A

PROJECT REPORT

ON

**Hangman Game**

Towards partial fulfillment of the requirement in

**2nd Semester BCA/B.Sc. IT/IMCA 2021-22**

**Submitted By:-**

2205101100152 Dipesh Rauniyar

2205101100103 Raghav Mahajan

2205101100104 Rahul Kumar

**Submitted To:-**



**Parul Institute of Computer Application,**

**Parul University.**

Under the Guidance of

Prof. Hardik Parmar, <Designation>

**Acknowledgement**

*The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.*

*I respect and thank Dr. Priya Swaminarayan, Dean, FITCS for providing us an opportunity to do the project work in BCA and giving us all support and guidance which made us complete the project duly. We are extremely thankful to Mam for providing her support and guidance, although she had a busy schedule managing the academic affairs.*

*We would not forget to remember Prof. Hina Chokshi, HOD, BCA department for her encouragement and more over for her timely support and guidance till the completion of our project work.*

*We owe our deep gratitude to our project guide Prof. Hardik Parmar , who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.*

*I am thankful to and fortunate enough to get constant encouragement, support and guidance from our Parents, all Teaching staff of the BCA Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.*

***Dipesh Rauniyar 2205101100152***

***Raghav Mahajan 2205101100103***

***Rahul Kumar 2205101100104***



**PARUL INSTITUTE OF COMPUTER APPLICATION**

**CERTIFICATE**

This is to certify that ***Dipesh Rauniyar, Raghav Mahajan and Rahul Kumar*** the student(s) of Parul Institute of Computer Application, has/have satisfactorily completed the project entitled ***“ Hangman Game ”*** as a part of course curriculum in BCA/B.Sc. IT/IMCA semester-II for the academic year 2021-2022 under guidance of ***Prof.Hardik Parmar.***

Enrollment Number: 2205101100152

Enrollment Number: 2205101100103

Enrollment Number: 2205101100104

| **Quality of work** | **Grade** | **Sign of Internal guide** |
| --- | --- | --- |
| **Poor / Average / Good /**  **Excellent** | **B /B+ / A / A+** |  |

Date of submission:

HOD, Principal,

Prof. Hina Chokshi Dr.Priya Swaminarayan

**INDEX**

| **Content** **Page No.** |
| --- |
| 1. Introduction to Project System 6 |
| 1. System Requirement Specification 7 |
| * 1. Introduction to SRS 7 |
| * 1. Hardware Requirement 7 |
| * 1. Software Requirement 7 |
| * 1. System Users 8 |
| * 1. Description of User Role 8 |
| * 1. System Modules 8 |
| * 1. Description of Modules 8 |
| * 1. Timeline Chart 9 |
| 1. System Flow Diagram 10 |
| 1. Data Flow Diagram (All Levels of DFDs) 11 |
| 1. Use Case Diagram 13 |
| 1. Data Dictionary 14 |
| 1. Screenshots of Development Phase -1 15 |
| 1. Screenshots of Development Phase -2 20 |
| 1. Screenshots of Development Phase -3 24 |
| 1. Conclusion 31 |
| 1. Future Enhancement 32 |
| 1. References 33 |

**ABSTRACT**

Hangman is a classic word-guessing game where the player has to guess a hidden word by guessing the individual letters that make up the word. The player is given a limited number of guesses and must guess the word before they run out of guesses or else they lose the game. The game also includes a hangman figure which is slowly built as the player fails to guess correctly.The Hangman game project is an engaging and interactive game that is designed to help users improve their language skills, vocabulary, and spelling. The game is built using the Python programming language and the Pygame, Tkinter, Pillow and Sqlite3 library, which provide a range of features and tools to create an immersive and visually appealing game experience. In addition to its entertainment value, the Hangman game project can also be used as an educational tool for teachers and educators. However, the proposed system does have some limitations, such as a limited word library and potential compatibility issues with older devices or operating systems. Nonetheless, the game's user-friendly interface, and appealing graphics and sound effects make it a valuable and entertaining addition to the gaming community.

**Chapter 1**

**Introduction to Project System**

Hangman is a popular word guessing game that is often used in educational settings to help students learn new words and vocabulary. The game is simple: players have to guess the word by guessing one letter at a time. If the players guess wrong, the game adds an element of suspense as a “hangman” is slowly drawn and one more wrong guess will result in the player losing the game. Hangman is a great game for learning new words, helping with spelling, and having some fun with friends and family. In this system we have used various python modules for building this game some of the modules are tkinter, pillow, pygame, sqlite3, threading and random. In this system the user primary have to register himself and only after that the user is allowed to login and start playing game. All the data provided by user is stored through sqlite3 and is stored in a local file. To play the user have to user have to click the character displayed in the game and guess the letter and according to it the system will give score. In this system we have also provided negative scoring for wrong guess made. In this system we have also implemented sound for making the user more enjoyable to play the game again and again. The pillow library is used for adding background image to the start page of the game. The words that are used for guessing are stored in a text file and are later used in the game we can later update the words by simply copying the words to the file. The system has been integrated of two windows tkinter and pygame. Through use of tkinter we created login page , registration page and start page. Where as in pygame we simply created the game page only. We have provided different frame sizes to different pages according to requirement. We have provided 500\*550 to login and registration page were as for start page we implemented 605\*305 frame sizes for providing better feel to the users.

**Chapter 2**

**System Requirement Specification**

**2.1 Introduction to SRS**

**2.1.1 What is SRS?**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

**2.1.2 Need of SRS**

In order to fully understand one’s project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on.

