Karshak - Farmers Market Place : Revolutionizing Agriculture Trade For Farmer Prosperity and Good Food for Good Life

Bandi Navaneetha

*Dept. of Computer Science and Applications*

*KLEF*

Guntur, India [2201600151@kluniversity.in](mailto:2201600176@kluniversity.in)

Dr.R.D.Sathiya

*Professor, Dept. of Computer Science and Applications*

*KLEF*

Guntur, India

[sathiya@kluniversity.in](mailto:2201600176@kluniversity.in)

***Abstract—Karshak-Farmers Market Place embodies a transformative vision, blending technological innovation with a steadfast commitment to farmer prosperity and the promotion of good food for a good life. Leveraging the power of Django, HTML, CSS, Bootstrap, JavaScript, and AJAX, this platform seeks to reshape the traditional agricultural marketplace, particularly in the context of India.By establishing a direct avenue for farmers to connect with consumers, Karshak-Market eliminates the reliance on intermediaries like brokers and dealers, ensuring fair prices and maximizing profits for farmers. Through an intuitive user interface and dynamic features driven by JavaScript and AJAX, the platform facilitates seamless access to high-quality, certified organic produce, promoting a healthier lifestyle and sustainable farming practices.***

***Karshak-Market's mission, encapsulated in "Good Food for Good Life," underscores its dedication to enhancing well-being and environmental stewardship. Stringent certification processes uphold the authenticity of organic products, instilling confidence in consumers and driving demand for sustainable agriculture.Furthermore, Karshak-Market serves as a catalyst for empowerment within Indian agricultural communities. By providing farmers with access to valuable resources such as market insights, agricultural best practices, and financial assistance, the platform equips them to make informed decisions and thrive in their endeavors.***

***In summary, Karshak-Market epitomizes the transformative potential of technology in revolutionizing agricultural trade, ensuring prosperity for farmers and enriching lives through access to nutritious, organic food. As India's backbone, agriculture stands as a pivotal sector, and Karshak-Market's innovative approach holds promise in shaping a more sustainable and equitable food system for generations to come.Additionally, recognizing the importance of government support, this paper also introduces the concept of an Android-based mobile application aimed at providing timely and relevant information on various government schemes for farmers across India. By addressing the shortcomings of existing applications, such as outdated information and security risks, this initiative aims to further empower farmers and enhance their access to crucial support mechanisms.***

***Keywords— Model Creation,Django, Python-based Web Framework, Django Library, Web Server, HTML, CSS, AJAX, JavaScript, Bootstrap, Organic Products,* *Direct Sales, Sustainable Agriculture, Farmer-Customer Interaction, Online Marketplace, E-commerce, Fair Trade, Agricultural Innovation.***

# INTRODUCTION

In an era characterized by growing awareness of environmental sustainability and conscious consumerism, the direct exchange of organic products between farmers and consumers emerges as a pivotal solution. This application serves as a catalyst in this evolving landscape, leveraging technological advancements and the comprehensive capabilities of Django to redefine the accessibility and distribution of organic goods.At its heart, this application serves as a vital bridge connecting farmers dedicated to cultivating organic produce with consumers who prioritize health and environmental responsibility. By harnessing Django's robust framework alongside HTML, CSS, AJAX, JavaScript, and Bootstrap, this platform orchestrates seamless transactions within an intuitive online marketplace.

The crux of this application lies in its transformative potential to empower farmers, offering them a direct platform to exhibit and sell their products, thereby circumventing conventional intermediaries. Beyond ensuring equitable remuneration for their labor, this approach nurtures a sustainable agricultural ecosystem by championing organic farming methodologies.For consumers, the platform offers a portal to a diverse array of organic offerings, each embodying the values of ethical production and environmental stewardship. Through an immersive user interface complemented by dynamic functionalities powered by AJAX and JavaScript, customers can navigate, procure, and advocate for sustainable agriculture effortlessly.

In essence, this application transcends mere technological innovation; it embodies a vision of fostering a more equitable and environmentally conscious food system. As we embark on this journey, we embrace the profound potential of technology to forge genuine connections between farmers and consumers, heralding a future where every purchase contributes to a brighter and healthier world for all.

1. **SYSTEM ARCHITECTURE**

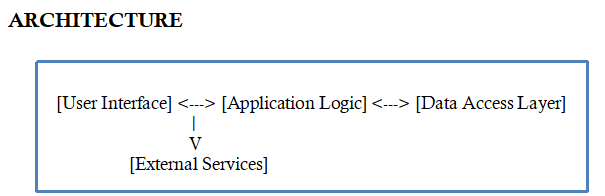
The application is based on Client Server Architecture. The user interacts with app through the UI of the application shown in the figure. It is 3 - tier architecture. The 3 tiers are:

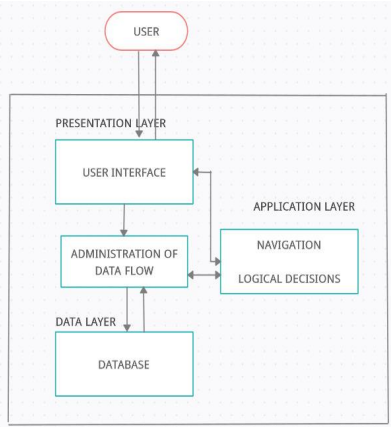
• Presentation Layer

• Business Layer

• Data Layer

The First tier is the UI and communication layer of the application, where the customer interacts with the application. Its main target is to show information to and gather details from the user. The second tier, also known middle layer, is the center of the application. In this tier, details gathered in the presentation layer is processed - sometimes against other details in the 3rd tier using a specific set of business protocols. The third tier or back-end, is where the details is processed by the application is stored and managed. The database used in this application is firebase.





