Employee Management System (Console-Based Application)

By:

Rishabh Raj

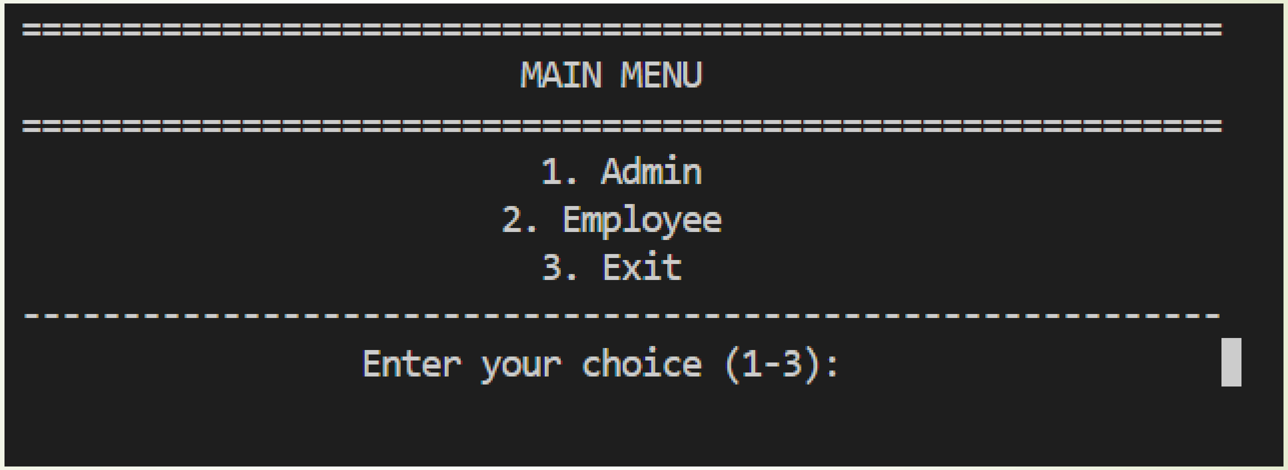
206633

Rishabh.Raj@amdocs.com

Introduction

- • A console-based application developed using Python.
- • Manages employee data, allowing admin CRUD operations.
- • Integrates with MySQL to store employee information.
- • Admin and Employee have different access privileges.

Main Menu



```
============================================================
                         MAIN MENU
============================================================
                         1. Admin
                        2. Employee
                         3. Exit
------------------------------------------------------------
                 Enter your choice (1-3):
```

Admin Functionalities

- • Admin has complete control over employee data.
- • Can perform CRUD operations:
-   - Insert new employee details.
-   - Update existing employee information.
-   - Delete employees from the system.
-   - View employee details (all or by ID).

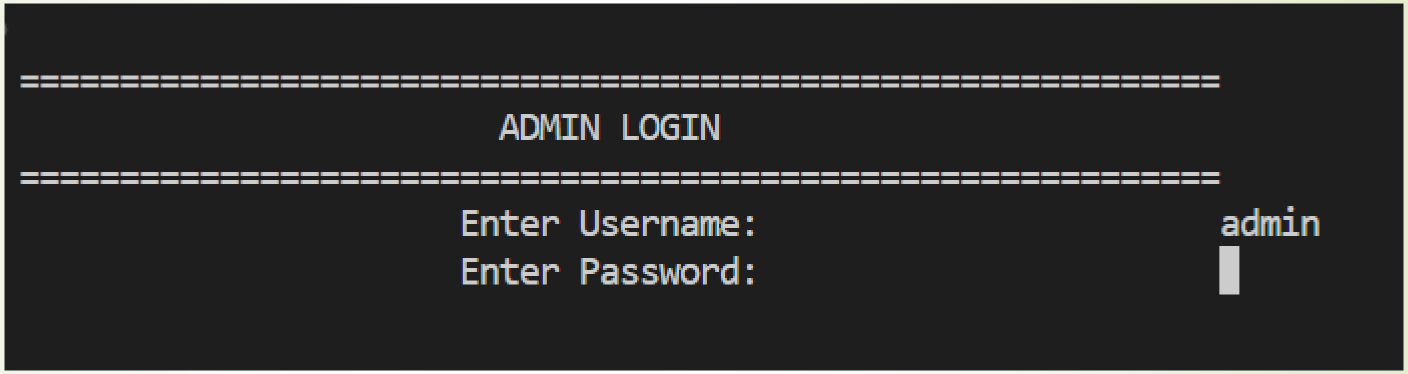
Employee Functionalities

- • Employees can register and manage their details.
- • Employees have the following actions:
-     - Register with name, age, department, username, password.
-     - Login to access their details.
-     - View and update their own details.