**2.2 Hardware Requirement**

| **Hardware Components** | **Specification** |
| --- | --- |
| Processor | Intel core I3,/I5 |
| RAM | 4GB/8GB |
| Hard disk | 512GB/1TB |
| Monitor | 15.6 colour monitor or advance |
| Device | Keyboard, Mouse |

**2.3 Software Requirement**

| **Name of component** | **Specification** |
| --- | --- |
| Operating System | WindowsXP,windows10 |
| Software development kit | Python Version 3 or above, Python Pycharm |
| Programming language | Python Programming [with MySql] |

**2.4 System Users**

**2.4.1 Player/User**

**2.5 Description of User Role**

**2.5.1 Player /User**

Player are the only user of this system .They simply register themselves in system and use the entered username and password to start playing the game. To play they use mouse to select the shown option or letters to guess the words.

**2.6 System Features**

2.6.1 Regestration

2.6.2 Login

2.6.3 Background sound

2.6.4 Lives System

2.6.5 Score System

2.6.6 2D animation

**2.7 Description of Features**

**2.7.1 Registration**

Data of the player or user is saved from registration page to use it as user authentication while logging to the game.

**2.7.2 Login**

Only the authenticate user are allowed to login and play the game.

**2.7.3 Background Sound**

Background sound is implemented in the system for proving the user a thrilling feeling while playing this game and enjoy the game.

**2.7.4 Lives System**

This system or game has lives system but unlike other games this game lives status is shown in the form of images. There are total 5 lives available in this game.

**2.7.5 Score System**

Scores are obtained on the basis of right guess and there is also a negative scoring system provided in it.

**2.7.6 2D Animation**

The game is implemented in a 2d format so that the user won’t be bored as many 1d animation games or console games are very boring for user to play.

**2.8 Timeline Chart**

| **Development phase** | ***75 Days*** | | | | | | **Duration**  **N**  **(days)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0to10 days | 11to20  days | 21to30 days | 31to40 days | 41to50 days | 51to75  days |
| **Requirement**  **Gathering** |  |  |  |  |  |  | 07 |
| **Analysis** |  |  |  |  |  |  | 09 |
| **Design** |  |  |  |  |  |  | 10 |
| **Development Phase 1** |  |  |  |  |  |  | 13 |
| **Development Phase 2** |  |  |  |  |  |  | 13 |
| **Development Phase 3** |  |  |  |  |  |  | 13 |
| **Documentation** |  |  |  |  |  |  | 10 |
| **Total time**  **(Days)** |  | | | | | | **75** |

