

Quiz Generator using MySQL and Python 3.12

Project Title: Python Quiz Management System with MySQL Integration

1. Preface:

The Python Quiz Management System with MySQL Integration project is designed to create an interactive quiz environment, covering topics such as lists, dictionaries, and strings in Python programming. The system allows participants to answer a set of multiple-choice questions (MCQs) and stores their results in a MySQL database for further analysis. The project aims to provide a hands-on experience in implementing a Python application that involves both user interaction and database integration.

2. Objectives:

The primary objectives of the project include:

- Creating a quiz system with 10 MCQs related to Python programming topics: lists, dictionaries, and strings.
- Allowing participants to answer questions interactively.
- Storing participant names and scores in a MySQL database.
- Retrieving and displaying results for all participants, including the participant with the highest score.
- Demonstrating the integration of Python with MySQL using the **mysql-connector-python** library.

3. Features:

- **Quiz Creation:** The system supports the creation of a quiz with predefined questions related to essential Python programming concepts.
- **Interactive Quiz Interface:** Participants can interactively answer the quiz questions, selecting the correct option for each MCQ.
- **MySQL Database Integration:** The project integrates with a MySQL database to store participant information, including names and quiz scores.
- **Results Display:** The system provides a feature to display the results of all participants, showcasing their names and scores.
- **Highest Scorer Recognition:** The application identifies and displays the participant with the highest score as a form of recognition.

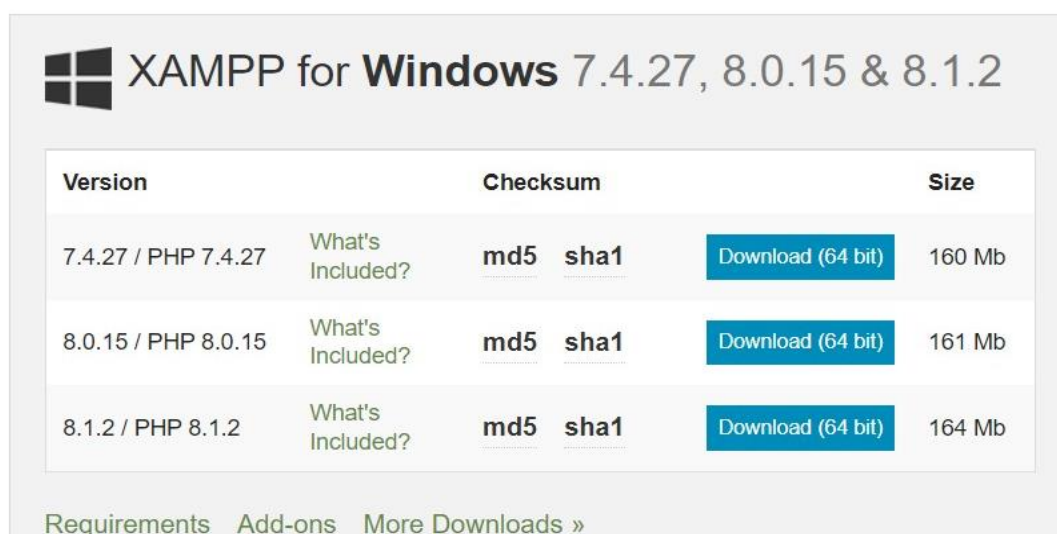
4. Technologies Used:

- **Python:** The core programming language for developing the interactive quiz application.
- **MySQL:** The chosen relational database management system for storing participant data.
- **mysql-connector-python:** A Python library used for connecting Python scripts to the MySQL database.

5) Installing the Required Software and platform

The XAMPP server is an open source localhost web server platform which consists Apache, MySQL phpMyAdmin features that are help to convert the normal system into a server that can be accessible by those computers whose are connected with them via. Local area network.

