SW Engineering CSC648/848

Application Name: NNSE
Section 02, Team 06

Name	Role
Anzara Ausaf	Team Lead / Frontend Team Lead
Harris Chan	Git Master / Document Master
Nicolas Carrillo	Database Engineer / Backend Team Lead
Michael Ho	Database Engineer
Chance Vodnoy	Backend
Casey Steven	Backend
Jiqing Liu	Backend

Milestone 2

February 25, 2025

Revision Table			
Revisions	Date	Author(s)	Description of Changes
1	02/25/2025	Harris Chan	Initial Document Created
2	02/28/2025	Nicolas Carrillo	High level Architecture, Database Organization, High Level UML
3	03/03/2025	Harris Chan	Key Risk Added
4	03/05/2025	Jiqing Liu	P6 Modified
5	03/06/2025	Anzara Ausaf	Designing UI Mockups
6	03/08/2025	Michael Ho	Prioritized Functional Requirements
7	03/10/2025	Casey Steven	P3 Modified
8	03/13/2025	Anzara Ausaf	Final Draft confirmed

1. Functional Requirements V2

Priority 1 (Must have):

Users:

1. The Searching Aspect:

This search engine will provide basic search features to help individuals find AI tools that fall under categories of Academic Assistance, productivity, career development, mental health support, and creative applications.

7. AI-Powered Tool Matching:

Users can input their needs, and the AI will recommend the best AI tools available. Example: A student looking for AI-powered research assistants will get recommendations for AI citation tools, summarizers, and research databases.

17. Language Preferences:

This feature will allow the user to set a language they are comfortable with from a list of languages provided by the search engine. E.g- English, Spanish, French etc.

19. The Relevancy Filter:

Users can filter search results by relevance and content type (music, school, research, etc.)

Admin:

2. User Authentication:

We want our users to be able to sign up and login. This helps the user to keep track of previously viewed searches and AI tools if they wish to revisit them. Additionally, users can apply to be verified AI app developers, which will be able to post and showcase their own AI apps.

2.1

While this has benefits for users, it also greatly benefits admins for the purposes of moderation. This would allow for the reporting and removal of users who may be posting malicious, dishonest reviews or programs

15. Feedback from Users:

Users will be prompted to give feedback after a successful AI tool implementation into their own project. This will allow for a growth in community and trust within the app.

15.1

Feedback from users would also allow the team to gain insight on what users like or dislike about the experience in order to help with the improvement of the search engine.

Priority 2 (Desired):

Users:

4. Lists sidebar:

The application will store every chat bot recommendation conversation and they will be listed in a way that will be easy to access, manage and delete. We hope to make it visually appealing. The sidebar of conversations will also allow users to rename conversations (if they would prefer to) with a title they would like to mark and remember the conversation by.

5. List sidebar part 2:

The list of conversations sidebar will also have a search bar if the user wants to look up a certain conversation(in order to find the correct AI tools). The renaming aspect of the sidebar also allows the user to look up the specific conversation more easily as they can title them according to their preference.

8. Bookmarking:

Users will be able to bookmark a certain search result they find useful instead of having to scroll through an entire re searched phrase. There will be a bookmark tab where users can directly access what information they have bookmarked.

14. Learning Summarize Mode:

Users will be able to activate learning mode, which will allow them to summarize the AI-tool information. Reading Documentation gets to be too much sometimes, so by summarizing this it will allow for a better understanding from the user.

16. AI-Generated Tool Descriptions & Reviews

AI can summarize user reviews and explain how AI tools work, helping users make informed decisions.

18. The Translator tool:

This feature allows the user to translate a piece of information.

20. Mental Wellness Support:

The Mental Wellness feature creates a safe, supportive space for users seeking AI-powered tools to enhance their emotional well-being. This thoughtfully designed interface helps users discover personalized mental health resources.

Admin:

3. UI :

As mentioned in our introduction, we believe in a minimalistic design. Our Search engine will be on a clean interface that is easy to look at and not overwhelming. This will help the user engage with the knowledge they want to acquire instead of getting lost in design that is unnecessary.

3.1

This design choice is based on initial thoughts from the team, but design preference would depend from user to user and realistically, any design that is functional and not too inconvenient would work.

Priority 3 (Opportunistic):

Users:

6: Cross-device synchronization:

User Conversations and search results will be accessible from all devices the user logs into. Allowing a smooth experience no matter the device the user chooses to use.

9. Exporting different files:

Users will be able to share images, PDFs, .txt, .mp4 files, etc., for reference during their searching experience.

