**KEEP MY BOOKMARKS**

Capstone project

# Introduction

End users will often access online web content via one of the many browsers that are freely available to a variety of devices.

Locating and consuming the required content will often take time. Bookmarks and Favorites are a way of saving links to content. This allows an end user to quickly locate the same content if it needs to be revisited.

Over time, Bookmark and Favorite lists can become bloated. Scrolling lists, inappropriately named links, multiple browsers, multiple devices and deep tree structures containing the links can all contribute to making it harder to find the required content. End users may also begin to save the links in text files, send emails to themselves with the links or paste the links into word documents and excel files to try to gain some sort of management over the content.

Keep my bookmarks is an application that can be used to save links and user defined metadata pertinent to the link into an online database. The content is then accessible from multiple devices.

Making the metadata searchable means the links are easy to locate and including the ability to add comments allows for easier recall as to why the link was retained.

Various applications with similar functionality, both free and subscription, are available.

# Product Description

## Architecture Diagram



TCP/IP 80

TCP/IP 80

Port 3000 frontend

Port 5000 backend

database

Port 5432

servers

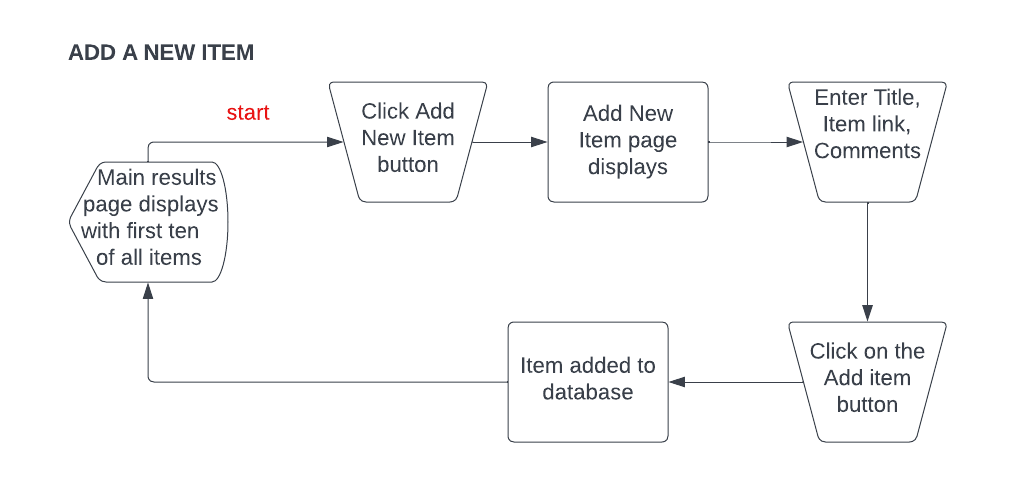
## USER STORIES

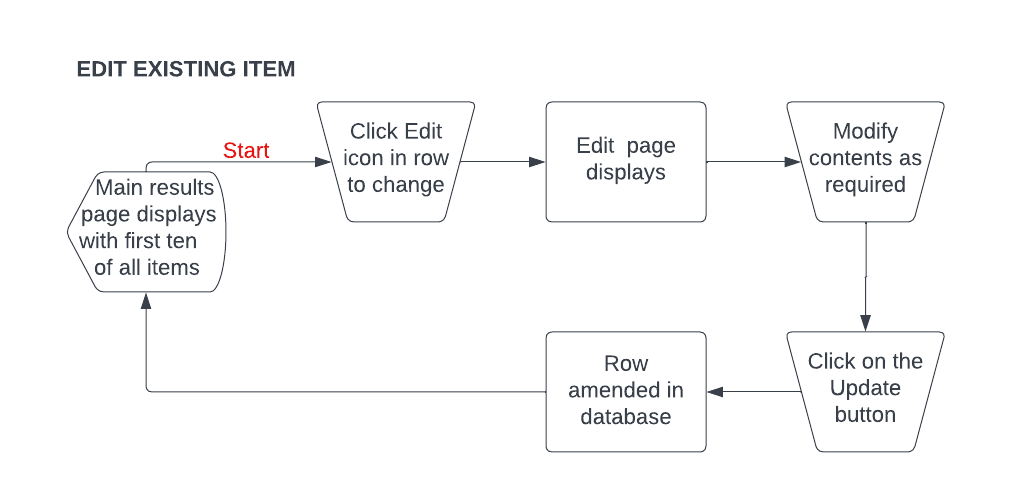
|  |  |  |  |
| --- | --- | --- | --- |
| Tittle | Description | Priority | Notes |
| Locating content | As an end user, I want an easy way to search my bookmarks/links to find relevant content | 2 |  |
| Availability | As an end user, I want the same content to be accessible across multiple devices. | 2 |  |
| Interface | As an end user, I want the interface to be intuitive and have a similar feel across multiple devices | 3 |  |
| Content | As an end user, I want the ability to add my own comments as a prompt to remember why I kept the information | 1 |  |
| Changing content | As an end user, I want the ability to edit any content I previously added | 1 |  |
| Housekeeping | As an end user, I want the ability to delete entries that I no longer require | 3 |  |

