DataSpark Instruction Manual

(Web Based Data Science Platform)

Technologies Used in the Application

Front-end

1. React

Back-end

1. Flask
2. MySQL

Deployment

1. Heroku
2. Firebase
3. FreeSql

Setup Instructions for Executing Web Application on Local Server

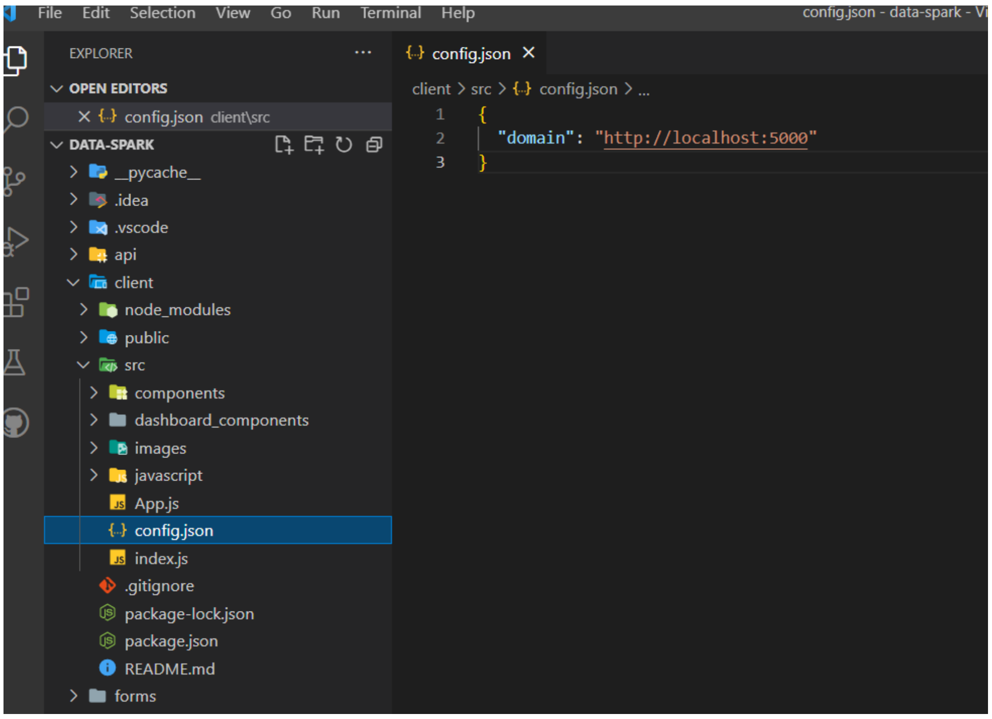
1. Extract the contents of the provided compressed (Zip) file, "`Assignment5_Group20`" into the desired location.
2. The `data-spark` folder inside contains the source code.

   Alternatively, the source code of the project can also be obtained by cloning the project from GitHub using the following URL.
   https://github.com/akashTharuka/data-spark.git

   This alternate step can be done by entering the following command in the command prompt. This will create a data-spark folder and clone the source code into that folder.

   ```
   \>git clone https://github.com/akashTharuka/data-spark.git
   ```

3. By default, the flask app runs at port 5000. Thus, 'domain' in the `config.json` file in the `data-spark/client/src` folder has been set as "`http://localhost:5000`", as indicated below. If you wish to change the port for back-end, make sure to apply the change in this file.



```
{
  "domain": "http://localhost:5000"
}
```

To get the Front-end running :-

4. Open command prompt in the `data-spark` folder (or open the `data-spark` folder in an IDE (e.g.: - Visual Studio Code) and open its terminal).
5. Then move into the `client` directory, using the following command.

```
\data-spark>cd client
```

