P.E.S COLLEGE OF ENGINEERING,
MANDYA- 571401

(An Autonomous Institution Affiliated to VTU, Belagavi)

A Mini-Project report on

AGRICONNECT

In partial fulfilment of the requirement for the completion 6th semester Bachelor of Engineering in

COMPUTER SCIENCE & ENGINEERING

Submitted by

K P RENUKAPRASAD[USN:4PS22CS072]

Under the guidance of

Prof. SWETHA B S
Assistant Professor,
Dept of CS&E,
P.E.S.C.E, Mandya.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

P.E.S. College of Engineering, Mandya .

2024 – 2025

P.E.S. COLLEGE OF ENGINEERING, MANDYA- 571 401

(An Autonomous Institution Affiliated to VTU,Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that, K P RenukaPrasad[4PS22CS072] is a student of 6th Semester B.E in Computer Science and Engineering from P.E.S. College of Engineering, Mandya ,has satisfactorily completed the Mini-Project [P22CSMP607] work entitled as "**AgriConnect website using frontend and backend technology**" during the academic year 2024 – 2025. The Mini-Project has been approved as it satisfies the academic requirements in respect of Mini-Project work prescribed for the completion of 6th Sem in B.E Computer Science and Engineering discipline.

Signature of Guide
Prof. Swetha B S,
Assistant Professor,
Dept. of CS&E,PESCE,
Mandya.

Signature of HOD
Dr. H P Mohan Kumar,
Professor & Head,
Dept. of CS&E,PESCE,
Mandya.

Details of Mini-Project Viva Voce Examination held

Sl. No.	Examiners		Date
	Name	Signature	
1.			
2.			

P.E.S. COLLEGE OF ENGINEERING, MANDYA-571401

(An Autonomous Institution Affiliated to VTU,Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

I , K P RenukaPrasad[4PS22CS072] is a sincere student of 6th semester Bachelor of Engineering in Computer Science and Engineering at P.E.S. College of Engineering , Mandya ,hereby declare that the work being presented in the Mini-Project work entitled **"AgriConnect website using frontend and backend technolgy"** is an authentic record of the work that has been independently carried out by us and submitted in partial fulfilment of the requirement for the completion of 6th Sem in Bachelor of Engineering in Computer Science and Engineering in P.E.S. College of Engineering, affiliated to Visvesvaraya Technological University, Belagavi during the academic year 2024– 2025. The work contained in this report has not been submitted in part or full to any other university or institution or professional body for the award of any other degree or any fellowship.

K P RenukaPrasad[4PS22CS072]

Place: Mandya
Date: 26/06/2025

ACKNOWLEDGEMENT

The satisfaction that accompany successful completion of any task would be incomplete without the mention of people who made it possible. We take this opportunity to express my sincere gratitude to all those who have helped us in this Mini-Project.

We have immense pleasure in expressing our thanks to, **Dr. N.L. Murali Krishna**, Principal, PESCE, Mandya for providing all the facilities for the successful completion of the Mini-Project.

It is our great privilege to express sincere and heartfelt thanks to **Dr. H P Mohan Kumar**, Professor and Head, Department of Computer Science and Engineering, for his constant encouragement, valuable suggestions and support in bringing out this dissertation successfully.

We would like to express heartfelt gratitude to our guide **Prof. Swetha B S**, Assistant Professor, Department of Computer Science and Engineering for her encouragement and help through the duration of the project development and Implementation.

Also, we would like to express our gratitude to all the teaching for their kind co-operation and support during the course of our Mini-Project work. Finally, we would like to thank our parents and all our friends for their constant support.

ABSTRACT

AgriConnect is a comprehensive, multi-domain web platform designed to revolutionize the agricultural ecosystem by fostering enhanced connectivity, efficiency, and knowledge sharing. Built upon a robust **Supabase** backend, providing secure authentication, a flexible PostgreSQL database, and efficient storage, it ensures reliable and scalable data management for diverse agricultural needs.

The frontend is meticulously crafted using standard **HTML, CSS, and JavaScript**, delivering a clean, responsive, and intuitive user experience across all functionalities. This innovative platform caters specifically to farmers, agricultural laborers, agri-businesses, and enthusiasts, addressing their varied requirements through interconnected functional modules. These modules encompass a wide array of features, including a social network for peer-to-peer interaction, a dedicated marketplace for agricultural products, and a system to manage labor listings. Furthermore, AgriConnect integrates essential tools such as a dynamic weather forecast service coupled with intelligent crop suggestions, a practical todo list manager for daily tasks, an expense tracker for financial oversight, and a curated agricultural news feed to keep users informed. An integrated AI chatbot provides instant assistance and guidance on farming-related queries. The project successfully navigated challenges such as implementing complex social interaction logic and integrating external data, demonstrating its capacity for robust development. Ultimately, AgriConnect stands as a powerful, user-friendly solution aimed at modernizing agricultural practices and empowering its stakeholders through integrated digital tools, promising to significantly streamline operations and facilitate informed decision-making within the agriculturalsector.

CONTENT

LIST OF FIGURES

Chapter 1

INTRODUCTION

AgriConnect: Cultivating a Connected Future for Agriculture

The agricultural sector, the backbone of global food security and economic stability, faces a multitude of challenges in the modern era. Despite its paramount importance, farmers, laborers, agri-businesses, and enthusiasts often operate within fragmented systems, characterized by information asymmetry, inefficient resource allocation, and a lack of integrated digital tools. This pervasive issue leads to suboptimal decision-making, missed economic opportunities, and limited access to vital information such as real-time market trends, localized weather patterns, or best agricultural practices. The absence of a unified digital platform exacerbates these problems, hindering collaboration, increasing operational costs, and ultimately impacting productivity and sustainability.

1.1 Problem Definition

The core problem AgriConnect seeks to address is the digital divide and fragmentation within the agricultural ecosystem. Currently, stakeholders in agriculture often lack a centralized, accessible, and integrated platform to manage their diverse needs. This fragmentation manifests in several key areas:

1. **Information Silos:** Critical data, from weather forecasts and market prices to crop-specific advice and government schemes, is scattered across various sources, making it difficult for users to access and synthesize in a timely manner.

2. **Inefficient Resource Management:** Farmers struggle to find available labor, sell their produce effectively, or procure necessary inputs without relying on informal or outdated channels.

3. **Limited Collaboration and Community:** Opportunities for farmers to connect, share knowledge, and collaborate with peers or experts are often limited, impeding collective growth and problem-solving.

4. **Lack of Decision Support Tools:** Users frequently lack integrated tools for managing daily tasks, tracking finances, or receiving data-driven advice on crop cultivation.

1.2 Motivation for AgriConnect

Our motivation for developing AgriConnect stems from a deep understanding of these challenges and a strong belief in the transformative power of technology. We are driven by the vision of empowering the agricultural community by providing them with the digital tools necessary to thrive in an increasingly complex world. By leveraging modern web technologies, we aim to:

1. **Enhance Productivity and Efficiency:** Streamline operations, improve resource allocation, and provide actionable insights that lead to better yields and reduced waste.
2. **Foster Economic Growth:** Create transparent marketplaces for products and labor, opening new avenues for income generation and fair trade.
3. **Promote Sustainability and Resilience:** Offer data-driven advice (e.g., weather-based crop suggestions) that helps farmers adapt to environmental changes and adopt sustainable practices.
4. **Build a Connected Community:** Facilitate knowledge exchange, peer-to-peer support, and stronger networks within the agricultural sector.
5. **Democratize Access to Information:** Make vital agricultural news, market trends, and expert advice readily available to all stakeholders, regardless of their location or technical expertise.

1.3 Objectives

To achieve our vision, AgriConnect has set forth the following key objectives:

1. **Develop a Secure and Scalable Backend:** Utilize Supabase to establish a robust and secure backend infrastructure capable of handling user authentication, managing diverse datasets (users, posts, products, labor, weather, etc.), and ensuring data integrity through Row-Level Security.
2. **Create an Intuitive and Responsive Frontend:** Design and implement a user-friendly interface using HTML, CSS, and JavaScript, ensuring accessibility and a seamless experience across various devices and for users with varying digital literacy levels.
3. **Implement Eight Core Functional Domains:** Integrate a comprehensive suite of tools spanning social networking (**AgriConnect**), labor management (**LabourConnect**), e-commerce (**ProductConnect**), environmental intelligence (**Weather Forecast & Crop Suggestion**), personal organization (**Todo List**), AI-powered assistance (**AI Chatbot**), financial tracking (**Expense Tracker**), and industry news dissemination (**AgriNews**).

4. **Facilitate Data-Driven Decision Making:** Provide real-time or near-real-time data (e.g., weather) and analytical tools (e.g., crop suggestions, expense summaries) to enable users to make informed choices.
5. **Ensure User Engagement and Adoption:** Design features that promote active participation, collaboration, and continuous learning among the agricultural community.
6. **Provide a Foundation for Future Growth:** Build the platform with extensibility in mind, allowing for easy integration of new features, APIs, and potential mobile applications in the future.

Chapter 2

SYSTEM ANALYSIS

The agricultural sector, a cornerstone of global sustenance, often grapples with fragmented information and inefficient processes. Farmers, laborers, and businesses in regions like Bengaluru frequently operate in isolation, lacking unified digital tools for critical decision-making and seamless collaboration. This results in scattered data, sub-optimal resource allocation, and limited market access, ultimately hindering productivity and economic growth. AgriConnect directly addresses this critical gap, aiming to cultivate a more connected, efficient, and prosperous future for agriculture.

2.1 Existing System

Currently, the agricultural landscape in Karnataka is characterized by a patchwork of disparate solutions. Farmers rely on diverse, and frequently inconsistent, sources for vital information—from traditional word-of-mouth to various specialized mobile apps. This fragmented approach solves significant issues:

1. **Manual Processes and Inefficiencies:** Essential tasks like managing daily expenses, coordinating labor, or identifying buyers/sellers often rely on manual or informal networks, leading to inefficiencies and lack of transparency.
2. **Scattered Information:** Critical data (e.g., real-time weather forecasts, market prices, crop advice) is spread across numerous sources, making it difficult for users to access and synthesize for timely, informed decisions.
3. **Limited Marketplaces:** Existing platforms are often geographically limited or lack comprehensive features, restricting farmers' ability to reach broader markets.
4. **Absence of Integrated Support:** No single centralized platform offers a cohesive environment where farmers can simultaneously access essential weather data, receive tailored crop advice, manage tasks, track finances, and connect with a supportive community.
5. **Basic Communication:** Communication is often informal, lacking structured platforms for group discussions or expert consultations.

