

# Assignment: Therapist Bot

## Why this assignment?

At **Skooc**, we build AI tools for mental wellness. This assignment is designed to test your ability to:

- **Prompt engineer** LLMs for mental health-aligned use cases
- Apply **basic guardrails** for sensitive conversations
- Deploy APIs on **AWS App Runner via Docker + ECR**

We want to see if you can go from **idea** → **API** → **deployed service** quickly and cleanly.

---

## Problem Statement

Build a simple LLM-powered API that acts like a **CBT therapist** and responds to emotional queries in a structured, compassionate way.

---

## Functional Requirements

### 1. FastAPI App

- Build an endpoint: `POST /respond`
  - Input: `{ "message": "I feel anxious about exams" }`
  - Output: `{ "response": "Let's explore that anxiety together. What's the worst that could happen?" }`
- Use **any LLM API**.

- Include a **system prompt**: *"You are a licensed CBT therapist helping users manage anxiety, stress, and negative thoughts. Speak warmly, don't overstep your boundaries."*

## 2. Prompt Engineering & Guardrails

- Engineer the prompt to **mimic a real therapist's tone**
  - Add **basic guardrails**, e.g.,:
    - No diagnosing
    - Avoid medical or crisis advice
    - Return: *"I'm not qualified to handle this. Please talk to a professional."* for red-flag keywords (e.g. suicide, self-harm)
- 

## Deployment (2 Parts)

### ♦ Part A: Docker + ECR + App Runner

- Containerize the app with Docker
- Push image to **Amazon ECR**
- Deploy to **AWS App Runner** using the ECR image








### ♦ Part B: GitHub Repo + App Runner

- Push source code to GitHub
  - Connect directly to **App Runner**
  - Set up build-from-source deployment
  - Optional: Set up GitHub Actions to automate deploys
-

## Bonus (Optional)

- Add a **minimal HTML frontend** (input box + response display)
  - Use **GitHub Actions** to auto-push Docker image to ECR
  - Log all requests/responses with timestamps
- 

## Final Submission Checklist

-  Working `/respond` endpoint with prompt-based LLM output
  -  Proper Docker setup
  -  App Runner deployments:
    -  Public URL (from Docker + ECR)
    -  Public URL (from GitHub)
  -  Clear documentation with setup + deployment steps
  -  Prompt is warm, helpful, and has soft guardrails
- 

## Evaluation Rubric

| Skill Area         | Criteria                                     |
|--------------------|--|
| Prompt Engineering | Warm, structured, focused on mental wellness |
| Guardrails         | Flags risky inputs and fails gracefully      |
| FastAPI Code       | Clean, async, well-structured                |
| Dockerization      | Lightweight, secure setup                    |
| AWS Deployment     | Can deploy via both Docker and GitHub        |

|                      |                              |
|----------------------|------------------------------|
| <b>Documentation</b> | Easy to follow, well-written |
| <b>Speed</b>         | Can ship a useful MVP fast   |