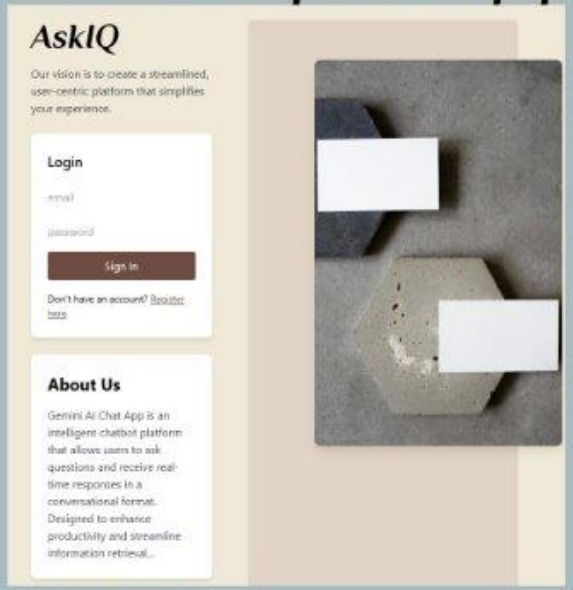
AskIQ

~AI Helper App~



https://askiq-live.netlify.app

Chingu Voyage 54
Tier 3-Team 35

Meet The Team

Doris John
Primary Scrum Master

Victoria Idris
Shadow Scrum master

Developers

Ahmed Sohail

Albert Ngodi

Conor barry

Maryam Hazrati

Zaid Hassan

Introduction & Project Overview

This presentation covers the AskIQ project. AskIQ is an innovative AI project designed to provide users with intelligent responses through a user-friendly interface. The project utilizes the Google Gemini API to analyze input from users and generate accurate and relevant answers based on data input via a pentagram form.

Project Development & Planning

AskIQ started off with a project kickoff getting to understand the team and their potential. After defining each of our roles, a low fidelity figma was created based on the MVP of the project and the selection of the tools like Jira to implement the agile practices and utilize the product backlog to initiate scrum/sprint planning, backlog refinement, and sprint review/retrospective.

MVP Live Demo

The key features and functionality of the AskIQ app are as follows:

1. **User Authentication & Registration**

Secure login and sign-up system allowing users to create and manage their accounts.

2. **Structured Query Input (Pentagram Format)**

Guided prompt input system using a five-point structure to help users craft high-quality queries.

3. **Google Gemini API Integration**

Seamless backend integration with the Gemini AI API to generate accurate and tailored responses.

4. **Database Connectivity**

Backend powered by PostgreSQL for storing user data, prompt history, and responses efficiently.

5. **Sidebar Prompt History**

Interactive sidebar that displays all of a user's previous queries for easy access and reuse.

6. **Response Display**

AI-generated answers are dynamically displayed after a prompt is submitted.

7. **Edit Previous Queries**

Users can modify and resubmit past prompts directly from their history.

8. **Form Validation & Error Handling**

Real-time form validation ensures inputs meet the required criteria, with user-friendly error messages on submission failures.

Tech Stack Used

The technology frameworks and tools used to develop the AskIQ app is as follows:

Front-end:

- React
- TypeScript
- Tailwind CSS

Back-end:

- Node.js
- Express.js
- PostgreSQL
- Google Gemini API
- Swagger (for API documentation)
- Docker

Project Management Tools:

- Jira, Mural, Zephyr, Figma, Discord , Github, Schej

Challenges & Mitigation Strategies

Deployment & Compatibility Issues

- Mitigation: Explored viable deployment options to ensure feasibility.

Team Resource Imbalance

- Mitigation: Conducted knowledge transfer sessions and pair programming to strengthen frontend capabilities.

Initial Communication Challenges

- Mitigation: Established dedicated Channel Threads for structured communication and frequent GitHub commits.

Time Constraints

- Mitigation: Used Schej and HammerTime to identify optimal meeting times, holding 3 one-hour meetings per week. Timeboxed events with prior agenda notifications.

Healthy Conflict

- Mitigation: Leveraged collaborative work assignments and skills mapping to enhance team bonding and task completion.

Future Development Opportunities & Lessons learned

Security Features Added

- Added libraries to check for malicious code, social authentication, email verification, pagination, enhanced history display, monitoring & logging of server response times, user password reset, and extended rate limits.

Lessons Learned

- Align early on product goals to maintain focus.
- Collaboration and communication drive continuous improvements and cohesive development.
- Embrace iteration and adaptability to pivot based on feedback.
- Prioritize user experience with simplicity and ease of use.

Feedback

Thank you for participating and any questions?

Github Repository link: https://github.com/chingu-voyages/V54-tier3-team-35/tree/development

AskIQ App Link: https://askiq-live.netlify.app/