Problem Statement 1: EchoMind - Your AI-Powered Emotional Co-Pilot

The Vision

In a world of constant digital noise, finding a private, non-judgmental space to reflect on our feelings is harder than ever. We share our highlights online, but where do we process the lows, the anxieties, and the moments of stress?

The Challenge: Build **EchoMind**, an intelligent and empathetic AI companion designed to be more than just a chatbot. It's an emotional co-pilot that actively listens, helps users understand their emotional patterns, and provides a personalized toolkit for mental well-being. Your goal is to create a digital sanctuary where technology fosters genuine self-awareness.

Core Modules

1. The Conversational Journal:

- **The Interface:** Design a clean, calming chat UI that feels like a private, safe space.
- **The Intelligence:** Move beyond canned responses. Use a small, efficient LLM (like Phi-2, Mistral, or Flan-T5) to generate truly dynamic, natural, and empathetic dialogue that adapts to the user's tone.
- **Multi-Modal Expression:** Allow users to express themselves through **text** or **voice**. The AI shouldn't just transcribe words; it should be prepared to understand the emotion behind them.

2. Deep Emotion Analysis Engine:

- **Beyond Positive/Negative:** Implement a model capable of detecting a nuanced spectrum of emotions (e.g., joy, sadness, anger, fear, anxiety, optimism) using models fine-tuned on datasets like GoEmotions.
- **Vocal Biomarkers (Advanced):** For voice inputs, analyze not just *what* is said, but *how* it's said. Analyze vocal tone, pitch, and pace to infer emotional states. A faster pace might indicate anxiety, while a lower pitch could signal sadness.

3. The Dynamic Mood Dashboard:

- **Actionable Insights:** Don't just show a line graph. Create an interactive dashboard that helps users discover patterns. For example: *"It looks like your stress levels tend to spike on Wednesdays. Is there something happening mid-week?"*
- **Visual Storytelling:** Use a heat map, calendar view, or an "emotional constellation" chart to visualize mood trends, helping users connect their feelings to real-world events.

The 'Wow' Factor (Bonus Ideas):

- **Proactive Check-ins:** If the AI detects a persistent negative trend, it could initiate a gentle, proactive conversation: *"Hey, I've noticed things might have been tough lately. Just checking in."*
- **Memory & Continuity:** Give the AI a long-term memory to recall past themes. *"You mentioned last week you were worried about a presentation. How did it go?"*
- **Emotionally-Aware TTS:** Use a modern text-to-speech engine to have the AI respond with a voice that matches the empathetic tone of its message (e.g., a calm, soothing voice).

Problem Statement 2: NutriSnap - Decode Your Plate

The Vision

You're looking at a plate of delicious food, but what's *really* in it? Calorie counting apps are tedious, and nutritional labels can be confusing. What if you could simply point your camera at any meal and instantly understand its impact on your health?

The Challenge: Build **NutriSnap**, an AI-powered dietary lens for your camera. Your mission is to empower users to make smarter food choices by instantly decoding the nutritional content of their meals. This isn't just a calorie counter; it's a comprehensive food-awareness tool.

Core Modules

1. The Instant Food ID Engine:

- **Image Recognition:** Train or fine-tune a Computer Vision model (e.g., MobileNetV2, EfficientNet) to accurately identify single food items and, more importantly, complex, mixed dishes (like *paneer butter masala*, salads, or pasta).
- **Text-Based Search:** For items without a photo, implement a robust text search that understands natural language queries.

2. From Pixels to Protein: The Data Engine:

- **Data Integration:** Connect to a rich nutritional database API (like USDA FoodData Central or Nutritionix) to pull comprehensive data for the identified food, including calories, macronutrients, vitamins, and minerals.
- **Data Presentation:** The challenge isn't just getting the data; it's displaying it beautifully. Use intuitive charts, graphs, and simple callouts to make the information easy to digest.

3. Beyond the Numbers: Actionable Health Insights:

- **Automated Insights:** Use a small language model to automatically summarize the key health benefits and potential drawbacks. For example: *"This meal is high in protein, which is great for muscle repair, but also high in saturated fat. Consider enjoying it in moderation."*
- **Dietary Context:** Tag foods with relevant labels like "High-Protein," "Keto-Friendly," "High-Fiber," or "Vegan."

The 'Wow' Factor (Bonus Ideas):

- **Recipe Analysis:** Allow users to paste a recipe URL or text, and have the AI calculate the nutritional information for the entire dish per serving.

- **Personalized Goal Tracking:** Let users set dietary goals (e.g., "eat 120g of protein daily"). The app can track their progress and provide feedback on their food choices.
- **Allergen & Sensitivity Alerts:** Allow users to specify allergies (e.g., nuts, gluten). The app should flag any identified foods that contain these ingredients.

Problem Statement 3: Campus Compass - The AI Oracle for Your College

The Vision

Every student knows the pain: a simple question about college policy ("*What's the fine for a late library book?*") sends you down a rabbit hole of broken links, 100-page PDFs, and outdated websites. The official information exists, but it's trapped in a digital labyrinth.

The Challenge: Slay the bureaucracy dragon! Build **Campus Compass**, an AI-powered chatbot that serves as the single source of truth for all college-related queries. Using a Retrieval-Augmented Generation (RAG) model, your bot won't make things up; it will find the precise answer from official documents and present it clearly, saving students time and frustration.

Core Modules

1. Taming the Knowledge Beast:

- **The Source Material:** Gather a corpus of official documents: the student handbook, academic calendar, library rules, fee structures, hostel regulations, etc.
- **The Processing Pipeline:** Build a robust pipeline to ingest these documents (PDFs, DOCX, TXT), clean the text, and split them into logical, indexed chunks.

2. Building the 'Brain': The Vector Knowledge Core:

- **Embedding Generation:** Use a high-quality sentence-transformer model to convert every document chunk into a meaningful vector embedding.
- **The Vector Database:** Store these embeddings in an efficient vector database (like ChromaDB, FAISS, or Pinecone) that allows for lightning-fast semantic searches.

3. The Oracle Interface: Ask Anything, Get Answers:

- **The RAG Pipeline:** When a student asks a question, your system will:
    1. Convert the question into a vector.
    2. Search the vector database to retrieve the most relevant document chunks.
    3. Feed these chunks and the original question as context to an LLM.
- **Trustworthy Responses:** The LLM's primary instruction is to generate an answer **based only on the provided context** and to cite its sources (e.g., "According to the Student Handbook 2025, page 42...").

The 'Wow' Factor (Bonus Ideas):

- **Multi-Document Synthesis:** Train the bot to answer complex questions that require information from multiple sources. Example: *"What's the last day to drop a course, and*

what's the financial penalty for doing so?" (This might involve the Academic Calendar and the Fee Structure document).

- **The Policy Summarizer (TL;DR):** Add a feature where a user can ask, *"Summarize the college's policy on plagiarism,"* and the bot provides a concise, bulleted summary.
- **Personalized Alerts:** Allow users to opt-in for alerts based on the academic calendar, such as reminders for registration deadlines or upcoming fee payments.