A Mini Project Report on

Swaasthya

CSE-3008: Introduction to Machine Learning

By

GROUP-6

20BCE7067	MADHYAM PATRA
20BCE7062	NIRDESH SINGH
20BCE7066	GNANENDRA AJITHNAG GADDE

School of Computer Science & Engineering

VIT-AP UNIVERSITY

MAY, 2022

INDEX

INTRODUCTION

Every 30 minutes, your kidneys, each the size of a computer mouse, filter all your blood. They put forth a lot of effort to get rid of waste, poisons, and extra fluid. They also aid in the management of blood pressure, the stimulation of red blood cell synthesis, the maintenance of healthy bones, and the regulation of vital blood molecules.

Healthy kidneys are essential for optimal health, but chronic renal disease affects more than one out of every seven persons in the United States (CKD).

Kidney disease (CKD) is a disorder in which the kidneys have been damaged and are unable to filter blood as effectively as they should. As a result, extra fluid and waste from the circulation linger in the body, potentially leading to various health issues like heart disease and stroke.

Other CKD-related health issues include:

- Anemia refers to a lack of red blood cells.
- Infections are becoming more common.
- Blood calcium levels are low, potassium levels are high, and phosphorus levels are high.
- Appetite loss or eating less.
- Depression implies a poorer standard of living.

CKD can be mild, moderate, or severe. Although therapy has been demonstrated to delay development, it normally grows worse with time. CKD can lead to renal failure and early cardiovascular disease if left untreated. Dialysis or a kidney transplant are required when the kidneys quit functioning. End-stage renal disease is kidney failure managed with dialysis or a kidney transplant (ESRD). Find out more about ESRD.

Renal disease does not always result in kidney failure. Control risk factors for CKD, get tested yearly, make lifestyle changes, take medicine as required, and contact your health care team periodically to help prevent CKD and minimize the chance of renal failure.

People with CKD may not notice any signs or feel poorly. Specific blood and urine tests are the only method to know for sure if you have CKD. Both the creatinine level in the blood and the protein level in the urine are measured in these assays.

- In 2019, total Medicare spending for patients with CKD totaled $87.2 billion, or $24,453 per Medicare member over the age of 65.
- In 2019, total Medicare fee-for-service spending for patients with ESRD or

renal failure totaled $37.3 billion, or $86,400 per person, accounting for around 7% of all Medicare paid claims expenses.

Creatinine is a waste product that comes from the normal wear and tear on muscles of the body. Everyone has creatinine in their bloodstream.

The typical range for serum creatinine is:

- For adult men, 0.74 to 1.35 mg/dL (65.4 to 119.3 micromoles/L)
- For adult women, 0.59 to 1.04 mg/dL (52.2 to 91.9 micromoles/L)

If it is out of range, the patient should undergo dialysis.

Everyone has experienced the difficulty of receiving varying creatinine results from different hospitals. So, what if we get a quick CKD test at our fingertips?

OVERVIEW OF THE APP

Swaasthya App does the following for the user:-

- Predicts if user has CKD.
- Keeps the records of user's medical history.
- Refers user the best doctors around him.
- Provides user the location of nearest medical stores.
- Emergency contacts.

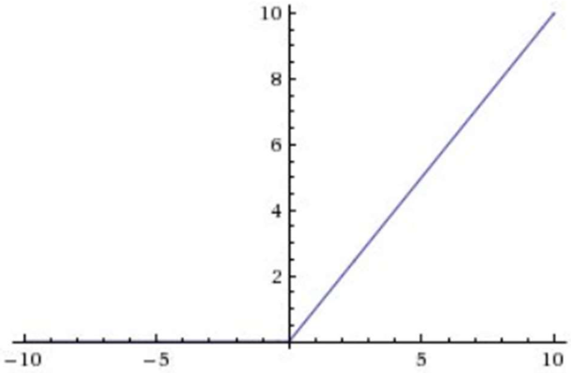
STUDY ON RELU ACTIVATION FUNCTION

The activation function in a neural network is responsible for converting the node's summed weighted input into the node's activation or output for that input.

The rectified linear activation function, or ReLU for short, is a piecewise linear function that, if the input is positive, outputs the input directly; else, it outputs zero. Because a model that utilizes it is quicker to train and generally produces higher performance, it has become the default activation function for many types of neural networks.

In deep learning models, the Rectified Linear Unit is the most widely employed activation function. If the function receives any negative input, it returns 0, but if it receives any positive input, it returns that value. As a result, f(x)=max(0,x) may be written.

Graphically it looks like this



It's amazing how well such a simple function (made up of two linear parts) can accommodate for non-linearities and interactions in your model. However, because the ReLU function performs well in most situations, it is commonly employed.

The fundamental objective of activation functions is to: 1) Assist in the accounting of interaction effects in a model.

What is the definition of an interactive effect? It occurs when one variable A has a varying impact on a prediction based on the value of B. For example, if my model needed to determine if a specific body weight indicated a higher risk of diabetes, it would need to know the person's height. For short persons, certain bodyweights suggest increased hazards, whereas for tall ones, they signify good health. As a result, the impact of body weight on diabetes risk is dependent on height, and we may say that weight and height interact.

2) Assist a model with non-linear effects. This simply implies that it isn't a straight line if I graph a variable on the horizontal axis and my predictions on the vertical axis. Alternatively, the effect of raising the predictor by one varies depending on the value of that predictor.

Imagine a single node in a neural network model with interactions. Assume it has two inputs, A and B, for the sake of simplicity. The weights into our node from A and B are 2 and 3, respectively. As a result, the node output is f(2A+3B). For our f, we'll utilise the ReLU function. So, if 2A+3B is positive, our node's output value is also 2A+3B. If 2A+3B is negative, our node's output value is 0.

Consider the following scenario: A=1 and B=1. The output is 2A+3B, and when A grows, the output increases as well. If B=-100, on the other hand, the output is 0, and if A is considerably increased, the output remains 0. As a result, A may or may not improve our production. It just relies on the value of B.

This is a straightforward example of a node capturing an interaction. The potential complexity of interconnections only grows as you add more nodes and layers. However, you should now be able to observe how the activation function assisted in the capturing of an interaction.

Non-linearities occur when the slope of a function is not constant. The slope of the ReLU function is always either 0 (for negative values) or 1 (for positive values) (for positive values). This is a particularly specific form of non-linearity.

However, two features regarding deep learning models allow us to build a variety of non-linearities by combining ReLU nodes in different ways.

For starters, almost all models have a bias term for each node. The bias term is just a fixed integer established during model training. Consider a node with a single input named A and a bias for simplicity.

When the bias term is set to 7, the node output is f(7+A). If A is less than -7, the output is 0 and the slope is 0 in this instance. If A is greater than -7, the output of the node is 7+A, with a slope of 1.

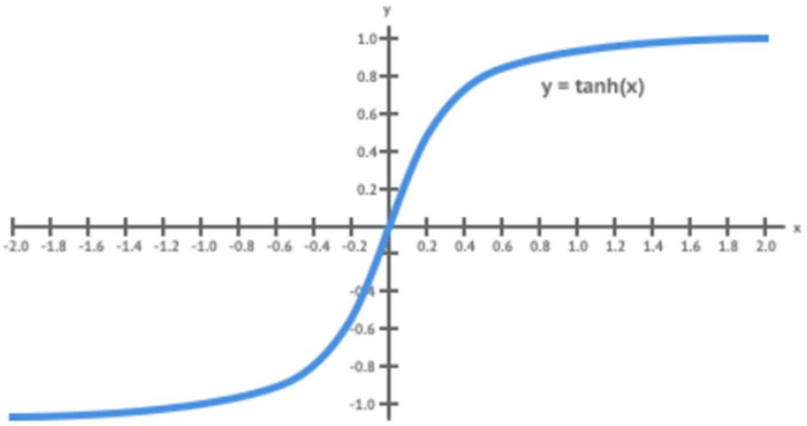
As a result of the bias term, we can move where the slope changes. So far, it looks that we can only have two distinct slopes.

Real models, on the other hand, have numerous nodes. Because each node (even within a single layer) might have a distinct bias value, each node can vary slope at various input values.

We obtain a combined function that changes slopes in numerous places when we add the resultant functions back up.

These models may generate non-linear functions and account for interactions well (if that will giv better predictions). The model's capacity to reflect these interactions and non-linearities grows as we add more nodes in each layer (or more convolutions if we're using a convolutional model).

S-shaped curves were once common in deep learning algorithms (like the tanh function below).



The tanh appears to have a few advantages. Even though it approaches flatness, it is never totally flat. As a result, its output is constantly reflective of changes in its input, which we would think is a positive thing. It is also non-linear in nature (or curved everywhere). One of the key goals of the activation function is to account for non-linearities. As a result, we anticipate that a non-linear function will perform well.

When utilizing the tanh function, however, researchers encountered a lot of trouble creating models with a lot of layers. Except for a very tiny range, it is essentially flat (that range being about -2 to 2). Unless the input is in this restricted range, the function's derivative is relatively tiny, and this flat derivative makes gradient descent difficult to enhance the weights. As the model grows in layers, the issue becomes more severe. The vanishing gradient issue was named after this.

Over half of its range, the ReLU function has a derivative of 0. (the negative numbers). The derivative is 1 for positive inputs.

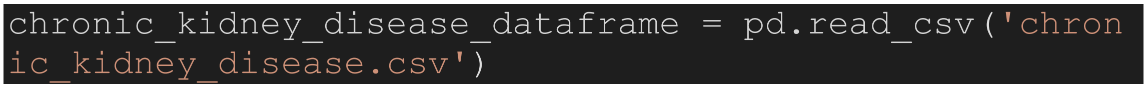
There will almost always be some data points yielding positive values to each given node when training on a suitably sized batch. As a result, the average derivative is rarely near to 0, allowing gradient descent to continue.

ML MODEL USED IN THE APP

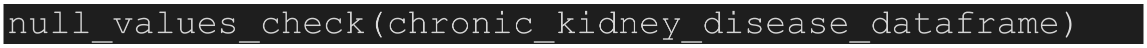
mLModel.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.impute import KNNImputer
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
from sklearn import metrics
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras.wrappers.scikit_learn im
port KerasRegressor
```

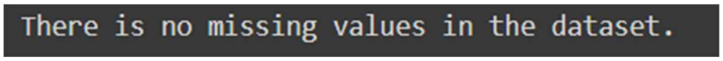
```
def null_values_check(df):
    #Error handling to prevent abnormal termination o
f operation
    try:
        #if-else statement for null value check
        if(df.isnull().values.any() == True):
            #if there are null values present, print
a column-wise summary of records with null values
            print('Number of null records within each
 column:\n' + str(df.isnull().sum()))
        else:
            print('There is no missing values in the
dataset.')

    except Exception as e:
        logging.error(e)
```



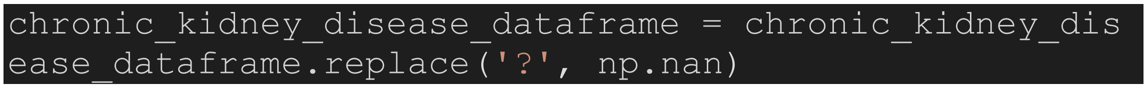
```
chronic_kidney_disease_dataframe = pd.read_csv('chronic_kidney_disease.csv')
```



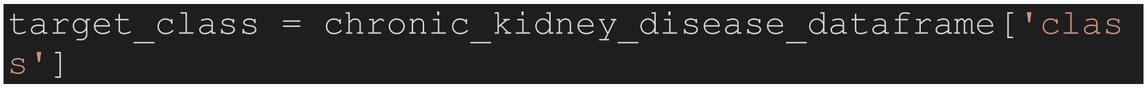
```
null_values_check(chronic_kidney_disease_dataframe)
```



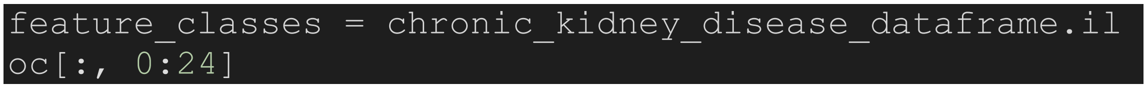
```
There is no missing values in the dataset.
```



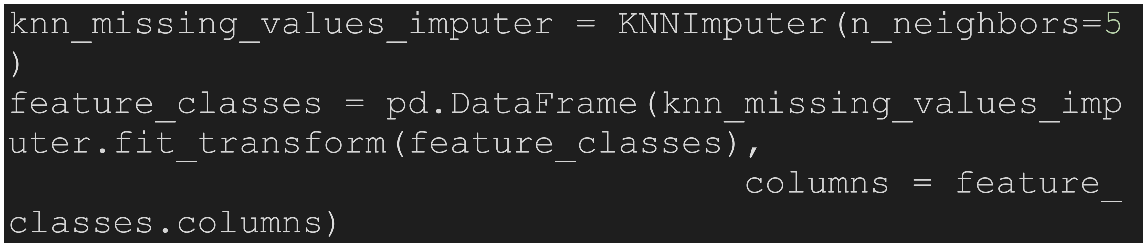
```
chronic_kidney_disease_dataframe = chronic_kidney_disease_dataframe.replace('?', np.nan)
```



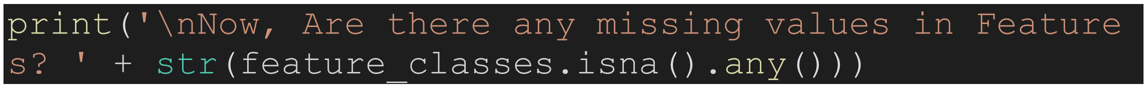
```
target_class = chronic_kidney_disease_dataframe['class']
```



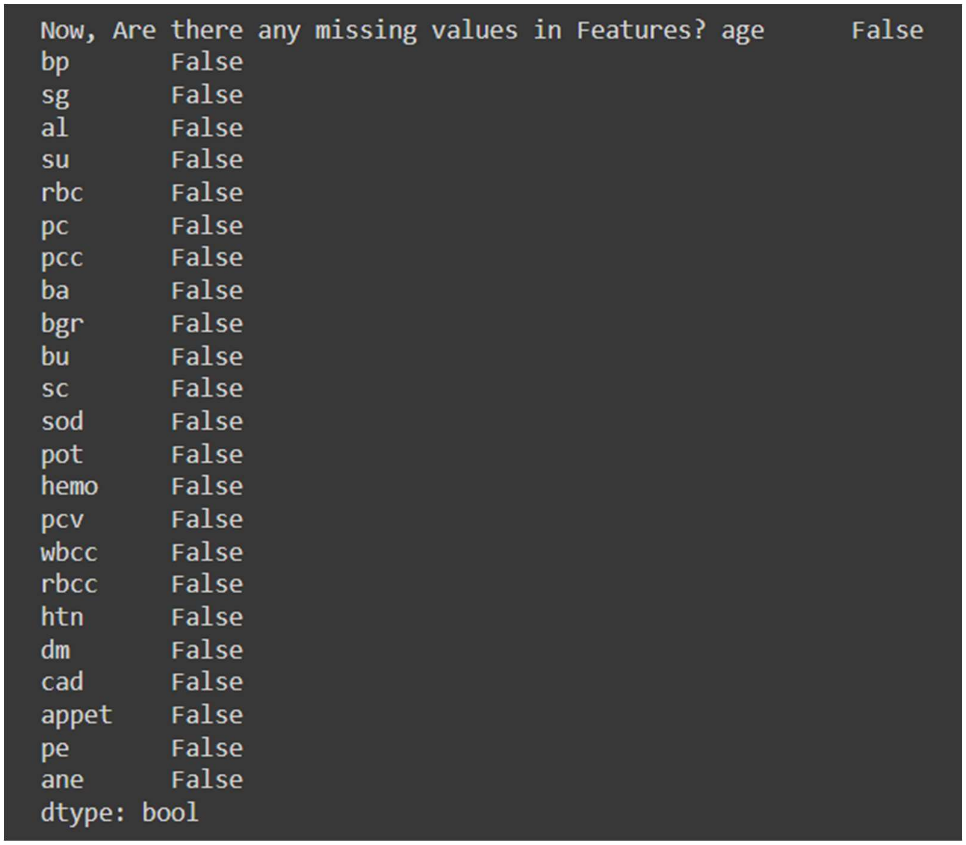
```
feature_classes = chronic_kidney_disease_dataframe.iloc[:, 0:24]
```



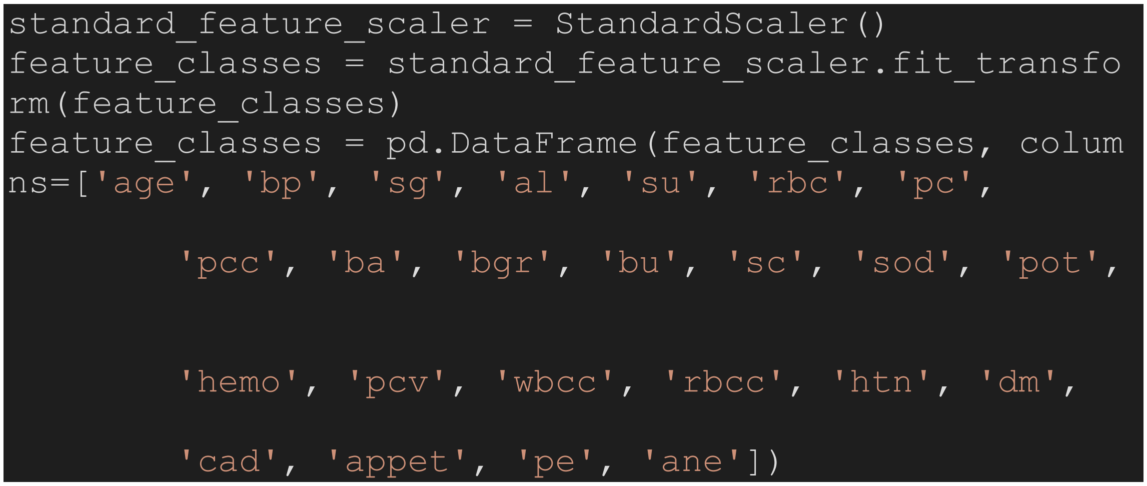
```
knn_missing_values_imputer = KNNImputer(n_neighbors=5)
feature_classes = pd.DataFrame(knn_missing_values_imputer.fit_transform(feature_classes),
                                   columns = feature_classes.columns)
```



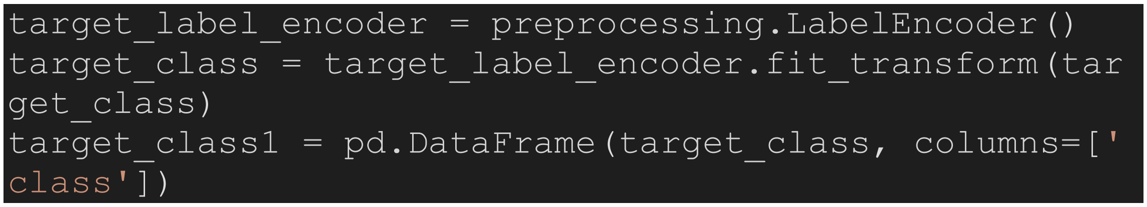
```
print('\nNow, Are there any missing values in Features? ' + str(feature_classes.isna().any()))
```