10. Interface mode:

Users can toggle between different modes, such as light mode/dark mode/any custom color for theme preferences. They will also be able to toggle different display modes that help to give the user an easy experience for their eyes, such as night shift mode, or be able to adjust warmer or cooler tones for their viewing experience preferences.

11. Citation generator:

For whatever AI program the user wishes to use, the citation feature automatically creates properly formatted citations, assisting them in avoiding acts of plagiarism and following honesty regulations.

13. Time Filter Options:

This advanced feature will allow users to filter their search results based on time. They can further filter their searches for time ranging from last 24 hours, past week, past month, past year, past decade, and all.

Admin:

12. Guest mode:

Users will be able to use the platform without having to register but that will remove the privilege to bookmark/save tools they found in their searches

12.1

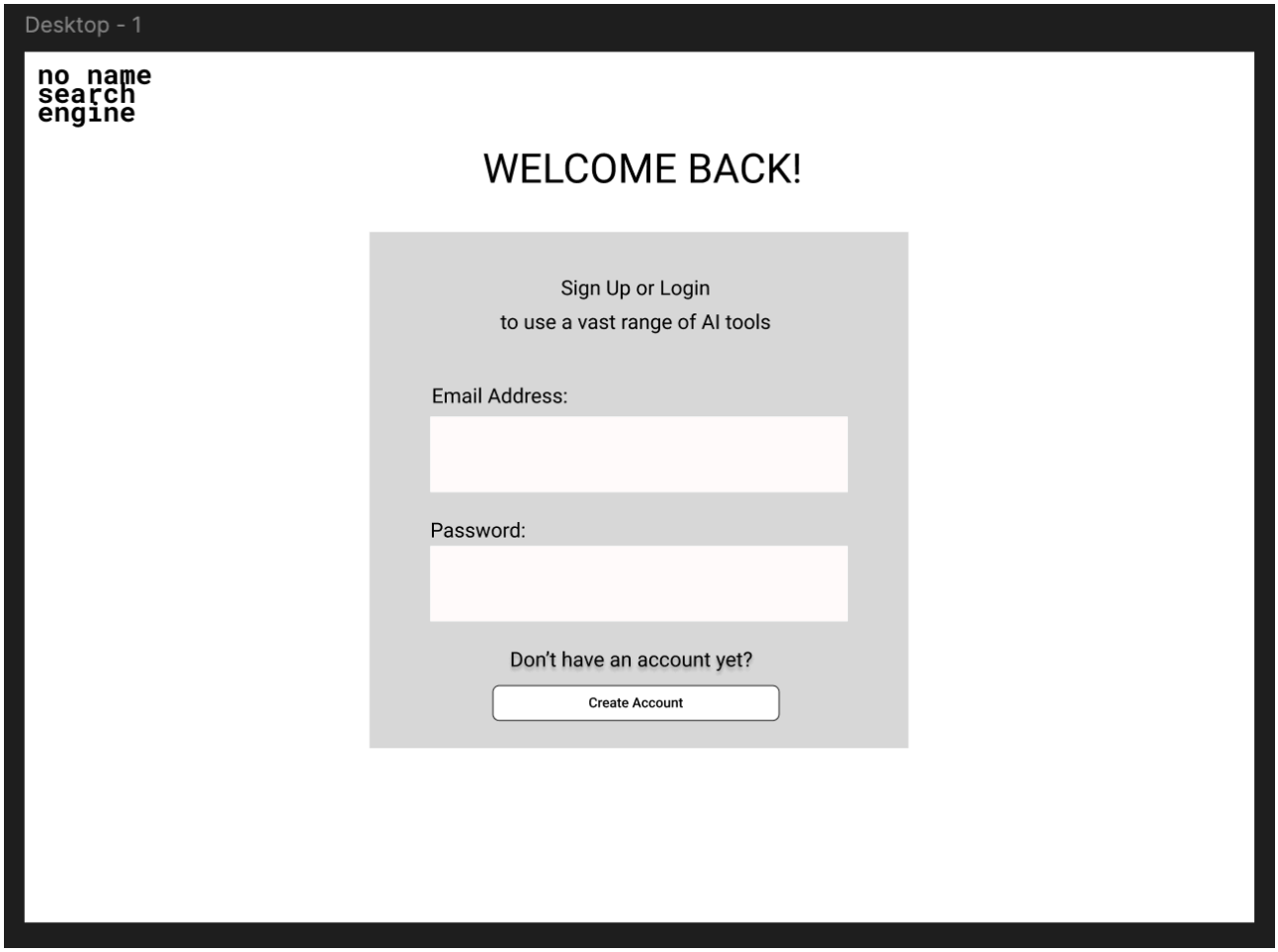
This may remove other privileges, an example being the ability to leave reviews on the ai tools. This would be to ensure that it’s not possible for unregistered guests or bots to spam products with falsely negative reviews.

