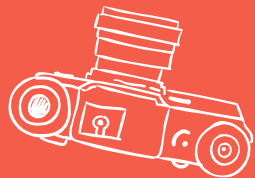


THE TECH LIBRARY

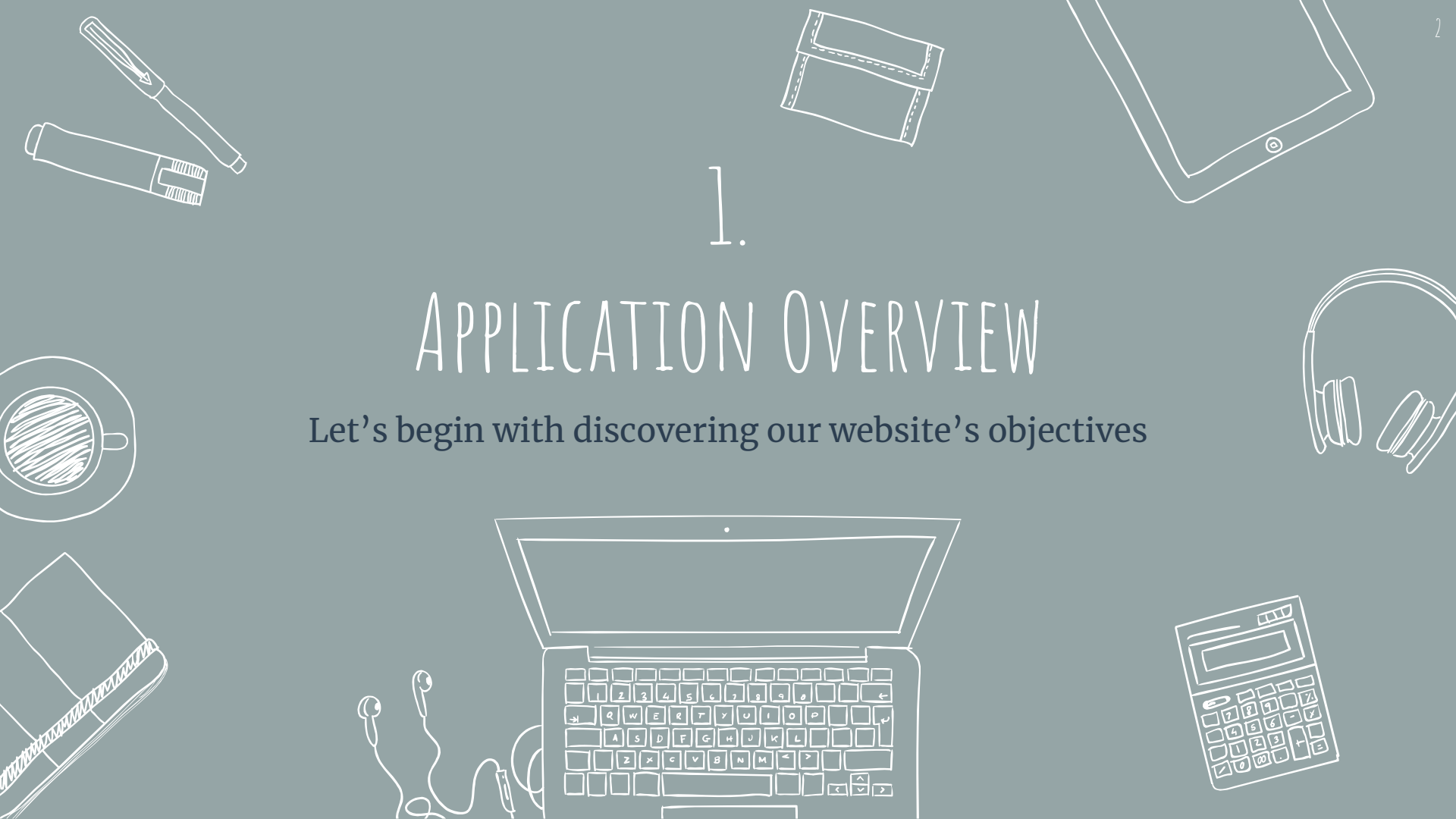
PROJECT BY: ANDREW CONNELL, BEN KOHAV, CHARLOTTE GORGEMANS,
ROXANNE MANTHY, RYAN HANNIGAN, SEAN SHI



1.

