PROBLEM STATEMENTS

1. FactCheck: The Social Media Integrity Guard

Scenario

AI-generated deepfakes and manipulated videos are increasingly common on social media, allowing misinformation to spread faster than verification. Users often lack a quick and reliable way to assess whether a video is genuine before trusting or sharing it.

The Hackathon Challenge

Build an **AI-powered Video Integrity Scanner** that analyzes short videos and helps users determine whether the content is authentic or manipulated.

The system should:

- Accept a **video upload or public video URL**.
- Apply **AI and forensic analysis** to assess authenticity.
- Generate a clear **Trust Score** with understandable explanations.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Video Input**
    - Upload video files or paste public URLs.
    - Display a preview of the analyzed video.
- **Metadata & Integrity Analysis**
    - Examine encoding details and metadata for anomalies.
- **Visual & Temporal Forensics**

    - CNN-based frame-level deepfake detection.
    - Detection of flickering or motion inconsistencies.
- **Audio–Visual Synchronization**
    - Analyze lip-sync mismatches between audio and video.
- **Trust Score & Explanation**
    - Classify videos as **Authentic**, **High Risk**, or **Manipulated**.
    - Highlight suspicious frames and failed checks.
- **User Interface**
    - Simple web or mobile dashboard showing the video, Trust Score, and concise findings.
    - Optional: browser extension to flag suspicious social media videos.

2. Women Safety Analytics: Eyes That Anticipate

Scenario

Ananya often travels alone after evening classes or late work shifts. Although CCTV cameras are deployed across the city, they mostly act as passive recorders rather than preventive tools. In poorly lit areas or during low footfall hours, she feels unsafe knowing that intervention usually happens only after an incident. Law enforcement teams, despite their efforts, find it difficult to continuously monitor thousands of camera feeds.

Ananya wishes there were an intelligent system that could **sense danger early**—by understanding who is present, how people are positioned, and whether behavior indicates risk—so that alerts could be generated *before* a situation escalates.

The Hackathon Challenge

Build a **Real-Time Women Safety Analytics System** that proactively identifies potentially unsafe situations involving women using video analytics and machine learning.

The solution should:

- Use **live or recorded camera feeds** to monitor public spaces.
- Detect individuals and **classify gender** in real time.
- Analyze **gender distribution and proximity patterns**.
- Identify **high-risk scenarios** based on context and behavior.
- Generate **timely alerts** to enable preventive action.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Person Detection & Gender Classification**
    - Detect people in the scene and classify them by gender in real time.
- **Contextual Risk Analysis**
    - Analyze gender ratios and spatial proximity to assess potential risk levels.
- **Vulnerable Situation Detection**
    - Identify scenarios such as:
        - A **lone woman during late hours**
        - A **woman surrounded by multiple men**
- **Gesture-Based SOS Recognition**
    - Detect simple predefined distress or SOS gestures.
- **Alert Generation & Logging**
    - Trigger real-time alerts when a risky situation is detected.
    - Maintain a basic alert log highlighting frequently flagged locations or time periods.

3. AudioScribe: Turn Your Ink Into Insights

Scenario

Vikram is a diligent student who fills his notebooks with handwritten notes, diagrams, and explanations during lectures. As exams approach, reviewing piles of notebooks becomes overwhelming. He realizes that time spent commuting or exercising could be used for revision—but carrying notebooks or reading while on the move isn't practical. Vikram wants a way to convert his handwritten notes into audio so he can revise anytime, hands-free.

The Hackathon Challenge

Build a **Learning Assistant** that converts handwritten notes into spoken audio, enabling students to listen to their own study material on the go.

The solution should:

- Accept **images of handwritten notes** captured using a smartphone or scanner.
- Convert handwritten content into **accurate digital text**.
- Read the converted content aloud using **natural-sounding audio**.
- Allow **hands-free interaction** to control audio playback.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Handwritten Text Recognition**
    - Use OCR to extract text from handwritten cursive or printed notes.
- **Text-to-Speech Conversion**
    - Convert extracted text into clear, natural-sounding speech.
- **Voice-Controlled Playback**
    - Support voice commands such as:
        - **Pause** – stop audio playback immediately
        - **Resume** – continue from the last position
        - **Backtrack** – replay the previous line or section
        - **Next** – skip to the next paragraph or section

- **User-Friendly Interface**
    - Simple interface for uploading images and controlling audio playback.

