# Question One

## Smart Borehole Monitoring System for Rural Water Management

### Problem Statement:

Many rural communities in Kenya rely on boreholes for access to clean water. These boreholes often suffer from sudden pump failures, water shortages, and inefficient maintenance due to lack of monitoring. The absence of real-time water level tracking and fault reporting makes water access unreliable. A smart system is needed to monitor borehole health, track water usage, and alert for maintenance needs to ensure sustainable water access.

### Project Task:

Design and implement a Smart Borehole Monitoring System (SBMS) with the following components:

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Identify and install sensors for water level and pump vibration monitoring
2. Assemble the sensor units using Arduino or Raspberry Pi
3. Provide a user-friendly troubleshooting guide for technicians in rural areas

#### Networking

1. Set up a GSM or Wi-Fi network (based on availability) for real-time data transmission
2. Configure IP addressing and implement a simple firewall for secure connections
3. Ensure offline data storage and sync when the network is restored

#### Fullstack Development

Build a mobile/web dashboard to:

1. Display water levels and pump status in real time
2. Send SMS/email alerts when thresholds are breached
3. Log historical data for analysis and reporting

#### IoT (Internet of Things)

1. Deploy ultrasonic or float sensors to measure water level and detect pump activity
2. Connect sensors to the internet using a microcontroller and GSM module
3. Ensure periodic data transmission to the server or cloud

#### Artificial Intelligence

1. Use AI to analyze sensor data and predict potential pump failures
2. Forecast future water availability based on current trends and seasonal patterns
3. Recommend optimal pump operation schedules to avoid dry running

#### System Administration

1. Host the dashboard and database on a Linux-based cloud server DigitalOcean, AWS, or locally
2. Configure secure SSH access, user roles, and automated daily backups
3. Monitor server logs, sensor errors, and usage metrics

# Question Two

## School ICT Lab Support Platform for Rural Schools

### Problem Statement:

Many rural schools in Kenya face challenges maintaining their ICT infrastructure due to limited technical personnel, delayed response from support teams, and lack of tracking tools. Equipment such as computers, projectors, and printers often break down and remain unused for long periods, disrupting learning. There is a need for a centralized system that allows schools to report issues, track repair status, and help support teams manage maintenance efficiently.

### Project Task:

Design and implement a School ICT Lab Support Platform (SILSP) that allows teachers or lab managers to report hardware issues, and enables technicians to respond, track, and analyze maintenance trends.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Identify common ICT equipment used in rural school labs (PCs, printers, projectors, etc.)
2. Create a fault reporting checklist for common hardware problems
3. Include a basic hardware troubleshooting guide within the system for lab assistants

#### Networking

1. Design a local area network or GSM-based connection for schools with limited infrastructure
2. Ensure secure communication between the school and support center
3. Implement basic IP management for LAN-based systems

#### Fullstack Development

1. Build a web-based or mobile platform to:
2. Submit maintenance tickets with categories and descriptions
3. Track repair status (Open, In Progress, Resolved)
4. Allow technicians to comment and update ticket status
5. Generate reports on most frequent issues, turnaround time, and equipment status

#### IoT (Internet of Things)

1. Use a simple Raspberry Pi-based monitor to auto-log basic system errors (e.g., device overheating, no boot)
2. Connect lab PCs to the system for automated reporting where possible
3. Include printer status checks via IoT-capable modules (e.g., page count, paper jam alerts)

#### Artificial Intelligence

1. Analyze historical maintenance data to:
2. Predict equipment failure patterns
3. Recommend preventive maintenance schedules
4. Highlight schools with frequent ICT issues for prioritization

#### System Administration

1. Host the platform on a local school server
2. Set up user roles (school users, technician, admin)
3. Configure system security, auto backups, and uptime monitoring
4. Enable data export features for county or ministry reporting

# Question Three

## Maternal Health Appointment and Alert System for Rural Clinics

### Problem Statement:

In rural Kenya, many expectant mothers miss antenatal visits or arrive late for delivery due to poor scheduling systems, long distances, and lack of timely reminders. Some clinics rely on paper records or verbal reminders, leading to missed critical checkups and unsafe deliveries. There is a need for a smart system that helps clinics manage appointments, send automated reminders, and educate mothers about their pregnancy stages.

### Project Task:

Design and implement a Maternal Health Appointment and Alert System (MHAAS) to support rural clinics in scheduling visits, sending reminders, and providing basic maternal health tips.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Identify and maintain clinic digital devices (e.g., tablets, printers, health record PCs)
2. Provide setup instructions and maintenance logs for clinic hardware
3. Create a health worker’s guide for using the system and troubleshooting common issues

#### Networking

1. Set up a secure network (Wi-Fi or GSM-based) for clinic systems to access centralized health databases
2. Ensure proper configuration for syncing data from rural clinics to central systems
3. Implement SMS gateway for alerts over mobile networks

#### Fullstack Development

1. Develop a mobile and web interface for:
2. Registering new mothers and pregnancy stages
3. Scheduling antenatal and postnatal appointments
4. Sending SMS reminders before visits
5. Viewing visit history and medical notes
6. Managing health worker accounts and dashboard summaries

