1.1 OVERVIEW:

The web application can be divided into two modules. First module for the customer and the second for the employee. With the customer's model, a user or a customer would be able to browse through the different products provided by the employee. The customer is also provided with an option to search a product just by uploading an image of the product to the web application. The user would also be able to add and remove the items for the cart and then checkout whenever required. Further, they'll be receiving an email once his orders have been placed or when then items those are added to his cart are about to be 'Out of Stock'.

With employee's module, an employee would be able to keep track of stocks available for each product and the employee would also be able to visualize these data points. Furthermore, the employee can order the products so that he won't be 'Out of Stock' for the product. The employer would also be alerted with an email when any product's quantity goes below any set threshold value. To develop this, AWS services like AWS Lambda, Zappa, S3, Dynamo DB, rekognition have been used.

2.1 DESCRIPTION:

The web application provides the following functionality through web interface.

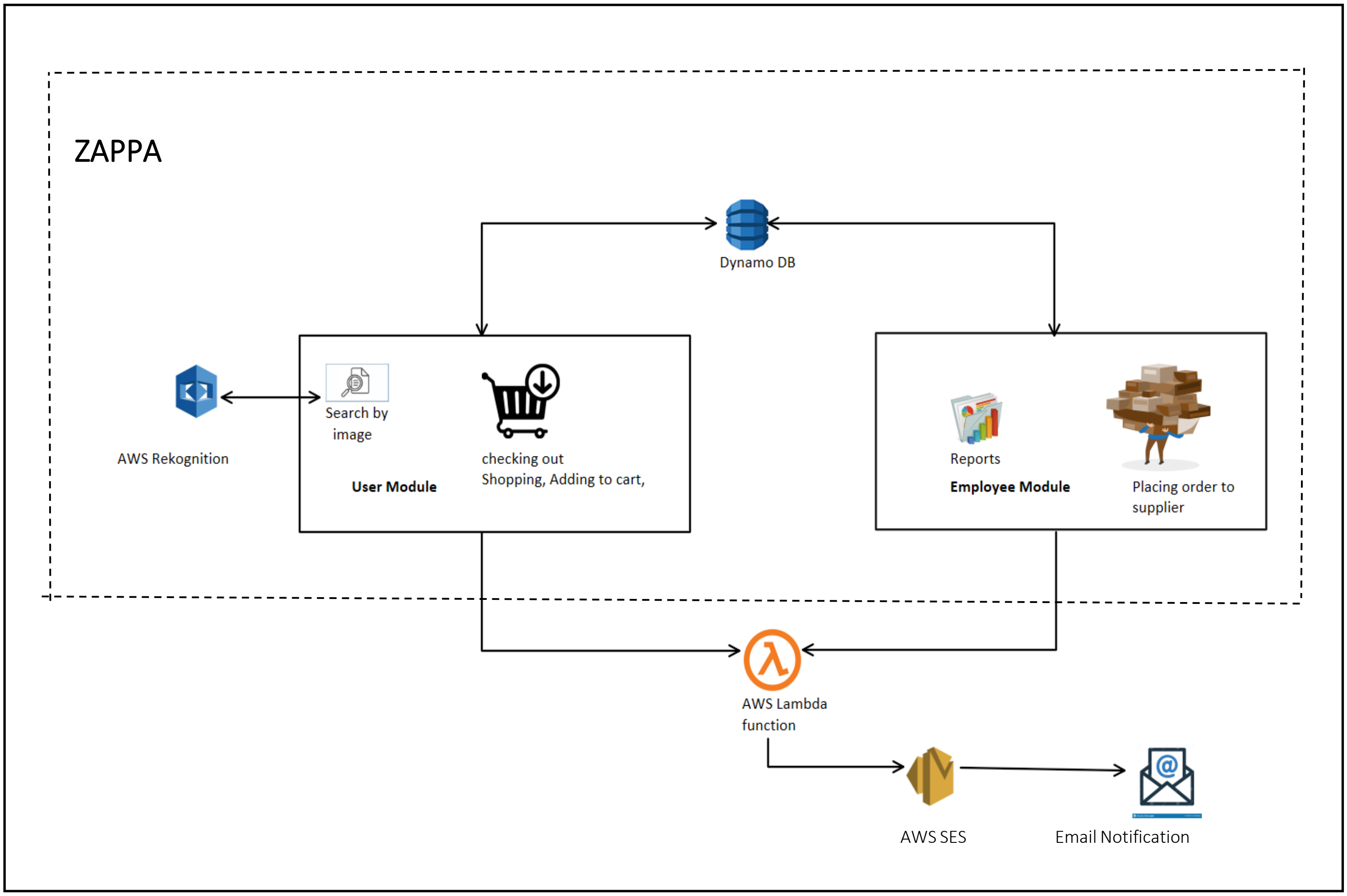
1. Login Page: This page provides authentication to access the webpage both for the customer and the employee separately. It also includes support for password recovery.
2. Employee Home Page: This page is accessible only to the employee and he would be able to see each product category separately along with the buttons that would navigate the employee to the page where he can visualize products available and the quantity of product present using graph and it would also help him to navigate to the page, from where he can place order to the supplier.
3. Customer Home Page: This page is accessible by the customer and the customer would be able to view the product categories which on clicking would take him to the respective product category page.

4. Product Category Page: This page displays different products from different manufacturer of the same category. It would also contain the name, description and cost of the product and the customer would be able to add the product to the cart directly from this page.
5. Search By Image Page: In this page, the customer would be able to navigate to respective product page upon uploading a similar image of the product.
6. Cart Page: This page displays all the items that were added to the cart by the user and the “Checkout” button would be enabled only when at least one product is added to the cart.
7. Order Summary Page: This page displays the order summary and the total cost of the purchase plus taxes. User will also be able to enter the address to which the products should be shipped to. Upon successfully placing the order, the customer would receive the mail confirming his order.

Apart from the web interfaces, few background lambda functions will be triggered upon meeting certain conditions.

1. After a customer successfully places an order, a lambda function will be triggered, which sends a confirmation email to the customer’s registered email, which is verified during the account creation stage.
2. Whenever a customer places an order, if the items is nearly a certain threshold, another lambda function would be triggered, which notifies the employee that this product is about to be ‘Out of Stock’ and need to be restocked again.

3.1 APPLICATION ARCHITECTURE:



AWD DynamoDB:

Amazon DynamoDB is a fully managed NoSQL database which uses a key-value pair and documents to store data.

In this web application, we use three tables cart, products, and users with their partition keys as email, product name and email respectively.

Each time a user checks out certain product, its quantity decreases in the quantity field which is further used as trigger for the lambda functions.

AWS Rekognition:

AWS is a cloud-based Software-as-a-service tool, which makes it easy to analyze images and videos. It also provides highly accurate facial analysis, face comparison, object recognition etc.