4. SignSpeak: The Silent Bridge

Scenario

Karthik is speech-impaired and communicates using Sign Language. While his friends understand him, everyday interactions at places like grocery stores, banks, or offices become challenging because most people do not know sign language. Typing messages on his phone is slow and breaks the natural flow of conversation. Karthik wishes he could simply **use his hands to speak**, and be understood instantly.

The Hackathon Challenge

Build a **Real-Time Sign-to-Speech Translator** that converts sign language gestures into spoken language, enabling seamless communication between sign language users and non-signers.

The solution should:

- Use a **smartphone camera** to capture hand gestures in real time.
- Translate **common signs and gestures** into meaningful text or sentences using an ML model.
- Convert the translated output into **real-time speech** using a Text-to-Speech (TTS) engine.
- Support **gesture-based controls** to repeat or clear the last spoken phrase.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Gesture Capture & Recognition**
    - Detect and interpret hand gestures using computer vision.
- **Sign-to-Text Translation**

    - Map recognized signs to words or sentences using a trained ML model.
- **Text-to-Speech Output**
    - Convert translated text into clear, audible speech instantly.
- **Gesture-Based Controls**
    - Specific gestures to **repeat** or **clear** the previous output.

5. DocFlow: Smart Document Approval & Tracking

Scenario

In many educational institutes, students are required to submit documents for verification and approval across multiple departments—for admissions, scholarships, internships, or certifications. Currently, this process is often manual and fragmented. Documents are rejected due to incorrect formats or missing details, approvals are delayed, and students have little visibility into where their document is or who needs to act next.

Students want a system that can **validate documents upfront**, route them automatically to the right authorities, and clearly show the approval progress at every stage.

The Hackathon Challenge

Build a **Web-Based Document Approval and Workflow Management System** that validates student-submitted documents, routes them through predefined approval stages, and provides end-to-end tracking.

The solution should:

- Allow students to **upload documents** through a web interface.
- **Automatically validate** document format and required fields.
- Route valid documents to the **appropriate department for review**.

- Support **multi-level approvals** by forwarding documents across departments.
- Provide **real-time status tracking** for students and approvers.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Document Submission & Validation**
    - Upload documents in supported formats.
    - Validate file type and check for mandatory details.
- **Approval Workflow Management**
    - Assign documents to the correct department for verification.
    - Enable approvers to **approve, reject, or request corrections**.
    - Allow approved documents to be **forwarded to additional departments**.
- **Status Tracking & Visibility**
    - Display current approval stage and pending authority.
    - Show a complete approval history and timestamps.
- **User Roles & Access Control**
    - Separate roles for **students, department reviewers, and administrators**.
- **Notifications & Alerts**
    - Notify users on submission, approval, rejection, or forwarding actions.

6. DDAS: Smart Download Watchdog

Scenario

In an institute, students, researchers, and faculty regularly download datasets, papers, and resources from the web. Over time, the **Downloads folder fills up with duplicate files**—the same dataset downloaded multiple times because users forget it already exists. Sometimes files have the same name but different content, and other times the content is identical but renamed, making manual checks unreliable.

A researcher wants a **smart browser assistant** that can detect *true duplicates* before a file is downloaded and warn them in advance, saving storage, bandwidth, and time—without falsely flagging different files that happen to share the same name.

The Hackathon Challenge

Build a **Chrome Extension–based Data Download Duplication Alert System (DDAS)** that monitors downloads and alerts users when an identical file already exists in their system.

The solution should:

- Work as a **Chrome browser extension**.
- Monitor **new download requests** initiated from the browser.
- Check the local **Downloads folder** for existing files.
- Detect **true duplicates** using multiple file attributes, not just filenames.
- Notify users **before the download completes** if a duplicate is found.
- Allow users to **proceed or cancel** the download.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Download Monitoring**
    - Track file downloads initiated through the Chrome browser.
- **Duplicate Detection Logic**
    - Compare files using multiple factors such as:
        - File size
        - File hash/checksum

        - MIME type or metadata
    - Avoid false positives when files share names but differ in content.
- **Pre-Download Alerting**
    - Display a browser notification or popup warning when a duplicate is detected.
    - Show details such as:
        - Existing file name
        - Download location
        - File size or timestamp
- **User Decision Control**
    - Allow users to **cancel or continue** the download after seeing the alert.
- **Extension Interface**
    - Simple UI to view recent downloads and detected duplicates.
    - Optional toggle to enable or disable duplicate checks.