#### IoT (Internet of Things)

1. Integrate a simple Bluetooth fetal heartbeat monitor or BP monitor and link readings to the mother’s record
2. Alert health workers if a reading is abnormal based on thresholds

#### Artificial Intelligence

1. Use AI to:
2. Predict high-risk pregnancies based on entered data (e.g., age, medical history)
3. Suggest personalized health advice based on the stage of pregnancy
4. Flag missed appointments for follow-up

#### System Administration

1. Host the application on a secure cloud server with data encryption (e.g., HTTPS, SSL)
2. Configure daily data backup, user access controls, and audit logs
3. Set up mobile-friendly access for clinics with tablets or phones

# Queston four:

## Smart Waste Bin Management System for Urban Estates and Markets

### Problem Statement:

Urban areas in Kenya, especially densely populated estates and open-air markets, struggle with uncollected waste due to poor communication between residents and waste collection services. Overflowing bins lead to health hazards, bad smells, and blocked drainage. A smart system is needed to monitor bin fill levels, notify collection teams, and track collection trends.

### Project Task:

Design and implement a Smart Waste Bin Management System (SWBMS) that uses sensors to monitor waste levels and notifies waste collection authorities in real-time.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up and maintain ultrasonic sensors in waste bins to detect fill levels
2. Use Arduino or ESP32 microcontrollers for reading and transmitting data
3. Prepare a quick maintenance guide for replacing sensors and checking battery levels

#### Networking

1. Configure GSM or Wi-Fi communication modules for sending data to the central system
2. Use IP addressing and secure messaging protocols for data transmission
3. Ensure offline storage and delayed sync for areas with poor coverage

#### Fullstack Development

1. Create a dashboard that:
2. Shows real-time status of bins (full, half, empty)
3. Notifies collection teams based on location and fill level
4. Logs collection history and patterns
5. Allows admin users to add/edit bins and assign collection routes

#### IoT (Internet of Things)

1. Deploy ultrasonic sensors inside prototype bins to detect vertical fill level
2. Integrate a microcontroller with a GSM module for data communication
3. Configure data collection intervals (e.g., every 30 minutes) and power management features

#### Artificial Intelligence

1. Analyze bin fill data to:
2. Predict peak waste generation times
3. Optimize waste collection schedules/routes
4. Alert for abnormal patterns (e.g., illegal dumping or sensor blockage)

#### System Administration

1. Deploy the dashboard on a cloud server or local municipality server
2. Set up user roles (admin, collection officer, supervisor)
3. Implement secure login, automated backups, and real-time error logging

# Question Five:

## Market Price Intelligence System for Farmers

### Problem Statement:

Small-scale farmers in Kenya often suffer losses due to lack of access to real-time market prices. Middlemen take advantage of this information gap, leading to low profits for farmers. There is a need for a digital platform that provides farmers with updated commodity prices, AI-driven price forecasts, and access to nearby markets or buyers.

### Project Task

Design and implement a Market Price Intelligence System (MPIS) that allows farmers to view current prices, analyze trends, and connect with potential buyers in their region.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up low-cost devices (e.g., tablets or smartphones) for data entry at local collection centers
2. Maintain devices used by market clerks or farmer cooperatives
3. Develop a maintenance checklist and training manual for field agents

#### Networking

1. Use GSM/4G or Wi-Fi networks to connect remote farm locations to the central database
2. Implement basic offline support with automatic sync when connectivity is restored
3. Secure data in transit using HTTPS or VPN tunneling

#### Fullstack Development

1. Build a mobile/web platform where farmers can:
2. View daily prices of crops from multiple markets
3. Upload produce availability (type, quantity, location)
4. Connect with registered buyers
5. Get notifications when prices rise/fall significantly
6. Visualize past trends via charts

#### IoT (Internet of Things)

1. Set up smart weight sensors at aggregation centers to auto-log quantity sold
2. Integrate a barcode/QR code system for produce tracking and inventory at collection points

#### Artificial Intelligence

1. Use AI models to:
2. Forecast future prices based on seasonality and past data
3. Recommend optimal sale periods to maximize profit
4. Cluster farmers and buyers for regional demand/supply analysis

#### System Administration

1. Deploy the platform on a scalable cloud server (e.g., Heroku, DigitalOcean)
2. Set up admin dashboards to manage users, price updates, and buyer verifications
3. Implement daily backups, monitoring tools, and access control systems

# Question Six

## Community Security Alert App for Neighborhood Safety

### Problem Statement:

Communities in urban and peri-urban Kenya experience increasing incidents of insecurity, including theft, assault, and vandalism. These issues are often underreported due to fear, lack of a structured system, or slow response from authorities. There's a need for a community-centered alert and reporting system that enables citizens to share real-time incident information with each other and with security agencies.

