



**K. J. Somaiya College of Engineering, Mumbai-77**  
(A constituent College of Somaiya Vidyavihar University)

**Batch: A4**

**Roll No.: 16010121208**

**Experiment No: 03**

**Group No: 12**

**Title: Prototype Implementation for the MiniProject.**

---

---

**Expected Outcome of Experiment:**

**CO3: Implement and prototype creation for the specified application.**

---

---

**Books/ Journals/ Websites referred:**

*[Students can mention websites/ books used in their project implementation]*

---

**Introduction:**

System Implementation uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder requirements and system requirements developed in the early life cycle phases. These system elements are then integrated to form intermediate aggregates and finally the complete system

The implementation and prototyping document should be presented with description of following steps.



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

**1. Modules Description:**

Write input-output, properties, scenarios of important modules in the code in the given format.

Client Side Rendering Architecture with Microservice Architecture is adopted for creating the Web App.

Module	Name, Definition, purpose
<i>Authentication Microservice</i>	<p><b>Definition:</b> <i>This stand-alone server of the application is used for issuing tokens.</i></p> <p><i>This auth server issues tokens after verifying credentials. This token is stored in the client's browser and used every time the browser makes a request to the Client facing server or Rest-API Server.</i></p>

Module	Name, Definition, purpose
<i>Client Facing Server Microservice</i>	<p><b>Definition:</b> <i>This server serves all the web pages and static content demanded by the end user.</i></p> <p><i>This server is the point of contact of the Web App with the end user (admins and doctors).</i></p>

Module	Name, Definition, purpose
<i>Rest-API Server Microservice</i>	<p><b>Definition:</b> <i>This server serves all the data using Rest-API's.</i></p> <p><i>This server serves the Rest API endpoints to access data. It does the authorization of user before returning date in the response.</i></p>

Module	Name, Definition, purpose
<i>Machine Learning Model Microservice</i>	<p><b>Definition:</b> <i>This model processes the images and gives the disease prediction</i></p> <p><i>This server serves the Machine Learning model to send and receive images and reply</i></p>



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

*Activity: Here, team members will mention important modules coding details and input-output parameters of respective modules.*

### *Authentication Microservice*

*This auth server issues tokens after verifying credentials. This token is stored in the client's browser and used every time the browser makes a request to the Client facing server or Rest-API Server.*

*Only this microservice server in the entire application possess the private key to sign the token. Other modules of the system verify the token by checking if the token is signed using the private key of authentication microservice. This verification is done using Public key. Authentication of different types of users is performed by viewing the "user\_type" in access token which is present in the payload of the token.*

*Only This microservice has access to the user credentials Database which stores the email and Password.*

*So this microservice receives login credentials and then verify them and then issue a token if the credentials are correct.*

*Once the token is logged out or expired : the other modules of the application will redirect to login page and this is how the authentication microservice is integral part of the application.*

### *Client Facing Server:*

*This server serves all the web pages and static content demanded by the end user.*

*This server is the point of contact of the Web App with the end user (admins and doctors). As a part of the scope of this application: this server also stores the images of the patients. This server also makes some calls to the Rest-API Server to fetch some data to perform some app logic or render some dynamically generated templates using jinja. This server is responsible for logging out the user by inserting the token into logged out tokens database.*

*This server is responsible for sending the image link to the ML/DL Model Server which does the prediction and then sends the response to Rest-API Server to insert into the Database.*

*The authorisation in certain protected routes exposed by this server are done by reading the access token from the cookie.*

### *Rest-API Server Microservice:*

*This server serves all the data using Rest-API's.*



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

*This server serves the Rest API endpoints to access data. It does the authorization of data access by reading the access token from the authorisation header of the request. The token also contains the user type in the payload. This is how the authentication of different users is performed. Each route “Depends” on auth module created in the server to verify the authorisation header [Concept of Dependency Injection utilised here]*

*This server has 4 main layers.*

*The 1<sup>st</sup> layer is of the Pydantic Models. They ensure that the correct format of data is present in the HTTP Body.*

*The 2<sup>nd</sup> layer is of authorisation.*

*The 3<sup>rd</sup> layer is the application logic within the route.*

*The 4<sup>th</sup> layer is Data Abstraction which connects to the database and gets the data (by ensuring safely opening and closing the connection)*

*Machine Learning Model Microservice:*

*This server runs the ML/DL model to do prediction.*

*This purpose of this server is to run gastrointestinal disease detection.*

*This server accepts the link of the image and then calls the client facing server to fetch the image and then perform prediction on it.*

*After the prediction is complete this server will call the rest api server to save the model's response in the database.*

## 2. Integration:

Integration is a critical step within software implementation and it involves migrating data, compiling all modules in one system. With proper integrations, one can expect the project to be producing expected output.

*Activity: Here, team members are required to document their integration strategy, mention about versions of module code if any, dependencies if any.*

*In our project we have 4 major components which needs to be integrated:*



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

- *Client Facing Server*
- *Rest API Server*
- *Authentication Server*
- *Machine Learning Model Server.*

*Client Facing Server, Rest API Server and Authentication Server are created using FAST API.*

- *Browser – Client Facing Server integration:*

*The Client Facing Server predominantly serves static web pages on front end.*

*The Client asks simply for html, css and js files which the client facing server returns.*

*This is how the Browser integrates with the Client facing server.*

- *Browser – Rest API Server integration:*

*A lot of jQuery is written in the static js files served by the client facing server. This jQuery code is used to dynamically load data by sending requests to Rest API Server from Client's browser and then render the data accordingly.*

*This call to the rest api server is not visible to the client on the url and thus is unnoticeable to the client unless they open the js files loaded in the network resources tab.*

*This is how the data is loaded in the web page*

*This is how the browser and the Rest API Server are integrated.*

*Rest API Server check the authorisation header for valid token before returning the data.*

- *Client Facing Server – Authentication Server integration.*

*When the login page is loaded by the client facing server and the user enters the credentials, the credentials are sent to authentication server.*

*Authentication server validates the credentials and generates the access token.*

*This token is then returned by the Authentication server to the Client Facing Server.*

*Accordingly the Client Facing Server shows the result on the web page.*



**K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

- *Client Facing Server – Machine Learning Model Server - Rest API Server integration.*

*When the doctor uploads an image, the request is served by the /uploads route of client facing server. Client facing server (after proper verification of doctor's access token and access rights for patient) then sends the image to the machine learning model server to do the prediction.*

*This is an asynchronous integration. Thus the prediction work will be “on” in the machine learning server and will not affect the browser’s interaction with the client facing server.*

*Once the prediction is completed, the machine learning server will use an api given by rest API server to insert the status of the image (Model’s response) into the Database.*

*This asynchronous integration improves overall performance of the entire system and the doctor doesn’t have to wait for the model’s response while uploading the image.*

*The Model’s response will be visible to the doctor in short time but not immediately.*

**3. Dataset link (if applicable) with source/ Process of dataset selection with sample data along with data dictionary.**

**Links of the Dataset:**

<https://www.kaggle.com/datasets/meetnagadia/kvasir-dataset>

**Dataset used in the project :**

- 1) The dataset used for the ML model is Kvasir Dataset developed by Vestre Viken Health Trust (VV) in Norway.
- 2) The images in the dataset are divided into 6 classes with 500 images in each category
  - Esophagitis
  - Polyps
  - Ulcerative-colitis
  - Normal-z-line



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

- Normal pylorus
- Normal cecum

3) The resolution in the images ranges from 720x576 up to 1920x1072 pixels. In some images there is shown arrangement of endoscope inside intestine

