Overview:

This application is a service for a web application like reddit. It features a users microservice and a posts microservice. These services operate under the REST API. Service functions include creating accounts, updating emails, incrementing/decrementing account karma, deactivating accounts, creating new posts, retrieving posts, listing recent posts, listing posts within a community, and deleting posts. The project was built using python and is intended to run on a linux platform.

Support Contacts:

Name	**Role**	**Contact**
Francis Nguyen	Application Developer	lqtruongnguyen@csu.fullerton.edu
Kevin Liang	Testing and Automation	kliang38@csu.fullerton.edu
BJ Warfield	Operations	bjwarfield@csu.fullerton.edu

Repository: https://github.com/constraster/CPSC449-Project1

Prerequisites:

Git
Python3
Python3-venv
Gunicorn3
Foreman
Caddy

Installing Prerequisites

In a linux console use the following commands to install our python dependencies

```
$ sudo apt-get install python3 python3-venv
```

To install git::

```
$ sudo apt-get install git-all
```

To install gunicorn:

```
$ sudo apt-get install gunicorn3 --yes
```

To install Foreman

```
$ gem install foreman
```

To install caddy

```
$ curl https://getcaddy.com | bash -s personal
```

Environment Setup:

Clone app repository and navigate to repo directory root

```
$ git clone https://github.com/constraster/CPSC449-Project1
$ cd CPSC449-Project1
```

Create and activate python virtual environment and install app requirements

```
$ python3 -m venv env
$ source env/bin/activate
$ pip3 install -r requirements.txt
```

Run Application:

Run wsgi processes

```
$ foreman start -c users=3,posts=3 -p 8000
```

*runs 3 user process and 3 post processes starting on port 8000

In a new console window, run Caddy reverse-proxy load balancer

```
$ ulimit -n 8192 && caddy
```

Now can access users on http://localhost:5000/v1/api/user
Posts on http://localhost:5000/v1/api/post

Tests:

VALIDATION TESTS:

These tests are used to verify that the service works using curl commands. Verify that the services are available before running tests in a separate terminal instance.

1. Go into: /TestScripts/validation_testing

2. Run test_all.sh to test all scripts, or run test scripts independently (Some commands are dependent on other scripts to run. Ordering defined in test_all.sh)

SYSTEM TESTING:

These tests cover simulated concurrent testing of all services with a heavy user load. We used Locust as our testing framework.

1. Run (if virtual environment not activated): . env/Scripts/activate

2. Go into: /TestScripts/system_testing

3. Run:

```
python3 -m locust --host=http://localhost:5000/v1/api --locustfile
locustfile.py --no-web -c 150 -r 30
```

This runs 150 users, with a given hatch rate of 30

API Routes:

URI	HTTP Method	Expected Response
/v1/api/user/register	POST	201: Creates User 409: Email or username Already Exists
/v1/api/user/add_karma	PUT	202: Added Karma 404: User not found

/v1/api/user/sub_karma	PUT	202: Removed Karma 404: User not found
/v1/api/user/update_email	PUT	202: Email Updated 404: User not Found
/v1/api/user/deactivate_acc/<string:username>	DELETE	202: User Deleted 404: User not Found
/v1/api/posts/make_post	POST	201: Post Created 409: Username does not exist
/v1/api/posts/remove_post/<int:pid>	DELETE	202: Post Deleted 404: Post not found
/v1/api/posts/retrieve_post/<int:pid>	GET	200: post data 404: post not found
/v1/api/posts/list_post_sub/<string:subreddit>/	GET	200: List of post data 404: posts not found
/v1/api/posts/list_all_posts/	GET	200: List of all posts