First of all, we need to install the XAMPP application form their website – <https://www.apachefriends.org/download.html>



The screenshot shows the XAMPP for Windows download page. At the top, it says "XAMPP for Windows 7.4.27, 8.0.15 & 8.1.2". Below this is a table with three rows, each representing a different version of XAMPP. The table has columns for Version, Checksum (md5 and sha1), and Size. Each row also includes a "What's Included?" link and a "Download (64 bit)" button. The latest version, 8.1.2, is highlighted.

Version	Checksum	Size
7.4.27 / PHP 7.4.27	md5 sha1	160 Mb
8.0.15 / PHP 8.0.15	md5 sha1	161 Mb
8.1.2 / PHP 8.1.2	md5 sha1	164 Mb

Requirements Add-ons More Downloads »

Fig. 5.1: Download XAMPP

Download latest version of XAMPP with PHP 8.1.2 (164MB) and Run the file from downloads '[xamppwindows-x64-8.1.2-2-VS16-installer.exe](#)' then click on next and accept terms and conditions. Then follow the process of installation.

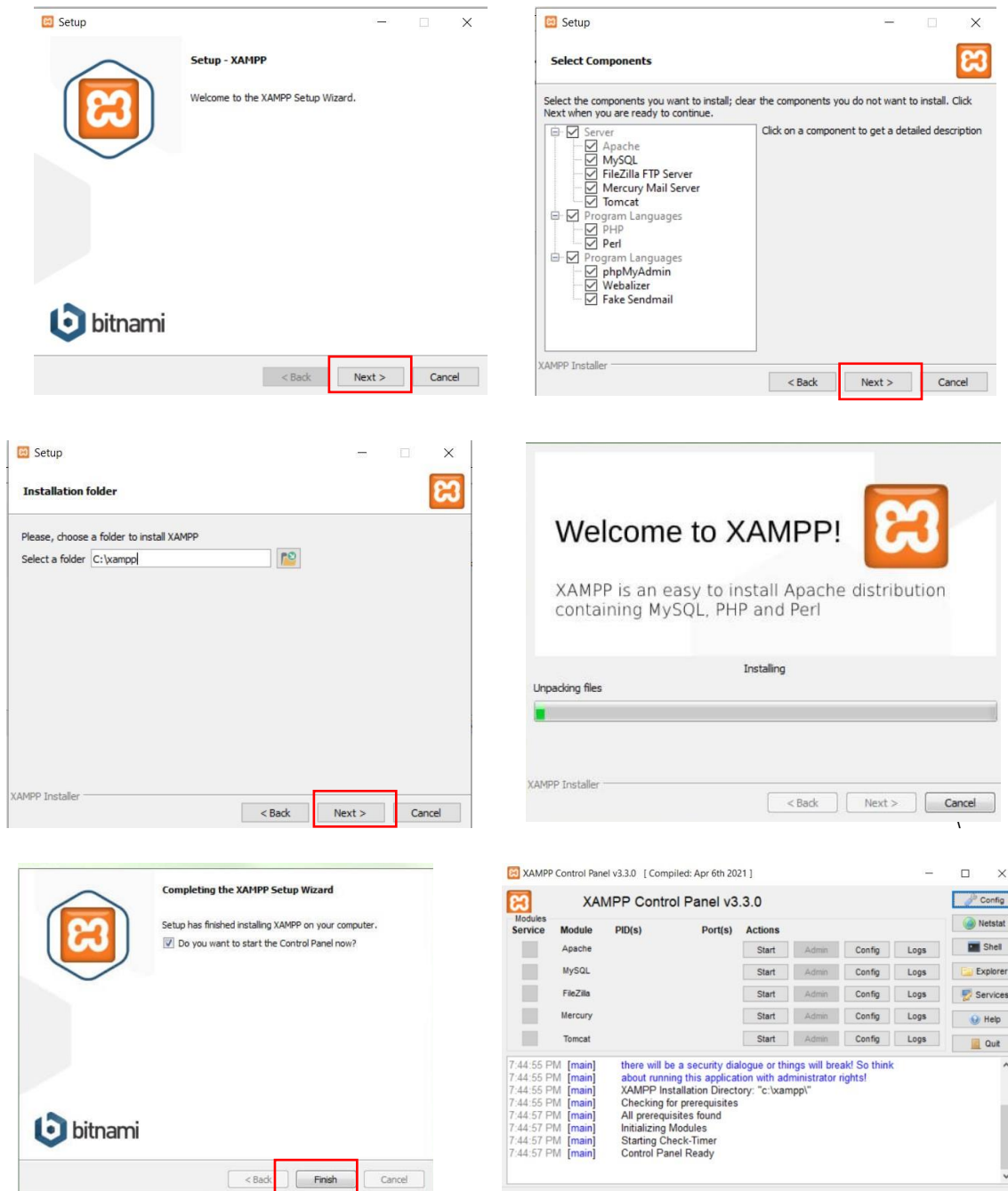


Fig. 5.2: Installation process of XAMPP

The above screenshot shows the complete installation process of XAMPP server in windows OS. This platform needs APACHE and MySQL server, so we have to start both of these services as we have to use them for this project.

