1. Assignment(hands on): My First Streamlit app

i. Tools / Libraries Used

- **Python** – Core programming language
- **Streamlit** – Web app framework for building interactive applications
- **Pandas** – Data manipulation and DataFrame handling

ii. Project Description

The **Smart Interactive Streamlit Dashboard** is a multi-feature web application built using Streamlit.

This application demonstrates various interactive components such as:

- Sidebar course selection
- Profile display section
- User input handling (text input & number input)
- Buttons and checkbox interactions
- Programming language selection
- Session-based counter functionality
- DataFrame display using Pandas
- CSV file upload and preview
- Image display from URL

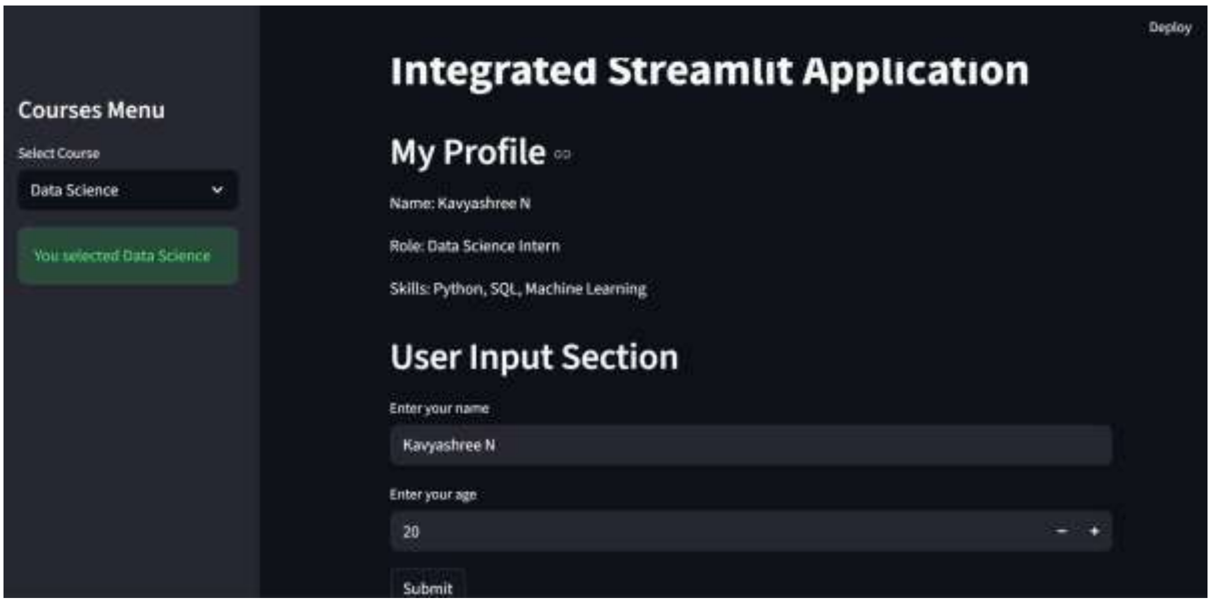
The project showcases how to build an **interactive, dynamic, and user-friendly web application** using Streamlit with minimal code. It integrates multiple UI elements and state management concepts, making it ideal for beginners learning web app development with Python.

iii. Project Code

```
import streamlit as st
import pandas as pd


#  TITLE
st.title("Integrated Streamlit Application")


# SIDEBAR MENU
st.sidebar.title("Courses Menu")
course = st.sidebar.selectbox(
    "Select Course",
    ["Data Science", "Full Stack Java", "Full Stack Python", "Dot Net"]
)

st.sidebar.success(f"You selected {course}")

#  PROFILE SECTION
st.header("My Profile")
st.write("Name: Kavyashree N")
st.write("Role: Data Science Intern")
st.write("Skills: Python, SQL, Machine Learning")

# - USER INPUT SECTION
st.header("User Input Section")

name = st.text_input("Enter your name")
age = st.number_input("Enter your age", min_value=0, max_value=100)

if st.button("Submit"):
    st.success("Button Clicked Successfully!")
    st.write(f"Hello {name}, you are {age} years old.")

# CHECKBOX SHOW/HIDE

if st.checkbox("Show Secret Message"):
    st.write("Welcome to Streamlit ")

#  SELECTBOX
st.header("Programming Language Selection")

language = st.selectbox(
    "Choose Programming Language",
    ["Python", "Java", "C++", "JavaScript"]
)

st.write(f"You selected: {language}")

# COUNTER
st.header("Simple Counter")

if "count" not in st.session_state:
    st.session_state.count = 0

if st.button("Increase Counter"):
    st.session_state.count += 1

st.write("Counter Value:", st.session_state.count)

#  DATAFRAME DISPLAY
st.header("Display Sample DataFrame")

data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Salary": [50000, 60000, 70000]
}

df = pd.DataFrame(data)
st.dataframe(df)

#  CSV FILE UPLOAD
st.header("Upload CSV File")

file = st.file_uploader("Upload a CSV file", type=["csv"])
```

```
if file is not None:
    uploaded_df = pd.read_csv(file)
    st.write("Uploaded Data:")
    st.dataframe(uploaded_df)

# IMAGE DISPLAY
st.header("Display Image")

st.image(
    "https://images.unsplash.com/photo-1519389950473-47ba0277781c",
    caption="Sample Image",
    use_container_width=True
)
```

