Employee Management System

Shruti Anandas

Employee ID: 207015

shruti.anandas@amdocs.com

Agenda

- Introduction
- Project Structure & Workflow
- CRUD Operations
- Technologies
- Conclusion

Introduction

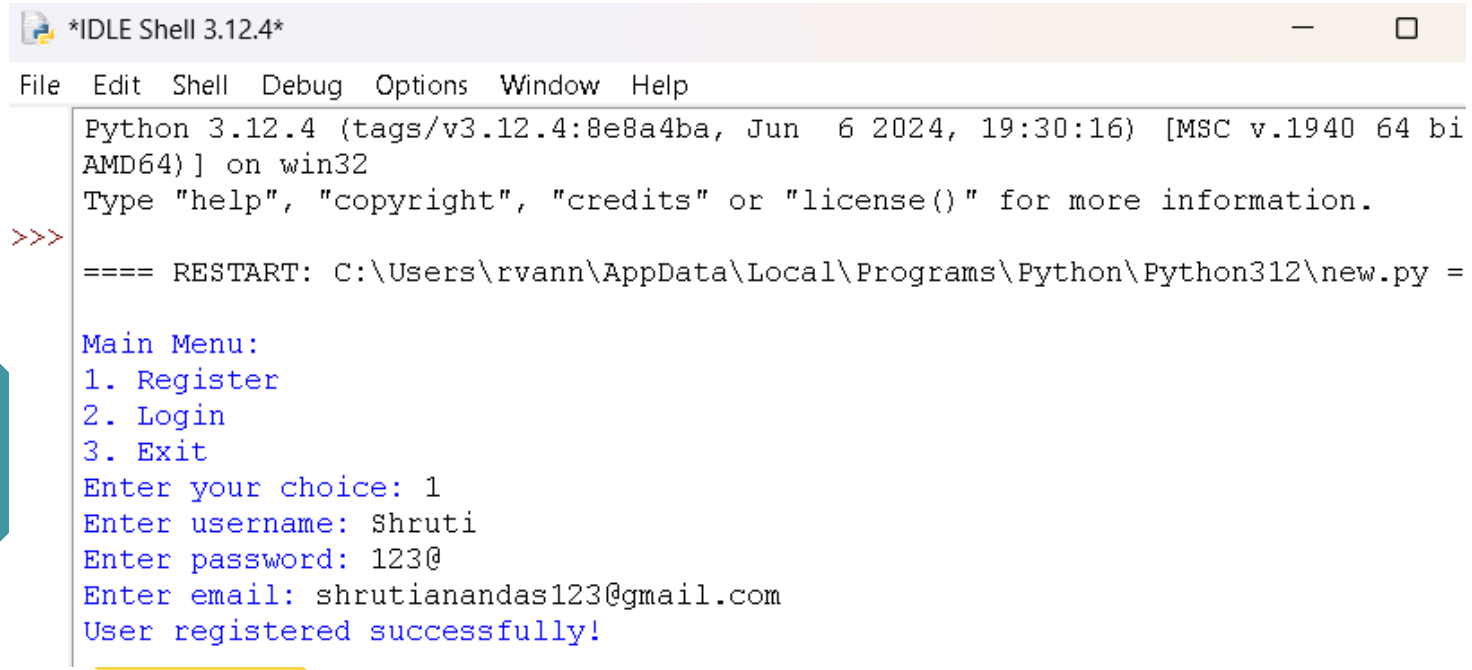
- **Objective**
  - Manage Employee Data: Add, View, Update, Delete
  - User Authentication with Registration & Login

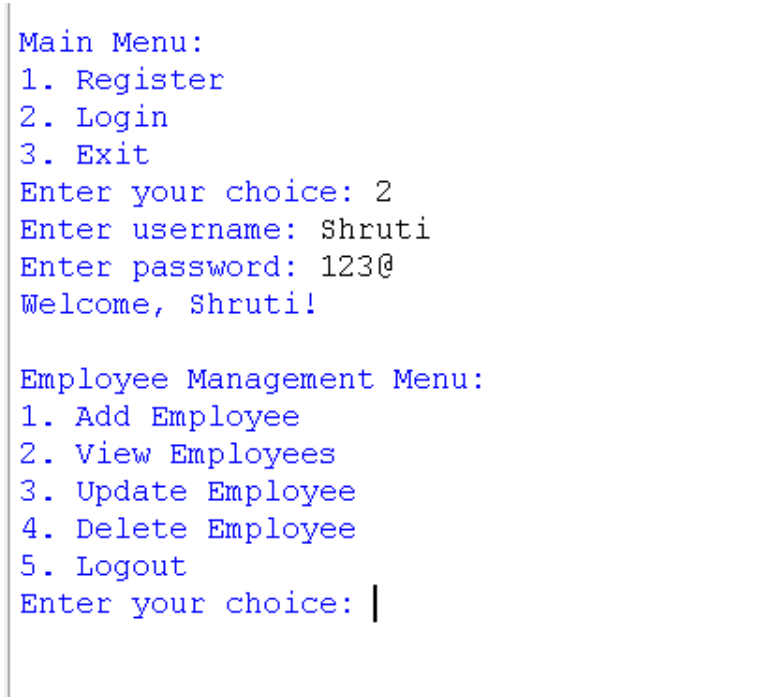
- **Technologies**
  - Python
  - MySQL Workbench
  - SHA-256 for password hashing