In the web application, we use AWS rekognition to identity the type of product uploaded by the image, so that user will be redirected directly to the page containing the similar products.

User Module:

The customer will be able to browse through different product categories and different products through the web interface. User would also be able to search any product category just by uploading an image of similar product. The customer would also be able to add or remove items from the cart any time before checking out.

Employee Module:

The employee will be able to see the overall summary reports on how many of each product is remaining and the employee would also be able to order any product from the supplier incase of any shortage of the product.

AWS Lambda function:

AWS lambda is an event driven, serverless computing platform which runs in response to events and its computing resources are automatically managed by AWS.

In the web application, we use lambda functions to use AWS SES functionality where, emails will be sent to the registered users when an event is triggered. In case of the web application, a lambda function will be triggered when the user checks out or when a particular product is about to go “Out of Stock”.

AWS Simple Email Service:

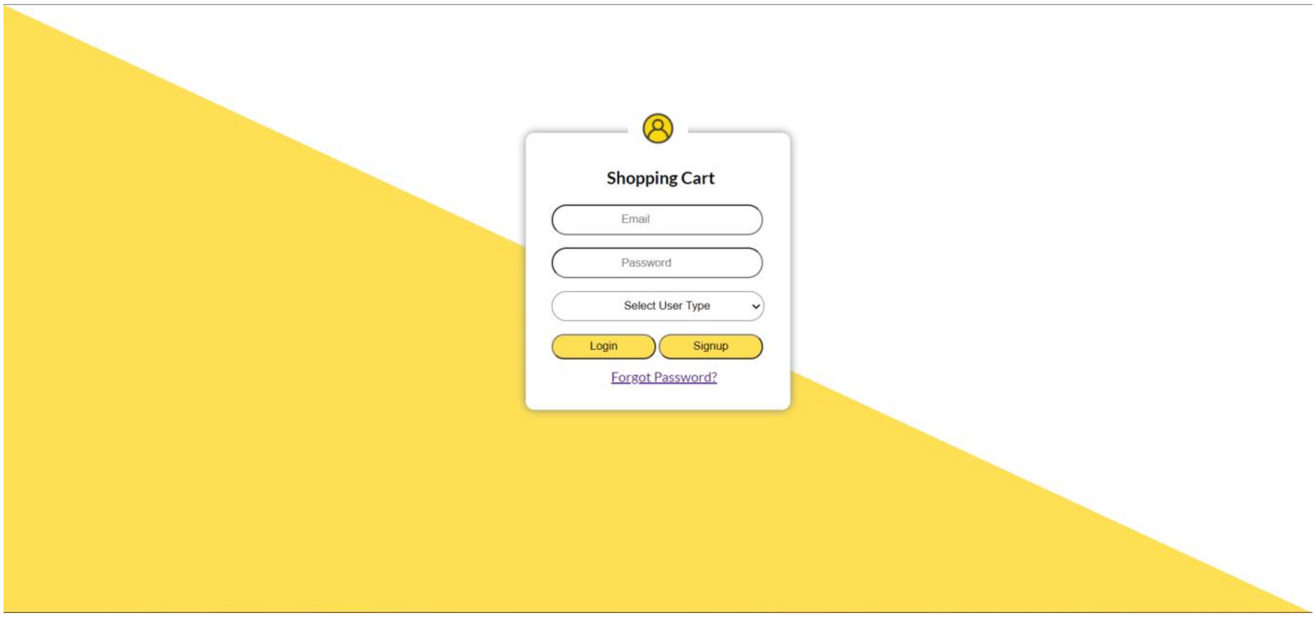
Amazon SES is a cost effective, flexible, and scalable email service that helps the users to send a mail from within any application. With AWS SES, user will be able to send mail securely, globally and at a large scale.

Zappa:

Zappa makes it easy to deploy flask application as a AWS lambda function.

4.1 WEB APPLICATION USER GUIDE:

LOGIN PAGE:



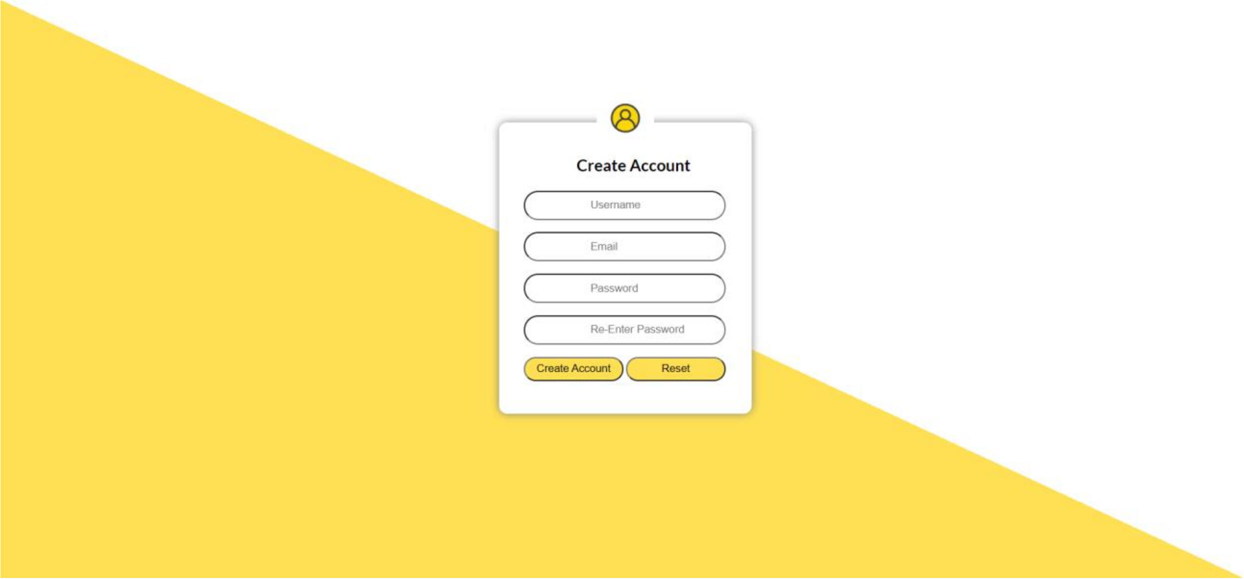
In the login page, the user will be able to login either as customer or employee, so that they will be navigated to their respective home pages.

The user would also be able to navigate to ‘Signup’ page and ‘Forget Password’ page from the login page.

SIGNUP PAGE:

The customer would be able register himself to the application from this page, upon which the customer will receive a verification mail.

