

AI VIETNAM
All-in-One Course
(TA Session)

Streamlit and Gradio

Extra Class: MLOps

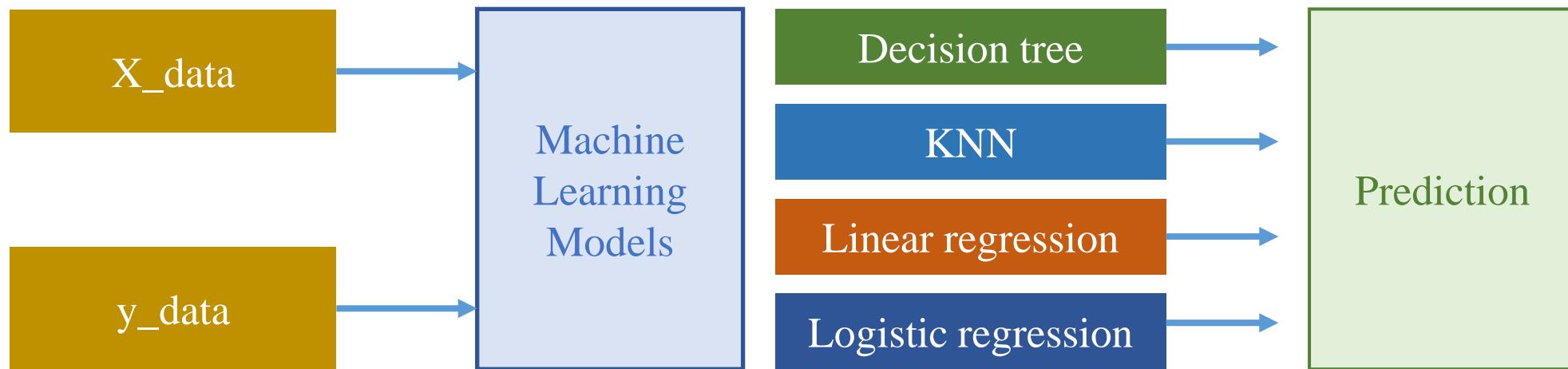


AI VIET NAM
[@aivietnam.edu.vn](http://aivietnam.edu.vn)

Nguyen-Thuan Duong – TA

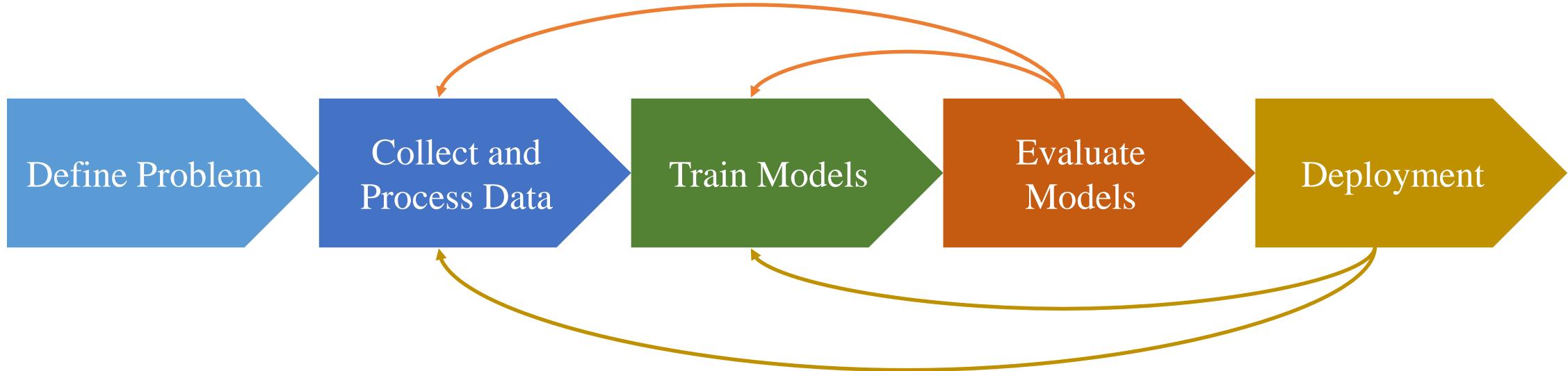
Machine Learning

❖ Machine Learning Models



Machine Learning

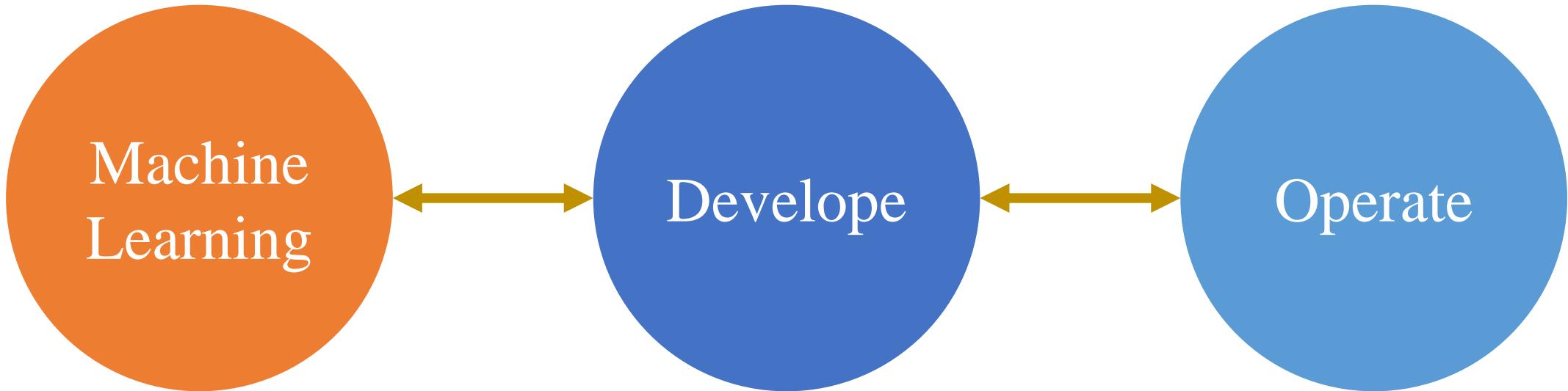
❖ Machine Learning Life Cycle



Machine Learning

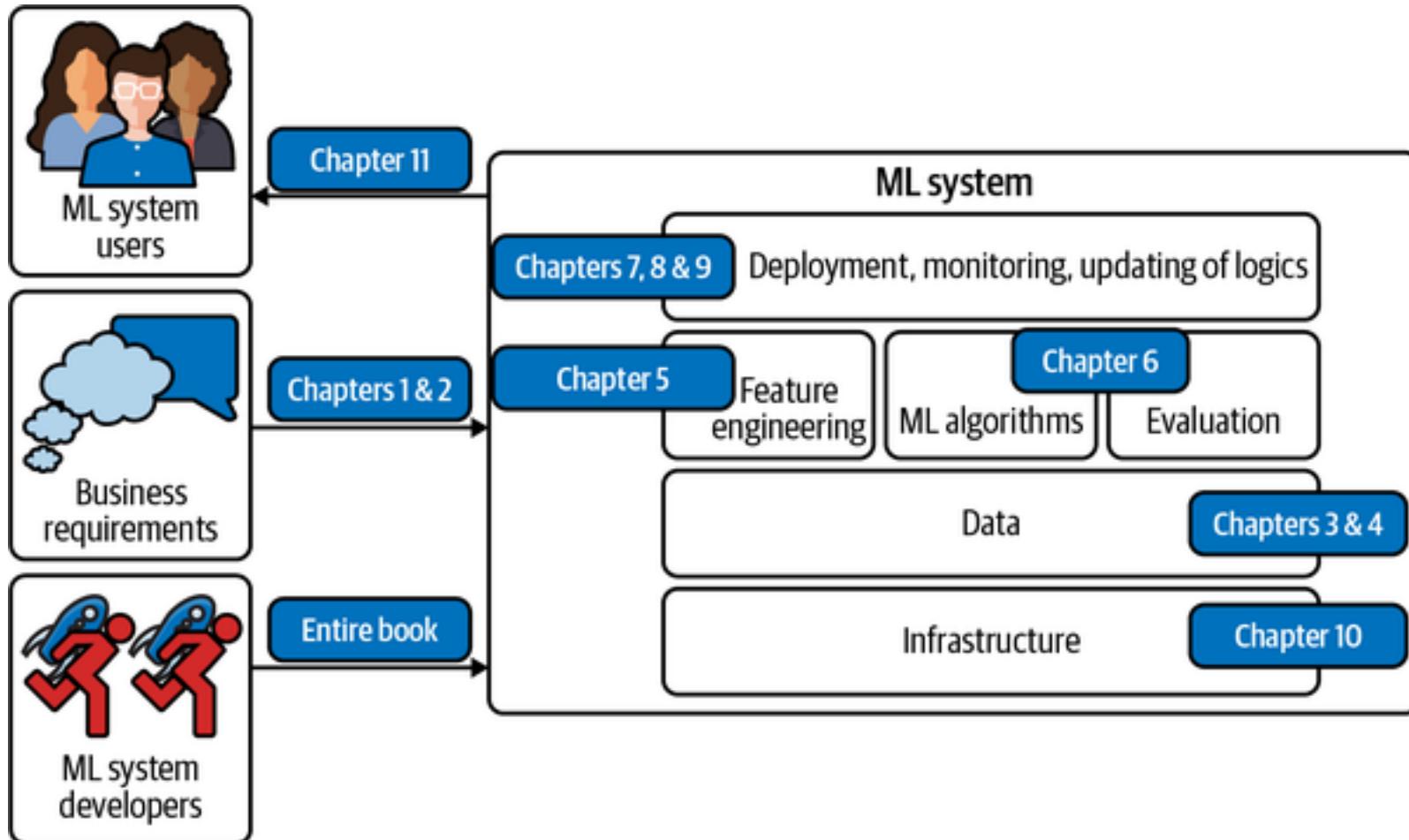
❖ Machine Learning Operation (MLOps)

- MLOps is a set of tools and best practices for bringing ML into production.



Machine Learning

❖ Machine Learning Systems



Outline

- Introduction
- Web Application
- Streamlit
- Gradio
- Hands-on
- Conclusion
- Question

Introduction

Introduction

❖ Getting Started

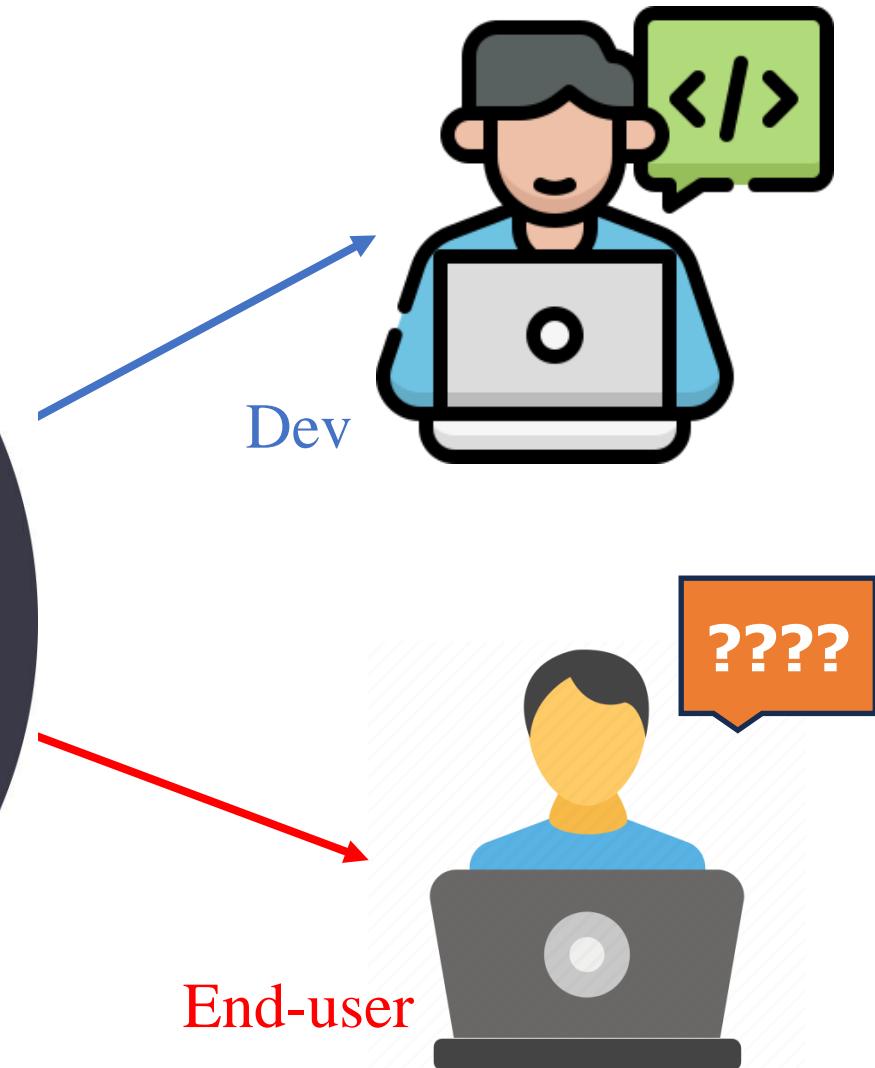
```
● ● ●  
1 from transformers import Wav2Vec2Processor, Wav2Vec2ForCTC  
2 import soundfile as sf  
3 import torch  
4  
5 processor = Wav2Vec2Processor.from_pretrained("nguyenvulebinh/wav2vec2-base-vietnamese-250h")  
6 model = Wav2Vec2ForCTC.from_pretrained("nguyenvulebinh/wav2vec2-base-vietnamese-250h")  
7  
8 # define function to read in sound file  
9 def map_to_array(batch):  
10     speech, _ = sf.read(batch["file"])  
11     batch["speech"] = speech  
12     return batch  
13  
14 # load dummy dataset and read soundfiles  
15 ds = map_to_array({  
16     "file": 'audio-test/t1_0001-00010.wav'  
17 })  
18  
19 input_values = processor(ds["speech"], return_tensors="pt", padding="longest").input_values  
20 logits = model(input_values).logits  
21  
22 # take argmax and decode  
23 predicted_ids = torch.argmax(logits, dim=-1)  
24 transcription = processor.batch_decode(predicted_ids)
```