Project Structure & Workflow

- User authentication: Register > Login
- Employee Operations (CRUD): Create, Read, Update, Delete
- Database Tables:
    1. users – Stores username, password & email
    2. employees – Stores employee details

Registration & Login Flow





Employee Operations

CRUD Operations

Add Employee

```
Main Menu:
1. Register
2. Login
3. Exit
Enter your choice: 2
Enter username: Shruti
Enter password: 123@
Welcome, Shruti!

Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 1
Enter employee name: Shruti
Enter department: IT
Enter salary: 40000
Employee added successfully!
```

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 1
Enter employee name: Anagha
Enter department: AI
Enter salary: 50000
Employee added successfully!
```

Similarly add all 5 employees

Read Operation

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 2

Employee List:
ID: 1, Name: Shruti, Department: IT, Salary: 40000.00
ID: 2, Name: Anagha, Department: AI, Salary: 50000.00
ID: 3, Name: Preeti, Department: UI/UX, Salary: 30000.00
ID: 4, Name: Sejal, Department: Docter, Salary: 560000.00
ID: 5, Name: Nandini, Department: Events, Salary: 45000.00
```

```
15
16 •   select*from employees;
17
18
```

Result Grid | Filter Rows:

emp_id	name	department	salary
1	Shruti	IT	40000.00
2	Anagha	AI	50000.00
3	Preeti	UI/UX	30000.00
4	Sejal	Docter	560000.00
5	Nandini	Events	45000.00
NULL	NULL	NULL	NULL

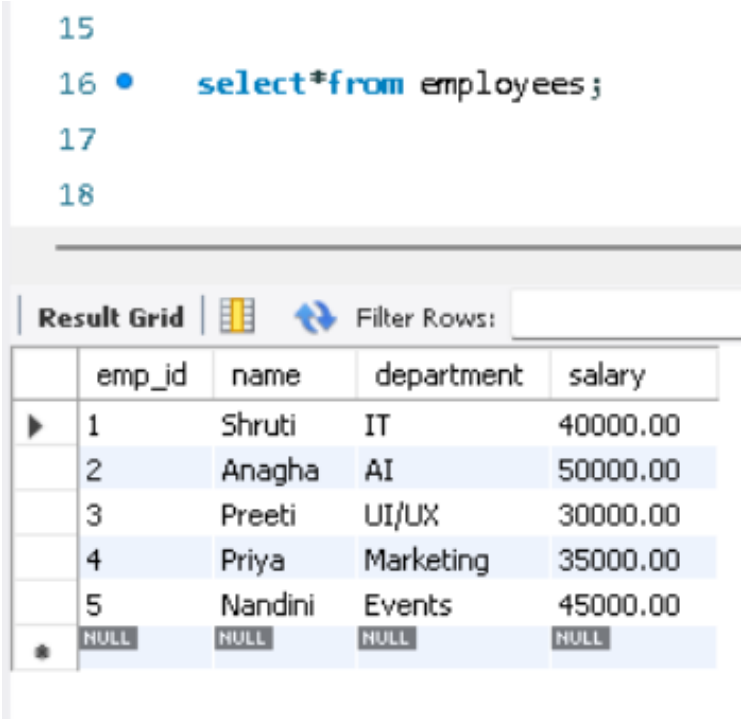
Workbench

Update Operation

Update Employee No. 4

```
Employee List:
ID: 1, Name: Shruti, Department: IT, Salary: 40000.00
ID: 2, Name: Anagha, Department: AI, Salary: 50000.00
ID: 3, Name: Preeti, Department: UI/UX, Salary: 30000.00
ID: 4, Name: Sejal, Department: Docter, Salary: 560000.00
ID: 5, Name: Nandini, Department: Events, Salary: 45000.00

Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 3
Enter employee ID to update: 4
Enter new name: Priya
Enter new department: Marketing
Enter new salary: 35000
Employee updated successfully!
```



Workbench

Delete Operation

Delete Employee No. 5

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 2

Employee List:
ID: 1, Name: Shruti, Department: IT, Salary: 40000.00
ID: 2, Name: Anagha, Department: AI, Salary: 50000.00
ID: 3, Name: Preeti, Department: UI/UX, Salary: 30000.00
ID: 4, Name: Priya, Department: Marketing, Salary: 35000.00
ID: 5, Name: Nandini, Department: Events, Salary: 45000.00
```

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 4
Enter employee ID to delete: 5
Employee deleted successfully!
```

```
16 •    select*from employees;
17
18
```

Result Grid | Filter Rows: | Edi

emp_id	name	department	salary
1	Shruti	IT	40000.00
2	Anagha	AI	50000.00
3	Preeti	UI/UX	30000.00
4	Priya	Marketing	35000.00
NULL	NULL	NULL	NULL

