Jai Parkash Mukand Lal Innovative Engg. & Technology Institute-JMIETI

Six weeks Industrial Training Report

(5th Semester)

ROCK PAPER SCISSOR GAME SOFTWARE-PROJECT

Submitted for the partial fulfillment of degree

Of

Bachelor of Artificial Intelligence and Machine Learning

Submitted By:- **Submitted To:-**

Abhishek Saini **Mr. Praduman**

DECLARATION

I am Abhishek Saini, students of B.Tech (Artificial Inteligence and Machine Learning) 5th semester,studying at Jai Parkash Mukand Lal Innovative Engg. & Technology Institute,RADAUR(Yamuna Nagar), hereby declare that the six week Industrial training Report,undergone at “Solitaire Infosys”,submitted to Kurukshetra University,Kurukshetra ,inpartial fullfilment of the award of degree of Bachelor of Artificial Inteligence and Machine Learning Engineering is the original work conducted by me.

The information and the data in the report is authentic to the best of my knowledge.

Abhishek Saini

ACKNOWLEDGEMENT

I, student of JMIETI college of engineering and technology, Radaur, have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I am highly indebted to the “Solitaire Infosys” for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project. I would like to express my gratitude towards my teachers for their kind co-operation and encouragement which help me in completion of this project.

Abhishek Saini

Certificate

- Table of Contents

Technology Used

- **Python**
- **Tkinter**

Python

- **Intro**
- **Why to learn Python**
- **Characteristics Of Python**
- **Applications Of Python**

Tkinter

- **Intro**
- **Why to learn Tkinter**
- **Characteristics of Tkinter**
- **Applications of Tkinter**

Replit

Introduction

Conculusion

Bibliography

- # INTRODUCTION

What Is Rock Paper Scissor Game Software?

What is Rock Paper Scissor Game Software? Rock Paper Scissor Game Software is an gaming application that can be utilized to play with computer.

The code will ask the user for a choice between rock, paper and scissors. Once the user enters their choice, the code will randomly choose one of those options as the computer's turn. The code then prints out the chosen option and the user's choice. Finally, it loops back to ask for another choice from the user.

- Technologies Used

1. Python
2. Tkinter

- Code Editor
- Replit

1. PYTHON

What is Python

Python is a dynamic, high-level, free open source, and interpreted programming language. It supports object-oriented programming as well as procedural-oriented programming. In Python, we don't need to declare the type of variable because it is a dynamically typed language.

For example, x = 10 Here, x can be anything such as String, int, etc.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics developed by Guido van Rossum. It was originally released in 1991.

- Easy To Learn and Readable Language.
- Interpreted Language.
- Dynamically Typed Language.
- Open Source And Free.
- Large Standard Library.
- High-Level Language.
- Object Oriented Programming Language.
- Large Community Support.

• Features of Python

There are many features in Python, some of which are discussed below as follows:

1. Free and Open Source

Python language is freely available at the official website and you can download it from the given download link below click on the **Download Python** keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to code

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Easy to Read

As you will see, learning Python is quite simple. As was already established, Python's syntax is really straightforward. The code block is defined by the indentations rather than by semicolons or brackets.

4. Object-Oriented Language

One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support

Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.

6. High-Level Language

Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. Large Community Support

Python has gained popularity over the years. Our questions are constantly answered by the enormous StackOverflow community. These websites have already provided answers to many questions about Python, so Python users can consult them as needed.

8. Easy to Debug

Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.

9. Python is a Portable language

Python language is also a portable language. For example, if we have Python code for Windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.

10. Python is an Integrated language

Python is also an Integrated language because we can easily integrate Python with other languages like C, C++, etc.

11. Interpreted Language:

Python is an Interpreted Language because Python code is executed line by line at a time. like other languages C, C++, Java, etc. there is no need to compile Python code this makes it easier to debug our code. The source code of Python is converted into an immediate form called **bytecode**.

12. Large Standard Library

Python has a large standard library that provides a rich set of modules and functions so you do not have to write your own code for every single thing. There are many libraries present in Python such as regular expressions, unit-testing, web browsers, etc.

13. Dynamically Typed Language

Python is a dynamically-typed language. That means the type (for example- int, double, long, etc.) for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.

14. Frontend and backend development

With a new project py script, you can run and write Python codes in HTML with the help of some simple tags <py-script>, <py-env>, etc. This will help you do frontend development work in Python like javascript. Backend is the strong forte of Python it's extensively used for this work cause of its frameworks like Django and Flask.