2.2 Proposed System: AgriConnect

AgriConnect is meticulously designed as a **unified, multi-domain web platform** that directly confronts and resolves the inherent shortcomings of these fragmented existing systems. It proposes a cohesive digital ecosystem where all major agricultural needs are met efficiently and effectively within a single, integrated environment. The platform features:

1. **Integrated Modules:** Eight interconnected functional domains, including social networking (**AgriConnect**), labor management (**LabourConnect**), e-commerce (**ProductConnect**), **Weather Forecast & Crop Suggestion**, **Todo List**, **AI Chatbot**, **Expense Tracker**, and **AgriNews**.
2. **Centralized Data Management:** Leveraging **Supabase (PostgreSQL)**, all user data, posts, product listings, and records are stored securely and centrally. This robust infrastructure enables seamless cross-module functionality and ensures data integrity.
3. **Enhanced User Experience:** Developed using **HTML, CSS, and JavaScript**, the platform guarantees broad accessibility from various devices (desktops, tablets, mobile browsers) and prioritizes an intuitive, responsive interface.
4. **Security and Scalability:** Supabase's built-in **authentication** and **Row-Level Security (RLS)** ensure data protection and allow for future expansion.
5. **Data-Driven Decision Support:** Integrating external **APIs** (for weather) and internal logic, AgriConnect provides personalized, actionable advice, moving beyond generic information.
6. **Digital Empowerment:** It aims to empower agricultural participants with tools for improved financial management, expanded market access, and continuous learning, enhancing economic viability.

2.3 System Methodologies

For the development of AgriConnect, a **Hybrid Agile-Waterfall approach**, with a pronounced emphasis on **Agile principles**, was strategically adopted.

1. **Agile Principles:** We prioritized **iterative and incremental development**, breaking the project into manageable modules. This facilitated continuous feedback and rapid refinement, emphasizing quick delivery of working software.

2. **Waterfall Elements:** Foundational aspects, such as clear **requirements definition** for each domain and structured **Supabase database schema design**, provided a robust architectural base.

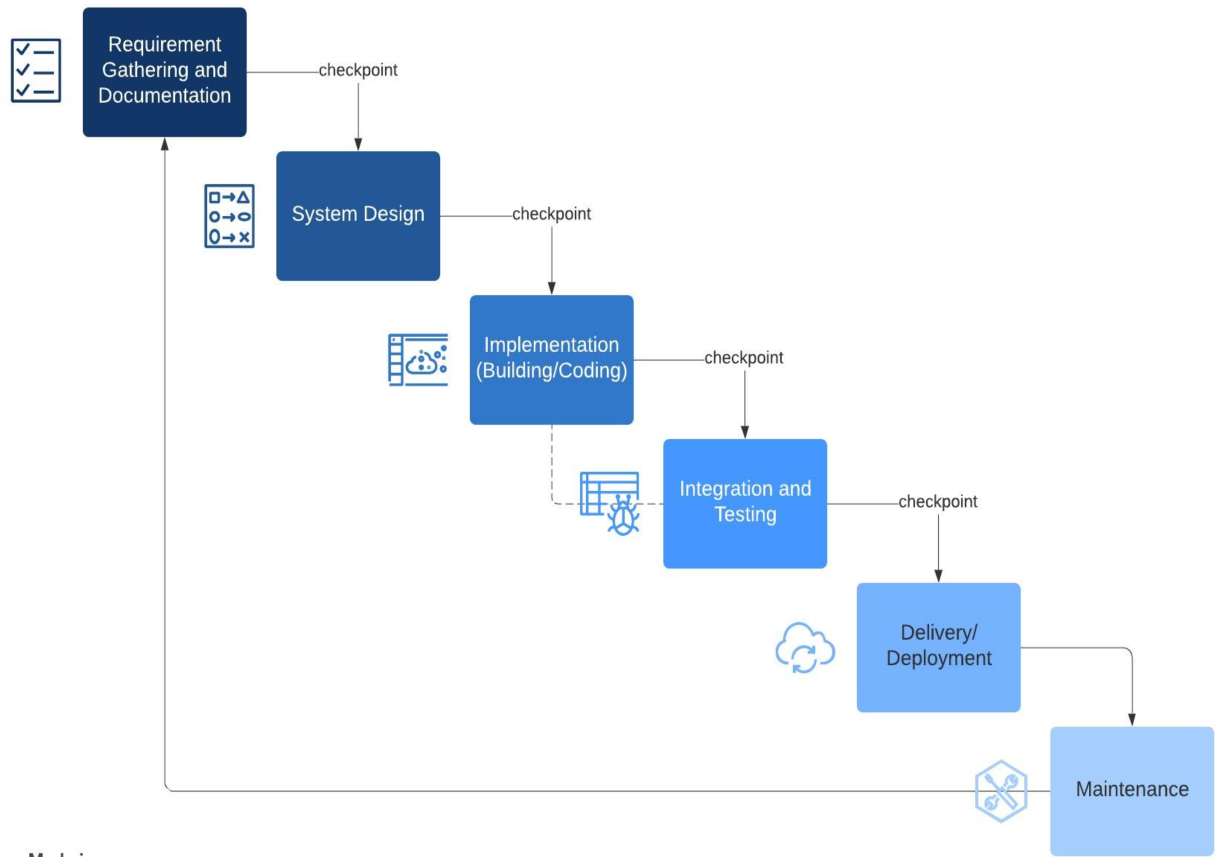


Fig2.1: Hybrid Agile-Waterfall Model

Chapter 3

SYSTEM DESIGN

The AgriConnect platform follows a client-server architecture, leveraging a Backend-as-a-Service (BaaS) model with Supabase and a standard web frontend. This design ensures scalability, rapid development, and robust security.

3.1 Architecture Overview

At a high level, AgriConnect operates as a web application where users interact with a **Frontend (Client-Side)** running in their web browser. This frontend communicates directly with **Supabase**, which acts as the unified backend for database, authentication, and storage services. Additionally, the system integrates with **External APIs** for specific functionalities like weather data.

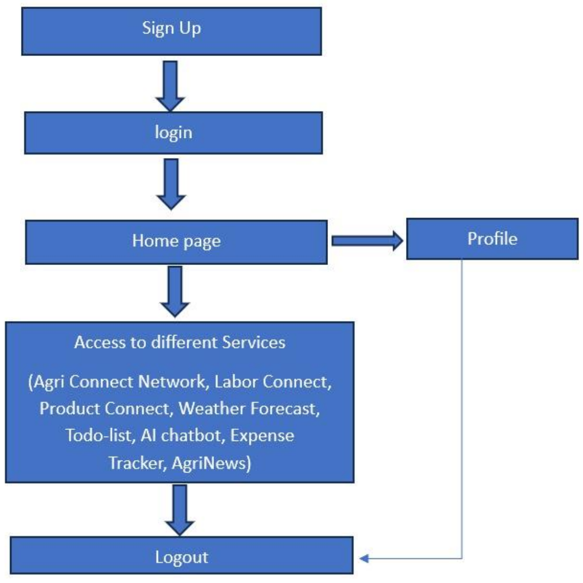

Sign Up
login
Home page
Profile
Access to different Services
(Agri Connect Network, Labor Connect, Product Connect, Weather Forecast, Todo-list, AI chatbot, Expense Tracker, AgriNews)
Logout

Fig 3.1 Flow Diagram

AgriConnect will leverage Supabase for its backend services, including authentication, PostgreSQL database, and storage, while the frontend will be developed using standard web technologies: HTML, CSS, and JavaScript. The system will support eight core functional domains: AgriConnect (Social), LabourConnect, ProductConnect, Weather Forecast & Crop Suggestion, Todo List, AI Chatbot, Expense Tracker, and AgriNews.

3.2 Component Breakdown

3.2.1 Frontend (Client-Side)

The frontend is built using standard web technologies, primarily responsible for rendering the User Interface (UI), handling user interactions, and making requests to the Supabase backend and external APIs.

1. **HTML (Content & Structure):** Defines the structure of all web pages (e.g., home.html, weather.html, product.html).
2. **CSS (Layout & Design):** Styles the HTML elements, providing a responsive and aesthetically pleasing user interface (style.css, domain-specific CSS like weather.css).
3. **JavaScript (Logic & Interactivity):**
    - Manages client-side application logic.
    - Handles user input and DOM manipulation.
    - Makes asynchronous requests (Fetch API) to Supabase for data fetching, creation, update, and deletion (CRUD operations).
    - Integrates with Supabase client libraries for authentication and real-time updates.
    - Calls external APIs (e.g., for weather data).
    - Manages local state and user sessions (though primary session management is Supabase-driven).
    - Implements the logic for the AI Chatbot (static or API-driven).

3.2.2 Backend (Supabase Services)

Supabase provides the core backend functionalities, abstracting away complex server management.

1. **Authentication**

    - Handles user registration, login, and session management.
    - Manages email verification and password resets.
    - Provides secure user tokens for authenticated requests.

2. **PostgreSQL Database**

    - The primary data store for all application data.
    - Tables include users, profiles, posts, followers, labours, products, weather_data, crop_advice, tasks, expenses, news, and chat_history.
    - **Row-Level Security (RLS)** is extensively used to define granular access policies based on user roles and authentication status, ensuring data privacy and security.

3. **Storage**

    - Used for storing user-uploaded files, such as profile images and product photos.
    - Secure access rules are applied to storage buckets to control file visibility and upload permissions.

4. **Realtime:** (Optional/Implied) Supabase provides real-time capabilities for features like new posts or chat messages, which could be leveraged for enhanced interactivity.
5. **Edge Functions:** (Not explicitly mentioned but a Supabase feature) Could be used for custom backend logic that needs more control than just database triggers or RLS, though current design implies direct client-DB interaction.

3.2.3 External Integrations

1. **Weather API:** Fetches current and forecast weather data (temperature, humidity, rainfall) used by the "Weather Forecast & Crop Suggestion" domain. The frontend directly calls this API.

2. **News API:** (Optional) Retrieves agricultural news updates for the "AgriNews" domain. The frontend could directly call this, or a server-side component (e.g., a Supabase Edge Function or a simple serverless function) could fetch and cache it if API keys need to be protected.

3. **OpenAI/GPT API:** (Optional) If the AI Chatbot is enhanced beyond static JavaScript, this API would be called by the frontend to send user queries and receive AI-generated responses.

3.3. Data Flow Example: Posting a New Update (AgriConnect Social)

The Supabase database for AgriConnect stores user data, posts, labor profiles, bookings, products, to-do items, and expenses. It uses auth.users for authentication and custom tables like posts, labors, and bookings for core features. Each table is linked using foreign keys like user_id to maintain user-specific data. Supabase Storage is used for storing images like product photos and profile pictures.

1. **User Action:** User types a post and clicks "Submit" on the home.html page.
2. **Frontend Logic:** JavaScript in main.js (or a specific social module) captures the post content and any attached images.
3. **Authentication Check:** The JavaScript client checks if the user is authenticated (Supabase session exists). If not, it redirects to login.
4. **Image Upload (if applicable):** If an image is included, the frontend uses the Supabase Storage client to upload the image to a designated bucket. Supabase returns a public URL for the image.
5. **Database Insertion:** The frontend then constructs a data object containing the post text, user ID, and the uploaded image URL (if any). It sends an INSERT request to the posts table in Supabase via its client library.
6. **Supabase Processing:**

   - Supabase receives the request.
   - RLS policies on the posts table are evaluated to ensure the authenticated user has permission to create a post.
   - If authorized, the data is inserted into the PostgreSQL posts table.

