

Group Assignment:

# Build a Scalable App on Clouds

Lecturer: Thanh N Nguyen

GROUP 8

Nguyen Vi Phi Long - S3904632

DATE

May 13, 2023



## Table of Contents

<b>Table of Contents.....</b>	<b>1</b>
<b>1. Introduction:.....</b>	<b>2</b>
<b>2. Relative Work:.....</b>	<b>3</b>
<b>3. SoftWare Design:.....</b>	<b>3</b>
3.1 Architecture:.....	3
3.2 Dataset:.....	4
<b>4. Implementation:.....</b>	<b>4</b>
<b>5. Application Overview:.....</b>	<b>5</b>
5.1 Sign In/ Sign Up:.....	5
5.1.1 Sign In:.....	5
5.1.2 Sign Up:.....	6
5.2 Admin Account:.....	8
5.2.1 Home:.....	8
5.2.1.1 Add:.....	9
5.2.1.2 Edit:.....	10
5.2.1.3 Delete:.....	11
5.2.2 Product:.....	12
5.2.3 User:.....	13
5.2.4 Order:.....	14
5.3 User Account:.....	14
5.3.1 Home:.....	15
5.3.2 Profile:.....	16
5.3.3 Product:.....	18
5.3.4 View Cart:.....	18
5.3.4.1 Check Out:.....	19
5.3.4.2 Complete:.....	20
<b>6. Limitations:.....</b>	<b>20</b>
<b>7. Further Recommendations:.....</b>	<b>20</b>
<b>Reflection.....</b>	<b>21</b>
<b>References:.....</b>	<b>22</b>

## **1. Introduction:**

myStore is an ecommerce website that was created to give a simple and safe platform for buying and selling various things online. myStore's concept is to offer a user-friendly shopping experience that allows customers to easily explore and purchase items while also allowing merchants to expose their products to a larger audience.

The website's backend is hosted by Google App Engine, which allows it to manage huge visitor levels and assures that the website is always online. It also stores product data, user information, and other associated data in Google Cloud Storage.

Users may utilise myStore to browse through numerous product categories and add items to their shopping basket. They may also check out safely utilising the platform's numerous payment choices. Sellers, on the other hand, may use a simple and easy dashboard to generate product listings, handle orders, and track sales.

myStore is necessary because it offers customers a simple and safe online purchasing platform and allows businesses to reach a larger audience. It also contains a number of features that make the purchasing and selling process simple and painless.

myStore, as a real-world application, may be used by businesses of all sizes to set up an online store and reach clients all over the world. Customers may shop for and buy items from the convenience of their own homes, while sellers can handle orders and track sales through a single platform.

myStore's benefits include a user-friendly interface, dependable hosting, and a secure payment gateway. It also offers real-time statistics and insights to assist sellers in optimising their sales strategy and improving customer experience. Overall, myStore provides both buyers and merchants with a smooth and fast ecommerce experience.

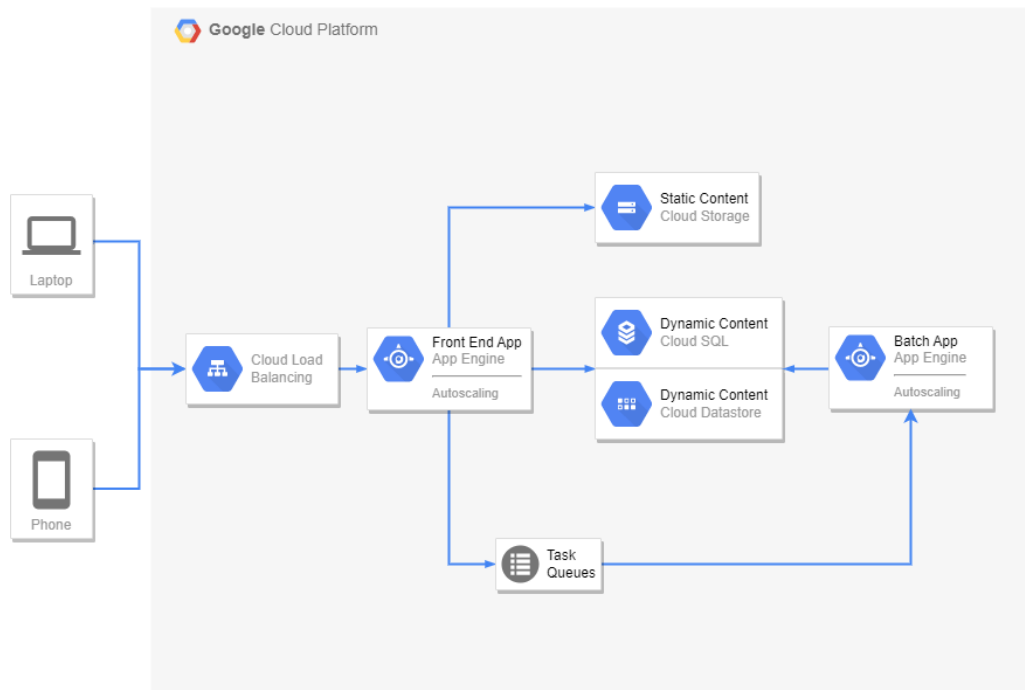
## **2. Relative Work:**

Many e-commerce websites employ technology comparable to myStore, such as Google App Engine and Google Cloud Storage. Best Buy is an example of an e-commerce website that uses Google Cloud Platform.

Best Buy is a well-known electronics retailer with both physical and online locations. It has a diverse product line that includes appliances, laptops, mobile devices, home entertainment equipment, and smart home gadgets. The firm is well-known for its educated sales representatives, Geek Squad assistance, and price-match promise. Best Buy also provides a variety of services, including installation, maintenance, and recycling. Best Buy is a popular place for individuals wanting to buy gadgets and appliances because of its wide selection and low pricing.

## **3. SoftWare Design:**

### **3.1 Architecture:**



myStore is an e-commerce website that stores and queries data using Google Cloud Platform's cloud storage and BigQuery capabilities. Data such as product information, user information, and other pertinent data are stored in CSV files within a storage bucket on the website. This enables easy data scalability and accessibility, as well as simple backup and restoration options. Furthermore, the website maintains product information in BigQuery to enable for rapid data searching and analysis. Using these services also improves the security and dependability of the website's data.

Unfortunately, the website was unable to transform the csv file in storage into the bigquery table anytime the csv file was modified, such as adding, updating, or deleting products. To address this issue, one possible option is to utilise Cloud Functions to automatically convert the CSV file into a BigQuery table whenever it is modified. Cloud Functions may be configured to monitor the Cloud Storage bucket for changes to the CSV file and execute a script that turns the file into a BigQuery table automatically. This script may be created in any programming language that Cloud Functions supports, such as Python or Node.js. Additionally, Cloud Functions may be set to run the script asynchronously to guarantee that the conversion process has no impact on the website's performance.

However, myStore was written in PHP. The only solution left is updating the bigquery manually.

### 3.2 Dataset:

User	
PK	<u>id int NOT NULL</u>
	name char(50) NOT NULL
	phone int
	email char(50) NOT NULL UNIQUE
	password char(50) NOT NULL
	role char(10) NOT NULL

Cart	
PK	<u>id int NOT NULL</u>
FK1	user_id int NOT NULL
FK2	product_id int NOT NULL
	price int NOT NULL
	quantity int NOT NULL
	subtotal int NOT NULL

Product	
PK	<u>id int NOT NULL</u>
	name char(50) NOT NULL
	type char(20) NOT NULL
	color char(50) NOT NULL
	description char(500)
	material char(60)
	price bigint
	in stock int

Order	
PK	<u>id int NOT NULL</u>
FK1	cart_id int NOT NULL
	first name char(50) NOT NULL
	last name char(50) NOT NULL
	address char(50) NOT NULL
	phone char(50) NOT NULL

These are the tables(csv files) in the Cloud Storage:

- User: the website's registered user. Every user has information such as their name, phone number, email address (as a username), password, and role (Admin or User). They will login in to the website using the registered email address and password.
- Product: the item for sale on the website. Every product contains information such as its name, kind, colour, description, material, price, and quantity in stock. All data will be displayed in the Admin account, however the User account can access all product information except the quantity of a product left.
- Cart: the user's shopping cart. The View Cart page allows users to view their cart. It will include the product name, price, quantity and total cost.
- Order: The order is placed when the user enters the shipping information such as name, phone number, and address. The order includes their shipping information (name, phone number, and address) as well as the product purchased via the cart id (product name, price, and quantity).

### 4. Implementation:

- Set up a Google Cloud Platform account.
- Install Google App Engine.
- Create a new project.
- Enable Google Cloud Storage, create a new bucket and upload all the csv files.