4) Each of the classes have a CSV file containing useful information like AutoColorCorrelogram, PHOG, Edge Histogram. These metadata help in more accurate detection of tumour

#### **4. Implementation details**

##### **Metrics used for the Model:**

- **True negative (TN)** The number of correctly identified negative samples, i.e., frames without an endoscopic finding which correctly is identified as a frame without an endoscopic finding.
- **False positive (FP)** The number of wrongly identified samples, i.e., a commonly called a "false alarm". Frames without an endoscopic finding which is erroneously identified as a frame with an endoscopic finding.
- **False negative (FN)** The number of wrongly identified negative samples. Frames without an endoscopic finding which erroneously is identified as a frame with an endoscopic finding.
- **Recall (REC)** This metric is also frequently called sensitivity, probability of detection and true positive rate, and it is the ratio of samples that are correctly identified as positive among all existing positive samples.
- **Precision (PREC)** This metric is also frequently called the positive predictive value, and shows the ratio of samples that are correctly identified as positive among the returned samples (the fraction of retrieved samples that are relevant).
- **Specificity (SPEC)** This metric is frequently called the true negative rate, and shows the ratio of negatives that are correctly identified as such (e.g., the fraction of frames without an endoscopic finding are correctly identified as a negative result).
- **Accuracy (ACC)** The percentage of correctly identified true and false samples.
- **Matthews correlation coefficient (MCC)** MCC takes into account true and false positives and negatives, and is a balanced measure even if



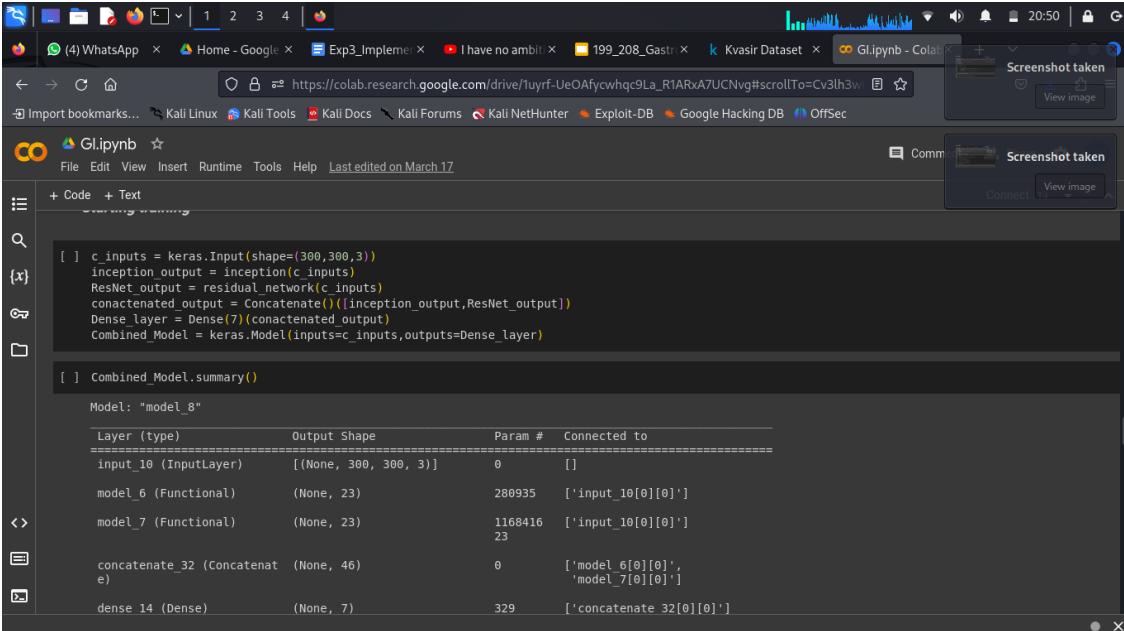
**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

the classes are of very different sizes.

- **F1 score (F1)** A measure of a test's accuracy by calculating the harmonic mean of the precision and recall.

## ML Model in depth insight.

### Implementation:



```
[ ] c_inputs = keras.Input(shape=(300,300,3))
inception_output = inception(c_inputs)
ResNet_output = residual_network(c_inputs)
concatenated_output = Concatenate()([inception_output,ResNet_output])
Dense_layer = Dense(7)(concatenated_output)
Combined_Model = keras.Model(inputs=c_inputs,outputs=Dense_layer)

[ ] Combined_Model.summary()

Model: "model_8"
-----
```

Layer (type)	Output Shape	Param #	Connected to
input_10 (InputLayer)	[None, 300, 300, 3]	0	[]
model_6 (Functional)	(None, 23)	280935	['input_10[0][0]']
model_7 (Functional)	(None, 23)	1168416	['input_10[0][0]', '23']
concatenate_32 (Concatenat e)	(None, 46)	0	['model_6[0][0]', 'model_7[0][0]']
dense_14 (Dense)	(None, 7)	329	['concatenate_32[0][0]']



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

The image shows two vertically stacked screenshots of Google Colab notebooks. Both notebooks have the title 'Gl.ipynb' and were last edited on March 17.

**Notebook 1 (Top): Assigning Training and Testing Dataset**

```
[ ] (TRAIN,TRAIN_LABEL) , (TEST,TEST_LABEL),labels = load_data()  
TRAIN = [os.path.join(cur_path,f"{i}") for index,i in enumerate(TRAIN)]  
TEST = [os.path.join(cur_path,f"{i}") for i in TEST]  
  
TRAIN_LABEL = TRAIN_LABEL.numpy().tolist()  
  
TEST_LABEL = TEST_LABEL.numpy().tolist()  
  
TRAIN,TRAIN_LABEL = shuffling(TRAIN,TRAIN_LABEL)  
  
TRAIN_DATASET = tf_dataset(TRAIN,TRAIN_LABEL)  
VALID_DATASET = tf_dataset(TEST,TEST_LABEL)  
  
print(TRAIN_DATASET)  
[1 1 1 ... 0 0 0]  
HI  
<_ PrefetchDataset element_spec=(TensorSpec(shape=<unknown>, dtype=tf.float32, name=None), TensorSpec(shape=<unknown>, dtype=tf.float32, name=None)>
```

**Notebook 2 (Bottom): Defining Metrics**

```
[ ] model_path = "Utils/CheckPoints"  
csv_path = "Utils/csv_path"  
metrics = [Precision(),Recall()]  
  
callbacks = [  
    ModelCheckpoint(model_path, verbose=1, save_best_only=True),  
    ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=5, min_lr=1e-7, verbose=1),  
    CSVLogger(csv_path),  
    TensorBoard(),  
    EarlyStopping(monitor='val_loss', patience=20, restore_best_weights=False)  
]  
lr = 1e-4
```

**Assigning Training and Testing Dataset**

```
[ ] (TRAIN,TRAIN_LABEL) , (TEST,TEST_LABEL),labels = load_data()  
TRAIN = [os.path.join(cur_path,f"{i}") for index,i in enumerate(TRAIN)]  
TEST = [os.path.join(cur_path,f"{i}") for i in TEST]
```