### Project Task:

Design and implement a Community Security Alert App (CSAA) that allows residents to report incidents, receive alerts, and track the response status of security teams.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up Android smartphones or tablets for use by community leaders or local security agents
2. Provide hardware maintenance guidelines (e.g., battery care, screen protection, storage expansion)
3. Train local administrators on handling device issues and app updates

#### Networking

1. Use mobile data (GSM/4G) or Wi-Fi to ensure seamless message transmission
2. Implement location-based data reporting using GPS
3. Apply encryption (HTTPS, AES) for sensitive information exchange

#### Fullstack Development

Build a Android app that allows:

1. Users to post alerts with location, description, and optional photos
2. Real-time feed of nearby incidents
3. Marking of incident resolution status
4. Admin view for verifying and escalating reports
5. Secure login for different user roles (resident, admin, security)

#### IoT (Internet of Things)

1. Connect surveillance cameras or motion sensors to the system (e.g., Raspberry Pi camera modules in shared areas)
2. Trigger alert in the app when a sensor or camera detects abnormal activity

#### Artificial Intelligence

1. Analyze reported incidents to:
2. Detect hotspots or patterns of crime over time
3. Predict areas at high risk during specific times
4. Recommend patrol schedules for security teams

#### System Administration

1. Host the app backend on a secure cloud or municipality server
2. Set up a control panel for managing users, reported content, and system logs
3. Implement logging and audit trails for all incident reports and user actions
4. Regularly back up data and update the system

# Question Seven

## Local Bus Tracking and Booking System for Urban and Rural Routes

### Problem Statement:

Commuters in many Kenyan towns and rural areas face uncertainty regarding matatu and bus schedules, routes, and availability. Overcrowding, long wait times, and inefficient route coverage are common problems. There is no centralized platform for booking seats, viewing real-time bus locations, or reporting issues. A smart system is needed to improve transport efficiency and commuter experience.

### Project Task:

Design and implement a Local Bus Tracking and Booking System (LBTBS) that enables users to track buses in real-time, book seats, and receive travel updates.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Install and maintain GPS trackers or Android phones in buses for location sharing
2. Provide power solutions (e.g., USB power banks) for devices used in public vehicles
3. Create a guide for operators on managing devices and keeping them functional

#### Networking

1. Configure GPS devices or mobile apps to transmit bus location data over GSM/4G
2. Set up a reliable backend to receive and process real-time location updates
3. Secure API endpoints for tracking, booking, and payment integration

#### Fullstack Development

1. Develop a commuter-facing mobile/web app with features to:
2. View real-time bus locations on a map
3. Book available seats and get fare estimates
4. Receive ETA and trip updates
5. Rate trips and report issues
6. Create a bus operator dashboard to manage schedules and bookings

#### IoT (Internet of Things)

1. Integrate GPS modules or use Android-based location services for real-time bus tracking
2. Collect operational metrics (e.g., speed, route completion times) for analysis
3. Send alerts if buses deviate from scheduled routes or speed limits

#### Artificial Intelligence

Use AI to:

1. Predict bus arrival times based on traffic and historical data
2. Recommend route optimizations based on demand patterns
3. Forecast peak travel times for planning

#### System Administration

1. Host the backend system on a cloud server Firebase
2. Implement user management for drivers, admins, and commuters
3. Set up automated system monitoring, backups, and alerting
4. Ensure role-based access and secure login (e.g., OTP, email verification)

# Question Eight

## Smart Classroom Attendance System for Universities and Schools

### Problem Statement:

Manual attendance tracking in classrooms can be time-consuming, inaccurate, and prone to fraud (e.g., proxy attendance). This process is particularly challenging in large university lecture halls or schools with many students. There is a need for an automated, efficient, and secure system that can track student attendance in real-time and reduce administrative burden.

### Project Task:

Design and implement a Smart Classroom Attendance System (SCAS) that automatically registers student attendance using technologies like facial recognition or RFID, and integrates with the school's student information system.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up facial recognition cameras or RFID sensors at classroom entrances
2. Provide maintenance for biometric devices, including camera calibration and sensor cleaning
3. Develop a guide for teachers and system administrators on device setup and troubleshooting

#### Networking

1. Configure the local network to connect cameras or RFID readers to the central server
2. Ensure secure data transfer between the devices and the student database
3. Implement offline functionality for remote classrooms with limited internet access

#### Fullstack Development

1. Create a web/mobile dashboard for:
2. Automatically detecting and marking students’ attendance based on facial recognition or RFID tags
3. Viewing attendance records, generating reports, and sending notifications
4. Allowing instructors to view real-time class presence and summary statistics
5. Providing students with a view of their attendance history and notifications on absences

#### IoT (Internet of Things)

1. Set up IoT sensors (e.g., RFID or cameras) to capture attendance as students enter the classroom
2. Configure IoT-enabled systems to sync data with a central cloud or local server
3. Implement a simple user interface for students to manually mark attendance in case of sensor failure

