

# MINI PROJECT

**CONSULTANT PROFESSOR**

**[Senthil kumar ]**

**TITLE OF PROJECT :** Quiz application

**REGISTER NUMBERs :** 192211385

192211469

192211393

**NAMEs :** P.vijay kumar

S.Devendra

**K.Manjunath Naidu**

**[Senthil Kumar]**

## 

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **1** | Abstract | **1** |
| **2** | Introduction | **2** |
| **3** | About Python | **3** |
| **4** | Flow Chart | **4** |
| **5** | Code | **5-6** |
| **6** | Output Screen | **7** |
| **7** | Conclusion | **8** |

**ABSTARACT**

The "Quiz Application" is a project designed to test users' knowledge on various topics through a series of multiple-choice questions. The primary goals of this project were to create an engaging and interactive platform for users to assess their understanding of diverse subjects. The application includes questions from fields such as chemistry, biology, physics, and general knowledge.

The project was carried out by implementing a structured code that presents users with a set of questions, each accompanied by multiple answer options. Users are prompted to select the correct answer, and their responses are evaluated for correctness. The application covers topics such as the periodic table, animal biology, atmospheric composition, human anatomy, and planetary science.

The outcomes of the project include a functional quiz application that provides immediate feedback to users based on their responses. The scoring system calculates the percentage of correct answers, allowing users to gauge their overall performance. The application aims to be user-friendly, encouraging individuals to test their knowledge in an enjoyable and educational manner.

The achievements of the project lie in the successful implementation of an interactive quiz platform, offering users an engaging experience while promoting learning. The code structure allows for easy expansion, enabling the addition of more questions and topics in the future. The Quiz Application serves as a valuable tool for self-assessment and knowledge enhancement across a diverse range of subjects.

STUDENT SIGNATURE

[K,Manjunath Naidu , P.Vijay kumar , S.Devendra]

|  |
| --- |
|  |

**INTRODUCTION**

The "Quiz Application" serves as a dynamic and interactive platform designed for individuals and teams to engage in a stimulating game of knowledge and skill. Quizzes, renowned for their entertaining and educational qualities, have long been integral to fostering learning and self-improvement. Rooted in the essence of competition, quizzes propel participants to answer a series of questions correctly, with the ultimate goal of achieving the highest score.

Beyond the realm of entertainment, quizzes have found a significant place in education, providing a structured approach to enhancing knowledge, abilities, and skills. The "Quiz Application" seeks to harness the power of this popular format, offering users an opportunity to test their understanding across diverse subjects. The user-friendly interface encourages active participation, making the learning experience both enjoyable and informative.

Scores, a fundamental aspect of quizzes, provide a quantifiable measure of participants' performance. The competitive edge adds excitement as individuals or teams strive to outscore their counterparts. Whether used for friendly competition or educational purposes, the "Quiz Application" aims to create an engaging and enriching environment, fostering continuous learning and personal growth.

**ABOUT PYTHON:**

1. **Ease of Learning and Readability:**

Python is known for its clean and readable syntax, making it easy for developers to understand and write code. This characteristic is particularly beneficial for projects like a quiz application, where clarity and simplicity are essential for efficient development.

1. **Rapid Development:**

Python is renowned for its emphasis on code readability and its concise syntax. These attributes contribute to faster development cycles, allowing developers to create functional applications with fewer lines of code. For a relatively straightforward project like a quiz application, this rapid development capability is advantageous.

1. **Extensive Libraries and Frameworks:**

Python boasts a rich ecosystem of libraries and frameworks that simplify the development process. In the case of the quiz application, the availability of libraries for user input, data manipulation, and user interface development (if applicable) can significantly streamline the implementation.

1. **Cross-Platform Compatibility:**

Python is platform-independent, meaning that code written in Python can run on various operating systems without modification. This cross-platform compatibility ensures that the quiz application can be easily deployed on different devices, enhancing its accessibility.