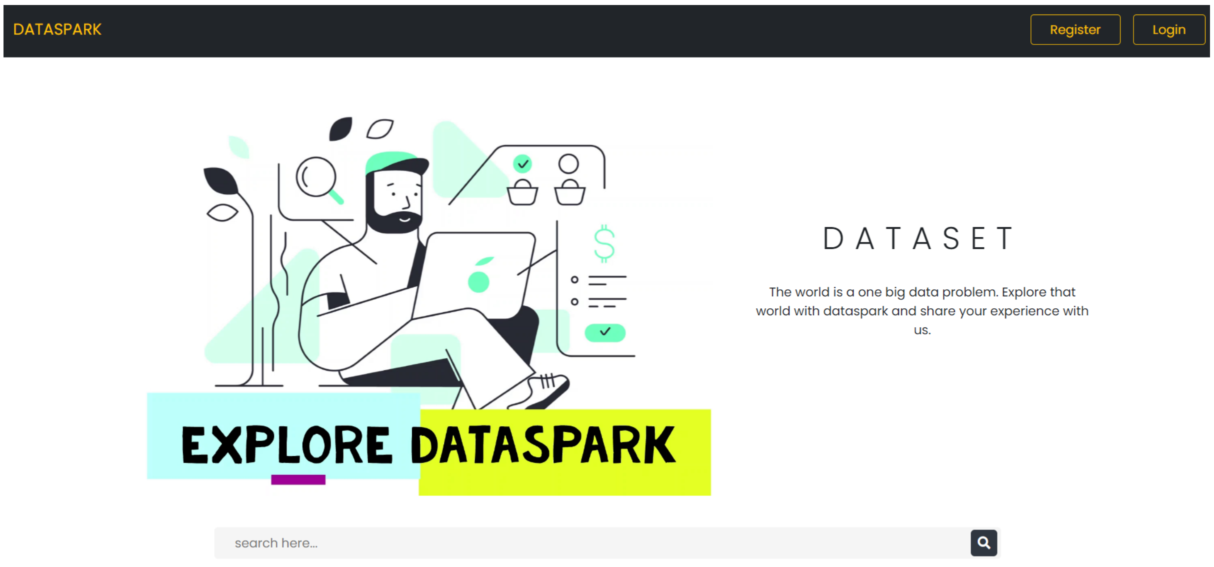
6. Use the following command to download dependencies defined in `package.json` file and generate node_modules folder with the installed modules.

```
\data-spark\client>npm install
```

7. To execute the scripts, enter the command given below.

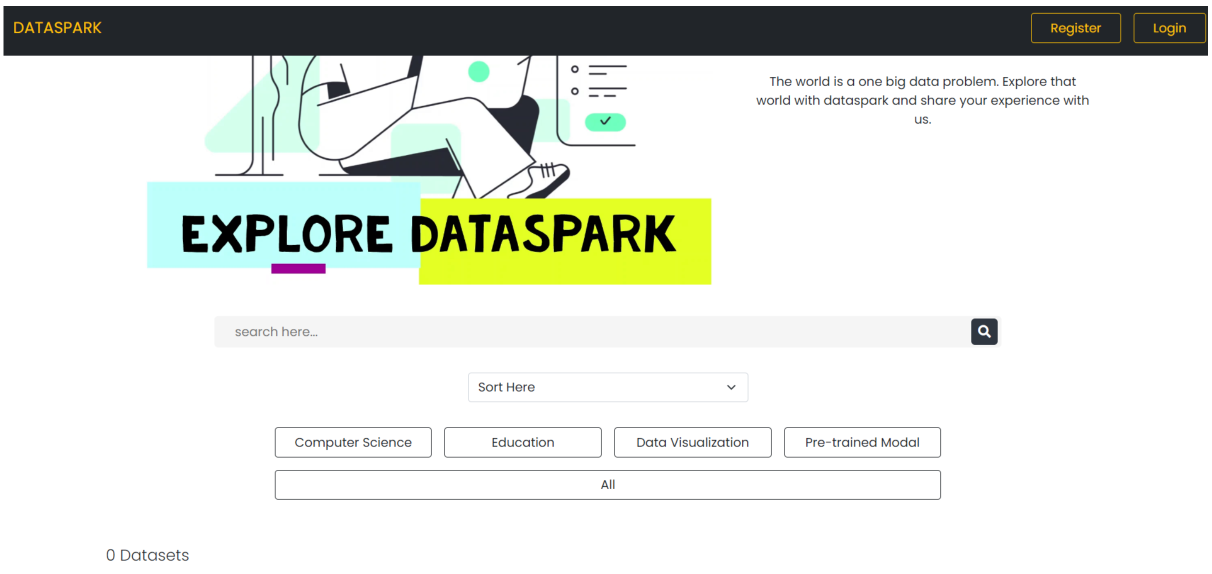
```
\data-spark\client>npm run start
```

This will open the front-end of the application in your web browser, as indicated in the figure below. (However, please note that this might produce some errors (e.g.: - Axios error, network error), since the back-end hasn't been set up yet)



DATASET

The world is a one big data problem. Explore that world with dataspark and share your experience with us.



The world is a one big data problem. Explore that world with dataspark and share your experience with us.

