**A PROJECT REPORT** **ON**

## “GROCMAN – AN IOT BASED GROCERY MANAGEMENT SYSTEM”

SUBMITTED TO

**SHIVAJI UNIVERSITY, KOLHAPUR**

IN THE PARTIAL FULFILLMENT OF REQUIREMENT

FOR THE AWARD OF DEGREE

**BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY**

## SUBMITTED BY

|  |  |  |
| --- | --- | --- |
| 1.Mr. Makarand P. Sahastrabuddhe | | 17UIT12029XX |
| 2.Ms. Rutuja S. Mali |  | 17UIT11032XX |
| 3.Ms. Monika M. Salunkhe |  | 17UIT11038XX |
| 4.Mr. Suhel S. Mujawar |  | 17UIT12039XX |
| 5.Ms. Anuja S. Mane |  | 17UIT51058XX |
| 6.Mr. Kayyum Sanadi |  | 16UIT12072XX |

**UNDER THEGUIDANCE OF**

## Prof. T. I. Bagban

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**D.K.T.E. SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, CHALKARANJI**

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

Accredited with 'A+' Grade by NAAC, An ISO 9001: 2015 Certified

YEAR 2020-2021

### D.K.T.E. SOCIETY’S TEXTILE AND ENGINEERING INSTITUTE, ICHALKARANJI

(An Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

Accredited with 'A+' Grade by NAAC, An ISO 9001: 2015 Certified

# DEPARTMENT OF INFORMATION TECHNOLOGY



**CERTIFICATE**

**This is to certify that, project work entitled**

## “GROCMAN – AN IOT BASED GROCERY MANAGEMENT SYSTEM”

**is a bonafide record of project work carried out by**

|  |  |  |
| --- | --- | --- |
| 1.Mr. Makarand P. Sahastrabuddhe | | 17UIT12029XX |
| 2.Ms. Rutuja S. Mali |  | 17UIT11032XX |
| 3.Ms. Monika M. Salunkhe |  | 17UIT11038XX |
| 4.Mr. Suhel S. Mujawar |  | 17UIT12039XX |
| 5.Ms. Anuja S. Mane |  | 17UIT51058XX |
| 6.Mr. Kayyum R. Sanadi |  | 16UIT12072XX |

**In the partial fulfillment of award of degree, Bachelor of Technology in Computer Science and Engineering prescribed by Shivaji University, Kolhapur for the academic year 2020-2021**

### Prof. T. I. BAGBAN (PROJECT GUIDE)

**Prof. (Dr.) D.V. KODAVADE Prof. (Dr.) P.V. KADOLE**

**(H.O.D. C.S.E.) (DIRECTOR)**

### EXAMINER

# DECLARATION

We are undersigned students of B. Tech Information Technology and declare that the field work report entitled Grocman-An IoT based grocery management system written and submitted under the guidance of Prof. T.I. Bagban is our original work. The empirical findings in this report are based on the data collected by us. The matter assimilated in this report is not reproduction from any readymade report.

Date:

Place: Ichalkaranji.

NAME PRN SIGNATURE

1.Mr. Makarand P. Sahastrabuddhe 17UIT12029XX

2.Ms. Rutuja S. Mali 17UIT11032XX

3.Ms. Monika M. Salunkhe 17UIT11038XX

4.Mr. Suhel S. Mujawar 17UIT12039XX

5.Ms. Anuja S. Mane 17UIT51058XX

6.Mr. Kayyum R. Sanadi 16UIT12072XX

## ACKNOWLEDGMENT

With great pleasure we wish to express our deep sense of gratitude to prof. T.I. Bagban for his valuable guidelines, support and encouragement in completion of project report.

Also, we would like to take the opportunity to thank our head of department Dr. D. V.

Kodavde for his co-operation in preparing this project report.

We feel gratified to record our cordial thanks to other staff members of the Information Technology department for their support, help and assistance which they extended as and when required.

Thank you.

1.Mr. Makarand P. Sahastrabuddhe 17UIT12029XX

2.Ms. Rutuja S. Mali 17UIT11032XX

3.Ms. Monika M. Salunkhe 17UIT11038XX

4.Mr. Suhel S. Mujawar 17UIT12039XX

5.Ms. Anuja S. Mane 17UIT51058XX

6.Mr. Kayyum R. Sanadi 16UIT12072XX

## ABSTRACT

GrocMan- An IoT based grocery management system is a smart and new way of

grocery shopping that acts as our helper and re-defines the approach towards grocery shopping. Here we are going to measure the weight under the fixed value and according to that the order will be placed for shopping on that particular grocery list. This system introduces us to the simple implementation and usable application with the low-cost solution to the common man.

the purpose of any innovation and development is to provide comfort and reliability to our life. The rise of the term online Grocery shopping in recent years is due to the possibility of applying Internet of Things (IoT) for the home. Our system provides Smart grocery level management using Internet of

Things (IoT) and AWS. The smart containers which are incorporated with sensors are used to collect the data about grocery level in it. This data is stored in the cloud platform accordingly with the help of

Wireless protocol. This process takes place daily.

The information includes the stock level of all items, items which need to buy and the quantity need to buy and items which we use often etc.

## KEYWORDS

AWS services

AWS IOT

AWS Amplify

S3 bucket

Loadcell sensor

LCD 16x2

AWS DynamoDB

Arduino uno controller

REST API

System Testing

System Maintenance

# INDEX

**CONTENTS PAGE**

## 1. INTRODUCTION

1.1 Problem Statement 9

1.2 Need of the project with motivating example 10

1.3 Objective of Project

1.4 Limitations and Scope

1.5 Timeline for project 11

1.6 Cost of project 12

## 2. BACKGROUND STUDY AND LITERATURE REVIEW 14

2.1 Technology review

2.2 Literature review

## 3. REQUIREMENT ANALYSIS 15

3.1 Functional requirements

3.1.1 User Interface requirements

3.1.2 Hardware interface requirements

3.1.3 Modular/ Component Requirements

3.2 System Requirements

3.2.1 Hardware Requirements

3.2.2 Operating System Requirements

3.2.3 Application or Web server requirements

3.2.4 Storage requirement

3.2.5 Tools and Technologies Requirements

## 4. SYSTEM DESIGN 17

4.1 Architecture Diagram

4.2 Use case Diagram 19

4.3 Algorithmic description of each module 20

4.4 Data Flow Diagram 21

4.5 Sequence diagram 20 4.6 Class Diagram 22

**5. IMPLEMENTATION** 26

## 6. SOFTWARE TESTING 32

6.1 Unit test cases generation and its testing reports

6.2 Integration test cases generation and its testing reports

6.3 System test cases generation and its testing reports

1. **OUTPUT SCREEN** 37
2. **PERFORMANCE ANALYSIS**
3. **APPLICATION**  38
4. **CONCLUSION:**
5. **GUIDE TO ACCESS APPLICATION AND USER MANUAL** 39
6. **ETHICS** 44
7. **REFERENCES**

**14.PLAGIARISM REPORT** 45

## INTRODUCTION

**1.1 Problem Statement**

A key component in effective kitchen management is inventory control. Grocery management is managing the grocery items which we have in our kitchen. This is done by keep tracking of the grocery items in each container. In our kitchen we are having many grocery items in many forms. Like solids or liquids or in powder forms. Each item is stored in its each respective container.

These grocery levels may fall depending upon our usage in daily days. This usage may vary for every day. If their level goes down, we have bought and again have to store. Grocery Management includes keep tracking their levels every day irrespective of the container and forms and indicating us that this particular grocery level is low which you have to buy if their level goes down.

In this hectic world people may don’t have the time to check into the grocery levels. If they continue using the grocery items without checking into it, one day they will gets emptied without their knowledge. At that time, it will affect the cooking process. If they are on the process of cooking, some dishes cannot be done without some grocery items.

In that situation they won’t have the time to go to the grocery store to buy that one particular item in order to complete cooking. It leaves only anger and frustration if they are in a hurry to complete it. Finally, the entire cooking will become mess. This is why we have kept track of the grocery items in order to manage it.

### 1.2 Need of the project with motivating example

Nowadays life for everyone has become so hectic and time consuming, at such times we require a smart system in our kitchen also. To put on records and observe all the grocery at home is difficult.

Most of the time we remain in the wrong belief that we have enough groceries in

our kitchen but we have to face empty bottles at the time of emergency when the requirement is must that gives us inconvenience. And to avoid this, sometimes we buy more than enough grocery & store it at our home for many days, which is also an inconvenience and can cause damage to the grocery. Both these situations are problems.