2.8Time line chart of Hangman Game

**Chapter 3**

**System Flow Diagram**

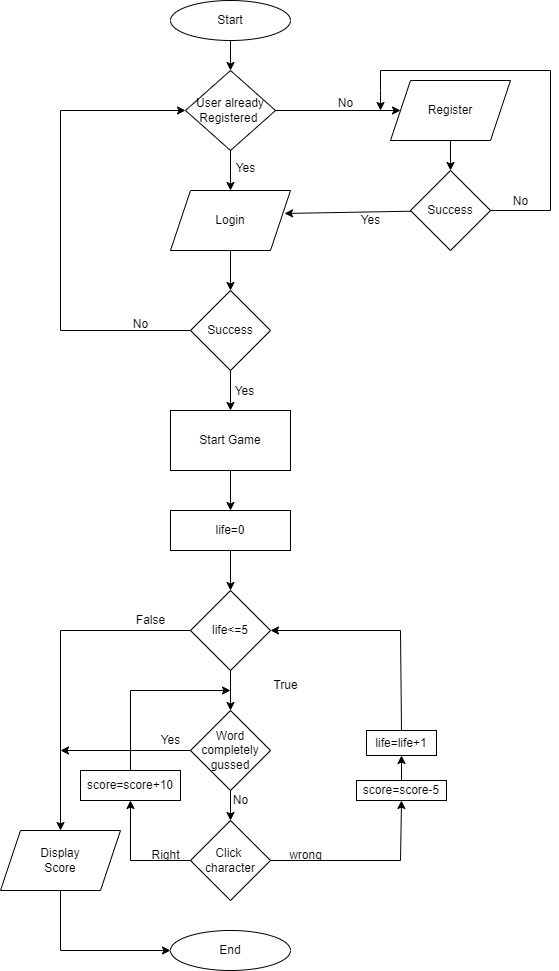
****

Figure 1. System Flow Diagram of Hangman Game

**Chapter 4**

**Data Flow Diagram ( All Levels)**

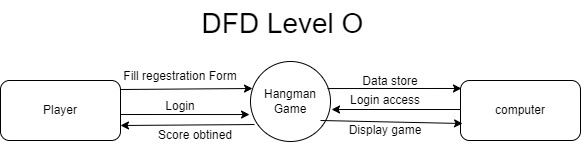
****

Figure 1. Data Flow Diagram level 0 of Hangman Game

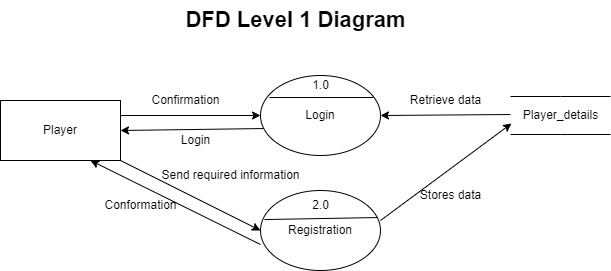
****

Figure 2. Data Flow Diagram level 1 of Hangman Game

**DFD Level 2 Diagram**

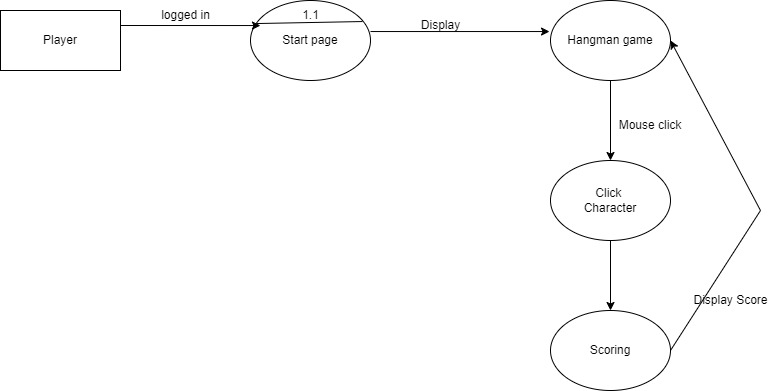
****

Figure 3. Data Flow Diagram level 2 of Hangman Game

****

Figure 4. Data Flow Diagram level 2 of Hangman Game

**Chapter 5**

**Use Case Diagram**

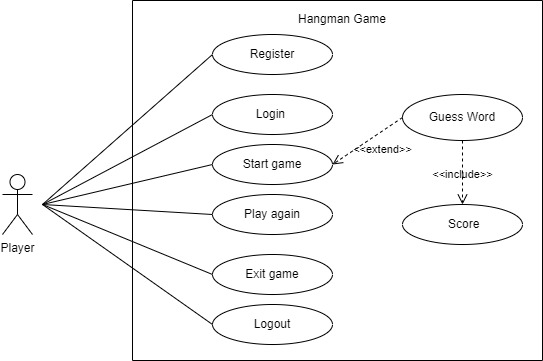


Figure 1. Use Case Diagram of Hangman Game

**Chapter 6**

**Data Dictionary**

**6.1 Player\_details**

Stores personal details of the player/user.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.no** | **Column** | **Data type** | **Size** | **KEY** | **Constrain** | **Description** | **example** |
| 1 | Player\_id | Int | - | PK | Al | Primary key of table | 1 |
| 2 | First\_name | Varchar | 20 |  | Not Null | First name of user | James |
| 3 | Last\_name | Varchar | 20 |  | Not Null | Last name of user | Bond |
| 4 | Contact\_no | Varchar | 13 |  | Not Null | Contact Number with Country Code | +91 9876543210 |
| 5 | Email | Varchar | 30 |  | Not Null | Email of player | James123@gmail.com |
| 6 | User\_name | Varchar | 20 |  | Not Null | User Name | champion |
| 7 | Password | Varchar | 10 |  | Not Null | Password | \*\*\*\*\*\* |

Table 1 : Player\_details

**Chapter 7**

**Screenshot of Development Phase 1**

**7.1 Login Page Design**

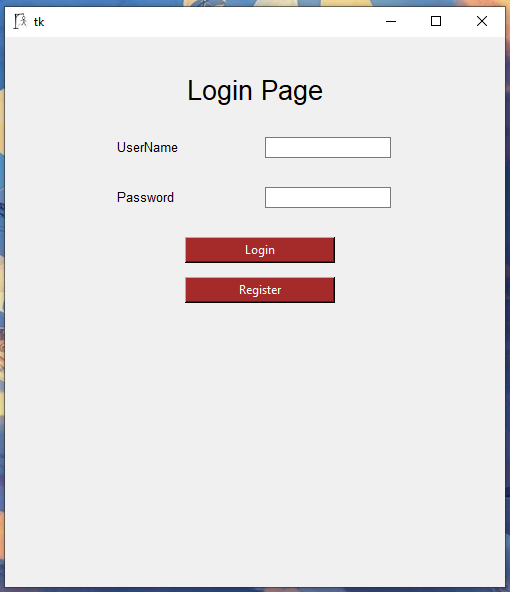
****

Figure 7.1 Login Page Design

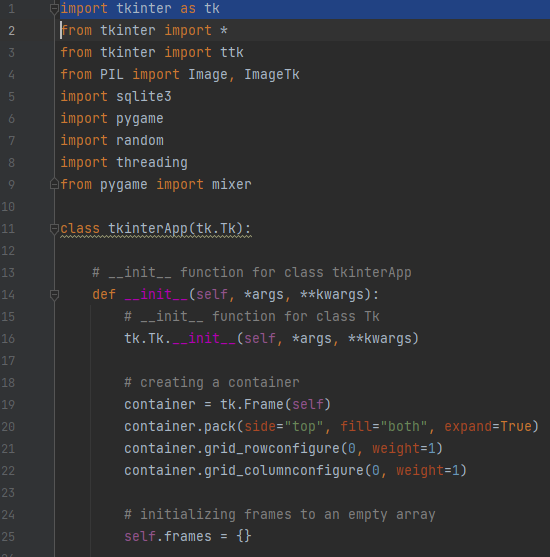
**7.2Code of Login Page **

Figure 7.2.1 Code of Login Page

In this section we imported various modules / libraries for different purpose. The module tkinter are used for GUI (Graphical User Interface), were as PIL /Pillow library is used for inserting images into the frames of software, Sqlite3 is used for storing the information obtained from players/users to further use it for authentication processs. Pygame modules are simply used to create 2d games .random module is used to choose random items. Threading is used for creating some time based intervals takes to be done. To Switch from different frames in tkinter we created a class called tkinterApp and as argument we provided tk.Tk which is used to create custom commands to create and manipulate GUL widgets. After that in tkinterApp class we created an \_\_inti\_\_ function which is used to take values for the user and store it in identifiers. In it we also created containers and provided different parameters for the container. The frames are we created a empty array so that we can iterate the frames one by one.