0 Datasets

To get the Back-end running :-

8. Without closing the terminal, open a new terminal in the `data-spark` folder.
9. Using the commands below, create an isolated virtual python environment.

```
\data-spark>py -3 -m pip install virtualenv
```

```
\data-spark>py -3 -m venv venv
```

10. Activate the scripts to use the virtual environment via the following command.

```
\data-spark>venv\Scripts\activate
```

    Please note that depending on the Python version on your machine, sometimes `pip3 install <dependency>` will work rather than `pip install <dependency>`. Thus give the commands involving pip accordingly.

11. Wheel is a built-package format that offers the advantage of not recompiling the software during every install. Install this package using the command below.

```
(venv) E:\data-spark>pip install wheel
```

12. All the dependencies required for the execution of the application have been included in the `requirements.txt` file in the `data-spark` folder. To install all those dependencies, simply use the command stated below.

```
(venv) E:\data-spark>pip install -r requirements.txt
```

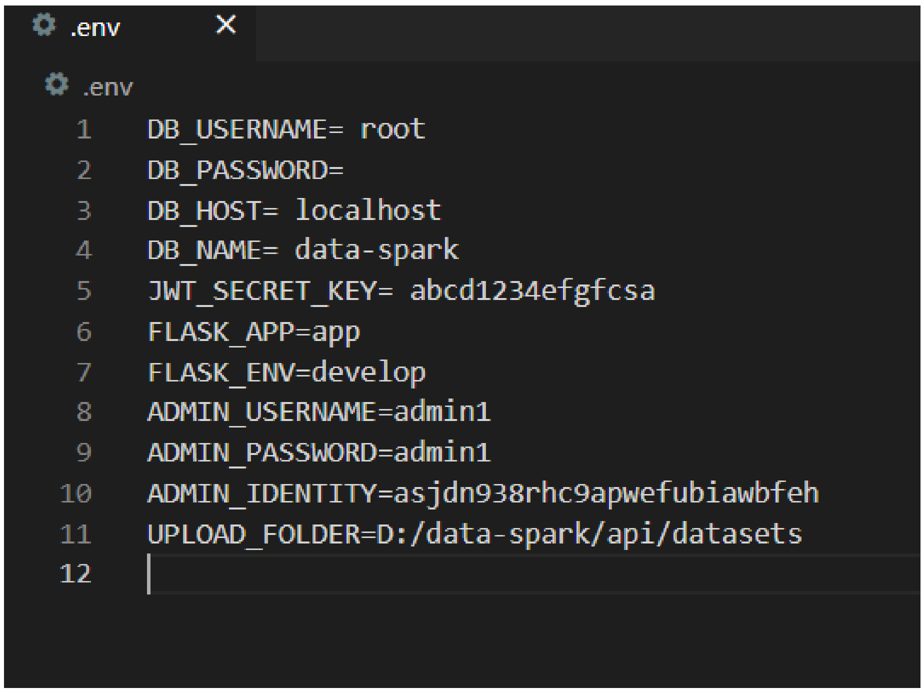
Setting up the Database :-

13. A sql file `data-spark.sql` is included in the “`Assignment5_Group20`” folder.
14. Open a software tool that can handle the administration of MySQL (e.g.: - phpMyAdmin, MYSQL Workbench).
15. Create a new database schema named ‘data-spark’.
16. Import the `data-spark.sql` file into the database created in the previous step.

Setting the Environmental Variables :-

17. Get a copy of the `.env.example` file into the `data-spark` folder and rename it as `.env` file.
18. To set the environmental variables, enter your database username, password and host details. The database name is data-spark (as per the one created above). JWT secret key, Admin username, Admin password and Admin identity can be set as you desire. Enter ‘app’ and ‘develop’ for FLASK_APP and FLASK_ENV respectively. The upload folder is where the uploaded datasets will be stored. Thus, choose a suitable

location for this purpose and enter the path of this folder to set that variable. An example of the `.env` file is shown below.



```
DB_USERNAME= root
DB_PASSWORD=
DB_HOST= localhost
DB_NAME= data-spark
JWT_SECRET_KEY= abcd1234efgfcsa
FLASK_APP=app
FLASK_ENV=develop
ADMIN_USERNAME=admin1
ADMIN_PASSWORD=admin1
ADMIN_IDENTITY=asjdn938rhc9apwefubiawbfeh
UPLOAD_FOLDER=D:/data-spark/api/datasets
```

19. Save the changes made to the file.

Running the Application :-

20. Go back to the terminal that was started for the back-end part and run the following command.

```
(venv) E:\data-spark>python app.py
```