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

```
inputs_residual = Input(shape=(300, 300, 3))
X = Conv2D(kernel_size=(7, 7), strides=(2, 2), filters=64, padding='same')(inputs_residual)
X = BatchNormalization()(X)
X = Activation('relu')(X)
X = MaxPooling2D(pool_size=(3, 3), strides=(2, 2), padding='same')(X)

# Stack residual blocks
for _ in range(3):
    X = residual_block(X, filters=64)

X = residual_block(X, filters=128, strides=(2, 2))

for _ in range(2):
    X = residual_block(X, filters=128)

X = residual_block(X, filters=256, strides=(2, 2))

for _ in range(2):
    X = residual_block(X, filters=256)

X = AveragePooling2D(pool_size=(3, 3), strides=(2, 2), padding='same')(X)
X = Flatten()(X)
X = Dense(512, activation='relu')(X)
X = Dropout(0.2)(X)
outputs = Dense(23, activation='softmax')(X)
```

```
[ ] from keras.layers import Input, Conv2D, MaxPooling2D, BatchNormalization, Activation, Concatenate, AveragePooling2D, Flatten, Dense, Dropout
from keras.models import Model

def residual_block(X, filters, strides=(1, 1)):
    shortcut = X

    # First convolutional block
    X = Conv2D(filters=filters, kernel_size=(1, 1), strides=strides, padding='same')(X)
    X = BatchNormalization()(X)
    X = Activation('relu')(X)

    # Second convolutional block
    X = Conv2D(filters=filters, kernel_size=(3, 3), strides=(1, 1), padding='same')(X)
    X = BatchNormalization()(X)
    X = Activation('relu')(X)

    # Third convolutional block
    X = Conv2D(filters=filters * 4, kernel_size=(1, 1), strides=(1, 1), padding='same')(X)
    X = BatchNormalization()(X)

    # Shortcut connection
    if strides != (1, 1) or shortcut.shape[-1] != filters * 4:
        shortcut = Conv2D(filters=filters * 4, kernel_size=(1, 1), strides=strides, padding='same')(shortcut)
```



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

The screenshot shows two instances of a Google Colab notebook. Both instances have the same URL: [https://colab.research.google.com/drive/1uyrf-UeOAfyccwhqc9La\\_R1ARxA7UCNvg](https://colab.research.google.com/drive/1uyrf-UeOAfyccwhqc9La_R1ARxA7UCNvg).

**Top Notebook (Left):**

```
e)
batch_normalization_76 (BatchNormalization)      1024    ['concatenate_22[0][0]']

global_average_pooling2d_2 (GlobalAveragePooling2D) 0      ['batch_normalization_76[0][0]']

dropout_6 (Dropout)      (None, 256)      0      ['global_average_pooling2d_2[0][0]']

flatten_6 (Flatten)      (None, 256)      0      ['dropout_6[0][0]']

dense_10 (Dense)         (None, 524)      134668  ['flatten_6[0][0]']

dropout_7 (Dropout)      (None, 524)      0      ['dense_10[0][0]']

dense_11 (Dense)         (None, 23)       12075   ['dropout_7[0][0]']

flatten_7 (Flatten)      (None, 23)       0      ['dense_11[0][0]']

=====
Total params: 280935 (1.07 MB)
Trainable params: 280423 (1.07 MB)
Non-trainable params: 512 (2.00 KB)
```

**Bottom Notebook (Right):**

```
Inception.summary()

Model: "model_6"
+-----+
| Layer (type)        | Output Shape | Param # | Connected to |
+-----+
| input_8 (InputLayer)| [None, 300, 3] | 0        | []           |
| conv2d_86 (Conv2D)  | (None, 300, 64)| 256     | ['input_8[0][0]']|
| conv2d_87 (Conv2D)  | (None, 300, 96)| 6240    | ['conv2d_86[0][0]']|
| conv2d_89 (Conv2D)  | (None, 300, 16) | 1040    | ['conv2d_86[0][0]']|
| max_pooling2d_4 (MaxPoolin| (None, 300, 64) | 0        | ['conv2d_86[0][0]']|
| g2D)                |
| conv2d_88 (Conv2D)  | (None, 300, 128)| 110720  | ['conv2d_87[0][0]']|
| conv2d_90 (Conv2D)  | (None, 300, 32) | 12832   | ['conv2d_89[0][0]']|
| conv2d_91 (Conv2D)  | (None, 300, 32) | 2080    | ['max_pooling2d_4[0][0]']|
| concatenate_22 (Concatenat| (None, 300, 256) | 0        | ['conv2d_86[0][0]',|
| e)                  |                 |          | 'conv2d_88[0][0]',|
|                      |                 |          | 'conv2d_90[0][0]',|
|                      |                 |          | 'conv2d_91[0][0]']|
```



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Screenshot taken

View image

File Edit View Insert Runtime Tools Help Last edited on March 17

+ Code + Text

```
inputs = keras.Input(shape=(300,300,3))
x = layers.Conv2D(filters=64,strides=(1,1),kernel_size=(1,1),padding='same')(inputs)
y = layers.Conv2D(filters=96,strides=(1,1),kernel_size=(1,1),padding='same')(x)
y_1 = layers.Conv2D(filters=128,strides=(1,1),kernel_size=(3,3),padding='same')(y)
z = layers.Conv2D(filters=16,strides=(1,1),kernel_size=(1,1),padding='same')(x)
z_1 = layers.Conv2D(filters=32,strides=(1,1),kernel_size=(5,5),padding='same')(z)
w = layers.MaxPool2D(strides=(1,1),pool_size=(3,3),padding='same')(x)
w_1 = layers.Conv2D(filters=32,strides=(1,1),kernel_size=(1,1),padding='same')(w)

filter_concat = Concatenate()([x,y_1,z_1,w_1])

bn = BatchNormalization()(filter_concat)
global_avg_pool = GlobalAveragePooling2D()(bn)
dropout = Dropout(0.2)(global_avg_pool)
flatten = Flatten()(dropout)
dense1 = Dense(524)(flatten)
dropout1 = Dropout(0.2)(dense1)
dense3 = Dense(23)(dropout1)
features = Flatten()(dense3)

inception = keras.Model(inputs = inputs, outputs = features)
```

Screenshot taken

View image

File Edit View Insert Runtime Tools Help Last edited on March 17

+ Code + Text

```
return dataset
```

Resnet.png#### Makina\_Inception\_V3



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Screenshot taken

Gl.ipynb

File Edit View Insert Runtime Tools Help Last edited on March 17

+ Code + Text

```
        return image,label
    image,label = numpy_function(_parse,[image,label],[np.float32,np.float32])
    ensure_shape(image,(H,W))
    return image,label
    return image, label

def tf_dataset(X,Y,batch_size=4):
    dataset = data.Dataset.from_tensor_slices((X,Y))
    dataset = dataset.map(tf_parse)
    dataset = dataset.batch(batch_size=batch_size)
    dataset = dataset.prefetch(10)
    return dataset
```

Screenshot taken

Gl.ipynb

File Edit View Insert Runtime Tools Help Last edited on March 17

+ Code + Text

```
print("HI")
label_mapping = {index: label for index, label in enumerate(label_encoder.classes_)}

return (train,train_labels), (test,test_labels), label_mapping

def shuffling(X,Y):
    X,Y = shuffle(X,Y,random_state=42)
    return X,Y

def read_image(PATH):
    PATH = PATH.decode()
    x = cv2.imread(PATH, cv2.IMREAD_COLOR)
    x = cv2.resize(x,(H,W))
    x = x/255.0
    x = x.astype(np.float32)
    return x

def tf_parse(image,label):
    def _parse(image,label):
        image = read_imagenet(image)
```



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

The image shows two side-by-side screenshots of Google Colab notebooks. Both notebooks have the title 'Gl.ipynb' and were last edited on March 17.