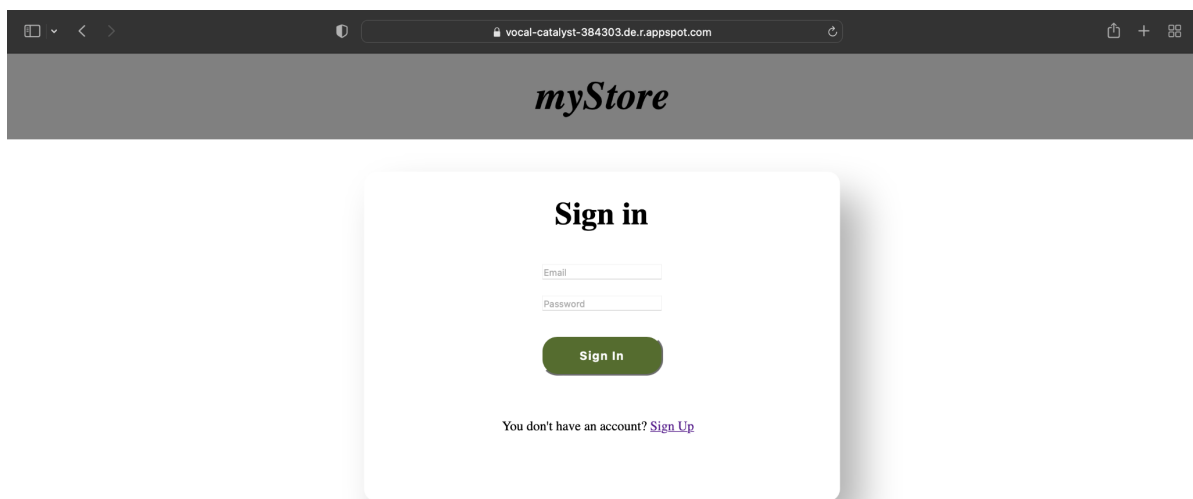
- Enable BigQuery, create a new dataset and create a new table in that dataset from Google Cloud Storage (product.csv)
- Build the Website. (code submitted in the zip file)
- Open Terminal and cd to that folder containing the code.
- Run gcloud init.
- Select 1 and continue. (it will ask you for an account and project you want to use)
- Run gcloud app deploy.
- Enter Y and continue.
- Run gcloud app deploy.
- Add to the url: /signin.

## 5. Application Overview:

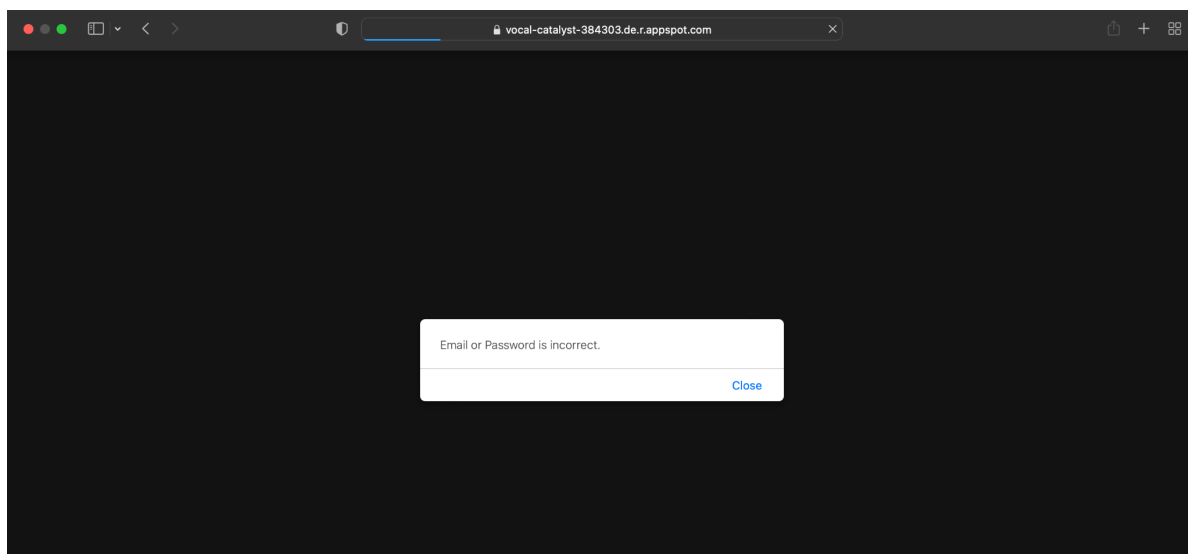
Link to the Application: <https://vocal-catalyst-384303.de.r.appspot.com/signin>

### 5.1 Sign In/ Sign Up:

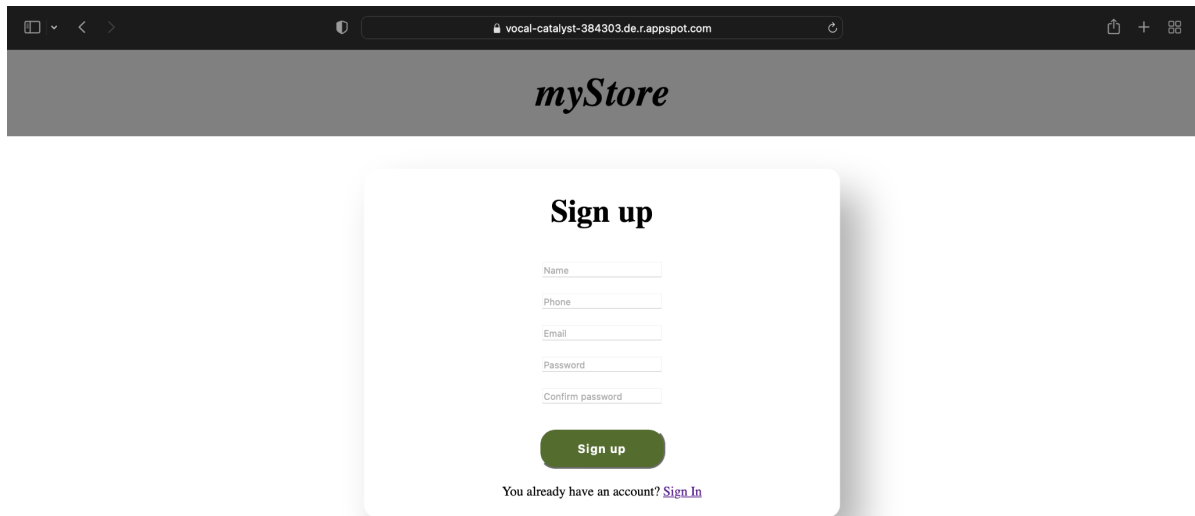
#### 5.1.1 Sign In:



This is the root page of the application. The user can sign in with a registered email and password. Or they can sign up if they don't have an account yet.

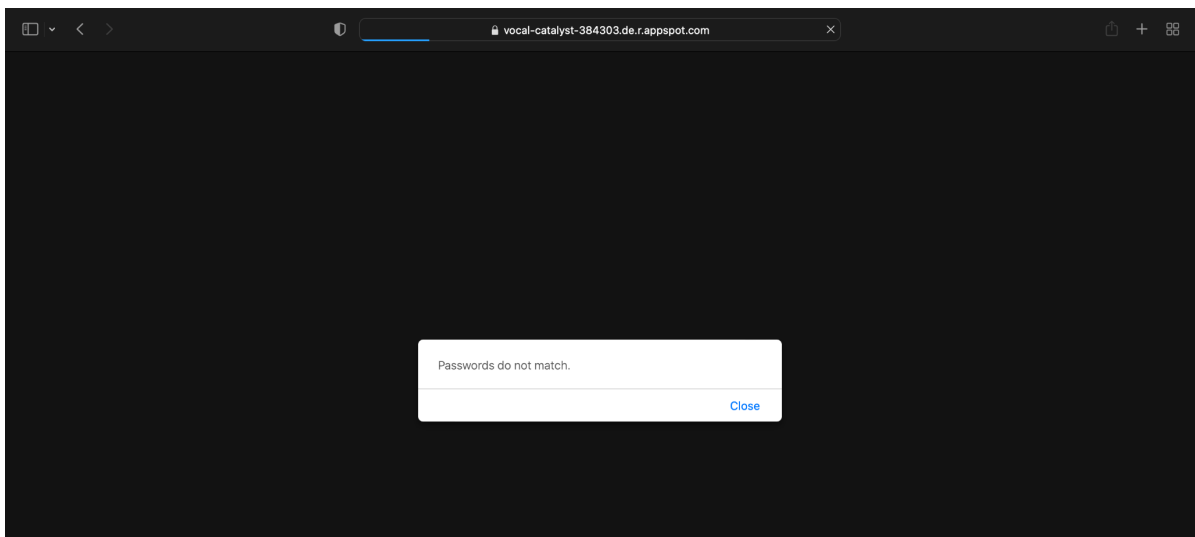
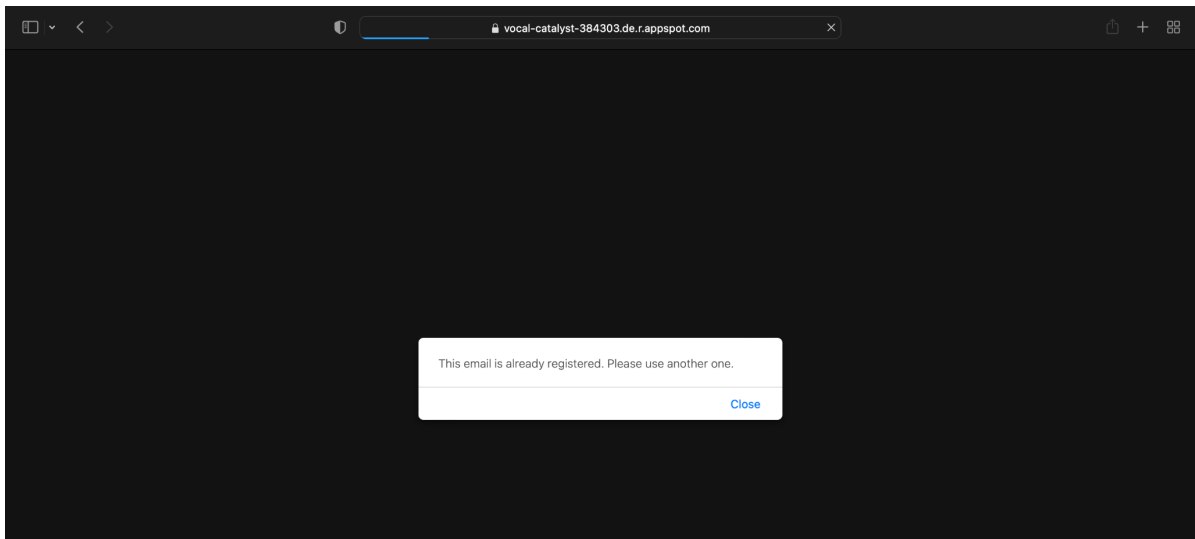
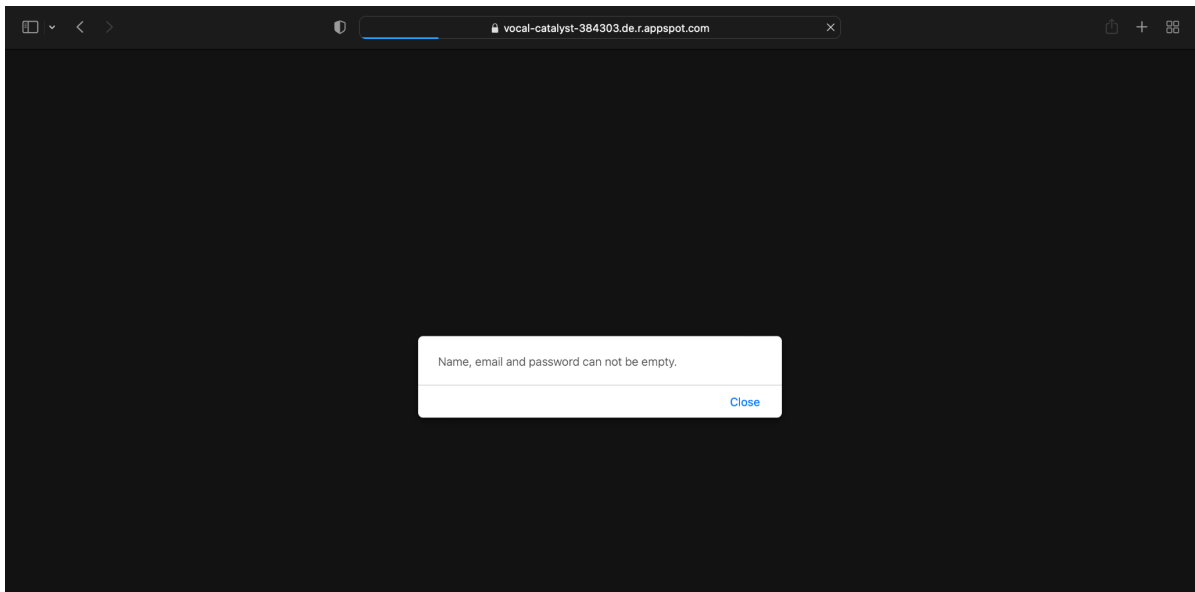


