Google Cloud H2S

Gen AI Exchange Hackathon

Team Name : Bug Slayer

Team Leader Name :Naresh Reddy Paluri

Problem Statement :Bridging the Digital and Physical Divide With Generative AI

Brief about the Prototype :

- **Definition**: A prototype is an initial version of your Gen AI project that demonstrates the core idea, functionality, and user experience.
- Purpose:
  - To showcase how your idea solves the given problem.
  - To let judges and mentors see the practical working of your AI solution.
  - To test the feasibility of your approach within the hackathon’s limited time.
- Features:
  - Doesn’t need to be perfect or production-ready.
  - Should clearly highlight the innovation, usability, and impact.
  - Often includes a demo interface (like a chatbot, web app, or dashboard) + a backend model (AI/ML model, APIs, workflows).
- Examples in Gen AI Hackathons:
  - A chatbot prototype using LLMs to answer domain-specific queries.
  - A text-to-image prototype generating designs for a given prompt.
  - A Gen AI-powered recommendation system for students or businesses.

Opportunity Prototype Breif :

❑ **1. How is it different from existing solutions?**

❑ Most existing platforms provide generic, one-size-fits-all outputs. Opportunity is different because it focuses on context-aware, personalized, and real-time insights, ensuring that users get solutions tailored to their specific needs rather than broad, unfocused results.

❑ **2. How will it solve the problem?**

❑ OpportunitQ takes raw or unstructured data, processes it using Generative AI techniques (reasoning, summarization, knowledge retrieval), and delivers clear, actionable

❑ outputs. This reduces complexity, saves time, and enables users to make better and faster decisions.

❑ **3. Unique Selling Proposition (USP)**

❑ The USP of Opportunity lies in its ability to combine Generative AI intelligence with real-time adaptability and personalization, creating a solution that is faster, smarter, and more user-centric than existing alternatives.

List Of Features Offered by the Solution :

❑ **How is it different?**

Most tools give generic results → Opportunity delivers context-aware & personalized insights.

❑ **How will it solve the problem?**

❑ Input (raw data) → Gen AI Processing (reasoning + summarization) → Output (clear, actionable insights)

❑ **USP**

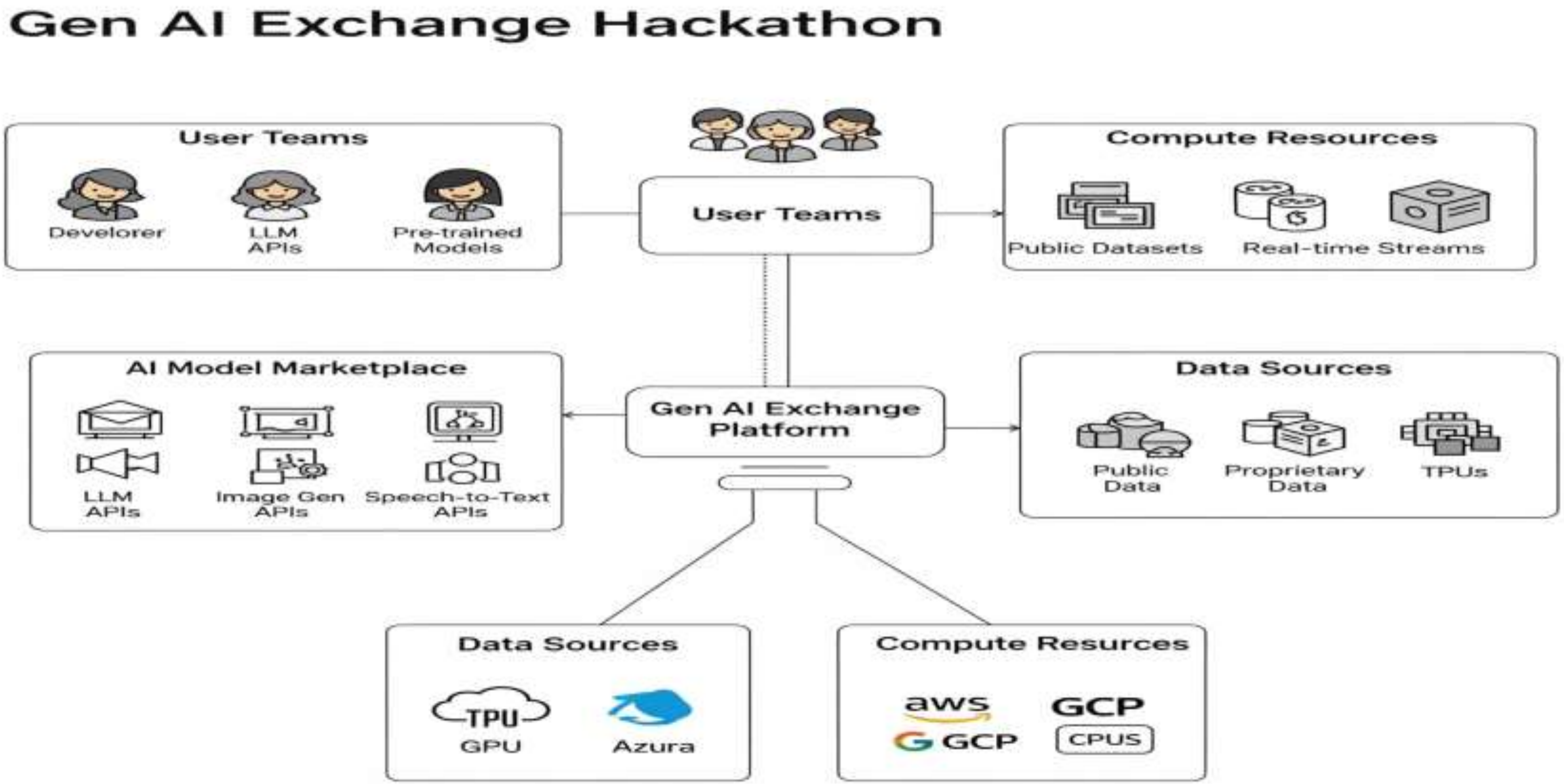
Real-time adaptability + personalization → A smarter, faster, user-centric solution.

