

A MINI PROJECT: BUILD A PYTHON APPLICATION ON

“A MATHS QUIZZER”

Submitted By –

Name –

SANOBAR SHAIKH – 25030421039

SRUSHTI LAMBDADADE – 25030421021

PRAJAKTA SURYAVANSHI – 25030421029

RIYA PILLAI – 25030421031

SUPERVISED BY
DR. DAWA CHYOPHEL LEPCHA
ASSISTANT PROFESSOR

Symbiosis Artificial Intelligence Institute (SAII)
Symbiosis International (Deemed University)

Lavale, Mulshi, Pune-411211, Maharashtra

3rd NOVEMBER 2025



SAIL (साई)

SYMBIOSIS ARTIFICIAL INTELLIGENCE INSTITUTE

Symbiosis Artificial Intelligence Institute (SAIL)

Symbiosis International (Deemed University)

Lavale, Mulshi, Pune-411211, Maharashtra

3rd NOVEMBER 2025

CERTIFICATE

This is to certify that the project work entitled “**THE MIND MATHS QUIZZER**” that is

submitted by –

SANOBAR SHAIKH – 25030421039

SRUSHTI LAMBDADADE – 25030421021

PRAJAKTA SURYAVANSHI – 25030421029

RIYA PILLAI – 25030421031

The results embodied in this report have been verified and found satisfactory.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without mentioning the people who made it possible and whose encouragement and guidance have crowned our efforts with success.

We would like to express our sincere gratitude and indebtedness to our project supervisor, **Dr. Dawa Chyophel Lepcha, Assistant Professor** at Symbiosis Artificial Intelligence Institute (SAII), for his invaluable support, patience, and guidance throughout our project. His insights and feedback were instrumental in shaping our application.

We also extend our gratitude to the other faculty and staff of the **Symbiosis Artificial Intelligence Institute (SAII)** for providing us with the necessary facilities and a conducive environment for our work.

Finally, we express thanks to our sir and friends for their moral support and encouragement during this project.

SANOBAR SHAIKH – 25030421039

SRUSHTI LAMBDARE – 25030421021

PRAJAKTA SURYAVANSHI – 25030421029

RIYA PILLAI – 25030421031

DECLARATION

We SANOBAR.S, SRUSHTI.L, PRAJAKTA.S, RIYA.P, here by declare that the report of the Mini Project work entitled “THE MIND MATHS QUIZZER “which is being submitted to the” **Symbiosis International (Deemed University) PUNE**”, in partial fulfillment of the requirement for the award of Degree of BACHELOR OF ARTIFICIAL Intelligence, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any University or Institution for the award of any degree.

Place: SIU,PUNE

Date:3rd NOVEMBER 2025

SANOBAR SHAIKH – 25030421039

SRUSHTI LAMBDADADE – 25030421021

PRAJAKTA SURYAVANSHI – 25030421029

RIYA PILLAI – 25030421031

ABSTRACT

The concept of using gamification in education to improve learning outcomes and user engagement is a well-established and intuitive approach. During the last several years, various programming languages have been introduced to create interactive learning tools. In this project, Python is used to develop an interactive "Mind Math Quizzer" application, with its user interface built using the Streamlit library. The application was designed to generate random arithmetic problems across four operations (addition, subtraction, multiplication, and division) and provide instant feedback to the user. The compressive, or rather, comparative logic was performed by checking the user's input against the correct, stored answer. As a result, it was found that this interactive format provides an engaging and effective way to practice mental math skills, with a clear system for tracking the user's score over a 10-question quiz.

A successful user experience was critical for this project. This was achieved by using Streamlit's simple and responsive UI components. We also implemented Streamlit's session state to manage the application's "memory." This was essential for tracking the user's score, the current question number, and the correct answer, ensuring the app state persists across user interactions and page reruns. The app was also configured to run on a "light" theme by default to ensure a clear and accessible user interface for all.