#### Artificial Intelligence

1. Use AI algorithms for facial recognition (e.g., OpenCV with a deep learning model) or integrate an existing AI-based API (e.g., Microsoft Azure or Google Vision) for accurate attendance detection
2. Use AI to detect patterns of absenteeism and alert administrators to potential issues (e.g., low attendance rates for specific students)
3. System Administration
4. Host the backend on a cloud or university server with data encryption
5. Set up automated attendance tracking, data backups, and scheduled updates
6. Implement role-based access for admins, instructors, and students, ensuring data privacy and security

# Question Nine

## Remote Medical Diagnostics System Using Artificial Intelligence

### Problem Statement:

Many people in remote areas of Kenya lack access to medical professionals, especially for early diagnosis of common illnesses. Clinics may be understaffed, and patients often rely on unverified home remedies or delay treatment. There is a need for a simple, AI-powered platform that can provide preliminary diagnoses based on symptoms and basic health inputs.

### Project Task:

Design and implement a Remote Medical Diagnostics System (RMDS) that allows users to input symptoms, receive AI-based diagnostic suggestions, and connect to nearby health providers.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Equip health posts or community centers with tablets or smartphones for system access
2. Create a simple setup guide for health workers or volunteers to manage the devices
3. Provide routine maintenance instructions for basic diagnostic equipment (e.g., thermometers, BP monitors)

#### Networking

1. Use GSM/4G or Wi-Fi for syncing patient data to a central server or cloud
2. Implement secure and encrypted communication between clients and servers
3. Support intermittent connectivity through offline-first architecture

#### Fullstack Development

1. Build a mobile/web application that:
2. Allows patients to select symptoms from a guided interface
3. Upload optional data like body temperature, images (e.g., rashes), or audio recordings (e.g., cough sounds)
4. Receives an AI-generated probable diagnosis and advice
5. Provides a map or list of nearby clinics and pharmacies
6. Logs diagnostic sessions for further consultation

#### IoT (Internet of Things)

1. Integrate with smart thermometers, pulse oximeters, or BP monitors to automatically capture patient data
2. Enable Bluetooth data transfer from low-cost health devices to the mobile app

#### Artificial Intelligence

1. Train or use a pre-trained model to:
2. Predict likely diagnoses based on symptoms and sensor data
3. Recommend next steps (e.g., self-care, visit doctor, emergency)
4. Flag high-risk conditions for urgent referral
5. Incorporate natural language processing (NLP) for understanding text or voice input (optional)

#### System Administration

1. Host the platform on a secure cloud server with HIPAA-like best practices for privacy
2. Set up data logs, encryption (at rest and in transit), and access control
3. Create role-based accounts for users, community health workers, and doctors
4. Schedule automated system health checks and backups

# Question Ten

## Digital Land Ownership Mapping System for County Governments

### Problem Statement:

In Kenya, land ownership disputes are common due to lack of digitized records, unverified land boundaries, and difficulty accessing official documents. Many rural areas rely on paper-based records which are vulnerable to loss or fraud. There is a need for a digital mapping and registration system to help county governments manage land data securely and transparently.

### Project Task:

Design and implement a Digital Land Ownership Mapping System (DLOMS) that digitizes land records, visualizes land parcels on a map, and provides secure access for verification and dispute resolution.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up and maintain data entry computers, tablets, and GPS surveying tools for field officers
2. Provide device maintenance guides and checklists for use during field mapping
3. Train land registry staff on backup procedures and basic troubleshooting

#### Networking

1. Establish secure networks between local land offices and central county data centers
2. Ensure data synchronization between mobile devices and servers, even with intermittent connections
3. Use secure communication protocols (HTTPS, SSL, VPN) for all data transactions

#### Fullstack Development

1. Build a web-based platform that:
2. Allows officials to register land parcels with coordinates, ownership details, and supporting documents
3. Visualizes parcels on an interactive map using GIS integration
4. Enables users to search and verify land ownership data
5. Logs change and provide downloadable ownership certificates

#### IoT (Internet of Things)

1. Integrate GPS receivers for accurate land boundary mapping during surveys
2. Automatically upload field-collected geolocation data to the land registry server

#### Artificial Intelligence

1. Use AI to:
2. Detect inconsistencies in land parcel records (e.g., overlaps or double ownership)
3. Analyze historical transactions for fraud or unusual patterns
4. Recommend likely boundaries using satellite imagery and GIS data (optional)

#### System Administration

1. Host the system on a secure county government server with daily backups
2. Implement role-based access for data clerks, surveyors, administrators, and the public
3. Log all changes to land records with timestamps and user IDs for transparency
4. Provide options for offline data collection and later synchronization

# Question Eleven

## Smart Poultry Farm Monitoring System for Smallholder Farmers

### Problem Statement:

Smallholder poultry farmers in Kenya face challenges in monitoring and controlling the farm environment, which affects poultry health and productivity. Manual checks for temperature, humidity, feeding, and water levels are inefficient and lead to losses. There is a need for a smart system that enables real-time monitoring, alerting, and basic automation to improve poultry farm efficiency.