Systems that can give continuous level measurement and can notify us about low levels of content are required to avoid these problems.

#### 1.3 Objective of Project

1.Build Smart containers which can measure weight and which are connected to Wi-Fi.

2**.**Store the status of groceries posted by container to the AWS cloud.

3. Calculate the ‘Out of Stock value’ and add those groceries to the cart.

4.Purchase the items added to the cart according to user’s approval.

#### 1.4 Limitations and Scope

1.4.1 Existing System

Existing system contains the facility to shop groceries and order it online but, in these systems, there is a need to add grocery items manually in the cart that we need to purchase.

1.4.2 Limitations

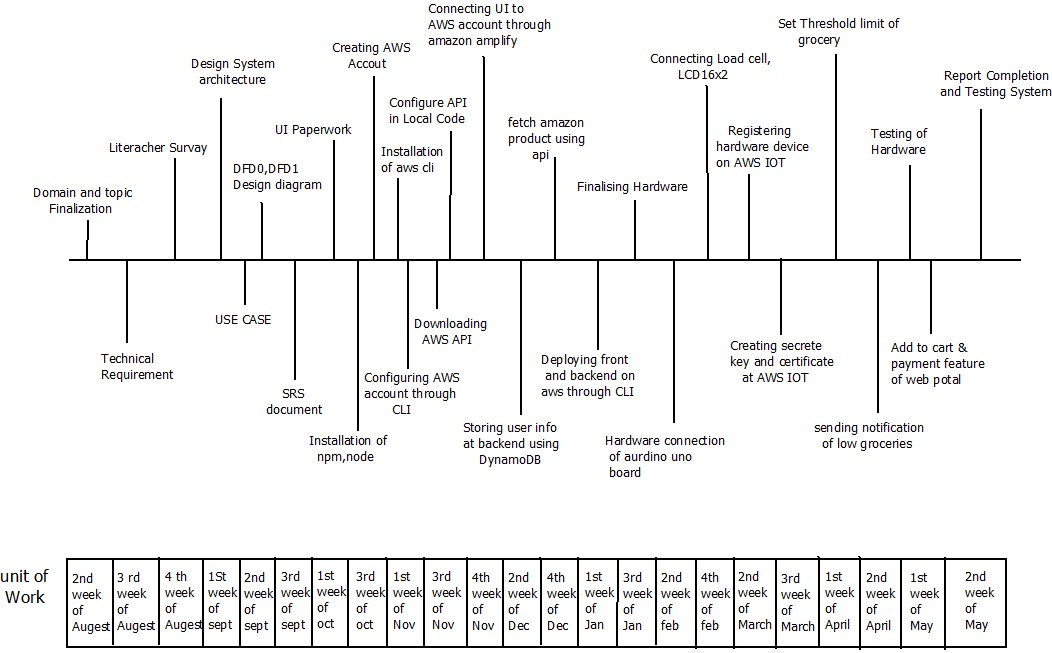
Existing system has following limitation-

* Limited list of groceries.
* Need manual approval of items.
* No android application available.
* Lack of Security.

1.4.3 Scope

1. Platform for solving monthly budget planning.
2. Users can plan their money for all needs.
3. Gain popularity by more interactivity with the platform.

#### 1.5 Timeline for project



**1.6 Cost of project:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Equipment | Details | Price (Rs) |
| 1. | AWS  charges | Consumed 8 services | 500 |
| 2. | Internet | 100 GB | 1000 |
| 3. | IoT kit & containers | Arduino, raspberry pi, LCD 16 X 2  , Load cell Weighing  Sensors | 2000 |

Estimated cost by considering other factors will be approx. **- Rs. 3500/-**

1.5.1 COCOMO Model

In this project, the Cost Estimation based on COCOMO (Constructive Cost Model) the formula for this Model is follows:

Effort = Constant × (Size) scale factor× Effort Multiplier

* Effort in terms of person-months
* Constant: 2.45 in 1998 based on Organic Mode
* Size: Estimated Size in KLOC
* Scale Factor: combined process factors
* Effort Multiplier (EM): combined effort factors

Functional Point Table

The function point range in between 1-10

Conversion of Functional point to Lines of Code (LOC) Total function points = 6

■ Estimated Size – 7820 LOC

The basic COCOMO equations take the form

Effort Applied (E) = ab (KLOC) bb [man-months]

Development Time (D) = cb (Effort Applied) db [months]

People required (P) = Effort Applied / Development Time [count]

Where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for a project. The coefficients ab, bb, cb and db are given in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ab | bb | cb | db |
| **Organic** | 2.4 | 1.05 | 2.5 | 0.38 |
| **Semi-detached** | 3.0 | 1.12 | 2.5 | 0.35 |

Organic Mode:

Effort Applied (E) = 2.4\*(7120) \*1.05=17942.4

Development Time (D) = 2.5\*(273) \*0.4= 273.0

People Required (P) = 17942.4/273.0 = 6 people

## 2. BACKGROUND STUDY AND LITERATURE REVIEW

### 2.1 Technology review

Amazon web services is a subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a pay-as-you-go basis.

Amazon Web Services offers a broad set of global cloud-based products including compute, storage, database, analytics, networking, mobile, developer tools, IoT, security and enterprise applications. These services help organizations move faster, lower IT costs, and scale.

We have used AWS IoT which is a managed cloud service that lets connected devices easily and securely interact with the cloud applications and other devices. And Amazon Cognito that lets you easily add user sign-ups and authentication to your web apps.

Also, we have used AWS DynamoDB which is a NoSQL database system. It has automatic scaling according to your load and no need for servers to manage. AWS platform and its services helps our grocery system to be serverless.

### 2.2 Literature review

We found many of the literature papers based on “Smart grocery management system” Some of the literatures are as follows-

2.2.1. Smart grocery management (Research paper 1 - IJRET)

Here system was developed using Intel Adison Arduino board and HC-SR04 ultrasonic sensor is used for level measurement which gives the signal of low level to the board. And messages were sent by MQTT protocol for communication to the cloud connectivity.

2.2.2. Smart kitchen automation (Research paper 2 - IJRET)

Smart kitchen automation system consists of ultrasonic sensor, Arduino microcontroller, Wi-Fi module and cloud. Grocery level information is pushed into the cloud using the Wi-Fi module. In the cloud the data is stored and analyzed. This analyzed result is visualized using a mobile application.

## 3. REQUIREMENT ANALYSIS

### 3.1 Functional requirements

3.1.1 User Interface requirements

1. User must create his account on the AWS console.
2. Users should be able to log in and then authenticate themselves on the AWS platform.
3. Users should be able to add a list of items.
4. Sensors should continuously check the weight of the grocery items.
5. This data should be stored in AWS cloud.
6. Users can continuously check the weight of groceries through sensors.
7. When the grocery weight goes below threshold level then the system should send the notification to the user through mail.
8. Users can place the order of groceries which is below threshold level.
9. Users can store and retrieve data on the cloud.

3.1.2 Hardware interface requirements

1. Loadcell sensor - Used to measure the weight of grocery items.
2. LCD 16x2 - LCD screen for displaying weight at client side.
3. Arduino uno controller - Arduino kit used to fetch data on cloud.

3.1.3. Modular/ Component Requirements

Project contain following modules/components which are the backbone of Grocery management System -

1. Smart Container - This component is used to place the item so that we can check the weight through sensors.

2. AWS CLI - This module is used to configure AWS on a local machine.

1. Weight Sensors - This component is used to check the weight of an item.
2. Amazon Web Services - We are developing Grocery management on AWS platform.
3. AWS Cloud - This module is used to access different services and to store the data.
4. DynamoDB - This module is used to store all information of item payments.
5. AWS IOT- This service is used to register hardware and fetch the data on cloud.