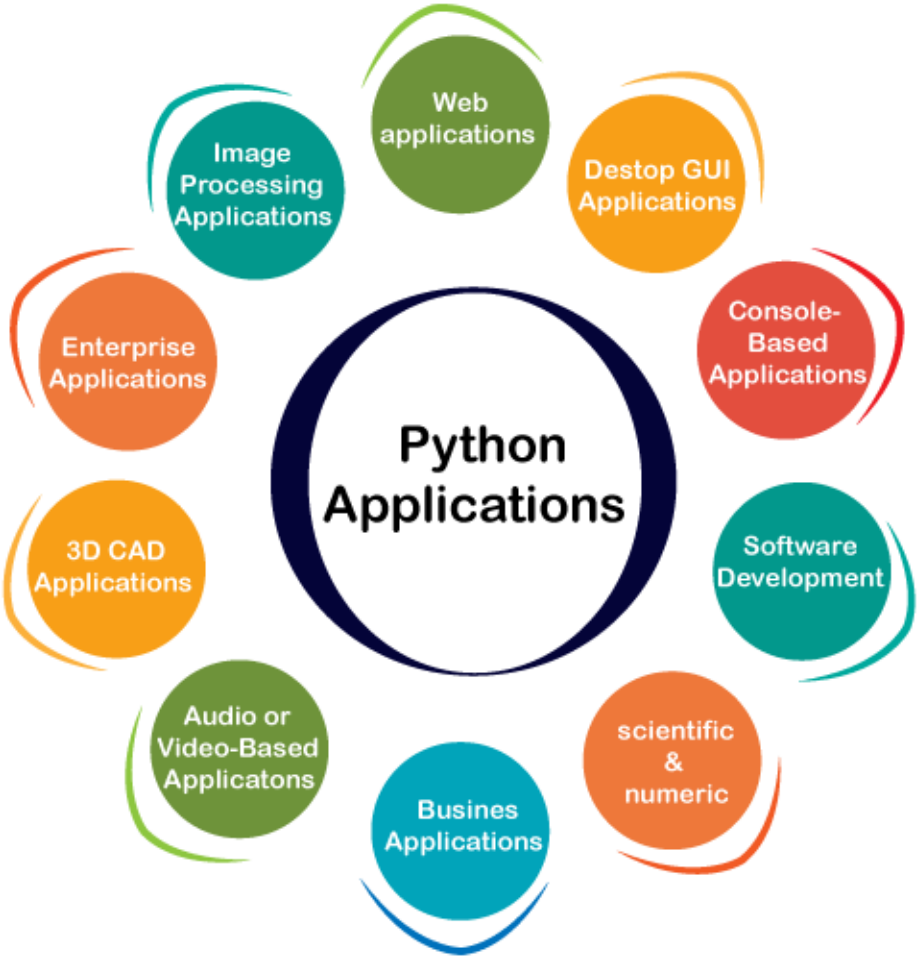
15. Allocating Memory Dynamically

In Python, the variable data type does not need to be specified. The memory is automatically allocated to a variable at runtime when it is given a value. Developers do not need to write int y = 18 if the integer value 15 is set to y. You may just type y=18.

• Applications of Python

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application.

Here, we are specifying application areas where Python can be applied.



1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser, etc. One of Python web-framework named Django is used on **Instagram**. Python provides many useful frameworks, and these are given below:

- Django and Pyramid framework(Use for heavy applications)
- Flask and Bottle (Micro-framework)

- Plone and Django CMS (Advance Content management)

2) Desktop GUI Applications

The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a **Tk GUI library** to develop a user interface. Some popular GUI libraries are given below.

- Tkinter or Tk
- wxWidgetM
- Kivy (used for writing multitouch applications)
- PyQt or Pyside

3) Console-based Application

Console-based applications run from the command-line or shell. These applications are computer program which are used commands to execute. This kind of application was more popular in the old generation of computers. Python can develop this kind of application very effectively. It is famous for having REPL, which means **the Read-Eval-Print Loop** that makes it the most suitable language for the command-line applications.

Python provides many free library or module which helps to build the command-line apps. The necessary **IO** libraries are used to read and write. It helps to parse argument and create console help text out-of-the-box. There are also advance libraries that can develop independent console apps.

4) Software Development

Python is useful for the software development process. It works as a support language and can be used to build control and management, testing, etc.

- **SCons** is used to build control.
- **Buildbot** and **Apache** Gumps are used for automated continuous compilation and testing.
- **Round** or **Trac** for bug tracking and project management.

5) Scientific and Numeric

This is the era of Artificial intelligence where the machine can perform the task the same as the human. Python language is the most suitable language for Artificial intelligence or machine learning. It consists of many scientific and mathematical libraries, which makes easy to solve complex calculations.

Implementing machine learning algorithms require complex mathematical calculation. Python has many libraries for scientific and numeric such as Numpy, Pandas, Scipy, Scikit-learn, etc. If you have some basic knowledge of Python, you need to import libraries on the top of the code. Few popular frameworks of machine libraries are given below.

- SciPy
- Scikit-learn
- NumPy
- Pandas
- Matplotlib

6) Business Applications

Business Applications differ from standard applications. E-commerce and ERP are an example of a business application. This kind of application requires extensively, scalability and readability, and Python provides all these features.

Oddo is an example of the all-in-one Python-based application which offers a range of business applications. Python provides a **Tryton** platform which is used to develop the business application.

7) Audio or Video-based Applications

Python is flexible to perform multiple tasks and can be used to create multimedia applications. Some multimedia applications which are made by using Python are **TimPlayer, cplay,** etc. The few multimedia libraries are given below.

- Gstreamer
- Pyglet
- QT Phonon

8) 3D CAD Applications

The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application by using the following functionalities.

- Fandango (Popular)
- CAMVOX
- HeeksCNC
- AnyCAD
- RCAM

9) Enterprise Applications

Python can be used to create applications that can be used within an Enterprise or an Organization. Some real-time applications are OpenERP, Tryton, Picalo, etc.