12.2

This is also mostly for moderation purposes so that admins can distinguish registered and unregistered users. Technically, registering and having an account could be a requirement to use the search engine.

2. UI Mockups and UX Flows

User Login Page:



For the Login Page (Desktop - 1):

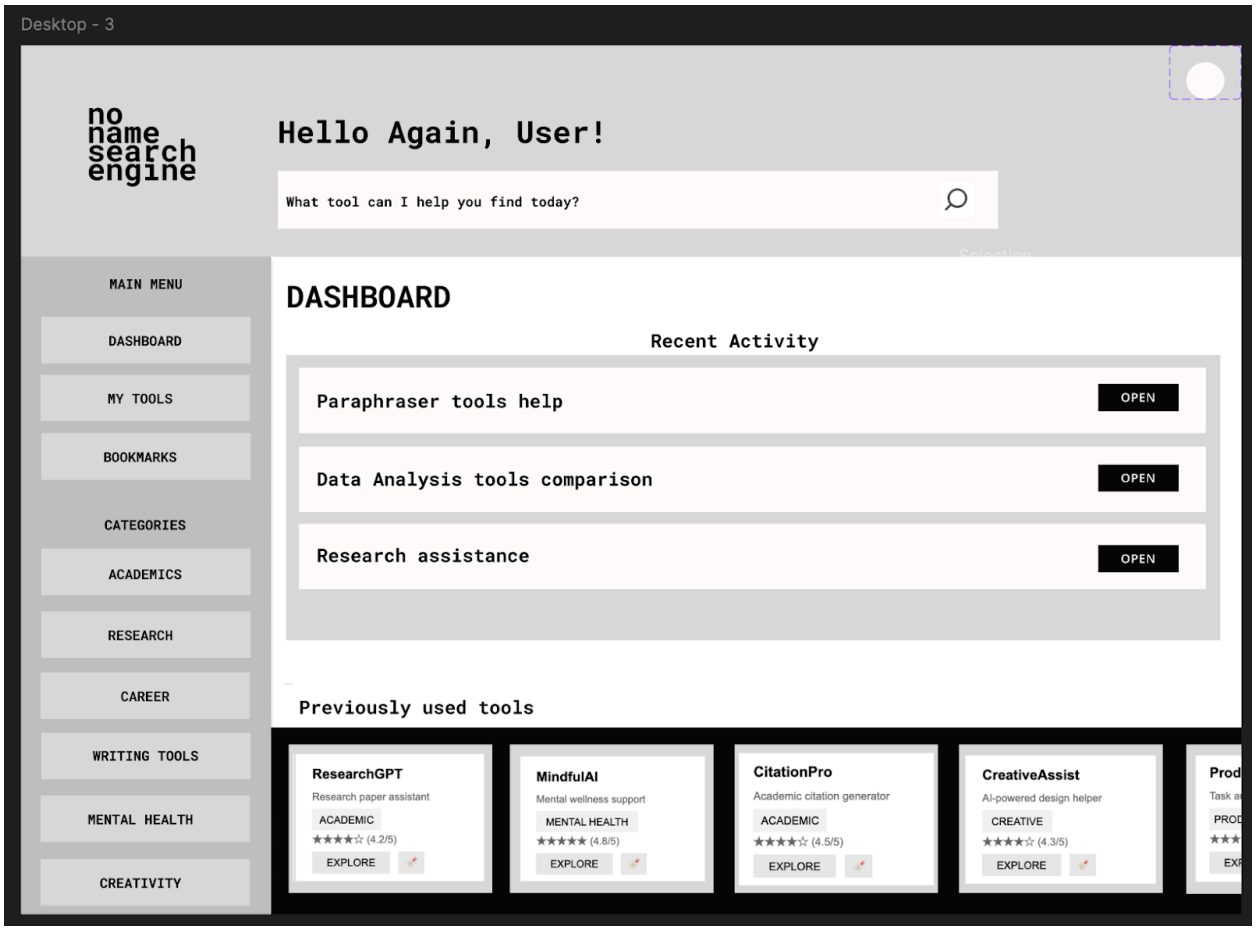
Enter your email and password to sign in. New users can create an account to access AI tools for academic research and productivity.

