# Problem Statement 1: Intelligent AI-Powered HR Recruitment Assistant

**Background** Modern organizations receive thousands of CVs stored in their HRMS (Human Resource Management System). When a new job opening is created, HR teams spend countless hours manually searching through resumes to find suitable candidates. This process is time-consuming, error-prone, and often overlooks strong candidates due to a lack of structured filtering and intelligent matching.

**Challenge** Design and develop an **AI-powered HR recruitment assistant** that integrates with an existing HRMS to automatically analyze the database of CVs when a new job post is created.

**Key Features**

1. **Automated CV Screening**:
   * When a new job is posted, the system should instantly identify and shortlist relevant CVs from the existing database.
   * Example: If 2,000 CVs exist, and 120 match closely with the job requirements, the system highlights those.
2. **Semantic Matching & Similarity Scoring**:
   * Use Natural Language Processing (NLP) and semantic analysis (via open-source AI models) to match CVs against job descriptions.
   * Consider parameters such as skills, education, universities, years of experience, and recency of the CV.
   * Provide similarity scores or percentage matches for HR to quickly evaluate.
3. **Intelligent Insights**:
   * Highlight candidate strengths: key skills, unique qualifications, and relevant experiences.
   * Provide a breakdown of why a CV was selected (e.g., “5 years in Java development, Master’s in Computer Science, matches required skills in cloud computing and APIs”).
4. **User-Friendly Dashboard**:
   * Display shortlisted candidates in an intuitive interface.
   * Allow HR managers to filter further (by location, graduation year, certifications, etc.).

**Expected Outcomes**

* Save time and effort in the recruitment process.
* Increase accuracy in matching candidates to jobs.
* Provide transparent, explainable AI-based decision support for HR managers.

**Student Tasks**

* Research and select appropriate open-source AI/NLP models (e.g., BERT, Sentence Transformers).
* Preprocess and clean CV text data (handling different formats like PDF, Word, plain text).
* Build algorithms for similarity scoring between job descriptions and CVs.
* Develop a prototype interface that demonstrates how HR can interact with the system.

**Deliverables**

* A working prototype of the AI-powered HR recruitment assistant.
* A short presentation explaining the architecture, model selection, and limitations.
* Documentation of future improvements, such as bias detection, multilingual CVs, and integration with LinkedIn profiles.

# Problem Statement 2: Voice-to-Voice AI Assistant for Universities and Enterprises

**Background** Websites today often struggle to provide clear, immediate, and interactive support for visitors. Students applying to universities, for example, may have to browse dozens of pages, PDFs, and FAQs to find basic answers about admissions, scholarships, fees, or hostel facilities. Similarly, businesses often maintain large sets of internal documents, Excel sheets, and knowledge bases that are difficult for users to navigate.  
 Traditional chatbots rely on rigid, rule-based scripts and fail to offer natural, human-like conversations. There is a need for a **voice-driven, AI-powered assistant** that can understand context, access structured and unstructured information, and respond in a natural, multilingual voice.

**Challenge** Design and develop a **Voice-to-Voice AI Assistant** that can be integrated with any website or enterprise portal to act as a conversational consultant.

**Key Features**

1. **Voice-to-Voice Interaction**

* Users can speak directly to the assistant in any supported language.
* The assistant responds back in voice, creating a two-way, natural dialogue.
* Multilingual support: English, Urdu, Arabic, and more.

1. **Dynamic Knowledge Integration**

* The assistant combines general AI knowledge (from models like ChatGPT, Gemini, DeepSeek, or open-source LLMs) with specific knowledge bases provided by the host organization.
* These sources may include:
* Website pages
* FAQs
* Policy documents
* Excel sheets and structured databases
* Example: A student visiting Namal University’s site can ask, “What is the admission deadline for Fall 2025?” and get an instant, spoken answer drawn directly from the university’s admission office documents.

1. **Domain-Specific Consultation**

* For universities: admissions, courses, faculty, scholarships, hostel facilities, and fee structures.
* For enterprises: product details, HR policies, customer support, and compliance documentation.

1. **Seamless Integration**

* Designed as an embeddable module for websites and portals.
* Can connect with APIs, document repositories, and databases for real-time updates.

1. **Explain the ability and Transparency**

* Assistant should cite the source of its answer (e.g., “According to the Admissions Handbook 2025…”).
* Provide both spoken and text transcript of the conversation for accessibility.

