Project Name: Game Application: Rock, Paper, and Scissor

Submitted To: Gowthami

Submitted By: ARULSELVI A

Coordinator Name: Pavimithra

D.O.B: 28-11-2025

GitHub Link:

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to everyone who supported me throughout the completion of my mini project titled **"Game Application: Rock, Paper, and Scissor."** This project helped me improve my knowledge of Python programming and gave me valuable practical experience.

First, I would like to thank **my project guide, Ms. Gowthami**, for her constant guidance, encouragement, and helpful feedback. Her support played an important role in shaping this project and helping me understand the concepts clearly.

I also extend my sincere thanks to **our Coordinator, Ms. Pavimithra**, for providing me the opportunity to work on this project and for motivating me throughout the process.

My heartfelt thanks go to all my **faculty members** of the Computer Science and Engineering department for teaching me the fundamentals of programming and helping me build the skills required to complete this project successfully.

I would also like to thank my **friends and classmates** who supported me with ideas, suggestions, and feedback that helped me improve my work.

Finally, I express my deep gratitude to my **parents and family** for their continuous encouragement, support, and motivation, which helped me stay focused and complete this project on time.

I am thankful to everyone who contributed directly or indirectly to the successful completion of this mini project.

Thank you.

1.INTRODUCTION TO PYTHON

Python is a high-level, general-purpose programming language created by Guido van Rossum in 1991. It is widely known for its simple syntax, readability, and extensive library support. easy-to-learn programming language widely used for developing applications, automation, and simple games like Rock, Paper, and Scissor. It provides a clean syntax and built-in libraries such as the random module, which helps generate computer choices in this game. The program uses basic Python concepts including functions to organize tasks, input and output statements to interact with the user, conditional statements to compare choices and decide the winner, and loops to repeat the game until the user wants to stop. Overall, Python's simplicity and powerful features make it an ideal language for building interactive applications like this mini project.

Why Python?

- Easy to learn
- Portable and platform independent
- High community support
- Supports both object-oriented & procedural programming

Features of Python

- Interpreted Language
- Dynamic Typing
- Huge Built-in Libraries
- Beginner Friendly Syntax

Applications of Python

- Web Development
- Machine Learning

- Data Science
- Game Development
- Automation
- Artificial Intelligence

Python is a perfect choice for building simple applications like Rock, Paper, Scissor Game.

2. INTRODUCTION ABOUT PROJECT

This mini project **"Rock, Paper, and Scissor Game Application"** is built using Python. It is an interactive game where the user plays against the computer.

Project Purpose

- To demonstrate Python fundamentals
- To implement user input and system-generated output
- To use the random module for computer choice
- To develop a simple command-line game

How the Game Works

1. User chooses Rock, Paper, or Scissor
2. Computer randomly selects a choice
3. The program compares both choices
4. Winner is declared based on rules
5. User can continue or exit

This project improves logical thinking, function usage, and Python programming skills.

3.CONCEPTS USED IN PROJECT

1. Variables

Used to store user input and computer choices.

2. Input and Output

- input() for reading user choice
- print() for showing results

3. Random Module

import random
random.choice()

Used to generate computer choice randomly.

4. Functions

Three functions are used:

- get_user_choice()
- get_computer_choice()
- determine_winner()

5. Conditional Statements

if–elif–else used for determining winner or tie.

6. Loops

while True: used for repeating gameplay until user exits.

7. String Comparison

Checking user and computer values.

8. Logic Implementation

Rules:

- Rock beats Scissor
- Scissor beats Paper
- Paper beats Rock

4.SOURCE CODE

```
import random

def get_computer_choice():
    choices = ["rock", "paper", "scissor"]
    return random.choice(choices)

def get_user_choice():
    user_input = input("Enter rock, paper or scissor: ").lower()
    return user_input

def determine_winner(user, computer):
    if user == computer:
        return "It's a tie!"
    elif (user == "rock" and computer == "scissor") or \
         (user == "paper" and computer == "rock") or \
         (user == "scissor" and computer == "paper"):
        return "You win!"
    else:
        return "Computer wins!"

while True:
    user_choice = get_user_choice()
    computer_choice = get_computer_choice()

    print(f"You chose: {user_choice}")
    print(f"Computer chose: {computer_choice}")

    result = determine_winner(user_choice, computer_choice)
    print(result)

    play_again = input("Play again? (yes/no): ").lower()
    if play_again != "yes":
        break

print("Thanks for playing!")
```

5.EXPLANATION ABOUT CODE

Importing Module

import random

Used to generate random computer choices.

Function: get_computer_choice()

- Creates a list of 3 values
- Uses random.choice()
- Returns computer's move

Function: get_user_choice()

- Reads user input
- Converts to lowercase using .lower()
- Returns the choice

Function: determine_winner()

- Compares both choices
- Checks tie condition
- Checks winning conditions
- Else → computer wins

Main Game Loop (while True)

- Repeats the game
- Displays choices
- Shows result
- Asks user to play again
- Breaks loop on "no"

6.OUTPUT


IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/friends/AppData/Local/Programs/Python/Python312/be.py ==========
Enter rock, paper or scissor: rock
You chose: rock
Computer chose: rock
It's a tie!
Play again? (yes/no): yes
Enter rock, paper or scissor: paper
You chose: paper
Computer chose: scissor
Computer wins!
Play again? (yes/no): yes
Enter rock, paper or scissor: scissor
You chose: scissor
Computer chose: paper
You win!
Play again? (yes/no): no
Thanks for playing!
>>>

7.Conclusion

Python has emerged as one of the most influential and widely used programming languages in the modern software industry due to its simplicity, readability, and versatility. Its clean and human-friendly syntax allows beginners to understand programming concepts easily while offering powerful features that support advanced applications. Python supports multiple programming paradigms such as procedural, object-oriented, and functional programming, making it a flexible choice for a wide range of software solutions. With rich built-in libraries and external modules, Python can handle various tasks including web development, automation, data analysis, artificial intelligence, machine learning, and scientific computing. Its strong community support and extensive documentation further enhance the learning experience, allowing developers to find solutions quickly and stay updated with new advancements. Whether it is writing basic scripts, performing mathematical operations, controlling logical flow through conditions and loops, or creating complex real-world applications, Python provides an efficient environment that improves productivity. Overall, Python stands out as a powerful, beginner-friendly, and future-ready programming language that continues to shape the world of technology. Its adaptability, continuous growth, and wide adoption ensure that Python will remain a key programming language for education, research, and industry applications for many years to come.

8.Bibliography

1. Python Software Foundation. *Python Official Documentation*. Retrieved from https://www.python.org/doc/
2. McKinney, Wes. *Python for Data Analysis*. O'Reilly Media, 2018.
3. Sharma, R. *Introduction to Python Programming*. Tech Publications, 2020.
4. TutorialsPoint. *Python Programming Tutorial*. Retrieved from https://www.tutorialspoint.com/python
5. W3Schools. *Python Tutorial*. Retrieved from https://www.w3schools.com/python

9.GitHub Link