Database Integration

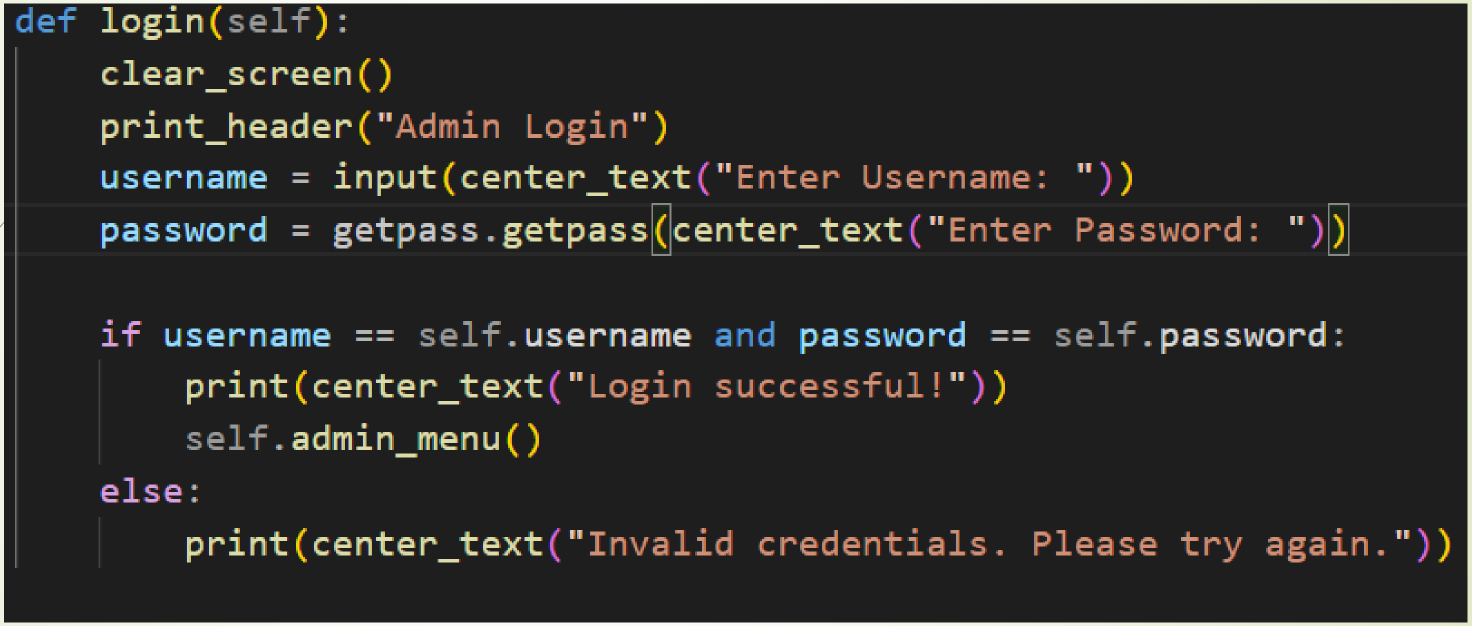
- • MySQL is used to store employee details in a relational database.
- • Database schema for employees includes fields:
-     - ID, Name, Age, Department, Username, Password.
- • All CRUD operations interact with MySQL tables via SQL queries.

Admin Login



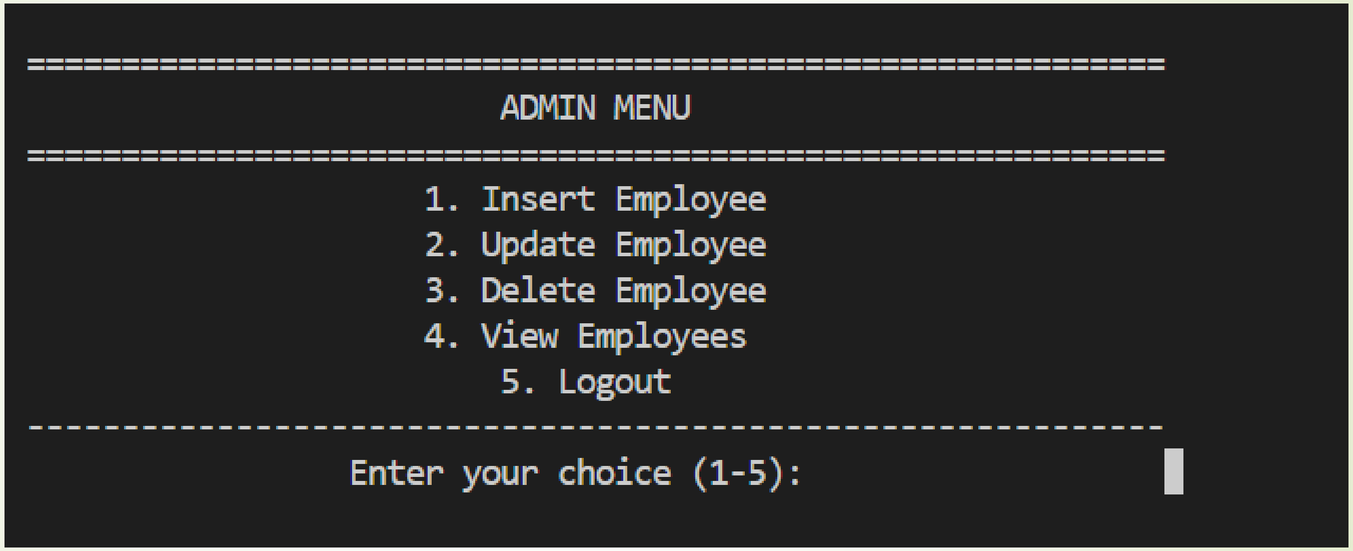
```
==========================================================================
                              ADMIN LOGIN
==========================================================================
                            Enter Username:                          admin
                            Enter Password:
```

Admin Login Code



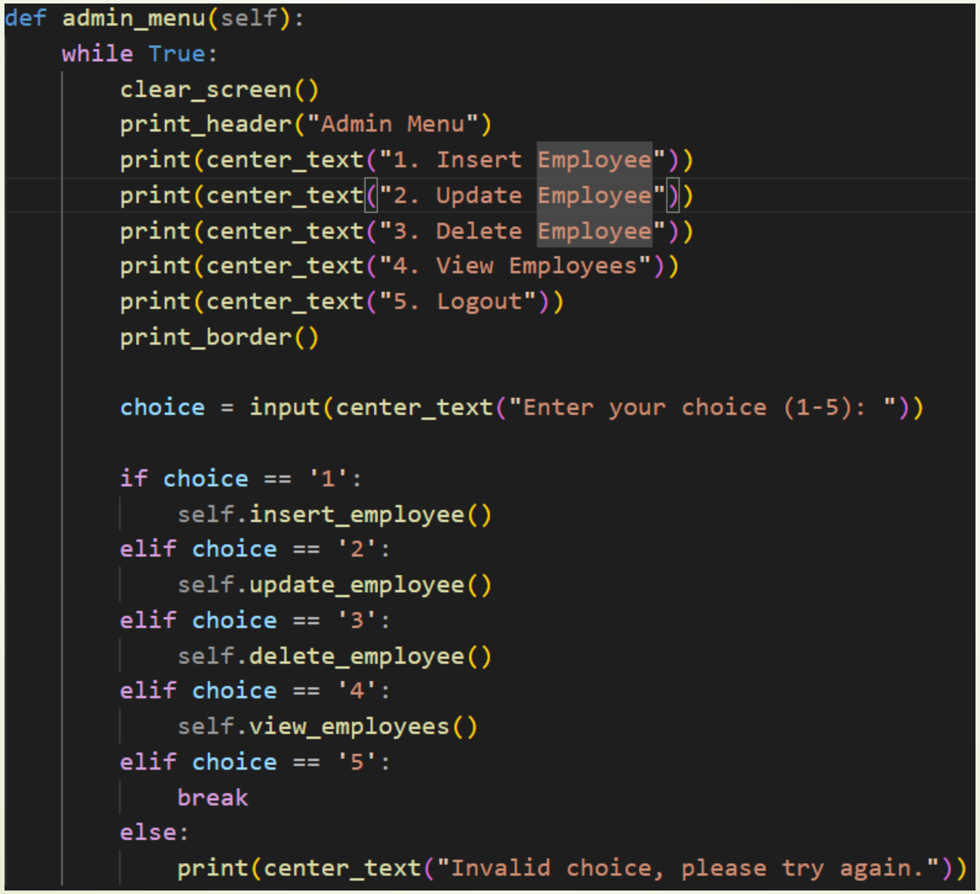
```python
def login(self):
    clear_screen()
    print_header("Admin Login")
    username = input(center_text("Enter Username: "))
    password = getpass.getpass(center_text("Enter Password: "))

    if username == self.username and password == self.password:
        print(center_text("Login successful!"))
        self.admin_menu()
    else:
        print(center_text("Invalid credentials. Please try again."))
```