**Expected Outcomes**

* Enhanced user experience on websites with 24/7 interactive support.
* Increased accessibility through multilingual and voice-based communication.
* Reduced workload for human support staff by handling repetitive queries.
* Stronger trust in AI systems due to transparency of sources.

**Student Tasks**

* Research and integrate speech-to-text (STT) and text-to-speech (TTS) technologies.
* Explore open-source LLMs and evaluate their suitability for domain-specific fine-tuning.
* Develop a system to ingest and index documents, webpages, and spreadsheets into the knowledge base.
* Build a prototype web integration showing how the assistant can be deployed on a university website.

**Deliverables**

* A working prototype of the voice-to-voice AI assistant.
* Demonstration of real-time Q&A using a provided dataset (e.g., university admission docs).
* Documentation describing architecture, technology choices, and integration methods.

·Recommendations for future improvements, such as emotion recognition, video avatar integration, or cross-platform deployment (mobile + web).

# Problem Statement 3: AI-Powered Human vs. Bot Authentication System

**Background**Artificial Intelligence is increasingly shaping our future, bringing both opportunities and risks. Alongside useful applications, AI is also being exploited in harmful ways. Malicious AI agents and bots can act like humans online, performing phishing attacks, stealing information from browsers, reading documents, or even impersonating real users. Traditional security measures such as CAPTCHA or simple password checks are no longer sufficient to detect advanced AI-driven bots.

To safeguard digital interactions, there is a need for a **robust AI-based identity inspection system** that can differentiate between humans and bots across devices, browsers, and applications.

**Challenge** Design and develop an **AI-driven security assistant** (extension/tool) that can be integrated with any device or web system. This tool should perform real-time

inspections whenever someone attempts to access a device, application, or website, and determine whether the user is a genuine human or an automated bot/agent.

**Key Features**

1. **AI Inspection Layer**
   * Every interaction with a website, application, or device passes through the inspection layer.
   * The system analyzes patterns of interaction: typing speed, mouse movement, touch behavior, voice, or programming/scripted activity.
2. **Multi-Modal Verification**
   * Use multiple biometric and behavioral parameters:
     + **Thumbprint recognition**
     + **Face recognition**
     + **Voice recognition**
     + **Keystroke dynamics** (typing rhythm)
     + **Mouse/touch gestures**
   * Combine signals to make a strong judgment of authenticity.
3. **Trusted Device Certification**
   * Once a device is verified, it receives a secure **“Human-Verified Badge”** linked to its MAC address or unique device ID.
   * This badge grants ongoing access to approved systems.
4. **Continuous Monitoring & Re-Verification**
   * The system continuously observes device behavior and user interaction patterns.
   * If unusual or suspicious activity is detected (e.g., sudden changes in typing speed, scripted behavior, or location anomalies), the badge is flagged.
   * In such cases, the device must **re-verify through biometric checks** before regaining trusted status.
5. **Integration & Extensibility**
   * Deployable as a browser extension, OS-level assistant, or API for websites.
   * Could integrate with enterprise security frameworks, educational portals, or e-commerce sites.
6. **Security and Privacy**
   * System should prioritize user privacy by storing biometric and behavioral data securely.
   * Must ensure compliance with data protection standards (GDPR-style guidelines).

**Expected Outcomes**

* Enhanced protection against AI-based attacks and impersonation attempts.
* A future-ready security mechanism that goes beyond traditional CAPTCHAs.
* A scalable solution applicable to universities, banks, enterprises, and governments.

**Student Tasks**

* Research methods of detecting AI agents vs. human interactions.
* Implement at least two biometric checks (e.g., voice + typing pattern).
* Build a prototype browser extension or standalone application that tags devices as human-verified.
* Develop anomaly detection to trigger re-verification when suspicious patterns are observed.
* Simulate attacks using bots/agents to test detection accuracy.

**Deliverables**

* A working prototype of the AI Human-vs-Bot Inspector.
* Demonstration of device verification, badge issuance, and re-verification on anomaly detection.
* Documentation describing system design, detection algorithms, and limitations.
* A proposal for future enhancements, such as integration with blockchain for secure device tagging.