```
Now, Are there any missing values in Features? age      False
bp       False
sg       False
al       False
su       False
rbc      False
pc       False
pcc      False
ba       False
bgr      False
bu       False
sc       False
sod      False
pot      False
hemo     False
pcv      False
wbcc     False
rbcc     False
htn      False
dm       False
cad      False
appet    False
pe       False
ane      False
dtype: bool
```



```
standard_feature_scaler = StandardScaler()
feature_classes = standard_feature_scaler.fit_transform(feature_classes)
feature_classes = pd.DataFrame(feature_classes, columns=['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc',

           'pcc', 'ba', 'bgr', 'bu', 'sc', 'sod', 'pot',


           'hemo', 'pcv', 'wbcc', 'rbcc', 'htn', 'dm',

           'cad', 'appet', 'pe', 'ane'])
```

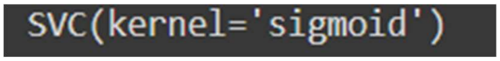


```
target_label_encoder = preprocessing.LabelEncoder()
target_class = target_label_encoder.fit_transform(target_class)
target_class1 = pd.DataFrame(target_class, columns=['class'])
```

```
train_features, test_features, train_target, test_tar
get = train_test_split(feature_classes, target_class,


                                   train_size = 0.7, test_siz
e = 0.3)
```

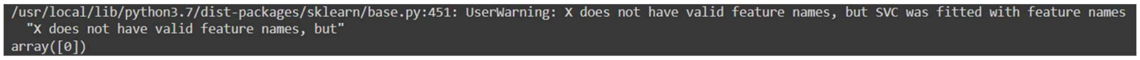
```
from sklearn.svm import SVC
clf = SVC(kernel='sigmoid')
```

```
clf.fit(train_features, train_target)
```



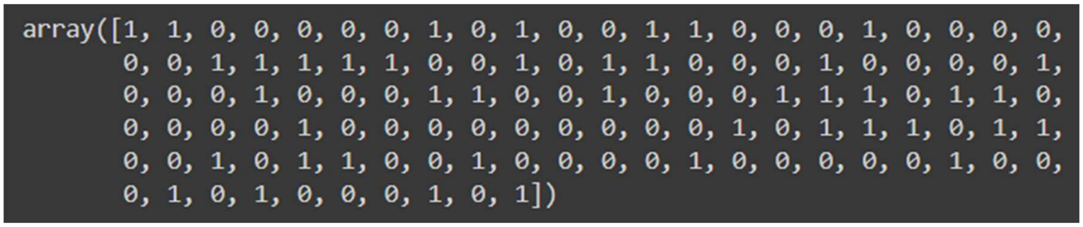
```
SVC(kernel='sigmoid')
```

```
clf.predict([[35,80,1.02,0,0,0,0,1,1,104,31,1.2,135,5
,16.1,45,4300,5.2,1,1,1,1,1,1]])
```



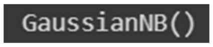
```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
  "X does not have valid feature names, but"
array([0])
```

```
clf.predict(test_features)
```



```
array([1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 1, 0, 1])
```

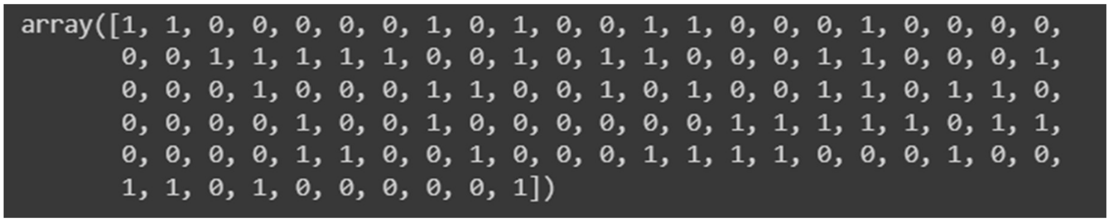
```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(train_features,train_target)
```



```
GaussianNB()
```

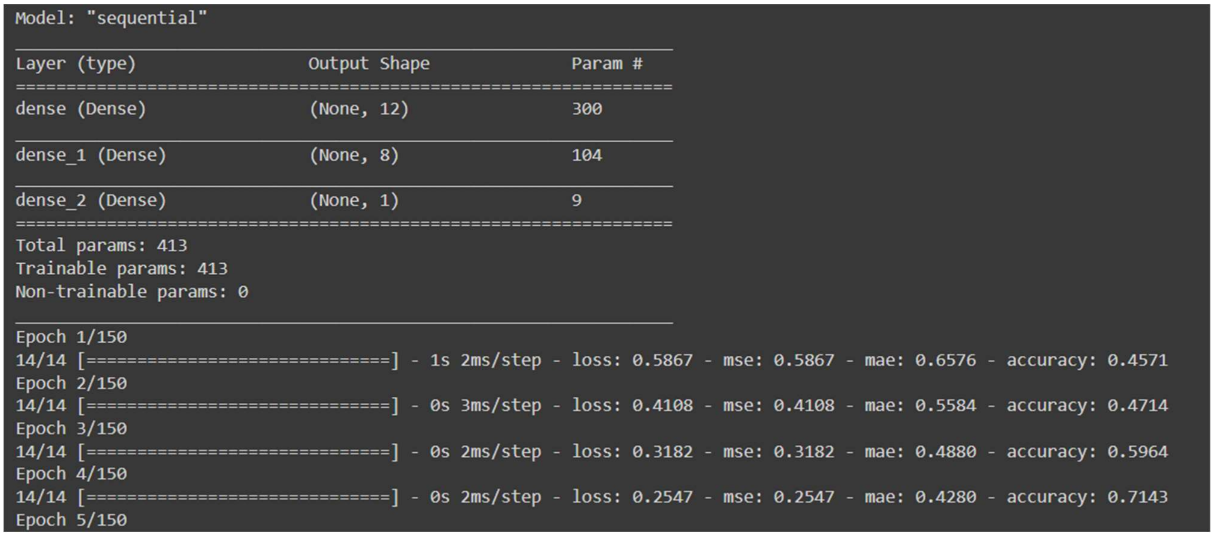
```
y_pred  =  classifier.predict(test_features)
```

```
y_pred
```

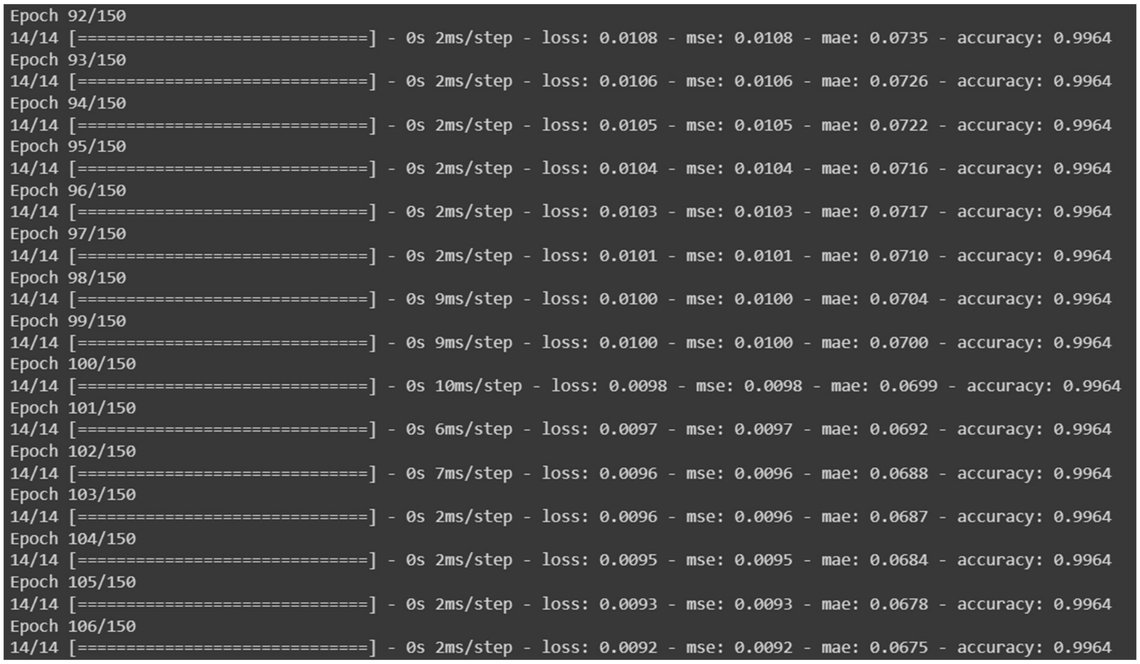


```
array([1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 1])
```

```
model=Sequential()
model.add(Dense(12, input_dim=24, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='linear'))
model.summary()
model.compile(loss='mse',optimizer='adam',metrics=['m
se','mae','accuracy'])
history=model.fit(train_features,train_target,epochs=
150,batch_size=20)
```



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 12)                300
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 104
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 9
=================================================================
Total params: 413
Trainable params: 413
Non-trainable params: 0
_________________________________________________________________
Epoch 1/150
14/14 [==============================] - 1s 2ms/step - loss: 0.5867 - mse: 0.5867 - mae: 0.6576 - accuracy: 0.4571
Epoch 2/150
14/14 [==============================] - 0s 3ms/step - loss: 0.4108 - mse: 0.4108 - mae: 0.5584 - accuracy: 0.4714
Epoch 3/150
14/14 [==============================] - 0s 2ms/step - loss: 0.3182 - mse: 0.3182 - mae: 0.4880 - accuracy: 0.5964
Epoch 4/150
14/14 [==============================] - 0s 2ms/step - loss: 0.2547 - mse: 0.2547 - mae: 0.4280 - accuracy: 0.7143
Epoch 5/150
```



```
Epoch 92/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0108 - mse: 0.0108 - mae: 0.0735 - accuracy: 0.9964
Epoch 93/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0106 - mse: 0.0106 - mae: 0.0726 - accuracy: 0.9964
Epoch 94/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0105 - mse: 0.0105 - mae: 0.0722 - accuracy: 0.9964
Epoch 95/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0104 - mse: 0.0104 - mae: 0.0716 - accuracy: 0.9964
Epoch 96/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0103 - mse: 0.0103 - mae: 0.0717 - accuracy: 0.9964
Epoch 97/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0101 - mse: 0.0101 - mae: 0.0710 - accuracy: 0.9964
Epoch 98/150
14/14 [==============================] - 0s 9ms/step - loss: 0.0100 - mse: 0.0100 - mae: 0.0704 - accuracy: 0.9964
Epoch 99/150
14/14 [==============================] - 0s 9ms/step - loss: 0.0100 - mse: 0.0100 - mae: 0.0700 - accuracy: 0.9964
Epoch 100/150
14/14 [==============================] - 0s 10ms/step - loss: 0.0098 - mse: 0.0098 - mae: 0.0699 - accuracy: 0.9964
Epoch 101/150
14/14 [==============================] - 0s 6ms/step - loss: 0.0097 - mse: 0.0097 - mae: 0.0692 - accuracy: 0.9964
Epoch 102/150
14/14 [==============================] - 0s 7ms/step - loss: 0.0096 - mse: 0.0096 - mae: 0.0688 - accuracy: 0.9964
Epoch 103/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0096 - mse: 0.0096 - mae: 0.0687 - accuracy: 0.9964
Epoch 104/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0095 - mse: 0.0095 - mae: 0.0684 - accuracy: 0.9964
Epoch 105/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0093 - mse: 0.0093 - mae: 0.0678 - accuracy: 0.9964
Epoch 106/150
14/14 [==============================] - 0s 2ms/step - loss: 0.0092 - mse: 0.0092 - mae: 0.0675 - accuracy: 0.9964
```

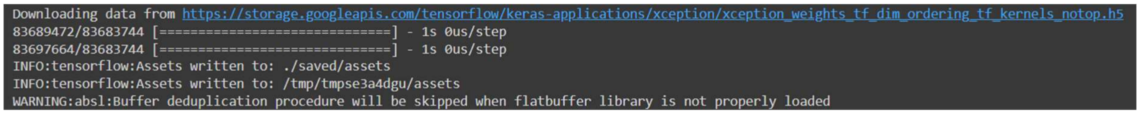
```
model.save('C:/Users/1only/Downloads/Madhyam/Mini-Project-master/result.h5')
```

```
from tensorflow.python import keras
import tensorflow as tf
modelnew=keras.models.load_model('C:/Users/1only/Downloads/Madhyam/Mini-Project-master/result.h5')
```

```
converter=tf.lite.TFLiteConverter.from_keras_model(modelnew)
```

```

# open("converted model.tflite", "wb").write(tflite_model)
```

```
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.applications.xception import Xception

class MyModel(tf.keras.Model):

  def __init__(self, input_shape, num_classes=5, dropout_rate=0.5):
    super(MyModel, self).__init__()
    self.weight_dict = {}
    self.weight_dict['backbone'] = Xception(input_sha
pe=input_shape, weights='imagenet', include_top=False
)
    self.weight_dict['outputs'] = Conv2D(num_classes,
 (1, 1), padding="same", activation="softmax")
    self.build((None,) + input_shape)

  def call(self, inputs, training=False):
    self.weight_dict['backbone'].trainable = False
    x = self.weight_dict['backbone'](inputs)
    x = self.weight_dict['outputs'](x)
    return x

input_shape=(256, 256, 3)
model=MyModel(input_shape)

model.compute_output_shape(input_shape=(None, 256, 25
6, 3))
model.save('./saved')
tflite_model=converter.convert()
```

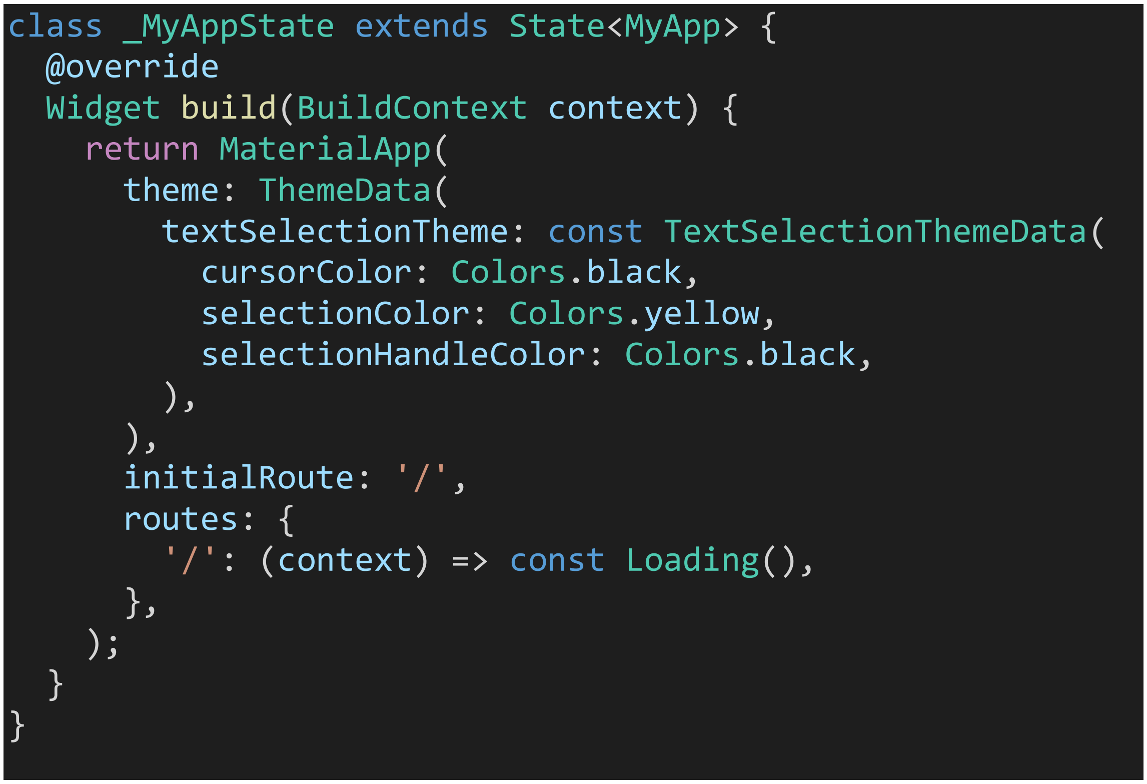


```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [==============================] - 1s 0us/step
83697664/83683744 [==============================] - 1s 0us/step
INFO:tensorflow:Assets written to: ./saved/assets
INFO:tensorflow:Assets written to: /tmp/tmpse3a4dgu/assets
WARNING:absl:Buffer deduplication procedure will be skipped when flatbuffer library is not properly loaded
```

