|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **PROJECT 3** |  | |  |  | |
| **Image-to-Text by CHIHAB EL OIDI**  Under supervision of Professor ML/DeepLearning: Dr. Abdelhak MAHMOUDI |

# Made using Gradio: Demo



## **Try it yourself**

[https://ceoeloidi-image-to-text-by-chihab-el-oidi-gradio.hf.space](https://ceoeloidi-image-to-text-by-chihab-el-oidi-gradio.hf.space/)

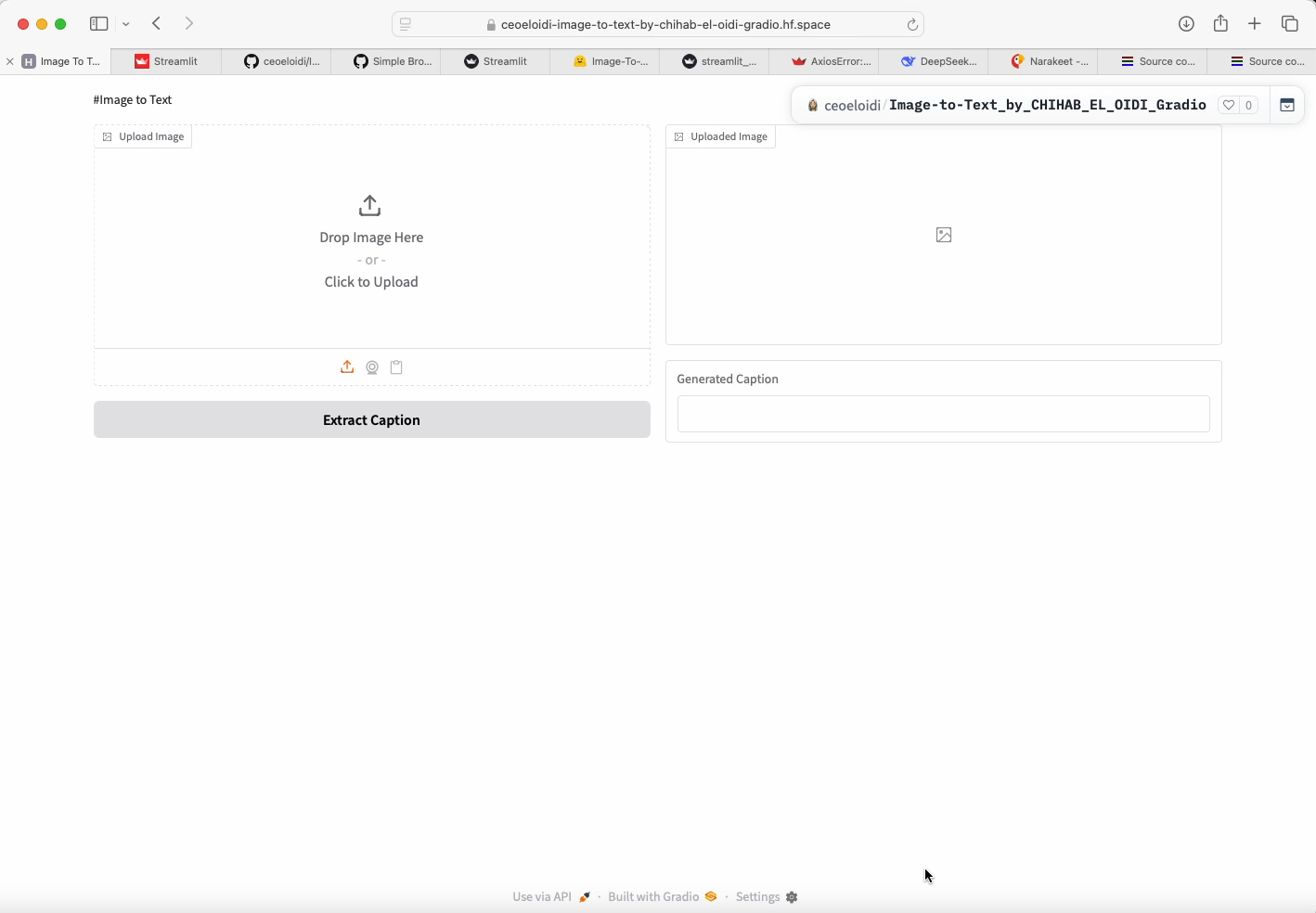
## **Setup and Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | **from** **transformers** **import** pipeline  **import** **gradio** **as** **gr**  **from** **PIL** **import** Image  *# Initialize the image captioning pipeline*  captioner = pipeline(“image-to-text”, model=”ydshieh/vit-gpt2-coco-en”)  **def** generate\_caption(image):  *“””Generate caption from uploaded image”””*  **if** image **is** **None**:  **return** **None**, “Please upload an image”    *# Open image and generate caption*  img = Image.open(image)  result = captioner(img)[0][‘generated\_text’]  **return** img, result *# Return both image and text*  *# Create Gradio interface*  **with** gr.Blocks(title=”Image To Text”) **as** app:  gr.Markdown(“#Image to Text”) *# Optional header for display*  **with** gr.Row():  **with** gr.Column():  upload\_file = gr.Image(type=”filepath”, label=”Upload Image”)  submit = gr.Button(“Extract Caption”)    **with** gr.Column():  output\_image = gr.Image(label=”Uploaded Image”, interactive=**False**)  output\_text = gr.Textbox(label=”Generated Caption”)  submit.click(  fn=generate\_caption,  inputs=upload\_file,  outputs=[output\_image, output\_text]  )  app.launch(share=**True**) *# Run the app* |

