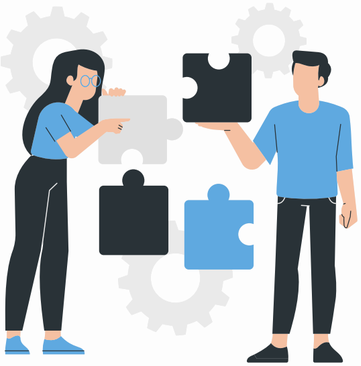
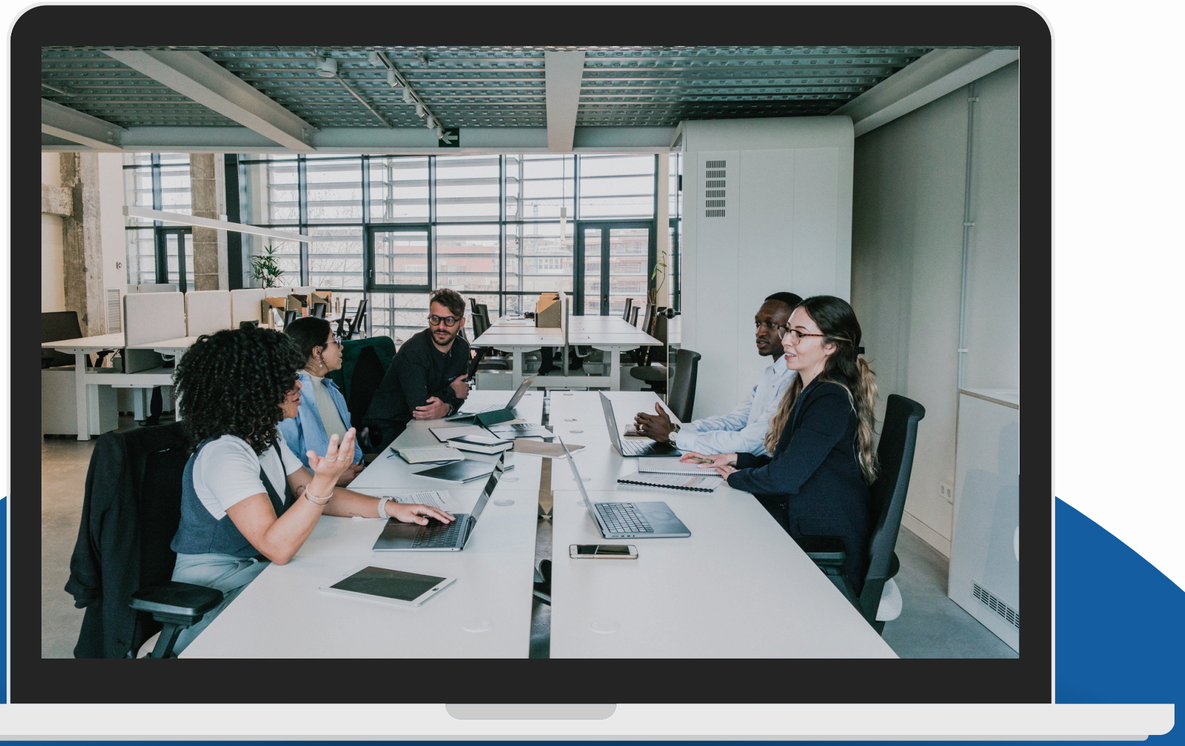
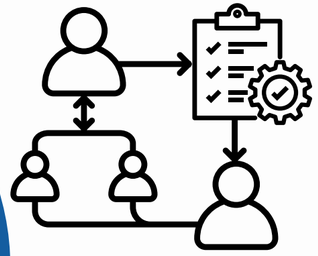


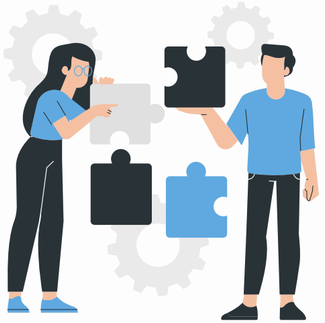
Student Management System

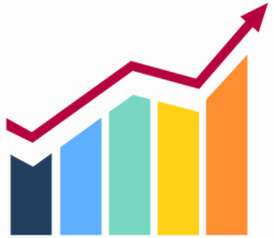


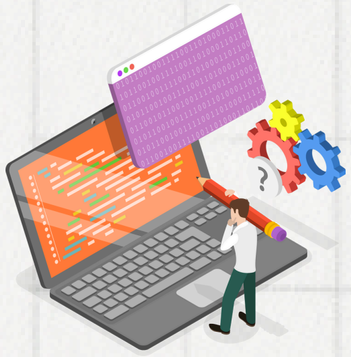
By : Varnika Chaudhary



Overview







PROJECT GOALS

01

Organize and Manage Data: Centralized storage for all student data, allowing quick access and updates.

02

Data Security: Structured database storage to secure sensitive student information.

03

Scalability: Designed for institutions of varying sizes, from small schools to larger universities.

04

Future Expansion: Can be upgraded with features like graphical interfaces and login credentials.