Key Words: Python, Streamlit, Quiz Application, Session State, User Interface (UI), Gamified Learning, Random Module

INTRODUCTION

The Mini Project titled “MATHS QUIZZER” is a simple python based educational application designed to help user practise basic mathematical operations interactively. The project demonstrate that how the programming logic can be applied learning in fun and easier way of engaging learning tool.

Mathematics is an essential skill that strengthens logical and analytical thinking. Through this project user can test their understanding of operations like addition, subtraction, multiplication and division in a quiz format. The quiz dynamically generate the random questions and evaluates user performances based on the answers given.

Python was chosen for this project because its simplicity ,readability ,and easy to understand. The project reflects a practical application of Python fundamentals such as loops, conditions, and user input handling.

OBJECTIVE

The main objective of this mini-project is to design and implement an interactive quiz application using Python that can test mathematical skills and provide instant feedback.

➤ **SPECIFIC OBJECTIVES INCLUDE:**

- To generate random arithmetic questions using Python's built-in libraries.
- To evaluate user responses and maintain a score counter.
- To create a simple and user-friendly interface for learning.
- To apply programming concepts such as control structures, loops, and functions in a real-world project.
- To encourage logical thinking and consistent practice through gamified learning.

METHODOLOGY

The development process of the project follows a step-by-step approach:

➤ Step 1: Problem Understanding

The goal was to create a short quiz that presents users with random math questions and checks the accuracy of their answers.

➤ Step 2: Planning

A flow diagram was prepared to visualize the working of the program. It helped in organizing the structure of question generation, score counting, and result display.

➤ Step 3: Implementation

Python was used to write the code. The random module was applied to generate random numbers for arithmetic operations. A simple loop structure allowed multiple questions to be asked in one quiz session.

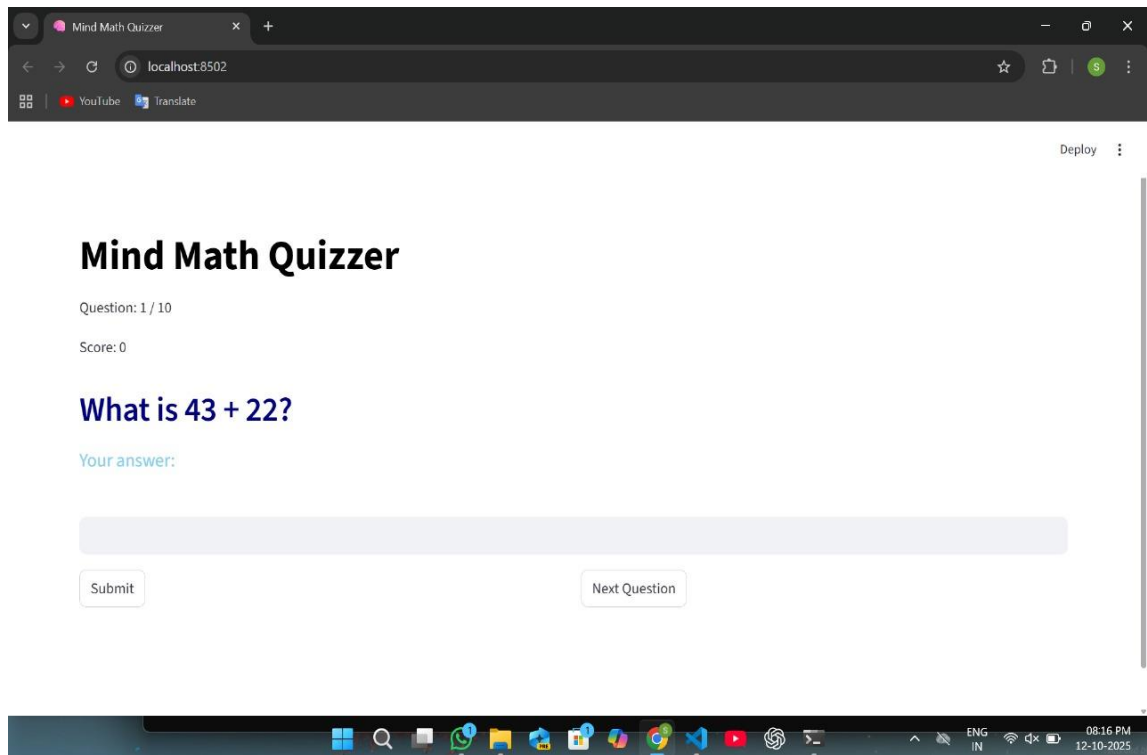
➤ Step 4: Testing

The program was tested for different inputs to ensure correct scoring and error-free operation.