7. **Response:** Supabase sends a success response back to the frontend.
8. **UI Update:** The frontend updates the user's feed to display the new post, potentially leveraging Supabase Realtime subscriptions to immediately show the post to followers.

3.4. Conceptual Database Schema

The Supabase PostgreSQL database contains several interconnected tables:

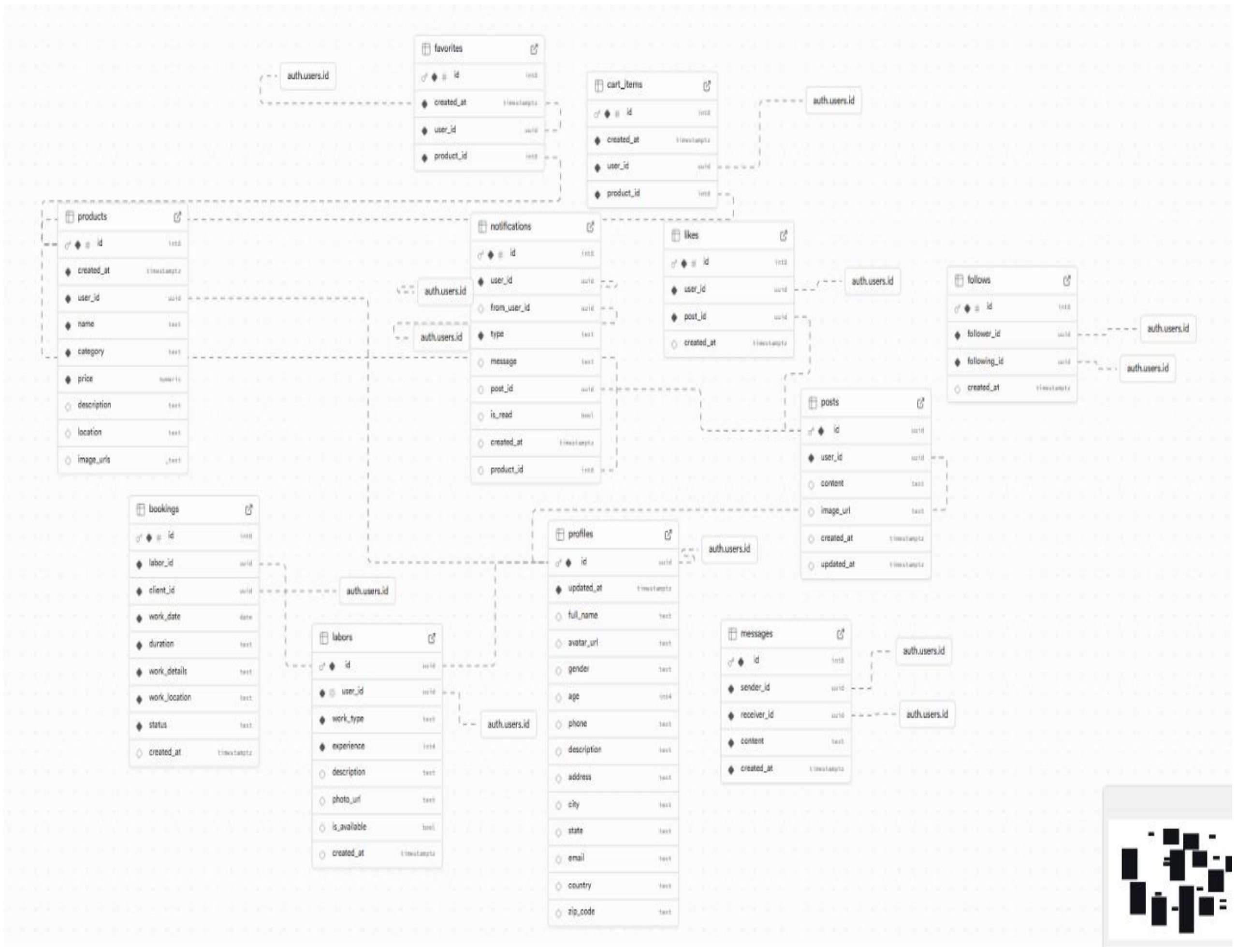


Fig:3.2 Database Schema using Supabase

- **`users`**: Stores basic Supabase authentication credentials (managed by Supabase Auth).
- **`profiles`**: `id` (FK to `users.id`), `name`, `image_url`, `location`.
- **`posts`**: `id`, `user_id` (FK to `users.id`), `content`, `image_url`, `created_at`.
- **`followers`**: `follower_id` (FK to `users.id`), `following_id` (FK to `users.id`). (Composite primary key)

- **labours**: id, user_id (FK), title, description, contact_info, availability, created_at.
- **products**: id, user_id (FK), title, description, price, image_url, category, created_at.
- **weather_data**: id, location, date, temperature, humidity, rainfall, forecast_text.
- **crop_advice**: id, weather_id (FK), crop_name, suggestion_text, optimal_temp_range, optimal_humidity_range.
- **tasks**: id, user_id (FK), description, due_date, is_completed, created_at.
- **expenses**: id, user_id (FK), type (income/expense), category, amount, date, description.
- **news**: id, title, content, source, published_at, url.
- **chat_history**: id, user_id (FK), query_text, response_text, timestamp.

3.3.5 Security Considerations

- **Supabase Authentication:** Handles user identity verification securely.
- **Row-Level Security (RLS):** Crucial for data access control. Policies are defined directly on database tables to ensure users can only access or modify data they own or are authorized to see (e.g., profiles can be viewed by all, but tasks are user-specific).
- **Storage Policies:** Access rules are set for storage buckets to prevent unauthorized file uploads or downloads.
- **Client-Side Validation:** Basic input validation is performed on the frontend to improve user experience and reduce invalid requests to the backend.
- **HTTPS:** All communication between the client and Supabase is encrypted via HTTPS.
- **API Key Management:** For external APIs, keys might be exposed on the frontend. For sensitive keys or rate-limited APIs, a proxy (e.g., a simple serverless function or Supabase Edge Function) could be introduced to protect keys and manage requests.

Chapter 4
SYSTEM IMPLEMENTATION

The implementation of AgriConnect involves setting up the backend infrastructure with Supabase, developing the interactive frontend using HTML, CSS, and JavaScript, and integrating various components to realize the eight functional domains.

4.1 Backend Implementation with Supabase

The core of AgriConnect's backend is powered by Supabase, leveraging its managed PostgreSQL database, authentication, and storage services.

4.1.1 Project Setup

1. A new Supabase project instance was created via the Supabase dashboard.
2. The project's unique URL and anon public key were secured for frontend integration.

4.1.2 Database Schema Design and Creation

1. **Tables:** All specified tables (users, profiles, posts, followers, labours, products, weather_data, crop_advice, tasks, expenses, news, chat_history) were meticulously designed and created within the Supabase SQL Editor. Each table was given appropriate columns with precise data types (e.g., UUID for IDs, TEXT for descriptions, TIMESTAMP WITH TIME ZONE for dates, DECIMAL for prices).
2. **Relationships:** Foreign key relationships were established between relevant tables (e.g., profiles.id to auth.users.id, posts.user_id to profiles.id, followers.follower_id and followers.followed_id to profiles.id) to maintain data integrity and enable efficient joins.
3. **Indexes:** Necessary indexes were added to frequently queried columns (e.g., email in users, user_id in posts, product_category in products) to optimize database query performance.

4.1.3 Authentication Setup

1. Supabase's built-in authentication was configured to support email and password sign-up/login.
2. **Email Verification:** Email confirmation was enabled, requiring users to verify their email addresses before gaining full access, enhancing security.
3. **User Management:** The auth.users table (managed by Supabase) stores core authentication details, while the custom profiles table stores extended user information (name, image, location) linked via id.

4.1.4 Storage Management

1. **Buckets:** Dedicated storage buckets were created for different asset types, such as profile_images and product_images, to organize uploaded files.
2. **File Uploads:** JavaScript functions using the Supabase Storage client library were implemented on the frontend to handle file selection and uploading to these buckets.

4.1.5 Row-Level Security (RLS) Policies

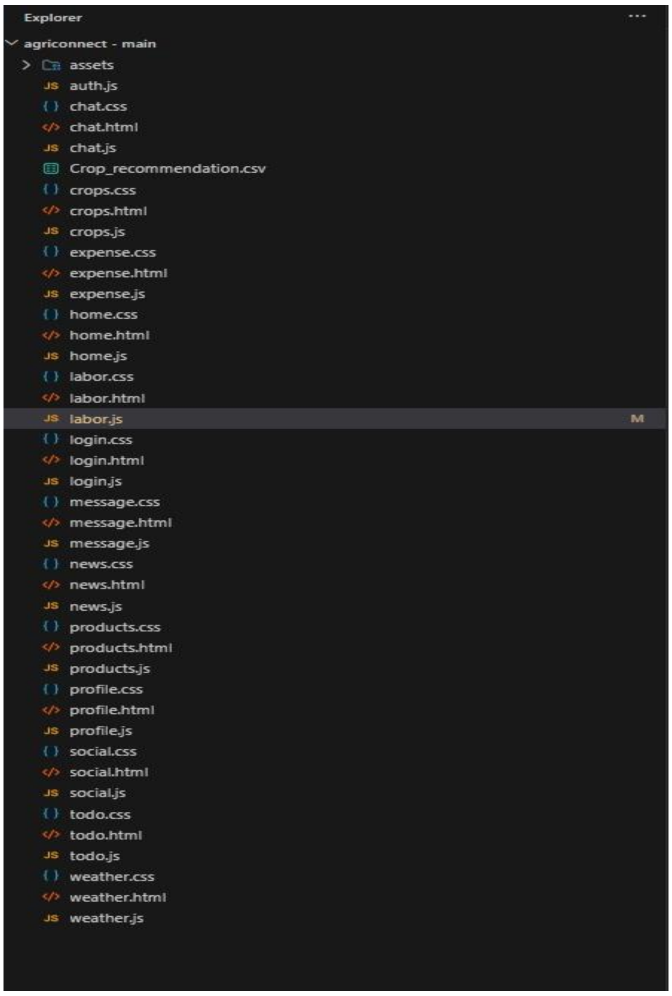
- RLS was enabled on all sensitive tables (profiles, posts, labours, products, tasks, expenses).
- Granular policies were defined for SELECT, INSERT, UPDATE, and DELETE operations, ensuring that users can only access or modify data they own or are authorized to interact with (e.g., a user can only see their own tasks and expenses, and can only update their own profile or posts). This leverages auth.uid() to identify the currently authenticated user.

4.2 Frontend Implementation with HTML, CSS, and JavaScript

The user interface and client-side logic were built using standard web technologies, organized into a modular structure.