iv. OUTPUT:



Integrated Streamlit Application

My Profile

Name: Kavyashree N

Role: Data Science Intern

Skills: Python, SQL, Machine Learning

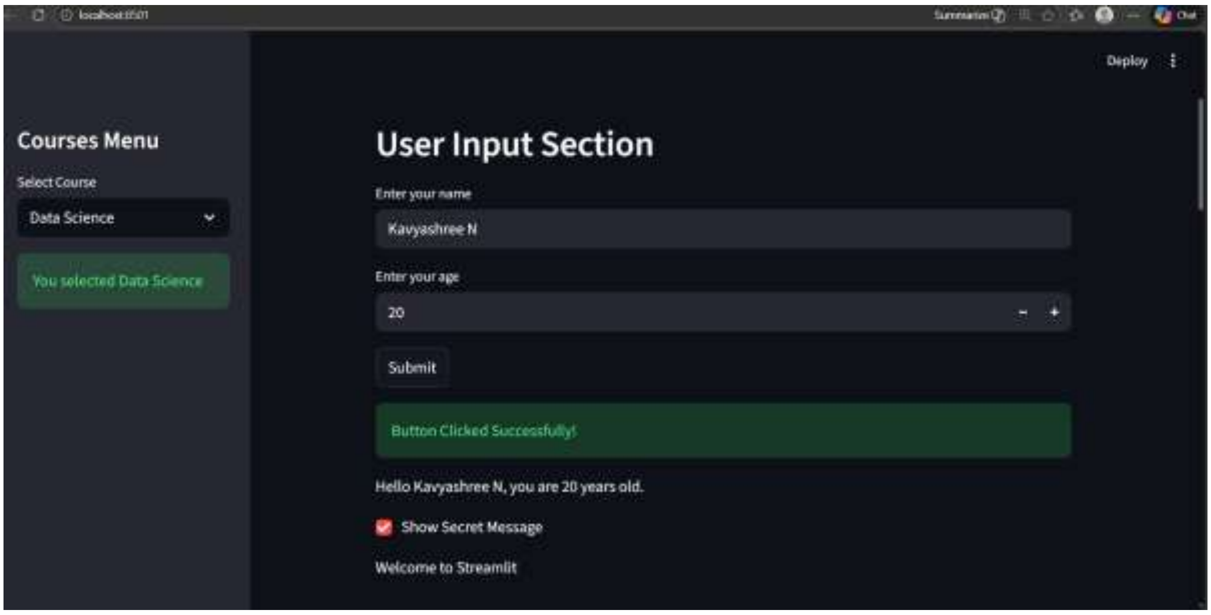
User Input Section

Enter your name

Kavyashree N

Enter your age

20



Courses Menu

Select Course

Data Science

You selected Data Science

User Input Section

Enter your name

Kavyashree N

Enter your age

20

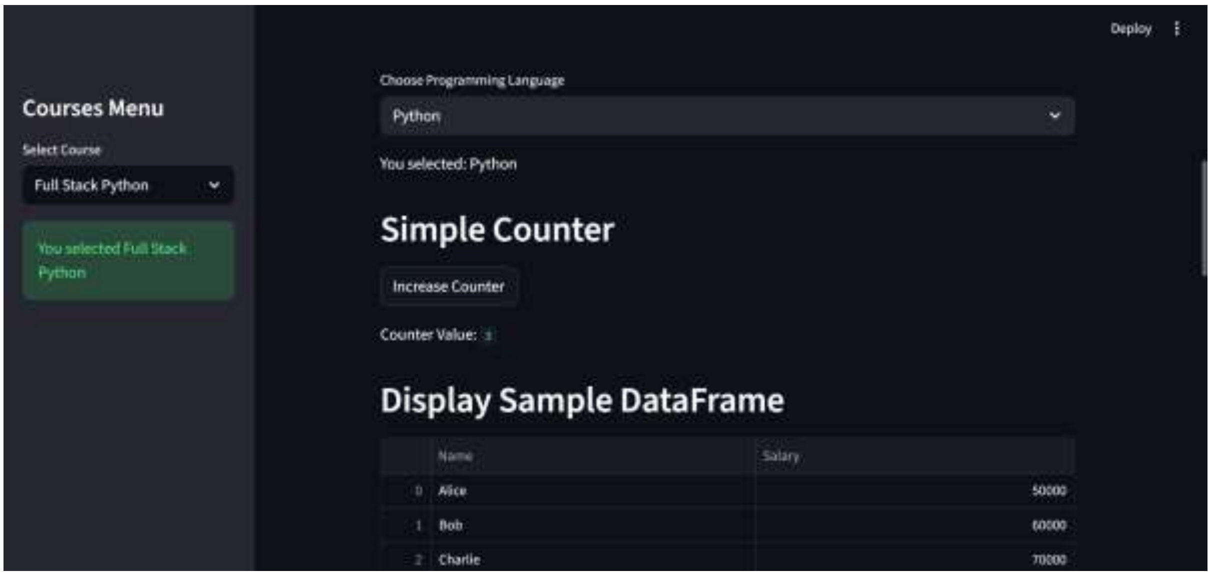
Submit

Button Clicked Successfully!

Hello Kavyashree N, you are 20 years old.