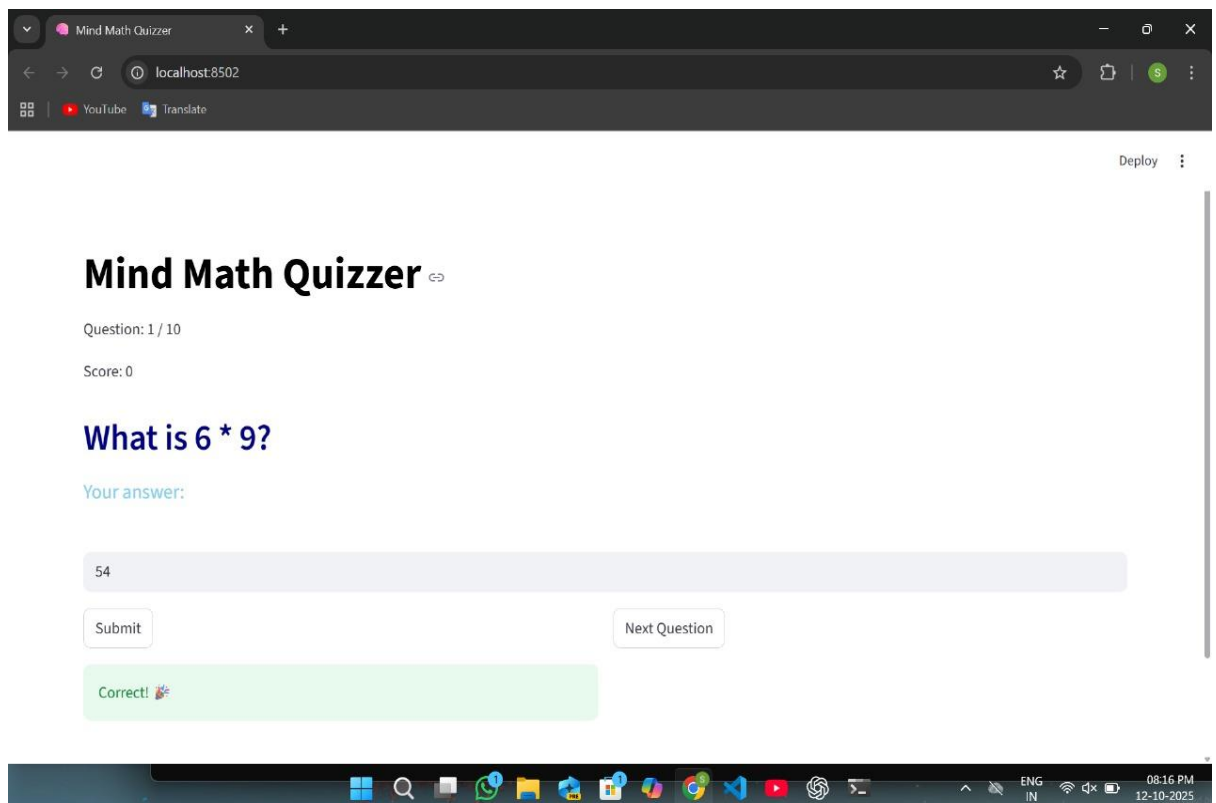
FLOW OF THE APPLICATION

1. Start the program
2. Display a welcome message.
3. Generate two random numbers and an arithmetic operation.
4. Take user input (answer).
5. Compare the user's answer with the correct result.
6. Update and display the score.
7. End the quiz and show the final score.