### 3.2 System Requirements

3.2.1 Hardware Requirements

1. Machine with minimum 8GB of RAM for developing environment and formation of the network.

3.2.2 Operating System Requirements

1.windows/ubuntu

3.2.3 Application or Web server requirements

1. Any type of browser which can be used to access the application, AWS CLI is used for configuration.

3.2.4 Storage requirement

1. We have used the cloud storage to store the data of users and grocery information.

3.2.5 Tools and Technologies Requirements

1. Node – Install the npm packages
2. React.js - for building the applications
3. AWS cli - to configure AWS on a local machine.
4. DynamoDB - to store the data of users and item information.
5. AWS IOT- for hardware connection

## 4. SYSTEM DESIGN

4.1 Architecture Diagram

Following diagram shows the interaction between components and modules of the project

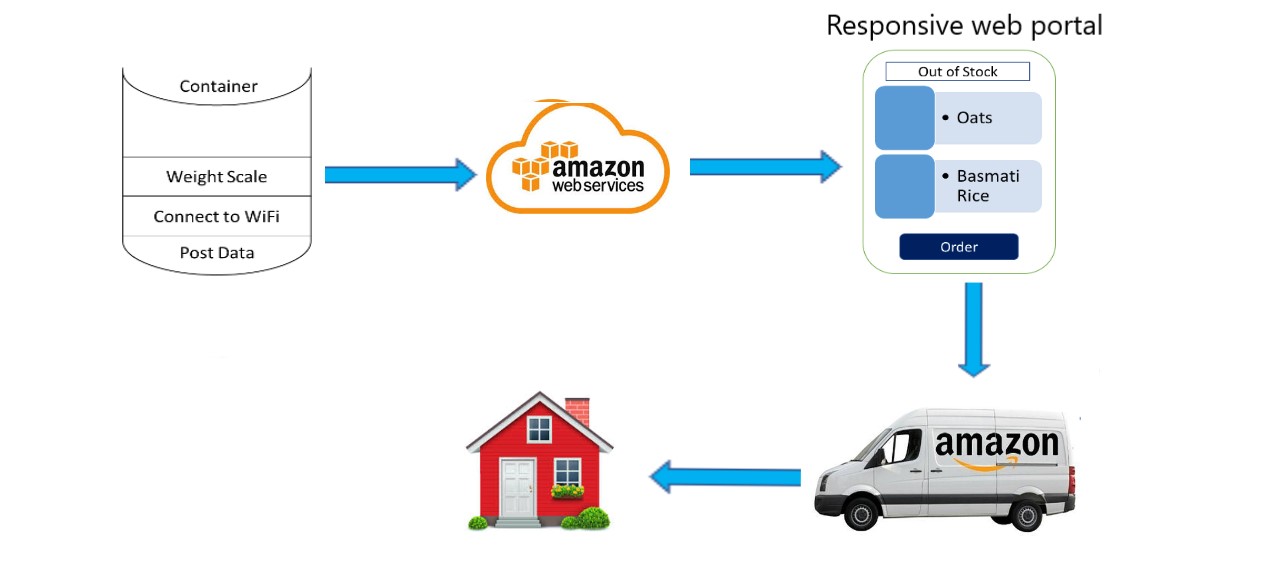


Fig 4.1 Architecture Diagram of Application

4.1.1 Components in Architecture diagram

1. User Interface

It is a responsive web online portal User Interface. By using this portal

every user can identify the inventory status of all the grocery items in the house at a glance and order it on a click of a button from the comfort of your home.

1. Container

This system contains an (IoT device) Smart Container that alerts the user when they are low on household grocery items. It stores the grocery items at home and is connected to a load cell to measure the weight.

1. AWS IoT

IoT device with load cells to be able to measure the weight and post the data on AWS Cloud. This IOT service helps the user to track on grocery items and send notifications to the user by using the MQTT protocol.

1. Place the order

When a user gets notified by system, he/she is able to place the order of that grocery item. Also, the user is able to add other items also and place the order and order will be delivered through amazon delivery service at specified address.

* 1. Use Case Diagram



Fig 4.2 Use Case Diagram

In the above diagram, main functionalities are shown. These functionalities are used by the users to manage the groceries at home

4.2.1 Algorithmic description of each module

1) Grocery item Notification-

Input – grocery stored in container which is connected through load cell

Output – Notification of low groceries

Description- This algorithm is based on grocery stored at home and the threshold Limit set by the user using AWS IOT.

1. Send user data on cloud-

Input – user data (grocery weight)

Output – data stored on cloud

Description- This algorithm is used to monitor the weight of grocery and

Store it on aws.

1. Place the Order -

Input – Items added to cart

Output – order placed successfully

Description- This algorithm is used to place the order of required grocery Items.

* 1. Data Flow Diagram
     1. DFD level 0

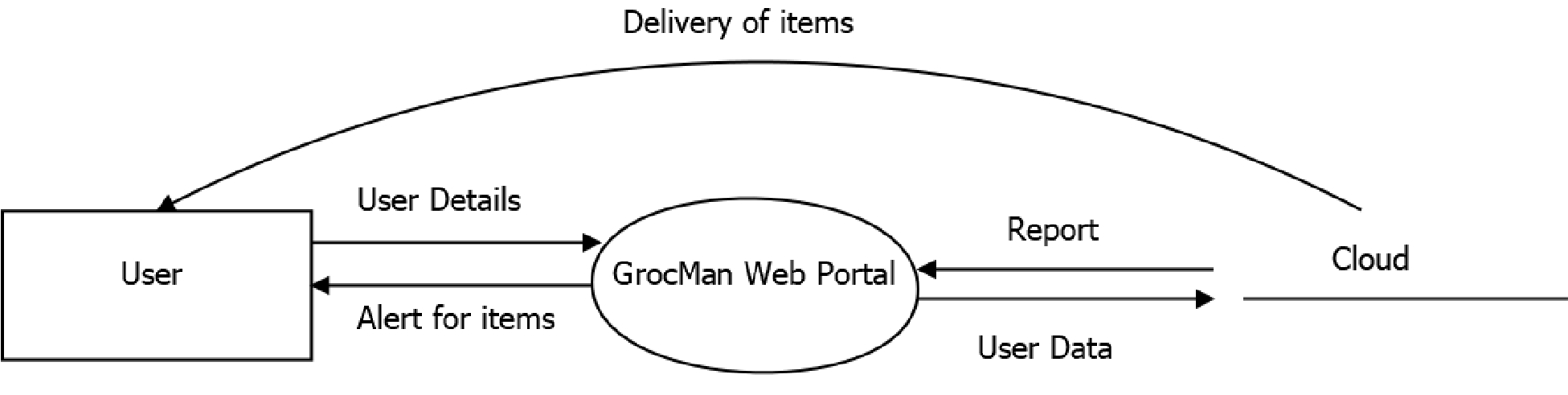


Fig 4.3.1 Diagram of DFD Level 0

As shown in the diagram firstly, users have registered on a web portal then and then only he/she is able to access the portal. And the user will get alert when they are low at grocery items.

* + 1. DFD level 1

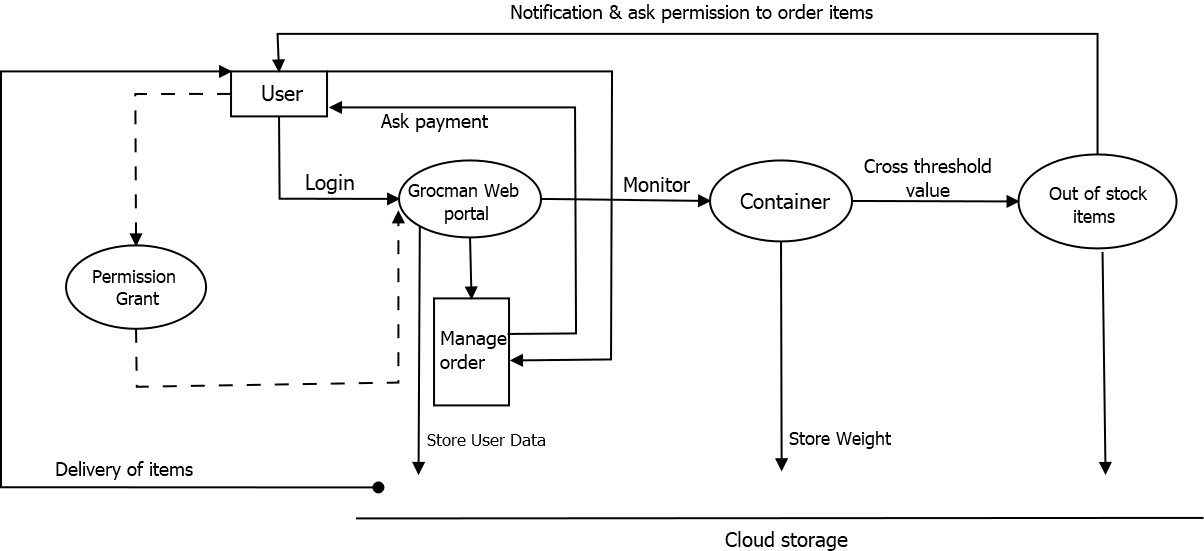


Fig 4.3.2 Diagram of DFD level 1

As shown in the diagram the specific functionality of each module is given. The corresponding input and output are shown on the arrow. The arrow shows the flow of the system

