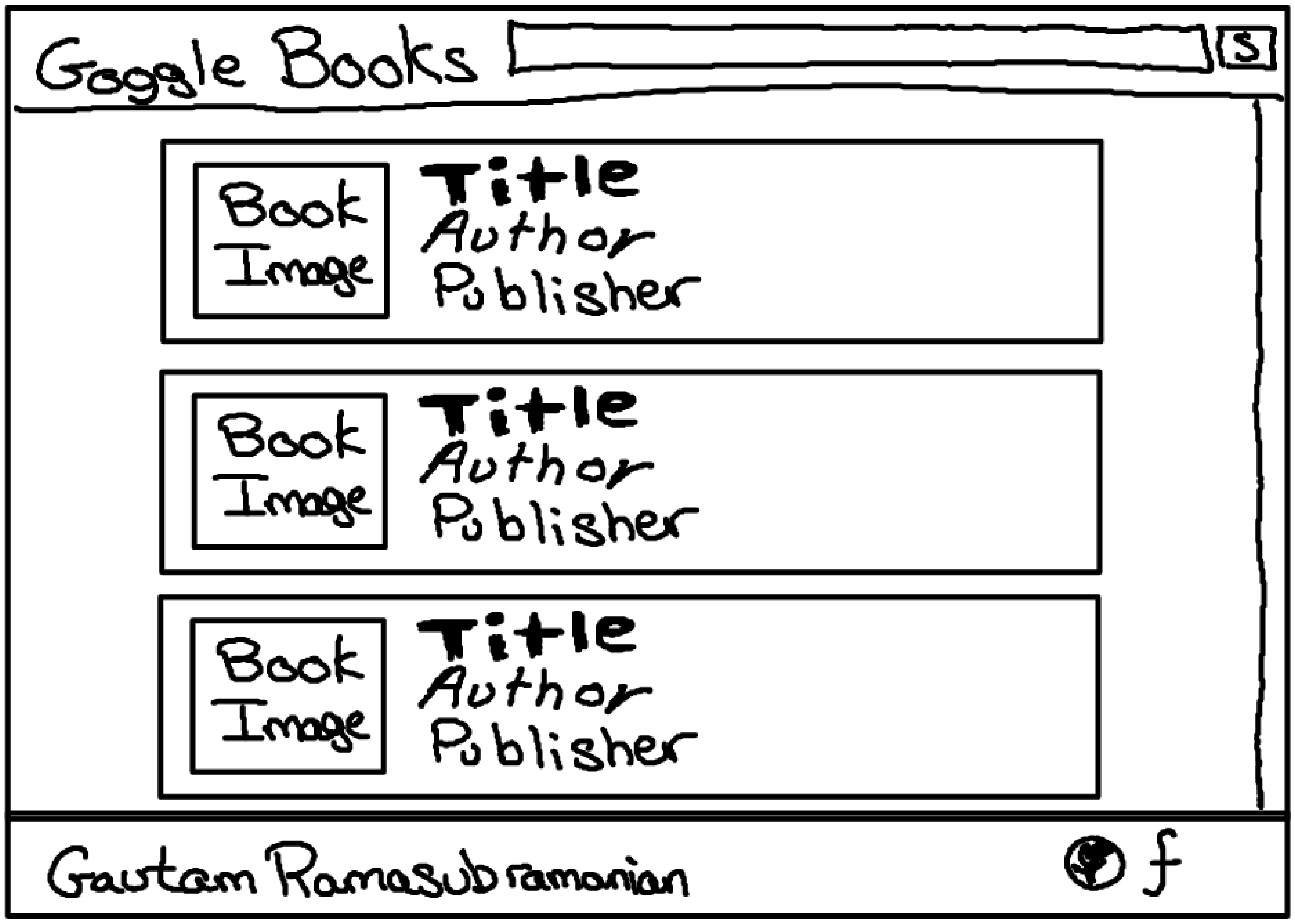
