



**A PROJECT REPORT
ON
QUIZ GAME**

Submitted by
SASMITHA S (927623BCS103)

in partial fulfillment for the completion of the course
CGB1121- PYTHON PROGRAMMING

in
DEPARTMENT OF FRESHMAN ENGINEERING

M.KUMARASAMY COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

KARUR– 639 113

May, 2024

**M.KUMARASAMY COLLEGE OF ENGINEERING
(AUTONOMOUS)**

KARUR – 639 113

BONAFIDE CERTIFICATE

Certified that this project report titled “**QUIZ GAME**” is the bonafide work of **SASMITHA S (927623BCS103)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a course was conferred on an earlier occasion on this or any other candidate.

Signature

Mr.V.RAJESHRAM, M.E.,
SUPERVISOR.

Department of Computer Science
and Engineering,

M.Kumarasamy College of
Engineering,
Thalavapalayam, Karur-639113.

Signature

DR.K.CHITIRAKALA, M.sc., M.Phill., Ph.D.,
HEAD OF THE DEPARTMENT

Department of Freshman Engineering,

M.Kumarasamy College of Engineering
Thalavapalayam, Karur-639113.

This project report has been submitted for the Project Work - End Semester viva
voice Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **“QUIZ GAME”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfillment of the requirement of the completion of the course **CGB1121- PYTHON PROGRAMMING**.

Signature

SASMITHA S

Place: Karur

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**M.Kumarasamy College of Engineering (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. B.S.MURUGAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. K.CHITIRAKALA , M.Sc., M.Phil., Ph.D.**, Head of the department of **FRESHMAN ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project guide **Mr. V. RAJESHRAM, M.E., (Ph.D)** Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

**M.KUMARASAMY COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Vision of the Institution

To emerge as a leader among the top institutions in the field of technical education

Mission of the Institution

- ✚ Produce smart technocrats with empirical knowledge who can surmount the global challenges

- ✚ Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students

- ✚ Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

Vision of the Department

To achieve education and research excellence in Computer Science and Engineering

Mission of the Department

- ✚ To excel in academic through effective teaching learning techniques


- ✚ To promote research in the area of computer science and engineering with the focus on innovation

- ✚ To transform students into technically competent professionals with societal and ethical responsibilities

Program Educational Objectives (PEOs)

- ✚ **PEO 1:** Graduates will have successful career in software industries and R&D divisions through continuous learning.

- ✚ **PEO 2:** Graduates will provide effective solutions for real world problems in the key domain of computer science and engineering and engage in lifelong learning

 **PEO 3:** Graduates will excel in their profession by being ethically and socially responsible.

Program Outcomes (POs)

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1- Professional Skills: Ability to apply the knowledge of computing techniques to design and develop computerized solutions for the problems.

PSO2- Successful career: Ability to utilize the computing skills and ethical values in creating a successful career.

ABSTRACT

This project aims to develop a user-friendly Python quiz game to enhance users' understanding of Python concepts through interactive quizzes. The objective is to provide an engaging and interactive learning experience for users interested in Python programming. The game will feature a main menu allowing users to start the quiz or quit. Once the quiz begins, questions will be displayed, and users will input their answers. The score will be calculated and displayed throughout or at the end of the quiz, fostering a rewarding learning experience.

TABLE OF CONTENTS

TITLE

| CHAPTER | | PAGE NO. |
|----------|---|----------|
| | ABSTRACT | vi |
| | LIST OF FIGURES | Xi |
| | LIST OF ABBREVIATIONS | 1 |
| 1 | INTRODUCTION | |
| | 1.1. Introduction to Python | 2 |
| | 1.1.1. Overview | 2 |
| | 1.1.2. Programming Paradigms | 2 |
| | 1.1.3. Standard Library | 2 |
| | 1.1.4. Third-Party Libraries and Frameworks | 3 |
| | 1.1.5. Versions of Python | 3 |
| | 1.1.6. Python Tools | 3 |
| | 1.1.7. Versatility and Adoption | 4 |
| 2 | PROJECT DESCRIPTION | |
| | 2.1. Project Introduction | 5 |
| | 2.1. Project Objective | 5 |
| | 2.3. Problem Statement | 5 |
| | 2.4. Libraries Used | 5 |
| 3 | SYSTEM ANALYSIS | |
| | 3.1. Existing System | 7 |
| | 3.1.1. Disadvantages | 7 |
| | 3.2 Proposed System | 8 |
| | 3.2.1. Advantage | 8 |
| 4 | SYSTEM DESIGN & MODULES | |
| | 4.1. Block Diagram | 9 |
| | 4.2. Module Description | 9 |
| | 4.2.1. Determining Question Index | 9 |
| | 4.2.2. Scoring Mechanism | 9 |
| | 4.2.3 Loading Question | 10 |
| | 4.2.4. Checking Answers | 10 |
| | 4.2.5 Displaying Scores | 10 |
| | 4.2.6. Managing Levels | 10 |
| | 4.2.7. Providing Feedback | 10 |
| | 4.2.8. Viewing Feedback | 10 |
| | 4.2.9. Updating Feedback | 10 |

| | | |
|----------|--|----|
| | 4.2.10. Handling Errors | 10 |
| | | 10 |
| 5 | CONCLUSION & FUTURE ENHANCEMENT | |
| | 5.1. Conclusion | 11 |
| | 5.2. Future Enhancement | 12 |
| 6 | APPENDICES | |
| | Appendix A-Source code | 13 |
| | Appendix B -Screen shots | 28 |
| 7 | REFERENCE | 31 |

LIST OF FIGURES

| FIGURE NO | TITLE | PAGE NO. |
|------------------|---------------|-----------------|
| 4.1 | BLOCK DIAGRAM | 9 |

LIST OF ABBREVIATIONS

| ABBREVIATION | | MEANING |
|---------------------|---|------------------------------------|
| IDE | - | Integrated development environment |
| VS Code | - | Visual studio code |
| NLP | - | Natural Language Processing |
| GUI | - | Graphical User Interface |
| APIS | - | Application Programming |

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO PYTHON

1.1.1. OVERVIEW

Python is a widely-used, high-level programming language renowned for its readability and simplicity, making it an ideal choice for both novice and seasoned programmers. Created by Guido van Rossum and released in 1991, Python's core philosophy emphasizes code readability and straightforward syntax, allowing developers to write clear and concise code more efficiently compared to other languages like C++ or Java.

1.1.2. PROGRAMMING PARADIGMS

Python supports various programming paradigms, including procedural, object-oriented, and functional programming. This flexibility, combined with a dynamic type system and automatic memory management, facilitates the development of a wide range of applications, from simple scripts to complex software systems.

1.1.3. STANDARD LIBRARY

The language's comprehensive standard library, often referred to as "batteries-included," provides built-in modules and functions for handling many programming tasks, such as file I/O, system calls, and even web

services. This extensive library helps streamline the development process by offering ready-to-use solutions for common programming challenges.

1.1.4. THIRD-PARTY LIBRARIES AND FRAMEWORKS

One of Python's significant strengths is its extensive ecosystem of third-party libraries and frameworks. Popular libraries such as NumPy and Pandas enable efficient data manipulation and analysis, while frameworks like Django and Flask streamline web development. In the realm of machine learning and artificial intelligence, libraries like TensorFlow and PyTorch are widely adopted for building and deploying sophisticated models.

