Gregory Robertson and Seth Milojevic

Dr. Xiao

Internet Systems Programming

4 December 2021

Final Report

Implementation Details

We structured the back-end of the server using MariaDB, Node.js with express.js, and bcrypt for security. For the front-end we used Gatsby, a flavor of React. For the actual design decisions of the website we decided to use an approach that separated all large elements from eachother using abstraction. Our server was the step between our static webpages and our API knowing nothing of either of the implementations of each of them, only knowing to serve webpages and send requests to the API. This type of linkage was used to try to inhibit the need to think about how these back-end functionalities would be implemented when working on the front-end. Our API is the connection between our server and our database and serves the purpose of preventing needing to think about the database connections when working on the functionality of getting requests or serving responses. The only piece of our software that was made to understand all of the functionalities of the design was our build process, however there is not a very large build step as most things once initialized are able to function standalone.

Contributions of Group Members

Seth worked on the front-end of the website largely focusing on how the API connections would integrate with the functions that the website needed to perform for users. He also focused on the aesthetic design of the website and many of the design decisions regarding UI and UX.

Greg developed the back end of the website developing the necessary API calls for the front-end to retrieve and deliver data, along with connecting expressjs and gatsby so that the gatsby generated pages

can be properly served to the user through express. Greg also designed and implemented the database for all of the user data tracked by the website.

Issues Encountered

- Getting React to work around scopes where the element is not yet rendered.
- Properly parsing data retrieved from the API for display on our pages
- Allowing an article to be marked as read for the user and properly displaying that they had already read the article on refresh
- Recognizing that a user had already logged in on the account page

Possible Future Work

- Implement better recognition of if user is logged in on account page
- Properly implement react elements rather than a mix of regular JS and react
- Track more stats on articles for users to see

Screenshots

404

Sorry! This page doesn't exist. :(

If you tried to find an article for this day, there aren't any (yet)!

Try selecting a different day using the calendar at the top of the page. To open the calendar, click the date at the top of the page.

If you've already logged-in once this session, just press "Submit" with both fields empty.

Login

Username:

Password:

Submit

Welcome, Seth! Articles You've Read:

Date Article Clicked:	Date Article Posted:
12/05/2021	08/20/2021/1
12/05/2021	08/20/2021/0
12/05/2021	11/09/2021/0
12/05/2021	12/01/2021/0
12/05/2021	12/01/2021/1