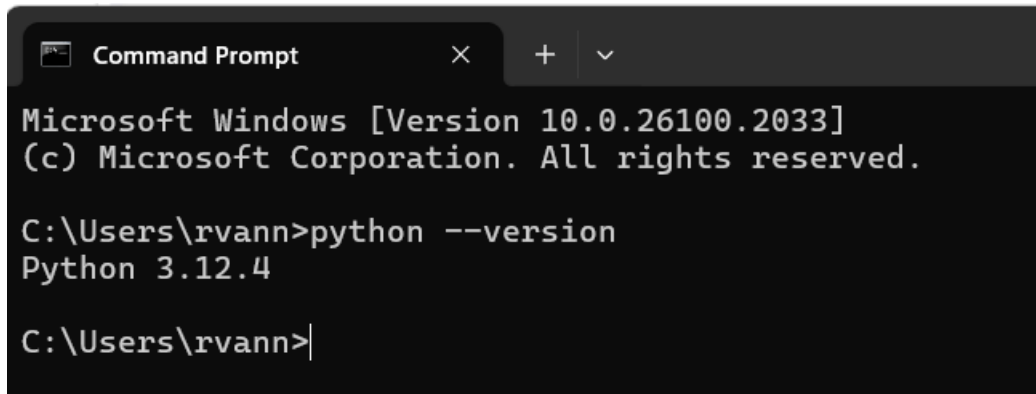


Python-project

- 1) Install MySQL Workbench 8.0 CE
- 2) Install Python 3.12.4, **IDLE (Integrated Development and Learning Environment)** is included by default.
- 3) After installation of Workbench go to **C:\Program Files\MySQL\MySQL Server 8.0\bin** paste this path in environment variable and save.
- 4) **Open cmd run below commands**



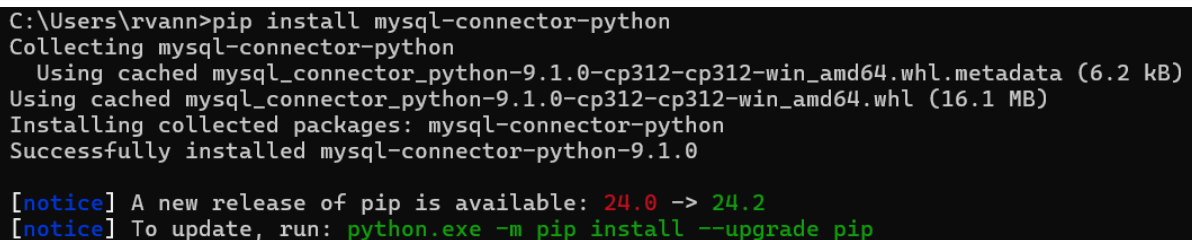
```
Command Prompt
Microsoft Windows [Version 10.0.26100.2033]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rvann>python --version
Python 3.12.4

C:\Users\rvann>|
```

- 5) Install **MySQL Connector**

The command `pip install mysql-connector-python` installs the **MySQL Connector for Python**, which is a library that allows Python programs to interact with a MySQL database. This package enables Python to connect to MySQL, execute queries, and retrieve or modify data within the database directly.



```
C:\Users\rvann>pip install mysql-connector-python
Collecting mysql-connector-python
  Using cached mysql_connector_python-9.1.0-cp312-cp312-win_amd64.whl.metadata (6.2 kB)
Using cached mysql_connector_python-9.1.0-cp312-cp312-win_amd64.whl (16.1 MB)
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.1.0

[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

Create database employee_system

The screenshot shows a SQL query editor with the following content:

```
1  
2 • CREATE DATABASE employee_system;  
3
```

Below the query editor is the 'Output' pane, which is set to 'Action Output'. It displays a single successful action:

#	Time	Action
✓ 1	13:35:06	CREATE DATABASE employee_system

The screenshot shows a SQL query editor with the following content:

```
1 • USE employee_system;  
2  
3
```

Below the query editor is the 'Output' pane, which is set to 'Action Output'. It displays two successful actions:

#	Time	Action
✓ 1	13:35:06	CREATE DATABASE employee_system
✓ 2	13:36:03	USE employee_system

Create table users

The screenshot shows a SQL query editor with the following content:

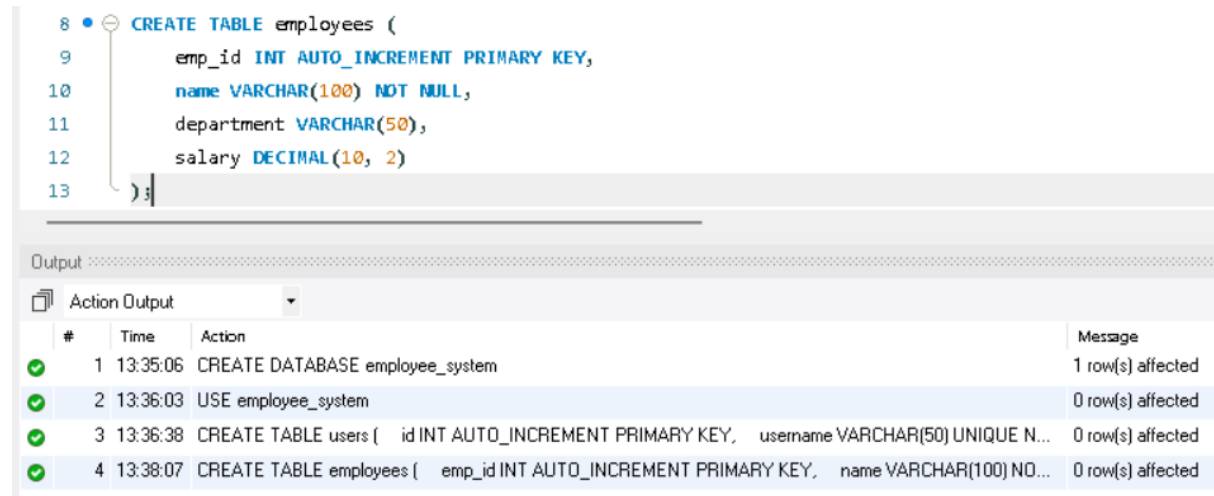
```
1 • USE employee_system;  
2 • CREATE TABLE users (  
3     id INT AUTO_INCREMENT PRIMARY KEY,  
4     username VARCHAR(50) UNIQUE NOT NULL,  
5     password VARCHAR(255) NOT NULL,  
6     email VARCHAR(100) UNIQUE NOT NULL  
7 ) ;  
8
```

Below the query editor is the 'Output' pane, which is set to 'Action Output'. It displays three successful actions:

#	Time	Action	Message
✓ 1	13:35:06	CREATE DATABASE employee_system	1 row(s) affected
✓ 2	13:36:03	USE employee_system	0 row(s) affected
✓ 3	13:36:38	CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(50) UNIQUE N...	0 row(s) affected

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

Create table employees



The screenshot shows a SQL query editor with the following code:

```
8 CREATE TABLE employees (  
9     emp_id INT AUTO_INCREMENT PRIMARY KEY,  
10     name VARCHAR(100) NOT NULL,  
11     department VARCHAR(50),  
12     salary DECIMAL(10, 2)  
13 );
```

Below the editor is an 'Output' window showing a table of database actions:

#	Time	Action	Message
✓ 1	13:35:06	CREATE DATABASE employee_system	1 row(s) affected
✓ 2	13:36:03	USE employee_system	0 row(s) affected
✓ 3	13:36:38	CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(50) UNIQUE N...	0 row(s) affected
✓ 4	13:38:07	CREATE TABLE employees (emp_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NO...	0 row(s) affected

PYTHON CODE

```
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()
```

Function to hash passwords for security

```
def hash_password(password):  
    return hashlib.sha256(password.encode()).hexdigest()
```

User Registration

```
def register_user():  
    username = input("Enter username: ")  
    password = input("Enter password: ")
```

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

```
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 Login

```
def login_user():
```

```
    username = input("Enter username: ")
```

```
    password = input("Enter password: ")
```

```
    hashed_password = hash_password(password)
```

```
    cursor.execute(
```

```
        "SELECT * FROM users WHERE username = %s AND password = %s",
```

```
        (username, hashed_password)
```

```
    )
```

```
    user = cursor.fetchone()
```

```
    if user:
```

```
        print(f"Welcome, {username}!")
```

```
        employee_menu()
```

```
    else:
```

```
        print("Invalid credentials. Please try again.")
```

Add Employee

```
def add_employee():
```

```
    name = input("Enter employee name: ")
```

```
    department = input("Enter department: ")
```