* 1. Sequence diagram

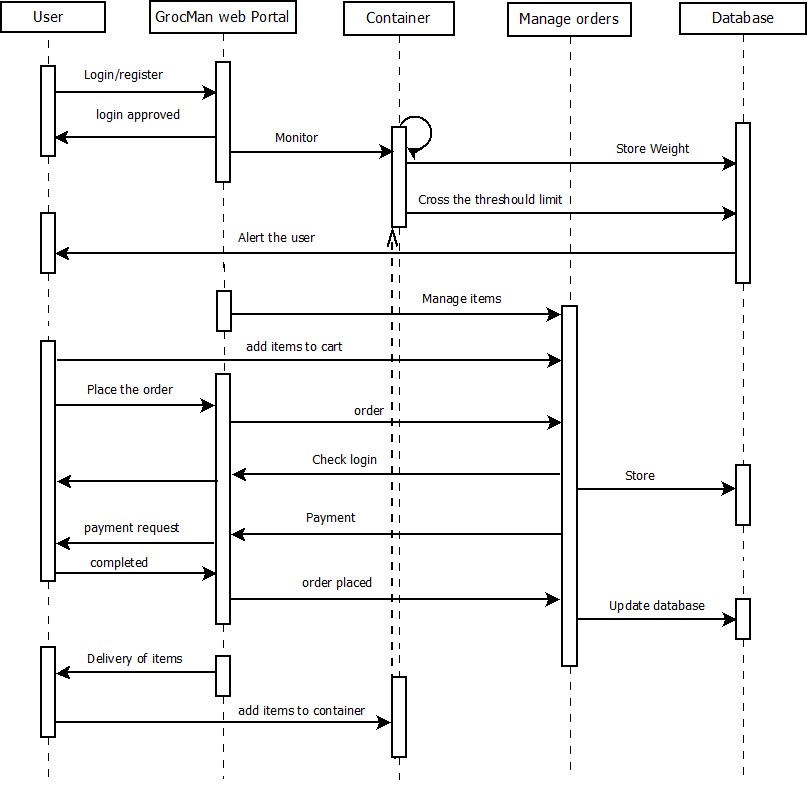


Fig 4.4 Sequence Diagram of Application.

In the above diagram the sequence of several activities is shown which are carried out in a specific manner.

* 1. Class Diagram

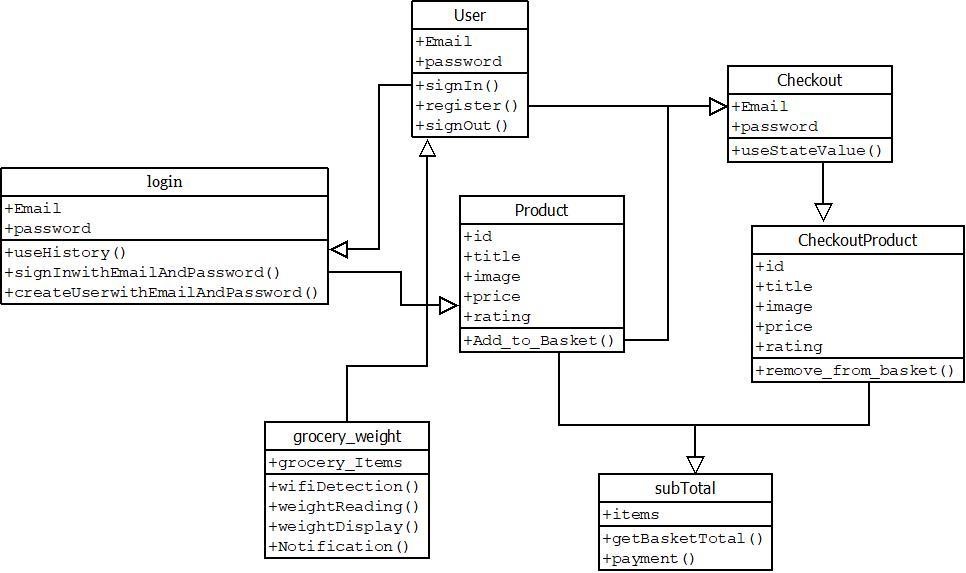


Fig 4.5 Class Diagram of Web application

There are seven classes named user, login, product, checkout, checkoutProduct, subtotal, grocery\_weight containing functions as shown in above figure.

* + 1. Description of each class in above figure is as follows,

### 1. class name: users

Description: User has unique ID i.e., Email id and password after registering with the web

app.

Data Members: EmailId, passwordMember Function Description: Users can register themselves, sign out from the application whenever they want.

### 2. class name: login

Description: While login or creating an account user should enter EmailId and password. And data provided by the user must be validated for security and to remove redundancy.

Data Members: EmailId, password

Member Function Description: User should firstly register with the web application. Then the user can log in into the application. Authentication is done while registering with the app.

### 3. class name: Product

Description: Users can choose products available on the web portal.

Data Members: EmailId, title, Image, price, rating.

Member Function Description: User can select the product whichever they want from the web portal. Each product item is provided with its information.

### 4. class name: Checkout

Description: Before going to finalize the items, system will check login credentials.

Data Members: EmailId, password.

Member Function Description: Before going to finalize the list of grocery system need to check the user has logged in with his own account credentials for security and better service.

**5. class name**: CheckoutProduct

Description: user will finalize the selected items.

Data Members: items present in basket.

Member Function Description: User can finalize the items and can remove the items from the basket as per user’s convenience.

### 6. class name: SubTotal

Description: Total amount will be displayed to the user.

Data Members: items present in basket.

Member Function Description: System will calculate the total bill for the user and the user needs to complete that payment to place the order.

**7. class name**: grocery\_weight

Description: user will get alert when they are low at grocery items.

Data Members: items in container.

Member Function Description: User will get a notification to alert the user when they are low on household grocery items including weight of that grocery.

### 5. IMPLEMENTATION

#### 5.1 Environmental settings for running the module –

1. Install NodeJS, ReactJs, npm.
2. Install AWS CLI.
3. Creating AWS account.
4. Creating and Configuring amazon S3 bucket.
5. Deploy the code through aws amplify.
6. Interconnection of hardware
7. Register hardware on AWS IOT.

8. Run the application.

#### 5.2 Installation

##### 5.2.1 Installation of Node

Node provides the basic infrastructure for installation of dependencies.

Commands to install node –

sudo apt-get update sudo apt install NodeJS

The NodeJS packet contains both node and npm binaries.

##### 5.2.2 NPM

1. NPM - Node Package Manager

Npm is installed with Node.js. It can install all the dependencies of a project. Npm is used for package installation, version management, and dependency management.

