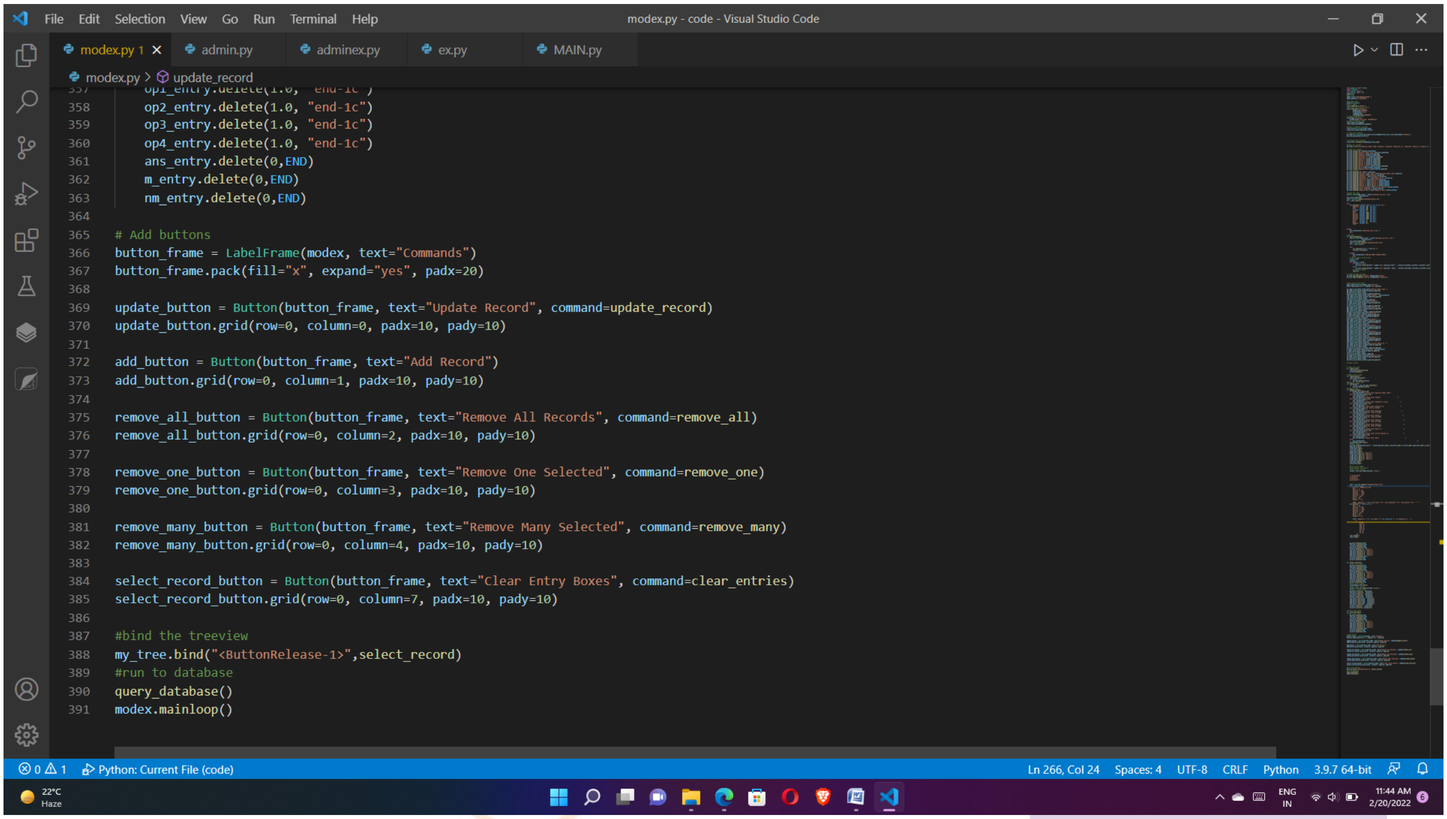
```python



admin = tk.Tk()
admin.geometry("300x300")
admin.title("Login")
admin.resizable(0,0)
admin.minsize(626,417)
adminimg=ImageTk.PhotoImage(file=r'F:\main project\photo\chemistryimage.jpg')
img1=tk.Label(admin,image=adminimg)
img1.place(x=0,y=0)


# Defining the first row
lblfrstrow = tk.Label(admin, text ="Username -", font=("Monotype Corsiva",'18'),
                justify=CENTER, fg="blue" )
lblfrstrow.place(x = 100, y = 40)

Username = tk.Entry(admin, width = 50,font=('Monotype Corsiva','14'))
Username.place(x = 300, y = 40, width = 180)

lblsecrow = tk.Label(admin, text ="Password - ",font=("Monotype Corsiva",'18'),
                justify=CENTER, fg="blue")
lblsecrow.place(x = 100, y = 100)

password = tk.Entry(admin, width = 50,font=('Monotype Corsiva','14'))
password.place(x = 300, y = 100, width = 180)

submitbtn = tk.Button(admin, text ="Login",font='elephant',
                      bg ='blue', command = submitact)
submitbtn.place(x = 300, y = 200, width = 55)
ret=tk.Button(admin,text='GO BACK',font=('impact','8'),
                justify=CENTER,fg='red',command=lambda:[adclose(),clmain()])
ret.place(x=550,y=350)
admin.mainloop()
```



```python
from PIL import Image, ImageTk
def cladmin():
    import admin
def ex():
    import ex
def mod():