Sign Up Page:

For the Create Account Page (Desktop - 2):

Enter your Full Name and e-mail, then create a password. Confirm your password by entering it twice.

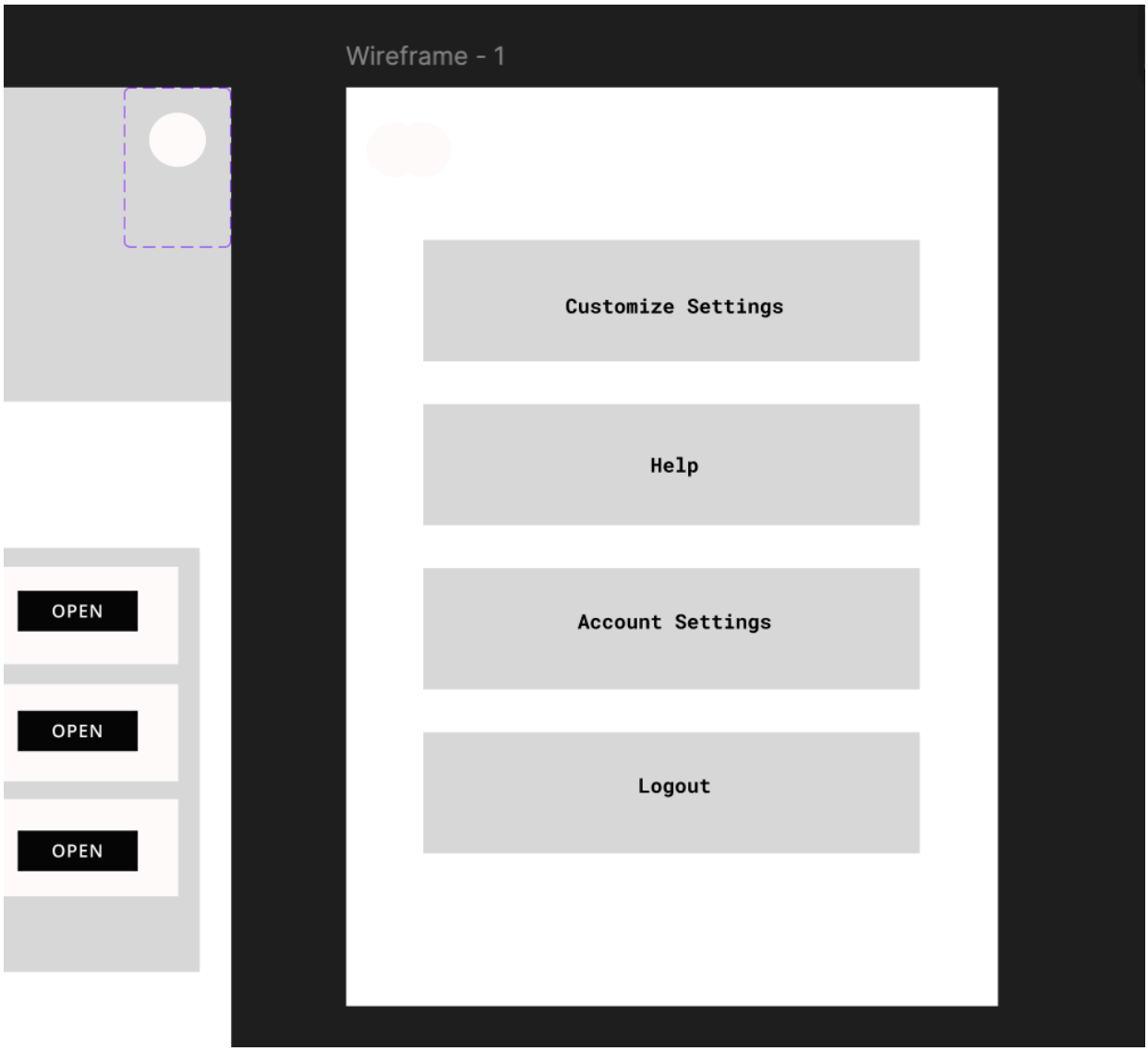
Main Page:



For the Dashboard (Desktop - 3):

View recent activity and previously used tools. Search for AI tools or browse by category using the left menu.