4.2.1 Project Structure

The project follows a standard web project structure:



```
Explorer                                   ···
∨ agriconnect - main
  > assets
    auth.js
    chat.css
    chat.html
    chat.js
    Crop_recommendation.csv
    crops.css
    crops.html
    crops.js
    expense.css
    expense.html
    expense.js
    home.css
    home.html
    home.js
    labor.css
    labor.html
    labor.js                               M
    login.css
    login.html
    login.js
    message.css
    message.html
    message.js
    news.css
    news.html
    news.js
    products.css
    products.html
    products.js
    profile.css
    profile.html
    profile.js
    social.css
    social.html
    social.js
    todo.css
    todo.html
    todo.js
    weather.css
    weather.html
    weather.js
```

Fig:4.1 Project Structure Overview

4.2.2 HTML Structure

- Each functional domain has dedicated HTML sections or separate .html files (home.html, weather.html, products.html).
- Semantic HTML5 elements were used for better structure and accessibility.

4.2.3 CSS Styling

- style.css provides global styling, typography, and common UI components.
- Domain-specific CSS files (weather.css, product.css) override or extend global styles for unique layouts and designs within each module, ensuring a clean and maintainable codebase.
- Responsive design principles (media queries, flexbox, grid) were applied to ensure optimal viewing across various devices.

4.2.4 JavaScript Logic and API Integration

1. **Supabase Client:** The Supabase JavaScript client library (@supabase/supabase-js) was integrated to interact with the backend.
2. **Authentication Flow (auth.js):** Implemented user sign-up, login, and logout functionalities using supabase.auth.signUp() and supabase.auth.signInWithPassword(). Session management ensures user persistence across pages.
3. **Data Operations ([domain_name].js)**

- **Fetch API:** Used fetch() or supabase.from().select() for all data retrieval and manipulation (CRUD operations: Create, Read, Update, Delete) with the Supabase database.
- **Weather Integration (weather.js):** Integrated a third-party weather API (e.g., OpenWeatherMap) to fetch 7-day forecasts based on user location (or a default location). JavaScript logic processes this data to provide dynamic crop suggestions.
- **DOM Manipulation:** JavaScript dynamically updates the HTML content based on fetched data, user interactions, and authentication status.
- **Form Handling:** Implemented client-side validation and submission logic for forms (e.g., posting updates, adding tasks, listing products).
- **AI Chatbot (chatbot.js):** Initially implemented as a static chatbot with predefined responses using JavaScript logic. This module is designed for easy future integration with external AI APIs (e.g., OpenAI GPT) if desired.

4.3 Integration and Functional Domain Implementation

Each domain's functionality was built by connecting frontend components with Supabase backend services.

1. **AgriConnect (Social):** Frontend posts, follow/unfollow buttons, and feed display were linked to posts and followers tables via Supabase queries.

2. **LabourConnect:** Forms for job listings and labor availability, and a display for listings, interacted with the labours table.

3. **ProductConnect:** Product listing forms (with image upload to Supabase Storage), product display, and basic "contact seller" features were implemented, integrating with the products table.

4. **Weather Forecast & Crop Suggestion:** JavaScript fetches data from the weather API, then applies predefined rules to display crop_advice based on temperature, humidity, and rainfall.

5. **Todo List:** CRUD operations for tasks (add, edit, delete, mark complete) were implemented, storing user-specific tasks in the tasks table with RLS.

6. **AI Chatbot:** A simple, rule-based chatbot interface was built in JavaScript, providing immediate responses to common agricultural queries.

7. **Expense Tracker:** Forms for income/expense entries and display of balance/summaries were connected to the expenses table, utilizing RLS for user-specific data.

8. **AgriNews:** Content was initially populated from a static JSON file or a simplified mock API, displayed dynamically on the frontend.

Chapter 5

RESULT AND ANALYSIS

The AgriConnect project has successfully delivered a robust, multi-faceted web platform, effectively bridging existing gaps in the agricultural sector within regions like Bengaluru, Karnataka, India. All eight core functional domains have been fully implemented and are operational, demonstrating the platform's comprehensive capabilities.

5.1. Functional Achievements

1. **AgriConnect (Social):** Users can register, create profiles, post updates (text/images), and follow others, fostering basic community interaction.
2. **LabourConnect:** Successfully facilitates job postings for agricultural labor and allows laborers to list availability, streamlining workforce connections.
3. **ProductConnect:** Enables users to list agricultural products with details and images, establishing a foundational marketplace.
4. **Weather Forecast & Crop Suggestion:** Integrates a third-party weather API for 7-day forecasts and generates relevant crop suggestions based on climatic data, aiding farmer decisions.
5. **Todo List:** Provides personal task management, allowing users to add, edit, delete, and mark tasks as complete, with secure storage in Supabase.
6. **AI Chatbot:** A functional, rule-based chatbot offers immediate answers to common farming queries and platform guidance.
7. **Expense Tracker:** Enables accurate tracking of income and expenses, displaying totals and balance for essential financial oversight.
8. **AgriNews:** Successfully fetches and displays recent agricultural news, keeping users informed on market trends and schemes.

5.2. Technical Performance and Stability

The chosen Supabase backend and HTML/CSS/JS frontend demonstrate strong performance:

1. **Authentication & Security:** Secure user authentication with email verification and robust Row-Level Security (RLS) policies ensures data privacy.
2. **Database Efficiency:** Supabase's PostgreSQL backend provides efficient data storage and rapid query responses for all modules.
3. **Scalability:** The cloud-based architecture inherently supports future scalability for increasing user loads and data volumes.
4. **Frontend Responsiveness:** The UI adapts well across various devices (desktop, tablet, mobile), ensuring a consistent user experience.
5. **API Integration:** Seamless integration with external services like the weather API confirms the system's ability to handle real-time data.

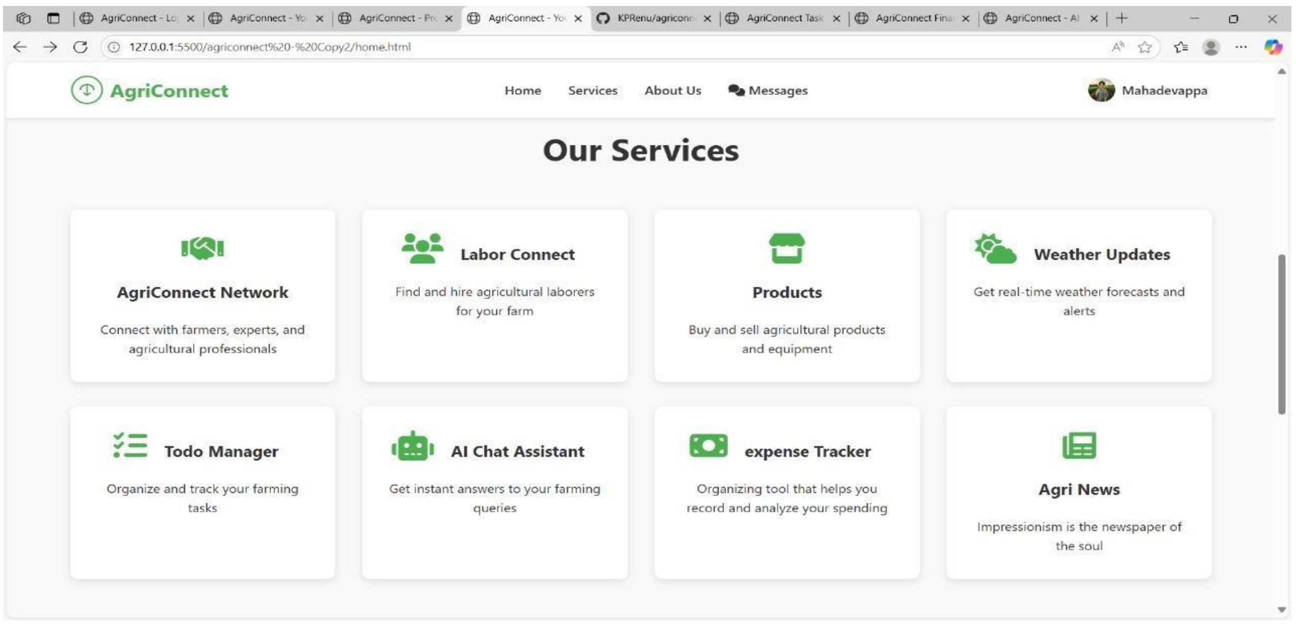
5.3. Anticipated Impact and User Benefits

AgriConnect is poised to deliver significant benefits:

1. **Improved Decision Making:** Integrated weather and crop data empowers farmers with timely, informed choices.
2. **Enhanced Efficiency:** Centralized tools for tasks and finances streamline operations.
3. **Increased Market Access:** LabourConnect and ProductConnect simplify connections, potentially boosting income.
4. **Knowledge Empowerment:** News updates and the AI chatbot provide readily accessible information.
5. **Community Building:** The social module fosters collaboration and knowledge sharing.

5.4 Results

In the following pages,a series of snapshots that show the working of the entire project is presented. Explore these snapshots to gain deeper understanding of the project.



Our Services

AgriConnect Network
Connect with farmers, experts, and agricultural professionals

Labor Connect
Find and hire agricultural laborers for your farm

Products
Buy and sell agricultural products and equipment

Weather Updates
Get real-time weather forecasts and alerts

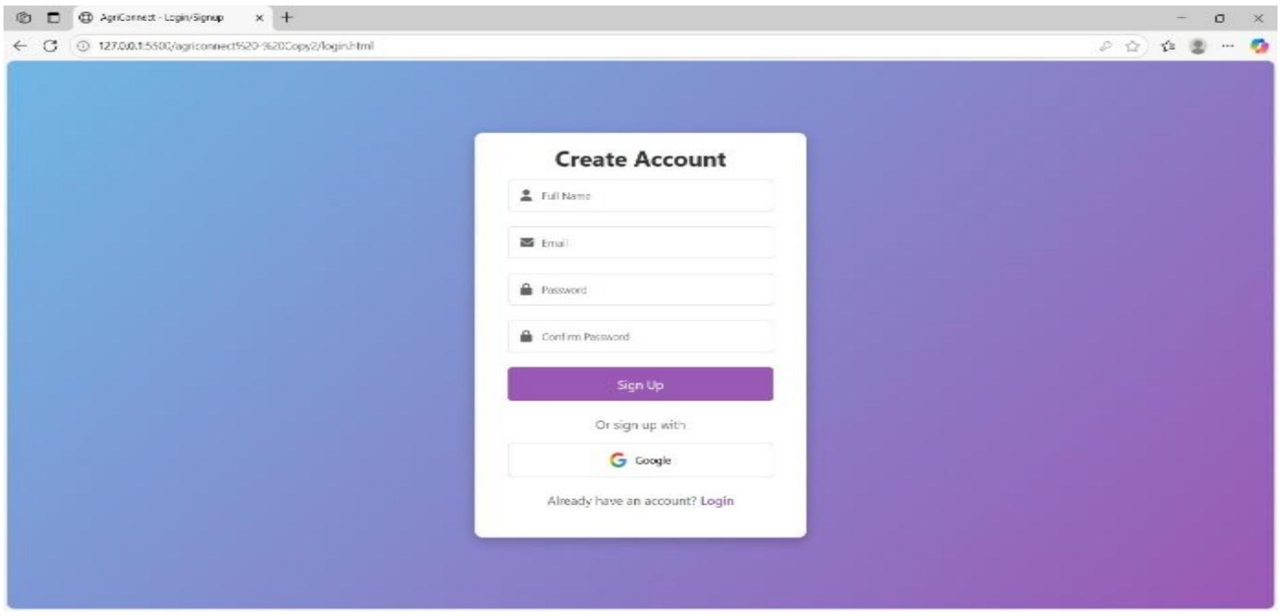
Todo Manager
Organize and track your farming tasks

