Code Mentor

Technical Challenges:

- **Session-based Key Management (BYOK):** Users provide their own API keys, which may be invalid or expire during use.
- **No Server-side Storage** Limits scalability, user tracking, and multi-session continuity
- **AI Accuracy in Defense Simulation:** Generating meaningful panel-style questions depends heavily on code structure recognition.

How will you address them?

- Validate keys upfront with in-browser scripts and offer step-by-step guidance.
- Use Web Workers for async code parsing and chunk-heavy file handling.
- Begin with rule-based question templates based on function patterns, naming conventions, and documentation gaps.

Skill Reflection: Developing this kind of system functionality is quite new to me specially handling api keys and the no server side storage since all api keys are stored temporarily in browser session. And I also don't have any experience train ai. Also it can be a new learnings to me while depeloping this system.

TruthCheck

Technical Challenges:

- **NLP Assessment Bias:** AI responses may inaccurately label content, especially in local contexts or slang.
- **Gamified Learning Precision:** Designing meaningful yet engaging tasks without oversimplifying media literacy.
- **User Progress Tracking:** Needs secure and flexible DB structure for quiz scores and learning paths.

How will you address them?

- Use NLP APIs with feedback calibration, confidence indicators, and bias disclaimers.
- Build multi-level challenges with varied media samples and score ranges.
- Design normalized DB tables for user profiles, quiz results, and badge progress. Use indexing for fast retrieval.

Skill Reflection:

FinBuddy PH

Technical Challenges:

- **Sensitive Financial Data Handling:** Risk of exposure even with browser-based tracking.
- **Real-time API Reliability:** Crypto, inflation, or market feeds can be inconsistent or slow.
- **Advice Personalization Logic:** AI must give useful and non-risky financial suggestions.

How will you address them?

- Encrypt sensitive fields and offer users control over storage preferences.
- Use cached results and fallback conditions when APIs are unavailable.
- Start with rule-based advice patterns based on user budget categories

Skill Reflection:

This system should more focus in the dashboard since this is a financial system and also need to be responsive and I'm not quite good about this one but handling the logics or data maybe I can. Also this quite more challenging to me since this more focus at the database logics and I'm not good enough with it. And it needs more secure and flexible DB structure.