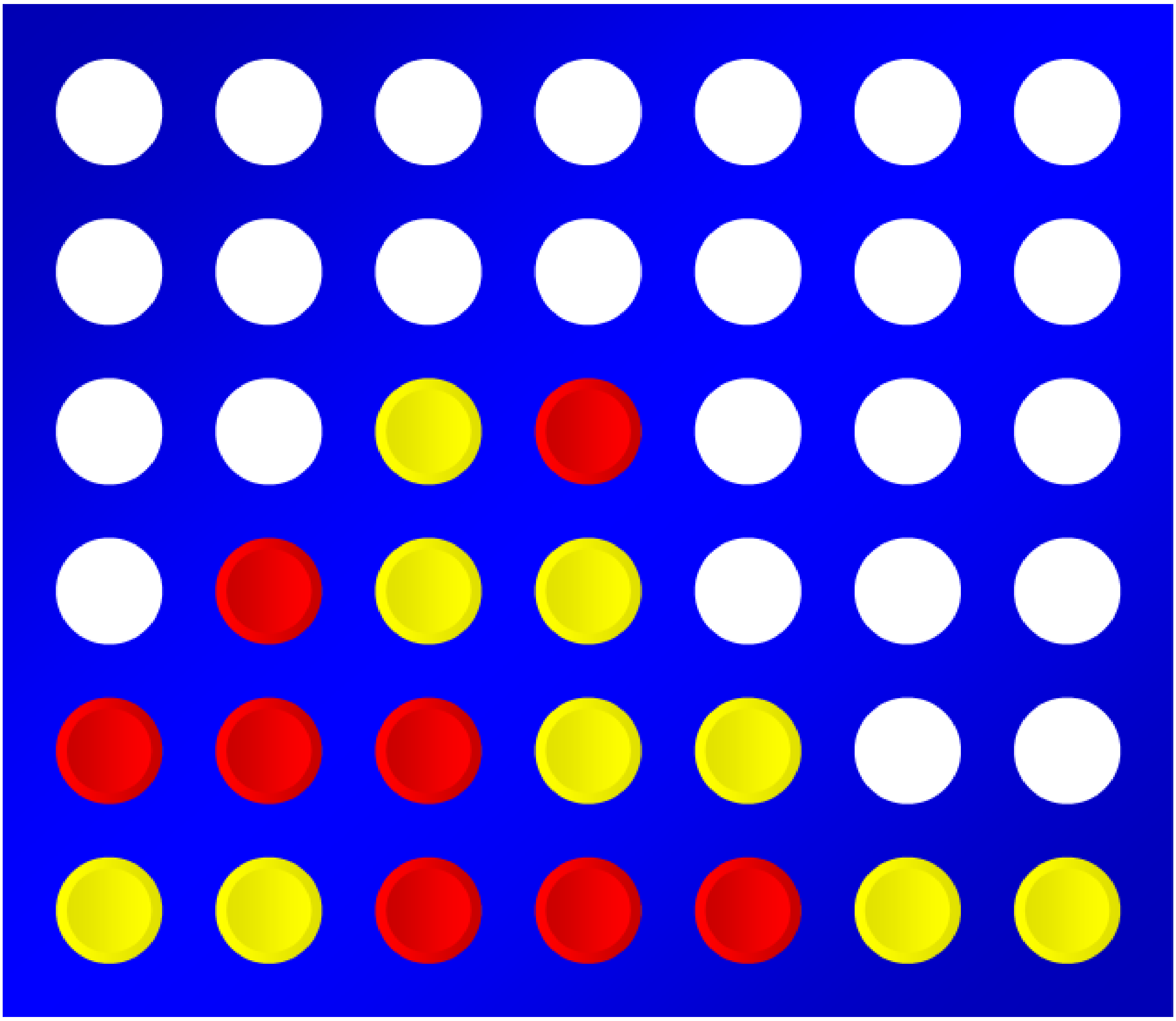
Connect Four

(Mini Project)



(Subject: Python for Computational Problem Solving)

Problem Statement:

Building a fully functional two-player game of Connect Four that will ask users for:

1. Their input:

   Designed using BreezyPythonGUI.

2. Verify that the input is valid:

   Using nested loops, user-defined functions in Python 3.11

3. Determine if the somebody has won the game:

   Using user-defined functions, saving data in DBMS(MySQL)

4. And display the board:

   Using MySQL, BreezyPythonGUI and ‘tabulate’ module.

Project Description:

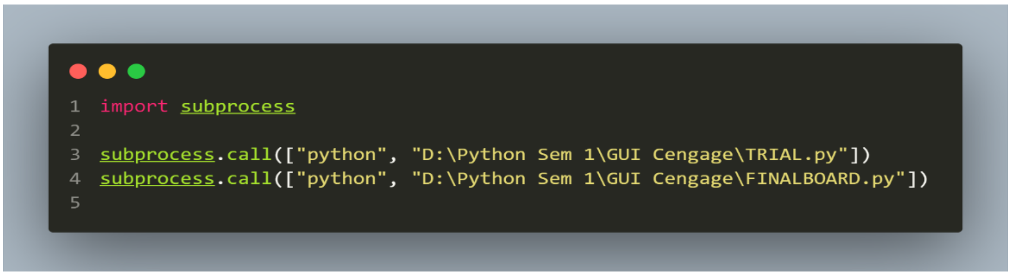
1. Welcome to my project on the classic strategy game, Connect Four.
2. Connect Four is a two-player strategy game that has a 6x7 grid and each player is given 21 coins of a colour (Red and Yellow respectively).
3. The players take alternate turns to play and the goal is to connect four coins of the same colour vertically, horizontally or diagonally.
4. During his/her turn, the player can select the column in which he wishes to place the coin, and the coin gets placed in the bottommost empty slot in the selected column.
5. The player who connects four coins of his colour first wins the game**.**

Program Modules Used:

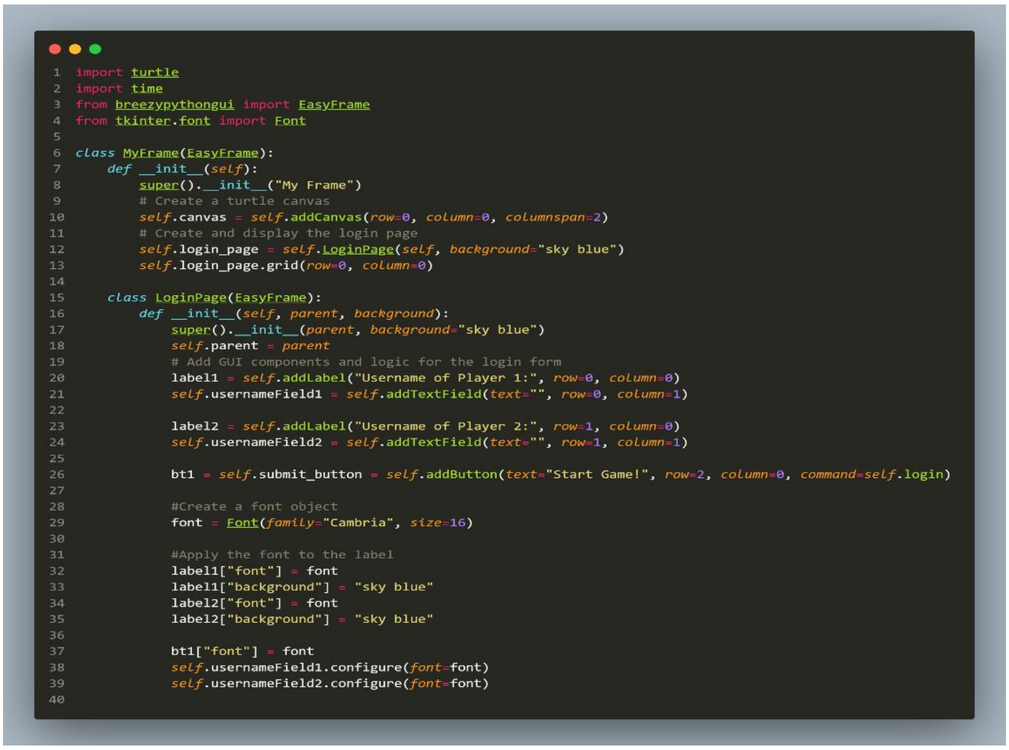
1. Turtle
2. Tkinter
3. BreezyPythonGUI
4. Time
5. Subprocess
6. Tabulate
7. mysql.connector
8. sys

Code Snippets:

Main file which invokes the other files.

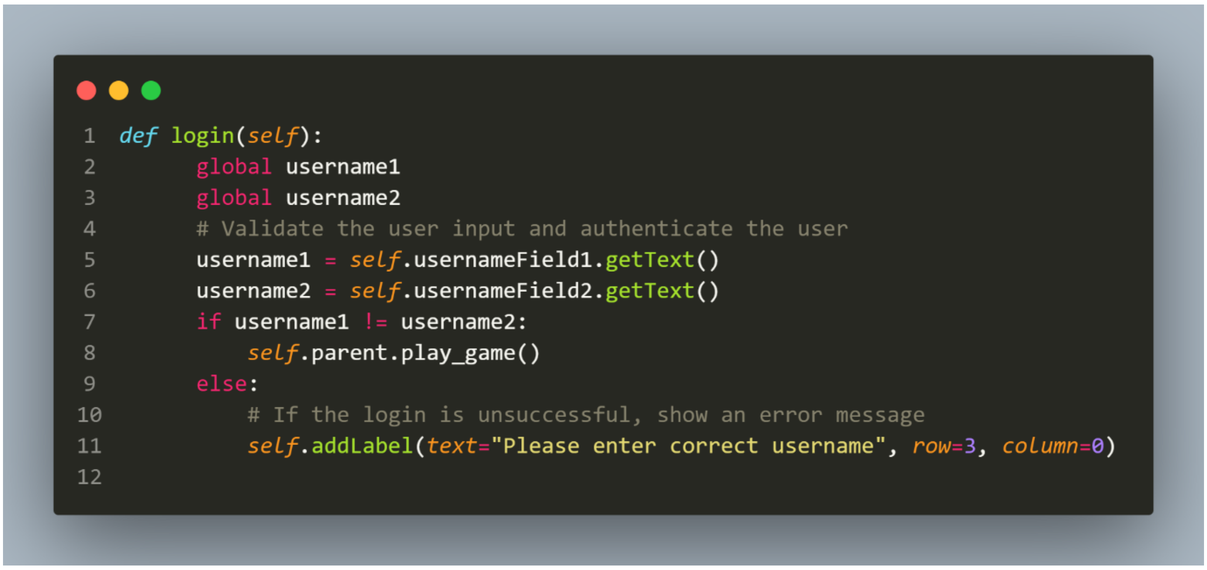


```python
import subprocess

subprocess.call(["python", "D:\Python Sem 1\GUI Cengage\TRIAL.py"])
subprocess.call(["python", "D:\Python Sem 1\GUI Cengage\FINALBOARD.py"])