Admin Menu



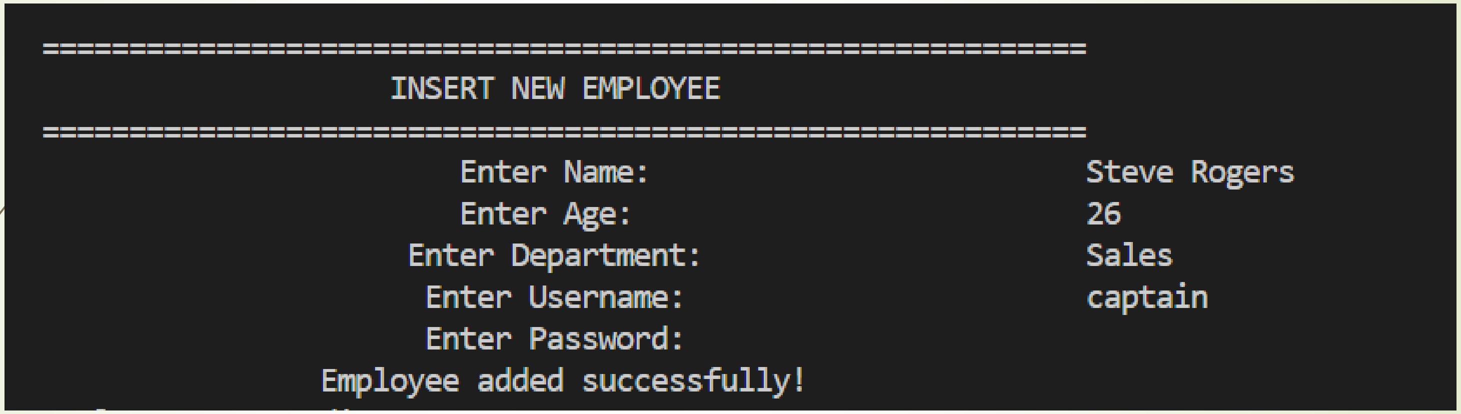
```
============================================================
                        ADMIN MENU
============================================================
                    1. Insert Employee
                    2. Update Employee
                    3. Delete Employee
                    4. View Employees
                        5. Logout
------------------------------------------------------------
                Enter your choice (1-5):
```

Admin Menu Code



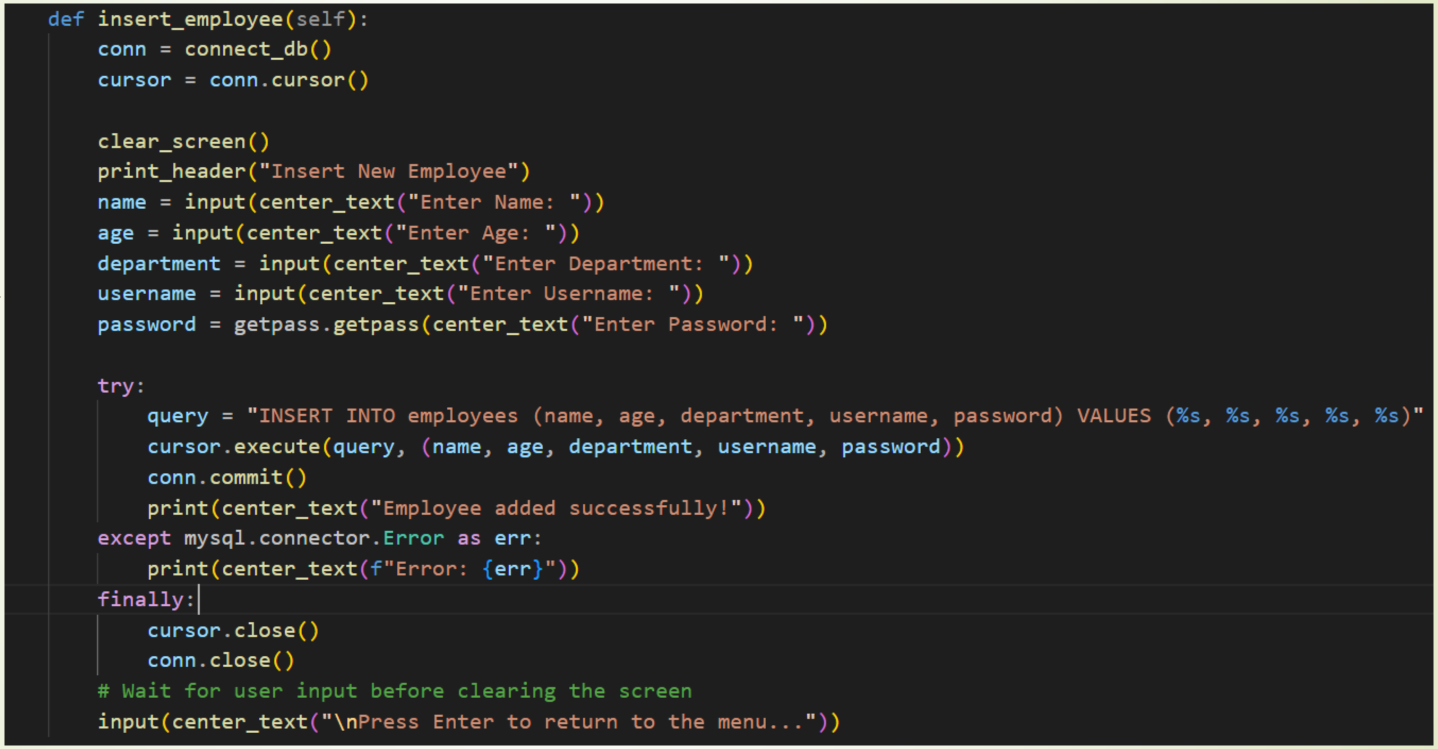
```
def admin_menu(self):
    while True:
        clear_screen()
        print_header("Admin Menu")
        print(center_text("1. Insert Employee"))
        print(center_text("2. Update Employee"))
        print(center_text("3. Delete Employee"))
        print(center_text("4. View Employees"))
        print(center_text("5. Logout"))
        print_border()

        choice = input(center_text("Enter your choice (1-5): "))

        if choice == '1':
            self.insert_employee()
        elif choice == '2':
            self.update_employee()
        elif choice == '3':
            self.delete_employee()
        elif choice == '4':
            self.view_employees()
        elif choice == '5':
            break
        else:
            print(center_text("Invalid choice, please try again."))
```