### 5.1.2 Sign Up:



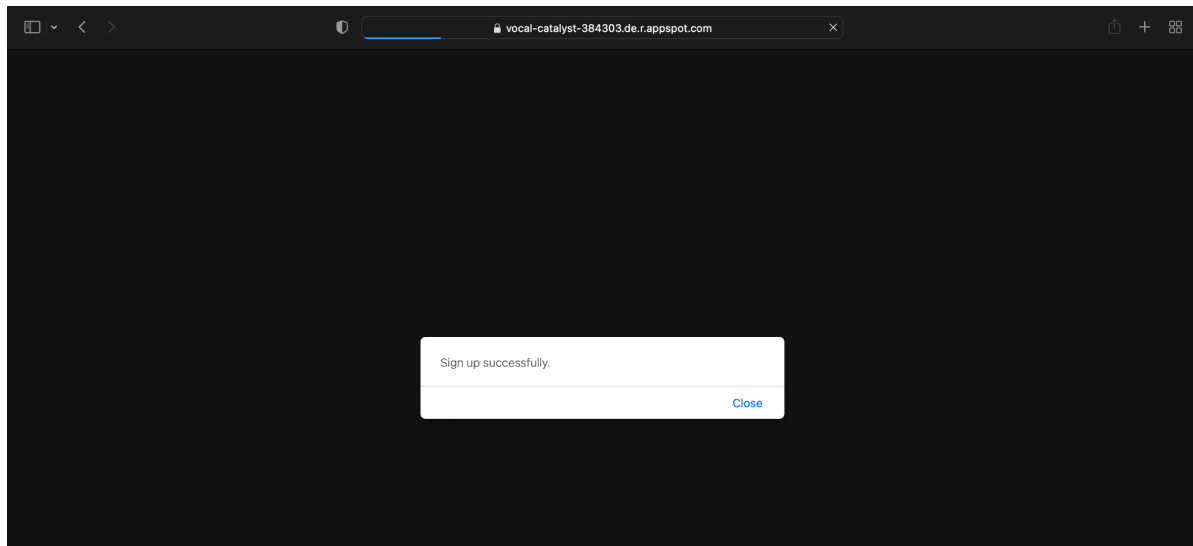
The screenshot shows a web browser window with the address bar displaying 'vocal-catalyst-384303.de.r.appspot.com'. The page header features the 'myStore' logo. The main content is a 'Sign up' form with the following fields: Name, Phone, Email, Password, and Confirm password. A green 'Sign up' button is located below the fields. At the bottom of the form, there is a link: 'You already have an account? [Sign In](#)'.

This is the sign up page. The user can sign up their email address and information. The application will validate the input and print out the error message. Here are all the error messages:



If everything is inputted correctly, this successful message will be displayed.





Then the page will be redirected to the sign up page. The User can click on the sign in href to sign in.

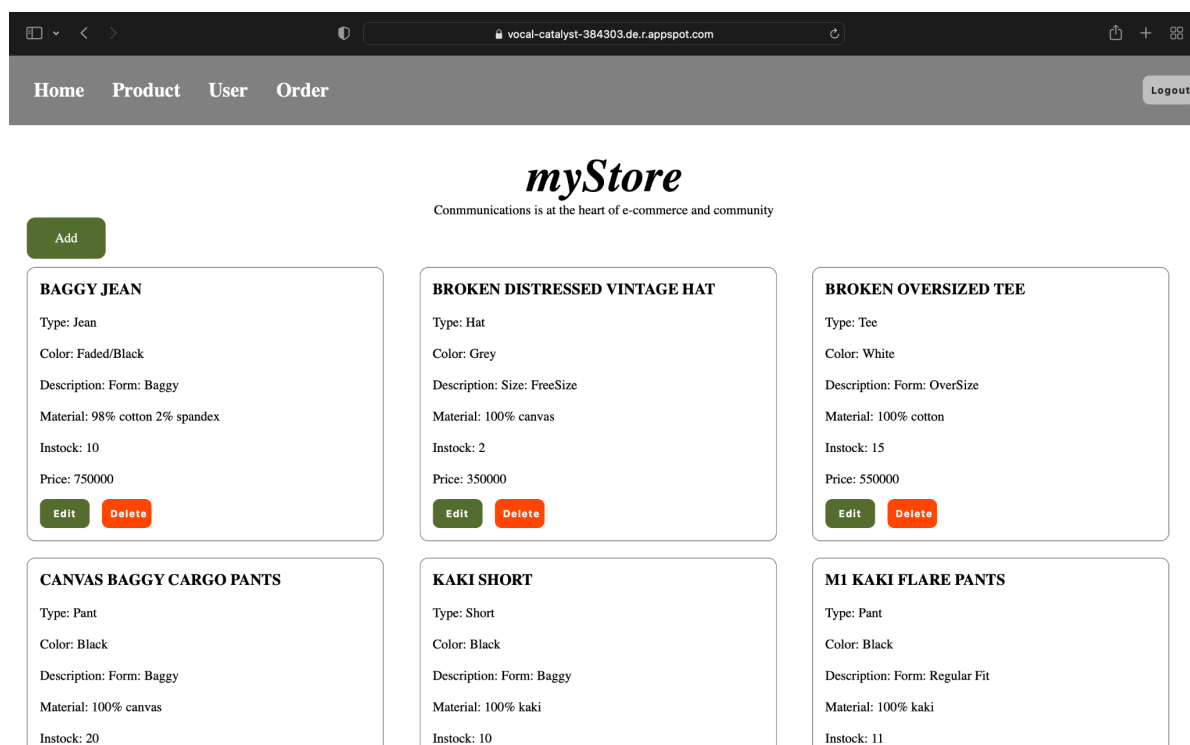
## 5.2 Admin Account:

To sign in as an Admin: email: s3904632

Password: 123

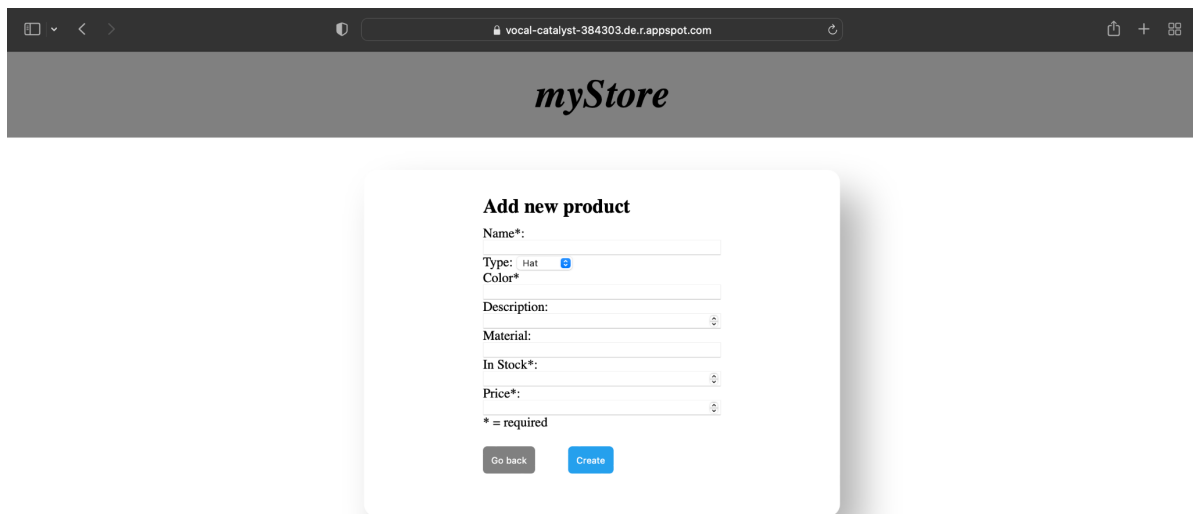
Then it will redirect to the home page of the staff account.

