



Alex Jackson
Seonaid McNabb
Imogen Minoli





Codify: The Idea

A collaborative app that provides tools to facilitate some of the practical, organizational needs of programmers, as well as spaces that encourage active learning.

Inspiration

Our own experience as bootcamp students.

Current Features

WHITEBOARD

-Visual note taking
during / after class

QUIZ

- Test your skills in specific topics
- Get interview ready

Q&A / HOW-TOS

-Learning
-Personal
Documentation

PRIORITIES LIST

-Self-reflection / planning



Technologies Used



Full project

- React.JS
- Express JS
- React Router
- React Reveal
- MySQL
- Canva
- Trello
- CSS, Bootstrap

Imogen

- Bcrypt
- JWT
- Postman
- Chakra UI
- QuizAPI

Seonaid

- DrawSQL App
- Postman
- Draft-JS
- DOM Purify
- HTML React Parser

Alex

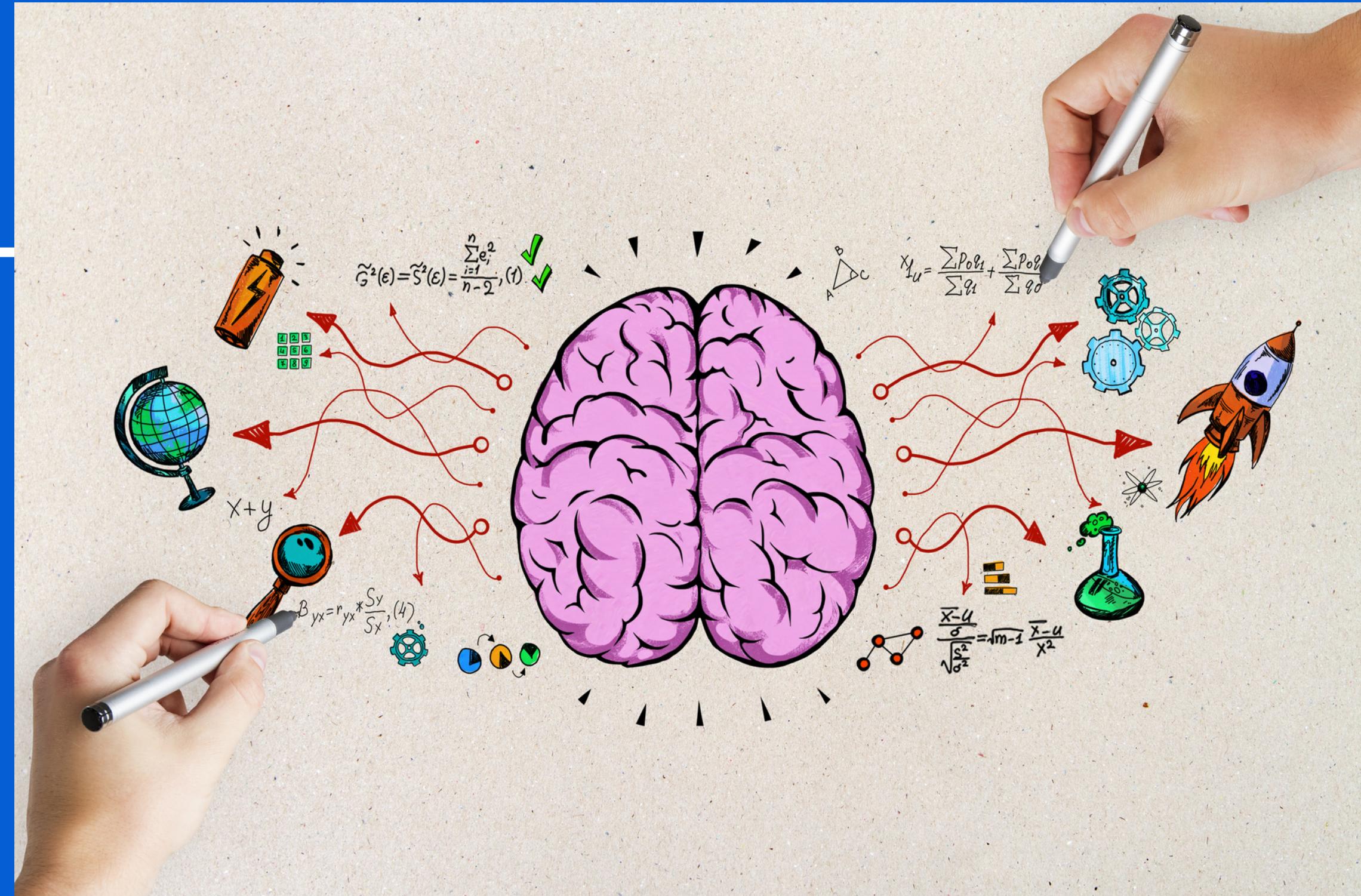
- HTML5 Canvas
- RoughJS
- Perfect
- Freehand
- React Color
- Flaticon
- Figma