    import modex

def exam1():
    exam=tk.Tk()
    exam.geometry('600x372')
    exam.title('EXAMINATIONS SECTION')
    exam.resizable(0,0)
    examimg=ImageTk.PhotoImage(file=r'F:\main project\photo\exam.jpg')
    exm=tk.Label(exam,image=examimg)
    exm.place(x=0,y=20)
    ex1=tk.Label(exam,text='Welcome To Examination Section',font=('elephant','20'),
                justify= CENTER,fg='black')
    ex1.place(x=10,y=50)
    exam1=tk.Button(exam,text='**NEW EXAM**',font=('impact'),
                justify=CENTER,fg='white',bg="black",command=lambda:[exam.destroy(),ex()])
    exam1.place(x=100,y=100)
    exam4=tk.Button(exam,text='**EXAM MODIFICATION**',font=('impact'),
                justify=CENTER,fg='white',bg="black",command=lambda:[exam.destroy(),mod()])
    exam4.place(x=100,y=150)
    exam2=tk.Button(exam,text='**SCHEDULES OF EXAM**',font=('impact'),
                justify=CENTER,fg='white',bg="black")
    exam2.place(x=100,y=200)
    exam3=tk.Button(exam,text='**PREVIOUS EXAM RECORDS',font=('impact'),
                justify=CENTER,fg='white',bg="black")
    exam3.place(x=100,y=250)
    ret1=tk.Button(exam,text='GO BACK',font=('impact','8'),
            justify=CENTER,fg='red',command=lambda:[exam.destroy(),cladmin()])
    ret1.place(x=540,y=330)
    exam.mainloop()
def cand():
    cand=tk.Tk()
```