### 5.2.1 Home:



There is a navigation bar in the header that includes connections to other sites. To log out of the account, click the logout button on the right side of the header. There is an add button. This button allows the administrator to add a new product to the csv file. There are also edit and delete buttons for each product in the database to alter the product details or remove the product.

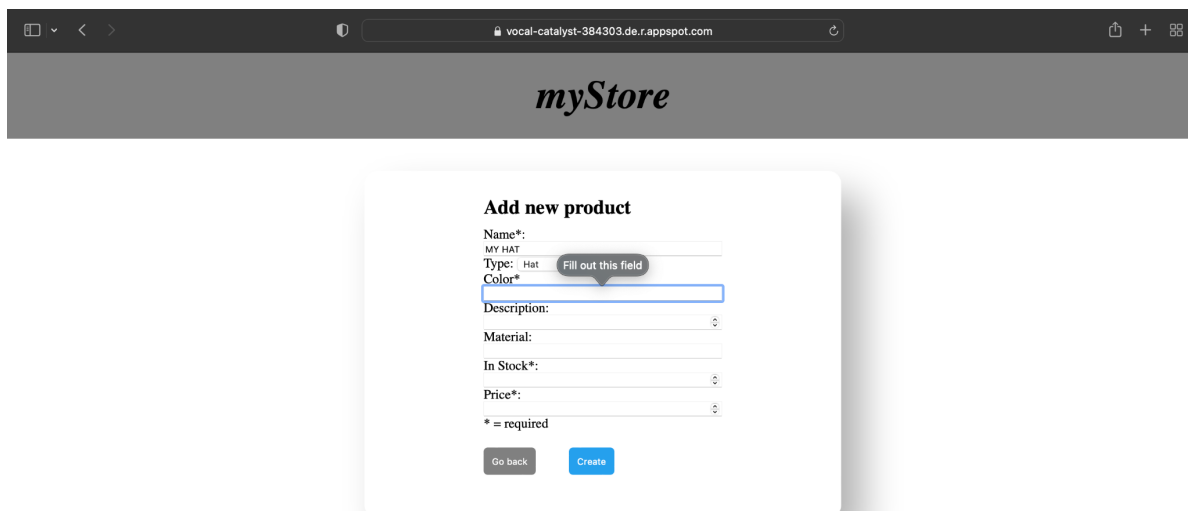
### 5.2.1.1 Add:



The screenshot shows a web browser window with the URL `vocal-catalyst-384303.de.r.appspot.com`. The page has a dark header with the *myStore* logo. Below the header is a white modal box titled "Add new product". The form contains the following fields: "Name\*" (text input), "Type:" (dropdown menu with "Hat" selected), "Color\*" (text input), "Description:" (text area), "Material:" (text input), "In Stock\*" (text input), and "Price\*" (text input). A note at the bottom of the form states "\* = required". At the bottom of the modal are two buttons: "Go back" and "Create".

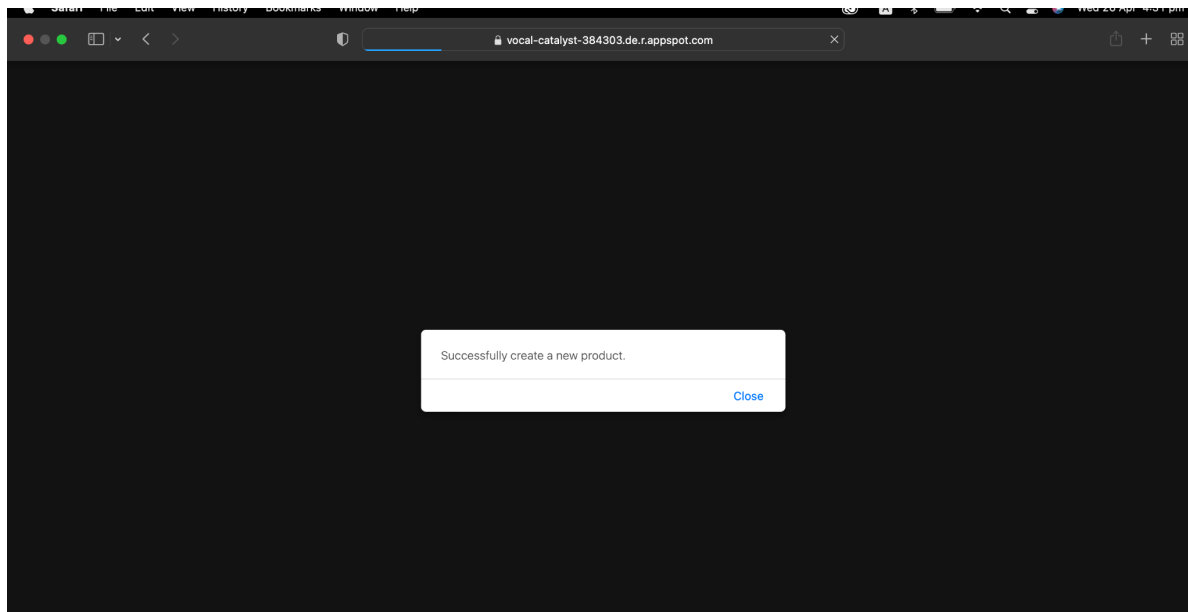
When the Admin clicks on the add button, this page will be displayed. The Admin can enter the product's information and select the product type, then they can click the create button to create the new product.

In case any important fields are missing, an error message will be displayed.



This screenshot shows the same "Add new product" form as the previous one, but with an error. The "Color\*" field is highlighted with a red border, and a red speech bubble with the text "Fill out this field" points to it. The "Name\*" field also has a red border. The "Type:" dropdown is still set to "Hat". The "Description:", "Material:", "In Stock\*", and "Price\*" fields are empty. The "Go back" and "Create" buttons are at the bottom.

If all the fields are entered correctly, a successful message will be displayed. After closing the error message, the page will be redirected to the home page.



The Admin can also use the go back button to quit whatever they are working on and go back to the home page.

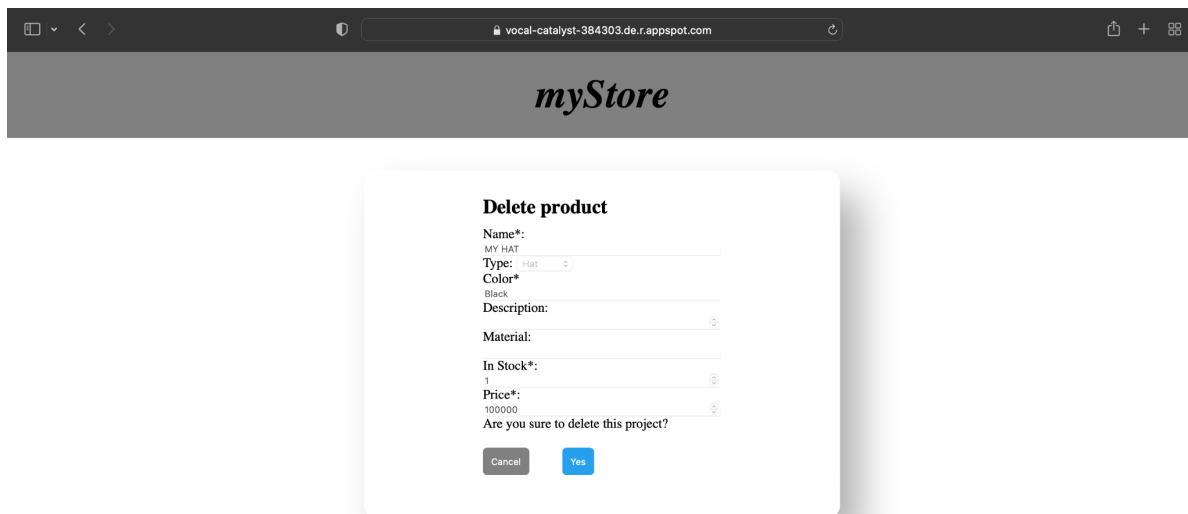
#### 5.2.1.2 Edit:

A screenshot showing the 'myStore' website header and an 'Update product' form. The header is a grey bar with the text 'myStore' in a stylized font. Below it, there is a white form titled 'Update product'. The form contains several input fields: 'Name\*' (filled with 'MY HAT'), 'Type' (a dropdown menu with 'Hat' selected), 'Color\*' (filled with 'Black'), 'Description:' (a text area), 'Material:' (a text area), 'In Stock\*' (a dropdown menu with '1' selected), and 'Price\*' (a dropdown menu with '100000' selected). Below the form, there are two buttons: 'Go back' and 'Update'. A note at the bottom of the form states '\* = required'.

