# **Hackathon Project**

## **Project Title:"SlangSavvy: Urban Google Gemini API-Powered Slang Decoder"**

## **Team Name: SlangGenix**

## **Team Members:**

* Soma Ananya Keerthi
* Abburi Yaswitha
* Kancharla Siri Reddy
* Gajula Sri Harshitha

## **Phase-1: Brainstorming & Ideation**

### **Objective:**

Develop an AI-powered slang decoder using Gemini Flash to help users understand and analyze slang terms in real time.

### **Key Points:**

1. **Problem Statement:**

* Many users struggle to understand rapidly evolving slang, especially across different regions and social platforms. Businesses and content creators also need accurate, context-aware slang interpretations for effective communication and engagement.

#### **Proposed Solution:**

* An AI-powered application leveraging Gemini Flash to decode slang by combining data from Urban Dictionary, social media, and online forums. The app provides real-time, region-specific slang meanings using advanced NLP techniques.

#### **Plan of Action:**

* + Data Collection: Gather slang terms from Urban Dictionary, social media platforms, and online forums for the most up-to-date definitions.
  + AI-powered NLP: Analyze slang contextually, offering accurate meanings based on region and usage.

#### **Target Users:**

* + Social media users and general audiences seeking slang meanings.
  + Businesses and marketers looking to engage audiences with modern language.
  + Non-native speakers needing assistance with slang comprehension.
  + Content creators aiming for accurate, trendy language use.

#### **Expected Outcome:**

#### A functional AI-driven slang decoding app that enhances communication, assists businesses in understanding language trends, and offers real-time, context-aware slang interpretations.

## **Phase-2: Requirement Analysis**

### **Objective:**

Define the technical and functional requirements for the SlangSavvy App.

