BOOKSHELF FINAL PROJECT

CSCI 3916 H01 – Web Api

Team Bookworms

Iris Vucic, Sadiqah Al-Masyabi, Giselle Manzano

Recorded Presentation URL

https://olucdenver-my.sharepoint.com/:v:/g/personal/sadiqah_2_al-masyabi_ucdenver_edu/EUMHaiX9fQpJuOprZXetYrsBJ8sOR6VTZdfzfxWYEeNrsA

Problem Statement

- Lack in social platforms for book lovers
- Many people love to read books and share their thoughts and recommendations with others.
- However, finding a reliable and convenient way to manage their personal book collections, track their reading progress, and connect with other book enthusiasts can be challenging.
- Therefore, there is a need for a bookshelf app social network that addresses these issues and provides a comprehensive and enjoyable experience for book lovers worldwide.

Solution Statement

Our solution is an image sharing site that connects book lovers through the books they have on their shelves. Our image site offers:

- A unique platform dedicated to books
- A community of book geeks, worms, and lovers who share their reading experiences and recommendations through images and captions
- A way to discover new books and genres

Join us today and start sharing your bookshelf with the world. What books are you reading right now?

Biggest Challenge

There were three main challenges we encountered in this project. We couldn't decide on which one.

1 - Code Made Separately Must Work Together

Before we even began this project, all of us has experienced this difficulty with previous projects. Something can work perfectly alone, but crash when combined with another piece of code. Because of this, we set aside a week to combing and work out any kinks created when we combined our code. Even with that, it was still a struggle to get everything working together and smoothly. Small things like variable names, naming conventions, and capitalizations were all things we had difficulty keeping consistent.

Biggest Challenge

2 - Working with Three Different Layers – Api, Web, DB

We also struggled with separating jobs between the API, the web app, and the database. Before taking Web API, we did not have any experience with MongoDB, JavaScript, and layered code. Understanding the jobs that make up each part was vital in us getting this project right. With any one small implementation we wanted to add, we found that we had to think very carefully and talk out each decision we made. We had to decide how that would work split up between the site, the API, and the database. This was a difficulty, but also something that we appreciated, because it made us step back and really think about the decisions we were making.

Biggest Challenge

3 - Working with Three Different Layers – Api, Web, DB - Debugging

We found that debugging across three different layers is extremely difficult. We struggled with finding where an error originated and would often spend hours trying to find why the web app wouldn't work. Both the Back-End team and the Front-End team had to work together to find out what was going wrong. Errors were vague and we had to go off instinct and/or asking Shawn for help.

- Posts not showing in our app was originally thought to be a back-end issue, then a front-end issue, and finally realized it was back-end. After hours of debugging and rereading and discussing code, we found it was a mix-up of error codes and resolved it.

Biggest Highlight

Combining all three stacks together

Our biggest highlights are

- Working together as team
- Worked long hours
- We learned so much from our mistakes
- Communicated with each other
- At the end, we succeeded as a team!

How Bookshelf was Built

- **Frontend -**
  - React
    - Used to build the user interface components for the website
  - Material-UI
    - Open-source library of React components where we sourced our icons for our header and post structures
  - Deployed with Firebase
    - Platform for hosting applications and storing the images that users upload

How Bookshelf was Built (ctd.)

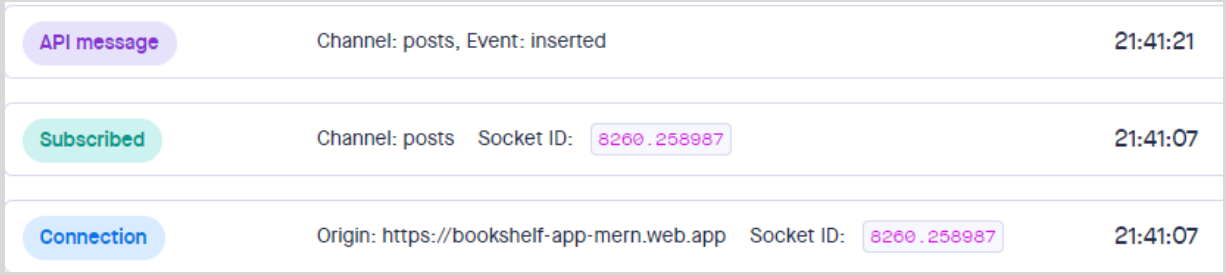
- **Backend -**
  - Node.js
    - Handles the backbone of the application, including connecting to the database, defining the "Post" schema,
  - Deployed with Render
    - The frontend (Firebase) connects to this /sync endpoint to display the posts on the website

How Bookshelf was Built (ctd.)

