

A Project Report On

EduMatrix

Submitted in partial fulfillment of the requirement for the
award of the degree

Bachelor of Computer Application
BCA

Academic Year 2025 - 26

Jenish Gohel
92300527156

Yuvrajsinh Zala
9300527141

Ravi Sav
92300527257

Internal Guide
Prof. Dipak Thanki



Marwadi
University
Marwadi Chandarana Group





Marwadi
University
Marwadi Chandarana Group



Faculty of Computer Applications (FCA)

Certificate

This is to certify that the project work entitled
EduMatrix
submitted in partial fulfillment of the requirement for
the award of the degree of
Bachelor of Computer Application
BCA
of the
Marwadi University
is a result of the bonafide work carried out by

Jenish Gohel (92300527156)
Yuvrajsinh Zala (92300527141)
Ravi Sav (92300527257)

during the academic year 2025-26

Prof. Dipak Thanki
Faculty Guide

Dr. Sunil Bajaja
HOD

Dr. R. Sridaran
Dean

DECLARATION

We hereby declare that this project work entitled **EduMatrix** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place : Marwadi University , Rajkot

Date : 6th June, 2025

Jenish Gohel (92300527156)

Signature : Jenish Gohel

Yuvrajsinh Zala (92300527141)

Signature : Yuvrajsinh zala

Ravi sav (92300527257)

Signature : Ravi Sav

CONTENTS

Chapters	Particulars	Page No.
1	SYNOPSIS	5
2	PREAMBLE General Introduction Module description	6
3	TECHNICAL DESCRIPTION Hardware Requirement Software Requirement	7
4	SYSTEM DESIGN AND DEVELOPMENT <ul style="list-style-type: none"> • Flowchart • Activity Diagram • Use Case Diagram • Database Design • Screen Design & Coding 	 <div style="text-align: right;"> 8 9 10 11 13 </div>
5	CONCLUSION	23
6	LEARNING DURING SIP	24
7	BIBLIOGRAPHY Online References Offline References	25

1. SYNOPSIS

- **EduMatrix** is an all-in-one interactive desktop application designed to help students and learners master essential programming concepts through quizzes, games, and instant reference tools. Built with Python and Tkinter, EduMatrix provides a seamless experience for practicing multiple-choice questions (MCQs), playing subject-oriented educational games, and accessing a rich language tool that includes syntax references, keyword explanations, error guides, and reusable code snippets.
- EduMatrix supports the following core programming disciplines:
 - Python
 - Java
 - C
 - Web Technologies
 - Operating System Concepts
- With a focus on engagement and effective learning, EduMatrix combines self-testing, game-based exercises, and handy code resources under a single, user-friendly platform. All progress and results are stored locally, ensuring privacy and offline accessibility.

2.PREAMBLE

2.1 General Introduction:

- **EduMatrix** is a desktop application developed in Python using Tkinter, designed to help students learn programming concepts interactively. It supports five key subjects: Python, Java, C, Web Technologies, and Operating Systems. EduMatrix combines quizzes, educational games, and a language reference tool in a single offline platform. Quiz history will be saved for Every time attempt of Quiz.

2.2 Module Description:

- EduMatrix is organized into four main modules, each serving a unique purpose within the application:

1.Quiz Module:

> The Quiz Module offers timed, randomized multiple-choice quizzes for Python, Java, C, Web, and Operating System subjects. Users receive immediate feedback and explanations for each answer, can review and change their responses before submission, and get a detailed result summary. All quiz attempts are logged for progress tracking.

2.Game Center:

> The Game Center features eight interactive games—including Flashcards, Hangman, Puzzle, Memory, Word Search, Typing Challenge, and more—tailored to reinforce programming concepts in a fun and engaging way. Each game adapts its content to the selected subject, promoting active recall and long-term retention.

3.Dashboard:

> The Dashboard provides users with a clear overview of their learning journey. It displays best scores, average performance, and a comprehensive quiz history for each subject, helping users identify strengths and areas for improvement

4.Language Tool :

> The Language Tool is a multi-tabbed reference window offering instant access to syntax guides, keyword definitions, error explanations, and a curated library of code snippets for all supported languages. It serves as a quick reference for study, revision, and practical coding assistance.

3. TECHNICAL DESCRIPTION

3.1 Hardware Requirements :

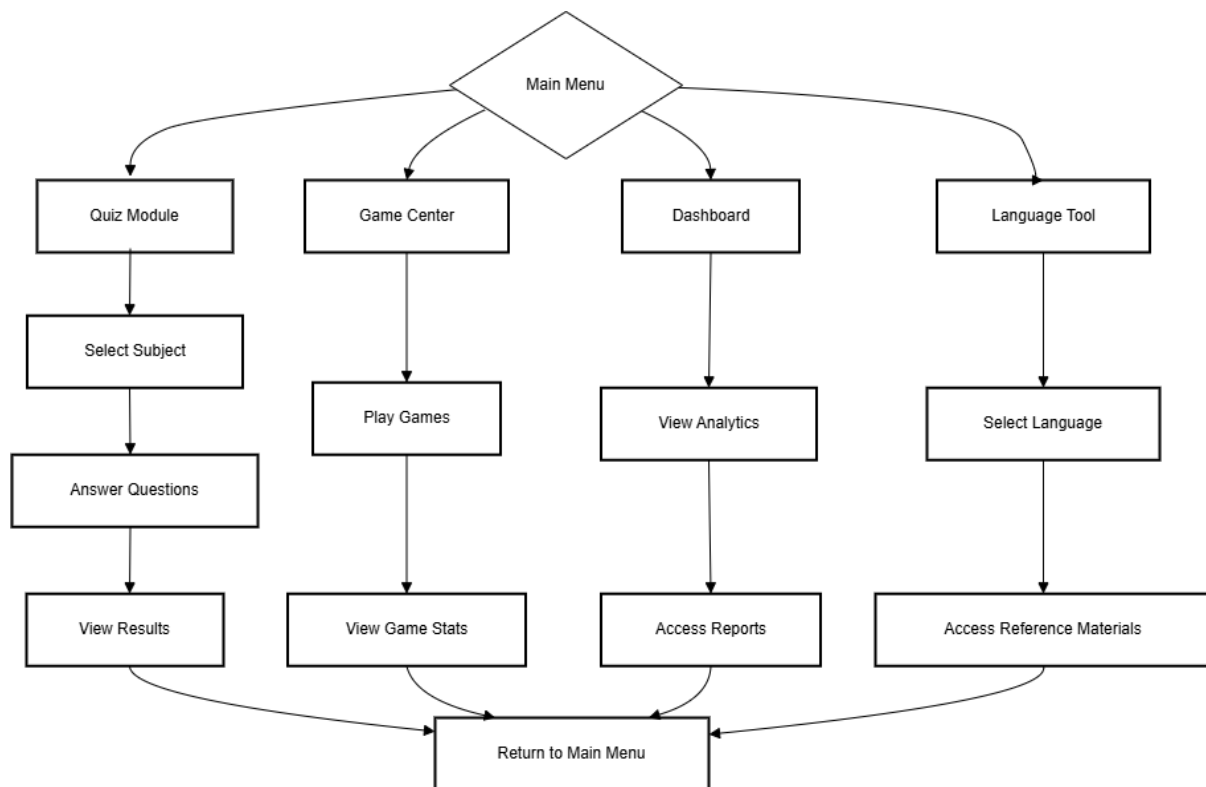
- ✓ A standard personal computer or laptop
- ✓ Minimum 2 GB RAM (4 GB or more recommended for optimal performance)
- ✓ At least 200 MB free disk space
- ✓ Display supporting at least 1024x768 resolution
- ✓ Keyboard and mouse or equivalent input devices

