Information Management System

Criterion A - Planning

Table of contents:

Defining the Problem:

Isht, a student, currently relies on a combination of pen and paper methods to manage his academic workload, creative ideas, and study notes. While functional to a degree, this system, relying on notebooks, loose sheets, and sticky notes, has become increasingly cumbersome and described by Isht as "walking on eggshells" due to its fragility and potential for errors.

Isht's workflow revolves around four core academic tasks: Note-Taking/Idea Recording, Planning and Task Management, Studying and Revision, and Self-Motivation/Mental Organisation (as detailed in the consultation - Appendix 1). Currently, these tasks are handled across various physical mediums, prioritizing task-specific convenience over overall process efficiency. This results in fragmented information scattered across multiple paper sources, particularly impacting the consistent and accurate recording of daily activities and actions.

Common errors arising from this paper-based system include: data misplacement (losing notes or task lists), redundant data entry (duplicating information across different locations), and incomplete or inconsistent records of daily tasks and ideas. For example, a key idea jotted down on a loose sheet might be easily lost or forgotten, leading to missed opportunities or duplicated effort. These errors frequently necessitate time-consuming and mentally draining "paper purges," where Isht must manually sort through accumulated papers to consolidate information onto a few organized sheets – a temporary fix before the system degrades back into disarray.

Ultimately, Isht seeks to minimize the time and cognitive load associated with managing his disparate paper-based systems. He desires a "plug and play" solution; one that requires minimal setup and is immediately usable, unlike more complex Personal Knowledge Management applications such as Obsidian or Notion, which present a significant initial time investment in configuration and learning.

(Refer to Appendix I)

Word Count: 274

Rationale for proposed solution:

Isht struggles with a fragmented paper-based workflow for notes, tasks, and ideas, leading to data misplacement, redundant entries, and wasted time. Existing Personal Knowledge Management (PKM) software like Obsidian/Notion are unsuitable due to subscription costs, complex setup, and plugin-related learning curves - hindering immediate convenience, a key client need.

Thus, a custom web application is the optimal solution, precisely tailored to Isht's workflow for a minimal learning curve and a truly plug-and-play experience. Its web-based nature ensures cross-device access, eliminating paper management.

For robust frontend development, Next.js with React, leveraging TypeScript, is chosen. React's component-based architecture ensures modularity and maintainability for this feature-rich application. Next.js enhances user experience with server-side rendering and a fast interface. TypeScript, a JavaScript superset with static typing, further improves code quality, reduces runtime errors, and enhances long-term maintainability. Firebase Firestore provides real-time data synchronization and a serverless backend, simplifying deployment and ensuring accessible data across devices without complex setup. Shadcn UI with Tailwind CSS offers pre-built, customizable UI components, accelerating development of a consistent, user-friendly interface, aligning with the 'intuitive user interface' success criterion. Clerk authentication ensures secure and efficient user management, a standard web application practice.

In conclusion, a bespoke web application using this technology stack, including TypeScript, is meticulously justified by Isht's need for a convenient, efficient, and immediately usable Information Management System. This tailored solution directly addresses paper-based inefficiencies, providing a user-friendly platform for streamlined academic tasks and idea management, saving valuable time and cognitive load.

Word Count: 244

Success Criteria for the Information Management System

The following success criteria were decided based on the first interview conducted (Appendix I) and the Work Shadowing (Appendix II) and were finalized after having a second interview (Appendix I) with the client.

Word Count: 33

General Functionality (Applies across Modules):

1. **Content Creation:** The user can create new items (cookies, doubts, ideas, notes, sections, columns, spaces, windows) within each module, specifying necessary attributes (name, description, title, content, etc.).
2. **Content Persistence:** Items created by the user within each module are saved in the application and remain accessible upon reopening the application.
3. **Content Editing:** The user can modify the attributes (name, description, title, content, solution, etc.) of existing items across all modules where editing is applicable.
4. **Content Deletion:** The user can delete items (cookies, doubts, ideas, notebooks, sections, columns, notes, spaces, windows) from each module, removing them permanently from the application.
5. **Content Reordering (Drag and Drop):** Where applicable (Cookie Jar, To-Do List, Continuous Info Space, Stage Manager), the user can reorder items using drag-and-drop functionality, and this new order is saved.
6. **Voting Functionality (Doubts & Ideas):** The user can upvote and downvote doubts and ideas to indicate their importance or value.
7. **Commenting Functionality (Doubts & Ideas):** The user can add, edit, and delete text-based comments on doubts and ideas for discussion and clarification.
8. **Status Management (Doubts & Ideas):** The user can manage the status of doubts and ideas, marking them as 'Resolved' (and providing a solution) and reopening them as 'Open'.

Module-Specific Criteria:

9. **Cookie Jar - Cookie Management:** The user can effectively manage cookies within the Cookie Jar, leveraging creation, persistence, editing, deletion, and reordering functionalities. Cookies refer to achievements of the user that they feel motivated by, the idea of a cookie jar has been inspired by David Goggins.
10. **Doubt Tracker - Doubt Resolution:** The user can effectively utilize the Doubt Tracker to log, manage, and resolve doubts, including marking resolution status and providing solutions.

11. **Curiosity Space (Idea Tracker) - Idea Resolution:** The user can effectively utilize the Curiosity Space to capture, manage, and resolve ideas, including marking resolution status and providing solutions.
12. **To-Do List - Task Management:** The user can effectively utilize the To-Do List to manage tasks, including section and column organization, note and subtask creation, task completion tracking (checking/unchecking), and task reorganization via drag-and-drop.
13. **Continuous Information Space - Notebook Management:** The user can effectively utilize the Continuous Information Space to create and manage notebooks, sections, columns, and notes, with hierarchical organization and note reordering within notebooks.
14. **Stage Manager - Space & Window Management:** The user can effectively utilize the Stage Manager to create and manipulate Spaces and Windows, including window movement and resizing, and ensure persistence of layouts across sessions.

General Application Success (Usability, Performance, Robustness):

15. **Intuitive User Interface:** The application features intuitive navigation and a clean, uncluttered user interface, making it easy to use with minimal setup ("Plug and Play").
16. **Cross-Browser & Device Compatibility:** The application functions correctly and is usable on major web browsers (Chrome, Firefox, Safari, Edge) and across different devices (desktop, laptop, tablet).
17. **User Authentication & Security:** The application securely authenticates users, protecting user data and ensuring only authorized users can access their information.
18. **Real-time Data Synchronization:** Data within the application is synchronized in real-time across different devices when logged in with the same user account.
19. **Fast Data Retrieval:** The application retrieves and displays user data (cookies, doubts, ideas, notes, etc.) quickly and efficiently, ensuring minimal loading times and a responsive user experience.
20. **Component-Based Architecture for Maintainability and Scalability:** The application is developed using a component-based architecture, promoting code modularity, reusability, and maintainability, and designed with a structure that can potentially accommodate a growing user base and future feature additions.