System Architecture

USER INTERFACE

Command-line interface providing a straightforward menu system.

BACKEND

Python code that connects and interacts with a MySQL database.

DATA FLOW

- User selects an operation (Create, Read, Update, Delete).
- Python functions execute the operation on the MySQL database.
- Results are displayed or updated in the database based on user commands.

DATABASE

MySQL, chosen for its reliability and efficiency in handling relational data.

Database Structure

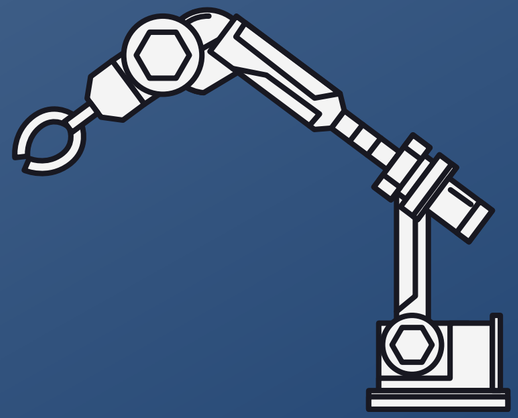
- **Database Name:** stud_management
- **Tables:**

1. users Table: Contains login details for authorized users.
2. s_students Table: Holds student-specific information like roll number, name, age, and course.

- Design Rationale: A simple, two-table structure allows efficient querying and data organization, keeping student records separate from user data for better security.

Database Setup

(Code Example)



MySQL80

uery Database Server Tools Scripting Help

Query 1

Limit to 1000 rows

```
1   CREATE DATABASE stud_management;
2   USE stud_management;
3
4   CREATE TABLE users (
5       username VARCHAR(100) PRIMARY KEY,
6       password_hash VARCHAR(64),
7       phone_number VARCHAR(15)
8   );
9
10  CREATE TABLE s_students (
11      roll_number VARCHAR(10) PRIMARY KEY,
12      s_name VARCHAR(100),
13      age INT,
14      course VARCHAR(50)
15      );
```

SQLAdditions

Automatic
disabled. U
manually
current ca
toggle a

Context Help Sni

Output

Action Output

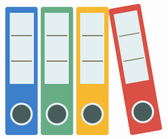
#	Time	Action	Message
1	14:57:04	CREATE DATABASE stud_management	1 row(s) affected
2	14:57:22	USE stud_management	0 row(s) affected
3	14:57:28	CREATE TABLE users (username VARCHAR(100) PRIMARY KEY, password_hash VARCHAR(64), phon...	0 row(s) affected
4	14:57:35	CREATE TABLE s_students (roll_number VARCHAR(10) PRIMARY KEY, s_name VARCHAR(100), age I...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser
5	14:57:44	CREATE TABLE s_students (roll_number VARCHAR(10) PRIMARY KEY, s_name VARCHAR(100), age I...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL ser
6	14:58:57	CREATE TABLE s_students (roll_number VARCHAR(10) PRIMARY KEY, s_name VARCHAR(100), age I...	0 row(s) affected

Search ENG IN

OVERVIEW: SETTING UP THE MYSQL DATABASE AND TABLES.

CODE EXPLANATION:

- CREATE STUD_MANAGEMENT DATABASE.
- DEFINE USERS TABLE TO MANAGE SYSTEM USERS.
- DEFINE S_STUDENTS TABLE TO STORE STUDENT INFORMATION.







Python-MySQL Connection

- Purpose: Establish a connection between Python and the MySQL database for data manipulation.
- Explanation:
- Python's mysql.connector library is used to connect to the MySQL server.
- Defines the server, user credentials, and database.

Purpose of Connection: Ensures Python can interact with the database to execute CVUD operations.

Run Terminal Help python

python2.py

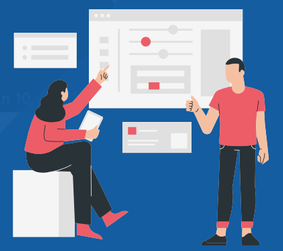
```python
import mysql.connector
import bcrypt

# Database connection
dataBase = mysql.connector.connect(
    host="localhost",
    user="root2",
    passwd="amrit76688",
    database="stud_management",
    auth_plugin='mysql_native_password'
)

# Creating a cursor object
cursorObject = dataBase.cursor()

# Function to register a new user
def register_user():
    username = input("Enter a new username: ")
    password = input("Enter a new password: ")
    phone_number = input("Enter your phone number (optional): ")

    # Hash the password before storing it
    hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
```

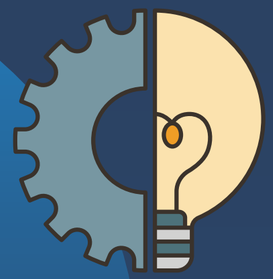
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

