

Willhelm International

```
# Wi_international
Willhelm International Care

## Description

An appointment website designed for Willhelm International.

## Tech stack
Django(full stack yay!), Sql, Bootstrap, jQuery.

<!-- **Webiste:** [Deployed on Pythonanywhere](http://jasonchan.pythonanywhere.com) -->

## Admin view:

Administrator can be created only through terminal of server by using "./manage.py
createsuperuser" command. An [administrator](username: test, password: test) has been
created, please feel free to test the data intergrity and consistency.

1. Administrator has Create, Read, Update, and Delete control of the user.
2. Administrator has Create, Read, Update, and Delete control of the appointment.
3. Administrator has Create, Read, Update, and Delete control of the prescription.

## Doctor view

1. Prescribe medication (medication/dosage/duration).
2. View scheduled appointments
3. Schedule Follow up appointment in the future
4. See patient appointment and medication history

## Patient View

1. book appointments with specific doctors - view only current medication and dosage.
2. Book appointment by types(primary care, flu, mental care, and etc)

# Django hooks

# Deploying options
## Deploying on AWS
1. Deploying a Django application to Elastic Beanstalk
https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html
### pros:
Easy to see the backend interface
### cons:
Will need time to manage
```

```
Need a domain name
```

Deploying on Heroku

pros:

No need domain or dbms

Easy to setup and free usage

cons:

<https://devcenter.heroku.com/articles/django-app-configuration>

'''

You're using the staticfiles app without having set the STATIC_ROOT setting to a filesystem path.

'''

1. Setting up for local deployment

Assuming that Python 3.9 or up and Dip has been installed

2. Cloning the repository

--> Clone the repository using the command below :

```
git clone https://github.com/killerfrost22/Wi_international
```

--> Move into the directory where we have the project files :

```
cd Wi_international
```

--> Create a virtual environment :

```
# Let's install virtualenv first
```

```
pip install virtualenv
```

```
# Then we create our virtual environment
```

```
virtualenv envname
```

--> Activate the virtual environment :

```
envname\scripts\activate
```

--> Install the requirements :

```
pip install -r requirements.txt
```

3. Setup Virtualenvironment

```
py -m venv venv # Create virtual environment
```

```
source venv/bin/activate # Activate virtual environment
```

```
pip install -r requirements.txt # Install requirements
```

```
python server.py # Run server
```

```
deactivate # Deactivate virtual environment
```

4.Migrate the database

```
python manage.py makemigrations
```

```
python manage.py migrate --run-syncdb
```

5. Running the App

--> To run the App, we use :

```
python manage.py runserver
```

⚠ Then, the development server will be started at <http://127.0.0.1:8000/>

Django Admin mode

<http://127.0.0.1:8000/admin/login/?next=/admin/>

Admin: ted@gmail.com

Pswd: josephao

Admin links admin functions and it links admin profile, it can also display change log
After running, the server and accessing the homepage, go to admin panel to add all the medicines,
their preparations and departments as they are all managed by the admin.