3.2 Software Requirements :

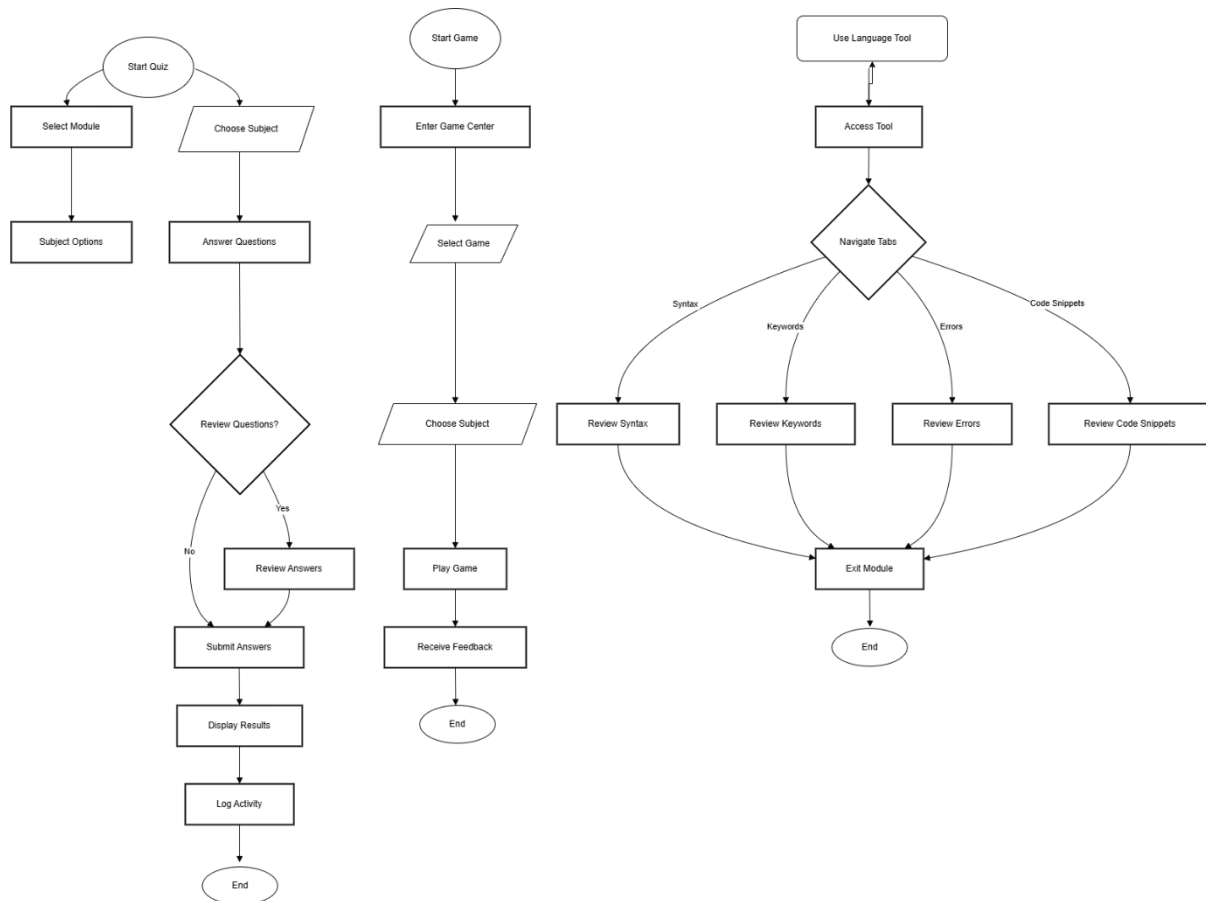
- ✓ **Operating System:** Windows, Linux, or macOS
- ✓ Python 3.x installed (version 3.7 or higher recommended)
- ✓ Tkinter library (comes pre-installed with most Python distributions)
- ✓ SQLite (included with Python standard library)
- ✓ No internet connection required after installation

4. SYSTEM DESIGN AND DEVELOPMENT

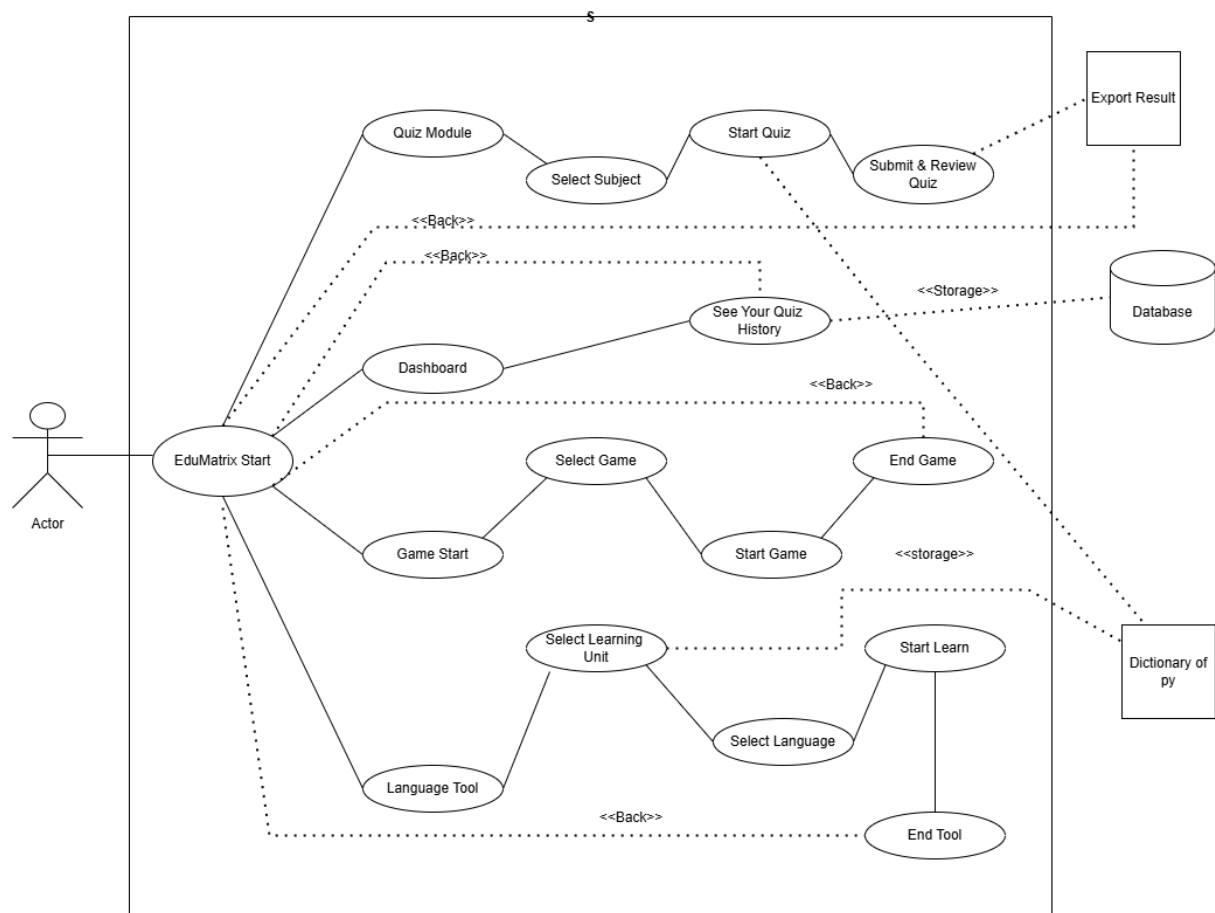
4.1 Flowchart:



4.2 Activity Diagram :



4.3 Use Case Diagram :



4.4 Database Design :

- **Database Information:**

The EduMatrix application uses an SQLite database, but this database is dedicated exclusively to the Quiz module. All other modules (such as games and reference tools) operate without persistent data storage.

- **Purpose :**

The database securely stores quiz-related data locally on the user's device. It keeps a record of each quiz attempt, enabling users to track their progress and review past results.

- **Main Table stores UserHistory :**

- Stores information about every quiz attempt.

Fields:

- `id` (primary key)
- `subject` (the topic of the quiz)
- `date` (when the quiz was taken)
- `score` (user's score)
- `time_taken` (duration of the attempt)
- `answers` (user's responses, stored as text or JSON)

- **Key Characteristics :**

- **Local Only:** Data is stored locally; no internet connection or external database is required.
- **Privacy:** All quiz history remains on the user's device.
Simplicity: Only quiz attempts are tracked—no user profiles, game scores, or other features require database storage.

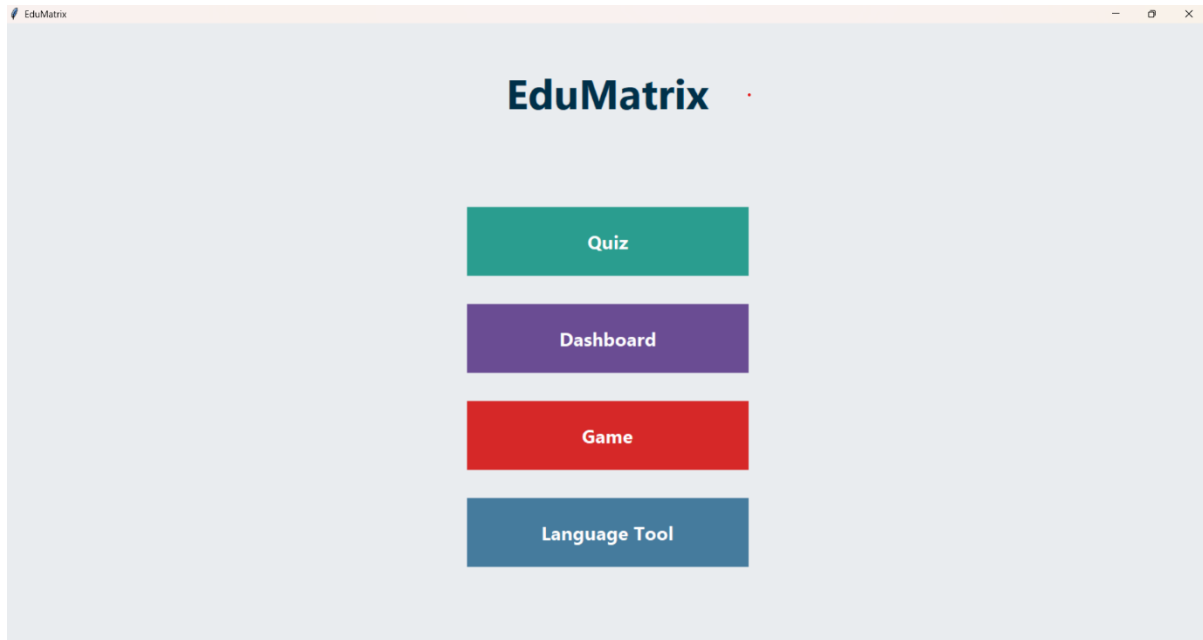
- **Example Table Schema (SQLite):**

```
```sql
CREATE TABLE UserHistory (
id INTEGER PRIMARY KEY AUTOINCREMENT,
subject TEXT NOT NULL,
date TEXT NOT NULL,
score INTEGER NOT NULL,
time_taken TEXT,
answers TEXT
);
```
```