10) Image Processing Application

Python contains many libraries that are used to work with the image. The image can be manipulated according to our requirements. Some libraries of image processing are given below.

- OpenCV
- Pillow

SimpleITK

In this topic, we have described all types of applications where Python plays an essential role in the development of these applications. In the next tutorial, we will learn more concepts about Python.

Tkinter

What is Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

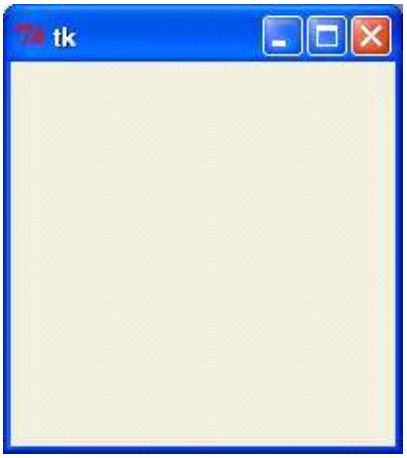
Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Example

```
#!/usr/bin/python

import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

This would create a following window −



Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table −

Sr.No.	Operator & Description
1	Button The Button widget is used to display buttons in your application.

2	Canvas The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	Checkbutton The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	Entry The Entry widget is used to display a single-line text field for accepting values from a user.
5	Frame The Frame widget is used as a container widget to organize other widgets.
6	Label The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	Listbox The Listbox widget is used to provide a list of options to a user.