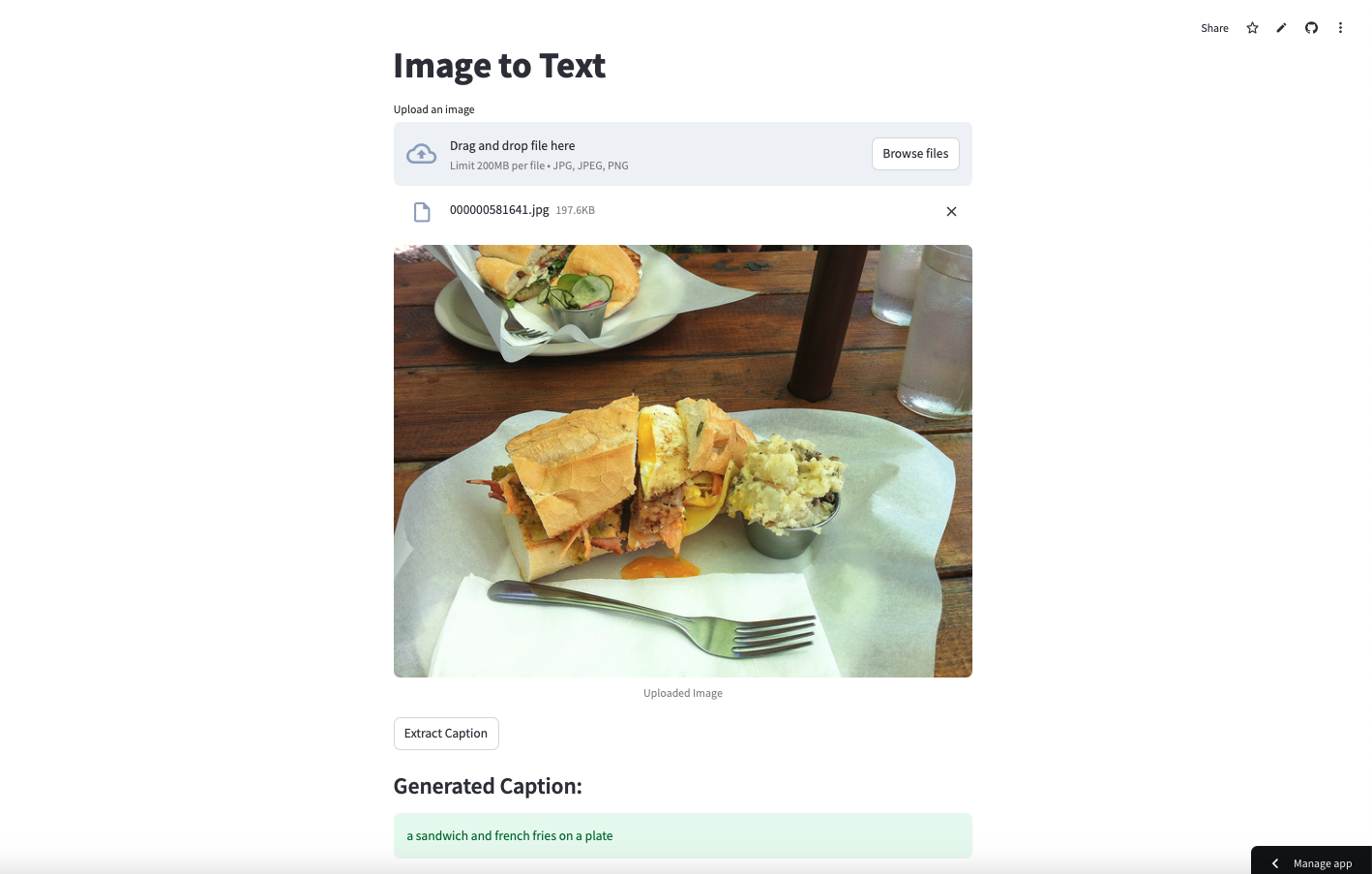
## **Result Demo**



## **Demo Animated**

****

# Made using Streamlit: Demo



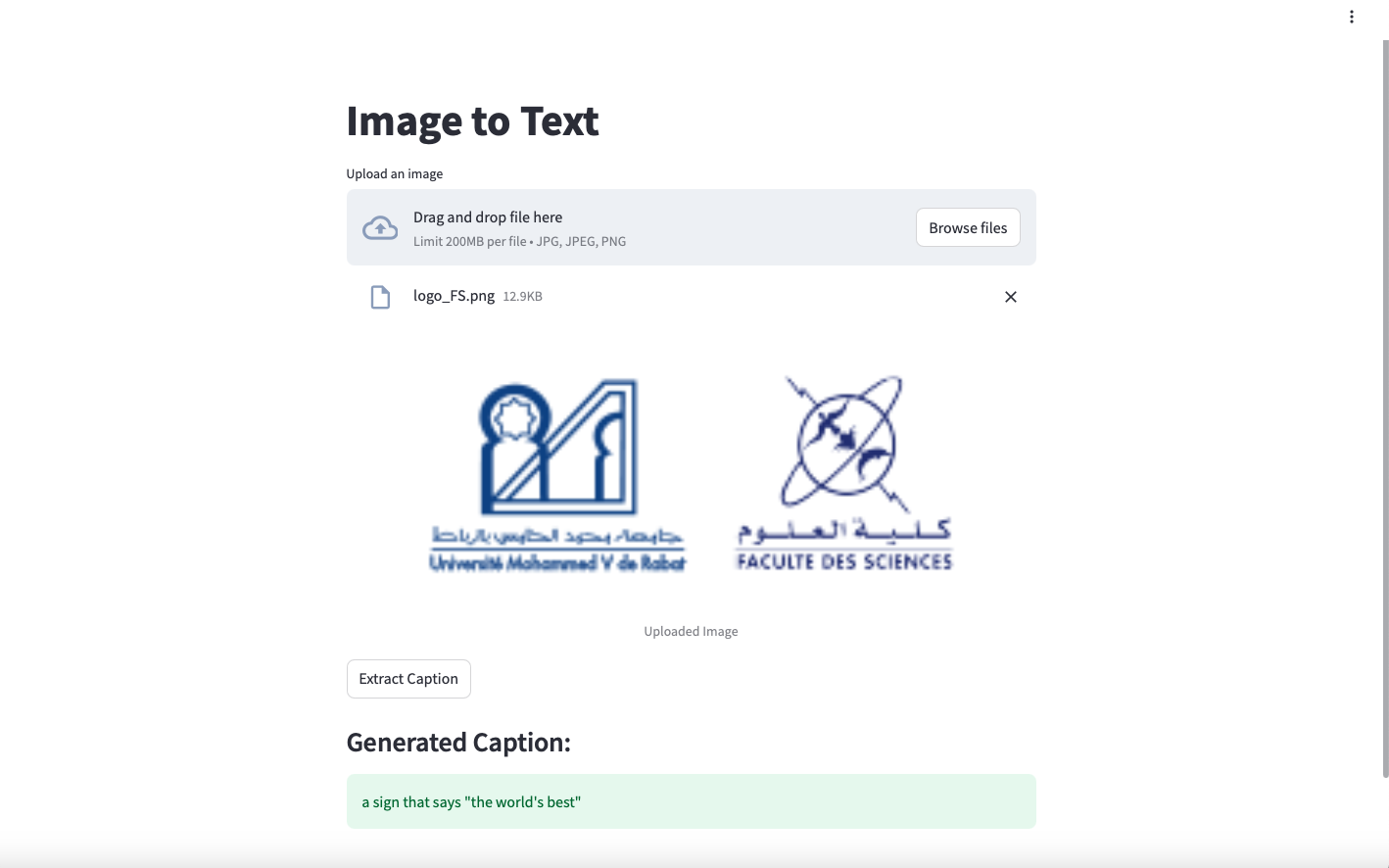
## **Try it yourself**

[https://image-to-textbychihabeloidi.streamlit.app](https://image-to-textbychihabeloidi.streamlit.app/)

