

comp90082-2024-s2-sa-redback.....	3
Background.....	5
Project Overview.....	6
Client Goal.....	8
Scopes.....	9
Stakeholders.....	12
Technical Consideration.....	14
Development Environment.....	16
Ethical Consideration.....	18
Cyber Security Consideration.....	20
Requirements.....	23
Personas.....	24
Prototype.....	26
User Stories.....	28
Product Backlog.....	40
Non-Functional Requirements.....	49
Sprint.....	51
Sprint 1.....	52
Plan for Sprint 1.....	53
Sprint 1 Retrospective.....	54
Sprint 1 Review.....	55
Sprint 2.....	57
Plan for Sprint 2.....	58
Sprint 2 Retrospective.....	62
Sprint 2 Review with Client.....	64
Sprint 3.....	66
Plan for Sprint 3.....	67
Architecture.....	72
MongoDB Database Model.....	73
API.....	74
Deployment Decision.....	83
Collaboration.....	85
Team Standards.....	86
About Us.....	88
Agile Ceremonies.....	89
Sprint 1 Planning Meeting.....	90
Sprint 1 Retrospective Meeting.....	92
Sprint 2 Planning Meeting.....	93
Sprint 2 Retrospective Meeting.....	95
Sprint 3 Planning Meeting.....	97
Code Review.....	99
Sprint 2 code Review.....	100
Sprint 2 Front-end Code Review.....	101
Sprint 2 Back-end Code Review.....	104
Sprint 2 Integration Code Review.....	107

AI (Front-end) Code Review	109
AI (Back-end) Code Review	115
Testing Strategy	118
Product Demonstration	120
Meetings	121
Client Meetings	122
19/09/2024 Client Meeting	123
01/09/2024 Client Meeting	126
07/08/2024 Client Meeting	129
Mentor Meetings	132
18/09/2024 Mentor Meeting	133
11/09/2024 Mentor Meeting	134
04/09/2024 Mentor Meeting	135
28/08/2024 Mentor Meeting	136
21/08/2024 Mentor Meeting	137
14/08/2024 Mentor Meeting	138
07/08/2024 Mentor Meeting	139
04/08/2024 Mentor Meeting	140
Team Meetings	141
10/09/2024 Team Meeting	142
03/09/2024 Team Meeting	144
29/08/2024 Team Meeting	145
27/08/2024 Team Meeting	147
20/08/2024 Team Meeting	148
15/08/2024 Team Meeting	150
13/08/2024 Team Meeting	152
06/08/2024 Team Meeting	154

👋 Welcome to the SWISP - AI Chatterbox Project Space!

About the Project

Welcome to SWISP Lab's exciting journey into the digital world! We're taking our fun paper chatterboxes and turning them into a smart digital tool. Our new AI chatbot is here to create interesting "what if" questions that make you think in new ways.

Our AI chatterbox is perfect for hackathons and learning activities. It helps you explore new ideas about the future and imagine "what if" scenarios. Just type in a topic, and the chatbot will come up with eight unique questions to spark your imagination.

Whether you're thinking about climate change, future technology, or how society might change, our chatbot is here to help you explore new ideas. It's not just fun to use – it also saves all the questions and answers so you can learn from them later.

Join us in this world of "what if" thinking. Let's see where your imagination can take you!

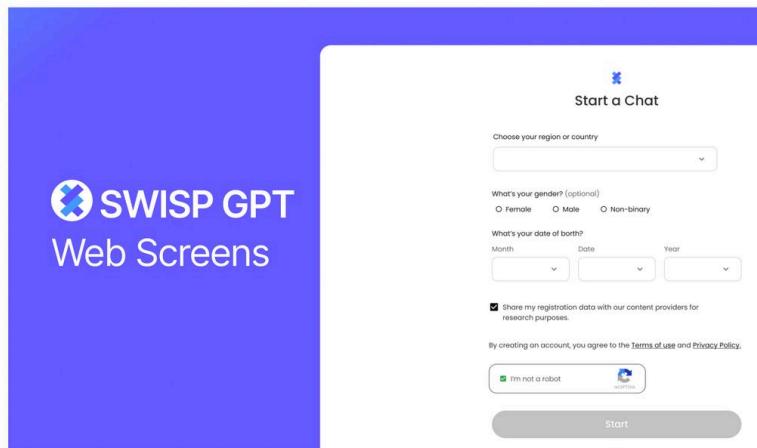


Table of Contents

[Collapse all](#)

[Expand all](#) [Collapse all](#)

Quick Links

[Trello Board](#)

[Github](#)

[Figma Prototype](#)

Meet the team



Yujun Yan

Mentor



Yiyao Li (Shaelyn)

Product Manager

[@YiYao Li](#)



Yingying Guo (Vicky)

Quality Assurance Leader

[@Yingying Guo](#)



Jionghao Song (Harry)

Developer

[@Jionghao Song](#)

Yiqun Liu (Elva)

Scrum Master

[@Yiqun Liu](#)



Leyao Lyu (Lydia)

Developer

[@Leyao Lyu](#)



Background

As technology advances, encouraging creative thinking is more important than ever. The SWISP Lab has developed paper "chatterboxes" to help people explore "what if" questions, sparking creativity. However, these tools face challenges in scaling and deep interaction. Many existing digital tools cannot also engage users meaningfully. The AI Chatterbox project, led by SWISP Lab, aims to solve this by using an AI-driven platform to bring the benefits of traditional tools into the digital age, making creative exercises more accessible and interactive for everyone.

- [Project Overview](#)
- [Client Goal](#)
- [Scopes](#)
- [Stakeholders](#)
- [Technical Consideration](#)
- [Development Environment](#)
- [Ethical Consideration](#)
- [Cyber Security Consideration](#)

Project Overview

Introduction:

The AI Chatterbox project seeks to bring the traditional paper-based chatterbox into the digital age by developing a web-based AI chatbot. Historically, the chatterbox has been used to prompt speculative thinking by asking "what if" questions, stimulating creativity and interdisciplinary dialogue. This project aims to replicate and enhance this experience through an AI-driven platform that generates speculative questions based on user input. Designed primarily for use in art galleries, this chatbot will engage users, particularly those aged 14-28, in thoughtful exploration of complex social and environmental issues.

Key Features:

Features

- **Web-Based Platform:** Accessible via a web interface, eliminating the need for app downloads. Users can engage with the AI Chatterbox from any device with internet access.
- **AI-Generated Questions:** The chatbot generates eight unique "what if" questions based on user input. These questions are designed to stimulate creativity and critical thinking on a wide range of topics.
- **Multilingual Support:** The platform supports multiple languages, making it accessible to a global audience. This feature ensures that users from different linguistic backgrounds can interact with the chatbot effectively.
- **No Login Required:** Users can start interacting with the chatbot immediately without creating an account or logging in, ensuring a seamless and hassle-free experience.
- **Adaptive Question Generation:** The system adapts its question generation based on user input, allowing for a diverse range of speculative questions tailored to various topics.
- **Data Collection and Analysis:** Collects and securely stores user interactions for future research and analysis. This feature helps understand user behaviour and improve the system based on feedback.
- **QR Code Access:** Simplifies user access by providing QR codes directly to the chatbot. This feature ensures easy and quick interaction, especially in public or gallery settings.
- **Output Filtering:** Ensures that generated questions adhere to safety and appropriateness standards by filtering out any potentially harmful or inappropriate content.
- **User Feedback Mechanism:** Provides options for users to give feedback on the questions and their experience, helping to refine and improve the chatbot's performance and content.

Benefits:

- **Enhanced Engagement:** The AI Chatterbox fosters deep engagement by encouraging users to explore speculative scenarios and think critically about complex issues.
- **Accessibility:** Being web-based and login-free, the platform is accessible to a wide audience, ensuring ease of use across different devices and locations.
- **Data-Driven Insights:** By collecting and storing interaction data, the platform allows for valuable research insights, helping to understand user behavior and preferences.
- **Cultural Relevance:** With multilingual support and adaptability to diverse inputs, the AI Chatterbox can be used across various cultural contexts, making it relevant and inclusive.
- **Budget-Friendly Implementation:** Utilizing OpenAI's API allows for a cost-effective solution that still provides high-quality, AI-generated content.
- **Sustainability:** The digital nature of the AI Chatterbox ensures long-term sustainability by reducing the need for physical materials and allowing for continuous updates and improvements, making it an environmentally friendly solution.

- **Seamless Accessibility:** The platform is engineered for effortless access, enabling users to quickly engage with the AI Chatterbox by scanning a QR code. This streamlined approach eliminates the need for downloads or complex setup, ensuring a smooth and efficient user experience.

Client Goal

1. **Web-Based AI Chatbot:** Develop a web-based AI chatbot that generates eight speculative "what if" questions based on user input. This tool will serve as a conversation starter in art galleries.
2. **Replacement of Manual Process:** The chatbot will replace the current manual method of generating speculative questions, streamlining the process.
3. **Critical Thinking Promotion:** The chatbot aims to encourage users to think critically about complex issues, including climate change, racism, and digital inclusion.
4. **Fully Digital System:** The system must be entirely digital, eliminating the need for printed materials or physical templates.
5. **No User Login Required:** The platform should not require users to log in, ensuring easy access and usability.
6. **User-Friendly Design:** The chatbot should be accessible and easy to use for individuals with varying levels of expertise in speculative inquiry.
7. **Theme-Specific Question Generation:** The generated "what if" questions will be directly related to the specific theme provided by the user's input.
8. **Low Budget Implementation:** The project has a low budget, and the client is open to using OpenAI's API for generating the speculative questions.
9. **Data Archiving for Research:** All generated data must be collected and archived, allowing for future analysis, which will contribute to ongoing learning and exploration.

Scopes

Inner Scope

Scopes	Objectives	Functionalities
AI Chat Box	<ul style="list-style-type: none">Develop an AI chatbot which allows generate 8 hypothetical questions based on user inputs.Users can interact with the chatbox, such as regenerate response.	<ul style="list-style-type: none">Automatically generate questions fostering creativity and critical thinking.Provide an interactive interface with users. Users can interact with the chatbox through keywords based on their preferences.
Data Collection	<ul style="list-style-type: none">Collect data on user interactions with the chatbot to improve system performance and study user behavior.	<ul style="list-style-type: none">Securely store data.Provide retrieval and feedback mechanisms.
Web Server	<ul style="list-style-type: none">Provide a stable, secure, and scalable server infrastructure to support the chatbot's operation.Allow users globally access the website.	<ul style="list-style-type: none">Implement high concurrency processing.Implement data security.Provide performance monitoring.
Web Development	<ul style="list-style-type: none">Create a user-friendly and responsive web interface seamlessly integrated with the AI chatbot.	<ul style="list-style-type: none">Provide an intuitive user interface.Optimize user experience suitable for various devices.
QR Code Accessing	<ul style="list-style-type: none">Simplify user access to the chatbot and related resources via QR codes.	<ul style="list-style-type: none">Generate QR codes for accessing the website.Providing instant access and usage tracking by scanning QR codes.
Input Filtering	<ul style="list-style-type: none">Identify discriminatory, offensive, or otherwise inappropriate content to ensure safety and appropriateness.	<ul style="list-style-type: none">Automatically detect any discriminatory, offensive, or impolite user inputs.Automatically filter any discriminatory, offensive, or impolite user inputs.
Output Filtering	<ul style="list-style-type: none">Filter output content to prevent responses that might lead to a poor user experience, ensuring the interaction with the chatbot complies with relevant laws and ethical standards, avoiding the spread of harmful, illegal, or inappropriate information.	<ul style="list-style-type: none">Automatically detect inappropriate or potentially harmful content output.Automatically filter inappropriate or potentially harmful content outputProvide user feedback options to report or flag inappropriate outputs to improve the content filtering mechanism.
History	<ul style="list-style-type: none">Provide users with access to and review their interactions with the	<ul style="list-style-type: none">Automatically record interaction history

	chatbot to facilitate continuous learning.	<ul style="list-style-type: none"> Provide search and review capabilities for history.
Multi-language	<ul style="list-style-type: none"> Expand the chatbot's multi-language support to cover more global users. 	<ul style="list-style-type: none"> Automatically detect and switch languages by users' IP address or location, ensuring accurate translation and cultural adaptation.
Admin Console	<ul style="list-style-type: none"> Develop an Admin Console that allows administrators to adjust system parameters, particularly for question generation, ensuring the system can meet diverse research needs and provide flexible outputs tailored to specific studies. 	<ul style="list-style-type: none"> Provide a user-friendly admin interface to allow authorized personnel to modify parameters related to the question generation process. Allow real-time updates to the question generation process based on adjusted parameters, ensuring instant reflection of changes in system outputs. Provide monitoring tools to track how parameter adjustments impact system performance and question relevance, allowing administrators to refine settings based on feedback and data. Incorporate analytics tools to help administrators fine-tune parameters based on past user interactions and question performance, optimizing for future research needs.
Data Archiving	<ul style="list-style-type: none"> Collect users' age, location, and input content to enhance user satisfaction and engagement for future analysis and review. 	<ul style="list-style-type: none"> Regularly and automatically store important user interactions and system performance data. Apply necessary encryption measures to archived data to ensure data security.

Outer Scope

Scopes	Objectives	Functionalities
User Login	<ul style="list-style-type: none"> Ensure secure authentication, preventing unauthorized access and safeguarding user privacy. The system should be user-friendly and adhere to security standards such as encryption protocols and password strength requirements. 	<ul style="list-style-type: none"> Provide a user login interface with email/username and password input fields. Support multi-factor authentication (MFA) for enhanced security. Implement "Remember Me" and session management features for user convenience. Enable password recovery functionality for users who forget their login credentials.

User Data Deletion	<ul style="list-style-type: none"> Ensure users have the right to delete their personal data at any time, maintaining their privacy rights. Follow data protection regulations, such as GDPR, to guarantee the effective execution of user data deletion requests. 	<ul style="list-style-type: none"> Provide a user-friendly interface enabling users to submit data deletion requests conveniently. Provide confirmation steps and execute data deletion promptly upon user request confirmation.
Advanced AI Features	<ul style="list-style-type: none"> Develop AI capabilities to handle more complex user requests and scenarios. The system should be able to not only handle basic conversations but also offer personalized recommendations, contextual awareness, and multi-step task execution. 	<ul style="list-style-type: none"> Provide contextual memory and reasoning for multi-turn conversations. Support advanced natural language understanding and generation to handle open-ended queries. Offer personalized recommendations based on user history and interactions. Enable multi-step task handling, allowing the AI to execute complex instruction chains. Dynamically adjust the conversational tone and style to suit different user preferences and scenarios.
Integration with External Services	<ul style="list-style-type: none"> Ensure AI Chatterbox can integrate effectively with external services (e.g., LMS or social media platforms), allowing users to sync data, track learning progress, and share content easily. 	<ul style="list-style-type: none"> Allow users to share AI-generated content on social media platforms, promoting social interaction. Support OAuth or other standardized authentication methods for secure external platform login. Provide real-time data synchronization, ensuring a consistent user experience across platforms. Enhance data exchange flexibility, supporting cross-platform operations and report generation.

ChangeLog

- Change Description:** "Move multi-language support from outer scope to inner scope."
- Reason:** "Client expressed a critical need for multi-language functionality during Sprint 1 review."
- Impact:** "This change will affect the timeline of other features in the inner scope but will ensure the client's priority needs are met."
- Approvals:** Approved by the client during Sprint 1 review meeting with client.

Stakeholders

Stakeholders	Description	Level of Interest	Level of Influence	Potential Impact	Communication Plan
Kate Coleman and Sarah Healy	As clients of SWISP Lab, they represent the end-users and educational goals of the project, providing requirements and reviewing project results.	High	High	Their feedback and requirements determine the overall direction and ultimate success of the project. Ignoring their opinions could lead to the project deviating from its educational objectives.	Hold a meeting after each sprint to provide a project progress report, obtain timely feedback, and maintain direct communication with the product owner. Regularly send project update emails and conduct periodic showcases of phase results.
Development Team	Responsible for the actual development of the AI chatbot.	High	High	The team's capabilities and efficiency directly impact the quality and timeliness of the project's delivery. If the team is not well-coordinated or lacks resources, the project may not be completed on time.	Daily standups for progress reports, weekly sprint reviews, and planning meetings. Regular internal communication and problem-solving meetings.
14-28 years old teenagers	The primary users of the AI chatbot, interacting to generate and explore hypothetical questions.	High	Low	User feedback influences product usability and user experience but does not directly affect the project's development timeline. Ignoring user experience may cause the project to lose interest among its target audience.	User experience surveys and feedback forms, regular user testing, and focus group discussions. Collect user data for continuous improvement of product design.
Science Gallery Melbourne	Collaborative partner involved in practical application and testing, providing feedback on usage and user experience improvements.	Medium	Medium	Their feedback helps improve the AI chatbot's user experience and ensures smooth implementation in public events. Ignoring their input may result in unresolved issues during practical applications.	Bi-weekly partner feedback meetings to ensure their suggestions are integrated into the development cycle. Showcase project milestones and discuss feedback during testing phases.
University of Melbourne Ethics Committee	Responsible for ensuring the project activities meet ethical standards, especially concerning research and educational activities involving minors.	Low	High	If the ethics committee does not approve or requires significant modifications to the project, it may lead to delays or inability to proceed as planned.	Submit detailed project plans and regular updates to ensure all activities comply with ethical requirements. If necessary, conduct face-to-face discussions and

					clarifications with committee representatives.
Experts in AI and Education	Provide technical support and consultation for the development of the AI chatbot, ensuring the effectiveness of the algorithms and the quality of question generation.	Medium	Medium	Their advice is crucial for the project's technical direction and the effectiveness of the AI features. Ignoring technical advice could lead to poor technical implementation.	Monthly technical advisory meetings to discuss technical challenges and progress. Ad-hoc communication and sharing of technical documentation as needed.
Data Privacy Experts	Ensure the project complies with data protection laws and handles intellectual property issues.	Low	High	Failure to comply with data privacy laws or to address intellectual property issues may lead to legal risks or project suspension.	Regular compliance checks and legal consultation meetings. Provide detailed legal review reports as necessary.
Legal Advisors	Ensure the project complies with data protection laws and handles intellectual property issues.	Low	High	Failure to comply with data privacy laws or to address intellectual property issues may lead to legal risks or project suspension.	Regular compliance checks and legal consultation meetings. Provide detailed legal review reports as necessary.

Technical Consideration

Client Requirement : Embrace any technology; strive for maximum inclusivity.

Component	Technology	Description	Reason
Front-End	React Framework	An open-source JavaScript library for building user interfaces, aimed at improving development efficiency and enabling fast UI rendering.	Team members have experience with React, facilitating quick project startup and minimizing time needed for adaptation.
Back-End	Node.js	Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser.	Node.js covers all project needs comprehensively, with team members having rich experience using it.
Database	MongoDB	An open-source NoSQL database that stores data in documents (JSON/BSON format). Offers flexibility and scalability compared to traditional relational databases.	Free version of MongoDB meets project requirements within the limited budget.
LLM API	ChatGPT	ChatGPT is a state-of-the-art large language model developed by OpenAI, capable of generating human-like text based on the prompts it receives. It excels in natural language understanding and generation, making it ideal for interactive applications.	ChatGPT was chosen because of its ability to generate creative and thought-provoking "what if" questions, aligning perfectly with our project's goal of inspiring critical thinking and imagination. Its adaptability and strong performance in language tasks make it an excellent choice for our speculative AI chatbot, ensuring high-quality interactions with users.
Hosting Platform	AWS	AWS (Amazon Web Services) is a comprehensive cloud computing platform that offers a wide range of services, including scalable computing power, database storage, and content delivery services. Its robust ecosystem supports continuous deployment, hosting, and management of applications, making it highly flexible for developers. AWS supports various technologies, including Node.js for backend development, which ensures seamless integration and efficient management of server-side operations.	We selected AWS as our hosting solution due to its extensive and reliable infrastructure, which allows us to focus more on building our AI chatbot rather than managing backend complexities. Our project, which utilizes React for the frontend, Node.js for the backend, and MongoDB for the database, benefits from AWS's ability to scale dynamically and handle different components of our full stack development. This provides a cohesive environment that supports our development needs while ensuring high availability and performance.

Development Environment

Our development environment is structured to support a robust, scalable, and collaborative workflow that integrates both frontend and backend technologies, ensuring seamless deployment and efficient management of the project. This development environment is designed to ensure efficiency, reliability, and scalability, supporting the project's goals while facilitating collaboration among the development team.

1. Programming Languages:

JavaScript:

- Primarily used for frontend development with the React framework.
- Utilized in the backend with the Django framework for handling server-side logic.

2. Frameworks and Libraries:

React: A JavaScript library used for building user interfaces, particularly single-page applications.

Node.js: A JavaScript runtime built on Chrome's V8 engine, enabling developers to execute JavaScript server-side. It is ideal for building fast, scalable network applications because it uses non-blocking, event-driven architecture, allowing for efficient handling of simultaneous connections without heavy use of system resources.

MongoDB: A NoSQL database used for storing and managing the application's data.

3. Integrated Development Environment (IDE):

Visual Studio Code: Our primary code editor, offering a versatile environment for coding in JavaScript, Python, and other languages with integrated debugging, Git control, and extensions.

4. Version Control System:

GitHub: Used for source code management, enabling version control, collaboration, and continuous integration/continuous deployment (CI/CD) pipelines.

5. APIs and Microservices:

ChatGPT API: Integrated for advanced conversational AI capabilities, allowing the chatbot to generate "what if" questions and interact intelligently with users.

6. Hosting and Deployment:

AWS: Used to host the frontend of the application, leveraging services like Amazon S3 and CloudFront to ensure fast, secure, and scalable content delivery. It supports continuous deployment from GitHub, allowing for seamless updates and management of the application directly from source control. This setup is particularly effective for high-availability applications requiring global reach and reliability.

7. Collaboration Tools:

Confluence: Used for documentation, project planning, and knowledge sharing among team members.

Trello: For task management, tracking project progress, and ensuring that all team members are aligned with project goals.

Slack: For daily communication and instant messaging.

WeChat: Used primarily for quick communication among team members.

Microsoft Teams: Primarily used for file sharing with clients, ensuring that all necessary documents and resources are accessible and up-to-date.

Zoom: Utilized for remote meetings and team discussions.

8. Operating Systems:

Our application is developed and tested to be fully compatible with both **Windows** and **macOS** operating systems, ensuring accessibility across different user environments.

9. Database Management:

MongoDB: Used as the primary database for storing and retrieving application data. It is chosen for its scalability and flexibility in handling large datasets.

10. Security and Monitoring:

Regular security audits and monitoring tools are in place to ensure the application remains secure, with data integrity maintained across all transactions.

Ethical Consideration

User Privacy and Data Protection:

- **Description:** Our AI chatbot collects and stores user input prompts and the questions it generates, making data privacy and protection paramount, especially for underage users. We must comply with data privacy regulations across different countries, implementing robust measures to secure user data and safeguard their privacy.
- **Reason:** Protecting minors' data is of utmost importance. By enforcing stringent data protection protocols, we can secure user trust and adhere to international and local legal frameworks, thereby mitigating potential ethical issues.

Bias and Fairness:

- **Description:** ChatGPT may inadvertently introduce biases related to race, gender, culture, or other factors in the content it generates. Given our global and culturally diverse user base, it is crucial to ensure that the generated content is fair, inclusive, and representative of diverse perspectives.
- **Reason:** Minimizing and addressing AI biases is essential for providing a fair and equitable experience for users from various backgrounds. Ensuring content inclusivity and respect for diversity is particularly critical when serving a global audience.

Content Appropriateness and Safety:

- **Description:** Considering our user demographic, which includes individuals aged 14-28, and especially minors, it is imperative to ensure that the questions and content generated are age-appropriate and free from potentially harmful or distressing elements. Strict content filtering and moderation mechanisms should be implemented to prevent the dissemination of inappropriate information.
- **Reason:** Safeguarding underage users from exposure to harmful content is both an ethical and legal responsibility. By ensuring content appropriateness, we create a safe and healthy environment that supports positive user interaction and learning.

Transparency and User Control:

- **Description:** Users should be adequately informed about how the AI system operates and have sufficient control over their data, particularly when minors are involved. It is essential to communicate how their data will be used and stored.
- **Reason:** Enhancing the awareness and control of young users over their data is crucial for protecting their interests. This approach not only helps in building trust but also ensures compliance with legal requirements, especially those concerning minors.

