



# Team Vision HealthCare



**KRYPTTHON**

## **OUR\_TEAM\_VISION**

M Sravani  
Aparna S  
Harshitha M  
Korrayi Saiteja



## PROBLEM STATEMENT

Here is the short brief intro of our project "**Web interface for disease prediction**"

**Problem statement:** To create a **user-friendly** web interface "**Swasthgarud**" for disease prediction. In this project we are going to predict diseases like Pneumonia, Cardiovascular disease, Malaria from X-ray, CSV datasets and Blood smear respectively by creating a machine learning model.



## INTRODUCTION

### PNEUMONIA

- Image Dataset
- Lung infection
- Detected from X-Ray images.
- CNN Model of Machine learning

### HEART DISEASE

- ECG, BP Dataset.
- Affect heart and Blood Vessels.
- Detected from ECG.
- Random Forest(RF)

### MALARIA

- Image Dataset
- Red Blood cells
- Detected from images of blood samples.
- CNN Model of Machine Learning.



## PROPOSED SOLUTION

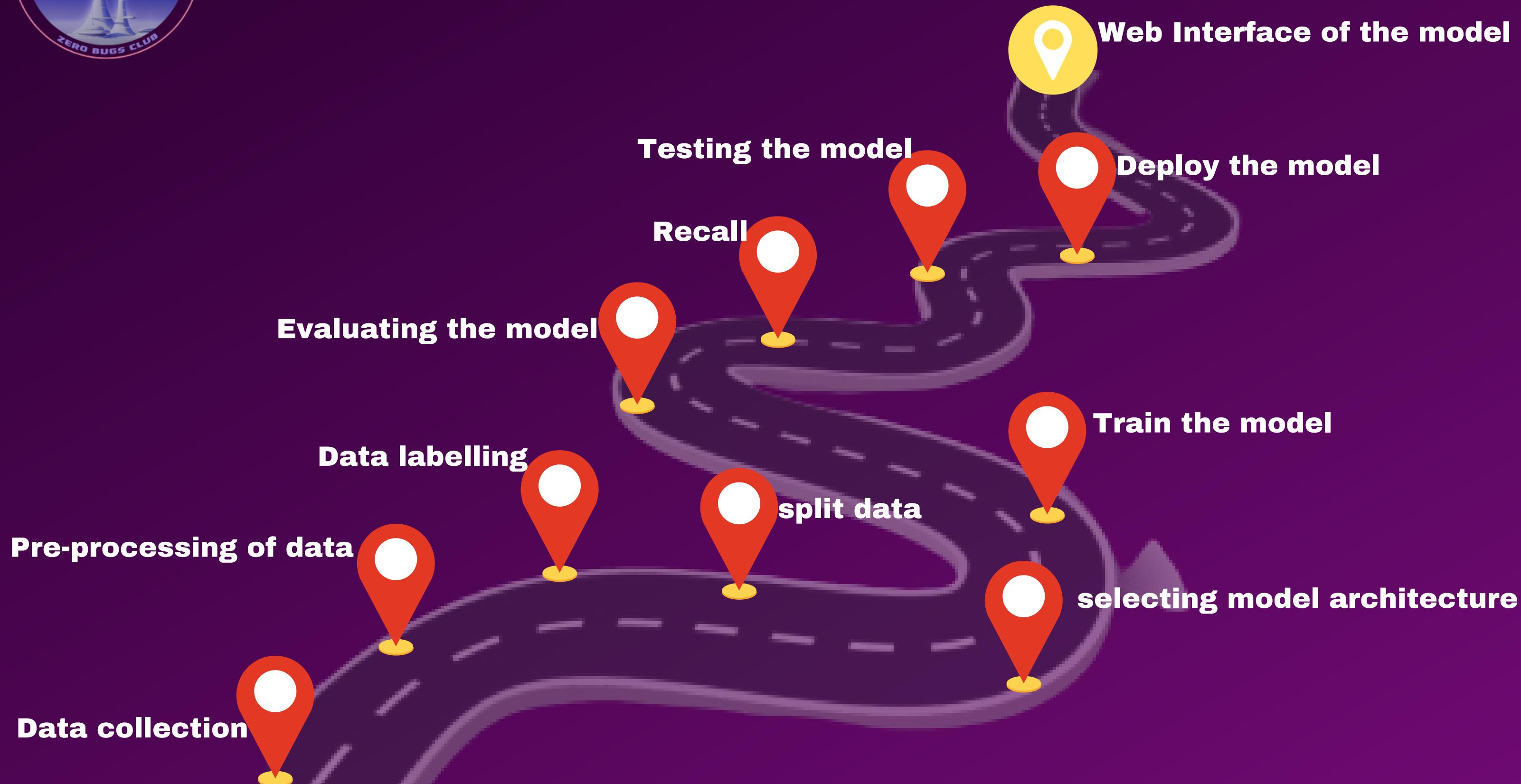
KRYPTTHON

- Collect the **datasets of Malaria, Pneumonia, Heart Diseases.**  
The following are the datasets used.
  - Malaria - Image dataset of **Red blood cells**
  - Pneumonia - **X-ray** images
  - Heart diseases - **ECG,BP.**
  - Select a machine learning model **CNN**.
  - Train the model using training set.
  - Evaluate the model.
  - Deploy the model using **Streamlit**.
  - Deploy **Streamlit** into the web app.
  - We created a web app named **Swasth Garud**.



# ROADMAP OF THE MODEL

KRYPTHON





# KRYPTON

## TOOLS/TECHNOLOGIES

ADOBE XD

→ To create prototype of our model

STREAMLIT

→ To deploy machine learning model into web interface

BACK-END

→ Python

## MACHINE LEARNING ALGORITHMS

- CNN (CONVOLUTIONAL NEURAL NETWORK)
