BIOTRACE

REVOLUTIONIZING FARM TO CONSUMER TRANSPERANCY

Prepared by:
Bio Coders

Connecting Sustainable Agriculture to Ethical Consumers

1) Team Name and Member Details

Team Name: "Bio Coders"			
Name	Role	Technologies	Key Tasks
Priyanshu Malusare	Full Stack Developer	Backend: Node.js, Express, REST APIs Frontend: React, TailwindCSS, ReactBits, ThreeJS	• Implement WebSocket for real-time notifications • Develop microservices for payment gateways and live support • Develop interactive 3D models using ThreeJS to enhance product visualization.
Hritik Giri	Project Manager + developer	Frontend : Bootstrap, ReactJS Backend: Python, Django, Reactbits	• Break down project into smaller tasks • Create sprint plan and timeline • Ensure smooth API integration • Monitor project schedule and task allocation
Shubham Gupta	Backend Developer	Backend: Python, Flask, REST APIs	• Implement farmer verification and authentication systems • Handle database interactions • Develop QR code generation and scanning endpoints
Dipesh Gaikar (Leader)	Full Stack Developer	Backend: Node.js, Express, REST APIs Frontend: ReactJS, TailwindCSS	• Create user-friendly dashboards and product pages • Integrate QR code scanning functionality • Ensure responsive design and accessibility • Develop user-friendly dashboards and ensure responsive design.

2) Problem Statement

- **Chosen Problem:** Bridging Natural Farmers and Conscious Consumers

- **Problem Analysis:**
  - Natural farmers struggle to access reliable markets due to a lack of digital presence and verification.
  - Consumers find it difficult to verify the authenticity of natural farming products, leading to trust issues.
  - Intermediaries inflate prices, reducing farmers' earnings and increasing costs for consumers.
- **Target Audience:**
  - Natural farmers looking for fair and direct market access.
  - Conscious consumers seeking verified, natural products.

3) Solution Overview

- **Brief Explanation:**
  A transparent marketplace platform that verifies natural farmers, provides real-time traceability of products via QR codes, and connects farmers directly to consumers, ensuring fair pricing and trust.
- **Approach:**
  - **Farmer Verification:** Digital certification system using **document uploads** and **admin verification**.
  - **Consumer Traceability:** QR codes for product origin, farming methods, and certification details.
  - **Direct Market Connection:** E-commerce interface for verified farmers to list products directly.
- **Uniqueness:**
  - **Digital Certificates:** PDF-based certifications viewable via QR codes.
  - **Real-time Traceability:** Detailed product insights through a centralized database.
  - **Elimination of Intermediaries:** Direct transactions between farmers and consumers.

4) Frameworks/Technologies

- **Tech Stack:**
  - **Backend:** Python (Flask for APIs), Node.js, Express.
  - **Frontend:** React, Bootstrap, TailwindCSS, QR code API, ThreeJS for 3D Models.
  - **Database:** MongoDB.
- **Reasoning:**
  - **Flask and Node.js:** Simplified API development for farmer verification and product management.
  - **React and TailwindCSS**: Responsive and user-friendly interfaces.

5) Feasibility and Implementation

- **Implementation Ease:**
  - Moderate complexity leveraging RESTful APIs for backend management.
  - QR code integration can be streamlined with open-source APIs like **qrcode.react**.
- **Effectiveness:**

