Design Document

Version 1.0

1 Frontend web application design

1.1 Application description

The purpose of this frontend web application is to have a webpage that will access and use the Eleox API. This application will be able to render and display a list of users that the server sends when making a request via HTTP. It will also be able to add and remove users from the server. The application also is able to perform protected routes and handle any errors without disrupting the user.

1.1.1 Key technologies

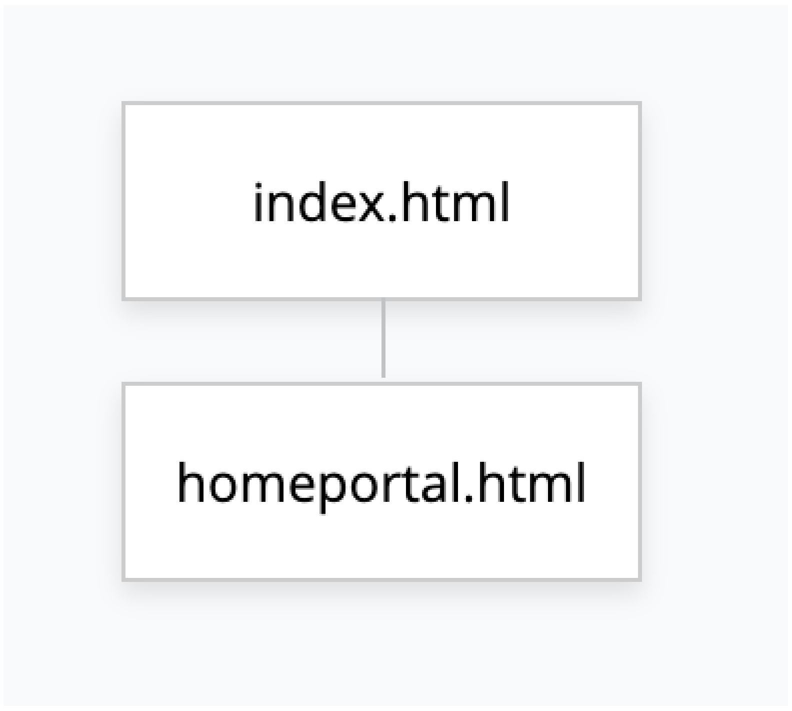
HTML is used to organise content on the webpage. CSS is used to give style to content on the webpage. Javascript is used to create a local webserver on the machine and update/change content to the webpage.

1.2 Authentication and authorisation

To access the main contents of the web application, you must have a valid login credential. The Eleox API uses a Bearer Authentication HTTP scheme that returns an authentication token that will be necessary when performing other API related requests.