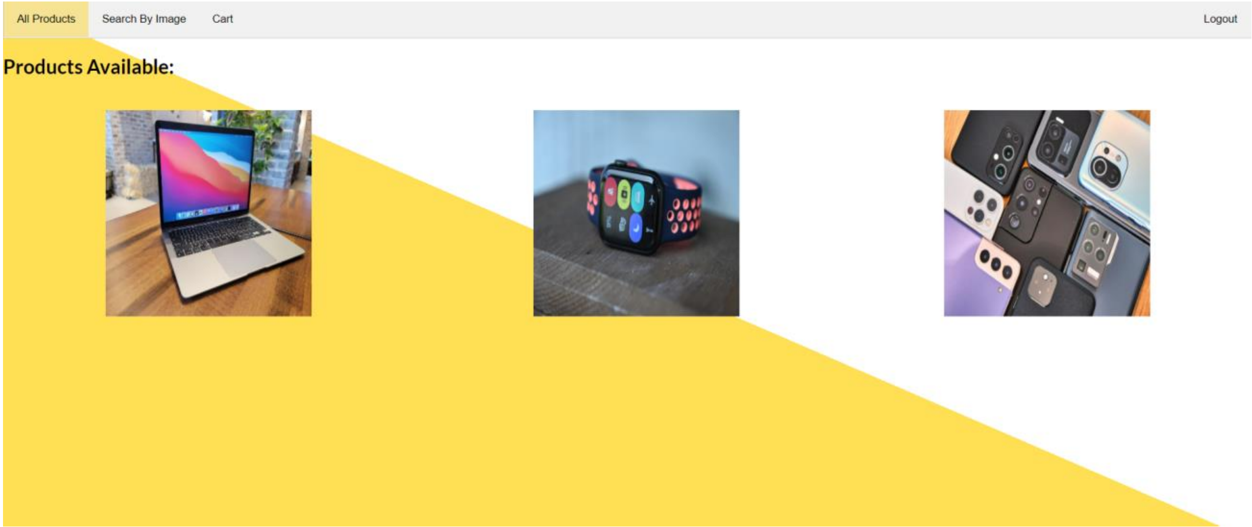
After verifying themselves using this mail, the customer would be able to access the application for shopping.

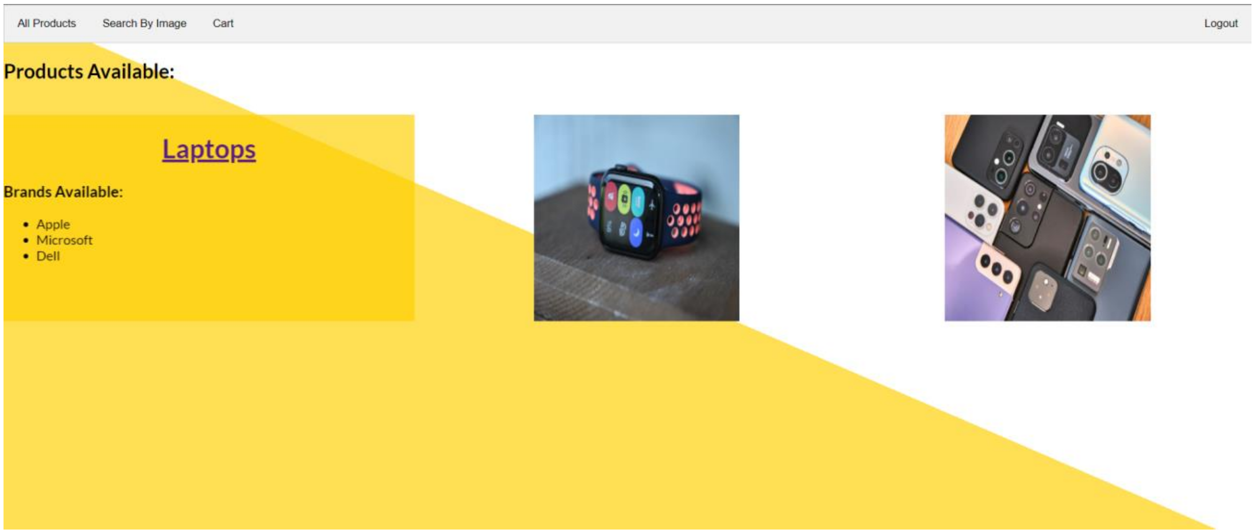


CUSTOMER HOME PAGE:

In the page, customer would be able to navigate to different product category page, where the customer would be able to view details of those products.

The customer can also check his cart from this page, and also would be able to navigate to 'Search by Image' screen.





Products Available:

Laptops

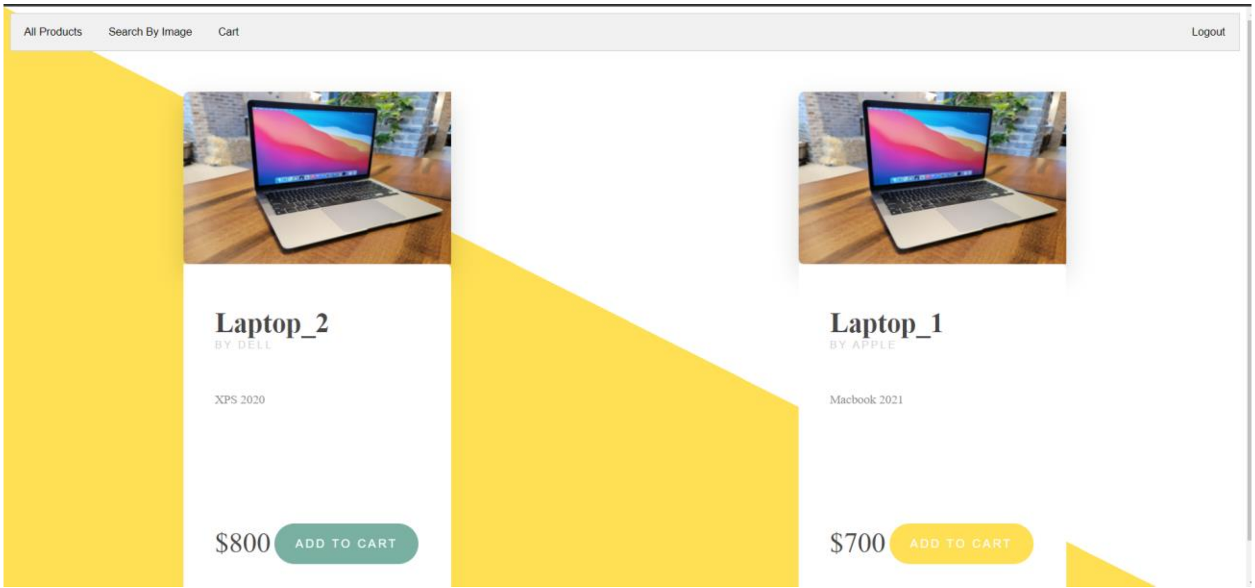
Brands Available:

- Apple
- Microsoft
- Dell

PRODUCT CATEGORY PAGE:

The customer would be able to view the all the products under the selected category, and they would also be able to view to product name, manufacturer, details, and cost of the product.