Project Process

- scaffolding
- design
- feature implementation

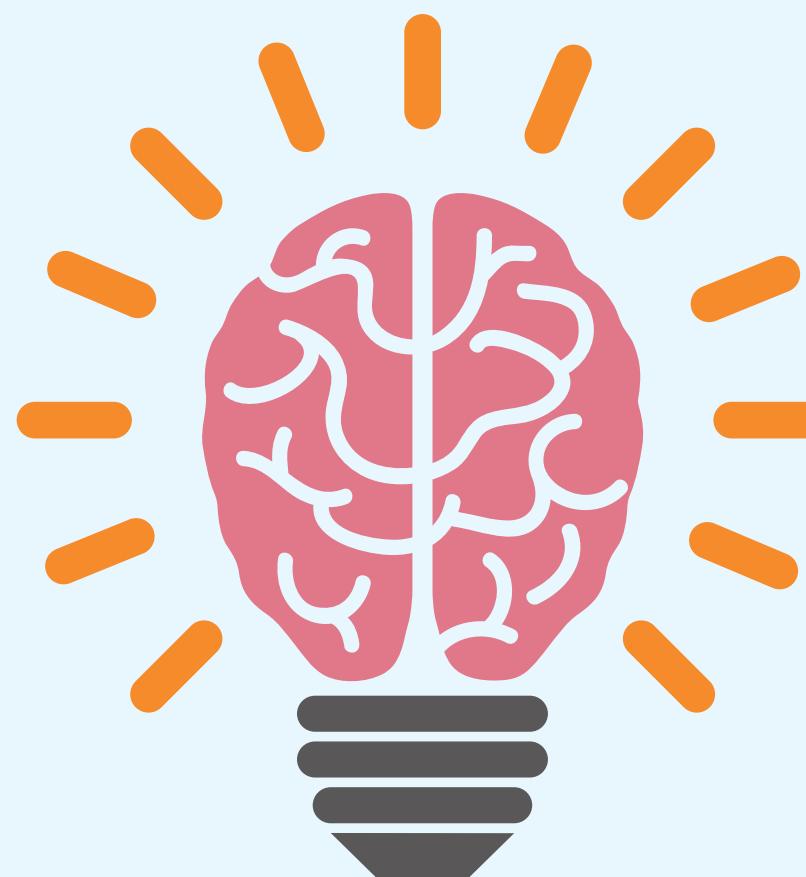


Brainstorming



Initial Meeting

In our initial meetings as a group, we discussed our different interpretations of 'education app' as an idea and what potential features this might have. We nailed this down to four main features as highlighted.



#USER-STORY: LEARNING PORTAL

1	User Story	Acceptance Criteria
User	As a ... bootcamp student	<input type="checkbox"/> Sign-up / Login page <input type="checkbox"/> Login area with username & password input <input type="checkbox"/> Successful login leads to homepage where there are two views (cohort & personal)
Need	I want ... an education portal where I can both interact with my cohort, and also track my personal learning process & resources,	
Reason	So that ... collaborative notes are easily accessible & visually appealing, I can improve my learning strategies, and I can access this information throughout the bootcamp and beyond	<input type="checkbox"/> Feature summary: sign-up, login, homepage, user & password database

Design



Logo example
Colours can be changed depending on final theme
Reading books within curly braces = learning and coding

Example name - can be changed
Codify = coding connotations and means to organise

Fonts:
Shrikhand for headings
Chivo Light for body text
Bold display vs simple serif

Welcome

Organise your learning

Simple form with login button

INSPIRATION BOARD

CODIARY

1 - onclick, opens resource view
2 - onclick, opens snippets view
3 - onclick, opens inspo whiteboard
4 - onclick, opens learning diary

