Student :Othello Mkhonto

Student number: 30224780

How the application works

The program contains a login and sign up page which prompts the user to either login or signup. When the user enters their details and clicks the signup button the program sends their data to the sever which is connected to mongo database online. The details are then stored on the online database. The users name, email and password are captured of which an id is allocated to each user for identification purposes. The user’s password is encrypted to prevent unauthorised access or third parties. The email is set to be unique for every user to prevent iterations of the same email clashing in the database. When user signs up, those details are stored and next time they just login.

Which algorithms were used to classify data

I tried to implement Luhns algorithm which classifies numeric values. The formula verifies a number against its included check digit, which is usually appended to a partial account number to generate the full account number. This number must pass the following test:

From the rightmost digit (excluding the check digit) and moving left, double the value of every second digit. The check digit is neither doubled nor included in this calculation; the first digit doubled is the digit located immediately left of the check digit. If the result of this doubling operation is greater than 9 (e.g., 8 × 2 = 16), then add the digits of the result (e.g., 16: 1 + 6 = 7, 18: 1 + 8 = 9) or, alternatively, the same final result can be found by subtracting 9 from that result (e.g., 16: 16 − 9 = 7, 18: 18 − 9 = 9).

Take the sum of all the digits (including the check digit).

If the total modulo 10 is equal to 0 (if the total ends in zero) then the number is valid according to the Luhn formula; otherwise it is not valid.

The integration of this algorithm collapsed my front end when I tried to concatenate it to the uploaded file. It iterates through the whole documents and finds errors due to the fact that it only recognizes characters.

Design choices that were made and why

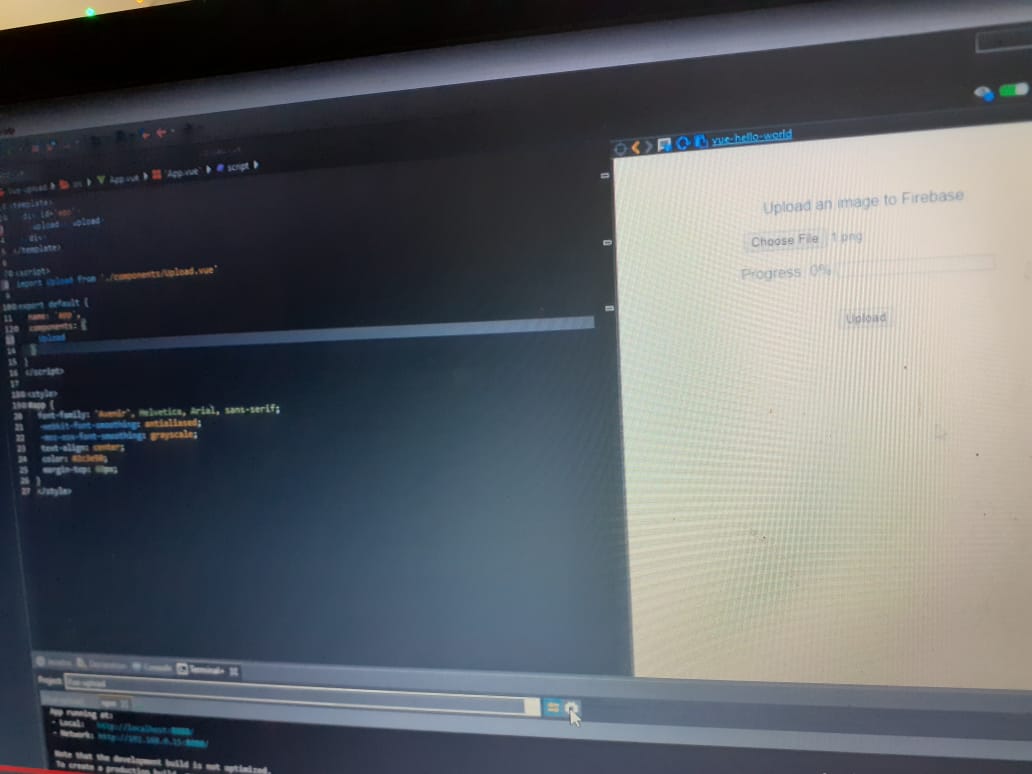
I choose to use vue.js to develop my websites because it is more user friendly and has more favourable benefits compared to other platforms.

**Here are the reasons I choose vue.js below:**

* Vue is small in size and it doesn’t take a lot of time for a user to download it.
* Vue is easy to understand, it has a well defined architecture for keeping your data, life-cycle methods and custom methods separate. It also has some great features like watchers, computed properties and directives which make process of building a modern web application a breeze.
* Vue simple integration, Vue.js most popular among the web developers because it facilitates them to integrate with the existing applications, this is mainly because it is based on the JavaScript framework and can be integrated into other applications built on Javascript.
* Vue is flexibly, it makes it possible for users to write there template in HTML file, JavaScript file, and pure JavaScript file using virtual nodes. This flexibility also makes it easy to understand for the developers of React.js,Angular.js, and any other new Javascript framework.
* Vue has two way communication, it facilitates two way communications because of its MVVM architecture which makes it quite easy to handle HTML blocks. It seems very close to Angular.js which also speeds up HTML blocks.
* Vue has great tooling, the new vue cli i.e.3.x which I used is arguably one of the best tool for a Javascript Framework. It allows you to kickstart new Project with things like Routing,Javascript,

State Store, Linting, Unit Testing, CSS Pre-processors, Typescript, PWA etc built.

**File formats that are supported**

****

For uploading files on vue I used a route called axios that optains the file from the “file” function. Its an add on extention that I installed using “npm install –save axios”

I used methods called onfile selected and onupload to execute the instructions from my route.

The file supported from this algorithm is a image.

I also added percentages to measure the input rate.

**The technologies used and why? they were used?**

The technologies I used include:

**Nodemond**

I installed nodemond on my server using ”npm install -g nodemon”

Nodemond is a utility that monitors for any changes in my source and automatically restarts my server.

**Mangoose**

I installed nodemond on my server using ” npm install --save express mongoose body-parser”

Mangoose is an object document mapper, this means that mongoose allows you to define objects with a strongly-typed schema that is mapped to mongoDB

**Axios**

I installed nodemond on my server using ” npm install --save axios”

Axios is a promise-based http client for JavaScript which can be used in your front-end application and my node.js backend.

**Cors**

I installed nodemond on my server using ” npm install --save cors”

Cors is the server telling the client what kind of http requests the client is allowed to make.

**Bcrypt**

I installed nodemond on my server using ” npm install --save bcrypt”

Bcrypt is a library for Nodejs , where a number then a salt will be generated inorder to hash a password.

**The API’s used in the website**

I used node two API’s

**Node server js:**

Node.js is a platfrom built on chrome’s javascript runtime for easily building fast and scalable network applications. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real time applications that run across a variety of distributed devices.

**Express js:**

This a framework I used that lets you structure a web application to handle multiple different htttp requests at a very specific url.

Express is a minimal but open source and flexible node.js web app framework designed to make developing websites,web apps and api’s much easier.