8	Menubutton The Menubutton widget is used to display menus in your application.
9	Menu The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	Message The Message widget is used to display multiline text fields for accepting values from a user.
11	Radiobutton The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	Scale The Scale widget is used to provide a slider widget.
13	Scrollbar The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	Text The Text widget is used to display text in multiple lines.

15	Toplevel The Toplevel widget is used to provide a separate window container.
16	Spinbox The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	PanedWindow A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	LabelFrame A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	tkMessageBox This module is used to display message boxes in your applications.

Let us study these widgets in detail −

Standard attributes

Let us take a look at how some of their common attributes.such as sizes, colors and fonts are specified.

- Dimensions
- Colors
- Fonts
- Anchors
- Relief styles
- Bitmaps
- Cursors

Let us study them briefly −

Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

- The *pack()* Method − This geometry manager organizes widgets in blocks before placing them in the parent widget.
- The *grid()* Method − This geometry manager organizes widgets in a table-like structure in the parent widget.
- The *place()* Method − This geometry manager organizes widgets by placing them in a specific position in the parent widget.

Replit

Replit combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

Available for macOS, Linux, and Windows

Replit supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.

Edit, build, and debug with ease

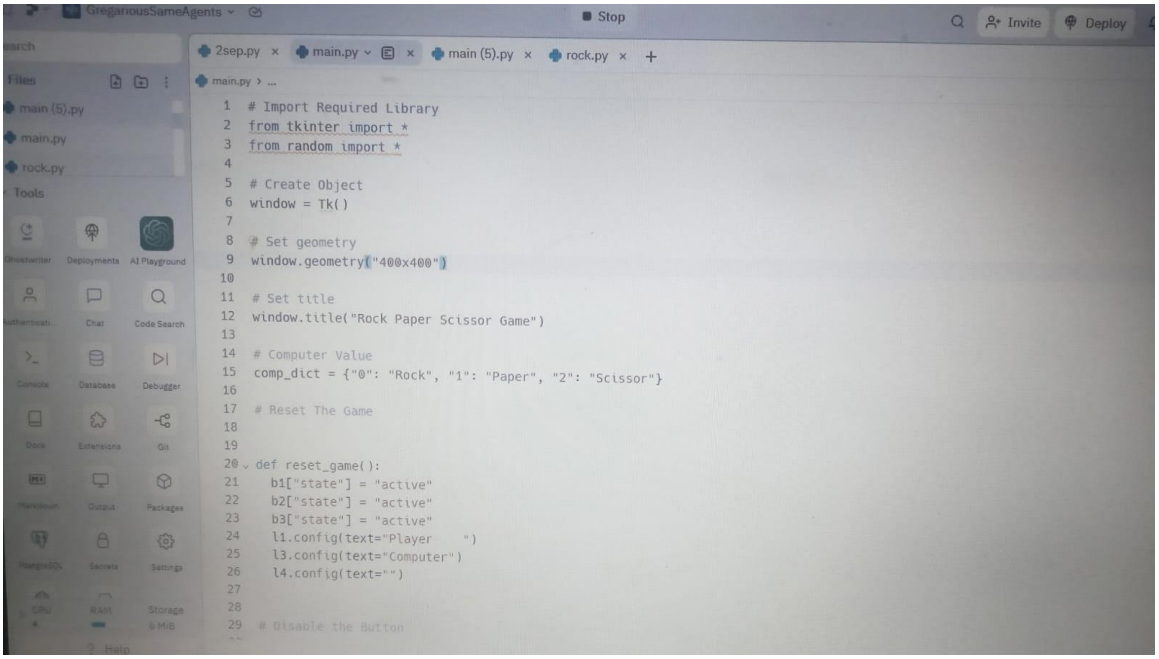
At its heart, Replit features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, Replit helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Replit includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

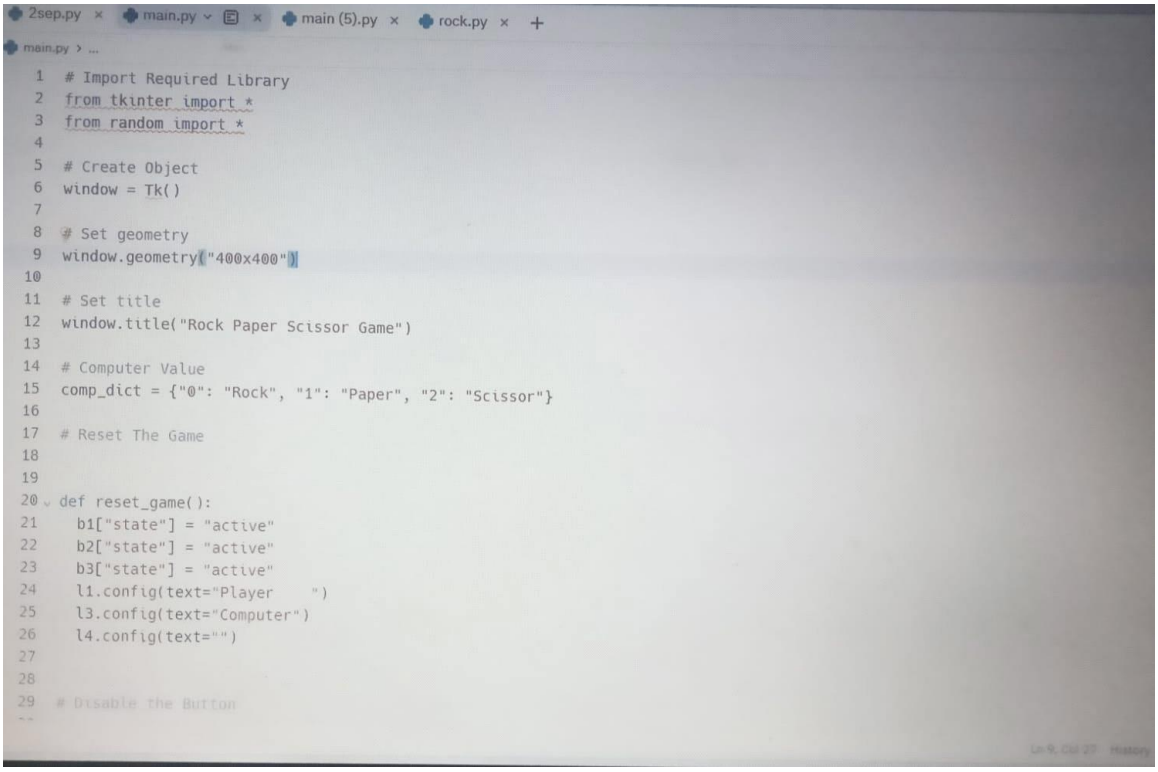
And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Replit includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

Replit also integrates with build and scripting tools to perform common tasks making everyday workflows faster. Replit has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

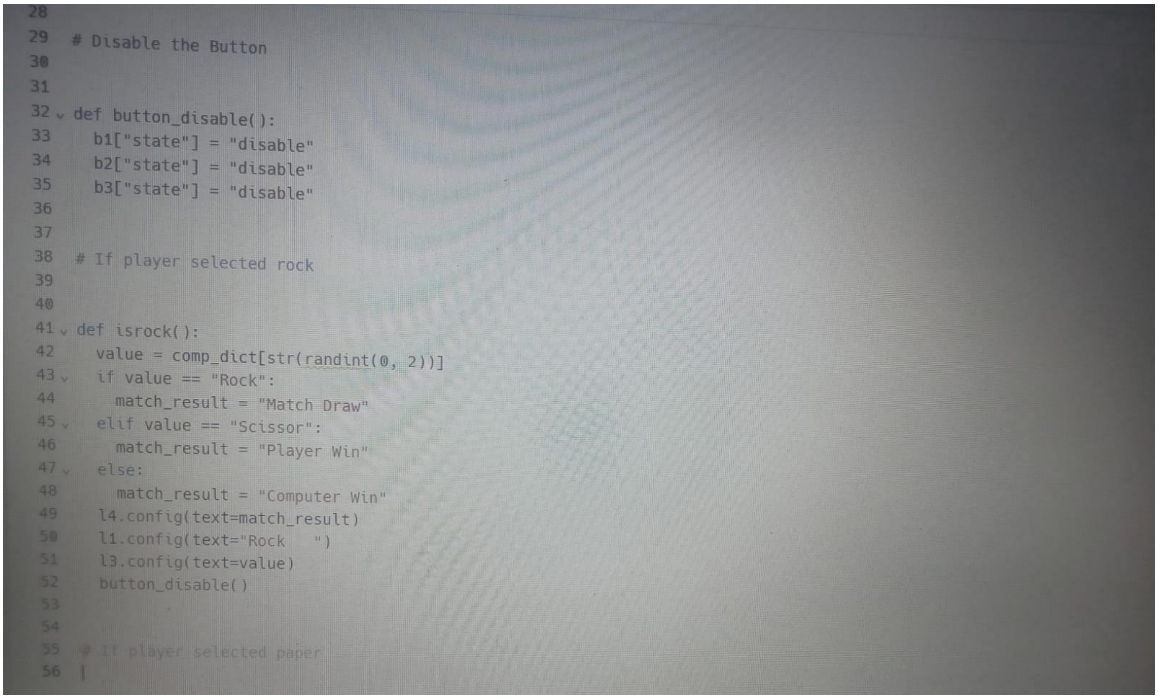
Replit Interface



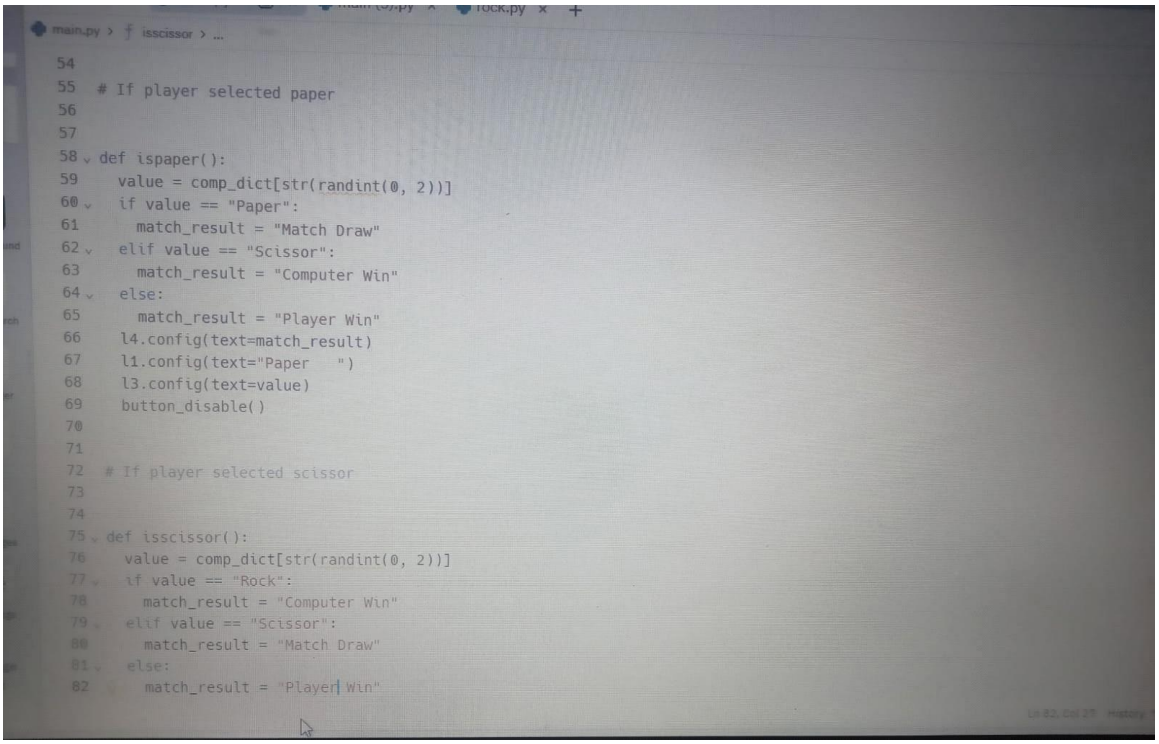
Source Code Of Project



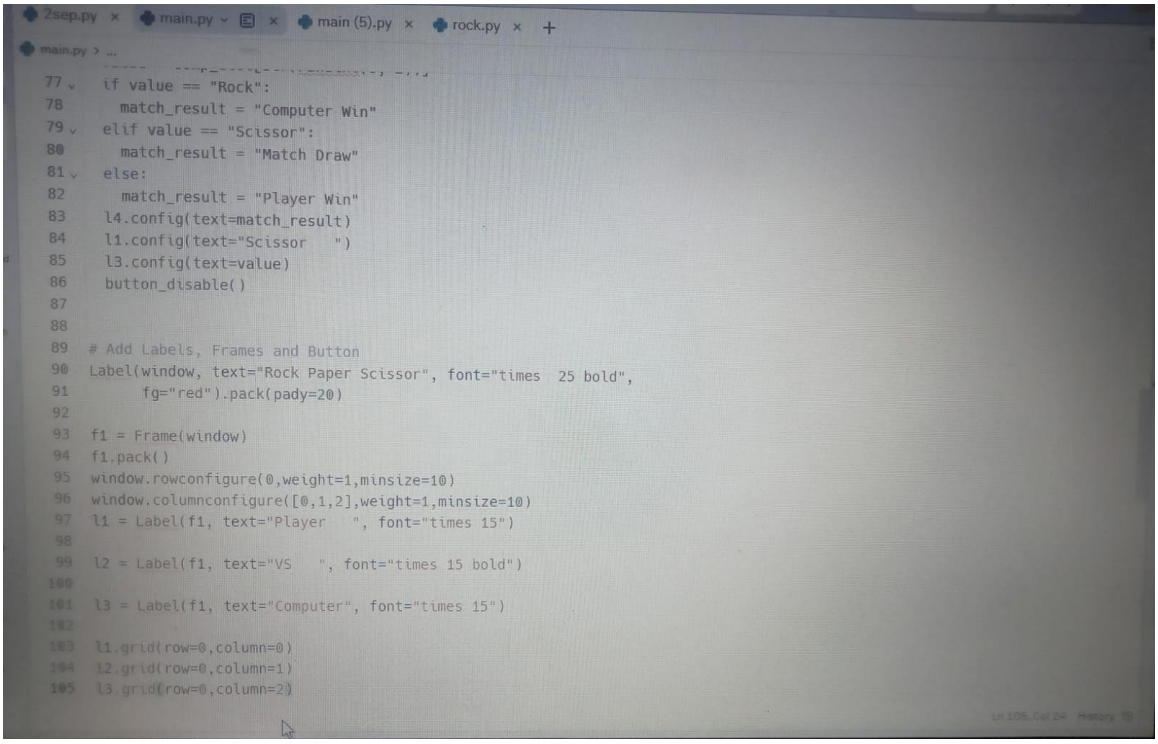
```
# Import Required Library
from tkinter import *
from random import *