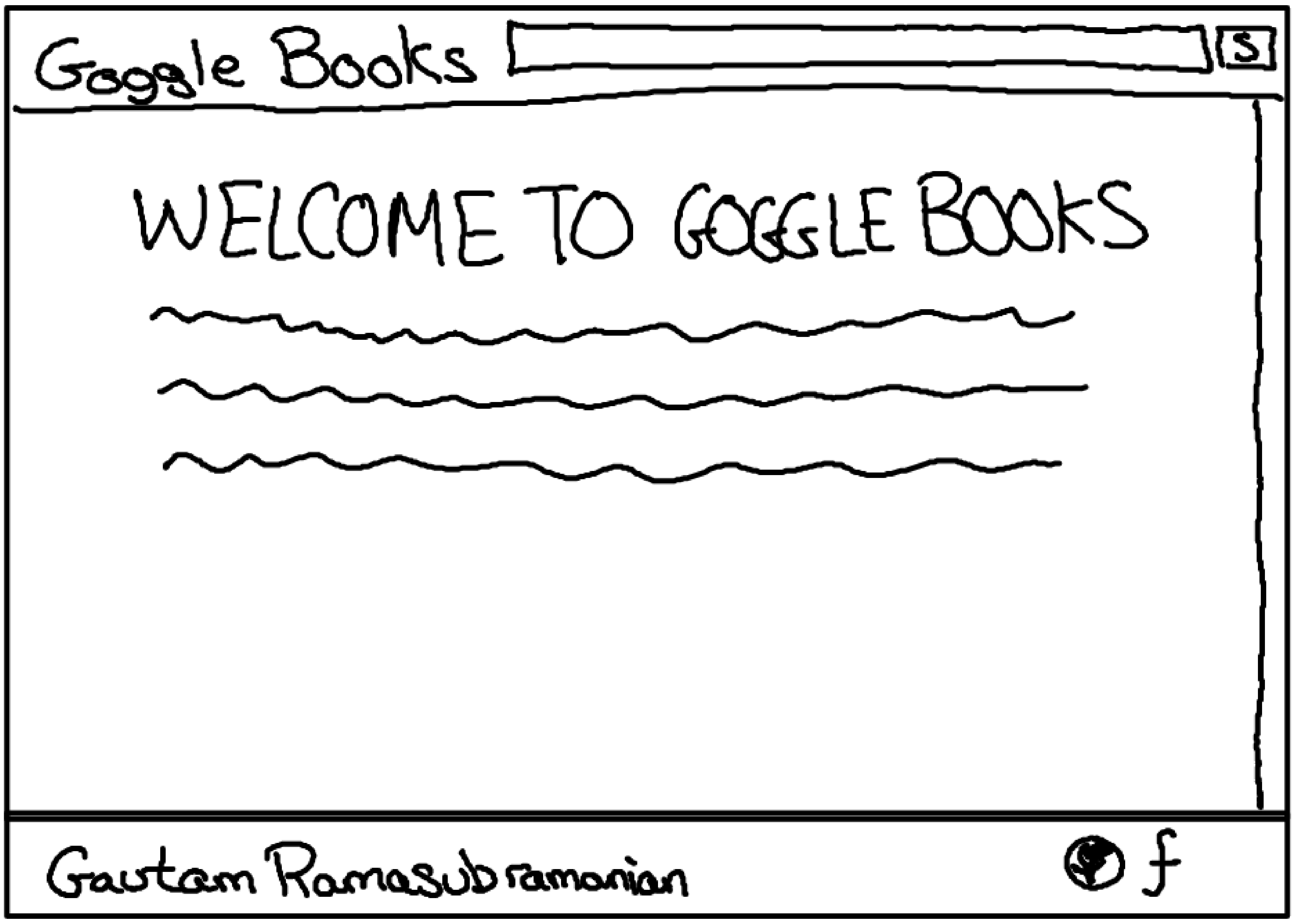