- **Data Flow:**
 - When a user completes a quiz, their attempt is saved in the 'UserHistory' table.
 - The Dashboard or results screen can retrieve and display this data to help users monitor their learning progress.
- **Note:** No other modules of EduMatrix (such as games or language tools) use the database. All database operations are confined to the Quiz module.

• 4.5 Screen Design & Coding :

1. Main Page :



Code :

```
import tkinter as tk
from tkinter import messagebox, font
import sqlite3
from datetime import datetime

# ... (other imports and subject/question data skipped for brevity)

class EduMatrixApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("EduMatrix")
        self.state('zoomed')
        self.configure(bg="#e9ecef")
        self.custom_font = font.Font(family="Segoe UI", size=18)
        self.title_font = font.Font(family="Segoe UI", size=40, weight="bold")
        self.button_font = font.Font(family="Segoe UI", size=18, weight="bold")
        self.subjects = list(SUBJECTS.keys())
        self.protocol("WM_DELETE_WINDOW", self.on_quit)
        init_db()
        self.show_main_menu()

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

```

def show_main_menu(self):
    self.clear()
    tk.Label(self, text="EduMatrix", font=self.title_font, bg="#e9ecef", fg="#003049").pack(pady=50)
    menu = tk.Frame(self, bg="#e9ecef")
    menu.pack(pady=40)
    tk.Button(menu, text="Quiz", width=25, height=2, font=self.button_font, bg="#2a9d8f", fg="white",
              activebackground="#52b788", relief="flat", command=self.show_subject_menu).pack(pady=18)
    tk.Button(menu, text="Dashboard", width=25, height=2, font=self.button_font, bg="#6a4c93", fg="white",
              activebackground="#b983ff", relief="flat", command=self.show_dashboard).pack(pady=18)
    tk.Button(menu, text="Game", width=25, height=2, font=self.button_font, bg="#d62828", fg="white",
              activebackground="#f77f00", relief="flat", command=self.show_game_menu).pack(pady=18)
    tk.Button(menu, text="Language Tool", width=25, height=2, font=self.button_font, bg="#457b9d", fg="white",
              activebackground="#a8dadc", relief="flat", command=self.show_language_tool).pack(pady=18)

def on_quit(self):
    if messagebox.askyesno("Exit", "Do you really want to quit EduMatrix?"):
        self.destroy()

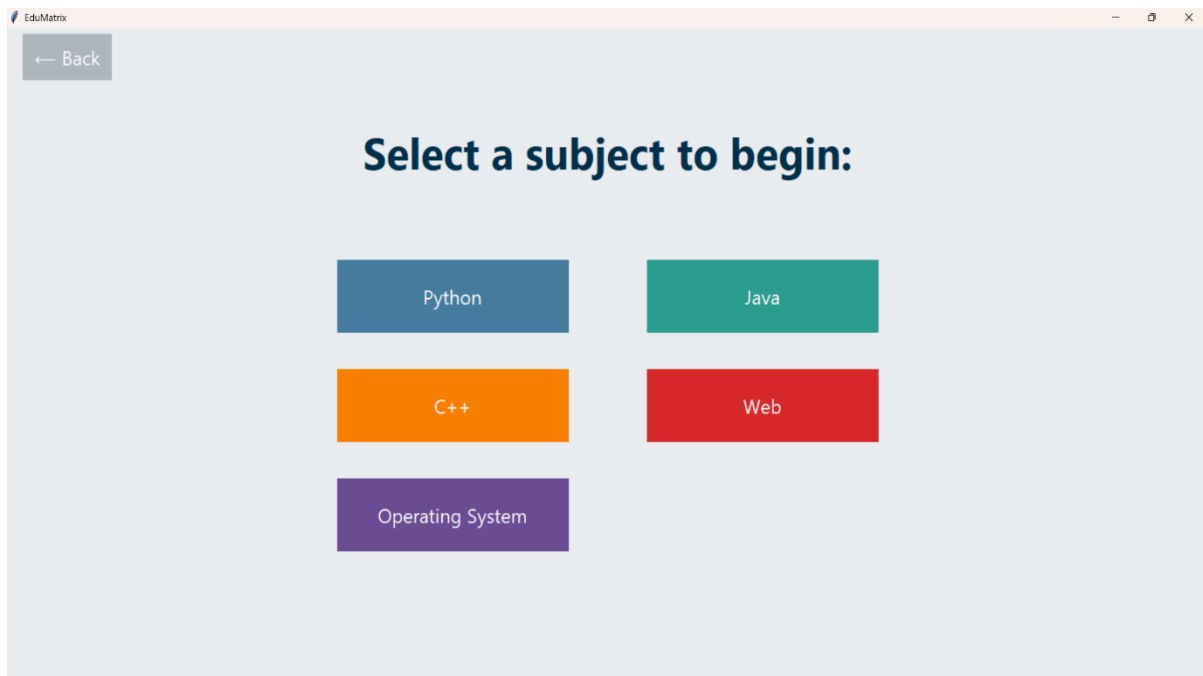
# The rest of your code for the other classes and logic...

if __name__ == "__main__":
    app = EduMatrixApp()
    app.mainloop()