AI Chat Assistant
Get instant answers to your farming queries

expense Tracker
Organizing tool that helps you record and analyze your spending

Agri News
Impressionism is the newspaper of the soul

Fig 5.1 Home page


AgriConnect - Login/Signup
127.0.0.1:5500/agriconnect%20-%20Copy2/login.html

Create Account

Full Name

Email

Password

Confirm Password

Sign Up

Or sign up with

Google

Already have an account? Login

Fig 5.2. Signup page

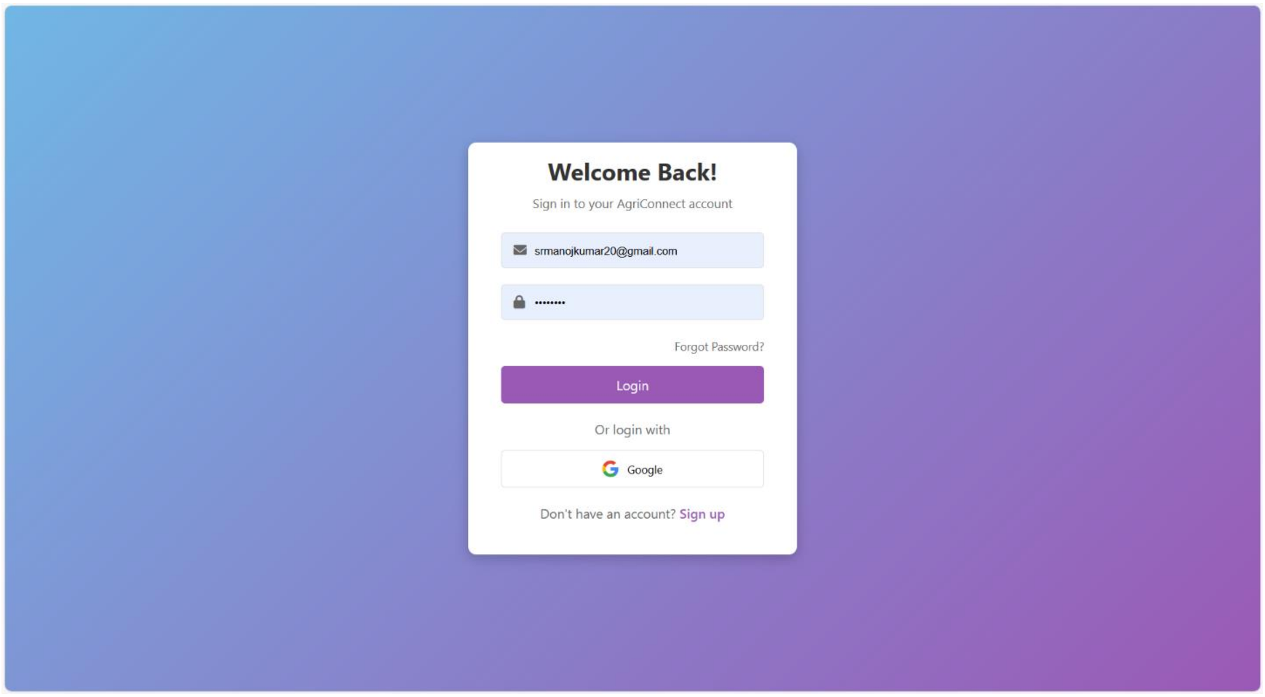
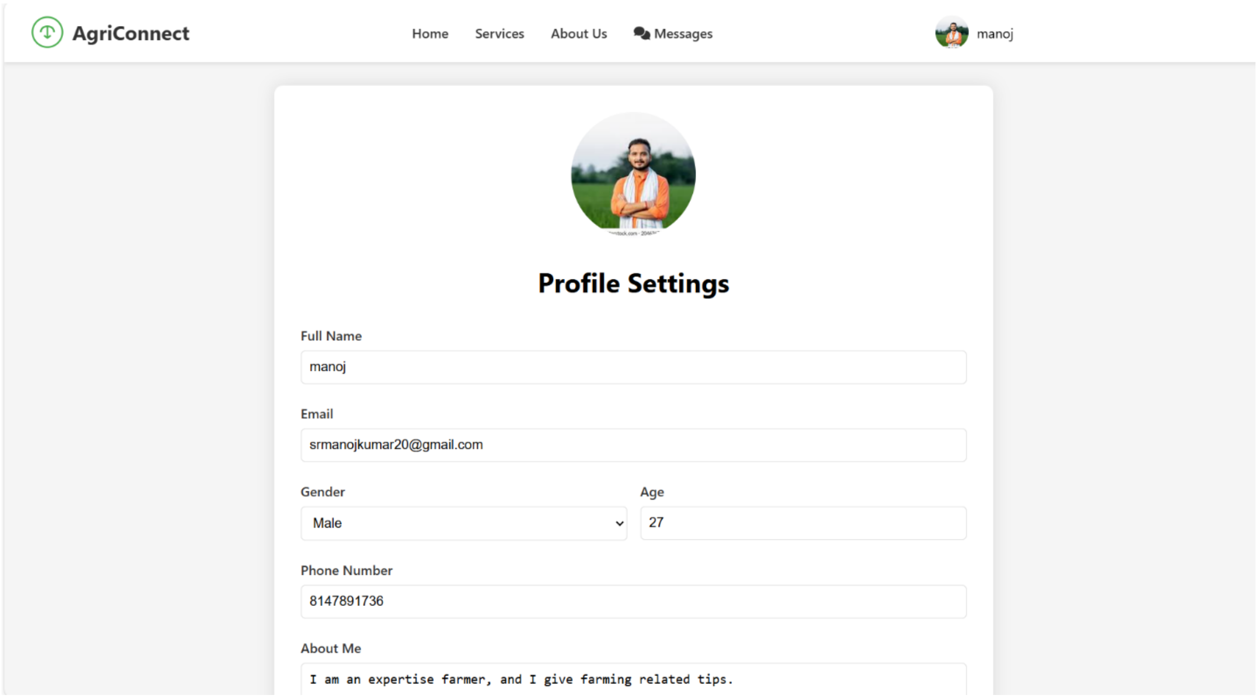


Fig 5.3 Login page



Profile Settings

Full Name: manoj

Email: srmanojkumar20@gmail.com

Gender: Male

Age: 27

Phone Number: 8147891736

About Me: I am an expertise farmer, and I give farming related tips.

