# ACKNOWLEDGEMENTS

Completing this project would not have been possible without the support, guidance, and assistance of various individuals. We extend our heartfelt gratitude to all those who played a significant role in making this project a reality.

First and foremost, we would like to express our most profound appreciation to our dedicated and knowledgeable faculty, Tanzilah Noor Shabnam (TnS1), from the Department of Electrical and Computer Engineering at North South University. Her continuous support, invaluable insights to our project significantly contributed to its successful completion. We are grateful for her mentorship and guidance throughout this journey.

We are indebted to the Department of Electrical and Computer Engineering at North South University for providing us with the necessary resources, infrastructure, and a conducive learning environment. Their support and encouragement greatly facilitated our project work.

In addition to our academic mentors, we would like to express our gratitude to our friends for their unwavering support and understanding during this project. Their encouragement and sacrifices have been a constant source of motivation.

We also want to acknowledge the broader academic community for the wealth of knowledge and research that served as the foundation for our project.

We recognize that this project results from a collaborative effort, and we are grateful to every individual and organization that contributed to its successful completion.

Nafiz Iqbal

2011765042,

Tofayel Ahmed Abhi

2112096642

# ABSTRACT

Project Title/Name: E-Agriculture Management System

Our project, titled "E-Agriculture Management System," aims to address the needs of both authentic farmers and the general population in our country. Currently, original farmers need help to receive fair compensation for their agricultural products due to the presence of intermediaries who purchase these products at low prices and subsequently sell them to consumers at inflated rates. This disconnect between farmers and consumers results in financial hardships for both parties. Our project aims to bridge this gap and eliminate the middleman.

Current Status: Presently, we have developed a platform that enables farmers to sell their products to consumers directly. This platform also features a one-to-one communication method to facilitate direct interaction between farmers and consumers. However, the current version of our platform has some limitations and challenges that we are actively working to overcome.

Methods Followed: To overcome these challenges, we have adopted a monolithic architecture for our web application, consolidating our codebase into a single unified framework. This approach has allowed us to address issues, particularly in developing a dedicated chatting methodology, which represents the innovation of our project. The monolithic architecture consists of three essential layers: frontend, logic, and database. We have harnessed Firebase as our database solution, which has proven instrumental in overcoming these challenges.

Impact/Significance of the Result: The findings of our project hold significant implications for the agricultural sector, the economy, and e-commerce in our country. Despite the prevalence of various e-commerce platforms, there needs to be a dedicated platform for authentic farmers. Our project stands to fulfill this critical need soon.

By providing a direct channel for farmers to connect with consumers, our platform will not only empower farmers to receive fair compensation for their products but also allow consumers to access high-quality, locally sourced produce. This innovative approach has the potential to revolutionize the agricultural landscape and promote economic growth while fostering a more sustainable and equitable marketplace.

Contents

[ACKNOWLEDGEMENTS 1](#_Toc150354432)

[ABSTRACT 2](#_Toc150354433)

[Chapter 1: Introduction 4](#_Toc150354434)

[1.1 Background and Motivation 4](#_Toc150354435)

[1.2 Purpose and Goal of the Project 4](#_Toc150354436)

[1.3 Organization of the Report 4](#_Toc150354437)

[Chapter 2: Research Literature Review 5](#_Toc150354438)

[2.1 Introduction 5](#_Toc150354439)

[2.2 Limitations of the Current Agricultural Landscape 5](#_Toc150354440)

[2.3 Missed Technological Advancements 5](#_Toc150354441)

[2.4 Economic and Social Consequences 5](#_Toc150354442)

[2.5 Conclusion 6](#_Toc150354443)

[Chapter 3 Methodology 6](#_Toc150354444)

[3.1 System Design [Described the project’s design flowchart, class diagram,database schema] 6](#_Toc150354445)

[Chapter 4: Investigation/Experiment, Result, Analysis, and Discussion 10](#_Toc150354446)

[4.1 Experiment and Investigation 10](#_Toc150354447)

[4.2 Results 10](#_Toc150354448)

[4.3 Analysis 10](#_Toc150354449)

[4.4 Discussion 10](#_Toc150354450)

[Chapter 5: Impacts of the Project 10](#_Toc150354451)

[5.1 Impact of this project on societal, health, safety, legal, and cultural issues 10](#_Toc150354452)

[5.2 Impact of this project on environment and sustainability 11](#_Toc150354453)

[Chapter 6: Project Management 12](#_Toc150354454)

[6.1 Gantt Chart 12](#_Toc150354455)

[6.2 Weekly Contribution: 12](#_Toc150354456)

[6.3 Pseudocode and algorithm 13](#_Toc150354457)

[6.5 Code Snippets: 20](#_Toc150354458)

[6.5 UI SNIPPET 30](#_Toc150354459)

[Chapter 7: Conclusion 37](#_Toc150354460)

[7.1 Summary 37](#_Toc150354461)

[7.2 Limitations 37](#_Toc150354462)

[7.3 Future Improvements 37](#_Toc150354463)

[References: 39](#_Toc150354464)

# Chapter 1: Introduction

## 1.1 Background and Motivation

In an age defined by rapid technological advancements and the ever-evolving demands of global agriculture, the concept of E-Agriculture emerges as a transformative solution poised to revolutionize the traditional farming and agribusiness landscape. At the intersection of agriculture and digital technology, E-Agriculture heralds a new era of efficiency, sustainability, and connectivity in the agricultural sector. This project proposal outlines a comprehensive E-Agriculture Management System aimed at seamlessly connecting farmers and customers, fostering transparency, optimizing resource utilization, and catalyzing rural development.

## 1.2 Purpose and Goal of the Project

The global agricultural industry faces unprecedented challenges. Increasing population, shifting dietary habits, and climate variability pressure farmers to enhance production while minimizing environmental impact. Traditional supply chains, often burdened with intermediaries, lead to inflated consumer prices and insufficient income for farmers. In this context, E-Agriculture presents a compelling solution that harnesses digital platforms, data analytics, and e-commerce to address these pressing challenges.

The proposed E-Agriculture Management System envisions a holistic ecosystem that engages stakeholders, from smallholder farmers to urban consumers. Through user-friendly web interfaces, farmers can digitize their operations, including product showcases, inventory management, and pricing strategies. Customers, on the other hand, gain direct access to a diverse array of fresh, locally sourced agricultural goods, enabling them to make informed purchasing decisions.

## 1.3 Organization of the Report

This project report is structured as follows:

Chapter 2 delves into a comprehensive literature review, exploring the existing e-commerce platforms and their limitations.

Chapter 3 provides insights into the methodology and system design employed in our project.

Chapter 4 presents the results of our project and offers a detailed analysis of our findings.

Chapter 5 discusses the societal and economic impact of the "E-Agriculture Management System."

Chapter 6 covers the project's management aspects, including planning and budget.

Chapter 7 concludes the report with a summary, limitations, and prospects for future improvements.

In summary, the proposed E-Agriculture Management System represents a transformative initiative that leverages technology to reshape the agricultural landscape. By bridging the gap between farmers and customers, promoting transparency, optimizing resource utilization, and fostering rural development, this project aims to create a paradigm shift that not only addresses existing challenges but also paves the way for a more sustainable and prosperous future for all stakeholders involved.

# Chapter 2: Research Literature Review

## 2.1 Introduction

In pursuing an E-Agriculture Management System, it is imperative to comprehend the compelling reasons underpinning the selection of this transformative project. The absence of such a system would perpetuate long-standing challenges within the agricultural sector, impeding progress and continued growth. This literature review delves into the key motivations behind our project.

## 2.2 Limitations of the Current Agricultural Landscape

1. Limited Market Access for Farmers:
2. With an E-Agriculture Management System, farmers, mainly smallholders, can access broader markets. Geographical constraints and a lack of information hinder their ability to reach consumers beyond their immediate vicinity, leading to reduced income opportunities, economic growth, and perpetuating cycles of rural underdevelopment.
3. Inefficient Supply Chains:
4. A digital platform is necessary to maintain efficient supply chains characterized by extended delivery times and increased post-harvest losses. Intermediaries play a central role, adding extra costs and price fluctuations that impact farmers and consumers. The absence of real-time inventory tracking and order management exacerbates these inefficiencies.
5. Lack of Transparency and Consumer Trust:
6. Consumers need access to comprehensive product information with an E-Agriculture Management System. This absence of transparency erodes consumer trust, as there is no way to verify the authenticity and quality of the products they purchase.

## 2.3 Missed Technological Advancements

1. Limited Access to Technological Advancements:
2. The absence of an E-Agriculture Management System means missed opportunities to leverage data analytics, digital payments, and other technological advancements. Farmers lack valuable insights into market trends and consumer preferences, restricting their ability to make informed decisions about crop selection and production planning.