Here is my
new model



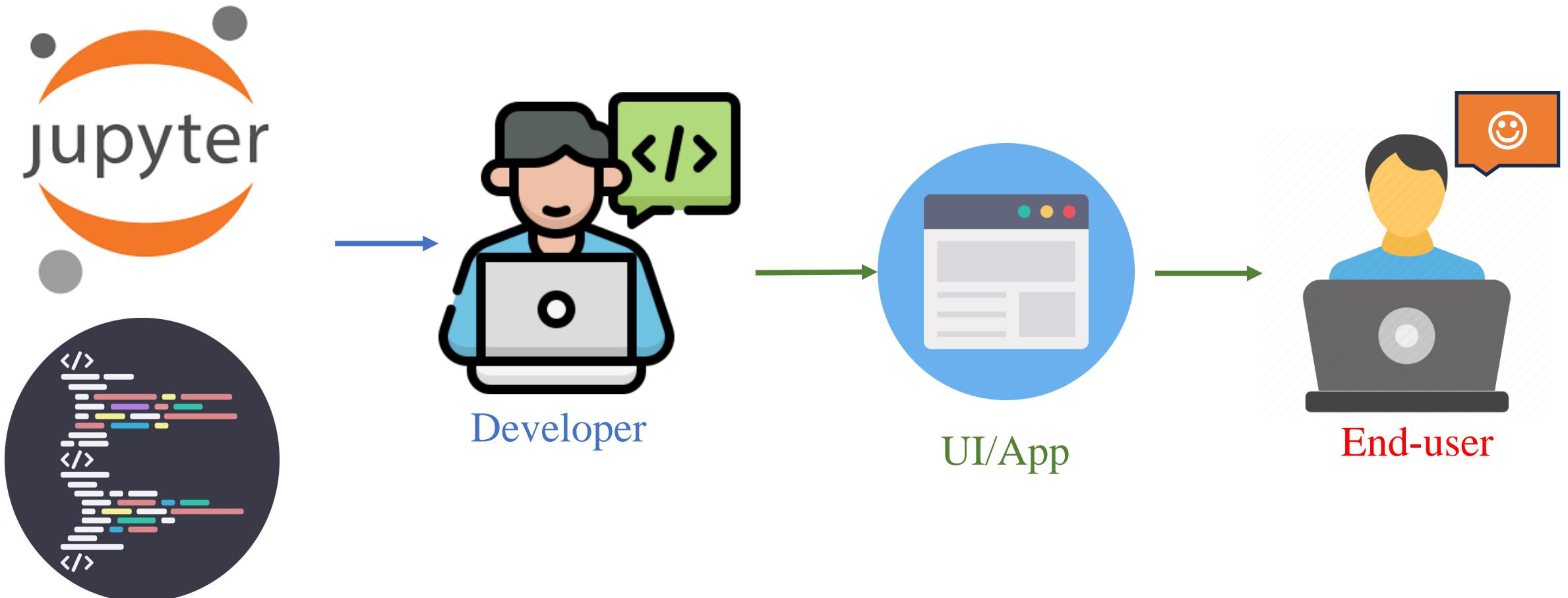
Introduction

❖ Developer vs End-user



Introduction

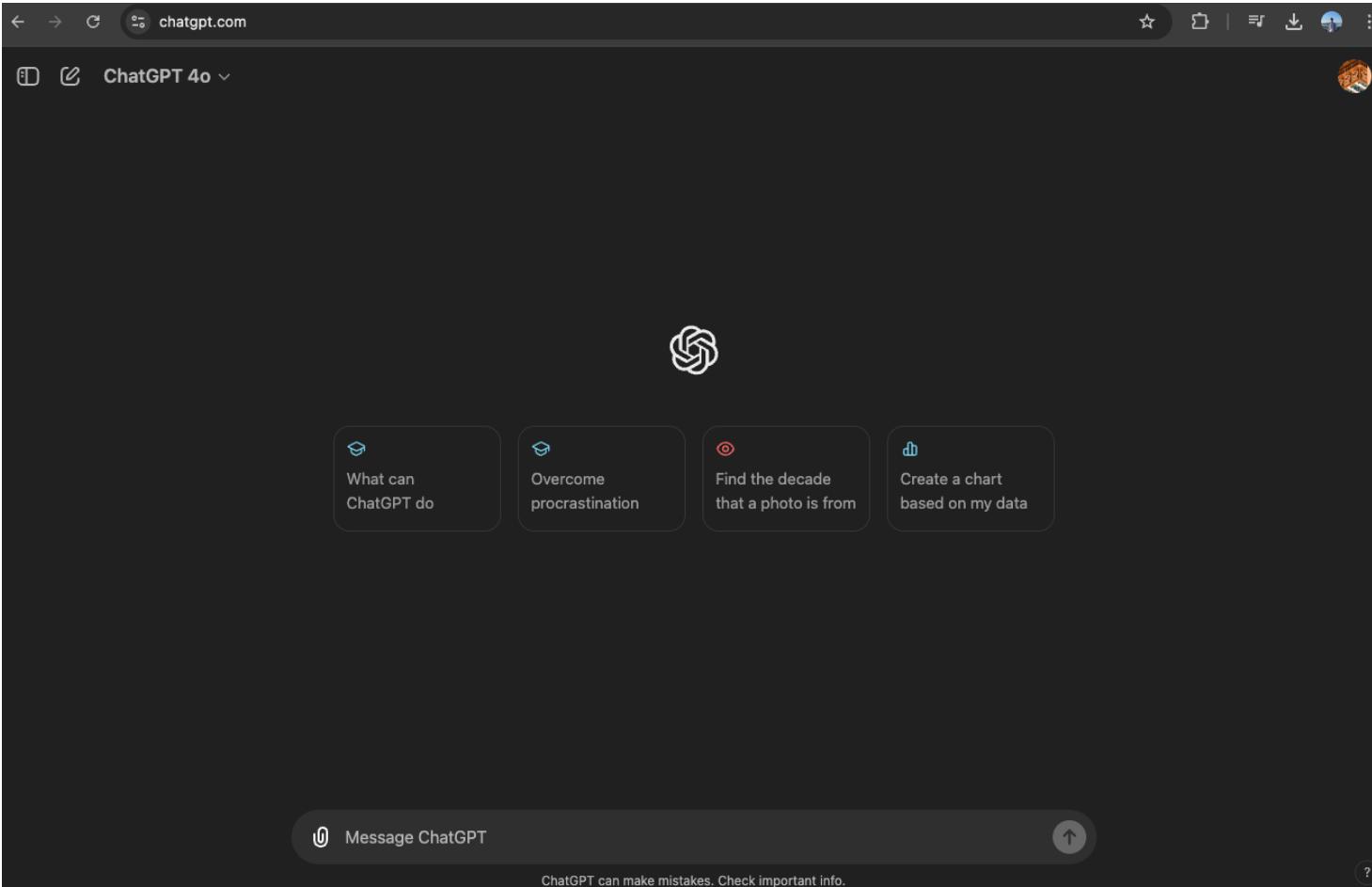
❖ Development



Web application

Web application

❖ ChatGPT web application



Introduction

❖ OCR demo

EasyOCR

Gradio demo for EasyOCR. EasyOCR demo supports 80+ languages. To use it, simply upload your image and choose a language from the dropdown menu, or click one of the examples to load them. Read more at the links below.

Input



language

abq	ady	af	ang	ar	as	ava	az	be
bg	bh	bho	bn	bs	ch_sim	ch_tra	che	
cs	cy	da	dar	de	<input checked="" type="checkbox"/> en	es	et	fa
fr	ga	gom	hi	hr	hu	id	inh	is
it	ja	kbd	kn	ko	ku	la	lbe	lez
lt	lv	mah	mai	mi	mn	mr	ms	mt
ne	new	nl	no	oc	pi	pl	pt	ro
ru	rs_cyrillic	rs_latin	sck	sk	sl	sq	sv	
sw	ta	tab	te	th	tjk	tl	tr	ug
uk	ur	uz	<input checked="" type="checkbox"/> vi					

Output



166.5s

Output 2

TEXT	CONFIDENCE
[H	0.3347064751478329
79D	0.24794097406733084
06:31879	0.181738747536989
THAO ĐIỀN	0.5234315327990876
HIỆP BÌNH	0.6633234946059156
CHÁNH	0.5597377871990743
TP_HCM SÊ THI TUYỂN QUỐC TẾ	0.5408944320326364
TƯỞNG QUY HOẠCH	0.9091566919586396
giây	0.9989028573036194
BẢN ĐÀO THANH ĐA	0.9408069748721736
TP HCM: Chủ dô	0.7445287340962483

Introduction

❖ Web application



Introduction

❖ HTML

```
● ● ●  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6      <title>Styled Text Input Form</title>  
7  </head>  
8  <body>  
9  
10     <div class="container">  
11         <h1>Styled Form</h1>  
12         <form action="#" method="post">  
13             <label for="input1">Input 1:</label>  
14             <input type="text" id="input1" name="input1" required>  
15             <label for="input2">Input 2:</label>  
16             <input type="text" id="input2" name="input2" required>  
17             <button type="submit">Submit</button>  
18         </form>  
19     </div>  
20  
21 </body>  
22 </html>  
23
```

Styled Form

Input 1: Input 2:

Introduction

❖ HTML + CSS

```
● ● ●  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6      <title>Styled Text Input Form</title>  
7      <link rel="stylesheet" href="index.css">  
8  </head>  
9  <body>  
10  
11      <div class="container">  
12          <h1>Styled Form</h1>  
13          <form action="#" method="post">  
14              <label for="input1">Input 1:</label>  
15              <input type="text" id="input1" name="input1" required>  
16              <label for="input2">Input 2:</label>  
17              <input type="text" id="input2" name="input2" required>  
18              <button type="submit">Submit</button>  
19          </form>  
20      </div>  
21  
22  </body>  
23  </html>  
24
```

Introduction

❖ HTML + CSS



```
1 body {  
2     font-family: Arial, sans-serif;  
3     background-color: #f4f4f9;  
4     margin: 0;  
5     padding: 0;  
6     display: flex;  
7     justify-content: center;  
8     align-items: center;  
9     height: 100vh;  
10 }  
11  
12 h1 {  
13     text-align: center;  
14     color: #333;  
15 }
```



```
1 form {  
2     background-color: #fff;  
3     padding: 20px;  
4     border-radius: 8px;  
5     box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
6     width: 100%;  
7     max-width: 400px;  
8 }  
9  
10 label {  
11     font-size: 16px;  
12     color: #555;  
13 }  
14  
15 input[type="text"] {  
16     width: 100%;  
17     padding: 10px;  
18     margin: 8px 0;  
19     border: 1px solid #ccc;  
20     border-radius: 4px;  
21     box-sizing: border-box;  
22 }
```



```
1 button {  
2     background-color: #28a745;  
3     color: white;  
4     padding: 10px 15px;  
5     border: none;  
6     border-radius: 4px;  
7     cursor: pointer;  
8     width: 100%;  
9     font-size: 16px;  
10 }  
11  
12 button:hover {  
13     background-color: #218838;  
14 }  
15  
16 .container {  
17     width: 100%;  
18     max-width: 400px;  
19     margin: 0 auto;  
20 }
```

Introduction

- ❖ HTML + CSS

Styled Form

Input 1:

Input 2:

Submit

Introduction

❖ HTML + CSS + JS

```
1 <body>
2   <div class="container">
3     <h1>Styled Form</h1>
4     <form id="concatForm" action="javascript:void(0);">
5       <label for="input1">First name:</label>
6       <input type="text" id="input1" name="input1" required>
7       <label for="input2">Last name:</label>
8       <input type="text" id="input2" name="input2" required>
9       <button type="submit">Submit</button>
10    </form>
11    <p id="result"></p>
12  </div>
13  <script src="index.js"></script>
14 </body>
```

```
1 document.getElementById('concatForm').addEventListener('submit', function(event) {
2   event.preventDefault();
3
4   const input1 = document.getElementById('input1').value;
5   const input2 = document.getElementById('input2').value;
6
7   const result = input1 + " " + input2;
8
9   document.getElementById('result').textContent = "Your name is: " + result;
10 });
11
```

Introduction

- ❖ HTML + CSS + JS

Styled Form

First name:

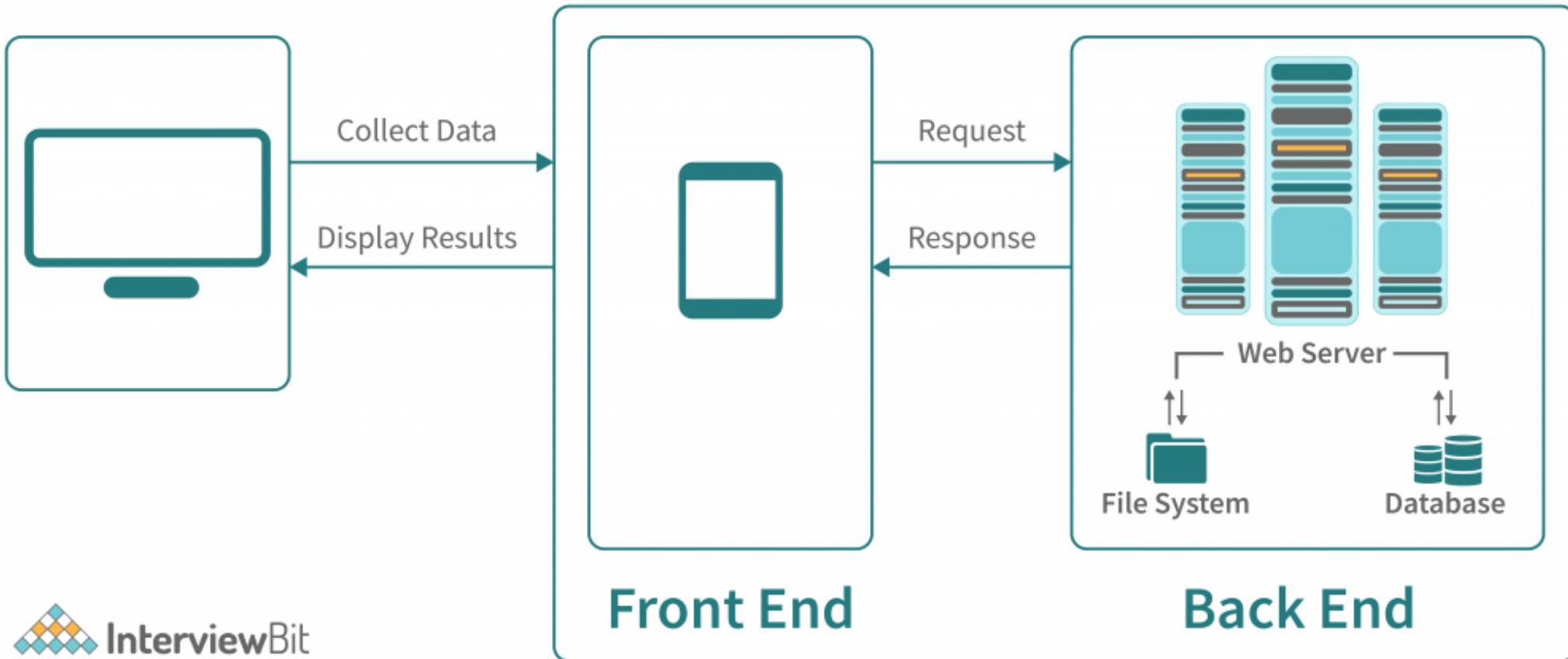
Last name:

Submit

Your name is: Duong Thuan

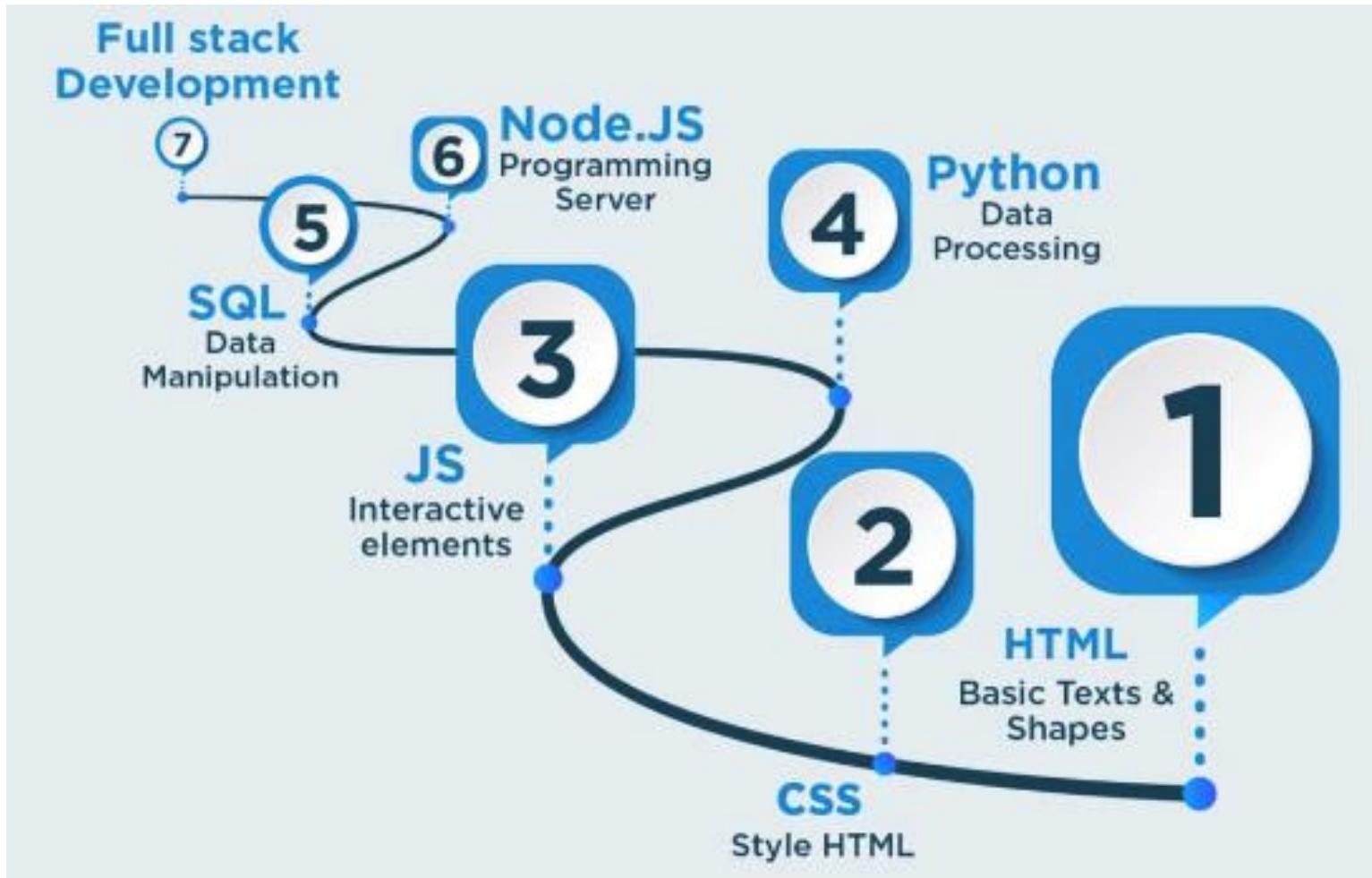
Introduction

❖ Web Development



Introduction

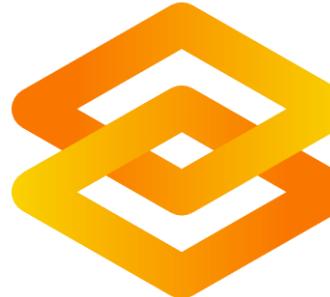
❖ Web Development



Introduction

- ❖ No HTML - No CSS - No JavaScript

Streamlit



gradio

Chainlit



Streamlit

Streamlit

❖ Getting Started

Streamlit's open-source app framework is a breeze to get started with.

The screenshot shows the Streamlit homepage. At the top is a dark navigation bar with the Streamlit logo, a magnifying glass icon, and the URL "streamlit.io". Below it is a light-colored header bar with a red notification badge that says "Latest news: Expanded dataframe support in 1.38, AI recipes, + new community forum design". The main content area features a large, bold, dark text headline: "A faster way to build and share data apps". Below the headline is a subtext: "Streamlit turns data scripts into shareable web apps in minutes. All in pure Python. No front-end experience required." At the bottom of the page are two prominent buttons: a red button on the left labeled "Try Streamlit now" and a white button on the right labeled "Deploy on Community Cloud (it's free!)".

Streamlit

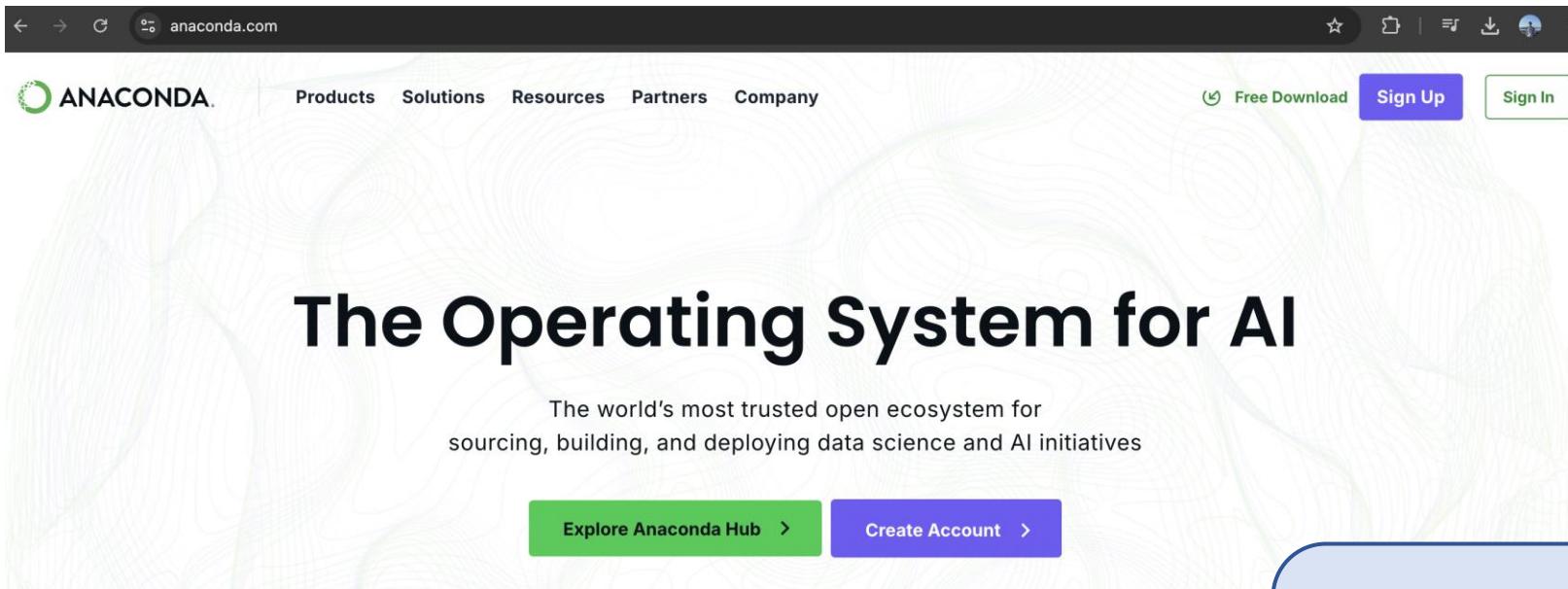
❖ Getting Started

Streamlit also supports deploying applications built with Streamlit to the cloud easily and quickly.

The screenshot shows the Streamlit Cloud homepage. At the top, there's a navigation bar with icons for back, forward, search, and user profile. The URL 'streamlit.io/cloud' is visible. Below the bar, a message says 'Latest news: Expanded dataframe support in 1.38, AI recipes, + new community forum design'. The main content area features a large title 'Community Cloud' in bold black font. Below it is a subtitle: 'Deploy, manage, and share your apps with the world, directly from Streamlit — all for free.' A prominent blue button labeled 'Get Started' is centered at the bottom of this section. The footer contains links for 'Cloud', 'Gallery', 'Components', 'Generative AI', 'Community', 'Docs', 'Blog', 'Sign in', and 'Sign up'.