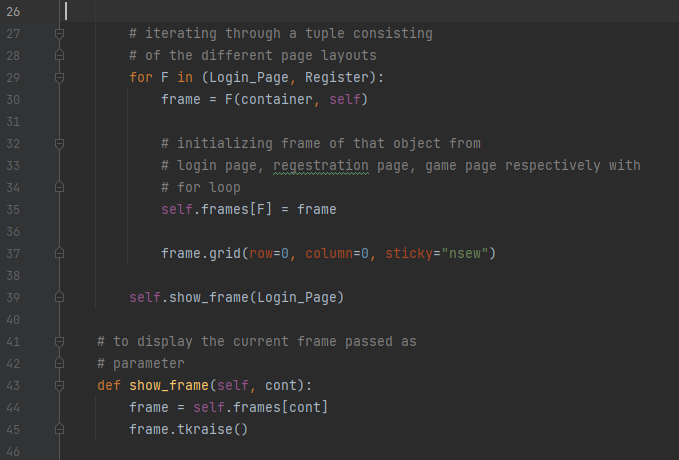


Figure 7.2.2 Code of Login Page

The frames are iterated using for loop and another function called show\_frame was created so that the frames would be raised / displayed according to given commands. The code in line 39 shows that which page to show while first opening the software which is login page.

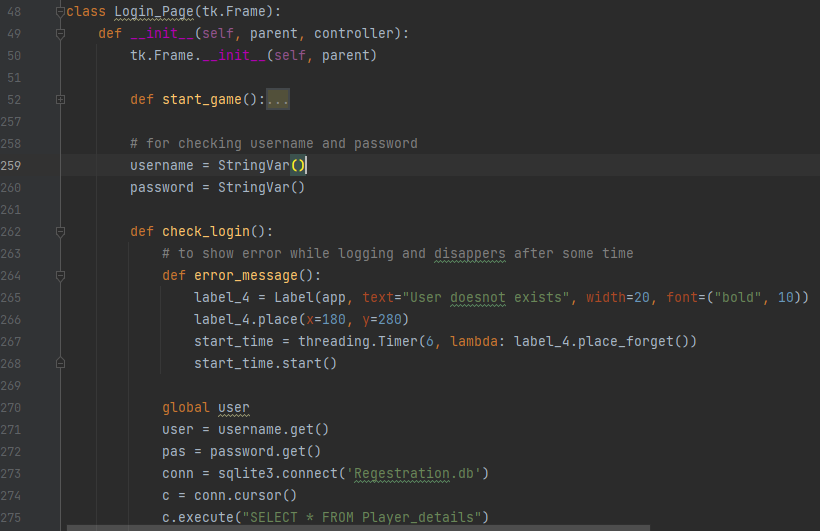


Figure 7.2.3 Code of Login Page

For Login Page we create a class called Login\_Page for frame. In it we created a function to check the logging authentication so for that we first connected to the existing database called Regestration.db and compare it with the existing data from the database for username and password. In the figure we also create function for error message to show error if the user entered wrong credentials and the error message is removed after 6 sec of showing which was implemented with the help of threading library.

****

Figure 7.2.4 Code of Login Page

If the user has entered correct credentials then the program will go to start page.



Figure 7.2.5 Code of Login Page

This image shows the codes used for creating logging page layout.

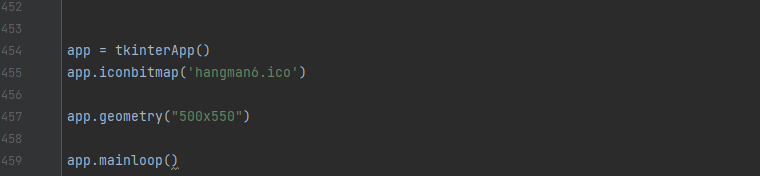


Figure 7.2.6 Code of Login Page

This is the last part for creating a frame and display it simply show that we have implemented icon to the system and it also includes the frame size .

**Chapter 8**

**Screenshot of Development Phase 2**

**8.1 Registration Page Design**

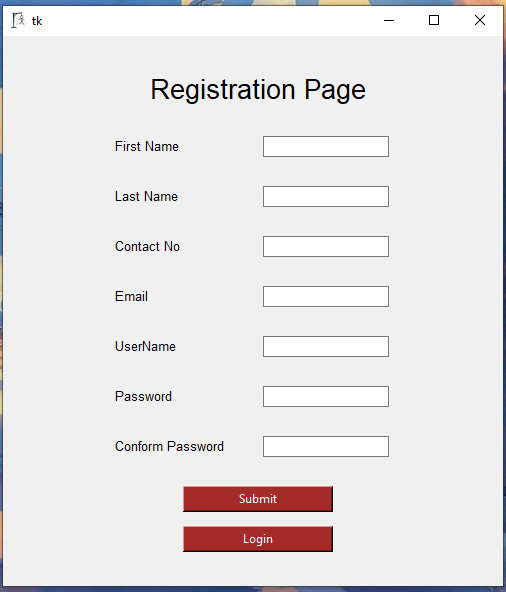
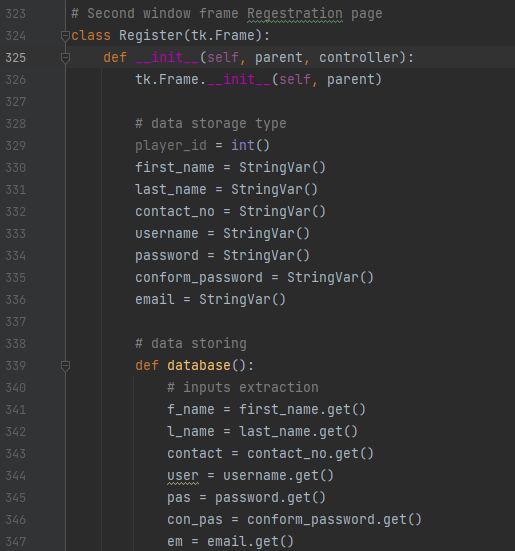
****

