Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	15 February 2025
Team ID	LTVIP2026TMIDS34355
Project Name	Flavour Fusion: AI-Driven Recipe Blogging
Maximum Marks	4 Marks

1. Technical Architecture

The architecture below outlines the end-to-end flow of the **Flavour Fusion** application, distinguishing between the local user environment and cloud-based AI services.

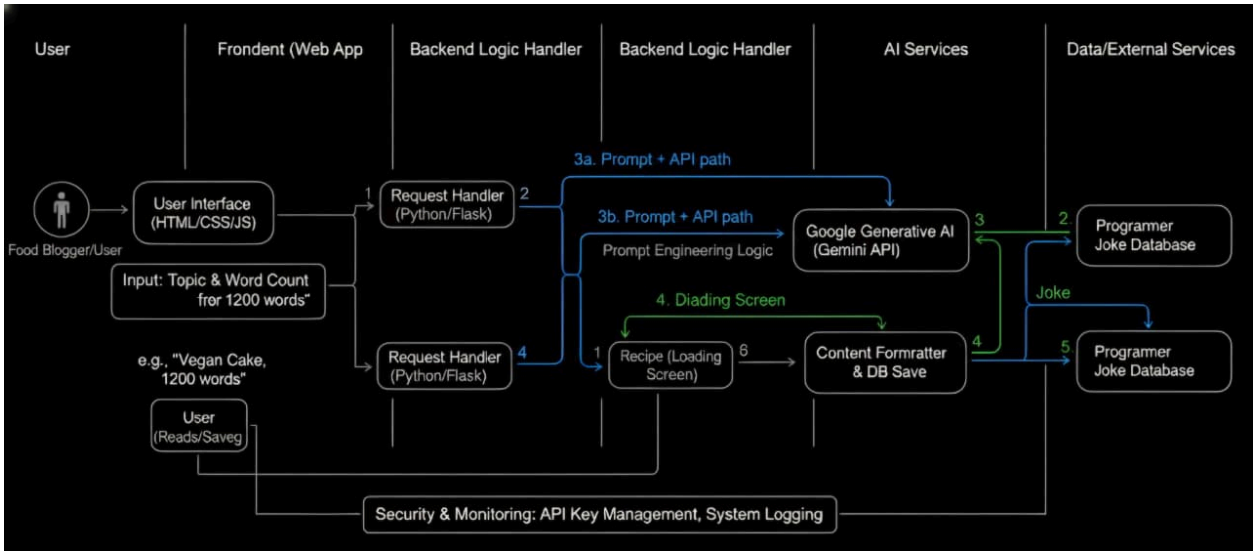

Table 1: Components & Technologies

This table details the specific technical choices used to build and run the application.

S.No	Component	Description	Technology
1	**User Interface**	Responsive web interface for topic input and blog display	HTML, CSS, JavaScript (React.js/Streamlit)
2	**Application Logic-1**	Core backend processing for user requests and API orchestration	Python
3	**Application Logic-2**	Prompt Engineering logic to control recipe word count and tone	Python / Google Generative AI SDK
4	**Application Logic-3**	Interaction handler for displaying programmer jokes during latency	Python (Randomization Logic)
5	**Database**	Storage for the library of programmer jokes	JSON / SQLite / NoSQL
6	**Cloud Database**	Cloud storage for user-generated recipe history	Firebase / MongoDB Atlas

7	**File Storage**	Local or cloud storage for generated blog exports (PDF/Doc)	Local Filesystem / AWS S3
8	**External API-1**	AI Engine for generating high-quality recipe content	Google Gemini API
9	**External API-2**	Potential integration for sharing blogs to social platforms	Pinterest / WordPress API
10	**Machine Learning Model**	Large Language Model (LLM) for creative text generation	Gemini-1.5-Flash (or similar)
11	**Infrastructure**	Application deployment and hosting environment	Cloud (Render, Vercel, or AWS)

Table 2: Application Characteristics

This table explains the non-functional requirements and design considerations that ensure the app is secure and reliable.

S.No	Characteristics	Description	Technology
1	**Open-Source Frameworks**	Use of community-driven libraries for rapid development	Streamlit, Flask, or React.js
2	**Security Implementations**	Protection of API keys and user data privacy	IAM Controls, Environment Variables (.env)
3	**Scalable Architecture**	3-tier architecture ensuring independent scaling of UI and Backend	Micro-services / Tiered Architecture
4	**Availability**	Ensuring the app remains online for global bloggers	Distributed Cloud Servers (Render/AWS)
5	**Performance**	Use of asynchronous calls to display jokes while AI processes	Async/Await Logic, Caching