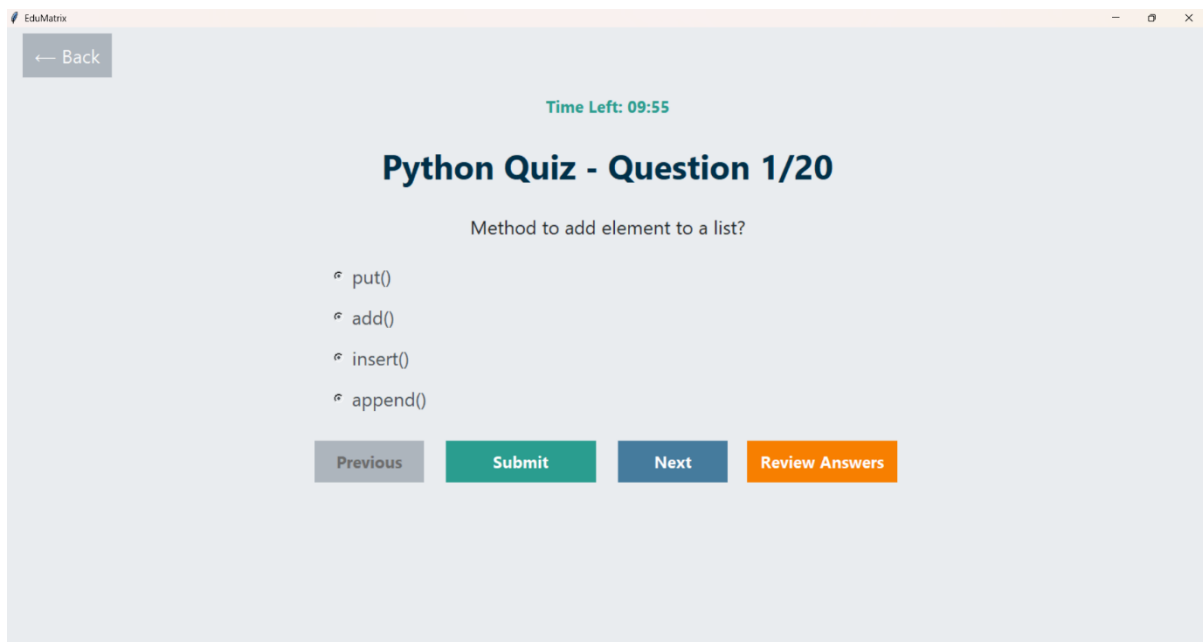
```

Note : Rest of the code will Have been submitted as edumatrix_main.py file which contains entire code file

2. Quiz Page :



2.1 Quiz Start Page :



The screenshot shows a web browser window with the EduMatrix logo in the top left corner. A 'Back' button is located in the top left of the page content. In the center, it says 'Time Left: 09:55' in green. Below that, the title 'Python Quiz - Question 1/20' is displayed in a large, bold, dark blue font. The question text 'Method to add element to a list?' is centered below the title. Four radio button options are listed: 'put()', 'add()', 'insert()', and 'append()'. At the bottom, there are four buttons: 'Previous' (grey), 'Submit' (green), 'Next' (blue), and 'Review Answers' (orange).

← Back

Time Left: 09:55

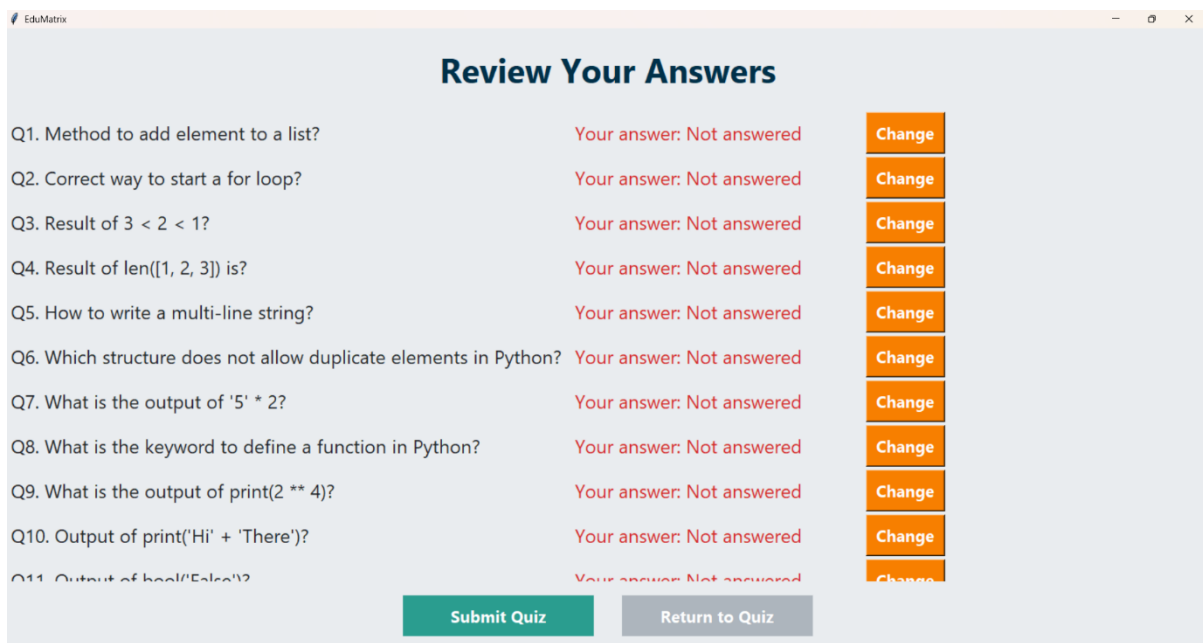
Python Quiz - Question 1/20

Method to add element to a list?

- ☐ put()
- ☐ add()
- ☐ insert()
- ☐ append()

Previous Submit Next Review Answers

2.2 Review Answer Page :



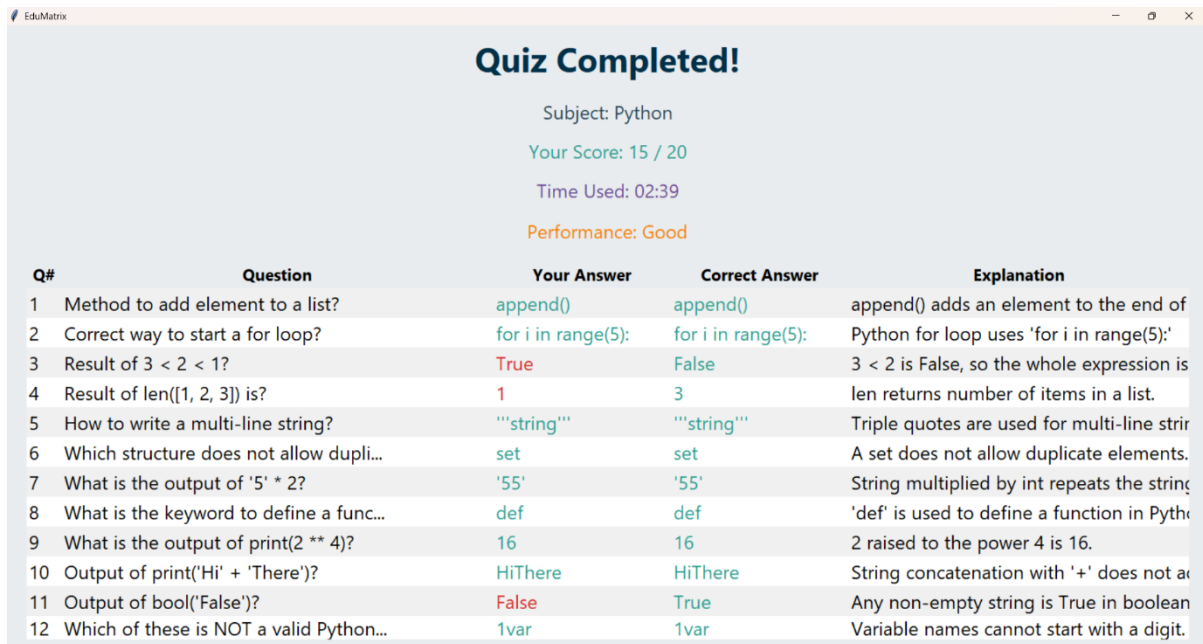
The screenshot shows a web browser window with the EduMatrix logo in the top left corner. The title 'Review Your Answers' is centered at the top in a large, bold, dark blue font. Below the title, there is a list of 11 questions (Q1 to Q11) with their corresponding answers. Each question is followed by the text 'Your answer: Not answered' in red. To the right of each question and answer, there is an orange 'Change' button. At the bottom of the page, there are two buttons: 'Submit Quiz' (green) and 'Return to Quiz' (grey).