✓ Show Secret Message

Welcome to Streamlit



Courses Menu

Select Course

Full Stack Python

You selected Full Stack Python

Choose Programming Language

Python

You selected: Python

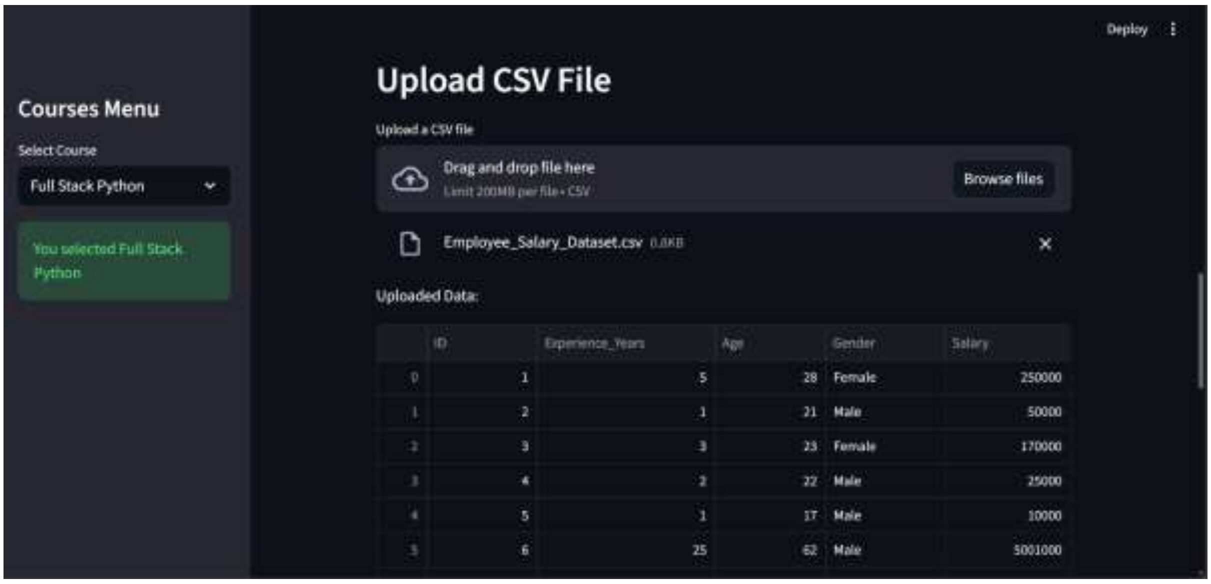
Simple Counter

Increase Counter

Counter Value: 1

Display Sample DataFrame

	Name	Salary
0	Alice	50000
1	Bob	60000
2	Charlie	70000



You selected Full Stack Python

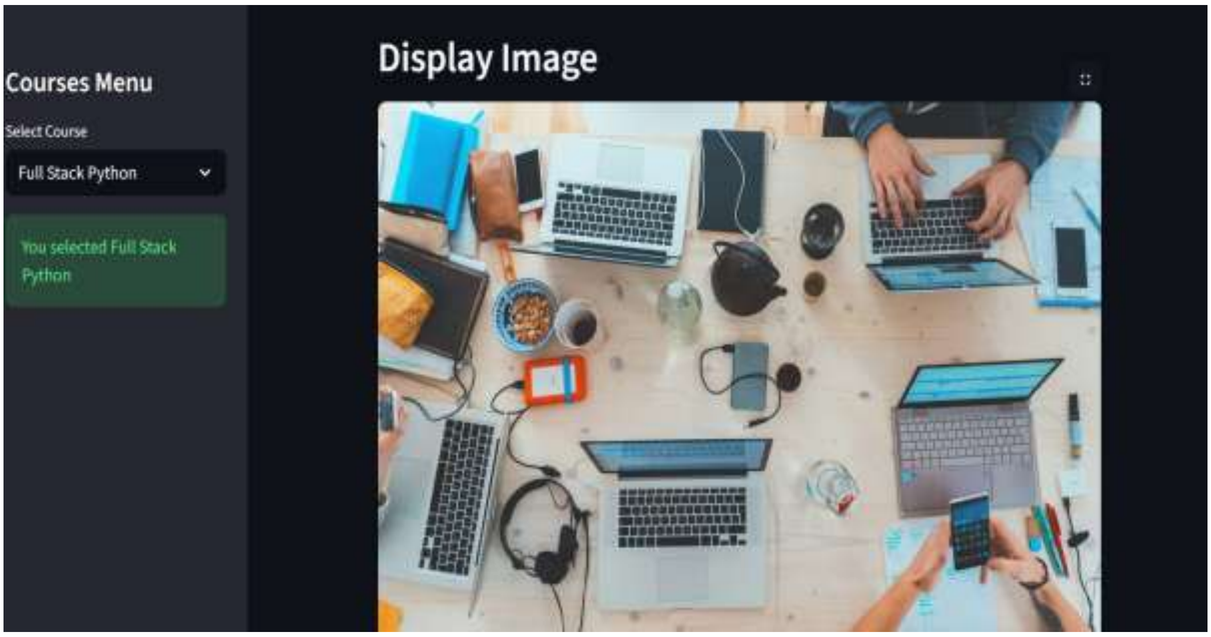
Upload CSV File

Upload a CSV file

Employee_Salary_Dataset.csv 0.8KB

Uploaded Data:

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5001000



You selected Full Stack Python

Display Image

2. Project Title : Role-Based Employee Salary Filter App

i. Tools / Libraries Used

- **Python** – Core programming language
- **Streamlit** – For building interactive web applications
- **Pandas** – For reading, filtering, and displaying CSV data

ii. Project Description

The **Role-Based Employee Salary Filter App** is an interactive Streamlit web application that allows users to:

1. **Select their role** (HR, Manager, Employee) from the sidebar.
2. View different sidebar messages based on their selected role.
3. Upload an employee CSV file.
4. Automatically filter and display employees whose salary is greater than ₹50,000.
5. Display the total number of employees matching the filter criteria.

Key Features:

- Role-based dynamic content display
- CSV file upload functionality
- Salary-based filtering using Pandas
- Conditional column validation
- Clean and interactive UI

This project demonstrates:

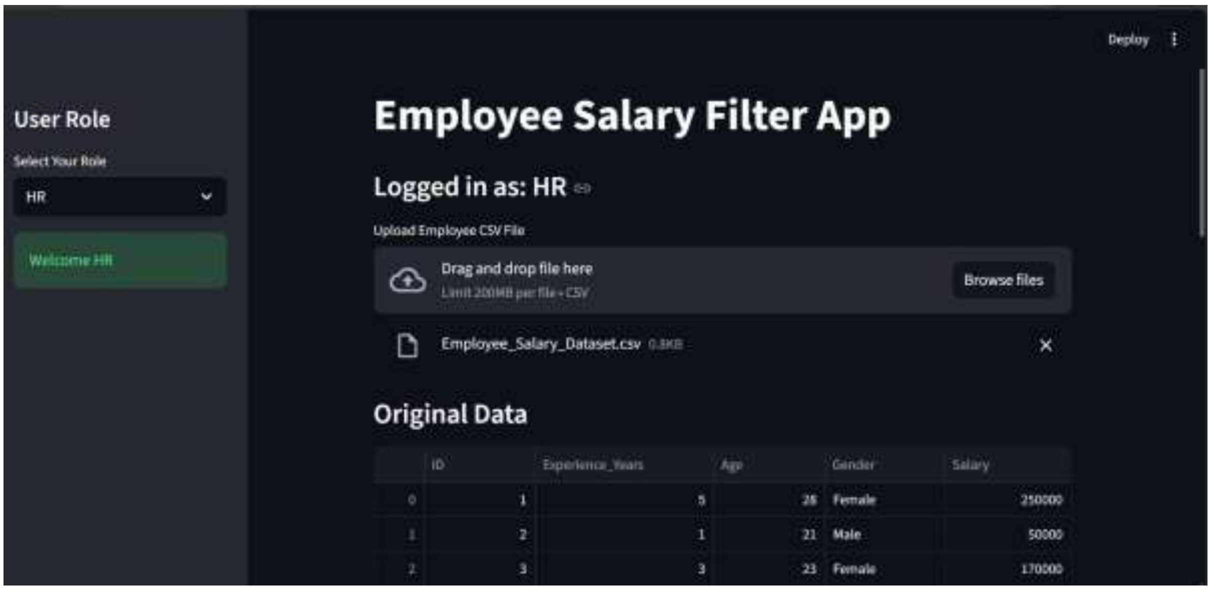
- Conditional rendering
- Data filtering
- File handling
- Role-based access logic
- Real-time data visualization

iii. Code Implementation:

```
import streamlit as st

import pandas as pd
```

```
st.title("Employee Salary Filter App")

#  ROLE SELECTION
st.sidebar.title("User Role")
role = st.sidebar.selectbox(
    "Select Your Role",
    ["HR", "Manager", "Employee"]
)

# Display content based on role
if role == "HR":
    st.sidebar.success("Welcome HR ")
elif role == "Manager":
    st.sidebar.info("Manager Dashboard ")
elif role == "Employee":
    st.sidebar.warning("Employee View ")

st.write(f"### Logged in as: {role}")

#  FILE UPLOAD
file = st.file_uploader("Upload Employee CSV File", type=["csv"])

if file is not None:
    df = pd.read_csv(file)

    st.subheader("Original Data")
    st.dataframe(df)

    #  SALARY FILTER
    if "Salary" in df.columns:
        filtered_df = df[df["Salary"] > 50000]
        st.subheader("Employees with Salary > 50,000")
        st.dataframe(filtered_df)

        st.success(f"{len(filtered_df)} employees found with salary > 50,000")
    else:
        st.error("Salary column not found in CSV file.")
```

iv. Output:



User Role

Select Your Role: HR

Welcome HR

Employee Salary Filter App

Logged in as: HR

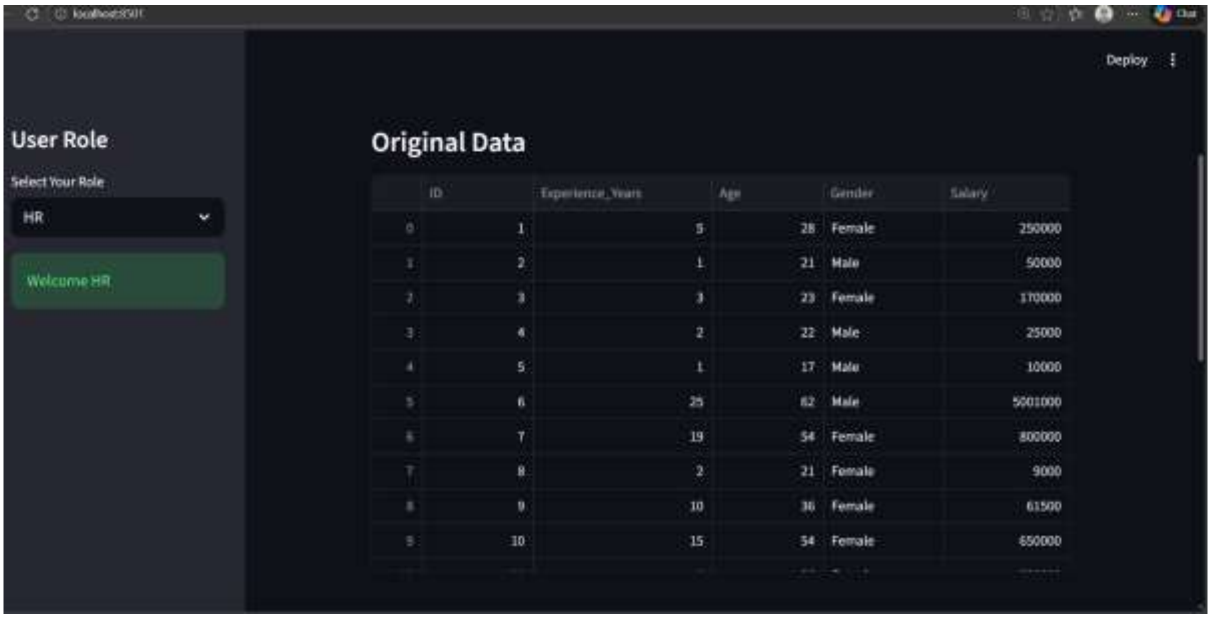
Upload Employee CSV File

Drag and drop file here
Limit 200MB per file • CSV

Employee_Salary_Dataset.csv 0.8KB

Original Data

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000


localhost:8501

Deploy

User Role

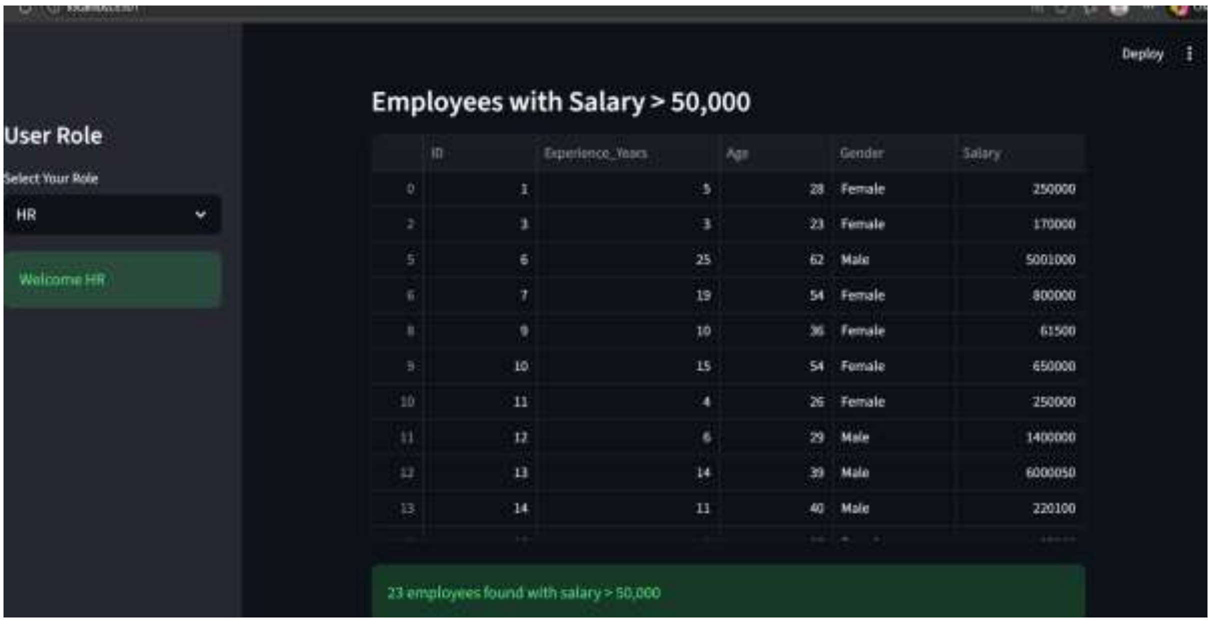
Select Your Role

HR

Welcome HR

Original Data

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
1	2	1	21	Male	50000
2	3	3	23	Female	170000
3	4	2	22	Male	25000
4	5	1	17	Male	10000
5	6	25	62	Male	5001000
6	7	19	54	Female	800000
7	8	2	21	Female	9000
8	9	10	36	Female	61500
9	10	15	54	Female	650000


Deploy

User Role

Select Your Role

HR

Welcome HR

Employees with Salary > 50,000

	ID	Experience_Years	Age	Gender	Salary
0	1	5	28	Female	250000
2	3	3	23	Female	170000
5	6	25	62	Male	5001000
6	7	19	54	Female	800000
8	9	10	36	Female	61500
9	10	15	54	Female	650000
10	11	4	26	Female	250000
11	12	6	29	Male	1400000
12	13	14	39	Male	6000050
13	14	11	40	Male	220100

23 employees found with salary > 50,000