

SECTION 1: BASIC DETAILS

Name: [Pragati Arya]

- ◆ AI Agent Title- **Zenquizion**

- ◆ Use Case: Zenquizion is an AI-powered quiz assistant designed to help students revise effectively through personalized quizzes. It adapts to each student's mood, learning progress, and weak areas using smart memory tracking.

SECTION 2: PROBLEM FRAMING

- ◆ 2.1 What problem does your AI Agent solve?

Students often struggle with structured revision and tend to lose motivation quickly. They forget what they previously got wrong and don't always know when to take breaks. Quizora solves this by offering memory-based quizzes and targeted revision. It also promotes wellness with timely breaks to avoid burnout and keep focus sharp.

- ◆ 2.2 Why is this agent useful?

Quizora transforms revision from stressful cramming into an engaging experience. It personalizes quizzes based on user mood and progress for better retention. By tracking mistakes and offering retry chances, it reinforces learning effectively. Wellness reminders keep the mind refreshed, balancing productivity with care.

- ◆ 2.3 Who is the target user?

Quizora is designed for school and college students facing academic pressure. Whether preparing for daily tests or final exams, it supports their learning journey. It's ideal for students who want intelligent revision without the clutter of heavy tools. Anyone looking to revise smarter, not harder, will benefit from Quizora.

- ◆ 2.4 What not to include?

We chose not to integrate OpenAI's API to ensure full offline functionality. No need for internet or API keys makes the agent accessible to everyone. The focus is kept on usability, speed, and student-friendly design. We also avoided over-complicating features to maintain a clear and smooth user experience.

SECTION 3: 4-LAYER PROMPT DESIGN

- ◆ 3.1 INPUT UNDERSTANDING

Prompt: "Welcome! Choose your subject and difficulty level."

The AI prompts users to enter their mood, subject, and difficulty for personalized quizzes. It flexibly accepts any student ID, even random ones like xyz or 123.

This ensures inclusivity and supports first-time or anonymous users as well.

◆ 3.2 STATE TRACKER

Prompt: Tracks student ID, name, mood, score, weak areas, and quiz history in memory.json.

Uses a simulated memory structure so each user has a persistent learning profile.

Every quiz session gets logged, including incorrect answers for future revision.

It supports both known and new student IDs dynamically without pre-registration.

◆ 3.3 TASK PLANNER

Prompt: Based on mood and past quiz performance, decides whether to start new quiz, revision, or break.

Branches intelligently: e.g., happy users get standard quizzes, frustrated users get wellness suggestions.

Also tracks skipped/incorrect answers to build future retry lists automatically.

Handles different tasks smoothly, guiding the student through a learning-friendly path.

◆ 3.4 OUTPUT GENERATOR

Prompt: Delivers feedback like “✅ Correct!” or “❌ Incorrect. Try again,” and plays reminder sounds.

Uses supportive, emoji-rich messages and smooth GUI transitions to keep students engaged.

Prompts for next actions with buttons like Continue, Break, and Exit for easy navigation.

Visually and emotionally designed to be calming, clear, and confidence-boosting.

🔍 SECTION 4: CHATGPT EXPLORATION LOG

◆ Attempt 1: “Generate quiz”

I initially used a simple prompt to generate a quiz. The output was too generic and lacked variety or challenge.

➤ I updated the prompt to include both difficulty level (easy/medium/hard) and the user's mood.

➤ This made the quiz more dynamic and tailored to the user's emotional state and learning level.

◆ Attempt 2: Added break feature

I introduced a wellness break feature with a 2-minute timer. The timer worked, but the whisper reminder sound didn't play consistently.

➤ I integrated the pygame library to handle the sound properly.

➤ This fixed the issue and allowed the whisper to loop after the break, enhancing the wellness experience.

◆ Attempt 3: Added GUI

I created a basic GUI using Tkinter, but the layout was cluttered and confusing.

➤ I redesigned the interface using proper layout management, added buttons, entry fields, and feedback messages.

➤ The result was a cleaner, more user-friendly interface with clear navigation and interactive features.

Attempt #	Prompt Variant	What Happened	What You Changed	Why You Changed It
1	"Generate quiz"	Output was very basic and generic	Added difficulty level and mood context	To personalize quizzes based on user mood/level
2	Added break feature	Timer started but sound did not play well	Integrated pygame for sound handling	To ensure whisper reminder plays after break
3	Added GUI	Initial GUI layout was messy and not usable	Rebuilt using Tkinter with layout fixes	For better UX with clear buttons and flow

SECTION 5: OUTPUT TESTS (Optional but Recommended)

Test 1: Normal Input – Quiz Flow

Input: Selected subject as Math or Science, and difficulty as Easy,Medium,Hard.

Expected Behavior: The system generated 95 easy-level math questions based on the user's selection.

Actual Output: Questions were displayed one by one. User scored 4 out of 95.

Result: On completion, an encouragement message and score summary were shown, enhancing student confidence.

Test 2: Trigger Wellness Break

Input: Clicked the "Wellness Break" button during the quiz session.

Expected Behavior: A 2-minute timer should begin, giving the student time to relax.

Actual Output: Break message displayed instantly. After 2 minutes, a whisper sound was played gently to alert the student.

Result: The break was successfully implemented, creating a calming pause before resuming learning.

Test 3: Revision Mode

Input: Selected "Revision Mode" after completing a quiz.

Expected Behavior: Should retry only the questions that were answered incorrectly in the past quiz.

Actual Output: Previously incorrect questions were presented again with fresh input boxes. Feedback was given after each attempt.

Result: Enabled focused revision and learning from past mistakes, strengthening weak areas effectively.

SECTION 6: REFLECTION

6.1 Hardest part?

Getting whisper sound to loop and sync with timer in GUI.

6.2 Most fun?

Making the GUI feel like a friendly learning buddy.

6.3 Improvements?

Would add charts or scoring trends.

6.4 What I learned?

Prompt tuning, mood-based adaptation, and GUI integration.

6.5 Did you get stuck?

Yes, when whisper sound kept breaking. Fixed it by switching to pygame with loops.

SECTION 7: HACK VALUE (Optional)

- Simulated long-term **student memory** using a `memory.json` file, allowing the AI agent to recall student names, moods, quiz history, and weak areas—even across sessions.
- Integrated a full **GUI interface** with interactive buttons, real-time feedback, and personalized greetings—making the experience smoother and more engaging for students.
- Added a soft, continuous **whisper sound reminder** using `pygame`, triggered after a 2-minute wellness break to gently bring students back to learning.
- Included **mood-based encouragement cards** after each quiz, providing emotional support based on how the student feels.
- Implemented **revision filtering**, where only previously incorrect questions are shown, creating a smart and efficient revision loop for better retention.