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

```
salary = float(input("Enter salary: "))
```

```
cursor.execute(
```

```
    "INSERT INTO employees (name, department, salary) VALUES (%s, %s, %s)",
```

```
    (name, department, salary)
```

```
)
```

```
conn.commit()
```

```
print("Employee added successfully!")
```

View Employees

```
def view_employees():
```

```
    cursor.execute("SELECT * FROM employees")
```

```
    employees = cursor.fetchall()
```

```
    if employees:
```

```
        print("\nEmployee List:")
```

```
        for emp in employees:
```

```
            print(f"ID: {emp[0]}, Name: {emp[1]}, Department: {emp[2]}, Salary: {emp[3]}")
```

```
    else:
```

```
        print("No employees found.")
```

Update Employee

```
def update_employee():
```

```
    emp_id = int(input("Enter employee ID to update: "))
```

```
    name = input("Enter new name: ")
```

```
    department = input("Enter new department: ")
```

```
    salary = float(input("Enter new salary: "))
```

```
    cursor.execute(
```

```
        "UPDATE employees SET name = %s, department = %s, salary = %s WHERE emp_id = %s",
```

```
        (name, department, salary, emp_id)
```

```
)
```

```
    conn.commit()
```

```
    print("Employee updated successfully!")
```

Delete Employee

```
def delete_employee():
```

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

```
emp_id = int(input("Enter employee ID to delete: "))
```

```
cursor.execute("DELETE FROM employees WHERE emp_id = %s", (emp_id,))
```

```
conn.commit()
```

```
print("Employee deleted successfully!")
```

Employee Management Menu

```
def employee_menu():
```

```
    while True:
```

```
        print("\nEmployee Management Menu:")
```

```
        print("1. Add Employee")
```

```
        print("2. View Employees")
```

```
        print("3. Update Employee")
```

```
        print("4. Delete Employee")
```

```
        print("5. Logout")
```

```
        choice = input("Enter your choice: ")
```

```
        if choice == '1':
```

```
            add_employee()
```

```
        elif choice == '2':
```

```
            view_employees()
```

```
        elif choice == '3':
```

```
            update_employee()
```

```
        elif choice == '4':
```

```
            delete_employee()
```

```
        elif choice == '5':
```

```
            print("Logged out.")
```

```
            break
```

```
        else:
```

```
            print("Invalid choice. Please try again.")
```

Main Menu

```
def main_menu():
```

```
    while True:
```

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

```
print("\nMain Menu:")

print("1. Register")

print("2. Login")

print("3. Exit")

choice = input("Enter your choice: ")

if choice == '1':

    register_user()

elif choice == '2':

    login_user()

elif choice == '3':

    print("Exiting...")

    break

else:

    print("Invalid choice. Please try again.")
```

Run the Program

```
if __name__ == '__main__':

    main_menu()
```

Close the database connection when the program ends

```
conn.close()
```

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

new.py - C:\Users\rwann\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()

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

# 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 Login
def login_user():
    username = input("Enter username: ")
    password = input("Enter password: ")
    hashed_password = hash_password(password)

    cursor.execute(
        "SELECT * FROM users WHERE username = %s AND password = %s",
        (username, hashed_password)
    )
    user = cursor.fetchone()
```


Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

```
new.py - C:\Users\rwann\AppData\Local\Programs\Python\Python312\new.py (3.12.4)
File Edit Format Run Options Window Help

if user:
    print(f"Welcome, {username}!")
    employee_menu()
else:
    print("Invalid credentials. Please try again.")

# Add Employee
def add_employee():
    name = input("Enter employee name: ")
    department = input("Enter department: ")
    salary = float(input("Enter salary: "))

    cursor.execute(
        "INSERT INTO employees (name, department, salary) VALUES (%s, %s, %s)",
        (name, department, salary)
    )
    conn.commit()
    print("Employee added successfully!")

# View Employees
def view_employees():
    cursor.execute("SELECT * FROM employees")
    employees = cursor.fetchall()

    if employees:
        print("\nEmployee List:")
        for emp in employees:
            print(f"ID: {emp[0]}, Name: {emp[1]}, Department: {emp[2]}, Salary: {emp[3]}")
    else:
        print("No employees found.")