# Create Object
window = Tk()

# Set geometry
window.geometry("400x400")

# Set title
window.title("Rock Paper Scissor Game")

# Computer Value
comp_dict = {"0": "Rock", "1": "Paper", "2": "Scissor"}

# Reset The Game


def reset_game():
  b1["state"] = "active"
  b2["state"] = "active"
  b3["state"] = "active"
  l1.config(text="Player    ")
  l3.config(text="Computer")
  l4.config(text="")


# Disable the Button
```



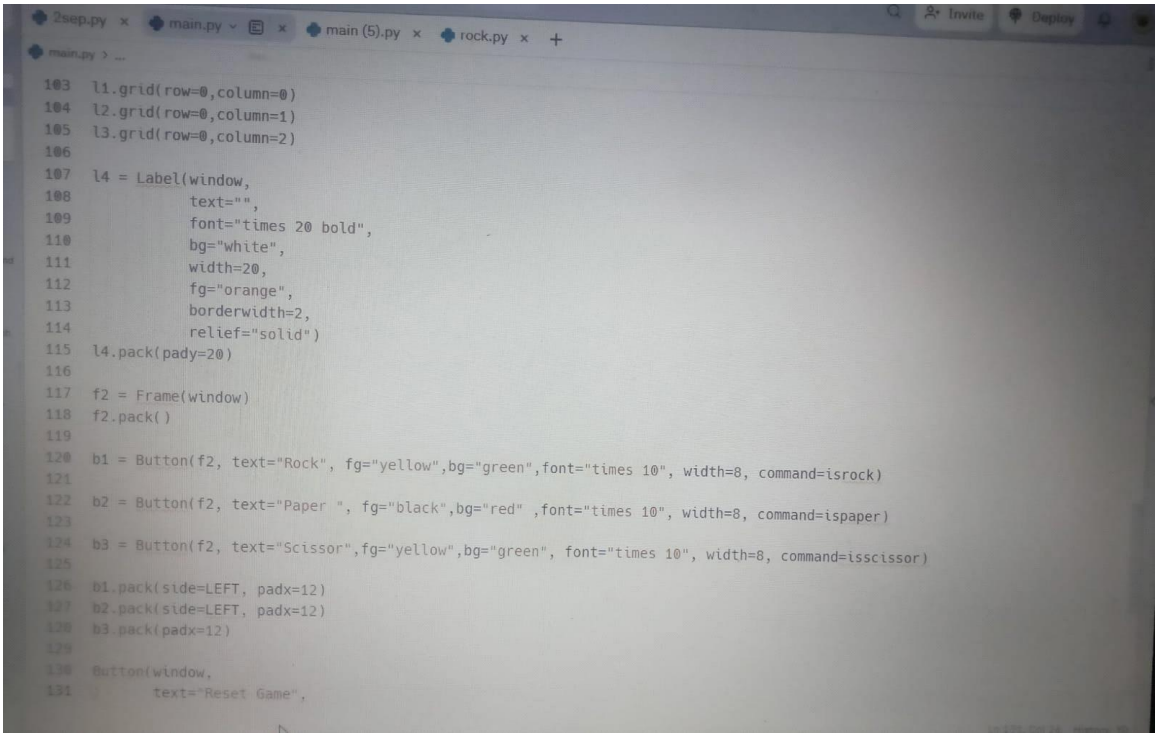
```

# Disable the Button


def button_disable():
  b1["state"] = "disable"
  b2["state"] = "disable"
  b3["state"] = "disable"


# If player selected rock


def isrock():
  value = comp_dict[str(randint(0, 2))]
  if value == "Rock":
    match_result = "Match Draw"
  elif value == "Scissor":
    match_result = "Player Win"
  else:
    match_result = "Computer Win"
  l4.config(text=match_result)
  l1.config(text="Rock   ")
  l3.config(text=value)
  button_disable()