Streamlit

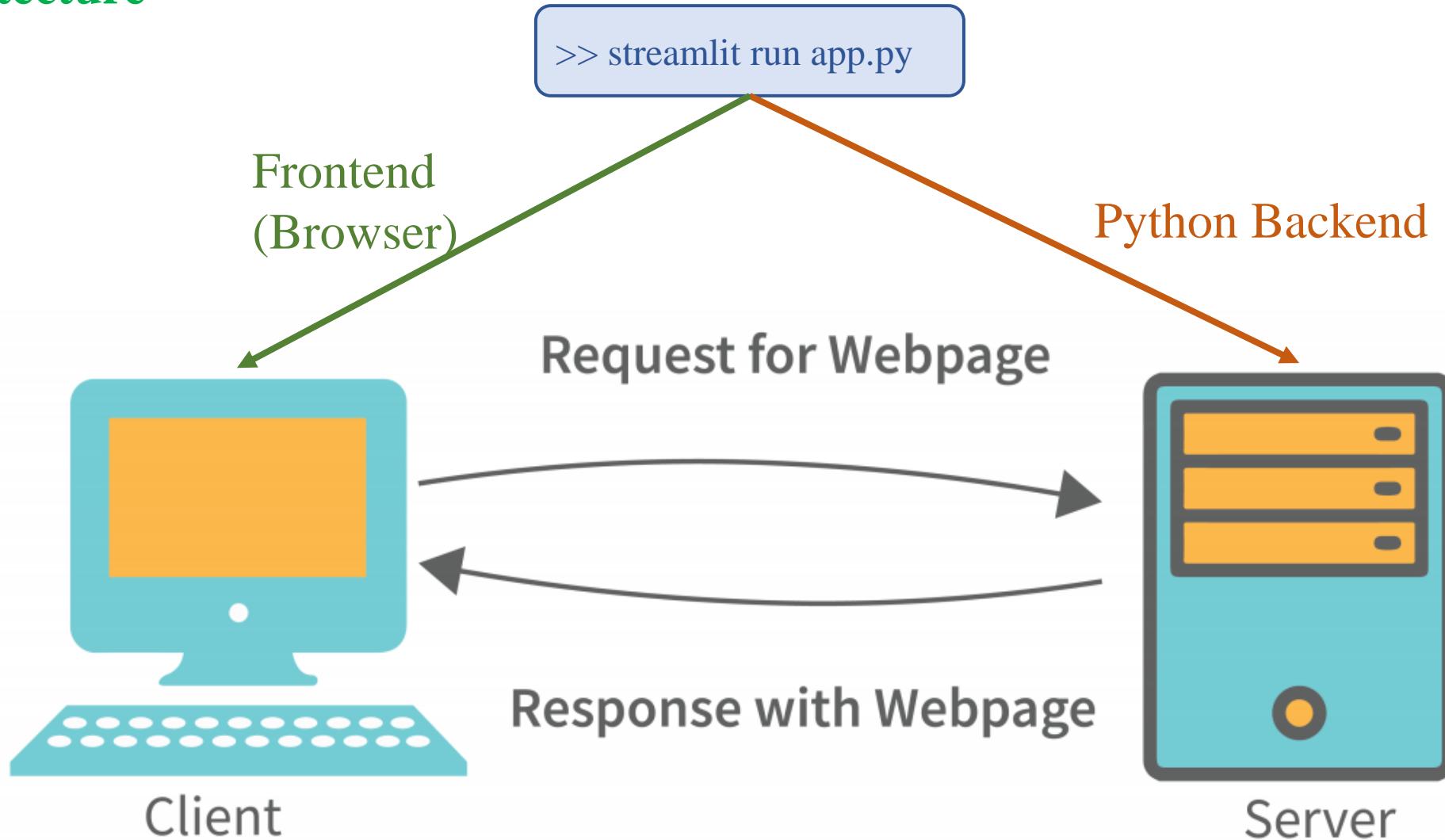
❖ Installation



```
>> conda create -n streamlit_env python=3.11.9  
>> conda activate streamlit_env  
>> pip3 install streamlit==1.39.0
```

Streamlit

❖ Architecture



Streamlit

❖ First app

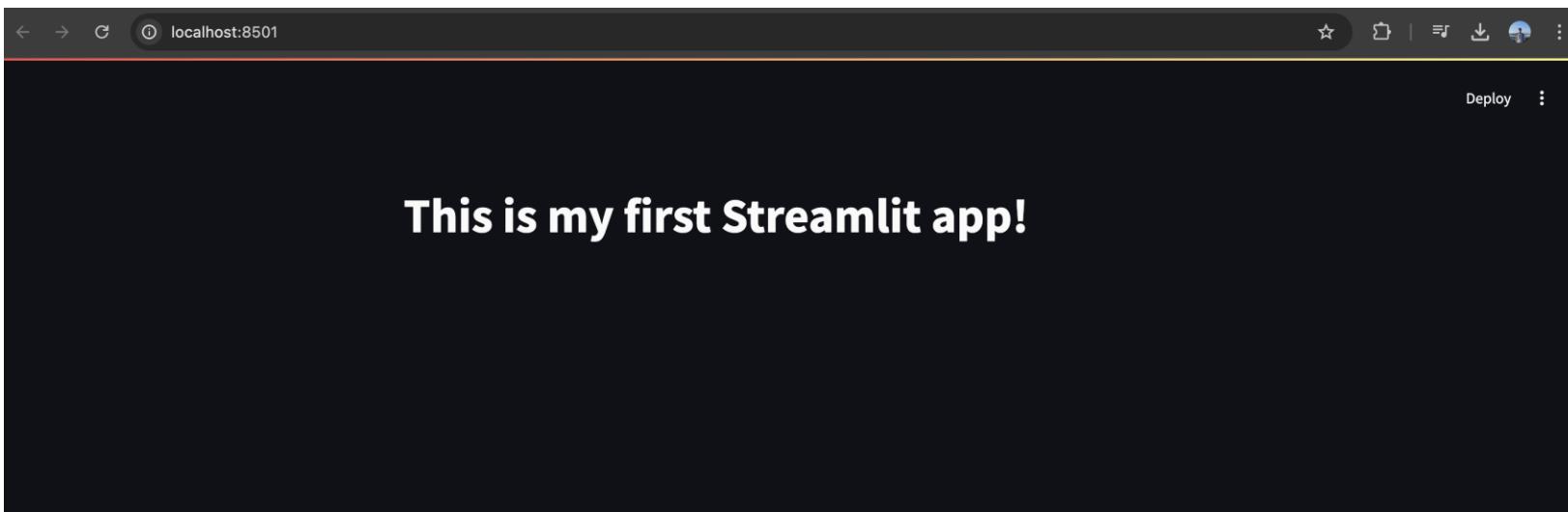
```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")
```

PROBLEMS 6 OUTPUT TERMINAL PORTS JUPYTER DEBUG CONSOLE

○ (aio-mlops-w1) thuann@MacNaN streamlit % streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>
Network URL: <http://192.168.1.5:8501>



Text element

Text element

❖ st.text

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 st.text("Streamlit is so easy!")
```

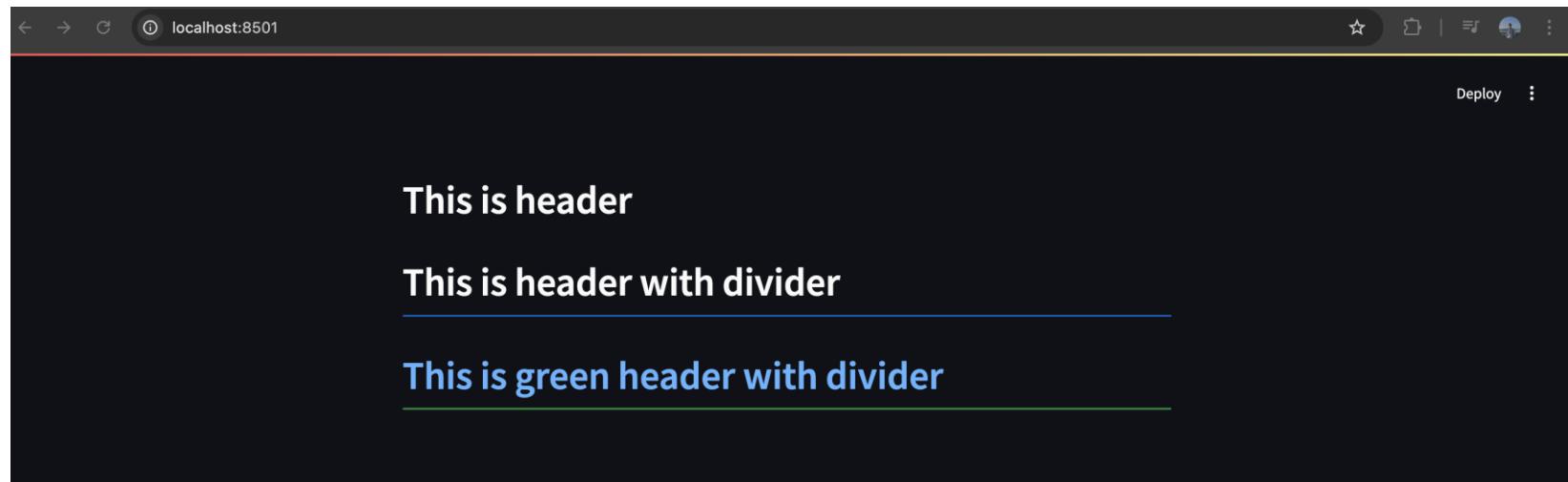
This is my first Streamlit app!

Streamlit is so easy!

Text element

❖ st.header

```
1 import streamlit as st
2
3 st.header("This is header")
4 st.header("This is header with divider", divider=True)
5 st.header(":blue[This is green header with divider]", divider=True)
```



Text element

❖ st.markdown

```
● ● ●  
1 import streamlit as st  
2  
3 st.header("AI02024 Contest", divider=True)  
4 st.subheader("Module 3 Examination", divider=True)  
5  
6 problem = """  
7     ***Bài toán***  
8 Cho một số nguyên không âm `x`, trả về căn bậc hai của `x` được làm tròn xuống thành số nguyên gần nhất. Số nguyên được trả về là số không âm.  
9  
10    ***Ví dụ***  
11    **** Ví dụ 1:  
12    - **Đầu vào**: `x = 4`  
13    - **Đầu ra**: `2`  
14    - **Giải thích**: Căn bậc hai của 4 là 2, vì vậy chúng ta trả về 2.  
15  
16    **** Ví dụ 2:  
17    - **Đầu vào**: `x = 8`  
18    - **Đầu ra**: `2`  
19    - **Giải thích**: Căn bậc hai của 8 là 2.82842..., và vì chúng ta làm tròn xuống thành số nguyên gần nhất, nên trả về 2.  
20  
21  
22 st.markdown(problem)
```

AI02024 Contest

Module 3 Examination

Bài toán

Cho một số nguyên không âm x , trả về căn bậc hai của x được làm tròn xuống thành số nguyên gần nhất. Số nguyên được trả về cũng phải là số không âm.

Ví dụ

Ví dụ 1:

- Đầu vào: $x = 4$
- Đầu ra: 2
- Giải thích: Căn bậc hai của 4 là 2, vì vậy chúng ta trả về 2.

Ví dụ 2:

- Đầu vào: $x = 8$
- Đầu ra: 2
- Giải thích: Căn bậc hai của 8 là 2.82842..., và vì chúng ta làm tròn xuống thành số nguyên gần nhất, nên trả về 2.

Text element

❖ st.code



```
1 import streamlit as st
2
3 st.header("AI02024 Contest", divider=True)
4 st.subheader("Module 3 Examination", divider=True)
5
6 code = """
7 class Solution(object):
8     def mySqrt(self, x):
9         return int(x**0.5)
10 """
11
12 st.code(code, language="python")
```

AI02024 Contest

Module 3 Examination

```
class Solution(object):
    def mySqrt(self, x):
        return int(x**0.5)
```

Input element

Input element

❖ st.text_input



```
1 # app.py
2 import streamlit as st
3
4 st.title("This is my first Streamlit app!")
5
6 name = st.text_input(
7     label="Enter your name: ",
8 )
9
10 st.write("Your name is: ", name)
```

This is my first Streamlit app!

Enter your name:

Thuận

Your name is: Thuận

Input element

❖ st.number_input

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 name = st.text_input(  
7     label="Enter your name: ",  
8     value="",  
9 )  
10 age = st.number_input(  
11     "Enter your age: ",  
12     value=0,  
13     step=1,  
14     format="%d"  
15 )  
16  
17 st.write("Your name is: ", name)  
18 st.write("Your age is ", age)
```

This is my first Streamlit app!

Enter your name:

Thuan

Enter your age:

19

- +

Your name is: Thuan

Your age is 19

Input element

❖ st.slider

```
1 # app.py
2 import streamlit as st
3
4 st.title("This is my first Streamlit app!")
5
6 name = st.text_input(
7     label="Enter your name: ",
8     value="",
9 )
10 st.write("Your name is: ", name)
11
12
13 age = st.slider("How old are you?", 0, 60, 20)
14 st.write("I'm ", age, "years old")
15
```

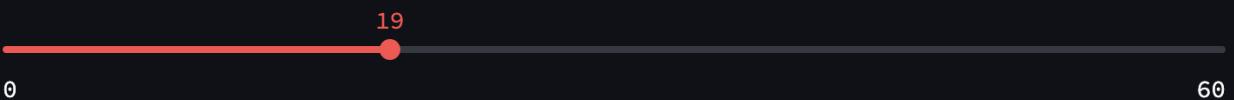
This is my first Streamlit app!

Enter your name:

Thuan

Your name is: Thuan

How old are you?