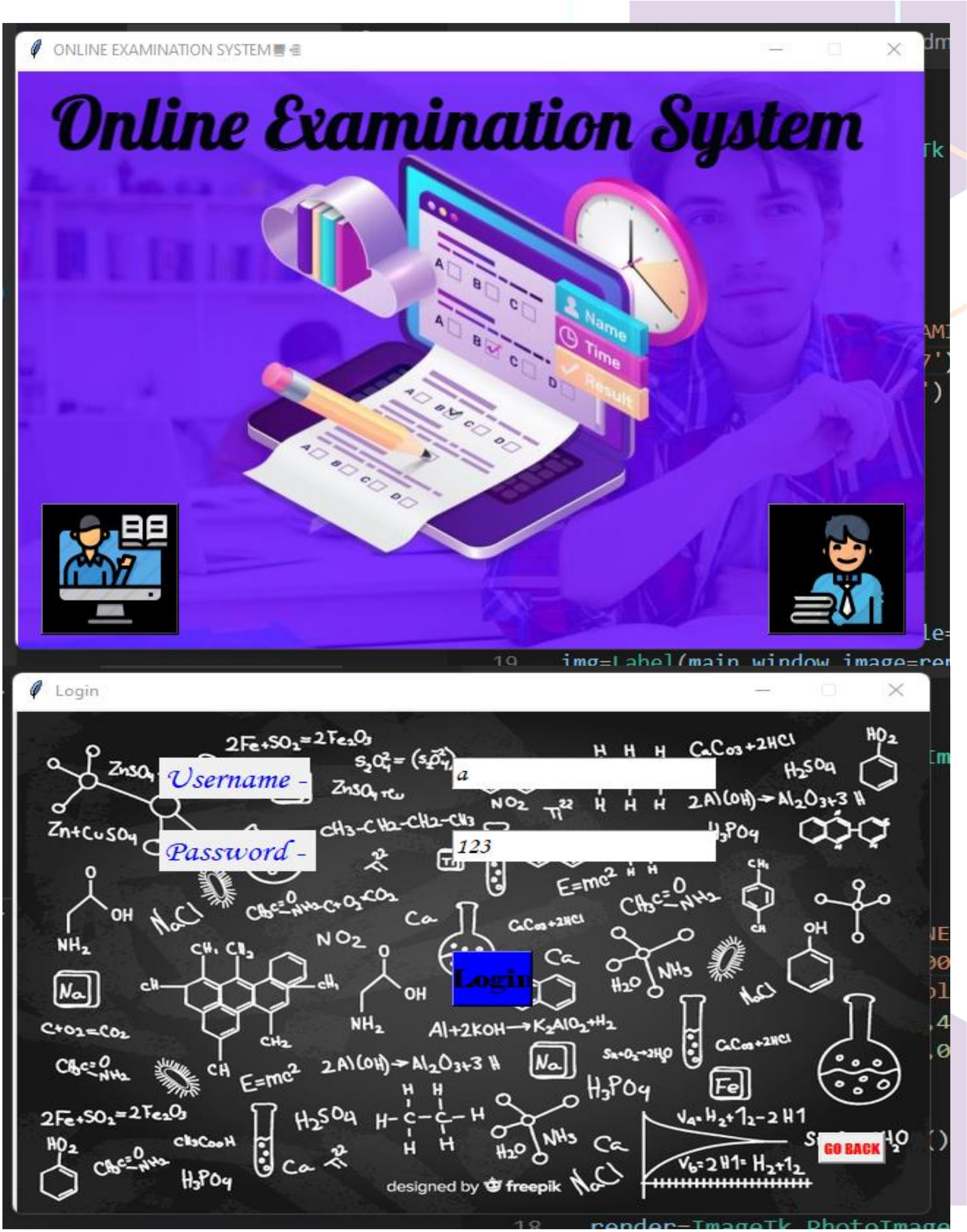

ex.py - code - Visual Studio Code
modex.py 1
admin.py
adminex.py
ex.py
MAIN.py
ex.py > col
Ln 71, Col 9 (407 selected)
Spaces: 4
UTF-8
CRLF
Python
3.9.7 64-bit
Python: Current File (code)

```python
                height=2,fg='black',bg='white')
Eop1.place(x=400,y=250)

Eop2=tk.Text(newex,width=50, #option 2
                height=2,fg='black',bg='white')
Eop2.place(x=400,y=300)

Eop3=tk.Text(newex,width=50, #option 3
                height=2,fg='black',bg='white')
Eop3.place(x=400,y=350)

Eop4=tk.Text(newex,width=50, #option 4
                height=2,fg='black',bg='white')
Eop4.place(x=400,y=400)

Eans=tk.Entry(newex,text='ans', #answer option no.
                fg='black',bg='white')
Eans.place(x=300,y=450,width=20)

Emar=tk.Entry(newex,text='mark', #marks
                fg='black',bg='white')
Emar.place(x=500,y=450,width=20)

Enmar=tk.Entry(newex,text='negative mark', #negative marks
                fg='black',bg='white')
Enmar.place(x=700,y=450,width=20)


nxt=tk.Button(newex,text='NEXT',font=('elephant','16'),
                justify=CENTER,fg='red',bg='black',command=col)
nxt.place(x=600,y=550,width=100)

uploadbt=tk.Button(newex,text='>>UPLOAD QUESTION PAPER<<',font=('elephant','18'),
                justify=CENTER,fg='AQUA',bg='black',command=upload)
uploadbt.place(x=400,y=650,width=400)

