DISTRIBUTED HEALTHCARE SYSTEM

**Group 4**

Priyanka Subramanyam

Karthika MadhavanPillai Sasidharan Nair

Srividhya Prakash Rangarajan

# 1.Table of Contents

Contents

[1.Table of Contents 2](#_Toc513642349)

[2.Introduction/Overview 3](#_Toc513642350)

[3.Design/Architecture: present the architecture of the project, including technology, tools and features used. 4](#_Toc513642351)

[4.Results and analysis: include screenshots of all the results. 5](#_Toc513642352)

[5.Future work: suggest potential project extensions. 6](#_Toc513642353)

[6.Conclusion 7](#_Toc513642354)

[7.References: list the articles, books, and web links that you have researched. 8](#_Toc513642355)

[8.Contribution: individual contribution of each member towards the project 9](#_Toc513642356)

[9.Appendices:a.embed the code filesb.include presentation slidesThe size of the report excluding the appendices should be within 5000 words (10 to 20 pages). You can include snapshots of the results withinthe text or as an appendix. 9](#_Toc513642357)

# 2.Introduction/Overview

# 3.Design/Architecture: present the architecture of the project, including technology, tools and features used.

The following is the workflow of our project.

1. **Sign up/Login**

The first page gives us two options:-1.Sign up and Login

This screenshot shows how the signup page looks like. Any new user can create his/her user account with the hospital. And the record will be updated in the database.

A screenshot of a cell phone

Description generated with very high confidence

Fig 1:Sign up Page

A screenshot of a cell phone

Description generated with high confidence

Fig 2:New Record is updated in the database

# A screenshot of a cell phone Description generated with very high confidence

Fig 3: Login in Page

A screenshot of a cell phone

Description generated with very high confidence

Fig 3: Login in Admin Page

1. **Patient Profile**

Using the credentials any patient can login and perform three major functions:-

1. Book appointment
2. Cancel appointment
3. Get Scan report

The following screenshots show the process:-

A screenshot of a cell phone

Description generated with very high confidence

Fig 4: Profile Page of the user

# A screenshot of a cell phone Description generated with very high confidence

Fig 5: Book Appointment

A screenshot of a cell phone

Description generated with very high confidence

Fig 6: Cancel Appointment

A close up of a logo

Description generated with very high confidence

Fig 7: Confirmation of Cancellation

A screenshot of a social media post

Description generated with very high confidence

Fig 8: Get scan report-Email sent to the patient’s mail id

1. **Doctor Profile**

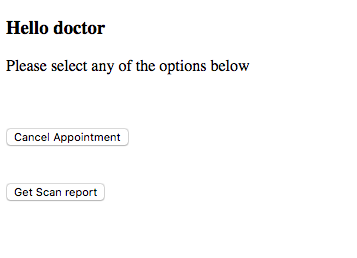
****

Fig 9: Doctor’s profile page

**A screenshot of a cell phone

Description generated with very high confidence**

Fig 10: Upload scan report

1. **Insurance Agent Profile**

The insurance agent can check the insurance status of patient.

A screenshot of a cell phone

Description generated with high confidence

Fig : Functions that an insurance agent can perform

A screenshot of a cell phone

Description generated with very high confidence

Fig : Searching for customers for a particular insurance companyA screenshot of a cell phone

Description generated with very high confidence

Fig : List of customers of “wellsfargo”

1. **Database**

A screenshot of a cell phone

Description generated with very high confidence

Fig : Patient Table

A screenshot of a social media post

Description generated with very high confidence

Fig :Doctor Table

# A screenshot of a social media post Description generated with very high confidence

Fig: Insurance Agent Table

**Future Extensions**

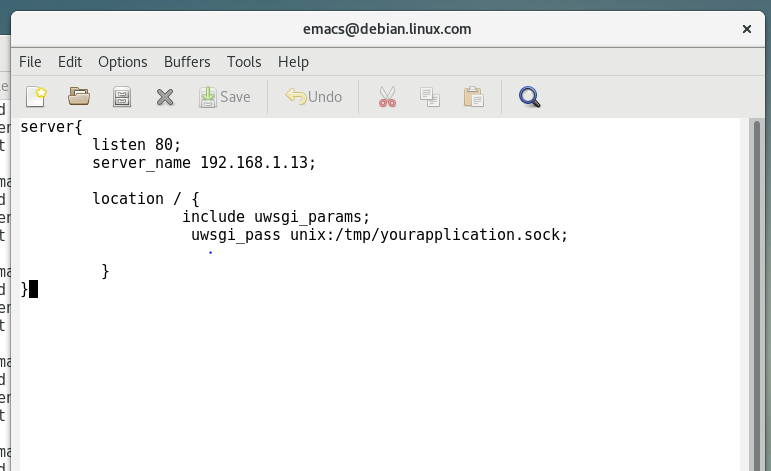
# Reverse Proxy Server

A **reverse proxy server** is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server. A reverse proxy provides an additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers.

Nginx is used as a load balancing server in our project

Steps for configuring and running nginx on port 80:

Starting nginx server: Listens to port 80



Configuration file for nginx. Configures the server to listen to port 80.