Figure 8.1 Registration Page Design

**8.2Code of Registration Page**

****

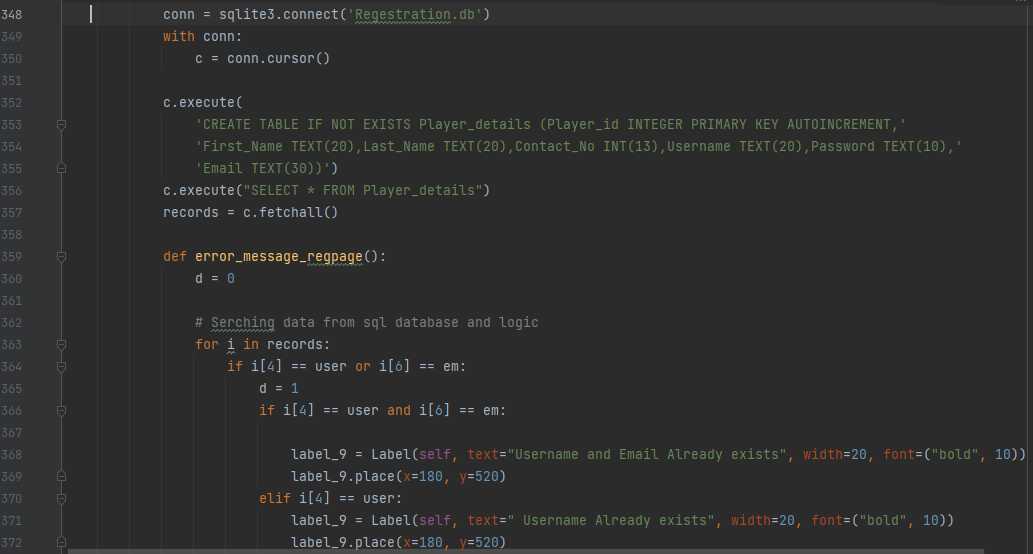
****

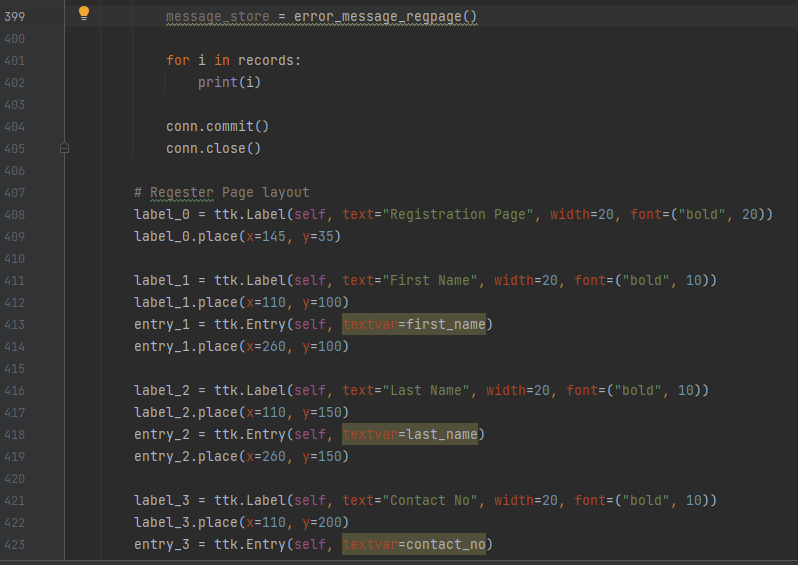
Figure 8.2.1 Code of Registration Page

The above images show that for creating a registration page we created a function called register where we first start with creation of database if doesn’t exist. After that we declared the variable data type which will be further be used for storing it in sql database. To store the data in sql we created another function called database where we declared another variables that will be used to store the information obtained from entry widget. In this function we connected our database called registration.db and started to create table called Player\_details and required table headers if not previously created along with its data type and constrains. If the database and table is already created then if fetches all data for further use. Then we created another function called error\_message\_repage which simply returns error messages case the user didn’t fill a box or the user and email already exist.

****

Figure 8.2.2 Code of Registration Page

If all fields are correctly typed then the data is now stored in table Player\_details and shows message success. The error or success messages are disappears after some interval.