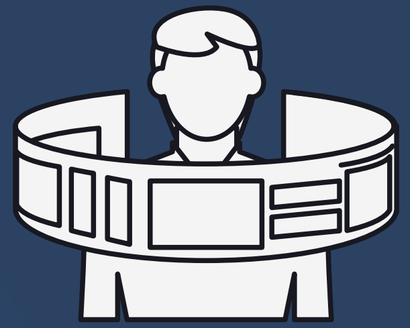
```
5. Logout
Enter your choice: 4
Student Records:
('1401', 'Simran', 22, 'Btech(CS)')
('2', 'Varnika', 24, 'B.tech')
('24', 'Pranjal', 34, 'Mca')
('2401', 'Simran', 22, 'B.tech(C.S)')


Student Management Menu:
1. Add Student
2. Update Student
```

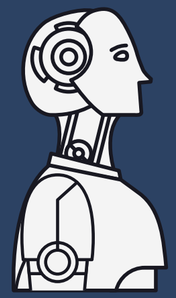
Code Search Error Share Code Link

Search









FUNCTIONALITIES AND MENU OPTIONS

- **ADD STUDENT** — Allows users to input new student data.
- **UPDATE STUDENT** — Provides options to modify details for an existing student.
- **DELETE STUDENT** — Removes a student record based on the provided ID.
- **VIEW STUDENTS** — Retrieves and displays all stored student records.
- **EXIT PROGRAM** — Ensures the database connection is closed securely upon exiting.

BENEFITS OF THE SYSTEM

DATA CONSISTENCY

The structured storage ensures accurate and organized data.

SECURITY

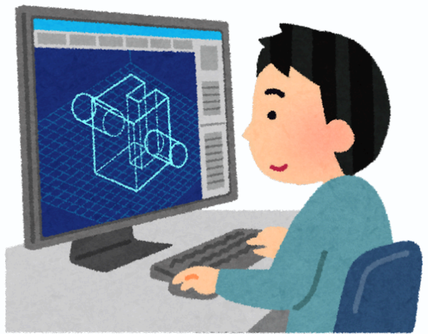
Sensitive student data remains in a protected database.

EASY MANAGEMENT

CVUD operations make it simple for users to manage records.

SCALABILITY

Easily accommodates growing data needs, like new students or course updates.



Challenges and Solutions

Challenge: Handling Errors (e.g., invalid input or missing records).

- **Solution:** Implement error-handling mechanisms in the code to provide meaningful error messages.

Challenge: Data Security and Integrity.

- **Solution:** Ensure primary keys (like roll_number) maintain uniqueness and data is well-structured.

Challenge: Database Connectivity Issues.

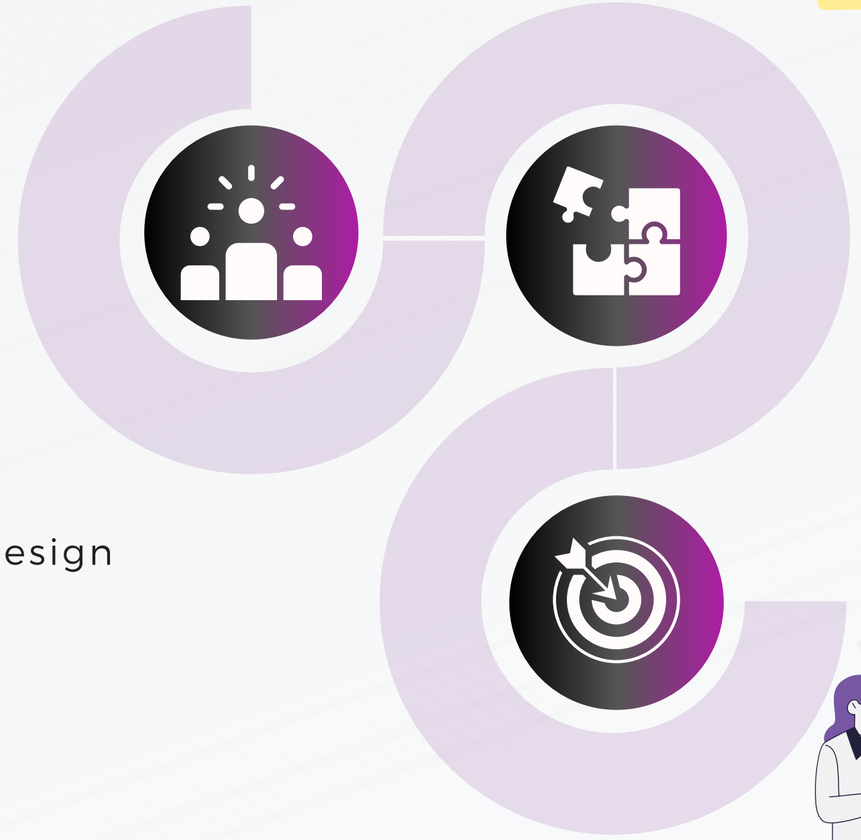
- **Solution:** Incorporate connection status checks and a structured database close method.

CONCLUSION

01

Summary:

- Successfully developed a Student Management System utilizing Python and MySQL.
- The project showcases efficient CRUD operations, data security, and a strong foundational structure for future enhancements.



02

Learning Outcomes:

- Gained practical experience with database design and Python-MySQL .
- Understood the importance of efficient data handling for real-world applications.
- Recognized the potential for scalability and modularity in software design.

THANK YOU!