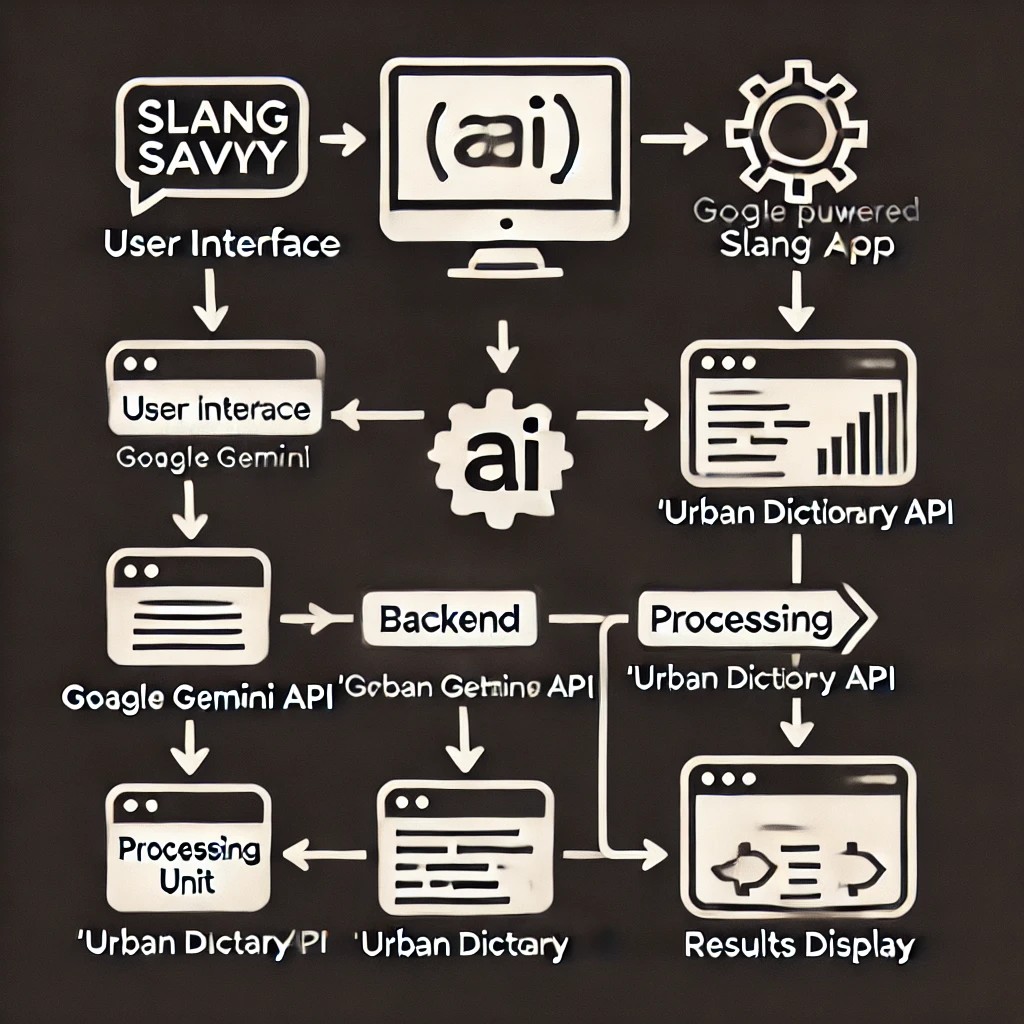
### **Key Points:**

* **Technical Requirements:**
  + Programming Language: Python, JavaScript
  + Backend: Google Gemini API for slang interpretation, Node.js (Express.js) for handling API requests
  + Frontend: Streamlit (Python-based UI Framework)
  + Database: Not required (Urban Dictionary)
* **Functional Requirements:**
  + Slang Search Bar: Instantly fetch slang meanings using AI-powered NLP.
  + Chatbot: Interactive chatbot for slang explanations and real-time discussions.
  + Trending Slang Updates: Track and display the latest slang trends from social media.
  + Personalized Explanations: Provide context-based slang meanings tailored to user preferences.
* **Constraints & Challenges:**
  + Ensuring Real-time Updates: Maintaining up-to-date slang definitions from multiple sources.
  + Data Accuracy: Filtering slang definitions to prevent misinformation.
  + UI Optimization: Ensuring smooth user experience with Streamlit, even with large datasets.

## **Phase-3: Project Design**

### **Objective:**

* To define the architecture, system design, and database schema for the project.
* To create the user interface (UI) and user experience (UX) designs.
* To prepare technical design documentation for implementation.



### **Key Points:**

**System Architecture:**

* Define the overall structure (frontend, backend, database).
* Integration of AI algorithms (Urban Genie) with Urban Dictionary API.
* Handling user queries, providing real-time responses, and tracking usage trends.

**Database Design:**

* Structuring the slang database (storing slang terms, meanings, examples, etc.).
* Designing scalable and efficient data models for easy search and retrieval.

**UI/UX Design:**

* Wireframes and mockups for the app’s interface.
* Prioritizing user-friendly navigation and fast, responsive design.

**Security and Privacy Design:**

* Encryption of user data.
* Ensuring compliance with privacy regulations.

**Testing Design:**

* Plan for user interface and interaction testing

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

### **Objective:**

* To break the project into manageable tasks and set clear milestones.
* To outline the timeline, sprints, and deliverables for each phase.
* To assign roles and responsibilities to the team members.