Inserting Employee



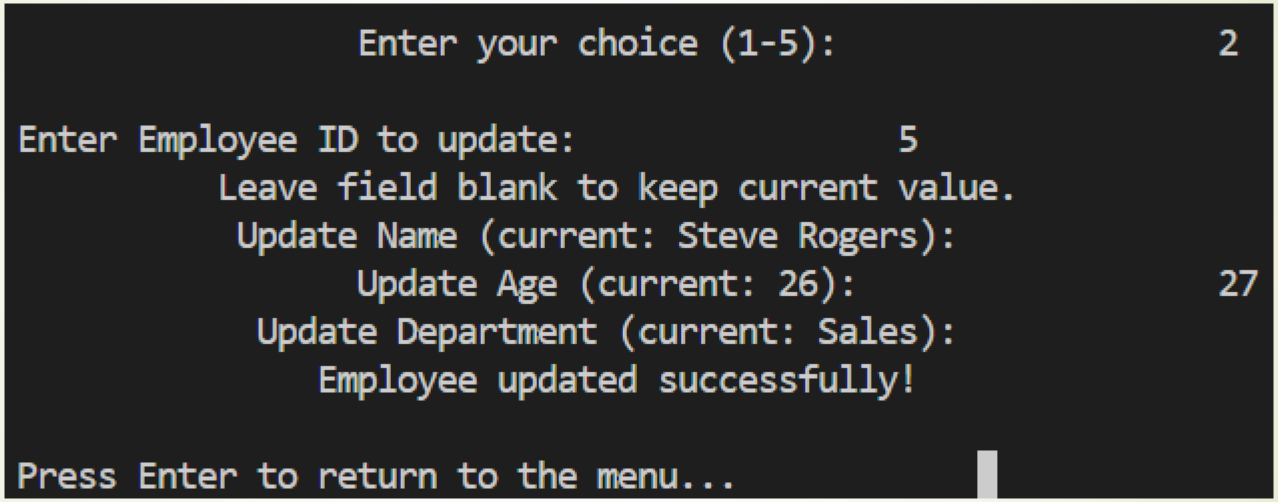
```
============================================================
                    INSERT NEW EMPLOYEE
============================================================
                        Enter Name:                         Steve Rogers
                        Enter Age:                          26
                     Enter Department:                      Sales
                      Enter Username:                       captain
                      Enter Password:
                  Employee added successfully!
```

Inserting Employee Code

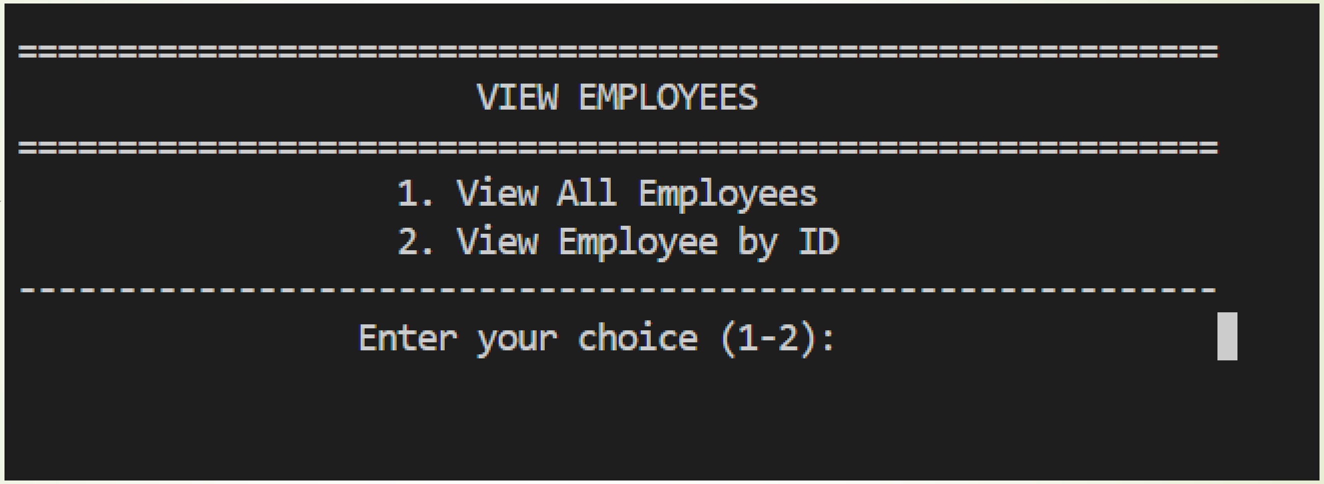


```python
def insert_employee(self):
    conn = connect_db()
    cursor = conn.cursor()

    clear_screen()
    print_header("Insert New Employee")
    name = input(center_text("Enter Name: "))
    age = input(center_text("Enter Age: "))
    department = input(center_text("Enter Department: "))
    username = input(center_text("Enter Username: "))
    password = getpass.getpass(center_text("Enter Password: "))

    try:
        query = "INSERT INTO employees (name, age, department, username, password) VALUES (%s, %s, %s, %s, %s)"
        cursor.execute(query, (name, age, department, username, password))
        conn.commit()
        print(center_text("Employee added successfully!"))
    except mysql.connector.Error as err:
        print(center_text(f"Error: {err}"))
    finally:
        cursor.close()
        conn.close()
    # Wait for user input before clearing the screen
    input(center_text("\nPress Enter to return to the menu..."))
```