**Fig.1 Karshak System Architecture**

**User Authentication:**

Prompt the user to log in with their credentials (username and password).Verify the user's credentials against the database.

**Main Menu Navigation:**

Once logged in, display the main menu options to the user.

Allow the user to choose from various features like viewing government schemes, accessing market information, etc.

**Access Market Information:**

Allow users to access market information such as crop prices, weather forecasts, etc.Integrate with external APIs or services to fetch real-time market data.

**Product Information:**

Include a section where users can access information about different agricultural products available in the market.Display details such as product name, description, cultivation methods, nutritional value, etc.

**Health Benefits:**

Provide information about the health benefits of various agricultural products.Highlight nutritional content, medicinal properties, and dietary advantages to encourage users to consume these products.

**Update Profile:**

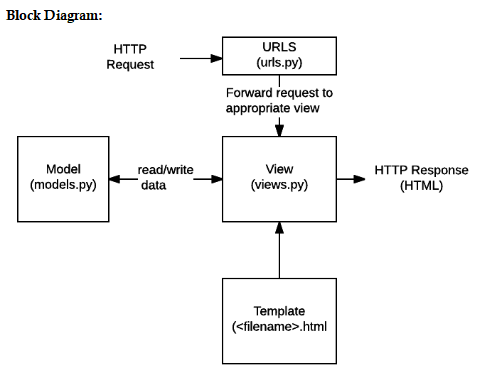
Enable users to update their profile information such as contact details, farm location, etc.Validate and save the updated information to the database.

**Feedback Mechanism:**

Implement a feedback mechanism where users can provide feedback or suggestions.Store feedback in the database for further analysis and improvement of the application.

**Logout:**

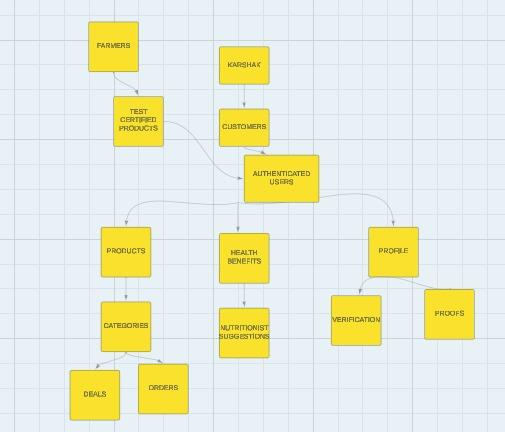
Provide an option for users to log out securely from their accounts.



**Fig.2 Block Diagram of Karshak**

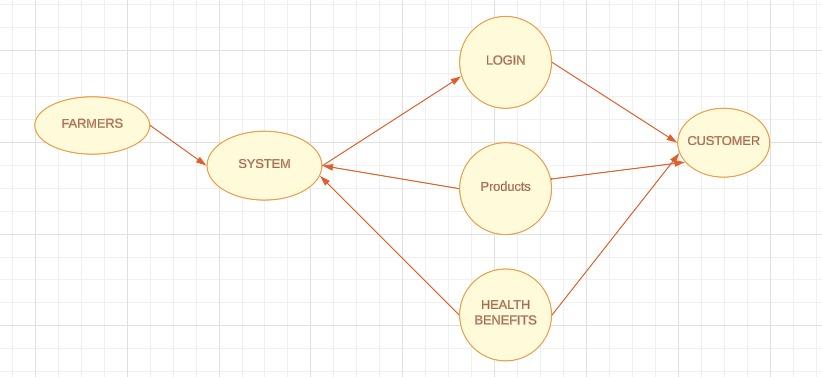
## UML DIAGRAMS:

The UML (Unified Modeling Language) diagram for the "Karshak" application provides a visual representation of its architectural components and their interactions. Here's some context for the UML diagram:

 **Fig.3 UML Diagram**

**USE CASE DIAGRAM:**

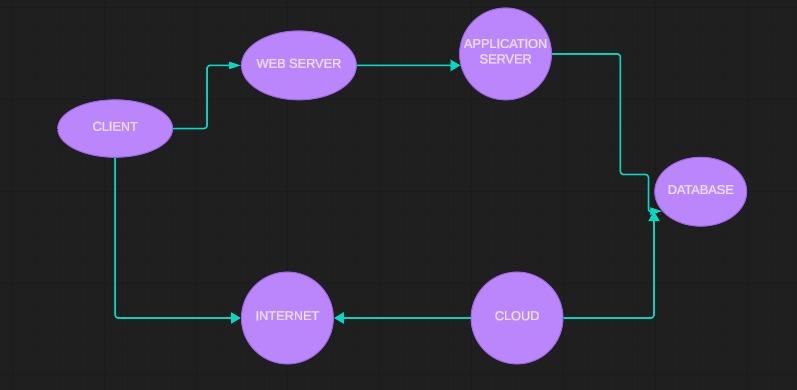
1. A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
2. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
3. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



