Udgam: A Community-Driven Organic Marketplace

A Project Report

Submitted by:

Aditya Dave (AU2140100)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

at

Ahmedabad University

School of Engineering and Applied Science (SEAS)

Ahmedabad, Gujarat

April, 2025

DECLARATION

I hereby declare that the project entitled “**Udgam: A Community-Driven Organic Marketplace**” submitted for the B. Tech. (**Computer Science And Engineering**) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of Student

Date: 24/04/2025

Place: Ahmedabad

CERTIFICATE

This is to certify that the project titled "**Udgam: A Community-Driven Organic Marketplace**" is the bona fide work carried out by **Aditya Dave**, a student of B. Tech. (**Computer Science And Engineering**) of School of Engineering and Applied Science at Ahmedabad University during the academic year 2024-2025, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science And Engineering** and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

This project was done under the supervision of the faculty mentor **Professor Sanjay Chaudhary**.

Signature of Faculty Mentor

Date: 24/04/2025

Place: Ahmedabad

Abstract

"Udgam" is an app developed to boost sales of organic products and encourage farmers to practice organic farming. During my research in this domain, I was unable to find any major platform where they have a community of individuals working in organic farming. To fill this void, I created a community where people can discuss organic farming. Marketplaces are there for organic products but not very useful for farmers producing in low volumes. When a person starts practicing organic farming, usually they start earning after a while. To promote organic farming practices, it is necessary to have some assistance in the initial years. Our marketplace model will act as an initial source of income for farmers. Contemporary technologies and machine learning algorithms are used to enhance usability.

Table of Contents

List of Figures

Gantt Chart

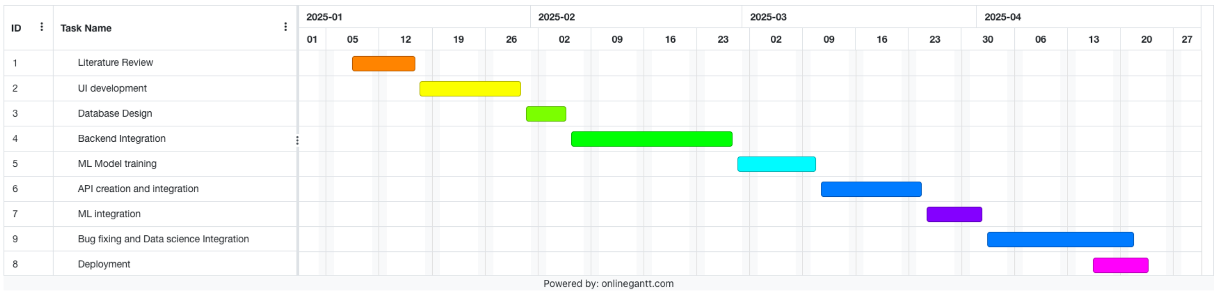


Figure 0.0.1: Gantt Chart

Chapter 1

Introduction

Food is necessary for human beings. A person's diet will decide how their body will function in the future. Vegetables, dairy products, and grains are part of every household's diet. Today, we are not able to meet our daily requirements of vegetables, and due to that, farmers are growing plants unconventionally. In this whole process, they are not aware of how they are affecting their farm's soil, consumers, and their own health. We are consuming products with high amounts of chemicals, which may not affect our bodies in the short run, but in the long run, they slowly harm our health, leading to deficiencies in vitamins and nutrients.

Our solution will emphasise awareness and availability. Awareness will open the doors for organic products. Availability will ensure that people get authentic products with minimal effort. I integrated both aspects in one app called Udgam.

1.1 | Project Definition

The goal is to develop an app with a community of individuals who are working in the field of organic farming. Marketplace will assist people in having genuine products from legitimate farmers. Cutting-edge technologies will assist with creating practical insights for farmers. This will make the app more meaningful and goal-oriented.

1.2 | Project Objectives

The main objectives of this project are

- To create a community of individuals where people will discuss organic farming.
- To create a category-wise filter for posts so that users can see the desired content only.
- To create an infinitely scrollable news feed where users can translate news into Hindi and listen to news in selected languages.
- To create an admin panel to manage orders and farmer details.
- To create a smooth onboarding process for farmers.

- To create a dashboard for farmers so that they can analyse their sales data and integrate decision science to provide actionable insights.
- To create a marketplace where people can buy authentic organic products.
- To create a recommendation system for users using a machine learning algorithm (matrix factorization) to give personalized recommendations.

Chapter 2

Literature Survey

Understanding the problem from the farmers' perspective leads to the creation of sustainable solutions. I spoke to a farmer named Vardha Rajan from Pune, who has been practicing organic farming. He told me that switching to organic farming involves high labor costs, greater maintenance of animals—since they help produce natural fertilizers—and lower crop yields in the initial years. After this conversation, I realized that the main bottleneck lies at the beginning: farmers lack access to resources that can help them generate income during the first few years of practicing organic farming.

Today's world is driven by technology, and we have almost all the solutions for our routine life. Specifically for agriculture, free solutions are provided by the government and due to that, there is a scope for improvement in this domain for the same understanding, contemporary available solutions are good to start with.

2.1 | Related Work

Here is the overview of existing solutions around our problem statement:

- FarmRise: This app provides services like daily news, crop doctor, mandi prices, etc. Good traits of the app are multilingual support, easy navigation and user interface. There is a scope for improvement in this app because there are so many functionalities in one place, so it becomes very difficult for users to use it efficiently.
- Kisaan Ai: This is one of the best websites existing on the internet for farmers. Here, farmers can chat with their AI model in a a regional language.
- Farmer.Chat: This website provides the same services as an AI chatbot, but the only difference is that it has some human involvement as well and focuses more on the USA region.

2.2 | Tools and Technologies

1. Front-End: Flutter is used for the user interface. Flutter is used cross-platform so that we can operate everything in a single codebase.

2. Back-end: For the back-end, we have used the Dart language. It is an object-oriented language.

3. Database: For the database we have used Supabase. It is built on top of PostgreSQL.

4. APIs: For news related to agriculture only, we have used a standard news API, which returns responses in JSON format. For the recommendation system, we converted the algorithm to Fastapi using Python.

5. Machine Learning: For the BERT model and the recommendation system, we have used ML libraries such as scikit, Hugging Face, and Transformers.

6. Tools: In Postman, we have done testing of our news API. For fine-tuning the BERT model, I have used Google Colab.

Chapter 3

Methodology

3.1 | Learn technology

The first step in the project was to learn the technologies required to create an application. I was familiar with Flutter. I learned how to use Supabase backend services and authentication services, Flutter, Fastapi service for recommendation systems, and API integration. I learned how to integrate this all with the front end. I explored various state management techniques and decided to work on the GET state management technique. After learning basic functionalities, I designed a use case diagram to understand the flow.