UIs OF THE APP

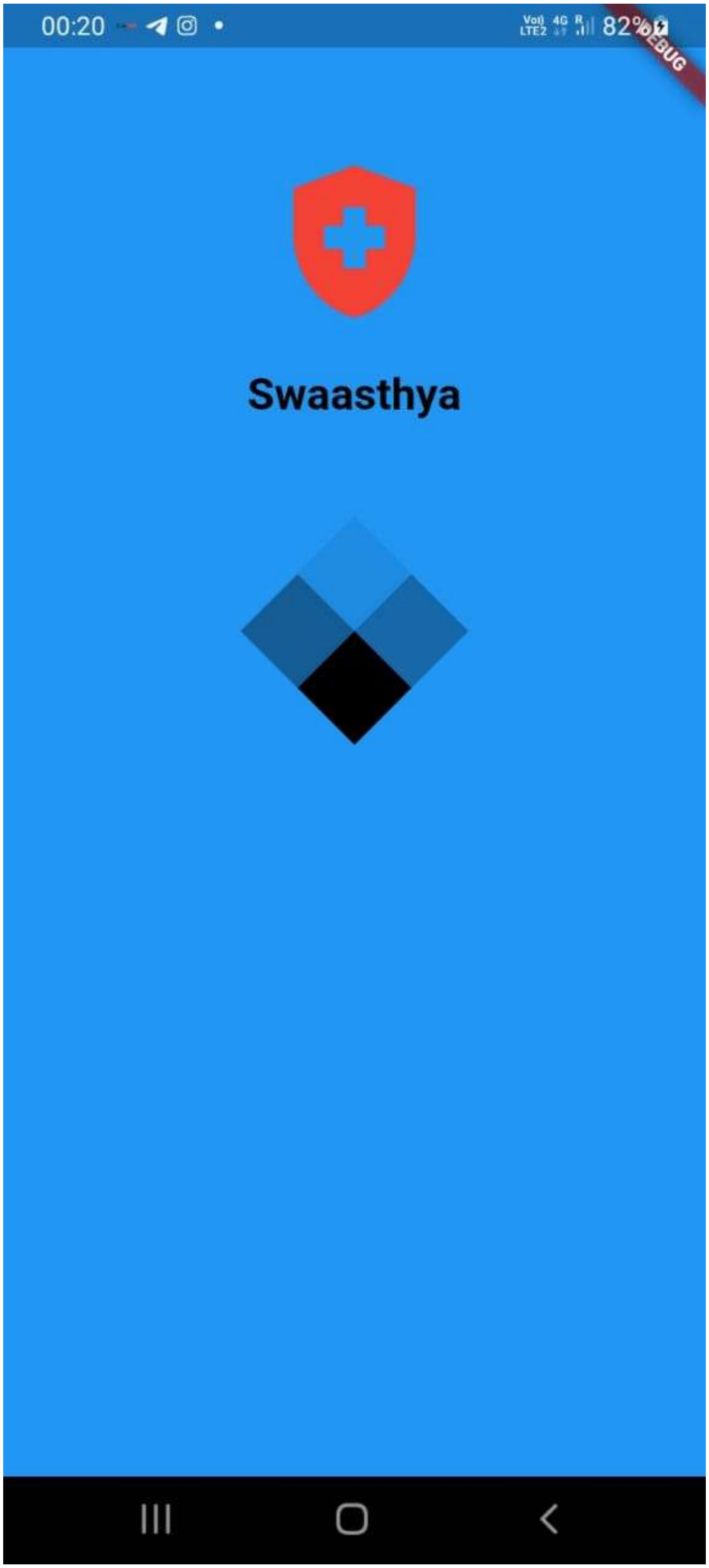
main.dart

```
import 'package:flutter/material.dart';
import 'package:swaasthya/pages/loading.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  State<MyApp> createState() => _MyAppState();
}
```



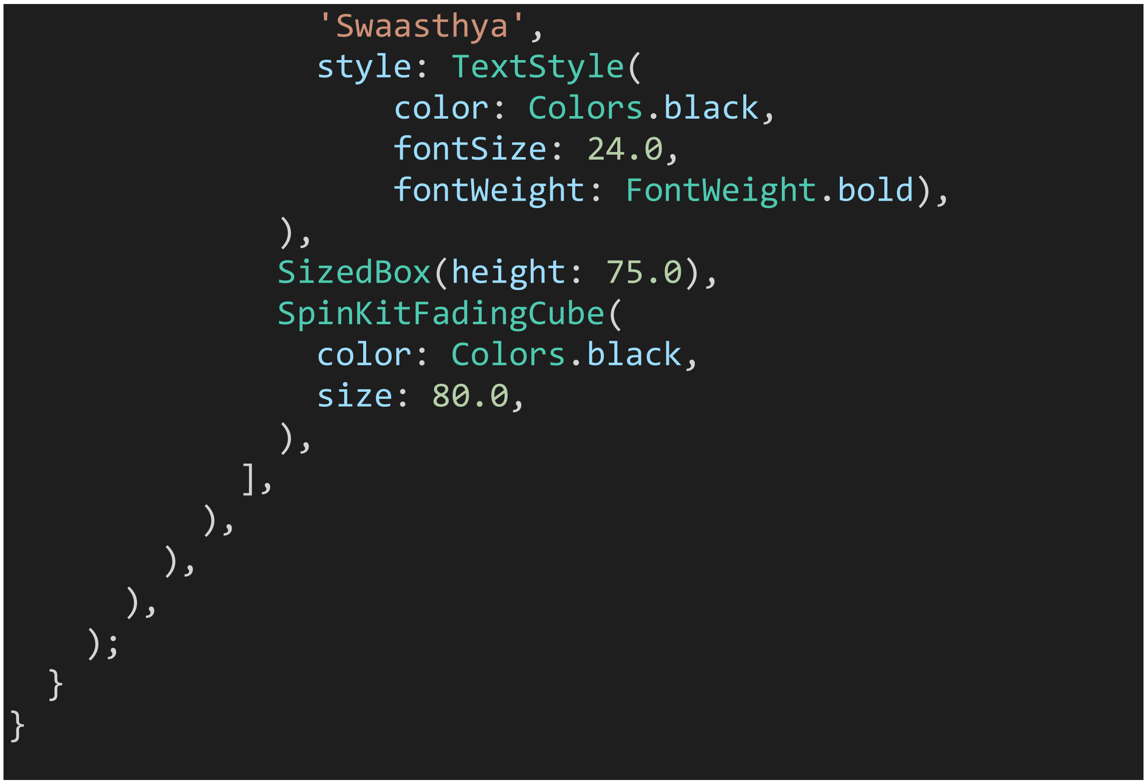
```dart
class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        textSelectionTheme: const TextSelectionThemeData(
          cursorColor: Colors.black,
          selectionColor: Colors.yellow,
          selectionHandleColor: Colors.black,
        ),
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => const Loading(),
      },
    );
  }
}
```

0. Splash Screen


00:20
82%
DEBUG
Swaasthya

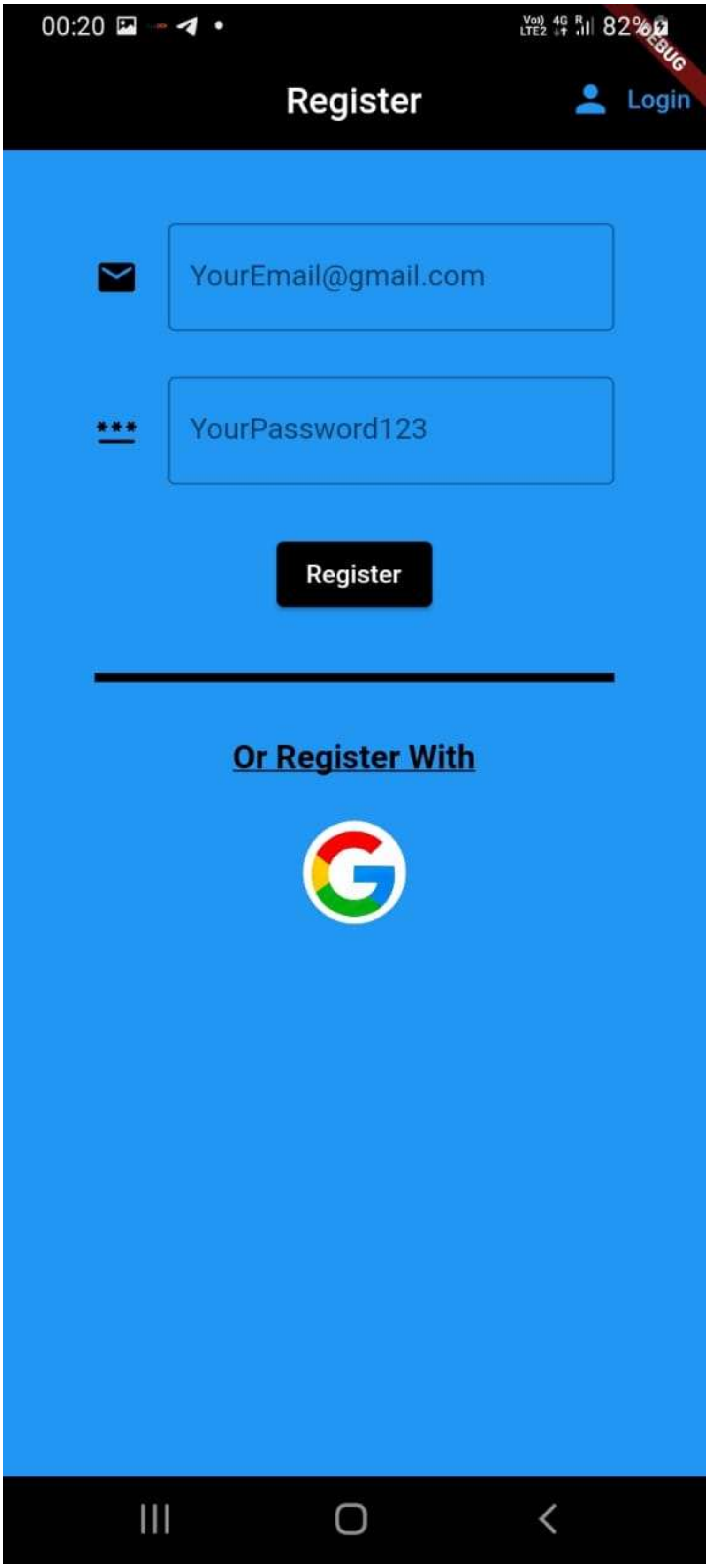
Code:

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';
import 'package:swaasthya/pages/register.dart';
// import 'package:swasthya/pages/symptoms.dart';

class Loading extends StatefulWidget {
  const Loading({Key? key}) : super(key: key);

  @override
  _LoadingState createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {
  @override
  void initState() {
    super.initState();
    Timer(
        const Duration(seconds: 2),
        () => Navigator.pushReplacement(context,
            MaterialPageRoute(builder: (context) => const
Register())));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      body: Padding(
        padding: const EdgeInsets.all(80.0),
        child: Center(
          child: Column(
            children: const <Widget>[
              Icon(
                Icons.health_and_safety,
                color: Colors.red,
                size: 100.0,
              ),
              SizedBox(height: 20.0),
              Text(
```

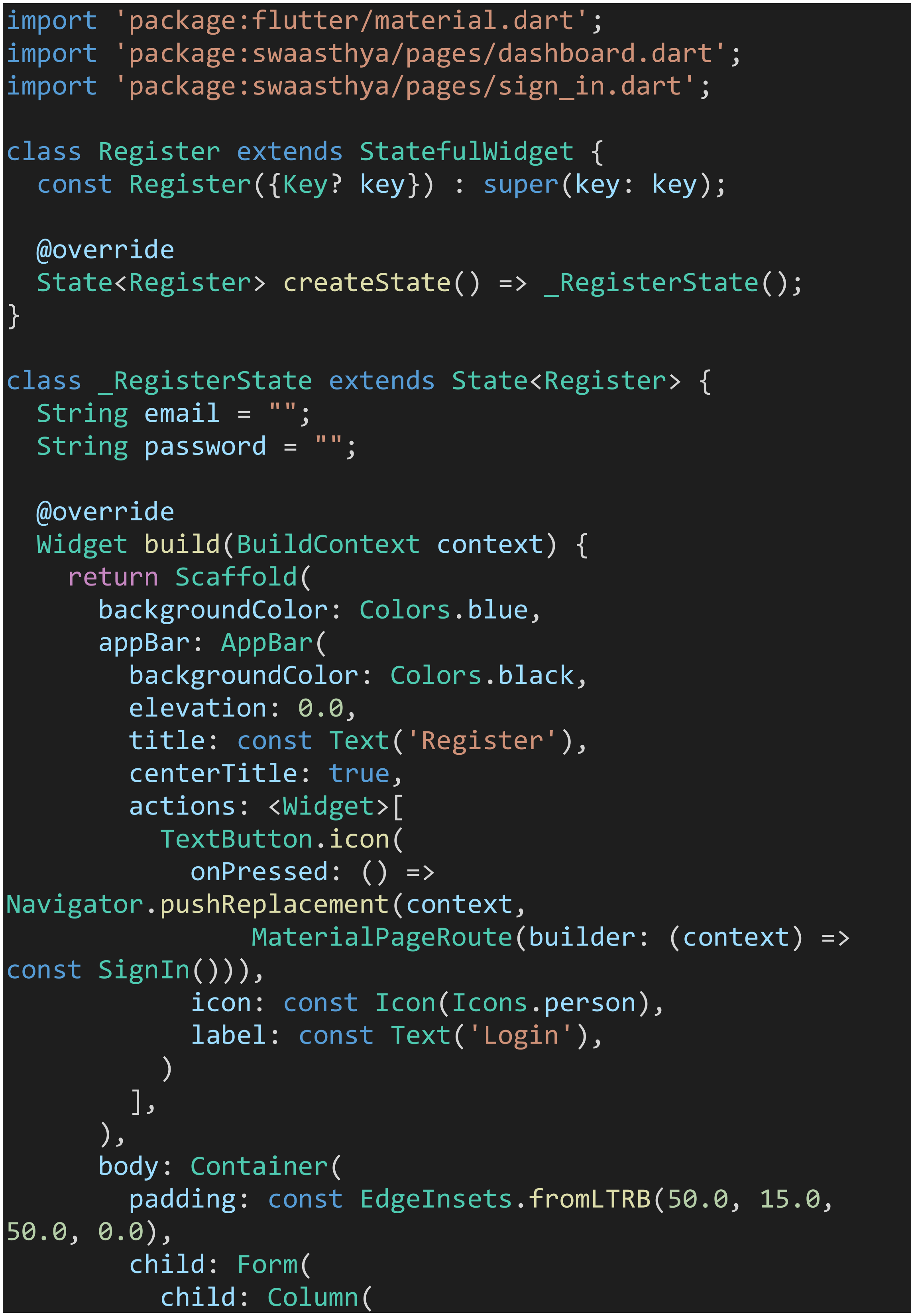


```
                'Swaasthya',
                style: TextStyle(
                    color: Colors.black,
                    fontSize: 24.0,
                    fontWeight: FontWeight.bold),
              ),
              SizedBox(height: 75.0),
              SpinKitFadingCube(
                color: Colors.black,
                size: 80.0,
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