<http://127.0.0.1:8000/admin>

Django patient account

Username: bucky@gmail.com

Pswd: josephao

Django doctor account

Username: da@gmail.com

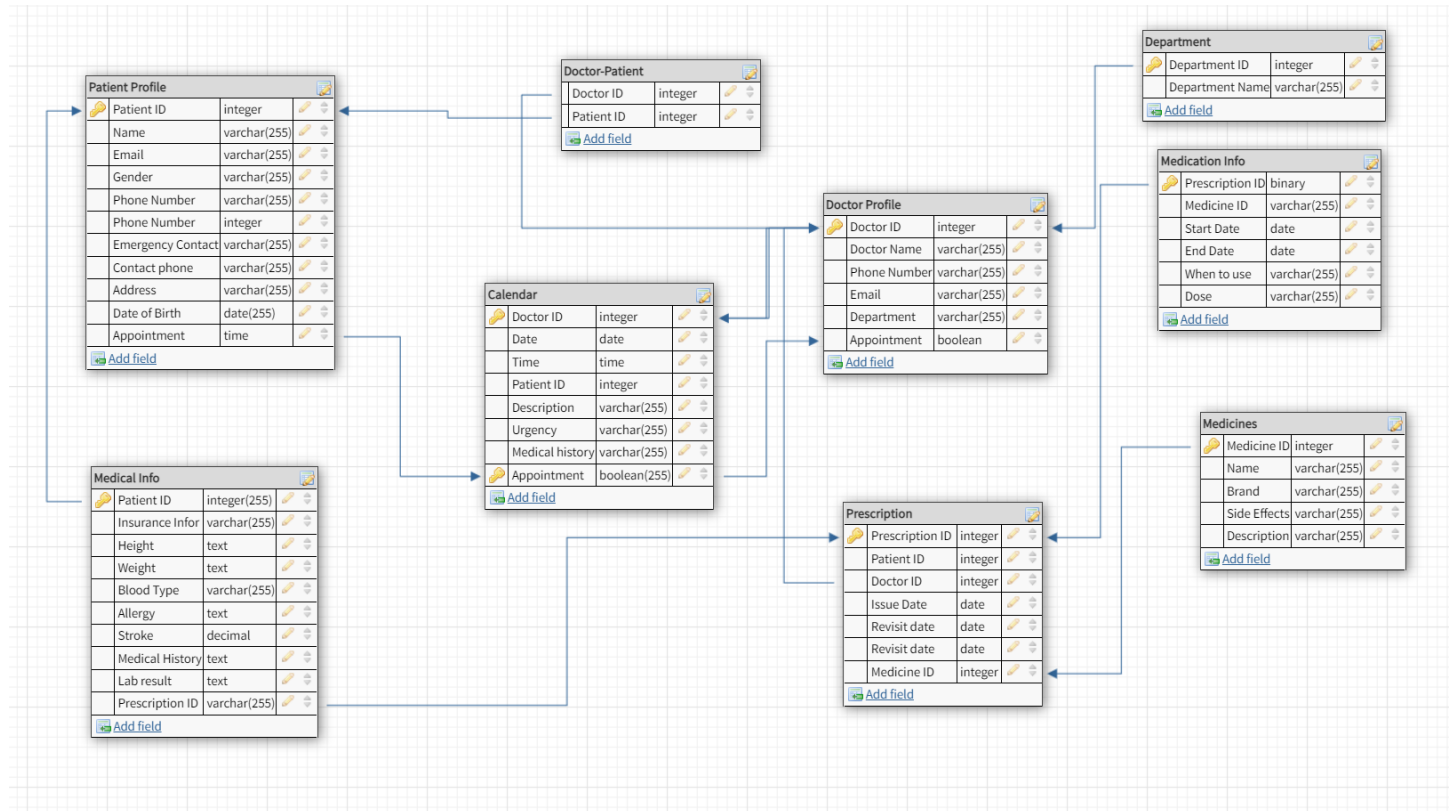
Pswd: josephao

Links doctor functions links to doctors profile Displays upcoming appts

Change password

In the command line python .\manage.py changepassword bucky@gmail.com

DataBase Design:



DB Browser for SQLite - C:\Users\josep\OneDrive\Desktop\jagddirect final\Actual\db.sqlite3