| **Sprint** | **Task** | **Priority** | **Duration** | **Deadline** | **Assigned To** | **Dependencies** | **Expected Outcome** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint 1 | Environment Setup & API Integration | 🔴 High | 6 hours (Day 1) | End of Day 1 | Yaswitha | Google API Key, Python, Streamlit setup | API connection established & working |
| Sprint 1 | Frontend UI Development | 🟡 Medium | 2 hours (Day 1) | End of Day 1 | Harshitha | API response format finalized | Basic UI with input fields |
| Sprint 2 | Vehicle Search & Comparison | 🔴 High | 3 hours (Day 2) | Mid-Day 2 | Siri | API response, UI elements ready | Search functionality with results |
| Sprint 2 | Error Handling & Debugging | 🔴 High | 1.5 hours (Day 2) | Mid-Day 2 | Ananya  Keerthi | API logs, UI inputs | Chatbot can process slang queries |
| Sprint 3 | Testing & UI Enhancements | 🟡 Medium | 1.5 hours (Day 2) | Mid-Day 2 | Yaswitha  Harshitha | API response, UI layout completed | Responsive UI, better user experience |
| Sprint 3 | Final Presentation & Deployment | 🟢 Low | 1 hour (Day 2) | End of Day 2 | Entire Team | Working prototype | Demo-ready project |

## **Phase-5: Project Development**

### **Objective:**

* To implement the project’s core features and functionalities.
* To integrate all components and ensure proper functioning.
* To conduct regular reviews and ensure the product meets the requirements.

### **Key Points:**

**Frontend Development:**

* Implementing the user interface based on the wireframes and mockups.
* Developing responsive designs for mobile and desktop.

**Backend Development:**

* Setting up servers, APIs, and databases.
* Integrating the slang API and Generative AI (Urban Genie).

**AI Integration:**

* Fine-tuning the AI model for personalized slang explanations.
* Testing the AI’s ability to offer real-time, context-specific slang definitions.

**Feature Implementation:**

* Search functionality for slang terms and providing usage examples.
* Developing a system for slang trend tracking and updates.

**Testing During Development:**

* Continuous testing (unit tests, integration tests) to catch bugs early.
* Collecting user feedback from early access or beta testing.

## **Phase-6: Functional & Performance Testing**

### **Objective:**

* To ensure the system meets functional and performance requirements.
* To identify and resolve any issues before release.

| **Test Case ID** | **Category** | **Test Scenario** | **Expected Outcome** | **Status** | **Tester** |
| --- | --- | --- | --- | --- | --- |
| TC-001 | Functional Testing | Query "What does ‘lit’ mean?" | Correct slang definition displayed. | ✅ Passed | Harshitha |
| TC-002 | Functional Testing | Query "Explain ‘vibe check’ in Gen-Z slang" | AI provides an accurate explanation. | ✅ Passed | Yaswitha |
| TC-003 | Performance Testing | API response time under 500ms | API should return results quickly. | ⚠ Needs Optimization | Siri |
| TC-004 | Bug Fixes & Improvements | Fixed incorrect slang responses. | Data accuracy should be improved. | ✅ Fixed | Developer |
| TC-005 | Final Validation | Ensure UI is responsive across devices. | UI should work on mobile & desktop. | ❌ Failed - UI broken on mobile | Ananya |
| TC-006 | Deployment Testing | Deploy app via Streamlit | App should be accessible online. | 🚀 Deployed | DevOps |

### **Key Points:**

1. **Functional Testing:**
   * Verify that all core features (slang search, definitions, usage examples) are functioning correctly.
   * Ensure proper handling of edge cases (e.g., slang terms not found in the database).
2. **Performance Testing:**
   * Stress testing to ensure the system can handle high user traffic and large amounts of slang data.
   * Load testing to check how the platform performs under different conditions (e.g., many simultaneous users).
3. **Usability Testing:**
   * Assess the user interface’s ease of use, ensuring the experience is intuitive.
   * Collect feedback on how users interact with the app.
4. **Security Testing:**
   * Ensure data protection, secure user authentication, and encrypted storage of personal data.
   * Test the application against potential vulnerabilities.
5. **Bug Fixes and Optimizations:**
   * Address bugs found during testing and optimize the platform for speed and reliability.
6. **Final Review & Deployment Preparation:**
   * Conduct a final project review and prepare for deployment.

## **Final Submission**

1. **Project Report Based on the templates**
2. **Demo Video (3-5 Minutes)**
3. **GitHub/Code Repository Link**
4. **Presentation**