Rajasthan

App and server code for Aanganwadi.

Component	Coverage
Server	

App and server code for Aanganwadi. Relevant documents like sprint reports can be found on Google Drive. Ask for access if you do not have it.

Working API

- `/user`
- `/user/login`
- `/user/logout`
- `/user/create`
- `/user/delete`
- `/form`
- `/form/list`
- `/form/create`
- `/form/delete`
- `/form/submit`
- `/form/responses`
- `/content`
- `/content/list`
- `/content/create`
- `/content/delete`
- `/category`
- `/category/create`
- `/category/delete`
- `/category/reorganize`
- `/report`

Endpoint Tests

To test heroku environment. We have to export the variable otherwise pytest does not pick it up.

```
cd server && python server.py
export TEST_HEROKU='true' && pytest tests
```

To test the local app `pytest tests`

To make the server use Mongo as a server instead of the in-ram database invoke the `server.py` file with proper environment variables set.

```
cd server && USE_MONGO=1 MONGODB_URI=mongodb://127.0.0.1:27017/aang python
server.py
```

Documentation about API

Run `cd server/docs && make html` to build the documentation. It is then available in `server/docs/_build/html/index.html`

Content Reordering

Already uploaded content can be reorganized by uploading an excel file via ADMIN in the format similar to given requirements file.

Server Specifications

The following are the server specs used for the Aanganwadi backend.

- **OS**
  - Ubuntu 16.04.3 LTS
- **CPU**
  - 8 cores
  - 8 threads
- **RAM**
  - 8 GB
- **Storage**
  - 8+20GB (root partition + storage)
- **Bandwidth** (as per lastest speedtest)
  - Download: 866.80 Mbit/s
  - Upload: 841.65 Mbit/s

SETUP

The server can be set up with the following steps.

```
sudo apt install python3-dev python-virtualenv
cd ~/rajasthan
virtualenv -p python3 venv
source venv/bin/activate
pip install -r requirements.txt


# Mongo setup
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
xenial/mongodb-org/3.6 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
org-3.6.list
sudo apt-get update
sudo apt-get install -y mongodb-org
sudo service mongod start
sudo service mongod enable

# nginx
sudo apt install nginx
```

To run the server then we can issue the following commands.

```
cd ~/rajasthan
source venv/bin/activate
cd server && USE_MONGO=1 MONGODB_URI=mongodb://127.0.0.1:27017/aang python
server.py
```

In order to set up an `nginx` proxy we can use the following config in `/etc/nginx/sites-available` and then create a symlink for it in `/etc/nginx/sites-enabled`

```
server {
        client_max_body_size 1G;
    listen 80;

    location / {
        proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;

        proxy_redirect off;
        if (!-f $request_filename) {
            proxy_pass http://127.0.0.1:8000;
            break;
```

```
        }
    }

}
```

The server is now available at `http://<your machine's ip>`

Default demo entries need to be inserted with `python dummy_setup.py`.

The data provided by DoIT&C can be uploaded by using the scripts present at this gist. Please note that the script does not upload MP4 and PNG. Those have to be manually uploaded in their respective categories.

NOTES

- To avoid `https strip` attacks, all endpoints never send sensitive data to the client.
- The server must be set up with `https`
- The server's Admin page contains a brief tutorial of how to use the Admin page.
- Admin setup requires inputs once. After that it works without prompt. On not receiving such input it prceeds with the defaults.