MINI PROJECT

(2020-2021)

Dr Dev

MID TERM REPORT

Institute of Engineering and Technology

Submitted By :-
Utkarsh Rastogi (181500770)
Sec- K

Praful Sharma (181500463)
Sec- C

Supervised By :-
Mr. Akash Kumar Choudhary
Technical Trainer
Department of Computer Engineering & Applications

Contents

Abstract

The doctor-population ratio in India is 1:1456 against the WHO recommendation of 1:1000. Assuming an 80-per cent availability, it was estimated that around 9.27 lakh doctors were available for active service. Besides, there were 7.88 lakh Ayurveda, Unani and homeopathy (AUH) doctors in the country and assuming an 80-per cent availability, it was estimated that around 6.30 lakh doctors practising traditional systems of medicine were available for service and considered together with allopathic doctors, it threw up a doctor-patient ratio of 1:860. With a country with limited number of doctors, weak medical facilities is a challenge in India more specially in the rural India which is home to the 70% of population. Issues like improper medication, diagnosis error, late diagnosis has caused approx 5 million deaths in 2018 in India itself.

Our project is to implement a solution to connect the rural population with better medical facilities. These facilities include Artificial neural network based medical image classifiers that can predict a disease like X-rays of pneumonia etc witch an accuracy of more than 90%. Thus removing diagnosis errors and improving diagnosis resulting in quicker decision making and better medication. Moreover it can help in reducing doctor and patient reach in India. The second part of the project includes, connecting the patient with a doctor across the country via internet. The patient will be able to book an appointment and after approval from the doctor the appointment is scheduled. With this connectivity feature a patient will be able to connect with a doctor without any need to travel, not in the case of emergency or follow up meetings can take place after a visit.

Artificial neural networks provide a powerful tool to help doctors to analyse, model and make sense of complex clinical data across a broad range of medical applications. Most applications of artificial neural networks to medicine are classification problems; that is, the task is on the basis of the measured features to assign the patient to one of a small set of classes. The labelled dataset is prepared with targeted area and is given as feed where an array of each image is generated. The generated results are forwarded for mean square error and model evaluation is done using cross validation. After successful evaluation a hyper parameter tuning is done optimal hyper parameter for the learning algorithms are decided. The values of node weights are learned by the model. These hyper parameter are used to control the learning process within the targeted area. It helps in maximising the model performance. Final evaluation is done and on success the model is saved.

Introduction

1.1 ***General Introduction to the topic***

An artificial neural network (ANN) is the piece of a computing system designed to simulate the way the human brain analyzes and processes information. It is the foundation of artificial intelligence (AI) and solves problems that would prove impossible or difficult by human or statistical standards. It is is based on a collection of connected units or nodes called artificial neurons, which loosely model the neutrons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

Neural networks learn (or are trained) by processing examples, each of which contains a known "input" and "result," forming probability-weighted associations between the two, which are stored within the data structure of the net itself. The training of a neural network from a given example is usually conducted by determining the difference between the processed output of the network (often a prediction) and a target output. This is the error. The network then adjusts its weighted associations according to a learning rule and using this error value. Successive adjustments will cause the neural network to produce output which is increasingly similar to the target output. After a sufficient number of these adjustments the training can be terminated based upon certain criteria. This is known as supervised learning.

Such systems "learn" to perform tasks by considering examples, generally without being programmed with task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge of cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the examples that they process.

1.2 ***Convolutional Neural Network***

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery.[1] They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.[2][3] They have applications in image and video recognition, recommender systems,[4] image classification, medical image analysis, natural language processing,[5] and financial time series.[6]

CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme.

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

1.3 ***Hyper Parameters for CNN***

Hyper Parameters are use to maximise the efficiency of the model, learning rates and regularisation is followed. They are dependent on number of filters, filter shape and max pooling shape.

1.4 ***Area of Computer Science***

CNN is widely implemented in application to improve the overall efficiency. Since CNN is based upon transfer learning, during classification, it is given preference over others. The following are the applications of CNN:

Image Recognition
Electromyography Recognition
Video Analysis
Natural Language Processing
Anomaly Detection
Checkers game
Time Series forecasting
Cultural Heritage and 3d Dataset
Health Risk Assessment
Biomarkers of aging Discovery

1.5 ***Fine-Tuning***

For many applications, the training data is less available. Convolutional neural networks usually require a large amount of training data in order to avoid overfitting. A common technique is to train the network on a larger data set from a related domain. Once the network parameters have converged an additional training step is performed using the in-domain data to fine-tune the network weights. This allows convolutional networks to be successfully applied to problems with small training sets.

1.6 ***Hardware and Software Requirements***

Hardware Requirements:

- Memory [4 GB Ram (or higher)]
- Intel core i3 64 bit-processor or higher

Software Requirements:

- Heroku
- Flask
- Django
- Any OS
- Dataset (primarily from Kaggle, Google Public Datasets, UCI machine learning repository)
- Google Colab
- Anaconda
- Text-editor
- Stripe for payment
- ML libraries (numpy, panda, pandas, keras, tensor flow, matplotlib)

Problem Definition

Year 2016, noted 5.2 million deaths in India due to medical errors. Similarly the British Medical Journal quoted that India like any other developing country is recording a lot of medical errors. These errors were wrong diagnosis of disease, poor medical facilities, quality assurance, inadequate outlay for health, expensive healthcare, wrong consultations and no timely access or shortage of doctors in urban areas. Even in the present, due to Covid-19 many hospitals have scaled back or postponed non-emergency care that has affected a lot of patients around the globe, more specifically countries that are developing or countries with poor medical healthcare.

According to a survey, There is one government doctor for every 10,189 people in India, whereas the WHO recommendation is 1:1000. Although six states in India like Delhi, Kerala, Karnataka, Tamil Nadu, Punjab and Goa have more doctors than the WHO norm, it is a highly imbalanced picture and most of them are unwilling to move to Bihar or Uttar Pradesh (UP), the states that suffer from an acute shortage of doctors. This issue has led to different errors mentioned above and this is the most significant cause. Finding a solution to this can solve a lot of problems.

Since the magnitude of this challenge is significant, this cannot be resolved by the government alone.
This has been a serious issue and an immediate action to improve this has to be taken.

Objectives

Part 1: To build a CNN model to analyse an X-ray of Disease (here pneumonia)

We first initialise ImageNet for our transfer learning and remove the upper and the bottom layers completely so we are left with learnings of the ImageNet and transfer it to a new model.

We decide the feature importance, load the Kaggle dataset.

Now we feed the new model with the Pneumonia dataset with average pooling 2d. Setting up the hyper parameters to 1024 and activating the rectified linear unit and ` targeting size with 224 x 224, Here targeting size represents the square where the model has to learn from in the given picture starting from the centre.

We decide the epoch to 5 depends on the dataset, that further divides the dataset into 5 parts for enhanced learning. And we update the model with the new dataset for classification of pneumonia. Since CNN is focused on particular category at a time it is advised to train different models for different diseases for higher accuracy and reduce loss during training.

Cross validation and model evaluation is done and evaluation on final test data is done.

We take the input from the user process it via flask supported by backend in python and generate the result.

Part 2: It compromises of login and signup functionality for different users (here doctor and patient).