APPLICATION OVERVIEW

Let's begin with discovering our website's objectives

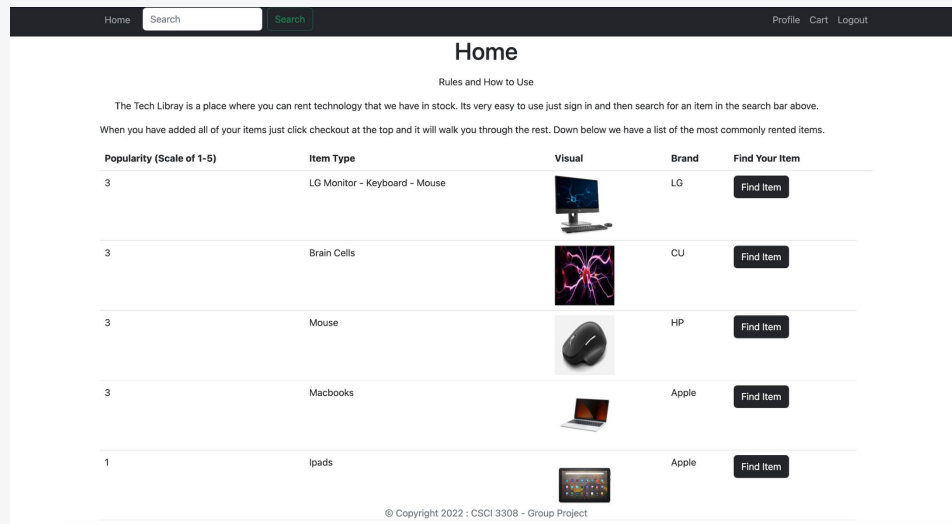


APPLICATION OVERVIEW

OBJECTIVE

Provide a forum for students to borrow essential school materials and equipment for day use.

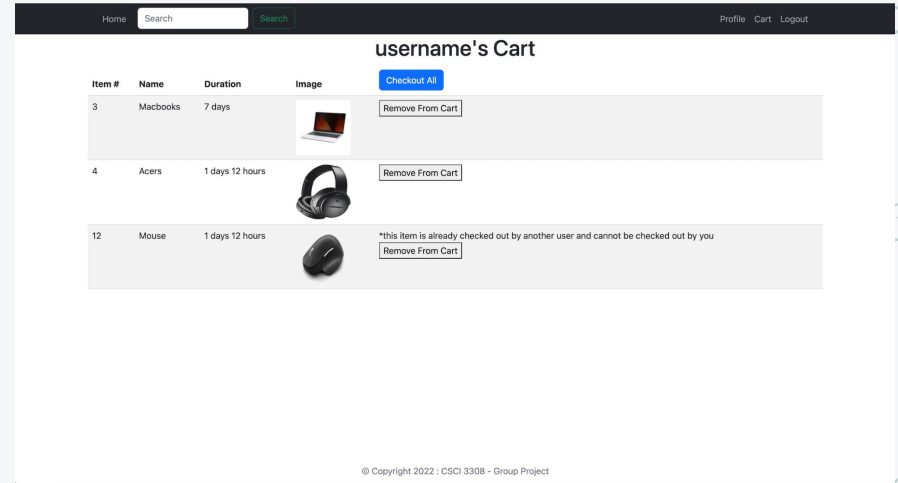
Scenario: Anna is a commuting student and has left her laptop charger at home. Our Tech Library application allows Anna to connect with a resource center on campus to borrow a charger for the day.