21. When running the application, if you encounter any ModuleNotFound errors, open a new command prompt (as administrator) and enter the following commands.

```
pip install Flask Flask-RESTful Flask-Cors Flask-JWT-Extended
pandas matplotlib numpy seaborn

pip install python-dotenv Flask-Login Flask-SQLAlchemy
PyMySQL
```

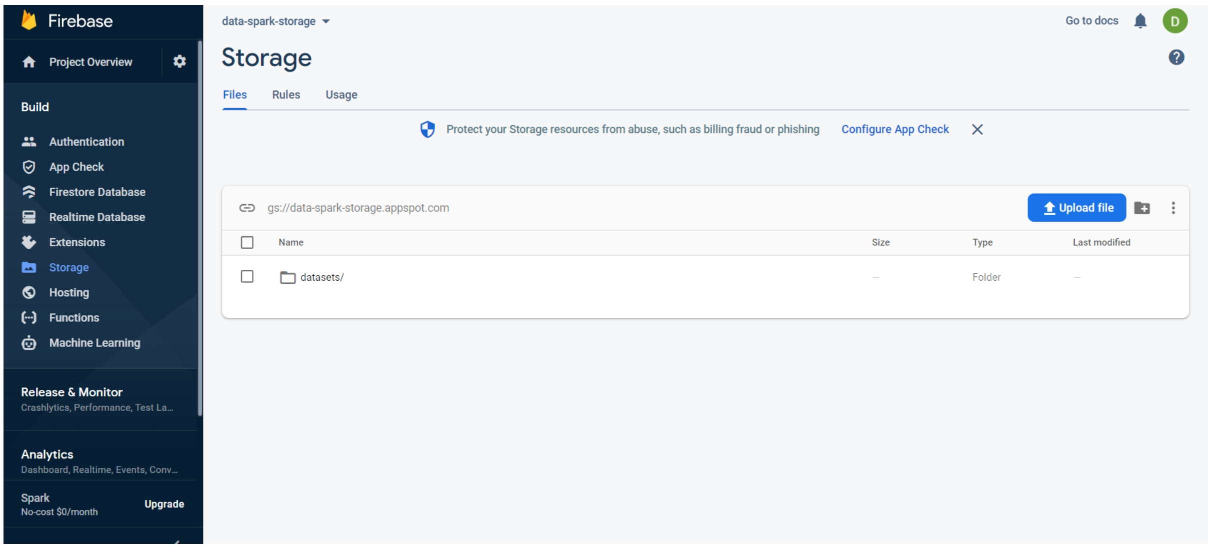
```
pip install Flask Flask-RESTful Flask-Cors Flask-JWT-Extended pandas matplotlib numpy seaborn
```

```
pip3 install python-dotenv Flask-Login Flask-SQLAlchemy PyMySQL
```

Heroku Deployment

Method - Using Heroku CLI

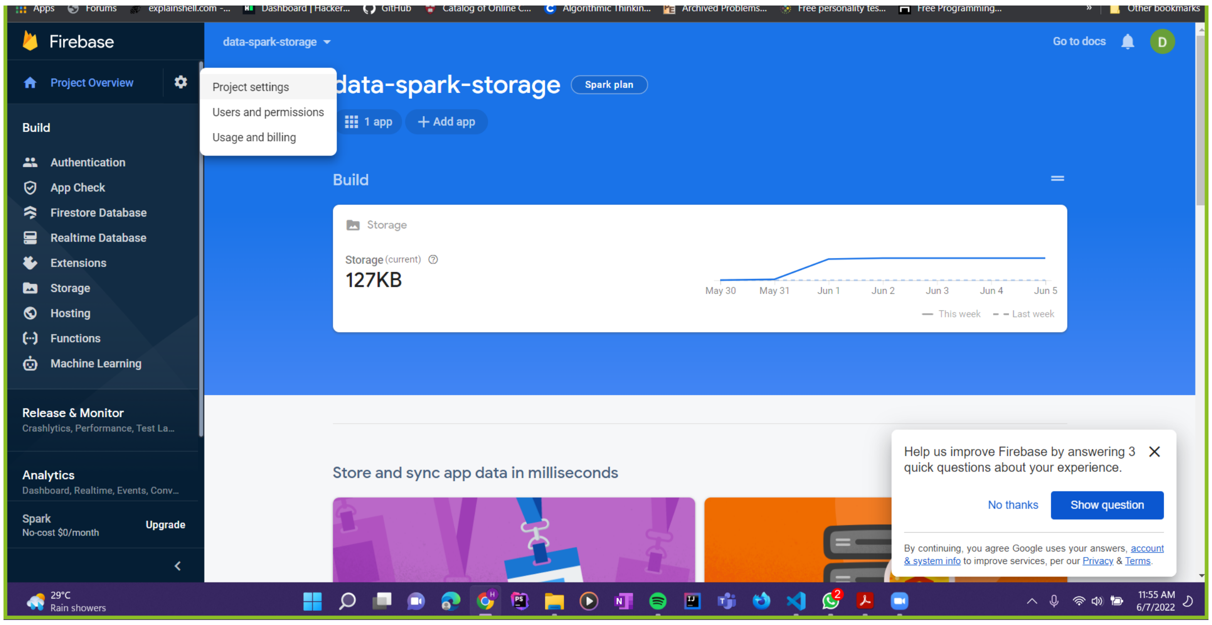
1. First Create an account in https://www.freesqldatabase.com/, if you don't have one already.
2. Then create a database (the database name and credentials will be automatically provided via an email)
3. Then import the sql file "`data-spark-deploy.sql`", which is available in the "`Assignment5_Group20`" folder into the database created above.
4. Create a Firebase web app and create a folder named datasets in storage.



