**Quizzical – Project Documentation**

**📌 Project Overview**

**Quizzical** is a gamified quiz web application built with **React**, **Supabase**, and **AI integration**. Users can enter their name, choose a category, difficulty, and timer, and then take an AI-generated multiple-choice quiz. After completing the quiz, their scores are stored and displayed on a **global leaderboard**, making the experience competitive and engaging.

The project’s main purpose is to:

* Practice **React development** with real-world features.
* Explore **Supabase** (PostgreSQL, realtime, and auth).
* Experiment with **AI-powered quiz generation**.

**🎯 Core Features (MVP)**

1. Enter name and select quiz settings.
2. Choose a category (topic).
3. Choose difficulty (Easy, Medium, Hard).
4. Choose a timer setting (per question or per quiz).
5. AI generates multiple-choice questions.
6. User answers and gets a score.
7. Score is stored in Supabase.
8. Global leaderboard shows results in real-time.

**🔮 Extra Features (Future)**

* **Authentication**: Supabase Auth (email/password or OAuth).
* **Personal quiz history**: Users can review past attempts.
* **Social sharing** (optional for later).
* **Admin dashboard**: Manage categories and leaderboards.
* **Dark mode / custom themes**.

**🛠 Tech Stack**

* **Frontend**: React + TailwindCSS (or shadcn/ui for components).
* **Backend**: Supabase (PostgreSQL, Realtime, Auth).
* **AI Integration**: OpenAI API (for question generation).
* **Deployment**: Vercel (frontend) + Supabase (backend hosting).

**🗄 Database Schema**

-- Users (anonymous for now)

create table users (

id uuid primary key default gen\_random\_uuid(),

name text not null,

created\_at timestamp default now()

);

-- Categories (topics)

create table categories (

id uuid primary key default gen\_random\_uuid(),

name text not null

);

-- Scores (with difficulty + timer recorded)

create table scores (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references users(id),

category\_id uuid references categories(id),

difficulty text check (difficulty in ('easy','medium','hard')),

timer int, -- in seconds

score int not null,

created\_at timestamp default now()

);

**🔄 App Flow**

1. **Landing Page**
   * User enters name.
   * Selects category, difficulty, and timer.
2. **Quiz Page**
   * AI generates questions based on category + difficulty.
   * Timer counts down.
   * Auto-submit if timer expires.
3. **Result Page**
   * Score displayed.
   * Score saved in Supabase (linked to category, difficulty, timer).
4. **Leaderboard Page**
   * Global leaderboard sorted by score.
   * Can filter by category & difficulty.
   * Uses Supabase Realtime for live updates.

**🤖 AI Integration Details**

* Prompt Example:

*“Generate 5 multiple-choice quiz questions about [CATEGORY] at [DIFFICULTY] level. Each question should have 4 options and 1 correct answer. Format the response as JSON with fields: question, options, correctAnswer.”*

* React parses the AI response into a structured quiz format.
* Questions are **not stored** in the database for MVP (generated fresh each time).

**🚀 Future Enhancements**

* Store AI-generated quizzes in Supabase for reuse.
* Add streaks, badges, or achievements for gamification.
* Show average scores per category/difficulty.
* Support multiplayer / head-to-head mode.

**📂 Suggested File Structure**

src/

├─ components/

│ ├─ Quiz.js

│ ├─ Leaderboard.js

│ ├─ Timer.js

│ └─ QuestionCard.js

├─ pages/

│ ├─ Landing.js

│ ├─ QuizPage.js

│ ├─ ResultPage.js

│ └─ LeaderboardPage.js

├─ utils/

│ ├─ supabaseClient.js

│ └─ aiService.js

├─ App.js

└─ index.js

**✅ Summary**

Quizzical provides a **fun, global, AI-powered quiz experience** with real-time leaderboards. It is designed to be simple enough for practicing React, Supabase, and AI integration, while flexible enough to extend into a portfolio-worthy project.