APPLICATION OVERVIEW & WEBSITE FLOW

MISSION EXECUTION

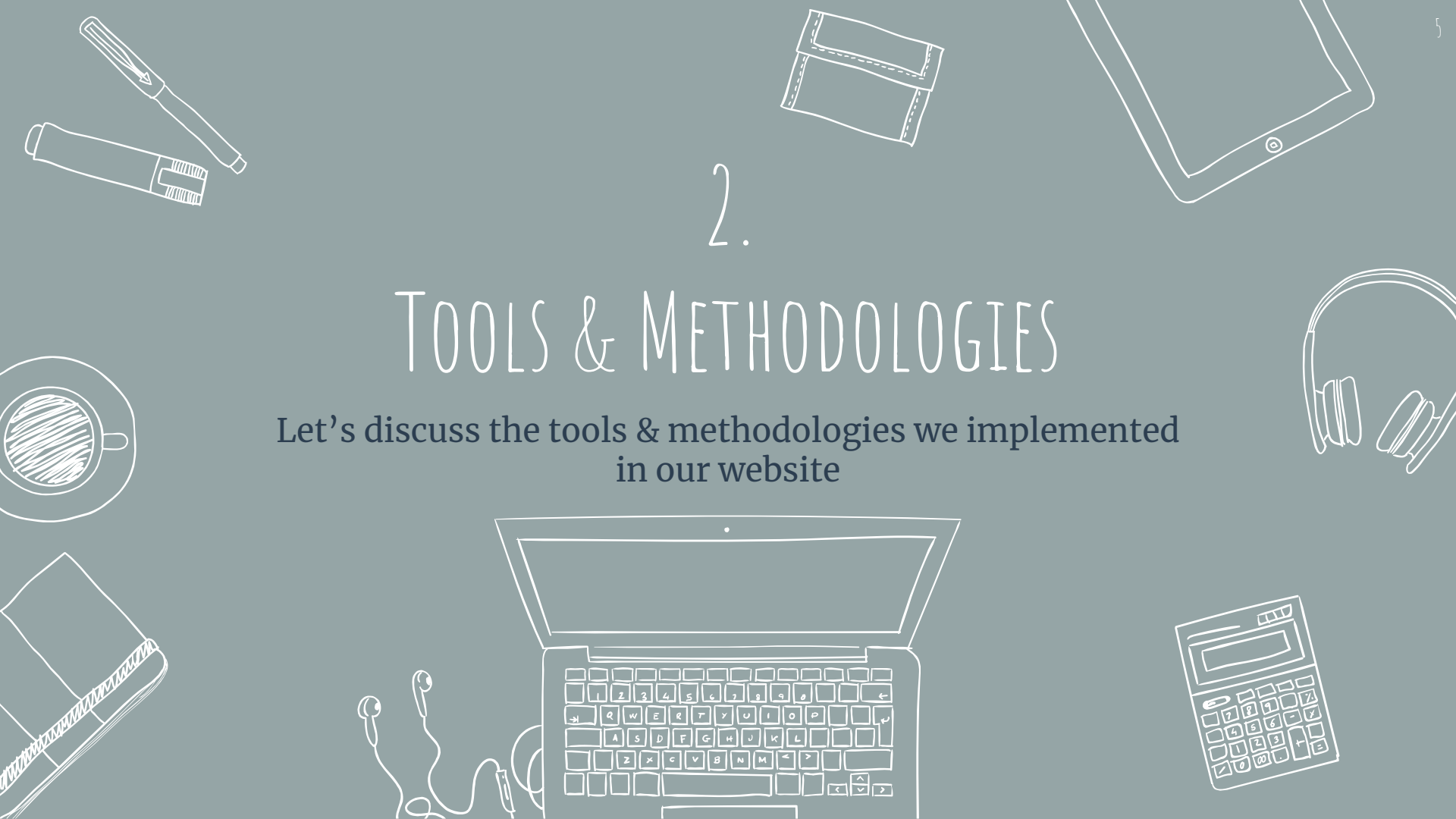
1. Register & Log In
 - a. User gains access to nav bar and home page
2. Home
 - a. User browses popular items & can select an item
3. Search
 - a. User can search for a specific item
4. Cart
 - a. User is redirected to a page, displaying their cart & selected items
5. Profile
 - a. User can modify their profile info and view what they have checked out.
6. Logout
 - a. User logs out of website & secures their account

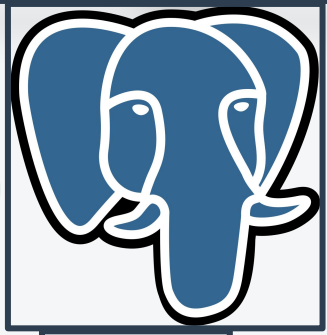


2.