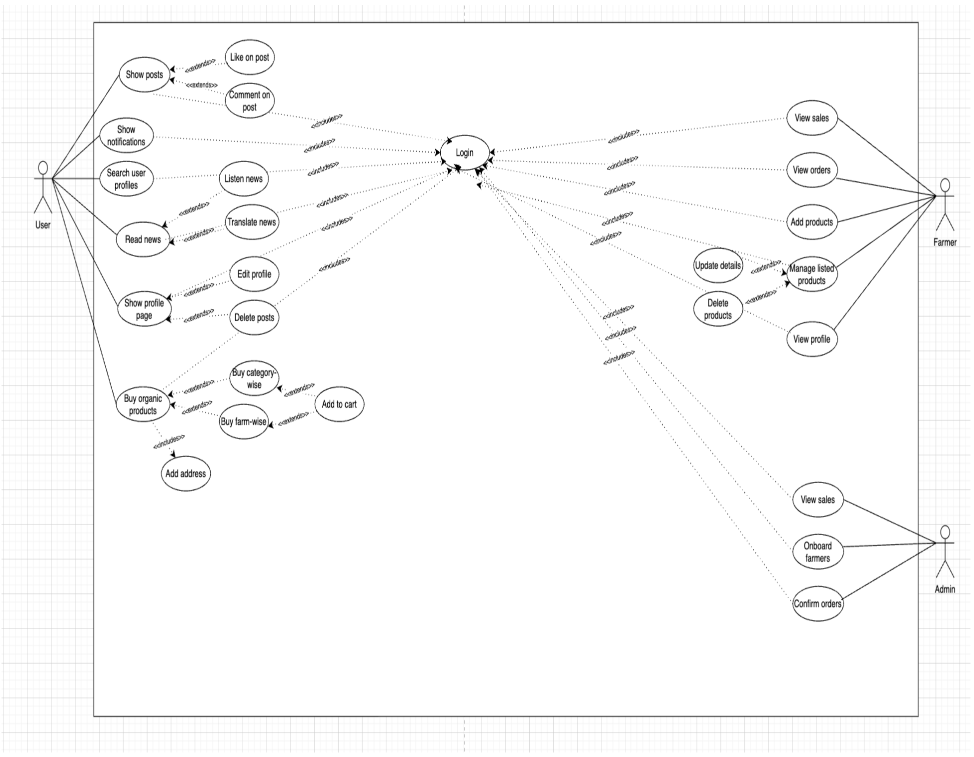


Figure 3.1.1: Usecase Diagram

3.2 | Setup and design database

I started with setting up an authentication system; it is essential to manage user-specific data and enable a personalised experience in the app. Supabase turned out to be the best substitute for Firebase because it has a built-in authentication service and is built on Postgresql, which aligned with my decision to work with RDBMS. I started with a users table and designed other tables along with development. The priority was to normalise the database for consistency and easy-to-retrieve data in the UI.

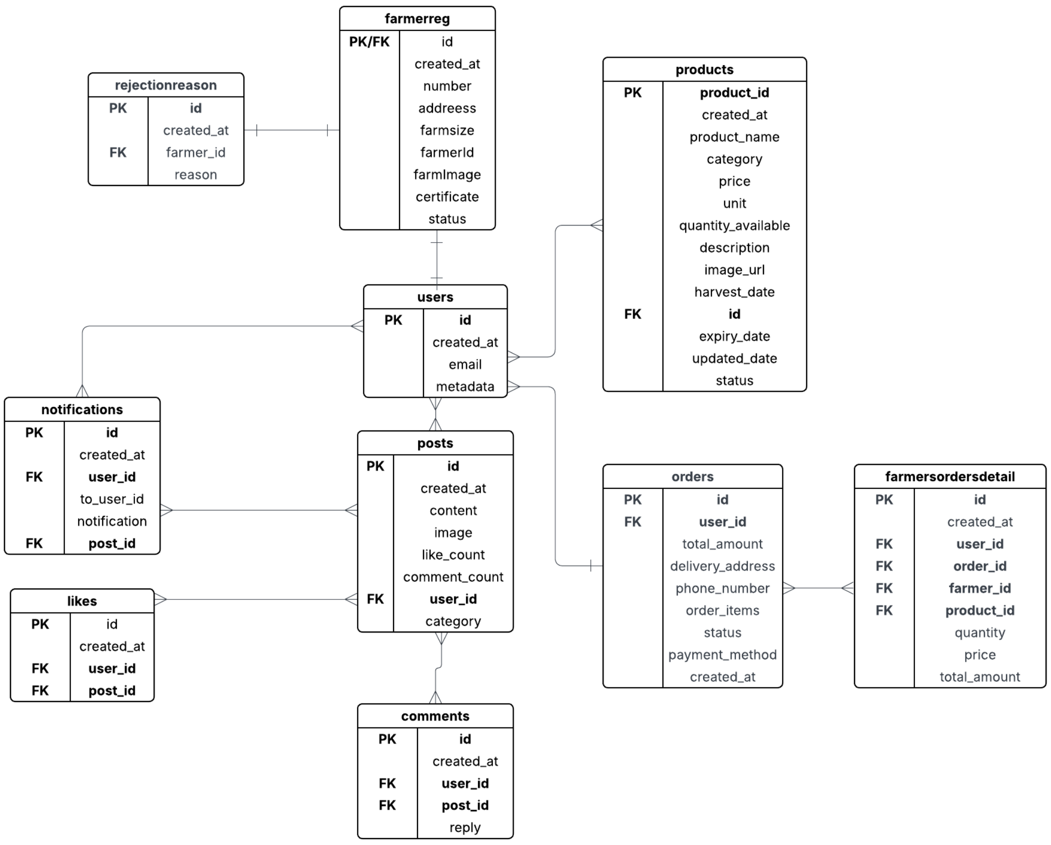


Figure 3.2.1: ER Diagram

3.3 | Feature addition

We have three major functionalities in the app: community, news, and marketplace. Started with the community feature. In the community, I first fetched the current user ID from local storage, which was stored at the time of login, and displayed profile details. On the add post page, users can add content in the form of images, video or text. On the main page, I listed all the posts sorted by upload date. For the marketplace, first created an onboarding process for farmers, then created farmer-side and user-side UI parts with basic backend services.

3.4 | API Integration

In the app, I have used the News API to display news related to agriculture only. I created a data model for the response coming from the API and stored it in a list. After that integrated with the frontend. With each news card, I integrated two features. The first one is live translation, and the second one is text-to-speech. For text-to-speech, I have used the Elevenlabs service. I set up which voice to use for specific languages. For the real-time translation, I have used Google's built-in package provided by Flutter.