**Notebook 1 (Top):** This notebook contains Python code for image processing. It includes imports for os, os.path, LabelEncoder, and train\_test\_split. The code reads images from a directory, encodes labels, splits the data into training and testing sets, and prints the total number of classes. It also defines a one-hot encoding function for depth.

```
path = os.path.join(cur_path)
label_encoder = LabelEncoder()

images = []
labels = []
for i in os.listdir(path):
    if i != "utils" and i != ".ipynb_checkpoints":
        for j in os.listdir(os.path.join(path,i,i)):
            images.append(os.path.join(i,i,j))
            labels.append(i)

labels = label_encoder.fit_transform(labels)
print(labels)
depth = 7

train,test = train_test_split(images,test_size = split,random_state=42)
train_labels, test_labels = train_test_split(labels,test_size = split,random_state=42)

train_labels = one_hot(train_labels,depth)
test_labels = one_hot(test_labels,depth)

print("HI")

label_mapping = {index: label for index, label in enumerate(label_encoder.classes_)}
```

**Notebook 2 (Bottom):** This notebook contains Python code for disease classification. It reads images from a directory named 'Kvasir'. The code counts the total number of classes, the number of diseases, and the number of images per class. It uses os.path, os.listdir, and os.scandir.

```
cur_path = os.path.join("/content/drive/MyDrive","Kvasir")
total_classes = 0
all_diseases = []
total_images_each_class = []
for i in os.listdir(cur_path):
    if(i!=".ipynb_checkpoints" and i!="Utils"):
        new_path = os.path.join(cur_path,i,i)
        number = 0
        total_classes += 1
        all_diseases += [i]

        for j in os.scandir(new_path):
            number+=1

        total_images_each_class+=[number]

print("Total number of classes: ",total_classes)
print("All diseases classes are: ",all_diseases)
print("Number of images in each class are: ",total_images_each_class)
```



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

The image shows two side-by-side screenshots of Google Colab notebooks.

**Top Notebook:**

- Header: "Gl.ipynb" - Colab
- Section: "Basic imports"
- Code:

```
import os
import cv2
import pandas as pd
from tensorflow import keras, data, numpy, function, ensure_shape, convert_to_tensor, zeros, GradientTape, float32, cast, one_hot
from keras import Sequential, layers, Model
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from keras.utils import to_categorical
from keras.callbacks import ReduceLROnPlateau, ModelCheckpoint, TensorBoard, CSVLogger, EarlyStopping
import numpy as np
from keras.models import Model
from sklearn.preprocessing import LabelEncoder
from keras.metrics import Recall, Precision
from keras.optimizers import Adam
from keras.layers import BatchNormalization, Conv2D, UpSampling2D, Activation, Concatenate, Input, MaxPooling2D, Concatenate, GlobalAveragePooling2D, Dropout
from keras.models import Model
from keras.losses import CategoricalCrossentropy
```

**Bottom Notebook:**

- Header: "Gl.ipynb" - Colab
- Section: "Download dataset from kaggle"
- Code:

```
!pip install kaggle
!pip install opencv-python
```

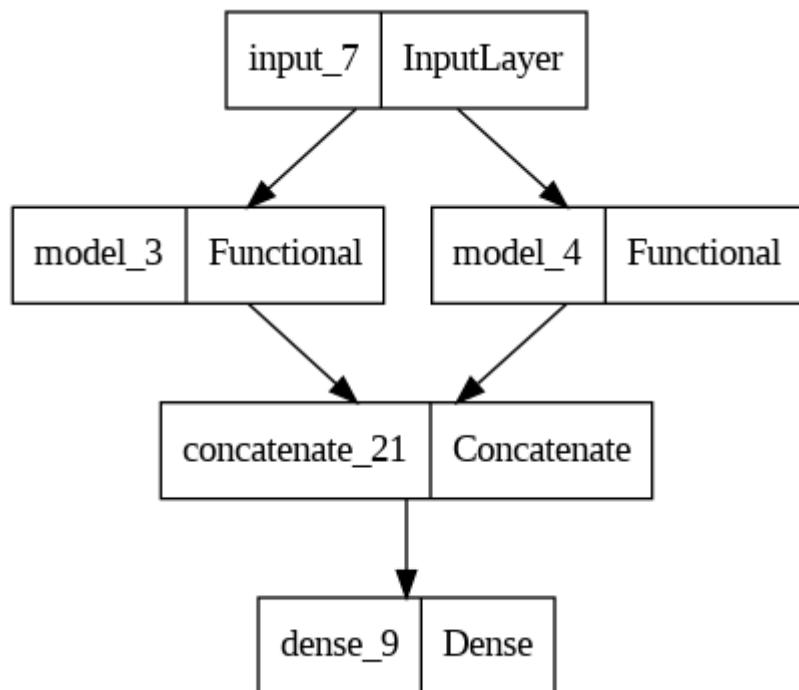
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.5.16)
Requirement already satisfied: six<=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.2.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tzdata in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.2)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.6)
Requirement already satisfied: opencv-python in /usr/local/lib/python3.10/dist-packages (4.8.0.76)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.10/dist-packages (from opencv-python) (1.25.2)

Double-click (or enter) to edit

- Section: "Basic imports"