# Problem Statement 4: Offline-Capable Voice-to-Voice AI Assistant for Farmers

**Background** AI assistants like ChatGPT are transforming the way people learn and interact. However, in Pakistan and many developing regions, their adoption faces two major barriers:

1. **Internet Dependency**: Most AI assistants require continuous online connectivity.
2. **Accessibility & Language Barriers**: Many farmers, workers, or rural citizens cannot type in English but could benefit from voice-based, local-language interactions.

Farmers in particular need real-time, trustworthy guidance about crops, soil, weather, seeds, pests, and medicine usage. Yet they lack digital solutions that are both **affordable and offline-capable**, allowing them to interact in their own languages, such as Punjabi, Urdu, or Sindhi.

**Challenge** Design and develop a **Voice-to-Voice AI Assistant Mobile App** that can work **both online and offline**. The assistant should serve as a guiding companion for farmers, offering agricultural advice, weather predictions, and practical recommendations in local languages.

**Key Features**

1. **Voice-to-Voice Interaction in Local Languages**
   * Farmers can speak in Punjabi, Urdu, Sindhi, or other local languages.
   * The assistant replies back in voice for accessibility (no need for reading/writing).
2. **Offline Knowledge Base**
   * The app stores a large library of agricultural data, including:
     + Crop cultivation guidelines
     + Common plant and soil diseases
     + Pest management
     + Fertilizer and pesticide usage
   * Farmers can query the system offline and receive instant advice.
3. **Online Sync & Personalization**
   * When internet access is available, the app syncs new data (weather updates, new farming practices, etc.).
   * The assistant adapts over time to the user’s behavior and frequently asked questions, personalizing guidance for their crops and region.
4. **Weather Integration**
   * The assistant provides weather forecasts when online.
   * It advises on actions based on weather (e.g., “Heavy rain expected tomorrow, delay irrigation”).
5. **Domain-Specific Assistant**
   * Focused first on farmers, but extensible to other domains (medical staff, shopkeepers, field workers).
   * Could evolve into a general-purpose, multilingual assistant for underserved communities.
6. **Usability & Accessibility**
   * Simple mobile app interface, designed for low-literacy users.
   * Visual icons and voice-first navigation (minimal typing required).
   * Lightweight enough to run on low-cost Android phones.

**Expected Outcomes**

* An affordable, offline-capable AI assistant that reduces dependency on premium services and internet availability.
* Increased agricultural productivity and informed decision-making for farmers.
* Empowerment of non-literate and rural populations through technology.

**Student Tasks**

* Build a prototype mobile application with offline and online modes.
* Integrate an offline speech-to-text (STT) and text-to-speech (TTS) system for at least one local language.
* Create a small offline agricultural knowledge base (crops, diseases, remedies).
* Implement weather API integration for online mode.
* Design the system to learn from user interactions and adapt over time.

**Deliverables**

* A working prototype of the offline-first voice-to-voice assistant.
* Demo in Punjabi or Urdu for common farming queries.
* Documentation explaining architecture (offline + online modules), model choices, and challenges.
* A proposal for scaling the system with larger datasets and multi-domain use.

# Additional Ideas

1. Performix

**Introduction**

The **“Perfomix”** will be a comprehensive solution for organizations seeking to efficiently manage employee performance. The system will have three main user roles: Admin, Line Manager, and Staff, each equipped with role-specific functionalities.

See full detail explanation here: [Performix](https://docs.google.com/document/d/1-yUDbuXAFRsnPrbMgLh4oRDJcacNhzVaI9XewSFZGRk/edit?tab=t.0)

1. ProjectIn

**Introduction**

The **“ProjectIn”** will be a comprehensive web-based solution for organizations seeking to efficiently manage tasks. The system will have three main user roles: Admin, Manager, and Staff, each equipped with role-specific functionalities.

See full detail explanation here: [ProjectIn](https://docs.google.com/document/d/1fo20DdTbZixGOmnXCXVmwbFhlqmuwM9R6vxiAfQgzvY/edit?tab=t.0)

1. AssetIn

**Introduction**

The **“AssetIn”** will be a comprehensive solution for organizations seeking to efficiently manage assets. The system will have two main user roles: Organization Owner and Asset Manager, each equipped with role-specific functionalities.

See full detail explanation here: [AssetIn](https://docs.google.com/document/d/1fNMxwyMOsUmlH0dEafsA-s8xTo2NpLWEbX4O-200Njc/edit?tab=t.0)