## 2.4 Economic and Social Consequences

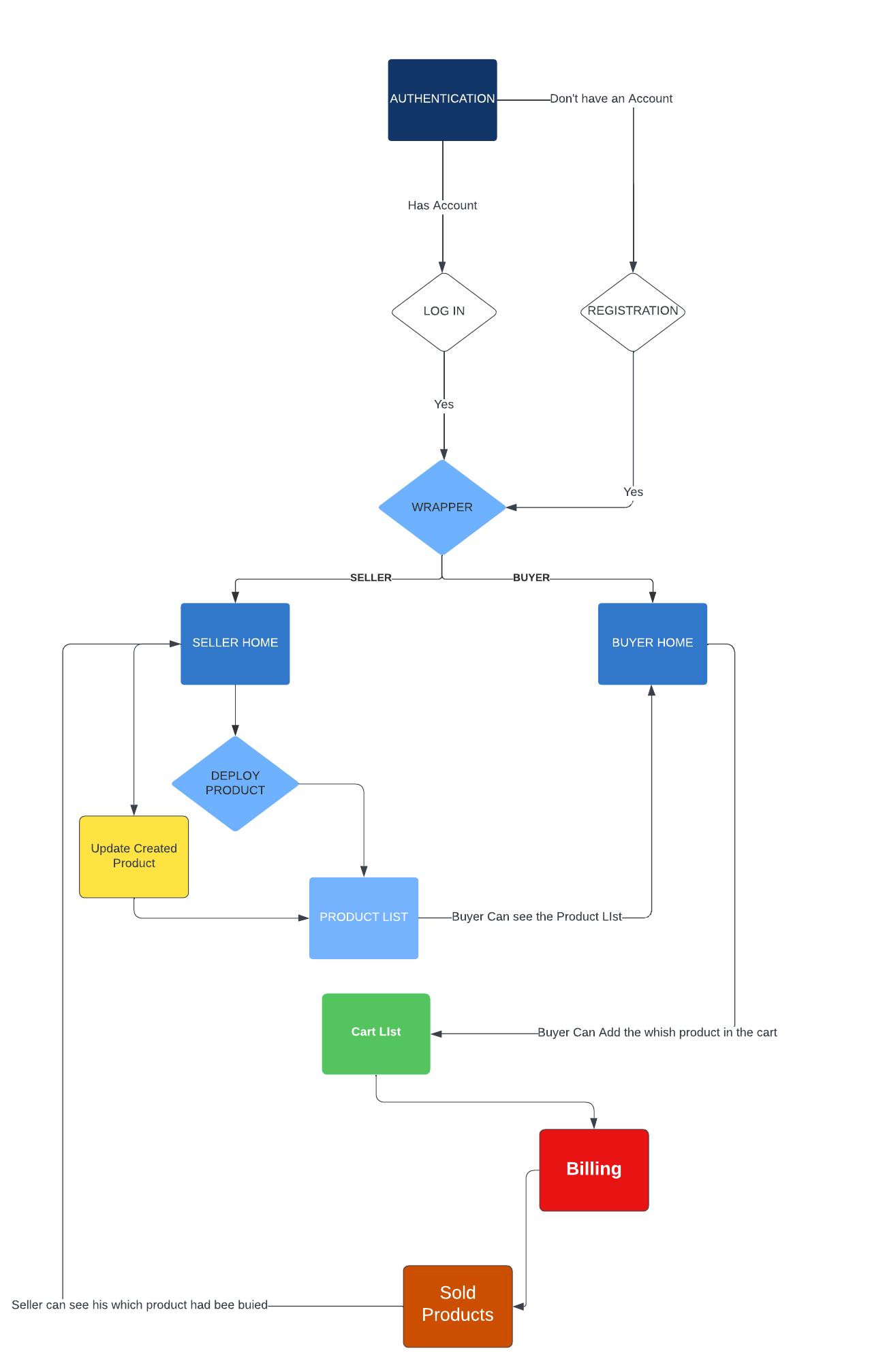
1. Limited Financial Inclusion:
2. The presence of digital payment mechanisms helps financial inclusion for both farmers and consumers. Cash-based transactions limit participation in the formal economy, posing challenges for economic growth.
3. Reduced Sustainability:
4. Resource optimization and sustainable agricultural practices facilitated by digital platforms still need to be explored. Precision farming techniques that optimize resource use and reduce environmental impact are underutilized, potentially contributing to unsustainable farming practices.
5. Missed Opportunities for Women's Empowerment:
6. The absence of an E-Agriculture Management System deprives women farmers of equal opportunities for market participation and financial transactions, perpetuating gender disparities and limiting women's empowerment.
7. Lack of Economic Growth in Rural Areas.
8. A digital platform for direct farmer-consumer transactions is necessary for economic growth and rural development. Limited market opportunities and reduced income potential for farmers contribute to stagnation in rural economies.

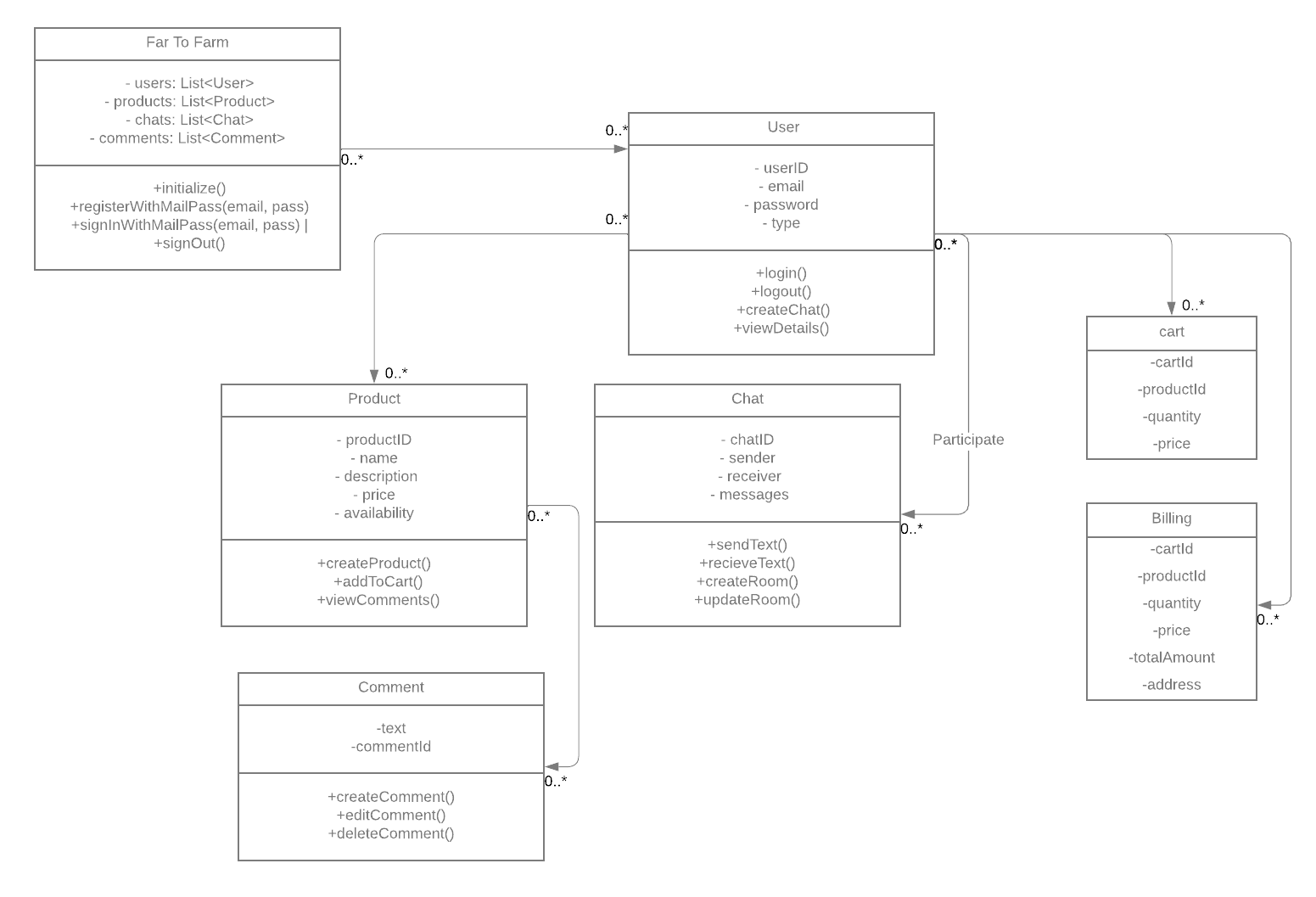
## 2.5 Conclusion

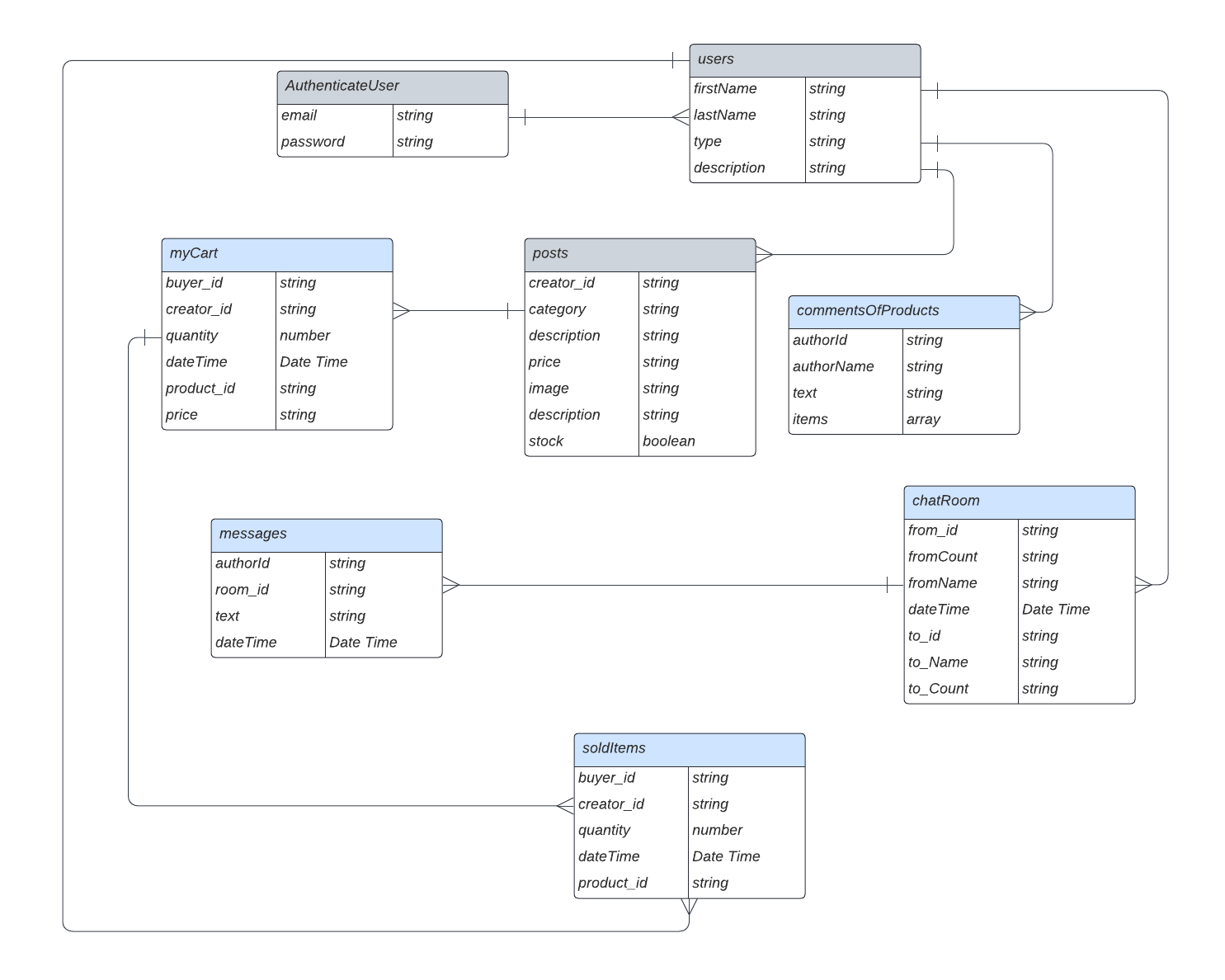
In conclusion, the absence of an E-Agriculture Management System perpetuates challenges within the agricultural sector, hindering market access, efficiency, transparency, sustainability, and financial inclusion. The transformative benefits offered by such a platform, including direct market access, transparency, and digital inclusion, remain unrealized, leaving farmers, consumers, and the agricultural sector at a disadvantage in an increasingly digital and interconnected world.