❑ **Visual tip:**

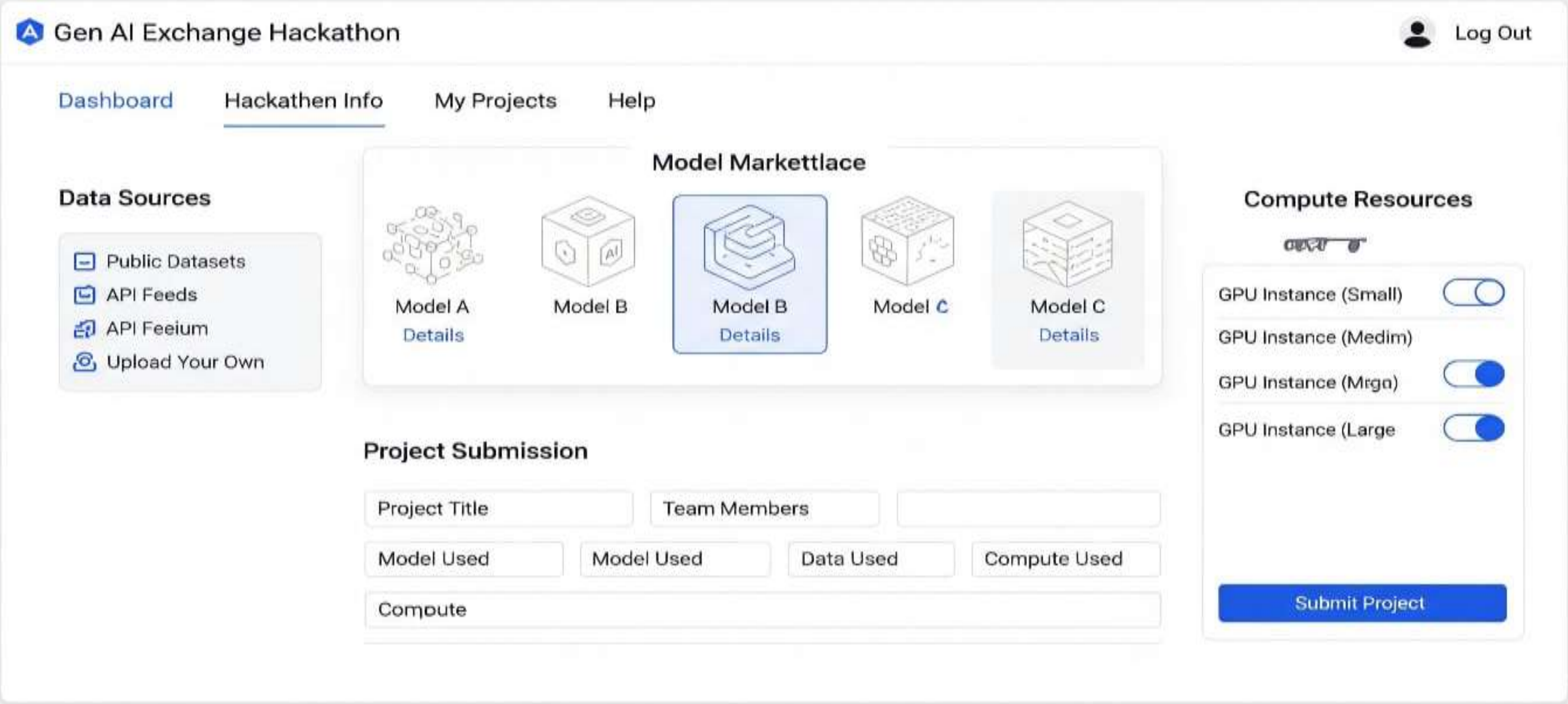
❑ Add a simple flow arrow diagram in the middle.