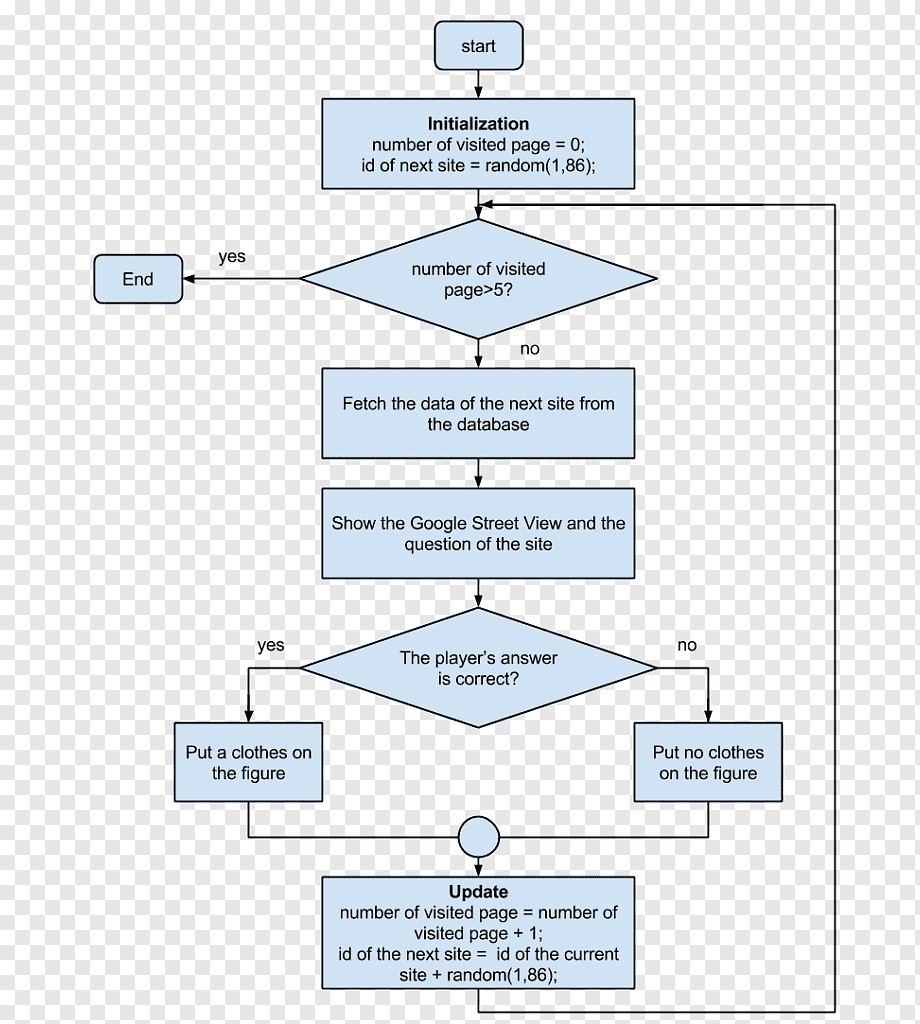
1. **Community Support:**

Python has a large and active community of developers. This community support translates to an abundance of resources, tutorials, and solutions to common programming challenges. For developers working on a quiz application, this support can be invaluable in troubleshooting and optimizing the code.

**6. Versatility:**

Python is a versatile language suitable for various types of projects, from web development to data science. This versatility provides flexibility for future expansions or modifications to the quiz application, allowing for easy integration of new features or improvements.

|  |
| --- |
| FLOW CHART |



**Quiz application (source code)**

questions = (

"How many elements are in the periodic table?: ",

"Which animal lays the largest eggs?: ",

"What is the most abundant gas in Earth's atmosphere?: ",

"How many bones are in the human body?: ",

"Which planet in the solar system is the hottest?: ",

"Who was the first Emperor of the Maurya Dynasty in ancient India?",

"What is the capital of France?: ",

"Which scientist developed the theory of general relativity?: ",

"In which year did World War II end?: ",

"Which planet is known as the 'Red Planet'?: ",

"What is the largest mammal in the world?: ",

"Which country is known as the 'Land of the Rising Sun'?: ",

"Who wrote 'Romeo and Juliet'?: ",

"What is the currency of Japan?: ",

"In which year did Christopher Columbus first reach the Americas?: "