This extension should demonstrate how intelligent client-side monitoring can **prevent redundant downloads** and improve everyday productivity without disrupting the user's browsing experience.

7. ExpireSense: Never Miss a Date

Scenario

Riya often buys groceries, medicines, and household items in bulk. Over time, products get pushed to the back of shelves and forgotten—only to be discovered after they've expired. This leads to wasted food, unused medicines, and unnecessary expenses. Checking expiry dates manually is tedious, and there's no single place to track everything.

Riya wishes she had a simple mobile app that could scan products when she buys them, automatically read expiry dates, organize items neatly, and remind

her before anything goes bad—so she can use products on time or replace them responsibly.

The Hackathon Challenge

Build a **Smart Expiry Reminder Mobile Application** that helps users track the lifecycle of everyday products and receive timely alerts before expiration.

The solution should:

- Allow users to **log in securely** using a simple authentication mechanism.
- Enable users to **scan packaged products** using a smartphone camera.
- Automatically **extract expiry dates** from packaging using OCR or AI-based text detection.
- Store products in a **structured inventory** with categories such as groceries, medicines, or personal items.
- Track expiry timelines and **generate reminder alerts** ahead of expiration.
- Notify users through **at least one working notification channel**.

Key Features to be Implemented

The prototype should demonstrate the following core capabilities:

- **User Authentication**
    - Basic login (email, phone, or social login).
- **Product Capture & Date Extraction**
    - Camera-based scanning of product labels.
    - OCR-based detection of expiry date text.
- **Item Organization**
    - Automatic or rule-based categorization of products.
    - Optional support for custom categories.
- **Expiry Monitoring & Alerts**
    - Default reminder (e.g., 7–30 days before expiry).
    - Simple configuration for reminder timing.

- **Notifications**
    - Push notification, email, or in-app alerts.

8. CoalZero: Measuring Today, Planning Net-Zero

Scenario

India's energy sector continues to rely heavily on coal mining to meet growing demand. While coal remains critical for energy security, mining operations contribute significantly to greenhouse gas emissions through excavation, transportation, fuel consumption, and electricity usage. Mine operators are increasingly expected to align with national climate commitments, yet many lack clear tools to measure their emissions or evaluate realistic paths toward carbon neutrality.

A coal mine manager wants a simple digital platform where operational data can be entered and instantly translated into emission estimates. More importantly, they want to explore *what-if scenarios*—such as adopting cleaner equipment, increasing renewable energy use, or planting trees—to understand how these actions could reduce the mine's carbon footprint over time.

The Hackathon Challenge

Build a **Web-Based Carbon Footprint and Carbon Neutrality Planning Tool** tailored for coal mining operations.

The application should:

- Allow users to **input basic operational data** related to mining activities.
- Estimate **activity-wise carbon emissions** using predefined emission factors.
- Calculate **per-capita emissions** for a mine based on workforce size.
- Estimate **existing carbon sinks** (e.g., green cover, plantations).
- Perform a **gap analysis** between emissions and absorption capacity.

- Suggest **simplified pathways toward carbon neutrality** through multiple strategies.

Key Features to be Implemented

The prototype should demonstrate the following capabilities:

- **Emission Estimation Module**
    - Calculate emissions from activities like excavation, transportation, and energy usage using fixed reference factors.
- **Carbon Sink Estimation**
    - Estimate absorption from existing or planned plantations using simplified assumptions.
- **Carbon Neutrality Pathways (Simulation-Based)**
    - Cleaner operations (e.g., partial electrification of vehicles).
    - Increased use of renewable energy to reduce grid dependence.
    - Afforestation-based offsets with estimated land requirements.
    - Indicative carbon credit estimation using sample market values.
- **Data Visualization**
    - Simple charts and graphs showing total emissions, sinks, gaps, and improvement scenarios.

9. GreenThumb: The Autonomous Plant Guardian – Hardware

Scenario

Priya enjoys maintaining a small balcony garden, but her work requires frequent travel. Traditional timer-based irrigation systems often overwater plants during rainy days or fail to distribute water evenly across all pots. Some

plants remain dry while others are overwatered. Priya wants a smarter solution—one that can **move between plants, assess their condition, and water them precisely based on actual need** without human intervention.