### Project Task:

Design and implement a Smart Poultry Farm Monitoring System (SPFMS) that uses sensors to track environmental conditions and notifies farmers of critical changes.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up and maintain sensors (temperature, humidity, feed/water level)
2. Assemble and connect the system using Raspberry Pi or Arduino boards
3. Provide a maintenance manual for sensor calibration, wiring, and enclosure protection

#### Networking

1. Use Wi-Fi or GSM to transmit sensor data to a central dashboard or mobile phone
2. Secure sensor data using basic authentication and encryption
3. Provide offline data logging and synchronization once the network is restore

#### Fullstack Development

Build a mobile/web interface that allows:

1. Real-time display of poultry house conditions (e.g., temperature, humidity)
2. SMS/Push alerts when thresholds are exceeded
3. Graphs and history logs of environmental data
4. Manual data entry for vaccination, feeding schedules, and mortality records

#### IoT (Internet of Things)

Deploy IoT sensors to measure:

1. Temperature and humidity for heat stress control
2. Feed and water container levels
3. Light control for poultry behavior regulation
4. Interface the sensor system with a microcontroller that sends data to the server/app

#### Artificial Intelligence

Use AI models to:

1. Predict risk of poultry diseases based on environmental trends
2. Suggest optimal feeding or ventilation adjustments
3. Forecast growth or egg-laying rates based on historical farm data

#### System Administration

1. Host the backend system on a cloud server or local farm server (Raspberry Pi)
2. Set up alerts for device failures or sensor disconnects
3. Implement user roles for farm workers, supervisors, and system admins
4. Ensure regular backups and performance monitoring

# Question Twelve

## Energy Usage Monitoring Dashboard for Homes and Schools

### Problem Statement:

Many households and educational institutions in Kenya experience high electricity costs due to unmonitored or inefficient power use. Without visibility into energy consumption patterns, it is difficult to identify energy wastage or plan for solar or backup options. A smart monitoring system is needed to track energy usage, analyze consumption trends, and promote responsible energy use.

### Project Task:

Design and implement an Energy Usage Monitoring Dashboard (EUMD) that collects, analyzes, and visualizes electricity usage data in real time for homes and schools.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Install and maintain smart plugs or energy meters (e.g., current sensors like SCT-013 or INA219 modules)
2. Assemble hardware using microcontrollers such as Arduino or ESP32
3. Provide a maintenance checklist for power sensor calibration, wiring, and safety inspections

#### Networking

1. Configure Wi-Fi or GSM modules for real-time data upload to a central dashboard
2. Ensure data encryption and network resilience during blackouts or connection loss
3. Implement local data caching in case of temporary network failures

#### Fullstack Development

Build a web/mobile platform that:

1. Displays real-time and historical electricity consumption
2. Shows usage by time of day, device, or room (if sensors are distributed)
3. Sends notifications for unusual spikes in consumption
4. Allows users to set energy-saving goals or usage limits

#### IoT (Internet of Things)

1. Deploy current/voltage sensors on mains lines or specific sockets
2. Transmit sensor data periodically to the server
3. Allow remote control of smart plugs (on/off) through the dashboard

#### Artificial Intelligence

Use AI models to:

1. Detect abnormal consumption patterns (e.g., faulty appliances)
2. Predict monthly electricity bills based on usage trends
3. Recommend energy-saving actions (e.g., reducing peak usage times)

#### System Administration

1. Host the dashboard on a secure cloud server or local school/home server
2. Set up user access for homeowners, facility managers, and technicians
3. Implement automated data backup, usage logging, and performance monitoring
4. Create a role-based access system to protect sensitive energy data

# Question Thirteen

## e-Waste Collection and Tracking System for Urban and Rural Kenya

### Problem Statement:

Kenya generates increasing amounts of electronic waste (e-waste) from discarded phones, computers, TVs, and other gadgets. Most e-waste ends up in landfills or is disposed of unsafely, posing health and environmental risks. There is no streamlined way for citizens to report or dispose of e-waste responsibly. A digital system is needed to manage e-waste collection points, track disposal, and raise public awareness.

### Project Task:

Design and implement an e-Waste Collection and Tracking System (EWCTS) that allows citizens to report e-waste, view drop-off locations, and enable authorities or recyclers to manage collections.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up and maintain devices (e.g., tablets, Android phones) used by collection agents or kiosks
2. Install QR/barcode scanners for tracking e-waste items
3. Provide manuals for managing basic device and scanner issues

#### Networking

1. Use GSM or Wi-Fi to connect field agents to the central system
2. Ensure secure and reliable data synchronization between users and the server
3. Apply basic data encryption to protect identity and location of contributors

Fullstack Development

Create a mobile/web app that enables users to:

1. Locate nearby e-waste drop-off points on a map
2. Submit item types, quantity, and photos for disposal
3. Receive reminders for community collection days
4. View stats on how much e-waste has been recycled

Develop an admin dashboard for recyclers or local governments to:

1. Approve drop-off locations
2. Track collection status
3. Analyze trends by region or item type

#### IoT (Internet of Things)

1. Integrate smart bins that notify authorities when they are full
2. Use GPS trackers for e-waste collection vehicles
3. Install sensors to monitor the temperature or condition of collected batteries