1.1.5. VERSIONS OF PYTHON

Python has undergone significant evolution since its inception, with two major versions in use today:

Python 2: Released in 2000, Python 2.x series was a major milestone and widely used for many years. However, it reached its end of life on January 1, 2020, and is no longer maintained.

Python 3: Introduced in 2008, Python 3.x series brought substantial improvements and changes to the language, such as better Unicode support, a more consistent syntax, and enhanced standard libraries. Python 3 is the recommended version for all new projects.

1.1.6. PYTHON TOOLS

Python's ecosystem includes numerous tools that enhance productivity and development experience:

- **IDEs and Code Editors:** Popular options include PyCharm, VS Code, and Jupyter Notebook, which offer features like syntax highlighting, code completion, and debugging.
- **Package Management:** Tools like pip and conda facilitate the installation and management of Python libraries and dependencies.
- **Virtual Environments:** virtualenv and venv allow developers to create isolated environments for different projects, ensuring dependency conflicts are avoided.
- **Testing Frameworks:** unittest, pytest, and nose are commonly used for writing and running tests to ensure code reliability and correctness.
- **Build Tools:** setuptools and wheel help in packaging Python projects, making them easy to distribute and install.
- **Documentation Generators:** Tools like Sphinx are used to create comprehensive documentation for Python projects.
- **Linters and Formatters:** pylint, flake8, and black help maintain code quality and consistency by enforcing coding standards and formatting.

1.1.7. VERSATILITY AND ADOPTION

Python's simplicity and versatility have led to its widespread adoption in various fields, including web development, data science, artificial intelligence, automation, and scientific computing. Its active community continually contributes to a rich repository of resources, tutorials, and documentation, making it easier for developers to learn and apply Python effectively.

CHAPTER 2

PROJECT DESCRIPTION

2.1. PROJECT INTRODUCTION

Python is a widely-used programming language known for its simplicity and readability. However, mastering Python concepts requires practice and reinforcement. To facilitate learning, this project introduces a Python quiz game designed to be user-friendly and interactive. By engaging in quizzes, users can solidify their understanding of Python concepts in a fun and immersive manner. This project aims to provide a valuable learning tool for beginners and enthusiasts alike, fostering a deeper comprehension of Python programming.

2.2. PROJECT OBJECTIVE

- **Develop a user-friendly Python quiz game:** Establishing the goal of creating a user friendly quiz game using Python.
- **Enhance understanding of Python concepts:** Aiming to improve users' understanding of Python through interactive quizzes .
- **Provide an interactive learning experience:** Offering users an engaging and interactive learning experience

2.3. PROBLEM STATEMENT

- **Lack of engaging programming education tools:** Identifying the absence of interactive tools for programming education .
- **Limited availability of Python quiz games for beginners:** Recognizing the scarcity of Python quiz games suitable for beginners.

2.4. LIBRARIES USED

- **Tkinter:** This library is used for creating the graphical user interface (GUI) for the quiz game. It provides classes and functions for creating windows, labels, buttons, text entry widgets, and other GUI elements.
- **PIL (Python Imaging Library):** PIL is used for image manipulation tasks, such as loading images from URLs and displaying them in the GUI. In the code, PIL is specifically used to open images from URLs and convert them to PhotoImage objects that can be displayed in Tkinter windows.
- **Requests:** This library is used for sending HTTP requests to retrieve data from URLs. In the code, requests is used to download image data from URLs.
- **io.BytesIO:** This library is used to create an in-memory binary stream (a BytesIO object) from the image data retrieved using requests. This is necessary to open the image data using PIL.
- **Tkinter.messagebox:** This submodule of tkinter provides functions for displaying message boxes, such as information boxes, error boxes, and warning boxes. In the code, messagebox is used to display error messages if there are issues loading images from URLs.

CHAPTER 3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

The current methods for learning Python programming often involve traditional educational resources such as textbooks, online tutorials, and static quizzes. While these resources provide valuable information, they lack the interactive and engaging elements that can make learning more effective and enjoyable.

3.1.1. DISADVANTAGES

1. **Lack of Interactivity:** Traditional learning methods often do not engage users in an interactive manner. Without interactivity, learners may struggle to stay motivated and retain information
2. **Limited Feedback:** Many educational resources do not provide immediate feedback on learners' progress, making it difficult for them to identify areas where they need improvement.
3. **Monotony:** Static quizzes and textbook exercises can become monotonous, leading to decreased motivation and engagement.
4. **Inefficient Self-Assessment:** Traditional resources often do not offer effective tools for self-assessment, making it hard for learners to gauge their understanding and track their progress over time.
5. **Lack of Personalization:** These methods often do not adapt to the individual learner's pace and style, which can hinder effective learning.

3.2. PROPOSED SYSTEM

The proposed system is a Python quiz game that addresses the shortcomings of traditional learning methods by providing an interactive and engaging platform for learning Python programming. This system is designed to make learning more enjoyable and effective through the use of gamification techniques.

3.2.1. ADVANTAGES

1. **Enhanced Interactivity:** The quiz game format ensures that users are actively engaged in the learning process. By answering questions and receiving immediate feedback, learners stay motivated and involved.
2. **Immediate Feedback:** The game provides real-time feedback on the users' answers, allowing them to quickly identify and correct mistakes. This helps reinforce learning and improve retention.
3. **Engagement through Gamification:** By incorporating elements of gamification, such as scoring and progress tracking, the system makes learning more fun and rewarding. This can lead to increased motivation and sustained interest in learning.
4. **Effective Self-Assessment:** The quiz game allows users to assess their knowledge and track their progress over time. This helps learners identify strengths and weaknesses and focus on areas that need improvement.
5. **Personalized Learning Experience:** The system can adapt to the user's pace, providing a more personalized learning experience that caters to individual needs and learning styles.
6. **Practical Application:** By answering questions and solving problems within the game, users get practical experience with Python programming concepts, which can enhance their understanding and skills.