Fig 5.4 Profile page

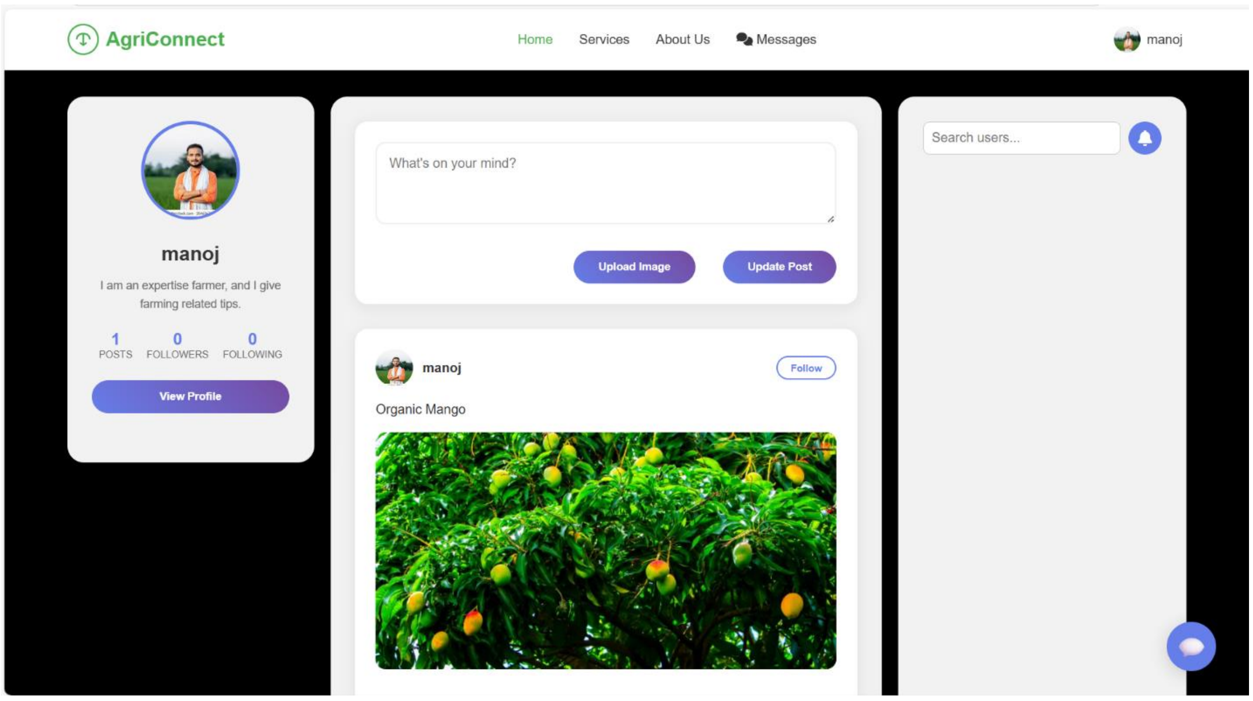


Fig 5.5 Agriconnect page

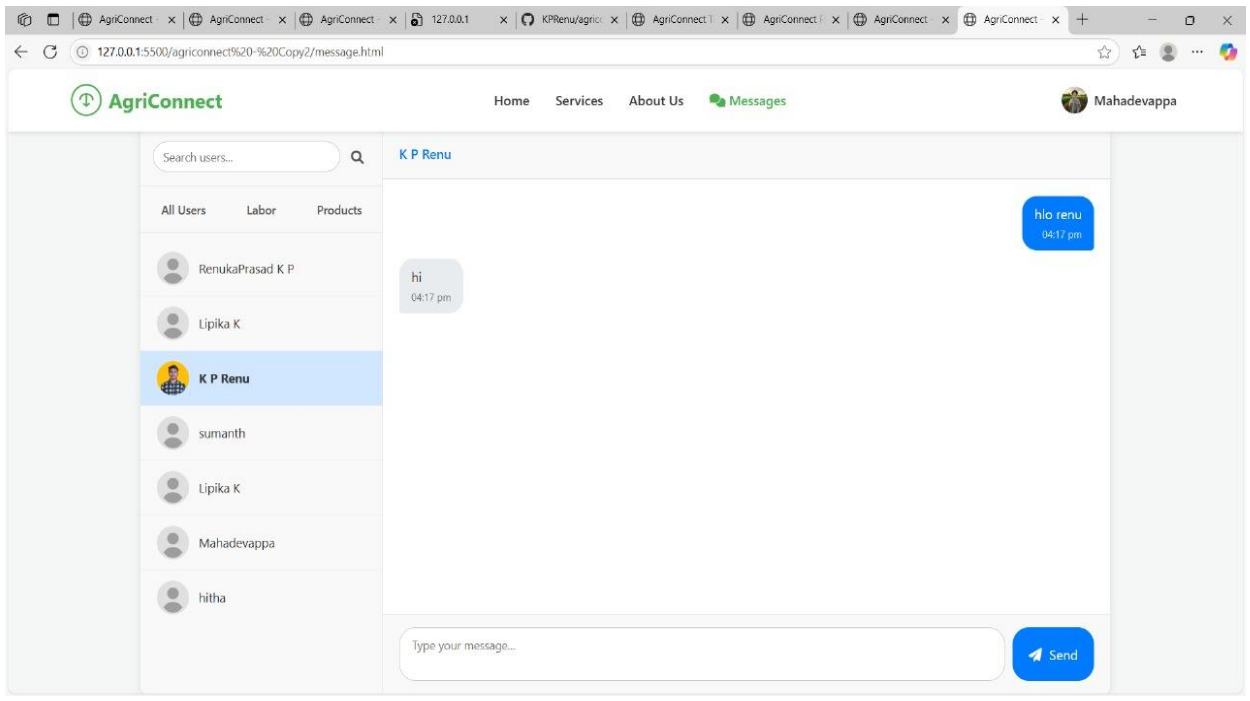
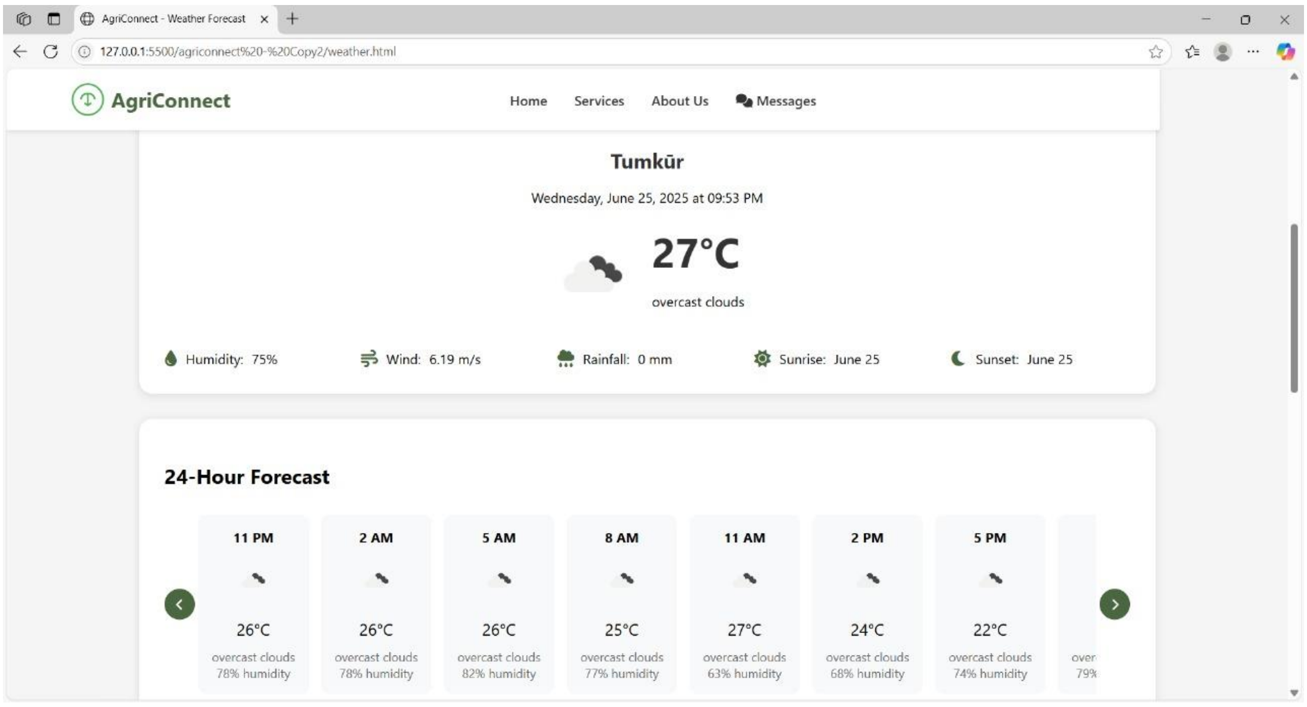


Fig 5.6 Message page


AgriConnect - Weather Forecast
127.0.0.1:5500/agriconnect%20-%20Copy2/weather.html
AgriConnect
Home
Services
About Us
Messages
24-Hour Forecast

Tumkūr

Wednesday, June 25, 2025 at 09:53 PM

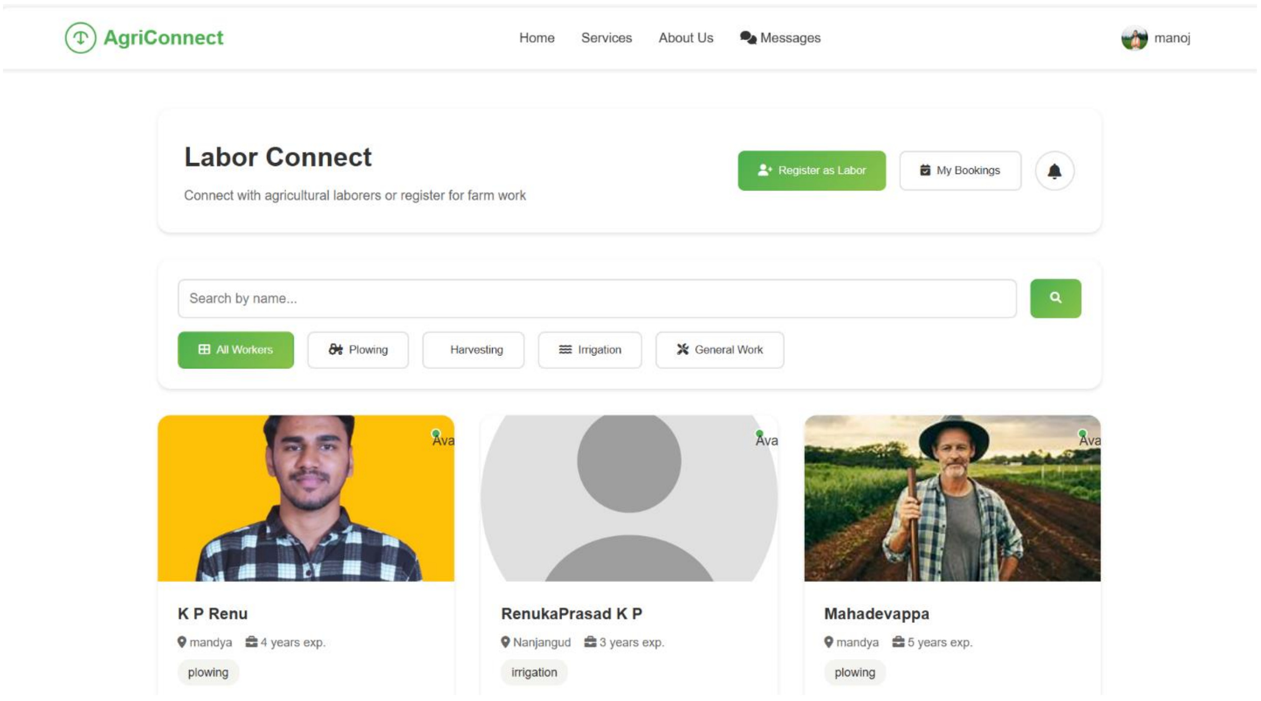
27°C

overcast clouds

Humidity: 75% Wind: 6.19 m/s Rainfall: 0 mm Sunrise: June 25 Sunset: June 25

11 PM	2 AM	5 AM	8 AM	11 AM	2 PM	5 PM
26°C	26°C	26°C	25°C	27°C	24°C	22°C
overcast clouds	overcast clouds	overcast clouds	overcast clouds	overcast clouds	overcast clouds	overcast clouds
78% humidity	78% humidity	82% humidity	77% humidity	63% humidity	68% humidity	74% humidity