#### Artificial Intelligence

Use AI to:

1. Classify uploaded item photos into e-waste categories (e.g., phones, batteries, computers)
2. Predict e-waste hotspots based on previous drop-off data
3. Recommend optimal collection routes for recyclers based on submissions

#### System Administration

1. Deploy the backend system on a cloud
2. Create user roles for public users, recyclers, drop-off agents, and administrators
3. Enable system logs, automated data backup, and user analytics
4. Implement a public API for NGOs or county governments to access anonymized data

# Question Fourteen

## Local Government Complaint Management System (LGCMS)

### Problem Statement:

Many citizens in Kenya lack a reliable and transparent way to report local service delivery issues such as water outages, road damage, corruption, or harassment. Paper-based reporting systems are slow, non-transparent, and prone to manipulation or neglect. There is a need for a digital platform to allow citizens to submit, track, and follow up on complaints lodged with local government authorities.

### Project Task:

Design and implement a Local Government Complaint Management System (LGCMS) that enables structured reporting, efficient follow-up, and data-driven transparency in local governance.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Deploy and maintain digital kiosks or tablets at chief’s offices or county offices for residents without smartphones
2. Provide a checklist for maintaining devices used for reporting and follow-up
3. Train ward administrators on how to troubleshoot and update devices

#### Networking

1. Configure secure mobile or Wi-Fi connections to sync complaints to the central database
2. Ensure encrypted transmission of sensitive complaint information
3. Implement fallback syncing mechanisms in case of internet failure

#### Fullstack Development

Build a web and mobile platform with features to:

1. Allow users to submit complaints with categories (e.g., sanitation, water, corruption)
2. Attach evidence (photos, video, documents)
3. Track complaint status (Submitted, In Review, Resolved)
4. Enable two-way feedback (citizen and government response)
5. View statistics (resolved, pending, most common issues)

Admin dashboard for county staff to:

1. Assign and escalate cases
2. Monitor performance of different departments
3. Generate reports and export complaint logs

#### IoT (Internet of Things)

1. Deploy smart sensors to automatically report certain issues (e.g., water level sensors triggering low water complaints)
2. Install environmental sensors (e.g., air quality, noise pollution) tied to public health complaint categories

#### Artificial Intelligence

1. Use NLP to auto-categorize and route complaints based on text input
2. Analyze complaint trends by ward, sub-county, or issue type
3. Predict service failure patterns (e.g., frequent sewer blockage in a region)

System Administration

1. Host the platform on a secure server
2. Set up role-based access control (citizens, sub-county admins, county directors)
3. Enable audit logs for transparency and accountability
4. Automate daily backups, system uptime alerts, and data recovery procedures

# Question Fifteen

## Youth Skills Marketplace Platform for Local Employment

### Problem Statement:

Many skilled youths in Kenya especially in rural and peri-urban areas struggle to find work or showcase their capabilities. At the same time, individuals and small businesses need services like plumbing, graphic design, web development, or tailoring but lack a centralized way to find and verify local talent. A digital platform is needed to connect skilled youth with job opportunities while promoting trust and visibility.

### Project Task:

Design and implement a Youth Skills Marketplace Platform (YSMP) where youth can register, showcase their skills, and connect with clients needing services.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Set up computer labs or kiosks at youth centers for profile registration and browsing jobs
2. Provide setup, maintenance, and troubleshooting guides for shared public devices
3. Train youth center staff on how to maintain the platform hardware (e.g., fingerprint scanners for ID verification)

#### Networking

1. Use internet or GSM connectivity to sync user profiles and job listings in real-time
2. Ensure privacy and security of communications between clients and service providers
3. Enable offline profile creation and sync when network is available (for rural centers)

#### Fullstack Development

Create a responsive web/mobile platform that allows:

1. Youth to create profiles with bio, skill tags, portfolio uploads (photos, videos, PDFs)
2. Clients to post jobs and browse by skill, location, and ratings
3. Ratings, reviews, and success tracking after job completion
4. Integration with mobile money for secure payments

Admin panel to:

1. Approve user registrations
2. Manage job categories, disputes, and payments
3. Generate analytics and reports

#### IoT (Internet of Things)

1. Use QR codes for identity verification at job locations
2. Biometric verification (e.g., fingerprint scanner) at public access points to prevent duplicate accounts

#### Artificial Intelligence

Use AI to:

1. Recommend jobs to youth based on their skillsets and location
2. Match clients with top-rated or nearby service providers
3. Detect spam, scams, or repeated negative reviews using sentiment analysis

#### System Administration

1. Host the platform on a secure server
2. Set up role-based access for youth, clients, and admins
3. Enable real-time backups, moderation tools, and data analytics
4. Provide reporting tools for tracking youth employment trends and skills demand

# Question Sixteen

### Disaster Early Warning and Alert System for Flood-Prone Areas

### Problem Statement:

Many regions in Kenya especially in low-lying and riverine areas are vulnerable to flooding. Communities often receive warnings too late, resulting in loss of life, property, and disruption of livelihoods. There is a need for a localized, technology-driven early warning system that can detect and predict disasters like floods, and send timely alerts to affected populations.

