Project PRD Document: Optimizer

1. Deep Description of the Project
Optimizer is an AI-powered ideation, optimization, and self-challenge assistant designed specifically for small startups. The platform acts as an AI co-founder, capable of building blueprints, analyzing project structures, optimizing business and technical strategies, and breaking the echo chamber effect in startup teams.

Small startups often face challenges like limited manpower for research and brainstorming, lack of diverse viewpoints, and missing optimization opportunities through automation or open-source tools. Optimizer directly tackles these issues using a multi-agent system that works collaboratively to simulate a full-scale strategic and technical discussion.

The system takes startup-related documents and transcripts as input, indexes them in a vector database using Sentence Transformers (e.g., all-mpnet-base-v2), and applies Retrieval-Augmented Generation (RAG) to reason over this data. It consists of multiple specialized agents:
BlueprintAgent: Generates technical and operational blueprints using Gemini Nano Banana to visualize the idea flow and structure. **CrawlerAgent:** Uses SerpApi to research similar projects, repositories, and solutions on GitHub, saving structured data into JSON. **OptimizerAgent:** Integrates insights from the crawler and RAG to propose technical, financial, and strategic optimizations using Gemini-2.5-flash. **EchoAgent:** Detects team bias and missing viewpoints by acting as a devil's advocate to enhance critical thinking and diversity of ideas. **SynthesisAgent:** Combines outputs from all agents to produce a final comprehensive report for founders or product teams. The backend is built on Flask and LangChain for orchestration. FAISS serves as the local vector database, while the frontend is developed using HTML/CSS/JS, creating a single-page or multi-tab user interface.

2. Team Working on the Project
The Optimizer project is developed by a cross-functional AI innovation team composed of the following members: **Ashok Suthar – Project Lead & AI Engineer:** Oversees the end-to-end development, responsible for RAG pipeline, model integration, and overall system design. **Priya Nair – Backend Developer:** Manages Flask APIs, LangChain agent orchestration, and FAISS vector database implementation. **Rohan Mehta – Frontend Developer:** Develops the interactive SPA interface and integrates APIs for seamless agent communication. **Arjun Patel – Research Engineer:** Works on prompt engineering, open-source tool research, and EchoAgent's bias detection logic. **Meena Sharma – Product Designer:** Designs UX flow, ensuring accessibility and simplicity for startup users.

3. Team Discussion Transcript

Ashok: So the main pain point we're solving is the lack of structured ideation and optimization for small startups. Many founders can brainstorm ideas but don't have the time to validate or improve them technically.
Priya: Exactly. That's where our BlueprintAgent and CrawlerAgent come in. The BlueprintAgent visualizes the workflow, and the CrawlerAgent brings real-world context from GitHub and research sources.
Rohan: Right, and once we have those insights, the OptimizerAgent uses Gemini-2.5-flash to refine everything — suggesting better architectures, financial trade-offs, and even open-source tool integration.
Arjun: I think the EchoAgent is our differentiator. Acting like an internal devil's advocate will really help startups catch blind spots in their strategies.
Meena: Yes, and from a design perspective, we'll make it super intuitive — founders can drag and drop their documents, and within minutes, they get an AI-generated report and blueprint.

Ashok: Perfect. Let’s also ensure that the SynthesisAgent merges everything seamlessly. The final output should feel like an investor-ready report.
Priya: Agreed. I’ll ensure that the backend handles the vector embeddings and FAISS indexing efficiently.
Rohan: Great! I’ll start integrating the frontend with Flask routes so we can demo a complete end-to-end prototype.