The customer would also be able to add the items directly into the cart from this page.

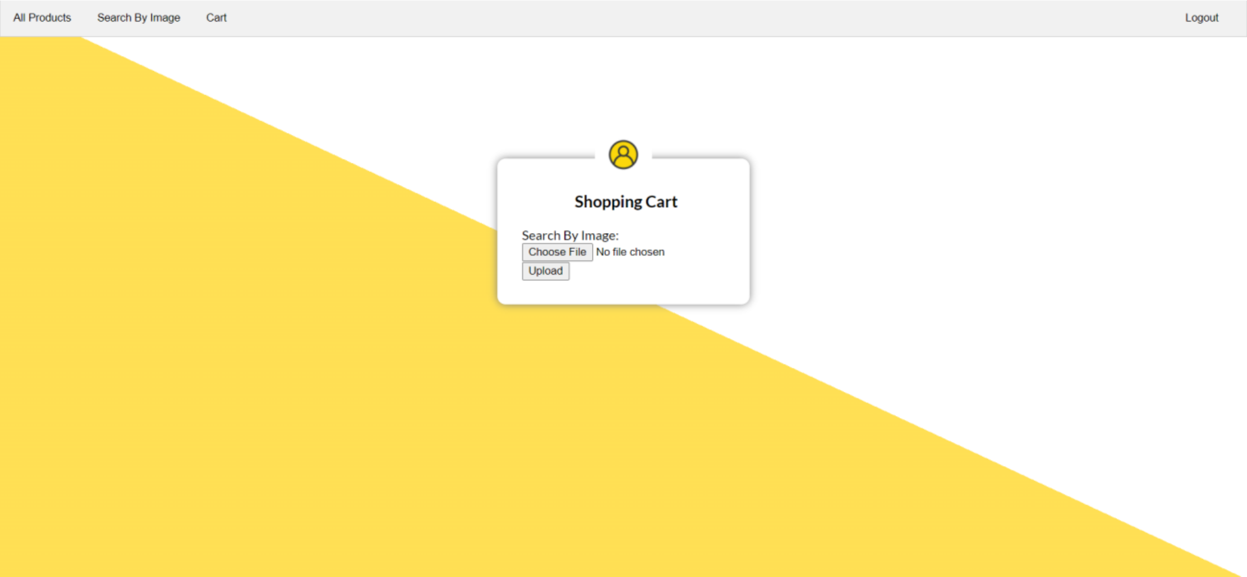


Laptop_2
BY DELL
XPS 2020
$800

Laptop_1
BY APPLE
Macbook 2021
$700

SEARCH BY IMAGE SCREEN:

Customer has also an option to search an product just be using a similar image of the upon, upon which the customer will be navigated to the respective product category page.



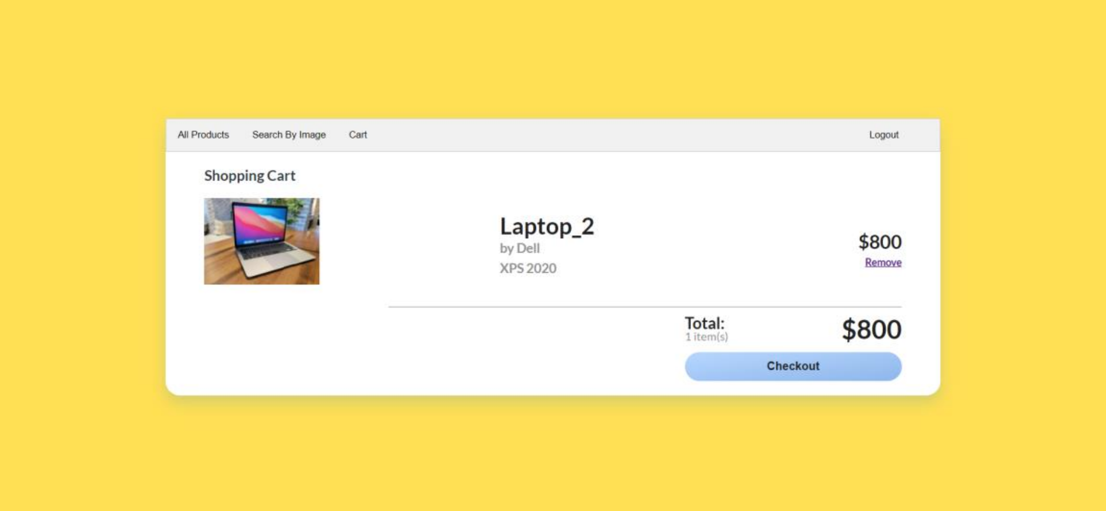
Shopping Cart

Search By Image:

No file chosen

Cart Page:

The customer would be able to look at all the items that were added to the cart in this page, along with the total amount.



Shopping Cart

Laptop_2
by Dell
XPS 2020

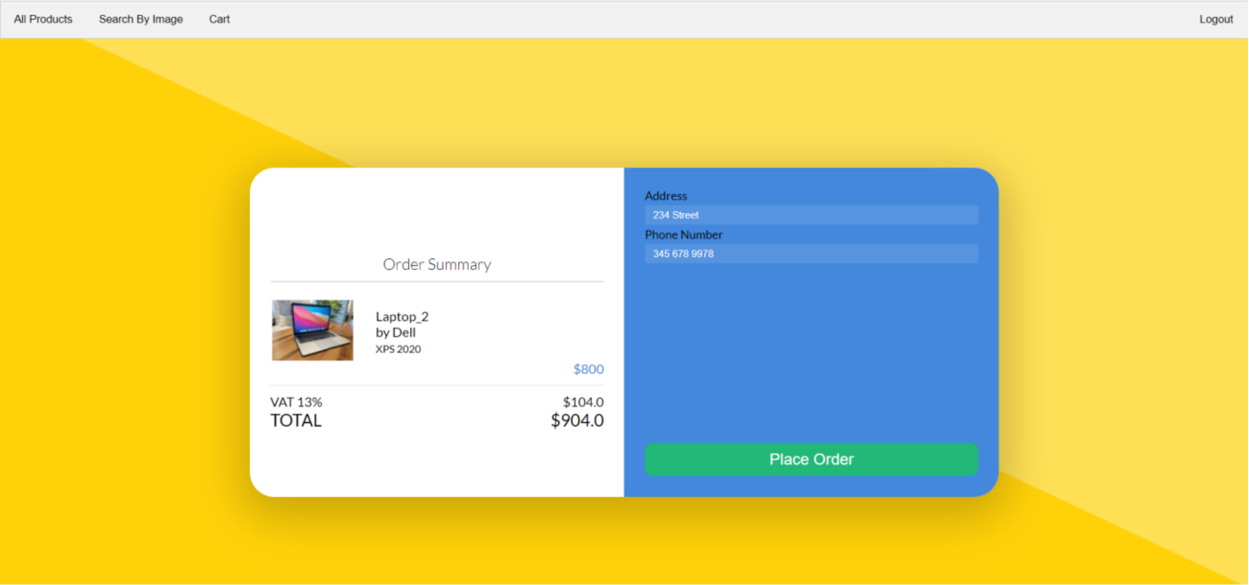
$800

Total:
1 item(s)

$800

CHECKOUT PAGE:

This page displays the customer the product he is about to order, along with the total cost along with tax. The customer will also be prompted to enter the details to which his products have to be delivered to.