### Project Task:

Design and implement a Disaster Early Warning and Alert System (DEWAS) that collects environmental data, detects risk conditions, and disseminates real-time alerts to communities and response teams.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Deploy and maintain water level sensors, rain gauges, and GSM modules in flood-prone areas
2. Use microcontrollers (e.g., Arduino/ESP32) to collect sensor data
3. Provide manuals for maintaining sensors and replacing parts during breakdown

#### Networking

1. Set up GSM-based or Wi-Fi networks to send real-time sensor data to a central dashboard
2. Ensure alerts can be triggered automatically and sent via SMS, USSD, or mobile app
3. Secure data flow from sensor nodes to the backend using SSL or VPN

#### Fullstack Development

Create a dashboard that:

1. Displays current environmental readings (water levels, rainfall intensity, etc.)
2. Triggers alerts when thresholds are crossed
3. Allows admin input from local authorities (manual overrides, announcements)
4. Sends location-based SMS/notification alerts to registered users

Mobile app/web portal for the public to:

1. Subscribe to alerts
2. View nearby flood zones
3. Access evacuation plans or emergency contacts

#### IoT (Internet of Things)

1. Install water level sensors near rivers or dams
2. Program microcontrollers to send regular readings (e.g., every 10 minutes)
3. Connect with solar-powered modules for continuous operation in remote areas

#### Artificial Intelligence

Train AI models to:

1. Forecast floods based on weather, historical, and sensor data
2. Provide risk scoring for different villages or wards
3. Recommend evacuation timing and safety zones

System Administration

1. Host the system on a server
2. Implement user management for residents, responders, and administrators
3. Create audit logs, backup protocols, and incident response templates
4. Monitor system uptime and sensor health

# Question Seventeen

## Digital Mental Health Support Platform for Youth and Rural Communities

### Problem Statement:

Mental health challenges among Kenyan youth are on the rise, yet stigma, lack of trained counselors, and limited access to support services prevent many from seeking help. Rural and underserved areas are especially affected. There is a need for a confidential, accessible, and supportive digital platform where individuals can get mental health resources, self-assess, and connect to professionals or trained volunteers.

### Project Task:

Design and implement a Digital Mental Health Support Platform (DMHSP) that offers self-care tools, anonymous chat support, booking with mental health providers, and community mental health education.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Equip youth centers and community health facilities with tablets or smartphones for mental health access points
2. Maintain public kiosks for safe, private access in rural areas
3. Train staff or volunteers to support users in navigating the platform and handling device issues

#### Networking

1. Use Wi-Fi or GSM to ensure continuous, secure access to the platform
2. Support low-bandwidth modes for areas with weak connections
3. Ensure data privacy using end-to-end encryption (e.g., HTTPS, SSL/TLS)

#### Fullstack Development

Build a responsive web/mobile app with features including:

1. Mental health self-assessment quizzes (depression, anxiety, stress)
2. Anonymous chat with trained peer counselors or AI-powered bots
3. Resource hub with videos, articles, and coping techniques
4. Appointment booking with professional counselors
5. Emergency contact and crisis alert button

#### IoT (Internet of Things)

1. Integrate wearable health data (e.g., heart rate, sleep pattern from fitness bands) to detect stress trends
2. Monitor mood via facial expressions or typing behavior with consent

#### Artificial Intelligence

Use AI to:

1. Provide conversational therapy or coping advice using NLP
2. Analyze mood from text/speech input to assess distress levels
3. Recommend personalized content based on usage and assessments

#### System Administration

1. Host the platform on a HIPAA-compliant or locally secure server
2. Set up admin roles for content managers, mental health professionals, and system auditors
3. Enable secure login, data encryption, and anonymized user logs
4. Provide data dashboards to help policy makers understand youth mental health trends

# Question Eighteen

## Local Climate Monitoring and Advisory System for Farmers

### Problem Statement:

Farmers in Kenya are increasingly affected by unpredictable weather patterns due to climate change, resulting in crop losses and reduced food security. Many rely on outdated information or traditional knowledge without access to real-time localized forecasts or actionable climate advice. There is a need for a digital platform that integrates localized weather monitoring with practical, timely agricultural advisories.