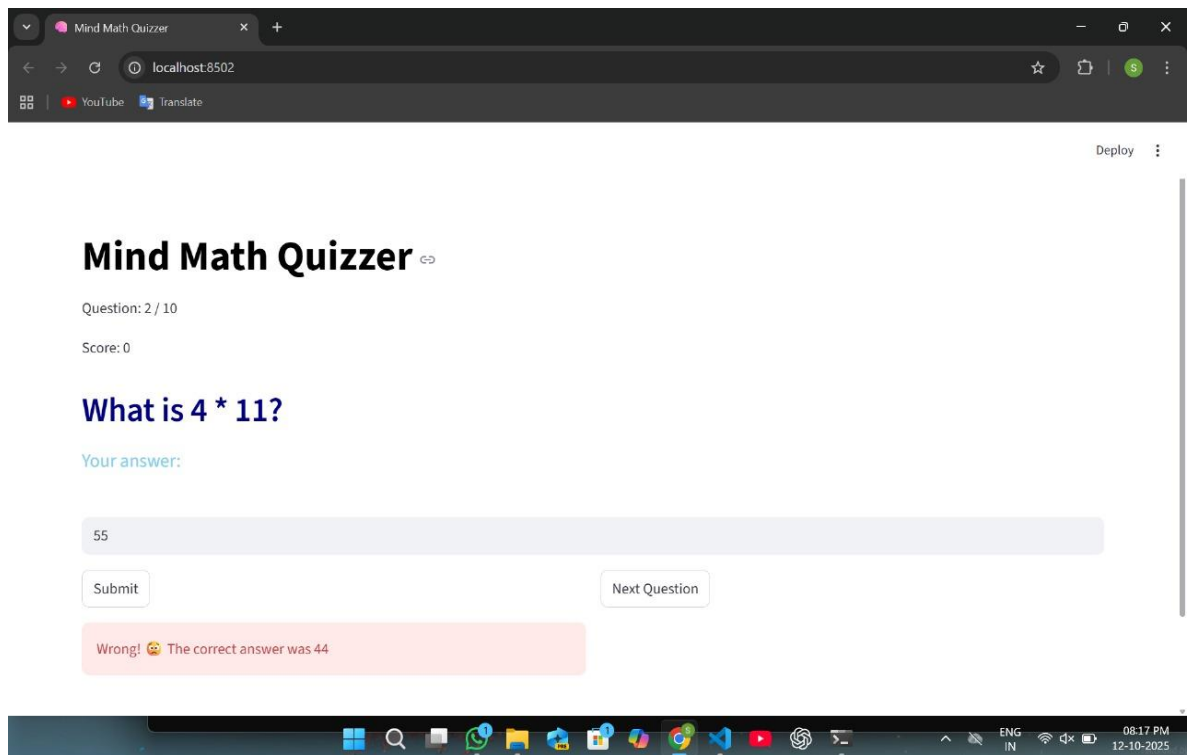
❖ The Start Screen



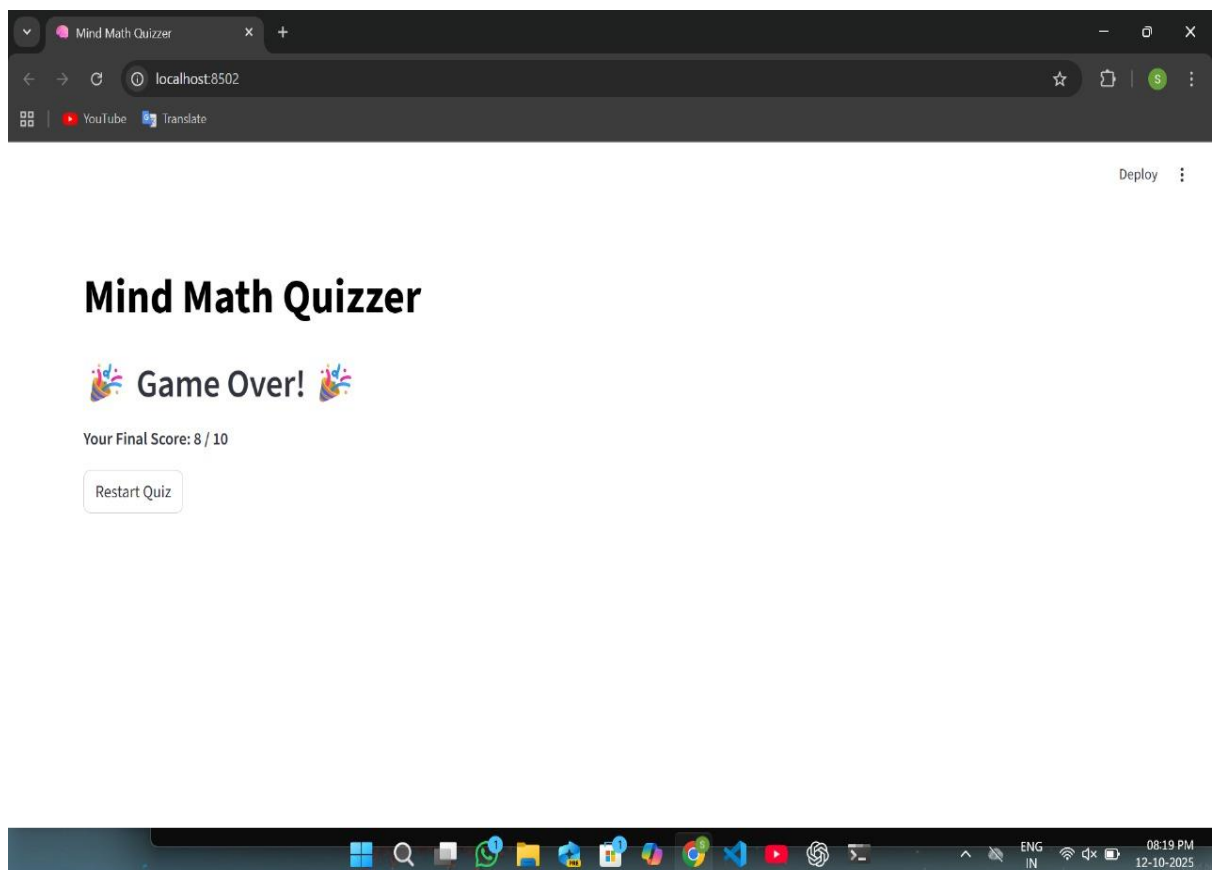
❖ The Correct Answer



❖ The Wrong Answer



❖ The Game Over



TOOLS AND TECHNOLOGIES USED

| <u>TOOLS</u> | <u>PURPOSE</u> |
|-------------------------------|---|
| Python 3.x | core programming language used for development |
| Random module | Generates random arithmetic questions |
| Input/Output Function | Takes user responses and shows output |
| Loops & Conditions | Check correctness of answers and repeat questions |
| VS Code | Development and testing environment |

IMPLEMENTATION

The project implementation involved writing Python code that can perform all basic arithmetic operations while interacting with the user.