When the Admin clicks on the edit button, this page will be displayed. All the product details will be automatically filled in the form. The Admin can update the product information by inputting in the fields.

The update button and the go back button have the same functionality as the buttons in the add page.

### 5.2.1.3 Delete:



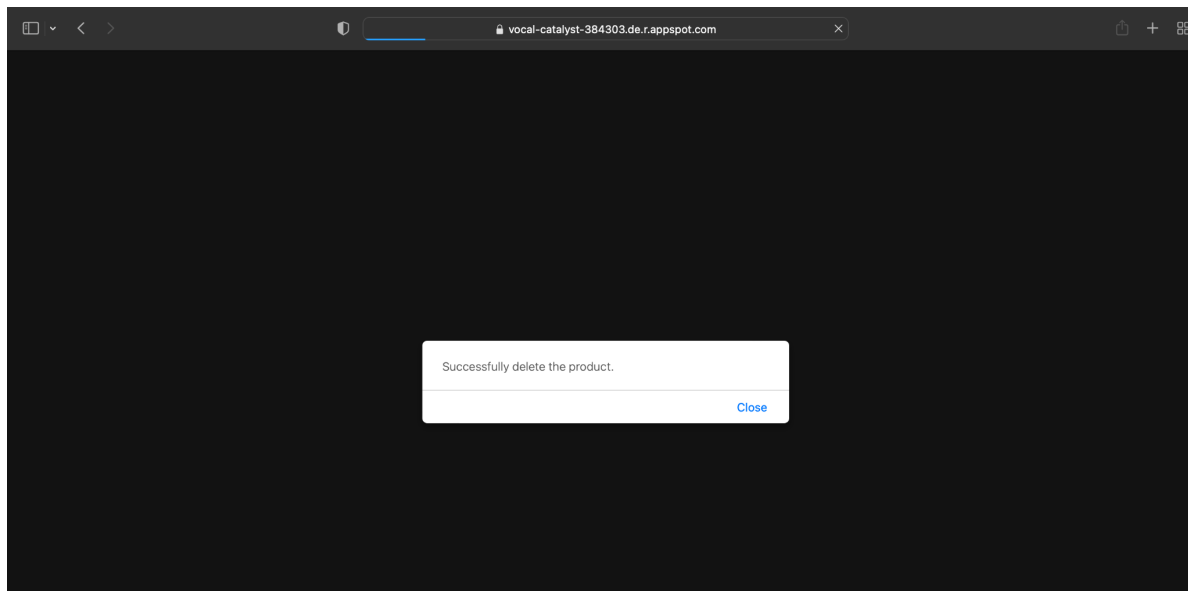
The screenshot shows a web browser window with the URL `vocal-catalyst-384303.de.r.appspot.com`. The page header displays the *myStore* logo. A modal titled "Delete product" is centered on the screen. The form contains the following fields, all of which are disabled (indicated by a grey background):

- Name\*: MY HAT
- Type: Hat (dropdown menu)
- Color\*: Black
- Description: (text area)
- Material: (text area)
- In Stock\*: 1
- Price\*: 100000

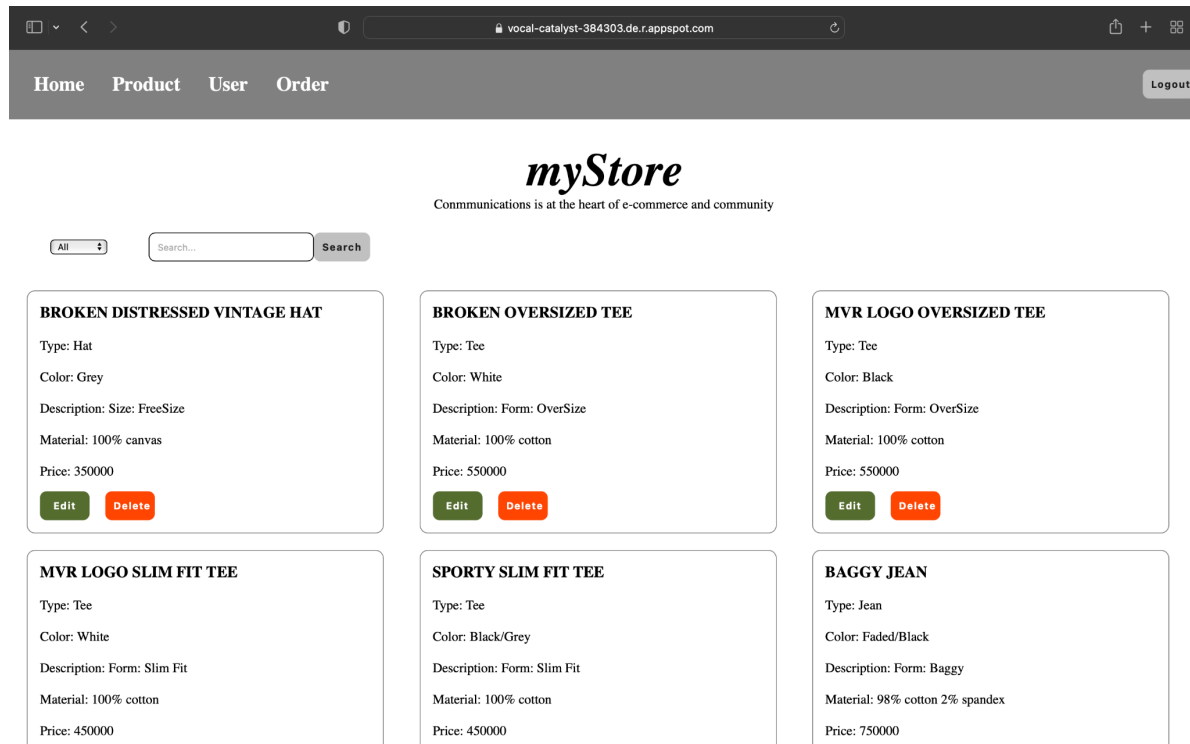
Below the form, there is a confirmation message: "Are you sure to delete this project?". At the bottom of the modal, there are two buttons: "Cancel" (grey) and "Yes" (blue).

When the Admin clicks on the delete button, this page will be displayed. All the product details will be automatically filled in the form. But all the fields will be disabled. The Admin can only read but not update the product details.

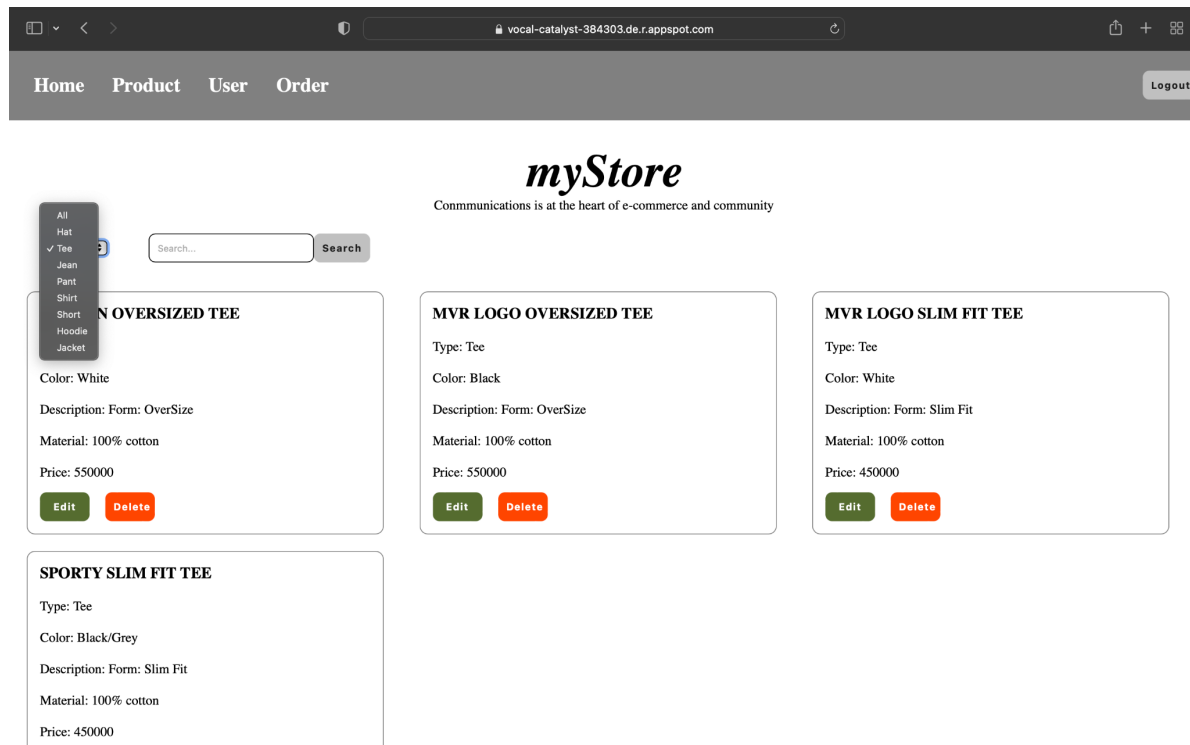
If the Admin clicks on the cancel button the page will be redirected to the home page. And if they click the yes button the product will be removed from the database (csv file) and a successful message will be displayed. Then the page will redirect to the home page.