- RF (RANDOM FOREST)



# KRYPTTHON

## IMPLEMENTATION PLANS

Pneumonia Detection using CNN.ipynb - Visual Studio Code

C:\> Users > saite > Downloads > Pneumonia-Detection-web-app-using-Streamlit-main > Pneumonia-Detection-web-app-using-Streamlit-main > Pneumonia Detection using CNN.ipynb > from tensorflow.keras.preprocessing import image

+ Code + Markdown ... Select Kernel

```
for i, _set in enumerate(['train', 'val', 'test']):
    set_path = input_path+_set
    ax[i].imshow(plt.imread(set_path+'/NORMAL/'+os.listdir(set_path+'/NORMAL')[6]), cmap='gray')
    ax[i].set_title('Set: {}, Condition: Normal'.format(_set))
    ax[i+3].imshow(plt.imread(set_path+'/PNEUMONIA/'+os.listdir(set_path+'/PNEUMONIA')[0]), cmap='gray')
    ax[i+3].set_title('Set: {}, condition: Pneumonia'.format(_set))
```

[2]

Python

The Marketplace has extensions that can help with '.ipynb' files

Search Marketplace Don't Show Again for '.ipynb' files

## CNN MODEL OF PNEUMONIA



# KRYPTTHON

## CNN MODEL OF MALARIA

```
File Edit Selection View Go Run Terminal Help
Deep_Learning_Custom_Images_Malaria (1).ipynb - Visual Studio Code
main.py Pneumonia Detection using CNN.ipynb Deep_Learning_Custom_Images_Malaria (1).ipynb X
C: > Users > saite > Downloads > malaria > Deep_Learning_Custom_Images_Malaria (1).ipynb > M+empty cell
+ Code + Markdown ...
plt.imshow(para_img)
[10] ... <matplotlib.image.AxesImage at 0x26eb38bdeb0>
</>
0 20 40 60 80 100 120 140
0 25 50 75 100 125
para_img.shape
[11] ... (148, 142, 3)
Python
uninfected_cell_path = train_path+'/uninfected/'+os.listdir(train_path+'/uninfected')[0]
uninfected_cell = imread(uninfected_cell_path)
plt.imshow(uninfected_cell)
[12] ... <matplotlib.image.AxesImage at 0x26eb5b024f0>
</>
```



