TaskandBudgetTracking

The task and budget tracking system is a web app for small businesses and non-profits to track employees and volunteers tasks and the organizations spending.

**Architecture**

The application consists of three parts. All three need to be built into a Docker Compose deployment.

React

The Front End is written in React and utilizes React Bootstrap to give a reactive UI.

Flask/PostgreSQL

The Back End is written in Flask and utilizes FlaskSQLAlchemy to store and manage a PostgreSQL database.

It also stores images in the filesystem of its container.

Nginx

The application uses an Nginx proxy server to handle directing of requests and encryption.

**Setup**

You will need docker, docker compose (these come bundled together if you are using docker desktop), and git installed in order to follow these instructions.

In order to get started with this application pull a clone a of the repository to your local environment.

Create your config files

sample files have been provided for ease of quick setup in their respective files.

1. Create a docker-compose.yml file following the instructions in docker-compose.yml

2. If you are deploying to a production environment and planning on using an ssl certificate create a nginx.prod.conf file from the nginx.prod.conf.patern file in the Nginx folder.

Build and Start the application

1. Make sure docker is installed.

2. Build the images. (if the front end fails to build run “npm install” in your command terminal before building

3. Then run the docker compose file from the root folder of the project with `docker-compose up`

Run the application tests

This process will clear any data in your local container volumes

1. Rename the docker-compose.test.yml docker-compose.yml

2. Build the compose file

3. Run the compose file with `docker-compose up --abort-on-container-exit`

These commands will produce a testing.txt file and a htmlcov folder in the flask folder the results of the tests can be viewed in the testing.txt file and the coverage report can be viewed by opening intex.html in the htmlcov folder. After you are done change the file names on the docker-compose.yml files. Make sure you rebuild the images before continuing.