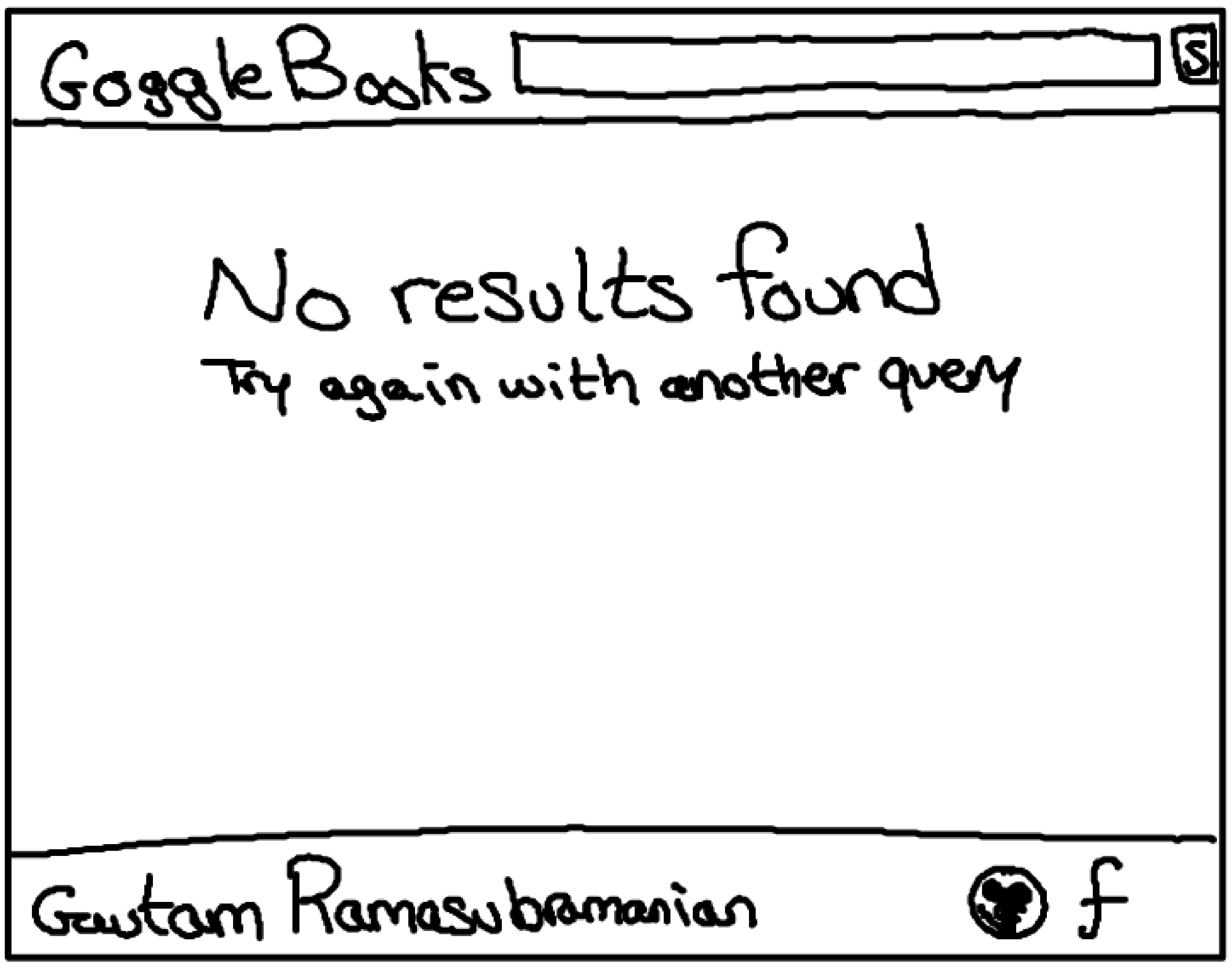
Project Name: Goggle Books