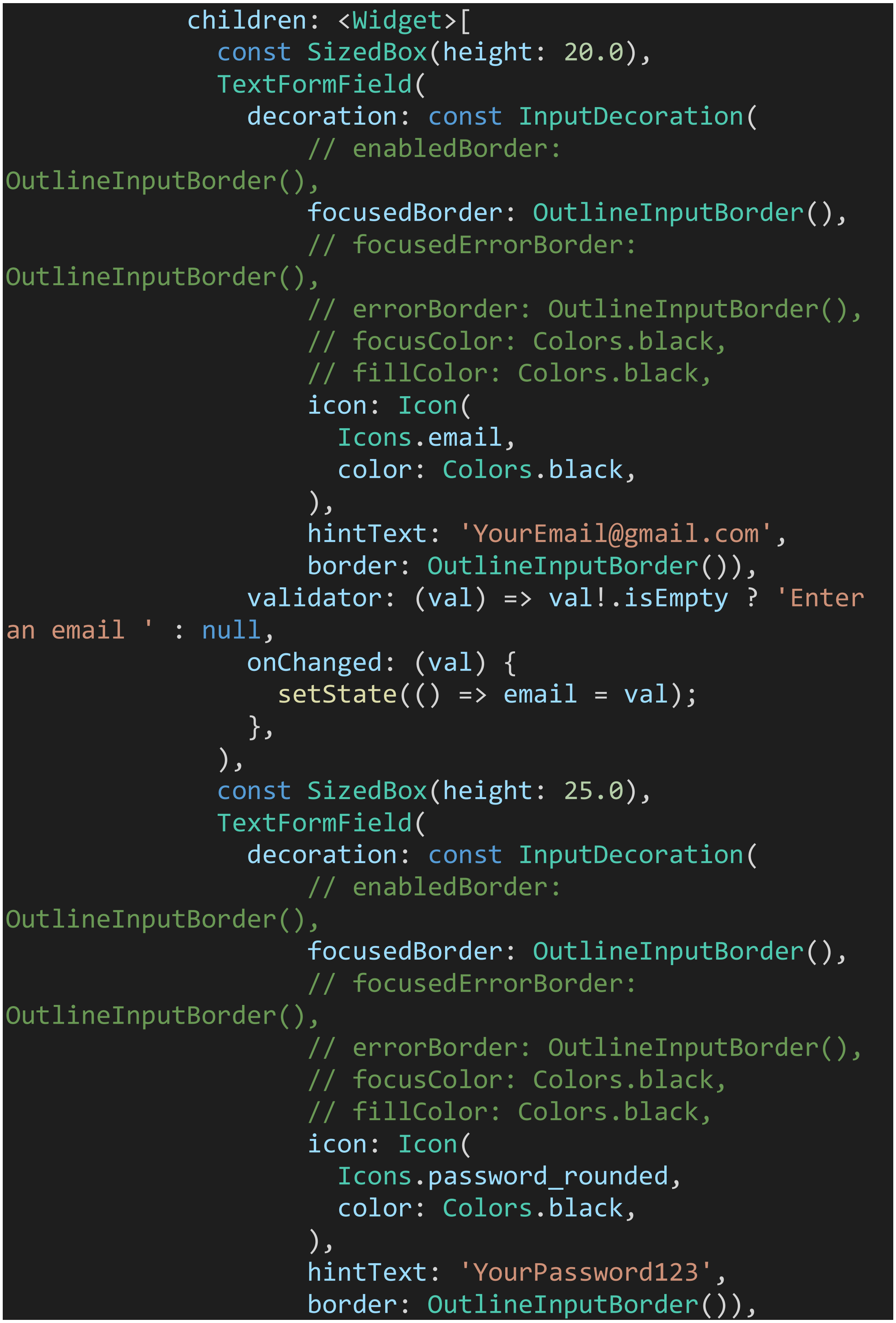
1. **Register Page**


00:20
82%
DEBUG
Register
Login
YourEmail@gmail.com
YourPassword123
Register
Or Register With

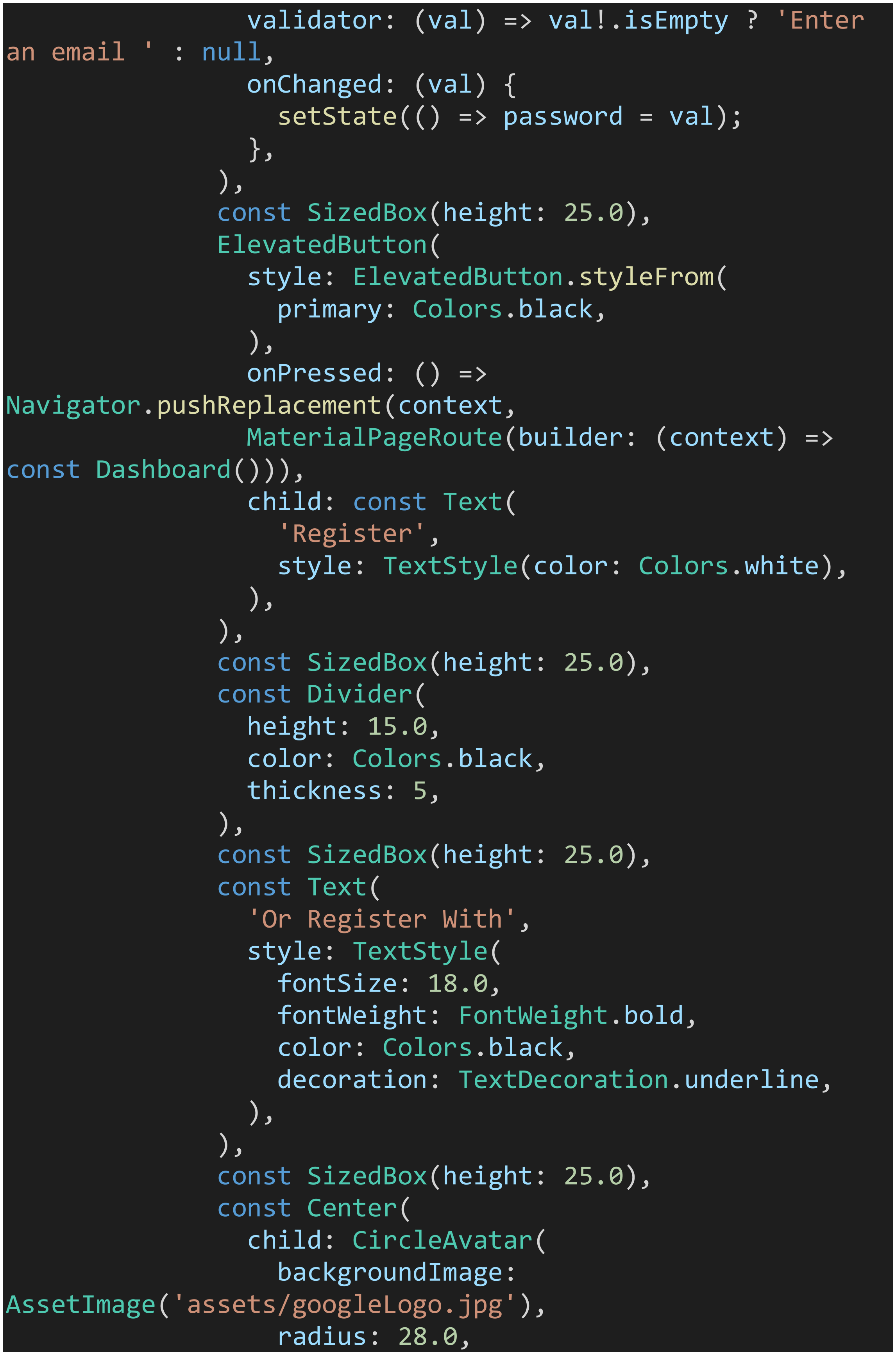
Code:



```
import 'package:flutter/material.dart';
import 'package:swaasthya/pages/dashboard.dart';
import 'package:swaasthya/pages/sign_in.dart';

class Register extends StatefulWidget {
  const Register({Key? key}) : super(key: key);

  @override
  State<Register> createState() => _RegisterState();
}

class _RegisterState extends State<Register> {
  String email = "";
  String password = "";

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      appBar: AppBar(
        backgroundColor: Colors.black,
        elevation: 0.0,
        title: const Text('Register'),
        centerTitle: true,
        actions: <Widget>[
          TextButton.icon(
            onPressed: () =>
Navigator.pushReplacement(context,
                MaterialPageRoute(builder: (context) =>
const SignIn())),
            icon: const Icon(Icons.person),
            label: const Text('Login'),
          )
        ],
      ),
      body: Container(
        padding: const EdgeInsets.fromLTRB(50.0, 15.0,
50.0, 0.0),
        child: Form(
          child: Column(
```

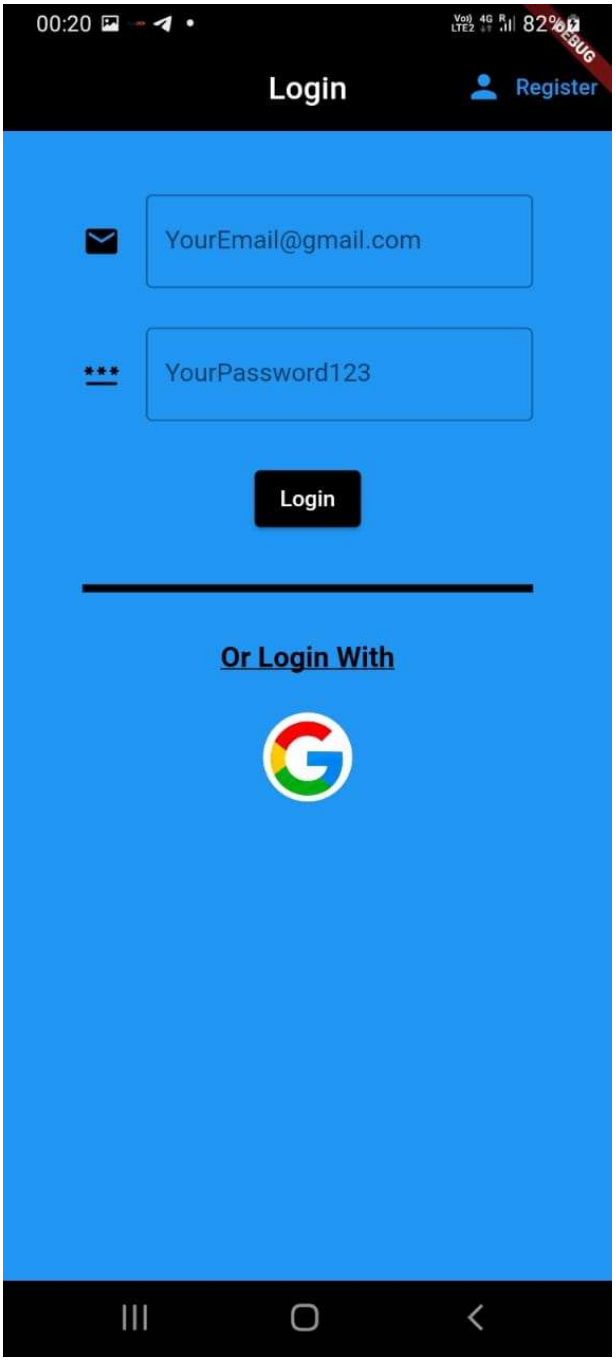


```
            children: <Widget>[
              const SizedBox(height: 20.0),
              TextFormField(
                decoration: const InputDecoration(
                    // enabledBorder:
OutlineInputBorder(),
                    focusedBorder: OutlineInputBorder(),
                    // focusedErrorBorder:
OutlineInputBorder(),
                    // errorBorder: OutlineInputBorder(),
                    // focusColor: Colors.black,
                    // fillColor: Colors.black,
                    icon: Icon(
                      Icons.email,
                      color: Colors.black,
                    ),
                    hintText: 'YourEmail@gmail.com',
                    border: OutlineInputBorder()),
                validator: (val) => val!.isEmpty ? 'Enter
an email ' : null,
                onChanged: (val) {
                  setState(() => email = val);
                },
              ),
              const SizedBox(height: 25.0),
              TextFormField(
                decoration: const InputDecoration(
                    // enabledBorder:
OutlineInputBorder(),
                    focusedBorder: OutlineInputBorder(),
                    // focusedErrorBorder:
OutlineInputBorder(),
                    // errorBorder: OutlineInputBorder(),
                    // focusColor: Colors.black,
                    // fillColor: Colors.black,
                    icon: Icon(
                      Icons.password_rounded,
                      color: Colors.black,
                    ),
                    hintText: 'YourPassword123',
                    border: OutlineInputBorder()),
```