The apache server tends to execute the localhost server and MySQL server, a database service that are used to create and manipulate the database using GUI as well as CUI.

The xampp application is a good choice for the developers to work with localhost and the database. The project takes both APACHE localhost and PhpMyAdmin for connecting to the database.

5.1. Starting APACHE and MySQL server

For starting the APACHE and MySQL server, the XAMPP control panel is required. In windows system, we can directly open XAMPP control from the installation path. For Mac OS, access the application menu by pressing command key and type XAMPP control panel. And for Linux, the installation path is must for opening the XAMPP control panel.

7. Python Database Connection

To connect Python to a MySQL database using the **mysql.connector** library, we first need to install the library using the following:

```
pip install mysql-connector-python
```

Once the library installed, we can use the following code as an example to establish a

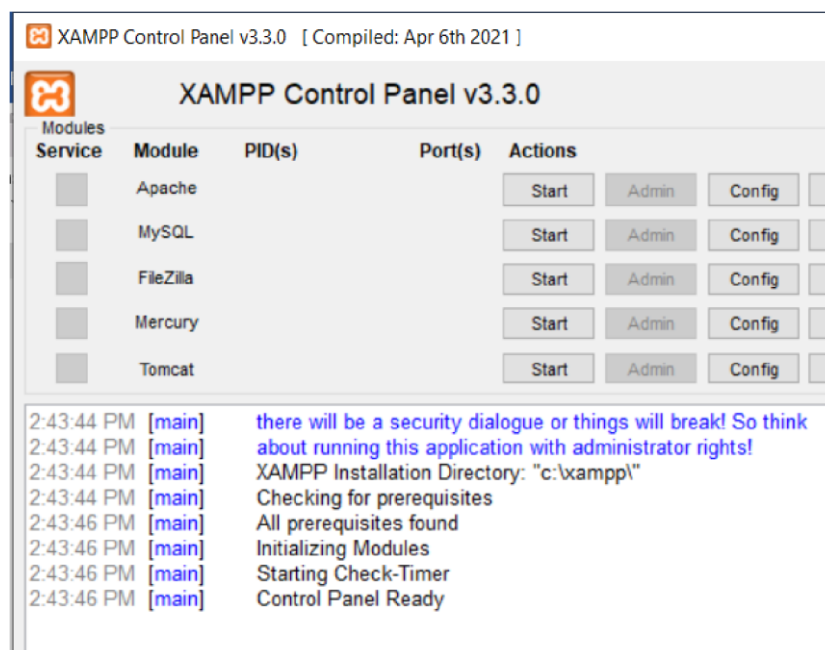
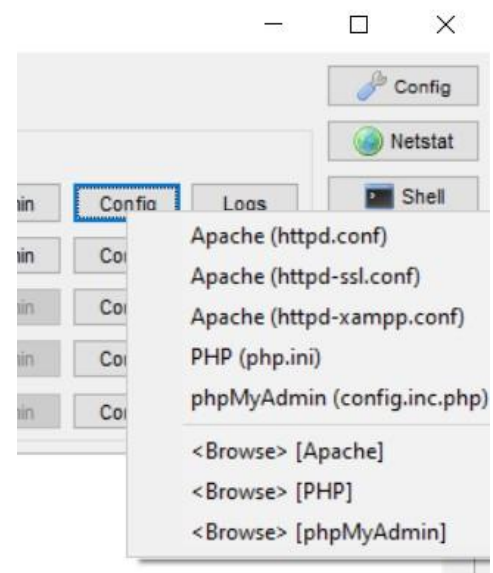