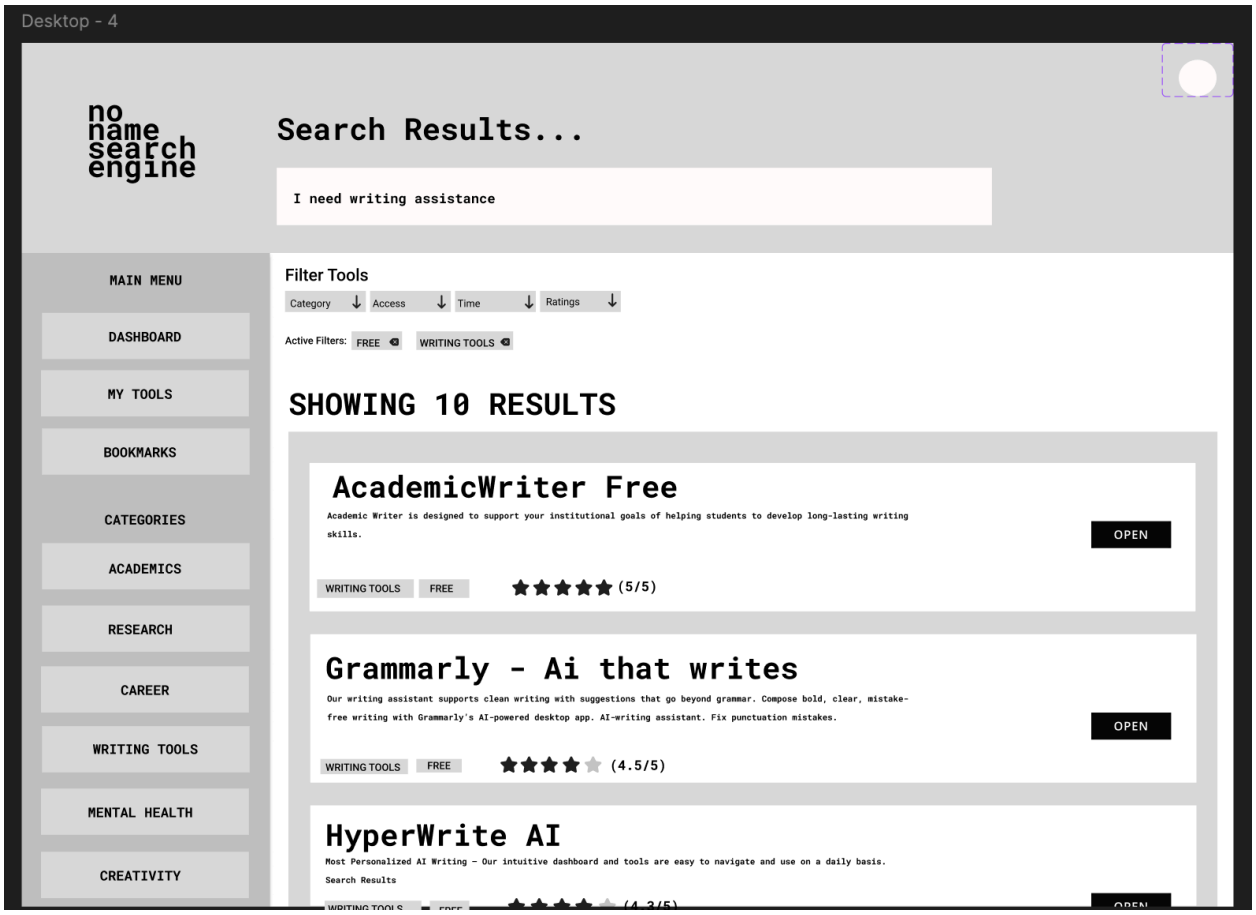
User Sign Drop Down Menu:



For the User Menu Dropdown (Wireframe - 1):

Manage your settings, get help, or sign out. Select "Customize Settings" for display preferences, "Account Settings" to update your profile, or "Logout" to sign out.