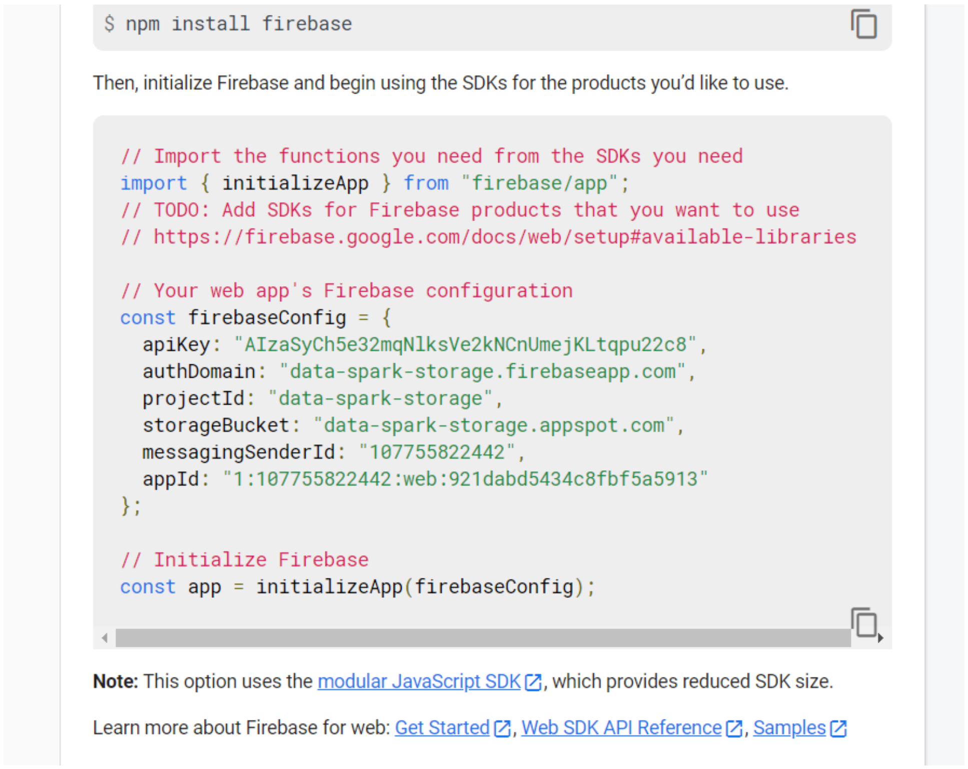
gs://data-spark-storage.appspot.com

Name	Size	Type	Last modified
datasets/	—	Folder	—

5. Then navigate to the Project settings from the menu indicated in the screenshot below.



6. Then obtain the app configurations from the General section of the Project Settings.



```
$ npm install firebase
```