I'm 19 years old

Input element

❖ st.checkbox

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 name = st.text_input(  
7     label="Enter your name: ",  
8     value="",  
9 )  
10 st.write("Your name is: ", name)  
11  
12 is_aio = st.checkbox("Are you AIO?")  
13 if is_aio:  
14     st.write("Are you finish the contest?")
```

This is my first Streamlit app!

Enter your name:

Thuận

Your name is: Thuận

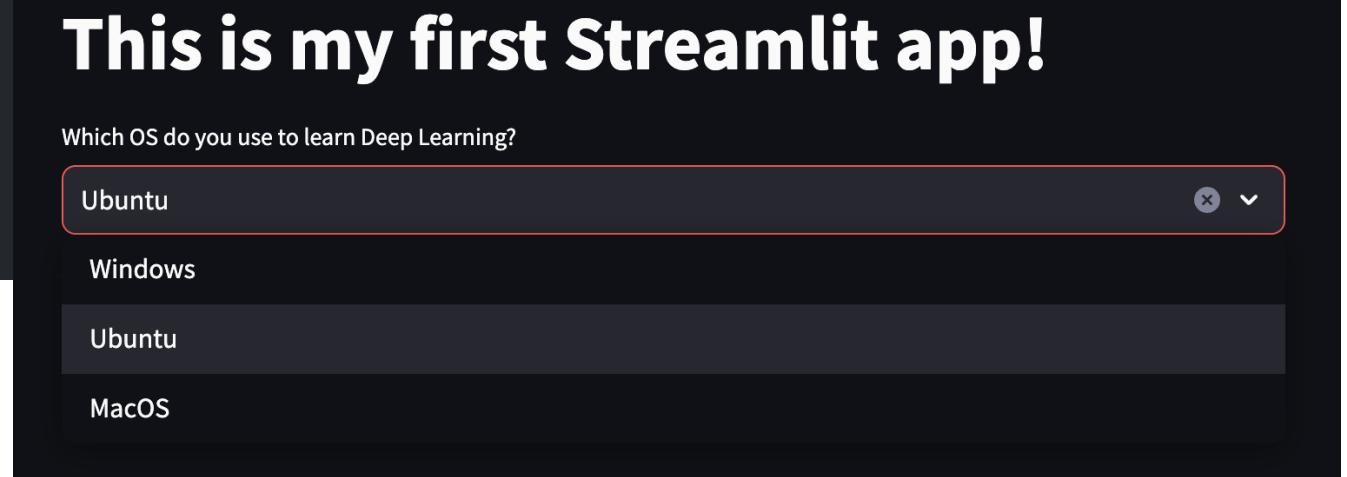
Are you AIO?

Are you finish the contest?

Input element

❖ st.selectbox

```
1 # app.py
2 import streamlit as st
3
4 st.title("This is my first Streamlit app!")
5
6 option = st.selectbox(
7     "Which OS do you use to learn Deep Learning?",
8     ("Windows", "Ubuntu", "MacOS"),
9     index=None,
10    placeholder="Select an OS...",
11 )
12
13 st.write("You choosed:", option)
```



Input element

❖ st.radio

```
1 # app.py
2 import streamlit as st
3
4 st.title("This is my first Streamlit app!")
5
6 option = st.radio(
7     "Which OS do you use to learn Deep Learning?",
8     ["Windows", "Ubuntu", "MacOS"],
9     captions=[
10         "Windows good to play games :video_game:",
11         "Ubuntu good for learning CLI :smile:",
12         "MacOS good for designer :vampire:",
13     ]
14 )
15
16 st.write("You choosed:", option)
```

This is my first Streamlit app!

Which OS do you use to learn Deep Learning?

Windows

Windows good to play games 🎮

Ubuntu

Ubuntu good for learning CLI 😊

MacOS

MacOS good for designer 💃'

You choosed: Ubuntu

Input element

❖ st.button

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 email = st.text_input(  
7     label="Type your email here: "  
8 )  
9  
10 if st.button("Submit"):  
11     if "@" not in email:  
12         st.write("Please type a valid email")  
13     else:  
14         st.write("Submitted")
```

This is my first Streamlit app!

Type your email here:

duongthuan

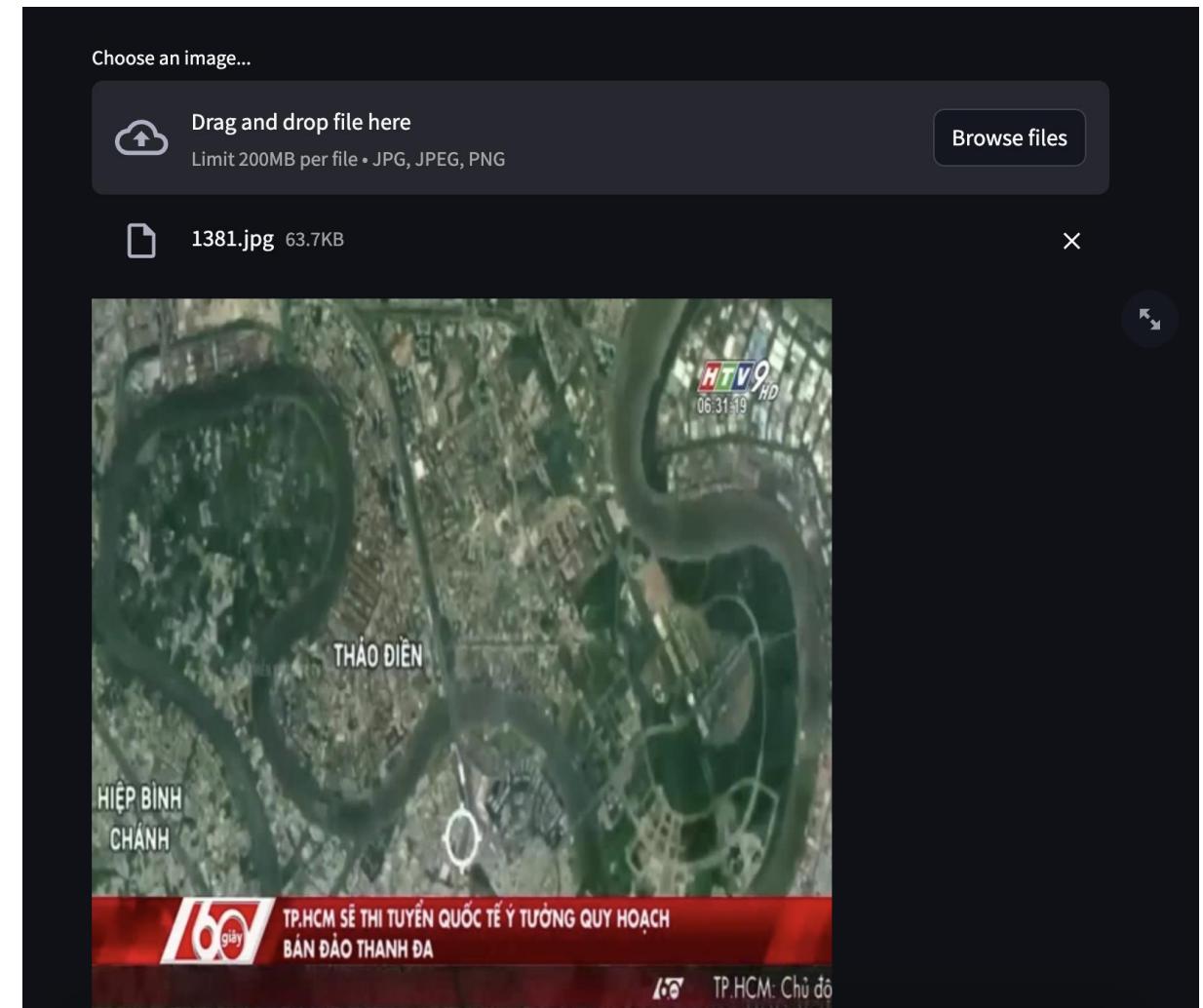
Submit

Please type a valid email

Input element

❖ st.file_uploader

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3 from PIL import Image  
4  
5 uploaded_file = st.file_uploader("Choose an image...",  
6                                   type=["jpg", "jpeg", "png"])  
7  
8 if uploaded_file is not None:  
9     img = Image.open(uploaded_file)  
10    img = img.resize((224, 224))  
11    st.image(img, caption="Uploaded Image.")
```



Media element

Media element

❖ st.image

```
1 # app.py
2 import streamlit as st
3
4 st.title("This is my first Streamlit app!")
5
6 st.image("./60s.jpg", caption="The 60s program")
```

This is my first Streamlit app!



The 60s program

Media element

❖ st.video

```
1 # app.py
2 import streamlit as st
3
4 st.title("This is my first Streamlit app!")
5
6 video_file = open("./L01_V001.mp4", "rb")
7 video_bytes = video_file.read()
8
9 st.video(video_bytes)
```

This is my first Streamlit app!



Layout element

Layout element

❖ st.columns

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 col1, col2, col3 = st.columns(3)  
7  
8 with col1:  
9     st.header("Information")  
10    name = st.text_input(  
11        label="Enter your name: ",  
12    )  
13    st.write("Your name is:", name)  
14  
15 with col2:  
16     st.header("An image")  
17     st.image("./60s.jpg")  
18  
19 with col3:  
20     st.header("An video")  
21     video_file = open("./L01_V001.mp4", "rb")  
22     video_bytes = video_file.read()  
23     st.video(video_bytes)
```

This is my first Streamlit app!

Information

Enter your name:

Thuận

Your name is: Thuận

An image



An video



Layout element

❖ st.container

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 problem = """  
7 ### **Bài toán**  
8 Cho một số nguyên không âm `x` , trả về căn bậc hai của `x` được làm tròn xuống thành số nguyên gần nhất.  
9  
10 ### **Ví dụ**  
11 ##### Ví dụ 1:  
12 - **Đầu vào**: `x = 4`  
13 - **Đầu ra**: `2`  
14 - **Giải thích**: Căn bậc hai của 4 là 2, vì vậy chúng ta trả về 2.  
15  
16 ##### Ví dụ 2:  
17 - **Đầu vào**: `x = 8`  
18 - **Đầu ra**: `2`  
19 - **Giải thích**: Căn bậc hai của 8 là  $2.82842\dots$ , và vì chúng ta làm tròn xuống thành số nguyên gần nhất  
20  
21  
22 with st.container(height=450):  
23     st.markdown(problem)  
24
```

This is my first Streamlit app!

Bài toán

Cho một số nguyên không âm x , trả về căn bậc hai của x được làm tròn xuống thành số nguyên gần nhất. Số nguyên được trả về cũng phải là số không âm.

Ví dụ

Ví dụ 1:

- **Đầu vào:** $x = 4$
- **Đầu ra:** 2
- **Giải thích:** Căn bậc hai của 4 là 2, vì vậy chúng ta trả về 2.

Ví dụ 2:

- **Đầu vào:** $x = 8$

Status element

Status element

❖ st.error / st.success

```
● ● ●  
1 # app.py  
2 import streamlit as st  
3  
4 st.title("This is my first Streamlit app!")  
5  
6 email = st.text_input(  
7     label="Type your email here:  
8 )  
9  
10 if st.button("Submit"):  
11     if "@" not in email:  
12         st.error("Please type a valid email",  
13                     icon="❗")  
14     else:  
15         st.success("Submitted",  
16                         icon="✅")
```

This is my first Streamlit app!

Type your email here:

duongthuan@gmail.com

Submit

✓ Submitted

This is my first Streamlit app!

Type your email here:

duongthuan

Submit

❗ Please type a valid email

Status element

❖ st.spinner