CHAPTER 4

SYSTEM DESIGN & MODULES

4.1. BLOCK DIAGRAM

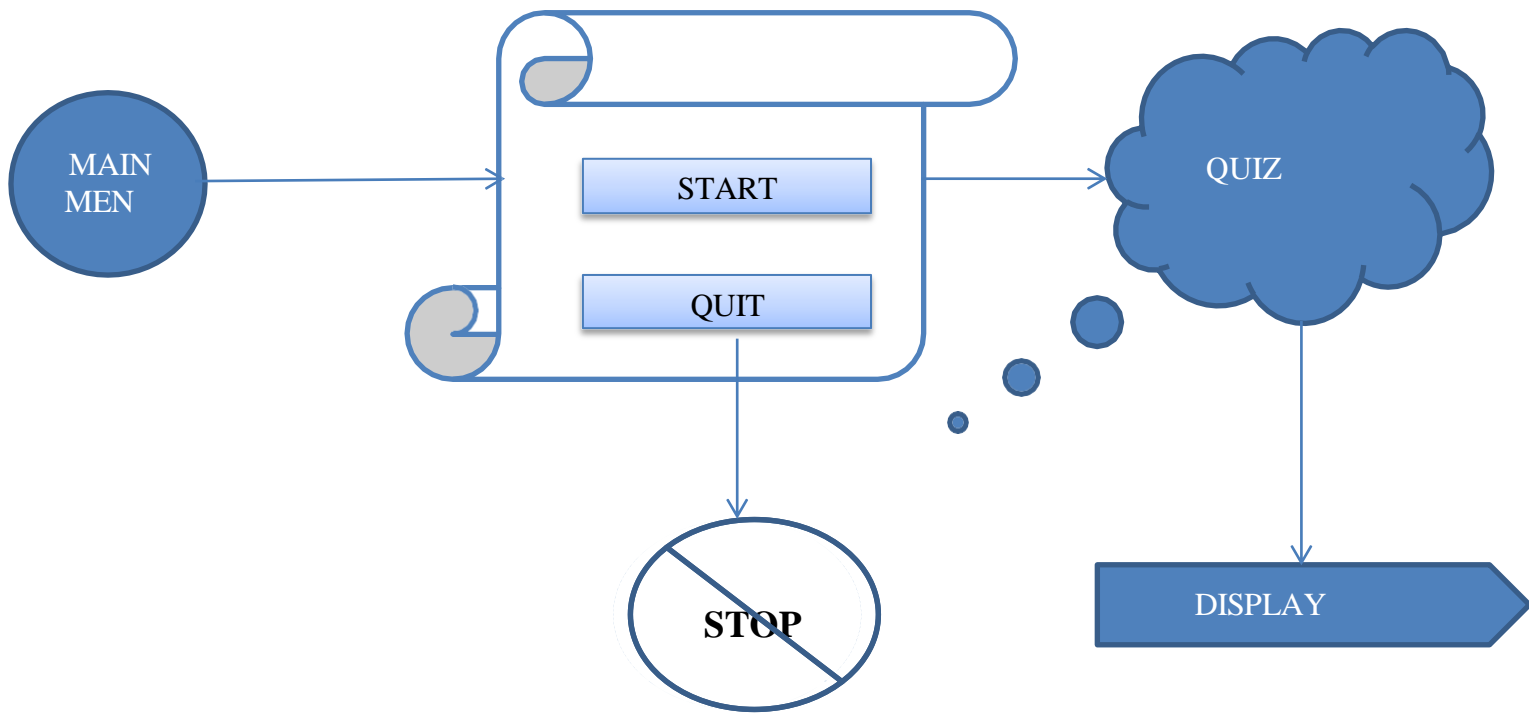


Fig. 4.1. Block Diagram

4.2. MODULE DESCRIPTION

4.1.1 Determining Question Index:

Description: This functionality determines the index of the current question based on the current level and the number of questions per level. It ensures that the correct questions are loaded for each level.

4.1.2 Scoring Mechanism:

Description: The scoring mechanism calculates the player's score based on their correct answers. It tracks the number of correct answers and awards points accordingly. The scoring mechanism may also determine the player's rank or reward based on their score at the end of each level.

4.1.3 Loading Questions:

Description: This functionality loads questions from the predefined list based on the current question index. It ensures that the correct set of questions is presented to the player during gameplay.

4.1.4 Checking Answers:

Description: The checking answers functionality compares the player's selected answer with the correct answer for the current question. It determines whether the player's response is correct or incorrect and updates the score accordingly.

4.1.5 Displaying Scores:

Description: After completing a level or the entire game, this functionality displays the player's score. It may also provide feedback on the player's performance, such as their rank or reward based on their score.

4.1.6 Managing Levels:

Description: This functionality manages the progression through different levels of the game. It tracks the current level and ensures that the appropriate questions are loaded for each level.

4.1.7 Providing Feedback:

Description: Allows players to provide feedback on the game. This feedback can be related to the questions, gameplay experience, or any suggestions for improvement.

4.1.8 Viewing Feedback:

Description: Displays feedback provided by players. This functionality allows the game developers to review feedback and make necessary improvements to enhance the game's quality.

4.1.9 Updating Feedback:

Description: Enables the modification or clearing of feedback data. This functionality ensures that the feedback system remains up-to-date and useful for improving the game.

4.1.10 Handling Errors:

Description: This functionality deals with any errors or exceptions that may occur during gameplay. It ensures a smooth user experience by gracefully handling unexpected situations.

CHAPTER 5

CONCLUSION & FUTURE ENHANCEMENT

5.1. CONCLUSION

The modulus operator proves to be a crucial element in programming, particularly when handling tasks that require cyclic behavior or periodic actions. In managing a list of questions, using the modulus operator to cycle through the questions is a simple yet effective strategy. It ensures that, regardless of how many times you iterate, you always loop back to the beginning of the list without needing complex conditional statements or additional logic. This method is particularly useful in educational software, quiz applications, and any scenario where repeated access to a fixed set of items is necessary. It makes the code cleaner, more maintainable, and less error-prone.

In the realm of scoring mechanisms, the modulus operator offers a robust solution for implementing bonus systems based on a specific pattern or threshold. By using the modulus operator, developers can easily check if a certain number of correct answers have been reached and apply bonuses accordingly. This approach can be tailored to various scoring rules, such as awarding extra points for every fifth correct answer or resetting the bonus count after each award. It simplifies the implementation of these rules, making the scoring system more flexible and easier to manage. Overall, the modulus operator enhances the functionality and efficiency of both list management and scoring systems, demonstrating its value in practical programming applications.

5.2. FUTURE ENHANCEMENT

Looking ahead, there are numerous opportunities to expand the use of the modulus operator in more complex and dynamic systems. One potential area of enhancement is in adaptive learning platforms. By integrating the modulus operator, these systems could not only cycle through a list of questions but also adjust the difficulty level based on the learner's performance. For example, the system could increase the challenge after a certain number of correct answers, using the modulus operator to determine when to escalate the difficulty. This could create a more personalized and engaging learning experience, keeping users motivated and adequately challenged.

In addition to educational applications, the modulus operator could play a significant role in other domains such as load balancing in distributed systems. Here, the modulus operator can be used to distribute tasks evenly across servers, ensuring that no single server is overwhelmed. By assigning tasks based on the task ID modulo the number of servers, a simple yet effective load balancing mechanism can be established. Similarly, in cyclic scheduling tasks, such as rotating duties among employees or scheduling periodic maintenance activities, the modulus operator can ensure a fair and balanced distribution of work.

Enhancing the scoring mechanism further could involve implementing more sophisticated bonus structures. For example, dynamic bonus thresholds that adjust based on user engagement or performance metrics could provide a more nuanced scoring system. Real-time feedback systems could use the modulus operator to trigger instant rewards or penalties, creating a more interactive and responsive user experience. These enhancements would not only improve the functionality of the systems but also provide richer, more adaptive experiences for users across various applications.

