Data classification

Documentation

CMPG 324

TSHWEU SEPHIRI

31255086

How It Works

1. On the home page user will either sign-up or login

Graphical user interface, application, Teams

Description automatically generated

1. If new user check 2.1 or if returning user check 2.2
   1. User clicks on sign-up button
      1. A form to sign-up will be displayed.

Graphical user interface, application

Description automatically generated

* + 1. the user will need to complete their email address.
    2. the user will need to complete their password.
    3. Then click on register button.
    4. if no error occur the login form will be display and you may proceed to 2.2
  1. User clicks on login/ is redirected after signing up.

Graphical user interface, application

Description automatically generated

* + 1. The user will need to complete their email address.
    2. The user will need to complete their password.
    3. Click on login button.
    4. If email invalid check error and retry.
    5. If password doesn’t match check caps on your keyboard and try again.

1. The first thing the user will see is the file upload page

Graphical user interface, application, Teams

Description automatically generated

3.1 The user should select a file type.[Text or excel]

3.2 Click the browse button to select a file to upload.

3.3 click upload button

Graphical user interface, application, Teams

Description automatically generated

3.4 if you uploaded an Excel file click select sheet button to select a sheet to retrieve data from and submit.

Graphical user interface, application

Description automatically generated

1. The data classified is then shown in the next screen

Graphical user interface, application

Description automatically generated

File Formats Supported

* txt files
* Excel Files

**Technologies Used**

Front-end: Angular

The font-end was developed in angular because of my prior experience with it and the ease-of-development. It provided simple functionality to upload data, authenticate users by using the users local storage to store a token, communicate with server by utilizing services, preventing unauthorized access to specific pages using guards. Material design bootstrap along with ordinary bootstrap both are supported in angular as well making designing the website easier while still providing a professional GUI.

Back-end: NodeJS

The biggest reason this was chosen was that I have prior experience with NodeJS. That along with the large number of resources, documentation, and libraries makes it extremely accessible and efficient. Libraries such as express

Database: MongoDB

The fields that need to be store vary widely because of this a document base database(NoSQL) like mongo dB that is flexible was chosen. This database is easy to use and with companies like Atlas you can host your databases on their free tier.

Hosting: Heroku

Heroku provides a good CD environment with even the initial deployment being very easy. It links to your GitHub repository and every time you push to the specified branch it automatically deploys your changes.