Cultural Sensitivity and Diversity:

- **Description:** Serving a global user base with diverse cultural backgrounds, our AI system must be culturally sensitive and avoid generating content that could be offensive or disrespectful to certain groups. We should consider cross-cultural differences and adjust the AI's content generation strategies accordingly.
- **Reason:** Respecting and understanding cultural diversity ensures broader acceptance of our AI system worldwide. It provides users with an experience that honours their cultural background, thereby preventing cultural conflicts and misunderstandings.

Responsibility and Accountability Mechanisms:

- **Description:** To address potential ethical issues, particularly in dealing with a global and diverse user base, we need to establish clear responsibility and accountability mechanisms to promptly respond and take corrective actions when necessary.
- **Reason:** A strong accountability framework is vital when managing a diverse user group across different cultures and age groups. It ensures we can effectively handle and resolve any ethical challenges, maintaining the integrity and safety of our system.

Ethical Use of AI

- **Description:** Ensure that the AI is used in a manner that aligns with the project's goals and does not manipulate users' emotions or opinions.
- **Reason:** The AI should foster open-ended exploration and critical thinking without exploiting users. Aligning the AI's function with ethical standards helps maintain the integrity of the project.

Impact on Mental Health

- **Description:** Monitor and mitigate any potential negative psychological impacts from the speculative questions, and provide resources for mental health support.
- **Reason:** Speculative "what if" questions could potentially cause anxiety or distress. It is important to consider the emotional well-being of users, particularly since the platform may engage with complex social and environmental issues.

Cyber Security Consideration

In this project, security considerations have been carefully planned to ensure robust protection, though they are yet to be fully implemented. These include using TLS encryption for securing data in transit and AES-256 encryption for data at rest. Role-based access controls and secure session ID management via Snowflake are proposed to safeguard access and user sessions. Additionally, session management will incorporate encrypted Snowflake IDs and rate limiting to prevent API abuse. Key vulnerabilities such as XSS, CSRF, and NoSQL injection will be mitigated through security protocols and proper input validation. Regular security audits and penetration testing are planned to maintain system integrity. Network security measures will include AWS security groups, Web Application Firewall (WAF), and BaoTa's firewall settings to restrict access. For data encryption, AWS KMS will be used to manage encryption keys. To prevent automated abuse, Google reCAPTCHA v3 will be introduced for bot protection. These proposed measures aim to ensure a secure and resilient system once fully implemented.

1. Secure Data Transmission and Storage

Data in Transit:

- Ensure that all communications between the client (browser) and the server are secured using **TLS (Transport Layer Security)** to prevent man-in-the-middle attacks. Enforce HTTPS for all traffic by configuring the server accordingly, especially if the front end is hosted on AWS.
- OpenAI API calls should be secured using TLS to protect API requests from being intercepted. Using the latest version of TLS (v1.3) is recommended.

Data at Rest:

- Sensitive data in MongoDB, such as user chat history and session tokens, should be encrypted using **MongoDB's Queryable Encryption**. This enables to store data in an encrypted form while still allowing efficient query operations on the encrypted fields.
- Chat histories stored locally in localStorage are susceptible to XSS attacks; ensure these are minimized by applying Content Security Policy (CSP) headers and secure JavaScript coding practices. Consider encrypting localStorage data using client-side encryption techniques.

Solution:

- Enable **MongoDB Queryable Encryption** to encrypt all sensitive data fields, ensuring that even if the database is breached, the data remains unreadable without proper decryption keys.
- For data in localStorage, use client-side encryption tools such as `crypto-js` to protect any stored chat history, ensuring that it remains secure even if accessed by malicious scripts.
- BaoTa can be configured to apply SSL certificates to web service.

2. Access Controls and Queryable Encryption

- MongoDB's **Queryable Encryption** provides field-level encryption, ensuring that sensitive data can only be accessed and queried by authorized users with the necessary decryption keys. This ensures that even database administrators or attackers with unauthorized access to the database cannot view sensitive data.
- Role-Based Access Control (RBAC) should be implemented to ensure that only authorized personnel can access the backend and MongoDB instances, limiting the exposure of encrypted data.
- For API-level access to the OpenAI API, restrict usage by securely managing API keys and limiting them to specific IP ranges if applicable.

Solution:

- Utilize MongoDB's **RBAC features** and Queryable Encryption to enforce strict access controls. Only users with proper decryption keys will be able to query and view encrypted data, ensuring that sensitive customer data remains secure.

- Use **Snowflake** for secure session ID generation and enforce server-side session management. Sessions can be encrypted using MongoDB Queryable Encryption to enhance protection against session hijacking.

3. Session Management and User Request Limits

- Use Snowflake to generate **unique session IDs**. These session IDs should be securely stored server-side and communicated to clients in a secure HTTP-only cookie to prevent XSS attacks.
- Enforce session expiration to limit the risk of long-lived sessions being hijacked.
- **Rate limiting** on the server side should be implemented to restrict the number of API calls a user can make, preventing abuse of the AI chatbot.
- To store chat history in **localStorage**, consider using encrypted session tokens, and only store non-sensitive history to avoid exposure.

Solution:

- Implement session management with session ID generation by Snowflake, with session IDs stored server-side and linked to user requests.
- Use Google reCAPTCHA to limit spam or bot access to chat system.

4. Prevention of Common Vulnerabilities

Cross-Site Scripting (XSS):

- Chatbot applications that allow users to send messages can be vulnerable to XSS. Use proper input validation and output encoding to prevent malicious scripts from being executed on the client-side.
- Ensure localStorage data isn't susceptible to XSS by applying Content Security Policy headers and input sanitization.

Cross-Site Request Forgery (CSRF):

- Even though this is a session-based system, ensure that all requests from users include a CSRF token to verify their legitimacy.

SQL Injection and NoSQL Injection:

- If querying MongoDB with user-generated input, ensure sanitizing the inputs and avoid allowing direct queries from clients.

Solution:

- Implement libraries like `helmet.js` for setting HTTP security headers (e.g., Content Security Policy, X-Frame-Options).
- Validate and sanitize all input fields on both the client and server sides. Use prepared queries when accessing MongoDB to prevent injection attacks.

5. Regular Security Audits and Penetration Testing

- Regularly conduct automated security audits and penetration testing on the chatbot application, especially after major updates. This includes vulnerability scanning and testing the AWS infrastructure, Node.js server, and MongoDB for weaknesses.
- BaoTa offers firewall and security modules to conduct regular audits and fortify any detected vulnerabilities.

Solution:

- Use security testing tools such as OWASP ZAP or Burp Suite to perform vulnerability scanning.
- Penetration testing should be scheduled regularly with third-party testers or in-house ethical hackers. Log and address any vulnerabilities discovered.

6. Network Configuration and Firewall

- Secure all AWS and BaoTa server instances using IP whitelisting, only allowing trusted IPs to connect to the servers. Configure firewall rules to restrict unwanted traffic.

- Enable a Web Application Firewall (WAF) to protect AWS-hosted application against common web attacks, such as SQL injection and XSS.

Solution:

- Use AWS security groups to restrict traffic to necessary ports (e.g., 443 for HTTPS).
- Implement BaoTa's firewall configuration to block unneeded inbound and outbound traffic.

7. Data Encryption and Key Management

- Implement **AES-256 encryption** for data at rest, including any backups stored on AWS.
- Encryption keys should be managed using AWS Key Management Service (KMS) to control and rotate keys regularly.

Solution:

- Use AWS KMS to encrypt sensitive data stored in MongoDB or BaoTa and ensure all backups are encrypted as well.

8. User Chat History Storage and LocalStorage Security

Since chat history is stored in localStorage and sensitive information should not reside client-side:

- Only store non-sensitive chat data in localStorage. If necessary, encrypt the chat history before storing it in localStorage.
- Implement a Content Security Policy (CSP) to minimize the risk of XSS.

Solution:

- Implement client-side encryption libraries such as `crypto-js` to encrypt data before storing it in localStorage.
- Ensure secure storage of session tokens and chat history in localStorage with client-side encryption and regular cleanup mechanisms.

9. Firewall and Intrusion Detection

- Use intrusion detection and firewall mechanisms to monitor the server for any abnormal activity.
- Configure the AWS WAF to block malicious requests and protect the application against DDoS and other attacks.
- For added security, deploy **Fail2Ban** on BaoTa to prevent brute force attempts.

Solution:

- Enable AWS Shield and WAF for DDoS protection and monitor traffic patterns. BaoTa's firewall should be configured to handle inbound traffic filtering and log abnormal access attempts.

10. Google reCAPTCHA Implementation

To prevent abuse, particularly from bots:

- Implement **Google reCAPTCHA v3** for invisible protection to monitor and limit bot access.
- ReCAPTCHA should be deployed on key request points, especially before sending requests to the OpenAI API, to avoid abuse of the chatbot.

Solution:

- Use Google reCAPTCHA v3 to silently monitor user interactions and prevent spam or automated attacks.



Requirements

- Personas
- Prototype
- User Stories
- Product Backlog
- Non-Functional Requirements

Personas



Image generated by AI

Lily Anderson
High School Student

Age: 16
Gender: Female
Ethnicity: Australian
Location: Melbourne

Art Design
Book Creativity

Major responsibilities

- Helps in planning and organizing activities for the school's art club, including arranging visits to galleries and inviting guest speakers.
- Actively participates in school-level art and design competitions, often winning or placing highly.
- Continuously experiments with different art mediums and techniques to broaden her artistic skills.

Goals and tasks

- Use the chatterbox to help herself and her friends better understand and discuss artworks during gallery visits.
- Stimulate creative thinking by posing and pondering hypothetical questions to explore the possibilities behind art.
- Broaden her artistic perspective by engaging in more in-depth discussions through interactive questions.

Physical, social, and technological environment

- Uses a smartphone and tablet for learning and social interaction.
- Follows artists and designers on platforms like Instagram and Pinterest for inspiration.
- Regularly participates in art-related online and offline events.

Quote

- "Visiting an art exhibition is not just about appreciating the works, but also about exploring the stories behind the creativity. With the chatterbox, I can engage in deeper discussions with my friends, sparking more inspiration."

Personality

Introvert	<div style="width: 40%;"></div>	Extrovert	<div style="width: 60%;"></div>	Thinking	<div style="width: 30%;"></div>	Feeling
Sensing	<div style="width: 10%;"></div>	Intuition	<div style="width: 90%;"></div>	Judging	<div style="width: 5%;"></div>	Perceiving



Image generated by AI

Lucas Martin
Undergraduate Student

Age: 22
Gender: Male
Ethnicity: European
Location: France

Art Design
Inspiration Creativity

Major responsibilities

- Participates in school and community art exhibitions and design workshops
- Develops and creates visual art pieces focusing on social and environmental issues

Goals and tasks

- To spark creative thinking and find new inspiration for art projects
- To explore and reflect on complex social issues and integrate them into his art
- To use the AI tool to generate deep "what if" questions that help break out of conventional thinking

Physical, social, and technological environment

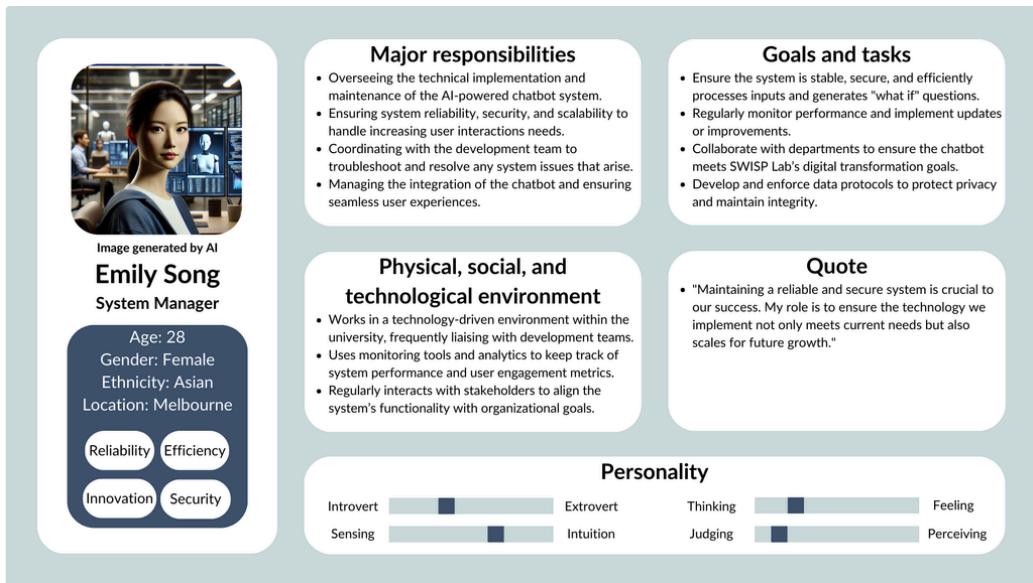
- Frequently works in the campus studio and library
- Actively interacts with peers and the artist community, participating in various art-related events
- Proficient in using design software and enjoys sourcing artistic inspiration through the internet and social media

Quote

- "I want my artwork to ignite conversations about social and environmental issues. I'm always on the lookout for new perspectives and innovative ideas to push the boundaries of what art can communicate."

Personality

Introvert	<div style="width: 20%;"></div>	Extrovert	<div style="width: 80%;"></div>	Thinking	<div style="width: 15%;"></div>	Feeling
Sensing	<div style="width: 5%;"></div>	Intuition	<div style="width: 95%;"></div>	Judging	<div style="width: 10%;"></div>	Perceiving



Role Overview:

The personas selected for this project represent the different user roles and demographics that interact with the AI system. They highlight different perspectives to ensure that the system is accessible, functional, and appealing to a broad user base.

1. Lily Anderson - Local User

Reason for Selection: Lily is representative of the local target audience, particularly young users who engage in creative thinking exercises with the chatbot. Her interactions with the system will test the usability of the system for non-professional, young users who use the system for artistic inspiration and personal projects.

Relevance to the Project: Lily is a high school student, reflecting the system's ability to appeal to a younger, less tech-savvy audience. Her focus on user-friendly interfaces and inspiring creativity was critical in assessing how intuitive the system would be to users who were new to AI-driven tools.

2. Lucas Martin - International User

Reason for Selection: As an international user, Lucas was able to gain insight into the system's functionality on a global scale. His needs focused on accessibility, localization and user experience in different languages and cultures.

Relevance to the Project: Lucas helps ensure that the system is optimized for use by different audiences around the world. His interactions were critical in testing the scalability and localization of the system to ensure that users from different countries could access and benefit from the platform regardless of geographic location.

3. Emily Song - Administrator

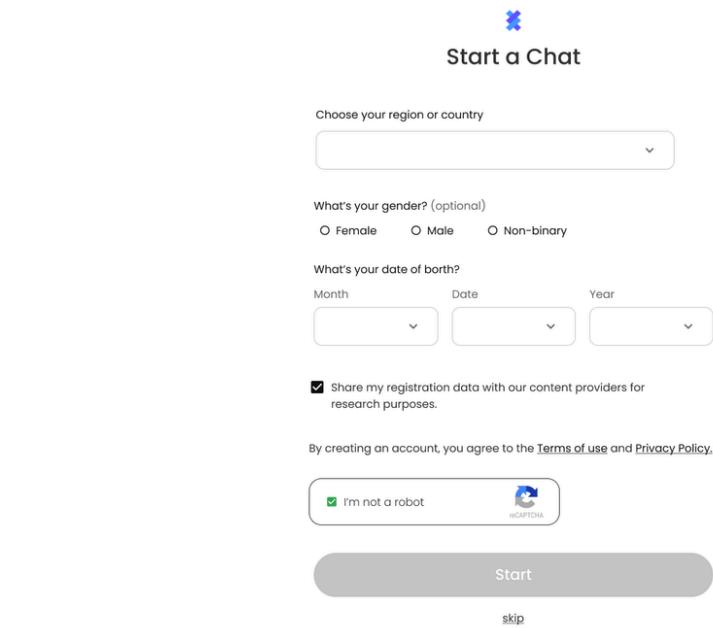
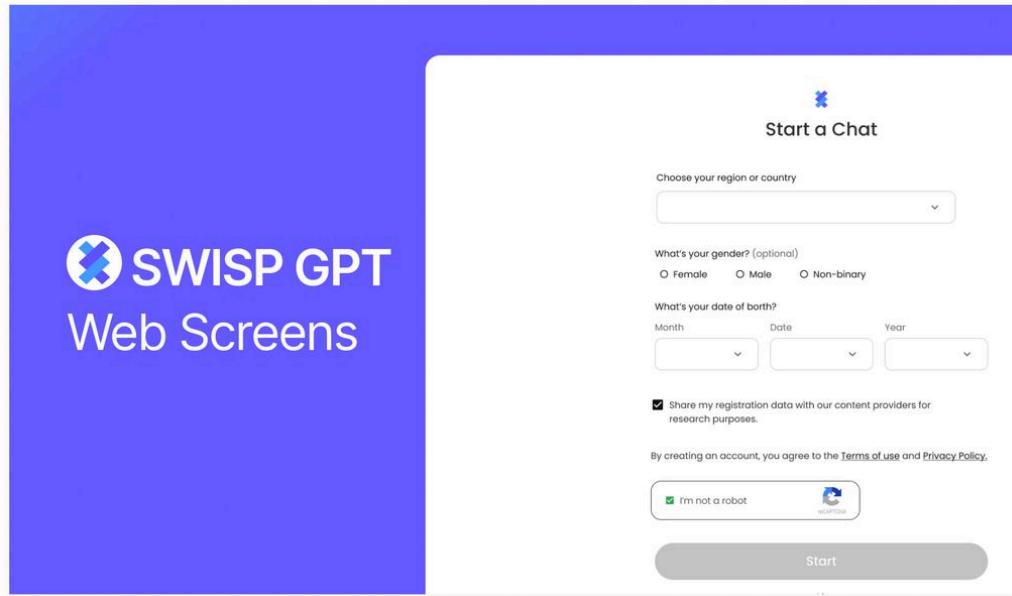
Reason for Selection: Emily was chosen to represent the administrative perspective, focusing on system administration, security, and data analysis. She will ensure that the system runs smoothly while complying with data privacy regulations.

Relevance to the Project: As Administrator, Emily interacts with the system from both a technical and operational perspective. Her input is critical to the system's backend functionality, data management, and overall reliability. Emily ensures the stability and scalability of the system, which is critical to expanding the use of the system.

Prototype

Reference

~ <https://www.figma.com/proto/ik7MYv4wULWS3Dhnu8QjKG/SA-Redback?page-id=0%3A1&node-id=21-212&viewport=583%2C1320%2C0.37&t=gGjkFiaqu2qhT8tz-1&scaling=min-zoom&content-scaling=fixed&starting-point-node-id=21%3A212&share=1>



SWISP GPT

+ New chat

Your conversations **Clear All**

- Benefits of AI in healthcare
- If humans could breathe underwater
- Climate change impacts**
- Last 7 Days
- Cities in 2050
- Sustainable urban planning strategies
- Effects of deforestation
- Future of work in a post-automation

EN

What's in your mind?...

Climate change impacts on agriculture

Here are eight "what if" questions in English focused on the impacts of climate change on agriculture:

1. "What if rising temperatures lead to a permanent drought in major agricultural regions?"
2. "What if climate change causes a global shift in crop growing seasons?"
3. "What if more frequent extreme weather events disrupt global food supply chains?"
4. "What if climate change forces a transition to vertical farming as the primary method of agriculture?"
5. "What if increased CO₂ levels enhance plant growth but decrease nutritional value?"
6. "What if traditional crops become unsustainable, and we have to rely on genetically modified organisms?"
7. "What if climate change makes certain pests and diseases more prevalent, drastically reducing crop yields?"
8. "What if rising sea levels result in the loss of fertile coastal farmland?"

These questions encourage further exploration of the varied impacts of climate change on agriculture.

SWISP GPT

+ New chat

Your conversations **Clear All**

- Benefits of AI in healthcare
- If humans could breathe underwater
- Climate change impacts**
- Last 7 Days
- Cities in 2050
- Sustainable urban planning strategies
- Effects of deforestation
- Future of work in a post-automation

EN

Please give more examples on rising sea levels result in the loss of fertile coastal farmland

Climate change impacts on agriculture

Here are eight "what if" questions in English focused on the impacts of climate change on agriculture:

1. "What if rising temperatures lead to a permanent drought in major agricultural regions?"
2. "What if climate change causes a global shift in crop growing seasons?"
3. "What if more frequent extreme weather events disrupt global food supply chains?"
4. "What if climate change forces a transition to vertical farming as the primary method of agriculture?"
5. "What if increased CO₂ levels enhance plant growth but decrease nutritional value?"
6. "What if traditional crops become unsustainable, and we have to rely on genetically modified organisms?"
7. "What if climate change makes certain pests and diseases more prevalent, drastically reducing crop yields?"
8. "What if rising sea levels result in the loss of fertile coastal farmland?"

These questions encourage further exploration of the varied impacts of climate change on agriculture.

Rising sea levels result in the loss of fertile coastal farmland

Certainly! Here are more examples exploring the impact of rising sea levels on fertile coastal farmland:

1. "What if rising sea levels lead to increased salinity in coastal aquifers, making the soil unsuitable for traditional crops?"
2. "What if frequent flooding from rising sea levels results in the loss of nutrient-rich topsoil along coastal areas?"
3. "What if rising sea levels cause saltwater intrusion into freshwater irrigation systems, compromising crop growth in coastal regions?"
4. "What if coastal erosion driven by rising sea levels destroys essential farmland infrastructure, such as irrigation channels and storage facilities?"
5. "What if the loss of fertile coastal farmland forces agricultural communities to relocate, impacting local economies and food security?"
6. "What if rising sea levels lead to the displacement of coastal communities and the abandonment of valuable farmland?"
7. "What if the inundation of coastal areas due to rising sea levels results in the degradation of important estuarine ecosystems that support agricultural productivity?"
8. "What if rising sea levels require costly adaptations and investments in coastal farmland protection, straining agricultural budgets and resources?"

These scenarios provide various perspectives on how rising sea levels could affect coastal farmland and agriculture.

What's in your mind?...