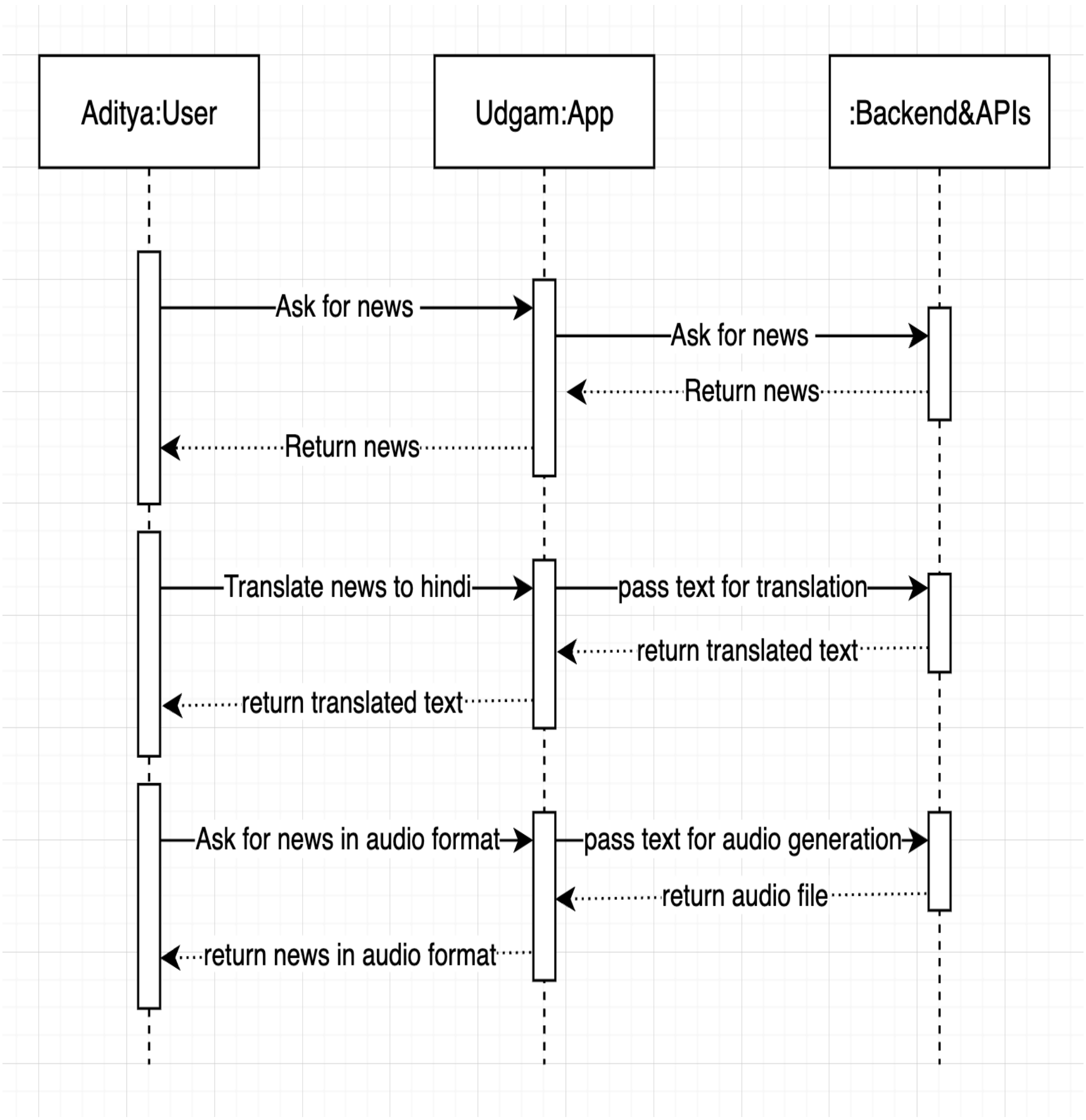


Figure 3.4.1: Sequence Diagram

3.5 | ML Model training

After creating a basic app, it is time to integrate ML into it. In the community, I wanted to create an algorithm that could work as a detector for the text so that we can maintain the decorum of the app by not allowing people to post anything irrelevant. For the same, I chose the BERT model from Hugging Face's transformer library. BERT is trained on words that have been pulled out from Wikipedia. So this BERT is used for fine-tuning for specific purposes, exactly what I wanted. I wrote around 50 sentences for organic farming just to train the BERT model for our use.

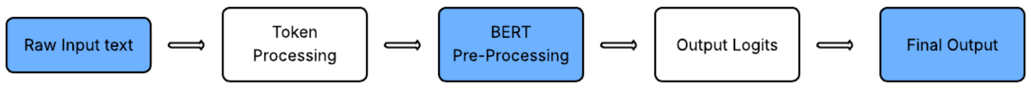


Figure 3.5.1: Bert Model

In the marketplace model, a recommendation system is a must for a better user experience. I chose ML's matrix factorisation algorithm for the recommendation system. It creates a sparse matrix for each user and performs operations on it. For training model, I gave order details and user details as input.

3.6 | ML Integration

I trained BERT on almost 100 sentences related to agriculture and non-agriculture by labeling them 1 and 0, respectively. Because the data is very small for training the algorithm for a large task, I tested it in Google Colab and did not integrate it in the first version of the app so that users can post freely. For the recommendation system, I first created a whole algorithm where we take the logged-in user ID and perform tasks on that. Output was product IDs. So I converted the algorithm into FastAPI so that users can directly use the API and the app do not have any load on itself.

3.7 | Deployment

We need a special account to host the app on the store. I created an APK using Flutter commands and shared it via various platforms. For the API, I hosted it locally and accessed it using an IP address. Eventually hosted FastAPI on Render.

Chapter 4

Results

The project aims to open doors for organic products. The result is an App that has functionalities like community and marketplace, using which we can fulfil requirements.

4.1 | Project Outcomes

4.1.1 | Authentication

For the login page in Figure 4.1.1, the user has to select their role. The default role is the user, and the other two roles are admin and farmer. In the figure 4.2.2, we have the farmer registration process. In this, the farmer has to upload all the details to register.