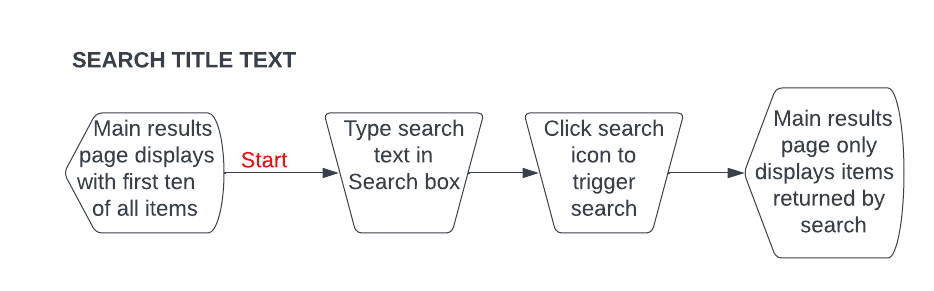
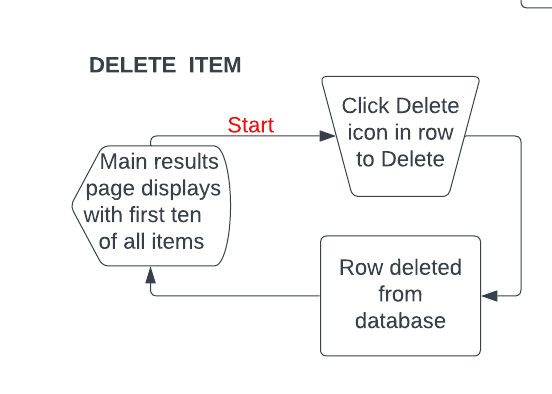
## User Flow

See the following link for an interactive user flow:

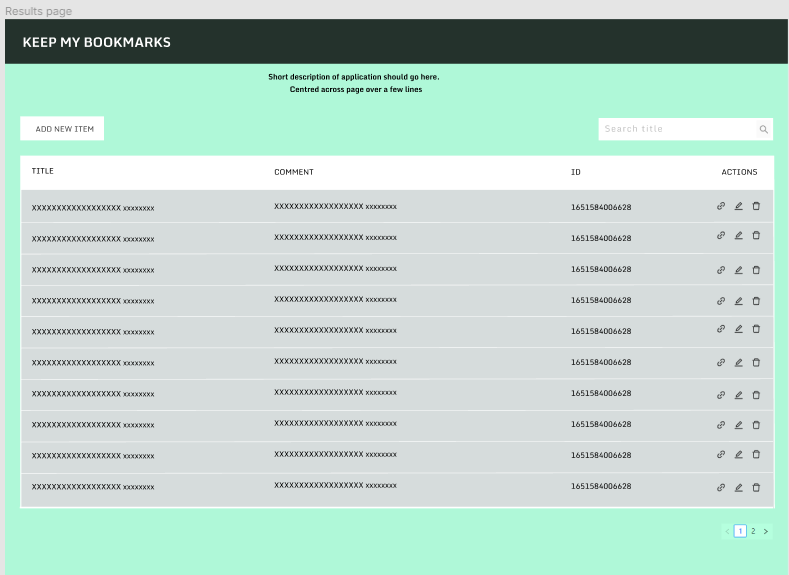
[Keep Bookmarks – Figma](https://www.figma.com/file/bVl4fetJtpHBaj2VJ2etB8/Keep-Bookmarks?node-id=0%3A1)



