

I began working on this assignment with zero background in full stack development. I have never used the Go language before, let alone used it to implement an entire RESTful API. My experience with React was also minimal, with most of my web dev experience revolving about only HTML and CSS implementations. At the beginning, I had my reservations about whether I could complete the assignment and meet the basic criteria in time. After all, I only had 2 months after finals ended to learn the bulk of full stack from scratch and implement it correctly. Thankfully, 1.5 months in, I have managed to put together a functional forum page supporting CRUD operations. This is my first-ever full stack project. Its functionality may be nothing compared to what veteran developers have made and there are of course plenty of room for improvement, but it is a major personal milestone. I have taken a step forward in improving myself, and I am proud of what I have implemented.

The learning journey has been highly fulfilling but also frustrating. The main reason is the steep learning curve within the short timeframe. I had to devote myself to studying online resources starting right after finals. Thankfully, my C background allowed me to quickly adapt to the Golang syntax, and there are plenty of online tutorials teaching the basics of React. Nonetheless, a major obstacle faced by many rookies is the never-ending source of errors. Throughout my development process, I have run into countless errors, most arising because I did not understand the correct way to implement something. Such errors can be as trivial as trying to access React state data directly after SetState to slightly more complex ones like failing to send cookie from server to client due to not configuring the server and client to include and use credentials respectively. Such errors are frustrating as I can take up to hours trying to debug, frantically surfing stackoverflow to figure out the issue. I have kept a log of all the errors I have encountered in my learning journey. Looking back at them, some errors were truly foolish, and I wonder how I even made such mistakes, but then again on hindsight everything seems easy and logical. Despite all the frustration, the reward is fulfilling. I learn best through making mistakes, instead of listening to someone teach me the right way to do something. What is important is that I do not make the same mistakes again in the future. Through all my mistakes, I have gained a deeper understanding of the underlying mechanisms in server-client communication and the workings of front and backend. I can now appreciate how a client browser sends requests to the server, how the server multiplexer redirect requests to handlers, how handlers interact with the database to process requests and send responses, and how the client receives the responses and extract data from them.

Being completely new to full stack, it is inevitable that I had to reference online tutorials and projects on Github to learn about correct practises to implement certain functionalities. However, there is no wholesale plagiarism of past works by others. The approach I have adopted is to make my forum project a melting pot of practises and approaches by different people. For one functionality, I may agree with someone's idea for implementing it, whereas for another functionality, I find someone else's method more effective for my context. Most of the time, I choose to blend various approaches, siphoning the best aspect of each approach to combine with my own input. For one, in my handlers I adopted the classic approach of using `sql*.db exec()` functions from the `sql` package to interact with the database, but I also added a slightly

uncommon step to check that rows in the database have indeed been modified for double confirmation.

There are also areas for improvements for my forum. If I had more time, or more expertise in the area, there are some functionalities that I wish to implement or change. Firstly, I would like the search bar to be able to find threads based on just keywords instead of needing to input the full thread title word by word, as most forum users would be searching by keyword to find the thread most relevant to their interests. This would require a different approach to the selection from my database, and I may even need to restructure how I store my threads data. Second, I wish to make it such that editing of threads and comments can be done directly on the same page instead of redirecting to a separate page. My current approach would make editing more inconvenient and breaks the flow of the page. My intuition for implementing this would be to insert a form element below comments/threads which can be edited. This form element shows nothing by default but when a user clicks on “edit comment/thread” button, an input space is dynamically generated, allowing the user to input the updated content. Finally, I would like to allow users to upload custom profile pictures and have their avatars displayed beside their usernames for threads and posts. Currently this is done using a standardised emoji which can be rather unappealing. This will require me to investigate the storage and uploading of custom image files, which is an area I lack experience in currently. In sum, there is a lot more for me to learn about web dev and full stack. When I embark on my next project, I hope to achieve more than what I have done for this forum and attain greater heights.

User Manual:

You can access the forum with or without a registered account. To register an account, you need to create a unique username and password which will be used for user verification. If you are not logged into a registered account, you can view all the posted threads and comments. You will also be able to register a new account, login to an existing account, and search for threads by their title using the search bar. However, you are not authorised to modify any information.

If you have logged in to a registered account, you will be authenticated and redirected to the authenticated page, where you can post threads and comments in addition to reading them. You can also update and delete threads and comments but limited to only the ones you posted. You may even edit user account information and delete your account altogether. For editing of user information, for extra security, you will need to verify your identity using your password once again. There will also be double confirmation needed for editing user info. After you are done with using the forum, do remember to logout to protect your account from misuse!

For usage on mobile phones and tablets, please ensure that the browser you use supports cross-site tracking for the jwt cookie to be sent successfully from the server. This can usually be enabled in the settings of the browser app. Landscape orientation is recommended when using mobile phone for optimal experience.

*Please expect slow responses and server lags from time to time as the forum is deployed for free on Render (on first load, multiple refreshes and/or clicks on the register or login button and navigation back to homepage may be needed)