# Update Employee
def update_employee():
    emp_id = int(input("Enter employee ID to update: "))
    name = input("Enter new name: ")
    department = input("Enter new department: ")
    salary = float(input("Enter new salary: "))

    cursor.execute(
        "UPDATE employees SET name = %s, department = %s, salary = %s WHERE emp_id = %s",
        (name, department, salary, emp_id)
    )
    conn.commit()
    print("Employee updated successfully!")
```

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

```
# Delete Employee
def delete_employee():
    emp_id = int(input("Enter employee ID to delete: "))
    cursor.execute("DELETE FROM employees WHERE emp_id = %s", (emp_id,))
    conn.commit()
    print("Employee deleted successfully!")

# Employee Management Menu
def employee_menu():
    while True:
        print("\nEmployee Management Menu:")
        print("1. Add Employee")
        print("2. View Employees")
        print("3. Update Employee")
        print("4. Delete Employee")
        print("5. Logout")

        choice = input("Enter your choice: ")

        if choice == '1':
            add_employee()
        elif choice == '2':
            view_employees()
        elif choice == '3':
            update_employee()
        elif choice == '4':
            delete_employee()
        elif choice == '5':
            print("Logged out.")
            break
        else:
            print("Invalid choice. Please try again.")
```

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

```
# Main Menu
def main_menu():
    while True:
        print("\nMain Menu:")
        print("1. Register")
        print("2. Login")
        print("3. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            register_user()
        elif choice == '2':
            login_user()
        elif choice == '3':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Please try again.")

# Run the Program
if __name__ == '__main__':
    main_menu()

# Close the database connection when the program ends
conn.close()
```

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

The output will appear in the **Python Shell** window in IDLE.

Register User

```
*IDLE Shell 3.12.4*
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bi
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\rvann\AppData\Local\Programs\Python\Python312\new.py =

Main Menu:
1. Register
2. Login
3. Exit
Enter your choice: 1
Enter username: Shruti
Enter password: 123@
Enter email: shrutianandas123@gmail.com
User registered successfully!

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: |
```

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

Add Employee 1

```
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!
```

Add Employee 2

```
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!
```

Add Employee 3

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 1
Enter employee name: Preeti
Enter department: UI/UX
Enter salary: 30000
Employee added successfully!
```

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

Add Employee 4

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 1
Enter employee name: Sejal
Enter department: Docter
Enter salary: 560000
Employee added successfully!
```

Add Employee 5

```
Employee Management Menu:
1. Add Employee
2. View Employees
3. Update Employee
4. Delete Employee
5. Logout
Enter your choice: 1
Enter employee name: Nandini
Enter department: Events
Enter salary: 45000
Employee added successfully!
```

View Employees

```
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
```

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

Output

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

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

Navigator

SCHEMAS

Filter objects

▶ chicken_coop

▼ employee_system

▶ Tables

▶ Views

▶ Stored Procedures

▶ Functions

▶ mirraw

▶ pet_shop

▶ sakila

▶ sys

▶ world

Administration Schemas

Information

No object selected

Query 1 x mirraw mirraw SQL File 5*

Limit to 1000 rows

14

15

16 • select*from employees;

17

18

Result Grid

Filter Rows:

Edit:

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

employees 1 x

Output

Action Output

#	Time	Action	Message
✓ 1	13:35:06	CREATE DATABASE employee_system	1 row(s) affected
✓ 2	13:36:03	USE employee_system	0 row(s) affected
✓ 3	13:36:38	CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, username VARCHAR(50) UNIQUE N...	0 row(s) affected
✓ 4	13:38:07	CREATE TABLE employees (emp_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NO...	0 row(s) affected
✓ 5	13:47:51	select*from employees LIMIT 0, 1000	5 row(s) returned

Name: Shruti Ramesh Anandas

Employee ID: 207015

Employee Management System

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: Doctor, 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!

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

Output

```
16 • select*from employees;
```

```
17
```

```
18
```

Result Grid   Filter Rows: Edit:

	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
	5	Nandini	Events	45000.00
•	NULL	NULL	NULL	NULL

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

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!

Output

15

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

Name: Shruti Ramesh Anandas
Employee ID: 207015
Employee Management System

Logout

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

Exit

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

Users table output

```
14  
15  
16 • select*from users;  
17  
18
```

	id	username	password	email
▶	1	Shruti	c1d0e46fdeb2b72758a6a5bd5eecf2622ff8b84a...	shrutianandas123@gmail.com
*	NULL	NULL	NULL	NULL