For Doctor and Patient we are using postgresql for database and Django as framework.

For frontend we are using HTML, CSS and for backend we are using python.

Part 3: It compromises of booking an appointment with the doctor.

The patient from his portal request for an appointment with a doctor and after approval from doctor the appointment is scheduled.

For video calling we are integrating zoom. For payment of fees we are using stripe and unified payment interface.

Part 4: Deployment on Heroku.

The final part consist of Deployment on Heroku. It is a cloud platform as a service which supports Java, Python, Ruby, PHP and Go.

It is further noted that the datasets are labelled and are verified before turning it to the model for training. The patient data is saved with his concern and can be avoided is he refuses too.

Progress

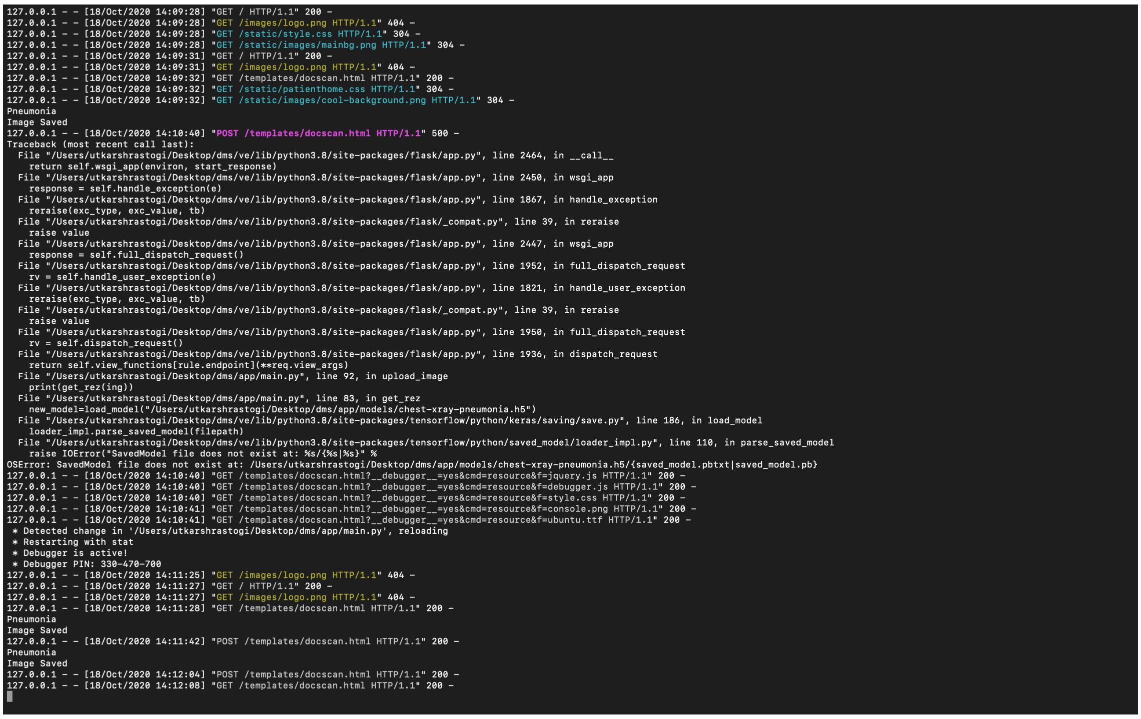
Part 1 is completed. The model trained with pneumonia along with the filters are set. Further models for diseases are in the go. Labelling for the data is completed and prediction for pneumonia is successful with high accuracy. Virtual Environment with flask, keras and tensor flow libraries is created.

Part 2 is completed. Home page, login and signup functionalities are ready. Different logins for patient and doctor, different sign up for the same are complete.

Part 3 is in progress. Integrating Zoom and Stripe and UPI for payment functionality is to be added.

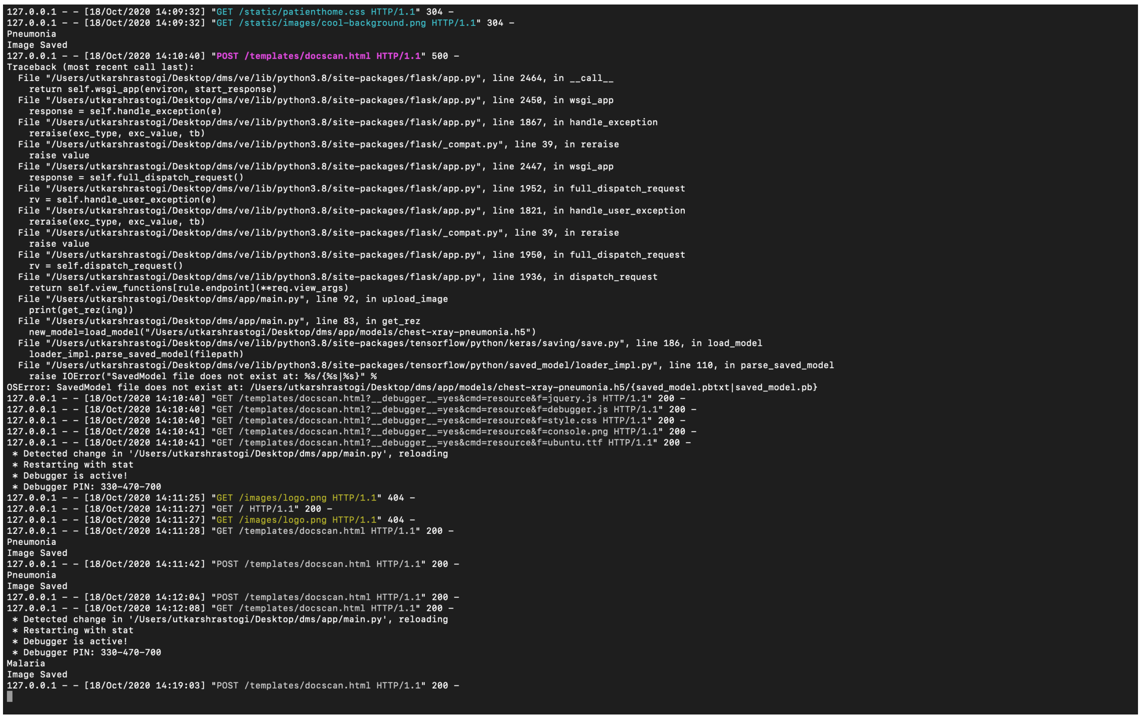
Part 4 will be deployed after the first 3 are completed.