Fig 5.7 Weather page



Labor Connect

Connect with agricultural laborers or register for farm work

K P Renu — mandya — 4 years exp. — plowing

RenukaPrasad K P — Nanjangud — 3 years exp. — irrigation

Mahadevappa — mandya — 5 years exp. — plowing

Fig 5.8 Labour page

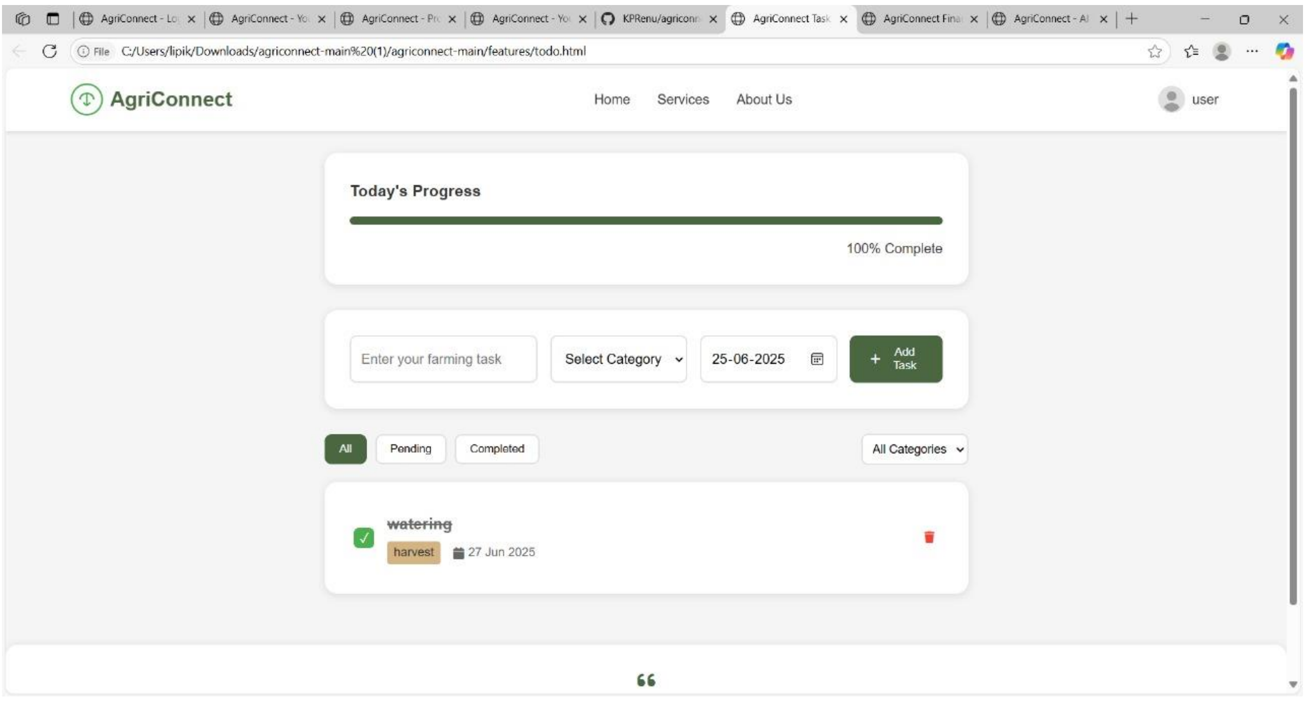
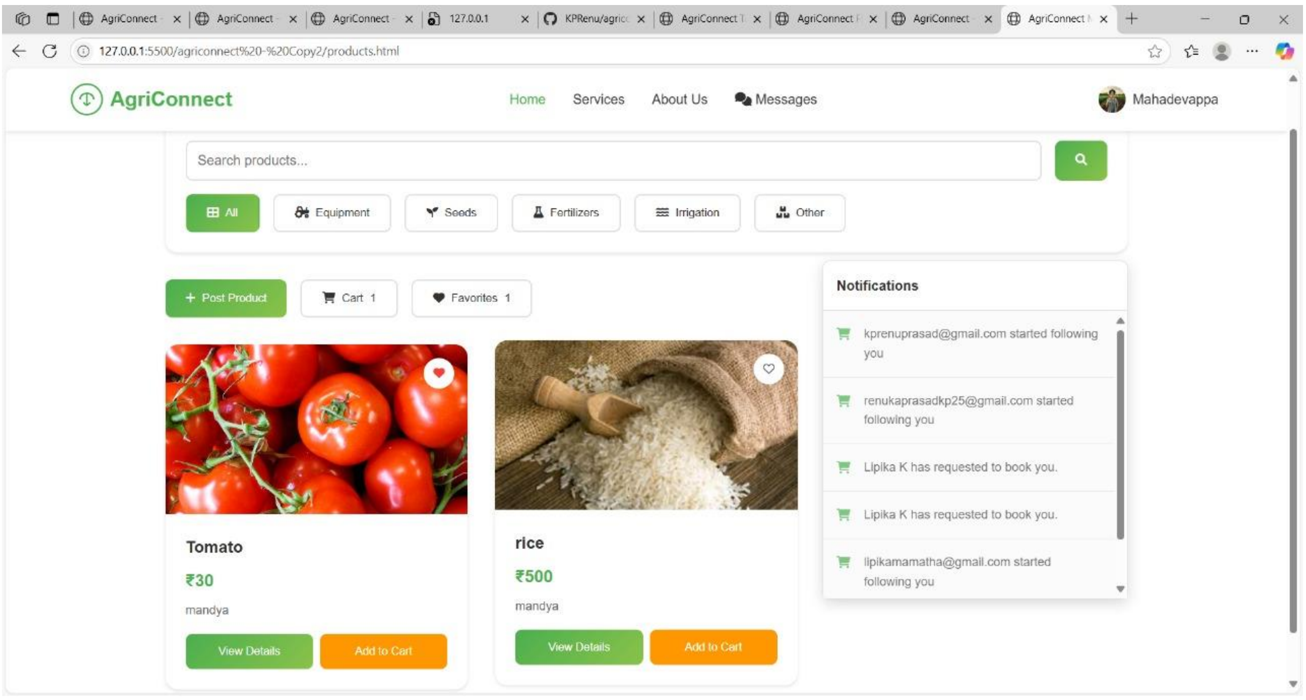


Fig 5.9 Todo page



kprenuprasad@gmail.com started following you

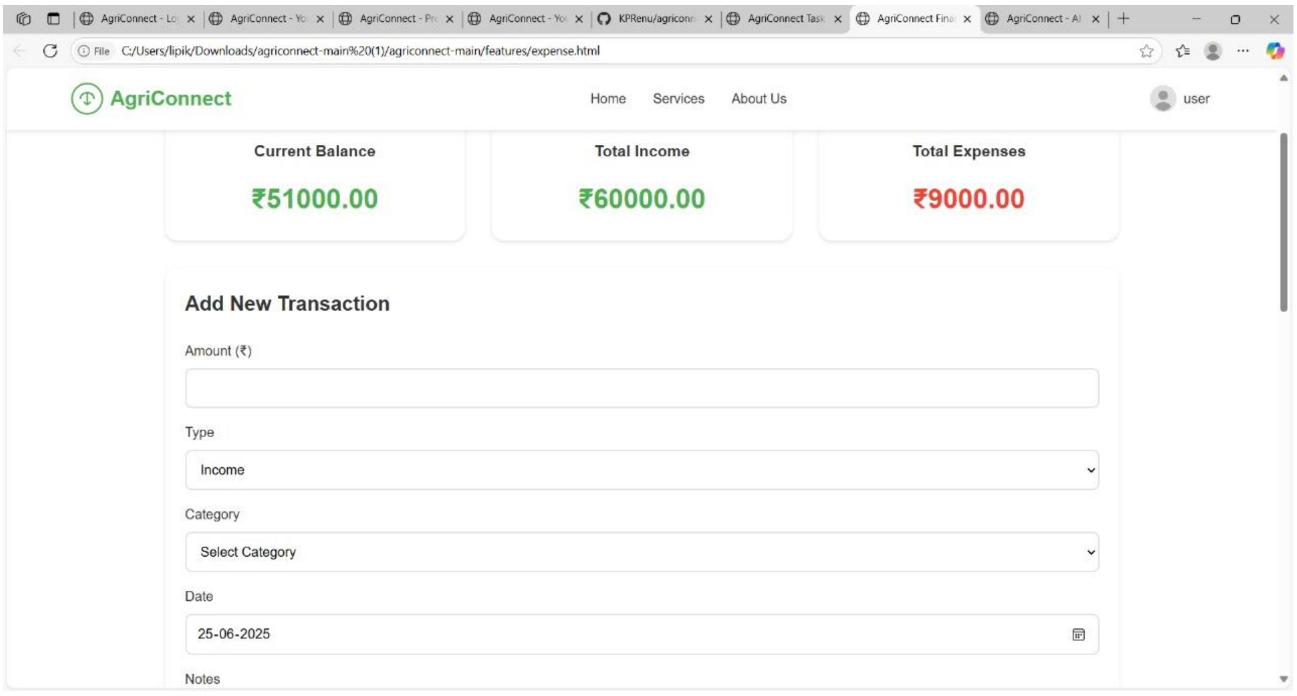
renukaprasadkp25@gmail.com started following you

Lipika K has requested to book you.

Lipika K has requested to book you.

lipikamamatha@gmail.com started following you

Fig 5.10 Product page



AgriConnect

Home Services About Us user

Current Balance ₹51000.00

Total Income ₹60000.00

Total Expenses ₹9000.00

Add New Transaction

Amount (₹)

Type: Income

Category: Select Category

Date: 25-06-2025

Notes

Fig 5.11 Expense page

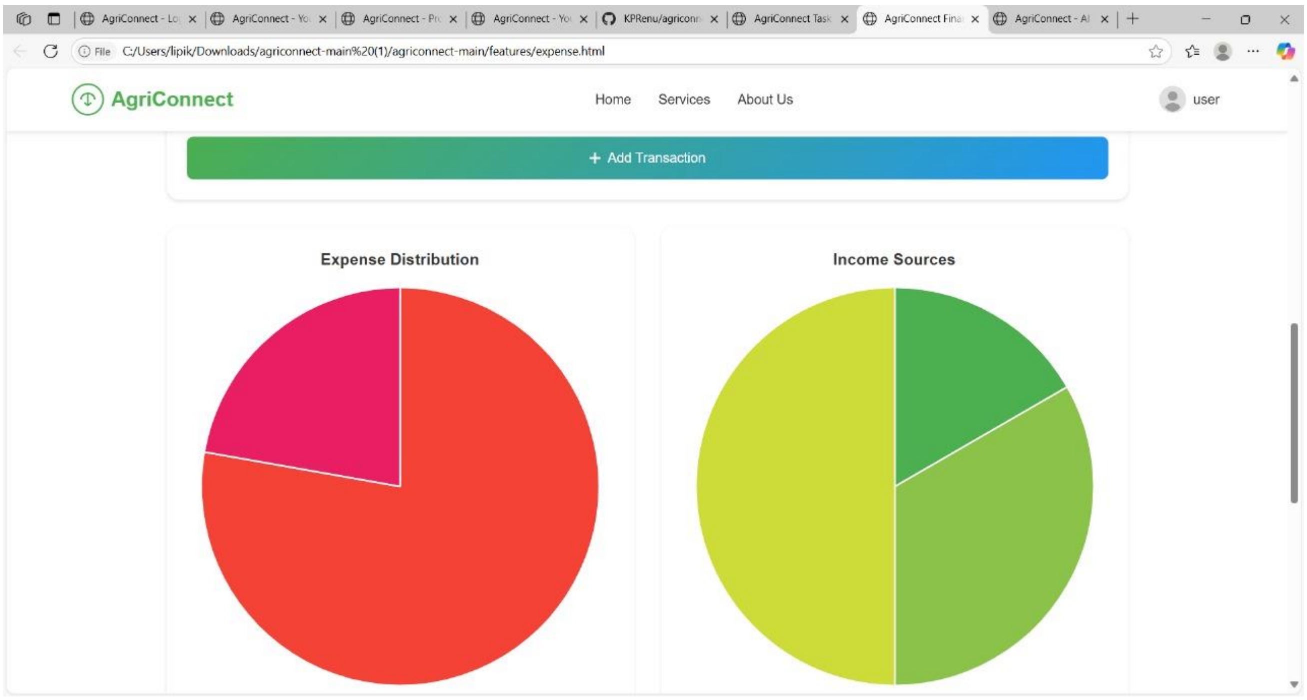
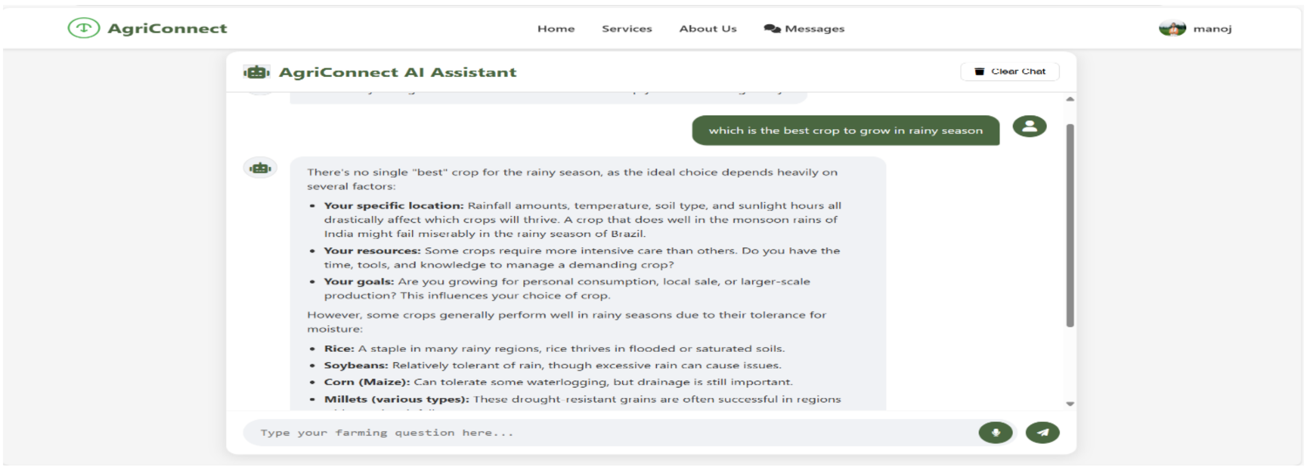