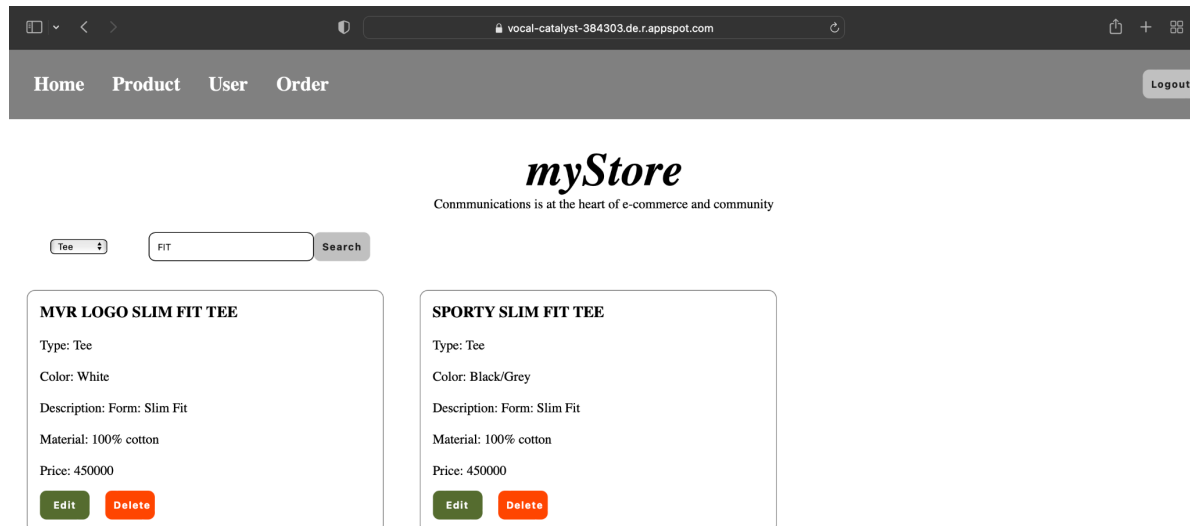
## 5.2.2 Product:



Similarly, a navigation bar and a logout button with the same functionality will be included. On this page, however, the add button has been replaced with a choose button for selecting a product type and a search bar for searching for a product name.

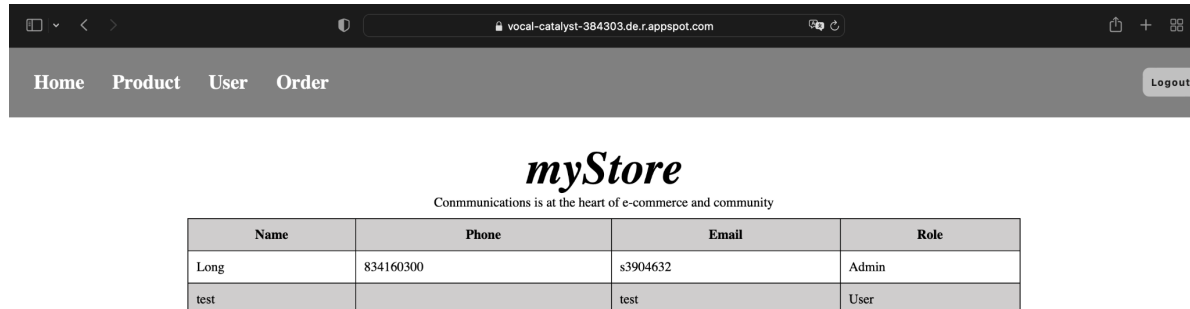


The product will be filtered based on the product type selected, the search box input, or both.



The Admin can also edit and delete the product by using the buttons inside each product's container.

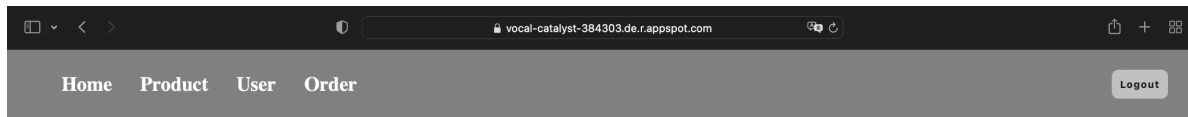
### 5.2.3 User:



In a table, the User page will list all of the website's registered users as well as the Admin account.

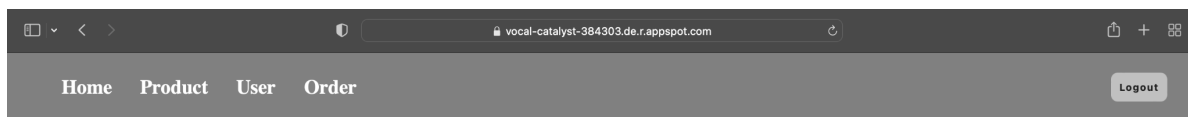
Whenever there is a new user, this will also automatically update.

### 5.2.4 Order:



The order page will display the order placed by the customer. There is a select box to select an order status and a submit button. There are four options: all, packaging, delivering and received. The admin has to click on the search button to search, the page can not automatically display the content.

The Admin can choose any status and filter the order.



**Orders:**

First Name:  
Last Name:  
Address:  
Post Code:  
City/Province:  
Phone:  
Order Status: Received

Product Name	Price	Quantity
CROP HOODIE	750000	1

Total:885000

First Name:  
Last Name:  
Address:  
Post Code:  
City/Province:  
Phone:  
Order Status: Packaging

Product Name	Price	Quantity
CROP HOODIE FULL ZIP	800000	1

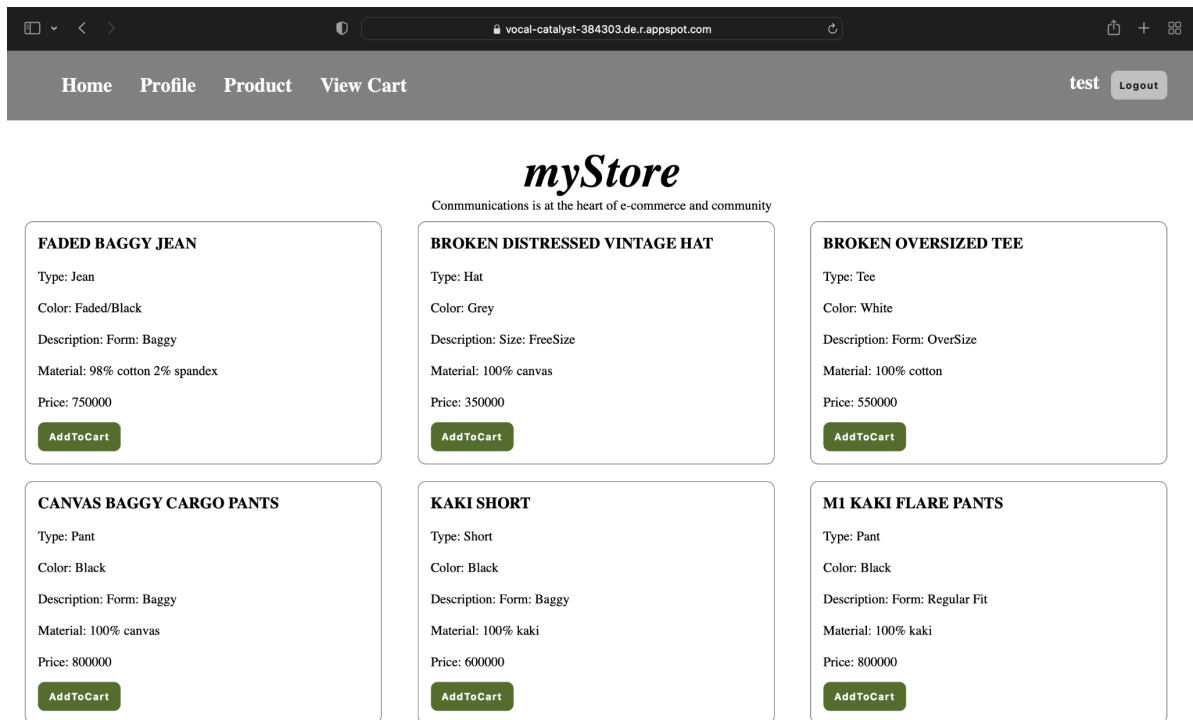
If the order status is packaging the Admin can use the to delivery button to change the order status to delivering.