Order Summary

Laptop_2
by Dell
XPS 2020

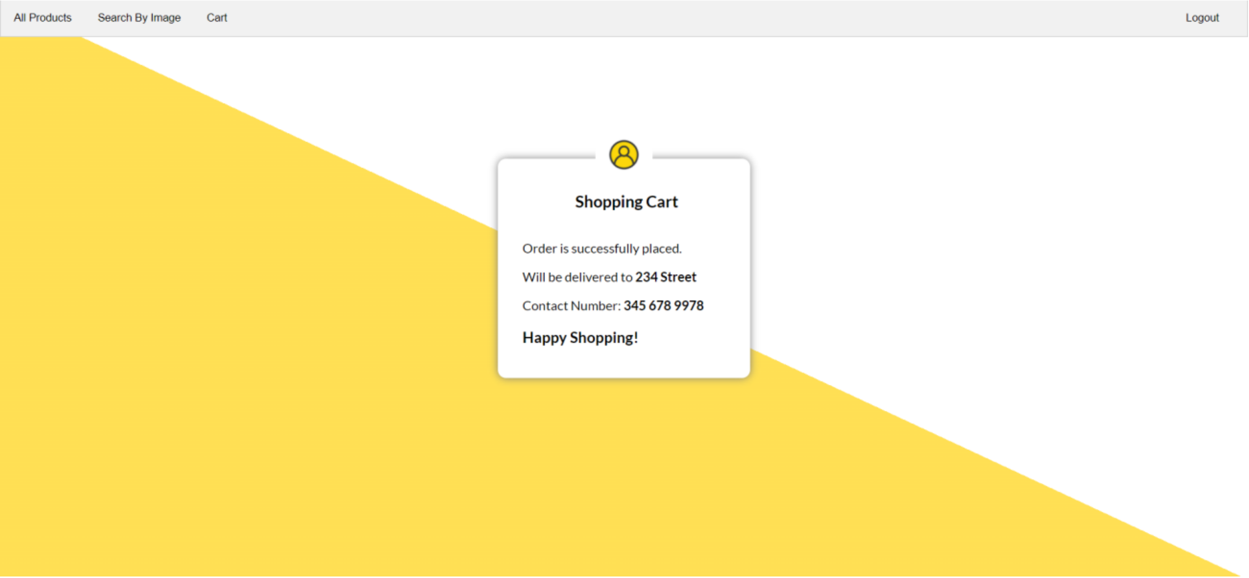
$800

VAT 13% $104.0
TOTAL $904.0

Address: 234 Street
Phone Number: 345 678 9978

CONFIRMATION SCREEN:

This page displays the information entered by the user and after this, the user would receive a mail about to order confirmation.



Shopping Cart

Order is successfully placed.

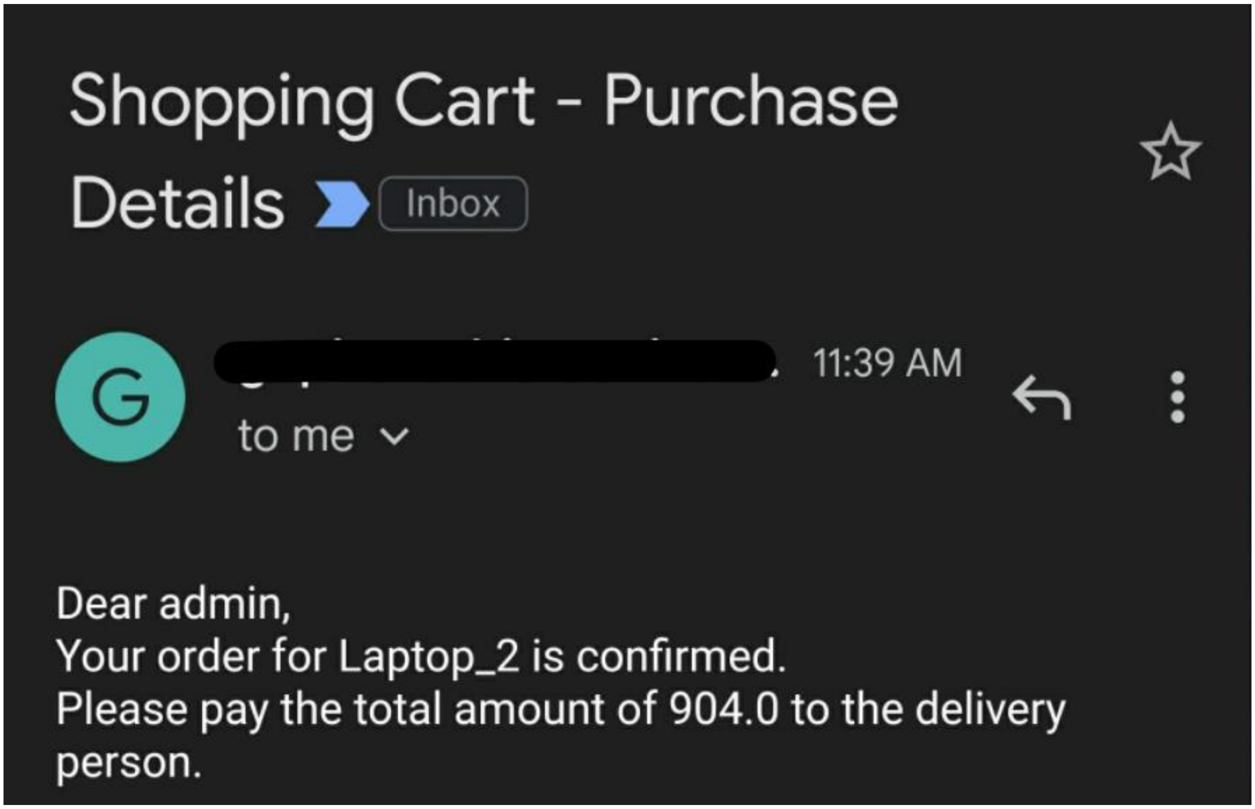
Will be delivered to **234 Street**

Contact Number: **345 678 9978**

Happy Shopping!