# KRYPTON

## RANDOM FOREST MODEL OF HEART DISEASE

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows files: main.py, app.py, prediction.py, and heart(using random\_forest).ipynb.
- Code Editor:** The active file is heart(using random\_forest).ipynb. It contains Python code for a machine learning model.

```
p = model.score(x_testScaler,y_test)
print(p)

...
print('Classification Report\n', classification_report(y_test, y_pred))
print('Accuracy: {}%'.format(round(accuracy_score(y_test, y_pred)*100),2))

cm = confusion_matrix(y_test, y_pred)
print(cm)

...
Classification Report
precision    recall   f1-score   support
          0       0.82      0.82      0.82      38
          1       0.81      0.81      0.81      37
   accuracy           0.81      0.81      0.81      75
  macro avg       0.81      0.81      0.81      75
weighted avg       0.81      0.81      0.81      75

Accuracy: 81.33%
[[[31  7]
 [ 7 30]]]

filename = 'heart.pkl'
pickle.dump(model, open(filename, 'wb'))
```
- Terminal:** Shows the output of the code execution.
- Status Bar:** Shows the path C: > Users > saite > OneDrive > Desktop > bio > models > heart(using random\_forest).ipynb > import numpy as np.



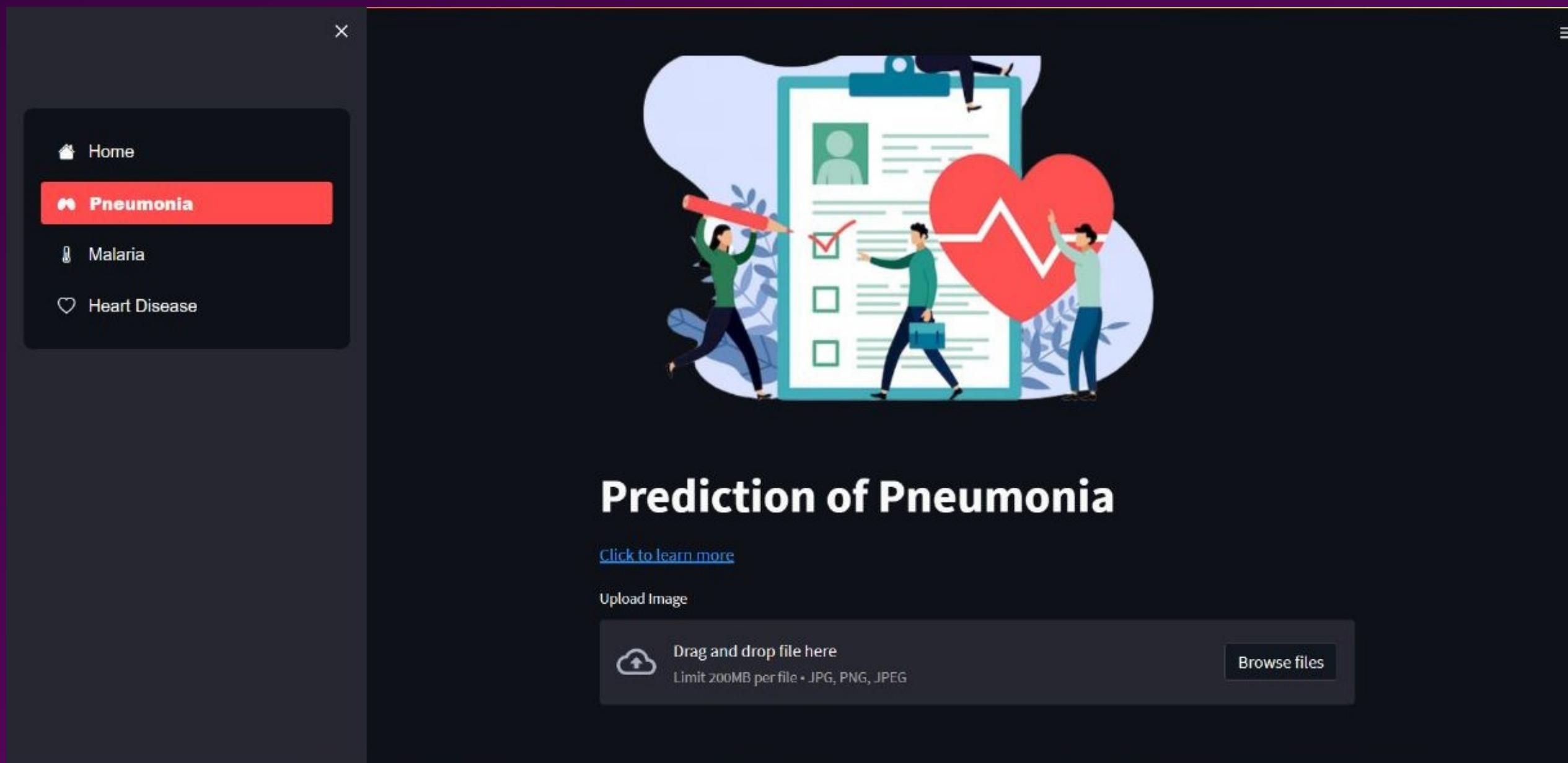
# KRYPTTHON

## LANDING PAGE OF SWASTH GARUD WEB INTERFACE



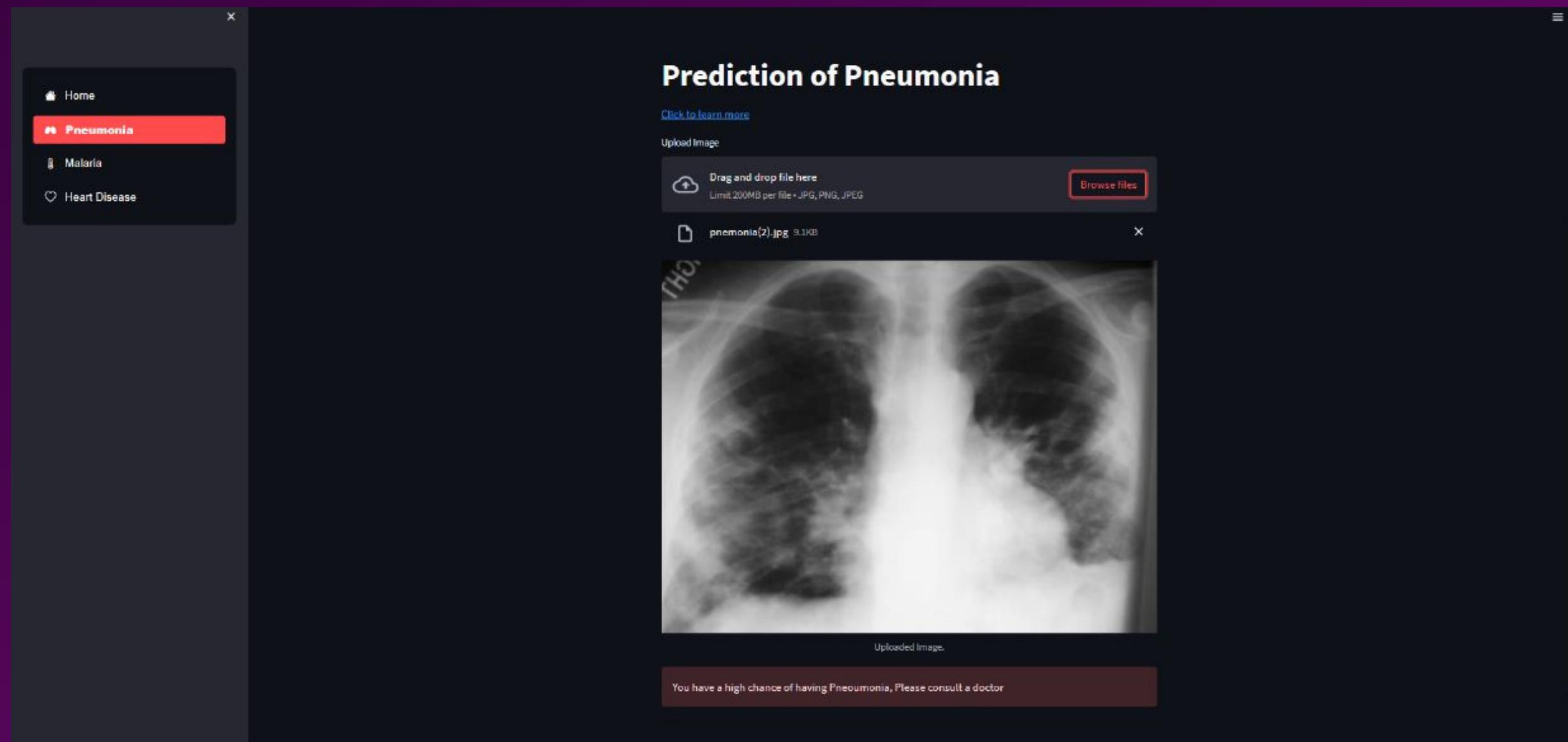
# PREDICTION OF PNEUMONIA

- Click on Browse files and upload the X-Ray of the patient.