**Fig.4 Use Case Diagram**

**DEPLOYMENT DIAGRAM:**

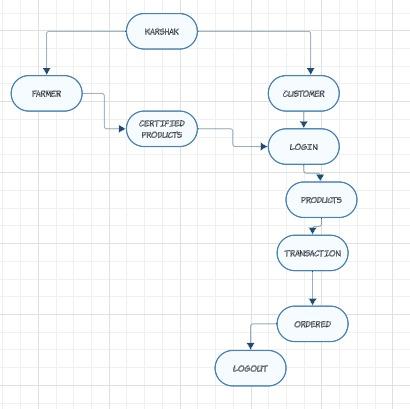
A deployment diagram for the "Karshak" application illustrates the distribution of software components across different nodes or hardware devices in a networked environment.



**Fig.5 Deployment Diagram**

This deployment diagram illustrates how the "Karshak" application is distributed across different nodes and components within a networked environment, enabling farmers and stakeholders to access essential agricultural information and resources seamlessly.

## ACTIVITY DIAGRAM:

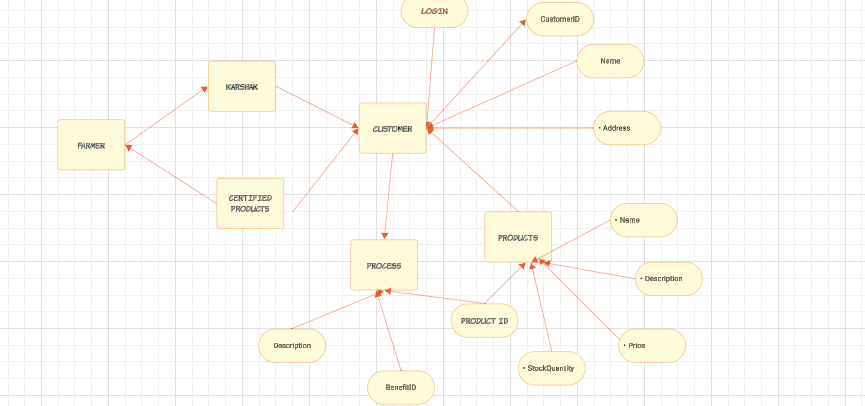
****

**Fig.6 Activity Diagram**

**ER DIAGRAM:**

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of the E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in a database, so by showing relationships among tables and their attributes, ER diagrams show the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.



**Fig.7 ER Diagram**

**Entities:**

Farmers,Customer,Product,Health Benefits.

**Relationships:**

1. Customer - Product (Many-to-Many):
   * Each customer can purchase multiple products.
   * Each product can be purchased by multiple customers.
2. Product - Health Benefits (One-to-Many):
   * Each product can have multiple health benefits.
   * Each health benefit is associated with one product.

**Attributes:**

1. Customer:
   * CustomerID (Primary Key)
   * Name
   * Email
   * Address
2. Product:
   * ProductID (Primary Key)
   * Name
   * Description
   * Price
   * StockQuantity
3. Health Benefits:
   * BenefitID (Primary Key)
   * Description
   * ProductID (Foreign Key)

This simplified ER diagram illustrates the relationships between customers, products, and health benefits in the "Karshak" application, focusing specifically on customer interactions.

# III. LITERATURE SURVEY

To gain a deeper understanding of the project, I perused several articles and web Pages. Here are some of the publications I reviewed, along with their findings.

1. *Related Work*

*In their paper titled "Kisan Soch – A Mobile App for Farmers," Joshi et al. (2021) highlight the importance of agriculture in India and the need for a mobile application to provide farmers with information about government schemes. They address the limitations of existing applications, such as outdated information and language barriers, by developing a multilingual mobile application using React Native and Neural Machine Translation (NMT) techniques.The authors discuss the system architecture, which is based on a Client-Server Architecture with three tiers: Presentation Layer, Business Layer, and Data Layer. They emphasize the use of React Native for cross-platform compatibility and Google Firebase for data storage. The application utilizes NMT with an attention mechanism to translate government scheme information from English to the user's selected language.*

*In the literature survey, the authors explore various research papers on NMT and transformer models to improve translation quality. They discuss the challenges associated with NMT and highlight the effectiveness of transformer models with attention mechanisms in achieving accurate translations.[1]*

1. *Survey of the Existing Applications and websites*
   * VaradhiFarms [2]
   * VamsiFarms [3]
   * Kisan Soch [4]
   * **FarmersMarket** [5]

**Search engines:**

Search engines such as Google Chrome, MicroSoft Edge, Opera and Yahoo.Not all accommodation alternatives may be advertised or easily found using search engines. Some Reasonal Products, Organic Products postings may not appear prominently in search engine results, thereby limiting products options.

