

SQL Connectivity In Streamlit

By Benak Deepak



AGENDA

- 01 Prerequisites
- 02 SQL and Streamlit
- 03 Implementing the project
- 04 Code Understanding
and conclusion

- 1)Python
- 2)VS Code
- 3)MySQL
- 4)Install packages:
 - i)pandas
 - ii)sql-connector
 - iii)sreamlit

01

Prerequisite

TERMINAL COMMANDS-

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

```
>>pip install streamlit
```

```
>>pip install pandas
```

02

SQL
Streamlit



SQL-CONNECTOR

Introduction	Key features	Advantages
<ul style="list-style-type: none">->Python Library->connect with Mysql->Python applications to execute SQL queries	<ul style="list-style-type: none">->Database connectivity->SQL queries->Data retrievals->Error Handling->Connecting Pooling	<ul style="list-style-type: none">->Easy to use->Performance->Flexibility



SQL COMMANDS IN MySQL
Are as follows....

Step1->CREATE DATABASE FarmManagement;

Step2->USE FarmManagement;

Step3->CREATE TABLE Farmer (
farmer_id INT PRIMARY KEY,
name VARCHAR(100) NOT NULL,
address VARCHAR(255),
contact_info VARCHAR(255));

----Creating the farm table----

```
CREATE TABLE Farm (  
  farm_id INT PRIMARY KEY,  
  location VARCHAR(255),  
  size DECIMAL(10, 2),  
  farmer_id INT, FOREIGN KEY (farmer_id) REFERENCES  
  Farmer(farmer_id)  
);
```

Recap on Streamlit:

- Title
- Sidebar
- Header
- Selectbox
- Write
- Form
- And many more....

Connections.....

Database connection

Function to execute query and return results

To run an action in python

Database connection

```
def get_connection():  
    return mysql.connector.connect(  
        host="localhost",  
        user="root",  
        password="xyz",  
        database="FarmManagement"  
    )
```

Function to execute query and return results

```
def run_query(query):  
    conn = get_connection()  
    cursor = conn.cursor()  
    cursor.execute(query)  
    data = cursor.fetchall()  
    columns = [col[0] for col in cursor.description]  
    conn.close()  
    return pd.DataFrame(data, columns=columns)
```

To run an action in python

```
def run_action(query, values):  
    conn = get_connection()  
    cursor = conn.cursor()  
    cursor.execute(query, values)  
    conn.commit()  
    conn.close()
```



03

Implementing
DBMS project



Project Explanation

Farm Management
System

A decorative border composed of a grid of squares. The top row has three squares: a large red one on the left, a medium gray one in the middle, and a small light gray one on the right. The bottom row has four squares: a small white one on the far left, a large light gray one, a medium red one, and a small light gray one on the far right. The central area of the slide is a large white rectangle.

Shifting to VS Code



04

Conclusion

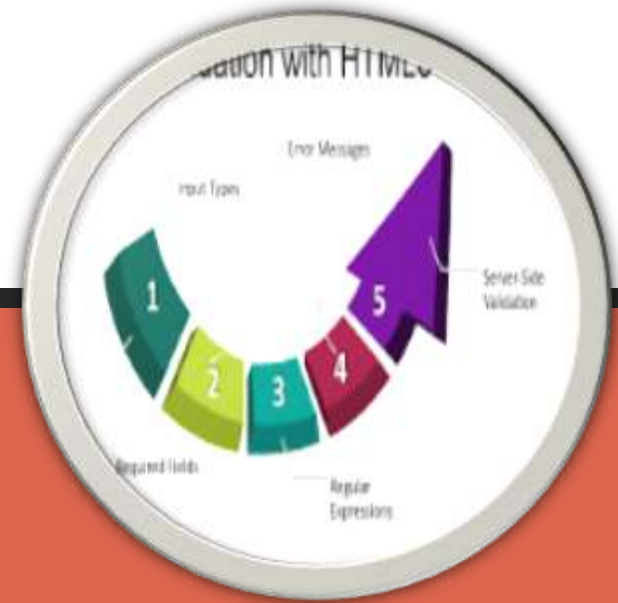
Future Extension



Multi media support



**Advanced Data
Visualization**



**Forms and User Input
Handling**