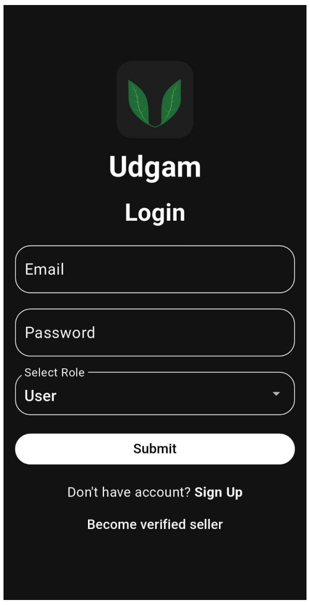


Figure 4.1.1: Login

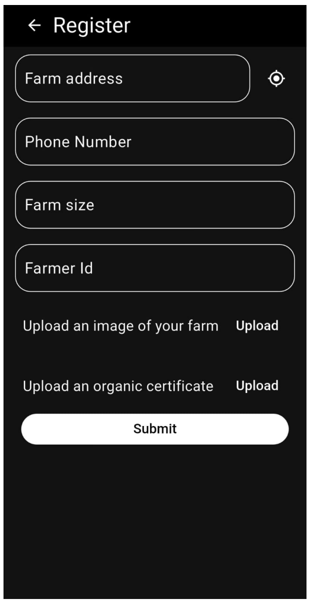
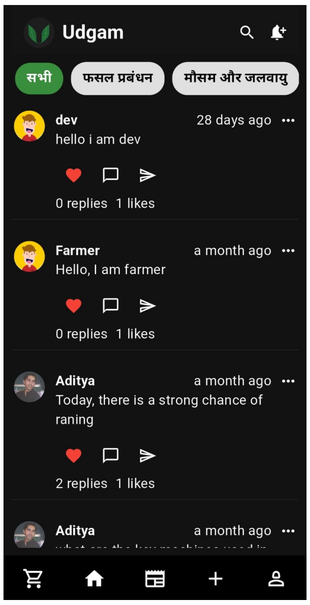

Register
Farm address
Phone Number
Farm size
Farmer Id
Upload an image of your farm Upload
Upload an organic certificate Upload
Submit

Figure 4.1.2: Farmer login

4.1.2 | Community

In the community feature, we have a home screen in Figure. Users will be able to see all the posts starting from the newest to the oldest. We have a search bar in Figure 2 to explore specific profiles. The notification button, where users can see who liked and commented on the post. In the top bar, we have a category-wise button to filter posts based on category. On the Profile page, the user can edit profile details like profile image and description.



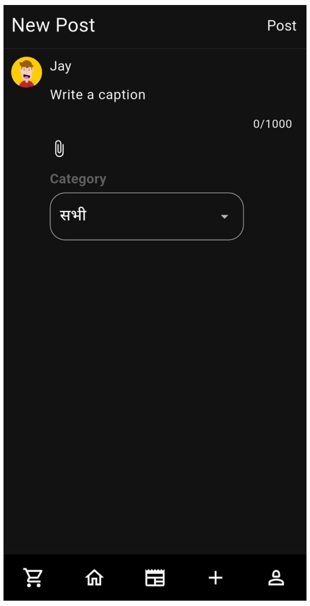
dev 28 days ago
hello i am dev
0 replies 1 likes

Farmer a month ago
Hello, I am farmer
0 replies 1 likes

Aditya a month ago
Today, there is a strong chance of raning
2 replies 1 likes

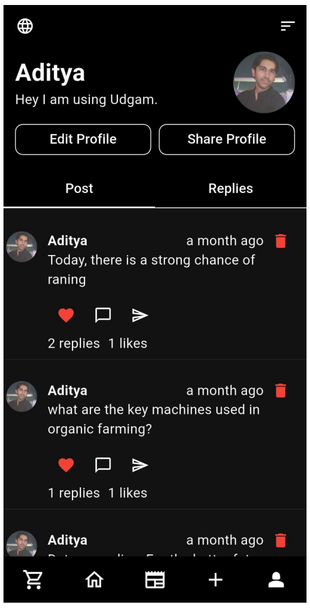
Aditya a month ago

Figure 4.1.3: Home



Jay
Write a caption
0/1000
Category
सभी

Figure 4.1.4: Add post



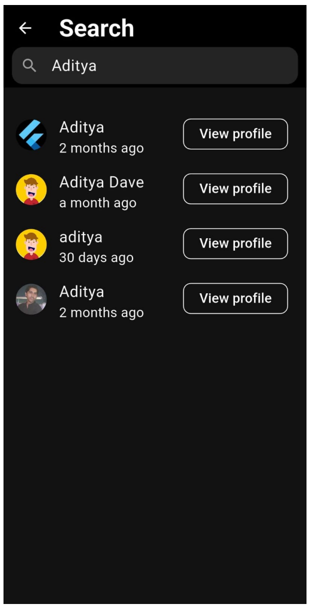
Aditya
Hey I am using Udgam.

Aditya a month ago
Today, there is a strong chance of raning
2 replies 1 likes

Aditya a month ago
what are the key machines used in organic farming?
1 replies 1 likes

Aditya a month ago

Figure 4.1.5: Profile



Aditya

Aditya	2 months ago	View profile
Aditya Dave	a month ago	View profile
aditya	30 days ago	View profile
Aditya	2 months ago	View profile

Figure 4.1.6: Search

4.1.3 | Marketplace

In the marketplace, we have category-wise and farm-wise listed products. Once the user adds all required products into carts, we have output of recommendation system's output. Below that we have the address where to place the order and the mobile number of the user. After that when the user clicks on the payment, it will navigate to the payment screen where the user can select the mode of payment and order items.

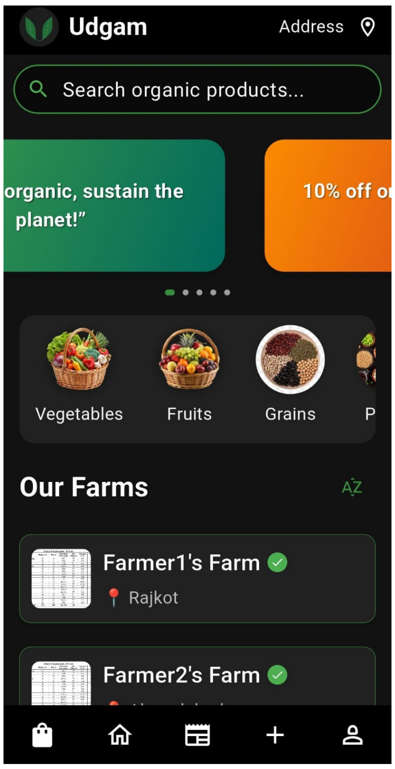

Udgam
Address
Search organic products...
organic, sustain the planet!"
10% off o
Vegetables
Fruits
Grains
P
Our Farms
Farmer1's Farm
Rajkot
Farmer2's Farm