Commands to install packages through npm install <package name>

1. NPX- Npm Package Runner

Commands to install packages through npm <package name>

Create application using npx

create-react-app my-app

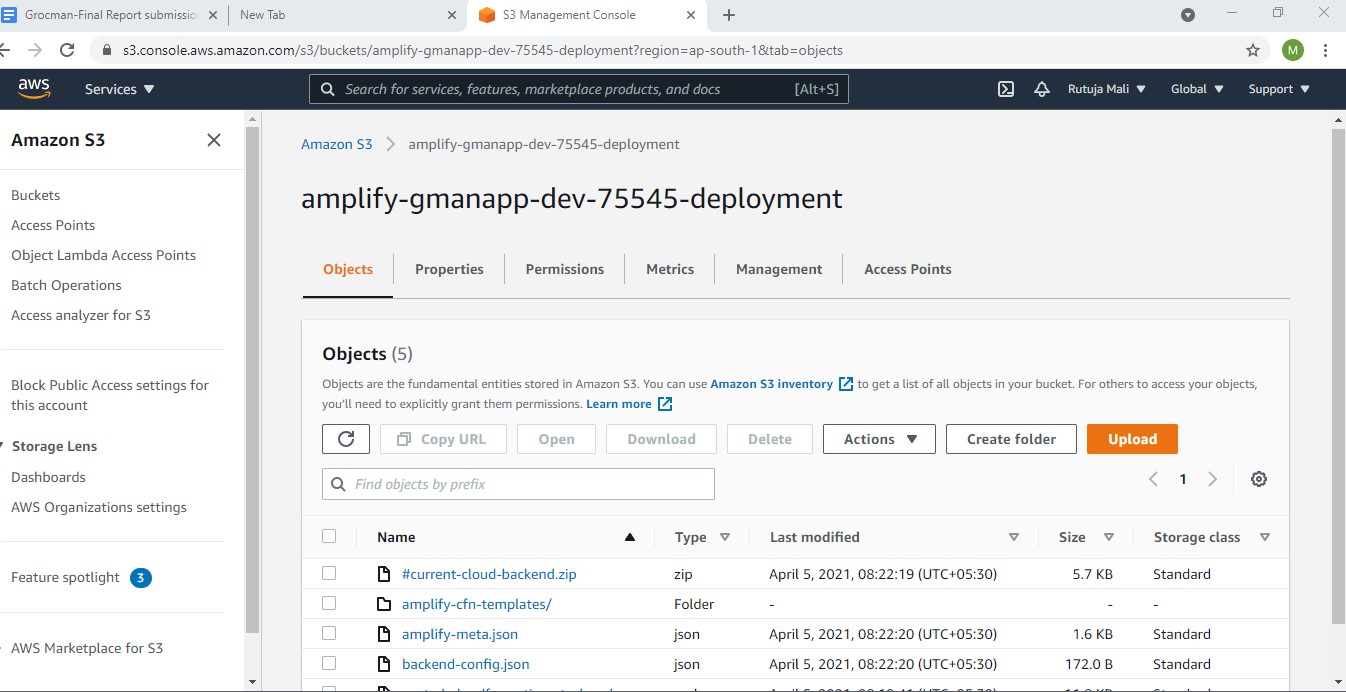
Npx will search for that package in the local and global registry

#### 5.3 Creating Account

After installation of software tools, we need to create an aws account for further Implementation.

#### 5.4 Creating and Configure amazon S3 bucket

To upload your data to Amazon S3, we must first create an Amazon S3 bucket in one of the AWS Regions. When we create a bucket, we must choose a bucket name and Region.



The above figure is displaying amazon s3 bucket named as ‘amplify-gmanapp-dev75545-deployment’ where the data will be stored and accessed.

#### 5.5. Deploy and configure through aws amplify

Amplify is an open-source library (under Apache 2.0) for interacting with cloud services that use JavaScript applications. Amplify comes with a with Authenticator for sign in-sign out functionality.

Amplify support-

1. **Auth**: Provides AWS credentials for signing as well as OIDC JWT tokens from

Amazon Cognito. Common functionality, such as MFA features, are supported.

.

1. **Analytics**: Tracking for authenticated or unauthenticated users in Amazon

Pinpoint. Extend this for custom metrics or attributes as you wish.

1. **API:** Interaction with RESTful APIs that leverage AWS Sigv4 in a secure manner. The API module is great on serverless infrastructures with Amazon API Gateway.
2. **Storage**: Simplified commands to upload, download, and list content in Amazon S3. Additionally, you can easily group data into public or private content on a per-user basis.
3. **Caching**: An LRU cache interface across web apps and React Native using implementation-specific persistence

- The AWS Mobile CLI can be installed from NPM:

Command to installnpm install -g awsmobile-cli

Import into the project using following way import Amplify from 'aws-amplify'; import aws\_exports from './aws-exports'; Amplify.configure(aws\_exports);

#### 5.6 Interconnection of hardware

To measure the weight of grocery items, present in container, we need to build the connection of hardware.

1. We need to attach a Loadcell sensor to the container for weighing.
2. Display weight to the user needs to connect the LCD which is attached with the

container.

1. This sensor and Lcd are connected to the Arduino uno controller.
2. we need to supply the power and Wi-Fi connection to fetch the data.

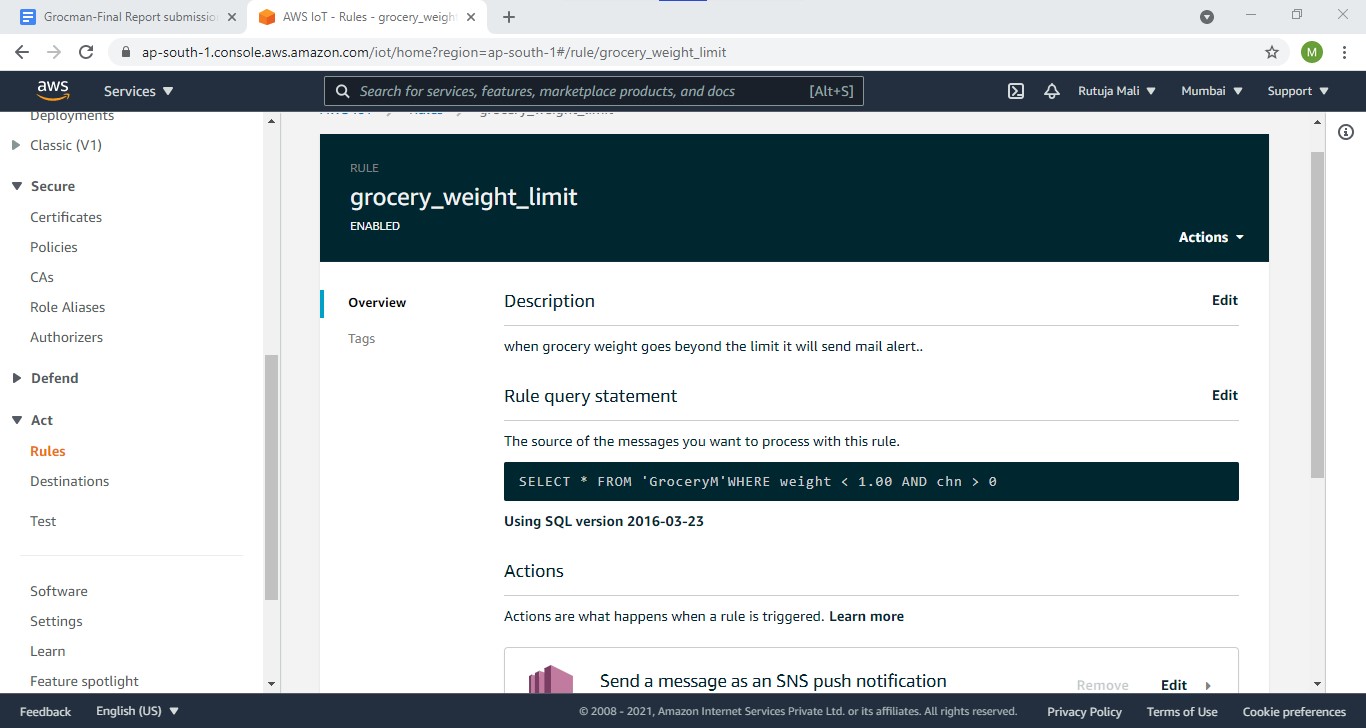
#### 5.7 Register hardware device on AWS IOT

To access the data from sensors on cloud we need to register the device on AWS IOT,