### Project Task:

Design and implement a Local Climate Monitoring and Advisory System (LCMAS) to collect local climate data, predict trends, and deliver tailored agricultural guidance to farmers.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Deploy and maintain low-cost weather stations (e.g., for rainfall, temperature, humidity, wind)
2. Use Raspberry Pi/Arduino with sensors for weather data collection
3. Train local extension officers to maintain devices and perform routine calibration

#### Networking

1. Set up GSM or LoRaWAN modules to transmit sensor data from farms to a central server
2. Support offline data storage with batch sync for areas with poor connectivity
3. Secure all data transfers using HTTPS and encrypted APIs

#### Fullstack Development

Develop a mobile/web dashboard that allows:

1. Farmers to view real-time weather data for their area
2. Receive SMS/Push alerts for extreme weather conditions (e.g., drought, frost, heavy rain)
3. Access AI-generated tips on when to plant, irrigate, or harvest
4. View historical climate trends for decision-making

#### IoT (Internet of Things)

1. Use environmental sensors (e.g., DHT22, rain gauge, UV sensor) to collect localized data
2. Send periodic weather readings (e.g., hourly) to the cloud
3. Allow sensor deployment on demo farms or school-based agricultural programs

#### Artificial Intelligence

Use AI models to:

1. Forecast weather patterns based on sensor and satellite data
2. Recommend crop planting dates or pest control schedules
3. Detect early signs of climate stress or anomaly patterns across seasons

System Administration

1. Host the system on a secure and scalable cloud platform (e.g., AWS, Azure, or Kenya-hosted servers)
2. Implement access controls for farmers, extension officers, and system admins
3. Provide system uptime monitoring, automated backups, and fault alerts
4. Enable multilingual support (e.g., Kiswahili, local dialects)

# Question Nineteen

## Public Sanitation Monitoring App for Urban Informal Settlements

### Problem Statement:

In many informal settlements across Kenya, public toilets and sanitation facilities are limited, poorly maintained, or entirely non-functional. This leads to outbreaks of waterborne diseases, especially among children and women. There is no real-time system for residents to report issues or for local authorities to monitor sanitation conditions. A smart solution is needed to improve visibility and responsiveness.

### Project Task:

Design and implement a Public Sanitation Monitoring App (PSMA) to report, track, and manage the state of shared sanitation facilities in urban and peri-urban areas.

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Equip toilets with QR codes or NFC tags to enable mobile reporting
2. Set up smartphones or tablets for facility caretakers or ward officers
3. Provide guides for maintaining these devices, including battery and data management

#### Networking

1. Use mobile data or Wi-Fi to submit reports to a centralized sanitation dashboard
2. Ensure data encryption for incident reports and user identities
3. Enable SMS-based reporting for users without smartphones

#### Fullstack Development

Build a mobile/web app that allows:

1. Residents to report issues (e.g., "toilet full", "no water", "damaged tap")
2. Upload photos and tag facility locations
3. See the operational status of nearby facilities (clean, dirty, under repair)
4. Get SMS updates when a facility is cleaned or fixed

Admin dashboard to:

1. Assign cleaning and repair tasks
2. Track response times and status by ward or contractor
3. View analytics (e.g., most frequently reported issues, high-traffic areas)

#### IoT (Internet of Things)

1. Install waste level or usage sensors in selected public toilets
2. Send automatic alerts when tanks are full or need cleaning
3. Log cleaning activity using sensor-based verification

Artificial Intelligence

1. Analyze report frequency and patterns to:
2. Predict which facilities need more frequent maintenance
3. Detect neglect or failure patterns by service providers
4. Recommend reallocation of resources (e.g., moving mobile toilets to higher-need areas)

#### System Administration

1. Host the system on a secure municipal or NGO cloud platform
2. Set up access for residents, cleaners, contractors, and county officials
3. Schedule daily backups, logging, and alert systems for delayed responses
4. Provide reporting and exporting tools for public health teams

# Question Twenty

## Smart Agricultural Support System for Small-Scale Farmers in Rural Kenya

### Problem Statement:

Small-scale farmers in rural Kenya often face challenges such as unpredictable weather, poor access to agricultural support services, and limited market information. There is a need for a smart system that can assist these farmers in maintaining equipment, getting real-time farming tips, predicting weather patterns, and accessing markets.

### Project Task:

Design and implement a Smart Agricultural Support System (SASS) with the following components:

### Project Requirements by Module

#### Hardware Maintenance and Support

1. Identify and maintain IoT sensors (e.g., soil moisture sensors, weather stations)
2. Assemble and test sensor kits using Raspberry Pi or Arduino
3. Create a troubleshooting guide for farmers

#### Networking

1. Set up a reliable local network (wireless or GSM-based) to connect IoT devices to a central system
2. Secure the network using basic firewalls and IP addressing

#### Fullstack Development

1. Develop a mobile/web dashboard where farmers can:
2. Register farms
3. View sensor data (e.g., soil moisture, temperature)
4. Access AI-based recommendations
5. Connect to local buyers or cooperatives

#### IoT (Internet of Things)

1. Deploy at least two sensors (e.g., soil pH, moisture, temperature) in a demo farm setup
2. Collect and transmit data to the cloud or local server
3. Integrate alert systems (e.g., SMS alerts for low moisture)

#### Artificial Intelligence

1. Train a basic ML model to predict crop disease based on sensor readings and uploaded leaf images
2. Provide smart tips for irrigation or fertilizer use based on past data

#### System Administration

1. Deploy the backend on a Linux server (cloud or local)
2. Configure security settings (SSH, users, cron jobs)
3. Monitor system logs and create a backup routine