Hackathon Project Phases Template for the **AutoSage App** project.

Hackathon Project Phases Template

Project Title:

Slangsavvy:Urban Google Gemini API-Powered Slang Decoder

Team Name:

(Pro-coders)

Team Members:

- Burra Mahesh
- Bandari Varun Sai
- Dongari Pramod
- Banoth Vijay Kumar
- Bathini Vamshi

Phase-1: Brainstorming & Ideation

Objective:

To develop an AI-powered slang decoder using the Google API,enabling users to accurately interpret and understand urban slang, regional dialects,and internet jargon in real time.

Key Points:

1. **Problem Statement:**

    - With the rapid evolution of language,urban slang,internet jargon,and regional dialects constantly change,making it difficult for individuals to understand and communicate effectively.

- The langauge barrier can lead to misunderstandings,misinterpretations and social disconnects,especially across different generations.

2. **Proposed Solution:**

   - An AI-powered application using **Gemini API** to develop **real-time slang decoder.**
   - Enhancing digital communication and real world conversations more inclusive and understandable.

3. **Target Users:**

   - **General internet users and older generations.**
   - **Non-Native speakers & language learners.**
   - **Students and Educators ,Journalists & researchers.**

4. **Expected Outcome:**

   - A functional **AI-powered slang decoder helps in improvement in understanding of a slang** and user-friendly & accessible platform.

Phase-2: Requirement Analysis

Objective:

To develop an AI-driven slang decoding platform that levearage the google gemini api. 3

Key Points:

1. **Technical Requirements:**

   - Programming Language: **python**
   - Backend: **python**
   - Frontend: **streamlit**

2. **Functional Requirements:**

   - Accepts text input and supports multiple input formats.
   - Detect slang words ,abbreviations ,acronyms, and informal expressions.
   - uses AI/NLP to interrupt slang based on sentence context.
   - Allow user a to select preferred formality levels for translations.

3. **Constraints & Challenges:**

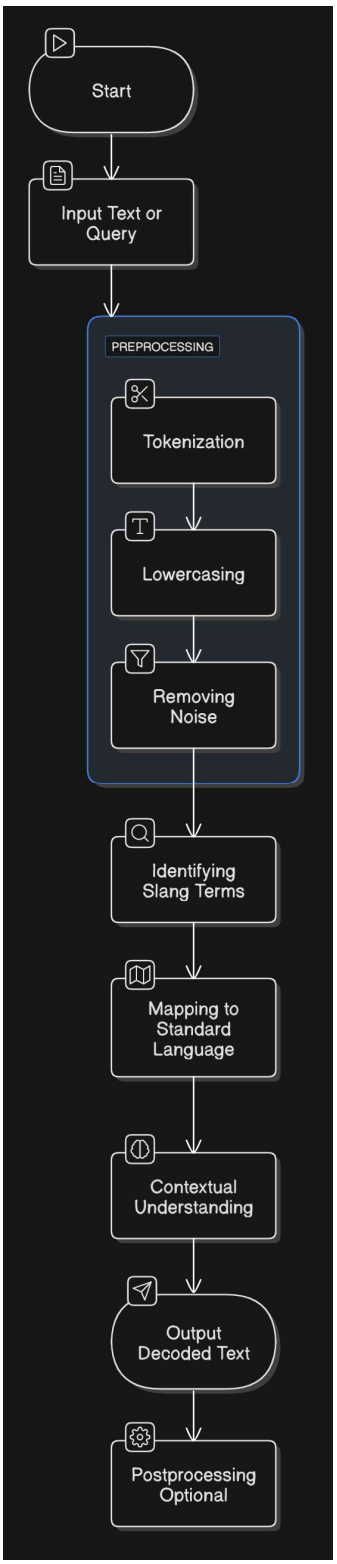
   - Constant evolution of slang and context ambiguity.
   - Natural Language processing & real-time processing speed.

- API and platform integration.

Phase-3: Project Design

Objective:

Develop the architecture and user flow of the application.