1. *SoftwareRequirement*

***Specifications Hardware:***

* + Processor: Intel Core i5
  + Memory: 8GB RAM
  + Hard disk: 1TB

***Software:***

* + *Technology: Python*
  + *OS: Windows10*
  + *IDE: VS CODE*

***Libraries Used:***

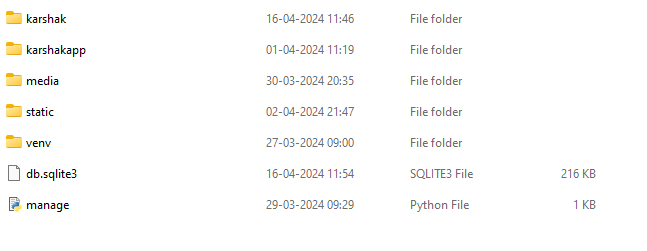
* + Django.contrib.admin
  + Django.urls
  + Django.contrib.settings
  + Django.http.JsonResponse
  + Django.shortcuts
  + Django.views.generic.base.View
  + Django.forms
  + Django.contrib.auth.decorators.login\_required
  + Django.utils.decorators.method\_decorator
  + Django.contrib.auth.views
  + Django.contrib.auth.forms
  + Django.db.models
  + Django.contrib.messages
  + Django.urls.reverse
  + Django.utils.html.format\_html

# IV. METHODOLOGY

1. *Development Environment Setup:*

Utilize Visual Studio Code (VSCODE) as the integrated development environment (IDE) for coding. Set up the development environment with Python 3.6 or higher as the primary programming language.

Install necessary libraries and frameworks, including Django for backend development, and utilize OS library for system-level operations.

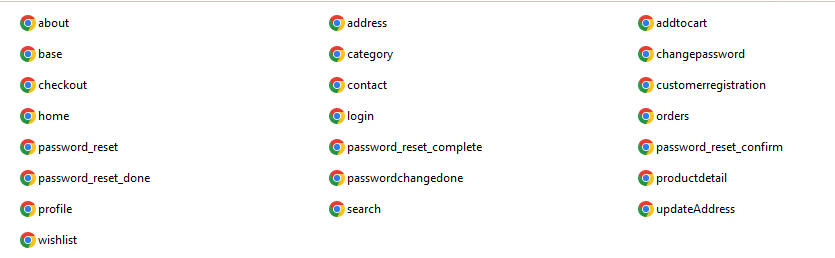


**Fig.8 Development Environment Setup**

1. *Frontend Development*

Utilize HTML, CSS, JavaScript, and Bootstrap for frontend development to create a responsive and visually appealing user interface.

Implement dynamic frontend features using JavaScript to enhance user interaction and experience. Leverage Bootstrap framework for efficient styling and layout of frontend components.



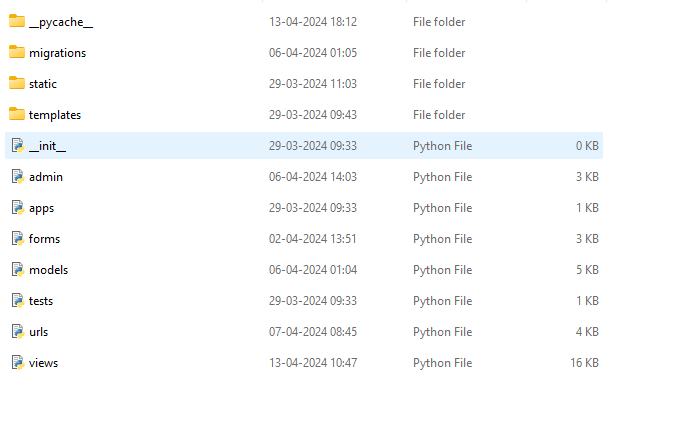
**Fig.9 Frontend Development**

1. *Backend Development:*

Develop the backend logic using Django, as high-level Python web framework, to handle server-side operations and data process.

Utilize Django's built-in features for user authentication, data modeling, and routing to streamline backend development.

Implement AJAX (Asynchronous JavaScript and XML) for asynchronous communication between the frontend and backend, enabling seamless data exchange without page reloads.