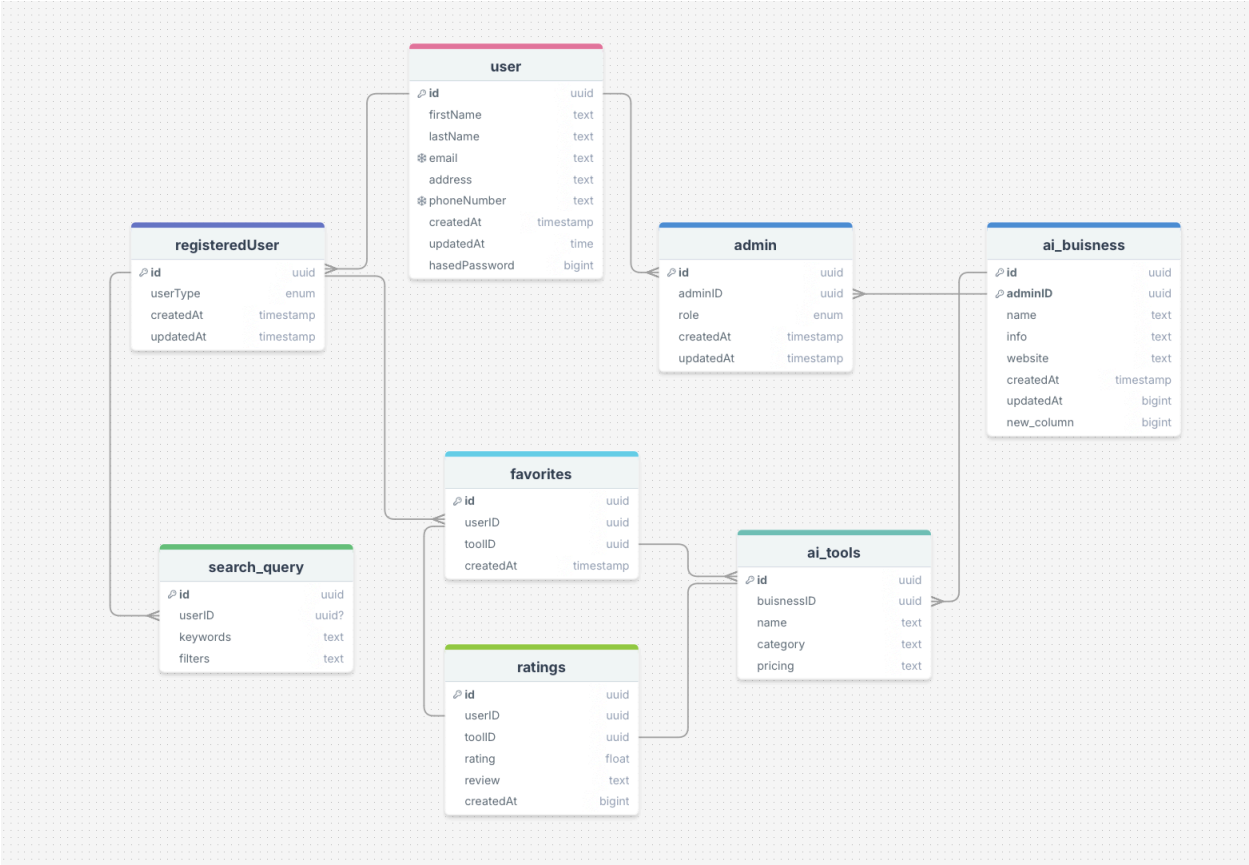
Search Results Page:



For the Search Results Page (Desktop - 4):

Showing what a search result from the search bar would look like. Browse filtered AI tools based on your search. Here, searched for writing assistance. Filter options available as listed - category, access, time, and rating filters to refine results. Currently showing free writing tools with detailed descriptions and ratings.

3. High-level Architecture, Database Organization



The database is structured to support the MVP of our application. This structure will allow us to have 2 different types of users, Registered common users and Admins. With these users we will be able to Post and Get different types of AI tools provided by different companies where users will be able to use review and manage AI-related Services

Below are the main tables:

1. User Management

- **user**: Stores general user information such as ID, name, email, address, phone number, timestamps, and password
- **registeredUser**: Defines different user types enum and timestamps for registration.
- **admin**: Holds admin-specific details, including role management and timestamps.

2. AI Business & Tools

- **ai_business**: Represents AI-related businesses, storing names, descriptions, websites, and admin IDs for AI business ownership tracking.

- **ai_tools**: Stores AI tools, their business association, category, and pricing, basically allowing the business to post what their AI tools are capable of doing.

3. User Interactions

- **favorites**: Allows users to save their favorite AI tools.
- **ratings**: Enables users to rate and review AI tools. We think this is going to be one of the most important user interactions because just like any other community based app, reviews are what gives people the confidence to go with that tool.
- **search_query**: Stores users' past searches, keywords, and filters for analytics or recommendation features, we want to make sure that the application is as smooth and easy for the user so we want to make things faster and cleaner.

Media Storage

- Since the system deals with AI tools and businesses things such as media like images, logos, or videos may be associated with them as well as description bios and pricing.
- **Decision**: Media files (images, videos) will be stored in a **file system** (e.g., AWS S3, Firebase Storage) instead of BLOBs in the database. The database will store file paths or URLs to the media.