Start
Input Text or Query
PREPROCESSING
Tokenization
Lowercasing
Removing Noise
Identifying Slang Terms
Mapping to Standard Language
Contextual Understanding
Output Decoded Text
Postprocessing Optional

Key Points:

1. **System Architecture:**

   - Slang decoder can be designed by natural language processing.
   - It is also uses machine learning,& a slang dictionary database.
   - social media posts, chat messages ,text input from users.
   - slang to standard text mapping.
2. **User Flow:**

   - Step 1: The user enters a message containing slang(e.g.,"That movie was fire!").
   - Step 2: If speech support is enabled ,the system converts speech to text before processing.
   - Step 3: Tokenization(splitting words).
3. **UI/UX Considerations:**

   - **Minimalist design(keep interface clean with a simple text box).**
   - **real-time feedback(provide instant decoding as the user types).**
   - **conversational UI(if chatbox-based).**

Phase-4: Project Planning (Agile Methodologies)

Objective:

Break down development tasks for efficient completion.

Sprint	Task	Priority	Duration	Deadline	Assigned To	Dependencies	Expected Outcome
Sprint 1	Data collections &pre-processing	High	6 hours (Day 1)	End of Day 1	Member 1	Slang datasets, text corpus.	Cleaned and structured slang dataset.
Sprint 1	Lexicon based mapping	Medium	3 hours (Day 1)	End of Day 1	Member 2	Pre-processed data	Basic lookup based slang translation
Sprint 2	ML model Training(supervised)	High	4 hours (Day 2)	Mid-Day 2	Member 1& 3	Processed dataset, embeddings	Trained slang t0 standard translation model
Sprint 2	API development &integration	High	3 hours (Day 2)	Mid-Day 2	Member 2&4	Model Pre-processing pipeline	REST API for slang decoding

Sprint 3	Context based NLP Enhancements	Medium	4 hours (Day 3)	Mid-Day 3	Member 1&3	Model API responses	Improved slang understanding in context
Sprint 3	UI Development (web/app)	medium	3 hour (Day 3)	Mid-day 3	Member 2&4	API integration	Frontend interface for slang input/output

Sprint Planning with Priorities

Sprint 1 – Setup & Integration (Day 1)

(**High Priority)** Set up the **environment** & install dependencies.
(**High Priority)** Integrate **NLP model for slang translation.**
(**Medium Priority)** Build a **basic UI with input fields for user interaction.**

Sprint 2 – Core Features & Debugging (Day 2)

(**High Priority)** Implement **slang detection & translation functionalities**.

(**High Priority)** optimize API calls for better response time

Sprint 3 – Testing, Enhancements & Submission (Day 2)

(**Medium Priority)** improve UI design for better user experience.

Phase-5: Project Development

Objective:

Develop a slang decoder that translates informal slang into standard language using AI and NLP techniques.

Key Points:

1. **Technology Stack Used:**

   - **Frontend:** Stream lit
   - **Backend:** open AI API/custom NLP Model
   - **Programming Language:** Python
2. **Development Process:**

   - Implement **API key authentication** and **integration with NLP Models**.
   - Train a slang to standard language model using a dataset of commonly used slang. Optimize **search and translation algorithm for accuracy and efficiency**.
3. **Challenges & Fixes:**

   - **Challenge:** Slow API response times for large inputs.
     Fix: implement caching for frequently translated slang terms.
   - **Challenge:** Inconsistent slang usage across different regions.
     Fix: use a diverse dataset with multiple slang variations.

Phase-6: Functional & Performance Testing

Objective:

Ensure that the AutoSage App works as expected.

Test Case ID	Category	Test Scenario	Expected Outcome	Status	Tester
TC-001	Functional Testing	User login	Successfully login	☑ Passed	Tester 1
TC-002	Functional Testing	lol	Laugh out loud	☑ Passed	Tester 2
TC-003	Performance Testing	brb	Be right back	☑ Passed	Tester 3

Final Submission

1. **Project Report Based on the templates**
2. **GitHub/Code Repository Link**
3. **Presentation**