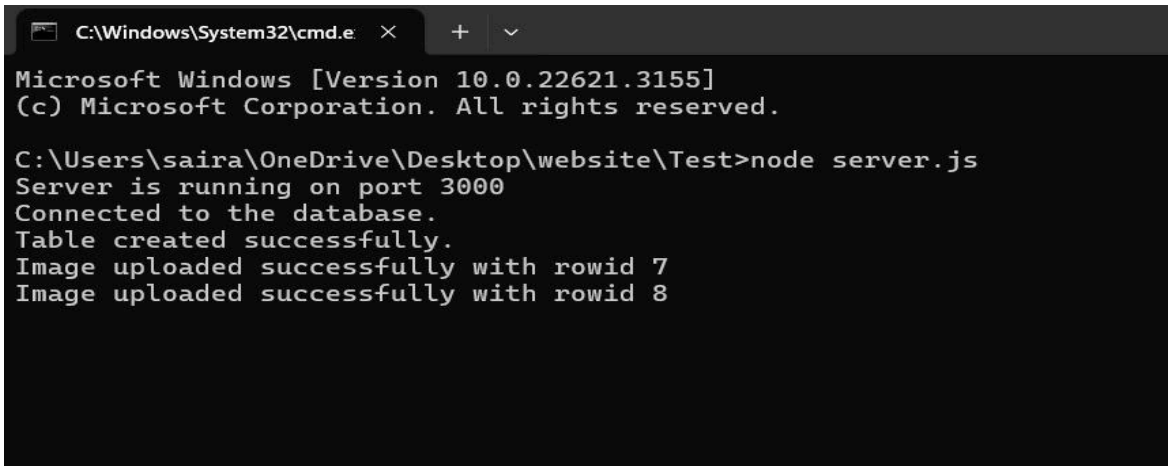
MediCaption Readme

Github link: https://github.com/aarthi03/MediCaption.git

MediCaption Prototype Explanation:
We have created a sample UI to upload a Medicine Image for the medicine name detection.
Sample UI: For the sample UI we have developed a web page to upload images by user and storing it into a database.
We can access the UI from the link: http://localhost:3000/
We have uploaded an image:

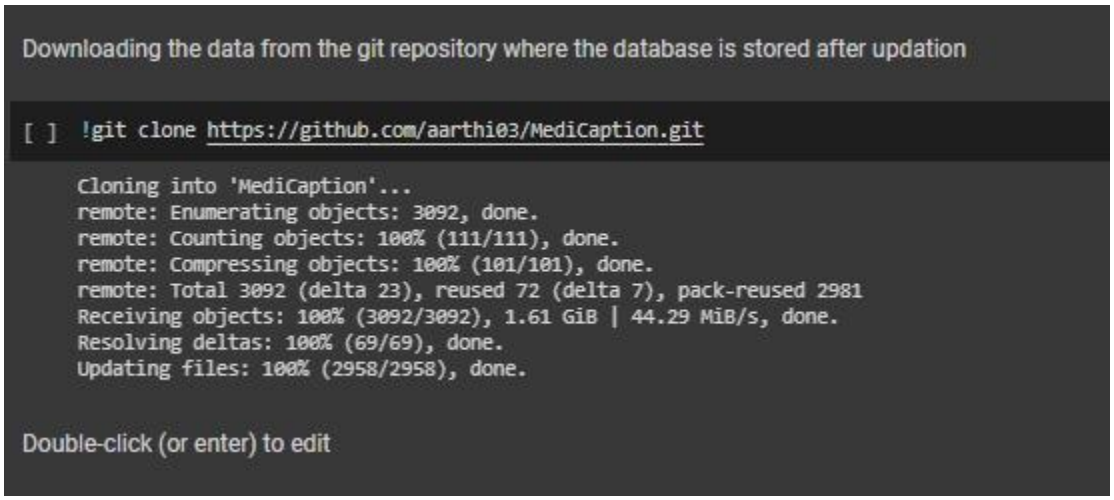
Input image:

Initialization of server.js file which helps us connect html file with the database. For every file we upload we can see the DB updation from server.js



Initially we trained the model using: “YoloV8_TRAIN,VALID,TEST.ipynb” From this file we got the best.pt with the model weights.

We have extracted the content of the database file by uploading the database into the git folder and we have also extracted the git folder files into the finalprototype.ipynb



The file best.pt is obtained while training the data using the code "YoloV8_TRAIN,VALID,TEST.ipynb" and then the file is stored in the GIT repository so that we can use the best.pt file which contains the weights of the model to predict the bounding boxes for the images that are being uploaded

From the image uploaded we have extracted the text of the medicine name by using OCR and Trained YoloV8 model :

```
# Perform OCR on the image
text1 = ocrfunc(image)

# # Print the extracted text
# print(image)
# print(text)

# cleaned_text = text.replace('\n', '')
# Initialize an empty list
cropped_images_list.append(image_path)
image_list.append(image)
fname.append(file_name)

items_list.append(text1)
# Display the image and extracted text
plt.imshow(image)
plt.axis('off')
plt.title(f"Text extracted from {file_name}:\n{text1}")
plt.show()
```

Based on the Medicine Name which we have obtained from the above steps We have generated the details about the medicine and the corresponding audio file.

```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
pd.set_option('display.max_colwidth', None)

output_text = ""
full_output_for_mp3 = ''

if not medicine_info.empty:
    for index, row in medicine_info.iterrows():
        output_text = ""
        output_text += f"Medicine Information:\n"
        output_text += f"- Name: {row['Name']}\n"
        output_text += f"- Generic Name(s): {row['GENERIC NAME(S)']}\n"
        output_text += f"- Uses: {row['Uses']}\n"
        output_text += f"- Side Effects: {row['Side Effects']}\n"
        # Display images associated with file names
        file_name = row['File_Path']
        image_path = f"//content/ultralytics/runs/detect/predict/crops/Medicine Name/{file_name}"
        img = Image.open(image_path)
        output_text += "\n"
        full_output_for_mp3 += output_text
        print(output_text)
        plt.imshow(img)
        plt.axis('off')
        plt.show()
```

```
Medicine Information:
- Name: Ciprofar
- Generic Name(s): Ciprofloxacin
- Uses: Treating bacterial infections such as urinary tract infections, respiratory tract.
- Side Effects: Nausea, vomiting, diarrhea, Headache, dizziness, Photosensitivity.
```