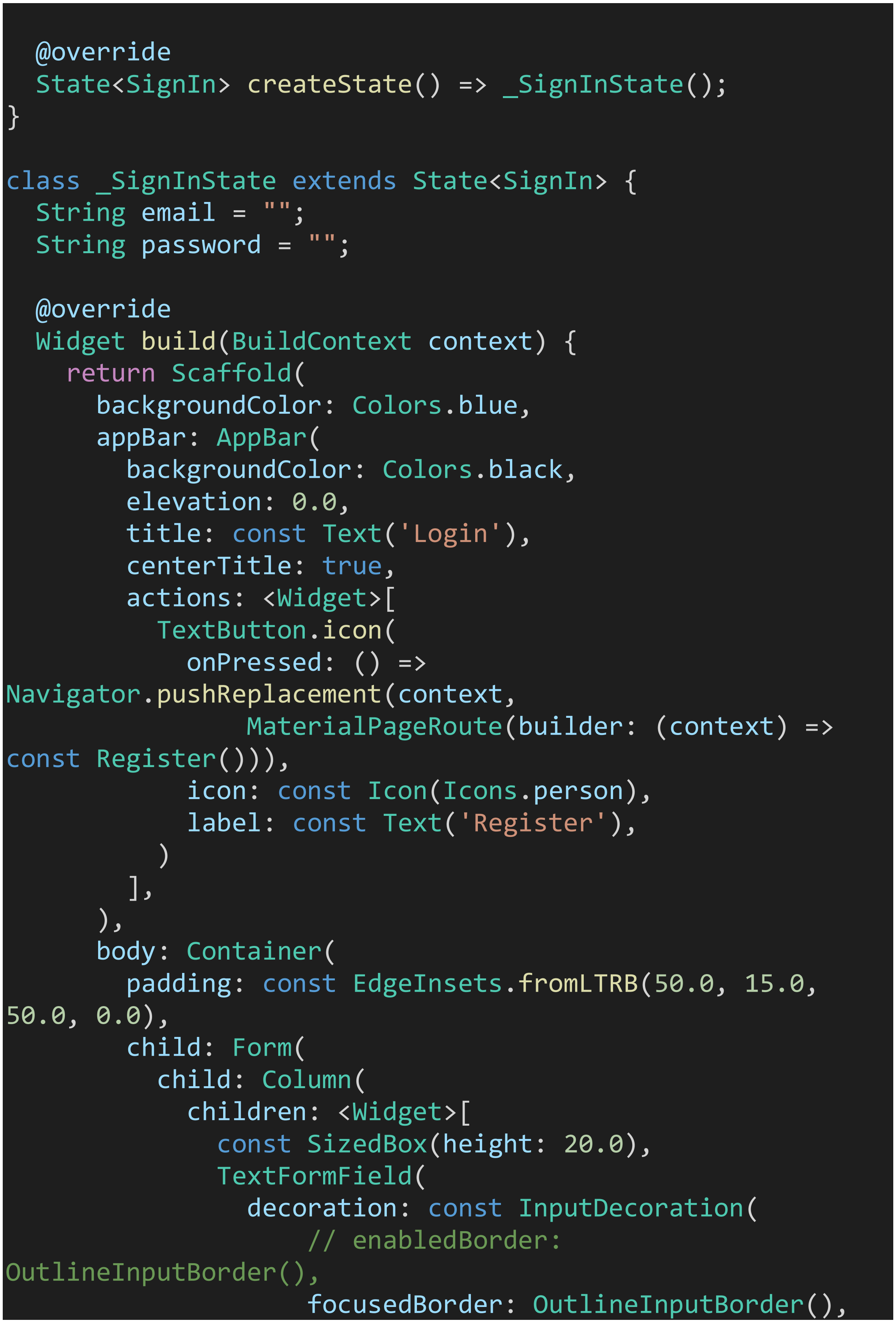
```
                validator: (val) => val!.isEmpty ? 'Enter
an email ' : null,
                onChanged: (val) {
                  setState(() => password = val);
                },
              ),
              const SizedBox(height: 25.0),
              ElevatedButton(
                style: ElevatedButton.styleFrom(
                  primary: Colors.black,
                ),
                onPressed: () =>
Navigator.pushReplacement(context,
                MaterialPageRoute(builder: (context) =>
const Dashboard())),
                child: const Text(
                  'Register',
                  style: TextStyle(color: Colors.white),
                ),
              ),
              const SizedBox(height: 25.0),
              const Divider(
                height: 15.0,
                color: Colors.black,
                thickness: 5,
              ),
              const SizedBox(height: 25.0),
              const Text(
                'Or Register With',
                style: TextStyle(
                  fontSize: 18.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.black,
                  decoration: TextDecoration.underline,
                ),
              ),
              const SizedBox(height: 25.0),
              const Center(
                child: CircleAvatar(
                  backgroundImage:
AssetImage('assets/googleLogo.jpg'),
                  radius: 28.0,
                  backgroundColor: Colors.white,
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

2. **Login Page**

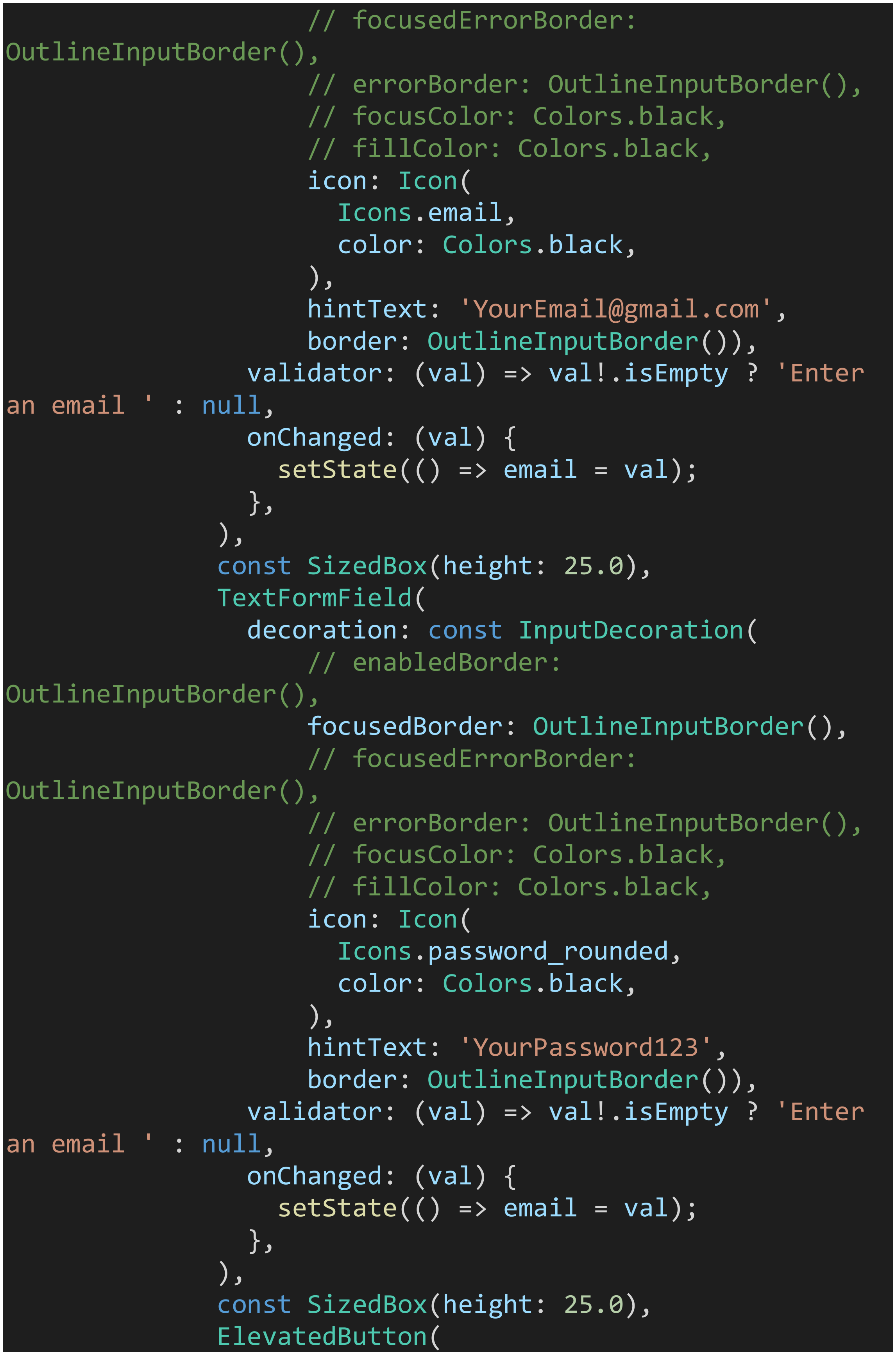


Code:

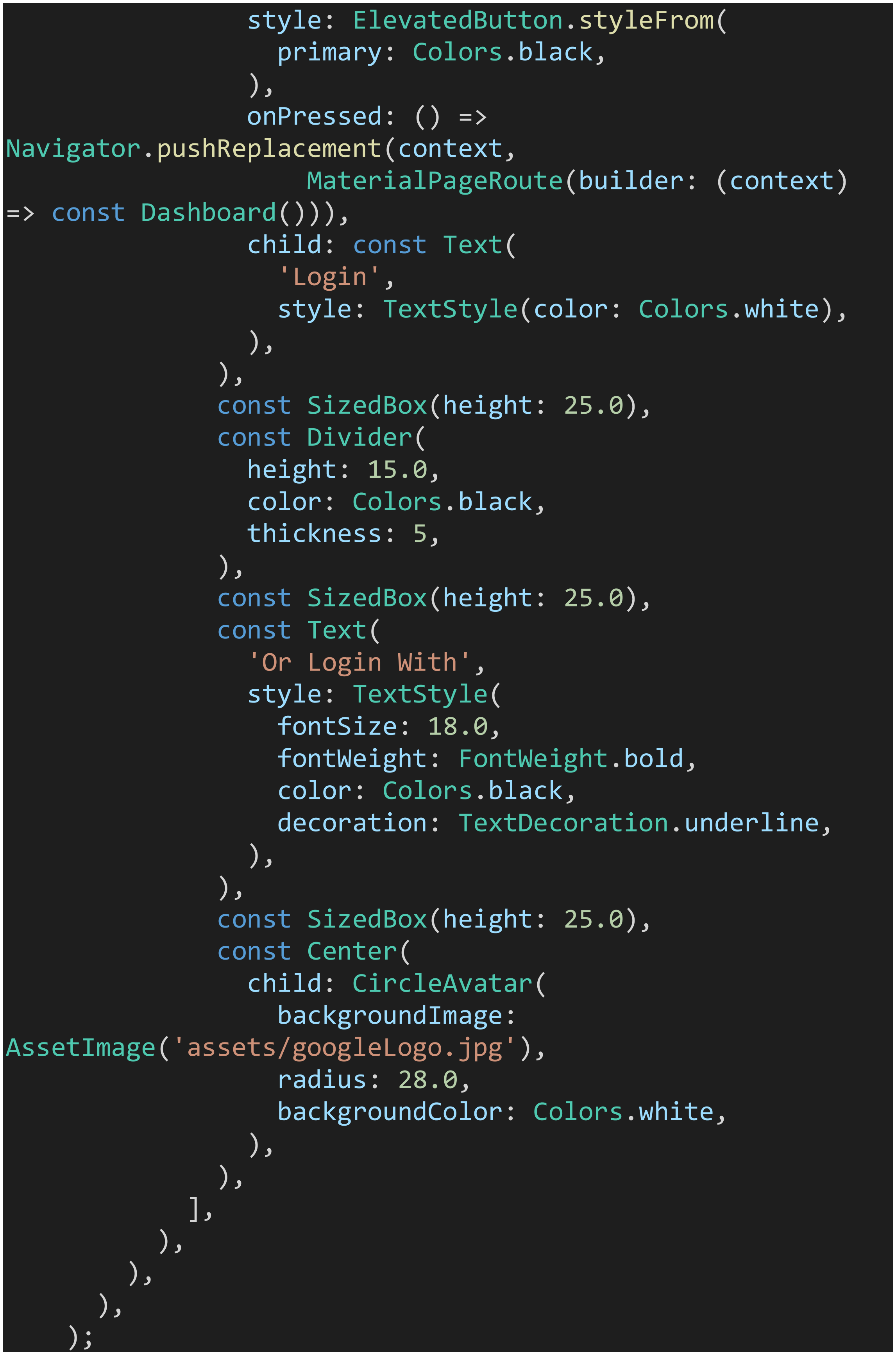
```
import 'package:flutter/material.dart';
import 'package:swaasthya/pages/dashboard.dart';
import 'package:swaasthya/pages/register.dart';

class SignIn extends StatefulWidget {
  const SignIn({Key? key}) : super(key: key);
```



```
  @override
  State<SignIn> createState() => _SignInState();
}

class _SignInState extends State<SignIn> {
  String email = "";
  String password = "";

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      appBar: AppBar(
        backgroundColor: Colors.black,
        elevation: 0.0,
        title: const Text('Login'),
        centerTitle: true,
        actions: <Widget>[
          TextButton.icon(
            onPressed: () =>
Navigator.pushReplacement(context,
                MaterialPageRoute(builder: (context) =>
const Register())),
            icon: const Icon(Icons.person),
            label: const Text('Register'),
          )
        ],
      ),
      body: Container(
        padding: const EdgeInsets.fromLTRB(50.0, 15.0,
50.0, 0.0),
        child: Form(
          child: Column(
            children: <Widget>[
              const SizedBox(height: 20.0),
              TextFormField(
                decoration: const InputDecoration(
                    // enabledBorder:
OutlineInputBorder(),
                    focusedBorder: OutlineInputBorder(),
```

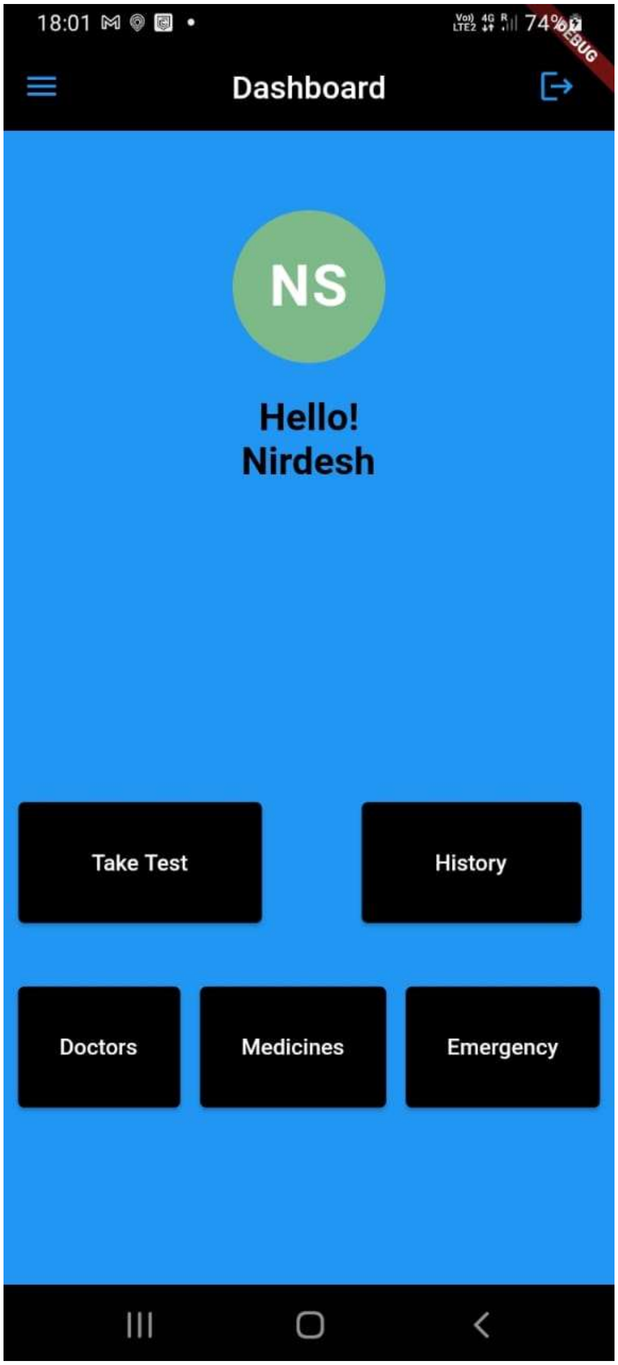


```
                    // focusedErrorBorder:
OutlineInputBorder(),
                    // errorBorder: OutlineInputBorder(),
                    // focusColor: Colors.black,
                    // fillColor: Colors.black,
                    icon: Icon(
                      Icons.email,
                      color: Colors.black,
                    ),
                    hintText: 'YourEmail@gmail.com',
                    border: OutlineInputBorder()),
                validator: (val) => val!.isEmpty ? 'Enter
an email ' : null,
                onChanged: (val) {
                  setState(() => email = val);
                },
              ),
              const SizedBox(height: 25.0),
              TextFormField(
                decoration: const InputDecoration(
                    // enabledBorder:
OutlineInputBorder(),
                    focusedBorder: OutlineInputBorder(),
                    // focusedErrorBorder:
OutlineInputBorder(),
                    // errorBorder: OutlineInputBorder(),
                    // focusColor: Colors.black,
                    // fillColor: Colors.black,
                    icon: Icon(
                      Icons.password_rounded,
                      color: Colors.black,
                    ),
                    hintText: 'YourPassword123',
                    border: OutlineInputBorder()),
                validator: (val) => val!.isEmpty ? 'Enter
an email ' : null,
                onChanged: (val) {
                  setState(() => email = val);
                },
              ),
              const SizedBox(height: 25.0),
              ElevatedButton(
```



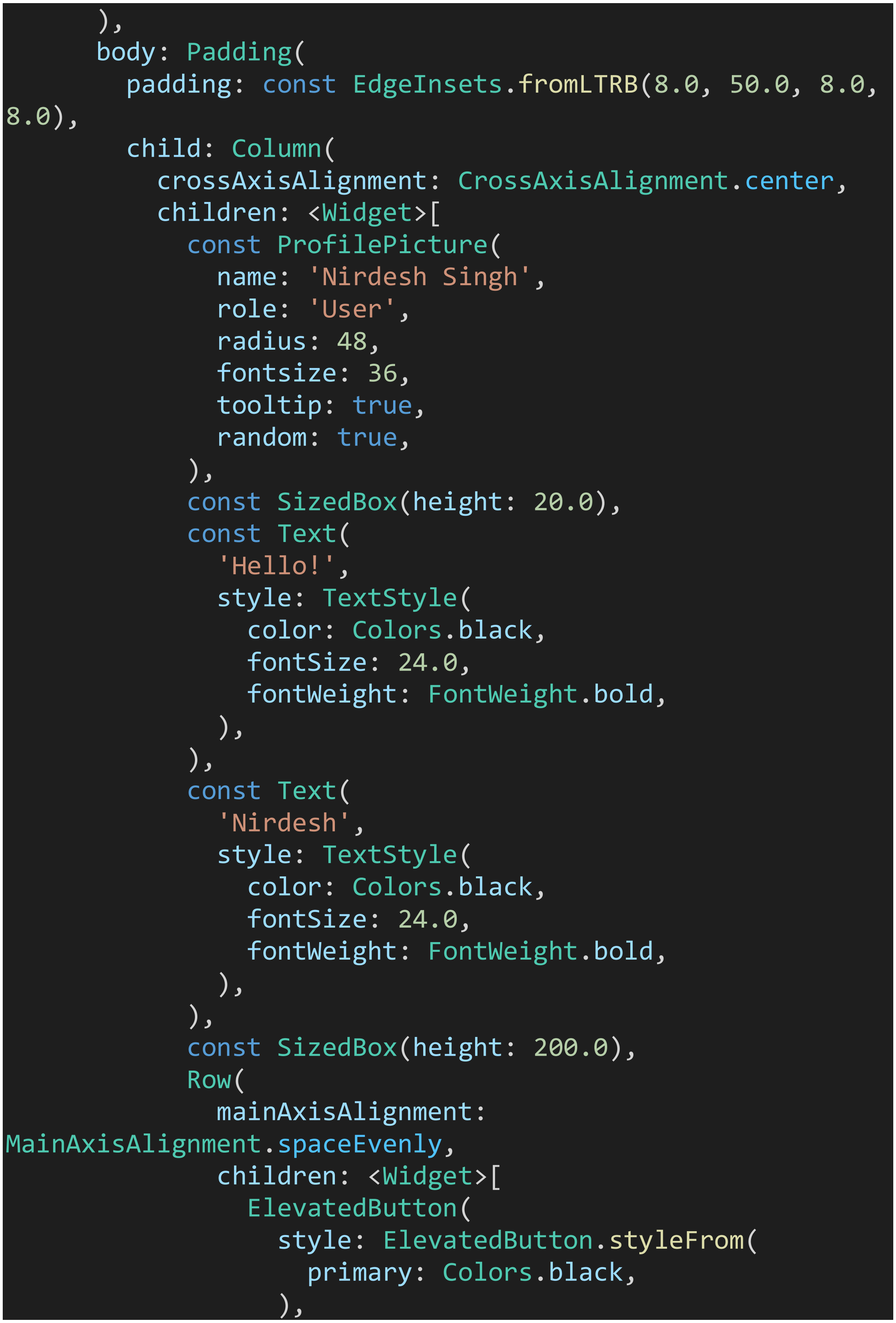
```
                style: ElevatedButton.styleFrom(
                  primary: Colors.black,
                ),
                onPressed: () =>
Navigator.pushReplacement(context,
                    MaterialPageRoute(builder: (context)
=> const Dashboard())),
                child: const Text(
                  'Login',
                  style: TextStyle(color: Colors.white),
                ),
              ),
              const SizedBox(height: 25.0),
              const Divider(
                height: 15.0,
                color: Colors.black,
                thickness: 5,
              ),
              const SizedBox(height: 25.0),
              const Text(
                'Or Login With',
                style: TextStyle(
                  fontSize: 18.0,
                  fontWeight: FontWeight.bold,
                  color: Colors.black,
                  decoration: TextDecoration.underline,
                ),
              ),
              const SizedBox(height: 25.0),
              const Center(
                child: CircleAvatar(
                  backgroundImage:
AssetImage('assets/googleLogo.jpg'),
                  radius: 28.0,
                  backgroundColor: Colors.white,
                ),
              ),
            ],
          ),
        ),
      ),
    );
    }
}
```

3. Dashboard

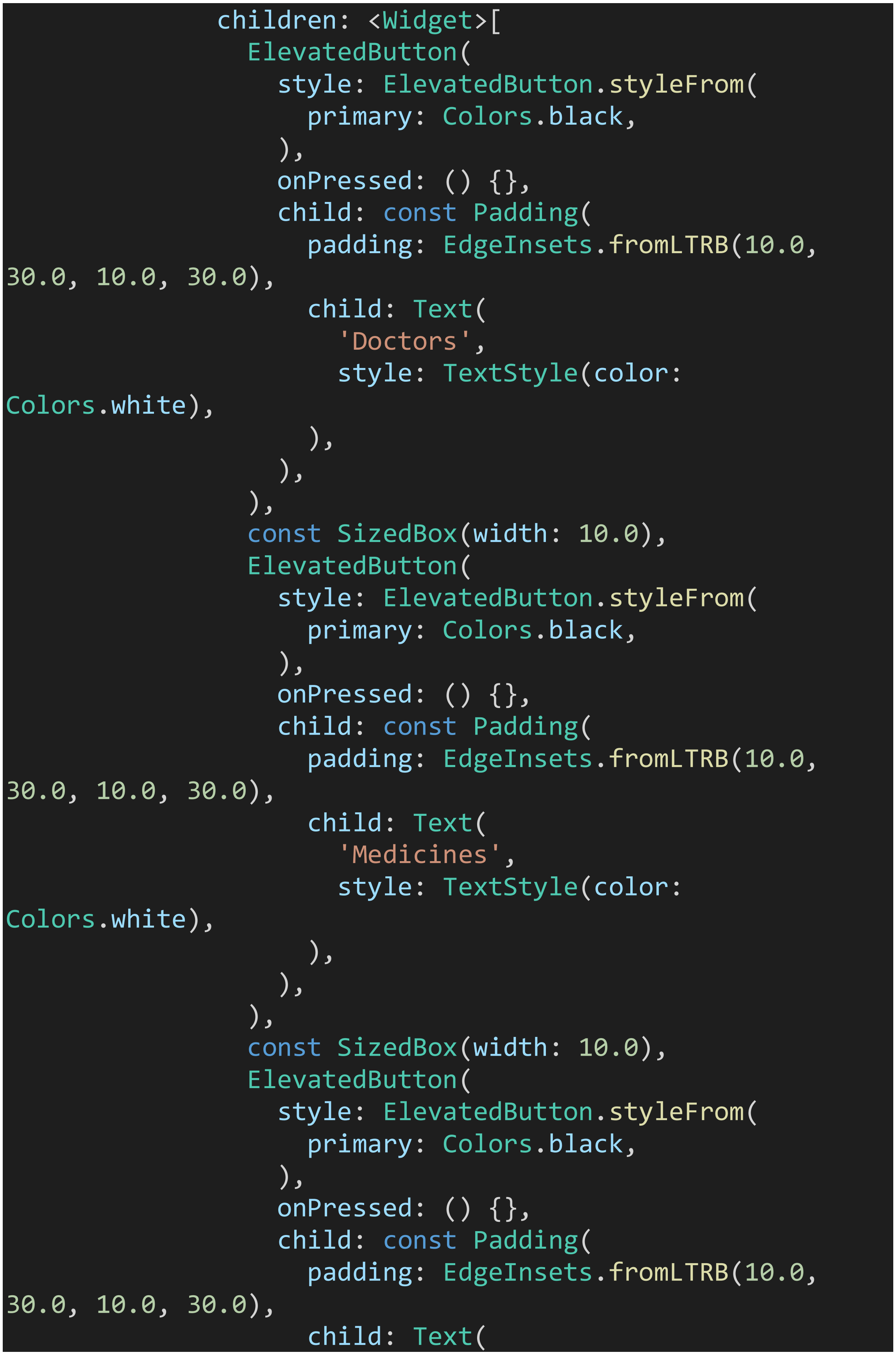

18:01
74%
DEBUG
Dashboard
NS
Hello!
Nirdesh
Take Test
History
Doctors
Medicines
Emergency

Code:

```
import 'package:flutter/material.dart';
import 'package:swaasthya/pages/questions.dart';
import 'package:swaasthya/pages/result.dart';
import 'package:swaasthya/pages/sign_in.dart';
import
'package:flutter_profile_picture/flutter_profile_picture.d
art';

class Dashboard extends StatefulWidget {
  const Dashboard({Key? key}) : super(key: key);

  @override
  State<Dashboard> createState() => _DashboardState();
}

class _DashboardState extends State<Dashboard> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      appBar: AppBar(
        backgroundColor: Colors.black,
        elevation: 0.0,
        title: const Text('Dashboard'),
        centerTitle: true,
        leading: TextButton.icon(
          onPressed: () {},
          icon: const Icon(Icons.menu),
          label: const Text(''),
        ),
        actions: <Widget>[
          TextButton.icon(
            onPressed: () =>
Navigator.pushReplacement(context,
                MaterialPageRoute(builder: (context) =>
const SignIn())),
            icon: const Icon(Icons.logout),
            label: const Text(''),
          ),
        ],
```



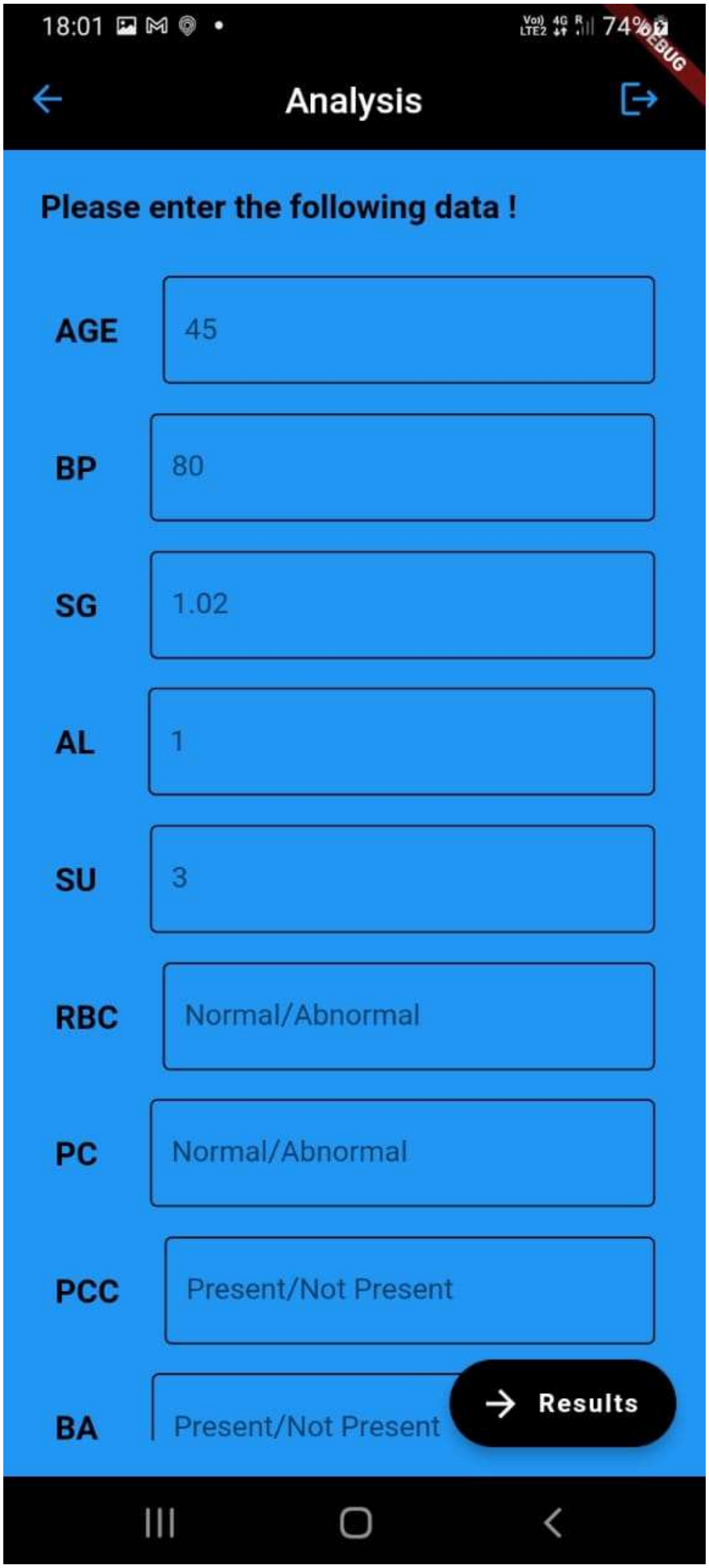
```
      ),
      body: Padding(
        padding: const EdgeInsets.fromLTRB(8.0, 50.0, 8.0,
8.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            const ProfilePicture(
              name: 'Nirdesh Singh',
              role: 'User',
              radius: 48,
              fontsize: 36,
              tooltip: true,
              random: true,
            ),
            const SizedBox(height: 20.0),
            const Text(
              'Hello!',
              style: TextStyle(
                color: Colors.black,
                fontSize: 24.0,
                fontWeight: FontWeight.bold,
              ),
            ),
            const Text(
              'Nirdesh',
              style: TextStyle(
                color: Colors.black,
                fontSize: 24.0,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(height: 200.0),
            Row(
              mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
              children: <Widget>[
                ElevatedButton(
                  style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                  ),
                  onPressed: () =>
Navigator.pushReplacement(
                      context,
                      MaterialPageRoute(
                          builder: (context) => const
Questions())),
                  child: const Padding(
                    padding: EdgeInsets.fromLTRB(30.0,
30.0, 30.0, 30.0),
                    child: Text(
                      'Take Test',
                      style: TextStyle(color:
Colors.white),
                    ),
                  ),
                ),
                const SizedBox(width: 60.0),
                ElevatedButton(
                  style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                  ),
                  onPressed: () =>
Navigator.pushReplacement(context,
                      MaterialPageRoute(builder: (context)
=> const Results())),
                  child: const Padding(
                    padding: EdgeInsets.fromLTRB(30.0,
30.0, 30.0, 30.0),
                    child: Text(
                      'History',
                      style: TextStyle(color:
Colors.white),
                    ),
                  ),
                ),
                const SizedBox(width: 10.0),
              ],
            ),
            const SizedBox(height: 40.0),
            Row(
              mainAxisAlignment:
MainAxisAlignment.spaceEvenly,
```



```
              children: <Widget>[
                ElevatedButton(
                  style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                  ),
                  onPressed: () {},
                  child: const Padding(
                    padding: EdgeInsets.fromLTRB(10.0,
30.0, 10.0, 30.0),
                    child: Text(
                      'Doctors',
                      style: TextStyle(color:
Colors.white),
                    ),
                  ),
                ),
                const SizedBox(width: 10.0),
                ElevatedButton(
                  style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                  ),
                  onPressed: () {},
                  child: const Padding(
                    padding: EdgeInsets.fromLTRB(10.0,
30.0, 10.0, 30.0),
                    child: Text(
                      'Medicines',
                      style: TextStyle(color:
Colors.white),
                    ),
                  ),
                ),
                const SizedBox(width: 10.0),
                ElevatedButton(
                  style: ElevatedButton.styleFrom(
                    primary: Colors.black,
                  ),
                  onPressed: () {},
                  child: const Padding(
                    padding: EdgeInsets.fromLTRB(10.0,
30.0, 10.0, 30.0),
                    child: Text(
```