Allows the user to enter a search string and get a detailed list of books corresponding to that query string.

Desktop Experience







Simple Way

→ No server - just files → HTTP, CSS, Javascript

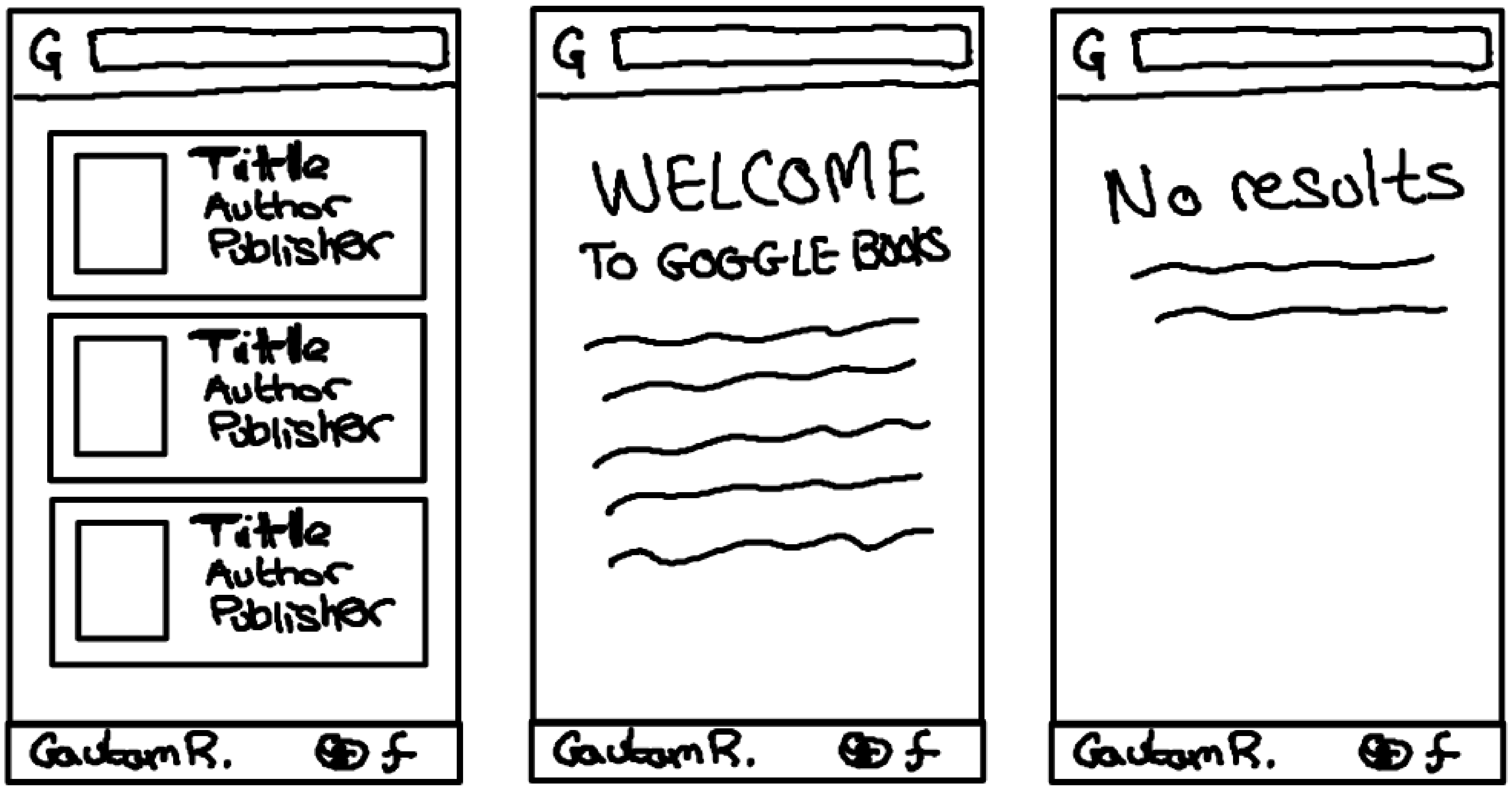
→ Advantages

- Do not have to learn fancy libraries
- Can use fetch, XMLHTTP-request for Google API

* - Testing will be awkward*

- ~~Host on Github pages.~~
- Host on Heroku

Mobile Experience



Create a project on Google Cloud Platform
Create an API key.
Set Up Heroku.
Set Up Project Structure → Techologies Used
Express, TailwindCSS, ZombieJS, Pug

Express Server has an endpoint for main webpage and another endpoint for displaying HTML for lists.

HTML has iframe to change.

Implement the bare minimum.