Review Your Answers

| | | |
|--|---------------------------|--------|
| Q1. Method to add element to a list? | Your answer: Not answered | Change |
| Q2. Correct way to start a for loop? | Your answer: Not answered | Change |
| Q3. Result of $3 < 2 < 1$? | Your answer: Not answered | Change |
| Q4. Result of <code>len([1, 2, 3])</code> is? | Your answer: Not answered | Change |
| Q5. How to write a multi-line string? | Your answer: Not answered | Change |
| Q6. Which structure does not allow duplicate elements in Python? | Your answer: Not answered | Change |
| Q7. What is the output of <code>'5' * 2</code> ? | Your answer: Not answered | Change |
| Q8. What is the keyword to define a function in Python? | Your answer: Not answered | Change |
| Q9. What is the output of <code>print(2 ** 4)</code> ? | Your answer: Not answered | Change |
| Q10. Output of <code>print('Hi' + 'There')</code> ? | Your answer: Not answered | Change |
| Q11. Output of <code>bool('False')</code> ? | Your answer: Not answered | Change |

Submit Quiz Return to Quiz

2.3 Quiz Completed Page :



Quiz Completed!

Subject: Python

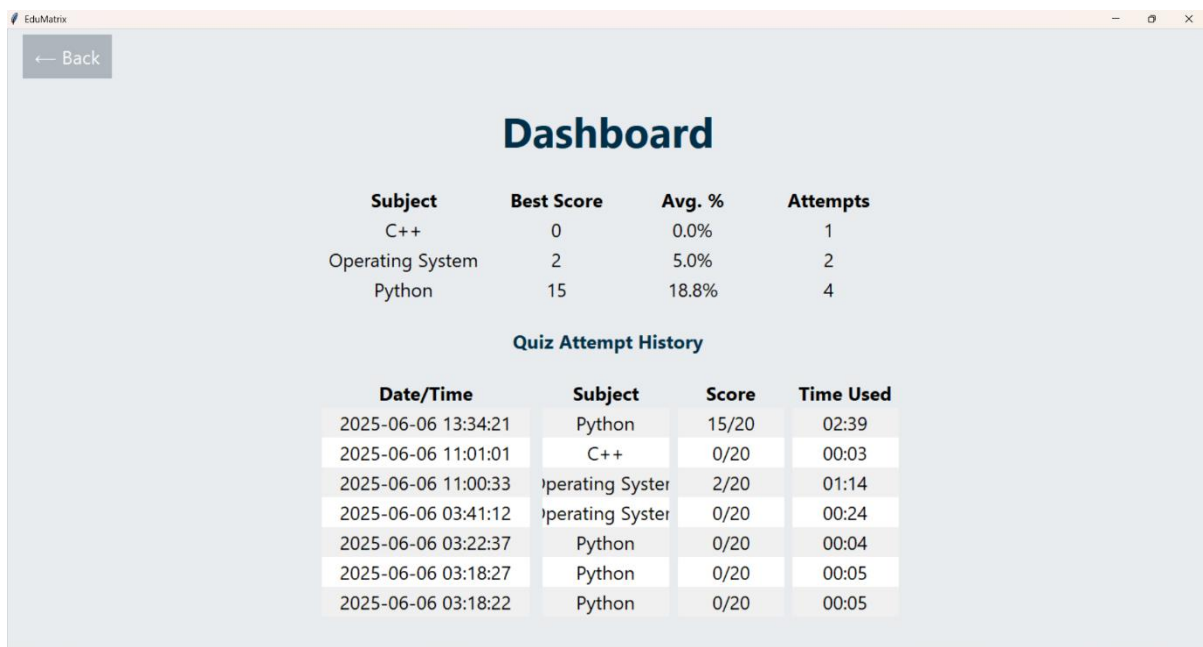
Your Score: 15 / 20

Time Used: 02:39

Performance: Good

| Q# | Question | Your Answer | Correct Answer | Explanation |
|----|---|--------------------|--------------------|---|
| 1 | Method to add element to a list? | append() | append() | append() adds an element to the end of |