Screenshots



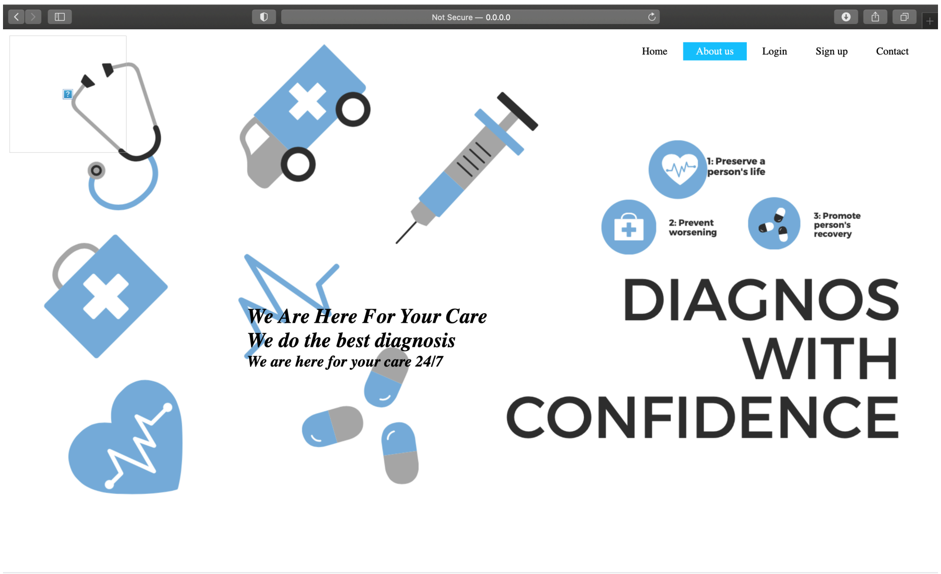
```
127.0.0.1 - - [18/Oct/2020 14:09:28] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:09:28] "GET /images/logo.png HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2020 14:09:28] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [18/Oct/2020 14:09:28] "GET /static/images/mainbg.png HTTP/1.1" 304 -
127.0.0.1 - - [18/Oct/2020 14:09:31] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:09:31] "GET /images/logo.png HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2020 14:09:32] "GET /templates/docscan.html HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:09:32] "GET /static/patienthome.css HTTP/1.1" 304 -
127.0.0.1 - - [18/Oct/2020 14:09:32] "GET /static/images/cool-background.png HTTP/1.1" 304 -
Pneumonia
Image Saved
127.0.0.1 - - [18/Oct/2020 14:10:40] "POST /templates/docscan.html HTTP/1.1" 500 -
Traceback (most recent call last):
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 2464, in __call__
    return self.wsgi_app(environ, start_response)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 2450, in wsgi_app
    response = self.handle_exception(e)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1867, in handle_exception
    reraise(exc_type, exc_value, tb)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 2447, in wsgi_app
    response = self.full_dispatch_request()
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1952, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1821, in handle_user_exception
    reraise(exc_type, exc_value, tb)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1950, in full_dispatch_request
    rv = self.dispatch_request()
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1936, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/Users/utkarshrastogi/Desktop/dms/app/main.py", line 92, in upload_image
    print(get_rez(ing))
  File "/Users/utkarshrastogi/Desktop/dms/app/main.py", line 83, in get_rez
    new_model=load_model("/Users/utkarshrastogi/Desktop/dms/app/models/chest-xray-pneumonia.h5")
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/tensorflow/python/keras/saving/save.py", line 186, in load_model
    loader_impl.parse_saved_model(filepath)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/tensorflow/python/saved_model/loader_impl.py", line 110, in parse_saved_model
    raise IOError("SavedModel file does not exist at: %s/{%s|%s}" %
OSError: SavedModel file does not exist at: /Users/utkarshrastogi/Desktop/dms/app/models/chest-xray-pneumonia.h5/{saved_model.pbtxt|saved_model.pb}
127.0.0.1 - - [18/Oct/2020 14:10:40] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=jquery.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:40] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=debugger.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:40] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=style.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:41] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=console.png HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:41] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=ubuntu.ttf HTTP/1.1" 200 -
 * Detected change in '/Users/utkarshrastogi/Desktop/dms/app/main.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 330-470-700
127.0.0.1 - - [18/Oct/2020 14:11:25] "GET /images/logo.png HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2020 14:11:27] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:11:27] "GET /images/logo.png HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2020 14:11:28] "GET /templates/docscan.html HTTP/1.1" 200 -
Pneumonia
Image Saved
127.0.0.1 - - [18/Oct/2020 14:11:42] "POST /templates/docscan.html HTTP/1.1" 200 -
Pneumonia
Image Saved
127.0.0.1 - - [18/Oct/2020 14:12:04] "POST /templates/docscan.html HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:12:08] "GET /templates/docscan.html HTTP/1.1" 200 -
```

Pneumonia



```
127.0.0.1 - - [18/Oct/2020 14:09:32] "GET /static/patienthome.css HTTP/1.1" 304 -
127.0.0.1 - - [18/Oct/2020 14:09:32] "GET /static/images/cool-background.png HTTP/1.1" 304 -
Pneumonia
Image Saved
127.0.0.1 - - [18/Oct/2020 14:10:40] "POST /templates/docscan.html HTTP/1.1" 500 -
Traceback (most recent call last):
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 2464, in __call__
    return self.wsgi_app(environ, start_response)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 2450, in wsgi_app
    response = self.handle_exception(e)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1867, in handle_exception
    reraise(exc_type, exc_value, tb)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 2447, in wsgi_app
    response = self.full_dispatch_request()
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1952, in full_dispatch_request
    rv = self.handle_user_exception(e)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1821, in handle_user_exception
    reraise(exc_type, exc_value, tb)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/_compat.py", line 39, in reraise
    raise value
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1950, in full_dispatch_request
    rv = self.dispatch_request()
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/flask/app.py", line 1936, in dispatch_request
    return self.view_functions[rule.endpoint](**req.view_args)
  File "/Users/utkarshrastogi/Desktop/dms/app/main.py", line 92, in upload_image
    print(get_rez(ing))
  File "/Users/utkarshrastogi/Desktop/dms/app/main.py", line 83, in get_rez
    new_model=load_model("/Users/utkarshrastogi/Desktop/dms/app/models/chest-xray-pneumonia.h5")
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/tensorflow/python/keras/saving/save.py", line 186, in load_model
    loader_impl.parse_saved_model(filepath)
  File "/Users/utkarshrastogi/Desktop/dms/ve/lib/python3.8/site-packages/tensorflow/python/saved_model/loader_impl.py", line 110, in parse_saved_model
    raise IOError("SavedModel file does not exist at: %s/{%s|%s}" %
OSError: SavedModel file does not exist at: /Users/utkarshrastogi/Desktop/dms/app/models/chest-xray-pneumonia.h5/{saved_model.pbtxt|saved_model.pb}
127.0.0.1 - - [18/Oct/2020 14:10:40] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=jquery.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:40] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=debugger.js HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:40] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=style.css HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:41] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=console.png HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:10:41] "GET /templates/docscan.html?__debugger__=yes&cmd=resource&f=ubuntu.ttf HTTP/1.1" 200 -
 * Detected change in '/Users/utkarshrastogi/Desktop/dms/app/main.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 330-470-700
127.0.0.1 - - [18/Oct/2020 14:11:25] "GET /images/logo.png HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2020 14:11:27] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:11:27] "GET /images/logo.png HTTP/1.1" 404 -
127.0.0.1 - - [18/Oct/2020 14:11:28] "GET /templates/docscan.html HTTP/1.1" 200 -
Pneumonia
Image Saved
127.0.0.1 - - [18/Oct/2020 14:11:42] "POST /templates/docscan.html HTTP/1.1" 200 -
Pneumonia
Image Saved
127.0.0.1 - - [18/Oct/2020 14:12:04] "POST /templates/docscan.html HTTP/1.1" 200 -
127.0.0.1 - - [18/Oct/2020 14:12:08] "GET /templates/docscan.html HTTP/1.1" 200 -
 * Detected change in '/Users/utkarshrastogi/Desktop/dms/app/main.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 330-470-700
Malaria
Image Saved
127.0.0.1 - - [18/Oct/2020 14:19:03] "POST /templates/docscan.html HTTP/1.1" 200 -
```