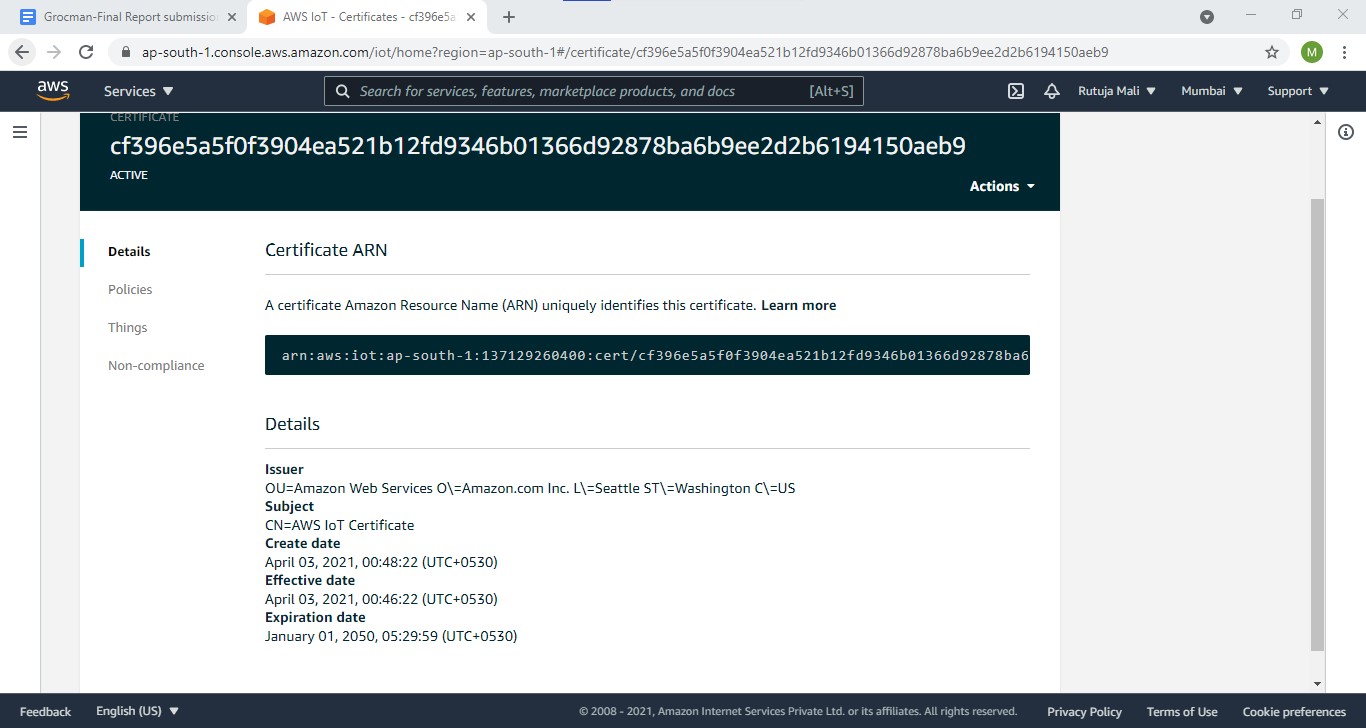
For this first we need to generate the certificate. On this portal we have set threshold limit and by using MQTT protocol can send notification to the user.

Query statement to set threshold limit-

SELECT \* FROM 'GroceryM'WHERE weight < 1.00 AND chn > 0



Above image describes the procedure of setting the threshold value

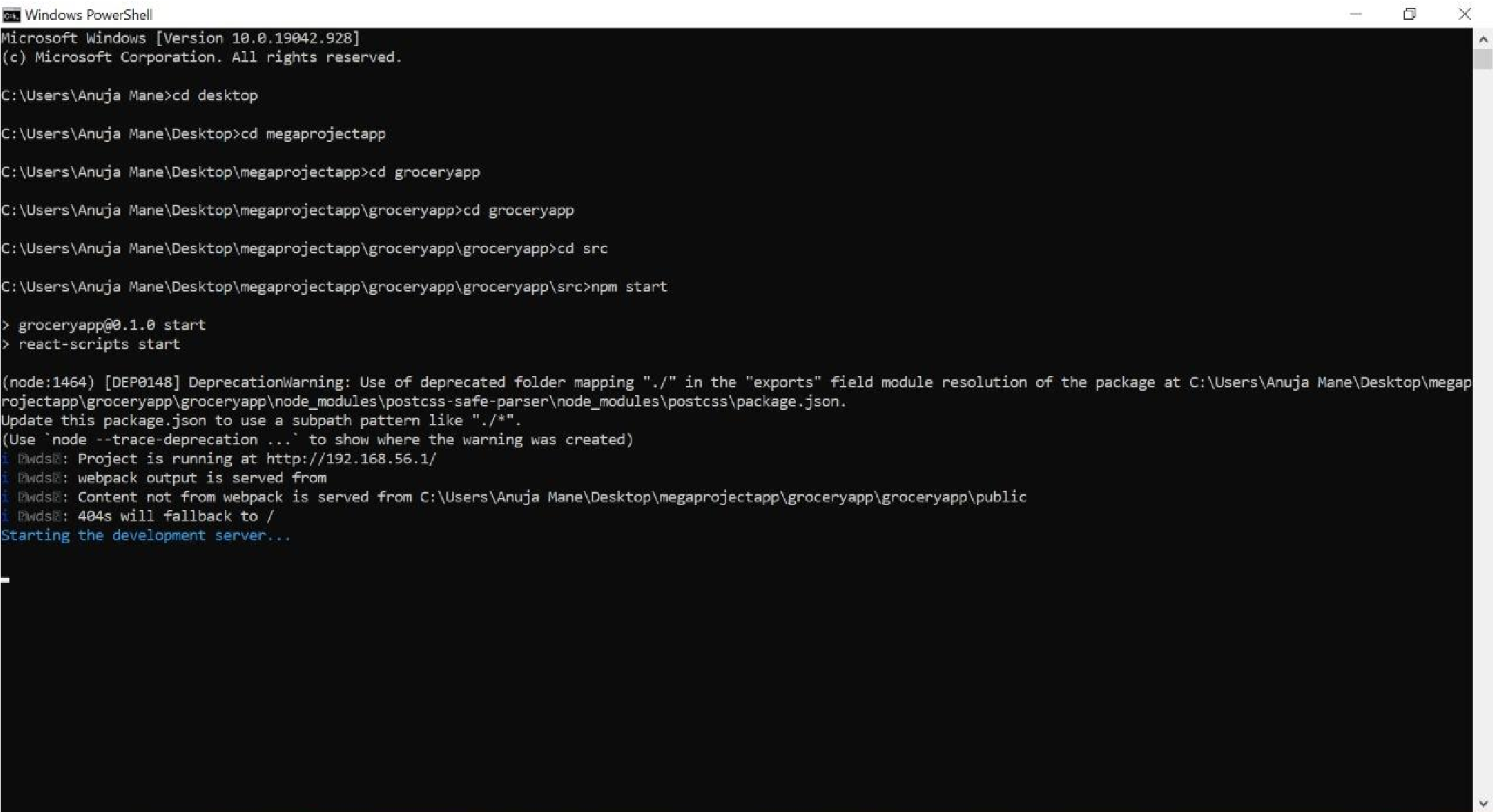


Above figure is displaying certificate for registering hardware device which provide secret key and this secret key is configured with a hardware device.

#### 5.8 Run the application

Run the application using command-

-npm start



Above image describes the way to start the application.

### 6. SOFTWARE TESTING

**6.1 Unit test cases generation and its testing reports**

6.1.1 Test cases for registration

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | Registration | Name,  Password, | Registration successful | Registration successful | Pass |

6.1.2 Test cases for Login

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | If User is  already registered | Name,  Password | Successful login | Successful login | Pass |
| 02 |  | If user is not  registered | Name,  Password | Give error message & redirect to sign up page | Give error and redirect  to sign up  page | Pass |

6.1.3 Test cases for add/remove item to the cart

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | Grocery item is added to  the basket | Add to cart | Successfully added to cart and to the  subtotal | Successfully added to cart and to the  subtotal | Pass |
| 02 |  | Grocery item is removed from the  basket | Remove  from the cart | Successfully removed from the cart and subtracted  from the  subtotal | Successfully removed from the cart and subtracted  from the  subtotal | pass |

6.1.4 Test cases for billing of a cart

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | Billing of the grocery items in the cart | Price of all the grocery items added  to the cart | Final bill of all the  grocery items in the cart. | Final bill of all the  grocery items in the cart. | Pass |

6.1.5 Test cases for weight measuring of an item

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | Weight of the grocery item | Weight | Weight is  successfully stored to the database | Weight is  successfully stored to the database | Pass |
| 02 |  | Weight of the grocery item below threshold value | Weight | User gets the message about less  grocery item | User gets the message about less  grocery items. | pass |