TOOLS & METHODOLOGIES

Let's discuss the tools & methodologies we implemented
in our website





The
Tech
Stack

TOOLS USED IN DEPLOYMENT

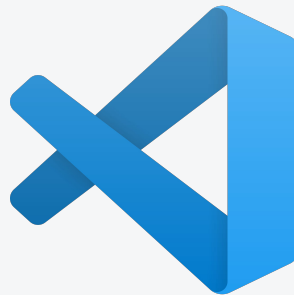
OVERVIEW

- EJS: 7/10
 - Easier creation of pages
 - Easier rendering of backend to frontend
 - Issues with syntax and unexpected behavior(html vs js)
- NodeJs: 8/10
 - APIs
 - Minor issues with syntax
- Postgres: 9/10
 - Functions
 - Useful data types
 - Easy data manipulation
- Docker: 7.5/10
 - Great service for running our application.
 - Issues with opening & initially running Docker.

TOOLS USED IN THE DEVELOPMENT STAGE

OVERVIEW

- **GitHub: 8.5/10**
 - Great service for version control.
 - Merge Conflicts
- **VS Code: 9/10**
 - Code is presented in a readable format, smoother programming process.
 - Easier Merge Conflicts
- **GroupMe: 8/10**
 - Easy to view all members' views.
 - No meeting support





METHODOLOGIES OVERVIEW (AGILE)



techLibrary Board



View 1 + New view

Filter by keyword or by field

Ice-Box 2



groupProject #36 ...



Rendering pages between actions work without error and data is updated between them