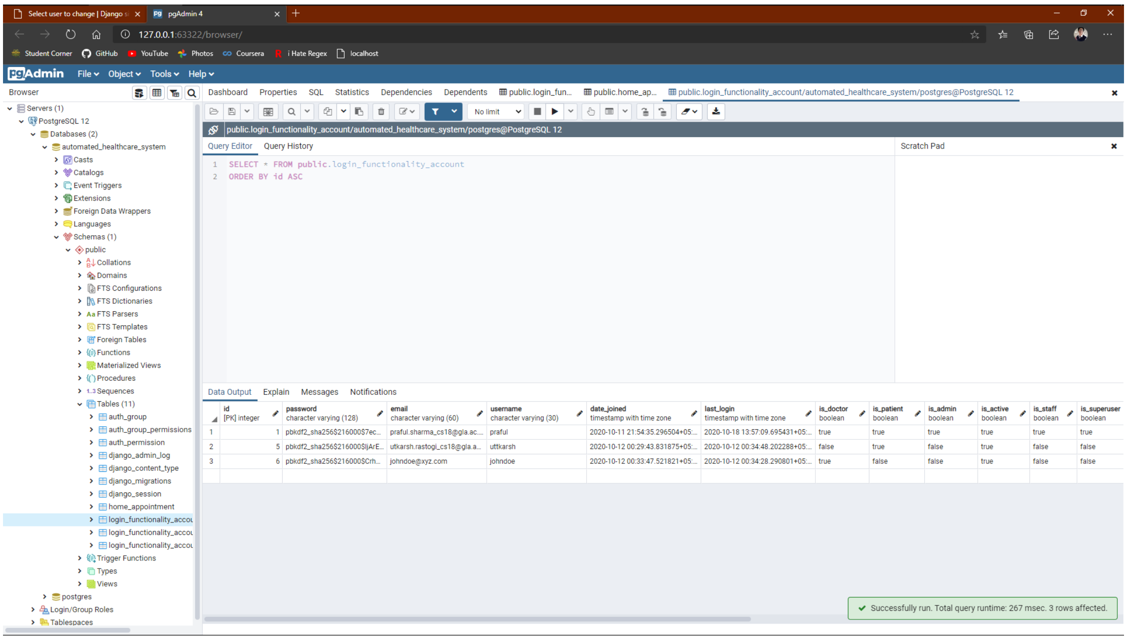
Malaria



We Are Here For Your Care
We do the best diagnosis
We are here for your care 24/7

DIAGNOS
WITH
CONFIDENCE

Landing Page[1]



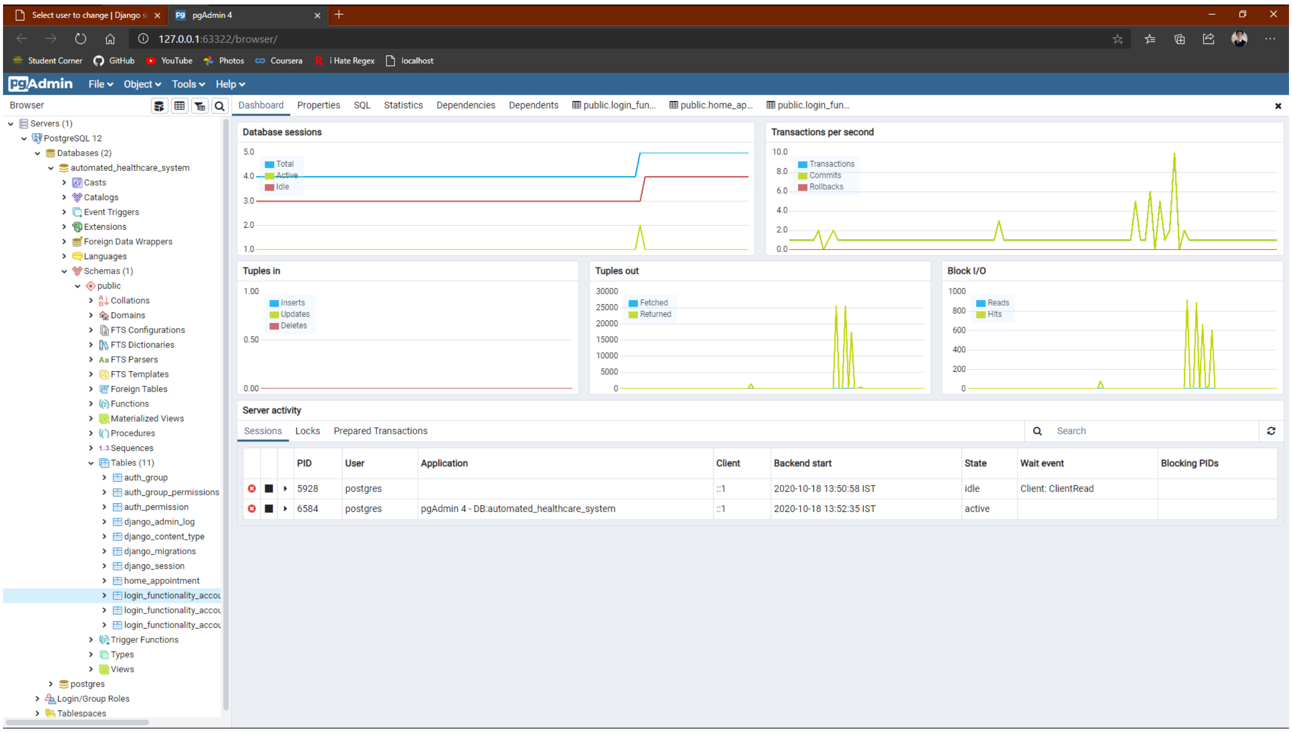
public.login_functionality_account/automated_healthcare_system/postgres@PostgreSQL 12

```
SELECT * FROM public.login_functionality_account
ORDER BY id ASC
```