Figure 4.1.7: Marketplace home

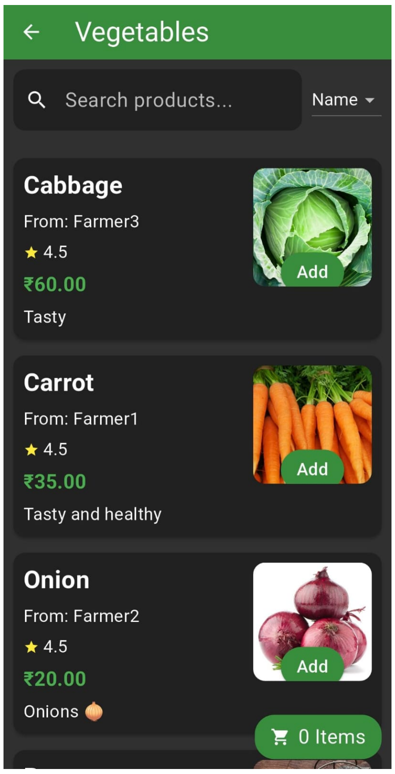

Vegetables
Search products...
Name
Cabbage
From: Farmer3
★ 4.5
₹60.00
Tasty
Add
Carrot
From: Farmer1
★ 4.5
₹35.00
Tasty and healthy
Add
Onion
From: Farmer2
★ 4.5
₹20.00
Onions
Add
0 Items

Figure 4.1.8: Category-wise products

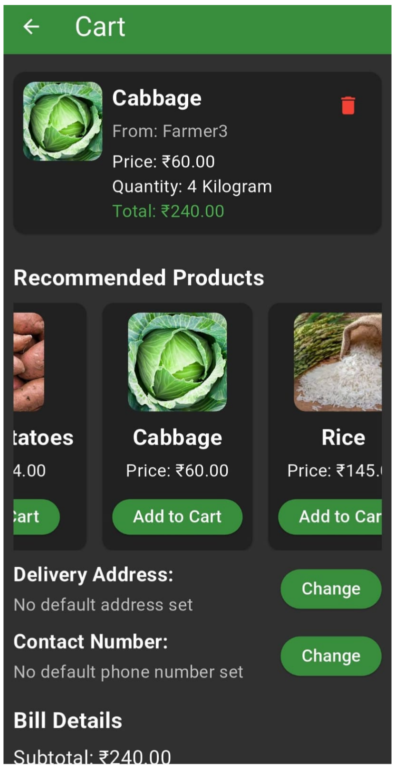
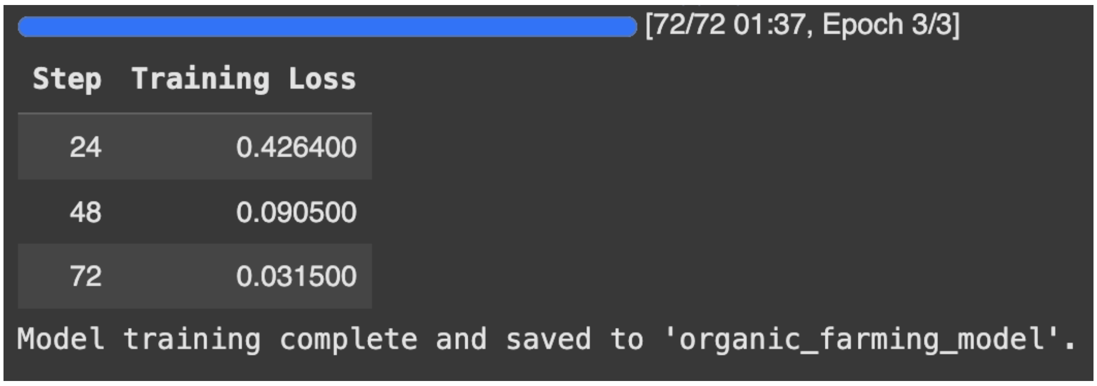

Cart
Cabbage
From: Farmer3
Price: ₹60.00
Quantity: 4 Kilogram
Total: ₹240.00
Recommended Products
tatoes
4.00
art
Cabbage
Price: ₹60.00
Add to Cart
Rice
Price: ₹145.
Add to Car
Delivery Address:
No default address set
Change
Contact Number:
No default phone number set
Change
Bill Details
Subtotal: ₹240.00

Figure 4.1.9: Cart

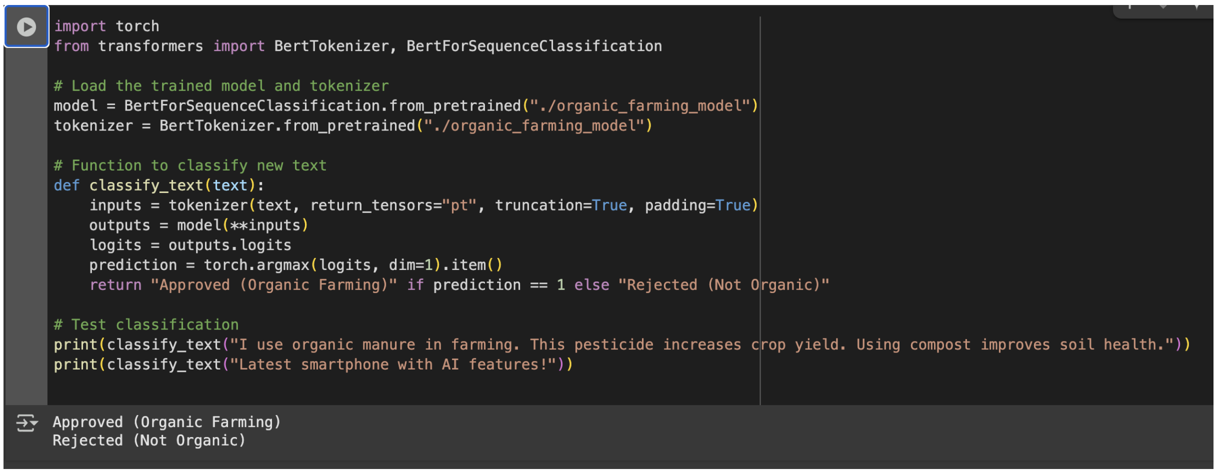
4.1.4 | BERT Model and Recommendation System Output

In the BERT model, as I mentioned earlier, we trained over 100 different sentences and labeled them with 0 and 1. The output of the tested sentences is displayed below.



```
[72/72 01:37, Epoch 3/3]
Step  Training Loss
  24       0.426400
  48       0.090500
  72       0.031500
Model training complete and saved to 'organic_farming_model'.
```

Figure 4.1.10: BERT Trained Model

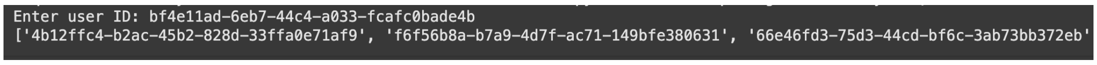


```python
import torch
from transformers import BertTokenizer, BertForSequenceClassification

# Load the trained model and tokenizer
model = BertForSequenceClassification.from_pretrained("./organic_farming_model")
tokenizer = BertTokenizer.from_pretrained("./organic_farming_model")

# Function to classify new text
def classify_text(text):
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    outputs = model(**inputs)
    logits = outputs.logits
    prediction = torch.argmax(logits, dim=1).item()
    return "Approved (Organic Farming)" if prediction == 1 else "Rejected (Not Organic)"

# Test classification
print(classify_text("I use organic manure in farming. This pesticide increases crop yield. Using compost improves soil health."))
print(classify_text("Latest smartphone with AI features!"))
```

```
Approved (Organic Farming)
Rejected (Not Organic)
```