)

options = (

("A. 116", "B. 117", "C. 118", "D. 119"),

("A. Whale", "B. Crocodile", "C. Elephant", "D. Ostrich"),

("A. Nitrogen", "B. Oxygen", "C. Carbon-Dioxide", "D. Hydrogen"),

("A. 206", "B. 207", "C. 208", "D. 209"),

("A. Mercury", "B. Venus", "C. Earth", "D. Mars"),

("A. Chandragupta Maurya", "B. Ashoka", "C. Bindusara", "D. Samudragupta"),

("A. Rome", "B. Berlin", "C. Paris", "D. London"),

("A. Isaac Newton", "B. Albert Einstein", "C. Galileo Galilei", "D. Marie Curie"),

("A. 1943", "B. 1944", "C. 1945", "D. 1946"),

("A. Jupiter", "B. Mars", "C. Venus", "D. Saturn"),

("A. Elephant", "B. Blue Whale", "C. Giraffe", "D. Polar Bear"),

("A. China", "B. South Korea", "C. Japan", "D. Thailand"),

("A. Charles Dickens", "B. William Shakespeare", "C. Jane Austen", "D. Mark Twain"),

("A. Dollar", "B. Euro", "C. Yen", "D. Rupee"),

("A. 1492", "B. 1510", "C. 1601", "D. 1620")

)

answers = (

"C", "D", "A", "A", "B", "A", "C", "B", "C",

"B", "B", "C", "B", "C", "A"

)

guesses = [] score = 0

question\_num = 0

for question in questions: print("----------------------")

print(question) for option in options[question\_num]:

print(option)

guess = input("Enter (A, B, C, D): ").upper() guesses.append(guess) if guess == answers[question\_num]:

score += 1 print("CORRECT!") else:

print("INCORRECT!")

print(f"{answers[question\_num]} is the correct answer") question\_num += 1

print("----------------------") print(" RESULTS ")

print("----------------------")

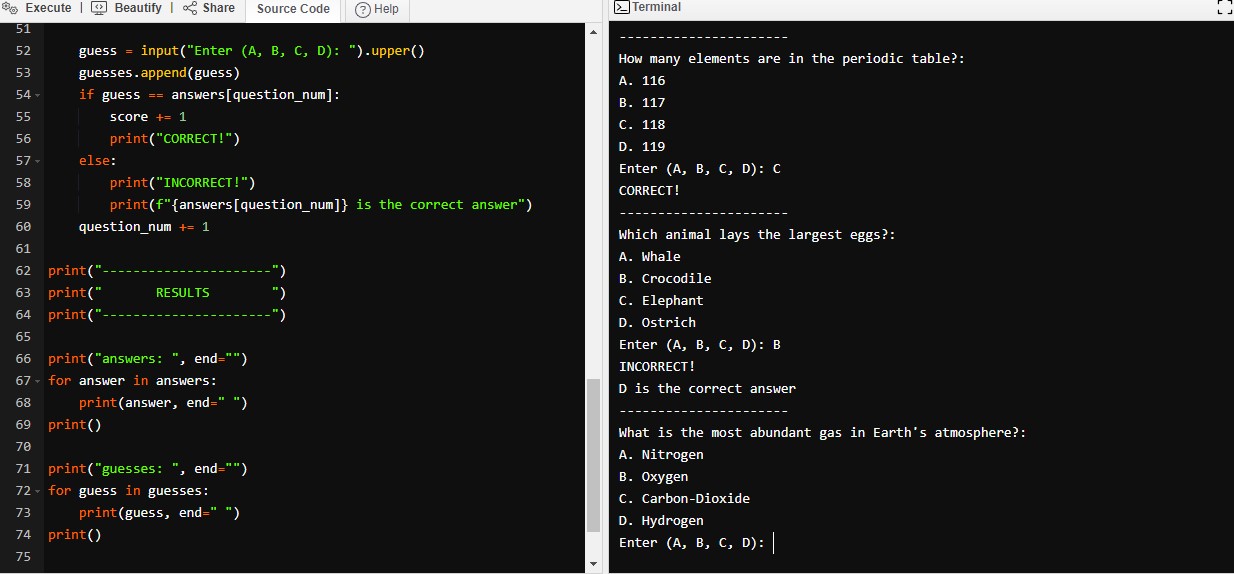
print("answers: ", end="") for answer in answers: print(answer, end=" ")