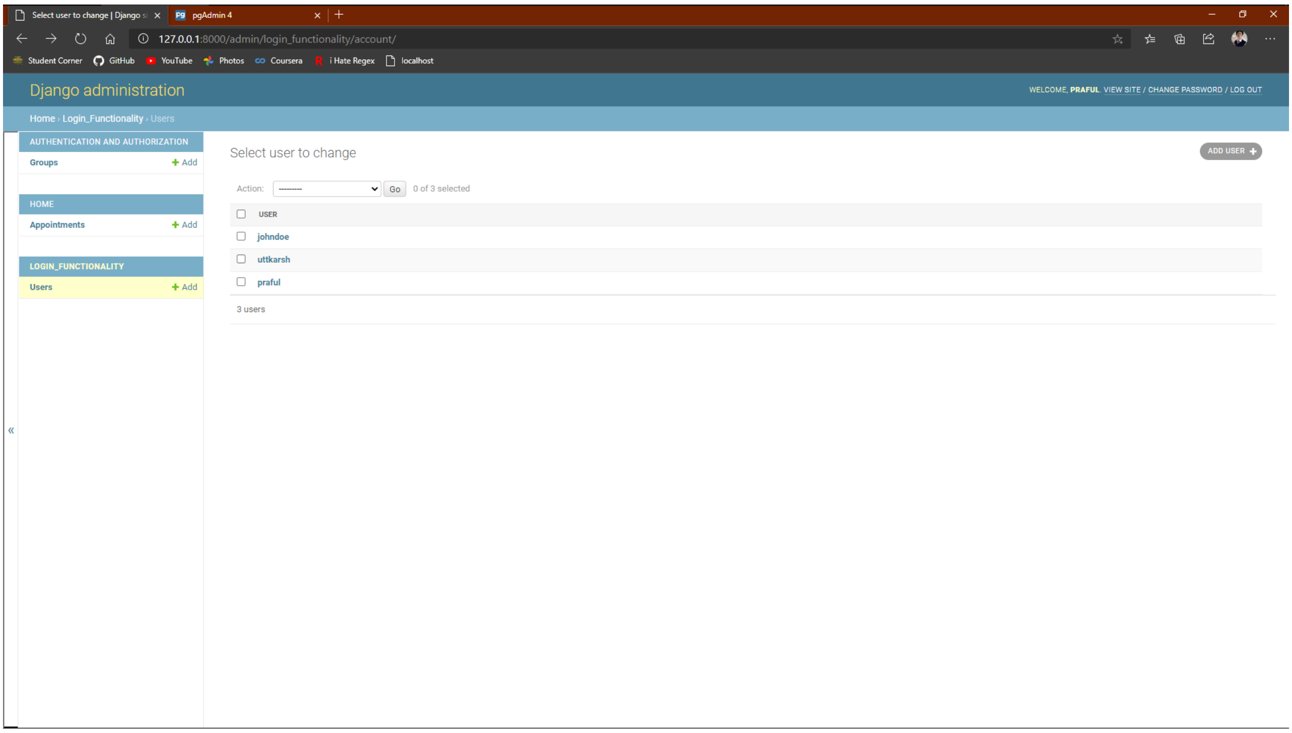
	id [PK] integer	password character varying (128)	email character varying (60)	username character varying (30)	date_joined timestamp with time zone	last_login timestamp with time zone	is_doctor boolean	is_patient boolean	is_admin boolean	is_active boolean	is_staff boolean	is_superuser boolean
1	1	pbkdf2_sha256$216000$7ec...	praful.sharma_cs18@gla.ac....	praful	2020-10-11 21:54:35.296504+05:...	2020-10-18 13:57:09.695431+05:...	true	true	true	true	true	true
2	5	pbkdf2_sha256$216000$IjArE...	utkarsh.rastogi_cs18@gla.a...	uttkarsh	2020-10-12 00:29:43.831875+05:...	2020-10-12 00:34:48.202288+05:...	false	true	false	true	false	false
3	6	pbkdf2_sha256$216000$Crh...	johndoe@xyz.com	johndoe	2020-10-12 00:33:47.521821+05:...	2020-10-12 00:34:28.290801+05:...	true	false	false	true	false	false

✓ Successfully run. Total query runtime: 267 msec. 3 rows affected.

[1] * The difference in the interfaces is due to remote development and functionalities are independent, therefore the work is divided in two parts and is assigned to us respectively. We will merge them in the later progress of the project before the final submission.



			PID	User	Application	Client	Backend start	State	Wait event	Blocking PIDs
⊗	■	▸	5928	postgres		::1	2020-10-18 13:50:58 IST	idle	Client: ClientRead	
⊗	■	▸	6584	postgres	pgAdmin 4 - DB:automated_healthcare_system	::1	2020-10-18 13:52:35 IST	active		



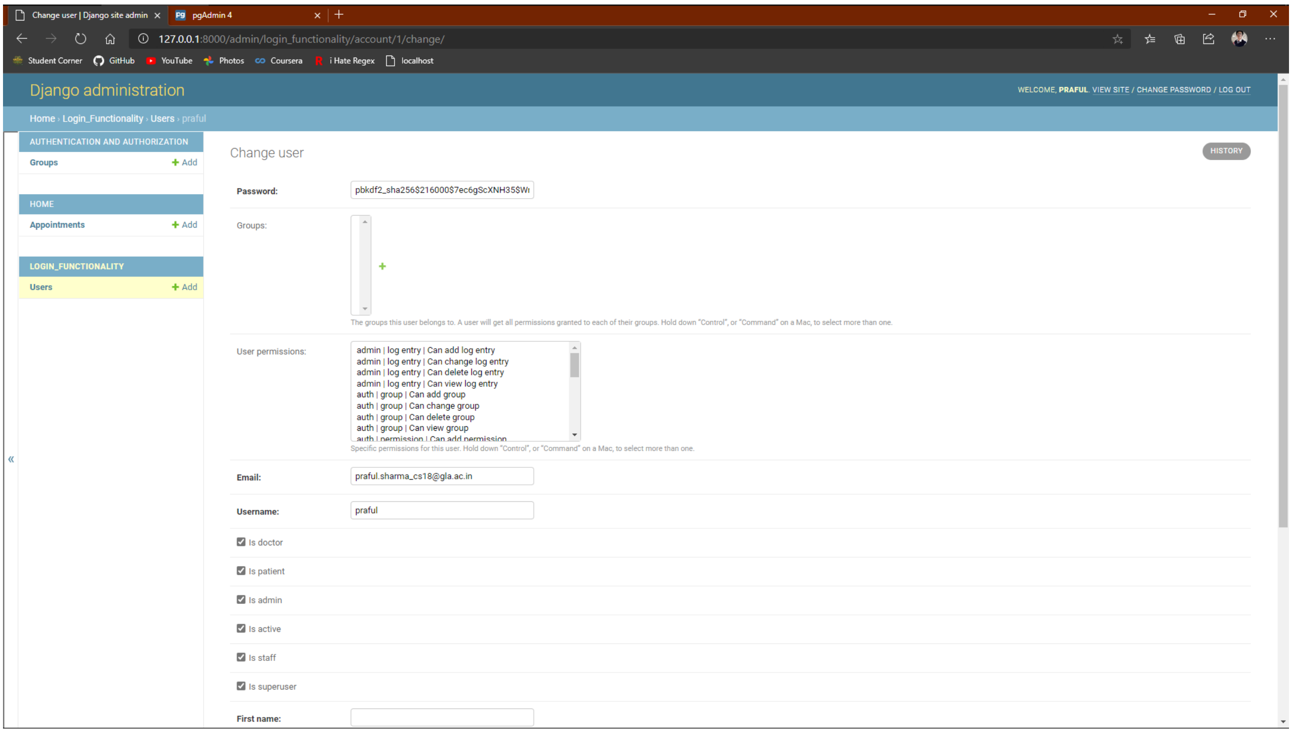
Select user to change | Django
pgAdmin 4
127.0.0.1:8000/admin/login_functionality/account/
Student Corner
GitHub
YouTube
Photos
Coursera
i Hate Regex
localhost
Django administration
WELCOME, PRAFUL. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Login_Functionality › Users
AUTHENTICATION AND AUTHORIZATION
Groups
+ Add
HOME
Appointments
+ Add
LOGIN_FUNCTIONALITY
Users
+ Add
ADD USER +
«

Select user to change

Action: --------- Go 0 of 3 selected

□	USER
□	johndoe
□	uttkarsh
□	praful

3 users


Change user | Django site admin
pgAdmin 4
Student Corner
GitHub
YouTube
Photos
Coursera
i Hate Regex
localhost
Django administration
WELCOME, PRAFUL. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Login_Functionality › Users › praful
AUTHENTICATION AND AUTHORIZATION
Groups
+ Add
HOME
Appointments
+ Add
LOGIN_FUNCTIONALITY
Users
+ Add
HISTORY

127.0.0.1:8000/admin/login_functionality/account/1/change/

Change user

Password: pbkdf2_sha256$216000$7ec6gScXNH35$W

Groups:

The groups this user belongs to. A user will get all permissions granted to each of their groups. Hold down "Control", or "Command" on a Mac, to select more than one.