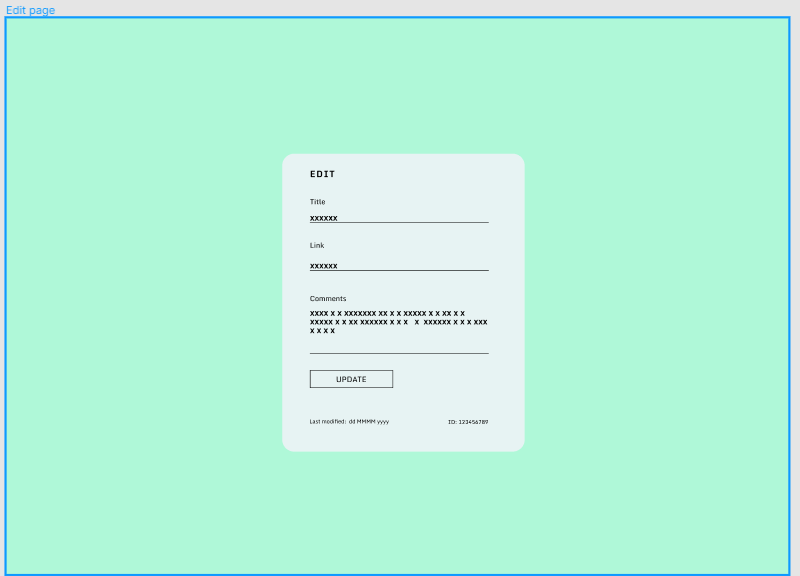




## Wireframe Design







## Scope

**In scope**

* Retrieving all items and navigate multiple pages
* Search by title
* Add new items
* Edit existing items
* Delete items
* Operational on multiple devices

**Open Questions/Out of Scope**

The following items are considered out of scope:

* Private content. The proof of concept application allows public access to all content.
* Automatic database creation. The proof of concept application is using a single database.
* Categories. Although use of categories was considered, assigning a category and using them as search criteria is currently out of scope.
* User customization. Currently there are no user customizations.

## Non-functional Requirements

**Security requirements**

The front and backend are hosted on AWS EC2.

The postgresql database is hosted by ElephantSQL.

The database name, password and username are stored in the backend and are added after the server is deployed and installed.

No personal details are stored. Any information added to the database is public. If links are added to access private or password-protected pages/sites, the user-specific security information will need to be entered by the user accessing the link.