# If player selected paper
```



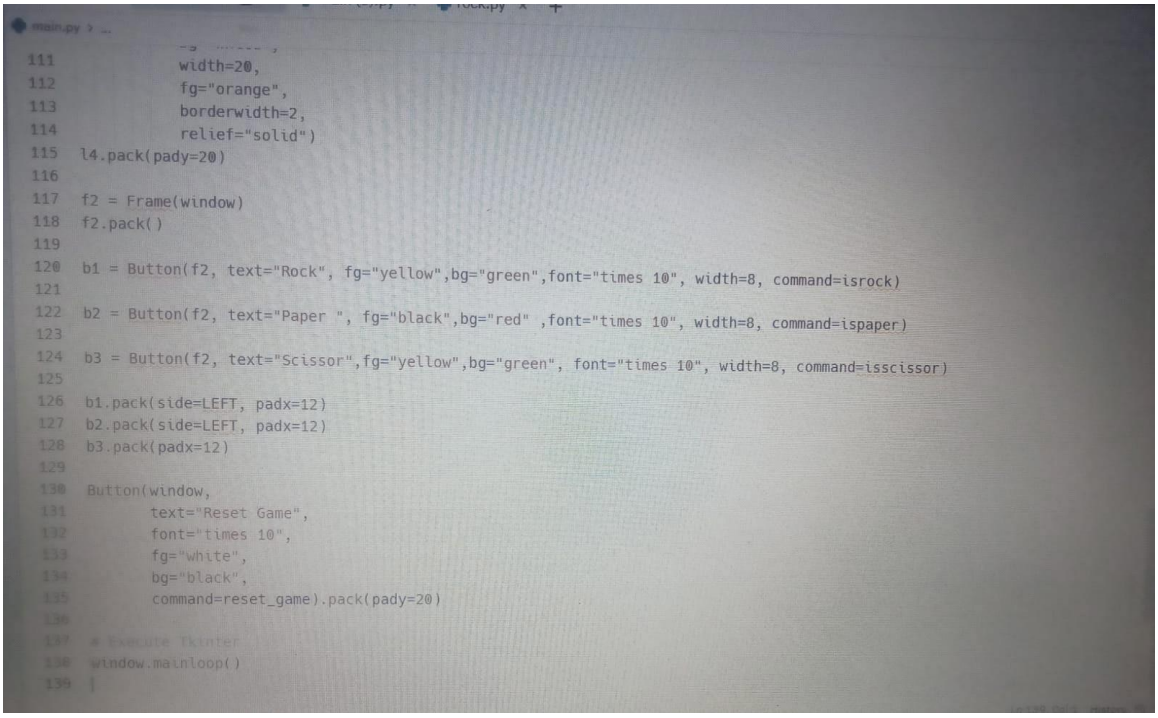
```
main.py > isscissor > ...

54
55  # If player selected paper
56
57
58  def ispaper():
59      value = comp_dict[str(randint(0, 2))]
60      if value == "Paper":
61          match_result = "Match Draw"
62      elif value == "Scissor":
63          match_result = "Computer Win"
64      else:
65          match_result = "Player Win"
66      l4.config(text=match_result)
67      l1.config(text="Paper   ")
68      l3.config(text=value)
69      button_disable()
70
71
72  # If player selected scissor
73
74
75  def isscissor():
76      value = comp_dict[str(randint(0, 2))]
77      if value == "Rock":
78          match_result = "Computer Win"
79      elif value == "Scissor":
80          match_result = "Match Draw"
81      else:
82          match_result = "Player Win"
```



```
    if value == "Rock":
      match_result = "Computer Win"
    elif value == "Scissor":
      match_result = "Match Draw"
    else:
      match_result = "Player Win"
    l4.config(text=match_result)
    l1.config(text="Scissor   ")
    l3.config(text=value)
    button_disable()


# Add Labels, Frames and Button
Label(window, text="Rock Paper Scissor", font="times  25 bold",
      fg="red").pack(pady=20)

f1 = Frame(window)
f1.pack()
window.rowconfigure(0,weight=1,minsize=10)
window.columnconfigure([0,1,2],weight=1,minsize=10)
l1 = Label(f1, text="Player   ", font="times 15")

l2 = Label(f1, text="VS   ", font="times 15 bold")

l3 = Label(f1, text="Computer", font="times 15")