```
1 # app.py
2 import time
3 import random
4 import streamlit as st
5
6 st.title("This is my first Streamlit app!")
7
8 def load_model():
9     with st.spinner('Model is being loaded...'):
10         time.sleep(3)
11     is_loaded = random.choice([True, False])
12     return is_loaded
13
14 model = load_model()
15 if model:
16     st.success("Model is loaded!")
17 else:
18     st.error("Model failed to load!")
```

This is my first Streamlit app!

 Model is being loaded...

This is my first Streamlit app!

Model is loaded!

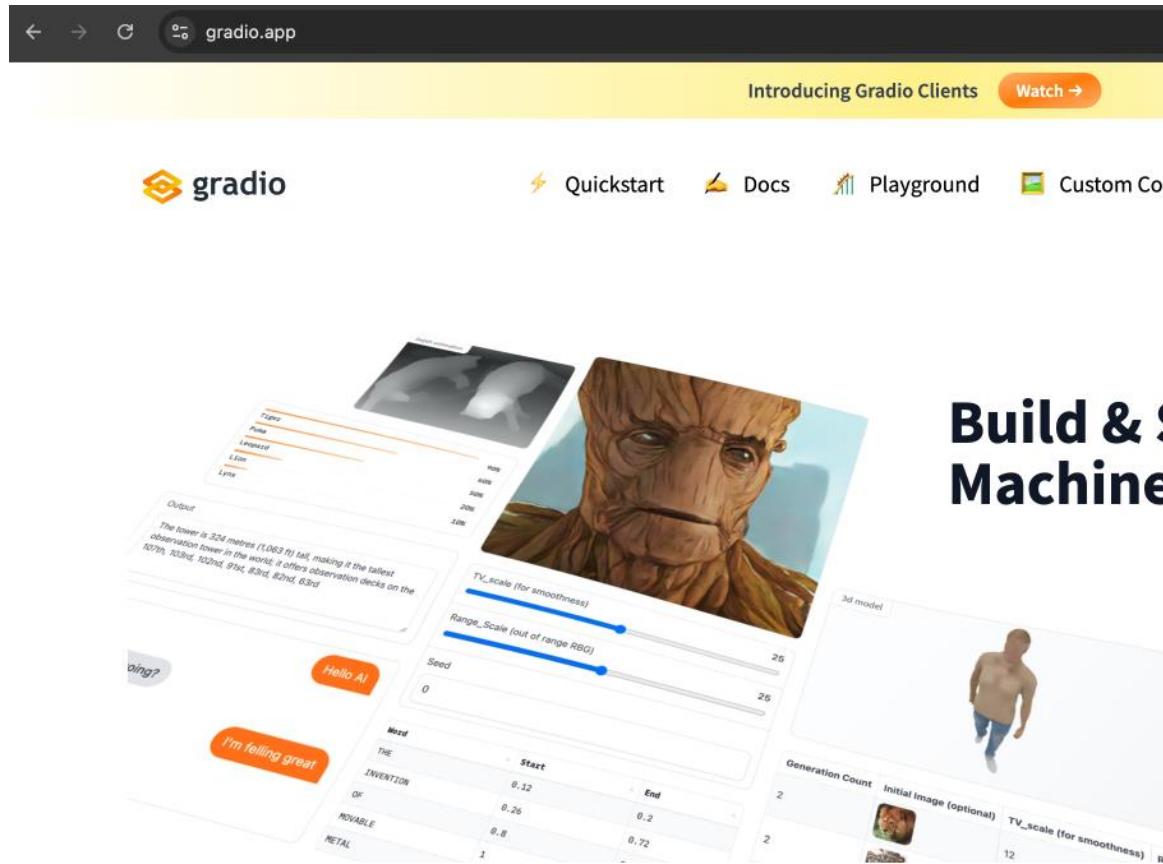
This is my first Streamlit app!

Model failed to load!

Gradio

Gradio

❖ Getting Started



The screenshot shows the Gradio homepage with a navigation bar featuring links for Quickstart, Docs, Playground, Custom Components, and Community. A search bar is also present. Below the navigation, there are two main demo sections. The left section, titled "Hello AI", includes a "Groot" face generation interface with sliders for TV_scale and Range_Scale, and a 3D model viewer. The right section, titled "3d model", shows a 3D human model. A large headline on the right side reads "Build & Share Delightful Machine Learning Apps". Below the headline, a descriptive text states: "Gradio is the fastest way to demo your machine learning model with a friendly web interface so that anyone can use it, anywhere!". At the bottom right, there are buttons for "Get Started", "Star" (with a count of 32485), and a "3D Model" link.

Introducing Gradio Clients [Watch →](#)

gradio

Quickstart Docs Playground Custom Components Community

Search 32485

Build & Share Delightful Machine Learning Apps

Gradio is the fastest way to demo your machine learning model with a friendly web interface so that anyone can use it, anywhere!

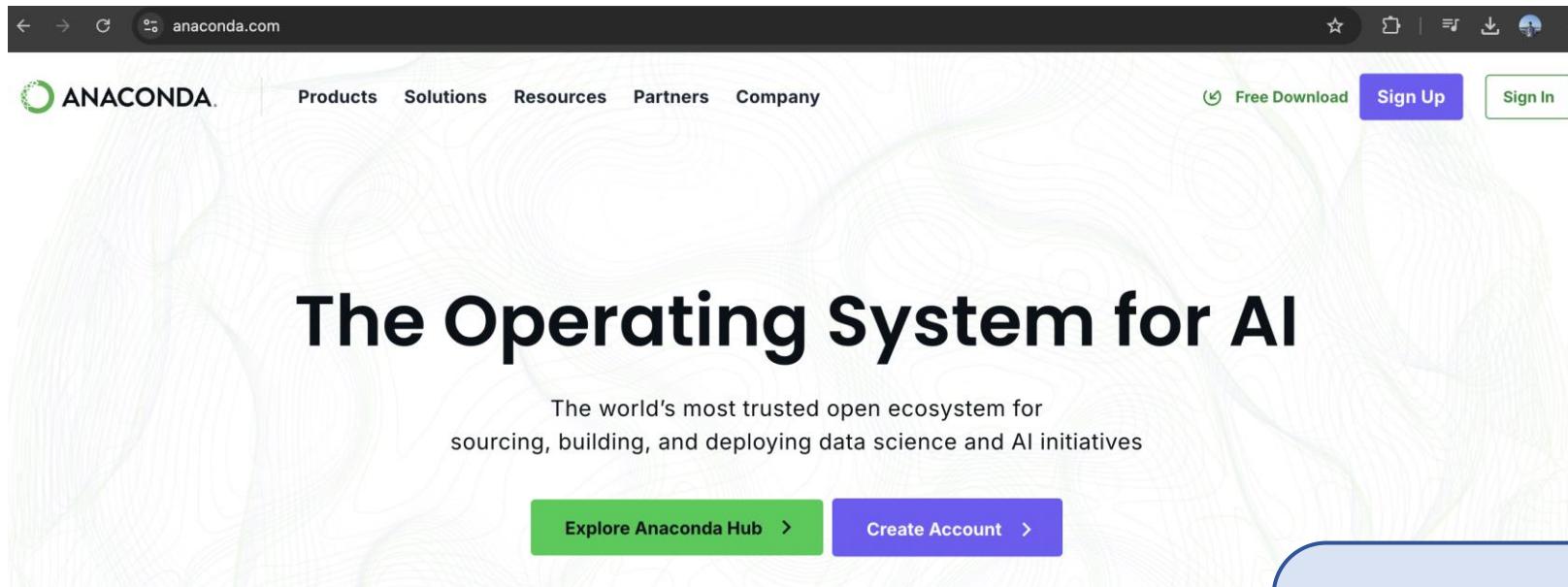
Get Started

Star 32485

3D Model

Gradio

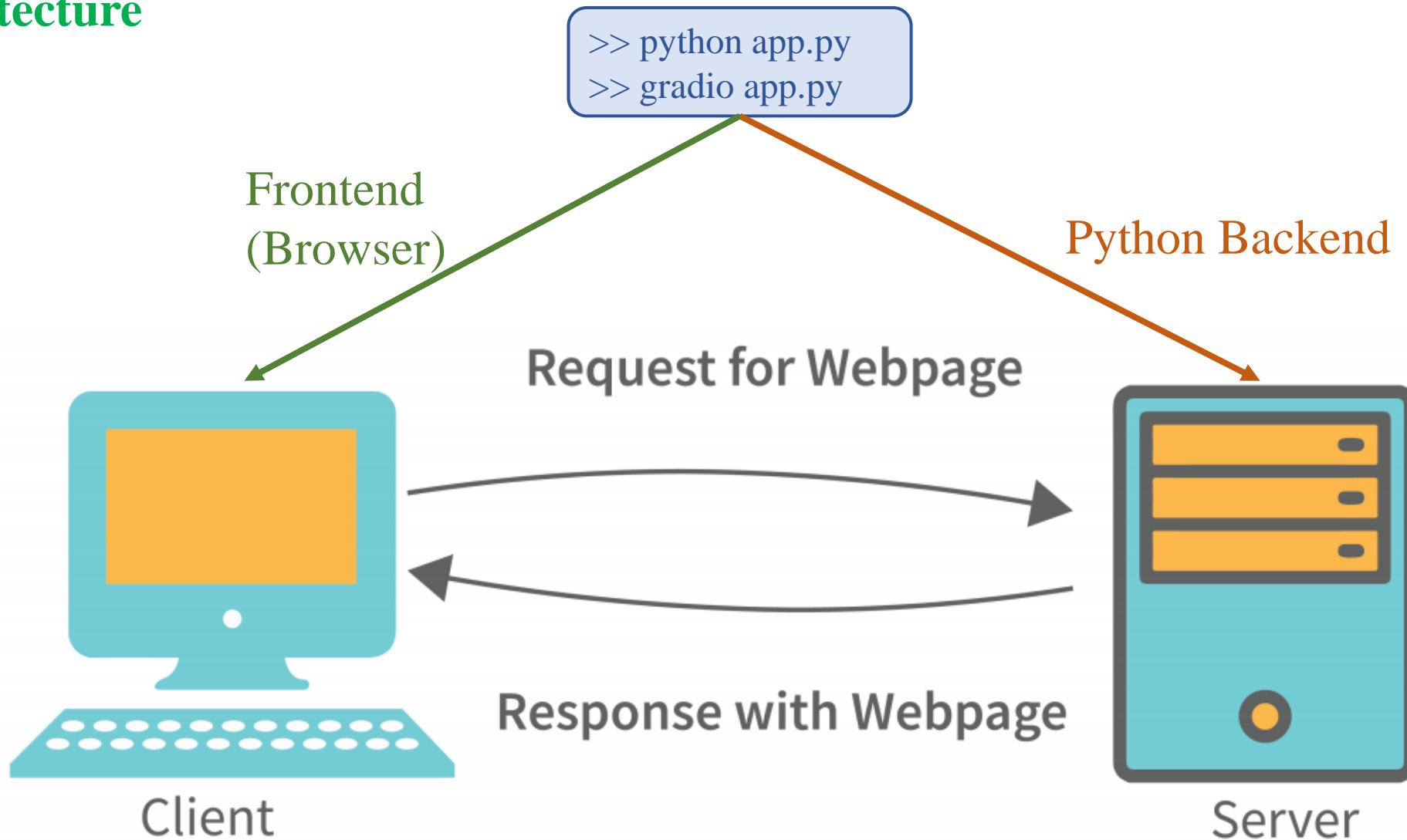
❖ Installation



```
>> conda create -n gradio_env python=3.11.9  
>> conda activate gradio _env  
>> pip3 install streamlit==4.44.1
```

Gradio

❖ Architecture



Gradio

❖ First app

The screenshot shows a Jupyter Notebook interface with two main sections: a code editor and a terminal/TUI area.

Code Editor:

```
import gradio as gr
def greet(name):
    return "Hello, " + name
demo = gr.Interface(
    fn=greet,
    inputs=["text"],
    outputs=["text"],
)
demo.launch()
```

Terminal/TUI Area:

PROBLEMS 6 OUTPUT TERMINAL PORTS JUPYTER DEBUG CONSOLE

o (aio-mlops-w1) thuannd@MacNaN gradio % python app.py
Running on local URL: http://127.0.0.1:7860
To create a public link, set `share=True` in `launch()`.

Browser-based Interface:

127.0.0.1:7860

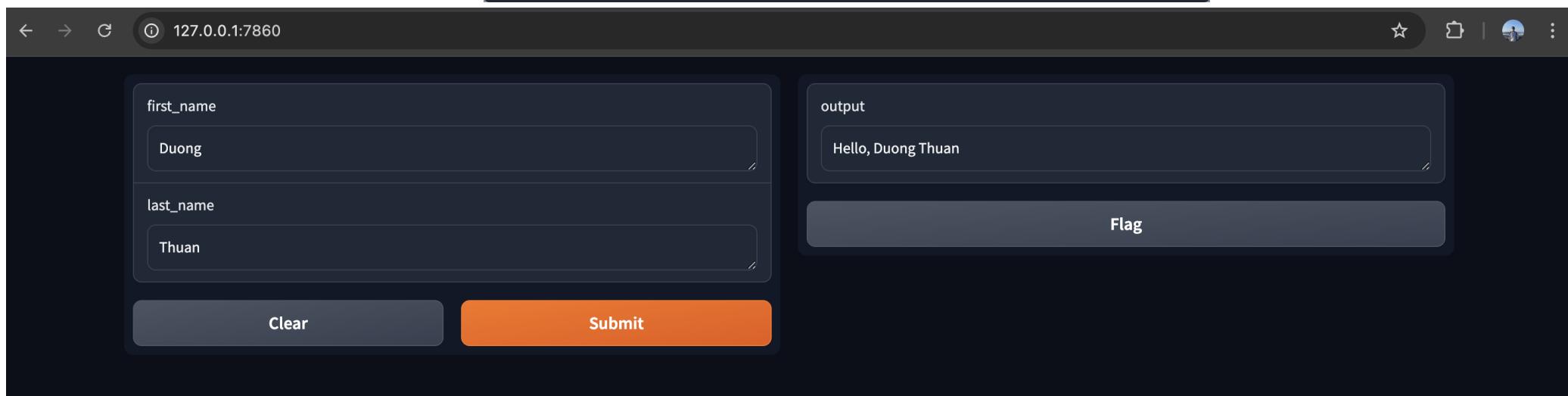
The interface has two input fields: "name" (containing "Thuan") and "output" (containing "Hello, Thuan"). It features three buttons at the bottom: "Clear", "Submit" (highlighted in orange), and "Flag".

Gradio with Interface

Interface

❖ Input element

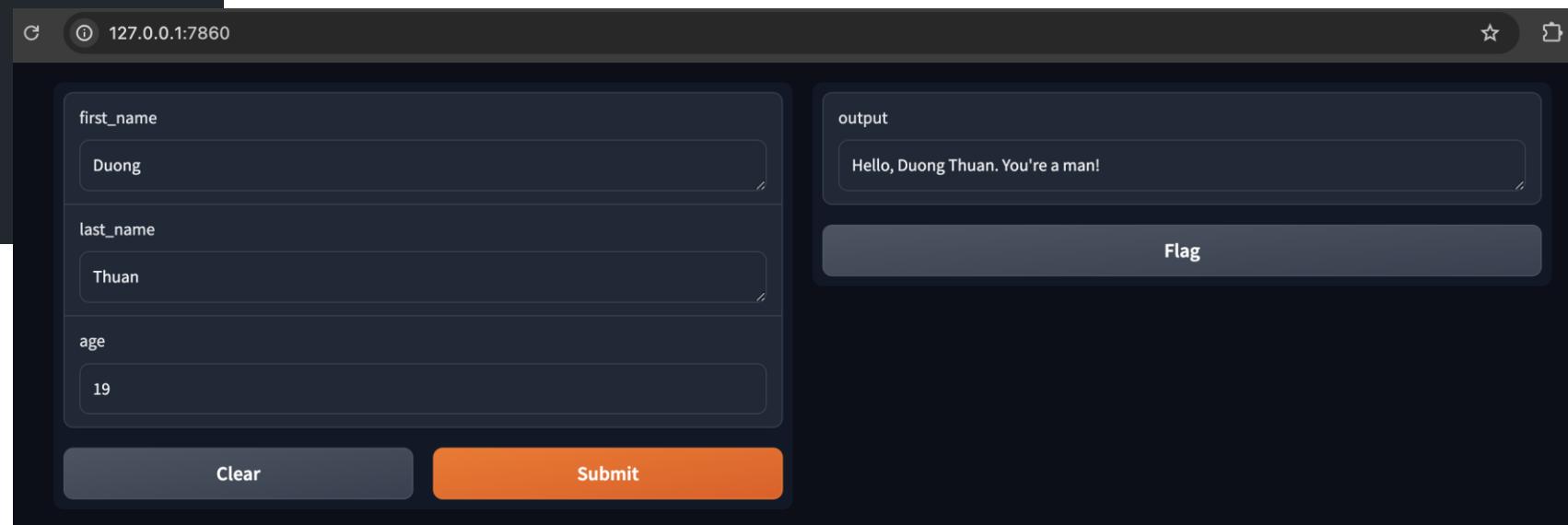
```
● ● ●  
1 import gradio as gr  
2  
3 def greet(first_name, last_name):  
4     return "Hello, " + first_name + " " + last_name  
5  
6 demo = gr.Interface(  
7     fn=greet,  
8     inputs=["text", "text"],  
9     outputs=["text"],  
10 )  
11  
12 demo.launch()
```



Interface

❖ Input element

