Owen Mount

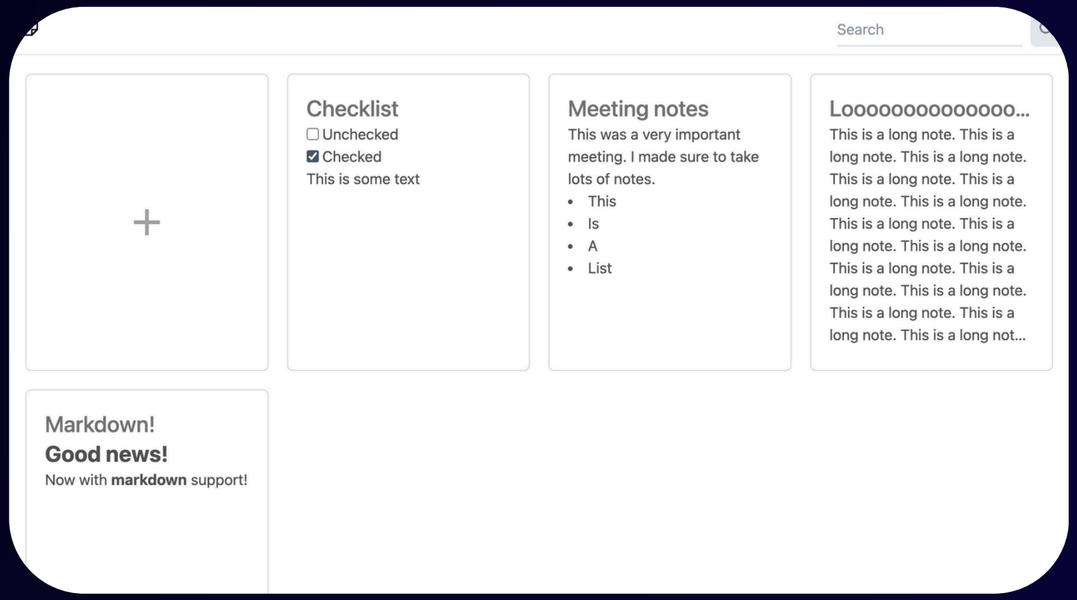
ONotes

CST-391 Milestone 6

ONotes

Application Overview

ONotes is a lightweight and flexible notes app that supports Markdown syntax. The application is entirely written in JavaScript, with the help of frameworks like ExpressJS for the back end, and React for the front end.



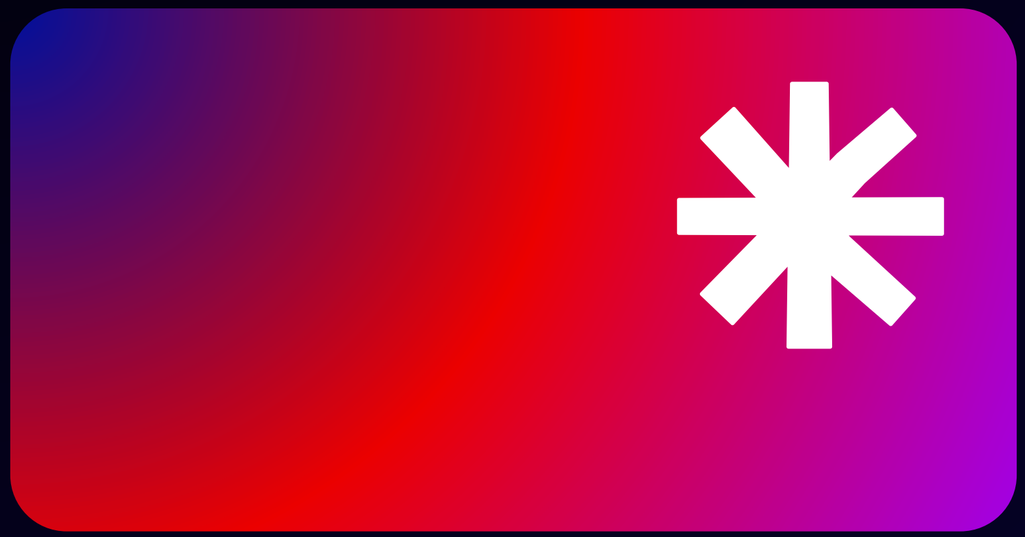
Simple, straightforward, lightweight.

