NAME : GEETHIKA RAVURI

# AGENTIC CONTENT GENERATION SYSTEM – PROJECT DOCUMENTATION

## Introduction:

This project implements a **fully modular, rule-based Agentic Content Generation System**.
The system takes a single product JSON as input and automatically generates:

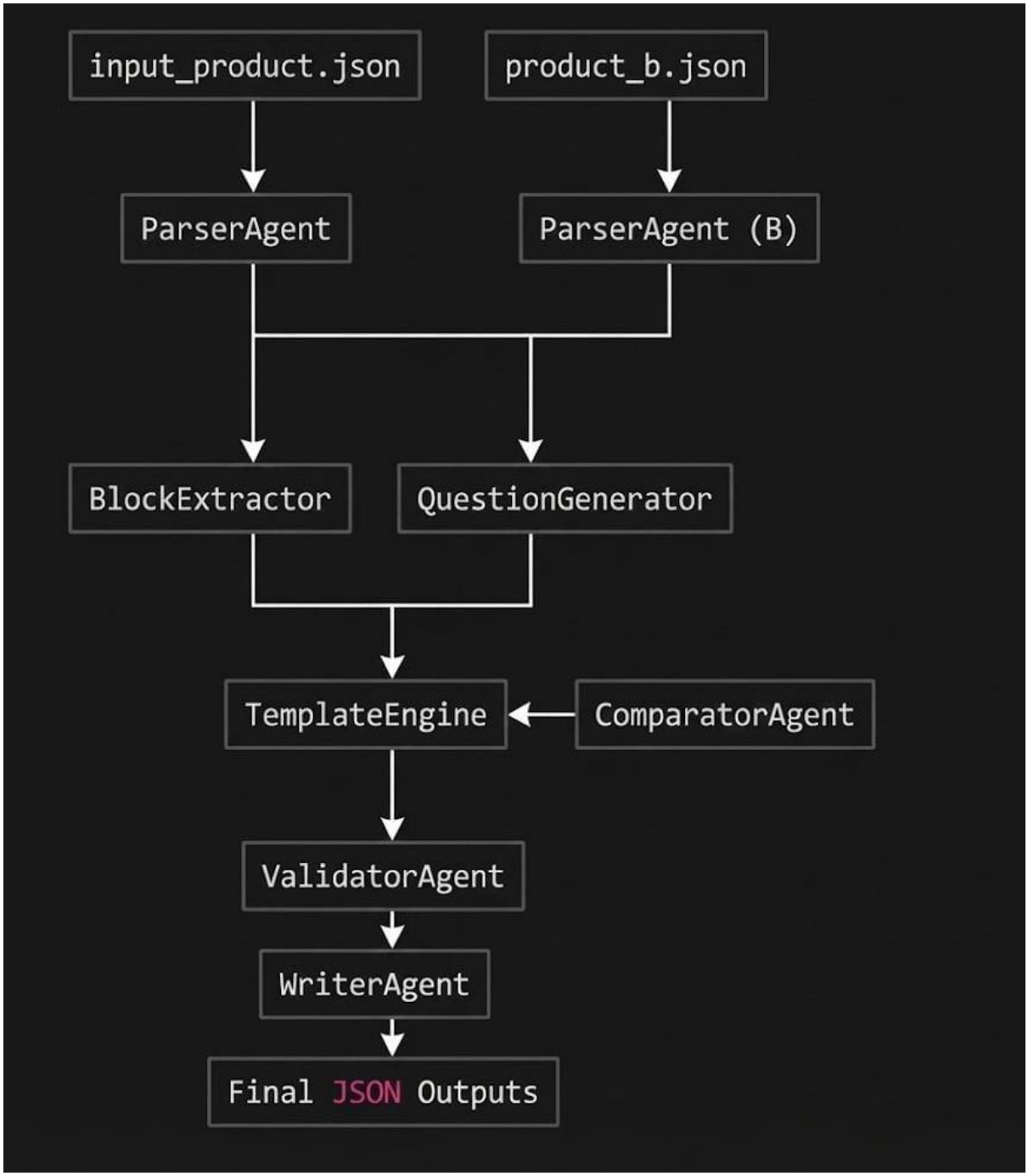
- A structured **Product Page**
- A 20-question **FAQ Page**
- A **Comparison Page** between Product A and Product B
- A **Manifest File** containing hashes for verification

All outputs follow strict schema validation and use **only the data provided in the input JSON**.
No external data sources, no AI writing, and no internet usage are involved.