Updating Employee

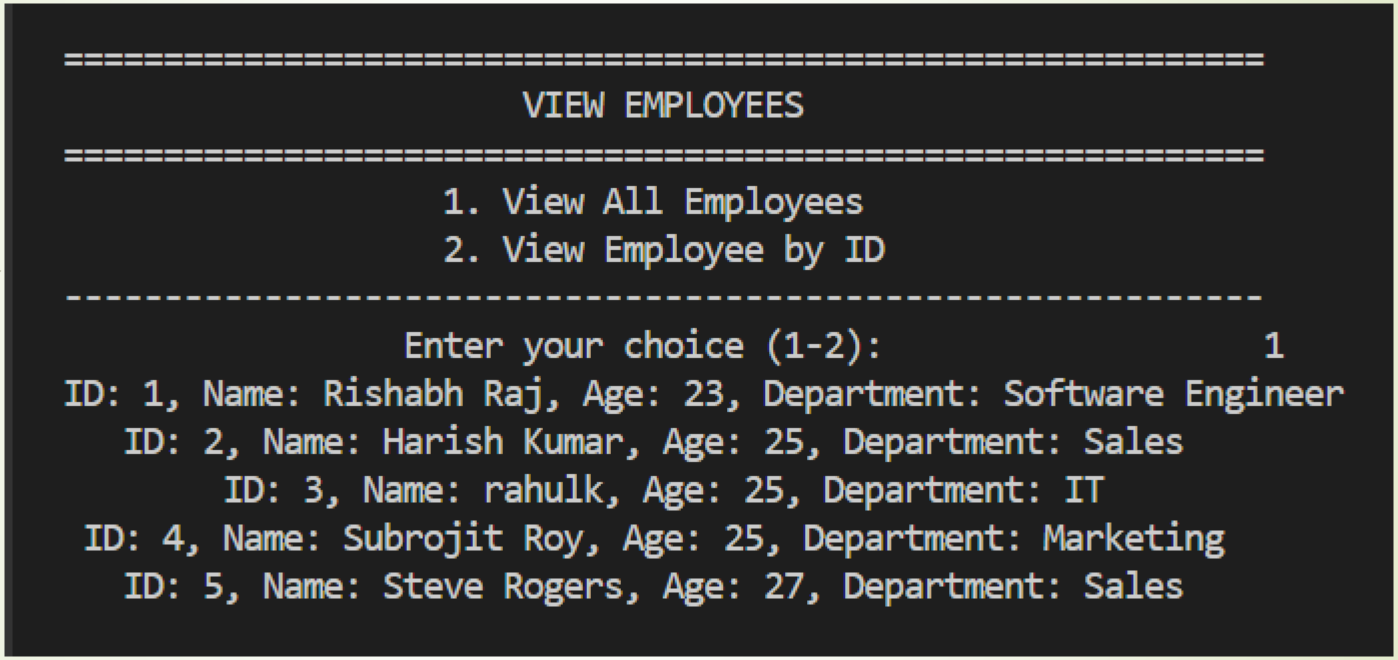

```
                 Enter your choice (1-5):                    2

Enter Employee ID to update:                5
          Leave field blank to keep current value.
           Update Name (current: Steve Rogers):
                 Update Age (current: 26):                   27
            Update Department (current: Sales):
               Employee updated successfully!

Press Enter to return to the menu...
```

View Employee Details



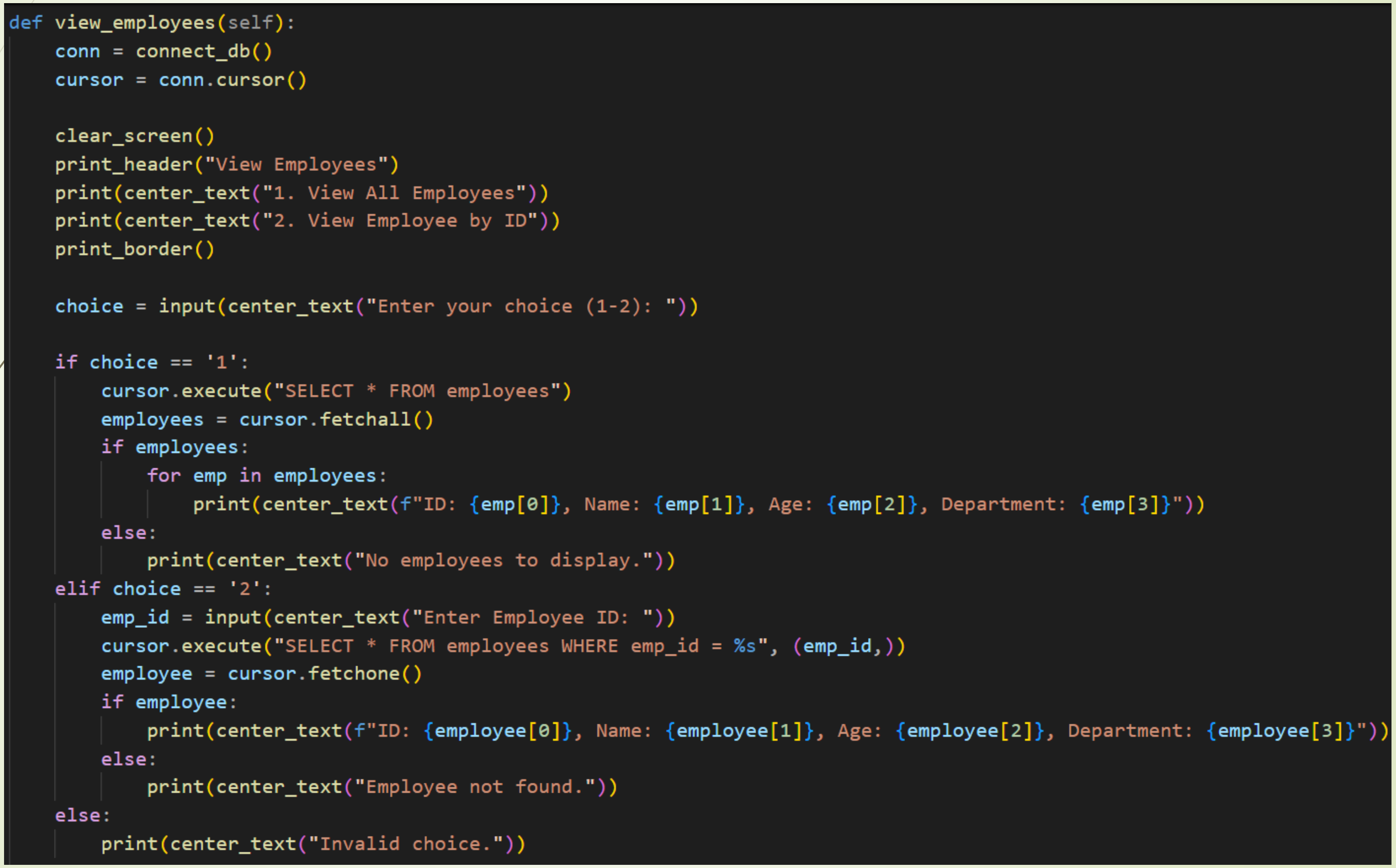
```
============================================================
                      VIEW EMPLOYEES
============================================================
                 1. View All Employees
                 2. View Employee by ID
------------------------------------------------------------
                Enter your choice (1-2):
```