File Edit View Tools Help

Database Structure Browse Data Edit Pragma Execute SQL

Table: patient_mysuer


	id	password	last_login	email	is_doctor	is_active	is_admin
	Filter	Filter	Filter	Filter	Filter	Filter	
1	pbkdf2_sha256\$320000\$9q1X4QuME...	2022-02-19 08:01:28.676022	admin@admin.com	0	1	1	
2	pbkdf2_sha256\$320000\$SdO7RvPICE...	2022-02-19 08:01:28.676022	doctor@gmail.com	1	1	0	
3	pbkdf2_sha256\$320000\$9MQAqGRF...	2022-02-19 08:01:28.676022	doctorstrange@gmail.com	1	1	0	
4	pbkdf2_sha256\$320000\$9ey5LEx37A...	2022-02-20 08:01:28.676022	curt@gmail.com	1	1	0	
5	pbkdf2_sha256\$320000\$9jwvSS4F...	2022-02-20 08:01:28.676022	otto@gmail.com	1	1	0	
6	pbkdf2_sha256\$320000\$9PymVTVQE3...	2022-02-20 08:01:28.676022	tsu@gmail.com	1	1	0	
7	pbkdf2_sha256\$320000\$3zP2aM4d3...	2022-02-20 08:01:28.676022	pym@gmail.com	1	1	0	
8	pbkdf2_sha256\$320000\$9F0eeW11j5...	2022-02-20 08:01:28.676022	nishi@gmail.com	1	1	0	
9	pbkdf2_sha256\$320000\$0m87nxd9M...	2022-02-20 08:01:28.676022	saurabhi@gmail.com	0	1	0	
10	pbkdf2_sha256\$320000\$T55uA833dR...	2022-02-20 08:01:28.676022	bruce@gmail.com	1	1	0	
11	pbkdf2_sha256\$320000\$9YUa6LLkueK...	2022-02-20 08:01:28.676022	sunny@gmail.com	0	1	0	
12	pbkdf2_sha256\$320000\$9guUQFvHb...	2022-02-20 08:01:28.676022	utkarshi@gmail.com	0	1	0	
13	pbkdf2_sha256\$320000\$9pH97H8FTY...	2022-02-20 08:01:28.676022	sharon@gmail.com	0	1	0	
14	pbkdf2_sha256\$320000\$9v6607Gb7p...	2022-02-20 08:20:28.676022	dinesh@gmail.com	0	1	0	
15	pbkdf2_sha256\$320000\$9sacrc9v8FFB3...	2022-02-20 08:20:28.676022	stevie@gmail.com	0	1	0	
16	pbkdf2_sha256\$320000\$9knnnQCGS1...	2022-02-20 08:20:28.676022	stark@gmail.com	0	1	0	
17	pbkdf2_sha256\$320000\$3YUuBuqL1...	2022-02-20 08:01:28.676022	bucky@gmail.com	0	1	0	
18	pbkdf2_sha256\$320000\$9MBC8gftVVC...	2022-02-20 08:01:28.676022	joe@gmail.com	1	1	1	
19	pbkdf2_sha256\$320000\$9xOABW1ZD...	2022-02-28 03:50:33.731580	ted@gmail.com	1	1	1	
20	pbkdf2_sha256\$320000\$9VuZfnnVomL...	2022-02-28 13:04:43.242581	da@gmail.com	1	1	0	


Go to: 1


SQL Log Plot DB Schema Remote

Overview of project:

Homepage


Home Yizhan Ao






Medical History
Records On-Demand

Need to see a doctor? View your medical records and prescriptions


MAKE APPOINTMENTS



24 Hours Service
On-Demand Doctors

Peek into our industry leading professional doctors on demand

VIEW DOCTORS



Medicine Schedule
Prescriptions

View your prescriptions and be on top of your health by taking your medicines on time

VIEW PRESCRIPTION

Doctor

Prescribe a Patient

Date

02/28/2022

Next Visit

mm/dd/yyyy

Reason

Enter the reason for visiting

Doctor Notes

Enter a note for future referencing

Patient

Vicky

Doctor

Yizhan Ao

ADD MEDICINE

SUBMIT

Patients View:

Search here ..

SEARCH

Vicky

Vicky@gmail.com

21331

VIEW FULL PROFILE

PRESCRIBE

Captain America

cap@gmail.com

21332

VIEW FULL PROFILE

PRESCRIBE

Googo

googo@gmail.com

21333

VIEW FULL PROFILE

PRESCRIBE

Halk

halk@gmail.com

21334

VIEW FULL PROFILE

PRESCRIBE

Black Widow

widow@gmail.com

21335

VIEW FULL PROFILE

PRESCRIBE

Steve Rogers

steve@gmail.com

21336

VIEW FULL PROFILE

PRESCRIBE

Tony Stark

stark@gmail.com

21337

VIEW FULL PROFILE

PRESCRIBE

Bucky Barnes

bucky@gmail.com

21338

VIEW FULL PROFILE

PRESCRIBE

Pan40

Medicine Price : Rs 12.00
Medicine Brand : Abbott
Medicine Side Effects : Nausea, Vomiting, Dizziness
Medicine Description : Gastritis

[VIEW DETAILS](#)

Paracetamol

Medicine Price : Rs 12.00
Medicine Brand : Tyleno
Medicine Side Effects : dark urine, clay-colored stools
Medicine Description : Helpful as a pain killer and in mild fever.

[VIEW DETAILS](#)

What could be done better?

1) Google Calendar API

`pip install google-api-python-client` (from the requirements already)

I can avoid running the server and storing all the appointment data which in the end messed up my code and I have to remove all of the calendar functions to be able to run the code successfully. If I got more time I would change the original schema to be hosted on google calendar and make the patient to send request and the doctor will receive emails and doctors can either choose approve or choose a different time.

2) AWS Beanstalk

This part is the pain in my project. My original deployment was on Heroku however, my code of calendar was malfunction and I have to go back to ground zero. And Heroku refuses to work with me again. So I have to redirect to the AWS beanstalk which works but partially.