User Stories

- **Story Points** - Using Fibonacci numbers (1, 2, 3, 5, 8, 13, etc.) for estimates.
- **Priority** - using the MoSCoW including must have, should have, could have, will not have
- **Name of Personas:** Lily (User), Lucas (International User), Emily (Admin)

Epic Feature 1: AI-Driven Question Generation System

ID	As a	I want to	So that	WIT H/WI THO UT GPT	Acceptanc e Criteria	Prio rity	Siz e	Dep end	Justification		Test Resu lt
									Esti mat	enc ion	
E1-US1	Lily	input a topic	I can start the creative thinking process	WIT HOU T Chat GPT	<ul style="list-style-type: none"> System accepts input Feedback confirming input received or invalid input 	Must Have	1	None	Aligns with Client Goal 1 (Web-Based AI Chatbot) and Goal 5 (No User Login Required). Core functionality that receives users' input of AI Chat Box within Inner Scope . Implementation includes a input text box and input processing.	Pass	
E1-US2	Lily	get 8 unique "what if" questions based on initial input	I can stimulate my creative thinking	WIT HOU T Chat GPT	<ul style="list-style-type: none"> 8 unique, diverse "what if" questions generated Clear presentation of questions 	Must Have	5	E1-US1	Directly fulfills Client Goal 1 (Web-Based AI Chatbot) and Goal 7 (Theme-Specific Question Generation). Core functionality that generates responses of AI Chat Box within Inner Scope . Implementation includes generating by interacting with LLM API and presenting 8 unique, diverse questions on chatbox.		
E1-US3	Emily	collect Lily information including location and gender	I can use Lily data for future analysis	WIT HOU T Chat GPT	<ul style="list-style-type: none"> Secure storage of collected data 	Should Have	1	None	Supports Client Goal 9 (Data Archiving for Research) by collecting user data for future analysis. However, priority is lowered to "Should Have" to balance with Goal 5 (No User Login Required). Advanced functionality that collects users information to improve user		

										experience of Data Collection within Inner Scope . Implementation includes secure storage of location and gender data.	
E1-US4	Lily	adjust question generation parameters	the system can meet various learning needs	WIT H Chat GPT	<ul style="list-style-type: none"> User-friendly interface for parameter adjustment< Parameters influence question generation 	Cou Id Hav e	3	E1-US2	<p>This story supports Goal 3 (Critical Thinking Promotion) by allowing fine-tuning of question generation. Priority remains "Could Have" due to budget constraints (Goal 8).</p> <p>Optional advanced Functionality that allows users to modify question generation parameters, enhancing flexibility for personalized learning within the main application.</p> <p>Implementation includes a user interface for setting question generation parameters.</p>		
E1-US5	Lily	get 8 new "what if" questions based on my responses	I can explore deeper aspects of the topic	WIT HOU T Chat GPT	<ul style="list-style-type: none"> New questions relate to previous answers Option to continue or return to original topic 	Show d Hav e	8	E1-US2, E1-US7	<p>Enhances Goal 3 (Critical Thinking Promotion) by allowing users to delve deeper into topics. Supports Goal 1 (Web-Based AI Chatbot) by extending the conversation capabilities.</p> <p>Core functionality that deepens engagement by generating new "what if" questions based on user responses in AI Chat Box within Inner Scope.</p> <p>Implementation includes tracking responses to generate contextually relevant questions.</p>		
E1-US6	Lily	regenerate 8 "what if" questions if unsatisfied	I can get more suitable questions	WIT HOU T Chat GPT	<ul style="list-style-type: none"> Clear "Regenerate" button New set of different questions generated 	Show d Hav e	3	E1-US2, E1-US5	<p>Enhances user experience (related to Goal 6 - User-Friendly Design) by allowing users to get new questions if the initial set doesn't meet their needs.</p> <p>Core functionality that allows users to regenerate questions if unsatisfied in AI Chat Box within Inner Scope.</p> <p>Implementation includes a "Regenerate" button that generates a fresh set of questions.</p>		
E1-US7	Lily	interact with the "what if" questions	I can explore different scenarios	WIT HOU T	<ul style="list-style-type: none"> Ability to select and 	Mus t	3	E1-US2	<p>Directly supports Goal 3 (Critical Thinking Promotion) by encouraging users to engage deeply with the</p>		

			based on my input	Chat GPT	respond to individual questions • The system follow-up questions based on Lily responses	Hav e			generated questions. Also aligns with Goal 1 (Web-Based AI Chatbot) by enabling interactive conversations. Core functionality that enhances user interaction by allowing response-based follow-up questions in AI Chat Box within Inner Scope . Implementation includes selecting questions on interface and generating follow-up based on user input.	
E1-US8	Lily	get a randomly selected "what if" question when I'm stuck	I can continue my creative thinking process	WIT HOUT Chat GPT	• System selects a random question from the generated set • Selected question is clearly highlighted	Show d Hav e	2	E1-US2	Enhances Goal 3 (Critical Thinking Promotion) by providing additional prompts when users are stuck. Supports Goal 6 (User-Friendly Design) by offering assistance during the thinking process. This story aligns with the "AI Chat Box" in the Inner Scope, specifically supporting the functionality to "Automatically generate questions fostering creativity and critical thinking." It enhances the interactive nature of the chatbot, helping users overcome creative blocks.	
E1-US9	Lily	directly receive random "what if" questions	I can start my creative process without a specific topic	WIT HOUT Chat GPT	• System generates random "what if" questions without initial input • Questions are diverse and thought-provoking	Cou ld Hav e	3	None	Supports Goal 3 (Critical Thinking Promotion) by allowing users to engage with random prompts. Enhances user experience (Goal 6) by providing an alternative starting point. This story relates to the "AI Chat Box" in the Inner Scope, extending the capability to "generate 8 hypothetical questions based on user inputs" by allowing generation without specific user input. It provides an additional way for users to interact with the chatbot, increasing its versatility.	

E1-US1-0	Lily	receive suggestions for random topics when I'm unsure what to input	I can get inspiration for starting my creative process	WIT HOU T Chat GPT	<ul style="list-style-type: none"> System provide s a "Sugge st Topics" button Clicking the button generates 3-5 random topic suggestions Topics are diverse and interesting 	Cou ld Hav e	3	Non e	<p>Enhances user experience (Goal 6) by providing starting points for users. Supports Goal 3 (Critical Thinking Promotion) by offering diverse topic suggestions.</p> <p>This story aligns with the "AI Chat Box" in the Inner Scope, particularly the functionality to "Provide an interactive interface with users."</p> <p>It extends the interaction capabilities by offering topic suggestions, which can help "foster creativity and critical thinking" as mentioned in the scope.</p>	
----------	------	---	--	--------------------	---	--------------	---	-------	--	--

Epic Feature 2: Interaction Interface

ID	As a	I want to	So that	WITH/ WITHOUT GPT	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
E2-US1	Lily	Have an Lily friendly interface for inputting topic prompts	I can quickly start using the system	WITHOUT ChatGPT	<ul style="list-style-type: none"> Interface is clean and clear Input process is smooth There is an input prompt box 	Must Have	3	E1-US1	<p>Directly addresses Goal 6 (User-Friendly Design) and supports Goal 5 (No User Login Required) by providing an easy-to-use interface for immediate access.</p> <p>Core functionality that improves user experience by providing a clean, user-friendly interface for entering topic prompts in AI Chat Box within Inner Scope.</p> <p>Implementation includes creating an intuitive input prompt box.</p>

E2-US2	Lily	Clearly view the 8 generated questions	I can easily read and deeply think about them	WITHO UT ChatG PT	<ul style="list-style-type: none"> Questions displayed clearly in numerical form Layout is logical and readable 	Must Have	3	E1-US2	<p>Supports Goal 6 (User-Friendly Design) and Goal 3 (Critical Thinking Promotion) by presenting questions in a clear, easy-to-read format that encourages engagement.</p> <p>Core functionality that enhances user experience by ensuring the clear presentation of the generated questions in AI Chat Box within Inner Scope.</p> <p>Implementation includes a logical layout and readable format for displaying the 8 questionss.</p>
E2-US3	Emily	Monitor system usage	I can ensure the system runs smoothly and user data is secure	WITHO UT ChatG PT	<ul style="list-style-type: none"> Admin panel is accessible only to authorized personnel The system logs and displays user activity 	Should Have	3	None	<p>This story supports Goal 9 (Data Archiving for Research) by ensuring proper system monitoring and data security.</p> <p>Core functionality that maintains system security and performance by providing admin monitoring capabilities in Web Server within Inner Scope.</p> <p>Implementation includes an admin panel for authorized personnel to monitor system usage.</p>
E2-US4	Lily	use the AI chatter box on different devices, including mobile phones	I can access the system anytime and anywhere	WITHO UT ChatG PT	<ul style="list-style-type: none"> Responsive design that adapts to different screen sizes Consistent functionality across devices Optimized performance on 	Should Have	5	E2-US1, E2-US2	<p>This story directly supports Goal 6 (User-Friendly Design) by ensuring the system is accessible on various devices, enhancing user experience and expanding reach.</p> <p>Core functionality that improves accessibility and usability across different platforms within the AI Chat Box in Inner Scope.</p> <p>Implementation includes responsive web design, mobile-friendly UI</p>

					mobile devices				components, and performance optimization for various devices.
E2-US5	Emil y	set usage limits for users	I can prevent excessive or meaningless inputs and manage the budget	WITHO UT ChatG PT	<ul style="list-style-type: none"> Admin can set daily/weekly usage limits System enforces set limits Users are notified when approaching limits 	Should Have	5	None	<p>Addresses budget concerns (Goal 8) by allowing control over system usage. Ensures system sustainability and prevents abuse.</p> <p>This story relates to the "Admin Console" in the Inner Scope.</p> <p>It supports the system can meet diverse research needs and control the budget.</p>
E2-US6	Lily	see a loading indicator while questions are being generated	I know the system is working on my request	WITHO UT ChatG PT	<ul style="list-style-type: none"> Clear loading indicator is displayed during question generation Loading indicator disappears when questions are ready 	Should Have	1	E1-US2	<p>Improves user experience (Goal 6) by providing feedback during processing time. Enhances perceived system responsiveness.</p> <p>This story is connected to both the "AI Chat Box" and "Web Development" areas of the Inner Scope.</p> <p>It improves the user interface by providing visual feedback, which is part of creating a "user-friendly and responsive web interface" as mentioned in the Web Development scope.</p>
E2-US7	Lily	have a friendly and engaging landing page	I can quickly understand the system's purpose and feel motivated to use it	WITHO UT ChatG PT	<ul style="list-style-type: none"> Landing page clearly communicates system purpose within 5 seconds Visually appealing design with engaging graphics 	Should Have	5	E2-US1, E2-US4	<p>Directly supports Goal 6 (User-Friendly Design) by providing an inviting entry point to the system. Also supports Goal 3 (Critical Thinking Promotion) by effectively communicating the system's purpose to users.</p> <p>Core functionality that enhances user experience by providing an engaging introduction to the system</p>

					<ul style="list-style-type: none"> • Clear call-to-action to start using the system • Responsive design for various devices • Page loads within 3 seconds on standard connection 				<p>in Web Development within Inner Scope.</p> <p>Implementation includes designing an appealing layout, creating engaging content, ensuring responsiveness, and optimizing for fast loading.</p>
--	--	--	--	--	---	--	--	--	--

Epic Feature 3: Data Storage and Retrieval

ID	As a	I want to	So that	WITH/WITHOUT GPT	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
E3-US1	Lily	Have the system save my interaction records and answers	I can review and reference them later	WITHOUT ChatGPT	<ul style="list-style-type: none"> • All interactions are saved • Historical records are accessible 	Should Have	3	None	<p>Supports Goal 9 (Data Archiving for Research) by saving user interactions for future analysis. Also enhances user experience, indirectly supporting Goal 6 (User-Friendly Design).</p> <p>Core functionality that ensures long-term value by saving all user interaction records in History within Inner Scope.</p> <p>Implementation includes automatic recording and accessibility of historical interactions, allowing users to reference past sessions for review and further exploration.</p>
E3-US2	Lily	Access historical generated questions and answers	I can gain new inspiration	WITHOUT ChatGPT	<ul style="list-style-type: none"> • Search functionality is provided 	Could Have	5	E3-US1, E2-US2	Enhances user experience (Goal 6) and supports ongoing learning (related to Goal 3). Priority remains

		locally			• Search results are relevant and quick				"Could Have" due to budget constraints (Goal 8). Core functionality that increases the system's reuse value by providing local access to historical questions and answers in History within Inner Scope . Implementation includes a search functionality that allows users to quickly and accurately retrieve past questions and answers, providing continued inspiration.
E3-US3	Emily	Ensure all Lily data is securely stored	I can protect Lily privacy and security	WITHOUT ChatGPT	<ul style="list-style-type: none"> • Data is encrypted in storage • Access control mechanism is in place 	Must Have	3	E3-US1	<p>Critical for maintaining user trust and complying with data protection regulations. Supports Goal 9 (Data Archiving for Research) by ensuring secure data storage.</p> <p>Enhances user experience (Goal 6) and supports ongoing learning (related to Goal 3). Priority remains "Could Have" due to budget constraints (Goal 8).</p> <p>Critical functionality that ensures legal compliance and data security by securely storing user data in Web Server and Data Collection within Inner Scope.</p>
E3-US4	Emily	Have the system store data long-term	I can support in-depth analysis and research	WITHOUT ChatGPT	<ul style="list-style-type: none"> • Data backup mechanism exists • Storage expansion capability is available 	Should Have	3	E3-US3	<p>Directly supports Goal 9 (Data Archiving for Research) by enabling long-term data storage for future analysis.</p> <p>Core functionality that supports future research and analysis by providing long-term data storage capabilities in Web Server within Inner Scope and Data Archiving within Outer Scope.</p> <p>Implementation includes data backup and expansion mechanisms, ensuring the system can store large</p>

									volumes of data securely over time.
E3-US5	Emily	Retrieve historical data based on region, topic, or other metadata	I can conduct multi-dimensional analysis	WITHOUT ChatGPT	<ul style="list-style-type: none"> Advanced retrieval functionality is provided Multiple filter conditions are supported 	Should Have	5	E3-US1, E3-US2, E3-US4	<p>Enhances the capabilities for Goal 9 (Data Archiving for Research) by providing advanced data retrieval options for comprehensive analysis.</p> <p>Advanced functionality that enhances data analysis capabilities by allowing retrieval of historical data using metadata in History within Inner Scope.</p> <p>Implementation includes multi-dimensional filters such as region, topic, and other metadata.</p>

Epic Feature 4:User Access

ID	As a	I want to	So that	WITH/WITHOUT GPT	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
E4-US1	Lily	Use the system as a guest without registering an account	I can use the system directly and quickly	WITHOUT ChatGPT	<ul style="list-style-type: none"> Guest access is provided. 	Must Have	1	None	<p>Directly fulfills Goal 5 (No User Login Required) and supports Goal 6 (User-Friendly Design) by enabling immediate access without registration barriers.</p> <p>Core functionality that simplifies user access by allowing guest usage without requiring account registration in QR Code Accessing within Inner Scope.</p> <p>Implementation includes generating QR codes and providing instant access and usage tracking by scanning QR codes.</p>

Epic Feature 5: Analytics and Reporting

ID	As a	I want to	So that	WITHOUT GPT	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
E5-US1	Emil y	Export Lily data and interaction records	I can analyze the data externally or create reports	WITHOUT ChatGPT	<ul style="list-style-type: none"> Data export functionality is available. Exported data is in a standard format Option to select specific data ranges or types for export. 	Must Have	3	E3-US1, E3-US4	<p>Enhances capabilities for Goal 9 (Data Archiving for Research) by allowing external analysis. Priority remains "Could Have" due to budget constraints (Goal 8).</p> <p>Optional functionality that provides the ability to export Lily's data and interaction records for external analysis or reporting purposes in Data Collection within Inner Scope and User Data Deletion within Outer Scope.</p>

Epic Feature 6: Ethics and Content Filtering

ID	As a	I want to	So that	WITHOUT/WITH GPT	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
E6-US1	Emil y	the system can automatically extract topic from Lily input	ensure content integrity and security	WITHOUT ChatGPT	<ul style="list-style-type: none"> Extract topic from Lily input 	Must Have	2 (was 3)	E1-US1	<p>Supports Goal 7 (Theme-Specific Question Generation) by ensuring accurate topic extraction from user input. Also contributes to system integrity and security.</p> <p>Core functionality that ensures system integrity by automatically extracting the topic from Lily's input in AI Chat Bot within Inner Scope.</p> <p>Implementation includes a reliable mechanism to detect and extract topics from user inputs, ensuring content is appropriately categorized and secured.</p>
E6-US2	Emil y	Have strong language and	I can maintain a healthy	WITHOUT ChatGPT	<ul style="list-style-type: none"> Input validation and filtering 	Must Have	2 (was 5)	E6-US1	This story is crucial for maintaining a safe and appropriate environment, indirectly supporting Goal 3

		content filtering capabilities	platform environment		• Flagging system for suspicious content				(Critical Thinking Promotion) by ensuring quality content. Essential functionality that provides strong language and content filtering to maintain a healthy platform environment in Input Filtering within Inner Scope . Implementation includes validation mechanisms and a flagging system to detect and filter inappropriate content.
E6-US3	Emily	Have system-generated questions and content follow ethical guidelines	I can engage in healthy, beneficial discussions	WITH ChatGPT	• Questions adhere to ethical standards • Content promotes positive engagement	Must Have	1 (was 5)	E6-US1, E6-US2	Supports Goal 3 (Critical Thinking Promotion) by ensuring generated content is ethically sound and promotes constructive discussions. Critical functionality ensuring system-generated questions follow ethical guidelines, fostering healthy and beneficial discussions in Output Filtering within Inner Scope . Implementation includes adherence to ethical standards in question generation.
E6-US4	Lily	Have the system consider local language and cultural differences	I can receive more relevant and relatable questions	WITHOUT ChatGPT	• Multi-language support • Culturally sensitive content generation	Could Have Must Have <i>(The Client cares about if different Language people can use this system)</i>	3 (was 8)	E1-US5	This story supports Goal 6 (User-Friendly Design) by making the system more accessible and relevant to a diverse user base. Optional functionality that enhances user experience by supporting local language and cultural sensitivity in Multi-language within Inner Scope . Implementation includes multi-language support with culturally sensitive content generation, providing users with more relatable and contextually relevant questions.

Note on Story Point Adjustments for Epic 6

Story points for Epic 6 have been reduced due to the utilization of OpenAI's built-in capabilities. OpenAI provides robust content filtering, ethical guidelines, and multi-language support, significantly decreasing the development workload. The remaining points reflect the effort required to integrate and customise these features within our system.

Epic Feature 7: Globalization

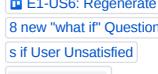
ID	As a [] I want to [] So that []			WITH/WITHOUT GPT	Acceptance Criteria	Priority	Size Estimation	Dependency	Justification
E7-US1	Luca	Access and use the system globally	I can benefit from the system regardless of my location	WITHOUT ChatGPT	<ul style="list-style-type: none"> System accessible from various countries Performance optimized for global use 	Could-Have Must Have <i>(The client cares about if different Language people can use this system)</i>	5	None	<p>This story supports Goal 6 (User-Friendly Design) by ensuring global accessibility.</p> <p>Optional functionality that broadens the system's reach by optimizing for global access in Web Server within Inner Scope and Multi-language within Inner Scope.</p> <p>Implementation includes ensuring the system is accessible worldwide, optimizing performance to accommodate high concurrency and diverse geographical locations, allowing users to access the system smoothly from various countries.</p>

Product Backlog

- **Story Points** - Using Fibonacci numbers (1, 2, 3, 5, 8, 13, etc.) for estimates.
- **Priority** - using the MoSCoW including must have, should have, could have, will not have
- **Name of Personas:** Lily (User), Lucas (International User), Emily (Admin)

Epic Feature 1: AI-Driven Question Generation System

User Story ID	Story/Scenario	Trello Link	Size	Sub Tasks	Priority	Status
E1-US1	As Lily, I want to input a topic so that I can start the creative thinking process	E1-US1: Input the Topic SPRINT 2 BACKLOG	1	1. Design UI for input field 2. Implement input field 3. Implement input field validation 4. Develop feedback system for input validation 5. Test input field functionality	Must Have	Done
E1-US2	As Lily, I want to get 8 unique "what if" questions based on initial input so that I can stimulate my creative thinking	E1-US2 : Generate 8 Unique "what if" Questions SPRINT 2 BACKLOG	5	1. Define criteria for question diversity 2. Design question generation algorithm 3. Integrate algorithm with input handling 4. Implement question display interface 5. Ensure diversity in generated questions 6. Test question generation algorithm	Must Have	Done
E1-US3	As Emily, I want to collect Lily's information including location and gender so that I can use her data for future analysis	E1-US3: Collect User Information SPRINT 2 BACKLOG	1	1. Create data collection form 2. Implement secure data storage 3. Configure data encryption 4. Develop data access controls 5. Ensure data privacy and compliance 6. Conduct privacy impact assessment	Should Have	Done

E1-US4	As Lily, I want to adjust question generation parameters so that the system can meet my various learning needs		3	<ol style="list-style-type: none"> 1. Define user-friendly adjustable parameters 2. Create user interface for parameter adjustment 3. Implement parameter validation 4. Ensure parameters influence question generation 5. Test parameter adjustments and their effects on learning experience <p>Could Have Done</p>	Could Have	Done
E1-US5	As Lily, I want to get 8 new "what if" questions based on my responses so that I can explore deeper aspects of the topic		8	<ol style="list-style-type: none"> 1. Design user interface for response tracking 2. Implement response tracking 3. Generate new questions based on previous answers 4. Implement navigation options 5. Provide options to continue or return to the original topic 6. Test question generation logic 	Should Have	Done
E1-US6	As Lily, I want to regenerate 8 "what if" questions if unsatisfied so that I can get more suitable questions		3	<ol style="list-style-type: none"> 1. Design user interface for the "Regenerate" button 2. Add "Regenerate" button 3. Develop algorithm to generate new set of questions 4. Ensure diversity in new questions 5. Test "Regenerate" functionality 6. Verify diversity of generated questions 	Should Have	Done
E1-US7	As Lily, I want to interact with the "what if" questions so that I can explore different scenarios based on my input		3	<ol style="list-style-type: none"> 1. Design user interface for question selection 2. Implement question selection and response functionality 	Must Have	Done

				3. Develop system to follow up based on responses 4. Test question selection and response functionality 5. Implement and test follow-up mechanism		
E1-US8	As Lily, I want to get a randomly selected "what if" question when I'm stuck so that I can continue my creative thinking process	E1-US8: Random "What If" Question Selection SPRINT 3 BACKLOG	2	1. Implement random selection algorithm 2. Design UI for displaying selected question 3. Integrate with existing question set 4. Test randomness and display functionality	Should Have	To Do
E1-US9	As Lily, I want to directly receive random "what if" questions so that I can start my creative process without a specific topic	E1-US9: Direct Random "What If" Question Generation SPRINT 3 BACKLOG	3	1. Develop algorithm for generating random questions 2. Design UI for requesting random questions 3. Ensure diversity in random questions 4. Test random question generation and display	Could Have	To Do
E1-US10	As Lily, I want to receive suggestions for random topics when I'm unsure what to input so that I can get inspiration for starting my creative process	E1-US10: Random Topic Suggestions SPRINT 3 BACKLOG	3	1. Develop algorithm for generating random topics 2. Design UI for "Suggest Topics" button 3. Implement topic suggestion functionality 4. Ensure diversity in suggested topics 5. Test topic suggestion feature	Could Have	To Do

Epic Feature 2: Interaction Interface

User Story ID	Story/Scenario	Trello Link	Size	Sub Tasks	Priority	Status
E2-US1	As Lily, I want to have a Lily-friendly interface for inputting topic prompts so that I can quickly start using the system	E2-US1: User-Friendly Input Interface SPRINT 2 BACKLOG	3	1. Define user requirements for the interface 2. Design interface layout	Must Have	Done

				3. Implement input prompt box 4. Ensure smooth input process 5. Gather feedback from initial users 6. Test interface usability		
E2-US2	As Lily, I want to clearly view the 8 generated questions so that I can easily read and deeply think about them	E2-US2: Clearly View the 8 Generated Questions SPRINT 2 BACKLOG	3	1. Define requirements for question display 2. Design layout for displaying questions 3. Implement question display in numerical form 4. Ensure readability and logical layout 5. Collect user feedback on display clarity 6. Test clarity of display	Must Have	Done
E2-US3	Admin wants to monitor system usage so that they can ensure the system runs smoothly and user data is secure.	E2-US3: Admin Interface SPRINT 3 BACKLOG	3	1. Define access levels and permissions for admin panel 2. Implement admin panel access control 3. Develop system logging for user activity 4. Implement alerts for unusual activity 5. Ensure system performance tracking 6. Test logging and performance tracking functionalities	Should Have	ToDo
E2-US4	As Lily, I want to use the AI chatter box on different devices, including mobile phones, so that I can access the system anytime and anywhere	E2-US4: Cross-Device Compatibility SPRINT 3 BACKLOG	5	1. Implement responsive design 2. Optimize UI components for mobile 3. Perform cross-device testing 4. Implement performance optimizations for mobile devices 5. Ensure consistent functionality across platforms	Should Have	ToDo
E2-US5	As Emily, I want to set usage limits for users so that I can prevent excessive or	E2-US5: Set User Usage Limits SPRINT 3 BACKLOG	5	1. Design user interface for setting limits	Should Have	ToDo

	meaningless inputs and manage the budget			2. Implement daily/weekly usage tracking 3. Develop limit enforcement mechanism 4. Create user notification system for approaching limits 5. Test limit functionality and notifications		
E2-US6	As Lily, I want to see a loading indicator while questions are being generated so that I know the system is working on my request	E2-US6: Question Generation Loading Indicator SPRINT 3 BACKLOG	1	1. Design loading indicator UI 2. Implement loading state management 3. Integrate loading indicator with the question generation process 4. Test loading indicator functionality	Should Have	ToDo
E2-US7	As Lily, I want a friendly and engaging landing page so that I can quickly understand the system's purpose and feel motivated to use it	E2-US7: Engaging Landing Page Design SPRINT 3 BACKLOG	5	1. Design visually appealing layout for landing page 2. Create engaging content explaining system's purpose 3. Implement interactive elements to increase user engagement 4. Ensure responsive design for various devices 5. Conduct user testing for feedback on landing page 6. Optimize landing page for fast loading	Should Have	ToDo