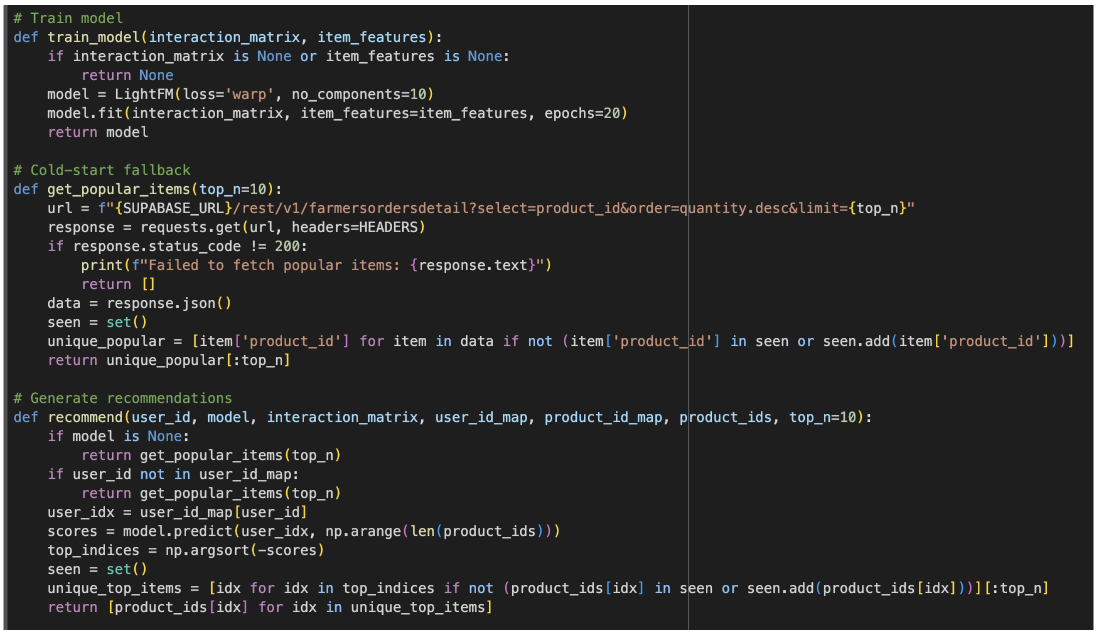
Figure 4.1.11: BERT Model Testing

For the recommendation system, instead of integrating it into the app, I converted the algorithm in FastAPI, but to display the output, I am using the algorithm that I tested in Google Colab.



```
Enter user ID: bf4e11ad-6eb7-44c4-a033-fcafc0bade4b
['4b12ffc4-b2ac-45b2-828d-33ffa0e71af9', 'f6f56b8a-b7a9-4d7f-ac71-149bfe380631', '66e46fd3-75d3-44cd-bf6c-3ab73bb372eb'
```

Figure 4.1.12: Recommendation System Model Testing



```python
# Train model
def train_model(interaction_matrix, item_features):
    if interaction_matrix is None or item_features is None:
        return None
    model = LightFM(loss='warp', no_components=10)
    model.fit(interaction_matrix, item_features=item_features, epochs=20)
    return model

# Cold-start fallback
def get_popular_items(top_n=10):
    url = f"{SUPABASE_URL}/rest/v1/farmersordersdetail?select=product_id&order=quantity.desc&limit={top_n}"
    response = requests.get(url, headers=HEADERS)
    if response.status_code != 200:
        print(f"Failed to fetch popular items: {response.text}")
        return []
    data = response.json()
    seen = set()
    unique_popular = [item['product_id'] for item in data if not (item['product_id'] in seen or seen.add(item['product_id']))]
    return unique_popular[:top_n]

# Generate recommendations
def recommend(user_id, model, interaction_matrix, user_id_map, product_id_map, product_ids, top_n=10):
    if model is None:
        return get_popular_items(top_n)
    if user_id not in user_id_map:
        return get_popular_items(top_n)
    user_idx = user_id_map[user_id]
    scores = model.predict(user_idx, np.arange(len(product_ids)))
    top_indices = np.argsort(-scores)
    seen = set()
    unique_top_items = [idx for idx in top_indices if not (product_ids[idx] in seen or seen.add(product_ids[idx]))][:top_n]
    return [product_ids[idx] for idx in unique_top_items]
```

Figure 4.1.13: Recommendation System Model Testing

4.1.5 | Admin

For the Admin we have a dashboard to showcase all the details. Farmers registration page from where admin will onboard new farmers or reject the farmer. In the third screen, the admin will confirm the order that was placed by the user.

