

Module 1: Virtual Profitability & Economics Dashboard

Requirement: "Virtual profitability simulations," "Comparative crop economics," "Highlight long-term profitability".

- **The Feature: "Fayda Calculator" (Profit Simulator)**
 - **Functionality:** A split-screen dashboard.
 - **Left Screen (Current Scenario):** The farmer enters their current crop (e.g., Paddy) and land size. The system calculates input costs (water, fertilizer, labor) and estimated profit based on current mandi rates.
 - **Right Screen (Oilseed Scenario):** The system simulates the profit if they switch to Mustard/Groundnut. It highlights **Higher Net Income** by factoring in:
 - Lower water usage cost.
 - Government subsidies (NMEO-OS).
 - Free seed minikits cost deduction.
 - **Visual Output:** A clear bar chart showing "Extra Income Earned" by switching.
 - **Team Tasks:**
 - **Dhiraj (Backend):** Create ComparativeEconomicsAPI. Logic: $(\text{Oilseed_Yield} * \text{Price}) + \text{Subsidy} - \text{Input_Cost}$ vs $(\text{Paddy_Yield} * \text{Price}) - \text{Input_Cost}$.
 - **Ujjwal (AI):** Build the **Yield Prediction Model** that inputs soil type and location to guess how much oilseed will actually grow there.
 - **Vishal/Janhvi (Frontend):** Design the "Split Screen" UI. Use green colors to highlight positive cash flow.
-

Module 2: Market Linkage & Price Alerts

Requirement: "Real-time price alerts," "Market linkages," "Address weak procurement systems," "Assurance tools".

- **The Feature: "Mandi Connect" with Digital Assurance**
 - **Functionality:**
 - **Price Alerts:** Push notifications sent to farmers when oilseed prices in nearby mandis hit a peak.
 - **Digital Assurance:** A booking system where farmers secure a buyer *before* harvest.
 - **The "Winning" Twist:** Use **Blockchain Smart Contracts** to generate a "Digital Pledge." This acts as the "Assurance Tool" that guarantees the buyer cannot back out, solving the risk of "weak procurement."
- **Team Tasks:**

- **Harsh (Blockchain):** Deploy a Smart Contract that records the Farmer_ID, Buyer_ID, Quantity, and Agreed_Price.
 - **Dhiraj (Backend):** Integrate **Agmarknet API** to fetch real-time prices for the alerts.
 - **Vishal/Janhvi (Frontend):** Create a "Sell My Harvest" button that lists nearby FPOs.
-

Module 3: Schemes & Automated Subsidies

Requirement: "Access to schemes and subsidies," "Integrate access to government support schemes like NMEO-OS".

- **The Feature: "One-Click Sahayata" (Support)**
 - **Functionality:**
 - **Auto-Eligibility Check:** The system checks the farmer's profile (Land size, District). If they are in an "Oilseed Focus District" (as per NMEO-OS guidelines), it flags them as eligible.
 - **Auto-Fill Application:** Instead of downloading forms, the app fills the government application digitally using stored user data and generates a PDF.
 - **Team Tasks:**
 - **Dhiraj (Backend):** Build a database table of Rules for NMEO-OS (e.g., if district == 'Bharatpur' and crop == 'Mustard': subsidy = True).
 - **Vishal/Janhvi (Frontend):** A "Claim Subsidy" dashboard that shows a progress bar (Applied -> Verified -> Disbursed).
-

Module 4: Weather Advisories & IoT Monitoring

Requirement: "Weather-based advisories," "Informed decision-making".

- **The Feature: "Khet Nighrani" (Field Watch)**
 - **Functionality:**
 - **Hyper-Local Weather:** Don't just show "Rain." Show "Rain in 2 hours – Cover your harvest."
 - **Soil Intelligence:** Real-time alerts from the field sensors. "Soil moisture is low (20%). Irrigate now to save yield."
- **Team Tasks:**
 - **Naman (IoT):** Program the ESP32 sensors to read Soil Moisture and Temperature every 30 mins and send to the server via MQTT.
 - **Dhiraj (Backend):** Set up **Django Channels** to receive this live stream.
 - **Harsh (Fullstack):** Create the API endpoint that Naman's device will call.

Module 5: Gamification & Incentives

Requirement: "Gamification and incentive-based features," "Encourage farmers to maintain/expand acreage".

- **The Feature: "Kisan Rewards"**
 - **Functionality:**
 - **Points for Actions:** Farmers earn points for: 1. Switching to oilseeds (500 pts), 2. Uploading soil test reports (100 pts), 3. Selling via the app (200 pts).
 - **Leaderboard:** "Top Oilseed Farmer of the Village" badge.
 - **Redemption:** Points can be redeemed for discounts on equipment rentals (Yantra Sathi).
 - **Team Tasks:**
 - **Dhiraj (Backend):** Create a Rewards model in Django. Logic: `user.points += 100`.
 - **Vishal/Janhvi (Frontend):** Design a "Trophy Room" or Leaderboard screen in the app.

Summary of the Workflow (How it connects)

1. Farmer opens app -> Sees **Price Alert** (Module 2).
2. Checks **Profit Simulator** -> Sees Oilseed makes more money (Module 1).
3. Decides to switch -> App auto-applies for **Subsidy** (Module 3).
4. Plants crop -> **IoT Sensors** guide them on water usage (Module 4).
5. Harvests & Sells via **Blockchain Contract** -> Earns **Reward Points** (Module 5).