groupProject #46



Help Page

Support 5

Todo 2



groupProject #47



Data for profile page (first name, last name, username, etc) should render

Profile 3

groupProject #67



modals for profile page should be updated

Profile

In Progress 5



groupProject #5



Profile

Profile 6

groupProject #3



Home Page

groupProject #71



Item Information After Search

Search 4

groupProject #87



Profile allows edits to name and such

Profile 4

Done 26



groupProject #90

Check for overdue items before checkout

Checkout 3

groupProject #17



Cart page - Get the base functionality shown up on our local host

groupProject #43



Make the data of users items render

Checkout 4

groupProject #63



Return Item

Profile 8

+ Add item

+ Add item

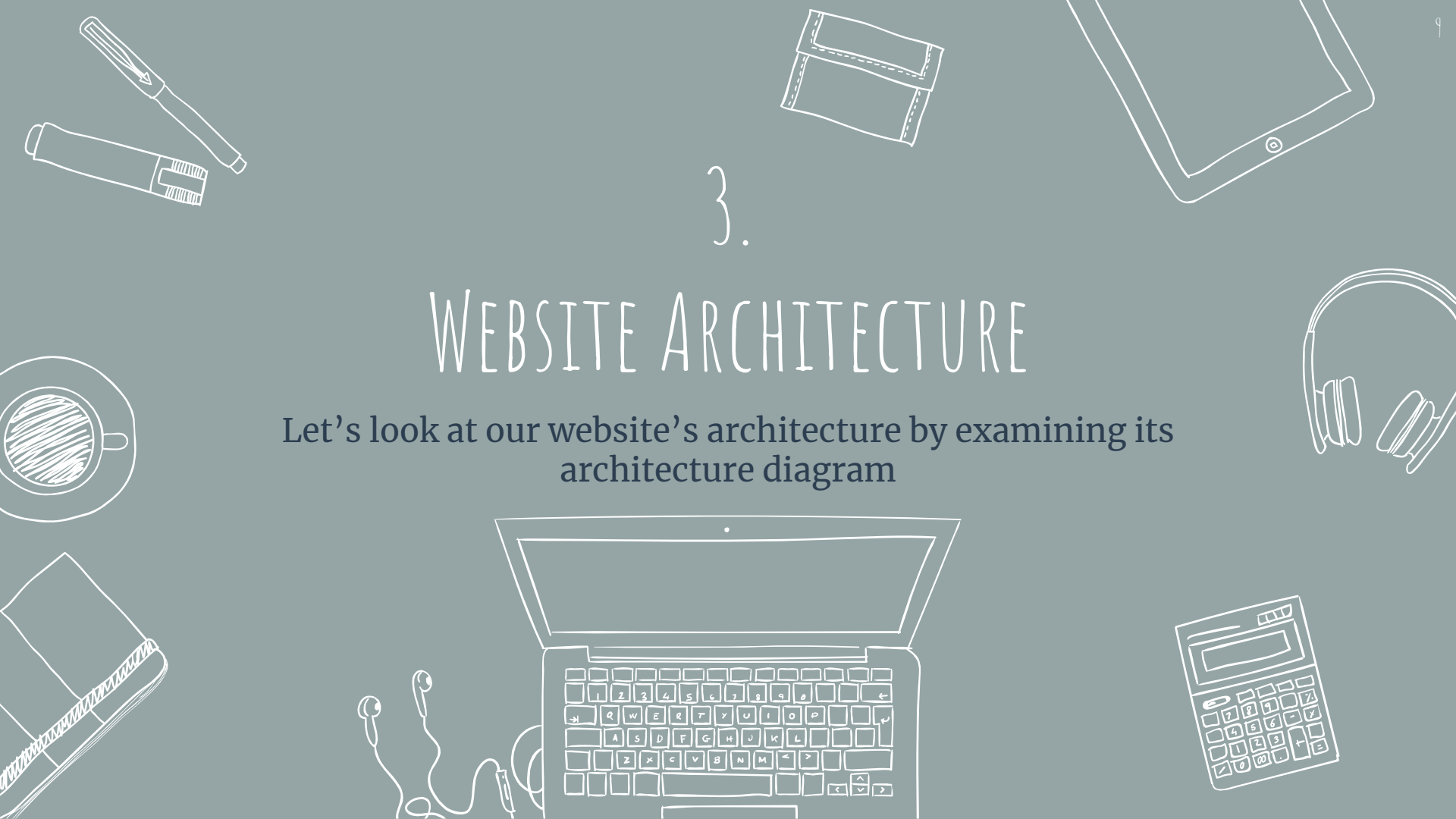
+ Add item

+ Add item

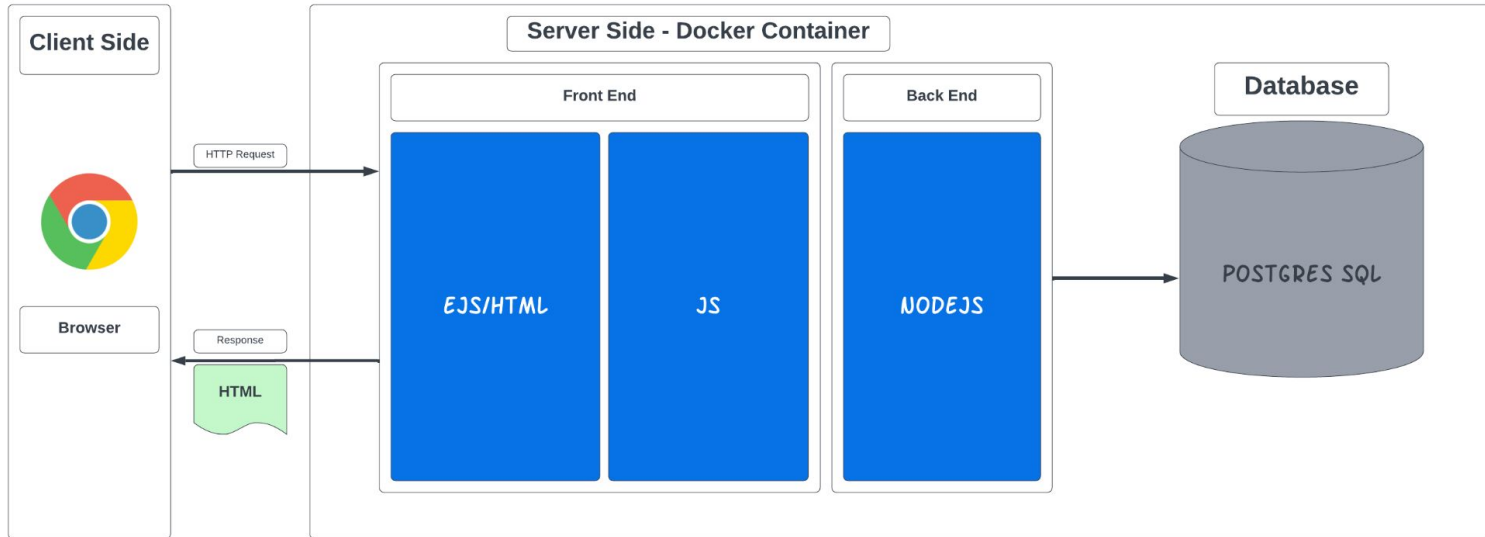
3.