View All Employee Details



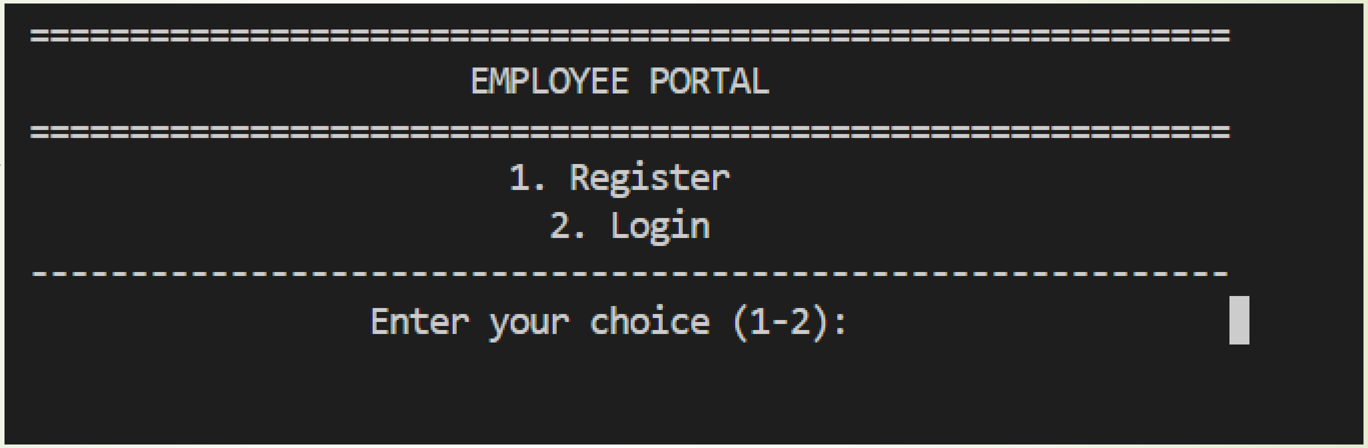
```
============================================================
                      VIEW EMPLOYEES
============================================================
                  1. View All Employees
                  2. View Employee by ID
------------------------------------------------------------
                Enter your choice (1-2):                  1
ID: 1, Name: Rishabh Raj, Age: 23, Department: Software Engineer
   ID: 2, Name: Harish Kumar, Age: 25, Department: Sales
        ID: 3, Name: rahulk, Age: 25, Department: IT
 ID: 4, Name: Subrojit Roy, Age: 25, Department: Marketing
   ID: 5, Name: Steve Rogers, Age: 27, Department: Sales
```

View Employee Details Code



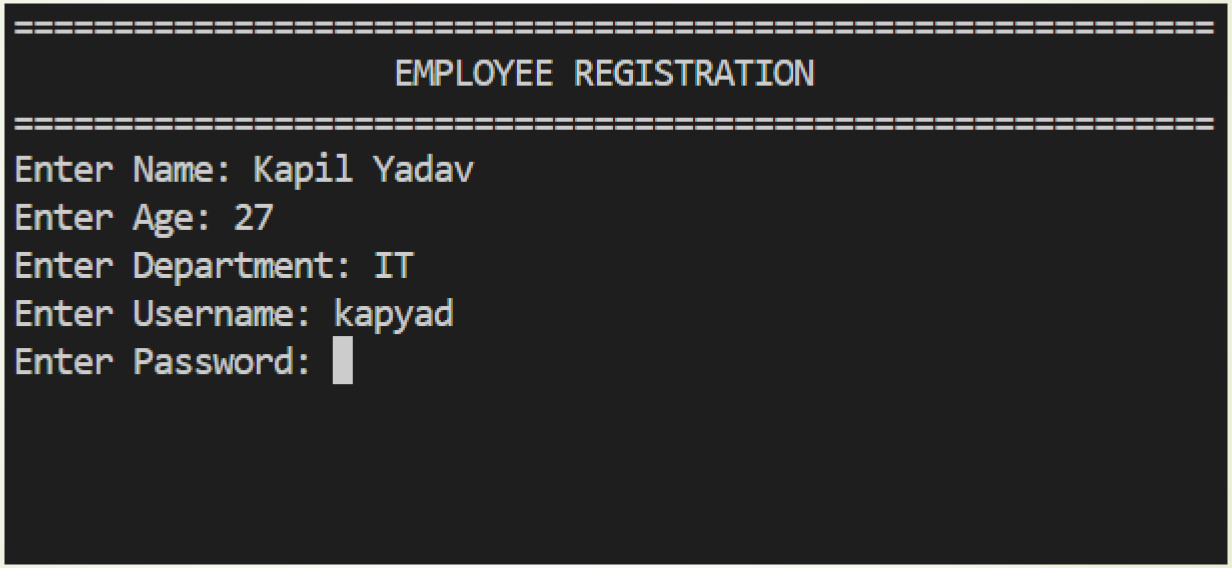
```python
def view_employees(self):
    conn = connect_db()
    cursor = conn.cursor()

    clear_screen()
    print_header("View Employees")
    print(center_text("1. View All Employees"))
    print(center_text("2. View Employee by ID"))
    print_border()

    choice = input(center_text("Enter your choice (1-2): "))

    if choice == '1':
        cursor.execute("SELECT * FROM employees")
        employees = cursor.fetchall()
        if employees:
            for emp in employees:
                print(center_text(f"ID: {emp[0]}, Name: {emp[1]}, Age: {emp[2]}, Department: {emp[3]}"))
        else:
            print(center_text("No employees to display."))
    elif choice == '2':
        emp_id = input(center_text("Enter Employee ID: "))
        cursor.execute("SELECT * FROM employees WHERE emp_id = %s", (emp_id,))
        employee = cursor.fetchone()
        if employee:
            print(center_text(f"ID: {employee[0]}, Name: {employee[1]}, Age: {employee[2]}, Department: {employee[3]}"))
        else:
            print(center_text("Employee not found."))
    else:
        print(center_text("Invalid choice."))
```

