MongoDB Editor and Quality Control

Developed by Trent Maas
Project Advisor: Dr. Giovani Abuaitah

Goals

The goal of this application was to allow any team to easily manage any MongoDB database they might have. The idea of this project comes from the time I spent at CO-OP and some ideas that were being passed around at the time of incorporating MongoDB. Spawned from this was an idea that would allow multiple people to start working with MongoDB Databases regardless of their knowledge of Mongo or databases as a whole. Alongside this was a desire to allow an admin to enact quality control by being able to accept and reject changes made to the database and have a detailed log of all changes made to the database.

Intellectual Merits

- This project leverages web development skills to provide intuitive editing of a MongoDB database without requiring the user to spend a considerable amount of time learning the software or MongoDB queries.
- Provides options for various modes of input using CKEditor for long text and simpler input methods for strings/integers and lists.
- Logging of edits made to database. Records added/removed data alongside other meta information such as the user who made the change and when the user made the change.

Broader Impacts

- My Application promotes collaboration through its design allowing multiple people on a team to easily work together to manage a shared database.
- Groups can easily leverage the benefits of MongoDB without first learning the MongoDB Query Language or designing their own application.
- The Application allows users to ease themselves into database management with quality control design aspects to create a safety net around the data that is being edited.
- Admins of the application can approve/reject changes to the Database to ensure the quality of the data being added to the database.

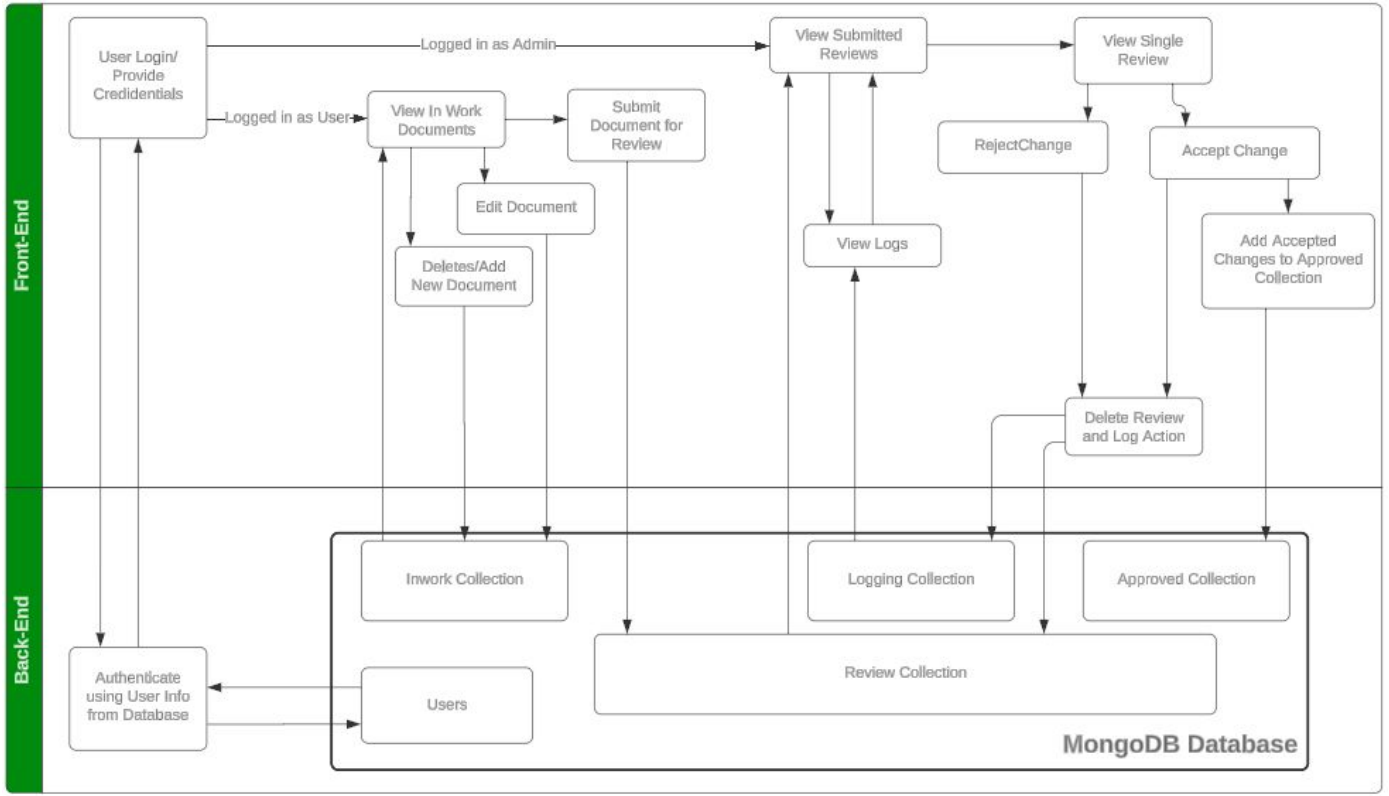
Technologies

The Technologies used in my product are:

- HTML, CSS, and Javascript for Building Front-End
- TinyMCE is used to provide a html editor entry option to the user
- Node.js was used to develop the Back-End.
- MongoDB is used to store users, and the data being editing on the application.

Design Specifications

- Front-End uses HTML, CSS, javascript
- Back-End uses Node.js
- Database is a MongoDB Database



Milestones

- October 22nd, 2023 - Foundation for both the Front-End and the Back-End was created.
- November 26th, 2023 - Connections to MongoDB made allowing for editing of database.
- February 16th, 2024 - Implemented Login Page through MongoDB
- March 14th, 2024 - Added Logging capabilities for when Edits are made to the database.
- April 8th, 2024 - Enhanced Login capabilities allowing more secure connections.
- April 12th, 2024 - Enhanced user features to allow submitting changes for review, added admin controls allowing for viewing of logs and accepting/rejecting submitted changes.

Results

Accomplished Feats:

- Implemented Back-End that can easily communicate with MongoDB and populate logs for quality control and easy manipulation of data.
- Made Intuitive Front-End that is User Friendly.
- Implemented Workflow allowing admins to accept and reject changes to the database.

Finishing Tasks:

- Ensure secure connections to application and database.
- Improve user experience further through quality of life updates.

Challenges

- Merging Front-end and back-end components. Persisting user data throughout the application proved difficult during development. This was accomplished by created a user Session that holds all of the pertinent data while a user was logged on. For awhile a large part of development this Session object was not being created correctly.
- User Authentication through MongoDB, allowing users to request accounts and storing their data on the MongoDB database.