Epic Feature 3: Data Storage and Retrieval

User Story ID	Story/Scenario	Trello Links	Size	Sub Tasks	Priority	Status
E3-US1	Lily wants to have the system save her interaction records and answers so she can review and reference them later.	E3-US1: Save Interaction Records and Answers SPRINT 2 BACKLOG	3	1. Define data storage schema for interactions 2. Implement interaction recording 3. Ensure data security and privacy	Should Have	Done

				4. Develop historical record access functionality 5. Test interaction recording and access functionality		
E3-US2	Lily wants to access historical generated questions and answers locally so she can gain new inspiration.	E3-US2: Access Historical Interaction Locally SPRINT 3 BACKLOG	5	1. Define search criteria and filters 2. Implement search functionality 3. Implement local data access 4. Develop relevant and quick search results display 5. Test search functionality and results accuracy	Could Have	ToDo
E3-US3	Emily wants to ensure all Lily data is securely stored so she can protect Lily's privacy and security.	E3-US3: Securely Store Data SPRINT 2 BACKLOG	3	1. Define encryption standards and protocols 2. Encrypt data in storage 3. Implement access control mechanisms 4. Implement audit logging for data access 5. Test encryption and access control effectiveness	Must Have	Done
E3-US4	Emily wants the system to store data long-term so she can support in-depth analysis and research.	E3-US4: Long-Term Data Storage SPRINT 2 BACKLOG	3	Define backup frequency and retention policies 1. Implement data backup mechanisms 2. Develop storage expansion capabilities 3. Test backup and restore processes 4. Monitor storage usage and performance	Should Have	Done
E3-US5	Emily wants to retrieve historical data based on region, topic, or other metadata so she can conduct multi-dimensional analysis.	E3-US5: Retrieve Historical Data SPRINT 3 BACKLOG	5	1. Define filter criteria and user requirements 2. Develop advanced retrieval functionality 3. Implement support for multiple filter conditions	Should Have	ToDo

				4. Design and implement user interface for advanced search 5. Test retrieval functionality with various scenarios 6. Optimize retrieval performance		
--	--	--	--	---	--	--

Epic Feature 4: Lily Management

User Story ID	Story/Scenario	Trello Links	Size	Sub Tasks	Priority	Status
E4-US1	Lily wants to use the system as a guest without registering an account so she can use the system directly and quickly.	E4-US1: Guest Access SPRINT 2 BACKLOG	1	1. Define access permissions for guest users 2. Implement guest access functionality 3. Design and implement guest user interface 4. Test guest access functionality	Must Have	Done

Epic Feature 5: Analytics and Reporting

User Story ID	Story/Scenario	Trello Links	Size	Sub Tasks	Priority	Status
E5-US1	Emily wants to export Lily data and interaction records so she can analyze the data externally or create reports.	E5-US1: Export Data and Interaction Record SPRINT 3 BACKLOG	3	1. Define export format specifications 2. Implement data export functionality 3. Ensure data is in a standard format 4. Add range/type selection 5. Implement export options for different data types 6. Test data export functionality	Must Have	ToDo

Epic Feature 6: Ethics and Content Filtering

User Story ID	Story/Scenario	Trello Links	Size	Sub Tasks	Priority	Status
E6-US1	Emily wants the system to automatically extract the topic from Lily's input so that it can ensure content integrity and security.	E6-US1: Automatic Topic Extraction SPRINT 2 BACKLOG	2 (wa s 3)	<ol style="list-style-type: none"> Define topic extraction criteria Implement topic extraction logic Implement error handling for extraction failures Test topic extraction for various inputs Review and refine extraction accuracy 	Must Have	Done
E6-US2	Emily wants to have strong language and content filtering capabilities so that she can maintain a healthy platform environment.	E6-US2: Strong Language Filtering SPRINT 2 BACKLOG	2 (wa s 5)	<ol style="list-style-type: none"> Define content filtering criteria and rules Develop input validation and filtering mechanisms Develop administrative controls for flagged content Implement a flagging system for suspicious content Test filtering and flagging functionality Review and adjust filtering rules based on feedback 	Must Have	Done
E6-US3	Emily wants system-generated questions and content to follow ethical guidelines so that she can engage in healthy, beneficial discussions.	E6-US3: Ethical Content Guidelines SPRINT 2 BACKLOG	1 (wa s 5)	<ol style="list-style-type: none"> Create a review process for ethical compliance Develop guidelines for ethical content Develop training materials for content creators Ensure questions adhere to these guidelines Implement feedback mechanism for ethical issues Test content for compliance 	Must Have	Done
E6-US4	Lily wants the system to consider local language and cultural differences so that she	E6-US4: Consider Local Language and Culture SPRINT 2 BACKLOG	3 (wa s 8)	<ol style="list-style-type: none"> Create a localization strategy for content 	Must Have	Done

can receive more relevant and relatable questions.		2. Develop culturally sensitive content generation 3. Develop a mechanism for continuous improvement based on feedback 4. Gather user feedback on language and cultural aspects 5. Implement multi-language support 6. Test for relevance and relatability		
--	--	--	--	--

Note: The size for the user story in Epic 6 changed ([User Stories](#))

Epic Feature 7: Globalization

User Story ID	Story/Scenario	Trello Links	Size	Sub Tasks	Priority	D
E7-US1	Lucas wants to access and use the system globally so he can benefit from the system regardless of his location.	E7-US1: Global System Access SPRINT 3 BACKLOG	5	1. Ensure system accessibility from various countries 2. Optimize performance for global use 3. Implement multi-region data centers 4. Develop a global content delivery network (CDN) 5. Monitor and address global performance issues	MustHave	ToDo

Non-Functional Requirements

Fairness & Transparency

- **Bias Detection and Diversity:** Implement mechanisms to minimize bias and ensure diverse content suitable for various user backgrounds.
- **Transparency:** Explain the AI decision-making process clearly and the logic behind generated questions, e.g., "This question is based on the user's input and relevant data."
- **Device Compatibility and Tool Support:** Support multiple devices and operating systems, and allow developers to use their preferred IDEs and version control tools.

Informed Consent

- **Transparency:** Provide a clear statement on data collection, usage, storage, and sharing before use.
- **Voluntariness:** Ensure user consent is voluntary and informed, with the option to accept or decline services.
- **Children's Privacy:** Implement special consent processes for minors, including parental or guardian approval.

Sustainability

- **Carbon Footprint:** Reduce paper use and resource waste through electronic means, lowering the carbon footprint.
- **Resource Optimization:** Align with sustainability goals to minimize environmental impact and promote eco-friendly practices.

Reliability

- **High Availability and Stability:** Ensure the system is operational and stable during peak times, with quick recovery from failures.
- **Error Handling and Recovery:** Offer robust error handling with friendly messages and automatic recovery attempts.

Usability

- **User Interface and Feedback:** Design an intuitive interface with clear feedback mechanisms for a smooth user experience.
- **User Feedback Mechanism:** Enable users to provide feedback on questions and show how AI adapts based on this feedback.

Scalability

- **Response Time and Throughput:** Generate questions within seconds and handle multiple requests efficiently. Ensure scalability to maintain performance with increasing users.
- **Elastic Scaling:** Support automatic resource allocation to manage high loads while maintaining response time and processing capacity.

Security & Data Protection

- **Data Security and Access Control:** Encrypt data in transit and storage, with access restricted to authorized users, especially for sensitive data.
- **Data Audit and Compliance:** Implement data auditing to track access and modifications, adhering to privacy and security regulations.

Maintainability

- **Code Quality and Maintenance:** Maintain clean, understandable code for easy updates and fixes, with real-time monitoring and logging.
- **Modular Design:** To avoid widespread issues, use modular design for independent maintenance, upgrades, and replacements.



Sprint

- › [Sprint 1](#)
- › [Sprint 2](#)
- › [Sprint 3](#)

Sprint 1

Plan for Sprint 1

Sprint Duration: 2 weeks (August 9 - August 22)

Sprint Goal:

- Establish foundational project documentation, including the background description, client goals, and motivation.
- Conduct a detailed analysis of project requirements, including user stories, product backlog, and non-functional requirements.
- Set up the development environment and confirm the technology stack, ensuring all team members are aligned on the tools and technologies to be used.

Sprint Priorities

- Completion of essential documentation, including background description, project overview, client goals, and initial requirement analysis.
- Initial setup of the development environment and confirmation of the technology stack, ensuring all team members are familiar with the tools and processes.

Sprint Review

- **Regular progress checks:** Ensure all documentation and setup tasks are completed by the end of the Sprint.
- **Feedback Collection:** Gather feedback on the initial setup and documentation to guide adjustments in the next sprint.

Sprint 1 Retrospective

Keep Doing

- Team Collaboration
- Clear Objectives
- Tool Efficiency
- Client Communication

Challenges

- Scope Understanding
- Time Management
- Tool Integration

Improvements

- Improving Requirement Clarity
- Simplifying Planning Meetings
- Tool Utilization

Actions

- Clarify Requirements
- Schedule Regular Check-Ins
- Optimize Tool Usage

Sprint 1 Retrospective

1. Keep Doing ➤	What went well?	2. Challenge ❌	What could have gone better?
Team Collaboration <ul style="list-style-type: none">• Team members actively shared ideas.• All meetings were attended by the entire team, with everyone actively participating and contributing.• Documentation was updated promptly whenever changes were needed.	Clear Objectives <ul style="list-style-type: none">• Sprint goals and tasks were clear to everyone.• The team understood the purpose and what needed to be completed by the end of the Sprint.	Tool Efficiency <ul style="list-style-type: none">• Tools like Confluence, Trello, Slack, and GitHub were used effectively, helping the team stay organized and on track during the planning phase.	Scope Understanding <ul style="list-style-type: none">• There were challenges in understanding the scope of work or requirements, requiring further clarification and explanation from the mentor.
Improving Requirement Clarity <ul style="list-style-type: none">• Engage in more detailed discussions with the internal team and mentor, and create clearer documentation.	Simplifying Planning Meetings <ul style="list-style-type: none">• Recommend shorter agendas, limiting discussion time, or breaking planning into smaller, focused meetings to increase efficiency.	Tool Utilization <ul style="list-style-type: none">• Determine which platforms are best suited. Consider and explore better platforms.	Time Management <ul style="list-style-type: none">• Meetings took longer than expected, indicating a need to simplify the meeting process.
Clarify Requirements <ul style="list-style-type: none">• Assign team members to contact the mentor and client to clarify any ambiguous requirements.	Schedule Regular Check-Ins <ul style="list-style-type: none">• Set up a regular schedule for internal team check-ins and client updates to ensure ongoing alignment.	Optimize Tool Usage <ul style="list-style-type: none">• Encourage idea sharing and consider testing alternative tools.	

Sprint 1 Review

Project Overview

During Sprint 1, our team focused on developing a prototype interface, finalizing the core technology stack, and defining the project's scope. The purpose of this review is to reflect on the work completed, gather client feedback, and outline the plan for the next phase of development.

Key Deliverables

1. Development of the Prototype Interface

We successfully developed the project's prototype interface and presented it to the client. This interface demonstrated the initial implementation of the core features and represents a significant milestone in the project's progress.

2. Scope Definition

The team clearly defined the project's Inner Scope and Outer Scope. The Inner Scope includes the essential features that will be the focus of the upcoming development phases, while the Outer Scope includes potential future improvements that are not prioritized in the current development cycle.

3. Finalization of the Technology Stack

After thorough consideration and discussion, the team finalized the technology stack, which includes React, Django, MongoDB, ChatGPT, and Netlify. These technologies will form the backbone of the project, ensuring stability and scalability.

4. Client Feedback

Following the presentation of the project's progress, the client expressed satisfaction with the overall direction of the project, particularly with the focus on ethical content generation and data privacy. While no major revisions were requested, the client emphasized the importance of content filtering and data security in future development.

Sprint 2 Planning

1. Primary Goals

In Sprint 2, we will focus on refining the core functionalities, including input processing, question generation, and data security. Additionally, we plan to implement basic content filtering mechanisms to ensure that generated content adheres to ethical standards.

2. Expected Improvements

We will make enhancements to the user input interface, question generation algorithm, secure data storage, and content filtering mechanisms. These improvements will enhance the system's user experience and expand its applicability across different scenarios.

Future Work Plan

1. Implement content filtering mechanisms in Sprint 2 to ensure that generated content adheres to ethical guidelines.
2. Strengthen data security and privacy by utilizing the university's secure AWS services (such as Media Flux) for data storage.
3. Ensure the system remains accessible and relevant across multiple languages and countries.

4. Based on client requirements, move Multi-language support, Admin Console, and Data Archiving from the Outer Scope to the Inner Scope.

Next Review Meeting

We plan to hold the next review meeting after the completion of Sprint 2, around September 20th, to summarize the outcomes of Sprint 2 and plan the subsequent phases.

Sprint 2

Plan for Sprint 2

Sprint Duration: 4 weeks (August 23 - September 20)

Sprint Goal:

- Implement the core functionalities of the AI-driven question Generation System, including input handling, question generation, and user information collection.
- Complete the initial development of the user interface, such as the user-friendly input interface and the display of generated questions.
- Ensure the security of data storage and the preservation of interaction records.
- **Note:** The interaction-related features will be developed in the next sprint.

Sprint Priorities

- **Must Have:** Completion of core functionalities, including input processing, question generation and display, secure data storage, and topic extraction.
- **Should Have:** Optimization of user experience and ensuring the long-term storage and security of data.

Sprint Review

- Regular progress checks to ensure that all Must Have tasks are completed by the end of the Sprint.
- Collection of issues and feedback during development for review and improvement in the next Sprint.

Sprint 2 Backlog

User Story ID	Story/Scenario	Trello Link	Size	Sub Tasks	Priority	Status
E1-US1	As Lily, I want to input a topic so that I can start the creative thinking process	https://trello.com/c/4JLQjVb P	1	1. UI Component Development 2. Responsive Design 3. Implement State Management 4. Implement Input Validation and Feedback 5. Implement Layout Based on Prototype Design 6. Implement Session Management in User Interface	Must Have	Done

E1-US2	As Lily, I want to get 8 unique "what if" questions based on initial input so that I can stimulate my creative thinking	https://trello.com/c/rQuEh6 6J	5 12	1. API Documentation Study 2. Environment Setup 3. Prompt Engineering Implementation 4. API Call Implementation 5. Command Line Interface (CLI) Integration 6. Create Creativity Level Buttons in User Interface	Must Have	Done
E1-US3	As Emily, I want to collect Lily's information including location and gender so that I can use her data for future analysis	https://trello.com/c/VRVLbU dv	1	1. UI Component Development 2. Responsive Design 3. Implement State Management 4. Implement Layout Based on Prototype Design 5. Backend Integration	Should Have	Done
E1-US6	As Lily, I want to regenerate 8 "what if" questions if unsatisfied so that I can get more suitable questions	https://trello.com/c/uWvliiR a	3	1. Input Detection for "Regenerate" Command 2. Backend Request Handling and Prompt Update 3. New Problem Display	Should Have	Done
E2-US1	As Lily, I want to have a Lily-friendly interface for inputting topic prompts so that I can quickly start using the system	https://trello.com/c/RKNC6p D9	3	1. Design UI/UX Design for Input Interface	Must Have	Done
E2-US2	As Lily, I want to clearly view the 8 generated questions so that I can easily read and deeply think about them	https://trello.com/c/kBDJFT Uf	3	1. Design the Question Display Layout	Must Have	Done
E3-US1	Lily wants to have the system save her interaction records and answers so she can review and reference them later.	https://trello.com/c/M0po3N AX	3	1. Design and Model MongoDB Schema 2. Securely Store OpenAI API Key in .env 3. Configure Database Environment and Database Settings 4. Implementation and Development of Thought Chain	Should Have	done

E3-US3	Emily wants to ensure all Lily data is securely stored so she can protect Lily's privacy and security.	https://trello.com/c/jmeoTv NY	3	1. Design and Model MongoDB Schema 2. Securely Store OpenAI API Key in .env 3. Configure Database Environment and Database Settings 4. Deployment and Compliance Check	Must Have	Done
E3-US4	Emily wants the system to store data long-term so she can support in-depth analysis and research.	https://trello.com/c/6SQcq1 JP	3	1. Design and Model MongoDB Schema 2. Securely Store OpenAI API Key in .env 3. Configure Database Environment and Database Settings 4. Deployment and Performance Optimization	Should Have	Done
E4-US1	Lily wants to use the system as a guest without registering an account so she can use the system directly and quickly.	https://trellio.com/c/RGK3 1S3I	1	1. Define Guest Access Mechanism 2. Implement Guest Access	Must Have	Done
E6-US1	Emily wants the system to automatically extract the topic from Lily's input so that it can ensure content integrity and security.	https://trello.com/c/vcy1Vil p was 3)	2 (was 3)	1. Implement Input Text Parsing and Topic Extraction Logic to ensure accurate identification of topics. 2. Integrate Content Filtering and Topic Validation in the backend to maintain security and track topic history.	Must Have	Done
E6-US2	Emily wants to have strong language and content filtering capabilities so that she can maintain a healthy platform environment.	https://trello.com/c/ZSkVqY 2Q	2 (was 5)	1. Implement Basic Language Filtering Logic and Set Up Content Moderation Rules. 2. Integrate Backend Detection for Profanity and Restricted Terms, ensuring real-time content filtering.	Must Have	Done
E6-US3	Emily wants system-generated questions and content to follow ethical guidelines so that she	https://trello.com/c/1Xwsy4 35	1 (was 5)	1. Define Ethical Guidelines for Question Generation and ensure alignment with	Must Have	Done

	can engage in healthy, beneficial discussions.			OpenAI's built-in content moderation. 2. Test Question Generation to verify it adheres to ethical standards and monitor for any violations.		
E6-US4	Lily wants the system to consider local language and cultural differences so that she can receive more relevant and relatable questions.	, https://trello.com/c/mE2xqXKs	3 (was 8)	1. Implement Language Detection and Localization Logic using OpenAI's API capabilities. 2. Test and Optimize the System's Ability to Generate Culturally Relevant Questions.	Could Have	Done

Sprint 2 Retrospective

Sprint Overview

- Sprint Duration: 4 weeks (August 23 - September 20)
- Planned Story Points: 34 (after adjusting Epic 6 story points)
- Actual Completed Story Points: 48 (including additional Sprint 3 tasks)

Achievements

1. Completed all planned Sprint 2 tasks (15 user stories)
2. Finished 3 tasks originally planned for Sprint 3:
 - E1-US4: User can Adjust question generation parameters (Could Have, Size: 3)
 - E1-US5: Generate 8 new "what if" questions based on user interaction (Should Have, Size: 8)
 - E1-US7: Interact with "what if" questions (Must Have, Size: 3)
3. Optimized Epic 6 (Ethics and Content Filtering) implementation by leveraging OpenAI's built-in capabilities

Keep Doing

1. Maintain high team efficiency and productivity
2. Continue successful integration of OpenAI API for high-quality question generation and content filtering
3. Preserve the team's ability to adapt quickly to new discoveries and optimize work processes
4. Sustain strong collaboration between frontend and backend teams
5. Keep the practice of completing tasks ahead of schedule when possible

Challenges

1. Initial sprint planning underestimated team capacity and OpenAI's capabilities
2. Mid-sprint adjustments to Epic 6 story points were somewhat abrupt
3. Information about OpenAI's full capabilities wasn't available at the sprint's start
4. Completing Sprint 3 tasks early may impact the next sprint's planning
5. Balancing rapid development with maintaining code quality and addressing technical debt

Improvements

1. Enhance sprint planning process to better assess external dependencies (like OpenAI features)
2. Develop a more flexible change management process for mid-sprint adjustments
3. Increase knowledge sharing about external services and APIs among team members
4. Improve workload balancing across sprints to maintain consistent productivity
5. Implement a more rigorous technical debt management strategy

Actions

1. Schedule a team workshop to refine the sprint planning process, focusing on better estimation techniques
2. Create a documented procedure for mid-sprint changes and adjustments
3. Organize a knowledge sharing session about OpenAI capabilities and best practices
4. Conduct a thorough review and adjustment of Sprint 3 planning, considering the pre-completed tasks

5. Implement a weekly technical debt review to ensure long-term code quality
6. Set up a cross-functional communication channel for sharing insights about external services and APIs
7. Develop a more comprehensive testing plan for new and adjusted features

Sprint 2 Review with Client

Project Overview

During Sprint 2, our team focused on refining core functionalities, implementing basic content filtering mechanisms, and enhancing the system's security and user experience. This review aims to reflect on the work completed, gather client feedback, and outline the goals for Sprint 3.

Key Deliverables

1. Refinement of Core Functionalities

- The team focused on improving the input processing and question generation algorithm. These improvements ensure smoother user interactions and better question relevance.

2. Implementation of Content Filtering

- We introduced basic content filtering mechanisms to ensure that all generated content adheres to ethical guidelines. This development aligns with our client's emphasis on ethical content generation.

3. Enhancements in Data Security

- Secure data flow between the front and back-end was further strengthened, leveraging AWS services for secure data storage. This enhancement ensures that all sensitive data is protected following the client's data privacy requirements.

4. User Interface (UI) Improvements

- The user interface was upgraded based on the client's feedback, particularly focusing on making it more engaging and user-friendly for a younger audience. The UI now features game-like interactions, improving user engagement.

5. Client Feedback

- The client expressed satisfaction with the improved UI and question generation but requested further UI enhancements to make the interface more playful and fun. Additionally, the client emphasized the need for a data export feature (CSV format) and data storage for up to 6 months.

Sprint 3 Planning

1. Primary Goals

- In Sprint 3, the team will focus on developing an admin interface for data export in CSV format, further enhancing the front-end UI/UX, refining the prompt generation process, and ensuring cross-device compatibility.

2. Expected Improvements

- Planned improvements include better sidebar navigation, optimization of local data storage, improved app performance, and continued security enhancements.

3. Client-Specific Requests

- Implementing the requested data export feature and limiting data collection to essential information (month and year of birth) to maintain privacy compliance. Additionally, we will add playful language and game-like question selection mechanisms.

Based on the results of the Sprint 2 Review meeting, we need to add the following new user stories:

1. Data Export Functionality

- As Emily, I want to export data in CSV format so that I can analyze user interactions and system performance.

2. More Engaging User Interface

- As Lily, I want a more playful and fun interface so that I can enjoy using the system more.

3. Gamified Question Selection Mechanism

- As Lily, I want game-like question selection mechanisms so that I can have a more engaging experience.

4. Cross-Device Compatibility

- As Lily, I want to use the system seamlessly across different devices so that I can access it anytime, anywhere.

5. Optimize Local Data Storage

- As Emily, I want to optimize local data storage so that the system can perform better and manage data more efficiently.

6. Limit Data Collection Scope

- As Lily, I want the system to only collect essential information (month and year of birth) so that my privacy is better protected.

Next Review Meeting

The next Sprint Review meeting is scheduled for October 9th, where the team will summarize the outcomes of Sprint 3 and plan the subsequent phases of development.

Sprint 3

Plan for Sprint 3

3 Updated Plan for Sprint 3

Sprint Duration: 4 weeks (September 21 - October 18)

Sprint 3 Goal:

- Implement admin interface for data export and analysis
- Enhance UI/UX with more playful and engaging features
- Optimize system performance and cross-device compatibility
- Implement additional security measures and data privacy enhancements

Sprint 3 Priorities:

- Must Have: Completion of critical functionalities essential for system operation and client requirements.
 - E5-US2: Export data and interaction records in CSV format (3 points)
 - E7-US1: Ensure global system access (5 points)
- Should Have: Important features that significantly enhance system functionality and user experience.
 - E1-US9: Random "what if" question selection (2 points)
 - E2-US3: Admin monitoring of system usage (3 points)
 - E2-US4: Cross-device compatibility (5 points)
 - E2-US5: Set usage limits for users (5 points)
 - E2-US6: Loading indicator for question generation (1 point)
 - E2-US7: Friendly and engaging landing page (5 points)
 - E3-US5: Retrieve historical data based on metadata (5 points)
- Could Have: Desirable features that will be implemented if time allows.
 - E1-US10: Direct random "what if" question generation (3 points)
 - E1-US11: Random topic suggestions (3 points)
 - E3-US2: Access to historical questions and answers locally (5 points)

Client Feedback Integration:

Based on the [Sprint 2 review](#), we've incorporated the following client requests:

1. Data Export Functionality
2. More Engaging User Interface
3. Gamified Question Selection Mechanism
4. Cross-Device Compatibility
5. Optimize Local Data Storage
6. Limit Data Collection Scope

Sprint 3 Capacity and Rationale:

For Sprint 3, we have planned 45 story points. This capacity is based on the following considerations:

1. Total story points in the backlog: The product backlog contains 45 story points (2+3+3+3+5+5+1+5+5+3+5).

2. Team velocity: Based on the successful completion of 48 story points in Sprint 2, we believe 45 points is an achievable target for Sprint 3.
3. Complexity distribution: The backlog includes a mix of small (1-2 points), medium (3 points), and large (5 points) tasks, allowing for flexible work management.
4. Priority distribution:
 - Must Have: 8 points (17.8%)
 - Should Have: 29 points (64.4%)
 - Could Have: 8 points (17.8%)

This distribution ensures that critical functionalities are prioritized while allowing for significant improvements. The higher proportion of Should Have items indicates our focus on delivering substantial enhancements and new features.