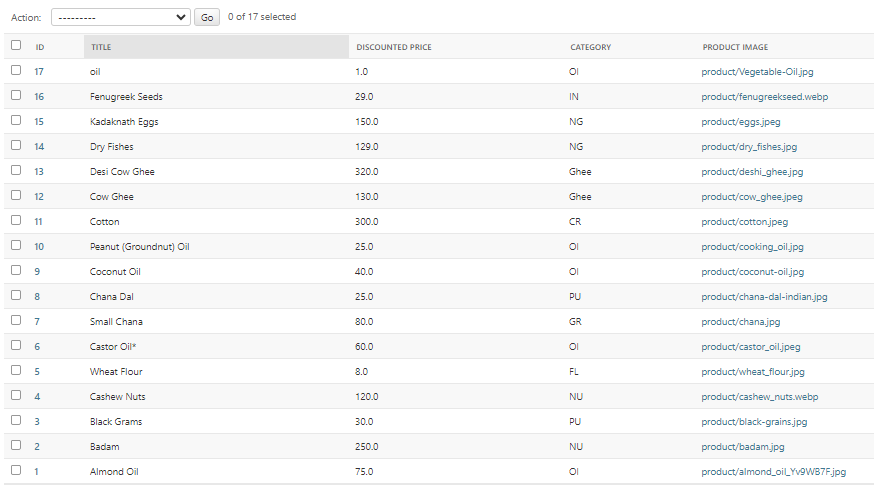


**Fig.10 Backend Development**

1. *Data Collection and Preprocessing:*

Gather relevant datasets containing agricultural information, government schemes, and other pertinent data for the application.

Preprocess the collected data using Python scripts to clean, filter, and organize it for efficient storage and retrieval.



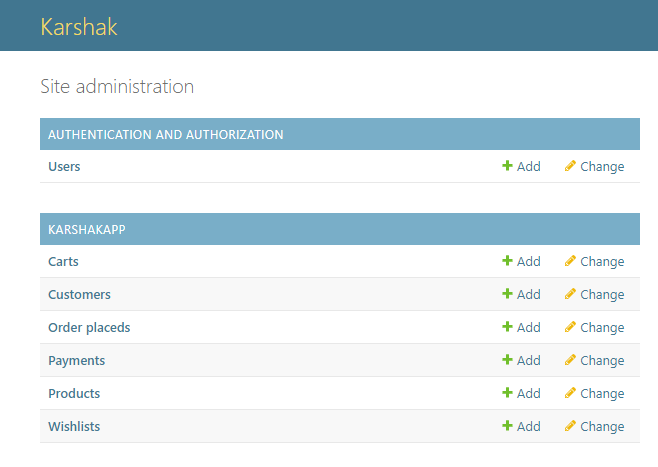
**Fig.11 Data Collection and Preprocessing**

1. *Integration and Deployment:*

Integrate the frontend and backend components to create a cohesive and functional application.

Deploy the application on a hosting platform compatible with Python, such as Python Anywhere, for accessibility and scalability.

Ensure smooth deployment by testing the application thoroughly in the production environment.



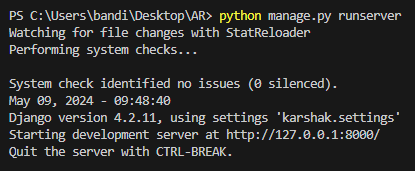
**Fig.12 Integration and Deployment**

1. Testing and Quality Assurance:

Conduct rigorous testing of the application to identify and address any bugs, errors, or usability issues.

Perform user acceptance testing (UAT) to ensure that the application meets the requirements and expectations of its target users.

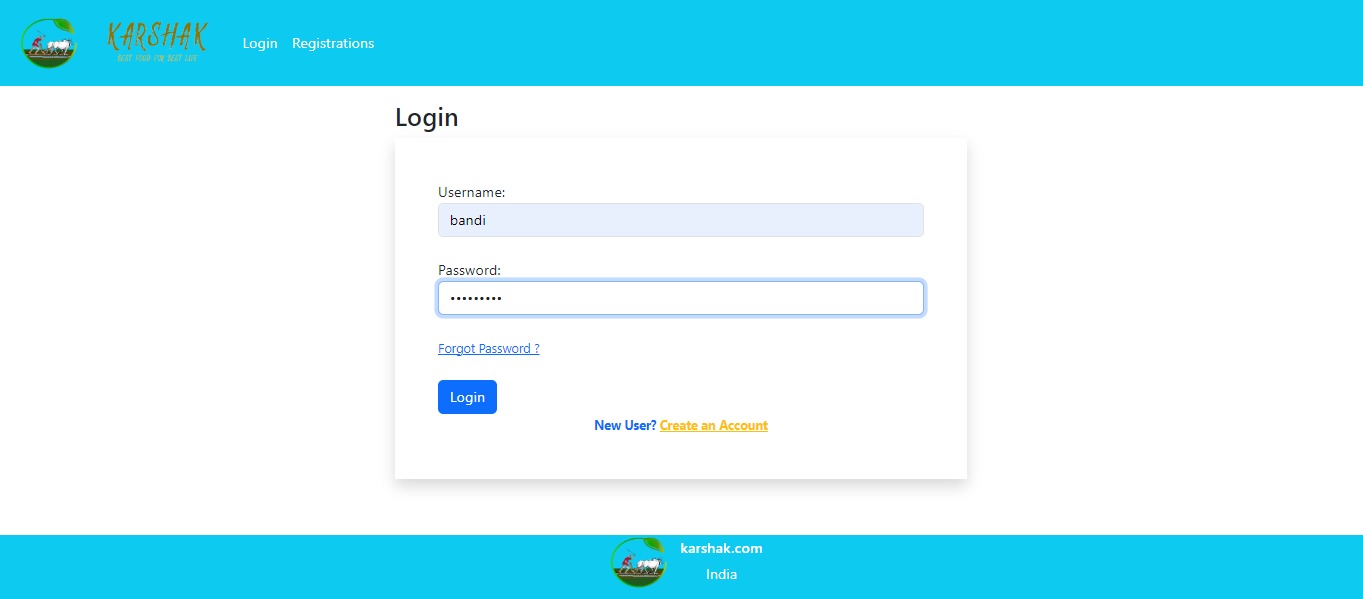
Implement continuous integration and continuous deployment (CI/CD) practices to automate testing and deployment processes for efficiency and reliability.



