**INDUSTRIAL INTERNSHIP REPORT ON**

**“QUIZ GAME”**

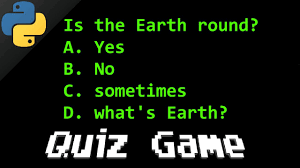
**PREPARED BY**

**S Jaswanth Kumar**

|  |
| --- |
| *Executive Summary* |
| This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).  This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.  My project was Quiz Game.  This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship. |

.**PROJECT REPORT:**

**QUIZ GAME PREPARATION USING PYTHON**

**INTROCUCTION**

The Quiz Game project aims to develop a console-based quiz game using the Python programming language. The game will present multiple-choice questions to the players and provide immediate feedback on their answers. It will track scores and display them at the end of the game.

**OBJECTIVES**

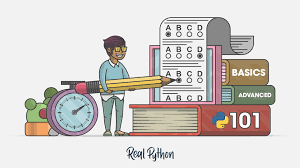
- Develop an interactive quiz game using Python.

- Create a database of questions with multiple-choice options.

- Track and calculate the scores of the players.

- Display the final score and provide feedback on each question.

- Allow players to play the game multiple times.



**TECHNOLOGIES USED**

- Python: Programming language used for game development.

 - SQLite: Database management system for storing questions and answers.

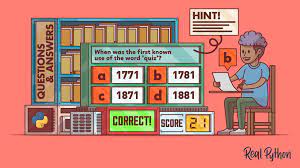
**SYSTEM DESIGN**

The Quiz Game project can be divided into the following components:

- Main Program Logic: Handles game flow, displays questions, and processes user input.

- Database Management: Stores and retrieves questions from an SQLite database.

- Score Tracking: Keeps track of the player's score and calculates the final score.

**** - User Interface: Displays questions, options, and feedback to the players.

**IMPLEMENTATION OF STEPS**

Step 1: Create an SQLite database to store the questions and answers.

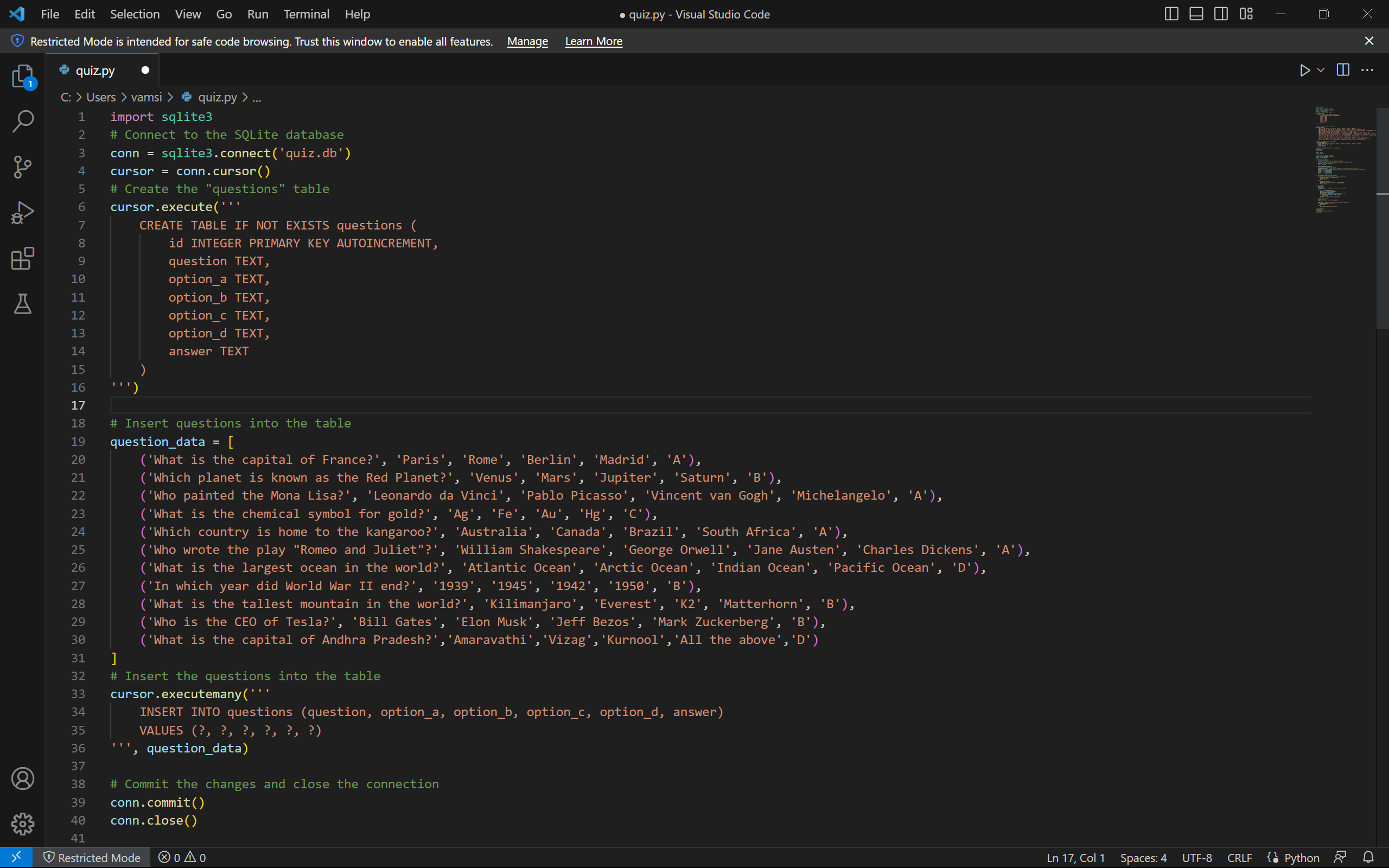
Step 2: Write Python code to connect to the database and retrieve questions.

Step 3: Develop the game logic to display questions and accept user input.

Step 4: Calculate and track the player's score based on their answers.

Step 5: Provide immediate feedback on each question.

Step 6: Display the final score and offer the option to play again.

**SAMPLE CODE SNIPPET**

**CONCLUSION**

The Quiz Game project successfully implements a console-based quiz game using Python. It allows players to answer multiple-choice questions, calculates their score, and provides immediate feedback. The game can be expanded by adding more questions to the database and enhancing the user interface. This project enhances programming skills and knowledge of database management with Python.

By implementing this project, users can enjoy playing an engaging quiz game while also improving their Python programming skills.