- Database -
  - MongoDB
    - Stores all info about posts, including the ID, caption, user, and image link (photos are stored on Firebase)
  - Pusher
    - Provides channels to integrate realtime features in the app when users upload posts

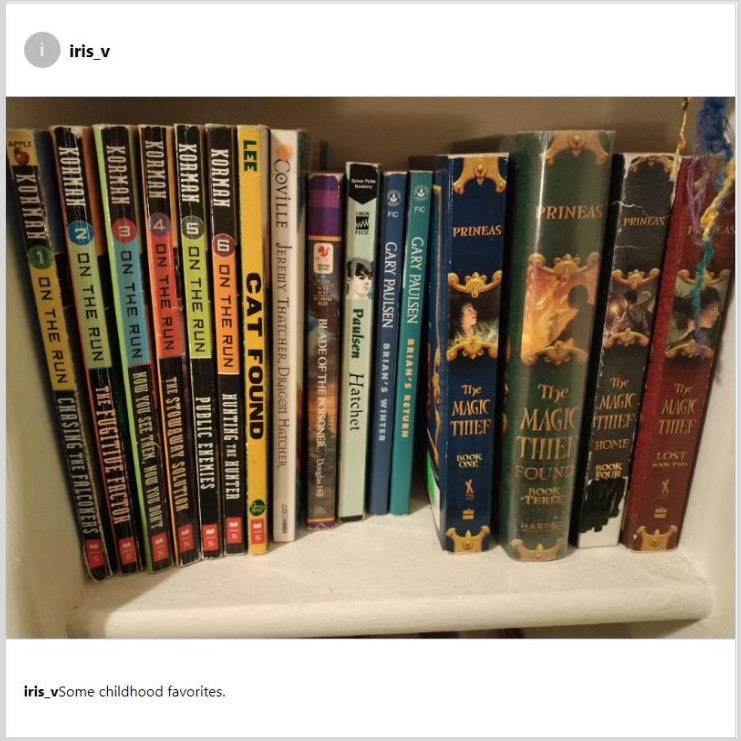

```
_id: ObjectId('6452bc1fe3c7fa395d8a42d0')
caption: "Some books and Lego sets! :)"
user: "iris_v"
image: "https://firebasestorage.googleapis.com/v0/b/bookshelf-app-mern.appspot…"
__v: 0

_id: ObjectId('6452bd29e3c7fa395d8a42d3')
caption: "Some childhood favorites."
user: "iris_v"
image: "https://firebasestorage.googleapis.com/v0/b/bookshelf-app-mern.appspot…"
__v: 0
```

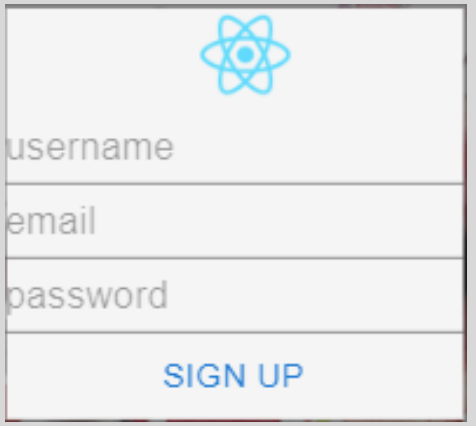



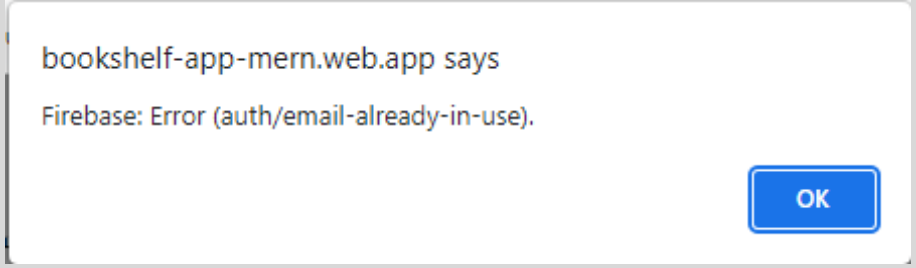
How Bookshelf was Built (ctd.)

Post on the Site



Sign-Up Screen (+ error example)





Future Development

In the future we would like to...

- Implement personalization features
- Implement it so that new posts load on the top of the page
- Add ways our users can filter and view the photo galleries
- Implement a bookshelf of the week where we randomly select a bookshelf and highlight it on our main page. This will encourage people to check out the highlighted bookshelf and try out new books.

Bookshelf GitHub Repos

- Frontend: https://github.com/irisvu/Bookshelf.git
- Backend: https://github.com/irisvu/Bookshelf-Backend.git

ONTO THE DEMO

Thank you!