| 2 | Correct way to start a for loop? | for i in range(5): | for i in range(5): | Python for loop uses 'for i in range(5):' |
| 3 | Result of 3 < 2 < 1? | True | False | 3 < 2 is False, so the whole expression is |
| 4 | Result of len([1, 2, 3]) is? | 1 | 3 | len returns number of items in a list. |
| 5 | How to write a multi-line string? | """string""" | """string""" | Triple quotes are used for multi-line strin |
| 6 | Which structure does not allow dupli... | set | set | A set does not allow duplicate elements. |
| 7 | What is the output of '5' * 2? | '55' | '55' | String multiplied by int repeats the string |
| 8 | What is the keyword to define a func... | def | def | 'def' is used to define a function in Pytho |
| 9 | What is the output of print(2 ** 4)? | 16 | 16 | 2 raised to the power 4 is 16. |
| 10 | Output of print('Hi' + 'There')? | HiThere | HiThere | String concatenation with '+' does not a |
| 11 | Output of bool('False')? | False | True | Any non-empty string is True in boolean |
| 12 | Which of these is NOT a valid Python... | 1var | 1var | Variable names cannot start with a digit. |

3. Dashboard Page :



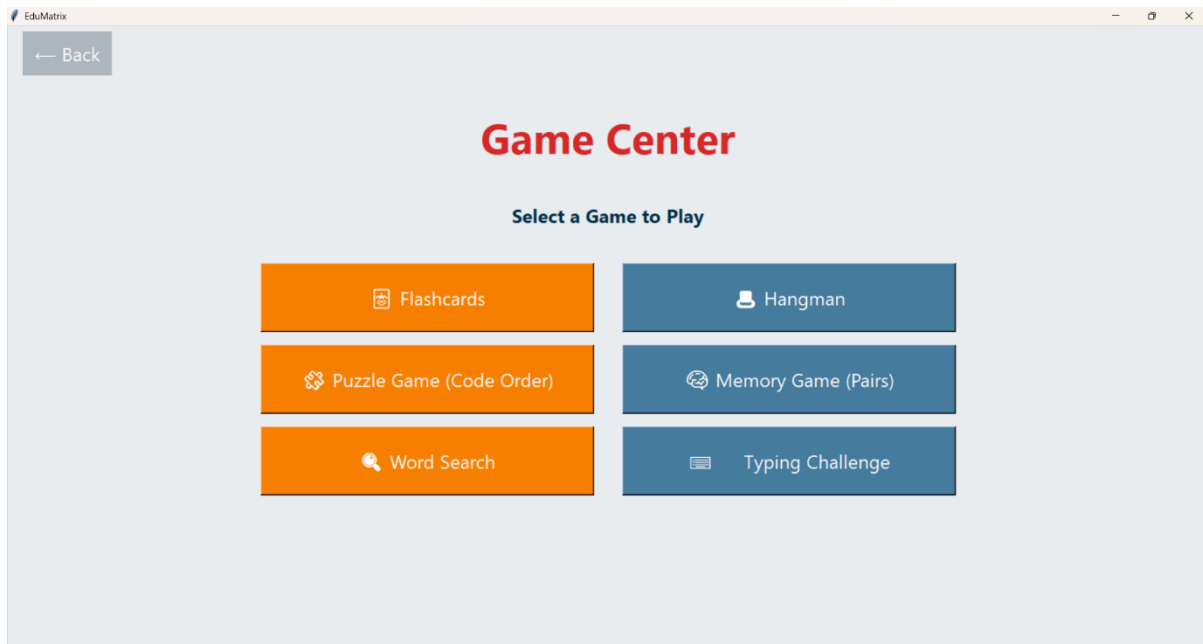
Dashboard

| Subject | Best Score | Avg. % | Attempts |
|------------------|------------|--------|----------|
| C++ | 0 | 0.0% | 1 |
| Operating System | 2 | 5.0% | 2 |
| Python | 15 | 18.8% | 4 |

Quiz Attempt History

| Date/Time | Subject | Score | Time Used |
|---------------------|------------------|-------|-----------|
| 2025-06-06 13:34:21 | Python | 15/20 | 02:39 |
| 2025-06-06 11:01:01 | C++ | 0/20 | 00:03 |
