# Report

## General Architecture

**1.1.1 Frontend: Vue + Vite**

I’ve implemented this assignment by using the Vue framework. At the beginning, I used Vite to generate a template of my project, which included a demo of Vue, then I implemented my own UI using this template.

It has two pages, which are implemented by Home.vue and Login.vue.

I alse made 2 shared components. The first one is SInput.vue, it is a input with lables. The other one is LinkCard.vue, it covers the UI design of each link post and some function such as liking, disliking and hiding.

About routing, I’ve learned Vue-Router from its documentation so that pages can be switched between the login page and the home(links) page.

**1.1.2 Vanilla vs Vue**

When using Vanilla, I can touch DOM directly and so I write code based on intuition. For example if I want to add multiple DOM elements to a parent node, I have to get the parent node DOM and add new elements by calling “appendChildt”.

While in Vue, the framework itself handles DOM logic already. Then what I should do is preparing good data for it. I can write less code to implement the same updating UI logic in Vue.

**1.2 Backend: Deno + Sqlite + deno-oak**

I chosed Sqlite as the database, because I feel its more easy to use. And when I was studying the documention of deno, I found oak(https://docs.deno.com/runtime/fundamentals/web\_dev/#oak), the demo looked good so I decided to used it to run my backend server.

Also, the backend is devided into three layers, which are “Model”, “Controller” and “Server”.

- The “Model” layer directly touches database, using Sqlite.

- The “Controller” layer handles business logic, such as verifying user login and processing data for frontend.It relies on “Model”.

- The “Server” layer is responsible for setting routes for frontend APIs and deals with frontend requests by calling method in “Controller”.

- URLs

'/link/all'

'/link/favorite/:userId

'/link/add'

'/link/update-like'

'/link/update-show'

'/member/query'

'/member/login'

'/member/register'

### Completed Features

**2.1 Login & Register**

- Users can login with existing accounts, which are stored in database.

- Users can register a new account, but they can’t register a user name which has been used already. If registration succeeded, they shall see a browser alert window and then it will auto login to the home page.(Note that every time backend server restart, the database will reset itself, that means newly registered account will be removed, users have to register again.

)

- Passwords are hashed by bcrypt(file: backend/utils/index.js);

**2.2 Home**

- Display current user’s name and user’s poitns on the header.

- Users can click logout on the header to get back to login page.

- Display links by three menus: All links, My links , Favorites. Users can switch among them by clicking menu on the header.

- Each link is displayed in a row.From left to right, each column represents “Rating”, “Title”, “Desc”, ”Link Owner(Owner points) / Post time”, “Actions(like, dislike, undo like and on)”.

- Users can like, dislike ,undo like, undo dislike a link in any of the three menu. Whenever the “like” status changes, the rating of all links will be recalculated using the Bayesian Average Ratings. Also user’s point shall change accordingly.

- Users can hide their own link in “My link” menu. Hiding a link won’t affect its rating or user’s points but this link won’t display in the “All link” menu until the user switch to “Show” it.

- “Favorite” menu only displays links that are liked by the login user.

- Users can sort columns. Only column “Rating” and column “Post time” allow sorting.By sorting it , users can lick the top of these column. Hovering on top of these columns show a title indicating sorting too. Sorting is implemented in the backend, together with the sorting params from frontend .

- In case some descriptions of links are too long to display. I use css to limit that only three rows of text can be displayed. If texts exceeds that limit ,there will be ellipsis. When user hover on desc. Column, it shows all the texts of desc. (I used the “title” attribute of HTML element to implement that)

- Users can post new links by Clicking “Post Link” on the header.

- About rating, as I have already covered above, I use the Bayesian Average Ratings. The rating ranges from 100 to 0. Another thing to note, if a link is not rated by any user, the rating of it remians 0.

help you received

### Received help

**3.1 Vue and Vue-Router usage**

I mainly learn from their cook books, and watched some video tutorials.

<https://cn.vuejs.org/guide/quick-start.html>

<https://router.vuejs.org/zh/guide/>

https://www.bilibili.com/video/BV1Zy4y1K7SH?spm\_id\_from=333.788.videopod.episodes&vd\_source=ceb0a2503fb1fddafe38727b5b20cb37&p=136

**3.2 deno-sqlite and deno-oak**

All in the deno website.

<https://deno.land/x/sqlite@v3.9.1>

<https://deno.land/x/oak@v17.1.3>

**3.3 Async/wait**

My codes look cleaner using async/await rather than promise.then

<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/await>