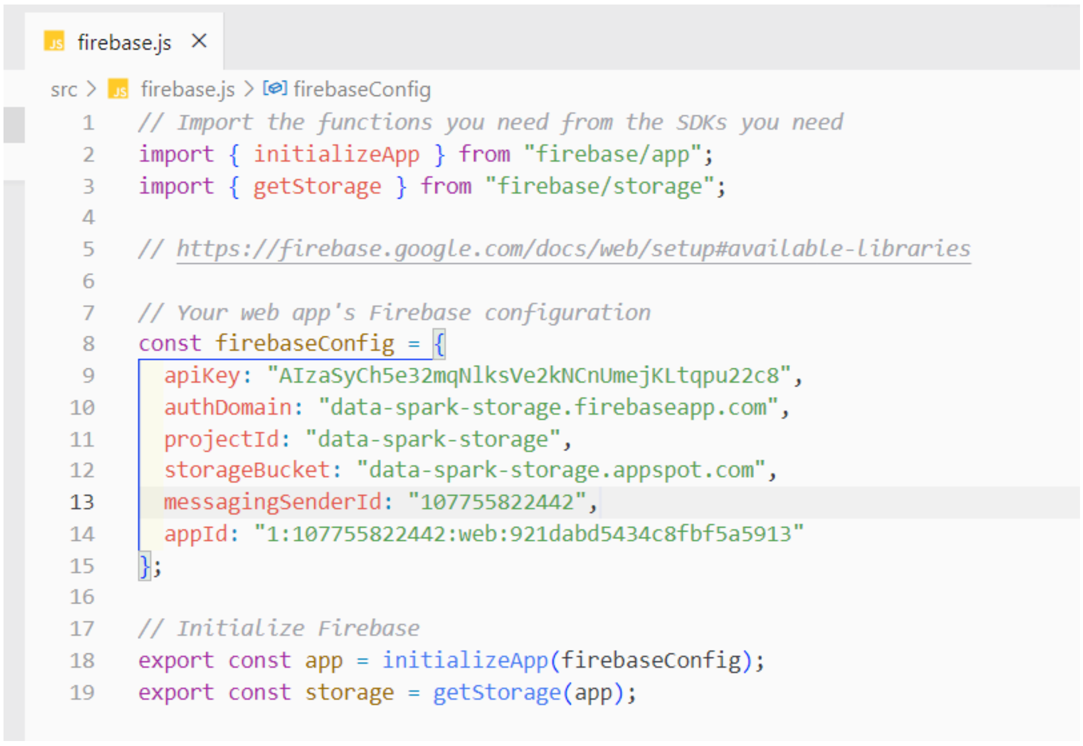
Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCh5e32mqNlksVe2kNCnUmejKLtqpu22c8",
  authDomain: "data-spark-storage.firebaseapp.com",
  projectId: "data-spark-storage",
  storageBucket: "data-spark-storage.appspot.com",
  messagingSenderId: "107755822442",
  appId: "1:107755822442:web:921dabd5434c8fbf5a5913"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

7. Then add those configurations to `firebase.js` (`data-spark-client-deploy/src/firebase.js`) file in the codebase of the client.



firebase.js

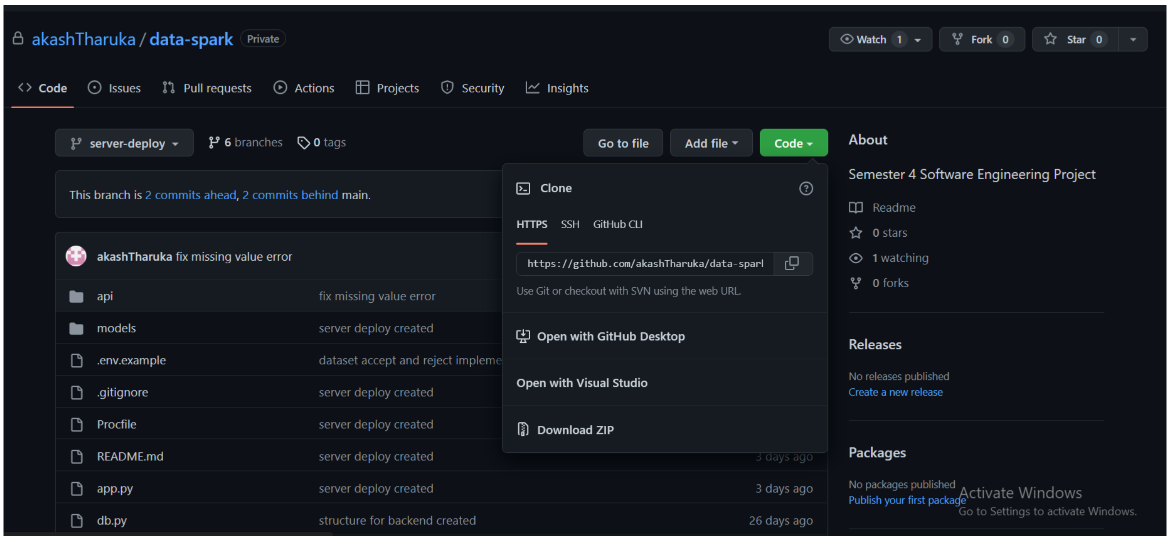
src > firebase.js > firebaseConfig

```javascript
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getStorage } from "firebase/storage";

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCh5e32mqNlksVe2kNCnUmejKLtqpu22c8",
  authDomain: "data-spark-storage.firebaseapp.com",
  projectId: "data-spark-storage",
  storageBucket: "data-spark-storage.appspot.com",
  messagingSenderId: "107755822442",
  appId: "1:107755822442:web:921dabd5434c8fbf5a5913"
};

// Initialize Firebase
export const app = initializeApp(firebaseConfig);
export const storage = getStorage(app);
```

