
AuShadha Documentation

Release 0.1

Dr. Easwar T.R and others (see credits)

Aug 15, 2018

Contents

1	Introduction to AuShadha Project	3
1.1	AuShadha (): Means medicine in Sanskrit.	3
1.2	License	3
1.3	Aims	3
1.4	How to Test	3
1.5	Project Structure	4
1.6	Plan	4
1.7	Pluggable Modules under Developement	5
1.8	Pluggable Modules Planned	5
1.9	For Developers	5
1.10	Repository:	5
2	Setup the PostgreSQL database for your OS	7
3	Creating and AuShadha PostgreSQL database	9
4	Installation of AuShadha	11
4.1	How to Test	11
5	DEPENDENCIES	13
5.1	Comments on Dependencies	13
6	AuShadha-Core	15
7	AuShadha-Stock Version	17
8	AuShadha-Pluggables	19
9	Resources & Links	21
9.1	For Developers	21
9.2	Google Groups	21
9.3	Project Website & Blog	21
9.4	Facebook Page	21
9.5	Twitter Handle	21
9.6	Documentation	22
9.7	Repositories	22
10	Indices and tables	23

Contents:

Introduction to AuShadha Project

1.1 AuShadha (): Means medicine in Sanskrit.

This is a Open Source Electornic Medical Records (EMR) & Public Health Management System for Small Clinics. It might be even suited for speciality medical practice after some customisation. It is developed with Python, Django and Dojo. For Development the code uses sqlite3 database. Deployment will use PostgreSQL.

1.2 License

AuShadha code is licensed under GNU-GPL Version 3. Django, Dojo, Icons and other thrid party modules are licensed as per their authors. This should be honoured.

1.3 Aims

There are many great Open Source EMR and Clinic Management projects, but most of them have very poor front ends. As in life, beauty and usability are key in medical practice. It is natural that more visually appealing it is more you want to use it. AuShadha will strive to achieve a balance between back and front end design. To achieve that aim Django and Dojo has been chosen. AuShadha is designed by me, a Medical Doctor in association with other Developers, so usability and practicability will always be foremost. AuShadha benefits from inputs, feature requests from my medical colleagues as the development progresses.

1.4 How to Test

1. For AuShadha requirements please refer to REQUIREMENTS.txt in docs/
2. AuShadha has been tested and developed with Python 2.7, Django 1.6x and Dojo 1.8x in Linux. It should work as long as dependencies are satisfied.

3. Ideally create a Python Virtual Environment. If virtualenv is not installed, please run in Debian systems `sudo easy_install virtualenv`
4. After installing virtualenv, Create a Python Virtual Environment `virtualenv python_env`
5. Change the working directory to the virtualenv folder `cd python_env` and activate it `source bin/activate`
6. With Python pip installed, run `pip install -r ../docs/REQUIREMENTS.txt` if you are in `python_env` directory . This will install all the necessary dependencies you need.

You may need to install Python development libraries if you are on Linux.

\$ apt-get install python-dev #in debian systems.

\$ yum install python-devel #in Redhat-like systems.

7. cd into src/AuShadha directory `cd src/AuShadha`
8. run `python manage.py runserver`
9. Use your browser to navigate to <http://localhost:8000/AuShadha/> . You will be greeted with a login page. Use username = admin, password = admin for a trial run.
10. Please read the issues, license before using. Currently AuShadha is under active development and is not fit in anyway for real world use.

1.5 Project Structure

2. docs : Contains the Requirements.txt, License.txt
3. src : Project Source Code, Media files(Icons,Images, File uploads), Javascript
4. README.md
5. LICENSE.txt

1.6 Plan

AuShadha project is split between AuShadha-stock (which contains AuShadha-core and some Stock applications) and Au-Pluggable (the pluggable modules for AuShadha). User can freely mix and match the modules he wants to create his own AuShadha brew.

In other words he is not stuck with what the developer has packaged and is free to repackage it in any way he wants. The AuShadha-core will help him / her create pluggable applications that integrate well into AuShadha, but user is also free to create a completely different Django application. As long as he / she sticks to the Django pluggable practices the application can be easily integrated into AuShadha. Ofcourse if he were to use AuShadha-core's API to develop he can do it more easily. AuShadha-core API is very young, mostly untested. It does not force the developer to re-learn anything that would not be useful outside AuShadha.

1.6.1 Completed Modules

1. AuShadha-core
2. AuShadha-PatientRegistration
3. AuShadha-Contact
4. AuShadha-Phone

5. AuShadha-Guardian
6. AuShadha-Demographics
7. AuShadha-MedicationList
8. AuShadha-AllergyList
9. AuShadha-History
10. AuShadha-OPD_Visit

1.7 Pluggable Modules under Developement

- currently under development
1. aushadha_demographics_us
 2. aushadha_demographics_india

1.8 Pluggable Modules Planned

1. aushadha_obs_and_gyn_india
2. aushadha_immunisation_india
3. aushadha_neonatal_india

1.9 For Developers

Suggestions and participation are welcome.

Please email me at dreaswar@gmail.com or Google Groups at aushadha@googlegroups.com

Visit project website at <http://aushadha.org>.

