Andrew Jensen

CS 4V95

5/1/21

Data and Application Security – Project Report

* Objectives of the project

The goal of this project would to create a reusable user system demonstrating access control, session layer usage, token validation, and private data hashing in order to provide a viable forum application foundation. I would hope to make access control into discrete package(s) in the process to practice Separation of Concerns.

* High level design (what you designed)

A person can create a page as an admin, or join a forum as the lowest rank on that forum. Each page has multiple ranks as well as feeds that users can contribute to. An admin can create ranks, assign permissions to certain ranks, and promote/demote users to specific ranks. An admin can admit/kick users and choose whether the forum is public or private.

If public the forum is open for users to join whereas if it is private, users would need to apply to join a forum. A forum can consist of multiple pages that limit access based of a users rank. Any page a user has access to, said user can post messages on it.

* **Implementation design**

Proposed

RBAC will be used to maintain proper access control.

The project will generally be a three tier architecture.

Presentation Tier – While it would be best to have a full web front with a , this would be very time intesive without adding much to the security demonstration over a terminal application. IRC can be used to display feeds.

Application Tier – The Application Tier will be an HTTP server with a series of semantic routes for authentication, forums, pages, and post submissions all of which will be controlled via the RBAC implementation.

Data Tier – The data layer will be based upon a MySQL database containing a schema with general and multipurpose tables for users, forums, pages, forum-memberships, and forum-ranks.

