# Malaria Detection by using Image Cells

**Here Ill describe in short that what I have done in malaria project Given by you**

## Approach:

### Tools used:

- VS code(Visual Studio code)
- Streamlit
- Jupyter kernel (inside VS code)
- Python-Programming
- Github(my trained saved model is not uploading because free user in Github can upload only till 25mb my saved trained model is 96mb)
- (NOTE:Also due to unable upload my trained model in github I cant able to deploy in streamlit,I have experience in deploying in streamlit and I can able to deploy in streamlit if paid account was given)
- (Rest I have done perfectly according to your instructions what you have given)

### Libraries used:

- Numpy
- Pandas
- Scikit-learn
- Tensorflow
- Jupyter
- Streamlit
- OpenCV-Python

## About Dataset:

The dataset you have provided me has image cells to identify malaria which has total of 27580 images and divided into two categories ,Parasitized-13780 and Uninfected contains 13780 image in .png format

This Dataset is a binary classification and we have to write code for binary classification

## APPROACH:

First we have to split into train and test data and dataset in raw format and not in train and test so I have done manually in a separate python file named test_train.py

So I have divided train-80% and test-20% so 22046 images belongs to train and 5512 belongs to test images

I used **ResNet50** for this dataset because I have already worked in ResNet model in my previous company and I have realtime experience and I have used in binary classification models and it works very well

**ResNet** has Deeper Architecture and captures complex relations in the image dataset, while I have also used VGG16,VGG19 it struggles with deeper architecture and I have experienced Vanishing Gradient Problem in my company while dataset I huge and we need to hyper tune ,it increases workload according to my previous work experience

So I have Applied Pretrained Transfer Learning CNN Model-**RESNET50** for this dataset

## Approach 1.1:

I have imported necessary libraries

# Data Augmentation:

This Basically rotates, flip, mirror, flip upside down etc to learn more about the images even if the images are vertical or horizontal etc.

To make our model train more I used Data Augmentation by using IMAGEDATAGENERATOR from TensorFlow

Then I applied ResNet50 to Augmented images

Finally I trained the model and I get the accuracy of 93.65%.

In **MAIN.PY** file I have done the streamlit code and run the model and it works fine

In code or in this document I have mentioned in short and crisp manner