**Project Proposal & Solutions Document: MindMate AI**

**1. Project Proposal**

**1.1 Executive Summary**

MindMate AI is a full-stack, agentic AI-powered application designed to address a critical bottleneck in mental healthcare: patient triage. In many community clinics, overwhelmed staff lack the tools to quickly identify patients with urgent mental health needs, leading to dangerous delays in care. Our solution automates and enhances the pre-visit intake process through a compassionate, multilingual conversational interface. It uses an AI agent powered by **IBM watsonx.ai** to analyze a patient's self-described symptoms, assign a clinical severity level, and provide an immediate, prioritized recommendation to clinical staff via a real-time dashboard. We have developed a functional end-to-end prototype that includes a React.js frontend, a Python/Flask backend, and a clinical dashboard, proving the viability and immediate impact of this solution.

**1.2 Problem Statement**

Community health clinics and other under-resourced healthcare settings face a silent crisis. Administrative staff are tasked with patient intake but are not trained clinicians. Faced with high patient loads, they cannot accurately differentiate between patients with low-acuity concerns and those in severe distress. This systemic inefficiency means individuals with urgent needs, such as severe depression or suicidal ideation, can get lost in a queue for weeks. This delay is not just a workflow issue; it is a critical failure that can have devastating consequences.

**1.3 Proposed Solution: MindMate AI**

MindMate AI is an intelligent assistant that serves as the digital front door for mental health services. It ensures every patient is heard and every urgent case is flagged instantly.

* **For Patients:** A simple, accessible web application allows patients to describe their feelings in their own words and preferred language, using either text or voice input.
* **For Clinicians:** A secure, real-time Clinical Dashboard provides an at-a-glance, prioritized view of all patient submissions. High-risk patients are automatically flagged and pushed to the top of the queue, enabling staff to intervene in minutes, not weeks.

The core of our solution is the **Triage Agent**, an AI-powered engine that analyzes patient input, reasons about its content and context, and autonomously assigns a severity score. This moves beyond simple data collection to provide actionable, clinical-style judgment at scale.

**1.4 Target Audience & Market Opportunity**

Our primary users are **community health clinics**, **university counseling centers**, and **primary care providers** who often serve as the first point of contact for mental health issues. The market for digital health tools is rapidly expanding, but many solutions focus on post-diagnosis therapy. MindMate AI targets the critical, underserved pre-diagnosis stage of intake and triage. By reducing administrative burden and preventing critical cases from slipping through the cracks, our solution offers immense value in improving patient outcomes and optimizing clinic resources.

**2. Solutions Document**

**2.1 Application Architecture**

Our prototype is a robust, full-stack application composed of three main components that work in concert. The architecture is designed to clearly separate the user-facing elements from the core business and AI logic.

Screens screenshot of a computer

AI-generated content may be incorrect.