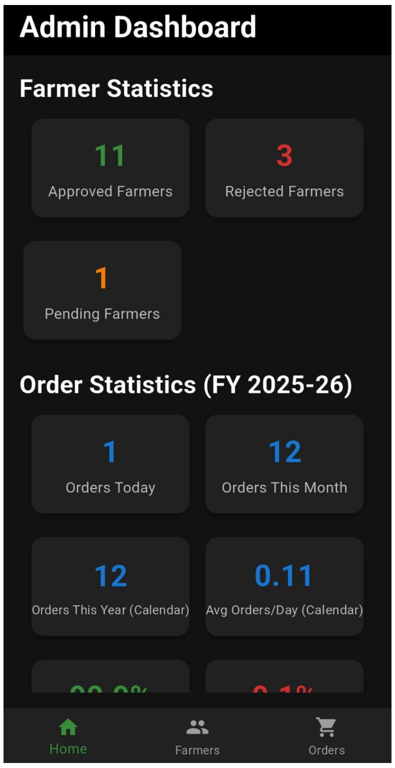


Figure 4.1.14: Admin dashboard

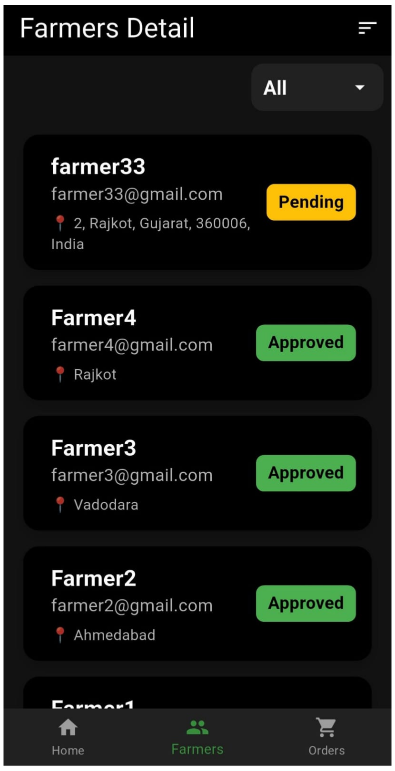


Figure 4.1.15: Farmers detail page

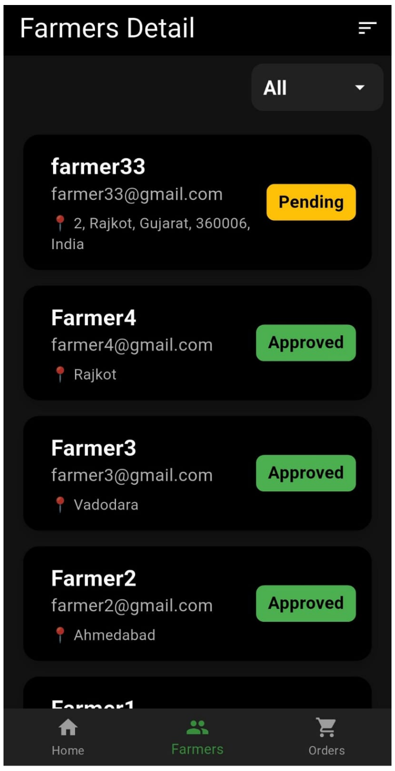
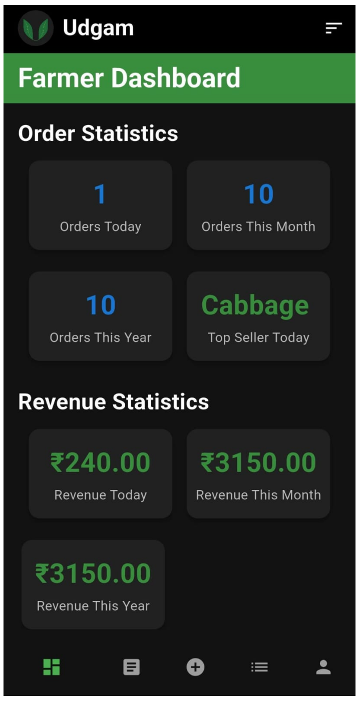


Figure 4.1.16: Order confirmation screen

4.1.6 | Farmers

For the farmer side, we have a dashboard with data analysis and an AI-powered suggestion system. Then we have order details, add a new product screen, list product screen and profile screen.


Udgam

Farmer Dashboard

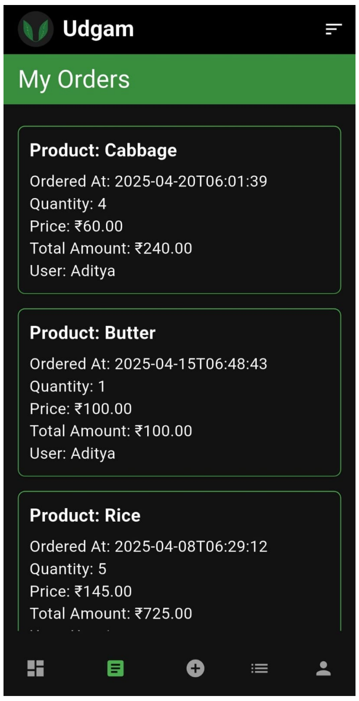
Order Statistics

1 Orders Today
10 Orders This Month
10 Orders This Year
Cabbage Top Seller Today

Revenue Statistics

₹240.00 Revenue Today
₹3150.00 Revenue This Month
₹3150.00 Revenue This Year

Figure 4.1.17: Farmer Dashboard


Udgam

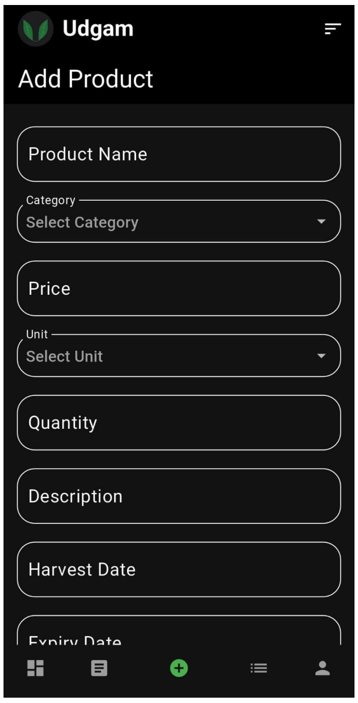
My Orders

Product: Cabbage
Ordered At: 2025-04-20T06:01:39
Quantity: 4
Price: ₹60.00
Total Amount: ₹240.00
User: Aditya

Product: Butter
Ordered At: 2025-04-15T06:48:43
Quantity: 1
Price: ₹100.00
Total Amount: ₹100.00
User: Aditya

Product: Rice
Ordered At: 2025-04-08T06:29:12
Quantity: 5
Price: ₹145.00
Total Amount: ₹725.00

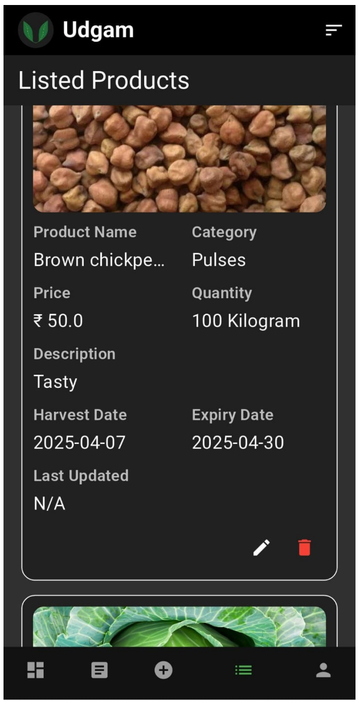
Figure 4.1.18: Farmer orders screen


Udgam

Add Product

Product Name
Category: Select Category
Price
Unit: Select Unit
Quantity
Description
Harvest Date
Expiry Date

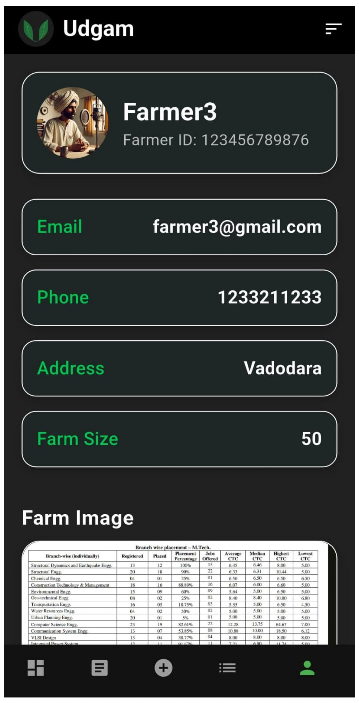
Figure 4.1.19: Add product screen


Udgam

Listed Products

Product Name: Brown chickpe...	Category: Pulses
Price: ₹ 50.0	Quantity: 100 Kilogram
Description: Tasty	
Harvest Date: 2025-04-07	Expiry Date: 2025-04-30
Last Updated: N/A	

Figure 4.1.20: Listed product screen


Udgam

Farmer3
Farmer ID: 123456789876

Email	farmer3@gmail.com
Phone	1233211233
Address	Vadodara
Farm Size	50

Farm Image

Figure 4.1.21: Farmer profile screen

4.2 | Learning Outcomes

In my undergrad, I learned subjects like database management systems, machine learning, operating systems, and human-computer interaction, and used them in Udgam.

