Project Report

Title: Basic quiz game

Submitted by: [Samiya Mujawar and Vishakha Padavalkar]
Student Roll No: [43,45]
Student EID: [2303111059 and 2303111035]
Submitted to: Mr. R. S. Khamitkar
Department: Electronics and Computer Engineering
Date: 21/4/2025

---

1. Project Overview

This project is a basic quiz game built using Python to test users' general knowledge. It presents multiple-choice questions and provides instant feedback on the selected answers. The game tracks the user's score and shows the final result at the end of the quiz. It uses simple Python concepts like lists, loops, and conditional statements.

---

2. Problem Statement

In today’s digital era, interactive applications are increasingly preferred for both entertainment and self-assessment. However, many existing quiz platforms are either overly complex or require constant internet connectivity. There is a need for a simple, accessible quiz game that can be used by anyone to test their knowledge, enjoy casual gameplay, or even practice for fun. This project aims to develop a basic quiz game using Python that presents multiple-choice questions, gives instant feedback, tracks scores, and offers a user-friendly experience suitable for users of all ages and purposes.

3. Technology Stack

- **Programming Language:** Python 3.x
- **IDE:** Python IDLE

---

4. Implementation

The implementation follows a simple logic:

1. Create a set of multiple-choice questions and store them in a Python dictionary or JSON file.

2. Implement the game logic to display questions in the console, collect user input, and track the score.

3. Use simple print statements and input functions to show the questions, options, and feedback, then display the final score.

---

5. Code Snippet

```
# Define the quiz questions and options
questions = [
    {
        "question": "What is the capital of France?",
        "options": ["Berlin", "Madrid", "Paris", "Rome"],
        "answer": "Paris"
    },
    {
        "question": "Which planet is known as the Red Planet?",
        "options": ["Earth", "Mars", "Jupiter", "Saturn"],
        "answer": "Mars"
    },
    {
        "question": "What is the largest ocean on Earth?",
        "options": ["Atlantic", "Indian", "Arctic", "Pacific"],
        "answer": "Pacific"
    }
]

# Function to run the quiz game
```

```
def run_quiz():
    score = 0
    for q in questions:
        print("\n" + q["question"])
        for i, option in enumerate(q["options"], 1):
            print(f"{i}. {option}")

        user_answer = input("Select the correct option (1/2/3/4): ")
        if q["options"][int(user_answer) - 1] == q["answer"]:
            print("Correct!")
            score += 1
        else:
            print("Incorrect!")

    print(f"\nYour final score is: {score}/{len(questions)}")

# Run the quiz
run_quiz()
```

---

6. Sample Test Cases

Question	Options	User Input	Result
What is the capital of France?	1. Berlin, 2. Madrid, 3. Paris, 4. Rome	3	Correct!
Which planet is known as the Red Planet?	1. Earth, 2. Mars, 3. Jupiter, 4. Saturn	1	Incorrect!
What is the largest ocean on Earth?	1. Atlantic, 2. Indian, 3. Arctic, 4. Pacific	4	Correct!
Final Score			2/3

---

7. Challenges Faced

During the development of the Basic Quiz Game, several challenges can arise. One key challenge is handling user input errors, as users may select invalid options outside the given range, which can disrupt the game flow. Another challenge is ensuring accurate score tracking, as it's important to update and display the score correctly after each question. Storing and retrieving questions efficiently from files like JSON or text can also be tricky, requiring careful structure to prevent errors. Finally, designing a clear and user-friendly text-based interface is essential, as it needs to be intuitive while presenting the game in a simple yet engaging manner.

---

8. Results and Observations

The quiz game successfully tracks user input, provides immediate feedback, and calculates the score accurately. It is easy to play, with a simple text-based interface, making it accessible and engaging for users of all ages.

Project Submission

Repository Link: [The repository Link]

9. Signatures

Student Signature: _____

Guide Signature: _____

Date: _____