| 2025-06-06 11:00:33 | Operating System | 2/20 | 01:14 |
| 2025-06-06 03:41:12 | Operating System | 0/20 | 00:24 |
| 2025-06-06 03:22:37 | Python | 0/20 | 00:04 |
| 2025-06-06 03:18:27 | Python | 0/20 | 00:05 |
| 2025-06-06 03:18:22 | Python | 0/20 | 00:05 |

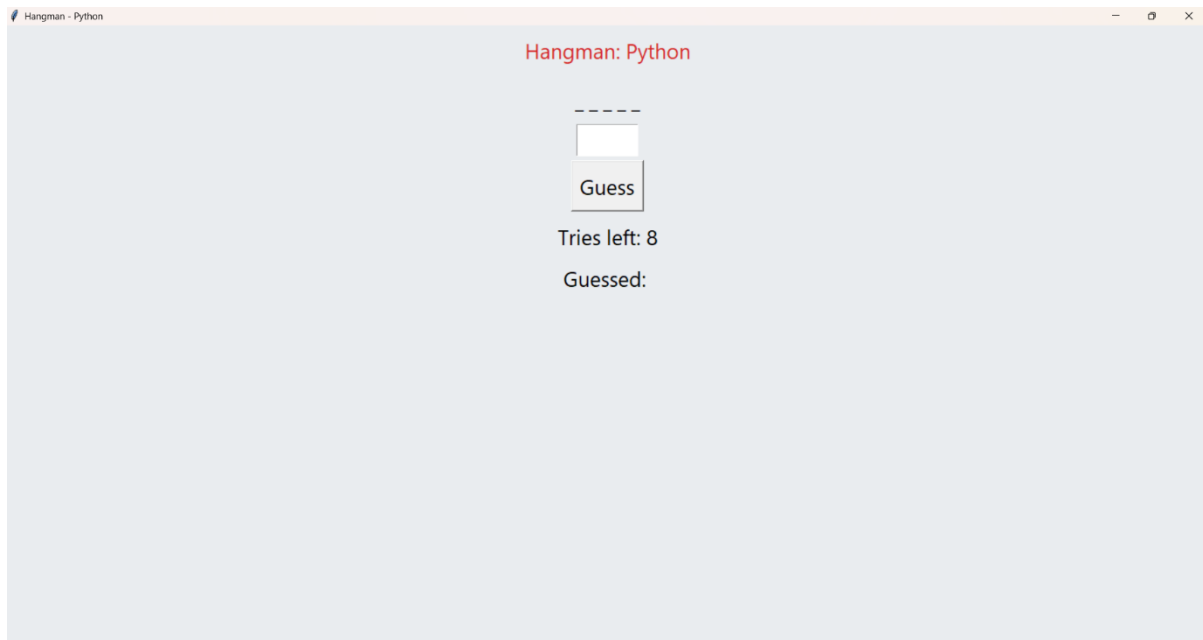
4. Game Page :



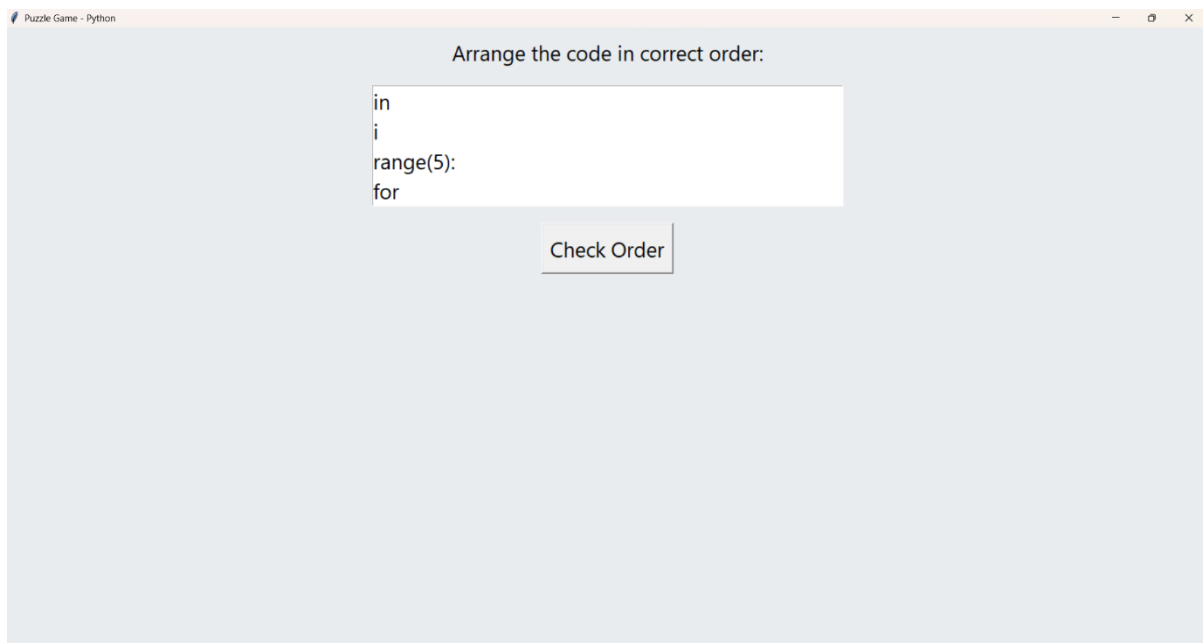
4.1 Flashcards :



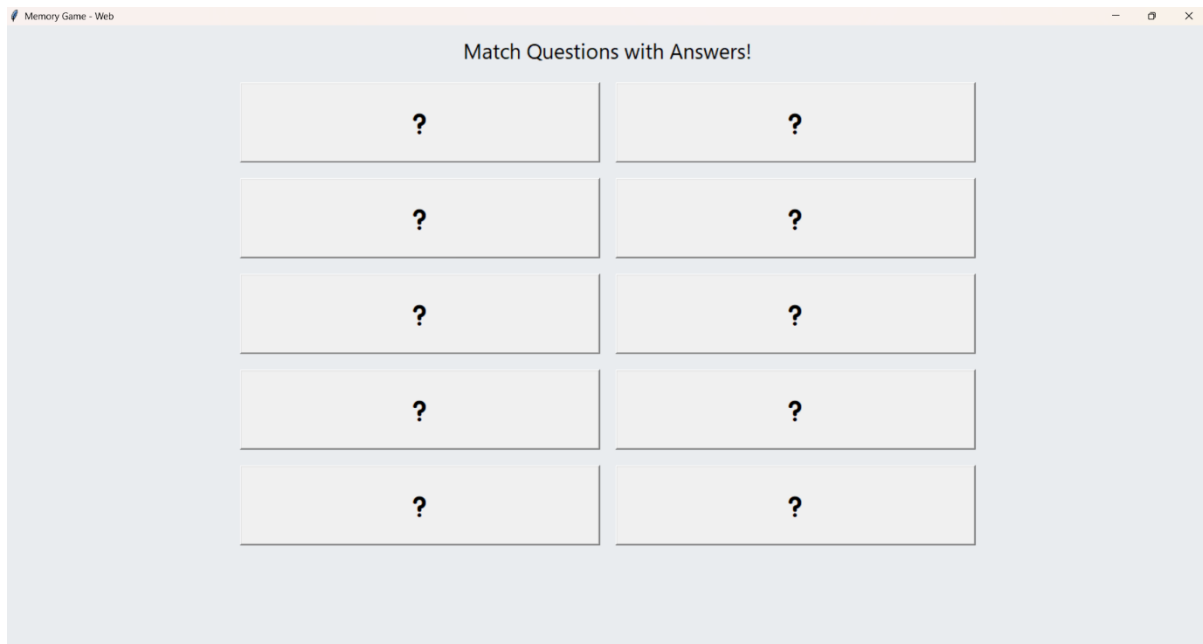
4.2 Hangman :



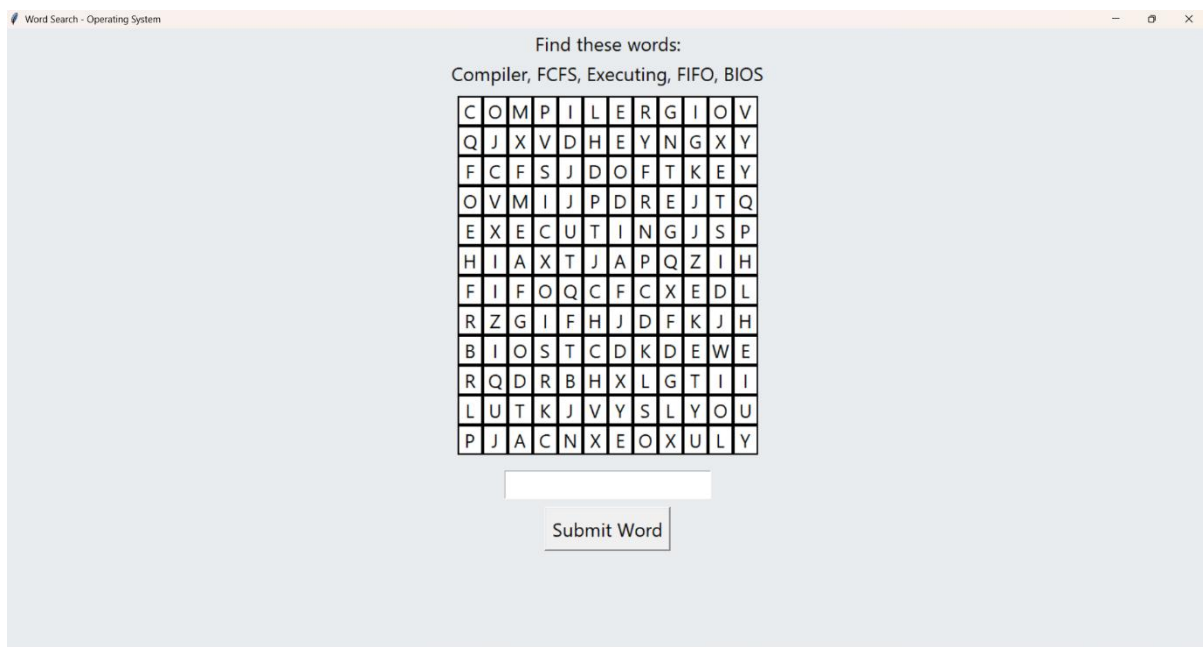
4.3 Puzzle Game :



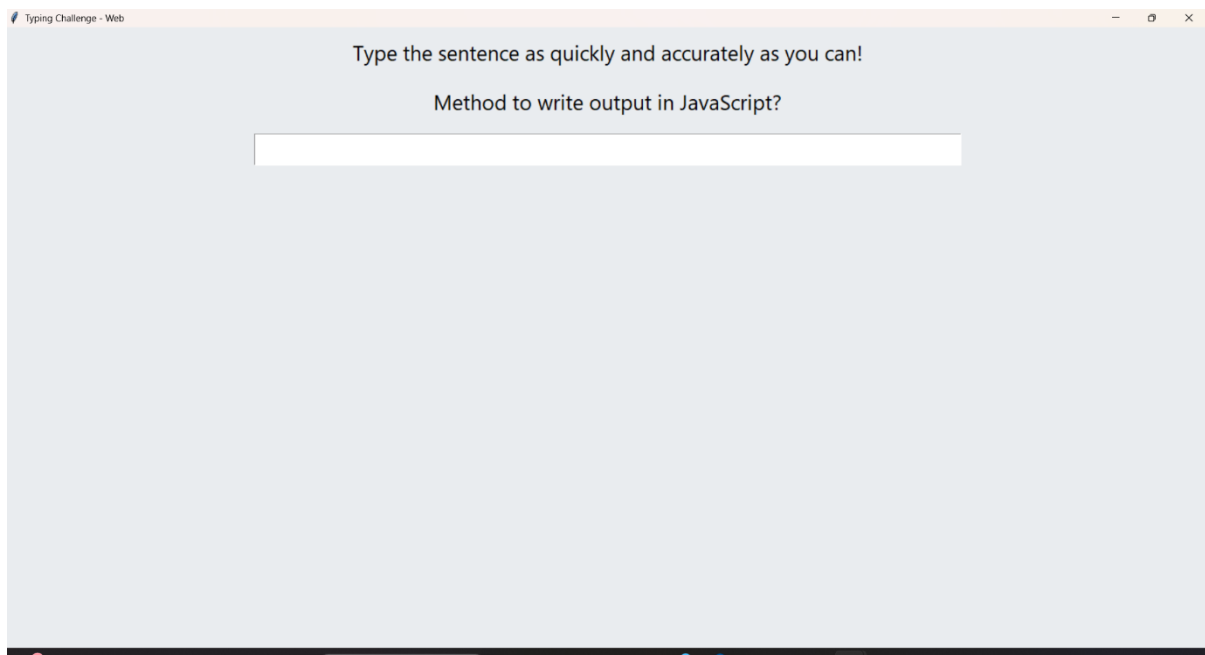
4.4 Memory Game :



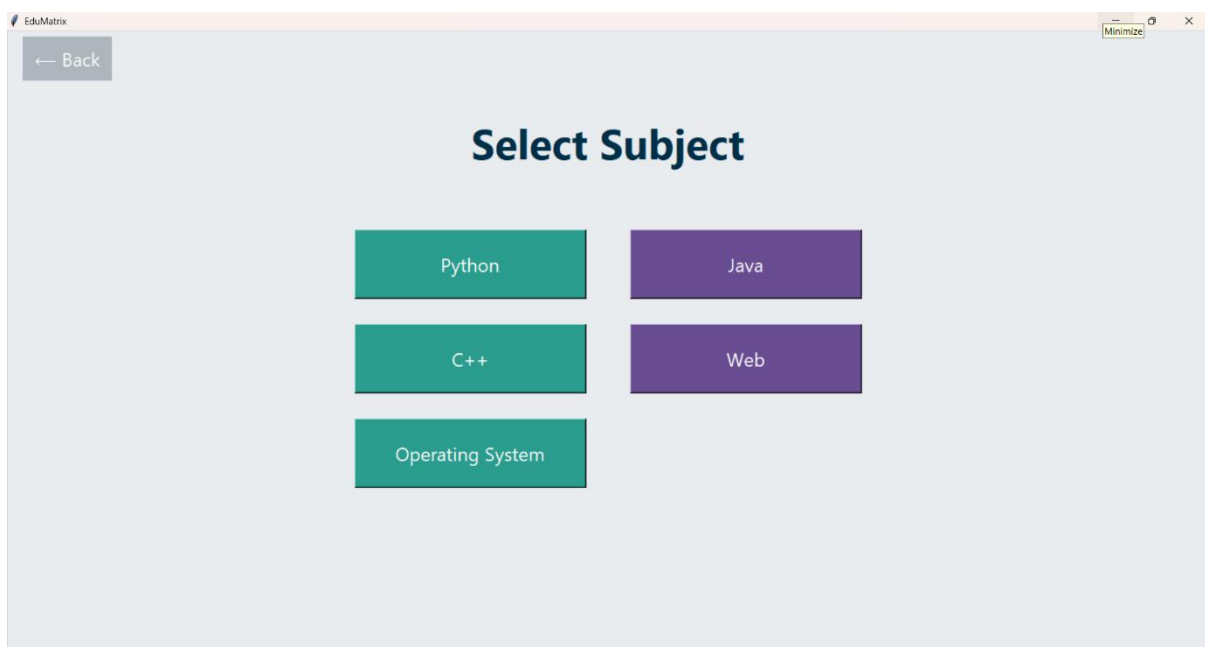
4.5 Word Search :



4.6 Typing Challenge :

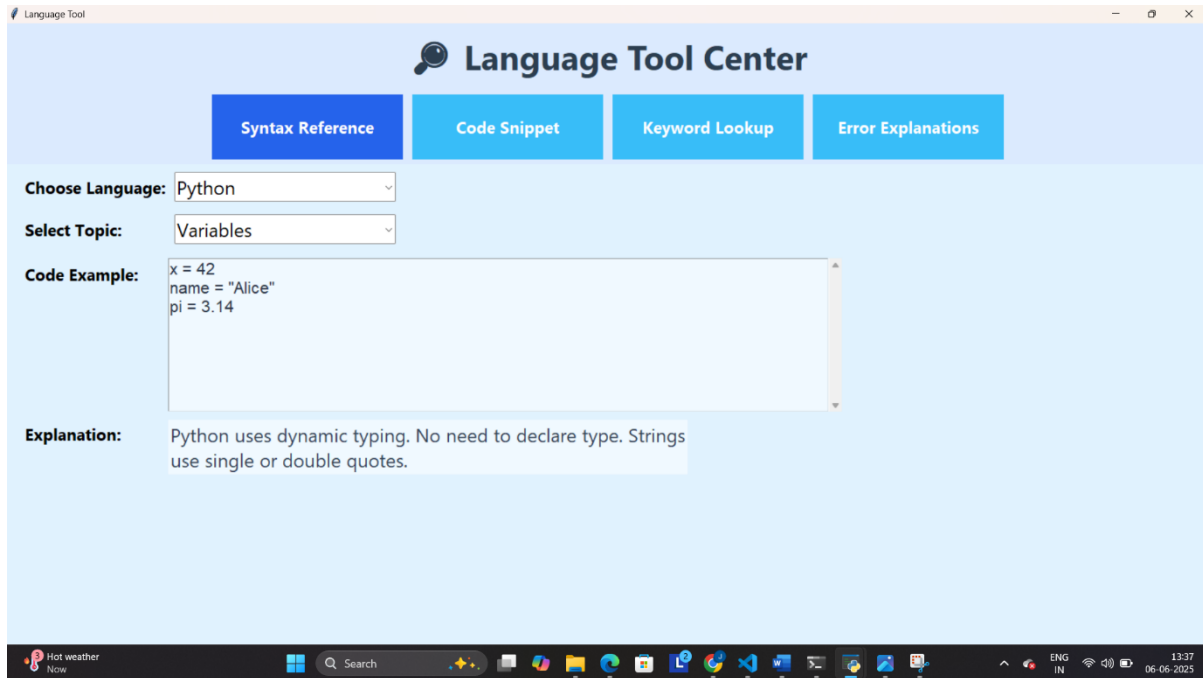


4.7 Common Subject Selection Page for all Games :

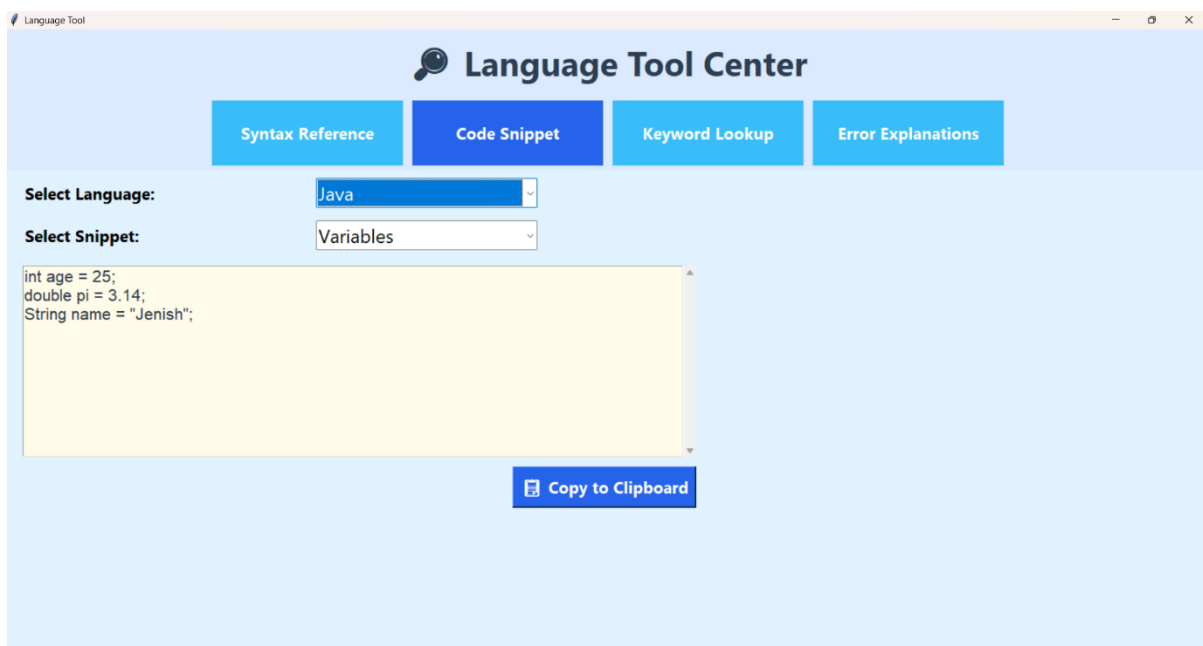


5. Language Tool Page :

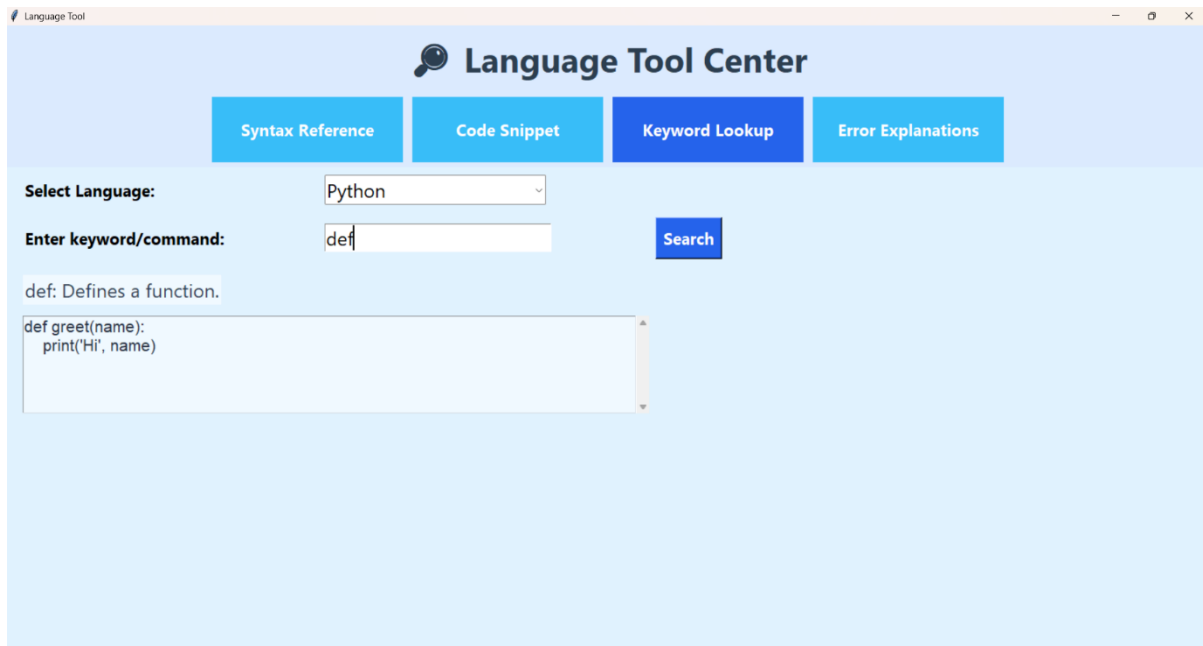
5.1 Syntax References :



5.2 Code Snippet :



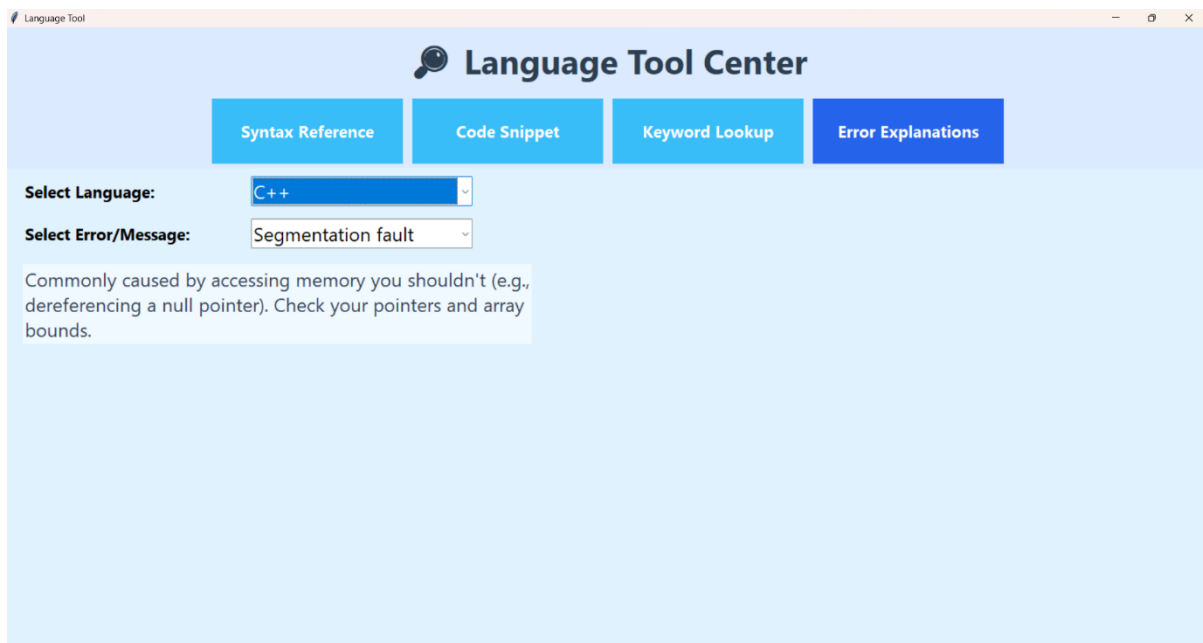
5.3 Keyword Lookup :



The screenshot shows the 'Language Tool Center' window. At the top, there are four tabs: 'Syntax Reference', 'Code Snippet', 'Keyword Lookup' (which is selected and highlighted in dark blue), and 'Error Explanations'. Below the tabs, there is a 'Select Language:' dropdown menu with 'Python' selected. To the right of the dropdown is a 'Search' button. Below the language selection, there is a text input field labeled 'Enter keyword/command:' containing the text 'def'. Below this input field, the search results are displayed: 'def: Defines a function.' followed by a code snippet in a text area:

```
def greet(name):  
    print('Hi', name)
```

5.4 Error Explanations :



The screenshot shows the 'Language Tool Center' window. At the top, there are four tabs: 'Syntax Reference', 'Code Snippet', 'Keyword Lookup', and 'Error Explanations' (which is selected and highlighted in dark blue). Below the tabs, there is a 'Select Language:' dropdown menu with 'C++' selected. Below the language selection, there is a text input field labeled 'Select Error/Message:' containing the text 'Segmentation fault'. Below this input field, the error explanation is displayed: 'Commonly caused by accessing memory you shouldn't (e.g., dereferencing a null pointer). Check your pointers and array bounds.'

5 . CONCLUSION

- The **EduMatrix** project provides an integrated, user-friendly educational platform focused on multi-disciplinary quizzes, interactive games, a comprehensive dashboard, and a robust language reference tool.
- Through careful system design—featuring clear activity diagrams, database structuring (with a focus on quiz data), and intuitive screen layouts—the application ensures both ease of use and effective learning outcomes.
- The use of Python's Tkinter for GUI development and SQLite for data persistence in the Quiz module allows for seamless offline operation, ensuring accessibility and privacy. The modular architecture supports easy maintenance and future enhancements.
- By combining assessment, gamification, and reference resources, EduMatrix delivers a holistic learning experience for users, fostering both knowledge retention and practical skills in a supportive, engaging environment.

6. LEARNING DURING SIP

Here is learning objectives during SIP :

1. Python GUI Development

> Working extensively with Tkinter improved understanding of graphical user interface design, event-driven programming, widget management, custom fonts, and application state handling.

2. Database Integration

> Implementing SQLite for quiz result storage provided hands-on experience with database schema design, CRUD operations (Create, Read, Update, Delete), and integrating persistent data storage within a desktop application.

3. Modular Programming Structure

> The project required designing and organizing code into logical classes and functions, enhancing the ability to write modular, maintainable, and reusable code.

4. User Experience (UX) Principles

> Attention to layout, navigation, color schemes, and user feedback (such as warnings, confirmations, and result displays) deepened the appreciation for user-centric application design.

5. Educational Content Structuring

> Curating and formatting multi-subject quiz questions and educational resources improved skills in information organization and content validation.

6. Debugging and Error Handling

> Encountering and resolving various bugs developed stronger debugging techniques, especially in the context of GUI and database-related errors.

7. Version Control and Documentation

> Maintaining clear documentation and managing project files instilled good practices for collaboration, code readability, and future scalability.

8. Gamification and Engagement

> Designing interactive games and dashboards provided insight into making educational software more engaging and effective for learners.

- Overall, this project fostered a comprehensive, practical understanding of desktop application development, from back-end logic to front-end user experience, equipping me with valuable skills for future projects.

7. BIBLIOGRAPHY

7.1 Online References :

1. **Tkinter Documentation** - Python Standard Library
<https://docs.python.org/3/library/tkinter.html>
2. **SQLite Official Documentation**
<https://sqlite.org/docs.html>
3. **Python Official Documentation**
<https://docs.python.org/3/>
4. **GeeksforGeeks** - Python Programming Examples
<https://www.geeksforgeeks.org/python-programming-examples/>
5. **Stack Overflow**
<https://stackoverflow.com/questions>
6. **TutorialsPoint** - Python SQLite
https://www.tutorialspoint.com/sqlite/sqlite_python.htm

7.2 Offline References :

1. E. Balagurusamy, "**Programming in Python**," McGraw Hill Education, 2017.
2. John Zelle, "**Python Programming: An Introduction to Computer Science**," Franklin, Beedle & Associates Inc, 2010.
3. Herbert Schildt, "**Java: The Complete Reference**," McGraw Hill, 2018.
4. Yashavant Kanetkar, "**Let Us C**," BPB Publications, 2018.