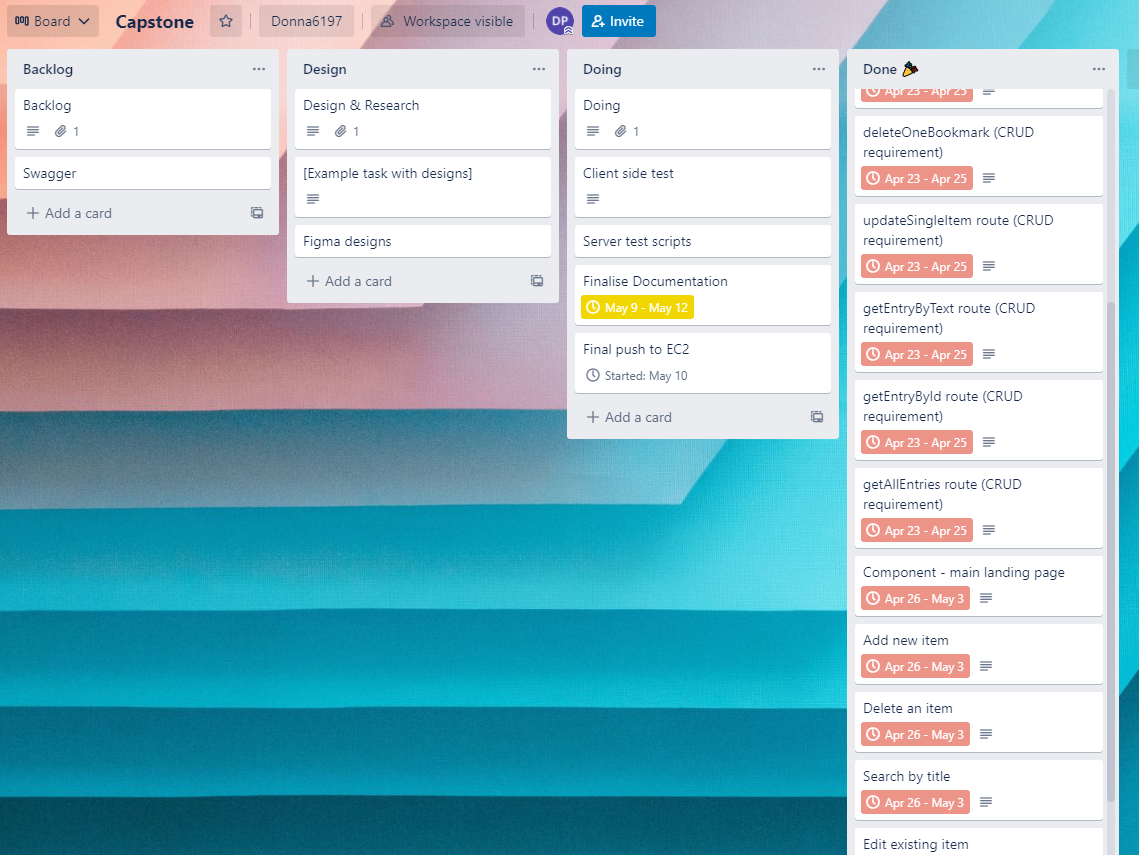
**Reliability**

The application response times depend on the reliability of the user connection.

# Project Planning

## TRELLO BOARD

Available at the following link: [Capstone | Trello](https://trello.com/b/L1tIcezq/capstone)



# Testing Strategy

Product quality was achieved by ensuring continual testing and development.

At each test, bugs were recorded. These then went back to the development stage to be fixed.

The following test scripts were run to check client side functionality:



Postman was used to test during api creation.

Swagger page is available to test GET connection [Swagger UI](http://localhost:5000/api-docs/) (or http://[insertpublicIP]:5000/api-docs)

# Implementation

To allow 24x7 access for any device, the software is deployed to AWS EC2 free tier. A user installing the same software has the option to choose their own hosting provider. IP is stored in a single configuration location detailed in the Read Me.

So as to not incur costs and allow users to not share their links, each user needs to set up their own database. Database configuration needs to be added post install. Instructions form part of the Read Me.

## EC2 INSTALLATION

Create an AWS EC2 instance.

Edit the inbound security group rules for the instance by opening ports 3000 (front end), 5000 (back end), 5432 (postgresql database).

Start the instance and connect to it. Make note of the public IP.

If Node.js isn’t installed, install it.  
<https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html>

Git clone repo from [donna6197/keepbookmarks: Capstone project. An application for storing, updating, deleting bookmark links. React and Expressjs. (github.com)](https://github.com/donna6197/keepbookmarks)

cd to keepbookmarks/frontend and install – npm i

Use Vim to edit keepbookmarks\frontend\src\component\env.js  
> cd to keepbookmarks\frontend\src\component\  
> type vi env.js and press enter  
> press I to switch to insert mode  
> change the IP to the public IP recorded earlier  
> press esc press :wq enter

Install the backend > cd keepbookmarks\backend – npm i

Use Vim to edit keepbookmarks\backend\db.config.js   
> cd to keepbookmarks\backend and type vi db.config.js and press enter  
> press I to switch to insert mode  
> modify file as appropriate  
> press esc press :wq enter

cd to keepbookmarks\backend and use nohup to start it  
nohup npm start > Output.out 2> Error.err < /dev/null &

cd to keepbookmarks\front and use nohup to start it  
nohup npm start > Output.out 2> Error.err < /dev/null &

# End-to-end solution

The software met all requirements defined in both the project brief and user requirements.

# References

Code and documentation relevant to this this project is available from the following github repository:

<https://github.com/donna6197/keepbookmarks>

The following resources were used:

React

Expressjs

Ant Design

Postgres

Nodemon

pgAdmin

body-parser

cors

swagger

supertest

node

Database is hosted by <https://www.elephantsql.com/>

Front and backend are hosted on AWS EC2