Milestone 3: REST API

Milestone 3 was focused on developing the back-end API for the application using the ExpressJS framework. The api supported full Create, Read, Update, and Delete operations. Additionally, the database design and implementation was done in this milestone. As one of the deliverables for the project, I wrote full OpenAPI documentation for the project

REST API & Data Layer

Challenges



Learning a brand-new framework for this project came with some challenges. In addition to learning the Express framework, getting comfortable using the node ecosystem also proved to be difficult initially

Thankfully, I have used other web frameworks like Spring Boot and ASP.NET MVC before, so applying the overall concepts was easier

Issues

Status & Last Insert ID

Last Insert ID

Because I used a string type as my primary key for note objects, MySQL is unable to return the ID of the last inserted item. Without changing the design of the note object, I would like to add the functionality to do this.

HTTP Status

Currently, all controller methods either return 200 OK or 500 Internal Server Error status codes to communicate success/failure. I would like to add more robust HTTP statuses to my controller methods.

Design,
Design,
Design

Lessons &

Takeaways

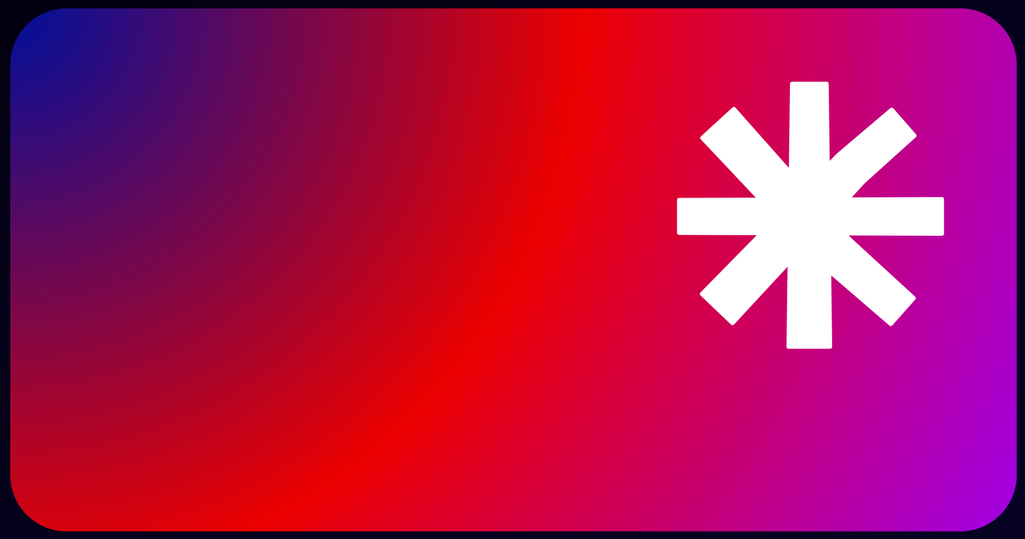
Completing this project further cemented the importance of design in my mind. Because I did all the design work beforehand, implementing it was only a matter of transferring my plans to code.

In this project, I designed the front-end of the application using Angular. I took the designs that I had created in previous milestones and implemented them using the Angular framework.

Milestone 4: Angular

Front end, using Angular

Challenges



Front-end development is not my favorite. I find myself getting caught up in the weeds of the small details, and this hinders me from running the project to completion. However, I found that focusing on design in the planning phase of the project helped me greatly in the execution. Additionally, using the tools that Bootstrap and Angular provide was extremely helpful in my design process.