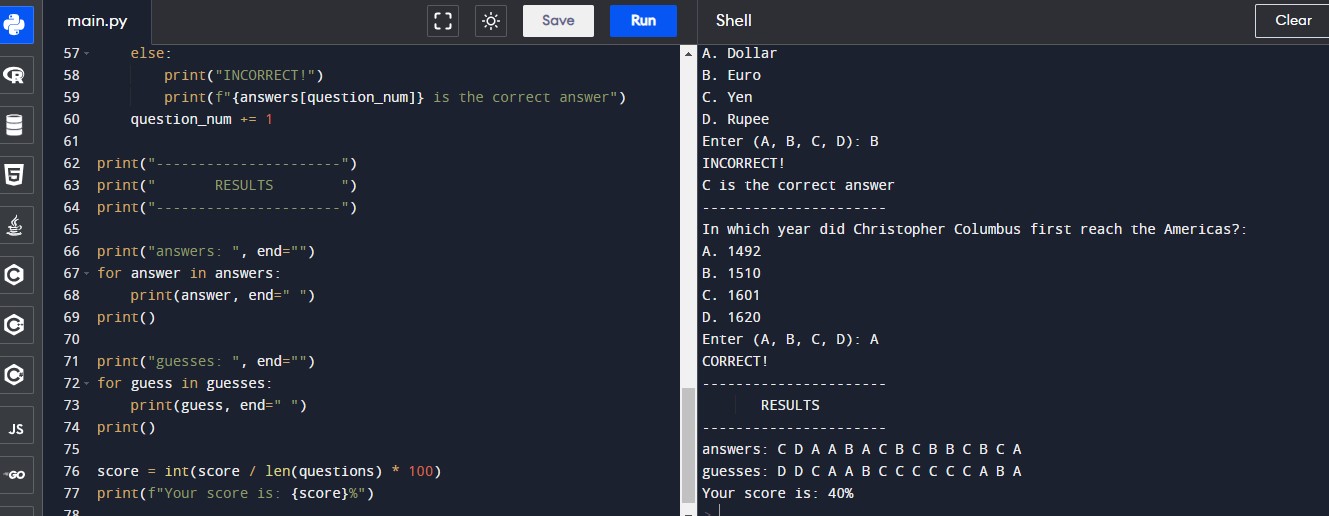
print()

print("guesses: ", end="") for guess in guesses: print(guess, end=" ") print()

score = int(score / len(questions) \* 100) print(f"Your score is: {score}%")

**OUTPUT SCREEN**:





**CONCLUSION:**

Quiz app system report in python. Thesis, we have focused on the automated system, which replaces the manual quiz system. But the interesting thing is that the thesis is not just a quiz system; it has its own intelligent capability. This was actually our target feature of our thesis. Quiz Applications Report in Python.We have tried to introduce at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by the design. We prefer the Top-down integration testing as a testing approach for our project. The Top-down integration testing is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module. The main control module is used as a test driver and others are substituted for all components directly subordinate to the main control module. Depth-first integration would integrate all components on a major control path of a structure. Selection of a major path is somewhat arbitrary and depends on applicationspecific characteristics.