```
● ● ●  
1 import gradio as gr  
2  
3 def greet(first_name, last_name, age):  
4     getting = "Hello, " + first_name + " " + last_name  
5     if age < 18:  
6         getting += ". You're a young!"  
7     else:  
8         getting += ". You're a man!"  
9     return getting  
10  
11 demo = gr.Interface(  
12     fn=greet,  
13     inputs=["text", "text", "number"],  
14     outputs=["text"],  
15 )  
16 demo.launch()
```



Interface

❖ Input element

```
● ● ●  
1 import gradio as gr  
2  
3 def greet(first_name, last_name, age):  
4     getting = "Hello, " + first_name + " " + last_name  
5     if age < 18:  
6         getting += ". You're a young!"  
7     else:  
8         getting += ". You're a man!"  
9     return getting  
10  
11 demo = gr.Interface(  
12     fn=greet,  
13     inputs=["text", "text", gr.Slider(1, 60, step=1)],  
14     outputs=["text"],  
15 )  
16 demo.launch()
```

The screenshot shows a Gradio interface with three input fields: 'first_name' containing 'Duong', 'last_name' containing 'Thuan', and 'age' set to 19. Below these inputs are two buttons: 'Clear' and 'Submit'. To the right, there is an 'output' section displaying the generated greeting: 'Hello, Duong Thuan. You're a man!' and a 'Flag' button.

Interface

❖ Input element

```
● ● ●  
1 import gradio as gr  
2  
3 def greet(name, over_18, age):  
4     getting = "Hello, " + name  
5     if not over_18:  
6         getting += ". You're a young!"  
7     else:  
8         getting += ". You're a man!"  
9     return getting  
10  
11 demo = gr.Interface(  
12     fn=greet,  
13     inputs=["text", "checkbox"],  
14     outputs=["text"],  
15 )  
16 demo.launch()
```

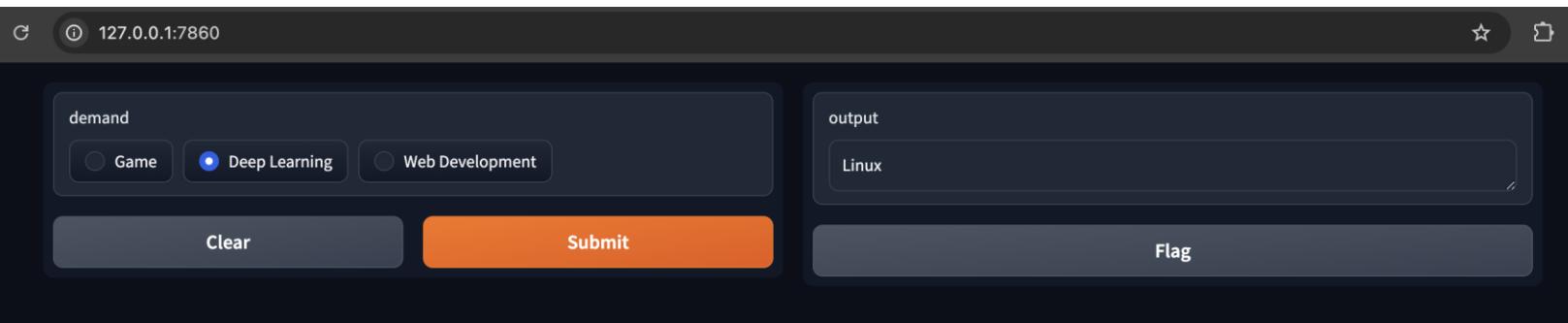
The screenshot shows a Gradio interface with the following components:

- Input Fields:**
 - A text input field labeled "name" containing the value "Thuan".
 - A checkbox input field labeled "over_18" which is unchecked.
- Buttons:**
 - A "Clear" button.
 - An orange "Submit" button.
- Output Field:** A text area labeled "output" displaying the message "Hello, Thuan. You're a young!".
- Flag Button:** A large grey button labeled "Flag".

Interface

❖ Input element

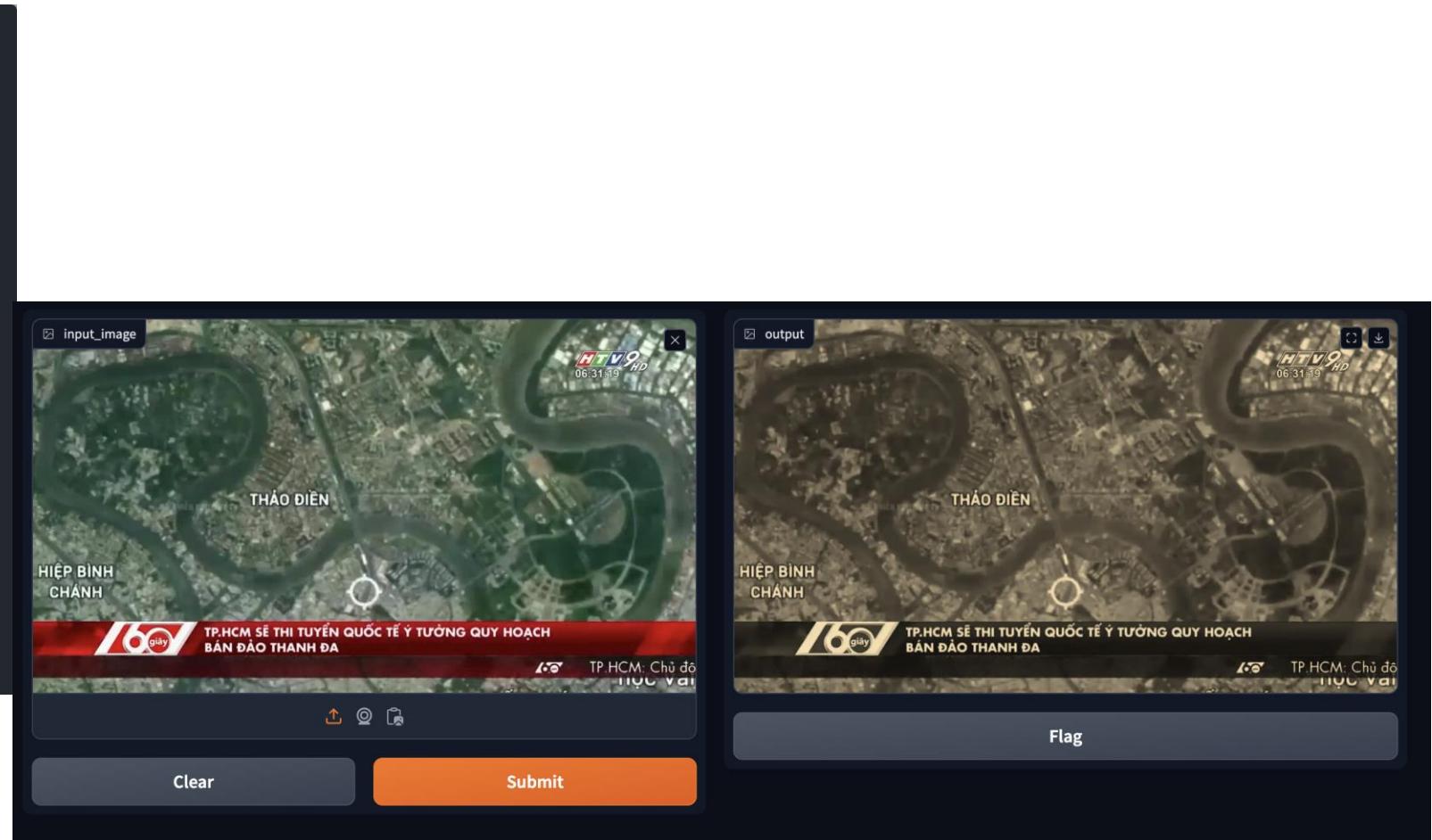
```
● ● ●  
1 import gradio as gr  
2  
3 def choice_device(demand):  
4     if demand == "Game":  
5         return "Windows"  
6     elif demand == "Deep Learning":  
7         return "Linux"  
8     else:  
9         return "MacOS"  
10  
11 demo = gr.Interface(  
12     fn=choice_device,  
13     inputs=[gr.Radio(["Game", "Deep Learning", "Web Development"])],  
14     outputs=["text"],  
15 )  
16 demo.launch()
```



Interface

❖ Input element