WEBSITE ARCHITECTURE

Let's look at our website's architecture by examining its architecture diagram



WEBSITE ARCHITECTURE

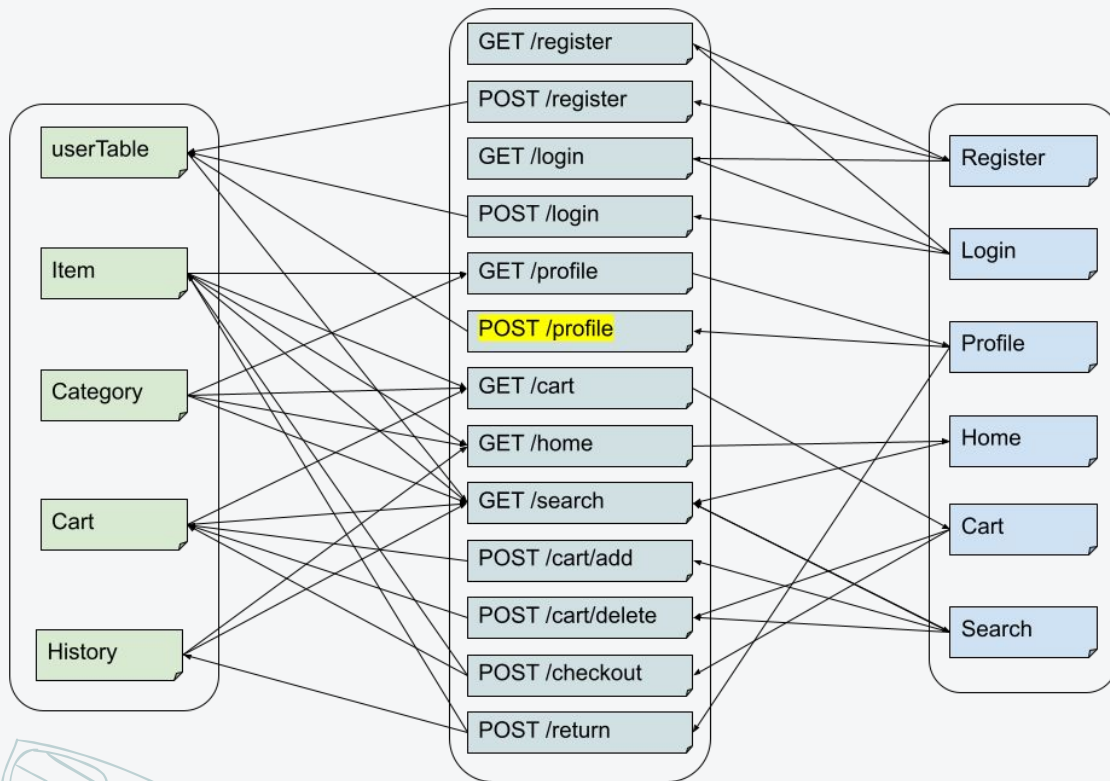


WEBSITE ARCHITECTURE

Database (Postgres)

API (Node JS)

Pages (EJS)



4.

CHALLENGES

We'll analyze the development of this website by
debriefing our challenges



CHALLENGES DURING DEVELOPMENT

CHALLENGES

- Deciding on a database & its organization
- Integrating our pages & combining all code together
- Determining where to store the data for the cart
- Discovering methods to develop a search functionality

RESOLUTIONS

- Maintaining open communication
 - Frequent updates via groupme
 - Asking for team's suggestions on questions
 - Clearly defined roles & responsibilities
- Seeking help & resources when needed
 - Visiting office hours
 - Collaborating with team members on tasks



LIVE DEMO!

We'll discover our website's full functionalities by a live demo!