**1. Project Summary**

As part of the CMAP 2025 internal competition, our team (Team B) is building an AI-powered asset finance assistant to streamline the customer onboarding process. The goal is to develop a full-stack web application with a smart conversational interface that can assess loan eligibility and guide users through the initial financing stages.

**2. Objectives**

* Build a working AI assistant that can collect applicant data via natural conversation.
* Automatically trigger loan eligibility assessments once sufficient information is gathered.
* Present suitable loan products with rates and documentation requirements.
* Provide a seamless experience between structured form inputs and chatbot interaction.

**3. Scope**

Our application will consist of:

* **Left Panel**: A dynamic, JSON schema–based form for structured inputs.
* **Right Panel**: A chatbot interface where the AI agent collects data and returns recommendations.

The AI Agent will:

* Collect asset finance information via dialogue.
* Detect when minimum data is complete.
* Call the backend API to assess loan eligibility.
* Display loan product suggestions and prompt for optional improvements.

**Clarification Requested:**  
We would like to confirm whether the chatbot’s role ends after product recommendation, or if it can be extended to approve and initiate loan disbursement. Is online loan approval/disbursement within the project scope?

**4. Team Members and Roles**

| **Name** | **Role** | **Responsibilities** |
| --- | --- | --- |
| Member A | Backend Developer | FastAPI backend, API integration |
| Member B | Frontend Developer | React UI, form rendering |
| Member C | AI Prompt Engineer | AI agent design, prompt tuning, OpenAI API usage |
| Member D | Full-stack / Integration | Connect backend/frontend, deployment |

**5. Technology Stack**

| **Component** | **Technology Used** |
| --- | --- |
| Frontend | React, Vite, Tailwind CSS |
| Backend | FastAPI (Python), Pydantic, JSON Schema |
| AI Integration | OpenAI API (GPT-3.5 / GPT-4), LangChain (optional) |
| Version Control | Git + GitHub |
| Deployment | Render / Vercel / Fly.io |

**6. Timeline & Milestones**

| **Week** | **Task** |
| --- | --- |
| 1 | Project planning, environment setup, team role alignment |
| 2 | Build basic React & FastAPI skeleton, connect GitHub |
| 3 | Implement chatbot + JSON-based form, start prompt tuning |
| 4 | Integrate backend loan API, test minimal information flow |
| 5 | Add improvements, refine AI logic, work on UI polish |
| 6 | Final integration, testing, documentation, and presentation |

**7. Resources & Token Budget**

* We will use the $50 token budget primarily for OpenAI API (gpt-3.5/gpt-4) during development and testing.
* Token usage will be tracked and monitored to avoid overruns.

**8. Risk Management**

| **Risk** | **Mitigation Strategy** |
| --- | --- |
| Inconsistent AI output | Improve prompt design and fallback messages |
| Backend API not ready in time | Mock responses for frontend integration and testing |
| Team bandwidth issues | Weekly sync-up and task redistribution if needed |
| Scope creep: loan approval functionality | Confirm with stakeholders whether it's in-scope early on |

**9. Success Criteria**

* Working AI assistant with both chatbot and form input.
* Accurate triggering of backend API at correct stages.
* Clear product recommendations and user feedback flow.
* Frontend/backend properly deployed and integrated.
* Code committed regularly with clear documentation.