APPENDICES

APPENDIX A-SOURCE CODE

```
import tkinter as tk
from tkinter import messagebox
from PIL import Image, ImageTk
import requests
from io import BytesIO

# Dummy questions data (You should have more comprehensive data for 100
# levels)
questions = [
    {"question": "What is the output of print(23)?", "options": ["6", "8", "9",
        "12"], "answer": "8", "reason": "23 in octal is 8 in decimal."},
    {"question": "Which of the following is a mutable data type in Python?",
        "options": ["Tuple", "string", "integer", "list"], "answer": "list",
        "reason": "Lists are mutable, meaning their contents can be changed."},
    {"question": "What is the keyword used to define a function in Python?",
        "options": ["func", "def", "define", "function"], "answer": "def",
        "reason": "The 'def' keyword is used to define a function in Python."},
    {"question": "What will be the output of print(type(5))?", "options": [
        "<class 'int'>", "<type 'int'>", "int", "integer"], "answer": "<class 'int'>",
        "reason": "The type() function returns the type of the object, which in
        this case is an integer."},
    {"question": "How do you create a dictionary in Python?", "options": [
        "{}", "[]", "()", "::"], "answer": "{}", "reason": "Dictionaries are created
        using curly braces {}."},
```



```

{"question": "Which of the following is used to handle exceptions in Python?", "options": ["try and catch", "try and except", "catch and except", "catch and finally"], "answer": "try and except", "reason": "Python uses 'try' and 'except' blocks to handle exceptions."},
{"question": "What is the output of 'Hello World'.lower()?", "options": ["hello world", " Hello World", "HELLO WORLD", "None the above"], "answer": "hello world", "reason": "The lower() method converts all characters in a string to lowercase."},
{"question": "Which of the following functions can convert a string to a list in Python?", "options": ["list()", "convert()", "split()", "parse()"], "answer": "split()", "reason": "The split() method splits a string into a list."},
{"question": "How can you remove all whitespace from the start and end of a string in Python?", "options": ["trim()", "strip()", "rstrip()", "lstrip()"], "answer": "strip()", "reason": "The strip() method removes whitespace from both ends of a string."},
{"question": "Which module in Python is used for regular expressions", "options": ["re", "regex", "rexp", "reg"], "answer": "re", "reason": "The 're' module is used for working with regular expressions in Python."},
{"question": "How do you get the length of a list my_list in Python?", "options": ["size(my_list)", "len(my_list)", "count(my_list)", "length(my_list)"], "answer": "len(my_list)", "reason": "The len() function returns the length of a list."},
{"question": "What does the expression 5 // 2 evaluate to in Python?", "options": ["2.5", "2", "2.0", "3"], "answer": "2", "reason": "The // operator performs integer division, discarding the remainder."},
{"question": "Which keyword is used to create an anonymous function in Python?", "options": ["lambda", "func", "def", "None of the above"], "answer": "lambda", "reason": "The 'lambda' keyword is used to create small anonymous functions."},

```

```

{"question": "Which of the following is not a valid variable name in Python?", "options": [" my_var", "var1", "1var", "_var"], "answer": "1var", "reason": "Variable names cannot start with a number."},
{"question": "What will be the output of the following code: print(10 % 3)?", "options": ["1", "2", "3", "0"], "answer": "1", "reason": "The '%' operator returns the remainder of the division."},
{"question": "Which function is used to read a single line from the user input in Python?", "options": ["input()", "raw_input()", " scan()", " read()"], "answer": "input()", "reason": "The input() function reads a line from input, converting it to a string."},
{"question": "What is the correct way to import a module in Python?", "options": ["include math", "import math", "using math", "from math include"], "answer": "import math", "reason": "The 'import' keyword is used to import modules in Python."},
{"question": "Which data type is used to store true/false values in Python?", "options": ["int", "float", "bool", "str"], "answer": "bool", "reason": "The 'bool' data type is used to store boolean values."},
{"question": "What will be the output of bool(0) in Python?", "options": [" True", " False", ") None", "0"], "answer": " False", "reason": "The bool() function returns False for zero values."},
{"question": "Which of the following statements is used to create a class in Python?", "options": ["class ClassName:", " class ClassName()", "create ClassName:", "define ClassName:"], "answer": " class ClassName:", "reason": "Classes in Python are created using the 'class' keyword."},
{"question": "What will be the output of the following code: print('Hello' + 'World')?", "options": ["Hello World", " HelloWorld", " HelloWorld", "Hello+World"], "answer": " HelloWorld", "reason": "The '+' operator concatenates strings."},

```

```

{"question": "Which of the following is the correct way to define a list in Python?", "options": [" list = (1, 2, 3)", "list = [1, 2, 3]", " list= {1, 2, 3}", "list = 1, 2, 3"], "answer": "list = [1, 2, 3]", "reason": "Lists are defined using square brackets [] in Python."},
{"question": "What will be the output of print(10 > 9)?", "options": ["True", "False", "10", "none above"], "answer": "True", "reason": "This expression evaluates to True since 10 is greater than 9."},
{"question": "Which of the following is not a Python built-in function?", "options": ["max()", "min()", "len()", "echo()"], "answer": "echo()", "reason": "The 'echo()' function is not a built-in function in Python."},
{"question": "How do you start a comment in Python?", "options": [" /*", "#", "/*", "<!--"], "answer": "#", "reason": "In Python, comments start with the '#' symbol."},
{"question": "Which method can be used to replace parts of a string in Python?", "options": ["replace()", " switch()", " substitute()", "change()"], "answer": "replace()", "reason": "The 'replace()' method is used to replace parts of a string in Python."},
{"question": "How do you create a tuple in Python?", "options": ["{1, 2, 3}", "[1, 2, 3]", " (1, 2, 3)", "<1, 2, 3>"], "answer": " (1, 2, 3)", "reason": "Tuples are created using parentheses () in Python."},
{"question": "What does the len() function do in Python?", "options": ["Calculates the length of a string", "Calculates the length of a list", "Calculates the length of a tuple", "All of the above"], "answer": "All of the above", "reason": "The len() function can be used to calculate the length of strings, lists, and tuples in Python."},
{"question": "How do you access the first element of a list my_list in Python?", "options": ["my_list[1]", "my_list[0]", "my_list[2]", "my_list[First]"], "answer": " my_list[0]", "reason": "List indexing in Python starts from 0, so my_list[0] accesses the first element."},

```

```

{"question": "Which of the following is used to define a block of code in Python?", "options": ["Curly braces {}", "Parentheses ()", "Indentation", "Square brackets []"], "answer": "Indentation", "reason": "In Python, blocks of code are defined by indentation."},
{"question": "Which keyword is used to end a function in Python?", "options": ["end", "return", "stop", "finish"], "answer": "return", "reason": "The 'return' keyword is used to exit a function and return a value."},
{"question": "Which of the following will create an empty set in Python?", "options": [{"set()"}, "set()", "[]", "set{ }"], "answer": "set()", "reason": "The 'set()' function creates an empty set in Python."},
{"question": "How do you convert a string s to an integer in Python?", "options": ["int(s)", "str(s)", "float(s)", "list(s)"], "answer": "int(s)", "reason": "The 'int()' function is used to convert a string to an integer in Python."},
{"question": "What is the output of print(5 == 5.0)?", "options": ["True", "False", "None", "Error"], "answer": "True", "reason": "The expression evaluates to True because 5 and 5.0 are considered equal in Python."},
{"question": "Which operator is used for string concatenation in Python?", "options": ["+", "*", "-", "/"], "answer": "+", "reason": "The '+' operator is used for string concatenation in Python."},
{"question": "Which of the following is a keyword in Python?", "options": ["eval", "assert", "yield", "print"], "answer": "yield", "reason": "'yield' is a keyword used in Python for generator functions."},
{"question": "How do you open a file for reading in Python?", "options": ["open(filename, 'r')", "open(filename, 'w')", "open(filename, 'a')", "open(filename, 'x')"], "answer": "open(filename, 'r')", "reason": "To open a file for reading, use 'r' as the mode parameter in the 'open()' function."},