EMAIL NOTIFICATIONS:

TO CUSTOMER:



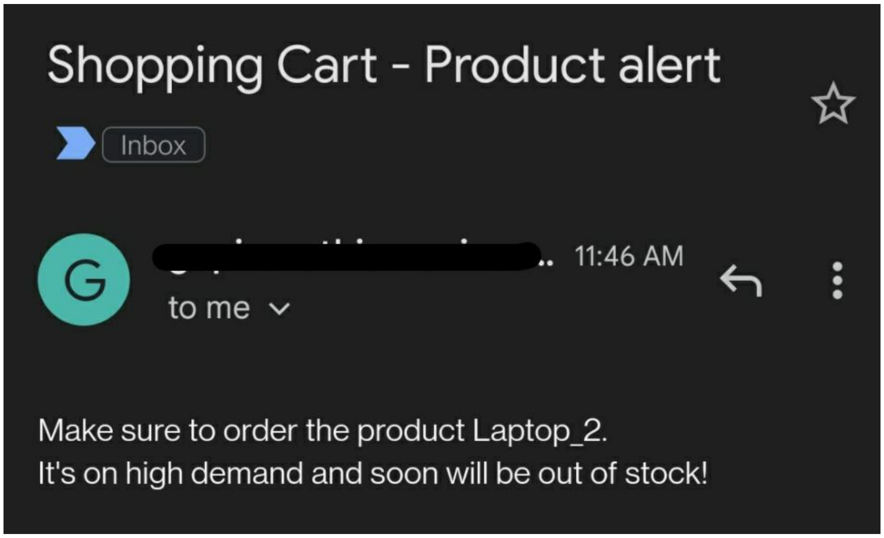
Shopping Cart - Purchase Details

[redacted] 11:39 AM

to me

Dear admin,
Your order for Laptop_2 is confirmed.
Please pay the total amount of 904.0 to the delivery person.

TO EMPLOYEE:



Shopping Cart - Product alert

[redacted] 11:46 AM

to me

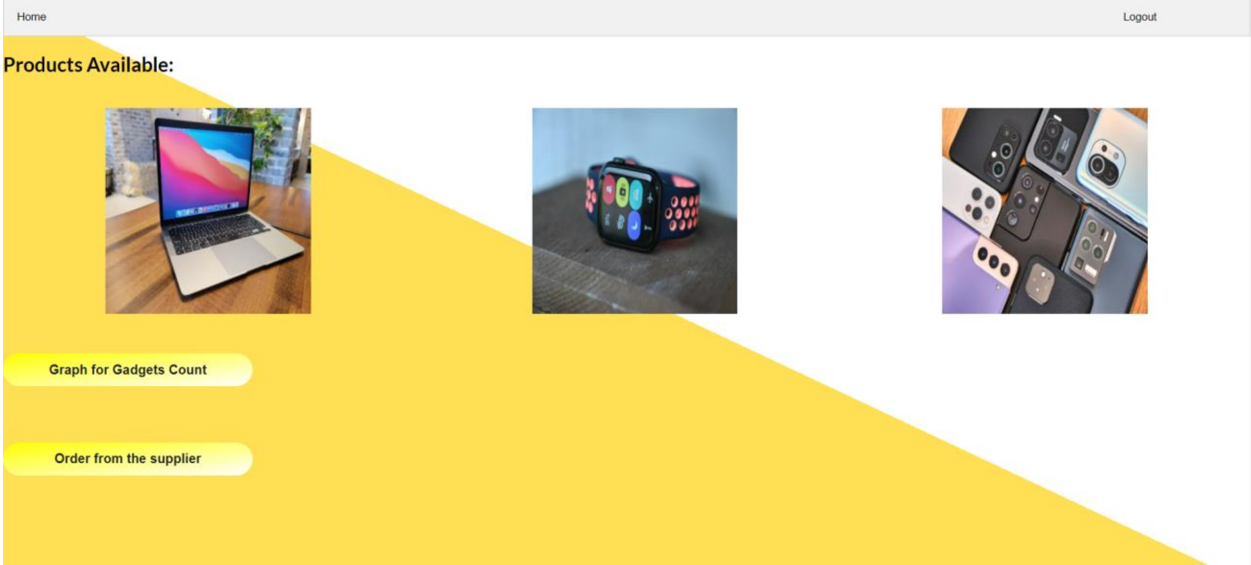
Make sure to order the product Laptop_2.
It's on high demand and soon will be out of stock!

EMPLOYEE USER GUIDE:

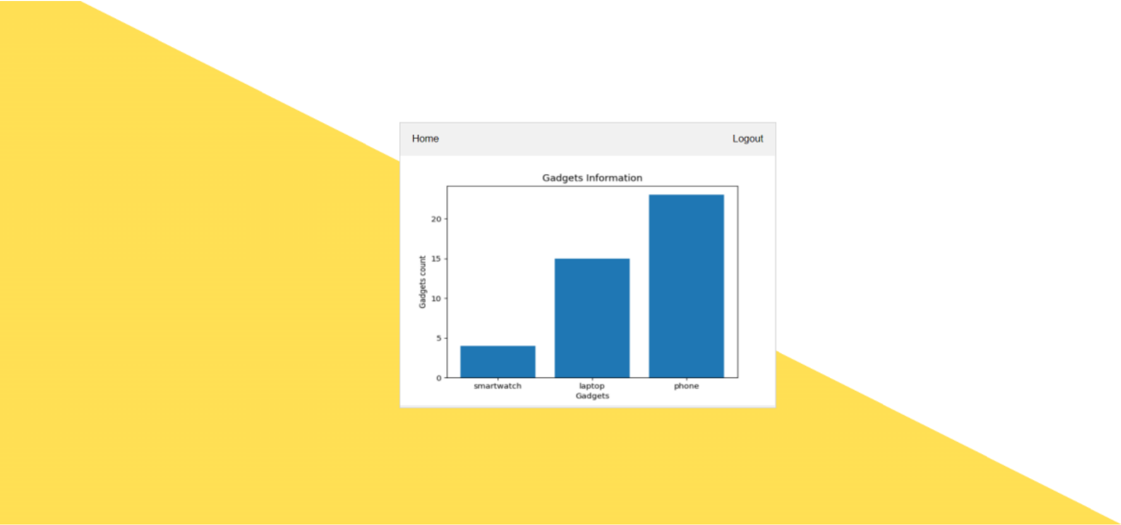
EMPLOYEE HOME PAGE:

This page is only accessible to the employee and the be will able to navigate to each product category screen from this page.