# Chapter 3 Methodology

## 3.1 System Design [Described the project’s design flowchart, class diagram,database schema]





****

# Chapter 4: Investigation/Experiment, Result, Analysis, and Discussion

## 4.1 Experiment and Investigation

The research conducted in this project involved a comprehensive analysis of the existing E-Agriculture Management Systems. Several limitations and challenges within these systems were identified. This section delves into the experiment, data collection, and investigative processes.

## 4.2 Results

The results obtained from the investigation are presented in this section, addressing key variables and aspects of the E-Agriculture Management Systems. Data is represented using appropriate figures and tables, each accompanied by concise captions, axis labels, and legends.

## 4.3 Analysis

In-depth analysis of the results is carried out to interpret the findings effectively. This section aims to provide insights into the significance of the data and how it relates to the identified challenges in E-Agriculture Management Systems.

## 4.4 Discussion

A constructive discussion follows the analysis, allowing us to draw meaningful conclusions from the results. The outcomes of the experiment are examined in detail, with a focus on their implications for the agricultural sector.

# Chapter 5: Impacts of the Project

## 5.1 Impact of this project on societal, health, safety, legal, and cultural issues

Our E-Agriculture Management System project holds significant implications for various aspects of society, health, safety, legal, and cultural domains. By seamlessly connecting farmers and consumers, fostering transparency, optimizing resource utilization, and catalyzing rural development, the project aims to bring about several positive impacts.

**Societal Impact:** The project contributes to the social welfare of farming communities by enhancing their economic prospects. With improved market access, farmers can increase their income, thereby reducing poverty and underdevelopment in rural areas. Furthermore, the project promotes gender equity by providing equal opportunities for women in the agricultural sector, contributing to social empowerment.

**Health Impact:** While not directly related to physical health, the project indirectly supports the well-being of farming communities by alleviating financial stress and offering access to a broader consumer base. This could lead to reduced pressure and an improved quality of life for farmers, which can positively affect their mental and emotional health.

**Safety Impact:** By enabling digital payments and reducing the reliance on cash transactions, the project enhances the safety of financial transactions for farmers and consumers. This can contribute to a safer economic environment, reducing the risk of theft and fraud associated with cash-based dealings.

**Legal Impact:** The project may have legal implications for data ownership, privacy, and e-commerce regulations. Ensuring compliance with these regulations is crucial to avoid potential legal issues and disputes, which could impact the project's sustainability.

**Cultural Impact:** Introducing digital platforms into traditional farming practices may change cultural norms and practices. It is crucial to monitor and manage these changes sensitively to ensure they align with the cultural values and traditions of the local community.

## 5.2 Impact of this project on environment and sustainability

The project also has noteworthy environmental and sustainability implications. By optimizing resource utilization and promoting sustainable farming practices, it contributes to a more environmentally friendly and resilient agricultural sector.

**Environmental Impact:** The project's emphasis on sustainable agricultural practices, including precision farming techniques, can lead to a reduced ecological footprint. By optimizing resource use, farmers can minimize the environmental impact of their activities, leading to more sustainable farming practices.

**Sustainability Impact:** Sustainability is at the core of this project. By fostering resource optimization, reducing waste, and enhancing economic opportunities for farmers, it contributes to the long-term sustainability of the agricultural sector. The project's focus on connecting farmers directly with consumers also reduces the carbon footprint associated with lengthy supply chains, contributing to sustainability.

In conclusion, our E-Agriculture Management System project goes beyond digital transformation. It brings about positive impacts on society, health, safety, legal compliance, and culture while also promoting environmental sustainability. By addressing these dimensions, our project aims to create a balanced and sustainable future for all stakeholders.

# Chapter 6: Project Management

## 6.1 Gantt Chart



## 6.2 Weekly Contribution:

Participator Name: Nafiz Iqbal

**Week 1:**

* Set up the React Environment.
* Configured the Firebase Console.
* Established a connection to the Firebase Database using API keys.
* Successfully implemented the authentication process.
* Created authentication components for the project.

**Week 3:**

* Developed routing for buyer and seller accounts upon user login.
* Implemented routing based on the user type.
* Created a wrapper component for user-specific routing.

**Week 4:**

* Conducted CRUD (Create, Read, Update, Delete) operations for the SELLER route.

**Week 5:**

* Displayed the product cart in seller accounts.
* Enabled real-time rendering and product deletion.
* Showcased the product cart in buyer accounts.
* Implemented comprehensive cart management for buyers, including insertions, updates, and item deletions.

**Week 6:**

* Enhanced product details display, including images and comprehensive product information.
* Initiated work on a nested comment section for individual product details pages.

**Week 7:**

* Completed the functionality for updating the quantity of each product for individual users.
* Calculated and displayed the total price of each product in real-time.
* Fully implemented a comment system with one-to-one individual communication, ensuring database connectivity.

**Week 8:**

* Completed the UI design for all components of the web app.
* Updated and enhanced the chat functionality throughout the entire application.

**Week 9:**

* Focused on resolving any existing component-related bugs.
* Conducted a project demo, showcasing the work completed.

**Week 10:**

* Created Report for the final submission.
* Created Presentation for the final presentation.

## 6.3 Pseudocode and algorithm

**User Registration Functions**

**# User Registration**

FUNCTION registerWithMailPass(email, pass)

IF email is empty

SET error to 'Please enter a valid email'

RETURN

END IF

IF pass is empty

SET error to 'Insert a valid password'

RETURN

END IF

TRY

CALL createUserWithEmailAndPassword(auth, email, pass)

CATCH error

LOG error

END TRY

END FUNCTION

**# User Login**

FUNCTION signInWithMailPass(email, pass)

IF email is empty

SET error to 'Please enter a valid email'

RETURN

END IF

IF pass is empty

SET error to 'Insert a valid password'

RETURN

END IF

TRY

CALL signInWithEmailAndPassword(auth, email, pass)

LOG 'Login Method Succeeded'

CATCH error

IF error code is 'auth/user-not-found'

LOG 'User not found. Please register first.'

ELSE

LOG 'Login Method Failed'

LOG error

END IF

END TRY

END FUNCTION

**# User Logout**

FUNCTION signOut()

TRY

CALL auth.signOut()

SET userid to 'null'

LOG 'Log Out Complete'

CATCH error

LOG error

END TRY

END FUNCTION

**Real-time Chat Functionality**

**# Sending a Chat Message**

const sendChatMessage = (message) => {

// Send a chat message to the recipient

// Store the message in the database

// Update the chat interface with the new message

};

**# Receive a Chat Message (Example of Real-time Chat Interface Update)**

const receiveChatMessage = (message) => {

// Display the received chat message in the chat interface

};

**User Type Functions**

**# Find User Type**

FUNCTION findType(userid)

Initialize an empty result variable (res)

Get a Firestore database reference (db)

TRY

Get a snapshot of the "users" collection from Firestore

FOR EACH document in the snapshot

IF the document's ID matches the provided userid

Extract the "type" data from the document

Set the result (res) to the extracted "type"

BREAK the loop

END IF

END FOR

CATCH any errors

Log an error message

END TRY

RETURN the result (res)

END FUNCTION

**# Send Text**

FUNCTION sendText(data)

Get a Firestore database reference (db)

Specify the collection where you want to add messages (collectionRef)

TRY

Add the provided data to the collection using the addDoc function

IF successful

Log a success message with the ID of the added document

ELSE

Log an error message

END IF

CATCH any errors

Log an error message

END TRY

END FUNCTION

**# Update Chat Room**

FUNCTION updateRoom(docId, data)

Get a Firestore database reference (db)

Specify the path to the document you want to update (docRef)

TRY

Use updateDoc to update specific fields in the document

IF successful

Log a success message

ELSE

Log an error message

END IF

CATCH any errors

Log an error message

END TRY

END FUNCTION

**# Get Room Data by ID**

FUNCTION getRoomDataById(docId)

Get a Firestore database reference (db)

Specify the path to the document you want to retrieve (docRef)

TRY

Get a snapshot of the document using getDoc

IF the document exists

Extract and return the document data

ELSE

Log a message that the document does not exist

Return null

END IF

CATCH any errors

Log an error message

Return null

END TRY

END FUNCTION

**# Create Room for Chat**

FUNCTION createRoomForChat(roomId, data)

Get a Firestore database reference (db)

Specify the document reference for the chat room (room)

TRY

Create the chat room with the provided data using setDoc

IF successful

Log a message that the room is created

ELSE

Log an error message

END IF

CATCH any errors

Log an error message

END TRY

END FUNCTION

**Comment Section Functions**

**# Insert Node**

FUNCTION insertNode(tree, commentId, item)

IF tree's ID matches the provided commentId

Create a new node with a unique ID, the provided item, and an empty items array

Add the new node to the tree's items array

RETURN the modified tree

ELSE

Initialize an empty array (latestNode)