****

****

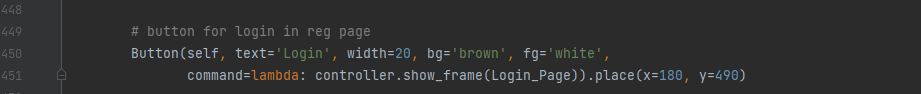
****

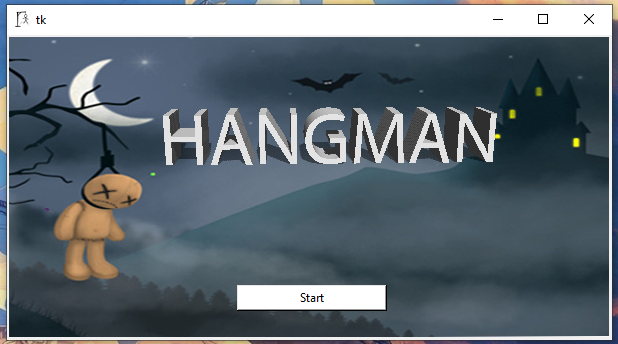
Figure 8.2.3 Code of Registration Page

The database if finally closed at last and after that all the codes are about the layout of registration page.

**Chapter 9**

**Screenshot of Development Phase 3**

**9.1 Game Design**

****

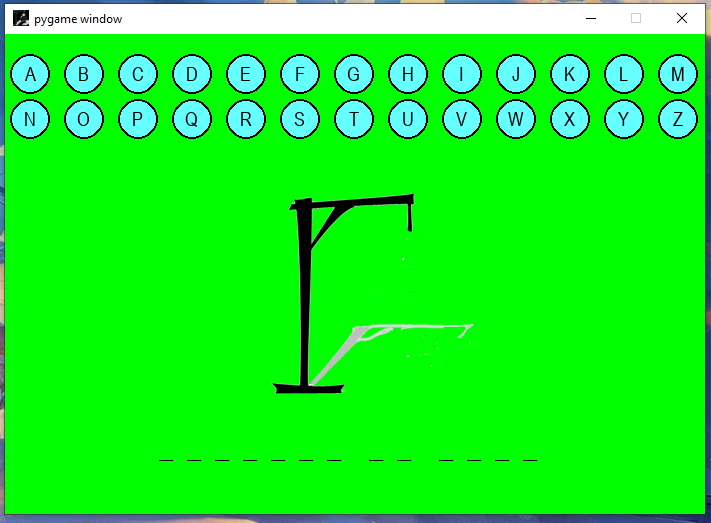
****

Figure 9.1 Game Design

**9.2Code of Game**

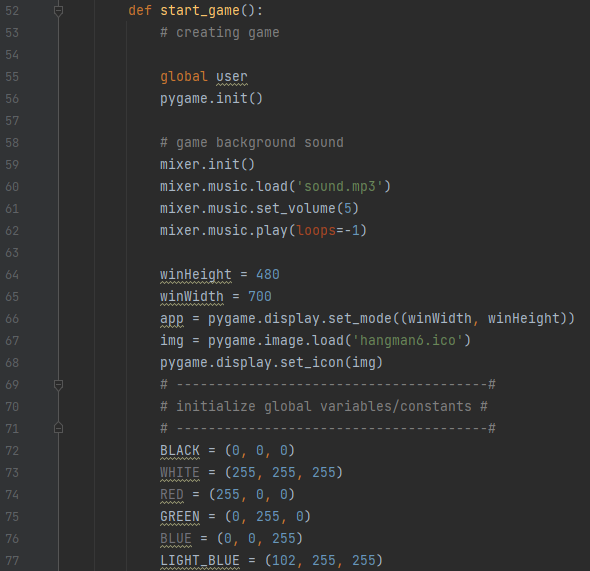
****

Figure 9.2.1 Code of Game

We have created a function called called start\_game so that the game can be called again and again if required. For game windows we have used init function of pygame which simply creates a window for game after that we have implemented background sound for that we used the mixer.music module. And for continuous playing of music we just loop if giving value -1 which does the same work. We then provided the windows display size and icon for the game. After that we provided different color variable to be used in the game.