newex.mainloop()
```



```python
    op1_entry.delete(1.0, "end-1c")
    op2_entry.delete(1.0, "end-1c")
    op3_entry.delete(1.0, "end-1c")
    op4_entry.delete(1.0, "end-1c")
    ans_entry.delete(0,END)
    m_entry.delete(0,END)
    nm_entry.delete(0,END)

# Add buttons
button_frame = LabelFrame(modex, text="Commands")
button_frame.pack(fill="x", expand="yes", padx=20)

update_button = Button(button_frame, text="Update Record", command=update_record)
update_button.grid(row=0, column=0, padx=10, pady=10)

add_button = Button(button_frame, text="Add Record")
add_button.grid(row=0, column=1, padx=10, pady=10)

remove_all_button = Button(button_frame, text="Remove All Records", command=remove_all)
remove_all_button.grid(row=0, column=2, padx=10, pady=10)

remove_one_button = Button(button_frame, text="Remove One Selected", command=remove_one)
remove_one_button.grid(row=0, column=3, padx=10, pady=10)

remove_many_button = Button(button_frame, text="Remove Many Selected", command=remove_many)
remove_many_button.grid(row=0, column=4, padx=10, pady=10)

select_record_button = Button(button_frame, text="Clear Entry Boxes", command=clear_entries)
select_record_button.grid(row=0, column=7, padx=10, pady=10)

#bind the treeview
my_tree.bind("<ButtonRelease-1>",select_record)
#run to database
query_database()
modex.mainloop()
```