```

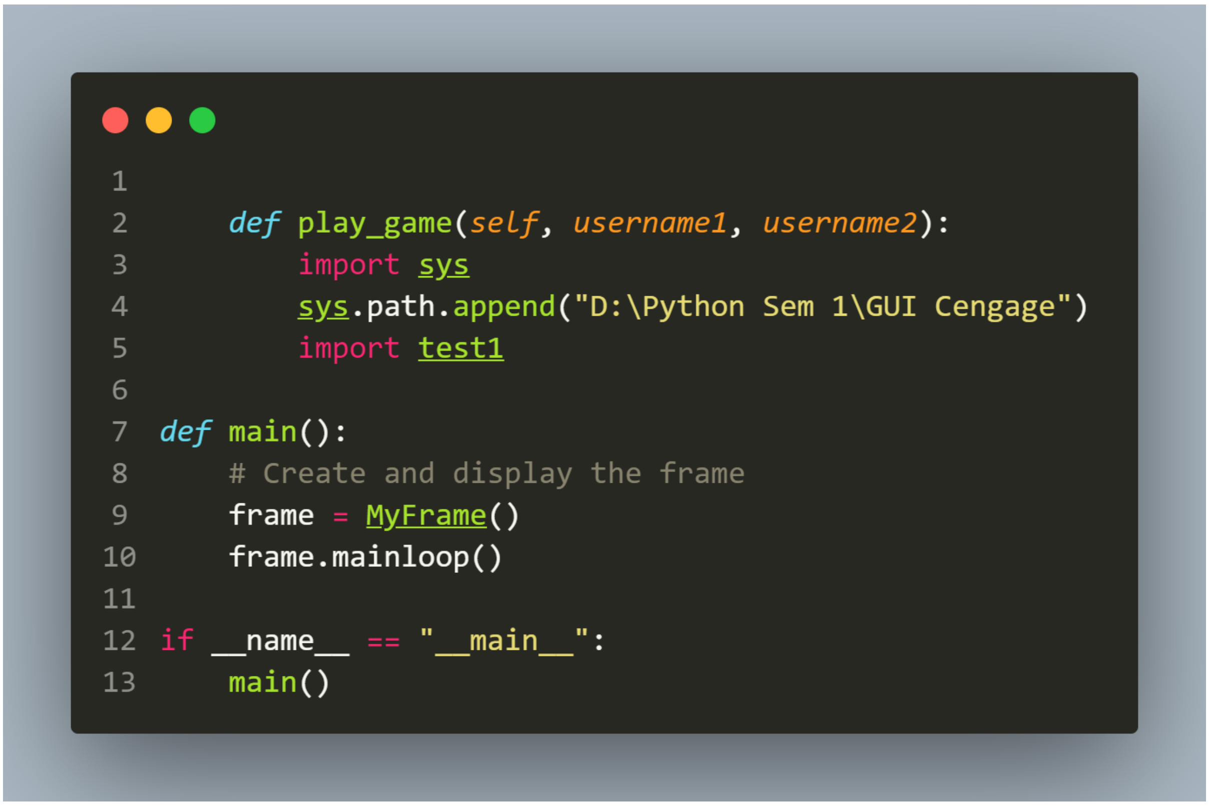


```python
import turtle
import time
from breezypythongui import EasyFrame
from tkinter.font import Font

class MyFrame(EasyFrame):
    def __init__(self):
        super().__init__("My Frame")
        # Create a turtle canvas
        self.canvas = self.addCanvas(row=0, column=0, columnspan=2)
        # Create and display the login page
        self.login_page = self.LoginPage(self, background="sky blue")
        self.login_page.grid(row=0, column=0)

    class LoginPage(EasyFrame):
        def __init__(self, parent, background):
            super().__init__(parent, background="sky blue")
            self.parent = parent
            # Add GUI components and logic for the login form
            label1 = self.addLabel("Username of Player 1:", row=0, column=0)
            self.usernameField1 = self.addTextField(text="", row=0, column=1)

            label2 = self.addLabel("Username of Player 2:", row=1, column=0)
            self.usernameField2 = self.addTextField(text="", row=1, column=1)

            bt1 = self.submit_button = self.addButton(text="Start Game!", row=2, column=0, command=self.login)

            #Create a font object
            font = Font(family="Cambria", size=16)

            #Apply the font to the label
            label1["font"] = font
            label1["background"] = "sky blue"
            label2["font"] = font
            label2["background"] = "sky blue"

            bt1["font"] = font
            self.usernameField1.configure(font=font)
            self.usernameField2.configure(font=font)

```

Login Page Validation:

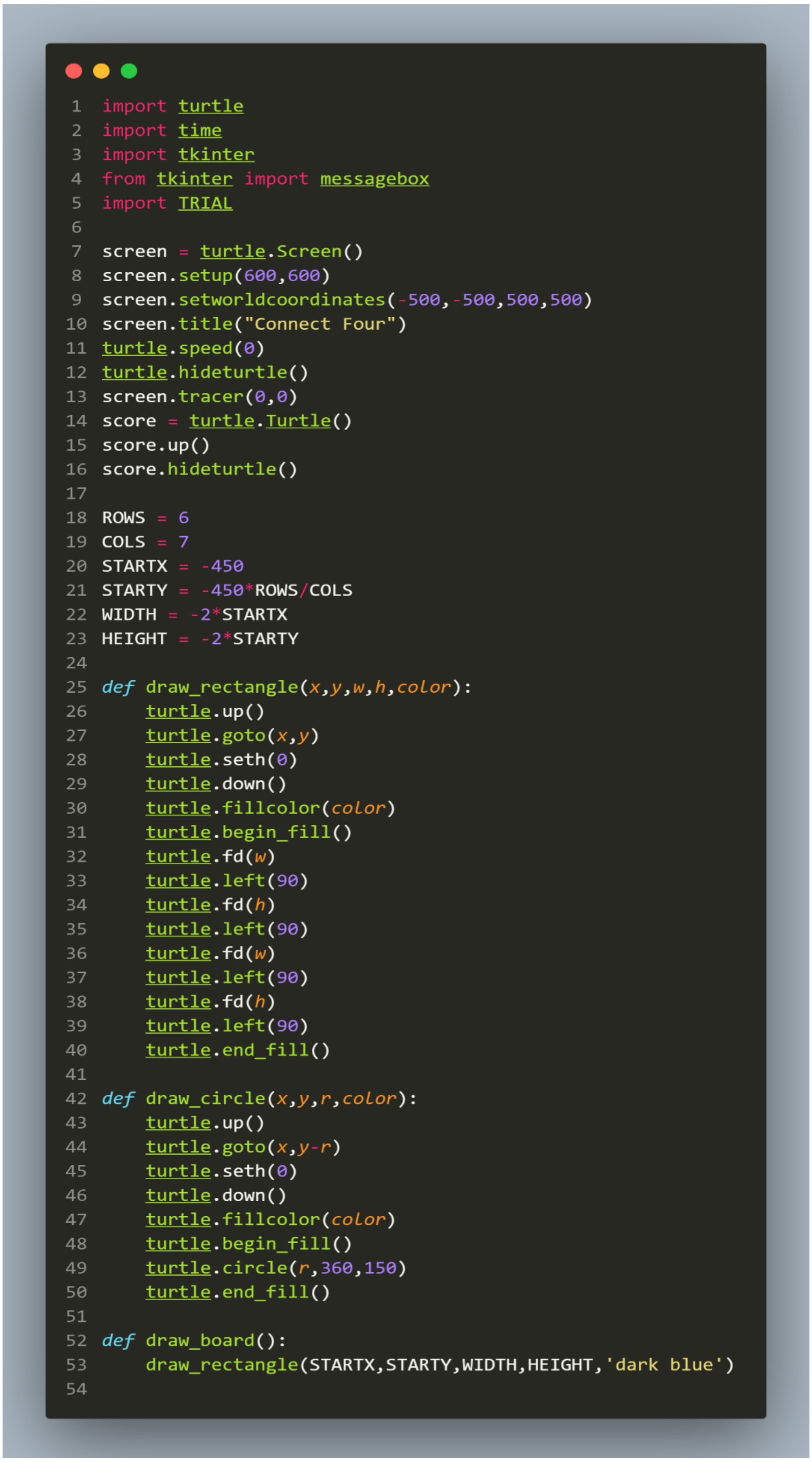


```python
def login(self):
      global username1
      global username2
      # Validate the user input and authenticate the user
      username1 = self.usernameField1.getText()
      username2 = self.usernameField2.getText()
      if username1 != username2:
          self.parent.play_game()
      else:
          # If the login is unsuccessful, show an error message
          self.addLabel(text="Please enter correct username", row=3, column=0)

```

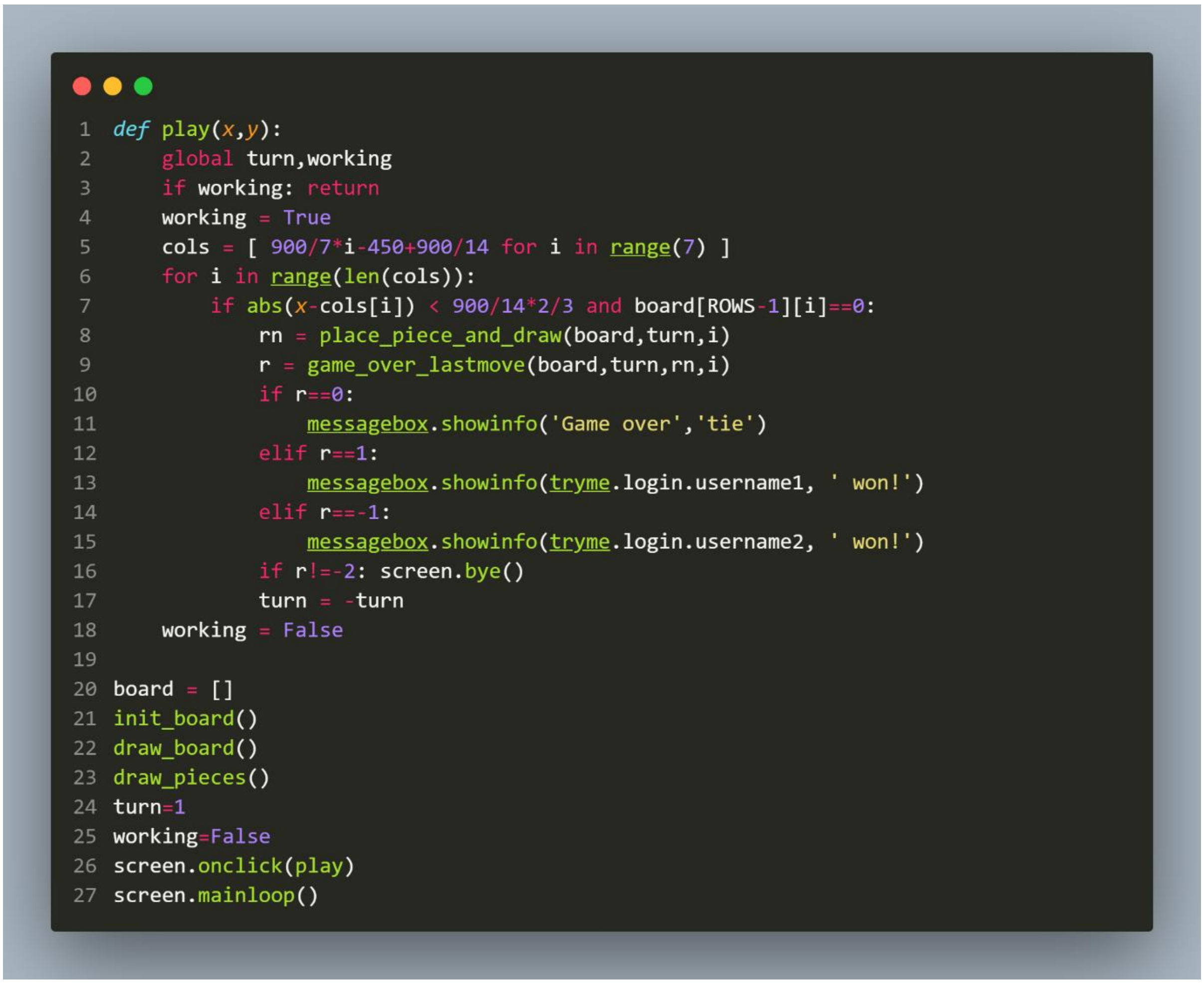


```python

    def play_game(self, username1, username2):
        import sys
        sys.path.append("D:\Python Sem 1\GUI Cengage")
        import test1

def main():
    # Create and display the frame
    frame = MyFrame()
    frame.mainloop()

if __name__ == "__main__":
    main()
```