2 Service specification

2.1 Sitemap

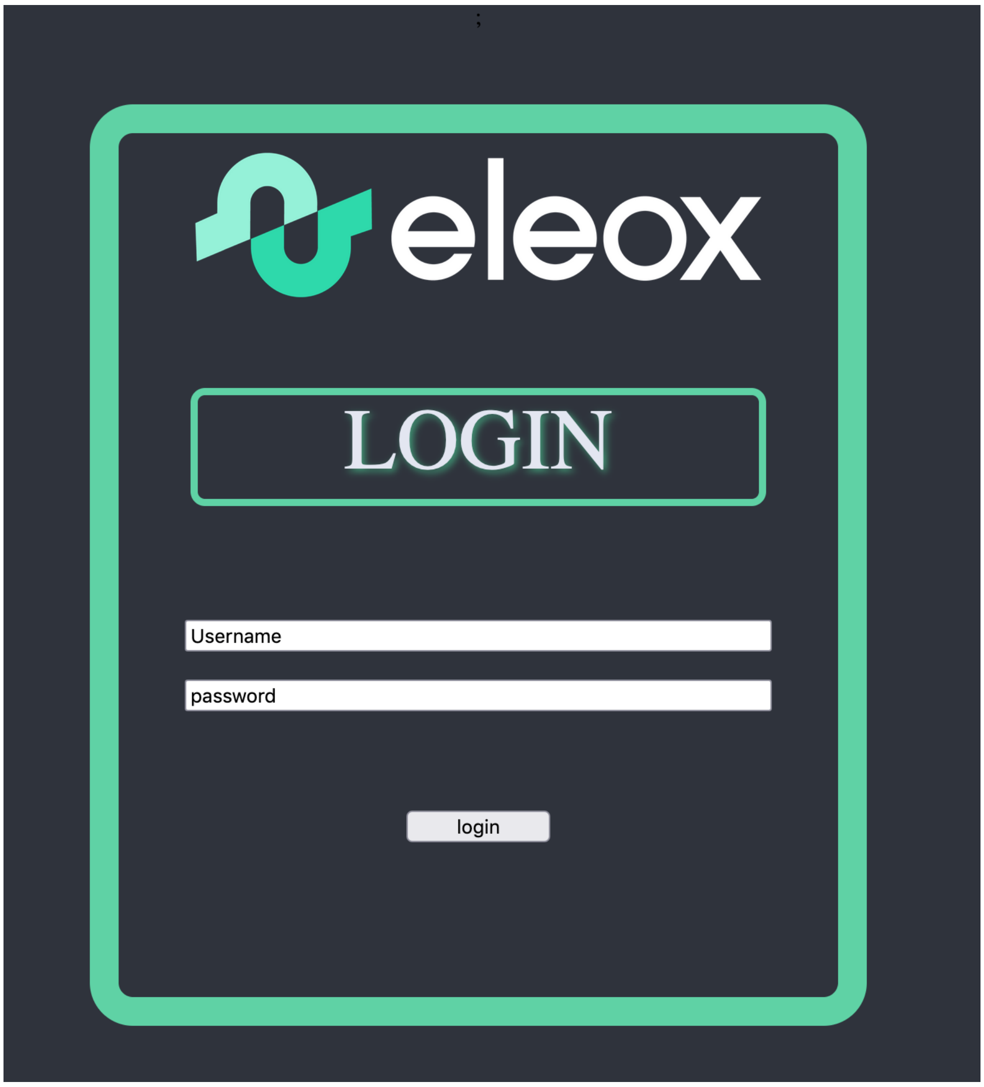


Although the sitemap looks relatively simple, most of the functionality is contained on the homeportal.html. Index.html is used as a “Sign-In” page to allow access to the homeportal.html page.

2.2 Sitemap In Depth

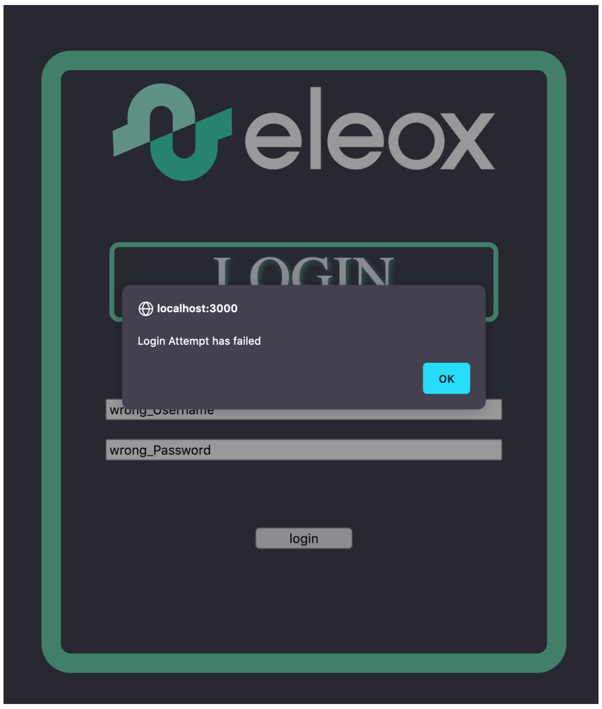
This section will go more in depth about the markup and style choices of each webpage. Followed by what functionality can be done.

2.3 index.html



Upon navigating to "localhost:3000" the user is prompted with a login screen. With valid credentials the user is able to bypass the login screen and go directly into the homeportal page.