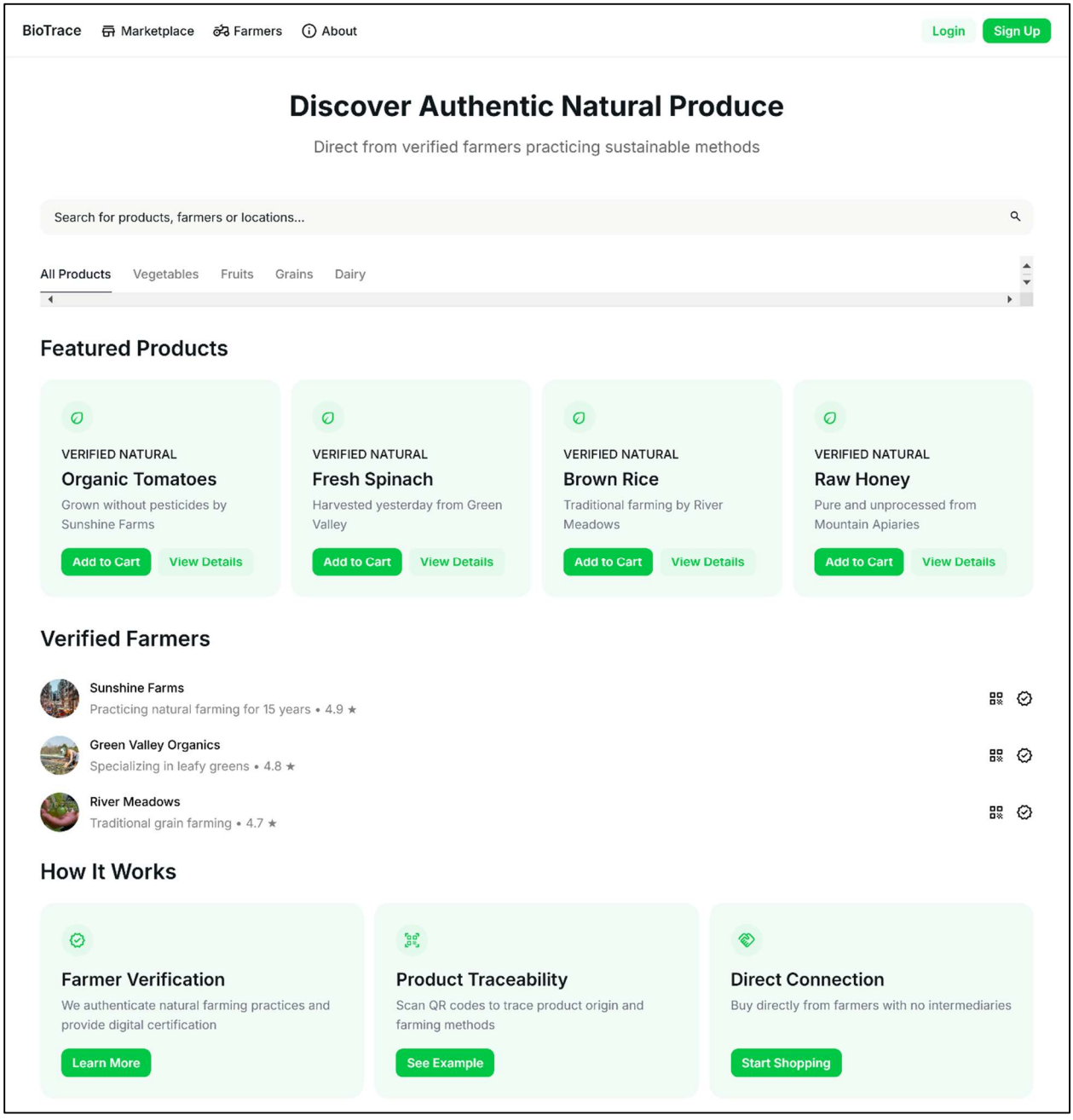
  Addresses the trust gap with traceability and certification.
  - Ensures fair pricing by eliminating intermediaries.

6) UI/UX Mock-up

- **Screens Overview:**
  - **Farmer Dashboard:** Upload products, view sales, certification status.
  - **Consumer Dashboard:** Search products, scan QR codes, view product history.
  - **Admin Panel:** Approve farmers, manage certifications.
- **User Flow:**
  1. Farmers register and submit proof of natural practices.
  2. Admin verifies and certifies farmers.