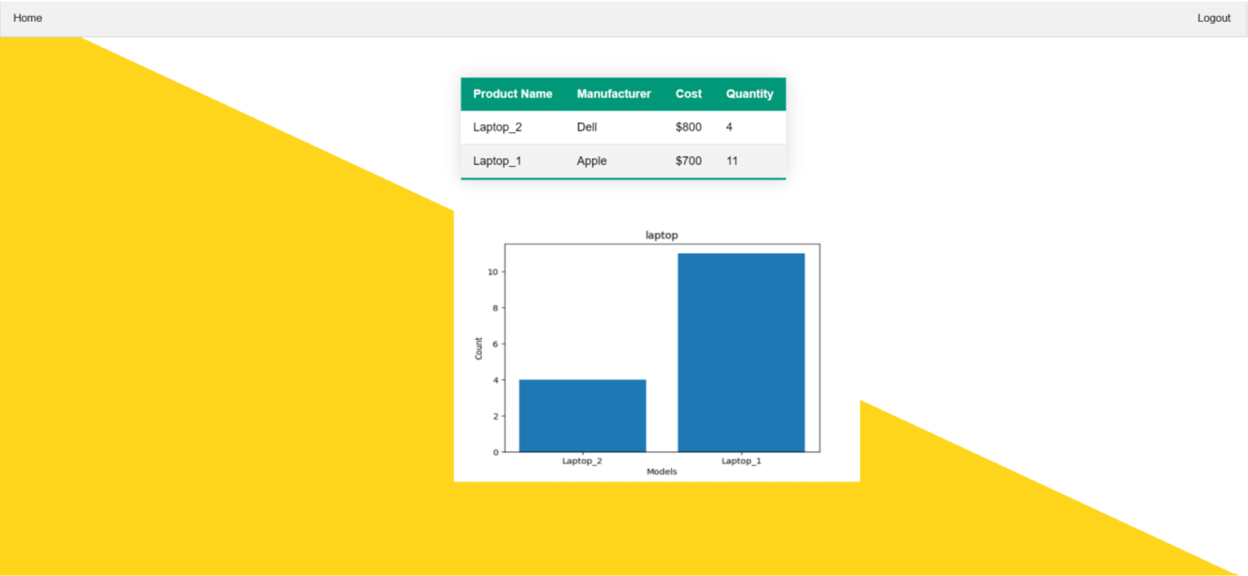
This page also contains two buttons from which the employee can navigate to overall reports page and the page from where the employee would be able to place order for every product to the supplier.



OVERALL REPORT SCREEN:



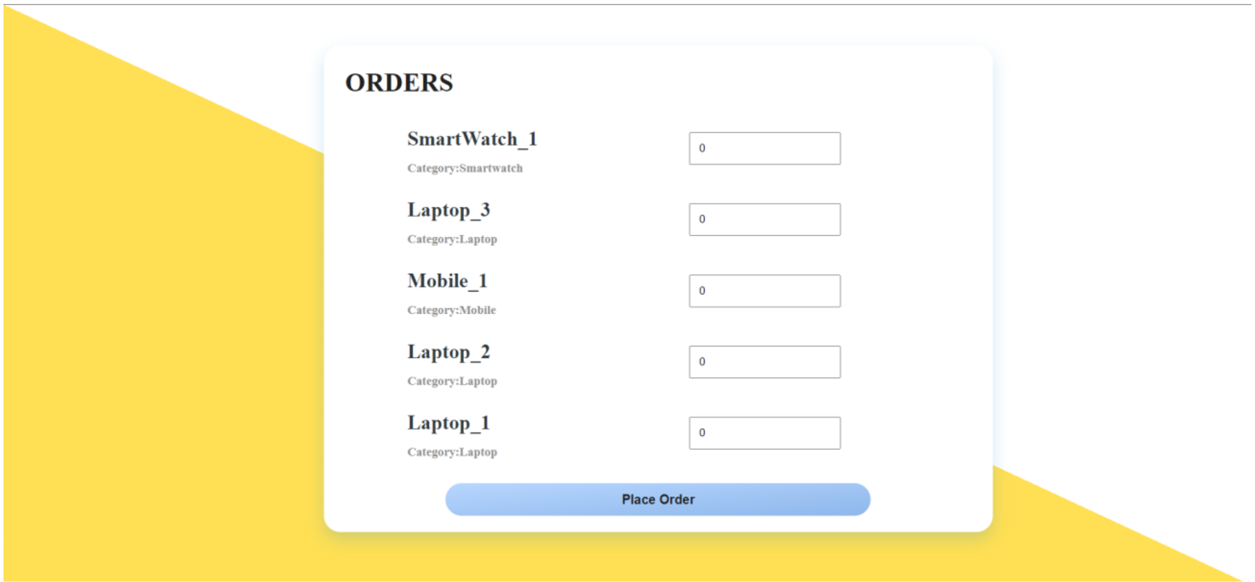
EACH PRODUCT CATEGORY SCREEN:



Product Name	Manufacturer	Cost	Quantity
Laptop_2	Dell	$800	4
Laptop_1	Apple	$700	11

PLACE ORDER TO SUPPLIER SCREEN:

The employee would be able to place order to the supplier from this page for any product whenever the product is in demand.



ORDERS

SmartWatch_1 Category:Smartwatch	0
Laptop_3 Category:Laptop	0
Mobile_1 Category:Mobile	0
Laptop_2 Category:Laptop	0
Laptop_1 Category:Laptop	0

Cost Modelling:

If initially when launching the application, there are 50 users who use our website frequently to buy products. With gradual increase in the users the users of the website after six months is considered 1000,000. We use AWS price calculator and AWS pricing documentation to estimate the price for our application. The AWS services we use, and the price estimation are as mentioned below.

AWS DynamoDB:

DynamoDB charges for functionalities that we use like read request, write request, data storage, data transfer etc. For our application we use read, write, data transfer and Data storage functionality.

Read and Write:

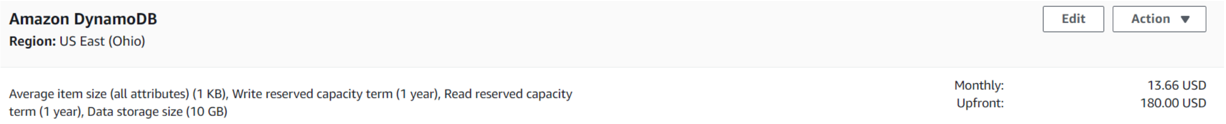
Write: $1.25 per million write request units

Read: $0.25 per million read request units

Data-storage: First 25 GB stored per month is free using the DynamoDB Standard table class. $0.25 per GB-month thereafter.

Data-transfer: AWS does not charge for data transfer between DynamoDB and other AWS services.

Considering there are 100 requests write and read from 10 user for 1 month, with peak request up to 400 one-month estimation of DynamoDB is as shown below.