Fig 5.12 Expense graph page



AgriConnect AI Assistant

which is the best crop to grow in rainy season

There's no single "best" crop for the rainy season, as the ideal choice depends heavily on several factors:

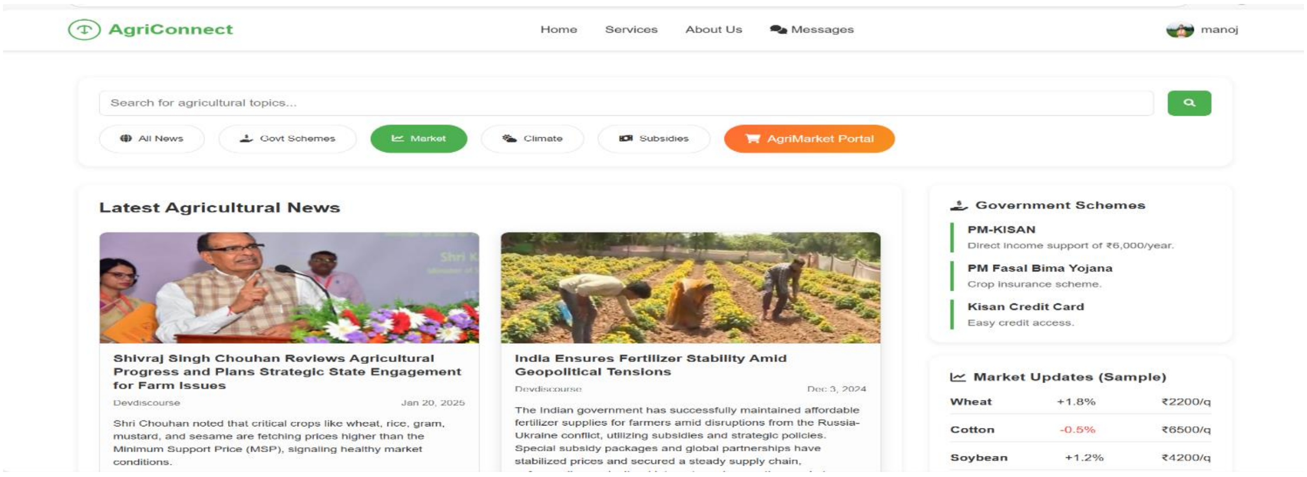
- **Your specific location:** Rainfall amounts, temperature, soil type, and sunlight hours all drastically affect which crops will thrive. A crop that does well in the monsoon rains of India might fail miserably in the rainy season of Brazil.
- **Your resources:** Some crops require more intensive care than others. Do you have the time, tools, and knowledge to manage a demanding crop?
- **Your goals:** Are you growing for personal consumption, local sale, or larger-scale production? This influences your choice of crop.

However, some crops generally perform well in rainy seasons due to their tolerance for moisture:

- **Rice:** A staple in many rainy regions, rice thrives in flooded or saturated soils.
- **Soybeans:** Relatively tolerant of rain, though excessive rain can cause issues.
- **Corn (Maize):** Can tolerate some waterlogging, but drainage is still important.
- **Millets (various types):** These drought-resistant grains are often successful in regions

Type your farming question here...

Fig 5.13 AgriConnect AI Assistant page



Latest Agricultural News

Shivraj Singh Chouhan Reviews Agricultural Progress and Plans Strategic State Engagement for Farm Issues

Devdiscourse Jan 20, 2025

Shri Chouhan noted that critical crops like wheat, rice, gram, mustard, and sesame are fetching prices higher than the Minimum Support Price (MSP), signaling healthy market conditions.

India Ensures Fertilizer Stability Amid Geopolitical Tensions

Devdiscourse Dec 3, 2024

The Indian government has successfully maintained affordable fertilizer supplies for farmers amid disruptions from the Russia-Ukraine conflict, utilizing subsidies and strategic policies. Special subsidy packages and global partnerships have stabilized prices and secured a steady supply chain,

Government Schemes

PM-KISAN
Direct income support of ₹6,000/year.

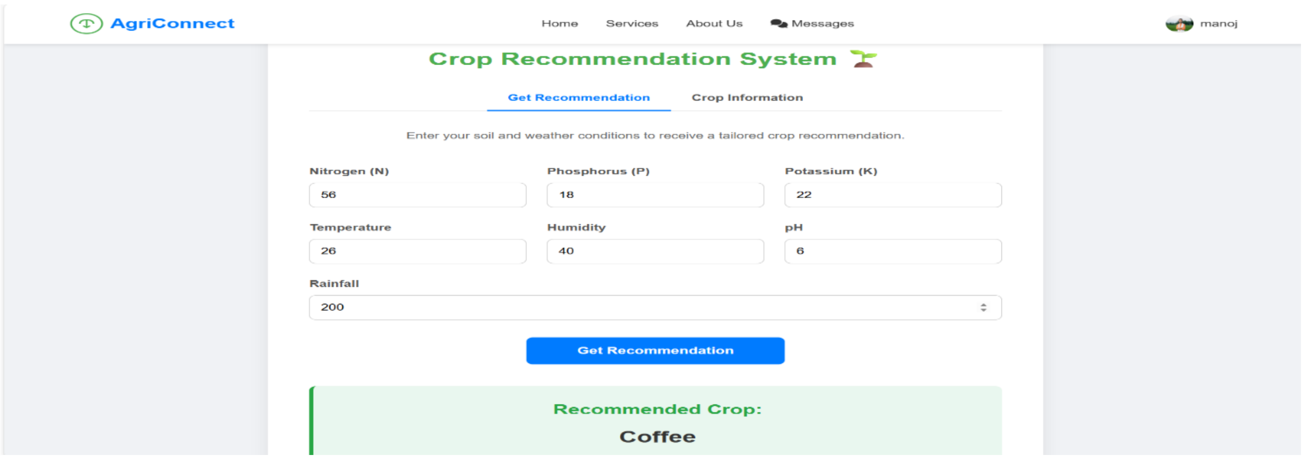
PM Fasal Bima Yojana
Crop insurance scheme.

Kisan Credit Card
Easy credit access.

Market Updates (Sample)

Wheat	+1.8%	₹2200/q
Cotton	-0.5%	₹6500/q
Soybean	+1.2%	₹4200/q

Fig 5.14 Agrinews page



Crop Recommendation System

Enter your soil and weather conditions to receive a tailored crop recommendation.

Nitrogen (N): 56	Phosphorus (P): 18	Potassium (K): 22
Temperature: 26	Humidity: 40	pH: 6
Rainfall: 200		

Get Recommendation

Recommended Crop:
Coffee

Fig 5.15 Crop recommendation page

Chapter 6

CONCLUSION

AgriConnect stands as a testament to the power of integrated digital solutions in transforming traditional sectors. By successfully implementing a multi-domain web platform, we've demonstrated a viable and effective approach to addressing the pervasive challenges of fragmentation and inefficiency within the agricultural ecosystem. Leveraging the robust backend capabilities of **Supabase** alongside a responsive frontend built with **HTML, CSS, and JavaScript**, AgriConnect now provides a centralized hub that caters to the diverse needs of farmers, laborers, agri-businesses, and enthusiasts.

The successful development and integration of eight core functional modules—from social networking and marketplace features to intelligent weather-based crop suggestions and financial tracking—underscore the platform's comprehensive scope. Every key feature, including secure user authentication with **Row-Level Security (RLS)**, efficient data management, and seamless third-party API integrations, is fully operational. This not only validates our chosen technical stack but also highlights our commitment to delivering a secure, scalable, and user-friendly experience.

AgriConnect is poised to make a tangible impact by enhancing decision-making, improving operational efficiencies, fostering economic growth through streamlined market access, and empowering the agricultural community with vital knowledge. While the project currently stands as a strong proof of concept, its foundation is solid, ready for future enhancements such as real-time messaging, payment integrations, or advanced AI capabilities. Ultimately, AgriConnect is more than just a platform; it's a digital bridge connecting the agricultural community of Bengaluru, Karnataka, India, and beyond, cultivating a more informed, efficient, and prosperous future.

Chapter 7

FUTURE ENHANCEMENTS

To further expand the utility and impact of AgriConnect, future updates will focus on enhancing accessibility, intelligence, and usability. One of the key improvements will be the development of a mobile application, allowing farmers and laborers to access services on the go with offline capabilities for low-network areas. Additionally, multilingual support will be introduced to cater to diverse users across India, making the platform easier to use for native language speakers. Role-based dashboards for farmers, laborers, and buyers will be added to provide a more personalized and efficient user experience.

To further enhance AgriConnect's value and reach, the following improvements are envisioned:

1. **Geo-location Based Services:** Enhance modules like **LabourConnect** and **ProductConnect** with geo-location features to show listings relevant to a user's proximity, improving efficiency in local sourcing and hiring.
2. **Payment Gateway Integration:** For **ProductConnect**, integrate secure online payment gateways (e.g., UPI, credit/debit cards) to enable direct, cashless transactions between buyers and sellers, streamlining the marketplace experience.

Chapter 8

BIBLIOGRAPHY

1. https://gnews.io/api/v4/search
2. https://agmarknet.gov.in/vn
3. WEATHER_BASE_URL =
   'https://api.openweathermap.org/data/2.5/weather'
4. FORECAST_BASE_URL =
   'https://api.openweathermap.org/data/2.5/forecast'
5. https://generativelanguage.googleapis.com/v1beta/models/gem
   ini-2.0-flash:generateContent';
6. https://renuagriconnect.netlify.app/login.html