#### 6.2 Integration test cases generation and its testing reports

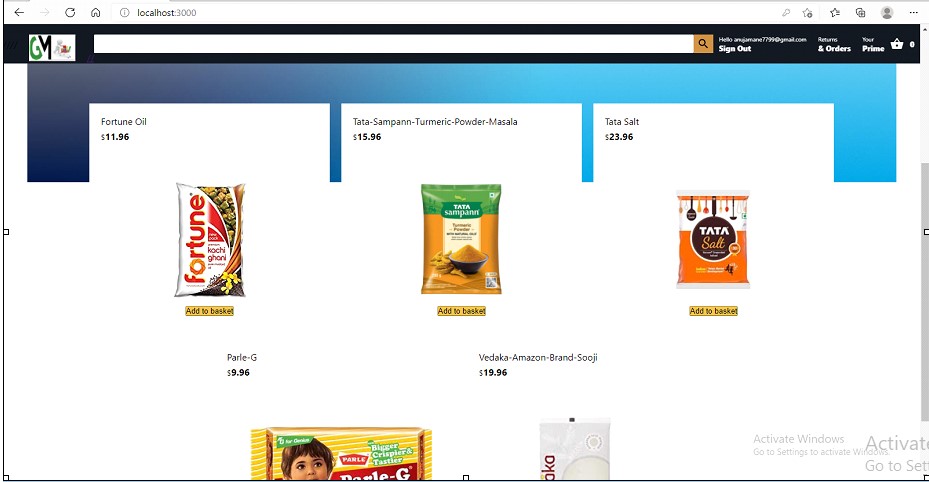
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | Login | Username password | Login successful | Login successful | Pass |
| 02 |  | Create new  account | username, Password, | Registration successful | Registration successful | Pass |
| 03 |  | Grocery item is added to  the basket | Add to cart | Successfully added to cart and to the  subtotal | Successfully added to cart and to the  subtotal | Pass |
| 04 |  | Grocery item is removed from the  basket | Remove  from the cart | Successfully removed from the cart and subtracted  from the  subtotal | Successfully removed from the cart and subtracted  from the  subtotal | pass |
| 05 |  | Billing of the grocery items in the cart | Price of all the grocery items added  to the cart | Final bill of all the  grocery items in the cart. | Final bill of all the  grocery items in the cart. | Pass |
| 06 |  | Weight of the grocery item | Weight | Weight is  successfully stored to the database | Weight is  successfully stored to the database | Pass |
| 07 |  | Weight of the grocery item below threshold value | Weight | User gets the message about less  grocery item | User gets the message about less  grocery items. | pass |

#### 6.3 System test cases generation and its testing reports

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test  No | case | Test case | Input | Expected  Output | Actual  Output | Status |
| 01 |  | Able to access the  application | Application name/link on network | Get application successfully. | Get application successfully. | Pass |
| 02 |  | Creating  account or  sign up | username and password | Get into  grocery management website  Successfully. | Get into  grocery management website  Successfully. | pass |
| 03 |  | Shopping for grocery items | Add or remove the  grocery items | Get the grocery time and billing of the groceries. | Get the grocery time and billing of the groceries. | pass |

## 7.OUTPUT SCREEN

Given below is the screenshot that shows the actual system working step by step.



## 8. PERFORMANCE ANALYSIS

In this project, we have done the analysis of the solutions available for the implementation of smart grocery systems using IoTs.

Smart grocery systems using IoT are cost effective and user-friendly systems for customers. It not only helps to induce organized grocery shopping; it additionally saves our time and money.

This platform can run without a central authority imposing its own rules during human interaction.

## 9. APPLICATIONS

1. This online portal helps to identify inventory status of all the grocery items in the house at a glance.
2. Order the required groceries on a click of button from the comfort of your home.
3. Users can save time and money by using this platform.

## 10. CONCLUSION

With a smart grocery system, one can relax and stop worrying about continuously following, and checking the grocery containers in our house. We will get notification about the low level of grocery and we can place orders for particular items.

This helps in ease of society and one step forward of making our city smart.

Future work or related work for smart grocery systems is that we can directly place orders by android application in our smartphones, so developing an application to automatically place order to grocery shops is further implementation in this system.

## 11. GUIDE TO ACCESS APPLICATION AND USER MANUAL

### ● Guide to Access Application

1. Connect to the network
2. Get the application address or link
3. Get the application

### ● User Manual

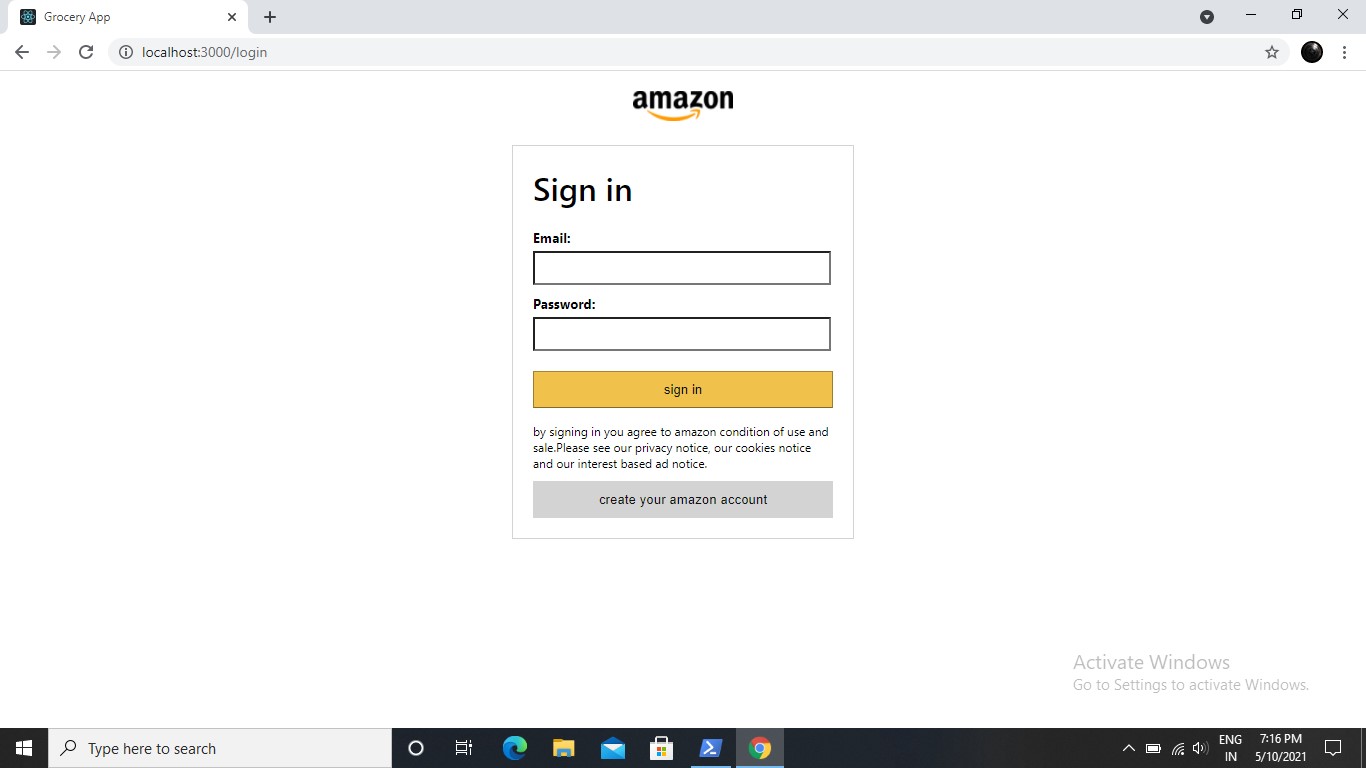
#### 11.1 The Screenshots of WebApp for this project

##### 11.1.1. Header of the page



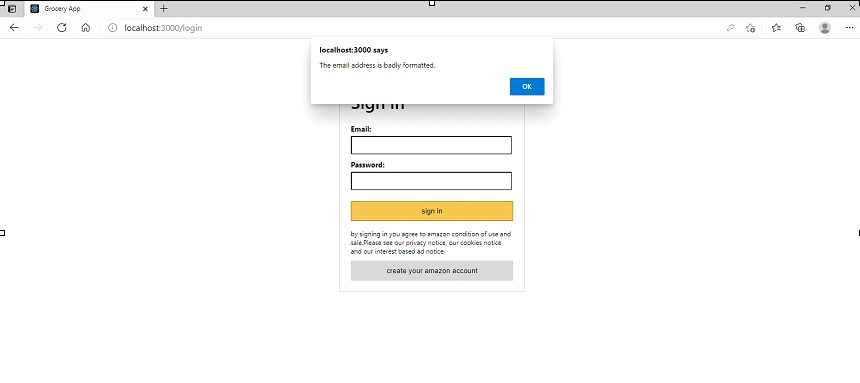
if you want to search products, then this facility for the account is also provided.

