

Routes:

/account

Account page route

/createdb

Creates db

/createlisting

HTTP Get and Post method

Post method takes listing information in http headers and images in the message body.

It then stores the images locally and writes all fields to the listing database

HTTP Get method simply returns the createlisting page.

/deletelisting/<listing_id>

Deletes a given user ensuring the user is correct

/deletedb

Deletes db

/getcurrentuser

Returns info on the current user

/listingpics/<path:path>

Returns a given listing image

/getlistings

HTTP Get function that takes fields to query the listing database returns results as json object of listings

/getuserinfo/<user_id>

Returns info on given user and if they are the current user

/getcategories

HTTP Get Method that returns listings

/

Home page route

/listing/<listing_id>

Returns the listings page for a given listing id

/listings

Listings page route

/login

Initial route for login

/redirect

Login Redirect

The backend consists of a quart web app that implements multiple app routes. Each app route runs an associated asynchronous function whenever a user goes to the specified address. Most routes are also login-protected such that users will be redirected to the login screen if they attempt to use a given route without being logged in. The backend also uses the QuartDB driver to transact with the database whenever it is necessary to do so.

To run the backend, you must have PostgreSQL and all python packages installed for python version 12.

You then must configure the environment variables to contain the values for the corresponding Google projects client ID and secret as well as the database password.

Then, you only need to invoke the app.py inside the python environment you configured