```
● ● ●  
1 import numpy as np  
2 import gradio as gr  
3  
4 def filter(input_image):  
5     sepia_filter = np.array([  
6         [0.393, 0.769, 0.189],  
7         [0.349, 0.686, 0.168],  
8         [0.272, 0.534, 0.131]  
9     ])  
10    sepia_img = input_image.dot(sepia_filter.T)  
11    sepia_img /= sepia_img.max()  
12    return sepia_img  
13  
14 demo = gr.Interface(  
15     fn=filter,  
16     inputs=["image"],  
17     outputs=["image"],  
18 )  
19 demo.launch()
```

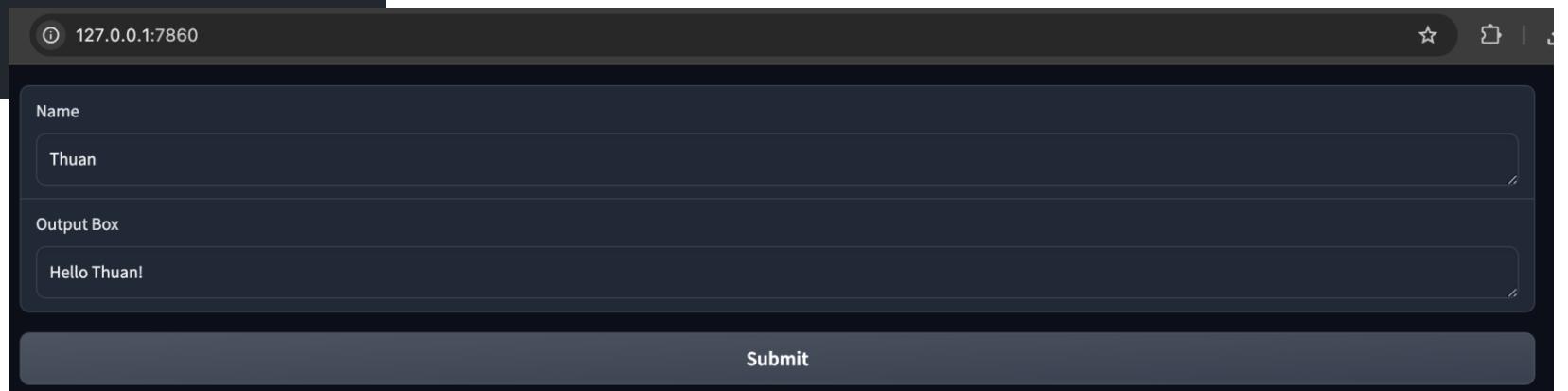


Gradio with Block

Block

❖ Click event

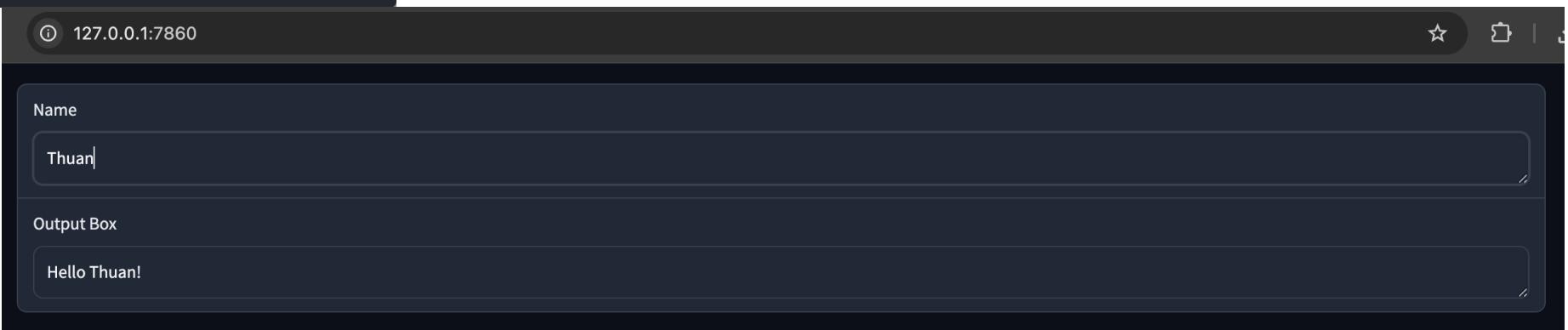
```
● ● ●  
1 import gradio as gr  
2  
3 def greet(name):  
4     return "Hello " + name + "!"  
5  
6 with gr.Blocks() as demo:  
7     input = gr.Textbox(label="Name")  
8     output = gr.Textbox(label="Output Box")  
9  
10    greet_btn = gr.Button("Submit")  
11    greet_btn.click(fn=greet, inputs=input, outputs=output)  
12  
13 demo.launch()
```



Block

❖ Change event

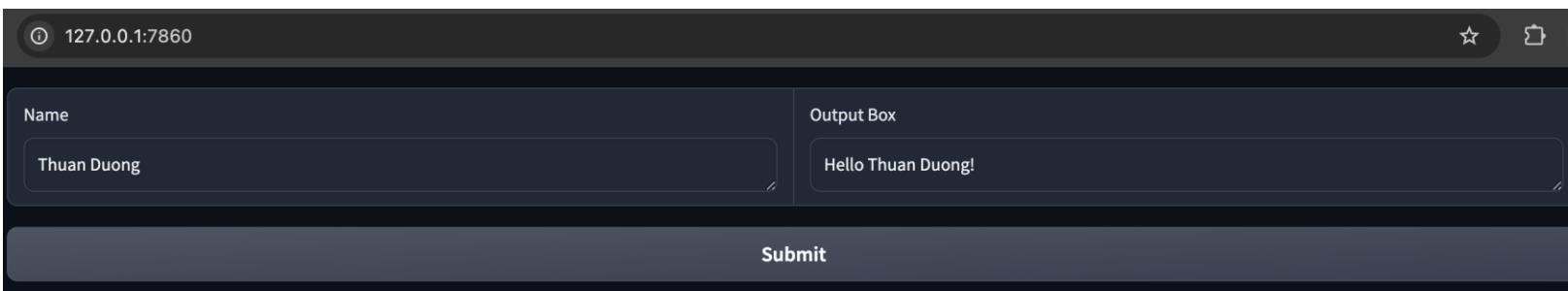
```
● ● ●  
1 import gradio as gr  
2  
3 def greet(name):  
4     return "Hello " + name + "!"  
5  
6 with gr.Blocks() as demo:  
7     input = gr.Textbox(label="Name")  
8     output = gr.Textbox(label="Output Box")  
9  
10    input.change(fn=greet, inputs=input, outputs=output)  
11  
12 demo.launch()
```



Block

❖ Layout

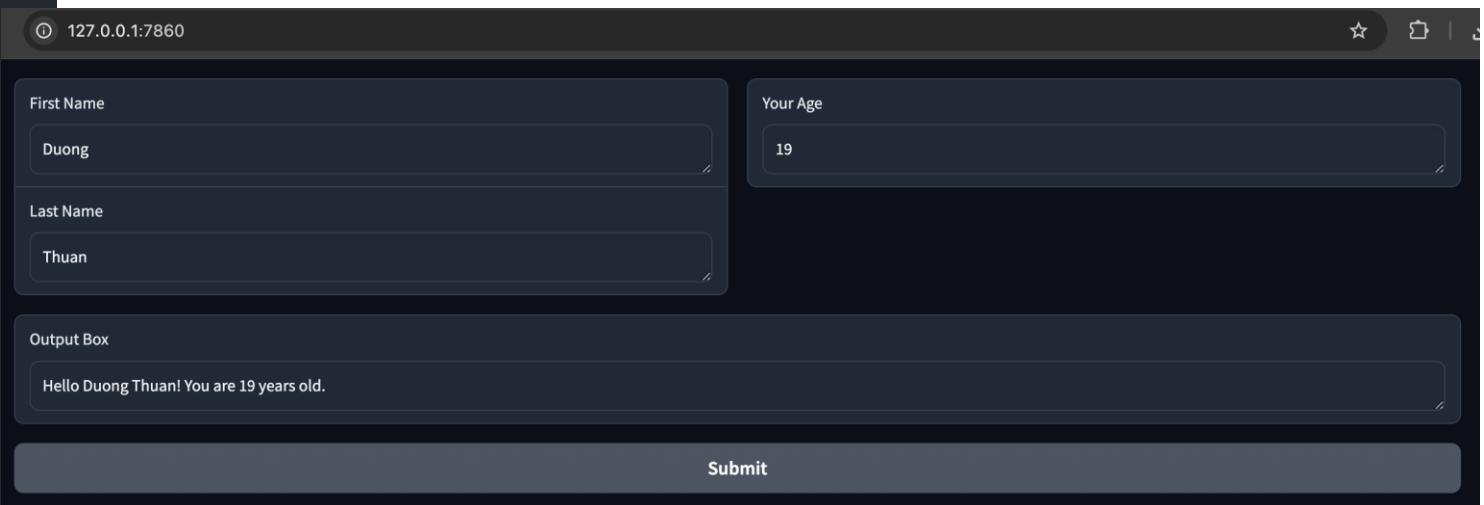
```
● ● ●  
1 import gradio as gr  
2  
3 def greet(name):  
4     return "Hello " + name + "!"  
5  
6 with gr.Blocks() as demo:  
7     with gr.Row():  
8         input = gr.Textbox(label="Name")  
9         output = gr.Textbox(label="Output Box")  
10        greet_btn = gr.Button("Submit")  
11        greet_btn.click(fn=greet, inputs=input, outputs=output)  
12  
13 demo.launch()
```



Block

❖ Layout

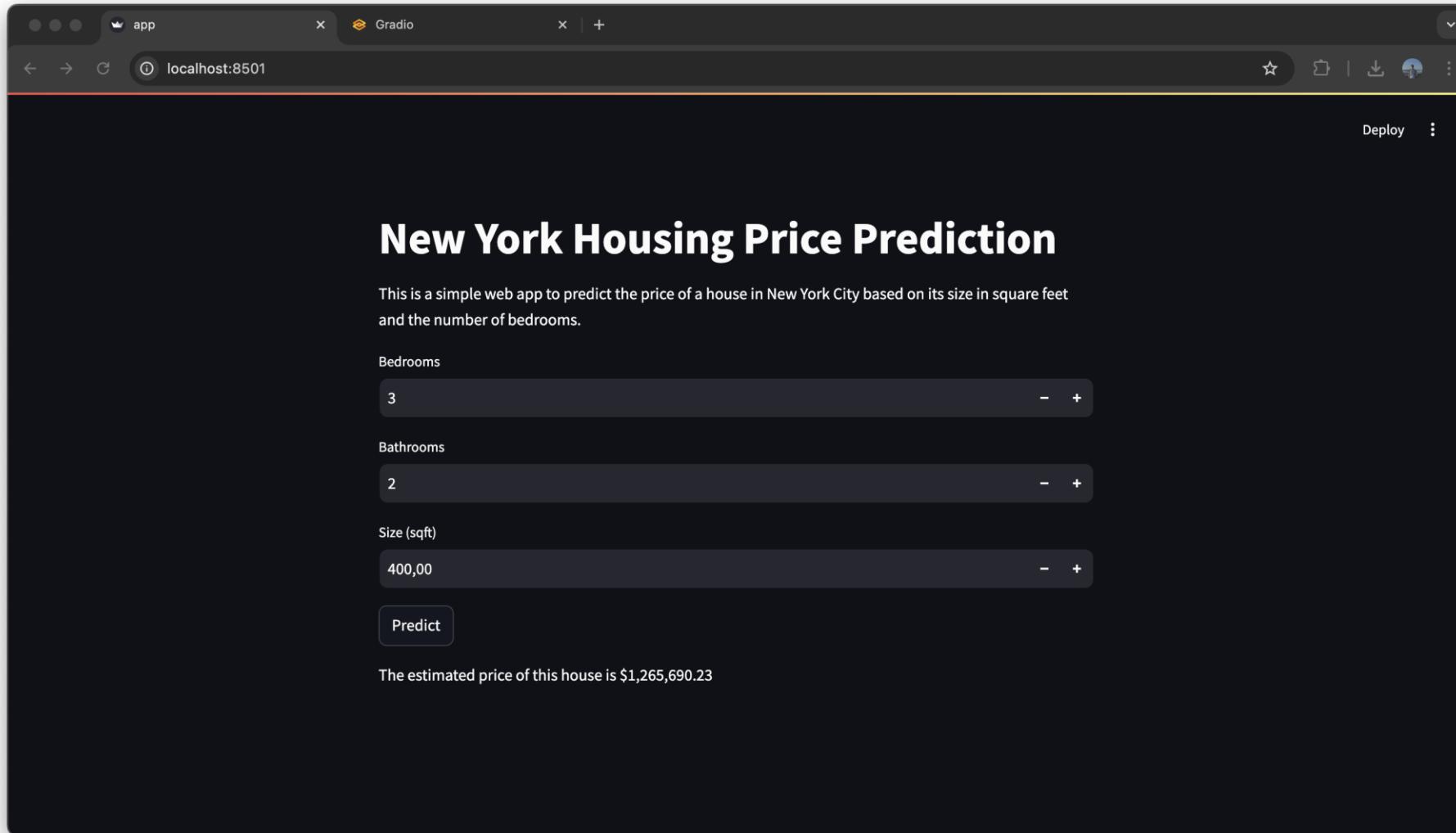
```
● ● ●  
1 import gradio as gr  
2  
3 def greet(first_name, last_name, age):  
4     output_1 = "Hello " + first_name + " " + last_name  
5     output_2 = "! You are " + age + " years old."  
6     return output_1 + output_2  
7  
8 with gr.Blocks() as demo:  
9     with gr.Row():  
10         with gr.Column():  
11             first_name = gr.Textbox(label="First Name")  
12             last_name = gr.Textbox(label="Last Name")  
13         with gr.Column():  
14             age = gr.Textbox(label="Your Age")  
15  
16     with gr.Row():  
17         output = gr.Textbox(label="Output Box")  
18  
19     greet_btn = gr.Button("Submit")  
20     greet_btn.click(fn=greet,  
21                     inputs=[first_name, last_name, age],  
22                     outputs=[output])  
23  
24 demo.launch()
```



Hands-on

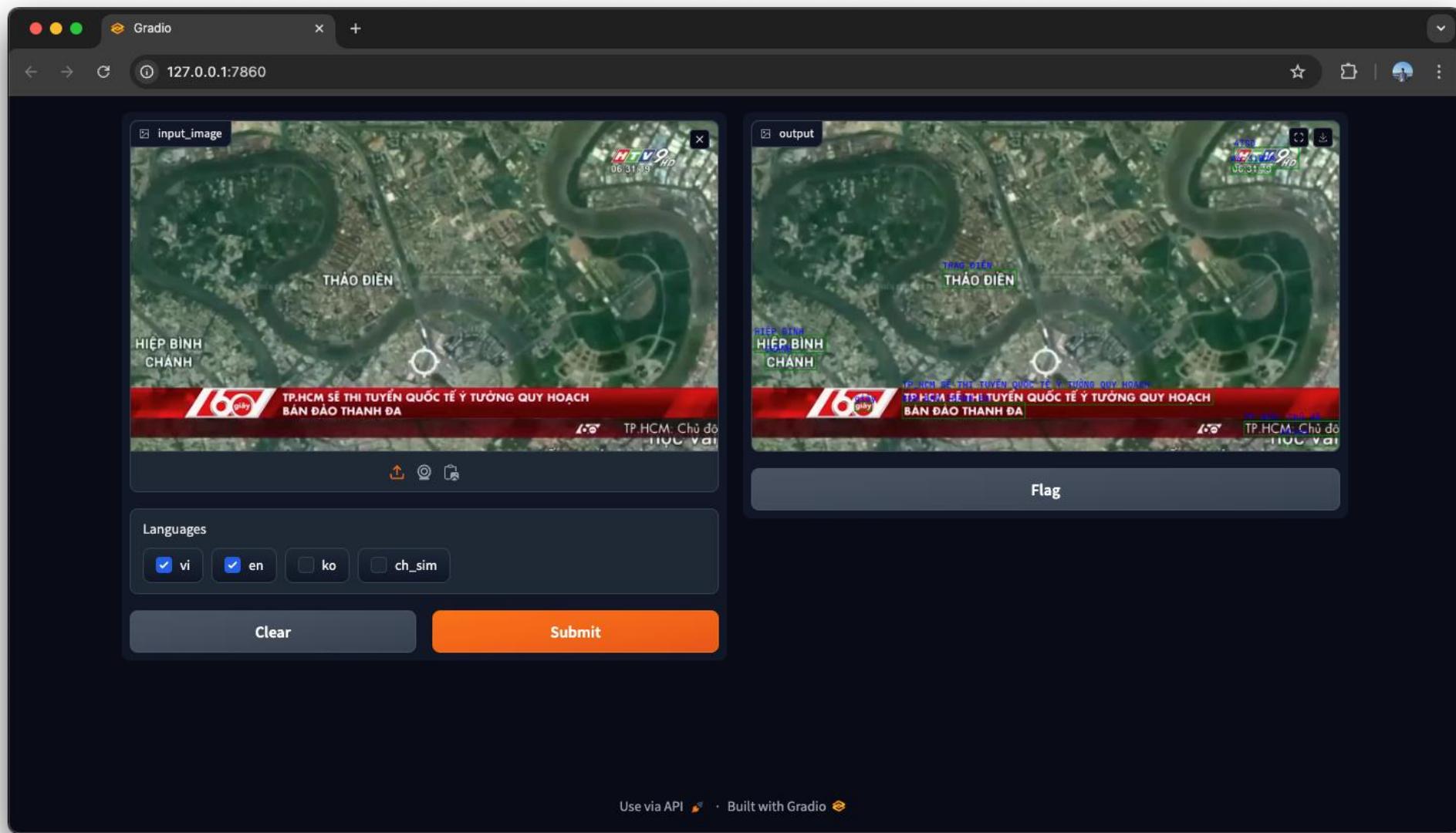
Hands-on

❖ Housing Price



Hands-on

❖ OCR



Conclusion

Conclusion



Question