Amazon DynamoDB
Region: US East (Ohio)

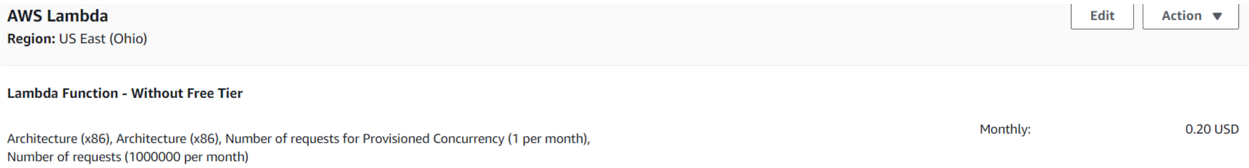
Average item size (all attributes) (1 KB), Write reserved capacity term (1 year), Read reserved capacity term (1 year), Data storage size (10 GB)

Monthly: 13.66 USD
Upfront: 180.00 USD

AWS Lambda:

Lamba charges price based on the number of requests for your functions and the duration it takes for your code to execute. The AWS Lambda free tier includes one million free requests per month and 400,000 GB-seconds of compute time per month. After free tier $0.0000166667 for every GB-second and $0.20 per 1M requests.

Assuming there are 1,000,000 requests from 10 user for the month, one month estimation of Lambda is as shown below.



AWS Lambda
Region: US East (Ohio)

Lambda Function - Without Free Tier

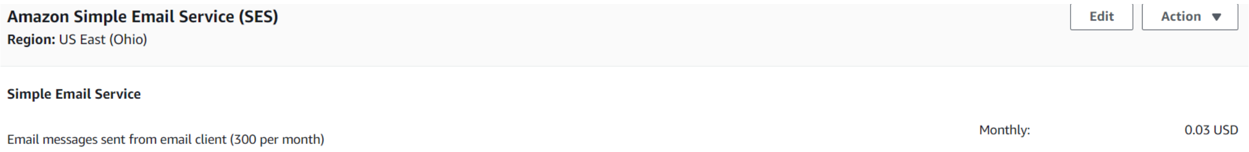
Architecture (x86), Architecture (x86), Number of requests for Provisioned Concurrency (1 per month), Number of requests (1000000 per month)

Monthly: 0.20 USD

AWS SES:

Assuming for single user 30 email is sent from email client. For 10 users one month estimation is as below. For an application hosted in EC2 amazon charges $0 for the first 62,000 emails you send each month, and $0.10 for every 1,000 emails you send after that.

For 10 user one month estimation of SES is as shown below. We assume for a month 10 users collectively receives 300 emails per month.



Amazon Simple Email Service (SES)
Region: US East (Ohio)

Simple Email Service

Email messages sent from email client (300 per month)

Monthly: 0.03 USD

AWS Rekognition:

Amazon free tier provides free processing of 5000 image per month. We use this service for users to find the required product with image upload. Amazon charges $0.001 per image for first 1 million images.

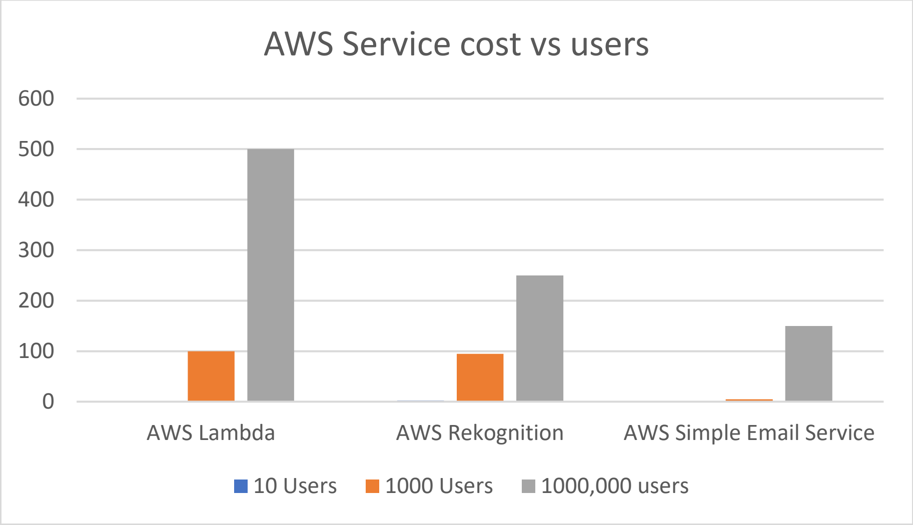
After six months:

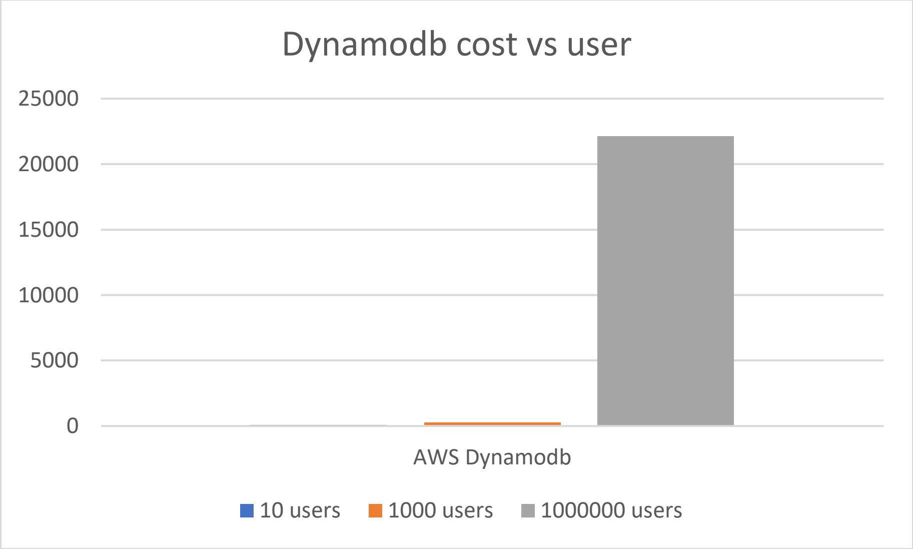
Assuming all the users are using our application frequently.

	10 Users	1000 Users	1000,000 users
AWS DynamoDB	81.96	270.81	22,127.75
AWS Lambda	1.2	100	500
AWS Rekognition	2	95	250

AWS Simple Email Service	0.18	5	150
TOTAL	85.34	470.81	23027.75

Below is the graphical representation of the data from table:


AWS Service cost vs users
AWS Lambda
AWS Rekognition
AWS Simple Email Service
10 Users
1000 Users
1000,000 users


Dynamodb cost vs user
AWS Dynamodb
10 users
1000 users
1000000 users