The Hackathon Challenge

Build a **Mobile Autonomous Plant Care System** that can navigate a defined area, assess plant health, and deliver the right amount of water to each plant.

The solution should:

- Navigate autonomously using a **mobile robotic base**.
- Visually identify plants and assess **basic health indicators**.
- Measure **soil moisture** directly at each plant.
- Dispense **controlled quantities of water** based on sensed conditions.

Key Features to be Implemented

- **Autonomous Navigation**
    - Wheeled or legged movement within a predefined area.
- **Plant Identification & Health Detection**
    - Use a camera to recognize plants and detect signs such as wilting or discoloration.
- **Soil Moisture Measurement**
    - Deploy a probe or sensor to measure moisture at the plant level.
- **Intelligent Water Dispensing**
    - Release water based on moisture readings and plant type.

10. PathFinder: The Smart Cane for the Visually Impaired

Scenario

Sameer is visually impaired and relies on a traditional white cane for mobility. While the cane effectively detects obstacles on the ground, it cannot sense hazards at upper-body level such as low-hanging branches, open vehicle beds,

or protruding signboards. These unseen obstacles often lead to injuries and reduce Sameer's confidence while walking independently.

Sameer needs a smarter assistive device that can **sense obstacles beyond ground level and warn him in advance**.

The Hackathon Challenge

Build a **Smart Assistive Cane Attachment** that detects obstacles at waist and head height and provides intuitive feedback to help visually impaired users navigate safely.

The solution should:

- Attach to a **standard walking cane**.
- Detect obstacles at **multiple height levels**.
- Provide **real-time feedback** to the user.
- Support an **emergency alert mechanism**.

Key Features to be Implemented

- **Obstacle Detection**
    - Use ultrasonic, LiDAR, or similar sensors to detect aerial obstacles.
- **Haptic Feedback System**
    - Provide vibration patterns indicating distance or severity of obstacles.
- **Fall or Emergency Detection**
    - Detect sudden impacts or inactivity.
- **SOS Alert Mechanism**
    - Send location or alert information to a predefined contact.

11. CropGuard: The Smart Field Sentry

Scenario

Karan is a farmer in a remote village who struggles with irregular soil moisture levels and frequent nighttime crop damage caused by wild animals. He often stays awake at night to guard his fields, which is exhausting and unsafe. He wants a system that can **automate irrigation based on soil conditions** and **detect intrusions**, alerting him immediately without requiring his physical presence.

The Hackathon Challenge

Build an **IoT-Integrated Agricultural Monitoring and Protection System** that manages irrigation intelligently and detects unauthorized animal movement in the field.

The solution should:

- Continuously monitor **soil and environmental conditions**.
- Automatically control irrigation based on crop needs.
- Detect animal intrusions in real time.
- Alert the farmer and initiate deterrent actions.

Key Features to be Implemented

- **Environmental Monitoring**
    - Use soil moisture, temperature, and humidity sensors to track field conditions.
- **Automated Irrigation Control**
    - Trigger water pumps when moisture drops below predefined thresholds.
- **Intrusion Detection**
    - Detect animal movement using PIR sensors or camera-based detection.
- **Alert & Deterrence System**
    - Activate buzzers or lights to scare animals.
    - Send SMS or app-based notifications to the farmer.

12. SafeStride: The Smart Mobility Aid

Scenario

Mrs. Kapoor is an elderly woman with limited mobility who uses a walker to move around her home. Her biggest concern is falling when she is alone, as she may not be able to call for help. Her family wants a reliable system that can **detect falls automatically**, warn about obstacles, and notify them instantly in case of an emergency.

The Hackathon Challenge

Build a **Smart Assistive Mobility Device** that enhances safety for elderly users by detecting falls, obstacles, and low-visibility conditions.

The solution should:

- Detect falls accurately.
- Alert emergency contacts automatically.
- Assist users in navigating obstacles safely.
- Improve visibility in low-light environments.