Search/Filter Architecture & Implementation

- **Search Algorithm**: A full-text search will be implemented using **SQL with LIKE queries** for partial matching on AI tool names, business names, and categories.
- **Indexing**: Relevant columns such as name, category, and info in ai_tools and ai_business will be indexed for optimized search performance.
- **Filters**: Stored in the search_query table, allowing users to filter by categories, pricing, and business names.

1. Authentication & User Management

- POST /register: Register a new user/admin.
- POST /login: Authenticate users and generate JWT tokens.
- GET /user/{id}: Retrieve user profile.

2. AI Tool Discovery

- GET /ai-tools: Fetch all AI tools (with optional search filters).
- GET /ai-tools/{id}: Fetch details of a specific AI tool.

3. Favorites & Ratings

- POST /favorites: Add an AI tool to favorites.

- DELETE /favorites/{id}: Remove an AI tool from favorites.
- POST /ratings: Submit a rating and review for an AI tool.
- GET /ratings/{toolID}: Fetch all reviews for an AI tool.

Significant Algorithms or Processes

- Ranking AI Tools: AI tools will be ranked using an average rating algorithm (weighted rating system based on review count).
- Recommendation System (Future Enhancement): Past searches and favorited items will be used to suggest AI tools.
- Language Preferences: Searches should be allowed in multiple languages, based on user preference.

Software Tools & Frameworks

- Frontend: React with Vite, (for performance and modern component-based UI).
- Backend: Fastify (lightweight Node.js framework for high-performance APIs).
- Database: MySQL
- Storage: Google Compute Engine/Cloudinary for Storage for media files.
- Authentication: FireBase
- Analytics: Google Analytics
- API Calls: Fetch for frontend API communication since we are already using Node

4. High-Level UML Diagram

The Architecture for both our Data Diagram and UML follows the needed OOP inorder to support our design and meet our MVP Requirements. Since the image UML Diagram is to small we have converted it into human readable structure

User Management System

1. **User (Base Class):**
    - Stores basic user information (ID, name, email, address, phone, password)
    - Tracks when accounts are created and updated
    - Handles core functions like registration, login, and profile management
2. **RegisteredUser (Extends User):**
    - Regular users of the platform
    - Can manage their favorite AI tools
    - Can submit ratings and reviews
    - Can search for AI tools
3. **Admin (Extends User):**
    - Special users with administrative privileges
    - Can manage AI businesses and their tools
    - Can manage other users on the platform

AI Business & Tools System

4. **AIBusiness:**
    - Represents companies that provide AI tools
    - Stores business details (name, description, website)
    - Tracks which admin manages the business
    - Can have multiple AI tools associated with it
5. **AITool:**
    - Individual AI tools offered by businesses
    - Includes details like name, description, category, and pricing
    - Can be rated and reviewed by users
    - Stores media like screenshots or logos

User Interactions System

6. **Favorite:**
    - Tracks which AI tools users have saved as favorites
    - Allows users to quickly access their preferred tools
7. **Rating:**
    - Stores user reviews and ratings for AI tools
    - Includes numerical score and written feedback
    - Essential for community-based recommendations
8. **SearchQuery:**

- Records user search history and preferences
- Stores keywords and filters used
- Supports analytics and future recommendation features

Service Layer

9. **AuthService:**
   - Handles user authentication and security
   - Manages user registration and login processes
   - Validates security tokens
10. **AIToolService:**
    - Manages all AI tool operations
    - Handles tool creation, updates, and deletion
    - Provides tool ranking algorithms
11. **User Interaction Service:**
    - Manages user interactions with the platform
    - Handles favorites, ratings, and search queries

Data Access Layer

12. **UserRepository:**
    - Provides database access for user-related operations
    - Handles finding, creating, updating, and deleting user records
13. **AIToolRepository:**
    - Provides database access for AI tool operations
    - Supports filtering and searching for tools
14. **InteractionRepository:**
    - Manages database operations for user interactions
    - Handles favorites, ratings, and search history

Key Relationships

- Users can be either regular users or administrators
- Regular users can manage favorites, submit ratings, and create search queries
- Admins manage AI businesses
- AI businesses contain multiple AI tools
- AI tools receive ratings from users
- Service classes use repositories to access data

5. Identify actual key risks for your project at this time

5.1 Skill Risks:

Risk:

- Some members are not certain how to perform certain jobs or functions with new configurations.

Mitigation Plan:

- Invite users to ask questions.
- Use provided class materials (tutorials, documentation) to get caught up.

Risk:

- A team member falls behind in learning or fails to complete necessary learning before coding activities.

Mitigation Plan:

- Learn the skills sooner to prevent cramming.
- Create study timetables for each person and review them with the group.
- When feeling overwhelmed, immediately notify and request further support or a lesser task size.