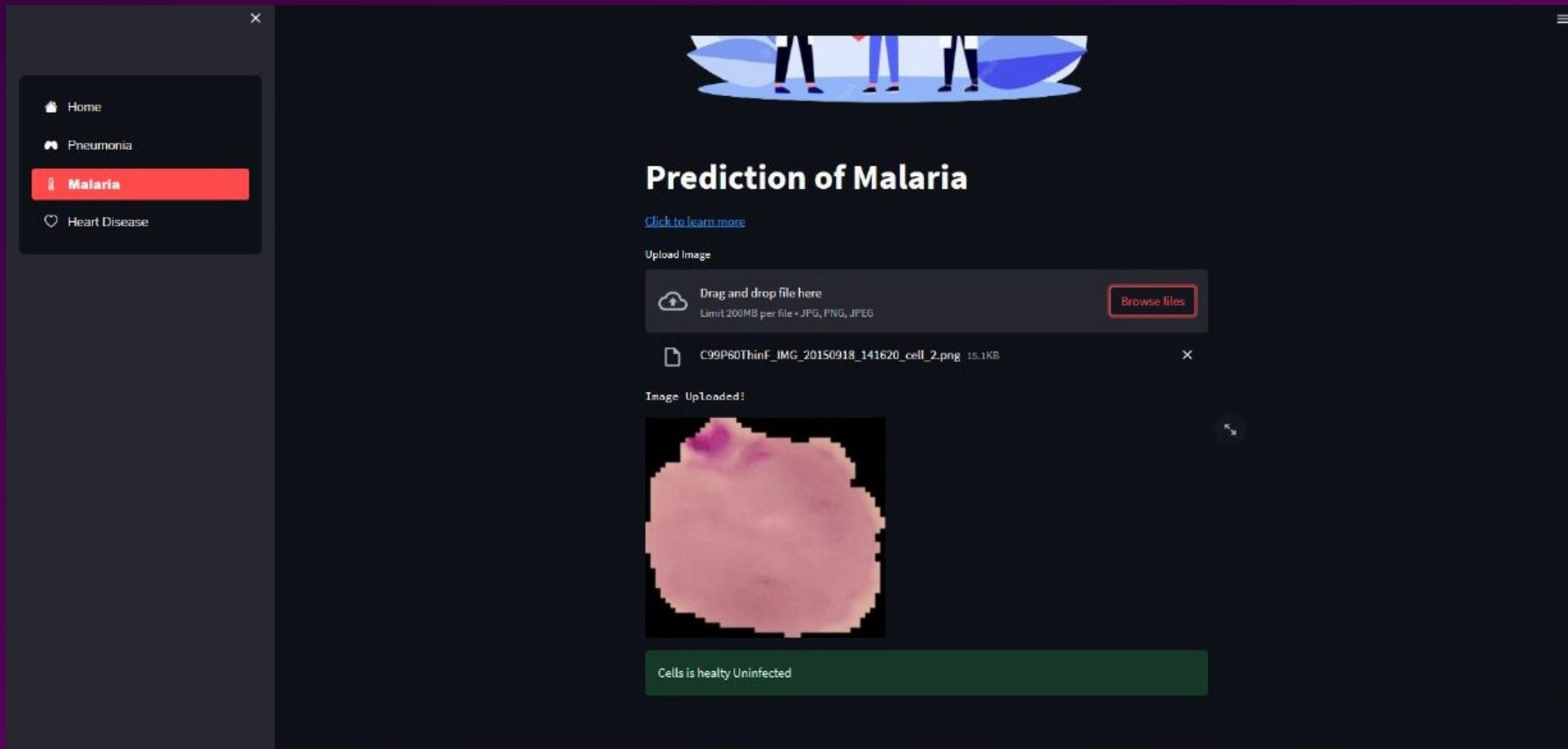
# PREDICTIIION OF PNEUMONIA

- X-Ray of a patient who is suffering from Pneumonia and the result is you have a high chance of Pneumonia.



# PREDICTION OF MALARIA

- X-Ray image of blood cells of a person who is healthy and the result is cells are healthily uninfected.



# PREDICTION OF HEART DISEASE

- By using Random Forest we are predicting heart disease

The screenshot shows a web-based application titled "Prediction of Heart Disease". On the left, there is a sidebar menu with options: Home, Pneumonia, Malaria, and Heart Disease, with "Heart Disease" highlighted in red. The main content area has a dark background with white text and features several input fields:

- select your age:** A slider with a red dot at 15, ranging from 15 to 100.
- Sex:** A dropdown menu set to "Male".
- Constrictive pericarditis:** A section with two radio buttons: "Yes" (selected) and "No".
- Heart Beat:** A slider with a red dot at 45, ranging from 45 to 180.
- Cholesterol:** Two sections, both with sliders. The first slider has a red dot at 108, ranging from 108 to 300. The second slider has a red dot at 88, ranging from 88 to 300.
- Rest ecg:** A section with a slider having a red dot at 0, ranging from 0 to 2.
- Maximum Heart Rate:** A section with a slider having a red dot at 0, ranging from 0 to 2.

Each input field includes a "Click to learn more" link in blue text above it.

# PREDICTION OF HEART DISEASE

The disease will be detected once the heart rate, cholesterol, and ECG levels of a patient are known.

Home

Pneumonia

Malaria

Heart Disease

Maximum Heart Rate  
100

Exercise induced angina: [Click to learn more](#)

Yes

No

Oldpeak  
0.80

Slope  
3

Coronary Artery Disease: [Click to learn more](#)

Thallium stress result: [Click to learn more](#)

Predict

There is chance of Heart disease



Datasets are classified by the Model

The ML model is deployed in Streamlit

Feature detection layer of CNN is trained

# The 35% Progress

Further, the three CNN models are integrated into a main python file

Sample image dataset is fed and an accuracy of 94.98% is observed.