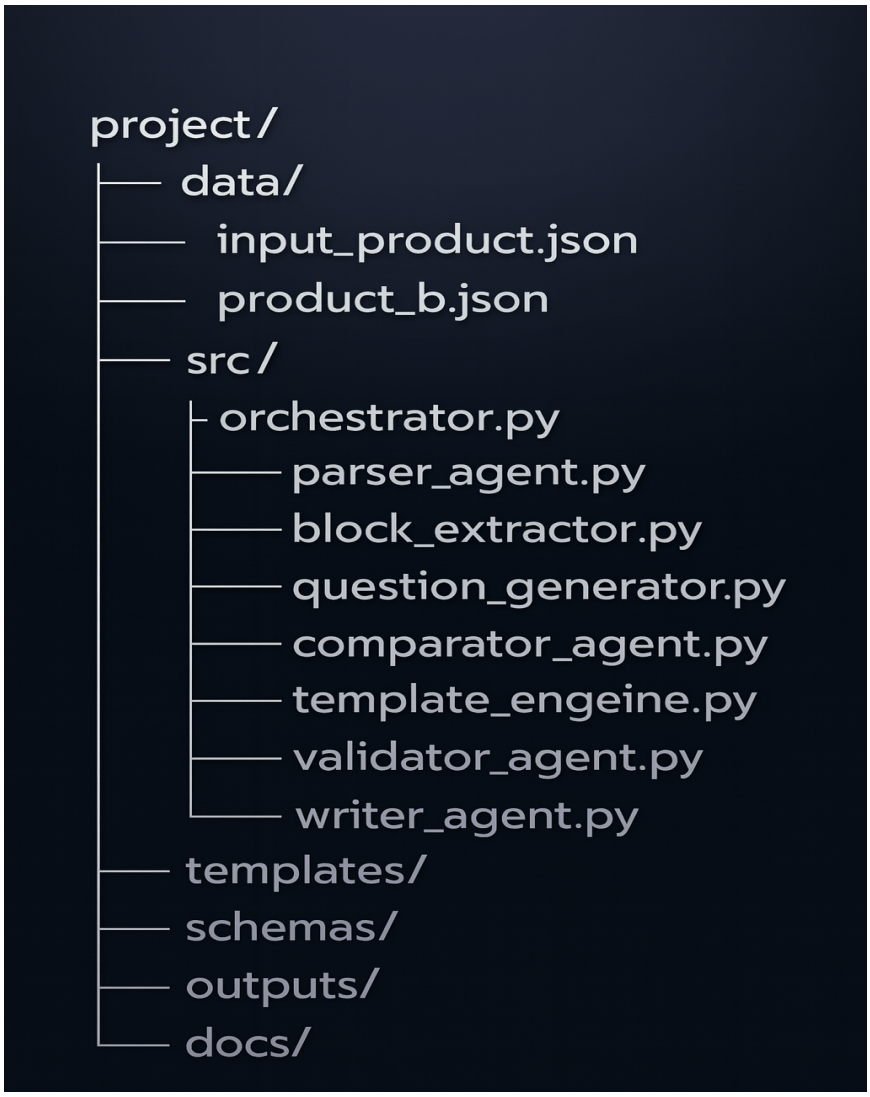
## 2. System Overview:

The system converts raw product data into three final JSON outputs using a pipeline of agents.
Each agent performs exactly one responsibility, making the architecture modular and maintainable.

3. Architecture Diagram:



4. Folder Structure:




5. Agents:

## ParserAgent:

- **Loads the product JSON and cleans the data.**
- **Converts price formats (₹699 → 699) and creates a product ID.**
- **Ensures lists like benefits, ingredients, and skin types are standardized.**
- **Produces a clean product object for other agents to use.**

## BlockExtractorAgent:

**->Breaks the product data into reusable content blocks.**

**->Generates blocks like title, benefits, usage, ingredients, safety, and price.**

**->These blocks are later used by the TemplateEngine.**

## QuestionGeneratorAgent:

**->Automatically generates exactly 20 questions using rule-based logic.**

**->Uses only product data such as benefits, usage, ingredients, price, etc.**

**->Ensures unique questions and uses fallback templates as filler.**

**->No external data or AI writing is used.**

## ComparatorAgent:

**->Compares Product A and Product B.**

**->Compares fields like price, concentration, benefits, and ingredients.**

**->Produces structured comparison results in JSON format.**

## TemplateEngine:

**->Builds final product and FAQ pages using JSON templates.**

**->Uses source, depends_on, rules, and blocks to assemble data.**

**->Joins sentences, limits characters, and merges blocks cleanly.**

## ValidatorAgent:

**->Ensures each output JSON follows the correct schema.**

**->Validates product page, FAQ page, and comparison page structures.**

**->Guarantees compliance with assignment requirements.**

## WriterAgent:

- **Writes all final JSON files to the outputs/ folder.**
- **Generates a manifest.json containing SHA256 hashes.**

- **Finalizes all output for submission.**

**Rule-Based Question Generation**

**Questions are generated using:**

- **Product name**
- **Benefits list**
- **Usage instructions**
- **Ingredients**
- **Side effects**
- **Price**

**This project successfully demonstrates a modular, rule-based agentic system that transforms product data into structured, validated content pages.**
**The system is deterministic, compliant, extensible, and fulfills all assignment requirements.**