Mentra Bounty Starter Pack

This document outlines topics, tools, and general tips and tricks that participants are encouraged to familiarise themselves with before the competition, should they choose to do this bounty.

This document does not describe the bounty. You're going to have to wait till the event for that 😀.

Difficulty Level

Difficulty: Accessible / Beginner–Intermediate

This bounty is designed to be accessible to participants with foundational technical knowledge.

Strong solutions will emphasize emotional intelligence, thoughtful design, and user experience, rather than technical complexity alone. Remember this is a psychology bounty at the end of the day.

Recommended Team Composition

Teams with at least one member comfortable with programming are recommended.

This bounty is particularly well suited to teams that combine technical skills with psychological or human-centered perspectives. Teams may benefit from having:

- 1 technical member responsible for implementation and prototyping
- Other members focusing on feature ideation, emotional experience design, ethics, and presentation

Participants from psychology, mental health, design, or related disciplines are strongly encouraged to contribute to concept development and user experience.

Bounty Prizes

3 teams will be chosen to be awarded with prizes.

- 1st Place winner: AED 500, with a certificate, feature across Mentra platforms, and an internship (positioned as "Top Award: Built for Impact")

- 2nd Place winner: AED 300, with a certificate and a 1-month mentorship check-in ("Technical Excellence")

- 3rd Place winner: AED 200, with a certificate only ("Most Creative")

Topics encouraged to look at

1. Psychological & Emotion-Centered Design

Participants are encouraged to explore how technology interacts with human psychology to support users during moments of vulnerability.

- Cognitive Load: Understanding that anxious users often have less mental energy. How do you design interfaces that don't overwhelm?
- Affect Labeling: The psychological benefit of "naming" an emotion. Research why identifying a feeling helps regulate it.
- Reflection vs. Fixing: Investigating features that help users look inward (e.g., reframing negative thoughts) rather than just providing a "quick fix."

Resources:

- Psychology Today: Why Labeling Emotions Matters
- Nielsen Norman Group: Sympathy vs. Empathy in UX
- Smashing Magazine: Designing For Mental Health

2. Trust, Safety & Digital Ethics

Participants are encouraged to deepen their understanding of how to build trust in sensitive digital environments.

- **Setting Boundaries:** Designing apps that know their limits. How do you signal to a user that an AI is a "tool" and not a "human"?
- **Privacy by Design:** Exploring how to create connection or personalization without demanding invasive personal data.
- **Safety Protocols:** Understanding how an app should react if a user inputs something concerning or risky.

Resources:

- Humane Tech: The Principles of Humane Technology
- Design Ethically: Toolkit for Ethical Design

3. Creative Use of LLMs (Beyond Conversation)

Participants are encouraged to explore how Large Language Models can work in the background to support creative expression and cognitive growth, rather than just building standard chatbots.

- Reframing: Using AI to help a user develop a "kinder inner voice" by rewriting harsh self-talk.
- Pattern Recognition: Summarizing emotional trends over time to help a user feel "seen."
- Adaptability: Using AI to change the tone or content of the app to match the user's mood.

Resources:

- OpenAI: Prompt Engineering Guide
- LangChain: Tagging & Extraction

4. Visual Expression & Adaptive Interfaces

Participants are encouraged to look into how interfaces can become "visually expressive," changing and reacting to the user's state.

- Calm Technology: Exploring interfaces that soothe the user rather than demand attention.
- Dynamic Design: Investigating how an app can "feel different" when the user feels different (e.g., changing color palettes, animations, or layout based on mood).
- Meaningful Gamification: Considering how to turn tracking into something beautiful (like growing a garden) rather than stressful (like maintaining a streak).

Resources:

- Calm Technology: The Principles of Calm Tech – *Designing tools that don't demand attention.*
- Growth Design: Psychology of Gamification – *Visual case studies on how apps use psychology to engage users.*
- Material Design: Understanding Motion – *How animation influences the "feel" of an app.*

5. Technical Implementation of LLMs

Participants are encouraged to familiarize themselves with how to technically integrate LLMs into a software application using standard APIs.

- API Basics: Understanding how to send a request to an LLM provider (like OpenAI) and parse the response.

- Prompt Engineering vs. System Prompts: Learning how to structure "System Instructions" to give the AI a specific personality or constraint.
- Handling Outputs: Exploring how to ensure the AI output can be used by your app (e.g., getting the AI to output JSON data instead of just a paragraph of text).

Resources:

- OpenAI: Quickstart Guide
- DeepLearning.AI: ChatGPT Prompt Engineering for Developers

6. Presenting & Pitching solutions

Participants are encouraged to think about how to clearly communicate their solution to both technical and non-technical audiences.

Useful areas to explore include:

- Clearly defining the problem being addressed
- Explaining why the solution matters and who it helps
- Demonstrating system behavior through simple workflows
- Communicating trade-offs, limitations, and future extensions

Strong presentations often focus on clarity and value, rather than implementation details alone.

Resources:

- Y Combinator: How to Pitch Your Product
- Guy Kawasaki: The Only 10 Slides You Need in a Pitch

Recommended tech stack

Participants are recommended to choose a technology where they can prototype quickly.

For Rapid Prototyping (Low-Code / AI-Assisted) participants can look at:

- Lovable: https://lovable.dev – Generate apps from text descriptions.
- Bolt: https://bolt.new/

For Python Developers:

- Streamlit: https://streamlit.io/
- Flask

For JavaScript/TypeScript Developers:

- Vite + React

If your team has strong engineering skills we encourage you to explore the following:

- Frontend: Flutter (Dart)
- Backend: PHP (Laravel) or Firebase

General Tips and Tricks

- If you are interested in using Flutter or Laravel or any of the suggested technologies, install SDKs and frameworks before the start of the competition. Troubleshooting installation errors during the event is the #1 way to lose time.

- Look into effective prompt engineering. Good prompt engineering means you can get better results if you use LLMS for ideation.