# The SlangLab project

## Project Title:

Rizzler: The Gen Z Slang Decoder & Quiz App

## Team Name:

Quack ‘O Debug

## Team Members:

Anamitra Joshi  
Monika Agarwal  
Keerthana Manam

## Phase-1: Brainstorming & Ideation

### Objective:

Rizzler is a chatbot designed to provide accurate meanings of Gen Z slang while restricting itself from responding to unrelated queries. It ensures an engaging learning experience by incorporating a daily quiz feature, making slang exploration both educational and entertaining.

### Problem Statement:

* Rapid Evolution of Slang – Slang terms constantly change, making it difficult for users to stay updated with their meanings and usage.
* Contextual Misinterpretation – Many slang words have multiple meanings based on context, leading to misunderstandings and miscommunication.
* Lack of Personalized & Accurate Resources – Existing slang dictionaries provide static definitions without AI-driven, context-aware insights tailored to users.
* Regional & Cultural Variations – Slang differs across regions and communities, making it challenging to understand localized expressions.

### Proposed Solution:

* An AI-powered chatbot named Rizzler that exclusively provides meanings and examples of Gen Z slang while refusing to respond to unrelated queries.
* A daily quiz feature to make learning slang fun and interactive.
* Uses Google Gemini API for accurate, real-time slang definitions.

### Target Users:

* Social media users who want to stay updated with trending slang.
* Parents, teachers, and professionals trying to understand Gen Z language.
* Linguists and language enthusiasts interested in evolving slang terms.

**Expected Outcome:**

* A functional AI-powered chatbot dedicated to Gen Z slang definitions.
* An engaging daily quiz feature that reinforces learning.
* A user-friendly and entertaining platform for slang exploration.

## Phase-2: Requirement Analysis

### Technical Requirements:

- Programming Language: Python  
- Backend: Google Gemini API  
- Frontend: Streamlit Web Framework  
- Database: Not required initially (API-based queries)

### Functional Requirements:

- Provide slang meanings with context and examples.  
- Restrict chatbot responses to slang-related queries only.  
- Offer a daily quiz with multiple-choice questions.  
- Maintain a fun and engaging persona to keep users entertained.

### Constraints & Challenges:

- Ensuring chatbot remains restricted to slang-related content.  
- Handling API rate limits and optimizing API calls.  
- Keeping the UI interactive and user-friendly.

## Phase-3: Project Design

### System Architecture:

- User inputs a slang word into the chatbot.  
- Query is processed using Google Gemini API.  
- AI model fetches and processes the definition.  
- The chatbot provides a response with meaning and examples.  
- Users can participate in a daily quiz to test their slang knowledge.

**User Flow:**

Step 1: User enters a slang term (e.g., "What does 'rizz' mean?").

Step 2: The chatbot fetches the meaning and provides an example.

Step 3: The user can participate in the daily quiz to test their knowledge.

Step 4: Quiz responses are evaluated, and the user receives instant feedback.

**UI/UX Considerations:**

* Minimalist, user-friendly interface for seamless interaction.
* Fun and engaging chatbot persona.
* Interactive daily quiz feature with progress tracking.

## Phase-4: Project Planning (Agile Methodologies)

Breaking down development tasks for efficient completion.

### Sprint Planning:

Sprint 1 – Setup & Integration:  
- Set up the environment & install dependencies.  
- Integrate Google Gemini API.  
- Build a basic UI with chatbot functionality.  
  
Sprint 2 – Core Features & Debugging:  
- Implement quiz functionality.  
- Debug chatbot responses.  
  
Sprint 3 – Testing & Deployment:  
- Test API responses, refine UI.  
- Final demo preparation & deployment.

## Phase-5: Functional & Performance Testing

### Test Cases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Category | Scenario | Expected Outcome | Status |
| TC-001 | Functional | Query 'What does mid mean?' | Correct slang definition is displayed | ✅ Passed |
| TC-002 | Functional | Query 'Tell me about politics' | Chatbot refuses to respond | ✅ Passed |
| TC-003 | Performance | API response time under 500ms | API should return results quickly | ⚠ Needs Optimization |
| TC-004 | Bug Fix | Fixed incorrect responses to non-slang queries | Chatbot properly filters questions | ✅ Fixed |
| TC-005 | Validation | Ensure UI is responsive | UI should work on all devices | ❌ Failed - Mobile issue |
| TC-006 | Deployment | Host app using Streamlit Sharing | App should be accessible online | ⚠Pending |