### 5.3 User Account:

To sign in as a User:      email: test  
   Password: 1

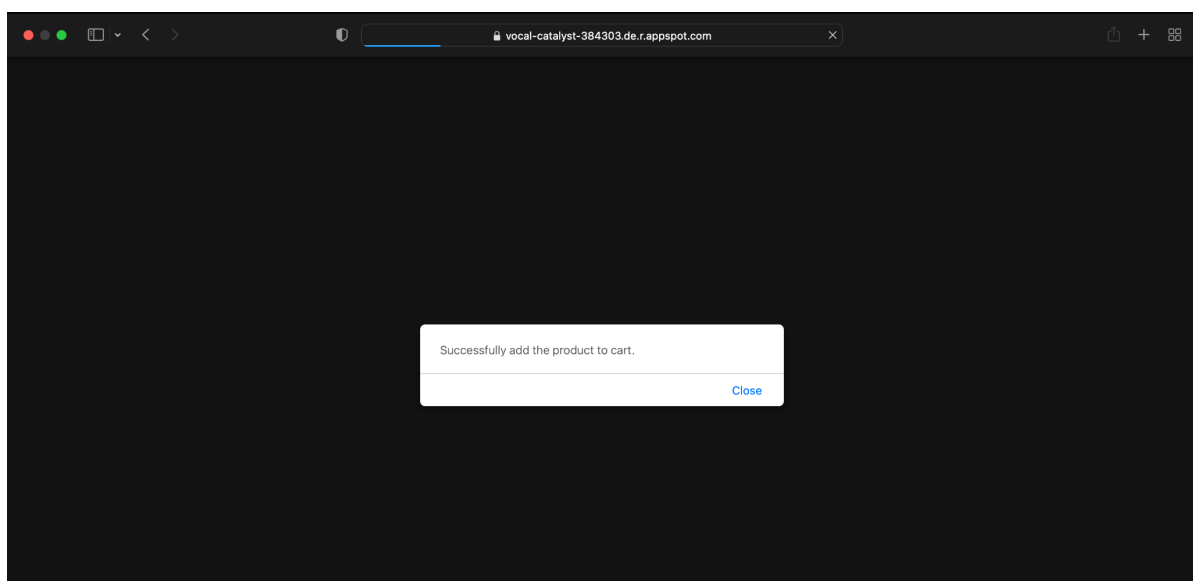
This account is only for demo, the user can register with their own email address then login to the website.

### 5.3.1 Home:



This is the user account's home page. Four links in the navigation bar lead to the pages for the home page, the profile page, the product page, and the view cart page. The sign-in email address will be shown on the left side of the header, and a logout button will be located next to the email.

Every card displayed on the body is a product. It contains the product details. The user can use the add to cart button to add the product to cart. After clicking on the button, this message will be displayed.



Closing the message, the page will be reloaded.



### 5.3.2 Profile:

Home Profile Product View Cart test Logout

myStore  
Communications is at the heart of e-commerce and community

**Account:**

Name: test  
Phone:  
Email: test  
Change Password

**Orders:**

First Name:  
Last Name:  
Address:  
Post Code:  
City/Province:  
Phone:  
Order Status: Packaging

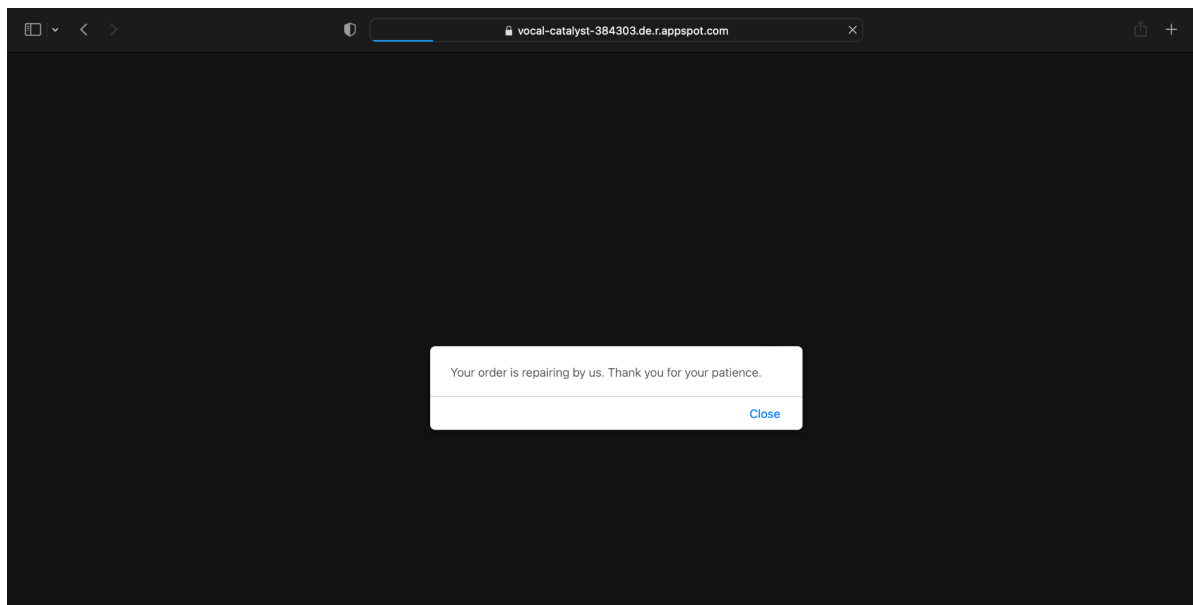
Product Name	Price	Quantity
CROP HOODIE FULL ZIP	800000	1

Received

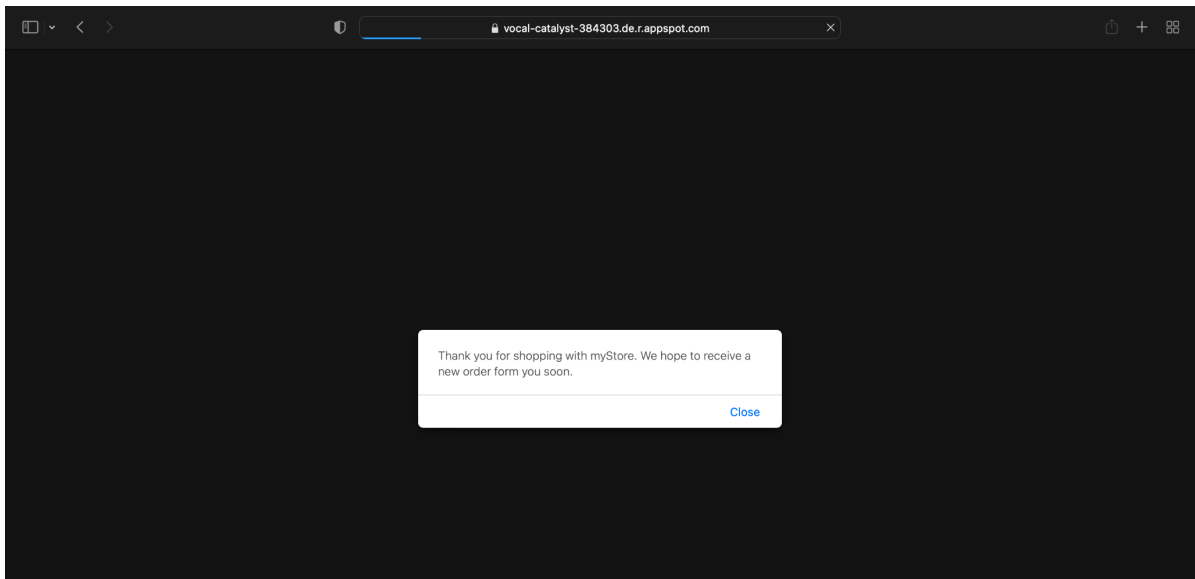
Total:940000

This is the profile page. There are two main sections in this page:

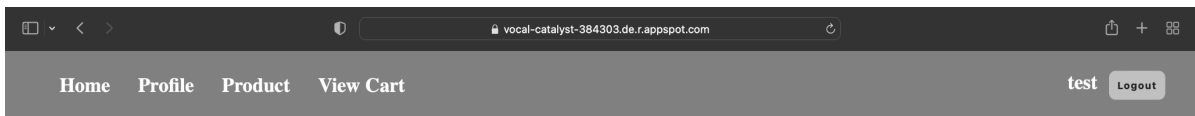
- Account: this container will display all the information of the user and a change password button.
- Orders: this is a list of containers. Every container is an order listing all the order details. There is a received button. The User can click on this button after they receive their order. In case, the order is still in the packaging process but the user clicks on the receive button, this error message will be displayed.



A successful message will be displayed in the right case: the Admin finishes repairing the product and the order status is changed to delivering.



After closing this message, the container will be hidden from the order list.