```

```

{"question": "What is the output of print(3 < 4 and 2 > 1)?", "options":
  ["True", "False", "3", "2"], "answer": "True", "reason": "The 'and'
  operator returns True if both conditions are True."},
{"question": "How do you create a new object from a class in Python?",
  "options": ["object = ClassName()", "object = ClassName()", "object =
  ClassName", "object = new ClassName()", "object = create
  ClassName()"], "answer": "object = ClassName()", "reason": "To create
  a new object from a class, you call the class with parentheses ()."},
{"question": "What will be the output of print(10 > 9)?", "options":
  ["True", "False", "10", "None of the above"], "answer": "True",
  "reason": "This expression evaluates to True since 10 is greater than
  9."},
{"question": "Which of the following is not a Python built-in function?",
  "options": ["max()", "min()", "len()", "echo()"], "answer": "echo()",
  "reason": "The 'echo()' function is not a built-in function in Python."},
{"question": "How do you start a comment in Python?", "options": ["//",
  "#", "/*", "<!--"], "answer": "#", "reason": "In Python, comments start
  with the '#' symbol."},
{"question": "Which method can be used to replace parts of a string in
  Python?", "options": ["replace()", "switch()", "substitute()",
  "change()"], "answer": "replace()", "reason": "The 'replace()' method is
  used to replace parts of a string in Python."},
{"question": "How do you create a tuple in Python?", "options": ["{1, 2,
  3}", "[1, 2, 3]", "(1, 2, 3)", "<1, 2, 3>"], "answer": "(1, 2, 3)", "reason":
  "Tuples are created using parentheses () in Python."},
{"question": "What does the len() function do in Python?", "options":
  ["Calculates the length of a string", "Calculates the length of a list",
  "Calculates the length of a tuple", "All of the above"], "answer": "All of
  the above", "reason": "The len() function can be used to calculate the
  length of strings, lists, and tuples in Python."},

```

```
{
    "question": "How do you access the first element of a list `my_list` in Python?",
    "options": ["my_list[1]", "my_list[0]", "my_list[2]", "my_list[First]"],
    "answer": "my_list[0]",
    "reason": "List indexing in Python starts from 0, so my_list[0] accesses the first element."
},
{
    "question": "What is the output of print('Python'[2:4])?",
    "options": ["Py", "th", "ho", "yt"],
    "answer": "th",
    "reason": "String slicing in Python returns the substring starting from the index specified before the colon (:) up to, but not including, the index specified after the colon."
},
{
    "question": "Which of the following will raise an error in Python?",
    "options": ["5 + 'a'", "5 + 5", "'a' + 'b'", "'5' + '5'"],
    "answer": "5 + 'a'",
    "reason": "You cannot perform addition between an integer and a string in Python."
},
{
    "question": "What is the correct file extension for Python files?",
    "options": [".pyth", ".pyt", ".pt", ".py"],
    "answer": ".py",
    "reason": "Python files typically have the '.py' extension."
},
{
    "question": "Which of the following is the correct way to define a list in Python?",
    "options": ["list = (1, 2, 3)", "list = [1, 2, 3]", "list = {1, 2, 3}", "list = 1, 2, 3"],
    "answer": "list = [1, 2, 3]",
    "reason": "Lists are defined using square brackets [] in Python."
}
```

]

```
class QuizGame:
    def __init__(self, root):
        self.root = root
        self.root.title("Python Quiz Game")
        self.root.geometry("800x600")
        self.root.config(bg="lightyellow")

        self.score = 0
        self.current_level = 1
```

```

self.current_question_index = 0

self.questions_per_level = 2


MAIN_MENU_IMAGE_URL = "https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQOqrTlMuhdDNlh8Qvo0o6T
AM137Xz-fqsC7A&usqp=CAU"

LEVEL_IMAGE_URL =
"https://t3.ftcdn.net/jpg/07/97/10/12/240_F_797101292_8XYB2vRnN
CkRT6Fx9bhd6JqbCIf9YOsX.jpg" # Replace with your level image
URL


self.main_menu_image =
self.load_image_from_url(MAIN_MENU_IMAGE_URL)
self.level_image = self.load_image_from_url(LEVEL_IMAGE_URL)


self.create_main_menu()


def load_image_from_url(self, url):
    try:
        response = requests.get(url)
        response.raise_for_status()
        image_data = response.content
        image = Image.open(BytesIO(image_data))
        return ImageTk.PhotoImage(image)
    except requests.exceptions.RequestException as e:
        messagebox.showerror("Image Load Error", f"Failed to load image
from {url}\n{e}")
        return None


def create_main_menu(self):

```

```

self.clear_widgets()
if self.main_menu_image:
    image_label = tk.Label(self.root, image=self.main_menu_image)
    image_label.pack(pady=200)
tk.Button(self.root, text="Start Game", font=("Arial", 18),
    command=self.start_game).pack(pady=10)
tk.Button(self.root, text="View Feedback", font=("Arial", 18),
    command=self.show_feedback).pack(pady=10)
tk.Button(self.root, text="Update Feedback", font=("Arial", 18),
    command=self.update_feedback).pack(pady=10)

def start_game(self):
    self.clear_widgets()
    self.display_level_map()

def display_level_map(self):
    self.clear_widgets()
    tk.Label(self.root, text="Select Level", font=("Arial", 24),
        bg="lightblue").pack(pady=20)

    frame = tk.Frame(self.root, bg="lightyellow")
    frame.pack(fill=tk.BOTH, expand=1)

    canvas = tk.Canvas(frame, bg="lightyellow")
    canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=1)

    scrollbar = tk.Scrollbar(frame, orient=tk.VERTICAL,
        command=canvas.yview)
    scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

```



```

canvas.configure(yscrollcommand=scrollbar.set)
canvas.bind('<Configure>', lambda e:
    canvas.configure(scrollregion=canvas.bbox("all")))

second_frame = tk.Frame(canvas, bg="lightyellow")
canvas.create_window((0, 0), window=second_frame, anchor="nw")

for level in range(1, 26):
    button = tk.Button(second_frame, text=f"Level {level}",
        command=lambda lvl=level: self.start_level(lvl))
    button.grid(row=(level - 1) // 5, column=(level - 1) % 5, padx=10,
        pady=5)

def start_level(self, level):
    self.current_level = level
    self.current_question_index = (level - 1) * self.questions_per_level
    self.score = 0
    self.load_question()

def load_question(self):
    if self.current_question_index < len(questions) and
        self.current_question_index < self.current_level *
        self.questions_per_level:
        self.clear_widgets()
        question_data = questions[self.current_question_index]

    if self.level_image:
        image_label = tk.Label(self.root, image=self.level_image)
        image_label.pack(pady=20)

```

```

self.question_label = tk.Label(self.root,
text=question_data["question"], font=("Arial", 18), bg="lightblue")
self.question_label.pack(pady=20)

```

```

self.option_buttons = []
for i, option in enumerate(question_data["options"]):
    button = tk.Button(self.root, text=option, font=("Arial", 14),
command=lambda opt=option: self.check_answer(opt,
question_data["reason"]))
    button.pack(pady=5)
    self.option_buttons.append(button)
tk.Button(self.root, text="Quit", font=("Arial", 14),
command=self.create_main_menu).pack(pady=10)

```

```

else:

```

```

    self.show_score()

```

```

def check_answer(self, selected_option, reason):

```

```

    correct_answer = questions[self.current_question_index]["answer"]

```

```

    if selected_option == correct_answer:

```

```

        self.score += 1

```

```

        messagebox.showinfo("Correct!", f"Correct answer! {reason}")

```

```

    else:

```

```

        messagebox.showerror("Incorrect", f"Incorrect answer. {reason}")

```

```

self.current_question_index += 1

```

```

if self.current_question_index % self.questions_per_level == 0:

```

```

    self.show_score()

```

```

else:

```

```

    self.load_question()

```

```

def show_score(self):
    self.clear_widgets()
    reward = "Gold" if self.score == 2 else "Silver" if self.score == 1 else
        "Bronze"
    score_text = f"Level {self.current_level} completed! Your score:
        {self.score} / {self.questions_per_level}\nReward: {reward}"
    tk.Label(self.root, text=score_text, font=("Arial", 18),
        bg="lightblue").pack(pady=20)

    feedback_button = tk.Button(self.root, text="Provide Feedback",
        font=("Arial", 18), command=self.provide_feedback)
    feedback_button.pack(pady=10)

    tk.Button(self.root, text="Next Level", font=("Arial", 18),
        command=self.start_next_level).pack(pady=10)
    tk.Button(self.root, text="Main Menu", font=("Arial", 18),
        command=self.create_main_menu).pack(pady=10)

def provide_feedback(self):
    self.clear_widgets()
    tk.Label(self.root, text="Please provide your feedback:", font=("Arial",
        18), bg="lightblue").pack(pady=20)

    self.feedback_text = tk.Text(self.root, height=10, width=50)
    self.feedback_text.pack(pady=20)

    submit_button = tk.Button(self.root, text="Submit Feedback",
        font=("Arial", 18), command=self.save_feedback)
    submit_button.pack(pady=10)

```

```

tk.Button(self.root, text="Back", font=("Arial", 18),
          command=self.show_score).pack(pady=10)

def save_feedback(self):
    feedback = self.feedback_text.get("1.0", tk.END).strip()
    if feedback:
        with open("feedback.txt", "a") as f:
            f.write(f"Level {self.current_level} Feedback: {feedback}\n")
        messagebox.showinfo("Feedback Received", "Thank you for your
        feedback!")
    self.show_score()

def start_next_level(self):
    if self.current_level < 25:
        self.start_level(self.current_level + 1)
    else:
        messagebox.showinfo("Quiz Completed", "You have completed all
        levels!")
        self.create_main_menu()

def show_feedback(self):
    self.clear_widgets()
    try:
        with open("feedback.txt", "r") as f:
            feedback_data = f.read()
        if feedback_data:
            tk.Label(self.root, text="Feedback from Users:", font=("Arial", 18),
            bg="lightblue").pack(pady=20)
            feedback_text = tk.Text(self.root, height=15, width=60)

```

```

        feedback_text.insert(tk.END, feedback_data)

        feedback_text.pack(pady=20)
    else:
        tk.Label(self.root, text="No feedback available.", font=("Arial",
18), bg="lightblue").pack(pady=20)
except FileNotFoundError:
    tk.Label(self.root, text="No feedback available.", font=("Arial", 18),
bg="lightblue").pack(pady=20)

tk.Button(self.root, text="Back to Main Menu", font=("Arial", 18),
command=self.create_main_menu).pack(pady=10)

def update_feedback(self):
    self.clear_widgets()
    try:
        with open("feedback.txt", "r") as f:
            feedback_data = f.read()
        if feedback_data:
            tk.Label(self.root, text="Feedback from Users:", font=("Arial", 18),
bg="lightblue").pack(pady=20)
            feedback_text = tk.Text(self.root, height=15, width=60)
            feedback_text.insert(tk.END, feedback_data)
            feedback_text.pack(pady=20)
            tk.Button(self.root, text="Clear Feedback", font=("Arial", 18),
command=self.clear_feedback).pack(pady=10)
        else:
            tk.Label(self.root, text="No feedback available.", font=("Arial",
18), bg="lightblue").pack(pady=20)
    except FileNotFoundError:

```

```
tk.Label(self.root, text="No feedback available.", font=("Arial", 18),  
bg="lightblue").pack(pady=20)
```

```
tk.Button(self.root, text="Back to Main Menu", font=("Arial", 18),  
command=self.create_main_menu).pack(pady=10)
```

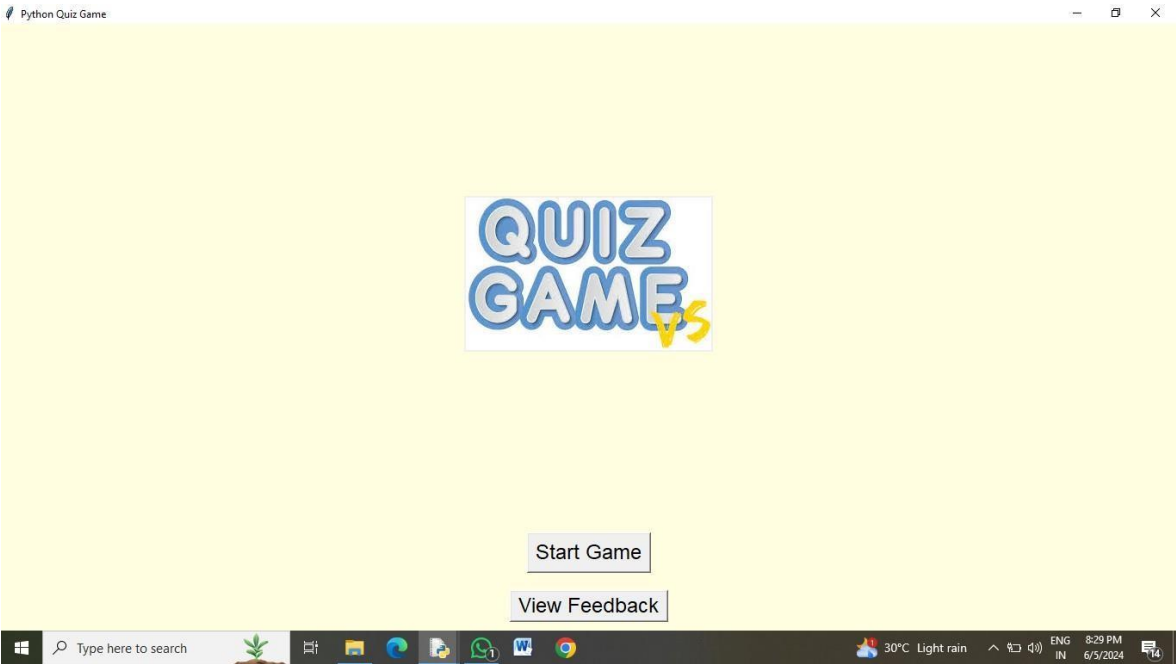
```
def clear_feedback(self):  
    with open("feedback.txt", "w") as f:  
        f.truncate(0)  
    messagebox.showinfo("Feedback Cleared", "All feedback has been  
cleared.")
```

```
def clear_widgets(self):  
    for widget in self.root.winfo_children():  
        widget.destroy()
```