Game code:

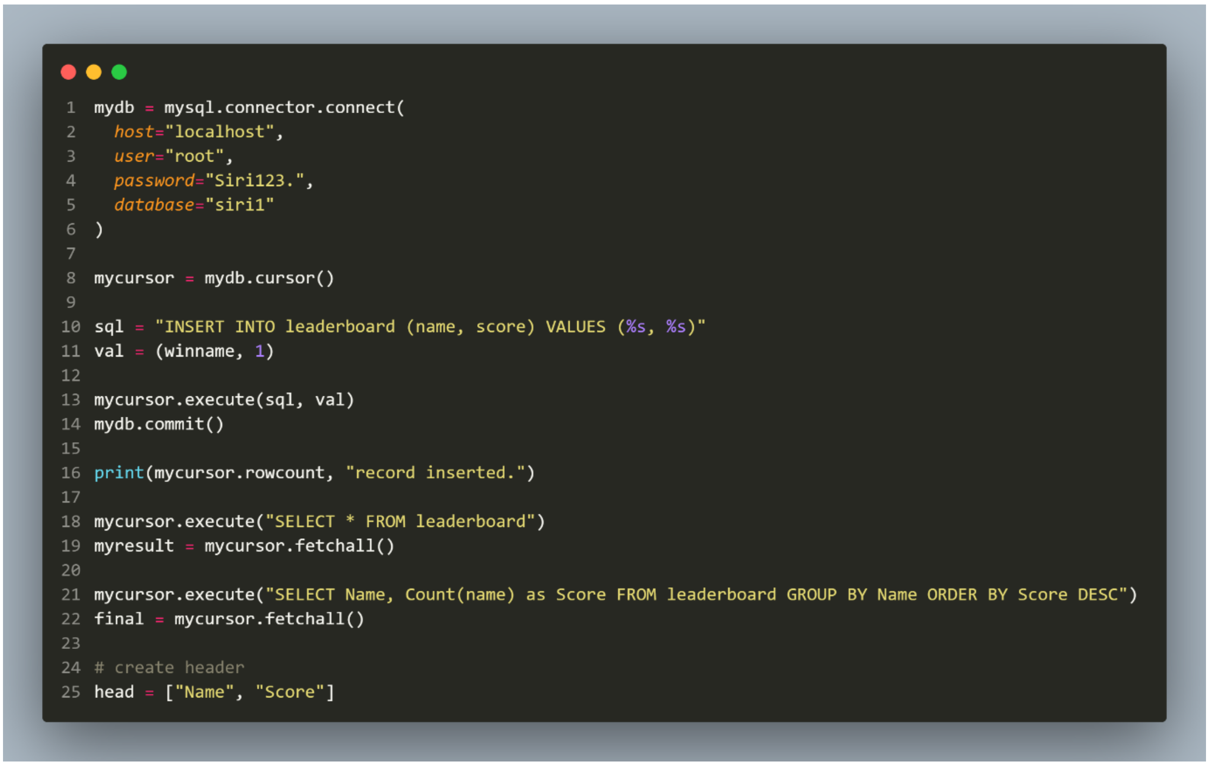


```python
import turtle
import time
import tkinter
from tkinter import messagebox
import TRIAL

screen = turtle.Screen()
screen.setup(600,600)
screen.setworldcoordinates(-500,-500,500,500)
screen.title("Connect Four")
turtle.speed(0)
turtle.hideturtle()
screen.tracer(0,0)
score = turtle.Turtle()
score.up()
score.hideturtle()

ROWS = 6
COLS = 7
STARTX = -450
STARTY = -450*ROWS/COLS
WIDTH = -2*STARTX
HEIGHT = -2*STARTY

def draw_rectangle(x,y,w,h,color):
    turtle.up()
    turtle.goto(x,y)
    turtle.seth(0)
    turtle.down()
    turtle.fillcolor(color)
    turtle.begin_fill()
    turtle.fd(w)
    turtle.left(90)
    turtle.fd(h)
    turtle.left(90)
    turtle.fd(w)
    turtle.left(90)
    turtle.fd(h)
    turtle.left(90)
    turtle.end_fill()

def draw_circle(x,y,r,color):
    turtle.up()
    turtle.goto(x,y-r)
    turtle.seth(0)
    turtle.down()
    turtle.fillcolor(color)
    turtle.begin_fill()
    turtle.circle(r,360,150)
    turtle.end_fill()

def draw_board():
    draw_rectangle(STARTX,STARTY,WIDTH,HEIGHT,'dark blue')