5. Potential challenges: We've accounted for a no-teaching week during which some team members have travel plans, potentially reducing available work time.
6. Increased efficiency: The team's deeper understanding of the project and improved technical skills following Sprint 2 is expected to boost productivity.

Explanation of Achievability:

The Sprint 3 plan is considered achievable for the following reasons:

1. Proven Capacity: The team demonstrated the ability to handle 48 points in Sprint 2, showing they can manage a similar workload of 45 points in Sprint 3.
2. Balanced Workload: The sprint includes a mix of task sizes (1, 2, 3, and 5 story points), allowing for flexible work management throughout the sprint. This variety helps maintain steady progress and allows for parallel work on different tasks.
3. Priority-Based Planning: 82.2% of points are allocated to Must Have and Should Have items, ensuring important work will likely be completed. The inclusion of Could Have items (17.8% of points) provides flexibility for addressing any unexpected challenges or overruns in higher-priority items.
4. Diverse Task Types: The sprint includes a mix of frontend improvements (e.g., E2-US7: Engaging Landing Page), backend enhancements (e.g., E5-US2: Export Data), and new features (e.g., E1-US9: Random Question Selection). This diversity allows for parallel work streams and efficient resource allocation.
5. Incremental Development: Many tasks build upon work from previous sprints (e.g., E2-US4: Cross-Device Compatibility), which should reduce initial setup time and leverage existing knowledge.
6. Team Dynamics: After working together through Sprint 2, the team has developed better communication and collaboration patterns, which should lead to smoother workflow and problem-solving.
7. Clearly Defined Sub-Tasks: Each user story in the backlog has well-defined sub-tasks, providing clear direction and making it easier to distribute work and track progress.
8. Global Considerations: The inclusion of E7-US1 (Global System Access) demonstrates the team's commitment to scalability and performance, which aligns with long-term project goals.
9. Flexibility: The presence of Could Have items allows for some flexibility if unexpected issues arise with higher priority tasks.

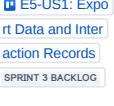
To ensure success, the team will:

- Conduct regular check-ins to monitor progress and make necessary adjustments.
- Prioritize Must Have items early in the sprint.
- Be prepared to re-prioritize tasks if unexpected challenges arise.
- Leverage the detailed sub-tasks to break down work and identify any potential bottlenecks early.

Given these factors, the Sprint 3 plan appears challenging but achievable, building upon the team's previous success while accounting for potential obstacles.

Sprint 3 Backlog:

User Story ID	Story/Scenario	Trello Link	Size	Sub Tasks	Priority	Status
E1-US8	As Lily, I want to get a randomly selected "what if" question when I'm stuck so that I can continue my creative thinking process.	E1-US8: Random "What If" Question Selection SPRINT 3 BACKLOG	2	1. Implement a random selection algorithm 2. Design UI for displaying the selected question 3. Integrate with existing question set 4. Test randomness and display functionality	Should Have	
E1-US9	As Lily, I want to directly receive random "what if" questions so that I can start my creative process without a specific topic.	E1-US9: Direct Random "What If" Question Generation SPRINT 3 BACKLOG	3	1. Develop algorithm for generating random questions 2. Design UI for requesting random questions 3. Ensure diversity in random questions 4. Test random question generation and display	Could Have	
E1-US10	As Lily, I want to receive suggestions for random topics when I'm unsure what to input so that I can get inspiration for starting my creative process.	E1-US10: Random Topic Suggestions SPRINT 3 BACKLOG	3	1. Develop algorithm for generating random topics 2. Design UI for "Suggest Topics" button 3. Implement topic suggestion functionality 4. Ensure diversity in suggested topics 5. Test topic suggestion feature	Could Have	
E2-US3	Admin wants to monitor system usage so that they can ensure the system runs smoothly and user data is secure.	E2-US3: Admin Interface SPRINT 3 BACKLOG	3	1. Define access levels and permissions for admin panel 2. Implement admin panel access control 3. Develop system logging for user activity 4. Implement alerts for unusual activity 5. Ensure system performance tracking 6. Test logging and performance tracking functionalities	Should Have	
E2-US4	As Lily, I want to use the AI chatter box on different devices, including mobile phones, so that I can access the system anytime and anywhere.	E2-US4: Cross-Device Compatibility SPRINT 3 BACKLOG	5	1. Implement responsive design for all UI components 2. Optimize UI components for mobile devices 3. Perform cross-device testing 4. Implement performance optimizations for mobile devices 5. Ensure consistent functionality across platforms	Should Have	
E2-US5	As Emily, I want to set usage limits for users so that I can prevent excessive or meaningless inputs and manage the budget	E2-US5: Set User Usage Limits SPRINT 3 BACKLOG	5	1. Design user interface for setting limits 2. Implement daily/weekly usage tracking 3. Develop limit enforcement mechanism 4. Create user notification system for approaching limits 5. Test limit functionality and notifications	Should Have	

E2-US6	As Lily, I want to see a loading indicator while questions are being generated so that I know the system is working on my request		1	1. Design loading indicator UI 2. Implement loading state management 3. Integrate loading indicator with the question generation process 4. Test loading indicator functionality	Should Have	
E2-US7	As Lily, I want a friendly and engaging landing page so that I can quickly understand the system's purpose and feel motivated to use it.		5	1. Design visually appealing layout for landing page 2. Create engaging content explaining system's purpose 3. Implement interactive elements to increase user engagement 4. Ensure responsive design for various devices 5. Conduct user testing for feedback on landing page 6. Optimize landing page for fast loading	Should Have	
E3-US2	Lily wants to access historical generated questions and answers locally so she can gain new inspiration.		5	1. Define search criteria and filters 2. Implement search functionality 3. Implement local data access 4. Develop relevant and quick search results display 5. Test search functionality and results accuracy	Could Have	
E3-US5	As Emily, I want to retrieve historical data based on region, topic, or other metadata so I can conduct multi-dimensional analysis.		5	1. Define filter criteria and user requirements 2. Develop advanced retrieval functionality 3. Implement support for multiple filter conditions 4. Design and implement user interface for advanced search 5. Test retrieval functionality with various scenarios Optimize retrieval performance	Should Have	
E5-US1	Emily wants to export Lily data and interaction records so she can analyze the data externally or create reports.		3	1. Define export format specifications 2. Implement data export functionality 3. Ensure data is in a standard format 4. Add range/type selection 5. Implement export options for different data types 6. Test data export functionality	Must Have	
E7-US1	Lucas wants to access and use the system globally so he can benefit from the system regardless of his location.		5	1. Ensure system accessibility from various countries 2. Optimize performance for global use 3. Implement multi-region data centers 4. Develop a global content delivery network (CDN)	Must Have	

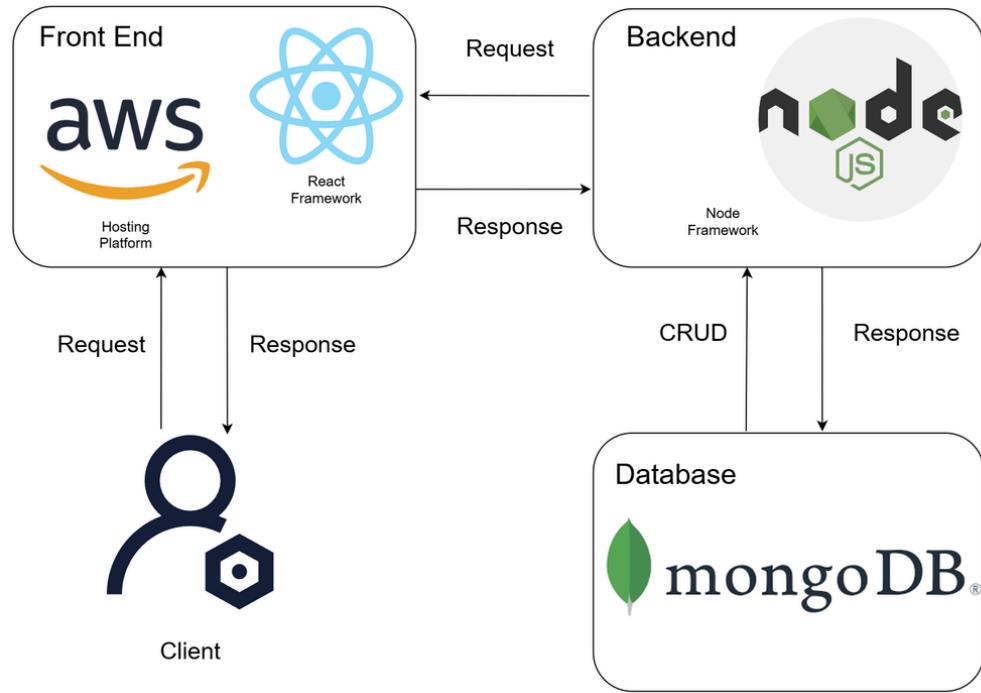
			5. Monitor and address global performance issues	
--	--	--	--	--

Notes:

1. Ensure all new features align with data privacy requirements, especially limiting data collection to essential information (month and year of birth).
2. Continue to leverage OpenAI's capabilities for content filtering and question generation.
3. Maintain focus on code quality and addressing technical debt alongside new feature development.
4. The team should be prepared to adjust priorities if needed, based on the outcomes of ongoing development and any new insights gained during the sprint.
5. Regular check-ins will be conducted to monitor progress and make necessary adjustments.

Architecture

- MongoDB Database Model
- API
- Deployment Decision



MongoDB Database Model

We chose MongoDB for our design due to its flexible document model, which is well-suited for handling complex and dynamic data structures like sessions, questions, prompts, and respond in our system. This model facilitates the easy storage and querying of structured or semi-structured data without predefined schemas, greatly enhancing development flexibility and speed. Moreover, MongoDB's high performance, robust scalability, and comprehensive querying capabilities make it an ideal choice for managing large-scale document storage and real-time data interactions, ensuring efficient processing and analysis of user data while supporting rapid system iterations and expansions.

MongoDB Design Example

1. users Schema

- Content:

- `_id` : Automatically generated unique identifier by MongoDB. This field is used to uniquely identify each document in the users collection, ensuring that each user has a distinct entry.
- `DoB` : Stores the user's birth date. It is used to calculate age, deliver age-appropriate content, or for demographic analytics.
- `gender` : Represents the user's gender. This information can be used for personalization and reporting purposes, allowing for gender-specific analysis or content customization.
- `location` : Captures the geographical location of the user. This can be used for localization, providing region-specific content, or for legal compliance related to geographic data handling.
- `userID` : A unique identifier for the user's session, generated by the server. This ID links the user's session data across the system, helping to track user interaction continuity and session-specific data without exposing the user's identity.

2. openaichats Schema

- Content:

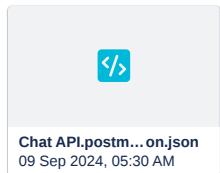
- `_id` : Automatically generated unique identifier by MongoDB for each document in the openaichats collection. This ensures each dialogue instance is recorded distinctly.
- `chatID` : Unique identifier for a dialogue or chat session. This identifier is crucial for referencing and retrieving specific dialogue instances in analytics or for follow-up interactions.
- `timestamp` : Unix timestamp indicating the exact time when the dialogue was recorded. This is crucial for time-series analyses, tracking response times, and organizing chat data chronologically.
- `creativityLevel` : Captures the creativity level set for the AI during that session. This field can influence the AI's response style and complexity, adapting the interaction according to predefined creativity settings.
- `userInput` : Text input by the user. This field is essential for understanding user queries or commands and analyzing interaction patterns.
- `response` : The AI's replies to the user input. This is critical for monitoring AI performance, tweaking response effectiveness, and training the AI on new data.
- `userID` : References the unique identifier from the users schema. This establishes a relationship between the user and their dialogues, allowing for personalized data tracking and user-specific analytics.

API

API Overview

API	Type	Main Features	How It Fits in the Project
OpenAI API (ChatGPT)	Language Model API	Generates human-like text based on prompts - Excels in natural language understanding and generation	Provides creative "what if" questions to enhance user engagement and stimulate critical thinking.
Node Framework	Web API Framework	Provides tools to build web APIs quickly - Handles serialization, authentication, and viewsets	Facilitates communication between the frontend and backend by creating RESTful APIs for data exchange.
MongoDB Atlas API	NoSQL Database API	Flexible document model - High performance and scalability - Supports complex queries	Manages and stores user sessions, prompts, generated questions, and feedback, ensuring efficient data handling.

Postman Workfile



API Naming

We follow RESTful API design principles to provide a consistent and intuitive interface. Each request path is terminated with a /, clearly indicating its functionality and target resource.

Node Framework

Request

- Method:** POST
- URL:** /api/conversations/
- Description:** Creates a new chat record and retrieves the corresponding reply.
- Request Body:**

```
1 {
2   "message": "Your message content"
3 }
```

- **Field Descriptions:**

- `message` (string): The content of the message sent by the user.

Response

- **Status Code:** 200 OK

- **Content:**

```
1 {
2   "reply": "Reply generated based on the user's message"
3 }
```

- **Field Descriptions:**

- `reply` (string): The generated reply to the user's message.

OpenAI API (ChatGPT)

Request

- **Method:** POST
- **URL:** <https://api.openai.com/v1/chat/completions>
- **Description:** Sends a message to the OpenAI GPT model and receives a generated reply.
- **Headers:**
 - Authorization: Bearer YOUR_API_KEY
 - Content-Type: application/json
- **Request Body:**

```
1 {
2   "model": "gpt-4o-mini",
3   "messages": [
4     {
5       "role": "user",
6       "content": "What if we could travel through time?"
7     }
8   ]
9 }
```

Response

- **Status Code:** 200 OK

- **Content:**

```
1 {
2   "id": "chatcmpl-xyz",
3   "object": "chat.completion",
4   "created": 1628171234,
5   "model": "gpt-4o-mini",
6   "usage": {
7     "prompt_tokens": 16,
8     "completion_tokens": 54,
9     "total_tokens": 70
10   },
11   "choices": [
```

```

12  {
13      "message": {
14          "role": "assistant",
15          "content": "Time travel is an exciting concept that opens up many possibilities! Here are some thought-provoking scenarios related to the idea of time travel: \\n\\n1. **What if you could witness any moment in history?** Which event would you choose to see firsthand, and what do you think you would learn from it?\\n\\n2. **What if you could change a pivotal moment in your own life?** Would you make a change, and if so, how do you think that would impact your present?\\n\\n3. **What if time travel was a one-time opportunity?** How would you plan your trip? Would you go to the past, the future, or somewhere else completely?\\n\\n4. **What if you could travel to a future where you met your descendants?** What questions would you want to ask them, and what kind of world do you imagine they'd be living in?\\n\\nFeel free to dive deeper into any of these scenarios or share your own ideas!"
16      },
17      "finish_reason": "stop"
18  }
19 ]
20 }Example

```

MongoDB Atlas API

Request

- Method:** POST
- URL:** /db/user-info
- Description:** This API is used to save user information.
- Request Body:**

```

1  {
2      "location": "Brizal",
3      "gender": "Male",
4      "DoB": "22-11-2000",
5      "sessionId": "002005"
6 }

```

- Field Descriptions:**

Field Name	Type	Required	Description
location	String	Yes	User's location, e.g., Brazil
gender	String	Yes	User's gender, e.g., Male
DoB	String	Yes	User's date of birth, formatted as DD-MM-YYYY
sessionId	String	Yes	Session ID, used to identify the current session

Response

- Status Code:** 201 OK
- Content:**

```

1  {
2      "sessionId": "002005",
3      "message": "User information saved successfully."
4 }

```

Postman Workfile

Request Name	Request Method	Request URL	Request Body	Response Body
Time Travel	POST	https://ai-chatterbox.mb6.top/openai/completion	<pre> "message": "What if we could travel through time?" </pre>	<pre> 1 { 2 "name": "https://ai- chatterbox.mb6.top/openai/completion", 3 "request": { 4 "method": "POST", 5 "header": [], 6 "body": { 7 "mode": "raw", 8 "raw": "{\n\"message\": \"What if we could travel through time?\"\n}", 9 "options": { 10 "raw": { 11 "language": "json" 12 } 13 } 14 }, 15 "url": { 16 "raw": "https://ai- chatterbox.mb6.top/openai/completion", 17 "protocol": "http", 18 "host": [19 "localhost" 20], 21 "port": "3000", 22 "path": [23 "api", 24 "conversations" 25] 26 } 27 }, 28 "response": [29 { 30 "name": "Travel Through time", 31 "originalRequest": { 32 "method": "POST", 33 "header": [], 34 "body": { 35 "mode": "raw", 36 "raw": "{\n\"message\": \"What if we could travel through time?\"\n}", 37 "options": { 38 "raw": { 39 "language": "json" 40 } 41 } 42 }, 43 "url": { </pre>

```

44           "raw":  

45             "https://ai-  

46               chatterbox.mb6.top/openai/completion",  

47                 "protocol":  

48                   "http",  

49                     "host": [  

50                       "localhost"  

51                     ],  

52                     "port": "3000",  

53                     "path": [  

54                       "api",  

55                     ]  

56                   },  

57                     "status": "OK",  

58                     "code": 200,  

59                     "_postman_previewlanguage": "json",  

60                     "header": [  

61                       {  

62                         "key": "X-  

63                           Powered-By",  

64                         "value":  

65                           "Express"  

66                         },  

67                         {  

68                           "key":  

69                             "Content-Type",  

70                               "value":  

71                                 "application/json; charset=utf-8"  

72                               },  

73                               {  

74                                 "key":  

75                                   "Content-Length",  

76                                     "value": "887"  

77                                     },  

78                                     {  

79                                       "key": "ETag",  

80                                         "value":  

81                                           "W/\\"377-ILcp8MTW9Hl2qm3WUMl7eHVF37A\\""  

82                                         },  

83                                         {  

84                                           "key":  

85                                             "Date",  

86                                               "value": "Tue,  

87                                                 03 Sep 2024 07:20:35 GMT"  

88                                               },  

89                                               {  

90                                                 "key":  

91                                                   "Connection",  

92                                                     "value": "keep-  

93                                                       alive"  

94                                                       },  

95                                                       {  

96                                                         "key": "Keep-  

97                                                       Alive",  

98                                                         "value":  

99                                                       "timeout=5"  

100                                                     }

```

					<pre> 88], 89 "cookie": [], 90 "body": "{\n 91 \"reply\": \"Time travel is an exciting\nconcept that opens up many possibilities!\nHere are some thought-provoking scenarios\nrelated to the idea of time travel:\n\\n\\n1. **What if you could witness any\nmoment in history?** Which event would you\nchoose to see firsthand, and what do you\nthink you would learn from it?\\n \\n2.\n**What if you could change a pivotal moment\nin your own life?** Would you make a\nchange, and if so, how do you think that\nwould impact your present?\\n\\n3. **What\nif time travel was a one-time opportunity?\n** How would you plan your trip? Would you\ngo to the past, the future, or somewhere\nelse completely?\\n\\n4. **What if you\ncould travel to a future where you met your\ndescendants?** What questions would you\nwant to ask them, and what kind of world do\nyou imagine they'd be living in?\\n\\nFeel\nfree to dive deeper into any of these\nscenarios or share your own ideas!\"\\n\"}\n </pre>
Climate Change	POST	https://ai-chatterbox.m6t6.top/openai/completion	"message": "Climate Change"	<pre> 1 { 2 "name": "Climate 3 Change", 4 "originalRequest": { 5 "method": "POST", 6 "header": [], 7 "body": { 8 "mode": "raw", 9 "raw": "{\n10 \"message\": \"Climate change\\n\"},\n11 \"options\": {\n12 \"raw\": {\n13 \"language\": \"json\"\n14 } 15 }, 16 \"url\": {\n17 \"raw\": \"https://ai- 18 chatterbox.m6t6.top/openai/completion\", 19 \"protocol\": \"http\", 20 \"host\": [\n21 \"localhost\"], 22 \"port\": \"3000\", 23 \"path\": [\n24 \"api\", 25 \"conversations\"] 26 } </pre>	

```

27     },
28     "status": "OK",
29     "code": 200,
30
31     "_postman_previewlanguage": "json",
32     "header": [
33       {
34         "key": "X-
Powered-By",
35         "value": "Express"
36       },
37       {
38         "key": "Content-Type",
39         "value": "application/json; charset=utf-8"
40       },
41       {
42         "key": "Content-Length",
43         "value": "1445"
44       },
45       {
46         "key": "ETag",
47         "value": "W/"5a5-YIC5jcx0rxBX3wKq8cbtU17JVmk\""
48       },
49       {
50         "key": "Date",
51         "value": "Tue,
03 Sep 2024 07:24:07 GMT"
52       },
53       {
54         "key": "Connection",
55         "value": "keep-
alive"
56       },
57       {
58         "key": "Keep-
Alive",
59         "value": "timeout=5"
60       }
61     ],
62     "cookie": [],
63     "body": "{\n  \"reply\": \"Climate change is a\n  significant and urgent issue impacting our\n  planet. Here are some key points and a few\n  \\"What-If\\\" questions related to\n  climate change:\\n\\n### Key Points:\\n-\n  **Causes:** Primarily driven by human\n  activities, including fossil fuel\n  combustion, deforestation, and industrial\n  processes, leading to increased greenhouse\n  gas emissions.\\n-\n  **Effects:** Rising\n  global temperatures, melting ice caps, sea-
  level rise, extreme weather events, and

```

```

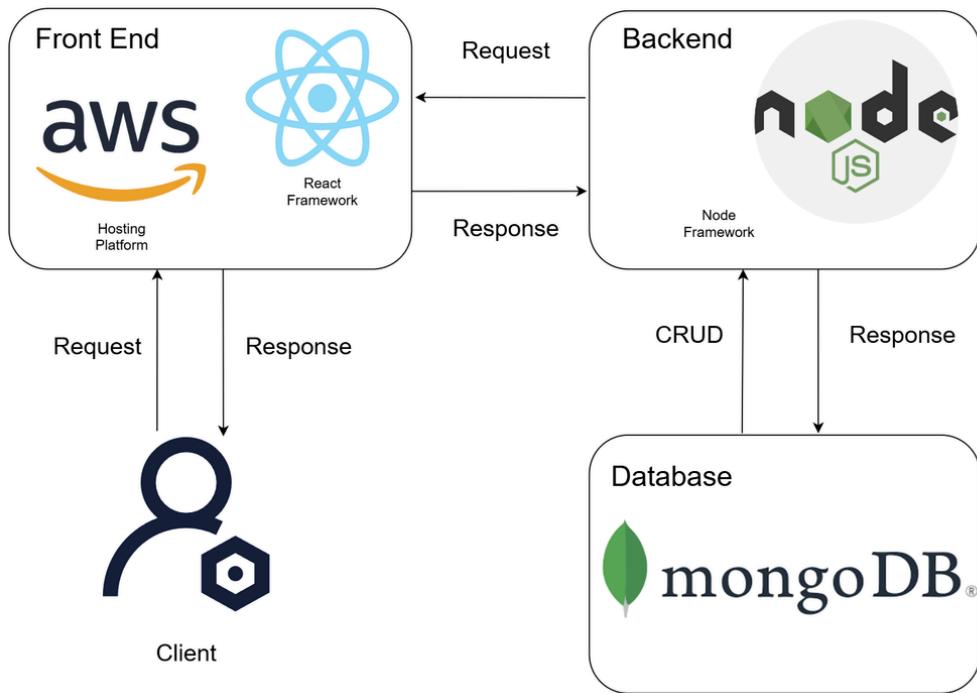
    effects on biodiversity and ecosystems.\n
    **Solutions:** Mitigation strategies
    include renewable energy adoption, energy
    efficiency, reforestation, and sustainable
    agriculture practices.\n\n### What-If
    Questions:\n1. **What if global efforts to
    combat climate change were fully
    successful?** How would the world look in
    50 years if we achieved carbon neutrality?
    \n
    2. **What if you had the power to
    implement one major environmental policy
    worldwide?** What policy would you choose,
    and how do you think it would affect future
    generations?\n\n3. **What if the entire
    world switched to renewable energy
    overnight?** What immediate challenges and
    long-term benefits would we likely
    experience as a society?\n\n4. **What if
    climate change continued unabated for the
    next century?** How would daily life change
    for future generations, and what
    adaptations might they need to make?
    \n\nFeel free to share your thoughts or
    let me know if you'd like more information
    on this topic!"\n"
63
        }
64
    ]
65
}

```

Database	POST	https://ai-chatterbox.mb6.to/db/user-info	<pre> { "location": "Brizal", "gender": "Male", "DoB": "22-11-2000", "sessionId": "002005" } </pre>	<pre> 1 { "name": "https://ai- chatterbox.mb6.top/db/user-info", "request": { "method": "POST", "header": [], "body": { "mode": "raw", "raw": "{\r\n \"location\": \"Brizal\",\r\n \"gender\": \"Male\",\r\n \"DoB\": \"22-11-2000\",\r\n \"sessionId\": \"002005\"\r\n }", "options": { "raw": { "language": "json" } } }, "url": { "raw": "https://ai- chatterbox.mb6.top/db/user-info", "protocol": "https", "host": ["ai-chatterbox", "mb6", "top"], "path": ["db", "user-info"] } } } </pre>
----------	------	---	---	---

```
25           "user-info"
26           ]
27       }
28   },
29   "response": []
30 }
```

Deployment Decision



Evaluation Criteria

- Performance and Reliability
- Scalability and Cost Efficiency
- Security and Compliance
- Development Efficiency and Maintenance
- Integration and Flexibility

Deployment Tool Options

Front End Deployment:

- Advantages: Leveraging AWS's extensive network and services like Amazon CloudFront for distributing React applications globally ensures fast, secure, and scalable deployments.
- Setup: Connect GitHub repository to AWS Amplify for continuous integration and automatic builds, ensuring frontend is always up to date.