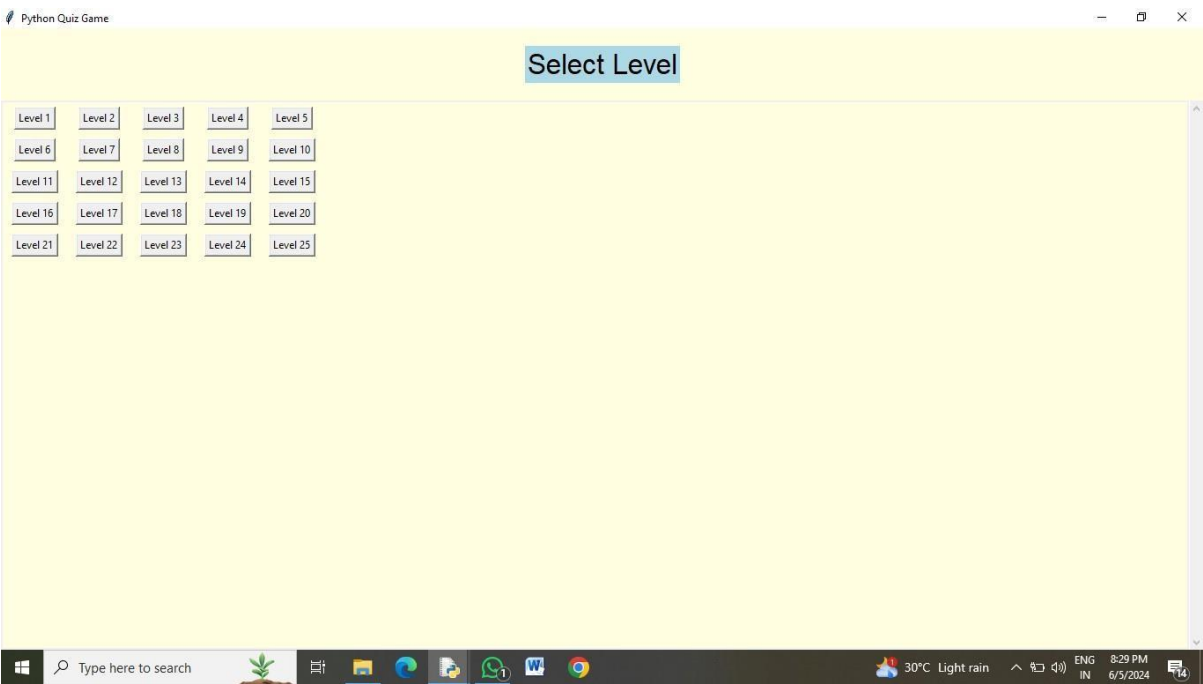
```
if __name__ == "__main__":  
    root = tk.Tk()  
    game = QuizGame(root)  
    root.mainloop()
```

APPENDIX B -SCREEN SHOTS

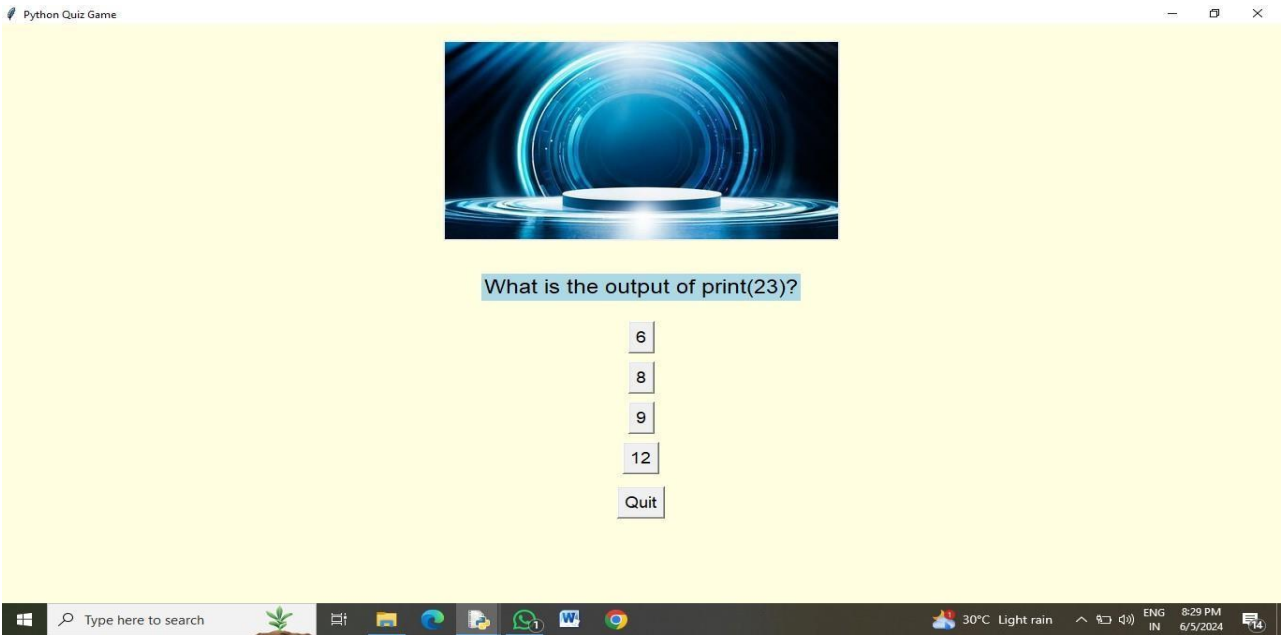
1. MAIN MENU:



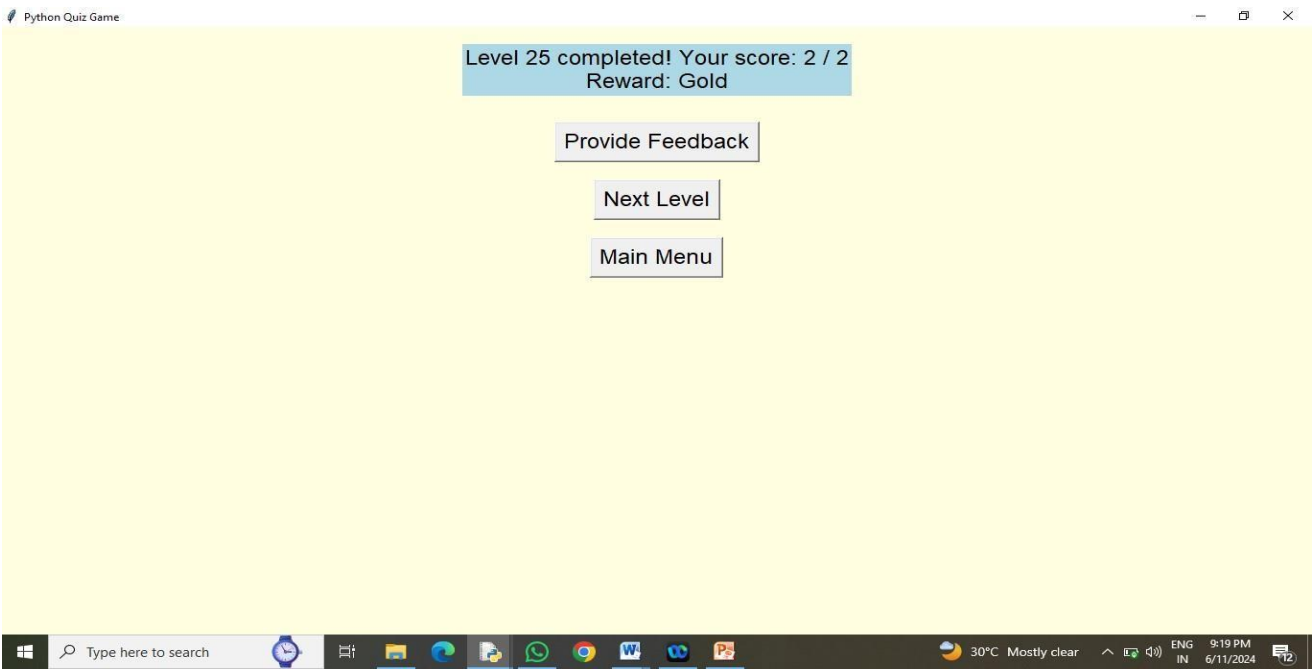
2. LEVEL DISPLAY:



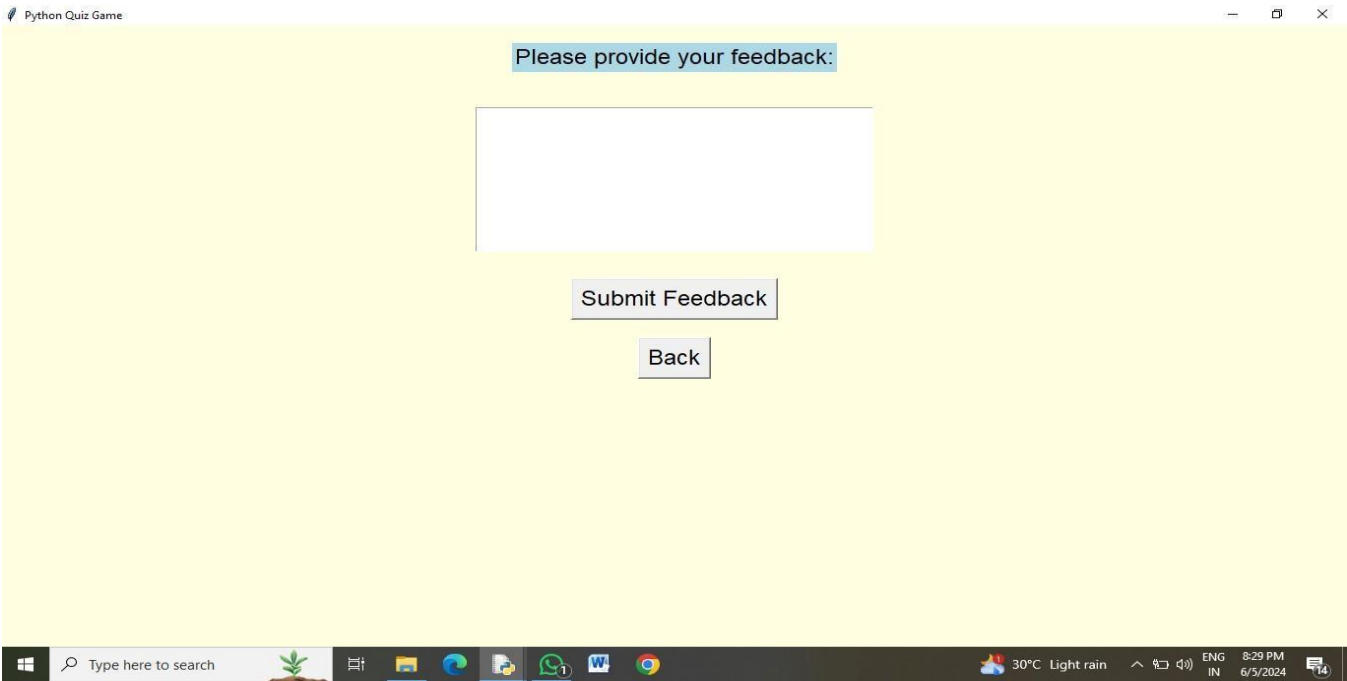
3. DISPLAY QUESTIONS AND QUIT OPTION IS GIVEN:



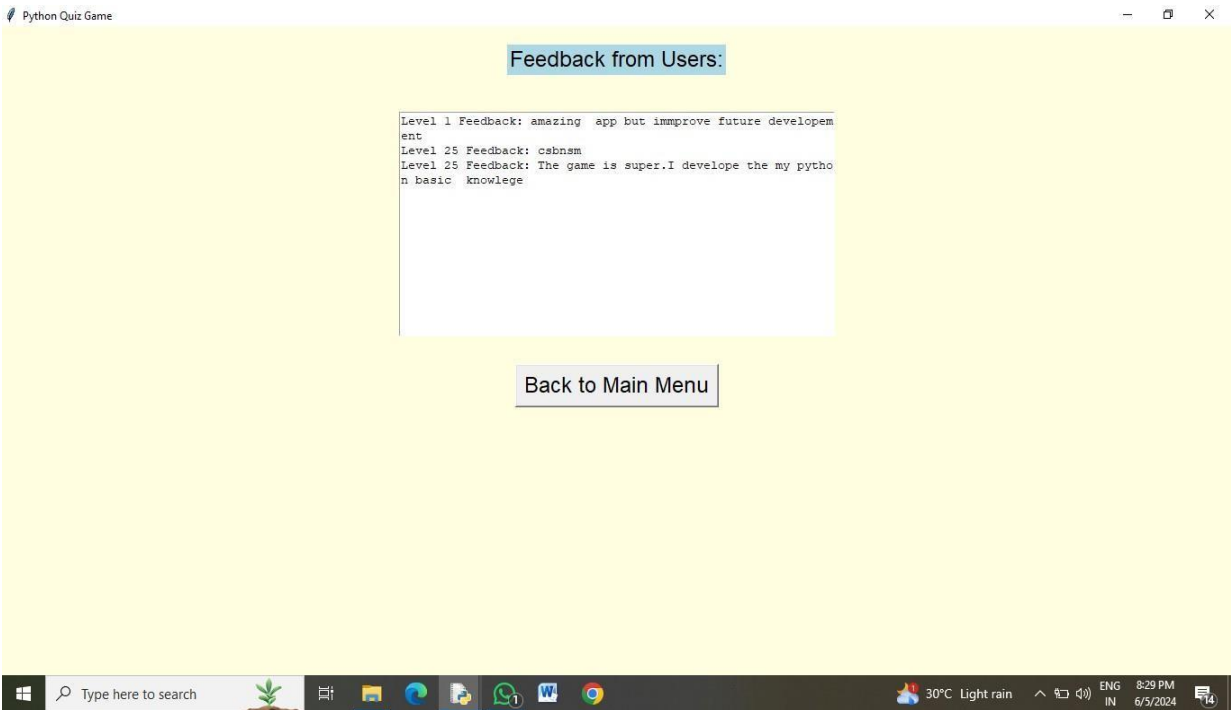
4. PROVIDE REWARDS AND SCORE:



5. PROVIDE USER FEEDBACK:



6. IEW USER FEEDBACK:



REFERENCE

1. **E-Learning and the Science of Instruction** by Ruth C. Clark and Richard E. Mayer - This book provides foundational knowledge on the principles of e-learning and instructional design.
2. **Gamification of Learning and Instruction** by Karl M. Kapp - This book explores the use of game-based mechanics in educational contexts, supporting the design and implementation of the quiz game.
3. **Interactive Learning Systems Evaluation** by Thomas C. Reeves - This book discusses methods for evaluating the effectiveness of interactive learning systems, providing a framework for assessing the quiz game's impact.