On the javascript side, an eventListener function is added to the "login" button element retrieved by traversing the DOM. The function first extracts the values from the form and stores the "username" value in the browser storage. The purpose of this is to have an easy and secure access to that value that will be used later on. After that has been done, the Fetch API is used to send a request to the Elenox server to validate the inputted credentials and return an authorization token if valid. The authorization token is stored in the web browser as well. If the credentials are invalid, then a warning is displayed to the users; as shown below.



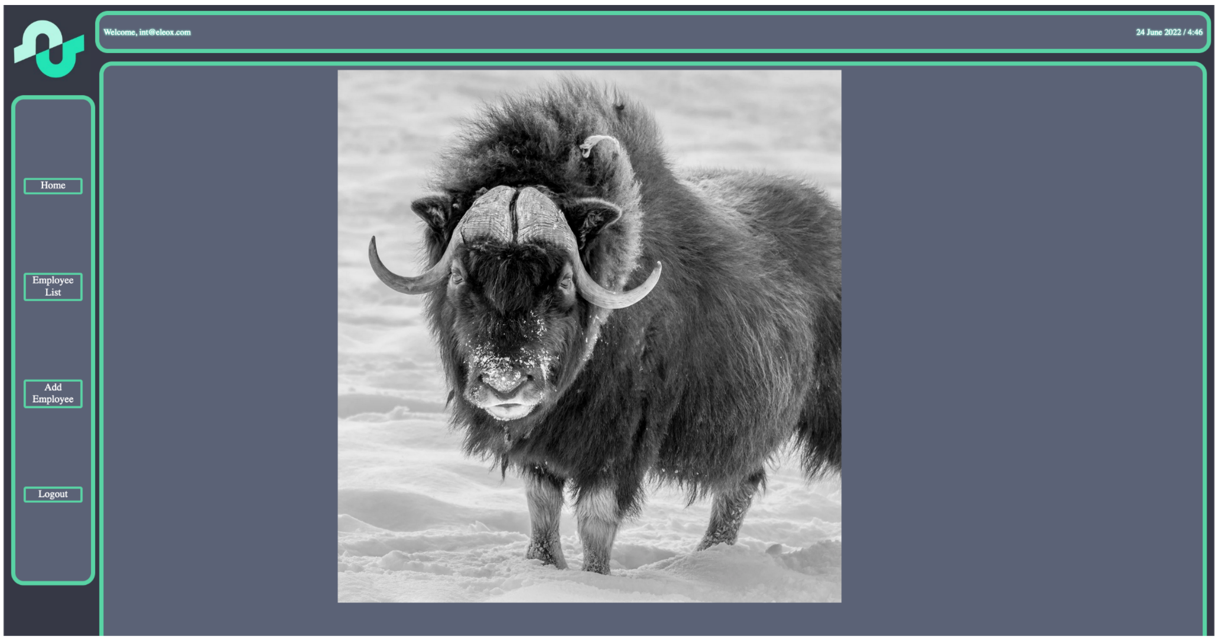
localhost:3000

Login Attempt has failed

wrong_Username

wrong_Password

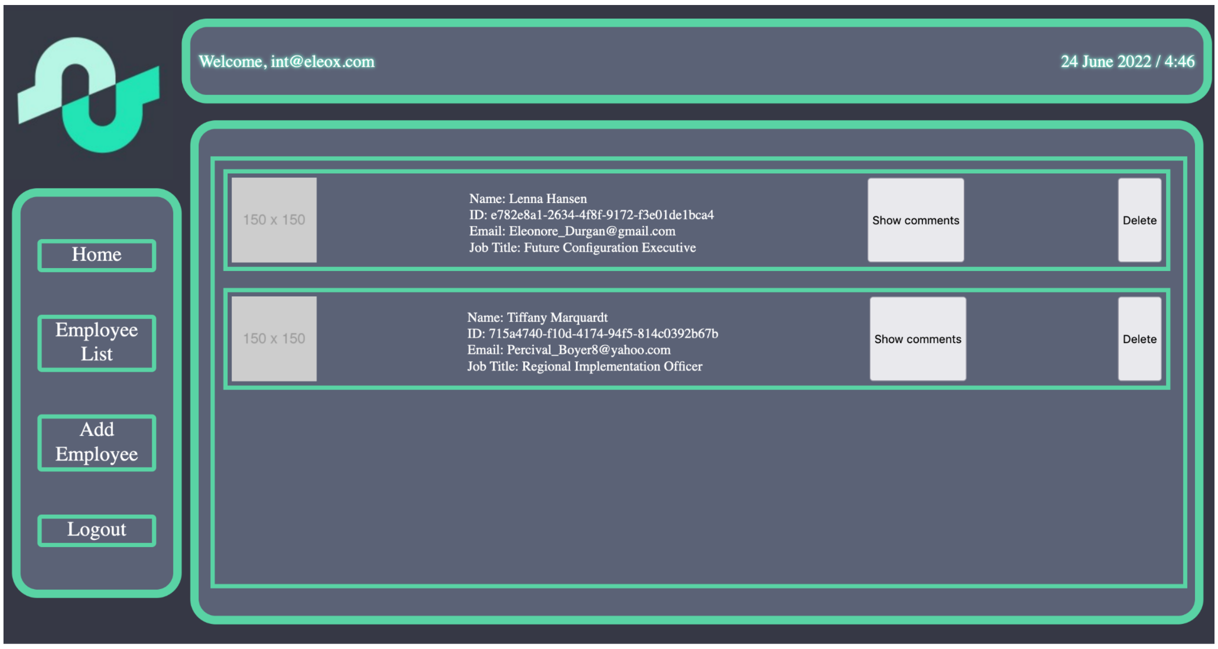
2.4 homeportal.html



Welcome, int@eleox.com

24 June 2022 / 4:46

Upon a successful login, the user is taken to the “Home Portal” page where the user can do many things. Clicking the “Home” button will return you to the home portal page. Clicking the “Logout” button will clear all the saved information in the browser and return the user to the login page(index.html). CSS Grid was used to help separate content based on “header'', “sidebar”, and “main” sections; with CSS Flexbox was used to position content in an orderly manner. The header contains a welcome message that displays the username retrieved from the browser storage, with a time and date pertaining to the local area of the user. The sidebar contains various links that are styled as buttons that the user can press. Upon clicking one of the “buttons”, content on the main section is cleared and redrawn to display new information.