```



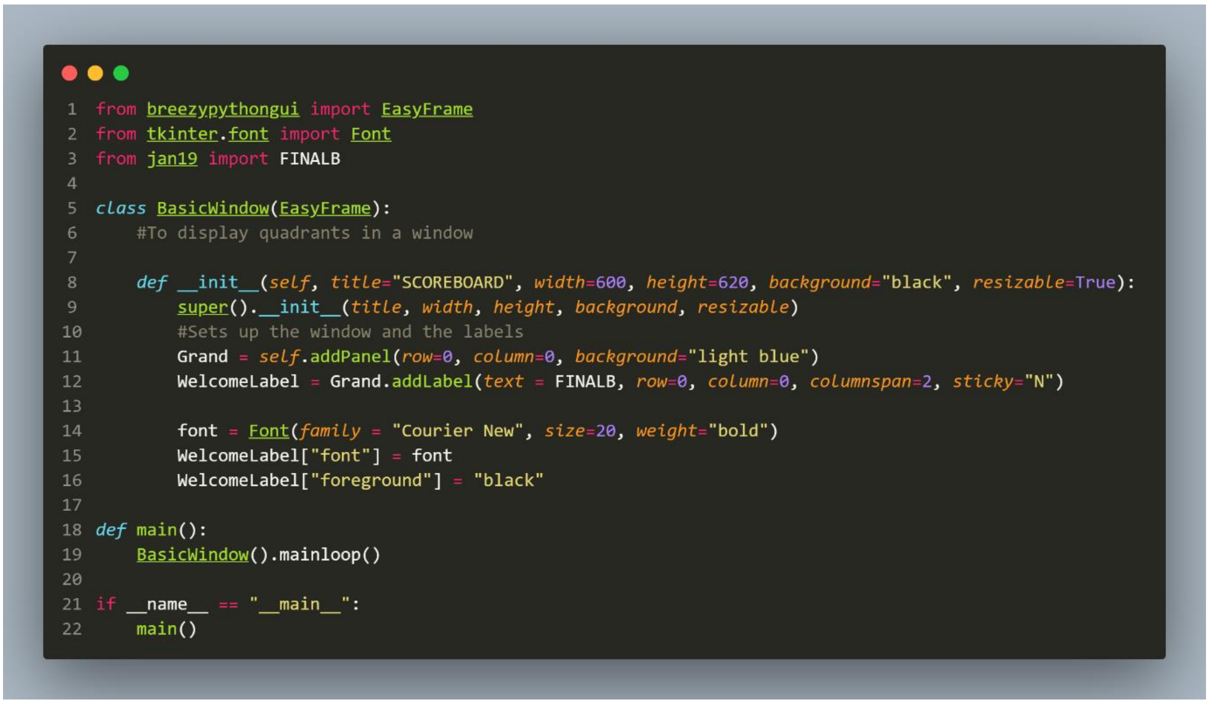
```python
def play(x,y):
    global turn,working
    if working: return
    working = True
    cols = [ 900/7*i-450+900/14 for i in range(7) ]
    for i in range(len(cols)):
        if abs(x-cols[i]) < 900/14*2/3 and board[ROWS-1][i]==0:
            rn = place_piece_and_draw(board,turn,i)
            r = game_over_lastmove(board,turn,rn,i)
            if r==0:
                messagebox.showinfo('Game over','tie')
            elif r==1:
                messagebox.showinfo(tryme.login.username1, ' won!')
            elif r==-1:
                messagebox.showinfo(tryme.login.username2, ' won!')
            if r!=-2: screen.bye()
            turn = -turn
    working = False

board = []
init_board()
draw_board()
draw_pieces()
turn=1
working=False
screen.onclick(play)
screen.mainloop()
```

Code which links MySQL database to Python



```python
mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="Siri123.",
  database="siri1"
)

mycursor = mydb.cursor()

sql = "INSERT INTO leaderboard (name, score) VALUES (%s, %s)"
val = (winname, 1)

mycursor.execute(sql, val)
mydb.commit()

print(mycursor.rowcount, "record inserted.")

mycursor.execute("SELECT * FROM leaderboard")
myresult = mycursor.fetchall()

mycursor.execute("SELECT Name, Count(name) as Score FROM leaderboard GROUP BY Name ORDER BY Score DESC")
final = mycursor.fetchall()

# create header
head = ["Name", "Score"]
```

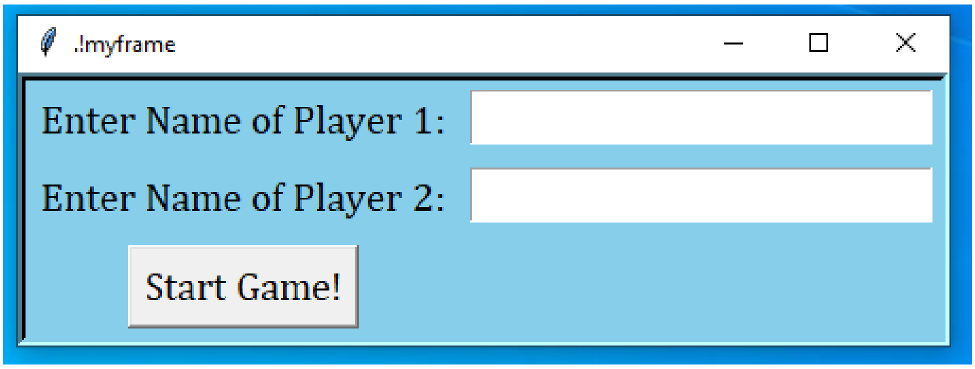
Displaying Scoreboard



```
from breezypythongui import EasyFrame
from tkinter.font import Font
from jan19 import FINALB

class BasicWindow(EasyFrame):
    #To display quadrants in a window

    def __init__(self, title="SCOREBOARD", width=600, height=620, background="black", resizable=True):
        super().__init__(title, width, height, background, resizable)
        #Sets up the window and the labels
        Grand = self.addPanel(row=0, column=0, background="light blue")
        WelcomeLabel = Grand.addLabel(text = FINALB, row=0, column=0, columnspan=2, sticky="N")

        font = Font(family = "Courier New", size=20, weight="bold")
        WelcomeLabel["font"] = font
        WelcomeLabel["foreground"] = "black"

def main():
    BasicWindow().mainloop()

if __name__ == "__main__":
    main()
```

Program Input:

1. Designed using BreezyPythonGUI and Tkinter. Font modules.
2. As soon as the user runs the program the game will start.
3. The program input for this game is the usernames of the two players.
4. The players are prompted to enter their usernames using two text fields, one for each player, on the login page.
5. The usernames are then validated to ensure that they are unique.