Server\_name : is domain name or ip

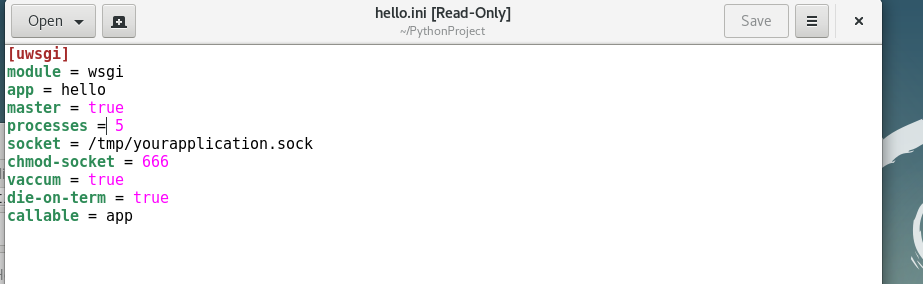
Location: socket path provided in the uwsgi ini file.

Command for restarting nginx:

sudo service nginx restart

Configuring uwsgi to act as a gateway for nginx to host flask app.

It creates a socket which listens to communications from nginx and pass it to flask

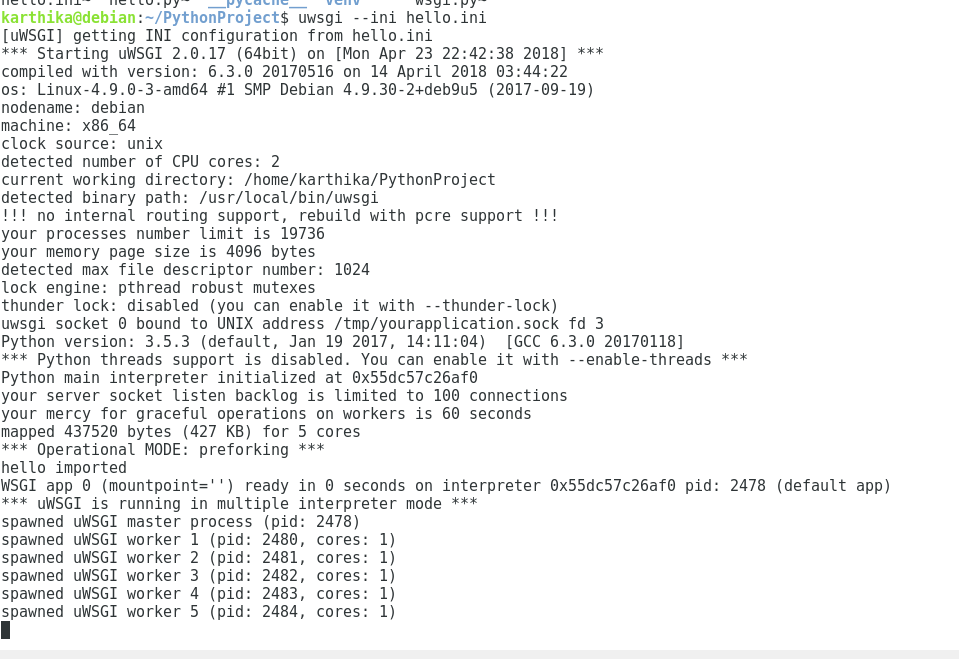


Run the uswgi.ini, so that the gate way is up and running

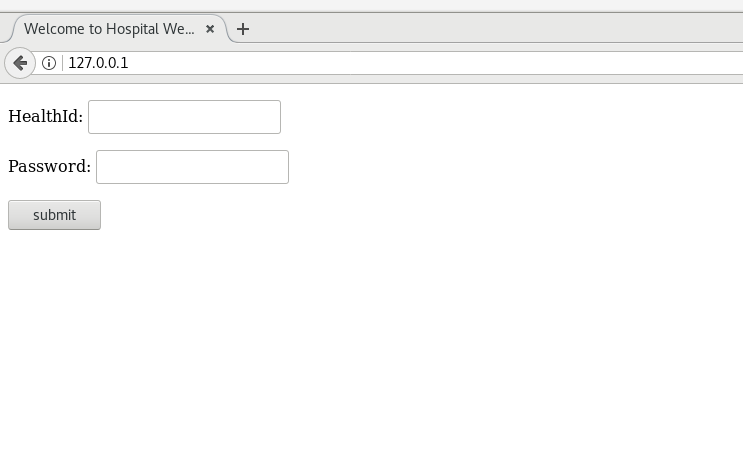
Command for running the .ini file of uwsgi

uwsgi --ini hello.ini

uwsgi is up and running. Since number of processes are given as 5 in the ini file, uwsgi has created 5 processes to handle the requests from nginx



After starting uwsgi, go to the browser and give the ip address. Port 80 is now hosting the hospital website.



# 4.Results and analysis: include screenshots of all the results.

# 

# 5.Future work: suggest potential project extensions.

# 6.Conclusion

# 7.References: list the articles, books, and web links that you have researched.

# 8.Contribution: individual contribution of each member towards the project

# 9.Appendices:a.embed the code filesb.include presentation slidesThe size of the report excluding the appendices should be within 5000 words (10 to 20 pages). You can include snapshots of the results withinthe text or as an appendix.