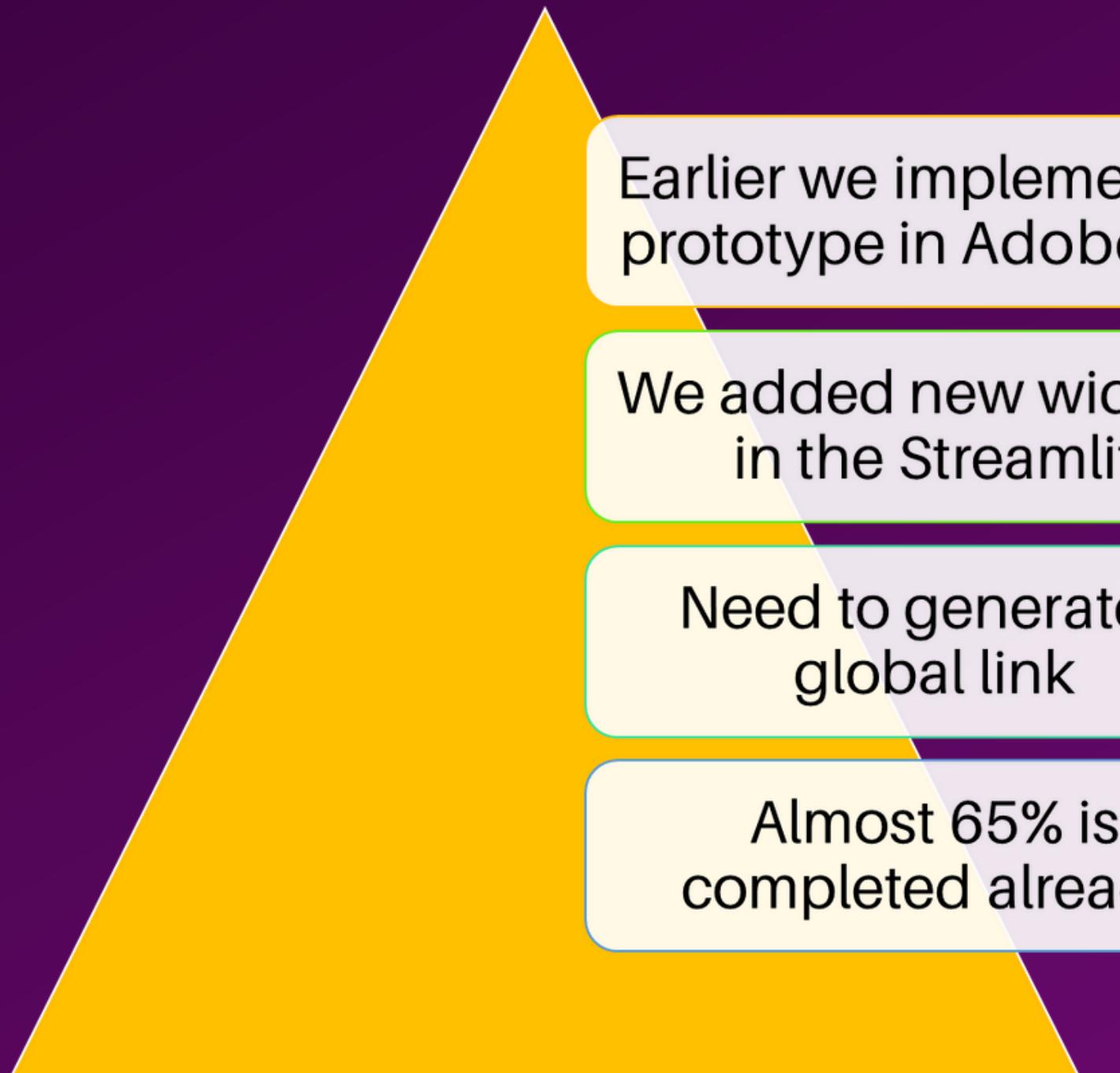
Three CNN models for three different diseases have been developed.



## POTENTIAL PROBLEMS FACED



# **Added features and recent updates**



Earlier we implemented prototype in Adobe XD

We added new widgets in the Streamlit

Need to generate a global link

Almost 65% is completed already

## FURTHER WORK

- We are planning to add some more diseases to predict.
- Advancing the features of the widgets.
- Usage of Session State and Callbacks.
- Converting from local host to the global host.

**Thank you**