CODECELL - CMPN, VESIT

SYRUS HACKATHON 2025

Category Code: C1

Problem Statement Title: KrishiConnect: AI-Powered Smart Marketplace for Indian Farmers

Team Name: Krishi Mitra

Institute Name: Vivekanand Education Society's Institute of Technology

Idea / Approach details (& implemented features)

1. **Direct Farmer-to-Consumer Marketplace:**
   - Eliminate middlemen with a seamless platform connecting farmers directly to buyers
   - Ensure fair pricing with transparent transactions and real-time market insights
2. **AI-Driven Dynamic Pricing System:**
   - Automatically adjusts prices based on real-time market data
   - Prevents price manipulation and ensures farmers receive competitive rates
3. **Voice-Assisted Management & Multilingual Support:**
   - List products using voice commands in multiple Indian languages
   - Overcome literacy and technology barriers for wider farmer adoption
4. **Comprehensive Tracking & Analytics:**
   - Real-time monitoring of inventory, orders, and market trends
   - Data-driven insights to optimize selling strategies

Innovation (Showstopper)

AI-Powered Agricultural Ecosystem:
In India's agricultural landscape, farmers typically receive only **33-50% of the final consumer price** due to multiple intermediaries. KrishiConnect revolutionizes this system by creating a **direct marketplace powered by advanced AI**.
Smart Crop Recommendation:

- AI analyzes, weather patterns, and market demand
- Predictive analytics to forecast market trends and pricing

Cold Storage & Logistics Integration:

- Real-time availability of nearby cold storage facilities
- AI-optimized logistics recommendations to reduce transportation costs
- Significant reduction in post-harvest losses

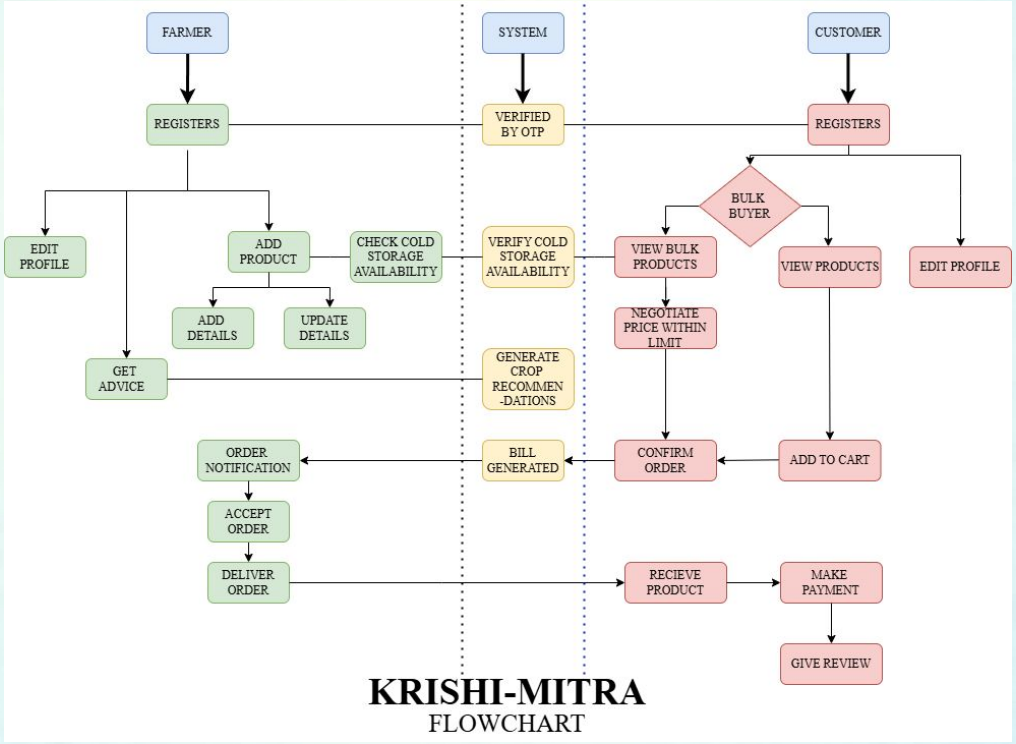
Voice-First Accessibility:

- Multilingual voice commands for farmers with limited tech literacy
- Voice-based market updates and alerts

