**Project:** Build Web and Mobile application to perform quoting against multiple insurance company rating algorithms and generate Premiums so that the user can compare the premiums from different insurers.

**Technology stack planned to be used:**

|  |  |
| --- | --- |
| Database | postgresql |
| Backend | Django |
| Frontend | Javascript/Javascript Framework(React) |
| mobile | React-native or Flutter |
| web server | nginx |
| WSGI | gunicorn |
| Version control | Git |
| Build and deploy | Docker |

**Installations:**

1. Install python3:
2. Install required libraries:  
   django,

gunicorn,

psycopg2,

1. Install nginx:
2. Install postgresql and pgadmin:
3. create server group and server in pgadmin.
4. Update the details in Django settings to connect to postgreql
5. Install Node.js and NPM

**Sample Project structure:**

A picture containing graphical user interface

Description automatically generated

**Django project setup:**

1. Create a project

– django-admin startproject <projectname>

1. Make the project a git repository and push to github.
2. Create apps

– manage.py startapp <appname>

1. Create the folders as needed and as shown in the project structure.
2. Create superuser

– manage.py createsuper

1. Update project settings to link to postgressql DB

DATABASES = {  
 **'default'**: {  
 **'ENGINE'**: **'django.db.backends.postgresql'**,  
 *# 'NAME': BASE\_DIR / 'db.sqlite3',* **'NAME'** :,  
 **'USER'** :,  
 **'PASSWORD'**:,  
 **'HOST'**:,  
 **'PORT'** :,  
  
 }  
}

1. Migrate admin tables in Django to postgres

– manage.py migrate

1. Start the application

– manange.py runserver

**React App setup:**

1. Create a React app

– npx create-react-app frontend

1. Create react app build

-- npm run build

1. Copy the app into Django project and integrate following
   1. Templates

TEMPLATES = [  
 {  
 **'BACKEND'**: **'django.template.backends.django.DjangoTemplates'**,  
 *# 'DIRS': [BASE\_DIR / 'templates'],* **'DIRS'**: [  
 os.path.join(BASE\_DIR,**'frontend/build'**)  
 ],

* 1. static files.

STATICFILES\_DIRS =[  
 os.path.join(BASE\_DIR,**'frontend/build/static'**),]  
 ]

==================Django application =========

Repository : <https://github.com/harishm11/Applications.git>

* git clone <https://github.com/harishm11/Applications.git>
* cd Applications

Create Virtual Env

* virtualenv myenv

Activate Virtual Env

* source env/bin/activate
* pip install <packages>
* python manage.py makemigrations
* python manage.py migrate
* python manage.py runserver

Ratemanager –

PDF – excel – exhibts and ROC

1. Transformation of exhibits into required format
2. Upload the file with exhibits based on state and company( PDF can different company’s/carriers and states ) – reuse current.
3. Put these exhibits into tables (create the models and push the data) – reuse current
4. We should be able to view these tables – UI

This should have filter by

1. State.
2. Product
3. Exhibits