# Fig.13 Testing and Quality Assurance

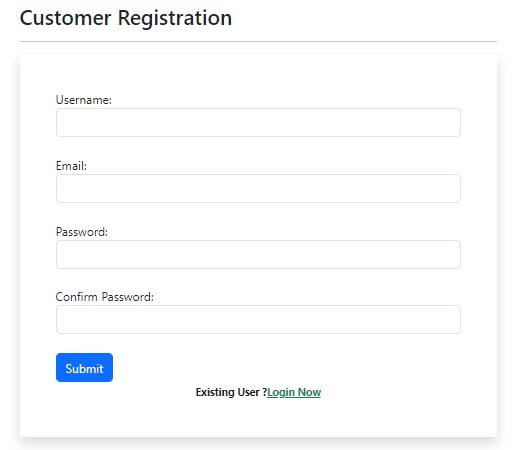
# V. RESULTS

**Home Page:** Users can view the Home

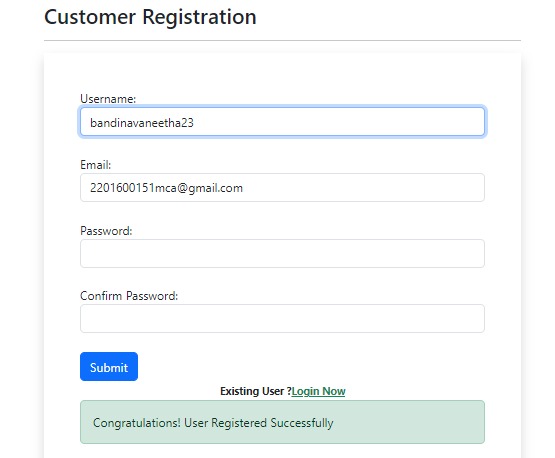


**Fig.14 Home Page(New User)**

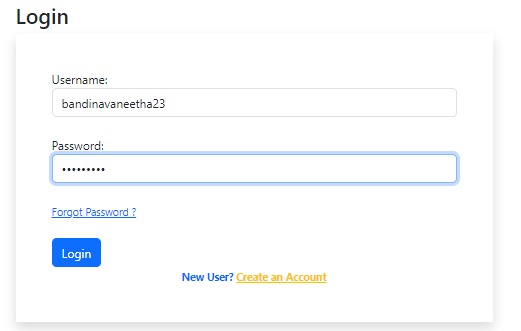
**User Registration page:** User can register with required details.



**Fig.15 Registration Page**



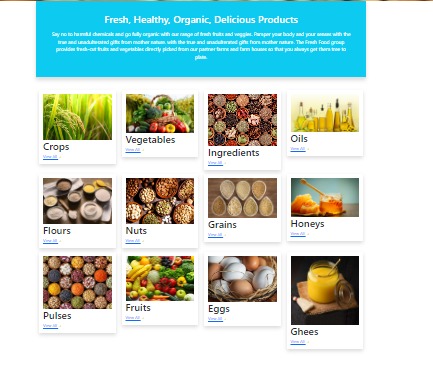
**Fig.16 Successfully Registered user**



**Fig.17 Login Page(Registered User)**

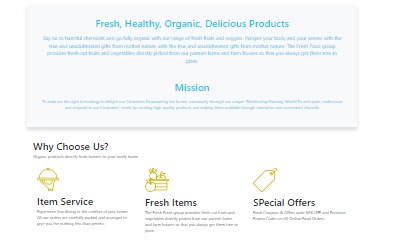
**Home Page:** Authenticated Users can view the Home page.





**Fig.18 Fig.19 Home Page**

**About Us:** This is the small information about project.



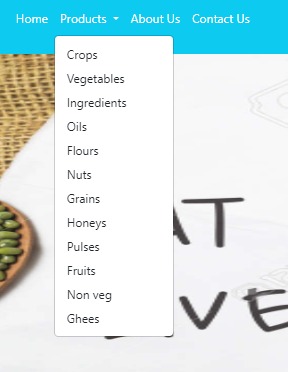
**Fig.20 About Us**

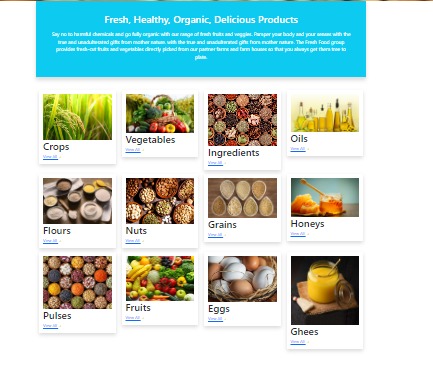
**Contact Us:** This is the small information about project.