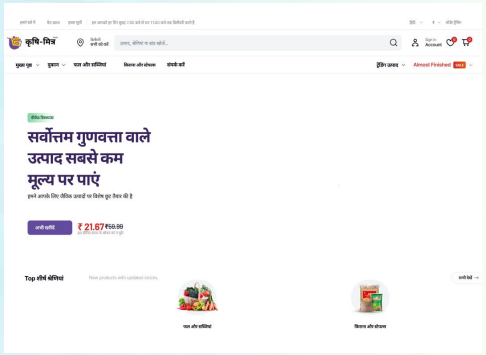
Tech Stack

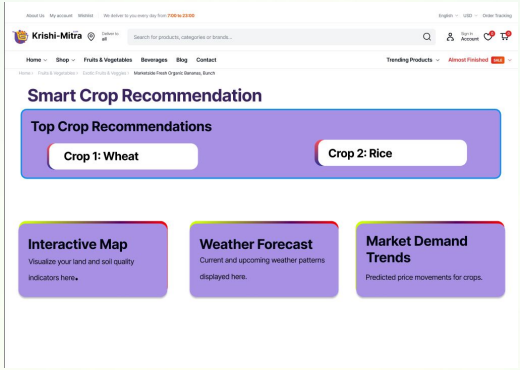
1. **Frontend: React + Tailwind CSS**
   - Responsive design for mobile-first approach
2. **Backend: Node.js + Express.js**
   - Scalable architecture for handling millions of transactions
3. **Database: MongoDB**
   - Flexible schema for diverse agricultural products
   - Efficient data retrieval for market analysis
4. **ML Models: TensorFlow, PyTorch**
   - Advanced algorithms for price prediction and crop recommendations
5. **Authentication: Firebase**
   - Secure user verification,Multi-device synchronization
6. **Tools & APIs:** LangChain, Hugging Face, Sarvam, Mistral Saba, Google DialogFlow
   - Voice recognition and natural language understanding
   - Multilingual support for regional languages

Implementation/Prototype/Use Case Diagram (screenshots)

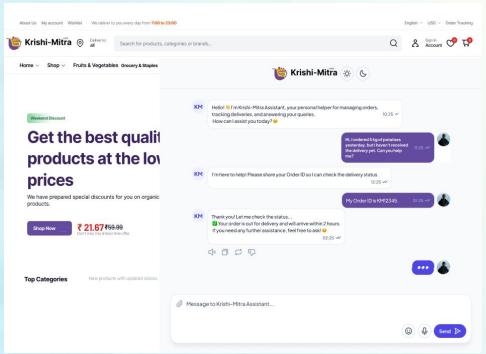


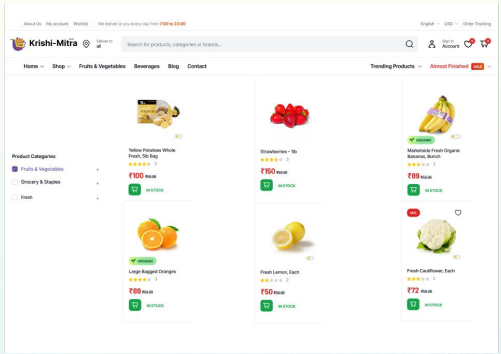
Implementation/Prototype/Use Case Diagram (screenshots)






```
const getOrderById = async (req, res) => {
  try {
    const order = await Order.findById(req.params.id).populate('products.product');
    if (!order) {
      return res.status(404).json({ message: 'Order not found' });
    }
    res.status(200).json(order);
  } catch (error) {
    res.status(500).json({ message: 'Error fetching order', error: error.message });
  }
};

// @desc    Update order status
// @route   PUT /api/orders/:id
// @access  Public
const updateOrderStatus = async (req, res) => {
  try {
    const { status } = req.body;

    if (!['Pending', 'Confirmed', 'Dispatched', 'Delivered'].includes(status)) {
      return res.status(400).json({ message: 'Invalid status value' });
    }

    const order = await Order.findById(req.params.id);

    if (!order) {
      return res.status(404).json({ message: 'Order not found' });
    }

    order.status = status;
    const updatedOrder = await order.save();

    res.status(200).json(updatedOrder);
  } catch (error) {
    res.status(500).json({ message: 'Error updating order', error: error.message });
  }
};
```






```
const registerUser = async (req, res) => {
  const { name, email, password, role } = req.body;
  try {
    const existingUser = await User.findOne({ email });
    if (existingUser) {
      return res.status(400).json({ message: "User already exists" });
    }

    const salt = await bcrypt.genSalt(10);
    const hashedPassword = await bcrypt.hash(password, salt);

    const user = new User({ name, email, password: hashedPassword, role });
    await user.save();

    res.status(201).json({ message: "User registered successfully" });
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};

// Login User
const loginUser = async (req, res) => {
  const { email, password } = req.body;
  try {
    const user = await User.findOne({ email });
    if (!user) {
      return res.status(401).json({ message: "Invalid credentials" });
    }

    const isPasswordValid = await bcrypt.compare(password, user.password);
    if (!isPasswordValid) {
      return res.status(401).json({ message: "Invalid credentials" });
    }

    const token = jwt.sign({ id: user._id, role: user.role }, process.env.JWT_SECRET, { expiresIn: '1h' });
```

Future Objectives

Enhanced AI Prediction Models:

- Implement advanced weather impact analysis on pricing
- Create predictive models for optimal harvest timing

Marketplace Expansion:

- Integrate with international export platforms
- Develop specialized categories for organic and specialty crops

Vision

Agricultural Ecosystem Transformation:

- Creating a self-sustaining, tech-enabled agricultural economy
- Aligning with UN SDGs 1, 2, 8, and 12

Farmer Financial Empowerment:

- Increasing farmer income by 40-60% through direct market access
- Reducing post-harvest losses by 30-50%