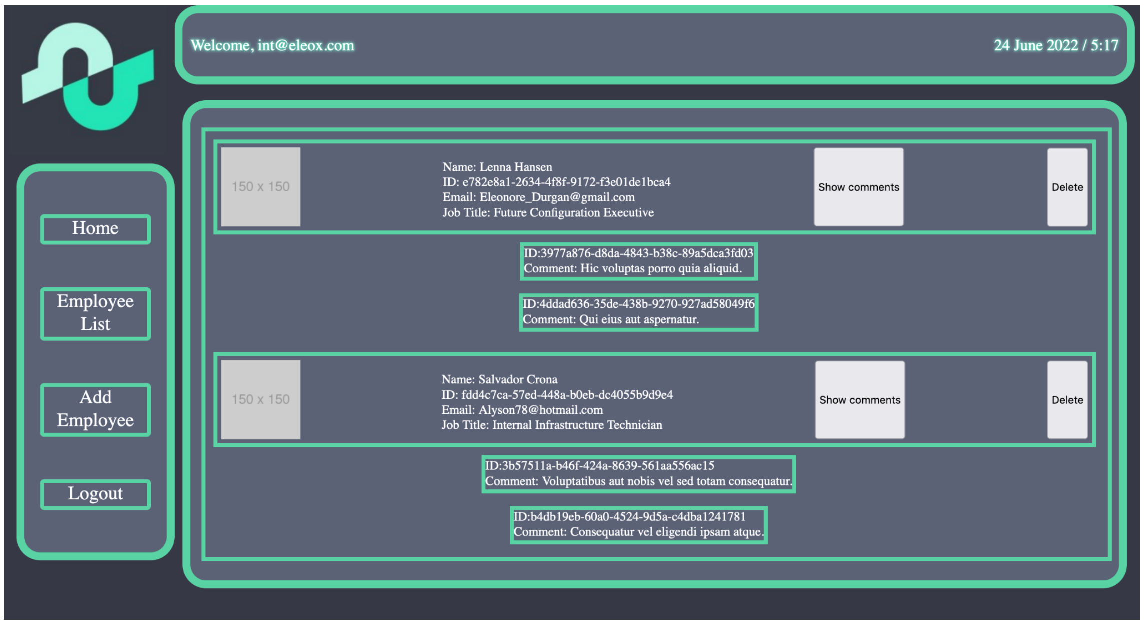
Welcome, int@eleox.com

24 June 2022 / 4:46

Name: Lenna Hansen
ID: e782e8a1-2634-4f8f-9172-f3e01de1bca4
Email: Eleonore_Durgan@gmail.com
Job Title: Future Configuration Executive

Name: Tiffany Marquardt
ID: 715a4740-f10d-4174-94f5-814c0392b67b
Email: Percival_Boyer8@yahoo.com
Job Title: Regional Implementation Officer

Upon clicking on “Employee List”, the main section is cleared and populated with a list of employees. In the code, a Fetch API request is made to the Eleox servers. When the request is returned and extracted from its JSON format, I created DOM elements that were populated with the returned data and added it to the main area for display.



Welcome, int@eleox.com

24 June 2022 / 5:17

Name: Lenna Hansen
ID: e782e8a1-2634-4f8f-9172-f3e01de1bca4
Email: Eleonore_Durgan@gmail.com
Job Title: Future Configuration Executive

ID:3977a876-d8da-4843-b38c-89a5dca3fd03
Comment: Hic voluptas porro quia aliquid.

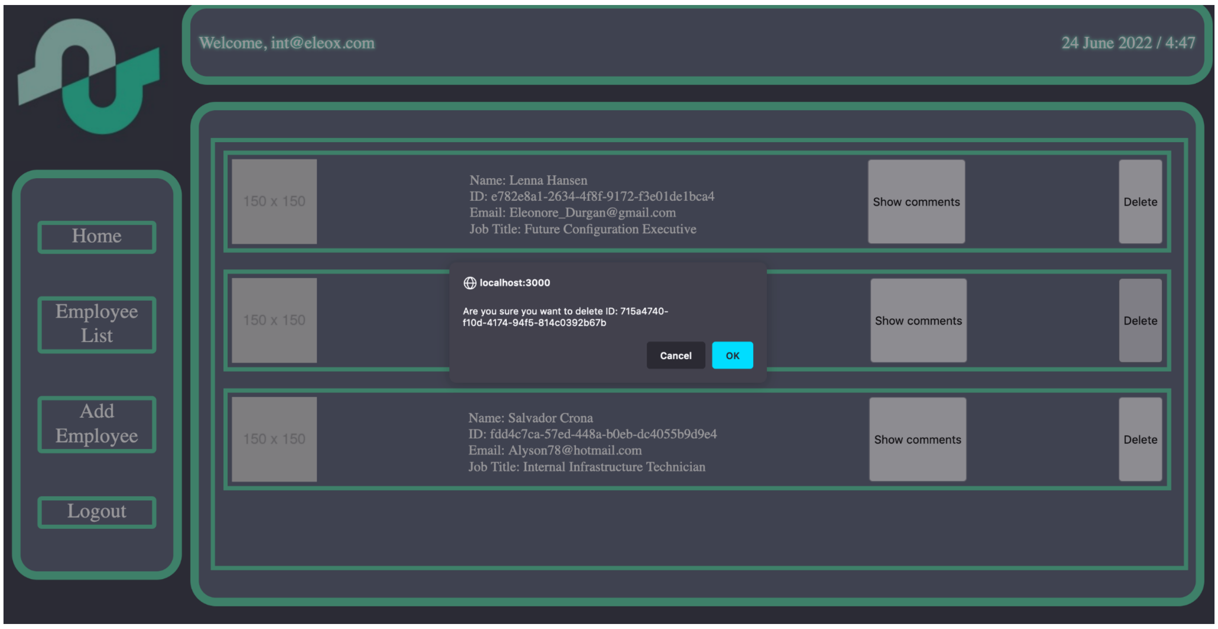
ID:4ddad636-35de-438b-9270-927ad58049f6
Comment: Qui eius aut aspernatur.