FOR EACH item in tree's items

Recursively call insertNode on each item

Add the result to latestNode

END FOR

Clone the tree and update its items with latestNode

RETURN the modified tree

END IF

END FUNCTION

**# Edit Node**

FUNCTION editNode(tree, commentId, value)

IF tree's ID matches the provided commentId

Update the tree's name with the provided value

RETURN the modified tree

ELSE

FOR EACH item in tree's items

Recursively call editNode on each item

END FOR

Clone the tree and its items

RETURN the modified tree

END IF

END FUNCTION

**# Delete Node**

FUNCTION deleteNode(tree, id)

FOR i from 0 TO the length of tree's items

Get the current item at index i

IF the current item's ID matches the provided id

Remove the current item from tree's items

RETURN the modified tree

ELSE

Recursively call deleteNode on the current item

END IF

END FOR

RETURN the modified tree

END FUNCTION

## 6.5 Code Snippets:

**USER LOGIN**

Description: This component handles user login functionality. It provides the user with a login form, allows them to enter their email and password, and sign in. After successful login, it navigates the user to the "Wrap" page. It also includes a password visibility toggle and a password reset option.

import { Navigate, Link, useNavigate } from "react-router-dom";

import { useAuth } from "../FireBase/Authentication/AuthContext";

import { auth } from "../FireBase/FireComp";

import { useEffect, useState } from "react";

import LogInPic from "../assets/loginPic.svg";

import { ScaleLoader } from "react-spinners";

const LogIn = () => {

// Component logic here

}

**USER REGISTRATION**

Description: This component manages user registration. Users can sign up by providing their information, including first name, last name, description, and choosing between "buyer" and "seller" type. After successful registration, it creates a user profile and navigates to the "Wrap" page. It also includes error handling and password visibility toggles.

import { Navigate, useNavigate } from "react-router-dom";

import { useAuth } from "../FireBase/Authentication/AuthContext";

import { auth } from "../FireBase/FireComp";

import { useEffect, useState } from "react";

import { ScaleLoader } from "react-spinners";

import { getFirestore, setDoc, doc } from "firebase/firestore";

export default function Registration() {

// Component logic here

}

**HOME**

Description: The Home component checks if the user is already logged in. If so, it redirects to the "Wrap" page; otherwise, it remains on the Home page. It uses Firebase's authentication state observer to determine the user's login status.

import { Link, Navigate } from "react-router-dom";

import { useAuth } from "../FireBase/Authentication/AuthContext";

import { auth } from "../FireBase/FireComp";

import { useEffect, useState } from "react";

import Wrapper from "./Wrapper";

import LogIn from "./Login";

import { onAuthStateChanged } from "firebase/auth";

const Home = () => {

// Component logic here

}

**WRAPPER**

Description: The Wrapper component is responsible for managing the user's role (buyer or seller) after login. It fetches user data from the Firebase Firestore database and redirects the user based on their role to either the "Seller" or "Buyer" page. It uses periodic data updates to check for role changes.

import { useEffect, useState } from "react";

import { useAuth } from "../FireBase/Authentication/AuthContext";

import { getFirestore, collection, getDocs } from "firebase/firestore";

import { Navigate, useNavigate } from "react-router-dom";

import { auth } from "../FireBase/FireComp";

import { BounceLoader } from "react-spinners";

const Wrapper = () => {

// Component logic here

}

**BUYER**

Description: The Buyer component handles the logic for logged-in users with the role of "buyer." It provides options to view products and access the shopping cart. Users can log out from this component, and it navigates them back to the login page upon logging out.

import { useEffect, useState } from "react";

import { useAuth } from "../FireBase/Authentication/AuthContext";

import { auth } from "../FireBase/FireComp";

import { Navigate, useNavigate } from "react-router-dom";

import Products from "./Product DIrectory/Products";

import BuyerBar from "./BuyerBar";

import carousoul from "./Carosoul/carosoul";

const Buyer = () => {

// Component logic here

}

**BUYER BAR**

Description: The BuyerBar component is a navigation bar for buyers. It provides options to navigate within the buyer's section, including viewing products and accessing the shopping cart.

import logout from "../assets/logout.svg";

import cartPic from "../assets/cart.svg";

import { useNavigate } from "react-router-dom";

import { auth } from "../FireBase/FireComp";

const BuyerBar = ({ logOut, cart }) => {

// Component logic here

}

**CART**

Description: The Cart component manages the user's shopping cart. It allows users to view and edit their cart items, calculate the total price of products, and remove items from the cart. It uses Firebase Realtime Database to store and update cart data. The component includes logic for calculating the total price of cart items.

import React, { useEffect, useState } from "react";

import { useLocation, useNavigate } from "react-router-dom";

import { getDatabase, ref, push, get, remove } from "firebase/database";

import CartManager from "../CartManager";

import Cart from "./Cart";

import backIcon from "../../assets/back.png";

const Carts = () => {

// Component logic here

}

**CART MANAGER**

Description: The CartManager class is responsible for managing cart operations, including adding, updating, and deleting items from the cart. It interacts with Firebase Realtime Database to perform these operations. It includes methods to add items to the cart, update the quantity of items, and delete items from the cart.

import { getDatabase, ref, push, get, remove, update } from "firebase/database";

class CartManager {

// Component logic here

}

**CART**

Description: The Cart component is responsible for managing individual items within the shopping cart. Users can view details of a product saved in their cart, update the quantity, and remove items. It interacts with the CartManager to handle cart operations such as updating quantity.

import React, { useEffect, useState } from "react";

import shoppingCart from "../../assets/shopping-cart.png";

import CartManager from "../CartManager";

import { useNavigate } from "react-router-dom";

const Cart = ({

buyer\_id,

creator\_id,

inStock,

product\_desc,

product\_id,

product\_image,

product\_name,

product\_price,

quantity,

path,

removeItem,

update,

}) {

// Component logic here

}

**CART PRODUCT DETAILS**

Description: The CartProductDetails component displays details of a product saved in the cart. Users can navigate back to their shopping cart.

import React from "react";

import { Navigate, useLocation, useNavigate } from "react-router-dom";

import backIcon from "../../assets/back.png";

import MainComment from "../../Comment/MainComment";

const CartProductDetails = () => {

// Component logic here

}

**PRODUCTS**

Description: The Products component displays a list of available products, and users can filter products by name and category. It fetches product data from Firebase Firestore and provides a list of products for users to view.

import { collection, getDocs, getFirestore } from "firebase/firestore";

import React, { useEffect, useState } from "react";

import Product from "./Product";

import Carousoul from "../Carosoul/carosoul";

const Products = ({ buyerId }) {

// Component logic here

}

**PRODUCT**

Description: The Product component represents an individual product in the product list. Users can view product details, add or remove products from their shopping cart, and check product availability. The component interacts with the CartManager to manage cart operations.

import { useEffect, useState } from "react";

import { useNavigate } from "react-router-dom";

import shoppingCart from "../../assets/shopping-cart.png";

import { getDatabase, ref, push, get } from "firebase/database";

import CartManager from "../CartManager";

import { useAuth } from "../../FireBase/Authentication/AuthContext";

const Product = ({

id,

name,

price,

desc,

creator\_id,

avail,

image,

buyerId,

}) {

// Component logic here

}

**DETAILS**

Description: The Details component displays detailed information about a product. Users can add the product to their shopping cart and initiate a chat with the product creator. The component interacts with the CartManager to add products to the cart and the chatClass to create chat rooms.

import React, { useState } from "react";

import { useLocation, useNavigate } from "react-router-dom";

import CartManager from "../CartManager";

import backIcon from "../../assets/back.png";

import MainComment from "../../Comment/MainComment";

import chatClass from "../../Chat Application/chatClass";

const Details = () => {

// Component logic here

}

**CHECKOUT**

Description: The Checkout component allows users to purchase products that they have added to their cart. It gathers user information such as name, email, address, and postal code for the delivery. It utilizes the CartManager to remove items from the cart and interacts with the checkoutClass to send an email confirmation and save the sold items' details in the "soldItems" collection.

import { useEffect, useState } from "react";

import { useLocation, useNavigate } from "react-router-dom";

import checkoutClass from "./checkoutClass";

import { auth } from "../../FireBase/FireComp";

import CartManager from "../CartManager";

import { serverTimestamp } from "firebase/database";

const CheckOut = () => {

// Component logic here

}

**checkoutClass**

Description: The checkoutClass class contains methods for sending email confirmations and saving sold items' details in the "soldItems" collection. The sendEmail method sends an email confirmation to the user, and the setCartPend method saves sold item data in the Firestore database.

import emailjs from "@emailjs/browser";

import {

equalTo,

get,

getDatabase,

orderByChild,

ref,

} from "firebase/database";

import {

addDoc,

collection,

doc,

getFirestore,

setDoc,

} from "firebase/firestore";

import { auth } from "../../FireBase/FireComp";

class checkoutClass {

// Methods for sending email confirmation and saving sold items

// Logic here

}

**Messenger Component**

Description: The Messenger component is responsible for managing a chat conversation. It allows users to send and receive messages, updates message counts, and scrolls to the latest messages. It retrieves messages and updates the chat room based on user type (seller or buyer).

import React, { useEffect, useRef, useState } from "react";

import { useLocation, useNavigate } from "react-router-dom";

import chatClass from "./chatClass";

import { useAuth } from "../FireBase/Authentication/AuthContext";

import { Timestamp, serverTimestamp } from "firebase/firestore";

const Messenger = () => {

const { userid, getCurrentUser } = useAuth();

const [text, setText] = useState("");

const [count, setCount] = useState(0);

const [message, setMessage] = useState([]);

const [refresh, setRefresh] = useState(false);

const chatContainerRef = useRef(null);

const location = useLocation();

const { roomId, name } = location.state;

const chat = new chatClass();

const navigate = useNavigate();

// Component logic for managing chat conversations

// ...

return (

// Render the messenger component

);

}

export default Messenger;

**ChatCart Component**

Description: The ChatCart component is responsible for displaying a chat card with information such as the sender's name and the last message's timestamp. It also includes a link to navigate to the messenger component.