Back End Using Netlify Functions:

- Advantages: AWS Lambda allows to run server-side code with high efficiency without managing a full server, ideal for building scalable applications with less overhead.
- Setup: Develop backend logic as a set of serverless functions in AWS Lambda.
- Deployment: Implement and deploy Lambda functions to handle API requests seamlessly alongside front end, using AWS API Gateway to manage interactions.

Database Connection:

- Using MongoDB Atlas as a database solution: Configure AWS Lambda environment variables to securely store and access MongoDB connection details.
- Set up API endpoints in AWS Lambda: Connect these endpoints with MongoDB using the AWS SDK, ensuring secure and efficient data management.
- Security and Integration:
 - Utilize AWS Secrets Manager: Securely manage environment variables and sensitive information (like database URIs).
 - Ensure secure HTTPS connections: Use AWS Certificate Manager to manage SSL/TLS certificates, ensuring secure connections between frontend and backend services.
 - Implement CORS: Configure CORS settings directly in AWS Lambda or API Gateway to handle cross-origin requests efficiently.

Final Decision

Component	Technology	Description	Reason
Front-End	React Framework	An open-source JavaScript library for building user interfaces, aimed at improving development efficiency and enabling fast UI rendering.	Team members have experience with React, facilitating quick project startup and minimizing time needed for adaptation.
Back-End	Node.js	Node.js is an open-source, cross-platform JavaScript runtime environment and library for running web applications outside the client's browser.	Node covers all project needs comprehensively, with team members having rich experience using it.
Database	MongoDB	An open-source NoSQL database that stores data in documents (JSON/BSON format). Offers flexibility and scalability compared to traditional relational databases.	Free version of MongoDB meets project requirements within the limited budget.



Collaboration

- [Team Standards](#)
- [About Us](#)
- › [Agile Ceremonies](#)

Team Standards

Introduction

This document outlines the coding standards and workflows for the development team. The goal is to ensure consistency, maintainability, and efficient collaboration across the project. All team members should follow these standards to ensure high-quality code and streamlined development processes.

GitHub Branch Strategy

- **Main Branch:** Stores stable, production-ready code. All feature branches are merged here after thorough testing and review.
- **Development Branches:** Specific branches for distinct modules ensure focused development and easier management.
 - **SP2-Development:** Manages new feature development and improvements.
 - **SP2-User-Interface:** Focuses on front-end UI developments.
 - **SP2-Data-Storage:** Handles database-related functionalities.
 - **SP2-Backend:** Dedicated to backend operations.

Coding Standards

- **Indentation:** Use 2 spaces for indentation to maintain code readability.
- **Line Length:** Limit line length to 80 characters to ensure code clarity and maintainability.
- **Naming Conventions:**
 - Use camelCase for variable and function names to enhance readability.
 - Use PascalCase for class names to follow industry practices.
- **Comments:** Include meaningful comments to clarify complex code segments.
- **Formatting:** Utilize Prettier for consistent code formatting across the project.

Code Review Process

- **Pull Request Reviews:**
 - Require at least one peer review for every pull request to maintain code quality and catch potential issues early.
 - Ensure all code adheres to the defined coding standards and passes all configured tests before it's eligible for merging.
- **Merge Practices:**
 - Employ 'Squash and Merge' to keep the commit history concise and meaningful.
 - Only merge after all CI/CD tests are passed to maintain the integrity of the main branch.

CI/CD Integration

- Configure continuous integration to automate testing and ensure that every commit meets quality standards.
- Continuous deployment settings should auto-deploy to respective environments post-testing for immediate feedback.

Issue Tracking and Project Management

- Utilize GitHub Issues for comprehensive tracking of bugs, enhancements, and tasks.
- Organize project progress and task allocation using GitHub Projects or Milestones, ensuring clear visibility and management of project timelines.

Documentation Standards

- **Branch-Specific READMEs:** Maintain detailed documentation in each branch's README, covering the purpose, setup, and dependencies specific to that branch.
- **Project Documentation:** Document overarching project setups and deployment instructions in the main branch to guide new team members and external collaborators.
- **Confluence:** Use Confluence for maintaining project-wide documentation, ensuring all team members have access to the latest information and updates.

Communication Protocols

- **Slack:** Utilize Slack for daily communications among team members to discuss project updates, issues, and more.
- **Code Comments:** Ensure that code comments are used to explain complex logic or functions, aiding in code understandability and maintainability.

Regular Documentation Review

- Regularly review and update documentation to ensure accuracy and relevance. This should align with the quarterly review of team standards to ensure documentation evolves with the project needs.

Testing Standards

- Each feature should be covered by unit tests.
- Minimum code coverage should be 80%.
- Integration tests should be implemented for key components.

Regular Review and Updates

- Team standards will be reviewed every quarter.
- Updates will be made based on team feedback and project needs.

About Us

Team Info

NAME	ROLE	CONTACT
@Yujun Yan	Mentor	yujun.yan.1@unimelb.edu.au
@YiYao Li Shaelyn	Product Owner	yiya03@student.unimelb.edu.au
@Leyao Lyu Lydia	Developer	lyull@student.unimelb.edu.au
@Jionghao Song Harry	Developer	jionghaos@student.unimelb.edu.au
@Yiqun Liu Elva	Scrum Master	yiqun2@student.unimelb.edu.au
@Yingying Guo Vicky	Quality Assurance Leader	yinguo3@student.unimelb.edu.au

Others Resources

- Trello  [COMP90082 2024 SM2 SA Redback](#)
- Github  <https://github.com/feit-comp90082/SA-RedBack>

Agile Ceremonies

For each Sprint, Agile ceremonies include **Sprint Planning**, **Sprint Review**, and **Sprint Retrospective**.

Sprint 1 Planning Meeting

Date

Aug 8, 2024 9 pm - 10 pm

Participants

- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)

Goals

- Sprint 1 Goal: Documentation(Background description, client goals, motivation...), Analysis of requirements, Development Environment, Plan
- Compile the tasks for Sprint 1 using the checklist
- Allocate tasks

Discussion topics

Time	Item	Presenter	Notes
10 mins	Sprint 1 planning	@Yiqun Liu	<ul style="list-style-type: none">• Define goals and tasks for Sprint 1
15 mins	Background Discussion	All	<ul style="list-style-type: none">• discuss and create a detailed structure• Background content including: Background description, Project overview, Client goals, Motivation, Scope, Goal model, Personas, Stakeholders
15 mins	Requirement Discussion	All	<ul style="list-style-type: none">• discuss and create a detailed structure• Requirement content including: User Stories, Product backlog, Prototypes, Non-functional Requirements
10 mins	Development Environment Discussion	@Jionghao Song	<ul style="list-style-type: none">• Confirm everyone's skill• Technology choices tentative

Action items

- [@Yiqun Liu](#) : writes the team meeting and client meeting notes on Confluence and drafts the initial background analysis and user stories.

- @Leyao Lyu : writes the mentor meeting notes on Confluence and drafts the initial personas part. @YiYao Li : drafts the initial product overview and client goals.
- @Jionghao Song and @Yingying Guo : draft the initial Motivation, Scope, Goal mode, Stakeholders and Development Environment Part.

⌚ Decisions

None



Sprint 1 Retrospective Meeting

Date

Aug 22, 2024 1 pm - 1:30 pm

Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)

Goals

- Review the team's performance in the Sprint, identify successful practices and challenges encountered, and discuss strategies for improvement.
- Through collective reflection and feedback, help the team to continuously optimize workflow, and improve efficiency and teamwork.

Discussion topics

Time	Item	Presenter	Notes
10 mins	Review Sprint 1 Objectives and Achievements	@Yiqun Liu	
10 mins	Identify What Went Well	All	
10 mins	Discuss Challenges Faced	All	
10 mins	Propose Improvements for Future Sprints	All	

Sprint 2 Planning Meeting

Date

August 22, 2024 9 PM - 10 PM

Participants

- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)

Goals

Sprint 2 Goal: Implement core functionalities of the AI-driven Question Generation System, complete initial user interface development, and ensure data storage security.

Discussion topics

Time	Item	Presenter	Notes
10 mins	Sprint 2 Planning Overview	@Yiqun Liu	Define goals and tasks for Sprint 2
15 mins	Core Functionality Discussion	All	Discuss implementation details for input processing, question generation and display, secure data storage, topic extraction, etc.
15 mins	User Interface Development	@Leyao Lyu	Discuss design for user-friendly input interface and display of generated questions
10 mins	Data Storage and Security	@Yingying Guo	Discuss data storage solutions and security measures
10 mins	Task Allocation	@Yiqun Liu	Assign tasks to team members based on Sprint Backlog
10 mins	Development Environment and Technology Choices Confirmation	@Jionghao Song	Confirm skill matching for all members, finalize technology choices

Action items

- @Yiqun Liu : Responsible for user interface implementation and prompt engineering
- @Leyao Lyu : Responsible for user interface implementation.
- @YiYao Li : Responsible for backend and prompt engineering
- @Yingying Guo : Responsible for data base using mongoDB
- @Jionghao Song** : Responsible for backend and openAI

Decisions

 Confirmed use of MongoDB as the database for storing user information and interaction records.

 Decided to use React as the frontend framework and Express.js as the backend framework.

 Agreed to store the OpenAI API key in a .env file to ensure security.

Sprint 2 Retrospective Meeting

Date

September 17, 2024 9 PM - 10:30 PM

Participants

@Yingying Guo

@Yiqun Liu

@Jionghao Song

@Leyao Lyu

@YiYao Li

Sprint Overview

- Sprint Duration: 4 weeks (August 23 - September 20)
- Planned Story Points: 34 (after adjusting Epic 6 story points)
- Actual Completed Story Points: 48 (including additional Sprint 3 tasks)

Discussion topics

Time	Item	Presenter	Notes
10 mins	Sprint 2 Achievements	@Yiqun Liu	<ol style="list-style-type: none">1. Completed all planned Sprint 2 tasks (15 user stories)2. Finished 3 tasks originally planned for Sprint 33. Optimized Epic 6 implementation
15 mins	Keep Doing	All	Identified 5 key practices to maintain
15 mins	Challenges	All	Discussed 5 main challenges faced during the sprint
15 mins	Improvements	All	Proposed 5 areas for improvement
20 mins	Action Items	@Yiqun Liu	Developed 7 specific actions to address challenges and improvements

15 mins	Open Discussion	All	Team members to share additional thoughts or concerns
---------	-----------------	-----	---

Sprint 3 Planning Meeting

Date

September 19, 2024, 9 PM - 10 PM

Participants

@Yingying Guo

@Yiqun Liu

@Jionghao Song

@Leyao Lyu

@YiYao Li

Goals

Sprint 3 Goal: Implement admin interface for data export and analysis, enhance UI/UX with more playful and engaging features, optimize system performance and cross-device compatibility, and implement additional security measures and data privacy enhancements.

Sprint 3 Capacity

- Total story points planned: 45
- Distribution:
 - Must Have: 8 points (17.8%)
 - Should Have: 26 points (57.8%)
 - Could Have: 11 points (24.4%)

Discussion topics

Time	Item	Presenter	Notes
10 mins	Sprint 3 Planning Overview	@Yiqun Liu	Define goals and tasks for Sprint 3
15 mins	Client Feedback Integration	All	Discuss implementation of client requests from Sprint 2 review
15 mins	UI/UX Enhancement	@Leyao Lyu	Discuss design for more engaging and playful interface
10 mins	Data Export and Analysis	@Yingying Guo	Discuss admin interface for data export and analysis
10 mins	Cross-Device Compatibility	@Jionghao Song	Discuss optimization for various devices

10 mins	Data Privacy Enhancements	@Yingying Guo	Discuss measures to limit data collection and ensure privacy
10 mins	Task Allocation	@Yiqun Liu	Assign tasks to team members based on Sprint Backlog

✓ Action items

- ✓ Go through the sprint 3 checklist together
- ✓ Confirmed the next week's available time for each member
- ✓ split the sprint 3 backlog into 4 weeks to ensure each week has its unique goal

⌚Decisions

- 👉 Confirmed the priority of user stories as outlined in the Sprint 3 backlog
- 👉 Agreed to conduct regular check-ins to monitor progress and make necessary adjustments throughout the sprint
- 👉 Decided to implement random "what if" question selection (E1-US9) as a key feature for enhancing user engagement



Code Review

› [Sprint 2 code Review](#)

Sprint 2 code Review

Sprint 2 Front-end Code Review

Total code review: 8 times (sprint 2)

Frontend Code Review - 1

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Aug 30, 2024

Code Commit: Yes ([Frontend Update](#))

Checklist

- The layout follows the design specifications. [@Yiqun Liu](#)
- The styling is consistent and aligned with the overall design. [@Yiqun Liu](#)
- The color and contrast are in line with the design. [@Yiqun Liu](#)
- The font size is consistent with the design specifications. [@Yiqun Liu](#)

Number of issues: 1

Action Taken: The font size was adjusted to match the design.

Frontend Code Review - 2

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 1, 2024

Code Commit: Yes ([Frontend Update](#))

Checklist

- The layout is responsive and displays correctly on different PC devices. [@Leyao Lyu](#)
- Interactive elements (buttons, links, etc.) function as expected. [@Yiqun Liu](#)
- Style changes on mouseover, focus, and activation states are consistent and provide clear user feedback. [@Yiqun Liu](#)
- Check if the code has any conflicts with the code before [@Yiqun Liu](#)

Number of issues: 1

Action Taken: Layout was adjusted for better responsiveness across PC devices.

Frontend Code Review - 3

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 3, 2024

Code Commit: No

Checklist

- Refined the Start button logic, ensuring it changes to blue when all required input fields are filled. [@Leyao Lyu](#)
- Make sure the layout of the information collection page is worked and fit different PC devices. [@Leyao Lyu](#)
- Check if the code has any conflicts with the code before [@Leyao Lyu](#)

Number of issues: 1

Action Taken: The problem of the turning "Start" chat button from grey to blue, from disabled to enabled is fixed

Frontend Code Review - 4

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 8, 2024

Code Commit: Yes ([Responsive Designs](#))

Checklist

- Refined the design by adjusting the components' sizes for a more cohesive and user-friendly layout. [@Leyao Lyu](#)
- Make sure the layout of the chat page is worked and fit different PC devices. [@Leyao Lyu](#)
- Check if the code has any conflicts with the code before. [@Leyao Lyu](#)

Number of issues: 0 (Standards compliant)

Action Taken: None

Frontend Code Review - 5

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 10, 2024

Code Commit: Yes ([Refine Conversation Design](#))

Checklist

- Finalised the chat side page design, including message bubbles, user and bot avatars, and message alignment. [@Leyao Lyu](#)
- Confirmed that the message reply and flow are visually aligned, improving the overall chat experience. [@Leyao Lyu](#)
- Check if the code has any conflicts with the code before. [@Leyao Lyu](#)

Number of issues: 1

Action Taken: Finalised the chat page design, ensuring other elements do not move when the conversation needs to scroll down, and the elements are in the right places.

Frontend Code Review - 6

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 11, 2024

Code Commit: Yes ([Local Storage](#))

Checklist

- Add local storage to save the conversation on local devices, and can access previous chat history without login [@Jionghao Song](#)
- Check if the code has any conflicts with the code before. [@Jionghao Song](#)

Number of issues: 0 (Standards compliant)

Action Taken: None

Frontend Code Review - 7

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 14, 2024

Code Commit: Yes ([Creativity & New Chat Buttons](#))

Checklist

- Added creativity buttons for better user interaction, allowing selection of different creativity levels. [@Leyao Lyu](#)
- Improved the message sending flow, restricting the user from sending another message before the bot responds. [@Leyao Lyu](#)
- Implemented auto-scroll functionality on the conversation side, so the latest messages are always visible without manual scrolling. [@Leyao Lyu](#)
- Developed the "add new page" button to clear the current conversation when clicked. [@Leyao Lyu](#)
- Check if the code has any conflicts with the code before [@Leyao Lyu](#)

Number of issues: 0 (Standards compliant)

Action Taken: None

Frontend Code Review - 8

Reviewer: [@Yiqun Liu](#) [@Yingying Guo](#) [@Jionghao Song](#) [@YiYao Li](#) [@Leyao Lyu](#)

Review Date: Sep 17, 2024

Code Commit: Yes ([Reformat & Add Comments](#))

Checklist

- Enhanced design details, such as red star indicators and country selection options. [@Leyao Lyu](#)
- Organized and cleaned up the code to improve readability and structure. [@Leyao Lyu](#)
- Check if the code has any conflicts with the code before [@Leyao Lyu](#)

Number of issues: 0 (Standards compliant)

Action Taken: None

Sprint 2 Back-end Code Review

Total code review times: 7 times (sprint 2)

Backend Code Review - 1

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 1, 2024

Code Commit: Yes ([Meet REST API Requirements](#))

Checklist

- Review the code of REST-API written by **@Jionghao Song** @Yingying Guo
- Use Postman to check if the API works properly and return the expected result in JSON format. **@Jionghao Song** @Yingying Guo

Number of issues: 0 (Standards compliant)

Action Taken: None

Backend Code Review - 2

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 3, 2024

Code Commit: Yes ([Beautify the code and add error handling](#))

Checklist

- Review the comment (whether clear/in English/necessary?) **@Jionghao Song** @Yingying Guo
- Review the error handling **@Jionghao Song**
- Review the code to check whether there is redundant code **@Jionghao Song** @Yingying Guo

Number of issues: 0 (Standards compliant)

Action Taken: None

Backend Code Review - 3

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 7, 2024

Code Commit: Yes([Prompt Engineering](#))

Checklist:

- Prompt Clarity and Efficiency: Enhanced clarity by refining the language used in prompts. @Yiqun Liu
- Handling of Diverse Inputs: Improved handling of diverse inputs in the system, ensuring robust response generation for unclear or multiple topic inputs. @YiYao Li @Yiqun Liu
- Ethical Considerations Integration: Integrated checks for ethical considerations, ensuring prompts do not generate harmful or sensitive content. **@Jionghao Song**
- Response Generation Optimization: Optimized the response generation process to ensure each generated "What If" question is unique and thought-provoking. **@Jionghao Song** @Leyao Lyu
- User Interaction Improvements: Streamlined user interactions by adjusting the follow-up question process, enhancing user engagement and response quality. @Yiqun Liu @YiYao Li

Number of issues: 1

Action Taken: We successfully use `SessionId` to ensure the interaction with the user is based on the historical.

Backend Code Review - 4

Reviewer: @Yiqun Liu @Yingying Guo @Jionghao Song @YiYao Li @Leyao Lyu

Review Date: Sep 8, 2024

Code Commit: Yes(Creativity and Diversity Level)

Checklist:

- The appropriate diversity is define @Yiqun Liu @YiYao Li
- The appropriate creativity level (low, medium, high) is define @Yingying Guo @Leyao Lyu @Jionghao Song

Number of issues: 0 (Standards compliant)

Action Taken: None

Backend Code Review - 5

Reviewer: @Yiqun Liu @Yingying Guo @Jionghao Song @YiYao Li @Leyao Lyu

Review Date: Sep 9, 2024

Code Commit: Yes(Separate file management prompts)

Checklist:

- Successfully separated prompt logic into `prompt.js`, enhancing modularity and maintainability of the codebase. @Yiqun Liu @YiYao Li @Jionghao Song
- Ensure that all prompt-related functions are well-documented with clear, concise comments explaining their purpose. @Yingying Guo @Leyao Lyu

Number of issues: 0 (Standards compliant)

Action Taken: None

Backend Code Review - 6

Reviewer: @Yiqun Liu @Yingying Guo @Jionghao Song @YiYao Li @Leyao Lyu

Review Date: Sep 10, 2024

Code Commit: Yes(Improve schema validation, route consistency, and OpenAI API handling)

Checklist

- Improved schema validation by adding trimming, regex validation for `DoB`, and proper timestamp handling in `UserSchema` and `OpenAIChatSchema`. Review the comment (whether clear/in English/necessary?) @Jionghao Song @Yingying Guo
- Enhanced route naming consistency and clarity by suggesting updated paths like `/users` and `/users/chat`. @Leyao Lyu @Yiqun Liu
- Emphasized the need for handling invalid inputs and database errors in route controllers. @Yingying Guo
- Refined OpenAI API response handling to ensure correct extraction of `choices[0].message.content` and added logging for better debugging. @Yingying Guo @YiYao Li

Number of issues: 0 (Standards compliant)

Action Taken: None

Backend Code Review - 7

Reviewer: @Yiqun Liu @Yingying Guo @Jionghao Song @YiYao Li @Leyao Lyu

Review Date: Sep 12, 2024

Code Commit: Yes(Improve schema validation, route consistency, and OpenAI API handling)

Checklist

- Verified that `SessionId` storage is secure and compliant with data protection regulations, using encrypted storage mechanisms.
@Yiqun Liu @Jionghao Song
- `SessionId` is well-integrated with the authentication system, ensuring that session states are maintained accurately across user logins and logouts. @YiYao Li @Yingying Guo
- Ensure that `SessionId` is always transmitted securely. @Leyao Lyu

Number of issues: 0 (Standards compliant)

Action Taken: None

Sprint 2 Integration Code Review

Total code review times: (sprint 2)

Integration Code Review - 1

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 10, 2024

Code Commit: Yes (Integrate prompting and fronted with creativity level)

Checklist

- The implementation integrates the frontend with creativity level controls effectively. @Leyao Lyu
- Creativity level selection changes are captured and passed correctly to the backend. **@Jionghao Song**
- The layout remains responsive and functional across different devices as per the new changes. @Yingying Guo
- Interactive elements such as dropdowns and buttons function as expected, updating the creativity level state appropriately. @YiYao Li
@Yiqun Liu
- Style changes on mouseover, focus, and activation states are consistent and provide clear user feedback. @Leyao Lyu **@Jionghao Song**
- Code merges have been handled correctly with no conflicts with the previous code base. @Yingying Guo

Number of issues: 0

Action Taken: Successfully integrated creativity level controls in the frontend, enhancing user interactivity and customization of the AI responses based on selected creativity levels.