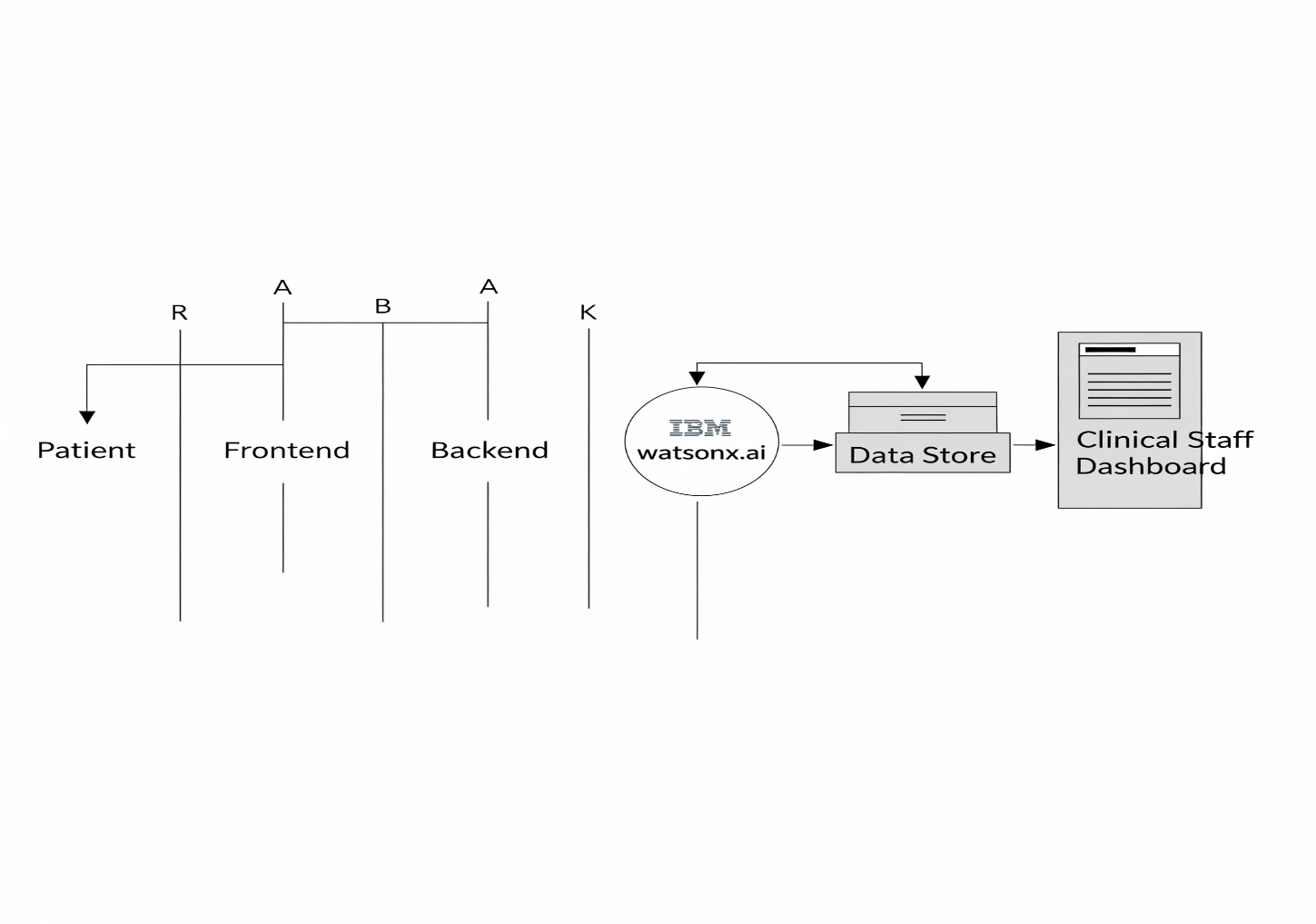
* **Frontend (Patient-Facing Interface)**
  + **Technology:** React.js, HTML5, CSS3
  + **Function:** Provides a responsive, multilingual user interface for patient data entry. It captures name, age, language, and symptoms.
  + **Key Features:**
    - Client-side language translation for UI elements.
    - Browser-native Speech Recognition and Speech Synthesis for accessibility.
    - Dynamic display of results based on severity returned from the API.
    - Sends a JSON payload to the backend via a POST request to the /analyze endpoint.
* **Backend (Orchestration & AI Logic)**
  + **Technology:** Python, Flask
  + **Function:** Serves as the central nervous system of the application.
  + **Key Responsibilities:**
    1. Provides two main API endpoints: /analyze for processing new submissions and /logs for serving data to the dashboard.
    2. Authenticates with the IBM Cloud IAM service to get an access token for watsonx.ai.
    3. Invokes the **Triage Agent** by sending a structured prompt to the watsonx.ai API.
    4. Parses the JSON response from the agent.
    5. Logs every transaction to a persistent data store (triage\_log.csv in the prototype).
* **Clinical Dashboard**
  + **Technology:** HTML5, CSS3, JavaScript
  + **Function:** A secure web interface for clinical staff.
  + **Key Features:**
    - Fetches all submission data from the backend's /logs endpoint.
    - Provides client-side filtering by severity and language, and includes pagination for usability.
    - Uses color-coded badges to visually highlight high-priority cases, enabling rapid clinical response.

**2.2 Agentic AI Architecture & Data Flow**

The intelligence of MindMate AI resides in its **Triage Agent**. This is not a complex multi-agent system but a single, powerful agent defined by its specific task and the way it is invoked. The sequence diagram below illustrates the complete interaction flow from patient to clinician.

A computer screen shot of a diagram

AI-generated content may be incorrect.



* **Agent Definition:** The Triage Agent is an instance of the **IBM Granite-3 8B Instruct model**, tasked with acting as a mental health triage assistant.
* **Activation & Reasoning:** The agent is activated when the Flask backend calls the watsonx.ai API. Its reasoning is guided by a **few-shot prompt** engineered in our app.py script. This prompt provides the model with its role, instructions, examples of low, medium, and high-severity cases, and a strict JSON output format. This technique effectively transforms a general-purpose language model into a specialized, reliable triage expert.

**2.3 Infrastructure Details**

The following outlines the infrastructure for both the current prototype and a scalable production environment on IBM Cloud.

* **Development / Hackathon Environment**
  + **Frontend:** Served locally (e.g., via Node.js dev server).
  + **Backend:** Running locally via flask run command on localhost:5000.
  + **Data Store:** A single triage\_log.csv file on the local filesystem.
* **Production Environment (Proposed IBM Cloud Architecture)**
  + **Compute:**
    - The **Flask Backend** and **React Frontend** will be containerized and deployed as separate applications on **IBM Code Engine**. This provides serverless, auto-scaling capabilities, ensuring the application is both cost-effective and can handle fluctuating loads.
  + **Data Store:**
    - The triage\_log.csv file will be replaced by a managed database solution. We recommend **IBM Cloud Databases** (e.g., PostgreSQL) or, for a more integrated AI workflow, **watsonx.data** to serve as a secure lakehouse for Protected Health Information (PHI).
  + **AI & Governance:**
    - The **Triage Agent** will continue to run on **watsonx.ai**.
    - We will integrate **watsonx.governance** to wrap the model deployment. This will provide an essential audit trail, explainability for each triage decision, and ongoing monitoring for model drift and bias.
  + **Networking & Security:**
    - A custom domain will be routed to the applications via **IBM Cloud Internet Services (CIS)** for DNS management and DDoS protection. All traffic will be encrypted end-to-end with SSL/TLS.
    - User authentication for the Clinical Dashboard will be implemented using **IBM Cloud App ID**.