User permissions:

admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission

Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select more than one.

Email: praful.sharma_cs18@gla.ac.in

Username: praful

☑ Is doctor

☑ Is patient

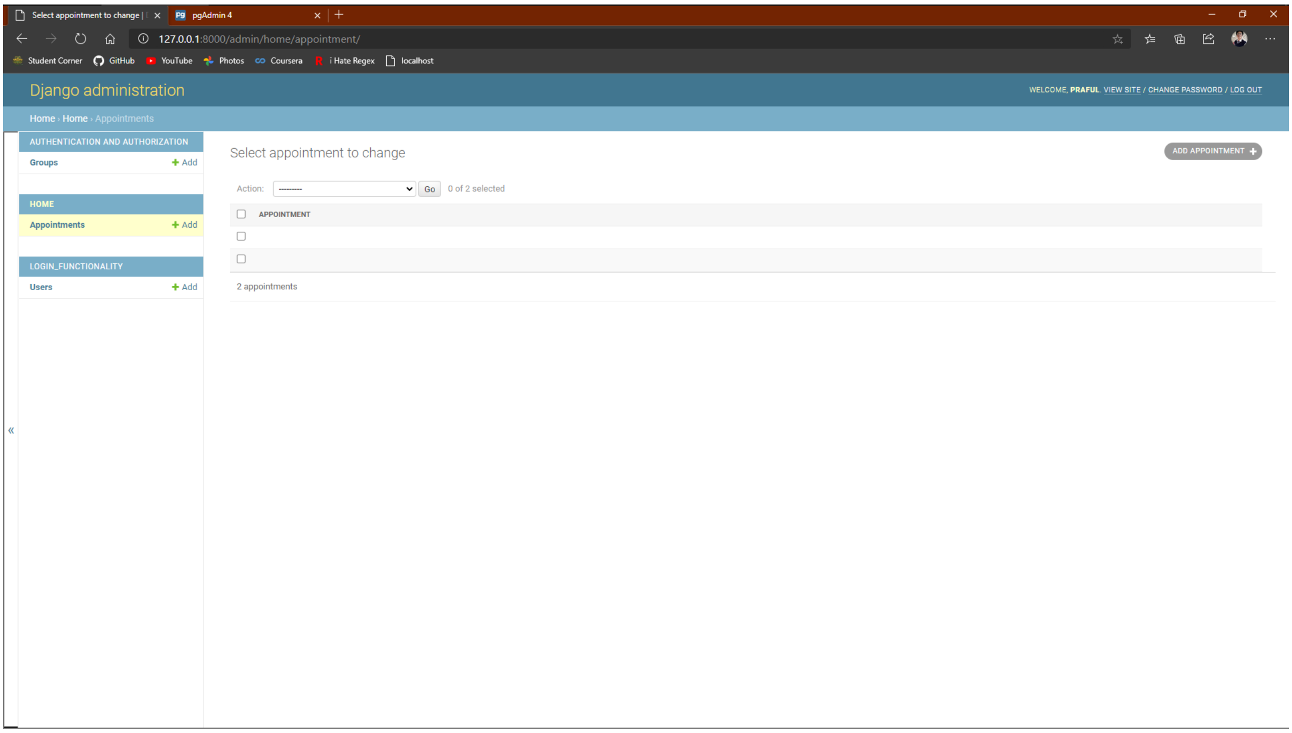
☑ Is admin

☑ Is active

☑ Is staff

☑ Is superuser

First name:



127.0.0.1:8000/admin/home/appointment/

Select appointment to change

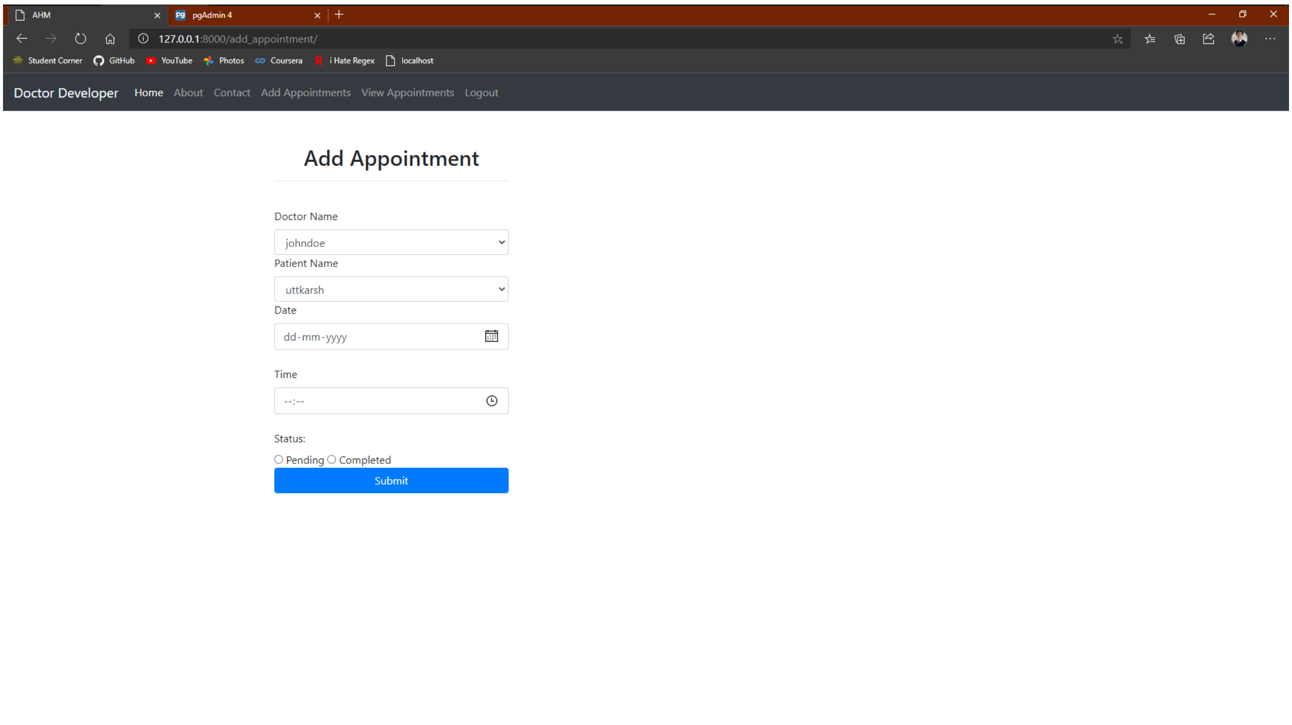
Action: --------- Go 0 of 2 selected

□ APPOINTMENT

□

□

2 appointments



Add Appointment

Doctor Name

johndoe

Patient Name

uttkarsh

Date

dd-mm-yyyy

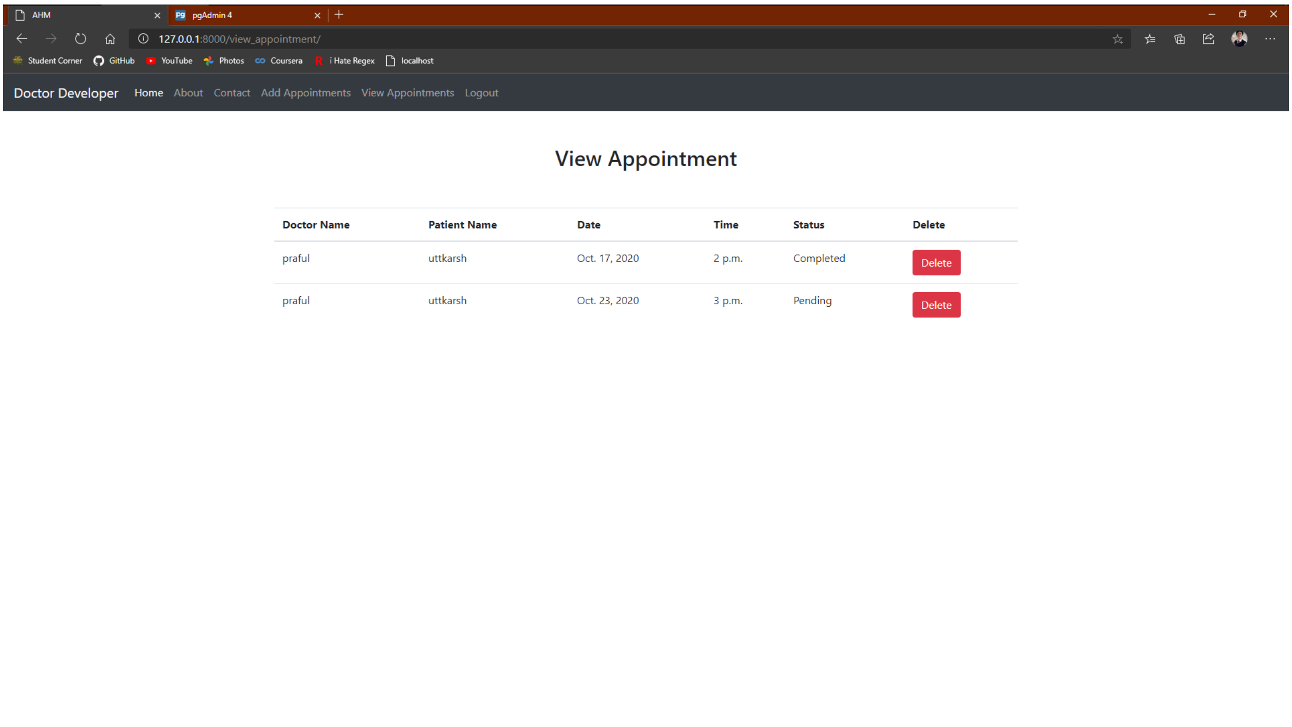
Time

--:--

Status:

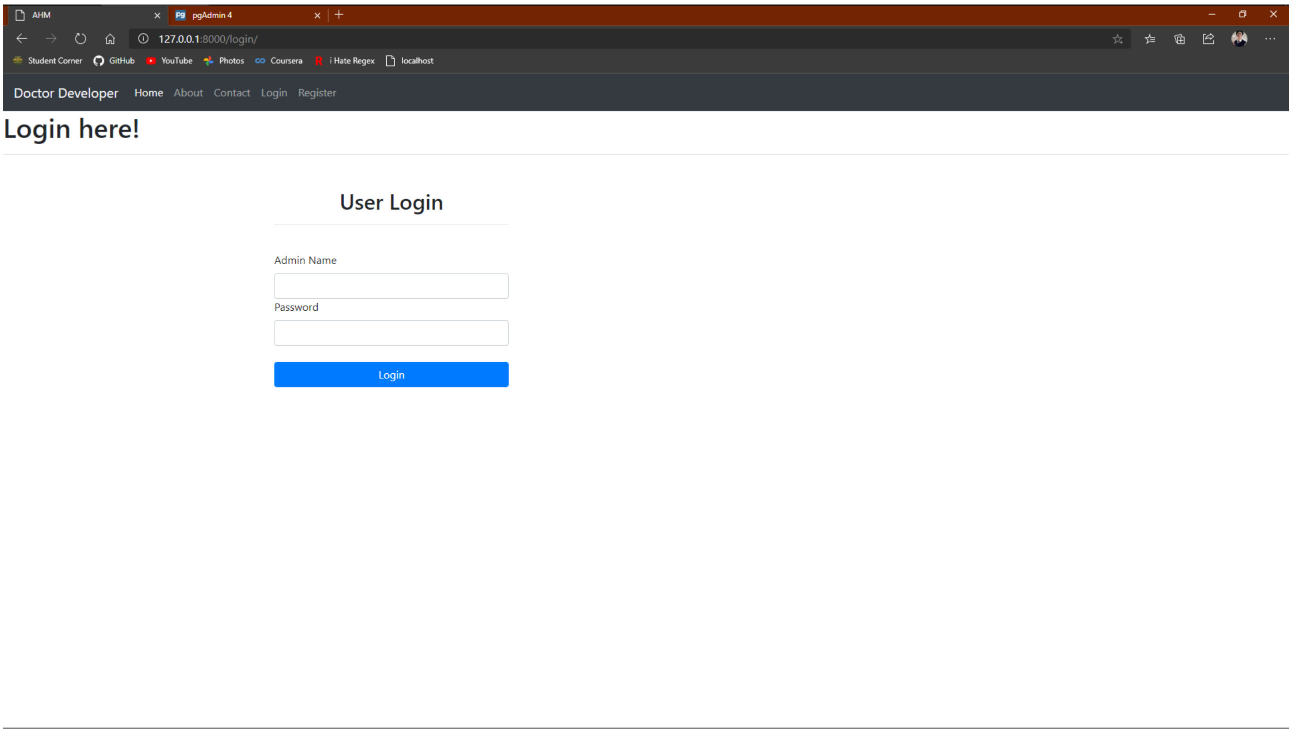
○ Pending ○ Completed

Submit



View Appointment

Doctor Name	Patient Name	Date	Time	Status	Delete
praful	uttkarsh	Oct. 17, 2020	2 p.m.	Completed	Delete
praful	uttkarsh	Oct. 23, 2020	3 p.m.	Pending	Delete