8. Install Heroku CLI in your machine. For more information refer https://devcenter.heroku.com/articles/heroku-cli

Hosting server application(flask app) in Heroku

9. Visit Heroku dashboard and create a new app for the server. If you don't have an account in heroku make sure to create one.
10. Use the following link to access the server-deploy branch of the GitHub repository. https://github.com/akashTharuka/data-spark/tree/server-deploy
11. Then download the zip code of this branch, as shown in the screenshot below.



akashTharuka / data-spark Private

Watch 1 · Fork 0 · Star 0

Code · Issues · Pull requests · Actions · Projects · Security · Insights

server-deploy · 6 branches · 0 tags · Go to file · Add file · Code

This branch is 2 commits ahead, 2 commits behind main.

akashTharuka fix missing value error

api	fix missing value error	
models	server deploy created	
.env.example	dataset accept and reject impleme	
.gitignore	server deploy created	
Procfile	server deploy created	
README.md	server deploy created	3 days ago
app.py	server deploy created	3 days ago
db.py	structure for backend created	26 days ago

Clone

HTTPS · SSH · GitHub CLI

https://github.com/akashTharuka/data-sparl

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Open with Visual Studio

Download ZIP

About

Semester 4 Software Engineering Project

Readme · 0 stars · 1 watching · 0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Activate Windows
Go to Settings to activate Windows.

12. Extract the zip file and go to the root directory, "`data-spark-server-deploy`".
13. Create a file named runtime.txt in the root folder and paste the following.

```
python-3.8.5
```

14. Install gunicorn using the following command.

    ```
    $ pip install gunicorn
    ```

15. Create a file named Procfile in the root directory. And paste the following line.

    ```
    web: gunicorn app:app
    ```

16. Then follow the following steps in the command line to deploy.
    a. Login to heroku account

       ```
       $ heroku login
       ```

    b. Initialise a git repository (Make sure you are in the root directory, “`data-spark-server-deploy`”)

       ```
       $ git init
       $ heroku git:remote -a <name of app created for server>
       ```

    c. Add a change in the working directory to the staging area using git

       ```
       $ git add .
       ```

    d. Save all staged changes using git commit.

       ```
       git commit -am "make it better"
       ```

    e. Push changes to heroku.

       ```
       git push heroku master
       ```

17. If you have successfully completed step 5, you can continue with step 7.
18. Visit Settings sections of your heroku app and put the following environment variables with relevant values in the Config vars section.