Fig. 5.3

In above figure, there are 5 servers to facilitate the module, i.e. Apache, MySQL, FileZilla, Mercury, Tomcat. The given project uses only two modules that are APACHE and MySQL.



connection to a MySQL database:

```
import mysql.connector

try:
    # Establishing a connection
    connection = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="quiz"
    )

    if connection.is_connected():
        print("Connected to MySQL database")

        # Perform database operations here

except mysql.connector.Error as e:
    print(f"Error: {e}")
```

Output:

Connected to MySQL database

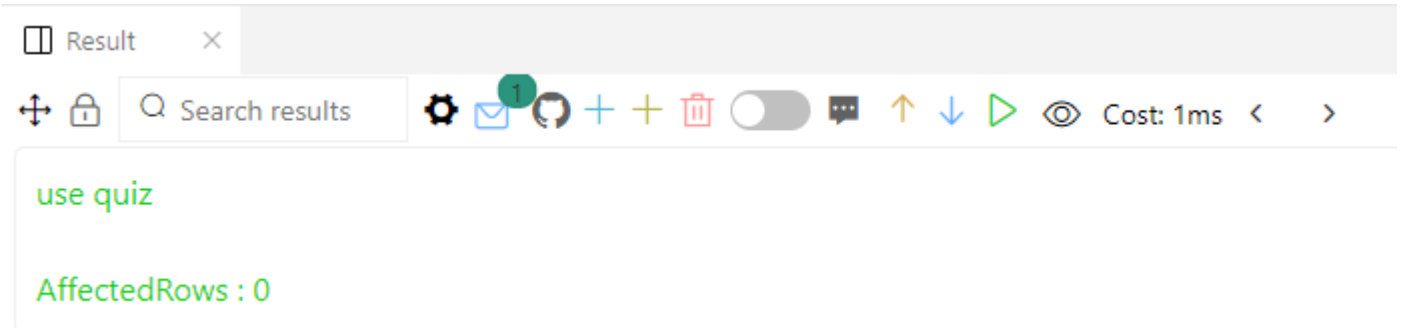
6. Implementation Steps:

1. **Database Setup:** Create a MySQL database with a table named **quiz_results** to store participant names and scores.
2. **Python Quiz Application:** Develop the interactive quiz application in Python, incorporating the **mysql-connector-python** library for database connectivity.
3. **Quiz Conductance:** Allow participants to answer the quiz questions interactively, calculating their scores based on correct answers.
4. **Database Interaction:** Implement functions to store participant results in the MySQL database and retrieve results for display.
5. **Results Display:** Develop a feature to display results for all participants, including the participant with the highest score.