❑ Keep text minimal, let visuals explain.

System Architecture :



Mock Diagram :



Data Sources

- Public Datasets
- API Feeds
- API Feeium
- Upload Your Own

Model Markettlace

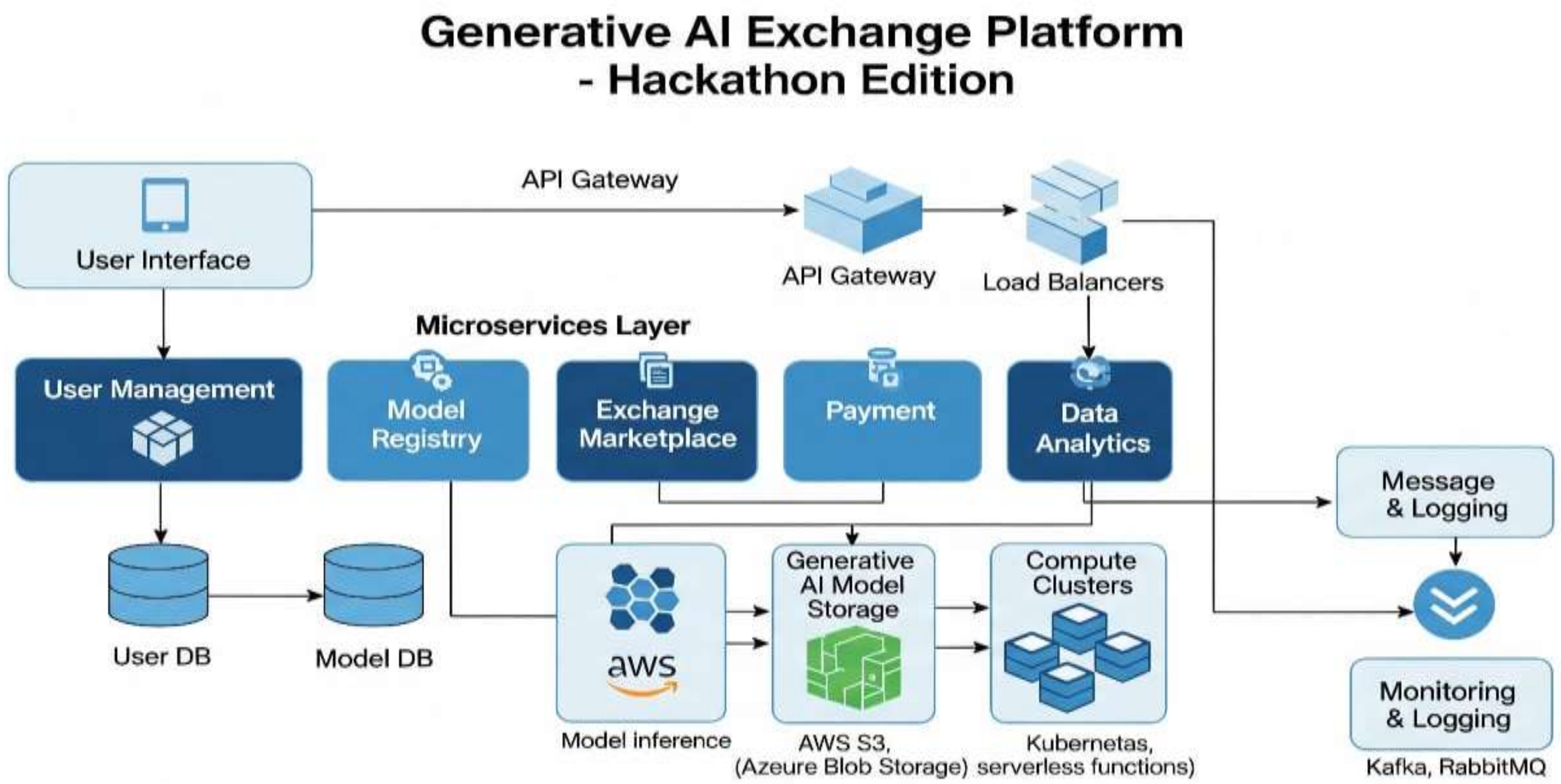
Compute Resources

- GPU Instance (Small) ○
- GPU Instance (Medim)
- GPU Instance (Mrgn) ●
- GPU Instance (Large ●

Project Submission

Project Title	Team Members		
Model Used	Model Used	Data Used	Compute Used
Compute			

Architecture Diagram Of the Proposed Solution :



Technologies to be Used:

❑ 1. Core AI / ML Stack

- ❑ Large Language Models (LLMs): OpenAI GPT, LLaMA, Falcon, or fine-tuned models.
- ❑ RAG (Retrieval Augmented Generation): For domain-specific knowledge.
- ❑ Embeddings & Vector Databases: Pinecone, FAISS, Weaviate.

❑ 2. Development Frameworks

- ❑ LangChain / LlamaIndex: For chaining prompts and building workflows.
- ❑ Hugging Face Transformers: Pre-trained models & fine-tuning.
- ❑ PyTorch / TensorFlow: For custom ML model development.

➢ 3. Application Layer

- ➢ Backend: FastAPI, Flask, Node.js.
- ➢ Frontend: React.js, Streamlit, or simple dashboards.
- ➢ Chat UI: Gradio, Streamlit, or custom web interface.

➢ 4. Infrastructure & Deployment

- ➢ Cloud Platforms: AWS, GCP, Azure (for GPU/compute).
- ➢ Model Hosting: Hugging Face Spaces, Replicate, or custom API.
- ➢ Containerization: Docker + Kubernetes (if scaling needed).

➢ 5. Supporting Tools

- ➢ Databases: PostgreSQL / MongoDB.