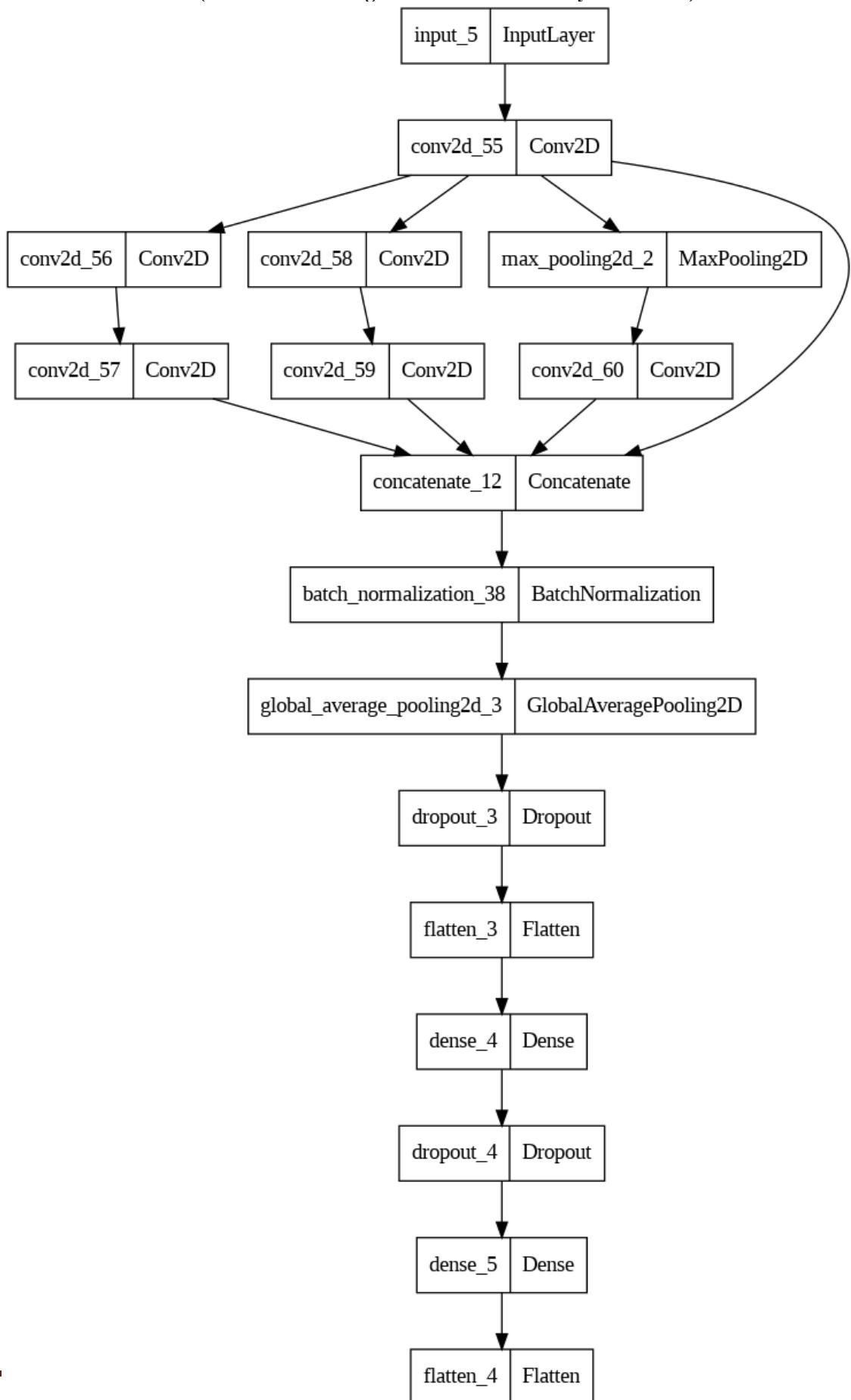


**K. J. Somaiya College of Engineering, Mumbai-77**  
(Autonomous College Affiliated to University of Mumbai)





**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**





**K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

### **How does the model work?**

- 1) For classification of image we use a combined model. We use the power of ResNet along with a block inception and concatenate the feature vectors thus obtained.
- 2) Why choose ResNet ? The reason is that ResNet has skip links that help in alleviating the vanishing gradient problem. This enables us to have deeper neural networks that can capture finer details
- 3) Why choose Inception block ?The Inception block helps in capturing details at various levels. Inception has filters of 1x1, 3x3, 5x5 which helps in capturing fine grained as well as details on a more global level. The inception block also helps in reducing the computations needed
- 4) The feature vectors obtained from both the models are then passed to a SVM classifier. Why use SVM ? SVM has the capacity to take the feature vector obtained from ResNet and Inception to higher dimensions where the vectors can then be distinguished from each other. SVM works well as compared to KNN and Softmax

**5. Implementation Screenshots: Each team will present implemented module screenshots in accordance of process flow with small description**



**K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

Healing Horizons Hospital Intranet

Email ID:

Password:

Login

127.0.0.1:8001 says

Invalid Credentials

OK

Healing Horizons Hospital Intranet

Email ID:

Password:

.....

Login



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

127.0.0.1:8001 says  
Valid Credentials

OK

Healing Horizons Hospital Intranet

Email ID:  
monika.mehta@healinghorizons.com

Password:  
\*\*\*\*\*

Login

Logout

### Welcome Admin

[View Existing Patients](#)

[View Patient by Id](#)

[Add a New Patient](#)

Logout

### Welcome Admin

The details of each patient is :

**Ashish Singh**

Patient Id: Ashish-Singh-114417841941

Blood Group: A+  
Doctor Assigned: rohit.singh@healinghorizons.com  
Birth Date: 2003-11-17  
Last Visit Date: 2023-09-12  
Mobile: 1234567890  
Email: email1@gmail.com

**Medical History**

Allergies

- allergy1
- allergy2
- allergy3

Surgeries



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

**Surgeries**

- endoscopy on 2013-09-12
- Cataract on 2016-04-19
- Heart bypass on 2019-07-13

**Medications**

- Crocin: Frequent headache, Once Every night, Temporary
- Calciferol: Vitamin D deficiency, Twice Every day, Lifetime
- Bonfeit K2-7: Calcium deficiency, Thrice Every day, Lifetime

**Operation Images**

Link: [View Image](#)  
Doctor's Comment: kjrthdfsgdfs  
Model Response: None

Link: [View Image](#)  
Doctor's Comment: dfsgdsfhsrf  
Model Response: None

Link: [View Image](#)  
Doctor's Comment: new comment  
Model Response: None

Link: [View Image](#)  
Doctor's Comment: New comment goes in here

Link: [View Image](#)  
Doctor's Comment: New comment goes in here  
Model Response: None

**Riya Shah**  
**Patient Id:** Riya-Shah-115107067255  
**Blood Group:** A+  
**Doctor Assigned:** radhika.gupta@healinghorizons.com  
**Birth Date:** 2006-11-19  
**Last Visit Date:** 2023-09-12  
**Mobile:** 1234567890  
**Email:** email1@gmail.com

**Medical History**

**Allergies**

- allergy1
- allergy2
- allergy3



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Search Patient

127.0.0.1:8001/static/admin\_view\_patient\_by\_id.html

Healing Horizons Hospital Logout

### Welcome Admin

Search for a Patient by ID:

Ashish-Singh-114417841941

Search

**Ashish Singh**

Patient Id: Ashish-Singh-114417841941

Blood Group: A+

Doctor Assigned: rohit.singh@healinghorizons.com

Birth Date: 2003-11-17

Last Visit Date: 2023-09-12

Mobile: 1234567890

Email: email1@gmail.com

**Medical History**

Allergies

- allergy1
- allergy2
- allergy3

127.0.0.1:8001/static/admin\_view\_patient\_by\_id.html

Healing Horizons Hospital Logout

### Welcome Admin

Search for a Patient by ID:

Ashish-Singh-1141

Search

Invalid Patient Id

Chrome File Edit View History Bookmarks Profiles Tab Window Help Aditya Ved 94% Sun 7 Apr 4:36:28PM

Add Patient

127.0.0.1:8001/static/admin\_add\_patients.html

Healing Horizons Hospital Logout

### Welcome Admin

**Patient Creation Process**

Step 1: In order to create a new patient, fill in the below details first:

No of Allergies: 2

No of Surgeries: 1

No of Medications: 1

Next



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

Healing Horizons Hospital

Logout

Welcome Admin

Patient Creation Process

Step 2: Now Fill the required details of the patient  
below :

First Name :

Last Name :

Birth Date :

 dd/mm/yyyy

Address :

Last Visit Date :

 dd/mm/yyyy

← → ⌛ 127.0.0.1:8001/static/admin\_add\_patients.html

Mobile Number :

Email Id :

Blood Group

Gender :

Doctor Assigned:

Allergies:

Allergy 1 :

Allergy Name :

Allergy 2 :

Allergy Name :

Surgeries:

Surgery 1



**K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

Add Patient

127.0.0.1:8001/static/admin\_add\_patients.html

Surgeries:

Surgery 1

Surgery Name :

Surgery Date :  dd/mm/yyyy

Medications

Medication 1

Medicine Name :

Medicine Reason :

Medicine Is Lifetime ? :

Medicine Frequency:

Submit

Apple Chrome File Edit View History Bookmarks Profiles Tab Window Help Aditya Ved 95% Sun 7 Apr 4:31:10PM

Healing Horizons Hospital Intranet

127.0.0.1:8001 says  
Valid Credentials

OK

Healing Horizons Hospital Intranet

Email ID:  rohit.singh@healinghorizons.com

Password:  .....  
Login



**K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

Healing Horizons Hospital

Welcome Doctor

View all Patients under your supervision

Add Image of a patient operations/surgeries

Create a new chat thread

Continue to Chat with other doctors

Logout

Healing Horizons Hospital

127.0.0.1:8001 says

Logged Out Successfully !!!