import { useNavigate } from "react-router-dom";

const ChatCart = ({ time, name, roomId, message }) => {

// Component logic for displaying chat information

const timestamp = time;

const date = new Date(timestamp.seconds \* 1000);

const formattedDate = date.toLocaleString();

const navigate = useNavigate();

const goToMessenger = () => {

navigate("/chat", {

state: {

roomId,

name,

},

});

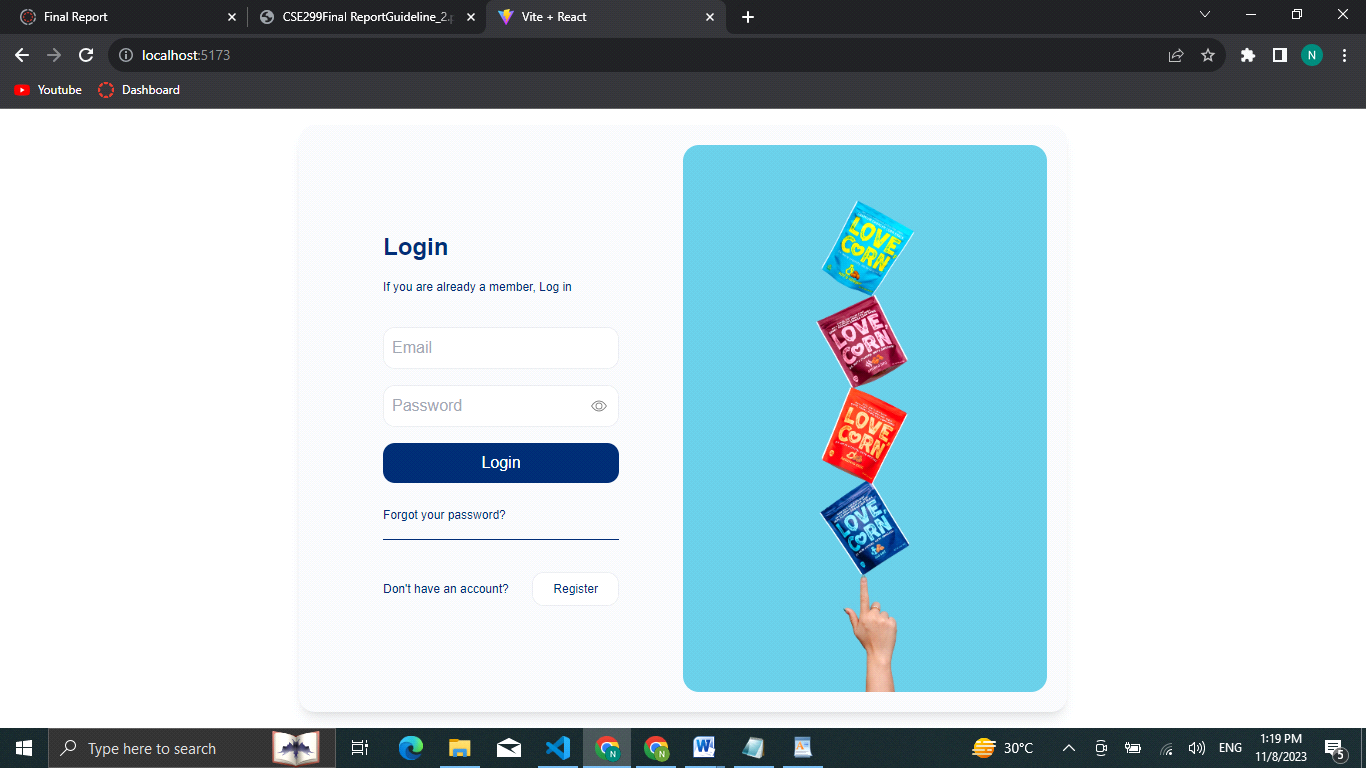
};

}

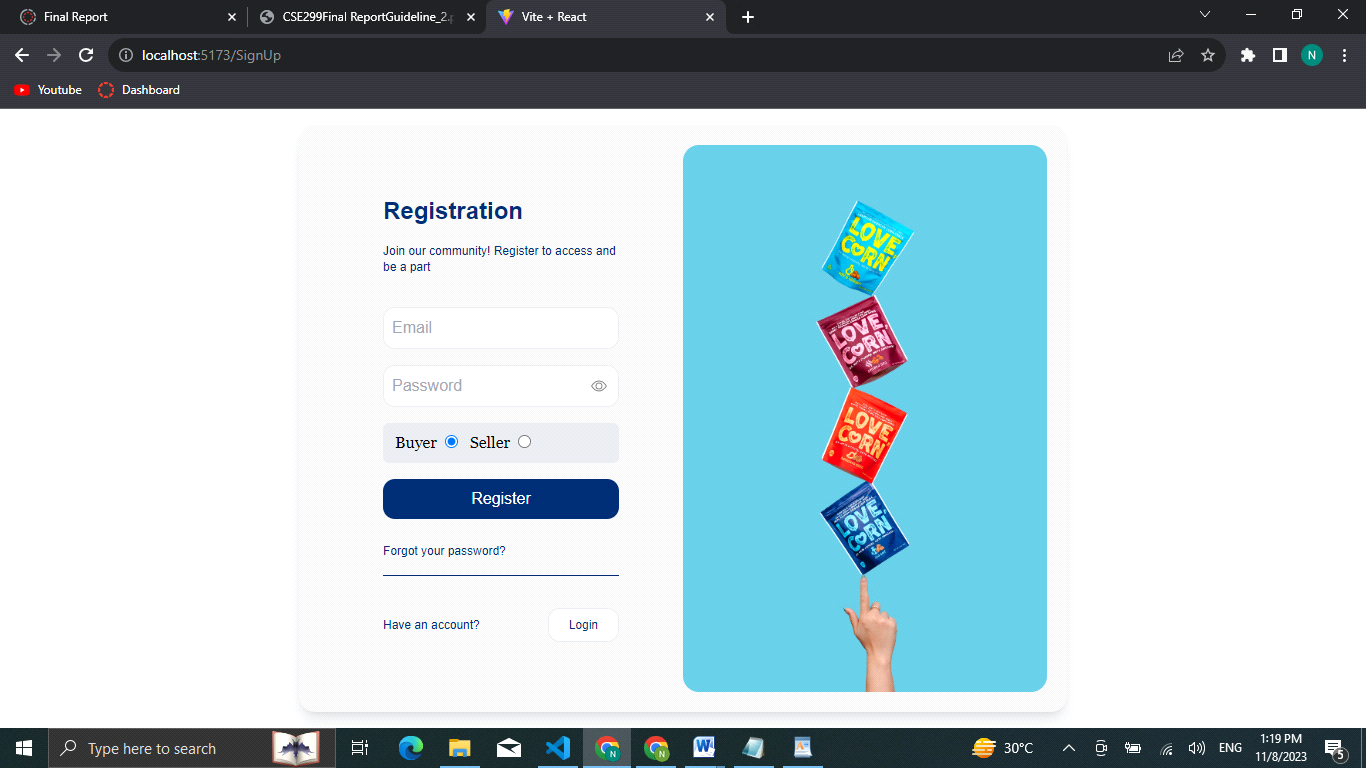
export default ChatCart;