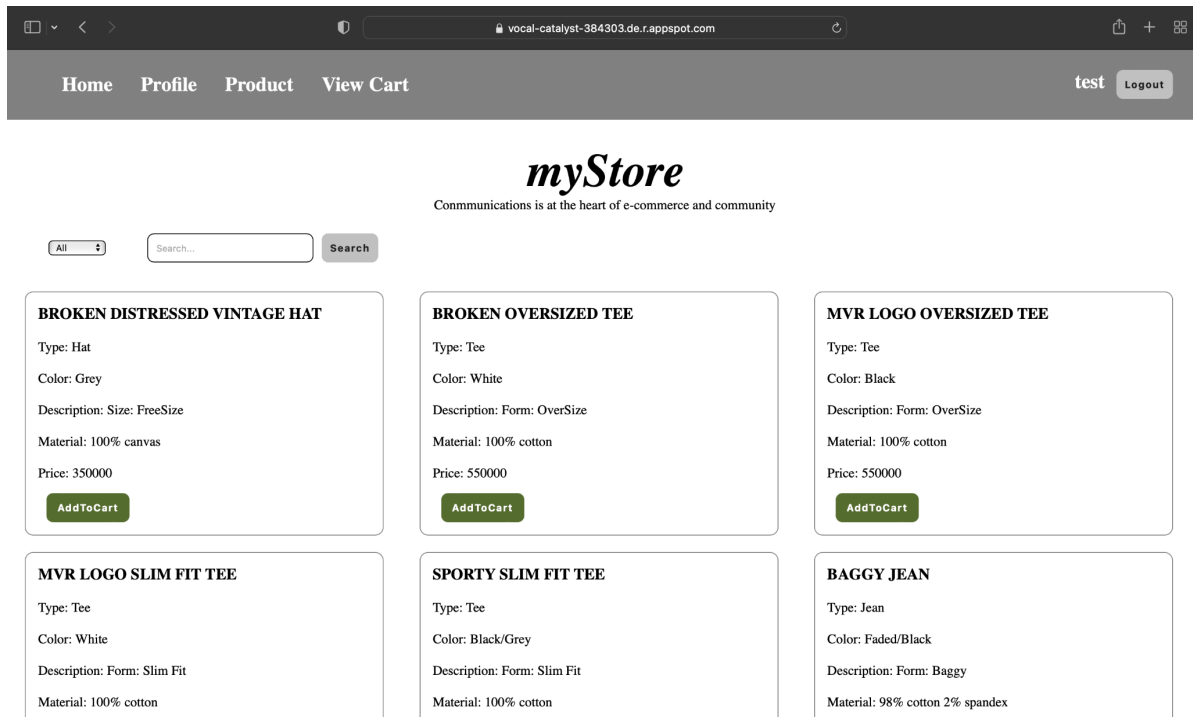
**Account:**

Name: test  
Phone:  
Email: test

Change Password

**Orders:**

### 5.3.3 Product:

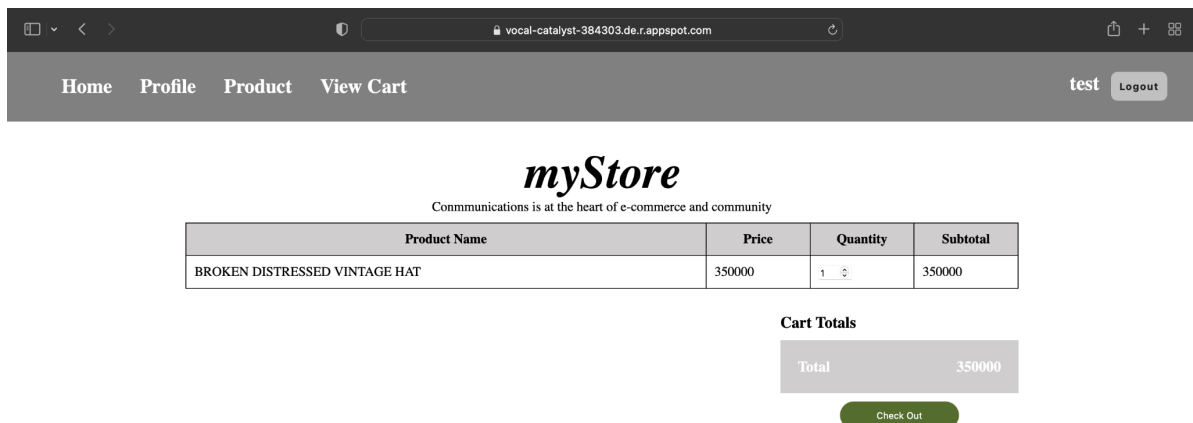


This page is similar to the product page of the staff product (5.2.2 Product) The only difference is the edit and delete buttons are changed to the add to cart button.

Due to the failure of automatically converting the csv file into table BigQuery, the add to cart button in this page will not work.

The other functionalities are working correctly.

### 5.3.4 View Cart:



A table of the chosen products (including the product name, price, quantity, and subtotal) may be found on the view cart page. The quantity input field is automatically adjusted to accept values between 1 and the number of products in stock. The user can access the check out page by clicking the check out button.

There is a limitation in this page. The subtotal will not automatically update due to the changing of the quantity.

### 5.3.4.1 Check Out:

Home Profile Product View Cart test Logout

**myStore**  
Communications is at the heart of e-commerce and community

**Ship To**

First Name Last Name

Address

Postal Code Province/City

Phone

CONFIRM

**Order Summary** [Edit Cart](#)

Merchandise: 350000

Shipping: 60000

Tax: 35000

Total: 445000

This is the check out page. There are two main objects in this page

- Shipping form: The shipping form is located on the left side. To complete the checkout, the user must fill out the entire form. This error message will appear in any field that is left empty.

Home Profile Product View Cart test Logout

**myStore**  
Communications is at the heart of e-commerce and community

**Ship To**

Fill out this field

First Name Last Name

Address

Postal Code Province/City

Phone

CONFIRM

**Order Summary** [Edit Cart](#)

Merchandise: 350000

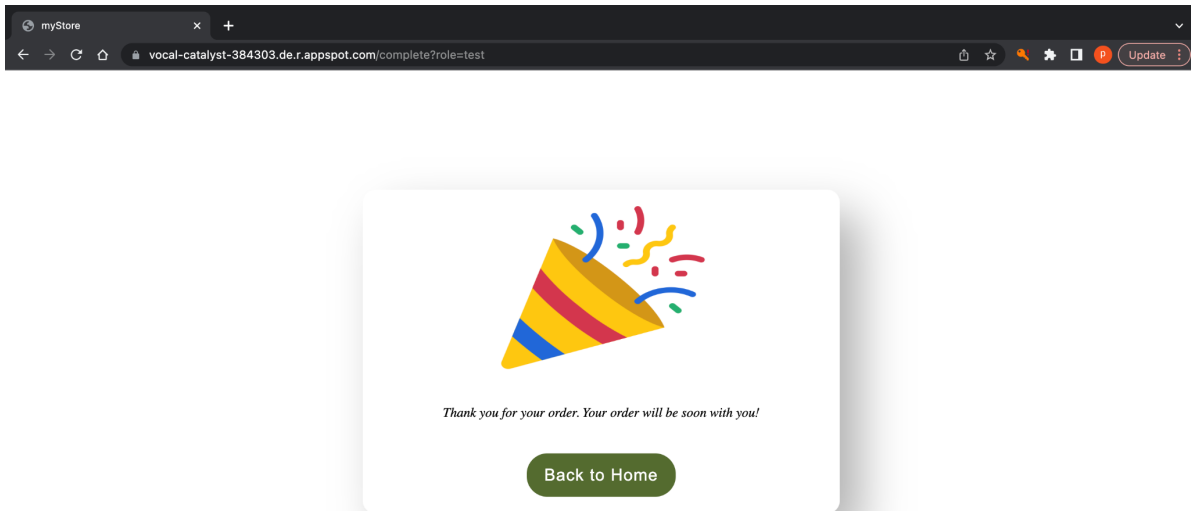
Shipping: 60000

Tax: 35000

Total: 445000

- Order summary: This summary will be found on the right side. A link to the view back page is included in this. Additionally, the billing information will be shown. The user can check again at this point.

### 5.3.4.2 Complete:



This entire page will appear if the user fills out the shipping form completely and hits the confirmation button. The back to home button can be used to return to the homepage.

The user has successfully placed an order if they are able to access this page. For them to keep track of it, this order will immediately change on their profile. Additionally, this order will be automatically listed in the Admin account's order page.

## 6. Limitations:

There are several limitations in this website:

- If the quantity is altered, the subtotal will not adjust automatically on the view cart page.
- The website is not connected to any external shipping or payment providers.
- The website is unable to convert a csv file into a BigQuery table automatically.
- The capacity of the website to manage a high volume of goods and orders is under question.

## 7. Further Recommendations:

There are a few things can be applied to update the website:

- Debug the subtotal problem.
- Connecting to a third party to handle the shipping and payment process.
- Generate a Google Cloud Function to automatically convert the csv file in the Google Cloud Storage into the table in Google BigQuery.
- Connect to a database, this will ensure the scaling of the website.
- Add images for each product.
- Improve the UI.
- Real-time chatting can be added to support the user as soon as possible.

## Reflection

Before the course, I had no prior knowledge of php, html, or CSS. Throughout the course, I have learned those things. After the first lab test, I learned more about those topics and felt more confident. When I am unable to reach my teammates. I made the decision to begin the group project alone rather than wasting time trying to find them.

At first, I wanted to create something easy and familiar to me, such as a weather app. However, after obtaining the lecturer's guidance and suggestions, I altered the topic to an e-commerce website. Nonetheless, it is not a difficult or novel issue. This is a completely new field to me. For my project's UI and functions, I had to conduct extensive study on various e-commerce websites on the internet. Furthermore, I had to simplify it enough that I could implement it myself.

Getting the assignment done on time was difficult. Fortunately, I was able to do the majority of the task. Because I managed the project alone, I had complete control over its application and execution. This endeavour provided me with a wealth of knowledge. Although the result is not as good as I expected and it has a lot of room for improvement. I'm grateful the lecturer let me handle things on my own.

## References:

[1]: Best Buy, [Online], Available: <https://www.bestbuy.com/>

[2]: Thanh N Nguyen, "Week 1 Online: Introduce to Cloud Computing", rmit.instructure, 2023, [Online], Assessed on: April 27, 2023, Available: [https://rmit.instructure.com/courses/118883/pages/week-1-online-intro-to-cloud-computing?module\\_item\\_id=4961242](https://rmit.instructure.com/courses/118883/pages/week-1-online-intro-to-cloud-computing?module_item_id=4961242)

[3]: Thanh N Nguyen, "Week 2 Online: Google Cloud Platform", rmit.instructure, 2023, [Online], Assessed on: April 27, 2023, Available: [https://rmit.instructure.com/courses/118883/pages/week-2-online-google-cloud-platform?module\\_item\\_id=4961243](https://rmit.instructure.com/courses/118883/pages/week-2-online-google-cloud-platform?module_item_id=4961243)