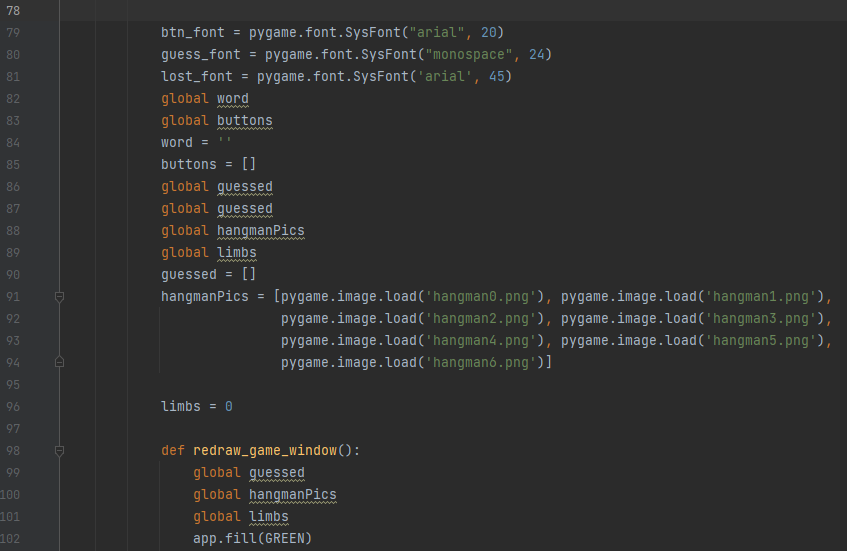
****

Figure 9.2.2 Code of Game

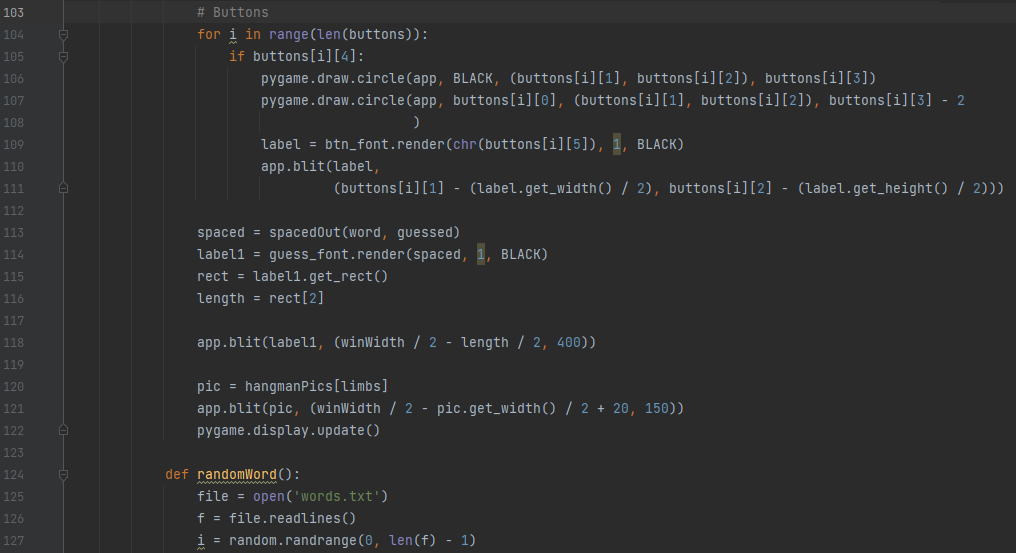
After that we gave the font type to the button and font going to be displayed in the game windows. We stored all the hangman image in an array for iterating it according to requirement in logic section. Then we created a function called redraw\_game\_window so that the window of the game can be updated time to time and set the background color green.****

Figure 9.2.3 Code of Game

After that we started to create the button for the game for that we used for loop for creation of the button. Bellow the button we created a area for the hangman image to be shown. Then for words guessting we created another function called randomWord which consist of file handling operation where we created a dot txt file called word.txt and store the words we need to be used in the game.

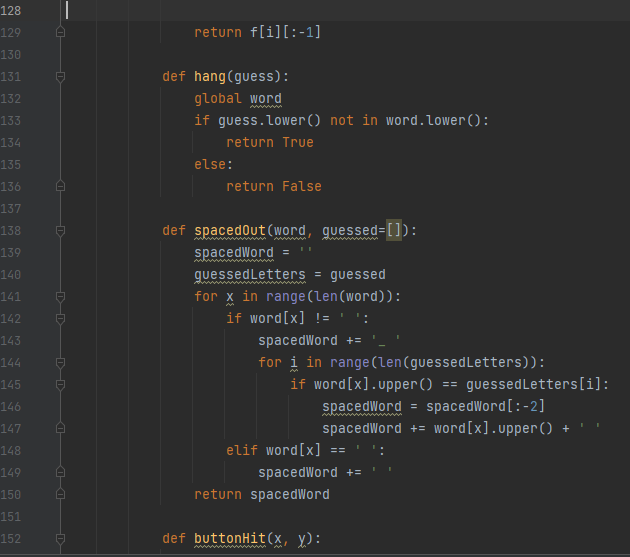
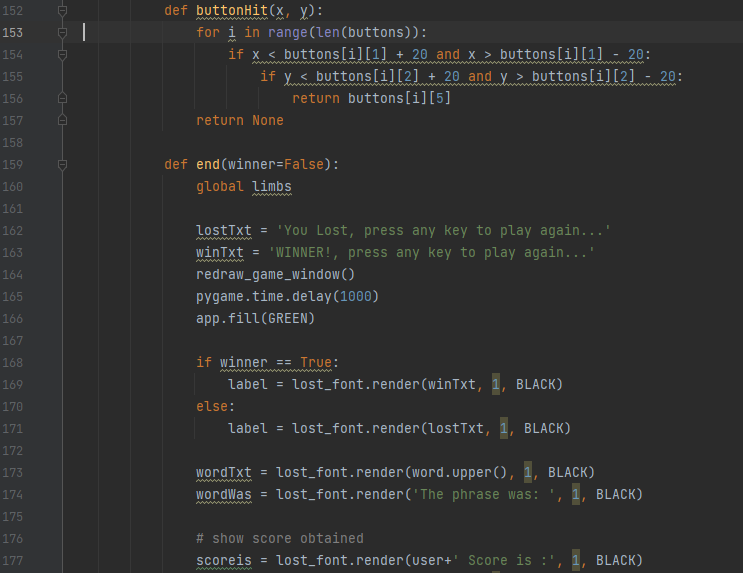
****

Figure 9.2.4 Code of Game

We created a hang function where we gave an argument called guess where we check if the guessed word matches with the work clicked and returns true if right guess. The another function called spacedOut is created for storing the guessed words for that we gave empty array as argument.