## 6.5 UI SNIPPET

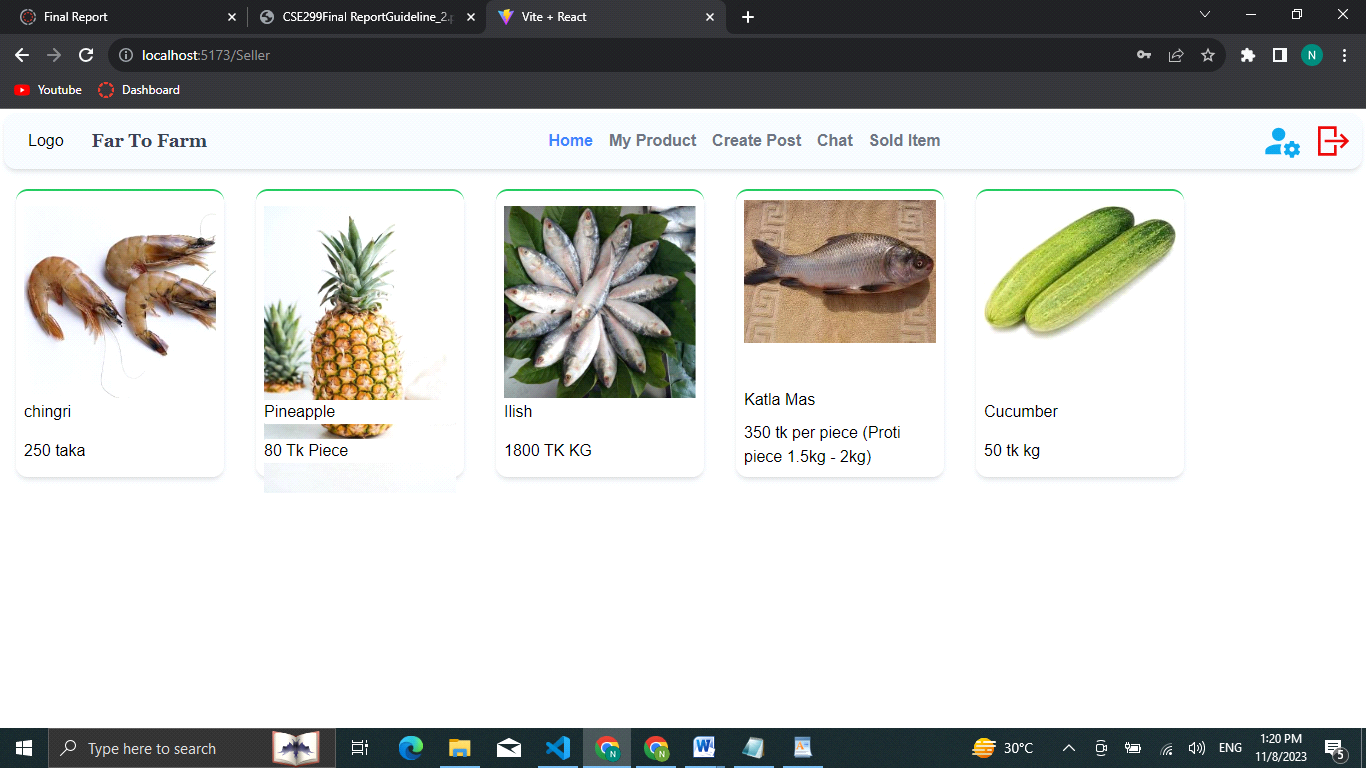
**LogIn**

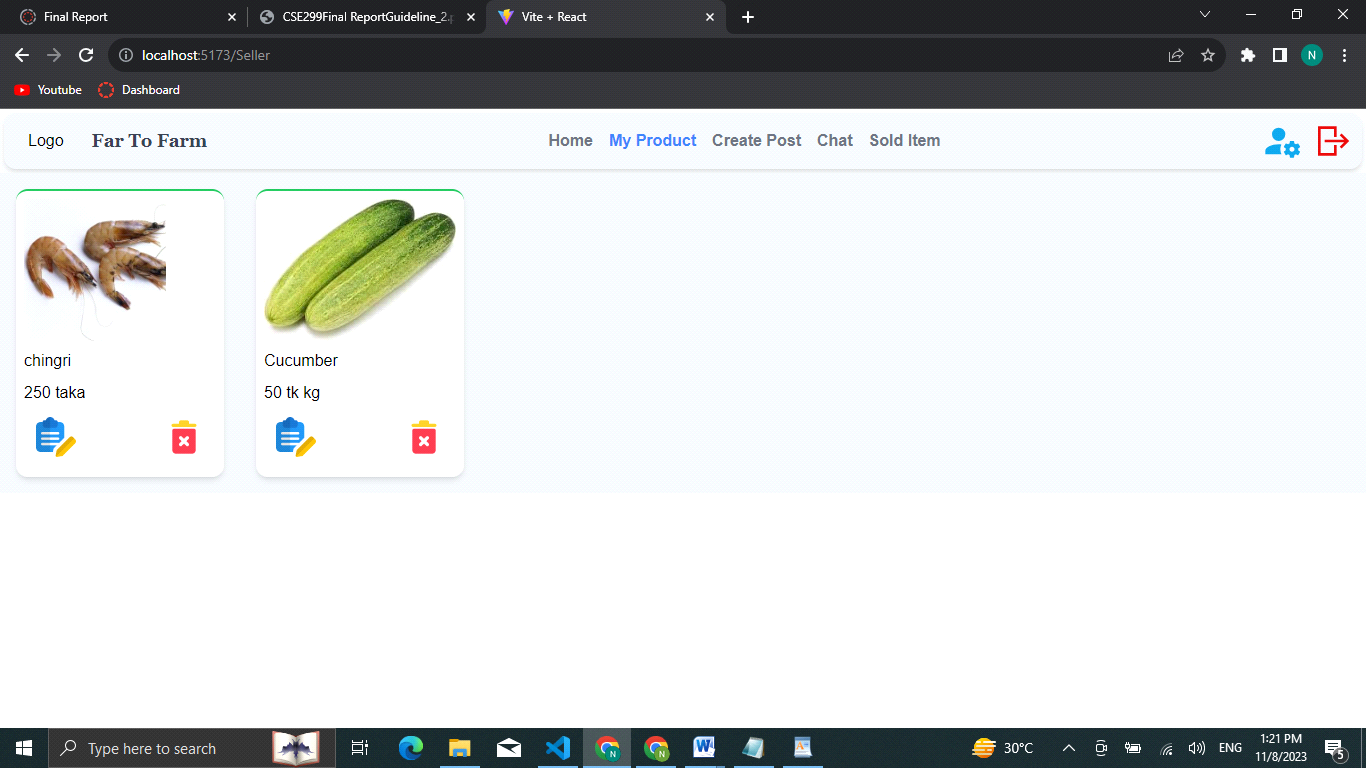


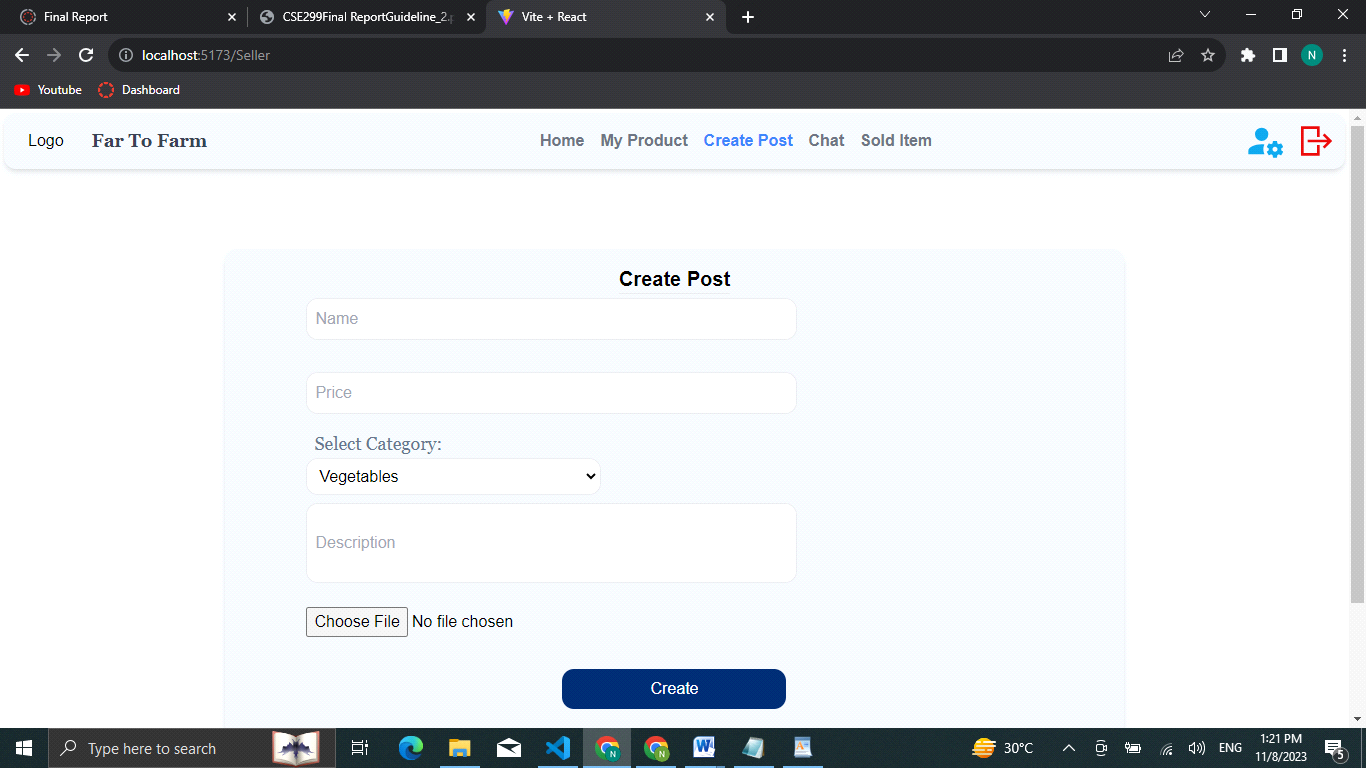
**Registration**

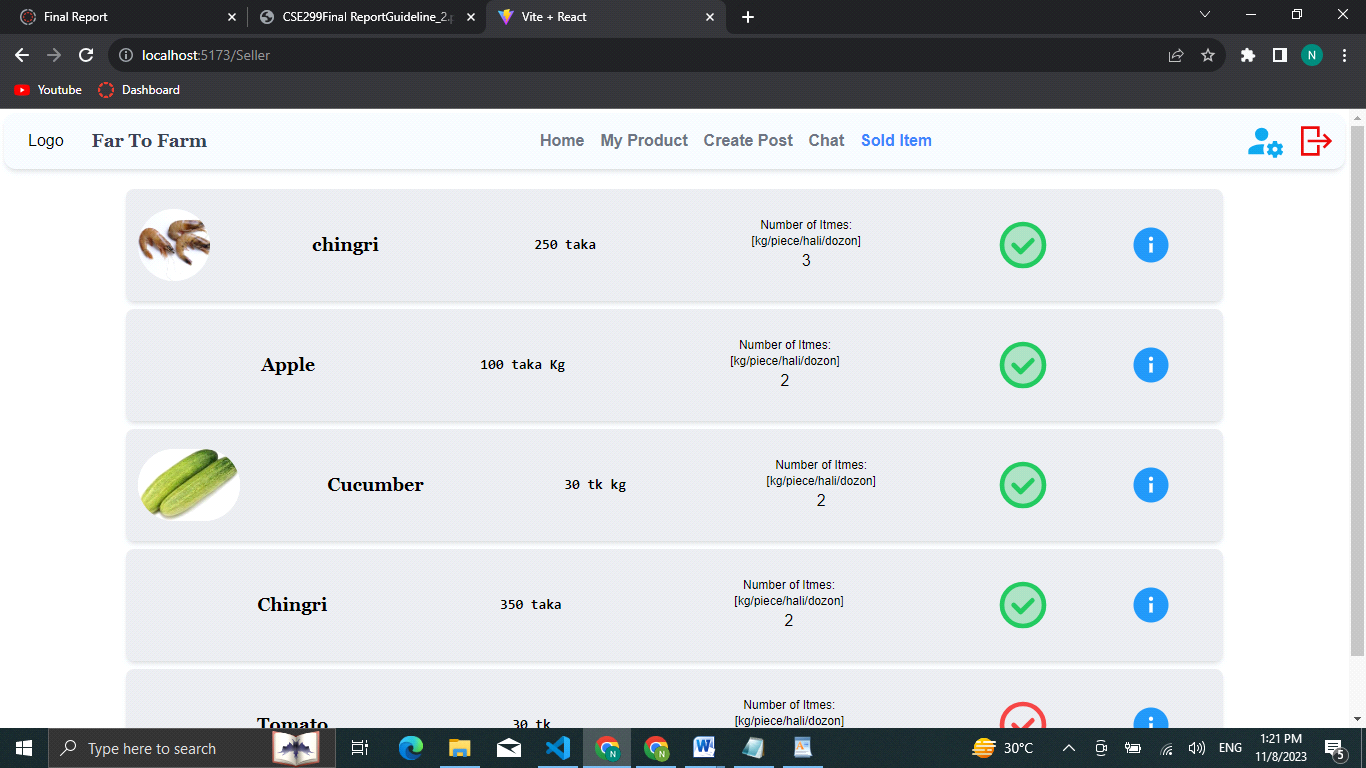
****

**Home [Seller]**

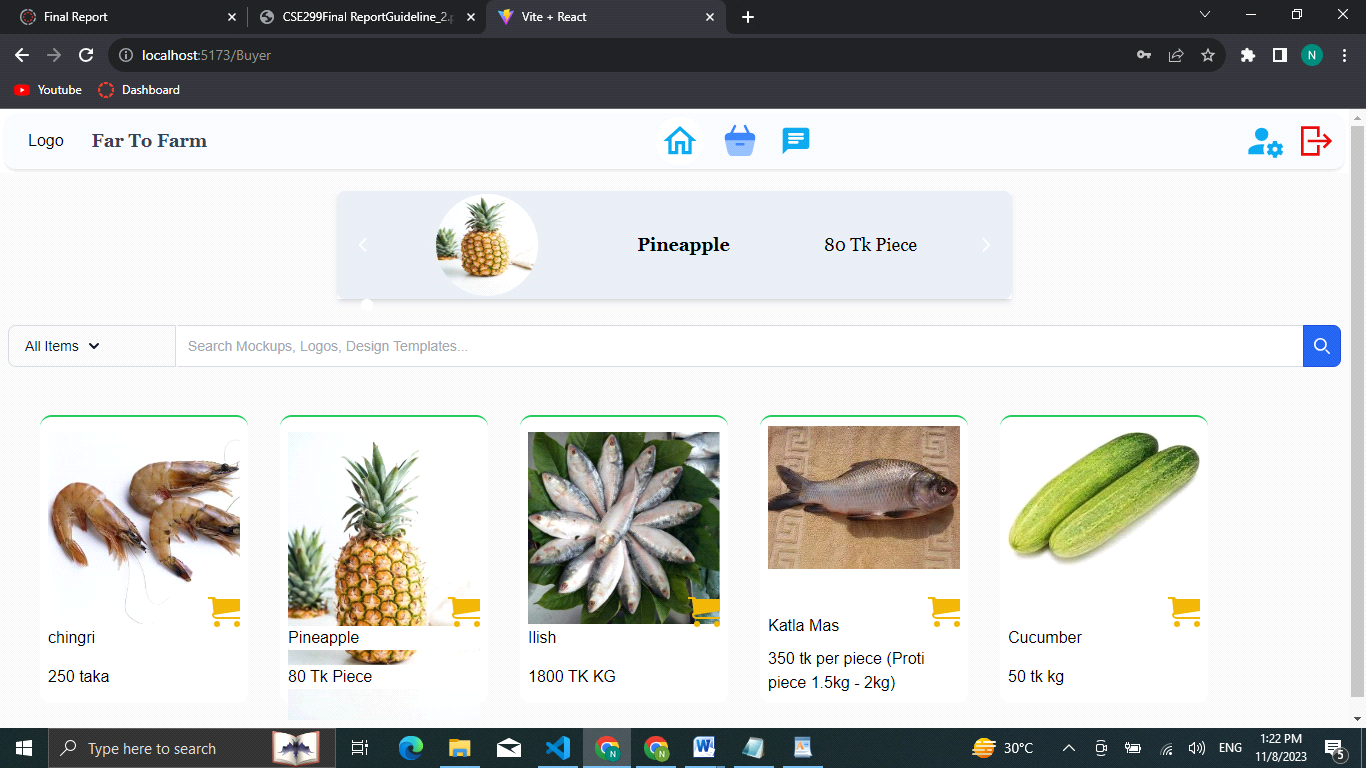
****

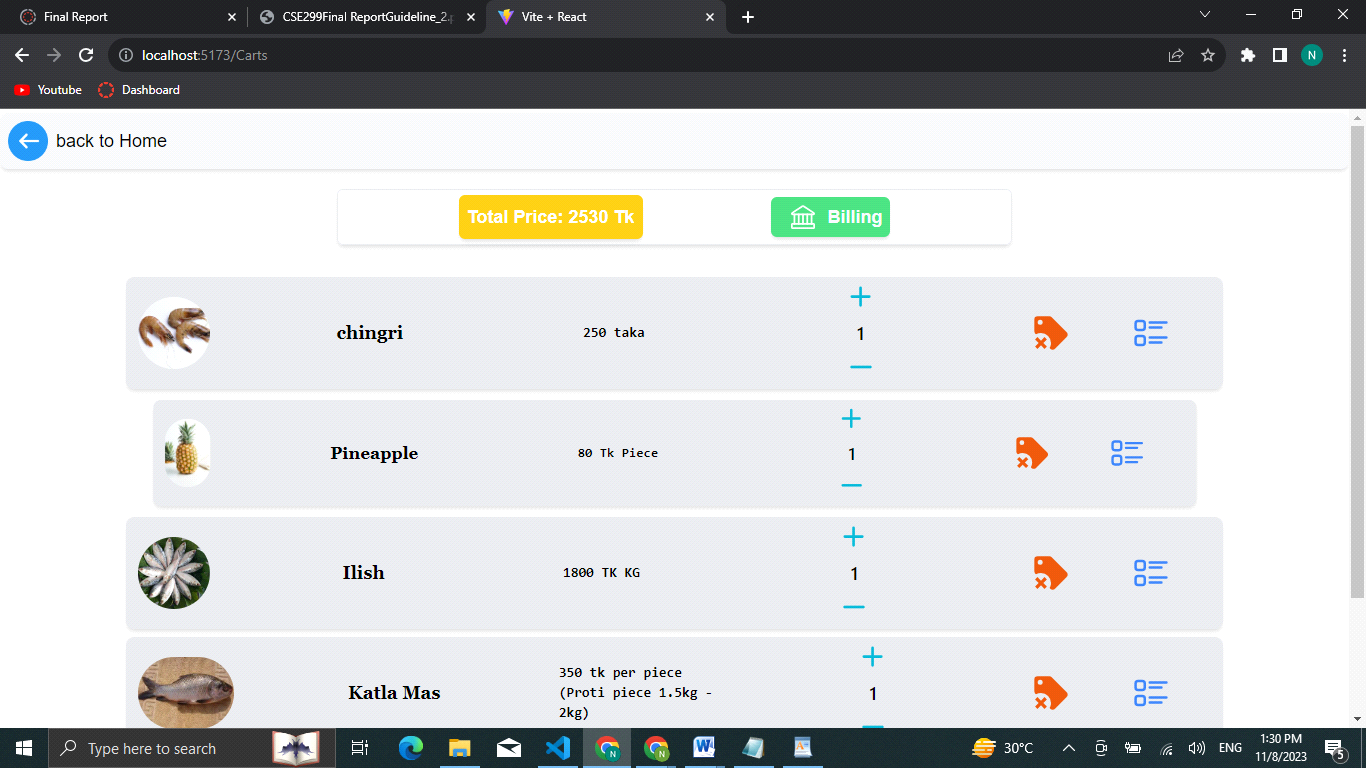


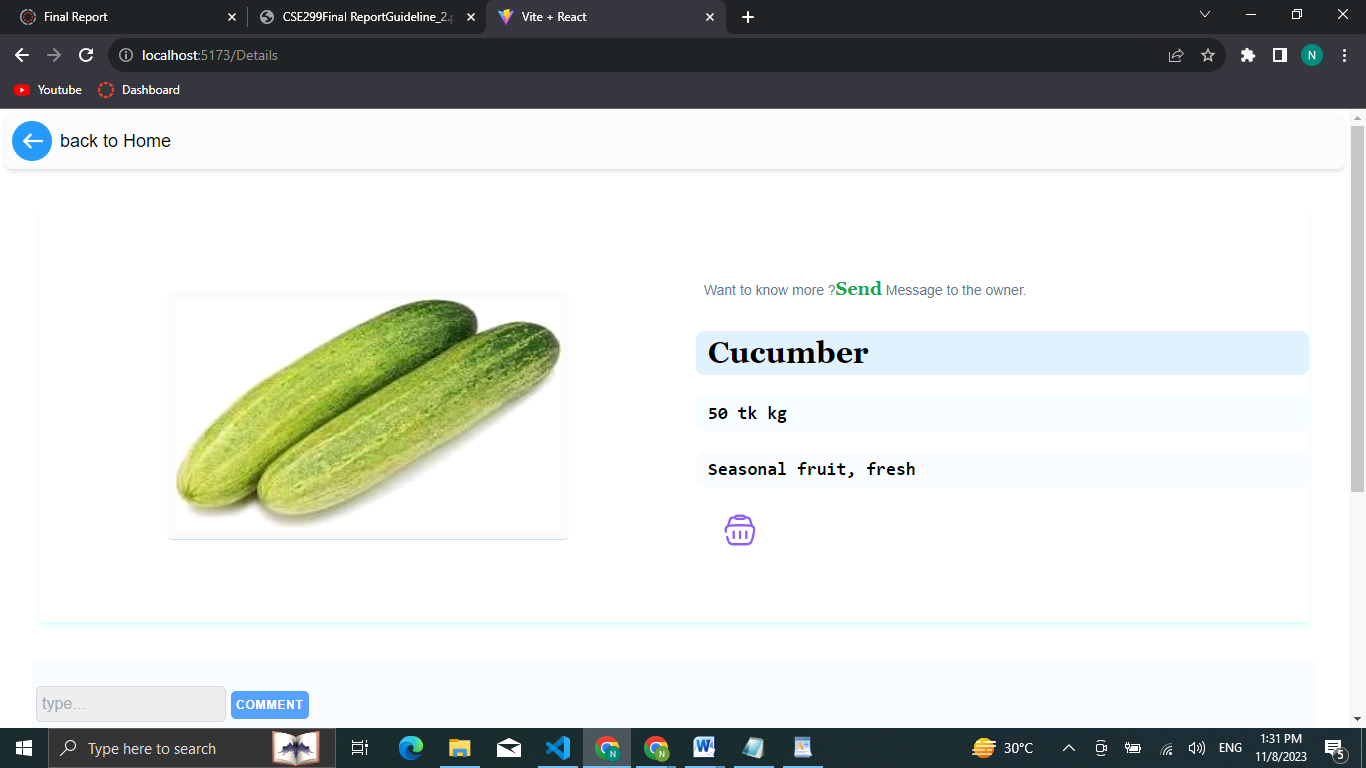


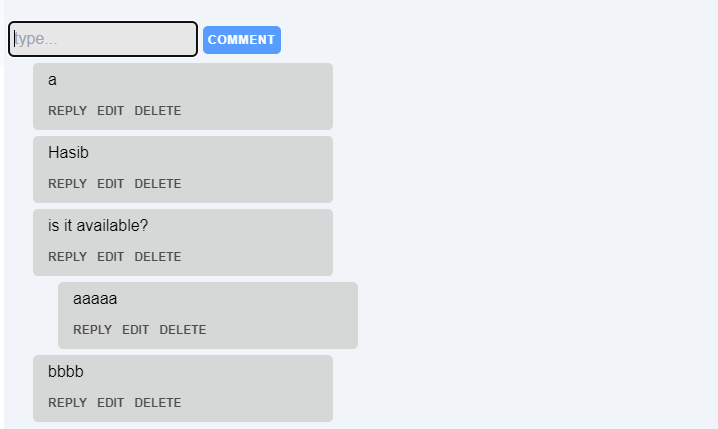


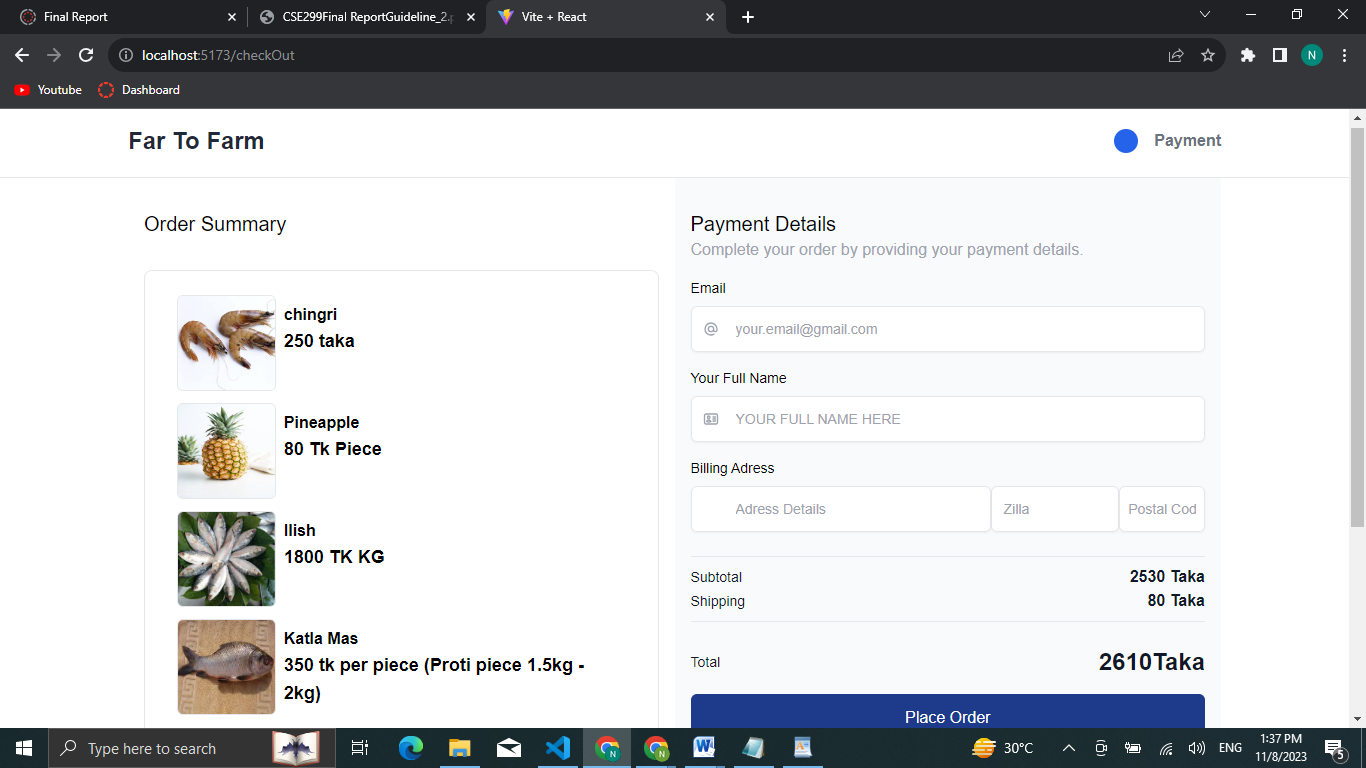
**Home [Buyer]:**

****

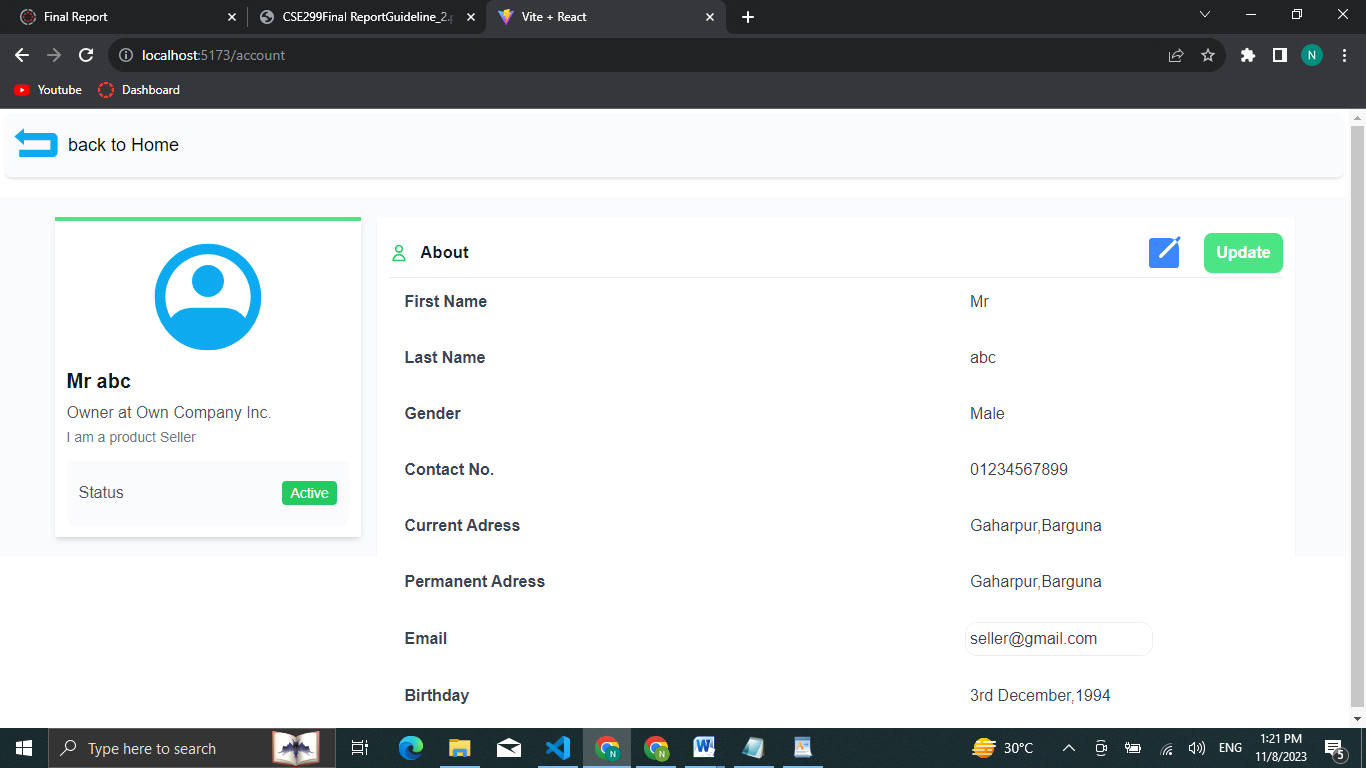








**Account**



# Chapter 7: Conclusion

## 7.1 Summary

The "E-Agriculture Management System Website" holds great potential to revolutionize the agricultural landscape. It aims to create an efficient, transparent, and inclusive ecosystem, leveraging digital technology to empower farmers and enhance consumer experiences. This