Name: Salvador Crona
ID: fdd4c7ca-57ed-448a-b0eb-dc4055b9d9e4
Email: Alyson78@hotmail.com
Job Title: Internal Infrastructure Technician

ID:3b57511a-b46f-424a-8639-561aa556ac15
Comment: Voluptatibus aut nobis vel sed totam consequatur.

ID:b4db19eb-60a0-4524-9d5a-c4dba1241781
Comment: Consequatur vel eligendi ipsam atque.

The user is able to press the "Show comments" button and populate the main area with comments under the specified user.


Welcome, int@eleox.com

24 June 2022 / 4:47

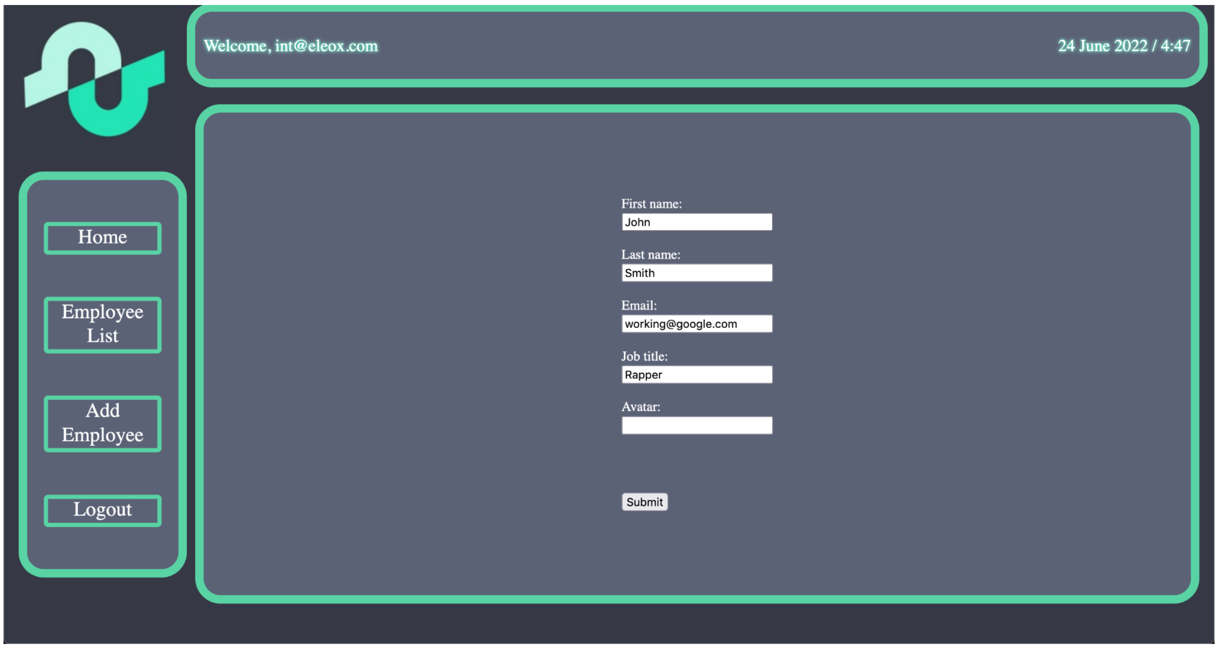
Name: Lenna Hansen
ID: e782e8a1-2634-4f8f-9172-f3e01de1bca4
Email: Eleonore_Durgan@gmail.com
Job Title: Future Configuration Executive

localhost:3000

Are you sure you want to delete ID: 715a4740-f10d-4174-94f5-814c0392b67b

Name: Salvador Crona
ID: fdd4c7ca-57ed-448a-b0eb-dc4055b9d9e4
Email: Alyson78@hotmail.com
Job Title: Internal Infrastructure Technician

The user is also able to delete employees from the list, and upon deletion the screen is cleared and redrawn to accurately display the list of employees.