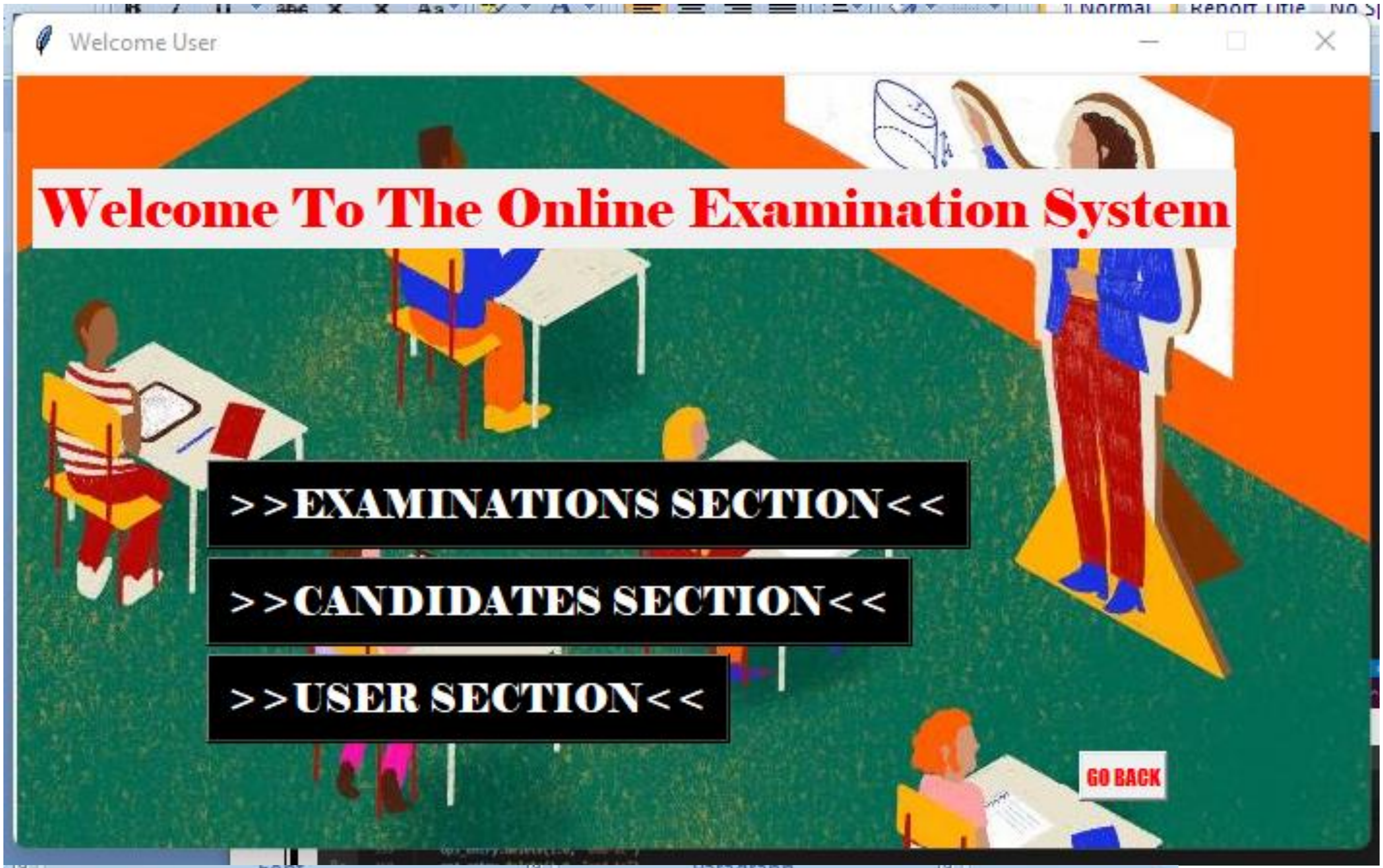
# *OUTPUT*



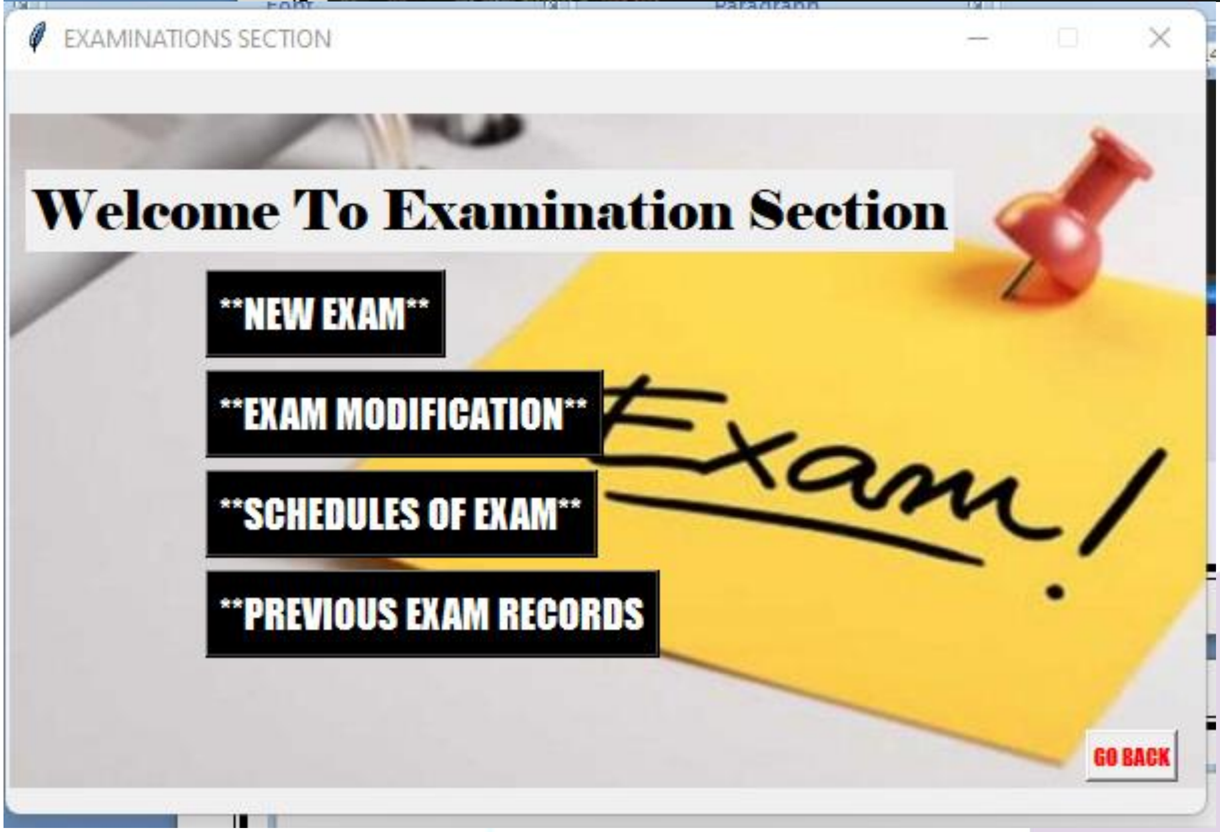
Username - a