##### 11.1.2 Login page



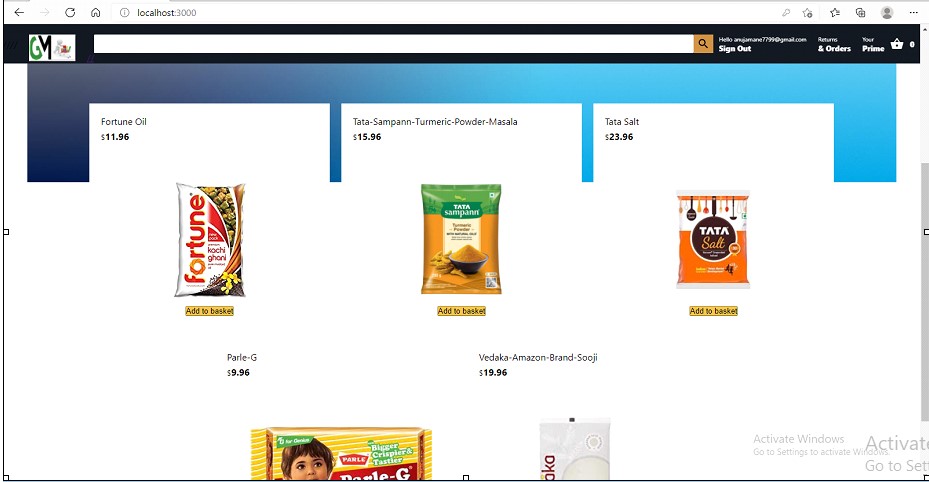
Users should enter valid username and password for entering into the app.

##### 11.1.3 Sign up page



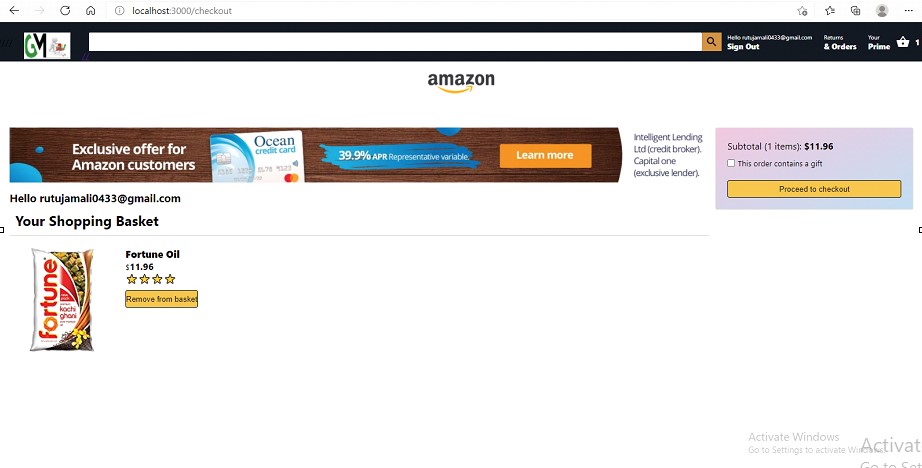
Users can create new accounts if required.

**11.1.4 Home page**



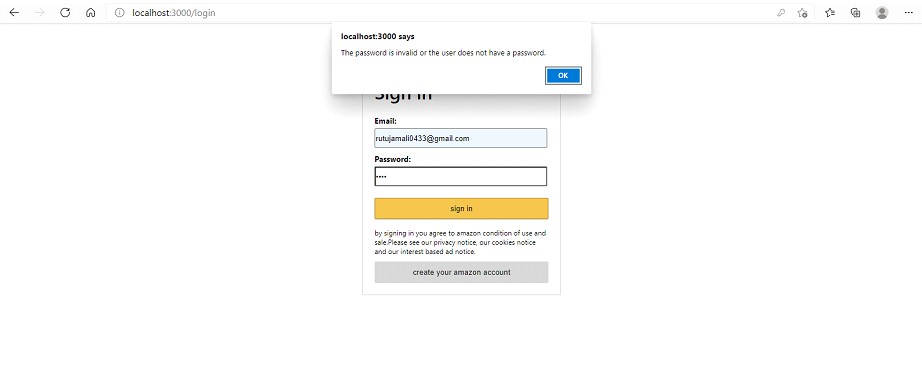
Welcome page or home page for users is available

##### 11.1.5 Cart page



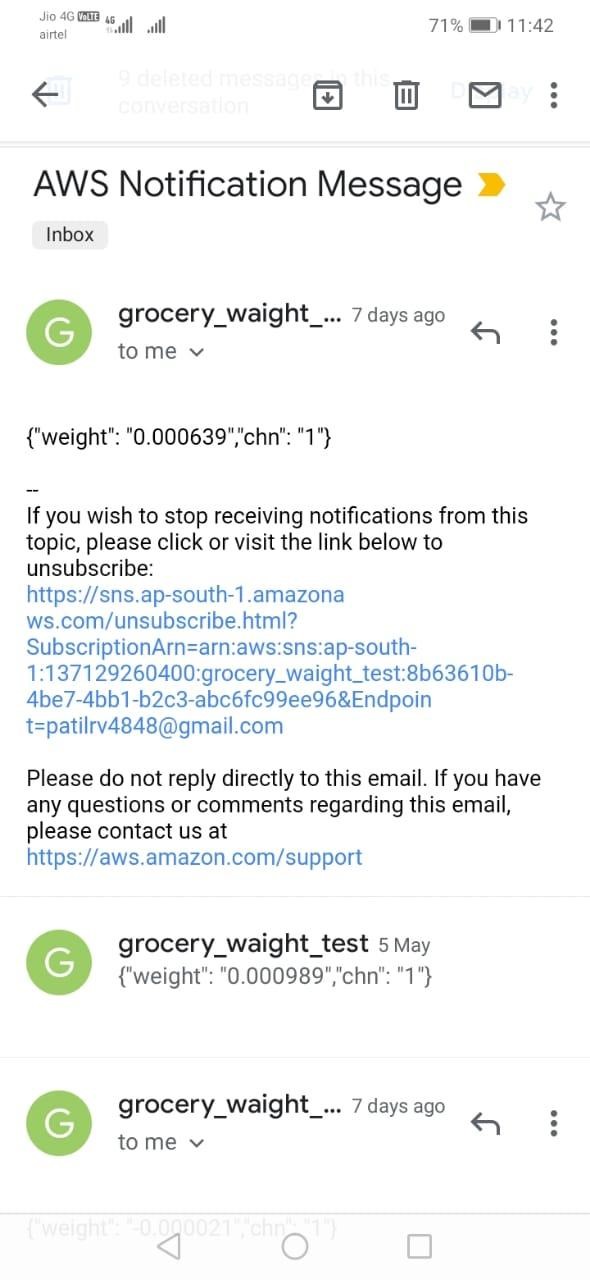
In application, checking for payment value can be done here.

##### 11.1.6 Validation of login credentials



Validation of username and password before proceeding to payment facility is available

##### 11.1.7 Notification received by the user



## 12. ETHICS

1. Users should not buy software with a single user license and then install it on multiple computers
2. Users should use weight sensors only to measure groceries not irrelevant products.
3. Users should not make a copy of the Grocman product for commercial use.

## 13. REFERENCES

* <https://ieeexplore.ieee.org/document/8065839/references#references>

(Publisher: IEEE - [2017](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [Fourteenth](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [International](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [Conference](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [on](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [Wireless](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [and](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding) [Optical Communications Networks (WOCN)](https://ieeexplore.ieee.org/xpl/conhome/8061194/proceeding)

* [https://www.semanticscholar.org/paper/IoT-based-grocery-monitoring-system-Desai-Guru vayurappan/17a5a8f1907a20860c57b920c14454d4394bf660](https://www.semanticscholar.org/paper/IoT-based-grocery-monitoring-system-Desai-Guruvayurappan/17a5a8f1907a20860c57b920c14454d4394bf660)

(IJRET: International Journal of Research in Engineering and Technology)

* [https://journal.uob.edu.bh/bitstream/handle/123456789/3739/paper-2.pdf?sequence=1&is](https://journal.uob.edu.bh/bitstream/handle/123456789/3739/paper-2.pdf?sequence=1&isAllowed=y)

[Allowed=y](https://journal.uob.edu.bh/bitstream/handle/123456789/3739/paper-2.pdf?sequence=1&isAllowed=y)

(International Journal of Computing and Network Technology)

* <https://www.ijsdr.org/papers/IJSDR1910014.pdf>

(The International Journal of Scientific Development and Research)

* <https://www.irjet.net/archives/V7/i7/IRJET-V7I7946.pdf>

(International Research Journal of Engineering and Technology (IRJET))

* AWS Fundamentals Course (Coursera online platform)

(Includes 1.AWS technical essentials 2. Addressing security risk 3. Migrating to the cloud 4. Building serverless applications

## 14. PLAGIARISM REPORT