Welcome, int@eleox.com

24 June 2022 / 4:47

First name: John

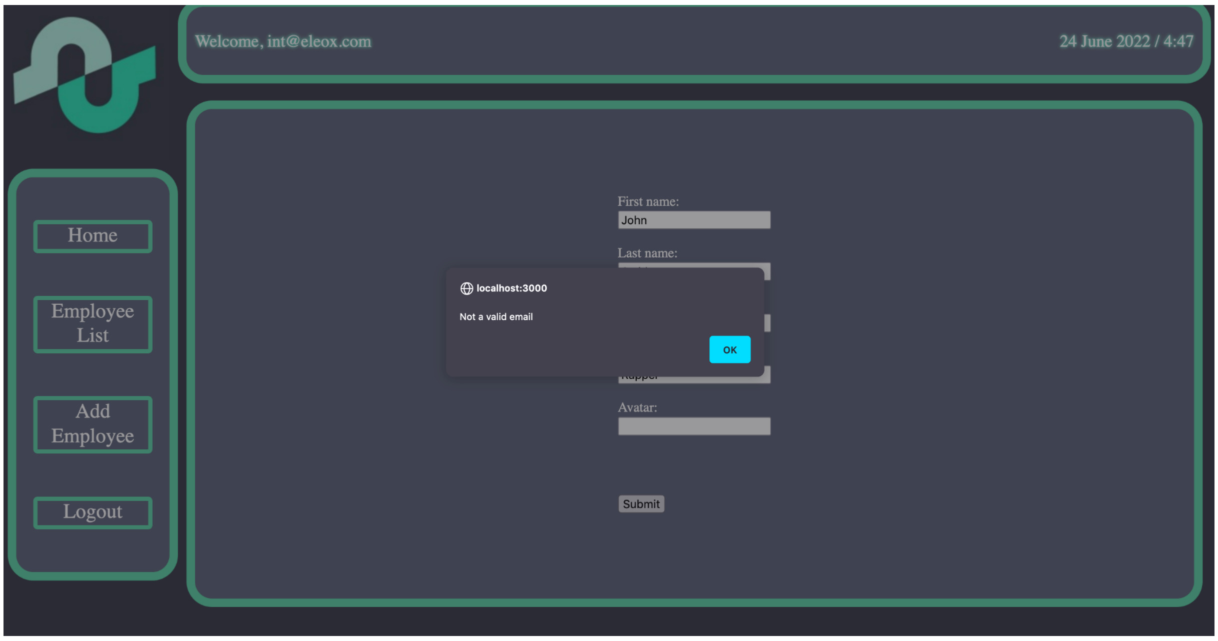
Last name: Smith

Email: working@google.com

Job title: Rapper

Avatar:

Upon hitting the "Add Employee" button, the user is brought to a form where they can enter information about a new employee. This is done by setting an eventListener function to the element object of the "Submit" button. Upon being clicked, the application will extract the values from the form and submit a Fetch API request to the Eleox server.


Welcome, int@eleox.com

24 June 2022 / 4:47

First name:

John

Last name:

Avatar:

localhost:3000

Not a valid email

The application also has the ability to handle any input error, such as bad email response.

3 Service operation

3.1 Startup and shutdown steps

3.1.1 Starting Up

This application REQUIRES NodeJS 18.4.0. This requirement is used to start a local webserver on the user's machine.

To start up, navigate to the directory containing server.js and turn the command in a terminal: "node server.js". To access the web application, open up a browser (highly recommend Firefox or Chrome) and enter in the url "localhost:3000" as the server is run on that part.

3.1.2 Shutting Down

To shut down, just close the terminal.