platform addresses critical challenges while capitalizing on opportunities to promote sustainability and economic growth.

## 7.2 Limitations

While the "E-Agriculture Management System Website" is promising, it's essential to acknowledge its limitations. These limitations may include technical challenges, infrastructure constraints, and the need for ongoing support and training for users.

## 7.3 Future Improvements

To maximize the platform's impact, potential future improvements can be considered. These improvements might involve expanding features, enhancing user interfaces, or developing mobile applications to increase accessibility.

In the conclusion, the "E-Agriculture Management System Website" demonstrates its potential to benefit both farmers and consumers by providing a virtual marketplace, promoting transparency, sustainability, and technological innovation. It has the power to bridge the rural-urban divide, empower women in agriculture, and shape the future of the agricultural sector.

Through this platform, we envision a future where agriculture is efficient, transparent, and economically rewarding for all stakeholders involved. As we embark on this journey toward agricultural transformation, the potential benefits of the "E-Agriculture Management System Website" are poised to enrich lives, empower communities, and contribute to a more sustainable and prosperous agricultural ecosystem.

# References:

1. OECD. "Technology and Digital in Agriculture." [Online]. Available: www.oecd.org. [Accessed: July 25, 2019].

2. FAO. "Information and Communication Technologies for Sustainable Agriculture." [PDF]. Available: www.fao.org. [Accessed: June 9, 2016].