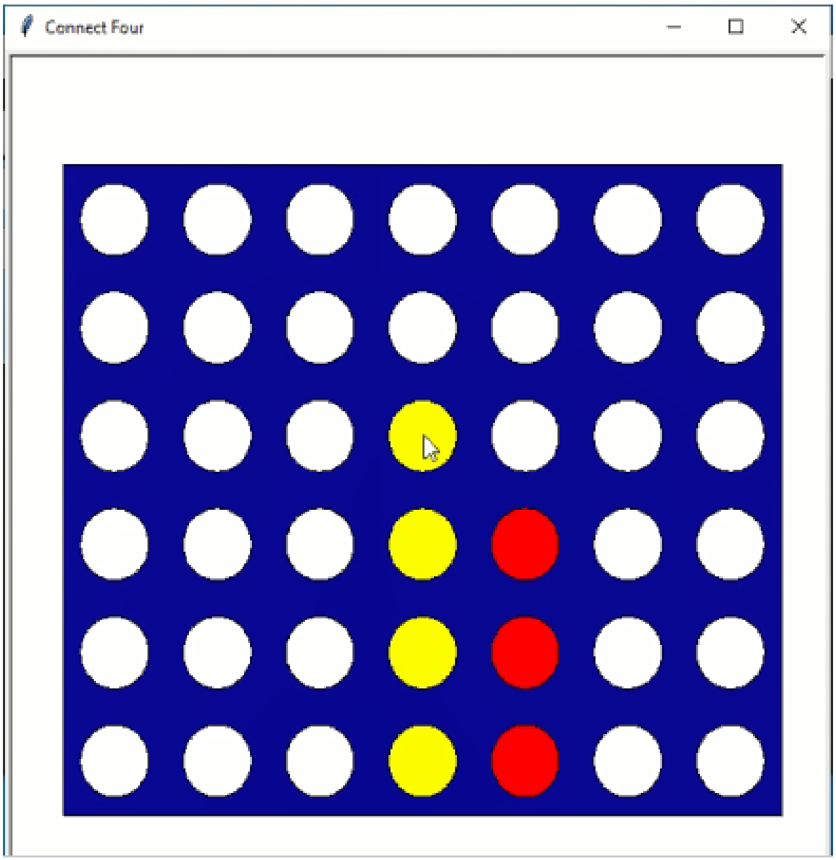


Program Output:

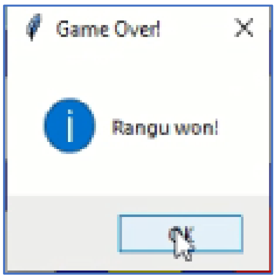
1. The game board is displayed on the screen. The turtle library is used to create the graphical elements, such as the board and the game pieces.
2. The game pieces are dropped into the board by the players, and they can be either yellow or red.
3. The condition for the winner (any one of the following)
    - four coins in horizontal row
    - four coins in vertical row
    - four coins in diagonal (top left to bottom right)
    - four coins in diagonal (top right to bottom left)

    This is done using nested for loops and def functions.
4. The game ends when one of the players connects four of their game pieces horizontally, vertically, or diagonally, and the winner is declared.
5. After the game ends, the winner's name and score are stored and a scoreboard is displayed on the screen showing the names of the previous winners and their scores.

Game Board



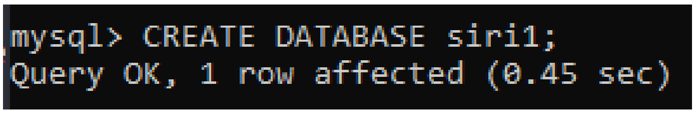
Pop-up box to display the winner’s name



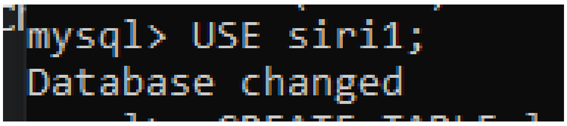
Game Over!

Rangu won!

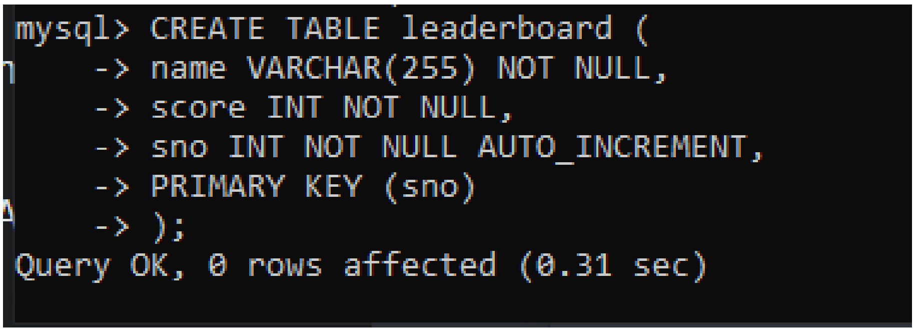
Database and Table created in MySQL:



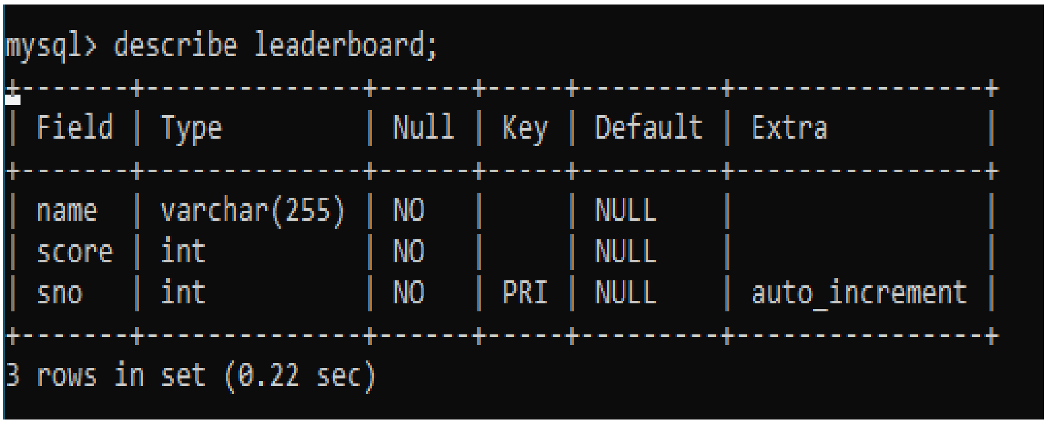
```
mysql> CREATE DATABASE siri1;
Query OK, 1 row affected (0.45 sec)
```



```
mysql> USE siri1;
Database changed
```

