

CVWO AY2023/2024
Mid-Assignment Submission: AskWhat

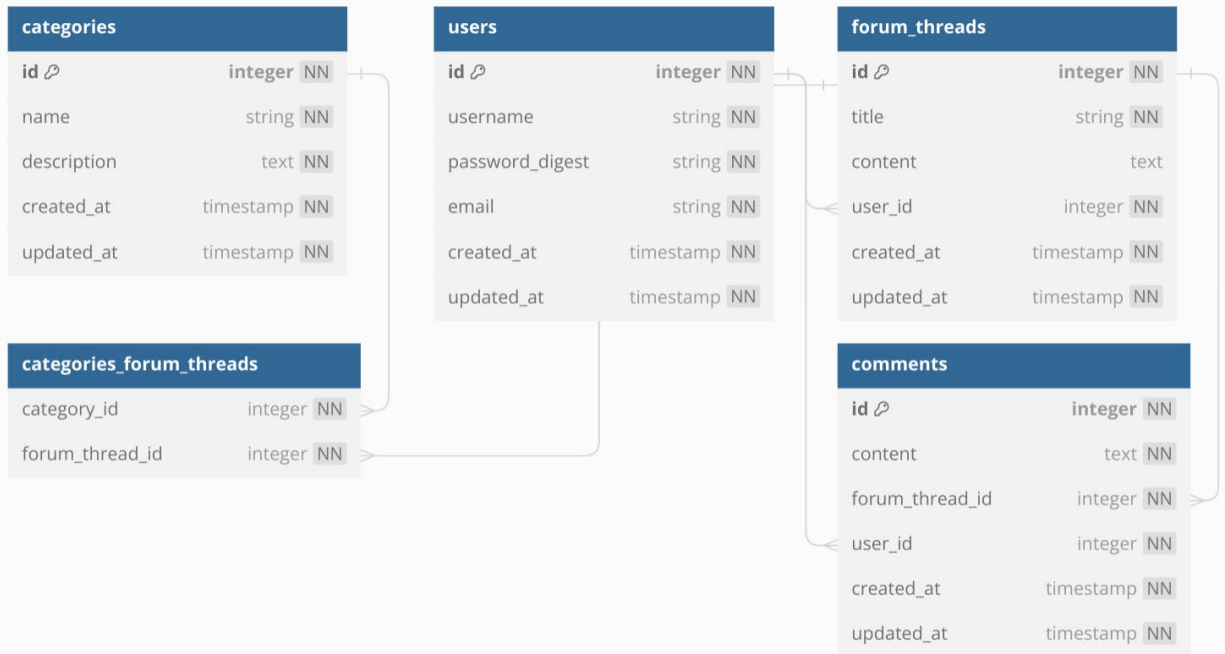
Introduction

AskWhat, a user-friendly web forum for community discussions on various topics.

Use Cases

Action	Basic Flow	Alternate Flow
User Registration	1. User fills in sign up form. 2. System validates and grants access.	1. User enters invalid information. 2. System denies access and prompts retry.
User Authentication	1. User enters credentials. 2. System validates and grants access.	1. User enters invalid credentials. 2. System denies access and prompts retry.
User Reads Thread	1. User access forum's main page/specific category. 2. System displays thread.	1. System implements pagination when data is excessive.
User Creates/Updates Thread	1. User enters thread details. 2. System validates and saves/modifies thread.	1. System displays an error message and prompts user to correct the input.
User Deletes Thread	1. User chooses thread to delete. 2. System deletes the thread and updates thread list.	1. System displays an error message and prompts user to try again.
User Read Comments	1. User views a thread. 2. System displays comments.	1. System paginated when data is excessive.
User Creates/Updates Comment	1. User inputs comment details. 2. System validates and saves/modifies comment.	1. System displays an error message and prompts user to correct the input.
User Deletes Comment	1. User chooses a comment to delete. 2. System deletes the comment and updates comment list.	1. System displays an error message and prompts user to try again.
User searches for Thread using Keyword/Tag	1. User inputs/select search feature. 2. System finds and displays searched results.	1. No results found.

Database Design



NN = Not Null

Tech Stack

The Model-View-Controller (MVC) framework will be used with the view as frontend and model and controller being the backend.

Frontend:

- **React.js with TypeScript** for developing responsive UI components.
- **Material-UI (MUI)** for a consistent and responsive design.

Backend:

- **Ruby on Rails** for handling server-side logic, using its MVC architecture.
- **RESTful APIs** for communication between frontend and backend.

Database:

- **PostgreSQL** for managing relational data.

Additional Tools: Ngix

- **Git and GitHub** for version control.

Current Progress

I have created user authentication with JSON Web Token (JWT) and have finished CRUD operations for threads and comments.

What's next?

- Category/Tagging system for thread organization.
- Search functionality for threads and comments.
- A lot of code refactoring and documentation.

Additional features I hope to implement:

- Upvote and downvote system.
- Moderation tools like reporting features.

Reflection

I have thoroughly enjoyed every step of this software development journey. While I had some prior experience in React, this project was my first in full-stack web development. I am thankful that I have managed to pick up some very crucial concepts, such as REST APIs to database management.

One area I need to focus on is increasing the frequency of documentation and code refactoring to enhance the codebase's quality.