Password - 123

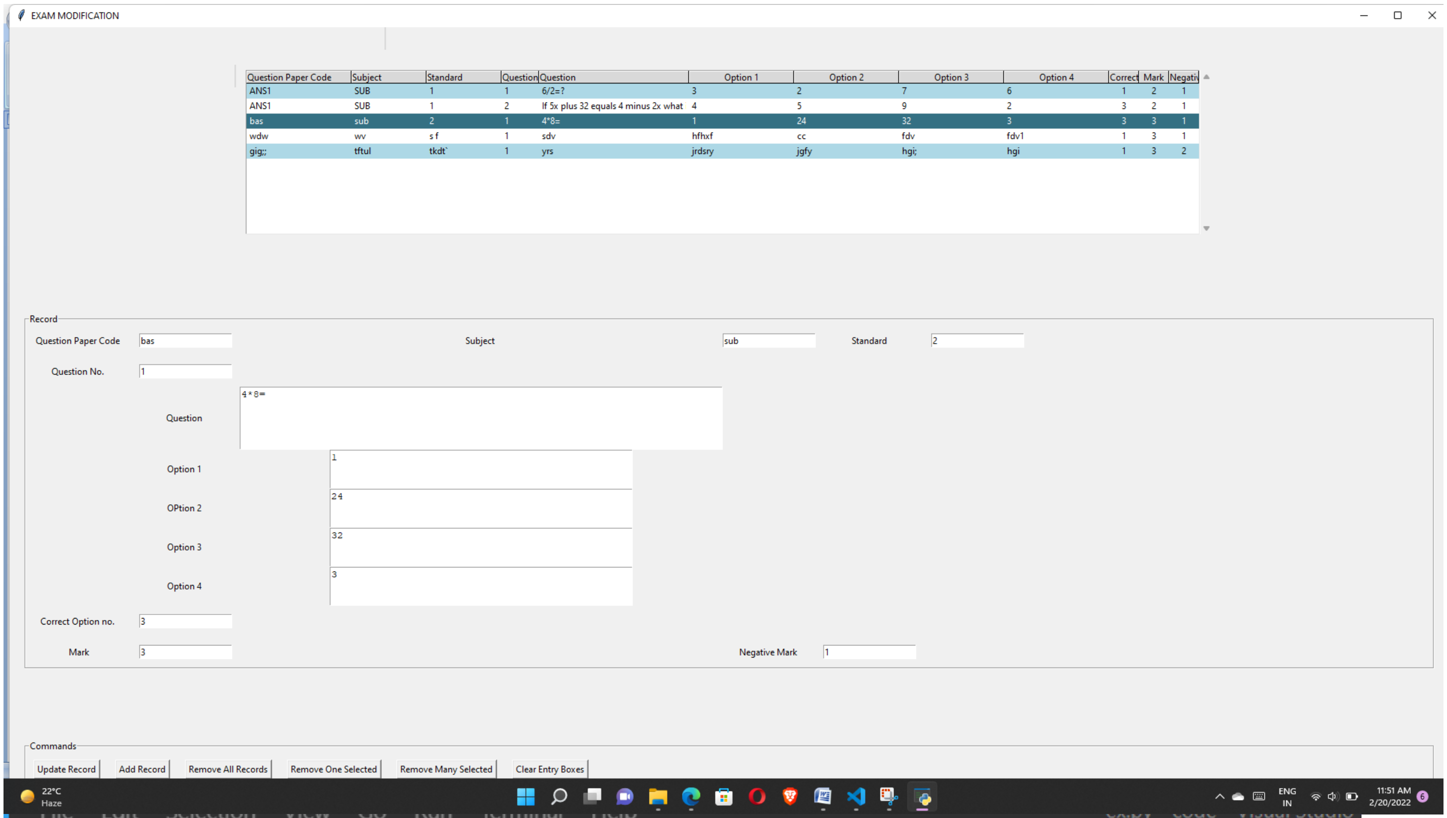
```
img=Label(main_window,image=ren
render=ImageTk.PhotoImage
```