OK

Welcome Doctor

View all Patients under your supervision

Add Image of a patient operations/surgeries

Create a new chat thread

Continue to Chat with other doctors



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Healing Horizons Hospital

Logout

### Welcome Doctor

The patients under your supervision are :

#### Ashish Singh

Patient Id: Ashish-Singh-114417841941

Blood Group: A+

Doctor Assigned: rohit.singh@healinghorizons.com

Birth Date: 2003-11-17

Last Visit Date: 2023-09-12

Mobile: 1234567890

Email: email1@gmail.com

#### Medical History

##### Allergies

- allergy1
- allergy2
- allergy3

##### Surgeries

- endoscopy on 2013-09-12
- Cataract on 2016-04-19
- Heart bypass on 2019-07-13

← → ⌂ 127.0.0.1:8001/static/doctor\_view\_all\_patients\_under\_supervision.html

#### Rohit Malhotra

Patient Id: Rohit-Malhotra-153003441493

Blood Group: A+

Doctor Assigned: rohit.singh@healinghorizons.com

Birth Date: 2003-11-17

Last Visit Date: 2023-09-12

Mobile: 1234567890

Email: email1@gmail.com

#### Medical History

##### Allergies

- allergy1
- allergy2
- allergy3

##### Surgeries

- endoscopy on 2013-09-12
- Cataract on 2016-04-19
- Heart bypass on 2019-07-13

##### Medications

#### Operation Images

Healing Horizons Hospital

Logout

### Welcome Doctor

Upload the image of patient Operation/Surgery

Patient Id:

Ashish-Singh-114417841941

Upload image of Patient:

[Choose file : operation\_img]

Doctor's Comment/Review for the Image:

XYZ Review

Upload File



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Upload Patient Image

127.0.0.1:8001

Healing Horizons Hospital

Logout

127.0.0.1:8001 says

Image Uploaded successfully !!!!

File name: Image-164345384253524496

Patient Id: Ashish-Singh-114417841941

Image Link: http://127.0.0.1:8001/static/med\_images/  
Image-164345384253524496

OK

Ashish-Singh-114417841941

Upload image of Patient:

Choose file : operation\_img

Doctor's Comment/Review for the Image:

XYZ Review

Upload File

127.0.0.1:8001

Healing Horizons Hospital

Logout

Welcome Doctor

Upload the image of patient Operation/Surgery

Patient Id:

Ashish-Singh-114417841941

Upload image of Patient:

Choose file : operation\_img

Doctor's Comment/Review for the Image:

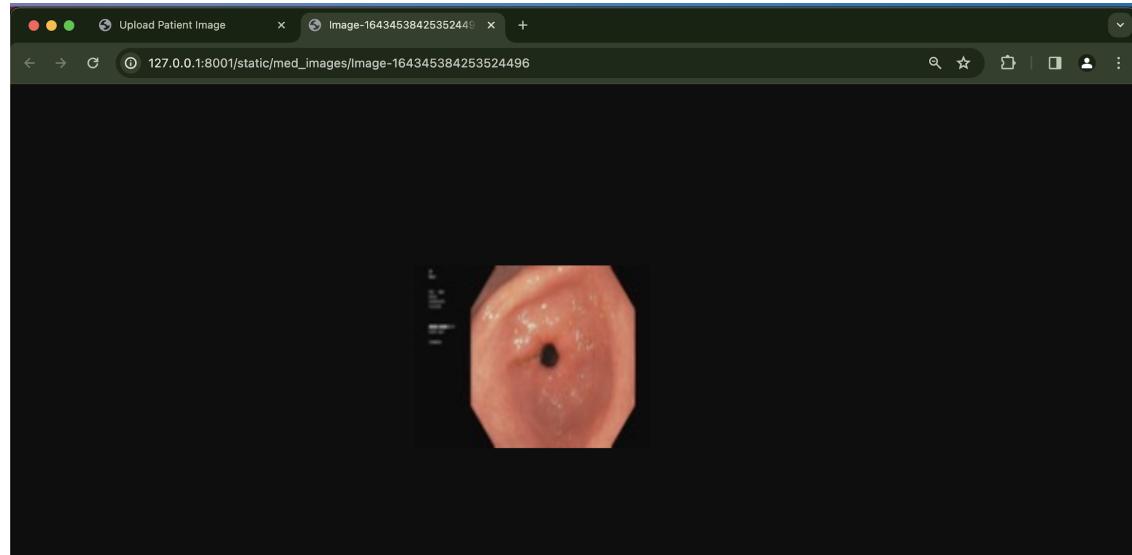
XYZ Review

Upload File

Image Uploaded Successfully :

Image Link Generated: http://127.0.0.1:8001/static/med\_images/Image-164345384253524496

[Click here to view the image](#)





## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

The screenshot shows the 'Chat Dashboard' page of a web application. At the top, there are four cards representing different chat sessions:

- new chat life**: Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com. A 'View Chat' button is present.
- Regarding Case 1**: Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com. A 'View Chat' button is present.
- Regarding Case 2**: Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com. A 'View Chat' button is present.
- Regarding Case 3**: Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com. A 'View Chat' button is present.

The screenshot shows the 'Regarding Case 1' chat room. The message history is as follows:

- Hi I am talking about case 1  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 10:32:39
- Hey are you listening ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:31:57
- Hey there 3rd time ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:32:26
- Hey there 4th time ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:33:04
- Hey there 5th time ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:34:29
- Hey there 6th time ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:35:51
- Hey there 7th time ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:36:04
- Hey there 8th time ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:37:41
- Finally Hey there !!!  
From: radhika.gupta@healinghorizons.com | Date & Time: 2024-03-30 13:37:55
- So I was saying  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:50:59

A text input field at the bottom says 'Type your message here...' and a 'Send' button is visible.



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Healing Horizons Hospital

Logout

127.0.0.1:8001 says

Message sent successfully !!!!  
Kindly Refresh the page to see the results and check for new messages !!!

OK

From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 15:19:38  
hey there !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 19:35:20  
hello there !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 22:59:58  
yeah I am free now please say  
From: radhika.gupta@healinghorizons.com | Date & Time: 2024-03-31 23:02:16  
Ok there !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 23:15:28  
Hey  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 23:20:40  
Demo new message  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 23:21:56  
Yeah Demo acknowledged !!!  
From: radhika.gupta@healinghorizons.com | Date & Time: 2024-03-31 23:22:58  
Hey there !

Send

Healing Horizons Hospital

Logout

You have any suggestions ?  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 13:54:28  
Tell fast now !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-30 15:18:51  
hey there !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 19:35:20  
hello there !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 22:59:58  
yeah I am free now please say  
From: radhika.gupta@healinghorizons.com | Date & Time: 2024-03-31 23:02:16  
Ok there !  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 23:15:28  
Hey  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 23:20:40  
Demo new message  
From: rohit.singh@healinghorizons.com | Date & Time: 2024-03-31 23:21:56  
Yeah Demo acknowledged !!!  
From: radhika.gupta@healinghorizons.com | Date & Time: 2024-03-31 23:22:58  
Hey there !

Type your message here... Send



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

The screenshot shows a web application window titled "Create Chat Thread". At the top, there's a header bar with "Healing Horizons Hospital" on the left and "Logout" on the right. A central modal window has a dark background with white text. It says "127.0.0.1:8001 says" at the top, followed by "Invalid Doctor Email Entered" in a smaller font. In the bottom right corner of the modal is a green "OK" button. Below the modal, the main page contains a form titled "Create a Chat Thread". It has two input fields: "Add email Id of the Doctor to communicate:" containing "ras.gupta@healinghorizons.com" and "Title of the chat thread:" containing "Regarding Case 4". At the bottom right of the form is a "Create Thread" button.

