MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY BHOPAL

MAFFICK & TECHNOSEARCH 2026

MANTHAN 1.0

Software Problem Statements:

Problem Statement 1(PS ID- SW1)

"Real-Time Chat Moderation Assistant: Build a lightweight service that ingests chat messages and flags/removes toxic or unsafe content in real time and suggests polite alternatives."

Description: The Real-Time Chat Moderation Assistant is a lightweight, fast, and automated system designed to monitor chat messages as soon as they are sent. Its main task is to detect toxic, abusive, unsafe, or inappropriate language in real time and prevent such content from appearing in the conversation. When harmful text is detected, the system should either flag, hide, or remove it and simultaneously provide the user with a polite and appropriate alternative message. This improves the overall quality, safety, and professionalism of online communications while maintaining natural conversation flow.

Objectives:

- **Detect harmful content** such as toxicity, hate, threats, or explicit language in real time.
- **Provide polite rephrasing** that preserves meaning while removing harmful tone.
- **Operate with low latency** to keep chat flow smooth and uninterrupted.
- **Allow configurable rules** for different platforms or communities.
- **Integrate easily** with existing chat systems through simple APIs.
- **Log moderation actions** for review and continuous improvement.

Problem Statement 2(PS ID- SW2)

“Smart Resume Parser + Role Match: Parse PDF/Plaintext resumes and output structured candidate profiles + match score against a job description.”

Description: The Smart Resume Parser + Role Match system automatically extracts key information from PDF or plain-text resumes and converts it into a structured candidate profile. The system then compares the extracted profile with a given job description to generate a relevance or match score. This tool helps recruiters quickly screen candidates and identify top matches with minimal manual effort.

Objectives:

- **Parse resumes accurately** to extract skills, experience, education, certifications, and personal details.
- **Generate structured profiles** in a consistent format for easy review or storage.
- **Compare candidates to job descriptions** and produce a match score based on skills, experience, and job requirements.
- **Support PDF and plaintext inputs** with reliable extraction.
- **Handle different resume formats** and writing styles with high accuracy.
- **Provide fast results** suitable for real-time or bulk screening.
- **Enable easy integration** via API or module for HR systems.

.

Problem Statement 3(PS ID- SW3)

“Plagiarism & Code Similarity Detector (for small assignments): Given multiple source code files, detect pairs that show suspicious similarity and highlight aligned code blocks.”

Description: The Plagiarism & Code Similarity Detector is a tool designed to analyze multiple source code files and identify pairs that exhibit suspicious similarity. It automatically compares code structure, logic flow, and patterns—beyond simple text matching—to catch copied or slightly modified code. The system also highlights the specific aligned or matching code blocks so instructors or reviewers can quickly verify potential plagiarism in small programming assignments

Objectives:

- **Compare multiple code files** and detect suspicious similarities between them.
- **Highlight matching or aligned code blocks** for easy visual inspection.
- **Handle minor modifications** (e.g., renamed variables, formatting changes) and still detect similarity.
- **Support various programming languages** commonly used in assignments.
- **Provide clear similarity scores** for each pair of files.
- **Run efficiently** for small batches of student assignments.

Problem Statement 4(PS ID- SW4)

"Image Duplicate Finder for Photo Albums: Find near-duplicate or highly similar photos in an album (help users clean storage)."

Description: The Image Duplicate Finder for Photo Albums is a tool designed to automatically identify duplicate or highly similar photos within a user's image collection. By analyzing visual features—such as colors, shapes, patterns, and embeddings—the system detects near-duplicate images even if they differ slightly (cropped, edited, resized, or filtered). This helps users clean up storage, reduce clutter, and keep photo libraries organized efficiently.

Objectives:

- **Detect exact and near-duplicate images** using visual similarity analysis.
- **Group similar photos** to help users review duplicates easily.
- **Provide a similarity score** for each pair or group of images.
- **Offer options to delete, archive, or keep selected photos.**
- **Work efficiently on large albums** with thousands of images.
- **Support multiple image formats** like JPG, PNG, HEIC, etc.

Problem Statement 5(PS ID- SW5)

"Deepfake Detector & Authenticity Verifier: Build a defensive tool that detects manipulated audio/video (deepfakes) and provides an explainable authenticity score and provenance hints. Emphasize ethics, consent, and transparency."

Description: The Deepfake Detector & Authenticity Verifier is a defensive tool designed to identify manipulated or AI-generated audio and video content. It analyzes media files to estimate how authentic they are and provides an explainable score that highlights suspicious regions, inconsistencies, or digital artifacts. The system also offers provenance hints—such as source metadata or tampering traces—to help users understand where the content may have been altered. The focus is on ethical use, user consent, and transparency while helping individuals, organizations, and platforms combat misleading or harmful deepfake media.

Objectives:

- **Detect manipulated audio/video** by identifying deepfake artifacts, face swaps, voice cloning, and frame-level inconsistencies.
- **Generate an authenticity score** that quantifies how genuine or manipulated a media file appears.
- **Provide explainable insights** showing which parts of the content triggered suspicion (e.g., heatmaps, segment flags).
- **Promote ethical usage** by emphasizing user consent, transparency, and responsible deployment.
- **Support multiple formats** of audio and video commonly used in social media or content-sharing platforms.
- **Enable practical integration** via APIs, plugins, or standalone tools for journalists, security teams, and content moderators.

Problem Statement 6(PS ID- SW6)

"Smart Parking & Micro Route Optimizer for Urban Deliveries Goal."

Description: Design a system that computes the most efficient delivery route for a fleet of vehicles while dynamically considering real-time parking availability and congestion in urban areas.

The solution should minimize total travel time + parking search time, and respect delivery time windows.

Goals and Objectives:

- **Optimize Delivery Routes:** Compute the most efficient delivery routes for multiple vehicles, minimizing total travel time.
- **Incorporate Real-Time Parking Data**: Use live parking availability to reduce parking search time near each delivery point.
- **Account for Traffic & Congestion:** Continuously adjust routes based on real-time traffic and road conditions.
- **Respect Delivery Time Windows:** Ensure each package is delivered within its allotted time window.