Implemented

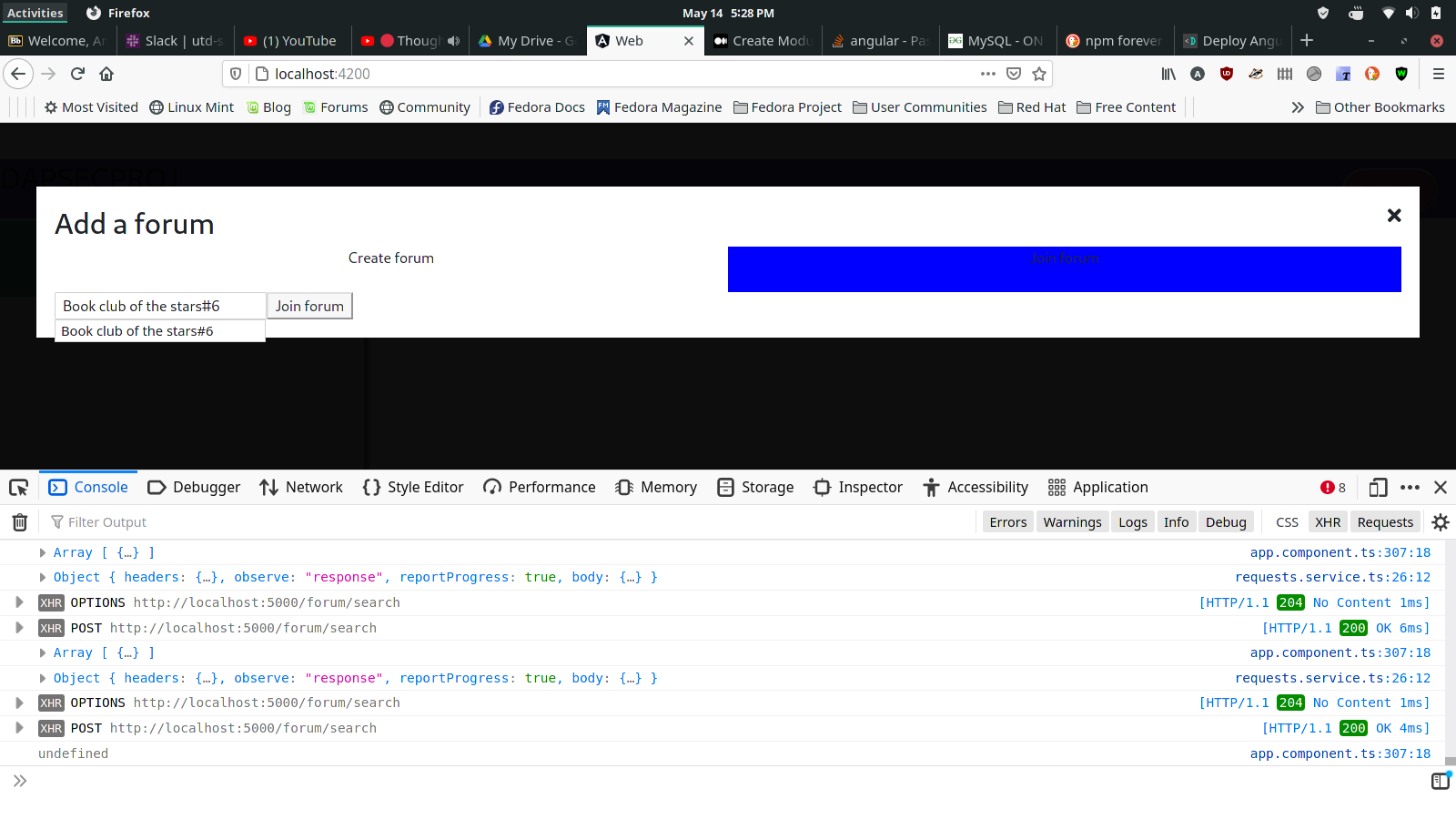
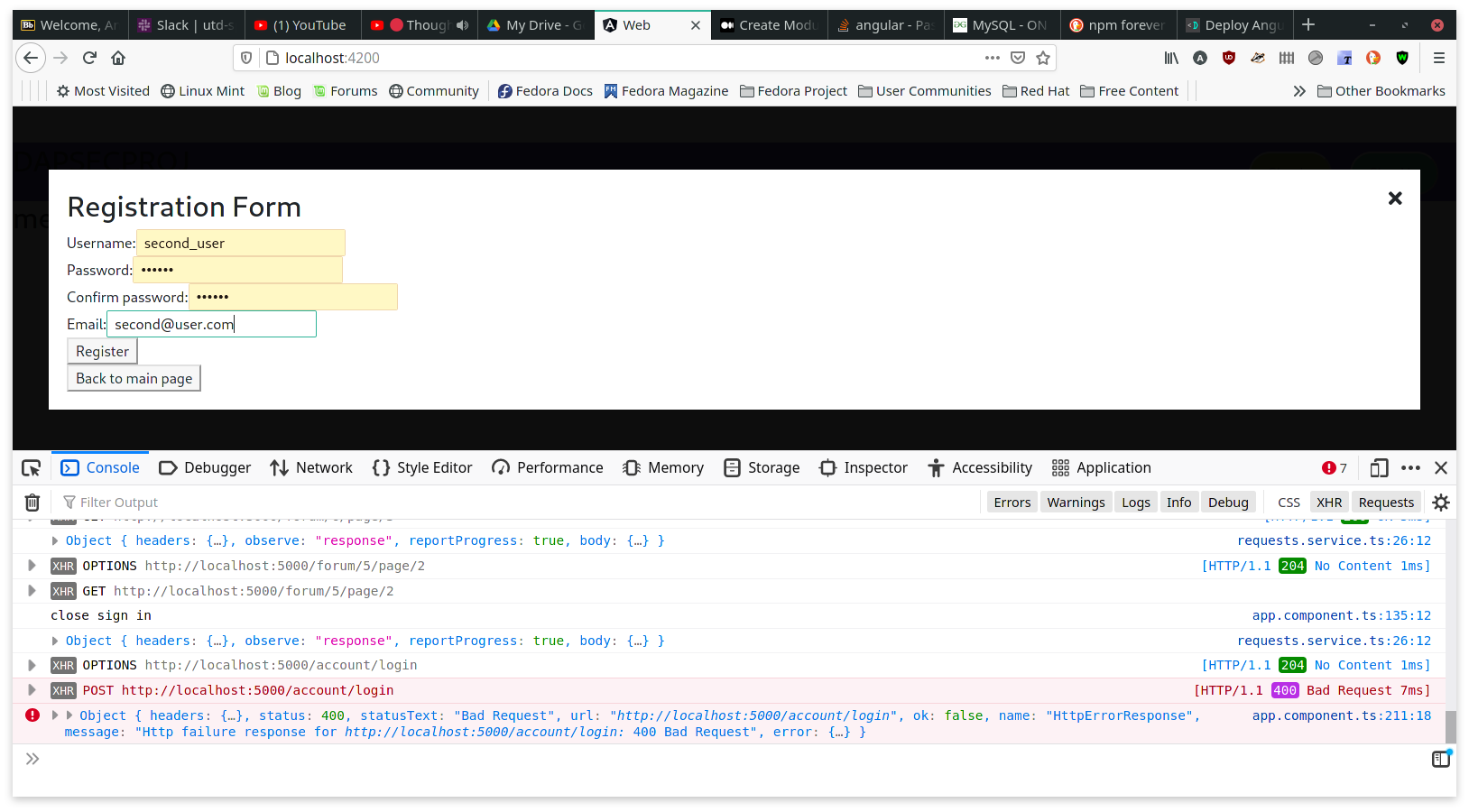
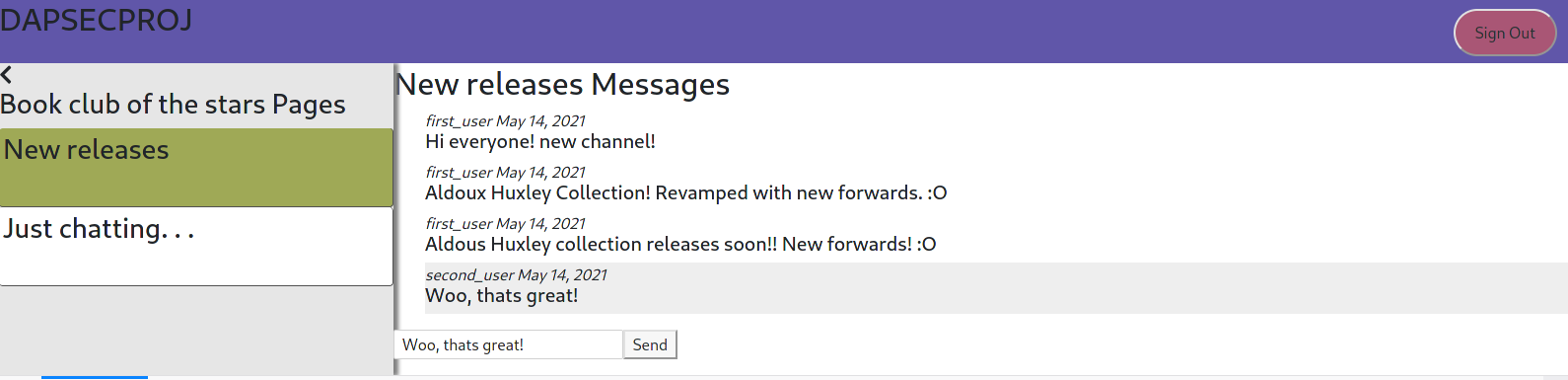
I wrote a full MyEAN Stack to build this application and to develop access control hands-on in a tangible context. MyEAN refers to Mysql, Express.js, Angular, and Node which handles all the needs for a small distributed web application with a proper DBMS for practice.

