**1. AI-Powered Plastic Waste Detection from Aerial Images**

Develop an AI/ML model that can identify and classify plastic waste in natural environments (beaches, rivers, forests) using drone or satellite imagery. The model should assist local authorities or NGOs in targeting clean-up operations.

Technology: Computer Vision, Satellite/Drones, CNNs, Object Detection (YOLO/Detectron2)

Solution Should Contain the following:

Data Collection & Annotation: The participating team need to collect drone/satellite imagery datasets (e.g., from [Planet Labs](https://www.planet.com/), Kaggle, Google Earth Engine). Annotate plastic waste manually or use open-source labeled datasets or if it is already available readymade.

Object Detection Model: You can use of implement models like YOLOv5, Faster R-CNN, or Detectron2. You can train use pre-trained model or can train the new model or fine-tune on plastic detection data.

**2. Smart Water Usage Monitoring and Forecasting System**

Build an IoT + ML platform that predicts household or agricultural water consumption patterns and suggests optimal usage schedules to reduce wastage and enhance water conservation.

Technology: Time Series Forecasting, IoT Sensors, Anomaly Detection

Solution Should Contain the following:

IoT Sensor Data Simulator: Simulate water flow data (per minute/hour); Include anomalies like leakage spikes or dry spells

Forecasting Engine: Use ML (Random Forest, ARIMA) or Deep Learning (LSTM); Predict future consumption

Anomaly Detection Module: Detect leaks, overuse, or unusual spikes; Trigger alerts using thresholds or statistical models

**3. Deforestation Risk Prediction and Alert System**

Use satellite data and ML to identify areas at risk of illegal deforestation. The system should analyze land use changes over time and send early warnings to authorities.

Technology: Remote Sensing, Geospatial Data, LSTM/Transformer models for temporal analysis

Solution Should Contain the following:

Satellite Data Acquisition: Use datasets from NASA, Sentinel, Google Earth Engine; Obtain time-series vegetation indices (e.g., NDVI)

ML Model for Risk Prediction: Train a model on historical data to forecast likely deforestation zones; Use features like proximity to roads, urban areas, economic activities

Alert Engine: Set thresholds for vegetation loss; Notify users/authorities via webhooks or email

**4. Urban Heat Island Effect Visualizer and Reducer**

Create an interactive tool that uses satellite and urban data to visualize heat islands in cities and recommend green architecture interventions (like rooftop gardens, tree plantation zones).

Technology: Geospatial ML, GIS Tools, Data Visualization, Recommender Systems

Solution Should Contain the following:

Urban Data Aggregation: Collect data like land surface temperature, vegetation cover, buildings. Sources: Landsat, MODIS, OpenStreetMap

Heat Island Detection: Use ML models to classify heat-affected areas; Create clusters of high/low temperature zones

Impact Analysis: Correlate temperature with pollution, health stats, or energy use; Identify most vulnerable regions

Green Intervention Recommendation Engine

Suggest urban forestry, reflective roofs, green walls

ML model trained on successful case studies

**5. Sustainable Living AI Assistant**

Design a personal AI assistant that guides users toward a sustainable lifestyle by tracking their carbon footprint from purchases, travel, and electricity usage—and suggests greener alternatives.

Technology: NLP, Recommendation Systems, User Behavior Modeling, Carbon Footprint Estimation APIs

Solution Should Contain the following:

User Activity Logging: Accept input (manual or via APIs) for electricity, travel, shopping; Store in structured format for analytics

Carbon Footprint Estimator: Estimate emissions from logged activities using standard coefficients (e.g., kg CO₂/unit)

NLP-based Chat Assistant: Build a chatbot using OpenAI/GPT/NLP techniques; Answer questions about lifestyle choices and sustainability

Recommender System: Suggest eco-friendly products, travel alternatives, renewable energy options; Use collaborative or content-based filtering

**6: Intelligent Home Power Consumption Monitoring and Optimization System**

Energy consumption in households is a major contributor to carbon emissions. Many households lack real-time insights into their energy usage patterns and miss opportunities for reducing waste. Smart energy management can lead to significant environmental and economic benefits.

Develop an AI-powered system that monitors power consumption of different household appliances, identifies energy-draining devices, predicts peak usage times, and recommends optimization strategies for sustainable and cost-efficient living.

**Technology: Backend**: Python (Pandas, Scikit-learn, TensorFlow, FastAPI). **Frontend**: React or Streamlit. **Visualization**: Plotly, Chart.js. **Data**: Open power usage datasets or simulated smart meter data

**Key Features / Goals:**

1. **Real-time Power Monitoring**: Visualize power usage per appliance.
2. **Appliance-Level Breakdown**: Classify power consumption by device types.
3. **Usage Prediction**: Forecast future energy usage using historical data.
4. **Anomaly Detection**: Identify faulty or unusually consuming appliances.
5. **Optimization Suggestions**: Recommend ways to reduce consumption (e.g., usage schedules, replacements, solar alternatives).