Integration Code Review - 2

Reviewer: @Yiqun Liu @Yingying Guo **@Jionghao Song** @YiYao Li @Leyao Lyu

Review Date: Sep 13, 2024

Code Commit: Yes (update database module and create openai module)

Checklist

- Database and OpenAI modules are updated and integrated correctly. @Yingying Guo
- Functionality for creating user records and OpenAI chat records is implemented without errors. @YiYao Li
- New schemas and route modifications support the functionality as expected. @Yiqun Liu
- User information from the frontend is successfully passed to the backend and stored in the MongoDB User schema. @Yiqun Liu
- Error handling is improved, with clear error messages for missing or incorrect data inputs. **@Jionghao Song**
- The new implementation handles session IDs effectively, ensuring unique identifiers for sessions. @Yingying Guo
- Timestamp conversion and formatting are accurately implemented using moment-timezone. @Yingying Guo
- Code refactoring enhances modularity and readability, adhering to project standards. @Yingying Guo **@Jionghao Song**
- Check for potential conflicts with existing code to ensure smooth integration. @Leyao Lyu

Number of issues: 0

Action Taken: Refactored the database access layer and introduced a dedicated module for handling OpenAI interactions, streamlining backend operations and enhancing data management capabilities. This includes ensuring user data from the frontend is captured and stored accurately in the MongoDB database.

Integration Code Review - 3

Reviewer: @Yiqun Liu @Yingying Guo @Jionghao Song @YiYao Li @Leyao Lyu

Review Date: Sep 14, 2024

Code Commit: fix sendmessagetoserver problem and integrate all script in 1 line

Checklist

- The `sendMessageToServer` function now correctly handles user inputs and server responses. @Leyao Lyu
- User information is accurately captured in the frontend and successfully transmitted to the backend. @Yingying Guo
- Backend properly processes and stores user information in MongoDB's User schema. @Yingying Guo
- Updated routes from GET to POST in `openai.route.js` align with HTTP standards for data submission. @Yiqun Liu
- Refactoring in `db.controller.js` ensures the userID is correctly processed without relying on `req.body`. @Jionghao Song
- All scripts have been consolidated into a single line in `package.json`, simplifying start-up scripts. @YiYao Li
- Error handling and logging are improved to provide clearer feedback during failures. @YiYao Li

Number of issues: 0

Action Taken: Resolved issues in the `sendMessageToServer` function, ensuring robust data handling and storage. Updated routing methods and consolidated start-up scripts enhance the application's operational efficiency and maintainability.

Integration Code Review - 4

Reviewer: @Yiqun Liu @Yingying Guo @Jionghao Song @YiYao Li @Leyao Lyu

Review Date: Sep 18, 2024

Code Commit: No(Deploy backend on Render and frontend on Netlify)

Checklist

- Backend deployment on Render is configured to manage connections and sessions effectively. @Yingying Guo @Jionghao Song
- Frontend deployment on Netlify is set up to serve static files and manage routing correctly. @Yingying Guo @Jionghao Song
- Environment variables and dependencies are appropriately configured in both deployments. @Yingying Guo
- Security configurations, such as CORS and headers, are properly set to ensure safe interactions between the frontend and backend. @Yiqun Liu

Number of issues: 2

1. Heartbeat Mechanism Missing for Render Deployment:

- As Render's free plan automatically puts inactive instances to sleep, the backend requires a regular heartbeat signal to keep the instance active.

2. Image Import Issue in Frontend Deployment on Netlify:

- The frontend application's `index.jsx` file directly references image paths without using module imports, leading to Vite build process failures as it cannot recognize or preprocess these image paths.

Action Taken:

- Scheduled to add a cron job for sending regular heartbeat signals to maintain the backend instance active on Render, and also add a heartbeat route to handle the heartbeat message. (add heartbeat route and beatify the structure)
- Initiated changes in the frontend code to adopt module imports for images, ensuring compatibility with Vite's build process and resolving display issues with user avatars on Netlify. (fix unseen images problem in build)

AI (Front-end) Code Review

Date: 18/09/2024

Participants: @Yiqun Liu @Jionghao Song @Yingying Guo

Code Review: Front-end Index.jsx

Feedback from AI (ChatGPT 4o)

Overview

The component appears to be part of a web application where users interact with a chatbot-like interface. The key functions include:

- Enabling/disabling a "Start" button based on certain form inputs.
- Handling communication with a server for message processing.
- Providing a robot verification checkbox.
- Functionality to skip certain steps and manage session IDs.

Suggestions for Improvement

1. Code Structure & Readability

AI Feedback: Break down the JSX into smaller components, such as extracting the "Start Button" or "Checkbox" into separate functional components.

Team Review:

We agreed with this suggestion as it indeed can improve code maintainability and readability.

Example code (before change):

```
1 return (
2   <div>
3     {/* Large amount of JSX code */}
4     <button
5       className={`button-26 ${isButtonEnabled() ? "button-turn_blue" : ""}`}
6       onClick={handleStartClick}
7       disabled={!isButtonEnabled()}
8     >
9       Start
10      </button>
11      {/* More JSX code */}
12    </div>
13 );
```

After change:

```
1 const StartButton = ({ isEnabled, onClick }) => (
2   <button
3     className={`button-26 ${isEnabled ? "button-turn_blue" : ""}`}
4     onClick={onClick}
5     disabled={!isEnabled}
6   >
7     Start
8   </button>
```

```

9  });
10
11 // In the main component
12 return (
13   <div>
14     {/* Other JSX */}
15     <StartButton
16       isEnabled={isButtonEnabled()}
17       onClick={handleStartClick}
18     />
19     {/* Other JSX */}
20   </div>
21 );

```

Action taken: We implemented this suggestion, extracting some frequently used UI elements into separate components. This not only improved code readability but also enhanced component reusability.

2. Error Handling

AI Feedback: Add more error handling in the `sendMessageToServer` function, especially for network-related errors.

Team Review:

We agreed with this suggestion as robust error handling is crucial for user experience.

Example code (before change):

```

1 const sendMessageToServer = async (message) => {
2   try {
3     const response = await fetch('/api/send-message', {
4       method: 'POST',
5       body: JSON.stringify({ message }),
6     });
7     if (!response.ok) {
8       throw new Error('Server response was not ok');
9     }
10    const data = await response.json();
11    // Process data
12  } catch (error) {
13    console.error('Error:', error);
14  }
15};

```

After change:

```

1 const sendMessageToServer = async (message) => {
2   try {
3     const response = await fetch('/api/send-message', {
4       method: 'POST',
5       body: JSON.stringify({ message }),
6     });
7     if (!response.ok) {
8       throw new Error('Server response was not ok');
9     }
10    const data = await response.json();
11    // Process data
12  } catch (error) {
13    console.error('Error:', error);
14    if (error.name === 'AbortError') {
15      setErrorMessage('Request timed out. Please try again.');
16    } else if (!navigator.onLine) {

```

```

17     setErrorMessage('You are offline. Please check your internet connection.');
18 } else {
19     setErrorMessage('An error occurred. Please try again later.');
20 }
21 }
22 };

```

Action taken: We enhanced the error handling mechanism, adding handling for network timeouts and offline states, and providing user-friendly error messages.

3. Session Management

AI Feedback: Consider logging out the user or resetting the state when there's a session ID mismatch.

Team Review:

This is a valuable suggestion, but we needed to decide how to handle it based on our specific application logic.

Example code (before change):

```

1 if (sessionId !== storedSessionId) {
2   console.error('Session ID mismatch');
3   // No further handling
4 }

```

After change:

```

1 if (sessionId !== storedSessionId) {
2   console.error('Session ID mismatch');
3   // Reset user state
4   resetUserState();
5   // Redirect to login page
6   navigate('/login');
7 }

```

Action taken: We implemented this suggestion, resetting the user state and redirecting to the login page when there's a session ID mismatch, ensuring the security and consistency of user data.

4. State Management Optimization

AI Feedback: Use a single state object or useReducer to manage multiple states more effectively.

Team Review:

This is a good suggestion that can simplify our state management logic.

Example code (before change):

```

1 const [isRobotChecked, setIsRobotChecked] = useState(false);
2 const [selectedCountry, setSelectedCountry] = useState('');
3 const [selectedGender, setSelectedGender] = useState('');
4 // ... more individual states

```

After change:

```

1 const [formState, setFormState] = useState({
2   isRobotChecked: false,
3   selectedCountry: '',
4   selectedGender: '',
5   selectedMonth: '',
6   selectedDate: '',
7   selectedYear: ''
8 });
9

```

```

10 // Function to update state
11 const updateFormState = (field, value) => {
12   setFormState(prevState => ({
13     ...prevState,
14     [field]: value
15   }));
16 };

```

Action taken: We adopted this suggestion, consolidating multiple related states into a single object, which simplified state update logic and improved code maintainability.

5. Repeated Function Calls

AI Feedback: Use useCallback to memoize the isButtonEnabled function to avoid unnecessary re-renders.

Team Review:

This is a valuable performance optimization suggestion.

Example code (before change):

```

1 const isButtonEnabled = () => {
2   return (
3     isRobotChecked &&
4     selectedCountry &&
5     selectedGender &&
6     selectedMonth &&
7     selectedDate &&
8     selectedYear
9   );
10 };

```

After change:

```

1 const isButtonEnabled = useCallback(() => {
2   return (
3     formState.isRobotChecked &&
4     formState.selectedCountry &&
5     formState.selectedGender &&
6     formState.selectedMonth &&
7     formState.selectedDate &&
8     formState.selectedYear
9   );
10 }, [formState]);

```

Action taken: We implemented this suggestion, using useCallback to memoize the isButtonEnabled function, which helps optimize performance, especially in large forms.

6. Styling & Conditional Classes

AI Feedback: Use template literals or the classnames library to simplify handling of conditional class names.

Team Review:

This is a useful suggestion that can improve code readability.

Example code (before change):

```

1 <div className={`button-26 ${isButtonEnabled() ? "button-turn_blue" : ""}`}>

```

After change:

```

1 import classNames from 'classnames';
2

```

```
3 // In the component
4 <div className={classNames('button-26', { 'button-turn_blue': isButtonEnabled() })}>
```

Action taken: We adopted this suggestion, introducing the classnames library to handle conditional class names, making the code clearer and easier to maintain.

7. Async/Await Handling

AI Feedback: Add a loading state while waiting for server response to provide better user experience.

Team Review:

This is a good user experience suggestion.

Example code (before change):

```
1 const sendMessageToServer = async () => {
2   // Logic to send message
3 };
```

After change:

```
1 const [isLoading, setIsLoading] = useState(false);
2
3 const sendMessageToServer = async () => {
4   setIsLoading(true);
5   try {
6     // Logic to send message
7   } finally {
8     setIsLoading(false);
9   }
10 };
11
12 // In the button
13 <button disabled={isLoading}>
14   {isLoading ? 'Sending...' : 'Send'}
15 </button>
```

Action taken: We implemented this suggestion, adding a loading state and visual feedback to improve user experience.

8. Data Validation

AI Feedback: Validate user input before sending a message.

Team Review:

This is an important suggestion for both security and user experience.

Example code (before change):

```
1 const sendMessageToServer = async () => {
2   // Directly send message
3 };
```

After change:

```
1 const validateForm = () => {
2   if (!formState.selectedCountry) {
3     setErrorMessage('Please select a country');
4     return false;
5   }
6   // Validation for other fields
7   return true;
8 };
```

```
9
10 const sendMessageToServer = async () => {
11   if (validateForm()) {
12     // Logic to send message
13   }
14 };
```

Action taken: We implemented this suggestion, adding form validation logic to ensure that only valid data is sent to the server.

Summary:

Our team carefully reviewed all the feedback provided by the AI and implemented most of the suggestions. These changes helped us improve code quality, performance, and user experience. Some suggestions (like session management) needed to be adjusted based on our specific application logic. The suggestions we didn't directly adopt were mainly due to potential conflicts with existing project structure or business requirements, but we still gained valuable insights from them and applied related principles where possible.

AI (Back-end) Code Review

Date: 18/09/2024

Participants:

Code Review: Back-end `server.js`

Feedback from AI (ChatGPT 4)

Overview

The code establishes a server using Express.js, integrates middleware for parsing JSON and URL-encoded data, handles CORS, connects to a database, and sets up routes associated with different functionalities (database, OpenAI, and heartbeat checking).

Suggestions for Improvement

Import Statements Consistency

- **AI Feedback:** Group and order imports for better readability and maintenance.
- **Team Review:** We agreed that this suggestion could improve code organization.

Example code (before change):

```
1 import express from 'express';
2 import cors from 'cors';
3 import bodyParser from 'body-parser';
4 import databaseRoutes from './routes/database';
5 import openaiRoutes from './routes/openai';
6 import heartbeatRoutes from './routes/heartbeat';
```

After change:

```
1 // Third-party modules
2 import express from 'express';
3 import cors from 'cors';
4 import bodyParser from 'body-parser';
5
6 // Route modules
7 import databaseRoutes from './routes/database';
8 import openaiRoutes from './routes/openai';
9 import heartbeatRoutes from './routes/heartbeat';
```

- **Action taken:** We implemented this suggestion, grouping imports by their nature, which improved code readability.

CORS Configuration

- **AI Feedback:** If specific origins are required for security, uncomment and use the origin array.
- **Team Review:** We recognized the importance of this for security but needed to align it with our specific requirements.

Example code (before change):

```
1 app.use(cors());
2 // app.use(cors({
3 //   origin: ['<http://localhost:3000>', '<https://yourdomain.com>']
```

```
4 // }));
```

After change:

```
1 const allowedOrigins = process.env.ALLOWED_ORIGINS.split(',');
2 app.use(cors({
3   origin: function(origin, callback) {
4     if (!origin) return callback(null, true);
5     if (allowedOrigins.indexOf(origin) === -1) {
6       return callback(new Error('The CORS policy for this site does not allow access from the specified
Origin.'), false);
7     }
8     return callback(null, true);
9   }
10 }));

```

- **Action taken:** We implemented a more robust CORS configuration that uses environment variables to specify allowed origins, enhancing security while maintaining flexibility.

Error Handling

- **AI Feedback:** Implement a global error handling middleware.
 - **Team Review:** We agreed this was a crucial improvement for robust error management.

Example code (added):

```
1 // At the end of all routes
2 app.use((err, req, res, next) => {
3   console.error(err.stack);
4   res.status(500).send('Something broke!');
5 });
```

- **Action taken:** We added a global error handling middleware to catch and respond to unhandled errors consistently.

Environment Variables for Port and Database Connection

- **AI Feedback:** Use environment variables for port number and sensitive configurations.
 - **Team Review:** We recognized this as an important security and configuration management practice.

Example code (before change):

```
1 const port = 3000;  
2 mongoose.connect('mongodb://localhost/mydatabase');
```

After change:

```
1 const port = process.env.PORT || 3000;  
2 mongoose.connect(process.env.MONGODB_URI);
```

- **Action taken:** We implemented the use of environment variables for the port and database connection, improving security and configuration flexibility.

Middleware Usage

- **AI Feedback:** Consider adding middleware for compression, security headers, or rate limiting.
 - **Team Review:** We agreed these were valuable additions for performance and security.

Example code (added):

```
1 import compression from 'compression';
2 import helmet from 'helmet';
3 import rateLimit from 'express-rate-limit';
4
5 app.use(compression());
6 app.use(helmet());
7 app.use(rateLimit({
8   windowMs: 15 * 60 * 1000, // 15 minutes
9   max: 100 // limit each IP to 100 requests per windowMs
10 }));
11
```

- **Action taken:** We added compression, security headers, and rate limiting middleware to enhance performance and security.
-

Logging

- **AI Feedback:** Use a more robust logging framework.
- **Team Review:** We agreed this would provide better insights and debugging capabilities.

Example code (added):

```
1 import winston from 'winston';
2 const logger = winston.createLogger({
3   level: 'info',
4   format: winston.format.json(),
5   defaultMeta: { service: 'user-service' },
6   transports: [
7     new winston.transports.File({ filename: 'error.log', level: 'error' }),
8     new winston.transports.File({ filename: 'combined.log' })
9   ]
10 });
11
12 // Instead of console.log
13 logger.info('Server started on port ' + port);
```

- **Action taken:** We implemented Winston for more comprehensive logging capabilities.
-

Route Structure

- **AI Feedback:** Move route definitions to separate files or directories.
 - **Team Review:** We agreed this would improve code organization and scalability.
 - **Action taken:** We restructured our routes into separate files grouped by functionality, improving code organization and maintainability.
-

Code Commenting and Documentation

- **AI Feedback:** Remove unused code and ensure comments are clear. Consider adding API documentation.
 - **Team Review:** We agreed this would improve code clarity and usability.
 - **Action taken:** We removed unused code, improved existing comments, and started work on implementing Swagger for API documentation.
-



Testing Strategy

Introduction

This document outlines the testing strategy for ensuring the quality and reliability of the software through unit and integration testing. The goal is to achieve high code coverage, prevent defects, and ensure the system meets the design requirements.

Testing Scope

The testing strategy covers all major components of the software, including unit tests for individual modules and integration tests for verifying the interactions between modules.

- **Unit Testing:** Focus on testing each individual function and module in isolation.
- **Integration Testing:** Ensure that the system works as expected when different modules interact.

Testing Tools and Technologies

We will use tools and frameworks that integrate with our tech stack and GitHub workflow:

- **Testing Framework:** We will use **Jest** for both unit and integration testing. Jest is well-suited for JavaScript projects and provides a robust environment for running tests.
- **CI Integration:** All tests will be integrated with **GitHub Actions** to run automatically on every push, pull request, or merge.

Test Case Design and Management

Test cases will be designed to cover a wide range of scenarios, including both positive and negative test cases. Each feature will have corresponding test cases to ensure full coverage.

- **Positive Test Cases:** Validate that the system behaves as expected under normal conditions.
- **Negative Test Cases:** Test for edge cases and ensure that the system handles erroneous or unexpected inputs gracefully.
- **Test Case Documentation:** Test cases will be documented and tracked using GitHub Issues for visibility and management.

Code Coverage Guidelines

The following code coverage standards will be applied to ensure the robustness of the code:

- **Minimum Code Coverage:** Each feature should have a minimum of 80% code coverage. This will ensure that most of the code is covered by automated tests.
- **CI Enforcement:** The CI pipeline will enforce the coverage standard. Code will not be merged unless it meets the 80% coverage requirement.

CI/CD Integration

Testing will be tightly integrated with our CI/CD pipeline to ensure continuous quality assurance.

- **Automated Testing:** Every time code is pushed or a pull request is created, the CI/CD pipeline will automatically trigger unit and integration tests.
- **CI/CD Workflow:** Code will only be merged into the main branch after passing all tests and meeting the coverage requirements.

Handling Test Failures

In the event of a test failure, the following process will be followed:

- **Failure Alerts:** Test failures will trigger alerts in the CI system, notifying the team of the issue.
- **Bug Tracking:** Any bugs identified through test failures will be logged in GitHub Issues and assigned to the appropriate developer for resolution.
- **Fix and Retest:** Developers will fix the issue and rerun tests to ensure that the problem is resolved.

Regular Review and Updates

Testing strategies should evolve with the project. Therefore, the following review processes will be in place:

- **Quarterly Reviews:** The team will review the testing strategy every quarter to ensure it remains relevant and effective as the project grows.
- **Strategy Updates:** If needed, adjustments will be made based on project feedback, new features, or technology upgrades.

Test Coverage Reporting

To maintain visibility, coverage reports will be generated and reviewed regularly:

- **Reports:** Coverage reports will be generated automatically by the CI pipeline.
- **Action on Low Coverage:** Any areas with low coverage will be identified, and additional test cases will be created to improve coverage.

Documentation

All test cases, strategies, and reports will be documented and maintained for easy access and future reference.

- **Confluence:** All testing documentation will be kept in Confluence for easy collaboration and access.
- **Test Case Documentation:** A repository of all test cases will be maintained to ensure traceability and completeness.

Conclusion

This testing strategy ensures that every piece of code is tested thoroughly and that the system remains stable as it evolves. By adhering to these guidelines, we aim to maintain high-quality code and reduce the risk of defects.

Product Demonstration

The product is deployed and the URL is: [sa-redback](#).



Meetings

- › [Client Meetings](#)
- › [Mentor Meetings](#)
- › [Team Meetings](#)

Client Meetings

19/09/2024 Client Meeting

Date

Sep 19, 2024 4:00PM - 5:10PM

Participants

- [@YiYao Li](#)
- [@Yiqun Liu](#)
- [@Leyao Lyu](#)
- [@Yingying Guo](#)
- [@Jionghao Song](#)
- The Client

Goals

- Review of Sprint 2 deliverables
- Gather client feedback
- Discuss plans for Sprint 3

Discussion topics

Time	Item	Presenter	Notes
5 mins	Welcome and Self-Introduction	@Jionghao Song	<ul style="list-style-type: none">• Welcomed everyone and introduced the meeting's purpose.• Confirmed the goal to review Sprint 2 work, gather feedback, and discuss plans for the next steps.
10 mins	Sprint 2 Deliverables Overview	@Jionghao Song	<ul style="list-style-type: none">• Presented the updated project goal• Review the technology stack and other development environment.
20 mins	Live Demo of Deployed Application	@Jionghao Song	<ul style="list-style-type: none">• Key features shown included a simple user-friendly input interface, dynamic AI-driven question generation, and secure data flow between the front and back-end.
15 mins	Q&A regarding the demo		<ul style="list-style-type: none">• Discussed and addressed design feedback. The client requested a more fun and engaging UI that is better suited for young users.• The client also specified additional design preferences and features they'd like to see implemented.

5 mins	Feedback Discussion	<ul style="list-style-type: none"> • Client Feedback: The client requested the development of an admin interface for data export, with the output format in CSV, and long-term data storage. • Questions: The client inquired about the timeline for the admin interface and CSV export, along with data storage capabilities for approximately 5000 records over a 3-month period. • Response: The team confirmed that the admin interface with CSV export and data storage for the specified duration will be considered for development in Sprint 3.
5 mins	Sprint 3 Planning Overview	<ul style="list-style-type: none"> • Presented the Sprint 3 goals: Developing an admin interface for data retrieval, optimizing front-end UI/UX, refining prompt generation, enhancing security, and ensuring cross-device compatibility. • Discussed planned improvements: Sidebar navigation, local data storage, prompt refinement based on feedback, and overall app performance and security enhancements.
10 mins	Q&A and Conclusion	<ol style="list-style-type: none"> 1. Q&A: Addressed additional questions from the clients about the UI and game play question generation. 2. Next Steps: Sprint 3 will focus on the planned goals, with a review meeting scheduled for the end of the sprint. 3. Conclusion: Thanked everyone for their feedback and time.

💡 Q&A Session Notes

Question	Response
What format should be used for data export?	The client requested that the data be exportable in CSV format, with a storage period of 3 to 6 months for the records.
How should the subdomain and certificates be managed?	The team discussed requesting a subdomain, and the client want the domain name start with "swisp- fortune-teller" and the team need an SSL certificate for secure access.

Additional Notes from Client

1. **Fun and Engaging Design:** The client emphasized the importance of making the UI more playful and engaging, particularly for younger audiences, with game-like interactions.
2. **Simplified Data Collection:** The client requested minimal data collection to avoid privacy issues, suggesting that only month and year of birth be collected.
3. **Creative Language Use:** The client stressed the need for playful language in the app, such as replacing "Start the Chat" with "Start the Game" or similar phrases that make the interaction more fun.
4. **Game-Like Question Selection:** The client proposed implementing a random question selection process, to make the experience feel more like a game.
5. **Usage Limitation:** The client expressed concern over unlimited use and suggested a restriction of around 6 uses, potentially enforced via login or set some obstacles.

Action items

- Focus on making the design more engaging and game-like, using playful language and interactive elements to appeal to younger users.
- Limit personal data collection to essential details (e.g., month and year of birth) and implement a usage cap (around 6 uses) to ensure responsible app use.
- Develop an admin interface for CSV data export, with data retention limited to 3-6 months.

Decisions

 Proceed with the current plan for Sprint 3, focusing on core functionalities and refine the UI design.

 Schedule the next Sprint Review Meeting after the completion of Sprint 3, around October 9th.

