

Overview

- **Core Features:** The app will feature AI-driven capabilities to generate custom architecture designs, user workflows, screens, and cost breakdowns based on conversations with the user. These features aim to translate non-technical inputs into comprehensive, developer-ready technical specifications.
- **User Workflow:** Non-technical founders will interact with the app through a chat interface, where they will input essential information such as core features, problem statements, user workflows, and software platform requirements. This input will be processed by an AI tool using a multi-prompt approach to generate a detailed technical report.
- **Problem Statement:** Non-technical founders, especially those looking to build tech-based startups, face significant challenges in finding trustworthy development partners. They often rely on word-of-mouth recommendations due to a lack of technical expertise, which can lead to miscommunications and suboptimal hiring decisions.
- **Software Platform:** The application will be developed for the web platform, utilizing AI and multi-prompt techniques to ensure accurate and practical outputs that align with software development best practices.

Cost Breakdown

Costs are calculated based on the estimated hours of a single full-stack developer at a rate of \$25 / hour

Description	Hours	Total (\$)
Input Project Requirements	40	\$1000
Generate Architecture Designs	60	\$1500
Create User Workflows	50	\$1250
Generate Cost Breakdowns	30	\$750
Produce Developer-Ready Specifications	70	\$1750
Review and Confirm Outputs	25	\$625
Download Technical Report	15	\$375
Secure User Authentication	20	\$500
Total	310	\$7750

User Stories / Tasks

- **Input Project Requirements:** As a non-technical founder, I want to input my project requirements using a chat interface so that I can easily communicate my ideas without needing technical expertise.
- **Generate Architecture Designs:** As a non-technical founder, I want the system to automatically generate architecture designs based on my inputs so that I can ensure the technical foundation aligns with my project needs.
- **Create User Workflows:** As a non-technical founder, I want the app to generate user workflows from my descriptions so that I can visualize how users will interact with my product.
- **Generate Cost Breakdowns:** As a non-technical founder, I want to receive a cost breakdown for the proposed architecture so that I can plan my budget and make informed financial decisions.
- **Produce Developer-Ready Specifications:** As a non-technical founder, I want the app to generate developer-ready technical specifications from my inputs so that I can directly hand them over to a development team without further translation.
- **Review and Confirm Outputs:** As a non-technical founder, I want to review and confirm the outputs generated by the AI, including architecture designs and cost estimates, on a summary screen so that I can ensure accuracy before finalizing the specifications.
- **Download Technical Report:** As a non-technical founder, I want to download the final technical report as a PDF so that I can share it with stakeholders and development teams.
- **Secure User Authentication:** As a non-technical founder, I want to securely log in to the application so that my project information and reports are protected and personalized.

Core Features

- **AI-Driven Custom Architecture Design:** This feature utilizes AI to automatically generate architecture designs for software projects based on user inputs. It translates non-technical descriptions into structured, developer-ready architecture diagrams, ensuring that foundational design aligns with best practices and user requirements.
- **Interactive Chat Interface:** A user-friendly chat interface where non-technical founders can input their project requirements. This interface serves as the primary interaction point, collecting essential information such as core features, problem statements, and platform needs, which are crucial for generating accurate technical specifications.
- **Multi-Prompt AI Processing:** An AI tool that processes inputs received through the chat interface using a multi-prompt approach. This feature ensures comprehensive understanding and interpretation of the user's requirements, facilitating the generation of detailed technical reports and specifications.
- **Cost Breakdown Generation:** This feature provides an estimated cost breakdown for the developed architecture. It helps users understand the financial implications of their design choices, aiding in budget planning and decision-making processes.
- **Developer-Ready Technical Specifications:** The final output of the app includes detailed, developer-ready technical specifications generated from non-technical user inputs. This feature ensures that the technical documents are precise, actionable, and ready to be used by development teams, reducing miscommunication and streamlining project kick-off.