- ➢ Visualization: Plotly, Power BI, Tableau.
- ➢ Collaboration: GitHub, Notion, Slack.

Estimated Implementation Cost :

- **1. Core AI & ML**
- Pre-trained LLM API usage (e.g., OpenAI / Anthropic): $50–150 / month (depending on API calls).
- Open-source LLM (e.g., LLaMA, Falcon) + fine-tuning: $200–500 (GPU credits).
- **2. Infrastructure**
- Cloud compute (AWS/GCP/Azure – basic GPU/CPU): $100–200 / month.
- Storage (vector DB + app data): $20–50 / month.
- **3. Development Tools**
- LangChain, Hugging Face, Streamlit: Free / Open-source.
- GitHub, collaboration, CI/CD: $10–20 / month.
- **4. Deployment**
- Hosting (Hugging Face Spaces / Replicate / Cloud VM): $30–70 / month.
- Domain & SSL: $10–15 / year.

Add as Per the Requirements For the Hackthon :

❑ **Prototype (Hackathon Stage)**

❑ AI/ML APIs & Tools → Free (credits/open-source)

❑ Hosting (basic) → $0–50

Total: Near Zero

❑ **Pilot (Post-Hackathon)**

❑ API Usage → $50–150 / month

❑ Cloud + Storage → $100–200 / month

❑ Deployment → $30–70 / month

Total: ~$200–400 / month

❑ **Scaling (Full Implementation)**

❑ Enterprise Cloud + GPUs → $500–1000 / month

❑ Advanced APIs/Databases → $300–500 / month

❑ Maintenance → $100–200 / month

Total: ~$1000–1500 / month

❑ **Visual Tip for Slide** → Use a 3-level pyramid or stacked bar with:

❑ Top: Prototype (~$0)

❑ Middle: Pilot ($200–400)

❑ Base: Scaling ($1000–1500)

Thank You!

Generative AI Exchange
Hackathon

See you next year!