This screenshot is similar to the one above, showing the "Create Chat Thread" interface. The modal now displays "127.0.0.1:8001 says" followed by "Chat Thread Created Successfully !!!". The "OK" button is visible. The underlying form fields and "Create Thread" button are identical to the previous screenshot.



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Healing Horizons Hospital

Welcome Doctor

Create a Chat Thread

Add email Id of the Doctor to communicate:  
radhika.gupta@healinghorizons.com

Title of the chat thread:  
Regarding Case 4

Create Thread

Details of newly created chat thread:

Chat Title: Regarding Case 4

Participant 1: rohit.singh@healinghorizons.com

Participant 2: radhika.gupta@healinghorizons.com

Chat Thread Id: rohit.singh@healinghorizons.com-radhika.gupta@healinghorizons.com-2024-04-07-16-46-23

Healing Horizons Hospital

Logout

Chat Dashboard

**new chat life**  
Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com  
[View Chat](#)

**Regarding Case 1**  
Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com  
[View Chat](#)

**Regarding Case 2**  
Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com  
[View Chat](#)

**Regarding Case 3**  
Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com  
[View Chat](#)

**Regarding Case 4**  
Participants: rohit.singh@healinghorizons.com, radhika.gupta@healinghorizons.com  
[View Chat](#)

6. **Git link (Maintain Coding standards)**
  - **Link**
  - **Screenshots of contribution over repository**



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

<https://github.com/orgs/GastroIntestinal-Disease-Detect-WebApp/repositories>

The screenshot shows the GitHub organization page for "GastroIntestinal-Disease-Detect-WebApp". The left sidebar has a "Repositories" section with filters for All, Public, Private, Sources, Forks, Archived, Mirrors, and Templates. The main area displays 5 repositories under the "All" tab:

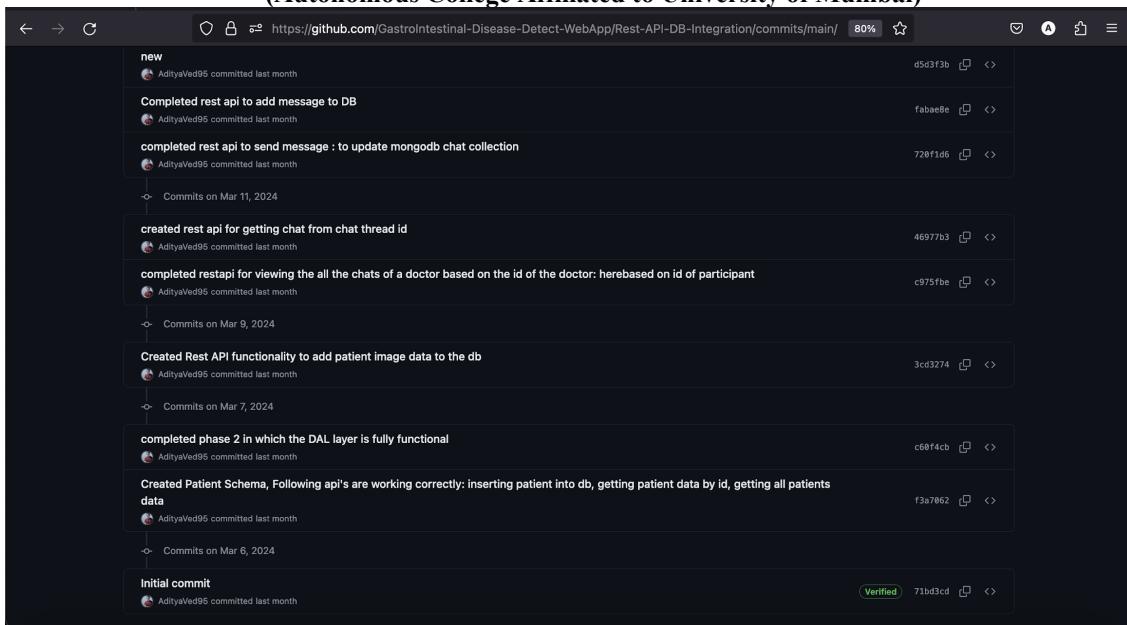
- Rest-API-DB-Integration** (Private)  
Creating rest apis to serve to client facing servers  
Python 0 stars 0 forks 0 updated last week
- Client-Facing-Server-New** (Private)  
All the requests that client makes comes to this server  
JavaScript 0 stars 0 forks 0 updated last week
- Client-Facing-Server** (Private)  
All the requests that client makes comes to this server  
JavaScript 0 stars 0 forks 0 updated last week
- AuthenticationSystem-Microservice** (Private)  
Python 0 stars 0 forks 0 updated last week
- Inception\_ResNetModel** (Private)  
0 stars 0 forks 0 updated last month

The screenshot shows the GitHub repository page for "Rest-API-DB-Integration" within the "GastroIntestinal-Disease-Detect-WebApp" organization. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The main area is titled "Commits" and shows a list of commits by user "AdityaVed95":

- Update README.md (Apr 1, 2024) - Verified commit f35addb
- Update README.md (Apr 1, 2024) - Verified commit 01b01e3
- complete all the rest api's for the web app to work correctly. Also added documentation of each route (Apr 1, 2024) - Verified commit 09f239e
- completed all web app api's with auth (Mar 30, 2024) - Verified commit bcc280c
- Commits on Mar 16, 2024:
  - completed chat application rest api (Mar 16, 2024) - Verified commit ed180d5
- Commits on Mar 13, 2024:
  - new (Mar 13, 2024) - Verified commit d5d3f3b

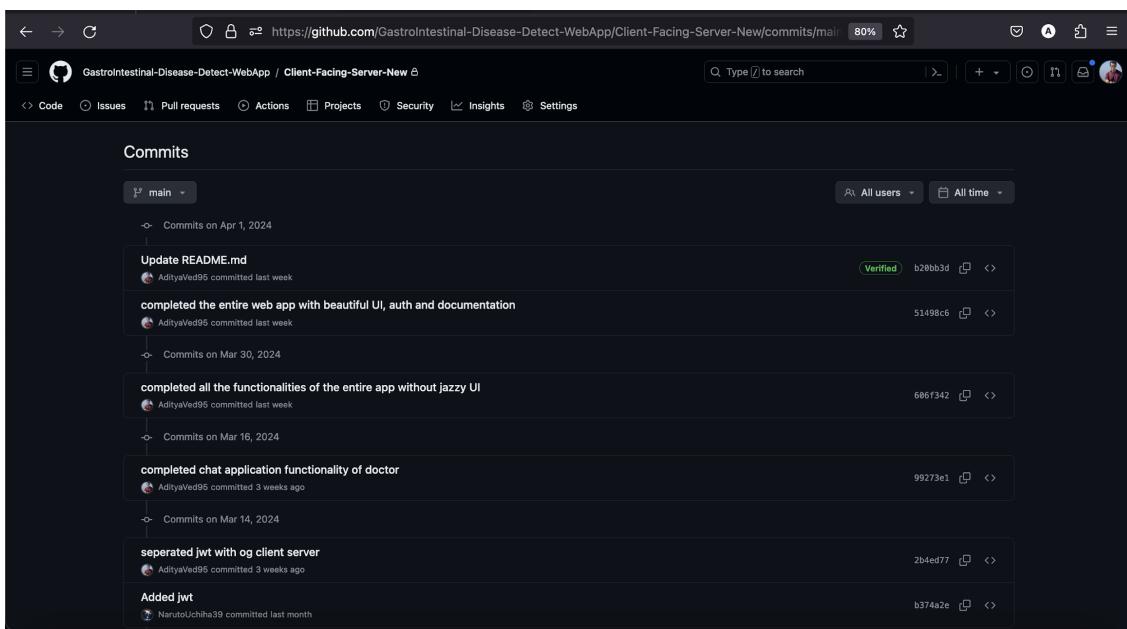


## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)