Issues

Last Updated & Search

Last Updated

On the note detail page, there is a “Last updated” area in the header that shows the date the note was last updated. When the save button is clicked, this field should update. However, for some reason it is not updating every time the save button is clicked, even though the note is being saved.

Search Results

If a user searches for a note while already on the results page for a previous search, the results are not updating. However, the search works full on every other page of the app.

Design,
Design,
Design x2

Lessons & Takeaways

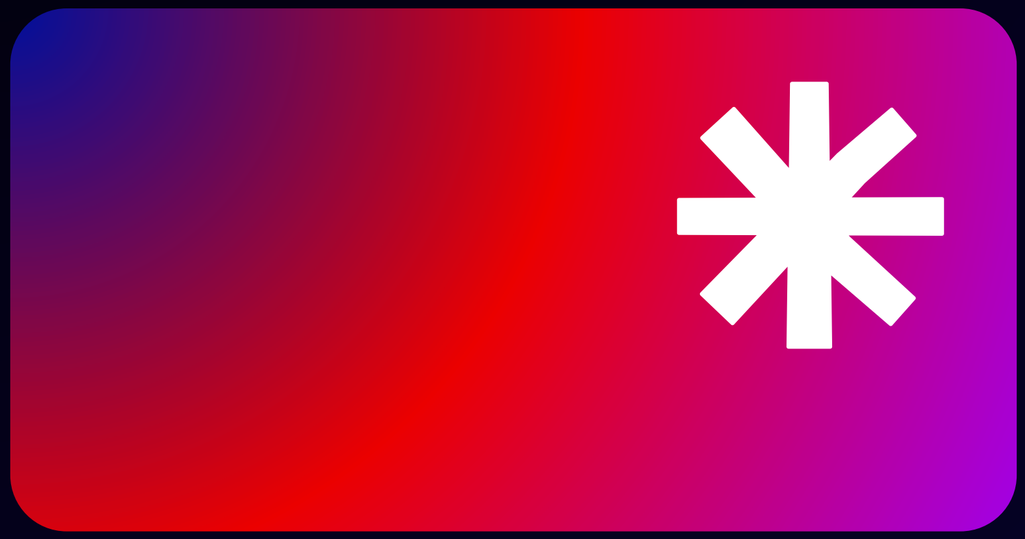
Similar to the last milestone, I learned the importance of design, especially when developing in the front end. It is important to go into development with strong design plans, because trying to figure out the design during development is very slow and labor-intensive.

In this project, I designed the front-end of the application using React. I took the designs that I had created in previous milestones, as well as the working version I had using Angular, and shifted them into React.

Milestone 5: React

Front end, now with React

Challenges



I chose to go with the TailwindCSS UI library for this project instead of Bootstrap. Tailwind is more utility-focused, as opposed to bootstrap which offers more pre-made components. Because of this, Tailwind has a steeper learning curve but also offers much more customization

Issues

Responsive design & Dark mode

Responsive design

In some areas of the application, particularly the single note display, the design comes apart at smaller display sizes. This can be fixed with additional tweaking of the Tailwind CSS classes

Dark mode

Tailwind provides utilities to support dark and light themes. Unfortunately, these utilities are not fully implemented throughout the app.

Features over Frameworks

Lessons & Takeaways

In this milestone, I learned to not let the tools dictate the design of my app, but to instead let the overall purpose of the app inform the tools I use. I was able to make an app that was nearly identical to the Angular/Bootstrap version, but quite different underneath.

Accessibility Considerations

From a Christian worldview, every person is made in the image of God and deserves dignity and inclusion. Website accessibility ensures that all users, including those with disabilities, can fully engage with digital content. This enhances user experience by removing barriers and showing love through thoughtful design. Best practices include using semantic HTML, providing alt text for images, ensuring sufficient color contrast, enabling keyboard navigation, and following standards and guidelines. Designing with accessibility in mind reflects compassion and a commitment to serving all people well.

Demo