Employee Portal



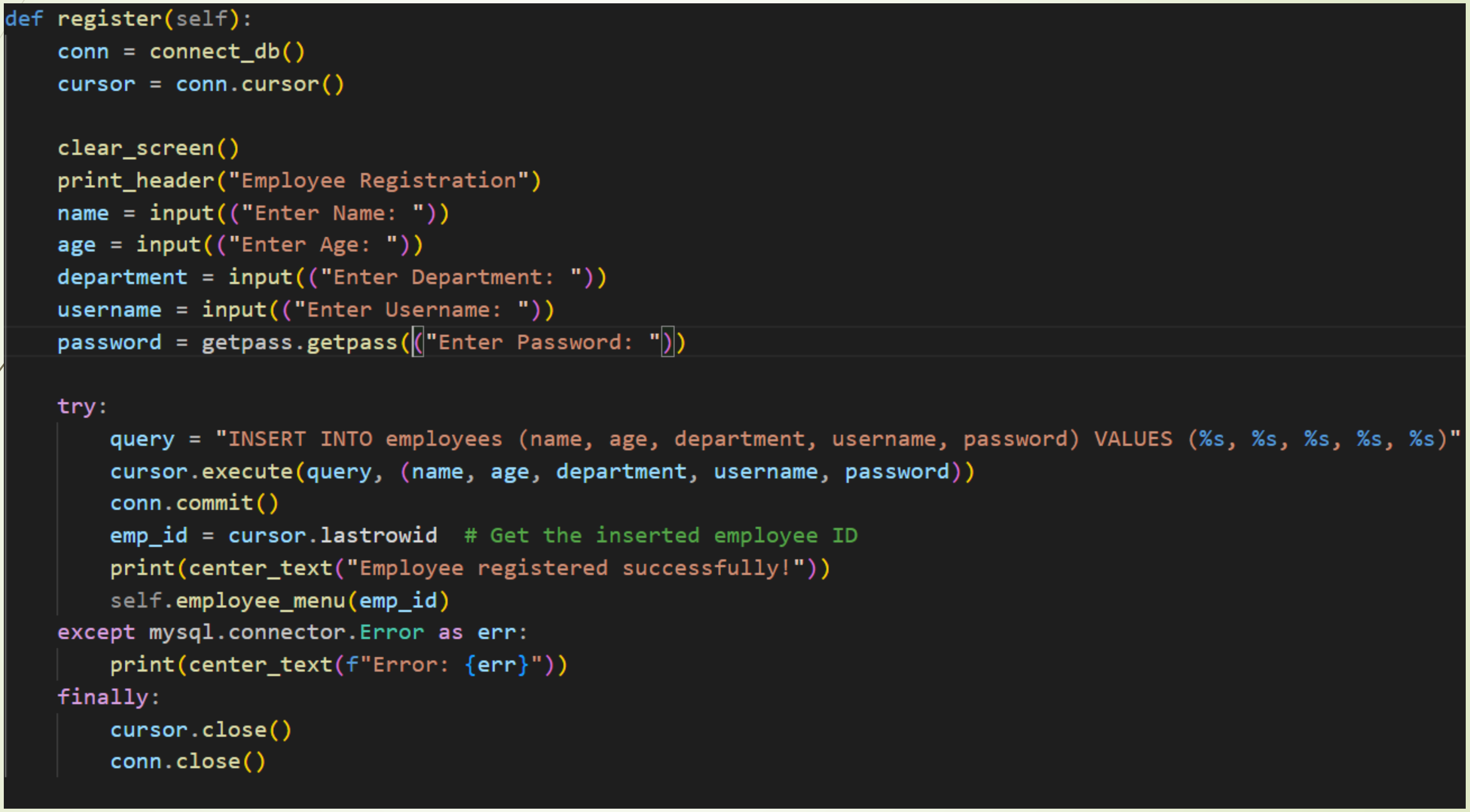
```
============================================================
                      EMPLOYEE PORTAL
============================================================
                       1. Register
                         2. Login
------------------------------------------------------------
                Enter your choice (1-2):
```

