

# Bookmark App Technical Design

## App

<https://libbydavis.github.io/>

## Tech Stack

Frontend | Angular

Styling | Bootstrap

Version Control | Github

Hosting | Github Pages

Extra packages | Eslint, Angular Bootstrap (Floating UI and PopperJS included as dependencies)

## Data

The app manages Bookmark objects which are structured like so:

```
id: number  
name: string  
url: string
```

## Creation

The Bookmarks are created with input from the user in the Bookmark Form component. The Bookmark is added to local storage and the bookmarksList variable which is managed in the Overview Page.

## Modification

The user can select to edit a Bookmark in the Bookmark List component. The Bookmark is then saved to a variable in the Overview Page and sent down to the Bookmark Form. When the user has finished making their changes they can save and the Bookmark will be changed in local storage and in the bookmarksList variable stored in the Overview Page.

## Deletion

The user can select to delete a Bookmark in the Bookmark List component. They will get a popup to confirm they want to delete and if so the Bookmark will be removed from local storage and the bookmarksList variable stored in the Overview Page.

## Validation

The Bookmark is validated in the Bookmark Form component using HTML validation, the custom url-validator directive and the onSubmit function. Most of the validation is done while the user is entering input and the submit button is disabled until the form is valid (excluding checking if the URL exists).

***Id***

This is required and must be unique but it is automatically created based on the id of the previous Bookmark so doesn't need to be validated.

***Name***

This is required. It must be at least one character and if not, a message is displayed automatically next to the input box.

***Url***

This is required, must be a valid URL format and must exist as a fetchable URL. If it is not at least one character and in the correct format then a message is displayed automatically next to the input box. The format is checked by trying to use the input string as a URL object. I considered using a regular expression but thought this could get complicated and creating a URL object should fail if it is not a valid format. The existence of the URL is checked when the user submits the form to avoid too many fetch calls (e.g. if the check was done onChange while the user was typing). A message is displayed next to the input box if the URL doesn't exist.

**Components****Pages*****App Root***

This page contains the header and the routing component so it will always display the header and then different content underneath depending on the current route.

***Overview Page***

Route = '/'

This page displays the Bookmark Form and Bookmark List components. It also manages the state of the bookmarks list and any bookmark to be edited. The bookmarks list is managed in the Overview Page as it is modified in both the Bookmark Form and Bookmark List. The bookmark to be edited is modified by the Bookmark List but stored by Overview Page so that it can be passed down to Bookmark Form so the bookmark to be edited can be used to populate the form.

***Results Page***

Route = '/results'

This page displays a message, the link that has just been added and a link back to the Overview Page.

**Reusable Components*****Bookmark Form***

This component displays a form to get the name and url for a bookmark, does validation and then saves the bookmark in local storage.

### ***Bookmark List***

This component displays a list of bookmarks with pagination (20 results displayed at a time). It uses the Angular Bootstrap pagination component

(<https://ng-bootstrap.github.io/#/components/pagination/overview>) for pagination but this is just a

UI component, the functionality to select and display the right content is still done manually in the Bookmark List component.

### **Limitations/Known Issues**

- The app will not be able to validate URLs if ad blockers are enabled on your browser
- No searching or sorting of the bookmarks
- Deleting bookmarks actually deletes them rather than just marking as delete as the use case is quite simple and didn't require a more complex delete feature
- Testing doesn't cover all cases but there are a few tests and they all pass