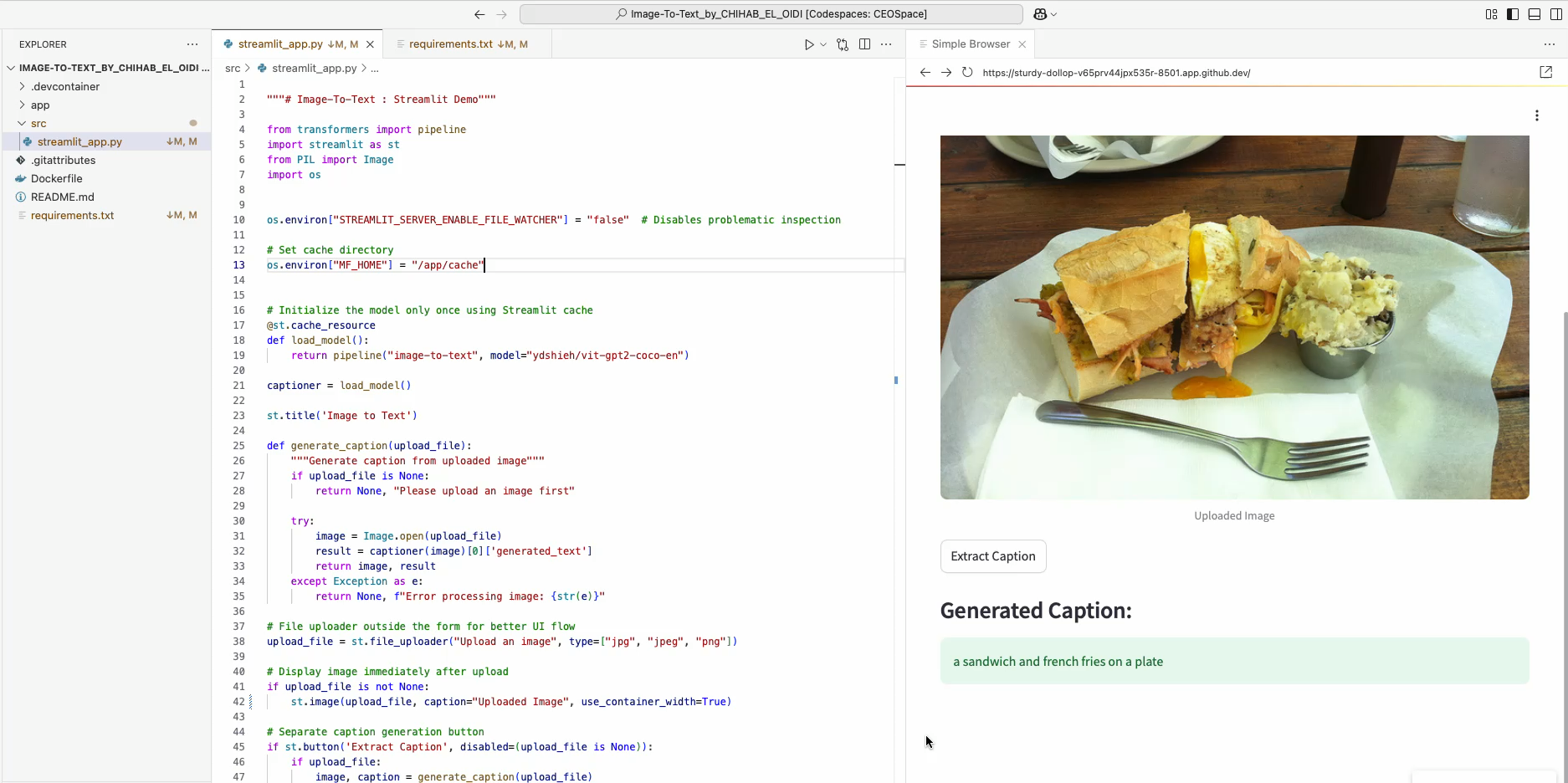
1. **Environment Setup and Code**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | *"""# Image-To-Text : Streamlit Demo"""*  **from** **transformers** **import** pipeline  **import** **streamlit** **as** **st**  **from** **PIL** **import** Image  **import** **os**  os.environ["STREAMLIT\_SERVER\_ENABLE\_FILE\_WATCHER"] = "false" *# Disables problematic inspection*  *# Set cache directory*  os.environ["MF\_HOME"] = "/app/cache"  *# Initialize the model only once using Streamlit cache*  @st.cache\_resource  **def** load\_model():  **return** pipeline("image-to-text", model="ydshieh/vit-gpt2-coco-en")  captioner = load\_model()  st.title('Image to Text')  **def** generate\_caption(upload\_file):  *"""Generate caption from uploaded image"""*  **if** upload\_file **is** **None**:  **return** **None**, "Please upload an image first"  **try**:  image = Image.open(upload\_file)  result = captioner(image)[0]['generated\_text']  **return** image, result  **except** **Exception** **as** e:  **return** **None**, f"Error processing image: {str(e)}"  *# File uploader outside the form for better UI flow*  upload\_file = st.file\_uploader("Upload an image", type=["jpg", "jpeg", "png"])  *# Display image immediately after upload*  **if** upload\_file **is** **not** **None**:  st.image(upload\_file, caption="Uploaded Image", use\_container\_width=**True**)  *# Separate caption generation button*  **if** st.button('Extract Caption', disabled=(upload\_file **is** **None**)):  **if** upload\_file:  image, caption = generate\_caption(upload\_file)  **if** image:  st.subheader('Generated Caption:')  st.success(caption)  **else**:  st.warning("Please upload an image first") |

## **Result Demo**



## **Demo Animated**

****