Welcome To The Online Examination System



Welcome To Examination Section



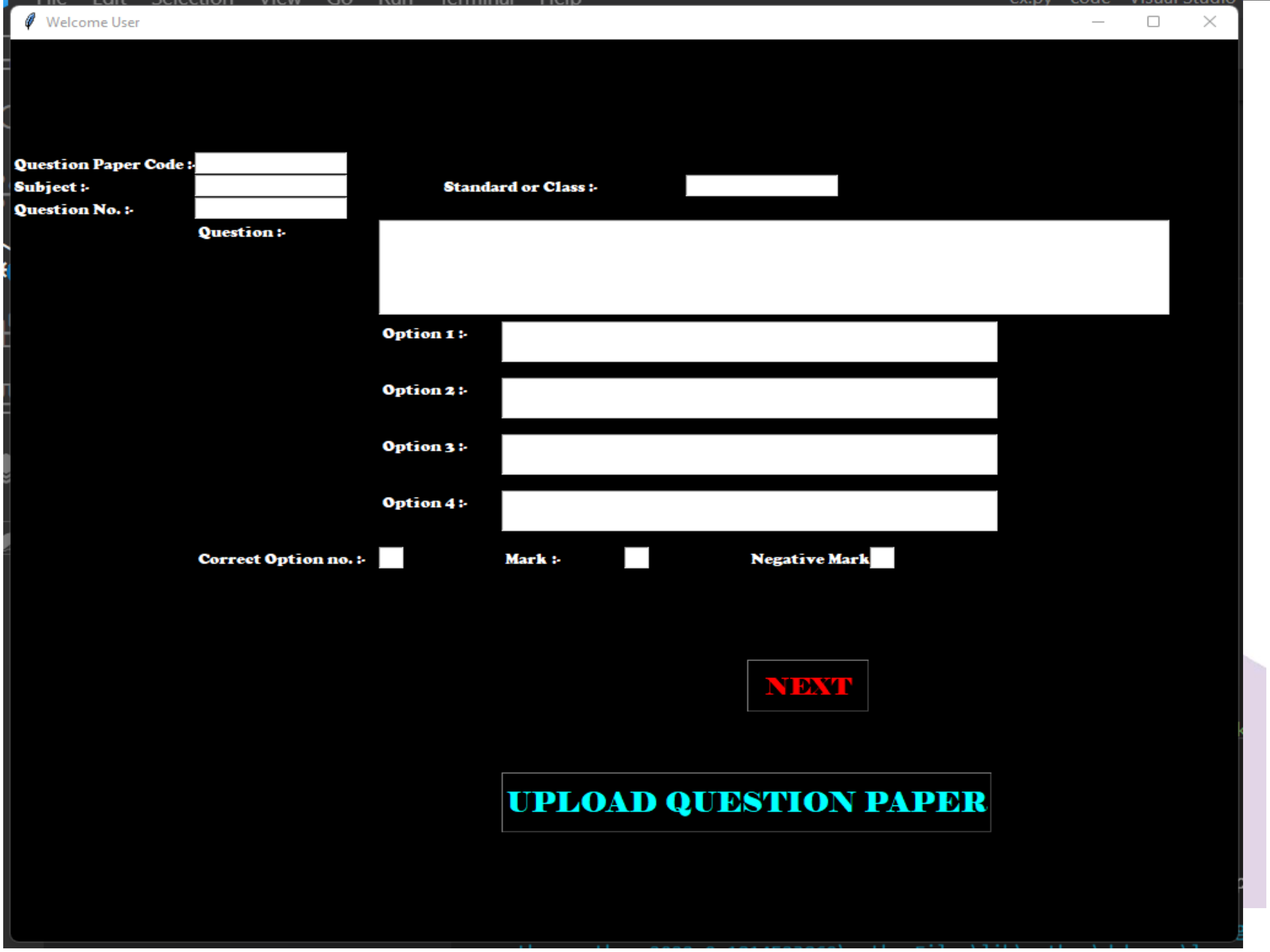
EXAM MODIFICATION

| Question Paper Code | Subject | Standard | Question | Question | Option 1 | Option 2 | Option 3 | Option 4 | Correct | Mark | Negativ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ANS1 | SUB | 1 | 1 | 6/2=? | 3 | 2 | 7 | 6 | 1 | 2 | 1 |
| ANS1 | SUB | 1 | 2 | If 5x plus 32 equals 4 minus 2x what | 4 | 5 | 9 | 2 | 3 | 2 | 1 |
| bas | sub | 2 | 1 | 4*8= | 1 | 24 | 32 | 3 | 3 | 3 | 1 |
| wdw | wv | s f | 1 | sdv | hfhxf | cc | fdv | fdv1 | 1 | 3 | 1 |
| gig;; | tftul | tkdt` | 1 | yrs | jrdsry | jgfy | hgi; | hgi | 1 | 3 | 2 |

Record

Question Paper Code: bas
Subject: sub
Standard: 2
Question No.: 1
Question: 4*8=
Option 1: 1
OPtion 2: 24
Option 3: 32
Option 4: 3
Correct Option no.: 3
Mark: 3
Negative Mark: 1

Commands



Welcome User

Question Paper Code :-
Subject :-
Question No. :-
Standard or Class :-
Question :-
Option 1 :-
Option 2 :-
Option 3 :-
Option 4 :-
Correct Option no. :-
Mark :-
Negative Mark