Before that, make it mobile friendly. → *Update Readme*

Now develop a set of css processors which I can call from npm scripts

☆ tailwind build input.css -o output.css

☆ purgecss --css input.css --content views/* --out output.css

☆ uglifyjs input.js --compress --mangle -o output.js

☆ uglifycss input.css --output output.css

☆ postcss input.css --use autoprefixer -o output.css

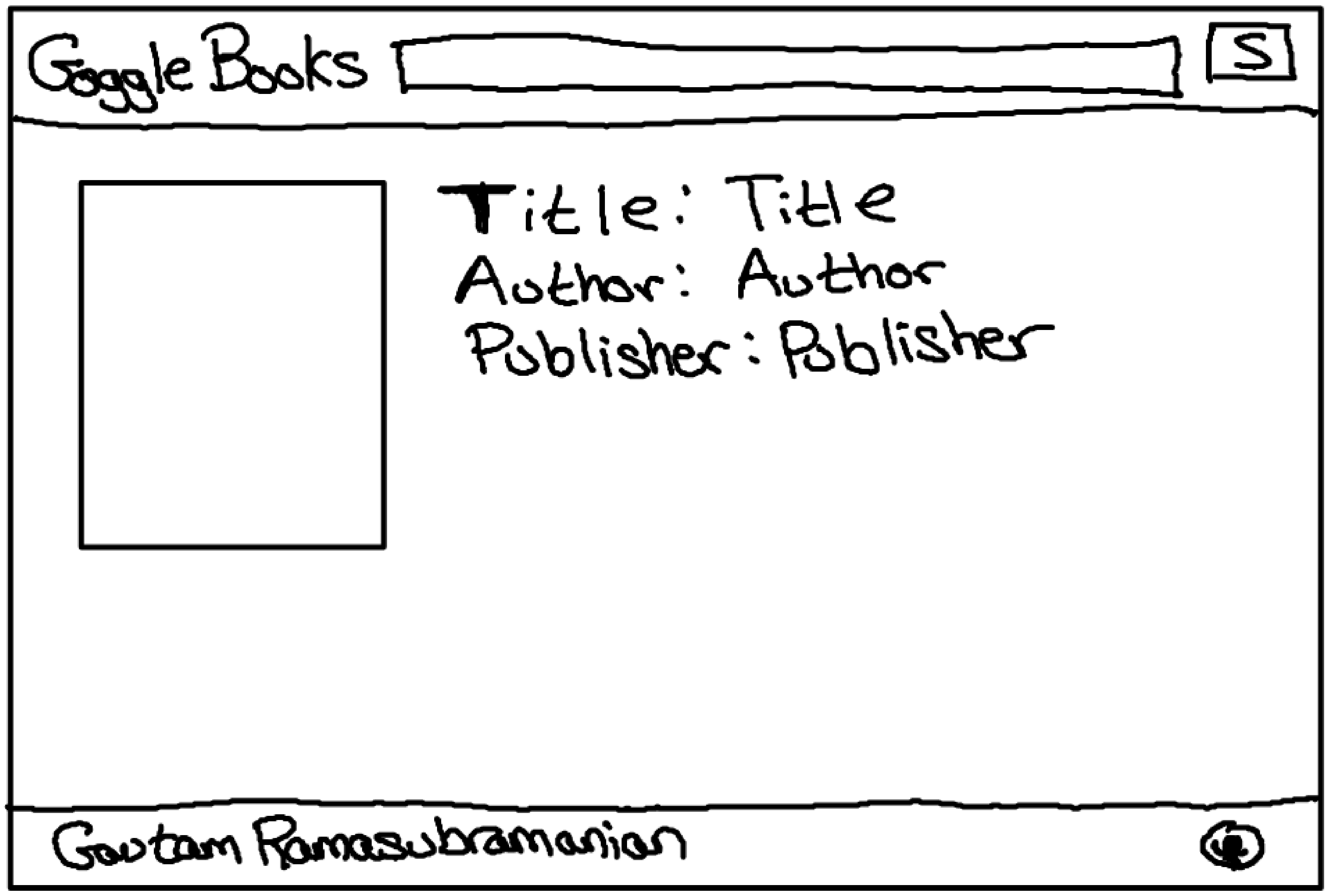
Make the welcome page. Make sure testing works.

The question is - what tool should I use for testing?

- Selenium JS? - Zombie JS? - Casper JS

- I will use NightmareJS.

Make a book details Page



What are the details I should include in this page.

Thumbnail, Title, Subtitle, Authors, Description
Publisher, Published Date, ISBN- . . .

Done.

Make comments + logging.

Use URL encoding.

Finish Readme Timeline.

6-22-19 → Created /search endpoint.
→ Allowed user to type in search query and get back a list of results.

6-23-19 → Created /volume/<id> endpoints.
→ Allow users to click on a search result to go to a page w/ details on that particular book.
→ Made Search Pages- More than one page of results

6-24-19 → Deployed new app on Heroku.
6-25-19 - 6-26-19 → Testing, Logging + Documentation.

Add Pictures.

Add Project Structure.

The directory structure of the project is as follows. . . .

Add more tests.

What do I wish to test exactly?

- Search Results. Make a list of search queries
- Get search ids and see if they match.