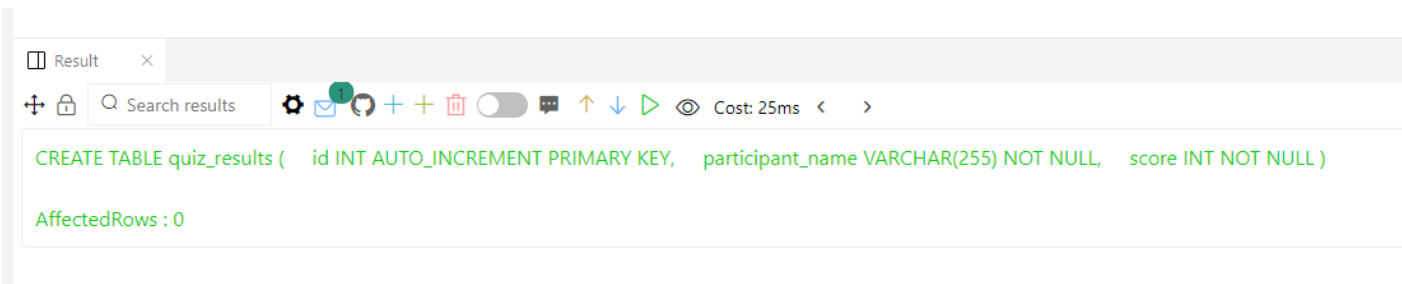
SQL Source Code

File://quiz.sql

```
CREATE DATABASE quiz;  
USE quiz;
```



```
CREATE TABLE quiz_results (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    participant_name VARCHAR(255) NOT NULL,  
    score INT NOT NULL  
);
```



Python Source Code

File://quizgenerator.py

```
import mysql.connector  
from mysql.connector import Error  
  
# Function to create a MySQL connection  
def create_connection():  
    try:  
        connection = mysql.connector.connect(  
            host='localhost',  
            database='quiz',  
            user='root',
```

```

        password=''
    )
    if connection.is_connected():
        print("Connected to MySQL database")
except Error as e:
    print(f"Error: {e}")
return connection

# Function to create the quiz
def create_quiz():
    questions = [
        {"question": "What is the correct way to declare a dictionary in Python?",
         "options": {"A": "{}", "B": "[]", "C": "()", "D": "<>"},
         "correct_answer": "A"},

        {"question": "What does the 'len()' function do?",
         "options": {"A": "Returns the length of a string", "B": "Returns the length of a list",
                     "C": "Returns the length of a dictionary",
                     "D": "All of the above"},
         "correct_answer": "D"},

        {"question": "How do you access the value associated with the key 'name' in a dictionary named 'person'?",
         "options": {"A": "person['name']", "B": "person(name)",
                     "C": "person.get('name')", "D": "name(person)"},
         "correct_answer": "A"},

        {"question": "Which method is used to convert a string to uppercase in Python?",
         "options": {"A": "upper()", "B": "capitalize()", "C": "title()",
                     "D": "uppercase()"},
         "correct_answer": "A"},

        {"question": "What does the 'join()' method do for strings in Python?",
         "options": {"A": "Joins two strings together", "B": "Splits a string into a list",

```

```

        "C": "Reverses the characters in a string",
        "D": "Converts a string to lowercase"},
        "correct_answer": "A"},

        {"question": "Which of the following is a correct way to
declare an empty list in Python?",
         "options": {"A": "list = []", "B": "list = {}", "C":
"list = ()", "D": "list = None"},
         "correct_answer": "A"},

        {"question": "What is the result of the expression 3 + 2
* 4?",
         "options": {"A": "20", "B": "15", "C": "11", "D": "25"},
         "correct_answer": "C"},

        {"question": "What is the purpose of the 'split()' method
for strings in Python?",
         "options": {"A": "Joins two strings together", "B":
"Splits a string into a list",
                     "C": "Reverses the characters in a string",
                     "D": "Converts a string to lowercase"},
         "correct_answer": "B"},

        {"question": "In Python, what is the purpose of the
'else' clause in a 'try...except' block?",
         "options": {"A": "It is used to define the exception",
                     "B": "It is executed if no exception occurs",
                     "C": "It is executed when an exception
occurs", "D": "It is optional and can be omitted"},
         "correct_answer": "B"},

        {"question": "What is the result of the expression
'hello' + 'world'?",
         "options": {"A": "'helloworld'", "B": "'hello world'",
                     "C": "'hello' 'world'", "D": "'h' 'e' 'l' 'l' 'o' ' ' 'w' 'o' 'r'
'l' 'd'"},
         "correct_answer": "A"},
    ]
    return questions

```


Function to conduct the quiz

```
def conduct_quiz(questions):
    score = 0
    for i, question in enumerate(questions, start=1):
        print(f"\nQuestion {i}: {question['question']}")
        for option, text in question['options'].items():
            print(f"{option}. {text}")

        user_answer = input("Your answer: ").upper()
        if user_answer == question['correct_answer']:
            score += 1

    return score
```