****

**Fig.21 Contact Us**

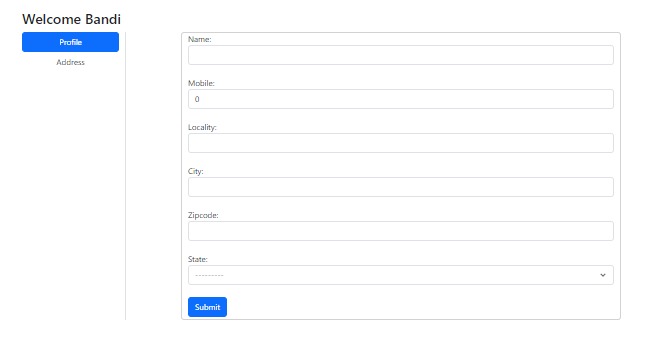
**Products:**

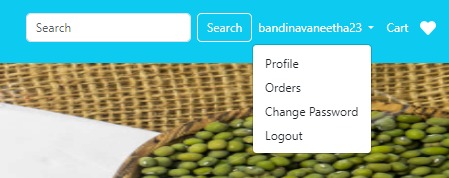
****

****

**Fig.22 Fig.23 Products Page**

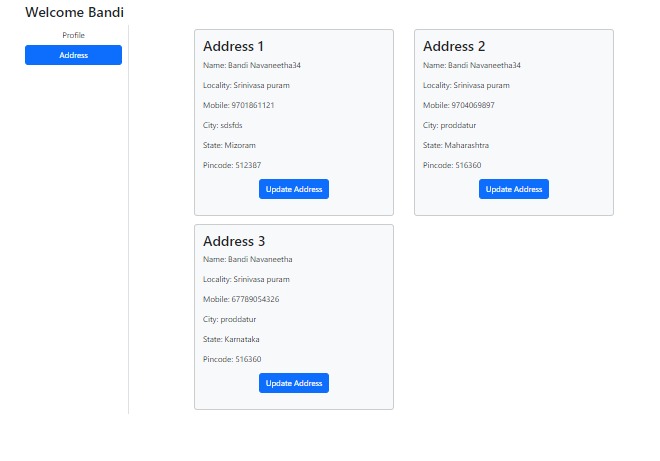
**Profile:**

****

****

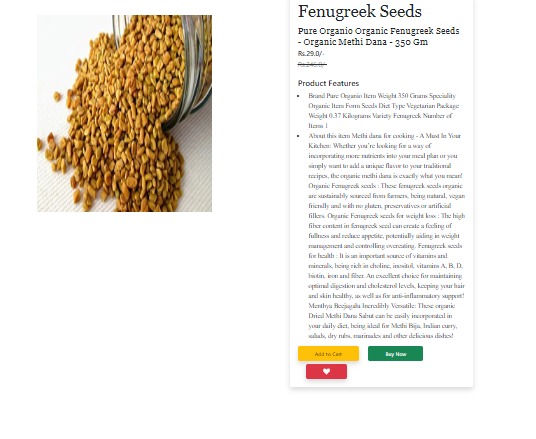
**Fig.24 Fig.25 Profile Page**

**Address Update:**

****

**Fig.26 Address Update Page**

**Add to cart:**

****

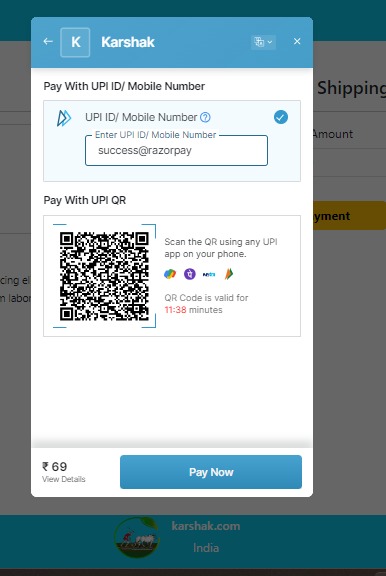
**Fig.27 Add to cart Page**

**Payment Page:**

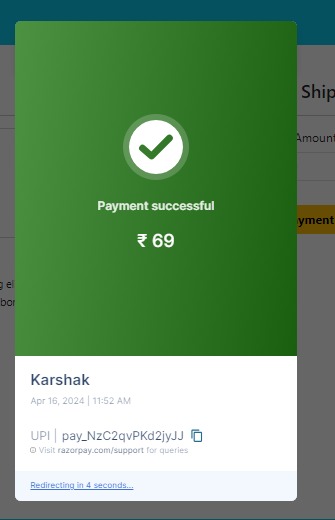
****

**Fig.28 Payment Page**

**Transaction:**

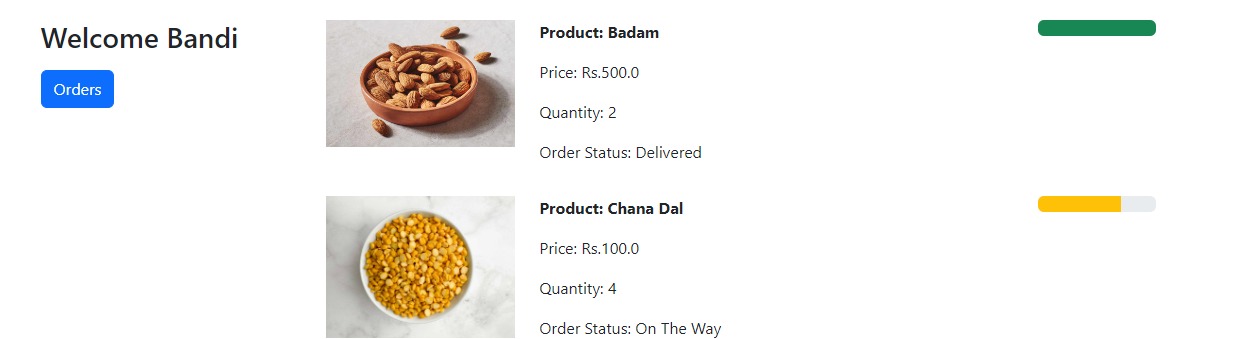
****

**Fig.29 Transaction Page**



**Fig.30 Transaction Successful**

**Order:**

****

**Fig.31 Orders Status Page**

**VI. CONCLUSION**

# In conclusion, our project "Karshak" has achieved its objectives of creating a comprehensive mobile application aimed at empowering farmers and enhancing agricultural practices. Through the utilization of advanced technologies such as React Native, Node JS, and Firebase, we have developed a user-friendly platform that provides valuable resources and support to farmers across India.

# By focusing on features such as government scheme updates, market access tools, and multilingual support, we have addressed key challenges faced by farmers, including access to information and communication barriers. Our application facilitates seamless interaction between farmers, experts, and stakeholders, enabling knowledge exchange, collaboration, and support within the agricultural community.

# Through extensive data collection, preprocessing, and machine learning techniques such as the Transformer model, we have ensured the accuracy and reliability of information provided through our application. By leveraging cloud deployment and multilingual capabilities, we have enhanced accessibility and usability for farmers of diverse backgrounds and regions.

# Overall, the "Karshak" project represents a significant advancement in agricultural technology, with the potential to drive positive change and impact the livelihoods of millions of farmers in India. By promoting sustainable practices, improving market access, and fostering education and awareness, our application contributes to the overall development and prosperity of rural communities. Moving forward, we aim to continue refining and expanding the capabilities of "Karshak" to further empower farmers and support the growth of the agricultural sector.