Workbench

Logout

Logging out & Existing the system

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 5
Logged out.
```

Exist

```
Main Menu:
1. Register
2. Login
3. Exit
Enter your choice: 3
Exiting...
>>>
>>>
```

Code Walkthrough - Modules

new.py - C:\Users\rvann\AppData\Local\Programs\Python\Python312\new.py (3.12.4)

File Edit Format Run Options Window Help

```
import mysql.connector
import hashlib

# Connect to the MySQL database (Update with your credentials)
conn = mysql.connector.connect(
    host="localhost",
    user="root",  # Replace with your MySQL username
    password="Akshay@123",  # Replace with your MySQL password
    database="employee_system"
)
cursor = conn.cursor()
```

Importing Required Modules

mysql.connector connect Python to a MySQL DB

hashlib to store user passwords securely

- SHA-256 hash

Database Connection Setup

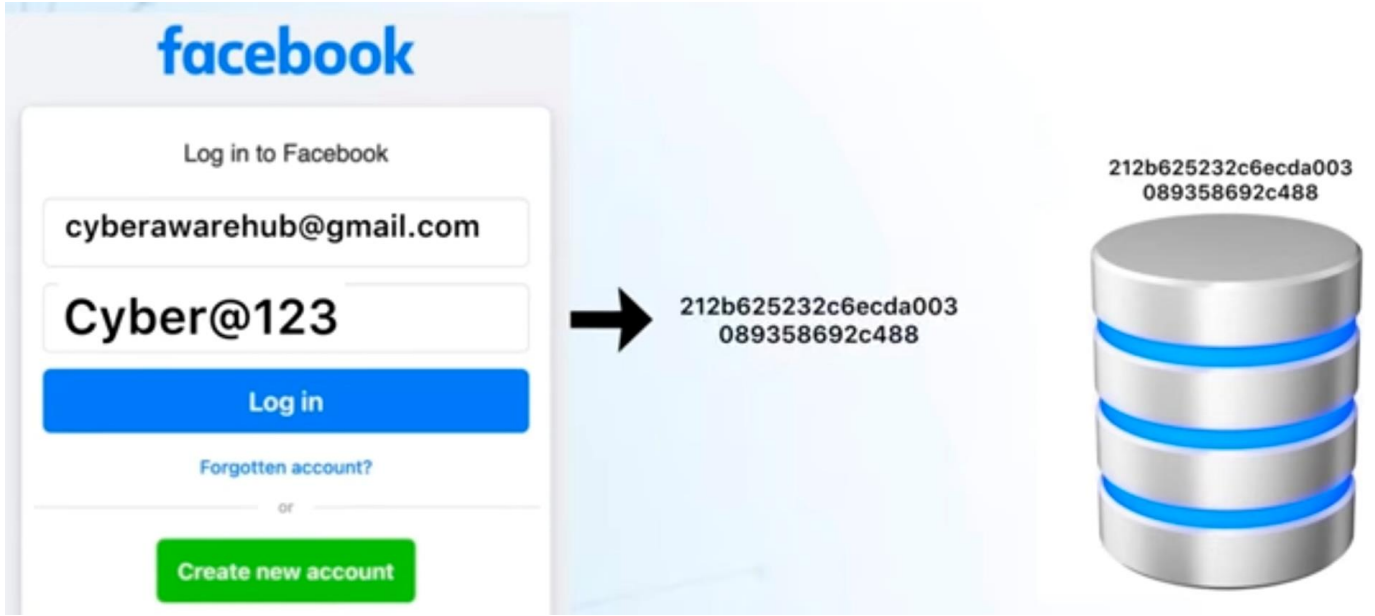
mysql.connector.connect()

cursor = conn.cursor() to execute SQL queries

Hashing Function

```
# Function to hash passwords for security
def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()
```

- Fixed size hash value
- Output is Unique for every input
- Protects against database leaks
- Irreversible hash ensures security



Code Walkthrough – User Registration

```python
# User Registration
def register_user():
    username = input("Enter username: ")
    password = input("Enter password: ")
    email = input("Enter email: ")
    hashed_password = hash_password(password)

    try:
        cursor.execute(
            "INSERT INTO users (username, password, email) VALUES (%s, %s, %s)",
            (username, hashed_password, email)
        )
        conn.commit()
        print("User registered successfully!")
    except mysql.connector.Error as err:
        print(f"Error: {err}")
```

User Registration Function

User Login Function

Add Employee Function

View Employees Function

Update Employee Function

Delete Employee Function

Error Handling

- Error Handling Example:

```
try:
    cursor.execute('...')
except mysql.connector.Error as err:
    print(f'Error:{err}')
```

- Prevent Program Crashes
- Execute Alternative Code
- Improves user experience by showing meaningful error messages.

Conclusion

- The project provides a solution for employee management with secure user authentication.
- **Secure**: Implements hashing to protect sensitive employee information.
- **Efficient Data Management**: Supports seamless CRUD operations for employee records.
- **User-Friendly**: Combines Python’s simplicity with SQL for fast, easy data handling.

Thank you

Shruti Anandas

Employee ID: 207015

shruti.anandas@amdocs.com