    ```
    ADMIN_IDENTITY

    ADMIN_PASSWORD

    ADMIN_USERNAME

    DB_HOST

    DB_NAME

    DB_PASSWORD

    DB_USERNAME

    JWT_SECRET_KEY
    ```

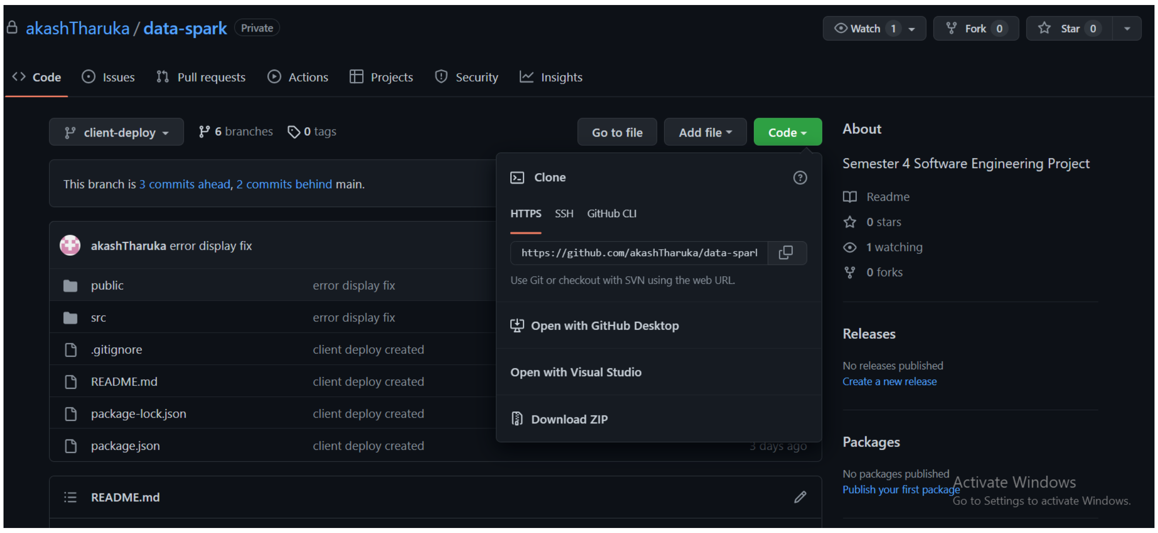
19. Push changes to heroku repository from your local repository.

    ```
    git push heroku master
    ```

Hosting client application(react app) in Heroku

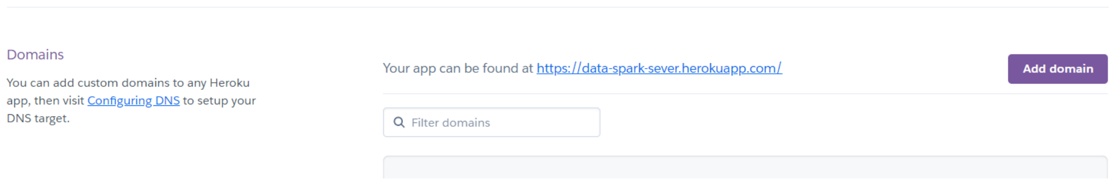
20. Visit Heroku dashboard and create a new app for the client.
21. Use the following link to access the client-deploy branch of the GitHub repository. https://github.com/akashTharuka/data-spark/tree/client-deploy

22. Then download the zip code of this branch, as shown in the screenshot below.



23. Extract the zip file and go to the root directory, “`data-spark-client-deploy`”.

24. Get the url for server application from the settings of server application as shown below.

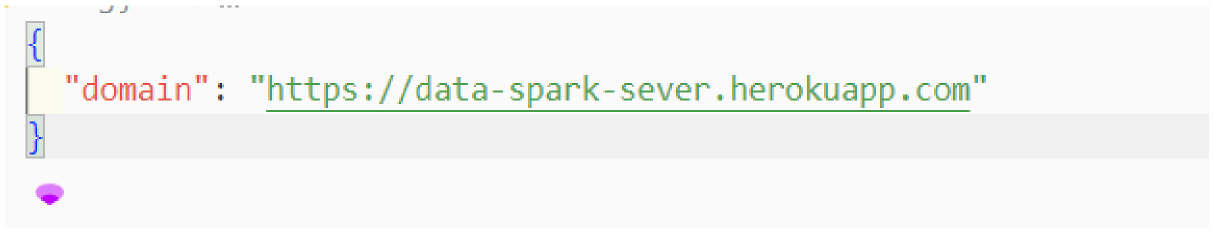


Domains

You can add custom domains to any Heroku app, then visit Configuring DNS to setup your DNS target.

Your app can be found at https://data-spark-sever.herokuapp.com/

And change the domain in the config.json file in the `data-spark-client-deploy/src` folder to that url as shown below.



```
{
  "domain": "https://data-spark-sever.herokuapp.com"
}
```

25. Then follow the following steps in the command line to deploy.
    a. Login to heroku account

    ```
    $ heroku login
    ```

    b. Initialise a git repository (Make sure you are in the root directory, “`data-spark-client-deploy`”)

    ```
    $ git init
    $ heroku git:remote -a <name of app created for client>
    $ heroku buildpacks:set mars/create-react-app
    ```

    c. Add a change in the working directory to the staging area using git

    ```
    $ git add .
    ```

    d. Save all staged changes using git commit.

    ```
    git commit -am "make it better”
    ```

    f. Push changes to heroku.

    ```
    git push heroku master
    ```