5.2 Schedule Risks

Risk:

- Unanticipated code changes could cause merge conflicts or damage the other parts of the app.

Mitigation Plan:

- Post immediately following any major update or change in Discord or the team's messaging channel.
- Always make a Pull Request (PR) and request code review before merging.

Risk:

- One team member is not able to finish the task within the deadline.

Mitigation Plan:

- Break down work into smaller bite-sized chunks and track progress within a project planning software (Trello, GitHub Projects).
- Seek help as soon as the delay is likely to happen—don't leave it till the end.

Risk:

- Not appreciating complexity results in scope creep and delayed milestones.

Mitigation Plan:

- Re-prioritize and review the activities weekly.
- Use the rolling-wave planning approach: detail close-term activities but leave the further-out activities relatively open.
- As the complexity of the feature grows, refactor or simplify it with the team.

5.3 Team risks

Risk:

- The team members are unaware of either the activities to undertake or stay jobless.

Mitigation Plan:

- Maintain a live task board (Trello) with deadlines and assignees.
- Approve new work items weekly or bi-weekly during planning meetings.
- Every individual is periodically checked by the team lead or the project manager to ensure a task is assigned.

Risk:

- Having many people contributing towards the same file/piece of code creates integration issues.

Mitigation Plan:

- Select a branching model (e.g., Git Feature Branch Workflow) and merge synchronously with regularity.
- Discord or Slack when developing a new branch or altering a crucial file.

Risk:

- Some members do not join the meetings or respond to the messages, causing delays.

Mitigation Plan:

- Document strives to contact unresponsive members and bring the issue to the professor when needed.
- Keep notes for meetings so absent members can follow with ease.
- Set clear expectations regarding attendance and responsiveness.

5.4 Legal/Content Risks

Risk:

- Usage of copyrighted material (images, text, sound/video) without licenses.

Mitigation Plan:

- Ensure all third-party content is open source, public domain, or properly licensed.
- Make a note of the type and origin of any outside material.
- In case of doubt, substitute dubious content with unequivocally licensed or homegrown resources.

Risk:

- Including libraries or frameworks with limited licenses or fees.

Mitigation Plan:

- Check the license (e.g., Apache, GPL, MIT) for each library before using it.
- Only use class-approved or popular open-source frameworks.

5.5. Appropriate Security and Privacy Risks

Risk:

- Potential data compromise or misuse of confidential user data.

Mitigation Plan:

- Secure user authentication using salting/hash passwords and SSL.
- Restrict database accessibility to needed roles.
- Store user data responsibly, following best practice or regulatory guidelines when handling actual user data.

Risk:

- Code base vulnerabilities, typically where third-party dependencies are used.

Mitigation Plan:

- Keep all dependencies current to patch known vulnerabilities.
- Implement code reviews and automated testing (at least security testing).
- Occasionally review the logs for suspicious usage whenever user data is stored.

5.6 Technical/Integration Risks

Risk:

- Blending front end, back end, and database with new frameworks may produce unforeseen mistakes.

Mitigation Plan:

- Create a first little "slice vertical" (like we are currently doing) and refine.
- Perform regular integration testing with code changes.
- Maintain documentation for architecture decisions (API endpoints and data schemes) and communicate them with the entire team.

Risk:

- Tool or environmental setup incompatibilities (e.g., production vs. local).

Mitigation Plan:

- Create a full instructions setup for all members of the team (step-by-step process, environment variables, etc.).
- Standardize the development environments with Docker or other containerization where appropriate.

6. Project management

In M2's plan our team manages tasks and communication through a structured workflow. The plan is to use Trello for task tracking, but for now we will mainly use Discord for real-time coordination and add the name of the person responsible for each task in the google doc, as well as a deadline at the bottom of the document.

Discord is currently the main communication channel for our team, so all front-end/back-end coordination happens in the general channel on the server.

In future project plans:

1. Our team plans to use a more standardized task tracking model (Trello), primarily by using shared Trello boards to separate tasks into columns (e.g., Pending, In Progress, Reviewed, Completed) and assigning cards to specific members and deadlines.
2. More detailed communication channel separation: create dedicated channels (e.g. #FrontendDevelopment, #BackendIssues) through Discord for instant discussion of technical details or unexpected issues.
3. Documentation: consider using doxygen to generate documentation (js is not the main supported language), or use the alternatives JSDoc Annotation + Markdown PDF

Team lead ensures that all members have read and understood this document.	Yes, 03/13/2025 Anzara Ausaf