# FUTURE WORKS

# Native Language

# Farmers Interface

# Providing different regional language options for the application.

# UI enhancement

# Government Schemes(Native Language)

# Adding GPS to automatically predict weather

# Adding schemes from different states

# 

# Fig.32 Government Schemes

# 

# Fig.33 In English

# 

# Fig.34 In Hindi

# 

# Fig.35 In Kannada

# UML DIAGRAM FOR FUTURE WORKS:

# 

# Fig.35 Future UML Diagram

# ACKNOWLEDGEMENT

I am grateful to Prof.D.R.Sathiya Ma’am, who is a professor in the Department of Computer Science and Applications at KL University, for his unwavering support, encouragement, and advice. Finally, without the unwavering encouragement and support from our instructors, friends, and family, I could not have finished this.

# REFERENCES

1. Yuki Kawara, Chenhui Chu and Yuki Arase, “Preordering Encoding on Transformer for Translation,”.

[2] Sahinur Rahman Laskar, Abinash Dutta, Partha Pakray and CSivaji Bandyopadhyay, “Neural Machine Translation: English to Hindi,”.

[3] Sufeng Duan1, Hai Zhao1, Junru Zhou1 and Rui Wang2, “Syntax aware Transformer Encoder for Neural Machine Translation,”.

[4] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, and G. Zhou, “Modeling source syntax for neural machine translation,” in ACL, 2017, pp. 688–697

[5] Kyunghyun Cho, Bart van Merrenboer, Dzmitry Bahdanau and Yoshua Bengio, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,”.

[6] https://reactnative.dev/docs/getting-started.

[7] https://firebase.google.com/.

[8] https://agricoop.nic.in/en.

[9] <https://openweathermap.org/api>.

[10] <https://nfsm.gov.in/nfmis/rpt/calenderreport.asp>

[11] Joshi, N. C., Chayapathi, A. R., Nair, A. A., Ajay, S., & Suhail, M. S. (2021). Kisan Soch – A Mobile App for Farmers. International Journal of Creative Research Thoughts (IJCRT), 9(7), 346-351.

[12] Reference a YouTube video (https://youtu.be/qwFBXuEeg1U?si=xCrfdMj4uFcBYbHe)