Build Python Code:

Import the libraries

Import the libraries

app.py 7, M | Diabetic_Retinopathy.ipynb M

app.py > index

```
import numpy as np
import os
import tensorflow as tf
from tensorflow import keras
from keras.models import load_model
from keras.preprocessing import image
from keras.applications.inception_v3 import preprocess_input
from flask import Flask, request, flash, render_template, redirect, url_for
from cloudant.client import Cloudant
from twilio.rest import Client
```

Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

```
model = load_model(r"Updated-xception-diabetic-retinopathy.h5")
app = Flask(__name__)
app.secret_key = "abc"
```

Create a database using an initiated client.

```
# Create a database using an initialized client
my_database = client.create_database('my_database')
if my_database.exists():
    print("Database '{0}' successfully created.".format('my_db'))
# default home page or route

user = ""
```

Render HTML page:

Here we will be using a declared constructor to route to the HTML page which we have created earlier.

In the above example, '/' URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the html page will be rendered. Whenever you enter the values from the html page the values can be retrieved using POST Method.

Configure the registration page

Based on user input into the registration form we stored it on data dictionary then we can validate the data using _id parameter with user input that we can store it on query variable then we can validate by passing the query variable into the my_database.get_user_result() method.Then we can check the docs length by using len(docs.all()) function.If the length of docs is 0 then user will register successfully on the platform and user data will store on the database.Otherwise its shows the message as user already registered please login and use our web application for DR prediction.

Configure the login page

Based on user input into the login form we stored user id and password into the (user,passw) variables. Then we can validate the credentials using _id parameter with user input that we can store it on query variable then we can validate by passing the query variable into the my_database.get_user_result() method.Then we can check the docs length by using len(docs.all()) function.If the length of doc is 0 then it means username is not found.Otherwise its validate the data that is stored on the database and check the username & password. If it's matched then the user will be able to login and use our web application for DR prediction.Otherwise the user needs to provide correct credentials.

For logout from web application.

Showcasing prediction on UI:

The image is selected from uploads folder. Image is loaded and resized with load_img() method. To convert image to an array, img_to_array() method is used and dimensions are increased with expand_dims() method. Input is processed for xception model and predict() method is used to predict the probability of classes. To find the max probability np.argmax is used.

For code : Refer github repo