 Add some user stories base on clients' requirements

01/09/2024 Client Meeting

📅 Date

Aug 30, 2024 2:00PM - 3:00PM

👥 Participants

- [@YiYao Li](#)
- [@Yiqun Liu](#)
- [@Leyao Lyu](#)
- [@Yingying Guo](#)
- [@Jionghao Song](#)
- Kate Coleman (Client)

📝 Goals

- Review of Sprint 1 deliverables
- Gather client feedback
- Discuss plans for Sprint 2

🗣 Discussion topics

Time	Item	Presenter	Notes
5 mins	Welcome and Self-Introduction	@YiYao Li (Product Owner)	<ul style="list-style-type: none">• Welcomed everyone and introduced the meeting's purpose.• Confirmed the goal to review Sprint 1 work, gather feedback, and discuss plans for the next steps.
15 mins	Sprint 1 Deliverables Overview	@YiYao Li	<ul style="list-style-type: none">• Developed the prototype interface.• Explained the distinction between Inner Scope (essential features) and Outer Scope (optional future improvements).• Finalized the technology stack (React, Django, MongoDB, ChatGPT, Netlify).• Demonstrated the developed interface.
10 mins	Feedback Discussion	@YiYao Li + Kate Coleman	<ul style="list-style-type: none">• Client Feedback: No major concerns; interested in ethical content generation.• Questions: The client asked about the content filtering mechanisms and data privacy.• Response: Explained basic content filtering implementation planned for Sprint 2.
15 mins	Sprint 2 Planning Overview	@YiYao Li	<ul style="list-style-type: none">• Presented the Sprint 2 goals: completion of core functionalities, including input processing, question generation, and data security.

			<ul style="list-style-type: none"> Discussed planned improvements: user-friendly input interface, question generation algorithm, secure data storage, and content filtering.
10 mins	Q&A and Conclusion	@YiYao Li + Kate Coleman	<ul style="list-style-type: none"> Q&A: Addressed additional questions from the clients about global accessibility and content relevance. Next Steps: Sprint 2 will focus on the planned goals, with a review meeting scheduled for the end of the sprint. Conclusion: Thanked everyone for their feedback and time.

💡 Q&A Session Notes

Question	Response
How will content be filtered for ethical standards?	Basic content filtering will be implemented in Sprint 2 to ensure the generated content adheres to ethical guidelines.
What measures are in place for data privacy and security?	Data will be encrypted and securely stored using access control mechanisms. The university's secure AWS services (Media Flux) will be used for storage.

💡 Additional Notes from Client

- Importance of Critical Thinking:** The client emphasized the importance of bringing critical thinking into everyday life and teaching users how to ask critical questions.
- AI's Role in Education:** The client mentioned that in university settings, students are often assumed to know how to ask and construct questions, but it's challenging. AI can help translate complex thoughts into structured questions.
- Ethical Content Generation:** There was a strong emphasis on not generating unethical questions.
- Data Storage Requirements:** The client prefers that data be stored securely within the university using services like Media Flux/Media Fox, which are secure AWS services paid for by the university.
- Reference for Data Services:** The client provided a link to the University of Melbourne's data services: [🔗 Research Data Management](#)
- Challenge of Question Generation:** The client noted that generating the right questions is challenging and should be a key focus. The system should cater to upper-level high school students, particularly those aged 14 to 23 (Year 9 to Year 10).
- Language and Accessibility:** The client stressed the importance of considering language diversity to ensure the system is accessible across multiple countries. The system should not prevent anyone from using it and should support diverse research needs.

✓ Action items

- Implement content filtering mechanisms in Sprint 2.
- Focus on secure data storage and privacy in the upcoming sprint.
- Ensure the system remains accessible and relevant across multiple countries and languages.

⬆️ Decisions

👉 Proceed with the current plan for Sprint 2, focusing on core functionalities and ethical content generation.

 Schedule the next Sprint Review Meeting after the completion of Sprint 2, around September 20th.

 Move Multi-language, Admin Console, and Data Archiving from Outer Scope to Inner Scope based on client requirements.

07/08/2024 Client Meeting

📅 Date & Time

Aug 7, 2024 3:15 pm - 4:05 pm

👤 Participants

- @Yiqun Liu
- @Yingying Guo
- **@Jionghao Song**
- @YiYao Li
- @Leyao Lyu
- The Client
- Other Group
- @Yujun Yan

📋 Goals

- Project introduction.
- Questions for the client.

🗣 Discussion topics

Time	Item	Presenter	Notes
5 mins	Self-Introduction	The client	
15 mins	Project and SWIP Lab Introduction	The client	<ul style="list-style-type: none">• What is the chatter box• Example of the chatter box
35 mins	Q&A Session	@YiYao Li and clients	

📝 Q&A Session Notes

Question from product owner	Response from the client
What is the most important requirement in this project	<ul style="list-style-type: none">• Efficiency: The process must be efficient, with a focus on digital output. Keep everything digital for flexibility.• Process Guidance: The system should guide users through a series of questions to input the correct information, enabling even those without expertise to drive their inquiry.

What about the dataset. How to build the dataset	<ul style="list-style-type: none"> Complex Issues: The dataset should address complex issues like climate change and racism, requiring manual cleaning and organization. Imagination & Testing: Encourage participants to use their imagination and challenge themselves.
Are we allowed to use the API of the open AI	<ul style="list-style-type: none"> GPT Integration: Any available API, including GPT, can be used.
How about the budget and the budget range	<ul style="list-style-type: none"> Small Budget: The budget is limited, with a need to keep discussions open.
How to validate the generated result	<ul style="list-style-type: none"> Validation: Validate results by checking if the generated questions offer plausible or preferable possibilities.
Beside the Chatter Box, what is other feature	<ul style="list-style-type: none"> Data Storage: Include a data bank. Ethics & Filters: Ensure ethical output with language and profanity filters.
What is the current framework, does "build on that" means user the previous code	<ul style="list-style-type: none"> Not build on the existing framework Creat our own
Is it a APP or Website	<ul style="list-style-type: none"> Website QR Code Integration: The website can support QR code functionality. Global Accessibility: The website should be accessible in other countries, particularly India, not limited to mobile devices.
What does the "previous result retrieval "means	<ul style="list-style-type: none"> No statically. Consider the place language and culture Keyword Search: Preference for picking up questions through a keyword search.
How long for the data storage	<ul style="list-style-type: none"> Perpetuity: The project should aim for long-term storage, not just immediate retrieval. Data Collection: Data will be played and collected for 12 months, possibly with self-deletion functionality after a certain period, as the focus is on capturing temporary insights.
Does the user input should only about the climate change	<ul style="list-style-type: none"> No Users will generate input on big, complex issues like climate change, encouraging their own inquiry process.

✓ Action items

- Understand the client's needs.
- Deepen the understanding of this project and the background
- Understand the feedback from the client.

 Decisions

4

Mentor Meetings

18/09/2024 Mentor Meeting

Date

Sep 18, 2024 6 pm - 6:20 pm

Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

Discussion topics

Time	Item	Presenter	Notes
15 mins	Report what we did last week	Everyone	<ul style="list-style-type: none">• Show ticket cards on Trello• Every team member reports what they did and plans to do for next week
5 mins	Mentor's Instructions	@Yujun Yan	<ul style="list-style-type: none">• Instructions in contribution• Document the changes made in sprint 2

Action items

- Mention user story updates in the Sprint 3 plan.
- Highlight success: completed one additional epic in retrospective meeting notes.

11/09/2024 Mentor Meeting

📅 Date

Sep 11, 2024 6 pm - 6:26 pm

👥 Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

🗣 Discussion topics

Time	Item	Presenter	Notes
10 mins	Report what we did last week	Everyone	<ul style="list-style-type: none">• Show ticket cards on Trello• Every team member reports what they did and plans to do for next week
15 mins	Mentor's Instructions	@Yujun Yan	<ul style="list-style-type: none">• Tell us what to do next• Tell us how to optimize the Trello board• Answer questions issued by Scrum Master
1 mins	Q&A	Everyone	All good 😊

✓ Action items

- Add a detailed description to the architecture diagram [@Jionghao Song](#)
- Use the Trello Review column before marking tasks as "Done." [@YiYao Li](#)
- Add additional documentation for the sprint plan, including story points calculation and person-day alignment [@Yiqun Liu](#)
- Add titles (e.g., Frontend, Backend, Full Stack) to each ticket, and use labels to clarify specialization. [@Yiqun Liu](#)

04/09/2024 Mentor Meeting

Date

Sep 4, 2024 6 pm - 6:31 pm

Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

Discussion topics

Time	Item	Presenter	Notes
10 mins	Report what we did last week	Everyone	<ul style="list-style-type: none">• Show ticket cards on Trello• Every team member reports what they did and plans to do
5 mins	Explain the backend	@Jionghao Song	<ul style="list-style-type: none">• Show the backend and explain the testing
15 mins	Mentor's Instructions	@Yujun Yan	<ul style="list-style-type: none">• Tell us what to do next• Tell us how to optimize the Trello board
1 mins	Q&A	Everyone	No more questions😎

Action items

- Ensure all tickets have detailed descriptions. Use sub-tasks for complex tickets, assigning each to a specific person.
- Strengthen error control mechanisms to improve reliability.
- Use AI tools (e.g., GPT, VS Code plugins) for regular code reviews. Perform reviews before merging, ideally on a weekly basis.
- Prioritize documentation and strictly follow the LMS checklist to ensure completeness.
- Include branch links in Trello ticket descriptions for better tracking and collaboration.
- Let the client know about changes to the backend technology.

28/08/2024 Mentor Meeting

Date

Aug 28, 2024 9 pm - 9:35 pm

Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

Discussion topics

Time	Item	Presenter	Notes
10 mins	Report what we did last week	@Yiqun Liu	<ul style="list-style-type: none">• Show ticket cards on Trello• Every team member reports what they did and plans to do
15 mins	Mentor's Instructions	@Yujun Yan	<ul style="list-style-type: none">• Tell us what to do next• Tell us how to optimize the Trello board
1 mins	Q&A	Everyone	No more questions😎

Action items

- Ensure all tickets are detailed and avoid using overly general or one-word titles.
- Specify the type of methods (e.g., guest method, PUT method) in the API documentation.
- Include input types and ensure an API key is documented for each feature.
- Track and follow up on team-client communication, especially during sprint review meetings.
- Add more specific tickets that align with the product backlog.
- Move backlog items back to the product backlog if incomplete.
- Create separate, detailed tickets for each backlog item, including relevant subtasks.
- Ensure changes are documented consistently across all related documents.
- Obtain client approval for any technology changes and document the rationale.

21/08/2024 Mentor Meeting

Date

Aug 21, 2024 6 pm - 6:35 pm

Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

Discussion topics

Time	Item	Presenter	Notes
20 mins	Self Report	@Yujun Yan	<ul style="list-style-type: none">• Everyone Report<ul style="list-style-type: none">a. What they have done for last weekb. Any Blockingc. Next Plan
10 mins	Go Through Sprint 1 Checklist	@Yujun Yan	<ul style="list-style-type: none">• To ensure that all requirements on the checklist have been covered and no important details are missed.
5 mins	Q&A	Everyone	

Action items

- Refine the name of branches on GitHub
- For each user story, link the priority with the scope
- Create “Sprint” content on confluence to record the sprint plan, review, and retrospective
- Make the subtask for product backlog

14/08/2024 Mentor Meeting

Date

Aug 14, 2024 6 pm - 6:20 pm

Participants

- @Leyao Lyu
- @Yiqun Liu
- @YiYao Li
- **@Jionghao Song**
- @Yingying Guo
- @Yujun Yan

Goals

- Report what we did last week
- Plan what to do for next week
- Issue blockers

Discussion topics

Time	Item	Presenter	Notes
10 mins	Report what we did last week	@Yiqun Liu	<ul style="list-style-type: none">• Show the documents on Confluence• Ask questions about ambiguous part
10 mins	Mentor's Instructions	@Yujun Yan	<ul style="list-style-type: none">• Help arrange the Trello board• Set up a template on Trello• Tell us what to do next

Action items

- Set up the Confluence landing page
- Add a link page with external links like Trello and GitHub
- Add a picture to the Persona section and format it as a template
- Organize Trello
- Gradually update the GitHub README
- Report the finalized tech stack to the client gradually
- Work on the product backlog
- Plan for Sprint 2.

Decisions

07/08/2024 Mentor Meeting

Date

Aug 7, 2024 6 pm - 6:30 pm

Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

Goals

- Introduction to Confluence

Discussion topics

Time	Item	Presenter	Notes
30 mins	Confluence Introduction	@Yujun Yan	<ul style="list-style-type: none">• Introduces the functionalities of Confluence and outlines the modules that showcase previous student work. Shows detailed components and content of each module.

Action items

- Start preparing for the Sprint 1 tasks
- Setup the Confluence

Decisions



04/08/2024 Mentor Meeting

📅 Date

Aug 4, 2024 3 pm - 3:20 pm

👥 Participants

- [@Leyao Lyu](#)
- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Yujun Yan](#)

📝 Goals

- Self-introduction
- Team Role Assignment
- Client meeting information

🗣 Discussion topics

Time	Item	Presenter	Notes
10 mins	Introduction	@Yujun Yan	<ul style="list-style-type: none">• Get to know each other
10 mins	Project Brief	@Yujun Yan	<ul style="list-style-type: none">• Record team member availability• Presenting upcoming client meetings

✓ Action items

- Team members introduce themselves
- Prepare for the [first client meeting](#) questions

⌚ Decisions

👉 [@Yiqun Liu](#) will serve as scrum master, where she is responsible for facilitating team collaboration, ensuring the Scrum process is followed, and removing obstacles to the team's progress.

👉 [@YiYao Li](#) will serve as a product owner, where she is responsible for communicating with clients and presenting questions at client meetings.

Team Meetings

10/09/2024 Team Meeting

Date

Sep 10, 2024 1:30 pm - 4:30 pm

Participants

- [@Yiqun Liu](#)
- [@Yingying Guo](#)
- [@YiYao Li](#)
- [@Leyao Lyu](#)
- [@Jionghao Song](#)

Goals

- Refine Trello board organization and update task priorities.
- Enhance understanding and implementation of MongoDB for optimized data handling.
- Streamline client settings and button interface for better user interaction.
- Ensure code quality and deployment efficiency through better branch management and review practices.

Time	Item	Presenter	Notes
30 mins	Review Trello Board	All	Discuss current board setup, identify missing tasks, and update priorities.
30 mins	Finalize Feature Checklist	All	Confirm which features are pending and assign responsibilities.
30 mins	Client Settings and Button Design	All	Discuss if the client settings import is properly configured and decide on the UI changes for settings button.
30 mins	Introduction to MongoDB	All	Basic overview of MongoDB operations relevant to our project.
40 mins	Analyze Data Storage Options	All	Deep dive into MongoDB's capabilities and limitations for our specific data needs.
30 mins	Branch and Code Review Strategies	All	Set standards for branch management and code review practices.

Action items

- Ensure everyone finalizes the tasks last week on Trello.
- All team members to review and pull the latest updates from the main branch before next meeting.
- Perform an initial code review session focusing on MongoDB integration by end of this week.

⌚ Decisions

👉 Change setting button into creativity level board

03/09/2024 Team Meeting

Date

Sep 3, 2024 4:30 pm - 6 pm

Participants

- [@Yiqun Liu](#)
- [@Yingying Guo](#)
- [@YiYao Li](#)
- [@Leyao Lyu](#)
- [@Jionghao Song](#)

Goals

- Organize Trello
- Analyze Data Storage Using MongoDB

Discussion topics

Time	Item	Presenter	Notes
30 mins	Organize Trello	All	
20 mins	Get to know MongoDB	All	
40 mins	Analysis data based on this project	All	

Action items

- Ensure everyone finalizes the tasks last week on Trello

Decisions



29/08/2024 Team Meeting

Date

Aug 29, 2024 11:30 am - 1:00 pm

Participants

- [@YiYao Li](#)
- [@Yiqun Liu](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)
- [@Jionghao Song](#)

Goals

- Review the checklist provided by the mentor.
- Allocate tasks for implementing the ChatGPT API, including prompt engineering.
- Provide more detailed subtasks on Trello for better task tracking.
- Assign tasks to each team member.
- Attempt to implement the user interface using React.
- Review the APIs to be used and carefully check their documentation.

Discussion topics

Time	Item	Presenter	Notes
30 mins	Checklist from mentor	@Yiqun Liu	<ul style="list-style-type: none">• Review the mentor's checklist and clearly identify the requirements for each team member.
15 mins	Subtasks on Trello	All	<ul style="list-style-type: none">• Review the product backlog for Sprint 2. Discuss and provide more detailed subtasks for each user story.
30 mins	Developing discussion	All	<ul style="list-style-type: none">• Address any blockers related to the implementation of the UI and ChatGPT API.
15 mins	Future work	All	<ul style="list-style-type: none">• Plan the next steps, discuss and assign tasks to each team member

Action items

- [@Yiqun Liu](#) [@Yingying Guo](#) [@Leyao Lyu](#) need to familiar with the frontend platform - React and try to implement it
- [@Yiqun Liu](#) [@Yingying Guo](#) [@Leyao Lyu](#) [@Jionghao Song](#) [@YiYao Li](#) need to familiar with the chatgpt developer documentation and know how to implement it with prompt engineering.

Decisions

 Schedule the next team meeting for August 30 and 31, 2024, at 3:00 PM, focusing on Sprint 2 development.

27/08/2024 Team Meeting

Date

Aug 27, 2024 2:30 pm - 4 pm

Participants

- [@YiYao Li](#)
- [@Yiqun Liu](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)
- [@Jionghao Song](#)

Goals

- Go through the sprint 2 checklist
- Week plan for Sprint 2
- Tasks allocation for each team member
- Go through the platforms that are going to use
- Finalise the API keys and URL

Discussion topics

Time	Item	Presenter	Notes
30 mins	Sprint 2 Checklist	@Yiqun Liu	<ul style="list-style-type: none">• Go through the sprint 2 checklist, identify the sprint requirements clearly for each team member
15 mins	Weekly plan	All	<ul style="list-style-type: none">• Go through the weekly plan for Sprint 2, discuss and allocate user stories for each week
30 mins	Platform discussion	All	<ul style="list-style-type: none">• Go through the platforms that are going to use• Finalise the chatGPT API keys and URL
15 mins	Future work	All	<ul style="list-style-type: none">• Decide what to do next, discuss and assign the tasks to each team member• Be prepared for the mentor meeting tomorrow

Action items

- [@Yiqun Liu](#) [@Yingying Guo](#) [@Leyao Lyu](#) need to familiar with the frontend platform - React
- [@Jionghao Song](#) [@YiYao Li](#) need to familiar with the backend platform - Django

Decisions



20/08/2024 Team Meeting

Date

Aug 20, 2024 12 pm - 4 pm

Participants

- [@YiYao Li](#)
- [@Yiqun Liu](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)
- [@Jionghao Song](#)

Goals

- Update Confluence content.
- Finalize technical decisions.
- Review and update GitHub repository.
- Check Trello board status and update tasks.

Discussion topics

Time	Item	Presenter	Notes
1hr30mins	Confluence Updates	All	<ul style="list-style-type: none">• Loading Page: Decided to include photos of each team member on the loading page.• Section Review: Reviewed all sections of Confluence for accuracy and consistency.• Font Size Consistency: Adjusted font sizes across Confluence to ensure uniformity.• Personas Region: Corrected the personas section for accuracy.• Prototype Interface: Added a new section for prototypes, including links and screenshots.• Architecture Section: Created and added content for the architecture section, including MongoDB Database Model and API details.
1hr	Technical Decisions		<ul style="list-style-type: none">• Stack Update: Agreed to replace Docker and AWS with Netlify for a serverless approach. This decision was made based on team discussions to simplify deployment and management.
1hr	GitHub Repository		<ul style="list-style-type: none">• README Review: Ensured all README content is accurate and up-to-date.• Figma Link Update: Corrected the Figma link in the README.

			<ul style="list-style-type: none"> • Changelog Inclusion: Decided to include the changelog directly within the README file rather than maintaining a separate <code>changelog.md</code>, for ease of updating each sprint.
30mins	Trello Board		<ul style="list-style-type: none"> • To-Do Review: Verified completion of the previous to-do items. • New To-Dos: Added and updated new tasks for the current sprint.

✓ Action items

- Update the loading page on Confluence with team member photos.
- Finalize the Confluence sections and ensure consistency in font size.
- Implement the new technical stack using Netlify.
- Make the necessary changes to the README on GitHub and include the updated Figma link.
- Verify and update tasks on Trello.

⌚ Decisions

- 👉 Use Netlify for serverless hosting instead of Docker and AWS.
- 👉 Include the changelog directly in the README file for easier management.
- 👉 Schedule the next team meeting for August 22, 2024, at 7:00 PM, focusing on Sprint 2 planning.

15/08/2024 Team Meeting

Date

Aug 15, 2024 2:30 pm - 3:30 pm

Participants

- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)

Goals

- Discuss and evaluate the work completed by each team member
- Allocate tasks for the next meeting on Tuesday
- Discussion on the design style of the website

Discussion topics

Time	Item	Presenter	Notes
10 mins	Work Presentation	All	<ul style="list-style-type: none">• Each group member introduce their work to others
40 mins	Prototype Design Discussion	All	<ul style="list-style-type: none">• Everyone discusses and shares opinions on website design in Confluence• Adjust the design of the Prototype Part
10 mins	Future Work	All	<ul style="list-style-type: none">• Check rest work with checklist• Decide what to do next• Be prepared for the team meeting next week

Action items

- [@Yiqun Liu](#) : needs to update in Product Backlog Part in Trello and Confluence; draft loading pages in Confluence.
- [@Leyao Lyu](#) : needs to update the Persona. allocate the Product Backlog in Trello.
- [@YiYao Li](#) : needs to ask mentor the details about the structure on Trello and how to deploy it; prepare for delivery to a client meeting; update Github README file.
- [@Jionghao Song](#) : needs to update the Technical Consideration and make a technical structure.
- [@Yingying Guo](#) : needs to draft Prototype and update Technical Consideration; write Team Meeting Notes.

Decisions

 @Yingying Guo will be the Quality Assurance Leader

13/08/2024 Team Meeting

Date

Aug 13, 2024 3 pm - 5 pm

Participants

- [@Yiqun Liu](#)
- [@YiYao Li](#)
- [@Jionghao Song](#)
- [@Yingying Guo](#)
- [@Leyao Lyu](#)

Goals

- Discuss and evaluate the work completed by each team member
- Prepare for the mentor meeting tomorrow
- Allocate tasks for the next meeting on Thursday

Discussion topics

Time	Item	Presenter	Notes
20 mins	Work Presentation	All	<ul style="list-style-type: none">• Each group member introduce their work to others
60 mins	Background Part Discussion	All	<ul style="list-style-type: none">• Everyone discusses and shares opinions on Background Part in Confluence• Adjust the structures of the Background Part
30 mins	Requirement Part Discussion	All	<ul style="list-style-type: none">• Everyone discusses and shares opinions on Requirement Part in Confluence• Adjust the structures of the Requirement Part
10 mins	Future Work	All	<ul style="list-style-type: none">• Decide what to do next• Be prepared for the mentor meeting tomorrow• Sprint1 - Individual Contributions

Action items

- [@Yiqun Liu](#) : needs to change in User Story Part; prepare for delivery to a mentor meeting
- [@Leyao Lyu](#) : sets up the Trello
- [@YiYao Li](#) : plans for Sprint2 plan
- [@Jionghao Song](#) needs to update the Scope Part and Non-Functional Part; write Team Meeting Notes
- [@Yingying Guo](#) : drafts Technical Consideration

 We decided to add a sign-in & sign up function

06/08/2024 Team Meeting

📅 Date

Aug 6, 2024 9:30 pm - 10:20 pm

👥 Participants

- [@Yiqun Liu](#)
- [@Yingying Guo](#)
- [@YiYao Li](#)
- [@Leyao Lyu](#)
- [@Jionghao Song](#)

📋 Goals

- Ice break
- Deepen the understanding of the project
- Discuss the current confusion and prepare for the client meeting

🗣 Discussion topics

Time	Item	Presenter	Notes
15 mins	Self-introduction	All	<ul style="list-style-type: none">• Get to know each other
10 mins	Project Brief	All	<ul style="list-style-type: none">• Deepen the understanding of this project
25 mins	Discuss the current questions	All	<ul style="list-style-type: none">• Prepare the questions for the client meeting

✓ Action items

- Everyone introduces themselves including their educational background
- Everyone discusses their current confusion or questions about the project
- Prepare for the first client meeting

⌚ Decisions

👉 [@YiYao Li](#) will be the representative who asks questions.

👉 [@Yiqun Liu](#) will be the recorder for the client meeting.