l1.grid(row=0,column=0)
l2.grid(row=0,column=1)
l3.grid(row=0,column=2)
```

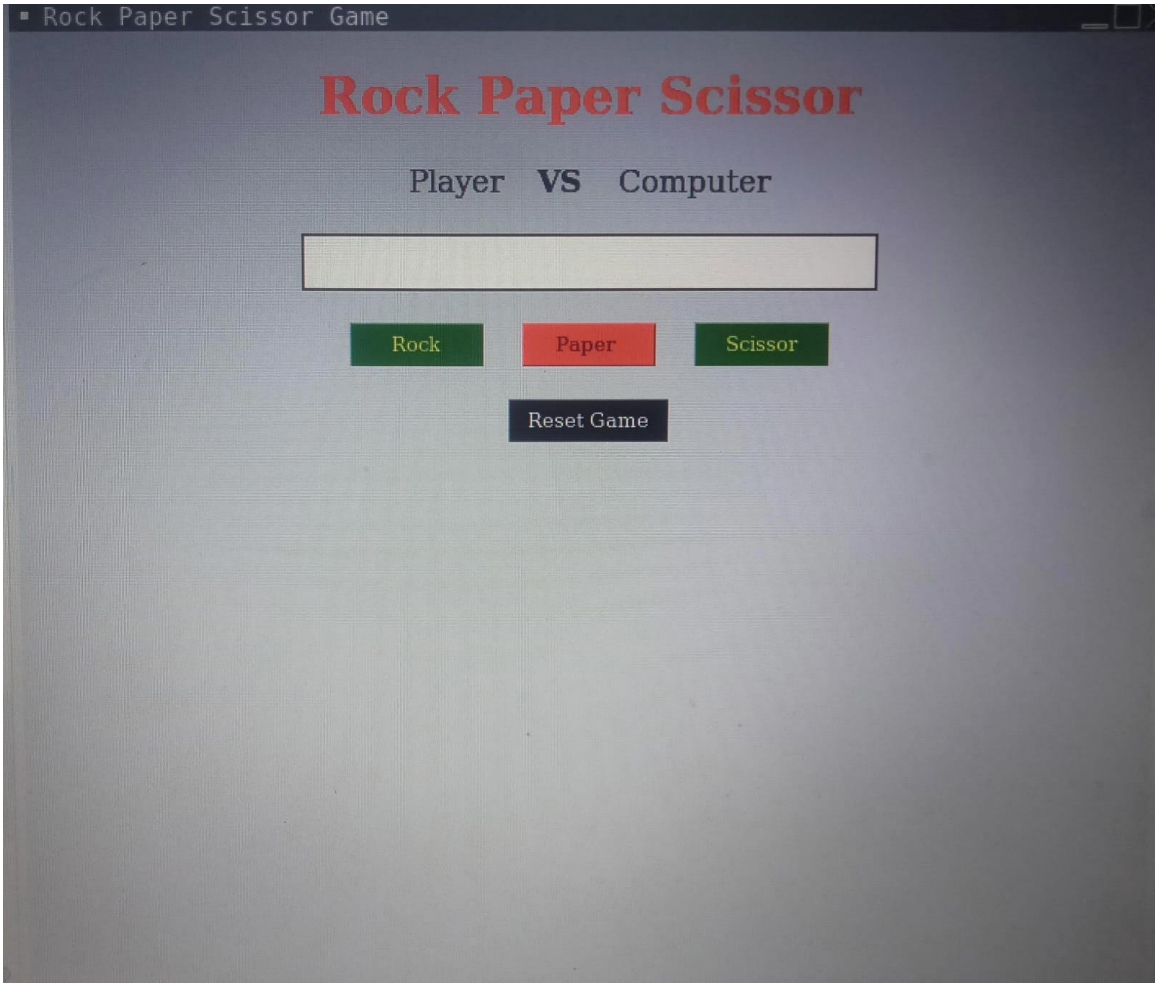


```
l1.grid(row=0,column=0)
l2.grid(row=0,column=1)
l3.grid(row=0,column=2)

l4 = Label(window,
           text="",
           font="times 20 bold",
           bg="white",
           width=20,
           fg="orange",
           borderwidth=2,
           relief="solid")
l4.pack(pady=20)

f2 = Frame(window)
f2.pack()

b1 = Button(f2, text="Rock", fg="yellow",bg="green",font="times 10", width=8, command=isrock)

b2 = Button(f2, text="Paper ", fg="black",bg="red" ,font="times 10", width=8, command=ispaper)

b3 = Button(f2, text="Scissor",fg="yellow",bg="green", font="times 10", width=8, command=isscissor)

b1.pack(side=LEFT, padx=12)
b2.pack(side=LEFT, padx=12)
b3.pack(padx=12)

Button(window,
       text="Reset Game",
```



```
            width=20,
            fg="orange",
            borderwidth=2,
            relief="solid")
l4.pack(pady=20)

f2 = Frame(window)
f2.pack()

b1 = Button(f2, text="Rock", fg="yellow",bg="green",font="times 10", width=8, command=isrock)

b2 = Button(f2, text="Paper ", fg="black",bg="red" ,font="times 10", width=8, command=ispaper)

b3 = Button(f2, text="Scissor",fg="yellow",bg="green", font="times 10", width=8, command=isscissor)

b1.pack(side=LEFT, padx=12)
b2.pack(side=LEFT, padx=12)
b3.pack(padx=12)

Button(window,
       text="Reset Game",
       font="times 10",
       fg="white",
       bg="black",
       command=reset_game).pack(pady=20)

window.mainloop()
```

PROJECT INTERFACE



CONCLUSION

Rock Paper Scissor game is a Python based Application or Software by the help of which we easly play with the computer ,by using this software we can make the game for future need.

Rock, Paper, Scissors. The familiar game of Rock, Paper, Scissors is played like this: at the same time, two players display one of three symbols: a rock, paper, or scissors. A rock beats scissors,

scissors beat paper by cutting it, and paper beats rock by covering it.

The rules state that rock smashes scissors, scissors cuts paper, and paper covers rock. So, rock wins over scissors, scissors win over paper and paper wins over rock.

Bibliography

Geeksforgeeks.com

Python.com

Replit.com

Tutorialpoint.com

Javatpoint.com