Architecture Design

- **Web Application:** The front-end of the MVP will be a simple, responsive web application designed to provide a user-friendly interface for non-technical founders. It will feature a chat interface where users can input their requirements. The web application will be developed using React.js due to its component-based architecture, which allows for efficient development of dynamic user interfaces. React.js is also widely supported and has a large community, making it a reliable choice for an MVP.
- **Backend:** The backend will handle business logic, AI interactions, and data management. It will be developed using Node.js, which allows for building scalable and fast server-side applications using JavaScript. This will simplify the development process as the same language can be used on both the front-end and back-end. Node.js is also well-suited for handling I/O-bound operations, which is essential for the chat-based interactions and AI computations in this application.
- **AI Model:** The core feature of the MVP involves AI-driven capabilities to generate technical specifications from non-technical user inputs. We will integrate a pre-trained NLP model using services like OpenAI's GPT-3, which can process natural language inputs and generate detailed, context-aware outputs. This approach minimizes the need for initial AI training and infrastructure, leveraging cloud-based AI services to handle the processing and complexity.
- **Data Storage:** User data, session information, and generated reports will be stored in a MongoDB database. MongoDB is a NoSQL database that offers flexibility and scalability, which is beneficial for MVPs that may evolve rapidly based on user feedback. It supports storing data in a JSON-like format, which aligns well with the JavaScript-based stack and simplifies the handling of data from the web application and backend.
- **Authentication:** To manage user sessions and secure access to the application, a simple authentication mechanism will be implemented. JSON Web Tokens (JWT) will be used for this purpose, providing a stateless and scalable method of user authentication. This will involve basic signup and login functionalities to ensure that user data and generated reports are protected and personalized.
- **Deployment:** The MVP will be deployed on a cloud platform like Heroku or AWS Elastic Beanstalk. These platforms offer managed services that simplify the deployment, monitoring, and scaling of web applications. They are cost-effective for MVPs, requiring minimal setup and maintenance, allowing the team to focus on product development and iteration rather than infrastructure management.

Users

- **Non-technical Founders:** This primary user group consists of individuals looking to start tech-based businesses but lack the technical expertise to articulate their needs into developer-ready specifications. They will use the app to input their ideas and requirements through a chat interface, which the AI will then translate into detailed technical reports, architecture designs, and cost analyses. This group is crucial as they are direct beneficiaries of the app's core functionality, helping them overcome challenges in hiring and communicating with development teams.

User Flow: Non-technical Founders

1. **Landing Page:** Users arrive at a simple landing page that introduces the app's purpose and core features. It includes a prominent 'Start Project' button to engage users immediately.
2. **Signup/Login:** Prompt users to create an account or log in to ensure personalized and secure access. This step is crucial for managing user sessions and storing individual project data.
3. **Initial Chat Interface:** After logging in, users are directed to the chat interface. Here, they start by describing their project needs, such as core features and problem statements. The AI uses prompts to guide users through providing the necessary information.
4. **AI Processing and Feedback:** The AI processes the inputs using multi-prompt techniques and generates preliminary architecture designs and cost estimates, which are then presented to the user for review.
5. **Summary and Confirmation Screen:** Users review the AI-generated project summary, including technical specifications and cost breakdowns. They can edit details or confirm to proceed. This step ensures that the user agrees with the generated outputs before finalizing.
6. **Final Technical Report:** Upon confirmation, the final detailed technical report is displayed. Users can download this report as a PDF. This report includes comprehensive details like architecture designs, user workflows, and UI/UX screens.
7. **Start New Project or Logout:** After downloading the report, users have the option to start a new project or log out. This provides a clear end to the session and an easy pathway to begin another project if needed.

Wireframe Screens

- **Chat Interface:** This screen serves as the primary interaction point for users. It features a clean and simple chat interface similar to popular messaging apps. At the top, there is a header with the app logo and a 'Help' button. The main area is dedicated to the chat window where users can type their inputs. The chat interface includes a text input field at the bottom with a 'Send' button to submit information. Above the input field, there's a small section displaying tips or examples of what kind of information the user can provide, aiding in guiding the conversation. The chat history is displayed above, where responses from the AI and user inputs are clearly differentiated by color and alignment.

App Logo	Help
Tips/Examples	
	Send

- **Summary and Confirmation Screen:** After the user has provided all necessary information through the chat interface, they are directed to this screen. It displays a summary of the inputs provided and the generated outputs including the architecture design, user workflows, screens, and cost breakdowns. Each section is presented in collapsible panels that the user can expand or collapse to view details. At the bottom of this screen, there are two buttons: 'Confirm' to proceed with the specifications or 'Edit' to return to the chat interface for modifications. This screen ensures that users can review and verify all details before final confirmation.

Summary and Confirmation Screen

Architecture Design
User Workflows
Screens
Cost Breakdowns
<input type="button" value="Confirm"/> <input type="button" value="Edit"/>

- **Technical Report Screen:** This screen is accessible after the user confirms the details on the Summary and Confirmation Screen. It displays the final technical report generated by the AI. The report is divided into sections such as 'Architecture Design', 'User Workflows', 'UI/UX Screens', and 'Cost Breakdown'. Each section can be viewed in detail by clicking on it, expanding to show comprehensive information and diagrams. The layout is clean with a focus on readability, using typography and whitespace effectively. There is a 'Download Report' button at the top right corner allowing users to download the report as a PDF, and a 'Start New' button at the bottom to initiate a new project.

Technical Report Screen	<input type="button" value="Download Report"/>
Architecture Design	
User Workflows	
UI/UX Screens	
Cost Breakdown	
<input type="button" value="Start New"/>	