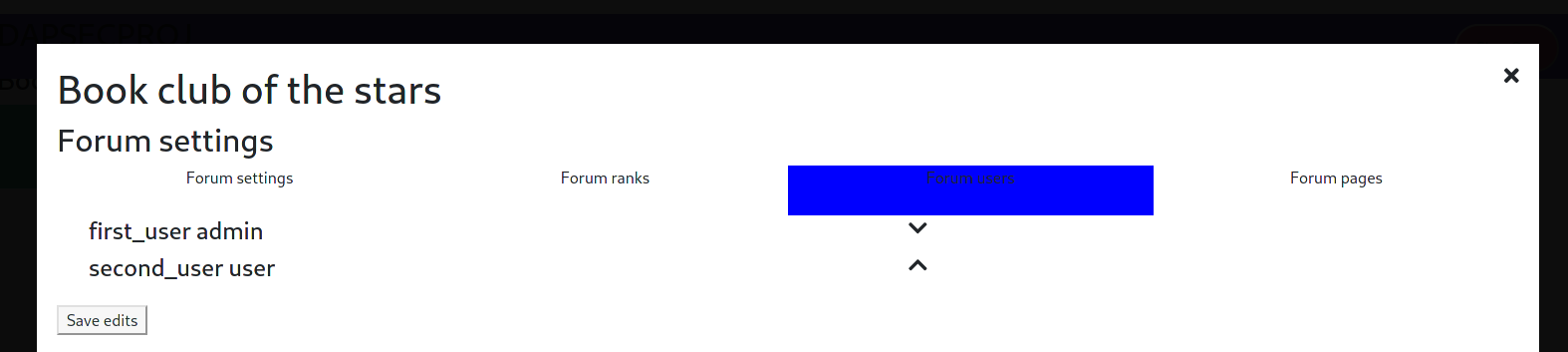
**Presentation Tier –** I wrote the frontend in Angular with reusable transmission and token storage services. The user can login or register on the right of the navbar. Once logged in, those options will no longer be available and will be replaced with a signout option that deletes your token and tells the server. Once loggedin, the user can view their forums in the left aside and click on it to view the forum pages. If a page card is clicked, the page messages are loaded. If the user is an admin, they can see a cog on the forum card which will allow them to interact with the forum settings. The forum settings allow the user to change the forums name and visibility, its ranks, user’s ranks/memberships, and it’s pages viewership/authorship.

**Application Tier –** I utilized Express for the middleware as it provides a clean interface to build HTTP api routes. Generally, all functionality is found in the `forum.ts` file as most data pertains to what happens in a forum. All the features mentioned in the presentation section are enabled through the application layer which submits prepared queries in a mysql connection pool. JSON Web Tokens (JWTs) are used to verify data was not modified during transmission and a user creds were contained inside it to determine access. Private forums do not appear in forum searches when joining forum. I was also able to pull out authentication and authorization into reusable functions.

**Data Tier –** I wrote a simple six table sql schema around the abstractions: user, forum, rank, membership, page, message. Membership is a join table between user, forum, and rank. Forum has many pages and each page has many messages.

* Output



* Challenges (e.g., the difficulties you encountered and why you think you had these challenges)

Managing the frontend was odd but having it to give user flow was pretty useful along the way with designing the middleware. There were several requests that had to be established but most of the attempts I made to containerize and generalize promise-based, asynchronous request listening were difficult and too specific. Thus I prepared a simplified solution. Forms were very time intensive. I was mixed on how I would relay user, rank, and forum data together.

* How can you improve the project in the future

In the future, I would hope to finalize the forum-user-settings page to modify user ranks. I would like to use a more defined tool for session management however the token was a good session manager as it would expire after a specified time. I would like to implement a data management system that performed constraint checking and query modification. I would like to take this frontend and see if I can modularize forms a bit more and have different input validators that are used depending on dev or production build methods. I don’t feel the need to deploy it since it would be costly, redundant, and useless unless testing under scale.