**Data Collection and Data Preprocessing**

* Collect power usage data from smart meters or simulate data per device.
* Format: Timestamped logs per appliance (kWh/hour).
* Aggregate by time intervals (hour/day/week).
* Classify appliances using clustering (unsupervised) or device IDs.

**Forecasting Engine**

* Train ML models (LSTM, Prophet, XGBoost) to predict next day/week usage.
* Include seasonality, weekends, holidays in the prediction.
* Use statistical thresholds or ML models to flag abnormal usage.
* Example: AC running while user is away.

**Recommendation & Optimization Engine**

* Suggest off-peak usage timings based on utility tariff plans.
* Recommend device upgrades (e.g., switch to energy-efficient fridge).
* Optional: Simulate solar panel benefits for the household.

**7: AI-Driven Eco-Friendly Gardening Assistant**

Gardening can significantly contribute to local biodiversity, reduce carbon footprints, and improve urban air quality. However, many gardeners—especially in urban areas—lack knowledge on sustainable gardening practices, plant compatibility, or how to minimize water and fertilizer usage.

Develop an AI-powered gardening assistant that helps users create and maintain **eco-friendly gardens** by recommending native, low-maintenance, and biodiversity-supporting plants, optimizing watering schedules, and detecting plant health issues from images.

**Technology**: **ML/AI**: TensorFlow/Keras for image classification, Scikit-learn for recommendations. **Backend**: FastAPI/Flask. **Frontend**: React.js or Streamlit. **Data Sources**: PlantNet API, OpenFarm, OpenWeather API. **Optional**: Integration with Arduino or Raspberry Pi for smart garden sensors

**Key Features / Goals:**

1. **Personalized Plant Recommendations** based on location, climate, soil type, and garden size.
2. **Watering and Care Optimization** using weather forecasts and plant-specific needs.
3. **Plant Disease and Health Detection** via image recognition.
4. **Eco-Impact Score** showing environmental benefit of gardening choices.
5. **Companion Planting Suggestions** for natural pest control and nutrient sharing.

**User Input & Environment Profiling**

* Collect inputs: location, soil type, sunlight hours, space available.
* Use APIs (OpenWeather, Plant APIs) to get regional climate and seasonal data.

**8: FairChain – AI-Powered Transparent Agri-Marketplace**

Farmers often receive minimal returns for their produce, while intermediaries and retailers capture most of the profits. Consumers, on the other hand, pay high prices without knowing the origin or value chain. There’s a pressing need for a **transparent, AI-assisted platform** that ensures **fair profit distribution** and **trust** among all stakeholders—**farmers, consumers, and intermediaries**.

Design a digital platform that connects farmers, intermediaries (logistics, aggregators), and consumers. The system should:

* Ensure transparent pricing and profit-sharing.
* Use AI to forecast demand and supply.
* Provide traceability of goods from farm to table.
* Allow smart contracts or rules for equitable profit splits.

**Technology**: **Frontend**: React Native (for app), Next.js (for web). **Backend**: Django, FastAPI, or Node.js. **AI/ML**: Time Series models (Prophet, XGBoost, LSTM) for forecasting. **Data Sources**: FAO data, local mandi prices, OpenWeather for crop trends. **Database**: PostgreSQL with PostGIS (for location), or Firebase. **Optional**: Blockchain simulation using Hyperledger or simple hash-chaining

**Key Features / Goals:**

1. **Farmer Onboarding** and produce listing.
2. **Demand Forecasting** using AI/ML for better planning.
3. **Dynamic Pricing Model** ensuring fair profit shares.
4. **End-to-End Traceability** of produce using digital tracking.
5. **Transparent Transaction Ledger** (can be blockchain-inspired, even if not using real blockchain).

**Stakeholder Onboarding & Profiles**

* **Farmer Profile**: Location, crop type, quantity, expected harvest time.
* **Intermediary Profile**: Logistics providers, aggregators, storage units.
* **Consumer Profile**: Retailers, end users, bulk buyers.

**Produce Listing & Supply Chain Builder**

* Farmers list produce, with images, quantity, and expected date.
* Intermediaries bid or register for logistics.
* Consumers see options with detailed supply chain info.

**AI-Powered Demand Forecasting (Fair Pricing & Profit Sharing Model)**

* Predict demand based on historical trends, seasonality, and weather data.
* Notify farmers about high-demand crops before sowing seasons.
* Calculate base price from input cost + market trend.
* Split profits transparently: e.g., 60% to farmer, 20% to logistics, 20% to platform/intermediaries.
* Let buyers see the exact breakdown.

**Traceability & Transparency Engine**

* Generate a **digital produce passport**: track origin, route, and intermediaries.
* Use QR codes or IDs for tracking and trust.
* Optional: Blockchain or hash-based logging for immutability.

**Payment & Trust Layer**

* Secure digital payment gateway with escrow option.
* Ratings for each transaction (farmer, transporter, consumer).
* Smart contract-style rules (e.g., only release payment after delivery confirmation).