1. Project Brief (Created by the Analyst)

This document captures the initial vision, goals, and high-level scope.

Markdown

```
# Project Brief: Meeting Assistant

## Introduction / Problem Statement

As a manager of a high-performing technical team, my day is fragmented by back-to-back meetings and an overwhelming volume of communications across email and Slack. This constant multitasking leads to missed information, creates significant follow-up work, and leaves little time for deep strategic thinking, technical supervision, and maintaining a healthy work-life balance. I need a personal assistant that can act as a second brain, specifically during meetings, to capture critical information, provide real-time context, and automate post-meeting administrative tasks, thereby restoring focus and maximizing efficiency.

## Vision & Goals

- **Vision:** To create a seamless, intelligent personal assistant that empowers managers to be fully present and strategically effective in their roles by automating administrative burdens and providing timely, contextual intelligence.
- **Primary Goals:**
  - Goal 1: Reduce time spent on post-meeting administrative tasks (summarizing, identifying action items) by at least 75% within the first month of use.
  - Goal 2: Increase the user's ability to recall key decisions and facts from meetings, measured by a significant reduction in "follow-up for clarification" emails or messages.
  - Goal 3: Provide real-time, relevant context during at least 50% of meeting discussion points, enabling more informed contributions without the need for manual context switching.

## Target Audience / Users

- **Primary User:** A leader or meeting facilitator within a dynamic organization. This individual is responsible for driving outcomes and is accountable for the effective use of their and their team's time. They are proficient with technology but are overwhelmed by the volume of meetings and digital communication, leading to cognitive load and context-switching. They work on their own hardware (Windows 10/11) and possess a large repository of existing documentation and artifacts that the assistant should use as a primary knowledge source for
```

providing context and insights. The user needs a solution that integrates deeply with their existing toolset (M365, Google Suite, Jira, Slack, etc.) to reclaim focus and improve personal productivity.

Key Features / Scope (High-Level Ideas for MVP)

- **Dynamic Pre-Meeting Briefings:** The assistant will analyze the user's calendar, communications, and knowledge base to generate an intelligent briefing before each meeting. This includes a "vibe check" on attendees and allows the user to add their own reminders.
- **Real-Time Intelligence Overlay:** During meetings, the assistant will display a widget providing a live transcript and "spoon-fed" contextual information pulled in real-time from the user's documents, emails, Slack, and other integrated sources.
- **Automated Post-Meeting Debrief Package:** After a meeting concludes, the assistant will produce a comprehensive package that includes a searchable, AI-tagged transcript, a summary of key points, a list of action items and decisions, and the AI's own observations and recommendations.

Post MVP Features / Scope and Ideas

- **Communications Tamer:** Develop a full suite of tools for email and Slack management.
- **Productivity Guardian:** Implement features for active time tracking and a "focus guardian."
- **Project & Team Orchestrator:** Build out features for team management and deeper workflow integrations.
- **Personal Aide & Admin:** Introduce capabilities for travel assistance, surveys, and approvals.
- **Advanced Agentic Features:** Explore a general-purpose background agent, adjustable AI "attitudes," and a dedicated research mode.

Known Technical Constraints or Preferences

- **Primary Operating System:** Windows 10 and Windows 11.
- **Hardware:** The application will run on user-owned hardware (Desktop/Laptop).
- **Deployment Preference:** The preferred architecture is a hybrid model: a native desktop application will serve as the primary UI/UX, while most heavy lifting will be handled by a scalable SaaS backend.
- **Knowledge Base:** The assistant must be able to index and use a large, existing corpus of the user's personal documentation and artifacts.
- **Database Preference:** A hybrid database structure is preferred. A NoSQL database for unstructured data and a relational database for structured data like user settings. Potential NoSQL options include Firebase, Supabase, or Couchbase.
- **Language Preference:** Python is the preferred language, but the user is open to other suggestions.

- **Cost & Technology Constraints:** Solutions should be cost-effective.
- **Core Integrations:** The system must be flexible enough to integrate with M365, Google Suite, Jira, Confluence, and Slack.
- **LLM Integration:** The system will use commercial LLM APIs (Claude, Gemini, OpenAI).

---

2. Product Requirements Document (PRD) (Created by the Product Manager)

This is the master plan, containing the formal requirements and the detailed Epics and Stories for the MVP.

Markdown

```
# Meeting Assistant Product Requirements Document (PRD)

## 1. Goal, Objective and Context
*(Content is identical to the Project Brief and is omitted here for brevity, but should be included in your file)*

## 2. Functional Requirements (MVP)
*(Content as defined and approved in our conversation)*

## 3. Non-Functional Requirements (MVP)
*(Content as defined and approved in our conversation)*

## 4. User Interaction and Design Goals
*(Content as defined and approved in our conversation)*

## 5. Technical Assumptions
*(Content as defined and approved in our conversation, ending with the key performance consideration for the Architect)*

## 6. Epic and Story Structure (MVP)
### Epic 1: Foundational Setup & Knowledge Ingestion
*(List of all 7 stories as defined)*
### Epic 2: The Pre-Meeting Experience
*(List of all 6 stories as defined)*
### Epic 3: The Live Meeting Intelligence Engine
```

(List of all 8 stories as defined, including the mode-switching story)
Epic 4: The Post-Meeting Debrief Wizard
(List of all 12 stories as defined, including the final data retention policy)

---

3. UI/UX Specification (Created by the Design Architect)

This document defines the application's look, feel, and flow.

Markdown

```
# Meeting Assistant UI/UX Specification

## 1. Overall UX Goals & Principles
*(Content as defined and approved, including the revised "Leader / Meeting Facilitator" persona)*

## 2. Information Architecture (IA)
*(Includes the Mermaid diagram for the site map)*

## 3. User Flows
*(Includes the Mermaid diagram for the primary user journey)*

## 4. Wireframes & Mockups
*(Content as defined, including the revised Dashboard and glassmorphic, autohiding Live Meeting Widget)*

## 5. Component Library / Design System Reference
*(Specifies the "headless" UI component strategy with Radix/Headless UI and Tailwind CSS)*

## 6. Branding & Style Guide Reference
*(Specifies the use of the provided logo and the full CSS variable theme for light/dark modes)*

## 7. Accessibility (AX) Requirements
- **Target Compliance:** WCAG 2.1 AA.

## 8. Responsiveness
*(Content as defined for the desktop-focused MVP)*
```

---

4. Architecture Document (Created by the Architect)

This is the master technical blueprint for the project.

Markdown

```
# Meeting Assistant Architecture Document

## Introduction / Preamble
*(Content as defined)*

## Table of Contents
*(As defined)*

## Technical Summary
*(Content as defined)*

## High-Level Overview
*(Includes the System Context Mermaid diagram)*

## Architectural / Design Patterns Adopted
*(As defined)*

## Component View
*(Includes the internal components Mermaid diagram)*

## Project Structure
*(Includes the two repository structures for backend and client)*

## API Reference
*(Includes the definitions for the three session management endpoints)*

## Data Models
*(Includes the schemas for MeetingDebrief, Contact, and ContactProfile)*

## Definitive Tech Stack Selections
*(The final, complete table of all technology choices, including Supabase, FastAPI, Electron, etc.)*
```