Follow the Project news at <http://www.facebook.com/AuShadha/>

1.10 Repository:

1. <http://github.com/dreaswar/AuShadha>
2. <http://github.com/dreaswar/Au-Pluggables>

CHAPTER 2

Setup the PostgreSQL database for your OS

Please refer to your OS notes on how to do this. AuShadha has been tested with PostgreSQL Server version 9.3 Once you have installed Postgres server and pgadmin3 client you should be able to run *psql* from terminal and get a prompt

Creating and AuShadha PostgreSQL database

Preferred DB for running AuShadha is PostgreSQL.

1. Set up PG for your OS Please see your OS website for instructions on setting up PG PG should be is setup and accesible from command line as >> psql prompt
2. Set up a System user with same name as the Postgres user Create an OS user with `<adduser>` command from terminal Create a `/home/<user_name>` directory for the user and run `chown -R <user_name> /home/<user_name>` The username used by default for DB and user is 'aushadha' with password of *aushadha*
3. Create an empty DB and Grant permissions

From terminal run the *createuser* to create and user with username 'aushadha' Run *psql* to get into the pg command line from the user that is allowed to run *psql*

4. Run the following

```
CREATE DATABASE aushadha WITH OWNER aushadha;
```

```
GRANT ALL ON DATABASE aushadha TO <system_os_user> WITH GRANT OPTION;
```

Installation of AuShadha

4.1 How to Test

1. For AuShadha requirements please refer to REQUIREMENTS.txt in docs/
2. AuShadha has been tested and developed with Python 2.7, Django 1.6x and Dojo 1.8x in Linux. It should work as long as dependencies are satisfied.

4.1.1 Creating a Virtual Environment to run:

3. Ideally create a Python Virtual Environment. If virtualenv is not installed, please run in Debian systems `sudo easy_install virtualenv`
4. After installing virtualenv, Create a Python Virtual Environment `virtualenv python_env`
5. Change the working directory to the virtualenv folder `cd python_env` and activate it `source bin/activate`

4.1.2 Installing Dependencies:

6. With Python pip installed, run `pip install -r ../docs/REQUIREMENTS.txt` if you are in `python_env` directory . This will install all the necessary dependencies you need.
7. You may need to install Python development libraries if you are on Linux.
\$ apt-get install python-dev #in debian systems.
\$ yum install python-devel #in Redhat-like systems.
8. cd into src/AuShadha directory `cd src/AuShadha`

4.1.3 Prepare and Download Dojo Javascript Library

9. Latest Dojo Library can be downloaded at : <http://dojotoolkit.org/download/>

Download the latest Dojo library and extract it into *AuShadha/src/AuShadha/AuShadha/media/plugins/dojo/*

After extracting the folder structure should be like: *dojo/dojo/ dojo/dojox/ dojo/dijit/*

AuShadha has been tested with Dojo 1.8.1

4.1.4 Prepare the database and install the fixtures

9. run *\$ python manage.py syncdb && python manage.py runserver*

4.1.5 See it in action & login

10. Use your browser to navigate to <http://localhost:8000/AuShadha/>

You will be greeted with a login page. Use username = admin, password = admin for a trial run.

11. Please read the issues, license before using.

Currently AuShadha is under active development and is not fit in anyway for real world use.

1. Django>=1.5.4
2. Jinja2>=2.7.1
3. MarkupSafe>=0.18
4. PyYAML>=3.10
5. Pygments>=1.6
6. South==0.8.1
7. Sphinx>=1.2b2
8. argparse
9. autopep8
10. docformatter>=0.5.5
11. docutils>=0.11
12. jsbeautifier>=1.4.0
13. pep8>=1.4.6
14. six>=1.3.0
15. untokenize>=0.1
16. wsgiref>=0.1.2

5.1 Comments on Dependencies

These dependencies are as of February 13, 2014. Most of it are indirect dependency of Sphinx, PyYAML.

Some of it are not strictly needed like autopep, jsbeautifier. These could be useful for the developer to check and modify the code.

All the current dependencies are listed in *AuShadha/docs/REQUIREMENTS.txt* file.

This can be run with pip to install all of them in one go

CHAPTER 6

AuShadha-Core

CHAPTER 7

AuShadha-Stock Version

CHAPTER 8

AuShadha-Pluggables

Resources & Links

9.1 For Developers

Suggestions and participation are welcome.

9.2 Google Groups

aushadha@googlegroups.com

9.3 Project Website & Blog

1. <http://aushadha.org>
2. <http://blog.aushadha.org>

9.4 Facebook Page

<http://www.facebook.com/AuShadha/>

9.5 Twitter Handle

[@aushadha_emr](https://twitter.com/aushadha_emr)

9.6 Documentation

AuShadha is documented on readTheDocs using reStructuredText and Sphinx

Find the Documentation here: <https://readthedocs.org/projects/aushadha/>

This documentation applies to the 'master' branch

9.7 Repositories

1. <http://github.com/dreaswar/AuShadha>

The main AuShadha Repository is located here. This houses the AuShadha-core and stock applications.

2. <http://github.com/dreaswar/Au-Pluggables>

The pluggable application repository is here. This repo has pluggable applications designed to replace / augment existing applications.

CHAPTER 10

Indices and tables

- `genindex`
- `modindex`
- `search`