Key Features to be Implemented

- **Fall Detection**
    - Use IMU sensors (accelerometer and gyroscope) to identify fall events.
- **Emergency Alert System**
    - Send GPS location details to predefined contacts upon fall detection.
- **Obstacle Detection**
    - Use ultrasonic sensors to detect obstacles or drop-offs.
    - Provide haptic feedback through vibrations.
- **Visual Assistance**
    - Activate LED indicators in low-light conditions to prevent tripping.

13. AquaSave: The Industrial Leakage Hunter

Scenario

In a large industrial facility, small leaks in water pipelines often go unnoticed until significant damage has already occurred. These leaks waste large amounts of water and increase operational costs. The facility manager needs an **early warning system** that can detect leaks in hidden or inaccessible areas and allow rapid action to prevent losses.

The Hackathon Challenge

Build a **Networked Water Leakage Detection and Control System** that identifies leaks early and enables rapid response.

The solution should:

- Monitor water flow and pressure continuously.
- Detect leak signatures using acoustic sensing.
- Identify the probable leak zone.
- Allow remote water shut-off when required.

Key Features to be Implemented

- **Flow & Pressure Monitoring**
    - Use sensors at pipeline junctions to detect anomalies.
- **Acoustic Leak Detection**
    - Detect high-frequency sounds associated with pressurized leaks.
- **Remote Shut-Off Control**
    - Control solenoid valves through a web-based interface.
- **Zonal Leak Mapping**
    - Visualize pipeline zones and highlight suspected leak locations.

14. The Infinite Canvas: Conflict-Free Collaborative Drawing

Scenario

Students, designers, and distributed teams often collaborate using digital whiteboards during brainstorming sessions, online classes, and design reviews. However, most existing tools struggle when multiple users draw simultaneously, leading to lag, overwritten strokes, or inconsistent views. As the canvas grows and the number of drawings increases, performance drops sharply, breaking the collaborative experience.

Teams need a **real-time, infinite digital canvas** where everyone can draw, write, and add visuals together—smoothly, reliably, and without conflicts—even as the board becomes densely populated.

The Hackathon Challenge

Build a **Real-Time Collaborative Whiteboard** that supports simultaneous multi-user drawing on an infinite canvas while maintaining consistency, performance, and responsiveness.

The solution should:

- Allow multiple users to **draw, write, and place images concurrently**.
- Maintain a **shared canvas state** without conflicts or data loss.
- Support an **infinite or dynamically expanding canvas**.
- Deliver smooth performance even with large numbers of drawing elements.

Key Features to be Implemented

- **Real-Time Collaboration**
    - Synchronize drawing actions across users using WebSockets or similar real-time protocols.
- **Conflict-Free State Management**
    - Use CRDT-based approaches to ensure consistent canvas state across clients.

- **High-Performance Rendering**
    - Efficiently render thousands of vector paths using optimized Canvas techniques.
    - Apply spatial indexing (e.g., QuadTrees) and off-screen rendering.
- **Infinite Canvas Navigation**
    - Support panning and zooming across an unbounded drawing area.
- **Multi-Content Support**
    - Enable freehand drawing, text input, and image uploads on the same canvas.

15. CaptionCraft: Context-Aware Caption Generator

Scenario

Social media creators, marketers, and small businesses regularly struggle to write engaging captions for posts. While images or videos may be visually appealing, coming up with the *right words*—ones that match the tone, context, and audience—often takes more time than creating the content itself. Generic captions fail to capture attention, and manually crafting captions for every post slows down productivity.

Creators want an intelligent assistant that can **understand the content and intent** and instantly generate captions that feel natural, relevant, and platform-ready.

The Hackathon Challenge

Build an **ML-based Caption Generation System** that automatically generates meaningful, context-aware captions for social media posts.

The solution should:

- Accept **input content** such as an image, short text prompt, or keywords.
- Understand **context, tone, and intent** of the input.
- Generate **multiple caption suggestions** suitable for social platforms.
- Allow basic **tone control** (e.g., professional, casual, promotional).

Key Features to be Implemented

- **Input Processing**
    - Accept images, keywords, or short descriptions as input.
- **Caption Generation**
    - Use NLP-based models to generate coherent and engaging captions.
- **Tone & Style Control**
    - Support different tones such as informative, friendly, or marketing-focused.
- **Multiple Suggestions**
    - Generate a set of alternative captions for user selection.
- **User Interface**
    - Simple dashboard displaying inputs and generated captions with copy/export options.

END