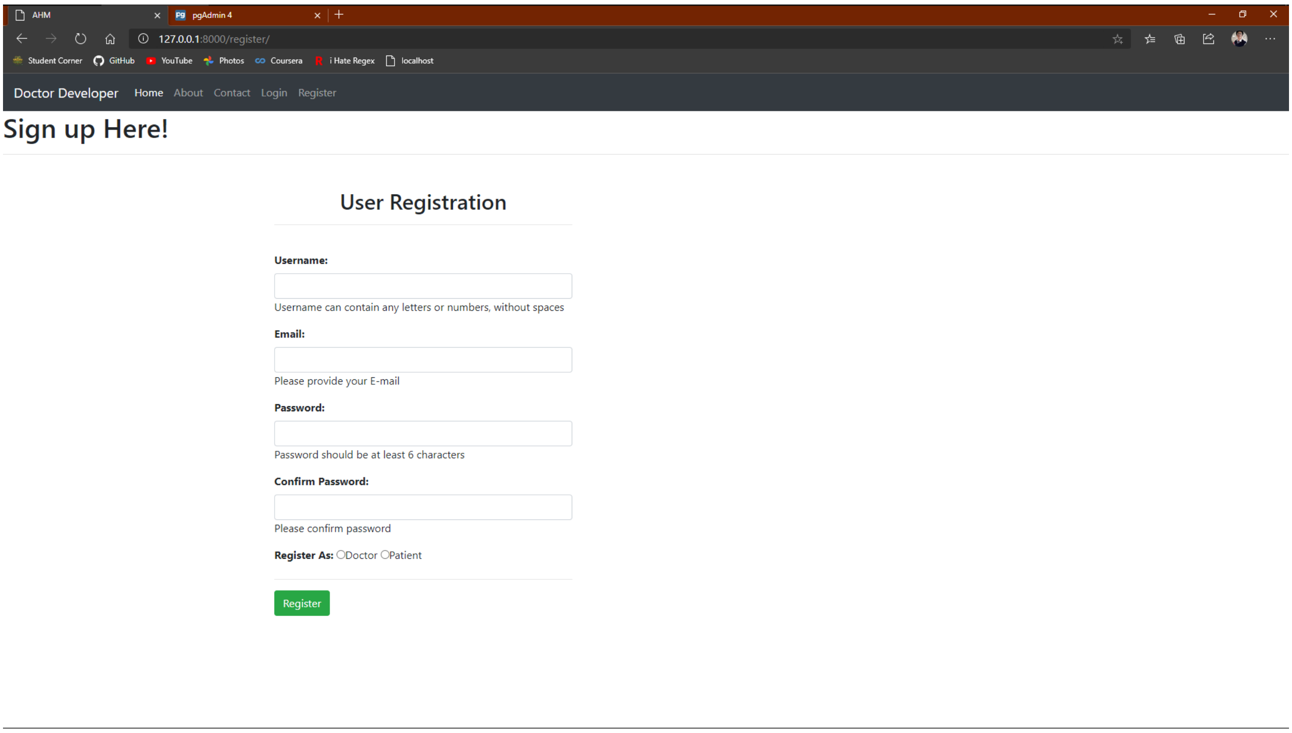


Login here!

User Login

Admin Name

Password



Sign up Here!

User Registration

Username:

Username can contain any letters or numbers, without spaces

Email:

Please provide your E-mail

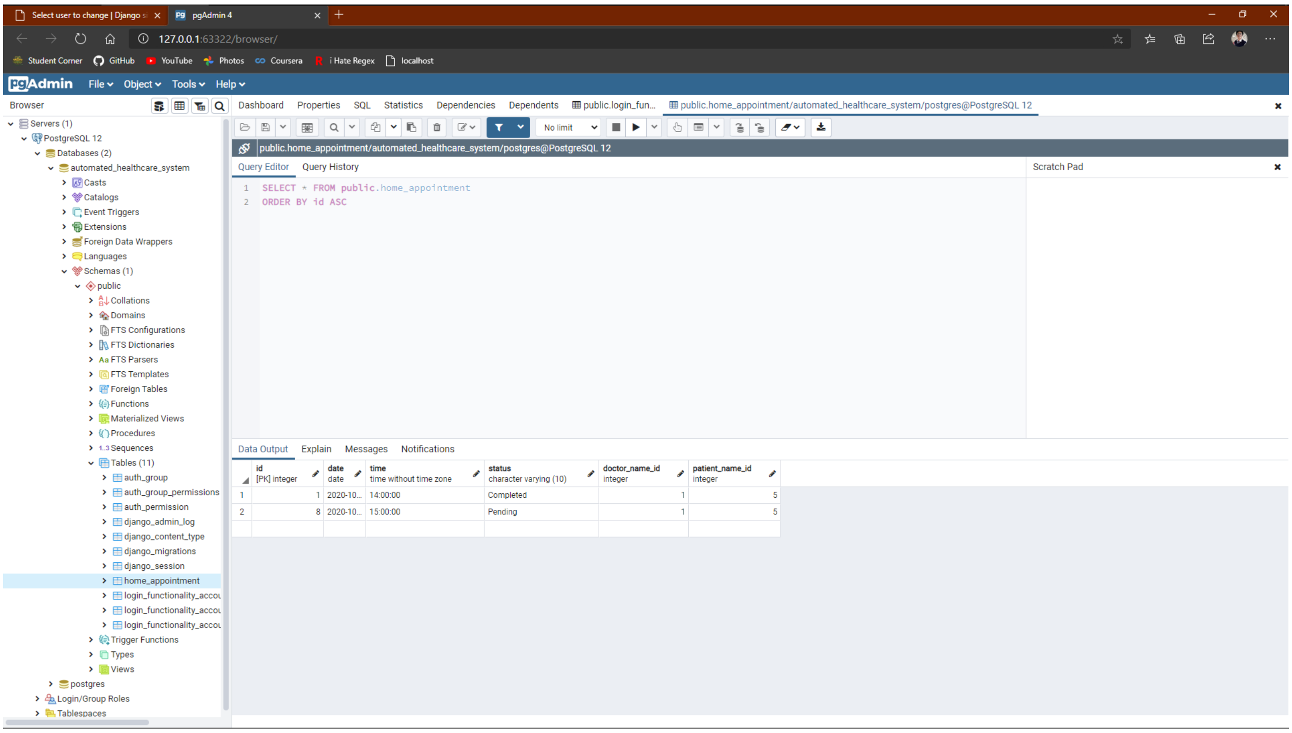
Password:

Password should be at least 6 characters

Confirm Password:

Please confirm password

Register As: ○Doctor ○Patient



public.home_appointment/automated_healthcare_system/postgres@PostgreSQL 12

```
SELECT * FROM public.home_appointment
ORDER BY id ASC
```

	id [PK] integer	date date	time time without time zone	status character varying (10)	doctor_name_id integer	patient_name_id integer
1	1	2020-10...	14:00:00	Completed	1	5
2	8	2020-10...	15:00:00	Pending	1	5

References

- Introduction to ANN
  https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp
- Introduction to CNN
  https://towardsdatascience.com/an-introduction-to-convolutional-neural-networks-eb0b60b58fd7
- Deep Learning CNN, Theano and Tensorflow
  https://www.udemy.com/course/deep-learning-convolutional-neural-networks-theano-tensorflow/learn/lecture/8944074?start=300#content