```
                      'Emergency',
                      style: TextStyle(color:
Colors.white),
                    ),
                  ),
                ),
              ],
            ),
            const SizedBox(height: 20.0),
          ],
        ),
      ),
    );
  }
}
```

4. Analysis



Please enter the following data !

AGE 45

BP 80

SG 1.02

AL 1

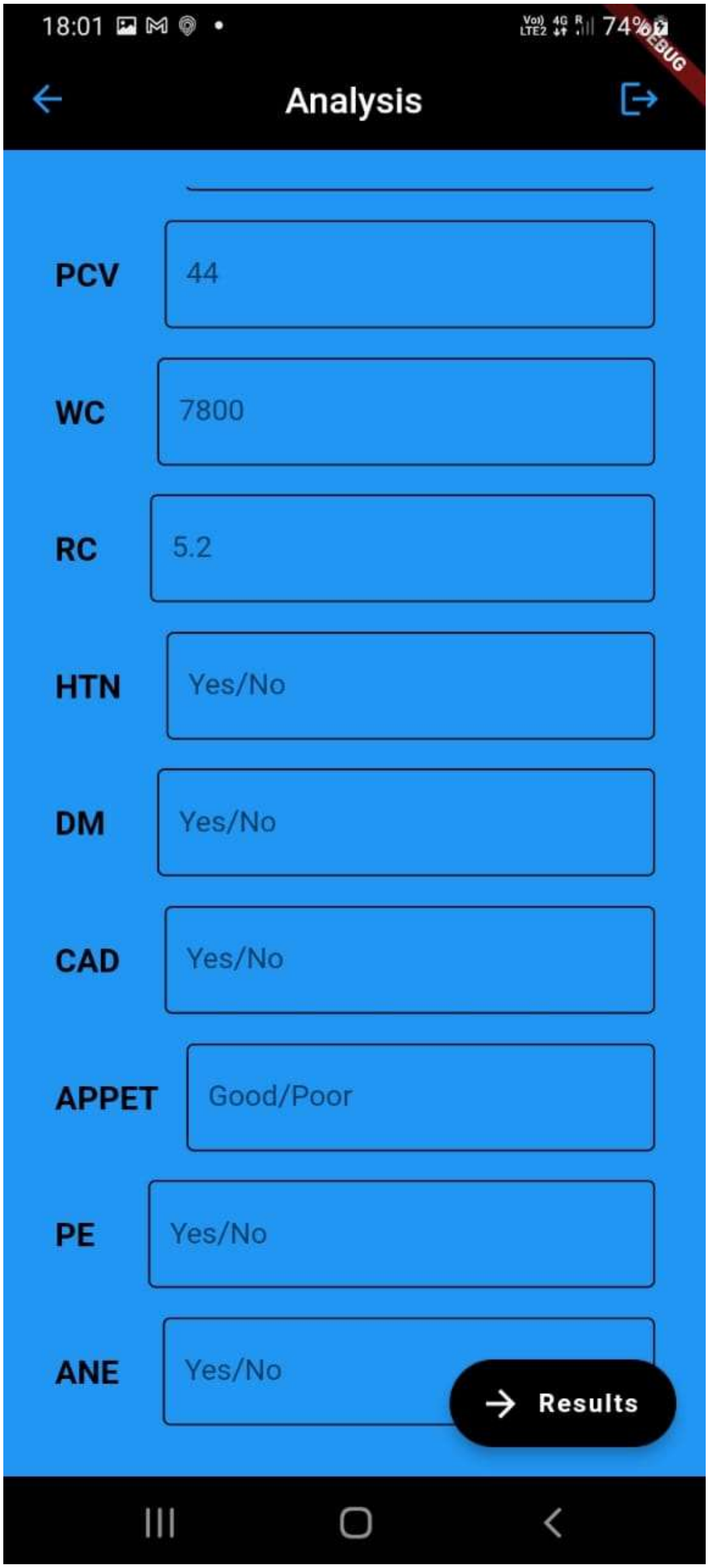
SU 3

RBC Normal/Abnormal

PC Normal/Abnormal

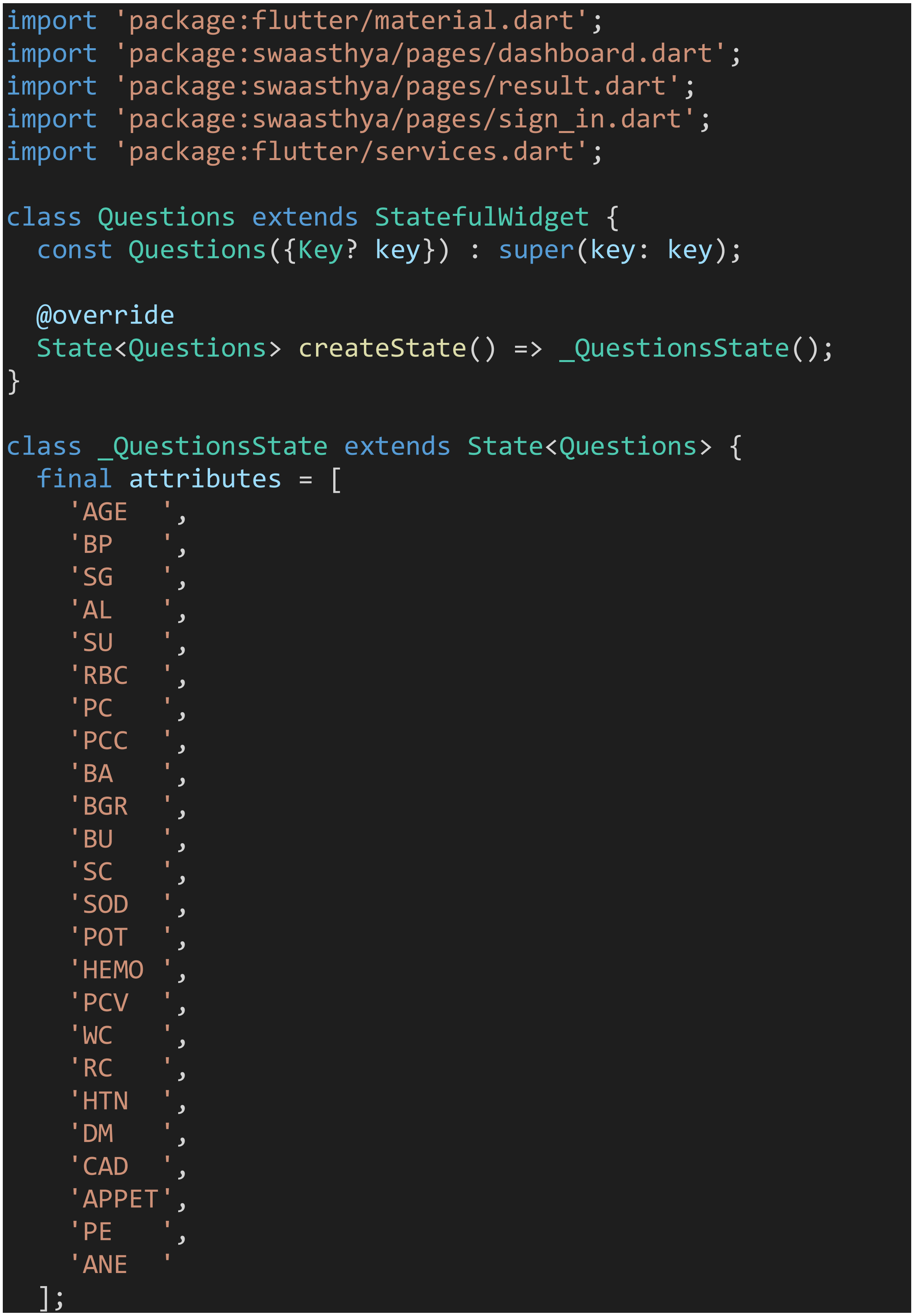
PCC Present/Not Present

BA Present/Not Present

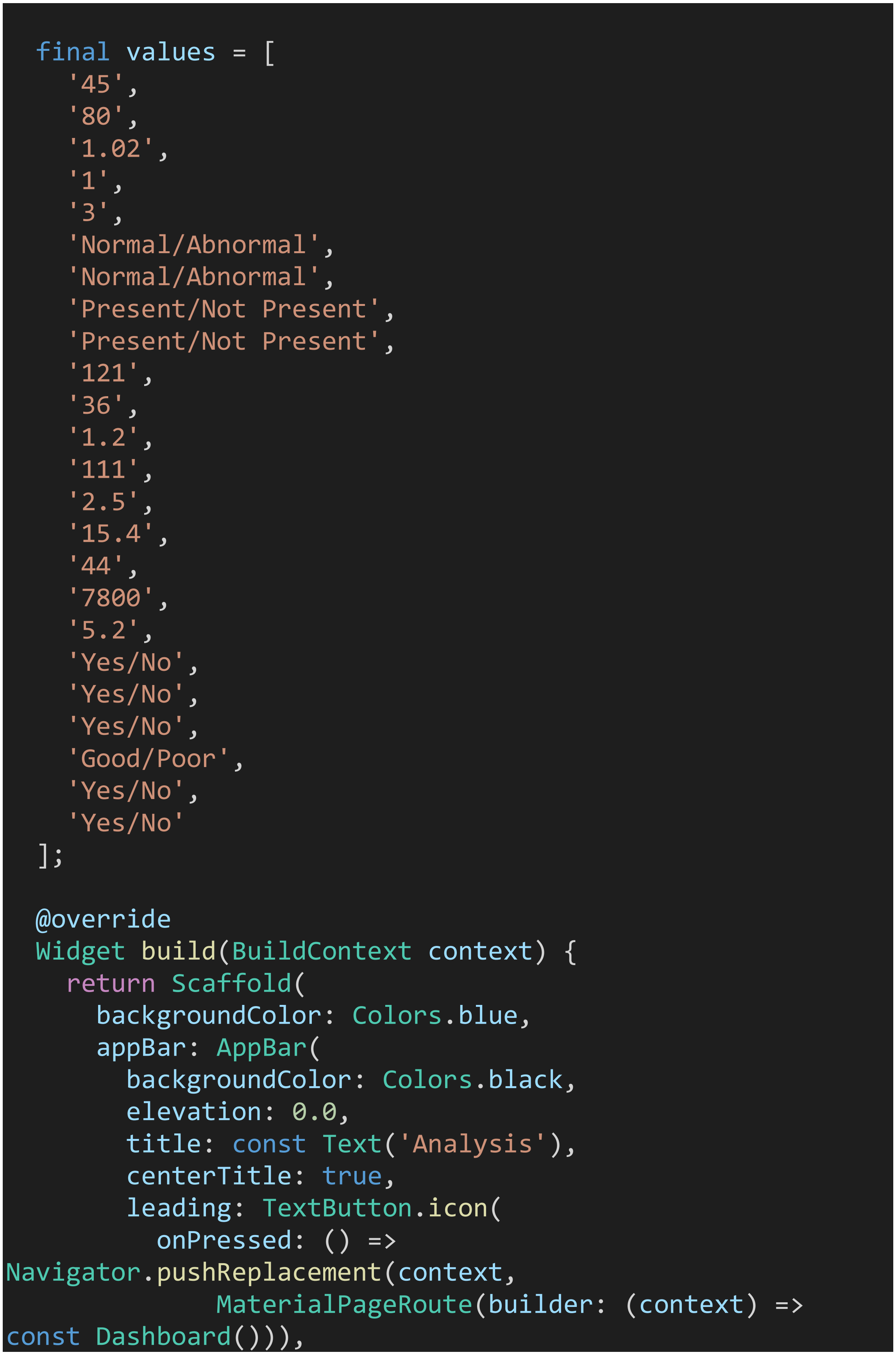


PCV	44
WC	7800
RC	5.2
HTN	Yes/No
DM	Yes/No
CAD	Yes/No
APPET	Good/Poor
PE	Yes/No
ANE	Yes/No

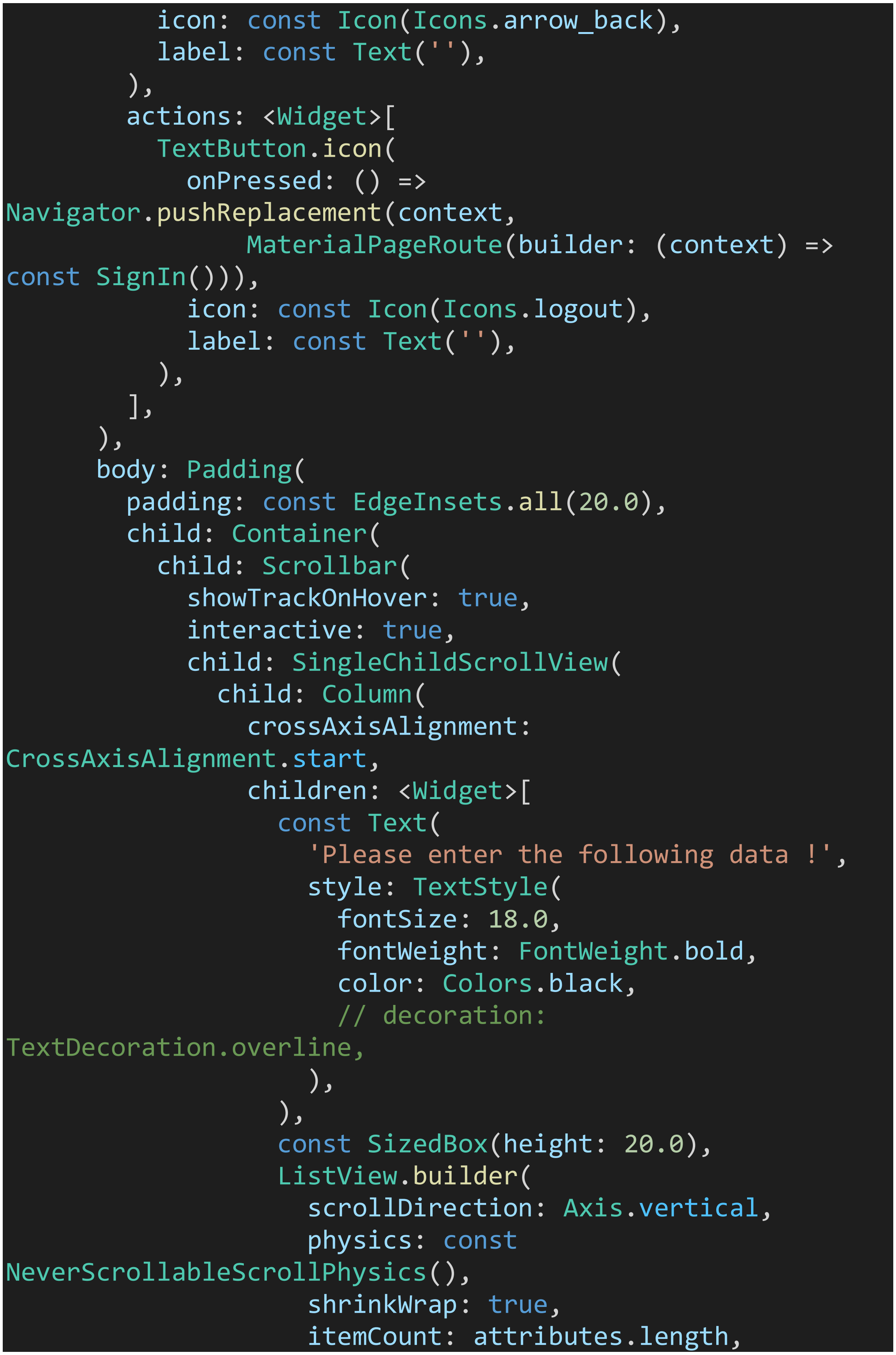
Code:



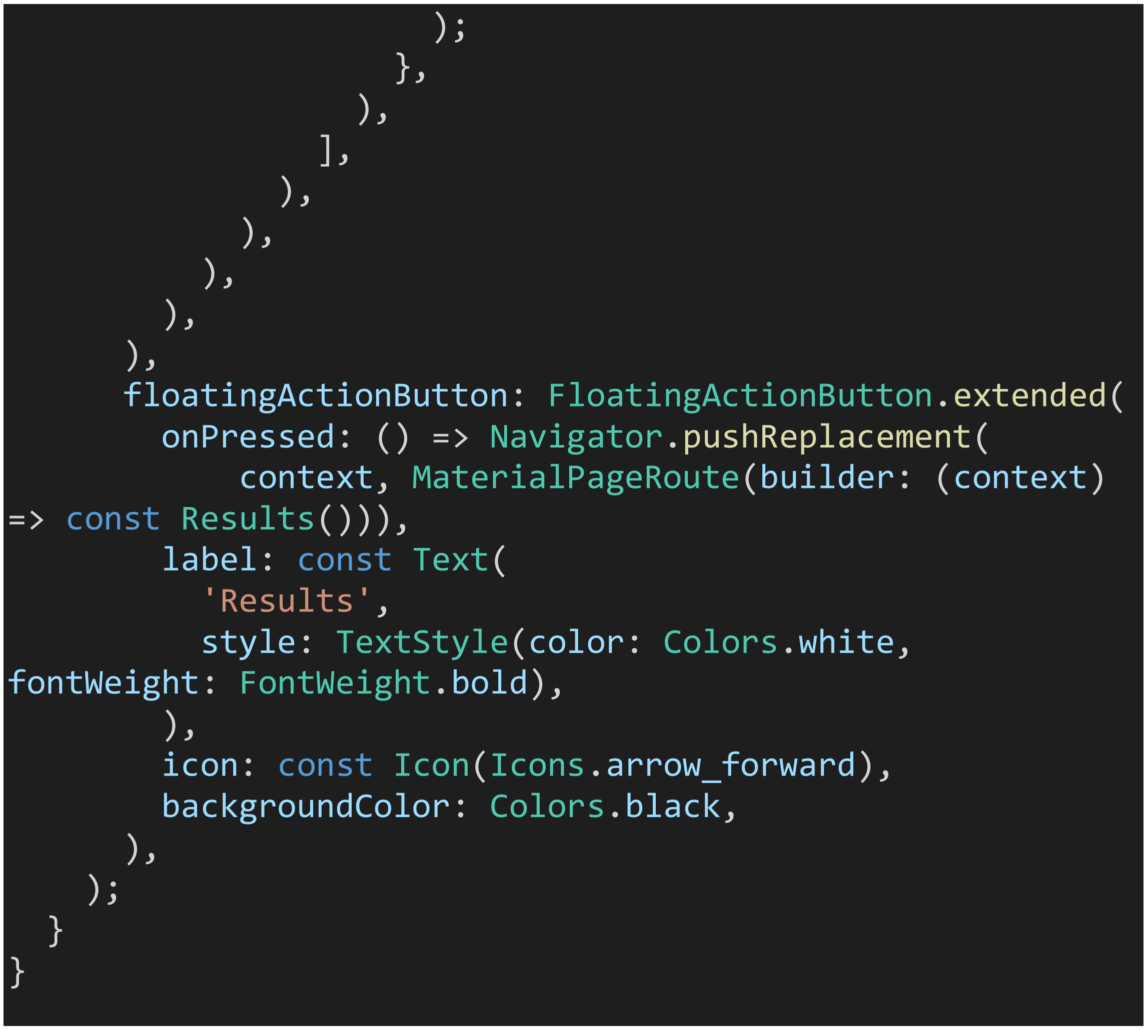
```dart
import 'package:flutter/material.dart';
import 'package:swaasthya/pages/dashboard.dart';
import 'package:swaasthya/pages/result.dart';
import 'package:swaasthya/pages/sign_in.dart';
import 'package:flutter/services.dart';

class Questions extends StatefulWidget {
  const Questions({Key? key}) : super(key: key);

  @override
  State<Questions> createState() => _QuestionsState();
}

class _QuestionsState extends State<Questions> {
  final attributes = [
    'AGE  ',
    'BP   ',
    'SG   ',
    'AL   ',
    'SU   ',
    'RBC  ',
    'PC   ',
    'PCC  ',
    'BA   ',
    'BGR  ',
    'BU   ',
    'SC   ',
    'SOD  ',
    'POT  ',
    'HEMO ',
    'PCV  ',
    'WC   ',
    'RC   ',
    'HTN  ',
    'DM   ',
    'CAD  ',
    'APPET',
    'PE   ',
    'ANE  '
  ];
```



```
  final values = [
    '45',
    '80',
    '1.02',
    '1',
    '3',
    'Normal/Abnormal',
    'Normal/Abnormal',
    'Present/Not Present',
    'Present/Not Present',
    '121',
    '36',
    '1.2',
    '111',
    '2.5',
    '15.4',
    '44',
    '7800',
    '5.2',
    'Yes/No',
    'Yes/No',
    'Yes/No',
    'Good/Poor',
    'Yes/No',
    'Yes/No'
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      appBar: AppBar(
        backgroundColor: Colors.black,
        elevation: 0.0,
        title: const Text('Analysis'),
        centerTitle: true,
        leading: TextButton.icon(
          onPressed: () =>
Navigator.pushReplacement(context,
              MaterialPageRoute(builder: (context) =>
const Dashboard())),
```

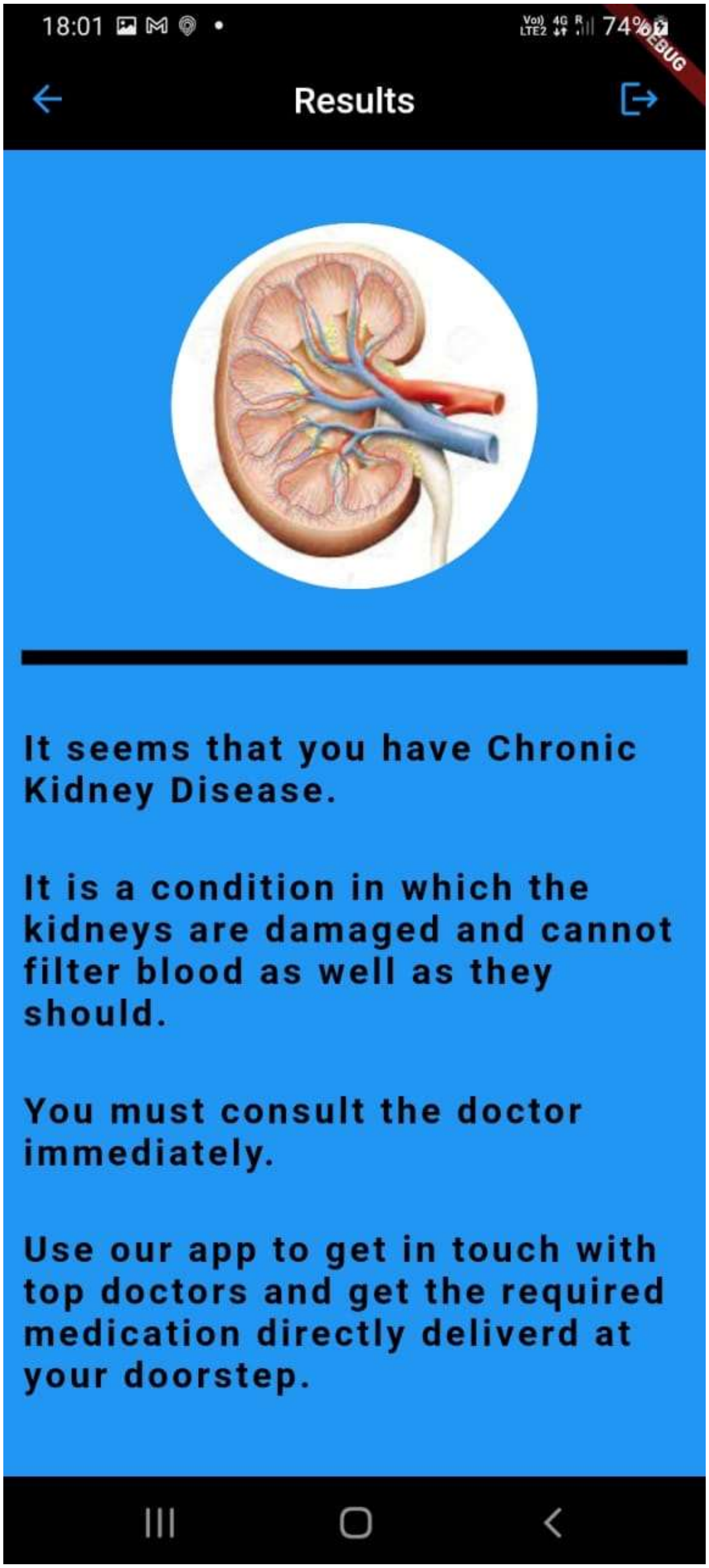