****

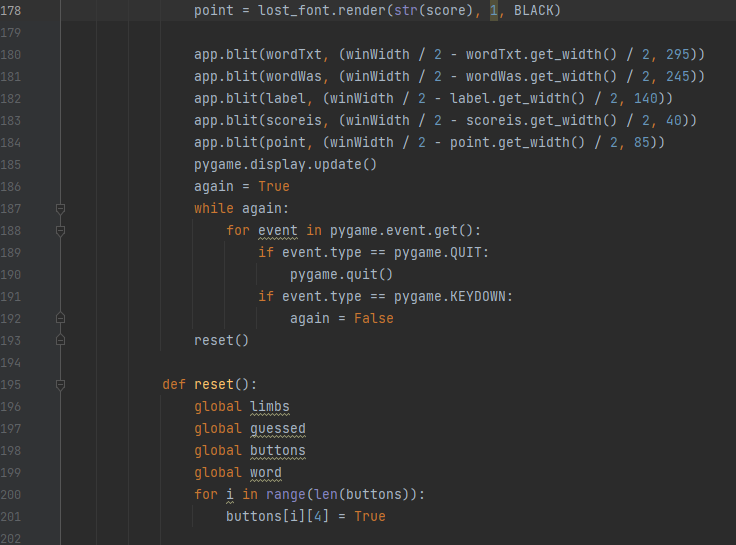
****

Figure 9.2.5 Code of Game

Next function is buttonHit which simply tracks which buttons are clicked and return that button. The function end simply is for showing the Winning or loosing text with the scores obtained along with username used for logging in and to play gain the user need to press any key. The reset function simply empties the array or makes the required variables null.

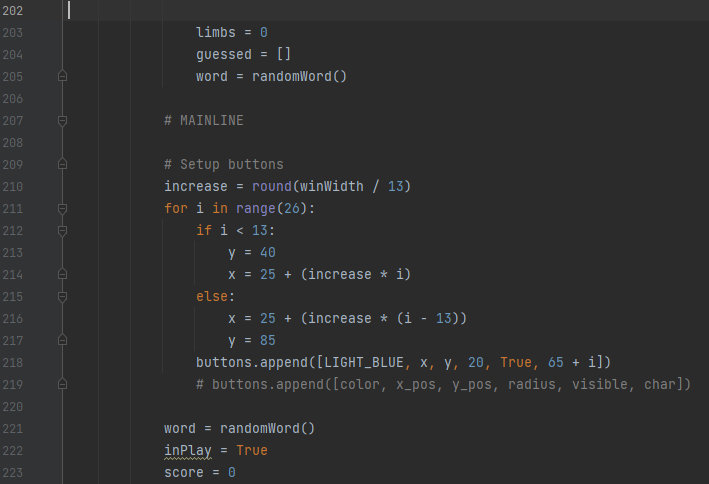


Figure 9.2.6 Code of Game

Not only the variable and array are made null but also resets the buttons that were clicked for guess and after all required objects are reset the game can be played again.

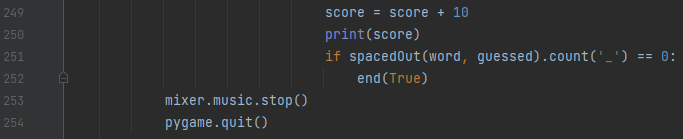
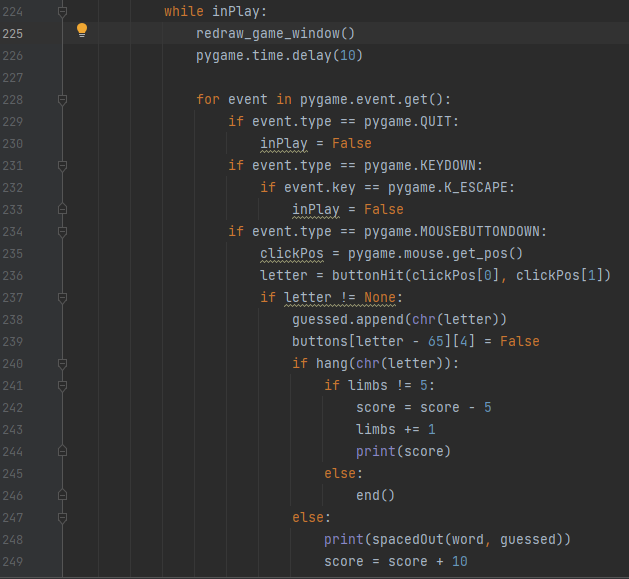
****

Figure 9.2.6 Code of Game

The while loop is implemented until the game is going on all game events are checked one by one using for loop and at last background music is stopped and game quits.

**Chapter 10**

**Conclusion**

The Hangman game project is an engaging and interactive game that is designed to help users improve their language skills, vocabulary, and spelling. The game is built using the Python programming language and the Pygame, Tkinter, Pillow and Sqlite3 library, which provide a range of features and tools to create an immersive and visually appealing game experience. In addition to its entertainment value, the Hangman game project can also be used as an educational tool for teachers and educators. However, the proposed system does have some limitations, such as a limited word library and potential compatibility issues with older devices or operating systems. Nonetheless, the game's user-friendly interface, and appealing graphics and sound effects make it a valuable and entertaining addition to the gaming community. Where as for the implementation of codes and development process there are many areas that this system needs to improve in to be called a quality product.

**Chapter 11**

**Future Enhancement**

1. Multiplayer Feature
2. Changing 2d graphics to 3d
3. Changing it to an online game
4. Leadership Board for global competition

**Chapter 12**

**References**

Website:

1. <https://www.w3schools.com/python/>
2. <https://www.google.com/>
3. <https://docs.python.org/3/library/sqlite3.html>
4. https://www.geeksforgeeks.org/python-sqlite/

Book:

1. “Python GUI Programming with Tkinter”, Alan D.Moore
2. “Morden Tkinter For Busy Python Developers 3rd Ed”, Mark Roseman