3. Consumers scan QR codes to verify authenticity.

- **Accessibility Considerations:**
  - Simple language and icons for farmers.
  - High-contrast and User-friendly design

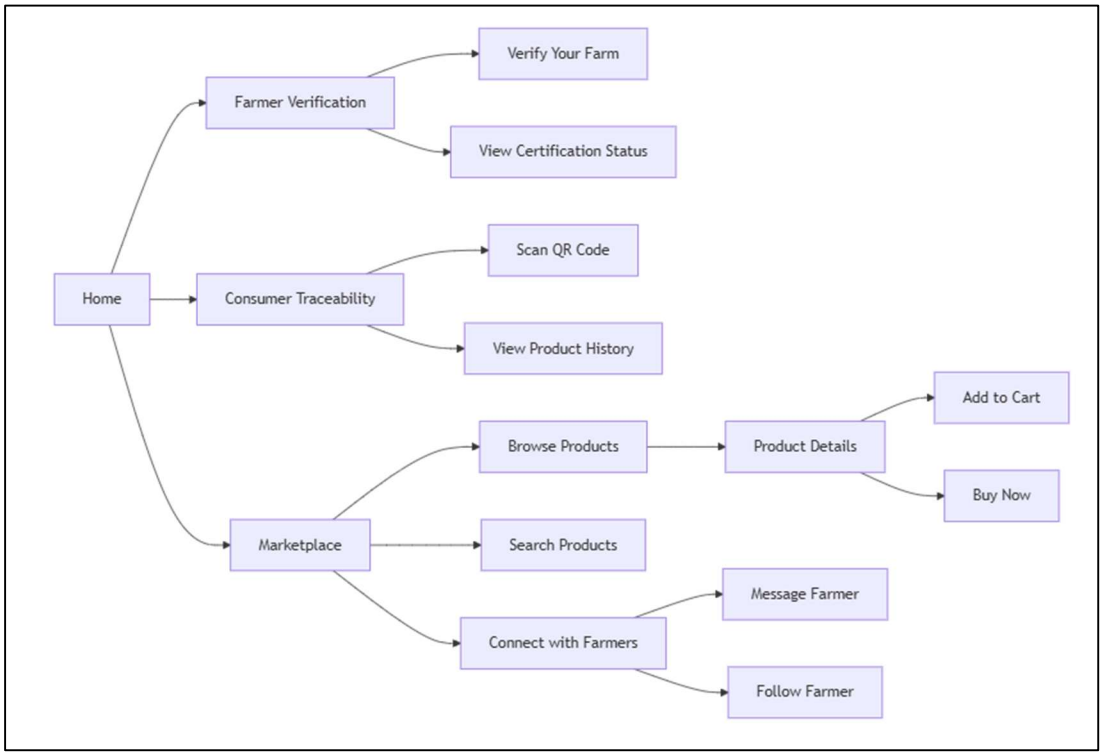


7) Business Scope and Use Case

- **Use Case Scenarios:**
  - A consumer in a metro city verifies farm products before purchase.
  - A natural farmer lists products directly, avoiding middlemen.
- **Market Need:**
  - Increasing demand for organic and verified natural products.
- **Revenue Model :**
  - Transaction fees for each sale.
  - Subscription for premium features (analytics for farmers).

8) System Design and Architecture

- **Technologies Overview:**
  - **Backend:** Flask APIs for farmer verification and QR code generation, Node.js for microservices.
  - **Frontend:** React for UI, MongoDB for data storage.
- **Design Patterns:**
  - **MVC (Model-View-Controller):** Clear separation of backend and frontend logic.
- **Functional Flow:**
  1. **Registration and Verification:** Farmers register and upload proof of natural practices.
  2. **Product Listing:** Admin approves and lists products.
  3. **Consumer Access:** Consumers scan QR codes for product information.



9) Coding Approach

- **Development Strategy:**
  - **Agile Methodology:**
    Adopt a sprint-based approach to ensure iterative progress and adaptability. Prioritize core features like farmer certification, marketplace integration, and QR code functionality in early sprints.
  - **Modular Design:**
    Implement a component-based architecture, separating modules for farmer certification, product listings, and traceability. This will enhance maintainability and scalability.
- **Coding Standards:**
  - **Code Reviews:**
    Establish peer review practices to ensure code quality and adherence to best practices. Leverage tools like ESLint for JavaScript and Pylint for Python to enforce consistent coding standards.