```
          icon: const Icon(Icons.arrow_back),
          label: const Text(''),
        ),
        actions: <Widget>[
          TextButton.icon(
            onPressed: () =>
Navigator.pushReplacement(context,
                MaterialPageRoute(builder: (context) =>
const SignIn())),
            icon: const Icon(Icons.logout),
            label: const Text(''),
          ),
        ],
      ),
      body: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Container(
          child: Scrollbar(
            showTrackOnHover: true,
            interactive: true,
            child: SingleChildScrollView(
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: <Widget>[
                  const Text(
                    'Please enter the following data !',
                    style: TextStyle(
                      fontSize: 18.0,
                      fontWeight: FontWeight.bold,
                      color: Colors.black,
                      // decoration:
TextDecoration.overline,
                    ),
                  ),
                  const SizedBox(height: 20.0),
                  ListView.builder(
                    scrollDirection: Axis.vertical,
                    physics: const
NeverScrollableScrollPhysics(),
                    shrinkWrap: true,
                    itemCount: attributes.length,
                    itemBuilder: (context, index) {
                      return Padding(
                        padding: const
EdgeInsets.all(8.0),
                        child: Row(
                          children: <Widget>[
                            Text(
                              attributes[index],
                              style: const TextStyle(
                                fontSize: 18.0,
                                fontWeight:
FontWeight.bold,
                                color: Colors.black,
                                // decoration:
TextDecoration.overline,
                              ),
                            ),
                            const SizedBox(width: 15.0),
                            Expanded(
                              child: TextFormField(
                                decoration:
InputDecoration(
                                  labelText:
values[index],
                                  enabledBorder: const
OutlineInputBorder(),
                                  focusedBorder: const
OutlineInputBorder(),
                                ),
                                // keyboardType: const
TextInputType.numberWithOptions(
                                //     decimal: true),
                                // inputFormatters:
<TextInputFormatter>[
                                //   FilteringTextInputFor
matter.digitsOnly
                                // ], // Only numbers can
be entered
                              ),
                            ),
                          ],
                        ),
```



```
                      );
                    },
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
      floatingActionButton: FloatingActionButton.extended(
        onPressed: () => Navigator.pushReplacement(
            context, MaterialPageRoute(builder: (context)
=> const Results())),
        label: const Text(
          'Results',
          style: TextStyle(color: Colors.white,
fontWeight: FontWeight.bold),
        ),
        icon: const Icon(Icons.arrow_forward),
        backgroundColor: Colors.black,
      ),
    );
  }
}
```

5. Results



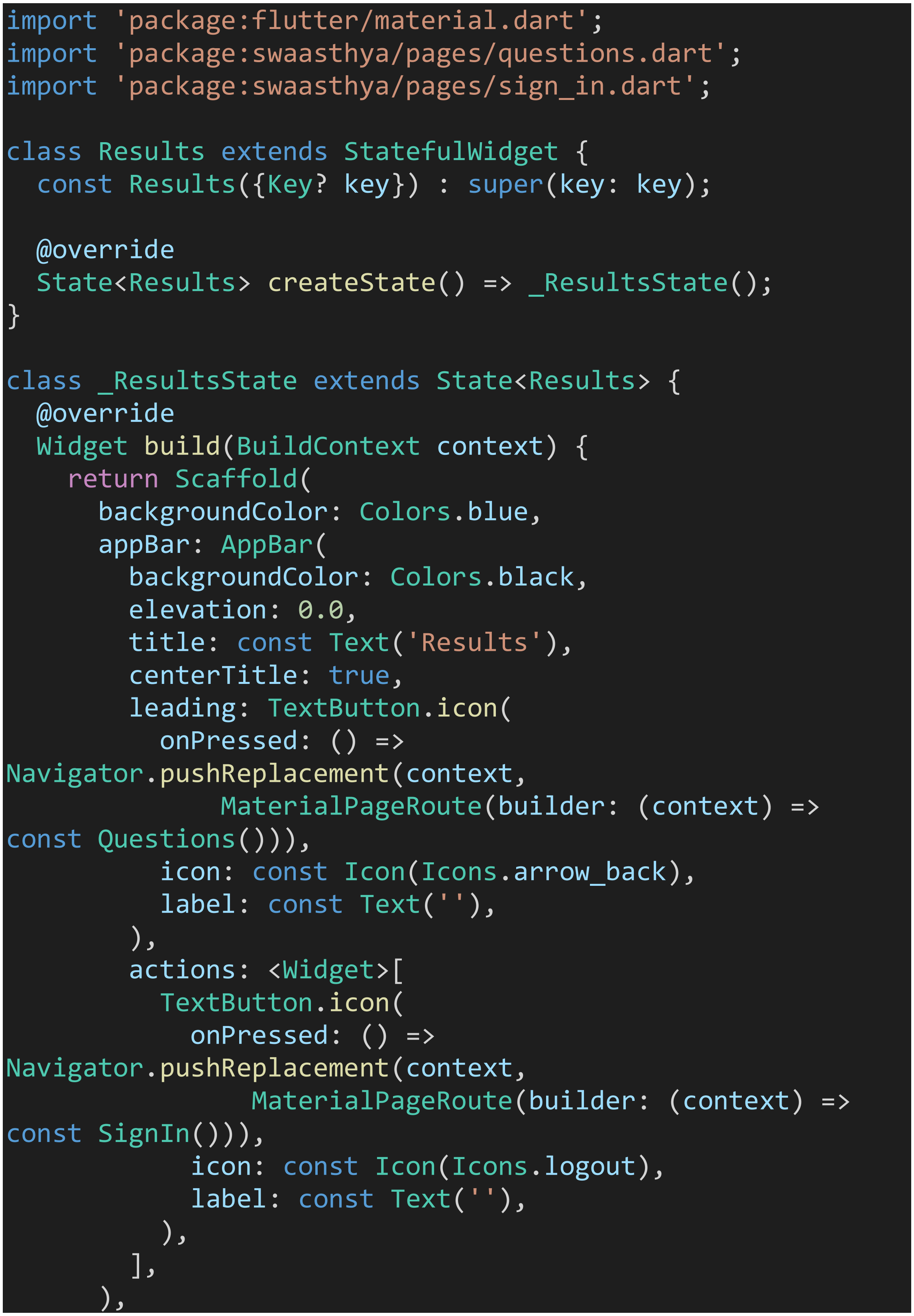
It seems that you have Chronic Kidney Disease.

It is a condition in which the kidneys are damaged and cannot filter blood as well as they should.

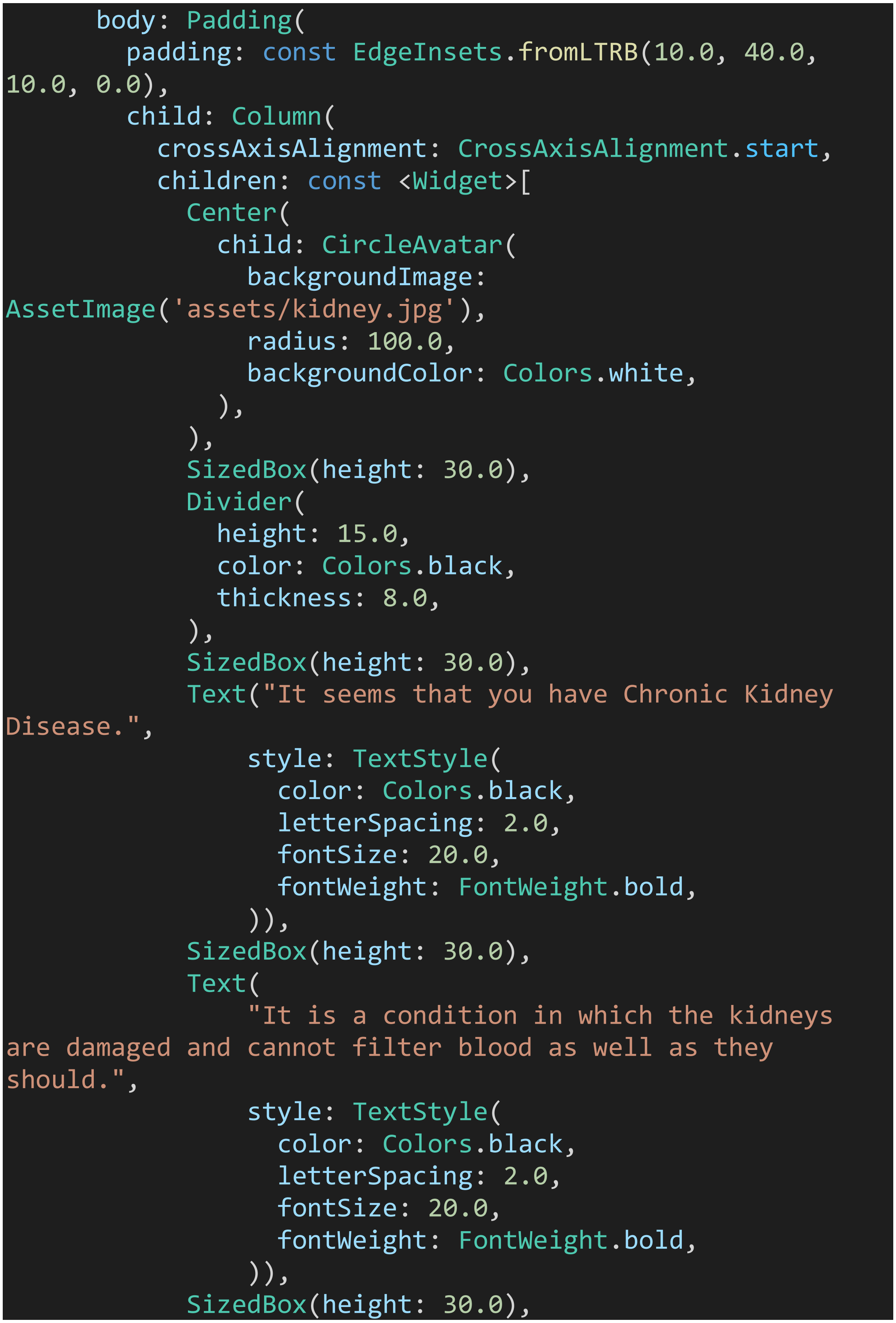
You must consult the doctor immediately.

Use our app to get in touch with top doctors and get the required medication directly deliverd at your doorstep.

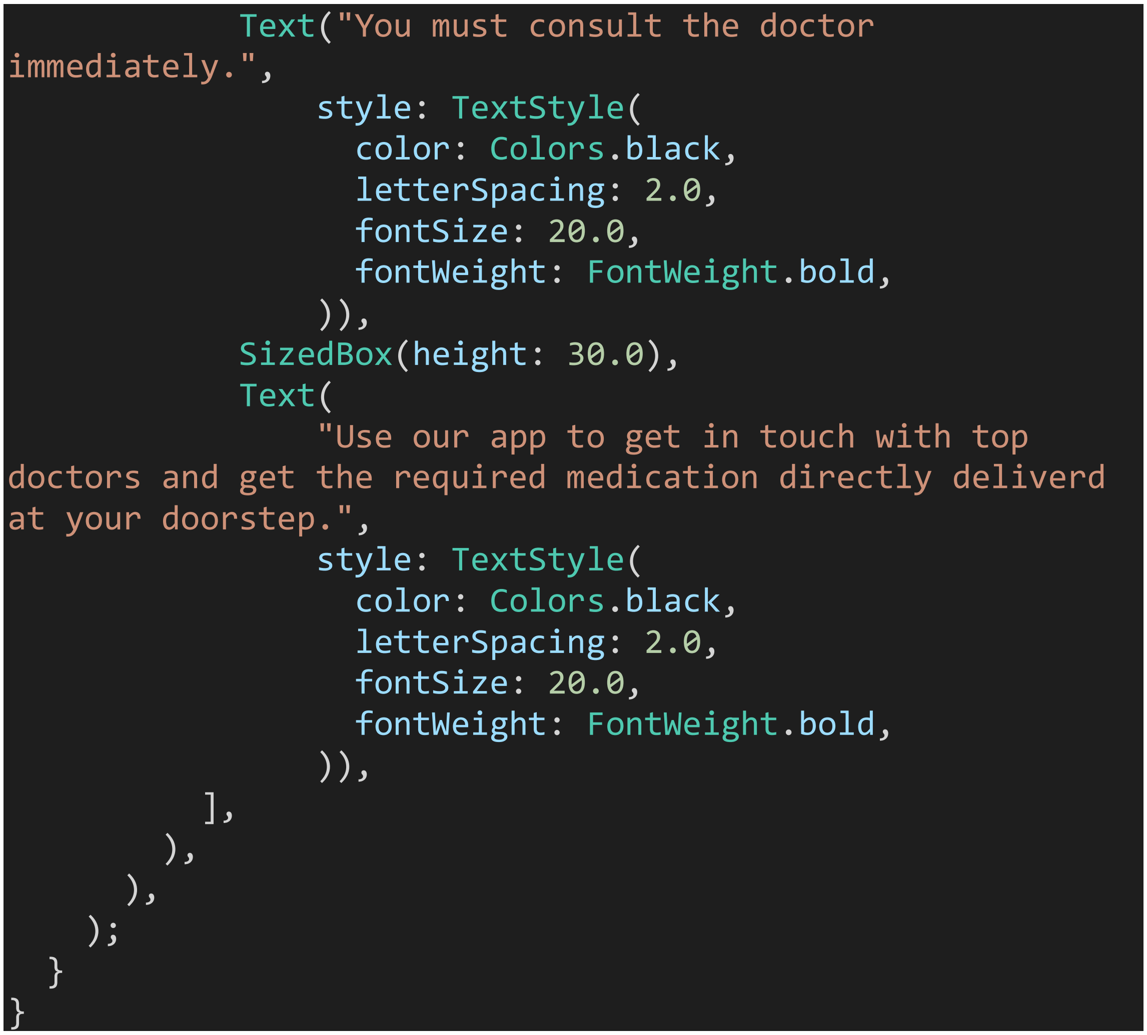
Code:



```
import 'package:flutter/material.dart';
import 'package:swaasthya/pages/questions.dart';
import 'package:swaasthya/pages/sign_in.dart';

class Results extends StatefulWidget {
  const Results({Key? key}) : super(key: key);

  @override
  State<Results> createState() => _ResultsState();
}

class _ResultsState extends State<Results> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue,
      appBar: AppBar(
        backgroundColor: Colors.black,
        elevation: 0.0,
        title: const Text('Results'),
        centerTitle: true,
        leading: TextButton.icon(
          onPressed: () =>
Navigator.pushReplacement(context,
              MaterialPageRoute(builder: (context) =>
const Questions())),
          icon: const Icon(Icons.arrow_back),
          label: const Text(''),
        ),
        actions: <Widget>[
          TextButton.icon(
            onPressed: () =>
Navigator.pushReplacement(context,
                MaterialPageRoute(builder: (context) =>
const SignIn())),
            icon: const Icon(Icons.logout),
            label: const Text(''),
          ),
        ],
      ),
```



```
      body: Padding(
        padding: const EdgeInsets.fromLTRB(10.0, 40.0,
10.0, 0.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: const <Widget>[
            Center(
              child: CircleAvatar(
                backgroundImage:
AssetImage('assets/kidney.jpg'),
                radius: 100.0,
                backgroundColor: Colors.white,
              ),
            ),
            SizedBox(height: 30.0),
            Divider(
              height: 15.0,
              color: Colors.black,
              thickness: 8.0,
            ),
            SizedBox(height: 30.0),
            Text("It seems that you have Chronic Kidney
Disease.",
                style: TextStyle(
                  color: Colors.black,
                  letterSpacing: 2.0,
                  fontSize: 20.0,
                  fontWeight: FontWeight.bold,
                )),
            SizedBox(height: 30.0),
            Text(
                "It is a condition in which the kidneys
are damaged and cannot filter blood as well as they
should.",
                style: TextStyle(
                  color: Colors.black,
                  letterSpacing: 2.0,
                  fontSize: 20.0,
                  fontWeight: FontWeight.bold,
                )),
            SizedBox(height: 30.0),
```



```
            Text("You must consult the doctor
immediately.",
                style: TextStyle(
                  color: Colors.black,
                  letterSpacing: 2.0,
                  fontSize: 20.0,
                  fontWeight: FontWeight.bold,
                )),
            SizedBox(height: 30.0),
            Text(
                "Use our app to get in touch with top
doctors and get the required medication directly deliverd
at your doorstep.",
                style: TextStyle(
                  color: Colors.black,
                  letterSpacing: 2.0,
                  fontSize: 20.0,
                  fontWeight: FontWeight.bold,
                )),
          ],
        ),
      ),
    );
  }
}
```

References

- https://drive.google.com/file/d/14nvAIWrup9dpI7dGpole8ELPzGkaiWar/view?usp=sharing
- https://docs.flutter.dev/
- https://www.tensorflow.org/
- https://www.karger.com/Article/FullText/504622
- https://www.journals.elsevier.com/advances-in-chronic-kidney-disease
- https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/#:~:text=The%20rectified%20linear%20activation%20function,otherwise%2C%20it%20will%20output%20zero.