Installation Guide

This guide shows how the code can be set up to run on a personal computer. The commands shown (in **bold** letters) are specific to the Mac Terminal. The steps will be the same for every computer, but the specific commands may vary slightly. The hosted version of the website can be accessed here.

1.) Open the terminal and create a new project directory and navigate to it.

mkdir radio-recording-project
cd radio-recording-project

2.) Clone the code from GitHub.

git clone https://github.com/lps116/radio_recording_app.git

Alternatively, if receiving the code as a zip file, unzip it and place the content in the current directory. Rename the unzipped folder to **radio_recording_app** if required.

3.) Create a new virtual environment.

python3 -m venv env

4.) Open the file env/bin/activate and add secret keys at the bottom of the file:

export SECRET_KEY="<random 32 character alphanumeric string>"
export DEBUG="True"
export REDIS_URL="redis://localhost:6379"

export RAPID_API_KEY_ONE="<get API key here>"
export RAPID_API_KEY_TWO="<same as RAPID_API_KEY_ONE>"

export AWS_ACCESS_KEY_ID="<get key by making S3 Bucket here>"
export AWS_SECRET_ACCESS_KEY="<add S3 secret access key> "
export AWS_BUCKET_NAME="<add bucket name> "

5.) Install, create and activate PostgreSQL database. Follow installation guide here. It is recommended to download the Postgres.app if using a Mac and the Postico graphical client, found on the same page. The Postico app allows you to create a new database once installed. The database can be started through the Postgres app.

6.) Open the file radio_recording_app/radio_app/settings.py and replace the existing database configuration with:

```
DATABASES = {
   'default': {
      'ENGINE': 'django.db.backends.postgresql_psycopg2',
      'NAME': '<db:name>',
      'USER' : '<username>',
      'PASSWORD' : '<db:password|optional>',
      'HOST' : 'localhost',
      'PORT' : '5432'
   }
}
```

7.) Activate the virtual environment.

source env/bin/activate

8.) Install all the libraries.

cd radio_recording_app
pip3 install -r requirements.txt

9.) Run database migrations.

python manage.py makemigrations
python manage.py migrate

10.) Seed the database.

python manage.py seed

11.) Open a new terminal window and start Redis.

source ../env/bin/activate
redis-server

12.) Open new terminal window and start Celery.

source ../env/bin/activate
celery worker -A radio_app --loglevel=info

13.) Activate server.

python manage.py runserver

14.) Open a browser and go to http://127.0.0.1:8000/

15.) Modify code.

The main files (views, models, templates and urls) which can be modified are summarized in Table C.1. Additional maintenance details can be found in section 6.6 of the dissertation.

16.) To run unit and integration tests run:

python manage.py collectstatic
python manage.py test