1. Database Management: Database is the core of Udgam. We designed a normalised database and used the same as a source of data. Implemented RLS policies for CRUD operations.
2. API creation: In the project I have used many APIs many times but never created them before. I created an algorithm first and then converted that into FastAPI using Python. In API creation learned about GET and POST requests.
3. Machine learning: Due to conflict with other classes, I was not able to opt machine learning course in my undergrad, but here I explored the popular scikit library and learned the matrix factorization algorithm.

4.3 | Real-world application

The app is designed for real-world use cases so all the features are developed in a way to use efficiency but here are some of the use cases of our app:

1. Ask queries to experts: We have so many domain-specific LLMs to answer industry-specific questions. But for some questions, humans are the reliable source. Keeping this in mind, the community feature will connect farmers to the experts.
2. Stay updated: In the news app we have categories nowadays, but for agriculture, news is available in English only, so we have two options: one for translation and one for audio on and off. This is how we made the experience more interactive.
3. Marketplace: Today we have so many marketplaces with more vegetables and fruits from common platforms like Zepto to specific platforms like Big Basket. The problem is organic farming requires time but due to losses in the initial years, farmers do not have the motivation to start practising it. Our platform will help those farmers to list their products and, in sales, to give support.

Chapter 5

Conclusion and Future work

In conclusion, Udgam is an app created with the intent to support farmers. Udgam is built on cutting-edge technology, which will help farmers to produce more and consumers to consume more. This app empowers small farmers to practice conventional farming. In future, we will try to scale this app as much as we can.

This project can cater to a large audience but some work can be done in future to enhance user experience on this platform:

1. Voice-operated shopping experience: AI is not the future; it is present. LLM's are revolutionizing the whole user experience. In the Marketplace model instead of a recommendation system, we can design voice assistance that will understand user requirements first and then LLM will search for required products from the app and will design a cart for them. For example, if the user tells the assistant that I want to prepare MatarPaneer for dinner, the assistant will suggest products like Paneer, peas, coriander etc.

2. AI-powered query solver: The community section is designed to spread education and awareness around organic farming and products. We can implement an AI agent that will answer farmer's questions posted on the app. So the process will be like a farmer will post the question in the app then our LLM will answer the question in the comment section and this is how people will interact with the AI bot.

3. Subscription model integration: Vegetables and dairy products are necessary products for every household. Instead of buying products every day, we can create a subscription model where people can buy for the next 7 days or 1 month in one click and we can set those triggers on the admin side to daily remind the delivery person to deliver the product. With this, we can integrate AI agents to suggest diet plans for the week to provide a healthy diet to our customers.

5.0.1 | Future Scope Suggested by Evaluation Committee

During my presentation, the evaluation committee members suggested several ideas to make the app more trustworthy. All these suggestions should be implemented in the upcoming version

- **KYC of Farmer:** Currently, we are collecting some basic details from the farmer and using that information to determine whether they are practicing organic farming. However, in this domain, farmers are required to certify their farms every year through an organization named APEDA. Since our system does not currently require such certification, we risk losing authenticity in the long run. To make the app trustworthy, we need to implement a proper KYC process for farmers. Without this, further development of the app should not proceed.
- **Focus on One Region:** Currently, we are allowing farmers from different cities to list on the app. However, to better understand the operations and effectiveness of the system, we should first focus on a single region and run the app from one central location. This approach is more practical from a business perspective. To implement this, modifications will be required in both the app and the database.
- **Understanding the logistics side:** After implementing the above two changes, our app will be ready for the next phase. One of the most vital tasks at this stage is to understand the logistics aspect of the model. We need to explore how farmers will deliver their produce to the central collection point, how the delivery will be carried out to the consumers, and how we will ensure that only authentic products reach them. To address these challenges, we must focus on the logistics part.

Chapter 6

Bibliography

1. FarmRise. (n.d.). FarmRise. https://farmrise.bayer.com/

2. KissanAI. (n.d.). https://kissan.ai/

3. Farmer.Chat. (n.d.). https://farmerchat.digitalgreen.org/

4. Tirodkar, A. (2025, March 13). How AI is Transforming Farming in India: The Baramati Experiment. Frontline. https://frontline.thehindu.com/news/ai-farming-microsoft-barabati-m
article69320825.ece

5. Chitransh, P. (2025, March 28). Ahmedabad Entrepreneur builds a Full-Suite CPAAS solutions platform... StartupPedia. https://startuppedia.in/startup-stories/
ahmedabad-entrepreneur-builds-a-full-suite-cpaas-solutions-platform-that-sends-200-m-sms-per-m

6. Home — APEDA. (n.d.). https://apeda.gov.in/

7. Threads. (n.d.). https://www.threads.net/?hl=en

Udgam: A Community-Driven Organic Marketplace

by Aditya Dave

Udgam: A Community-Driven Organic Marketplace

ORIGINALITY REPORT

6%	3%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University College London Student Paper	1%
2	Submitted to University of Canberra Student Paper	1%
3	Submitted to Auckland University of Technology Student Paper	1%
4	Submitted to University of Greenwich Student Paper	1%
5	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
6	www.hnbumuexams.com Internet Source	<1%
7	publications.iowa.gov Internet Source	<1%
8	Submitted to University of Waikato Student Paper	<1%
9	www.pmu.edu.sa Internet Source	<1%
10	Stéphane Leman-Langlois. "Technocrime, Policing and Surveillance", Routledge, 2012 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Aditya Dave

Udgam: A Community-Driven Organic Marketplace

Quick Submit

Quick Submit

Ahmedabad University

Document Details

Submission ID

trn:oid:::1:3225752142

Submission Date

Apr 23, 2025, 10:16 AM GMT+5:30

Download Date

Apr 23, 2025, 10:18 AM GMT+5:30

File Name

BTEP_Report_Daveaditya.pdf

File Size

2.3 MB

22 Pages

3,344 Words

17,928 Characters

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large-language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.