This screenshot shows the GitHub commit history for the 'Rest API - DB Integration' repository. The commits are listed in chronological order from top to bottom:

- new**  
AdityaVed95 committed last month
- Completed rest api to add message to DB**  
AdityaVed95 committed last month
- completed rest api to send message : to update mongodb chat collection**  
AdityaVed95 committed last month
- Commits on Mar 11, 2024**
- created rest api for getting chat from chat thread id**  
AdityaVed95 committed last month
- completed restapi for viewing the all the chats of a doctor based on the id of the doctor: herebased on id of participant**  
AdityaVed95 committed last month
- Commits on Mar 9, 2024**
- Created Rest API functionality to add patient image data to the db**  
AdityaVed95 committed last month
- Commits on Mar 7, 2024**
- completed phase 2 in which the DAL layer is fully functional**  
AdityaVed95 committed last month
- Created Patient Schema, Following api's are working correctly: inserting patient into db, getting patient data by id, getting all patients data**  
AdityaVed95 committed last month
- Commits on Mar 6, 2024**
- Initial commit**  
AdityaVed95 committed last month



This screenshot shows the GitHub commit history for the 'Client-Facing-Server-New' repository. The commits are listed in chronological order from top to bottom:

- main**  
Commits on Apr 1, 2024
- Update README.md**  
AdityaVed95 committed last week
- completed the entire web app with beautiful UI, auth and documentation**  
AdityaVed95 committed last week
- Commits on Mar 30, 2024**
- completed all the functionalities of the entire app without jazzy UI**  
AdityaVed95 committed last week
- Commits on Mar 16, 2024**
- completed chat application functionality of doctor**  
AdityaVed95 committed 3 weeks ago
- Commits on Mar 14, 2024**
- seperated jwt with og client server**  
AdityaVed95 committed 3 weeks ago
- Added jwt**  
NarutoUchiha39 committed last month



## K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

The screenshot shows a list of commits from the repository `GastroIntestinal-Disease-Detect-WebApp/Client-Facing-Server-New`. The commits are as follows:

- completed the entire web app with beautiful UI, auth and documentation (commit `51498c6`)
- completed all the functionalities of the entire app without jazzy UI (commit `686f342`)
- completed chat application functionality of doctor (commit `99273e1`)
- seperated jwt with og client server (commit `2b4ed77`)
- Added jwt (commit `b374a2e`)
- hello test (commit `aazfc8f`)
- new2 (commit `099b168`)

Each commit includes the author's name (AdityaVed95), the date it was committed, and a detailed description of the changes made.

The screenshot shows a list of commits from the repository `GastroIntestinal-Disease-Detect-WebApp/AuthenticationSystem-Microservice`. The commits are as follows:

- Commits on April 1, 2024
- Update README.md (commit `a5adbea`, Verified)
- Commits on March 30, 2024
- completed standalone token issuing authentication microservice (commit `f946b43`)
- Commits on March 25, 2024
- completed logout functionality of api (commit `9377c47`)
- Commits on March 24, 2024
- completed authentication using jwt tokens, login functionality and securing routes (only api part) (commit `7cdf10e`)
- Commits on March 16, 2024
- Initial commit (commit `7df6226`, Verified)

Each commit includes the author's name (AdityaVed95), the date it was committed, and a detailed description of the changes made.

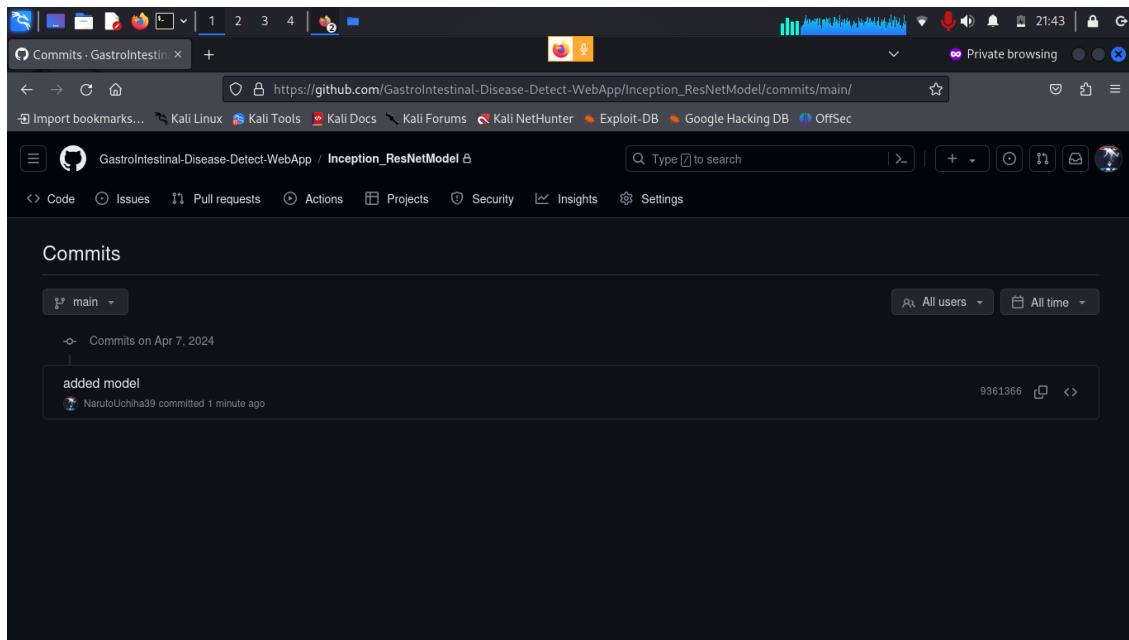
Aditya Ved : Worked on the Front End and Back End of the Web Application.

Aniruddha Ta: Worked on the Deep Learning Model to perform Gastrointestinal disease detection



**K. J. Somaiya College of Engineering, Mumbai-77  
(Autonomous College Affiliated to University of Mumbai)**

## Github Link:



**Only 1 commit as we used colab and did training there without extensive use of github**

## Conclusion:

**So via this project we learnt how different concepts of computer science integrate to make one big application.**

**We implemented a microservices architecture. We understood how each microservice has their own database and how they interact with each other to make the entire application work.**

**We implemented the concept of Authentication system as a separate microservice which helps in segregating responsibilities even more correctly.**

**We used the concept of asymmetric key cryptography to sign the token with private key which only authentication microservice has and then ensured that other components of the system have the public key to verify validity of token.**



**K. J. Somaiya College of Engineering, Mumbai-77**  
**(Autonomous College Affiliated to University of Mumbai)**

**We implement client side rendering through which we learned how data can be dynamically loaded in a web page by keeping the url the same or without redirecting to another web page.**

**As we see envision how this system can go live in the real world scenario, we see a lot of issues of data pipeline, MLops pipelines etc. Our project is the 1<sup>st</sup> step which has taken to a level to understand the need of such advanced concepts.**

**We also learnt about computer vision and deep learning as we explored the domain of predicting gastrointestinal diseases.**

**Thus multiple concepts were implemented in this project and it was a ecstatic experience to see them integrate into one big application. The ideas explored in the project opens doors to the world of possibilities and features that we have and are yet to explore.**