Below is a simplified code snippet that demonstrates the working of the Maths Quizzer:

```
quiz_app.py 1 X
> Users > Srushti > OneDrive > quiz_app_folder > quiz_app.py > ...
1  import streamlit as st
2  import random
3
4  # --- PAGE CONFIG ---
5  st.set_page_config(
6      page_title="Mind Math Quizzer",
7      page_icon="🧠",
8      layout="wide"
9  )
10
11 def generate_question():
12     """Generates a new random math problem (+, -, *, /)."""
13     ops = ['+', '-', '*', '/']
14     op = random.choice(ops)
15
16     if op == '+':
17         num1 = random.randint(10, 100)
18         num2 = random.randint(10, 100)
19         question = f"What is {num1} + {num2}?"
20         correct_answer = num1 + num2
21     elif op == '-':
22         num1 = random.randint(20, 100)
23         num2 = random.randint(10, num1)
24         question = f"What is {num1} - {num2}?"
25         correct_answer = num1 - num2
26     elif op == '*':
27         num1 = random.randint(2, 12)
28         num2 = random.randint(2, 12)
29         question = f"What is {num1} * {num2}?"
30         correct_answer = num1 * num2
31     elif op == '/':
32         divisor = random.randint(2, 10)
```

```

quiz_app.py 1 X
C: > Users > Srushti > OneDrive > quiz_app_folder > quiz_app.py > ...
11 def generate_question():
31     elif op == '/':
32         divisor = random.randint(2, 10)
33         answer = random.randint(2, 10)
34         dividend = divisor * answer
35         question = f"What is {dividend} / {divisor}?"
36         correct_answer = answer
37
38     return question, correct_answer
39
40 def on_next_question():
41     """Callback function to handle 'Next Question' button click."""
42     st.session_state.question_number += 1
43     st.session_state.question, st.session_state.answer = generate_question()
44     if 'user_input' in st.session_state:
45         st.session_state.user_input = ""
46
47 def restart_game():
48     """Callback function to reset the game state."""
49     st.session_state.score = 0
50     st.session_state.question_number = 1
51     # Generate the first question directly
52     st.session_state.question, st.session_state.answer = generate_question()
53     if 'user_input' in st.session_state:
54         st.session_state.user_input = ""
55
56
57 if 'score' not in st.session_state:
58     restart_game()
59
60
61 st.markdown('<h1 style="color: black;">Mind Math Quizzer</h1>', unsafe_allow_html=True)

```

```

quiz_app.py 1 X
C: > Users > Srushti > OneDrive > quiz_app_folder > quiz_app.py > ...
63
64 if st.session_state.question_number <= 10:
65     st.write(f"Question: {st.session_state.question_number} / 10")
66     st.write(f"Score: {st.session_state.score}")
67     st.markdown(f'<h2 style="color: navy;">{st.session_state.question}</h2>', unsafe_allow_html=True)
68     st.markdown('<p style="font-size: 20px; color: skyblue;">Your answer:</p>', unsafe_allow_html=True)
69     user_answer = st.text_input("Your answer:", label_visibility="hidden", key="user_input")
70
71     col1, col2 = st.columns(2)
72
73     with col1:
74         if st.button("Submit"):
75             try:
76                 user_answer_int = int(user_answer)
77                 if user_answer_int == st.session_state.answer:
78                     st.success("Correct! 🎉")
79                     st.session_state.score += 1
80             except:
81                 st.error(f"Wrong! 😞 The correct answer was {st.session_state.answer}")
82             except ValueError:
83                 st.warning("Please enter a valid number.")
84
85     with col2:
86         st.button("Next Question", on_click=on_next_question)
87
88 else:
89     # --- GAME OVER SCREEN ---
90     st.balloons()
91     st.header("🎉 Game Over! 🎉")
92     st.write(f"**Your Final Score: {st.session_state.score} / 10**")
93     st.button("Restart Quiz", on_click=restart_game)
94

```

RESULT AND DISCUSSION

After successful execution, the program displays five random questions and evaluates user answers instantly. The final score is printed at the end of the quiz.

The project demonstrates the use of Python for creating small interactive applications that combine fun and learning. Users can easily run this code in a terminal or notebook and get immediate feedback on their math skills.

❖ This project can also be expanded to include:

- A timer to make the quiz more challenging.
- A scoring leaderboard.
- More Graphic Representation
- A Sound or Visual Effect

CONCLUSION AND LIMITATION

The “Maths Quizzer” mini project successfully gives its objective of building an interactive Python-based application that tests mathematical abilities in a quiz format.

It helped in understanding key programming concepts such as loops, functions, and conditionals. The project also enhanced problem-solving and logical thinking skills.

❖ LIMITATIONS:

- Only limited to basic arithmetic operations.
- Lacks user data storage or performance tracking.
- Lack of Modular Design

THANK YOU!!