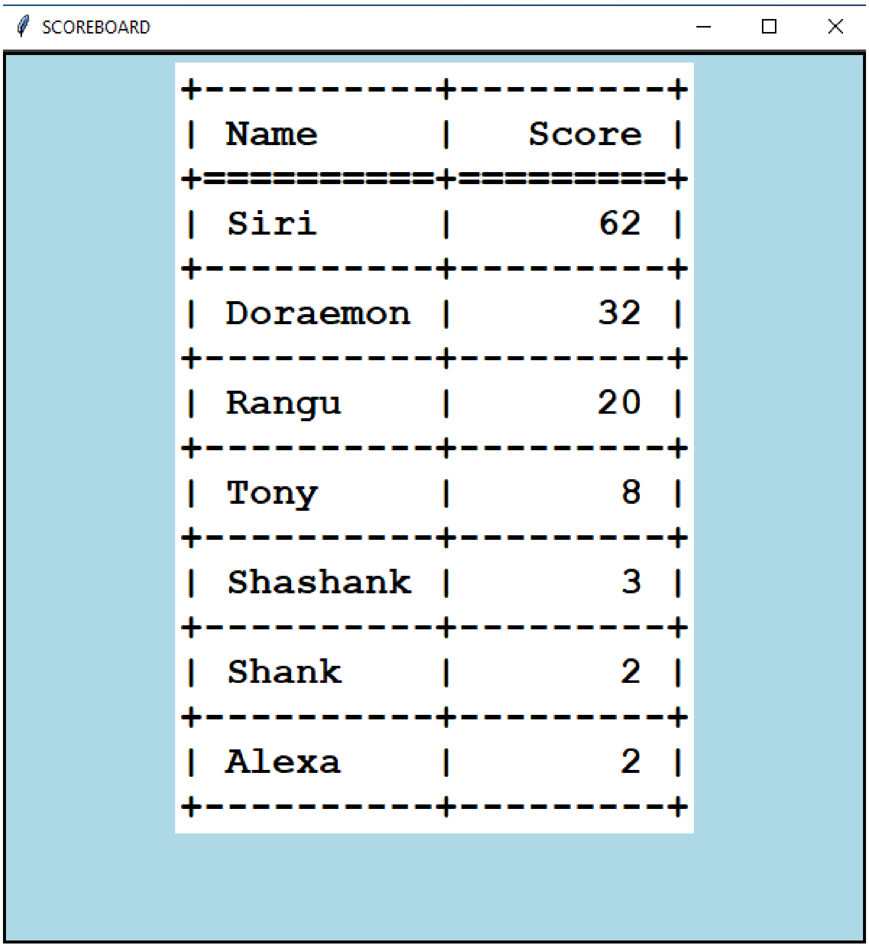


```
mysql> CREATE TABLE leaderboard (
    -> name VARCHAR(255) NOT NULL,
    -> score INT NOT NULL,
    -> sno INT NOT NULL AUTO_INCREMENT,
    -> PRIMARY KEY (sno)
    -> );
Query OK, 0 rows affected (0.31 sec)
```



```
mysql> describe leaderboard;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| name  | varchar(255) | NO   |     | NULL    |                |
| score | int          | NO   |     | NULL    |                |
| sno   | int          | NO   | PRI | NULL    | auto_increment |
+-------+--------------+------+-----+---------+----------------+
3 rows in set (0.22 sec)
```

Scoreboard:



```
+----------+---------+
| Name     |   Score |
+==========+=========+
| Siri     |      62 |
+----------+---------+
| Doraemon |      32 |
+----------+---------+
| Rangu    |      20 |
+----------+---------+
| Tony     |       8 |
+----------+---------+
| Shashank |       3 |
+----------+---------+
| Shank    |       2 |
+----------+---------+
| Alexa    |       2 |
+----------+---------+
```

Acknowledgements:

1. I would like to express my heartfelt gratitude to my respectable Assistant Professor, **Swathy Mugundan**, for her guidance and support throughout this project. Her invaluable insights and expertise were instrumental in helping me to understand the project and to bring it to completion.

2. I would like to extend my sincere appreciation to the following resources and books that helped me in the completion of this project.

   - Fundamentals of Python, by Kenneth A Lambert
   - PESUAcademy: https://www.pesuacademy.com/Academy/
   - W3Schools: https://www.w3schools.com/ to learn MySQL and tabulate module

Conclusion:

1. This project has successfully implemented a Connect Four game using Python programming language and GUI library BreezyPythonGUI. The game allows two players to play the game, and the game board is displayed on the screen where players can insert their coins. The game also implements four conditions to determine the winner of the game and stores the data of the winner in a database. The project also includes a leaderboard page which displays the winner's data.
2. Throughout the project, I have implemented various concepts of Python programming such as classes, functions, and GUI library.
3. In summary, this project has helped me to gain a deeper understanding of Python programming and GUI library, and also helped me to develop problem-solving skills. I believe that the skills and knowledge I have gained from this project will be beneficial for me in the future.

Done by:

Siri N Shetty - PES2UG22CS556

Semester-1, Section-I, Batch-I2

Branch – Computer Science and Engineering (CSE)