Employee Registration



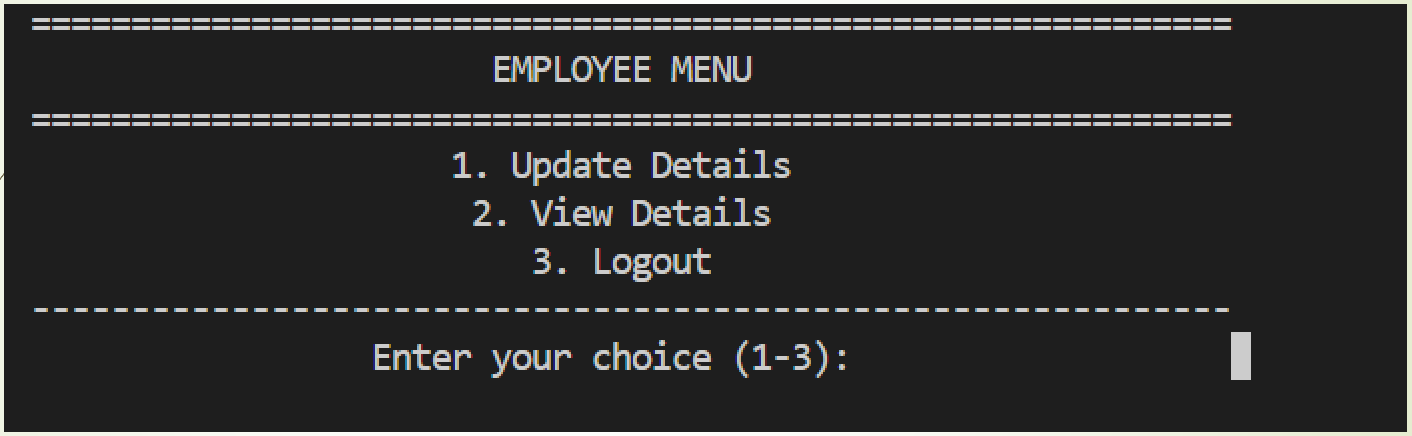
```
============================================================
                  EMPLOYEE REGISTRATION
============================================================
Enter Name: Kapil Yadav
Enter Age: 27
Enter Department: IT
Enter Username: kapyad
Enter Password:
```

Employee Registration Code



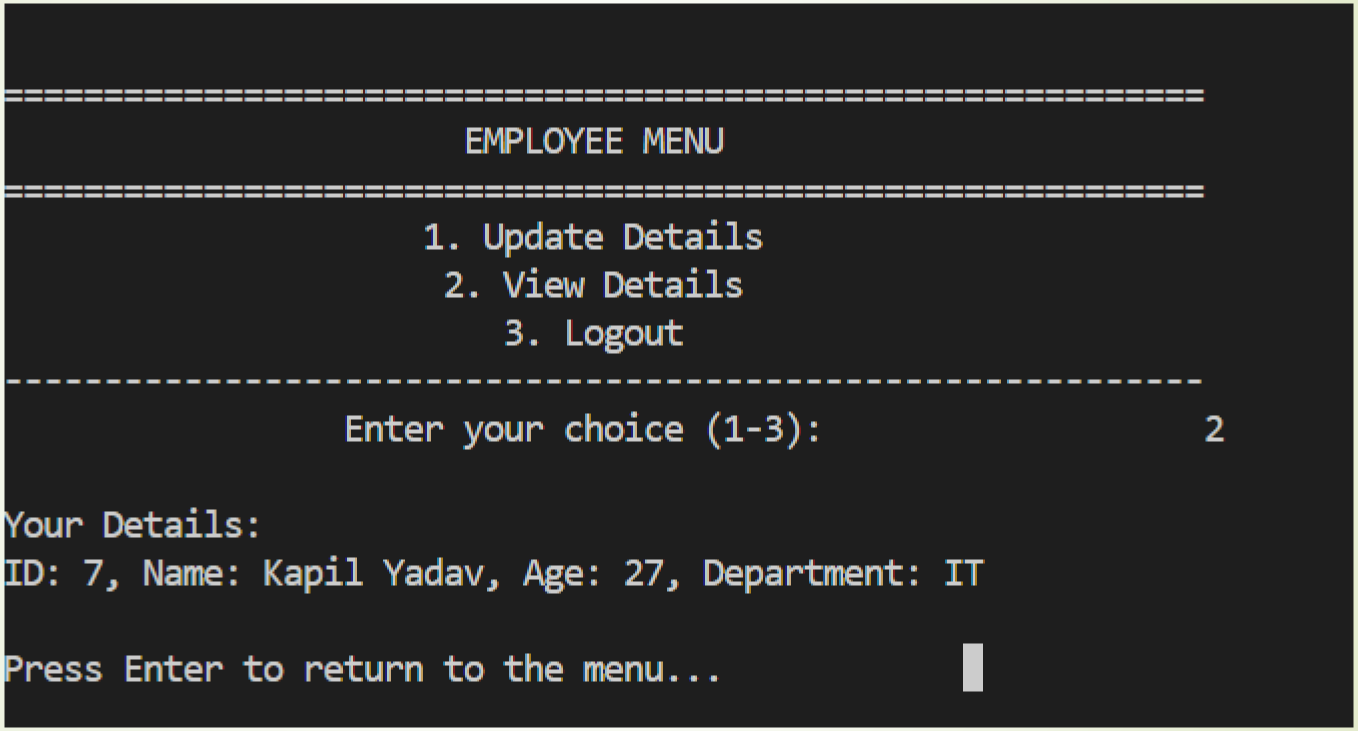
```python
def register(self):
    conn = connect_db()
    cursor = conn.cursor()

    clear_screen()
    print_header("Employee Registration")
    name = input(("Enter Name: "))
    age = input(("Enter Age: "))
    department = input(("Enter Department: "))
    username = input(("Enter Username: "))
    password = getpass.getpass(("Enter Password: "))

    try:
        query = "INSERT INTO employees (name, age, department, username, password) VALUES (%s, %s, %s, %s, %s)"
        cursor.execute(query, (name, age, department, username, password))
        conn.commit()
        emp_id = cursor.lastrowid  # Get the inserted employee ID
        print(center_text("Employee registered successfully!"))
        self.employee_menu(emp_id)
    except mysql.connector.Error as err:
        print(center_text(f"Error: {err}"))
    finally:
        cursor.close()
        conn.close()
```

Employee Menu



```
============================================================
                       EMPLOYEE MENU
============================================================
                    1. Update Details
                     2. View Details
                        3. Logout
------------------------------------------------------------
                Enter your choice (1-3):
```

Employee's own details



```
============================================================
                       EMPLOYEE MENU
============================================================
                     1. Update Details
                      2. View Details
                         3. Logout
------------------------------------------------------------
                 Enter your choice (1-3):                   2

Your Details:
ID: 7, Name: Kapil Yadav, Age: 27, Department: IT

Press Enter to return to the menu...
```

Conclusion

- • The Employee Management System is a robust application for admin and employees.
- • It leverages Python for backend operations and MySQL for persistent data storage.
- • The system is easily extendable and suitable for small to mid-sized organizations.

Thank You!