Function to store results in the database

```
def store_results(connection, participant_name, score):
    try:
        cursor = connection.cursor()
        query = "INSERT INTO quiz_results (participant_name,
score) VALUES (%s, %s)"
        data = (participant_name, score)
        cursor.execute(query, data)
        connection.commit()
        print("Results stored successfully.")
    except Error as e:
        print(f"Error: {e}")
```

Function to display results of all participants

```
def display_results(connection):
    try:
        cursor = connection.cursor()
        query = "SELECT * FROM quiz_results"
        cursor.execute(query)
        results = cursor.fetchall()

        if not results:
            print("No results found.")
        else:
```

```

        print("\nResults of all participants:")
        print("Participant Name\tScore")
        print("-----\t-----")
        for result in results:
            print(f"{result[1]}\t\t\t{result[2]}")

        highest_score = max(results, key=lambda x: x[2])
        print(f"\nParticipant with the highest score:
{highest_score[1]} with a score of {highest_score[2]}")

    except Error as e:
        print(f"Error: {e}")

# Main program
if __name__ == "__main__":
    # Create a quiz
    quiz_questions = create_quiz()

    # Get participant's name
    participant_name = input("Enter your name: ")

    # Conduct the quiz
    participant_score = conduct_quiz(quiz_questions)

    # Create a database connection
    connection = create_connection()

    # Store the results in the database
    if connection:
        store_results(connection, participant_name,
participant_score)

    # Display results of all participants
    if connection.is_connected():
        display_results(connection)

    # Close the database connection
    if connection.is_connected():
        connection.close()

```

```
print("Connection closed.")
```

Output:

Enter your name: Gaurav

Question 1: What is the correct way to declare a dictionary in Python?

- A. {}
- B. []
- C. ()
- D. <>

Your answer: A

Question 2: What does the 'len()' function do?

- A. Returns the length of a string
- B. Returns the length of a list
- C. Returns the length of a dictionary
- D. All of the above

Your answer: A

Question 3: How do you access the value associated with the key 'name' in a dictionary named 'person'?

- A. person['name']
- B. person(name)
- C. person.get('name')
- D. name(person)

Your answer: A

Question 4: Which method is used to convert a string to uppercase in Python?

- A. upper()
- B. capitalize()
- C. title()
- D. uppercase()

Your answer: A

Question 5: What does the 'join()' method do for strings in Python?

- A. Joins two strings together
- B. Splits a string into a list
- C. Reverses the characters in a string
- D. Converts a string to lowercase

Your answer: A

Question 6: Which of the following is a correct way to declare an empty list in Python?

- A. list = []
- B. list = { }
- C. list = ()
- D. list = None

Your answer: A

Question 7: What is the result of the expression $3 + 2 * 4$?

- A. 20
- B. 15
- C. 11
- D. 25

Your answer: c

Question 8: What is the purpose of the 'split()' method for strings in Python?

- A. Joins two strings together
- B. Splits a string into a list
- C. Reverses the characters in a string
- D. Converts a string to lowercase

Your answer: B

Question 9: In Python, what is the purpose of the 'else' clause in a 'try...except' block?

- A. It is used to define the exception
- B. It is executed if no exception occurs
- C. It is executed when an exception occurs
- D. It is optional and can be omitted

Your answer: D

Question 10: What is the result of the expression 'hello' + 'world'?

- A. 'helloworld'
- B. 'hello world'
- C. 'hello' 'world'
- D. 'h' 'e' 'l' 'l' 'o' ' ' 'w' 'o' 'r' 'l' 'd'

Your answer: B

Connected to MySQL database

Results stored successfully.

Results of all participants:

Participant Name	Score
------------------	-------

-----	-----
-------	-------

Alok	5
Sarthak	3

Gaurav 7

Participant with the highest score: Gaurav with a score of 7
Connection closed.