FRANKLIN

Topic: Coding

Dashboard

Whiteboard

Self Reflection

About Us

Contact Us

< 4/8 >



Features

- Whiteboard
- Quiz
- Header & Login
- Q&As/How-Tos
- Priorities List



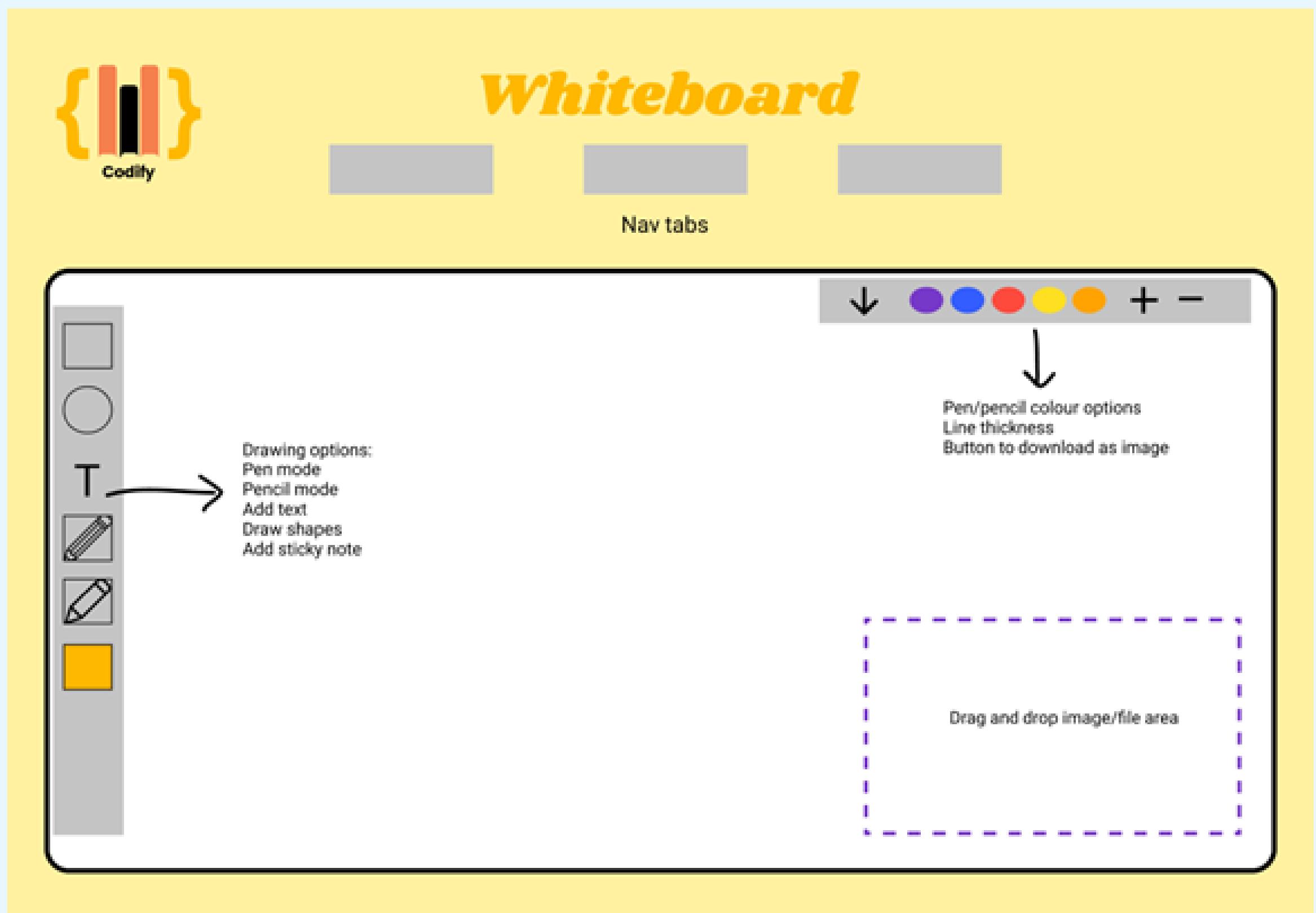
Visual notes and
diagrams



Whiteboard

People learn in different ways. For students new to coding, it may be difficult to really understand how something works by reviewing lines of code or written notes. The whiteboard feature is a way for students to visualise bootcamp material during and after class lectures to make more sense of complex coding concepts.





Started with an initial vision

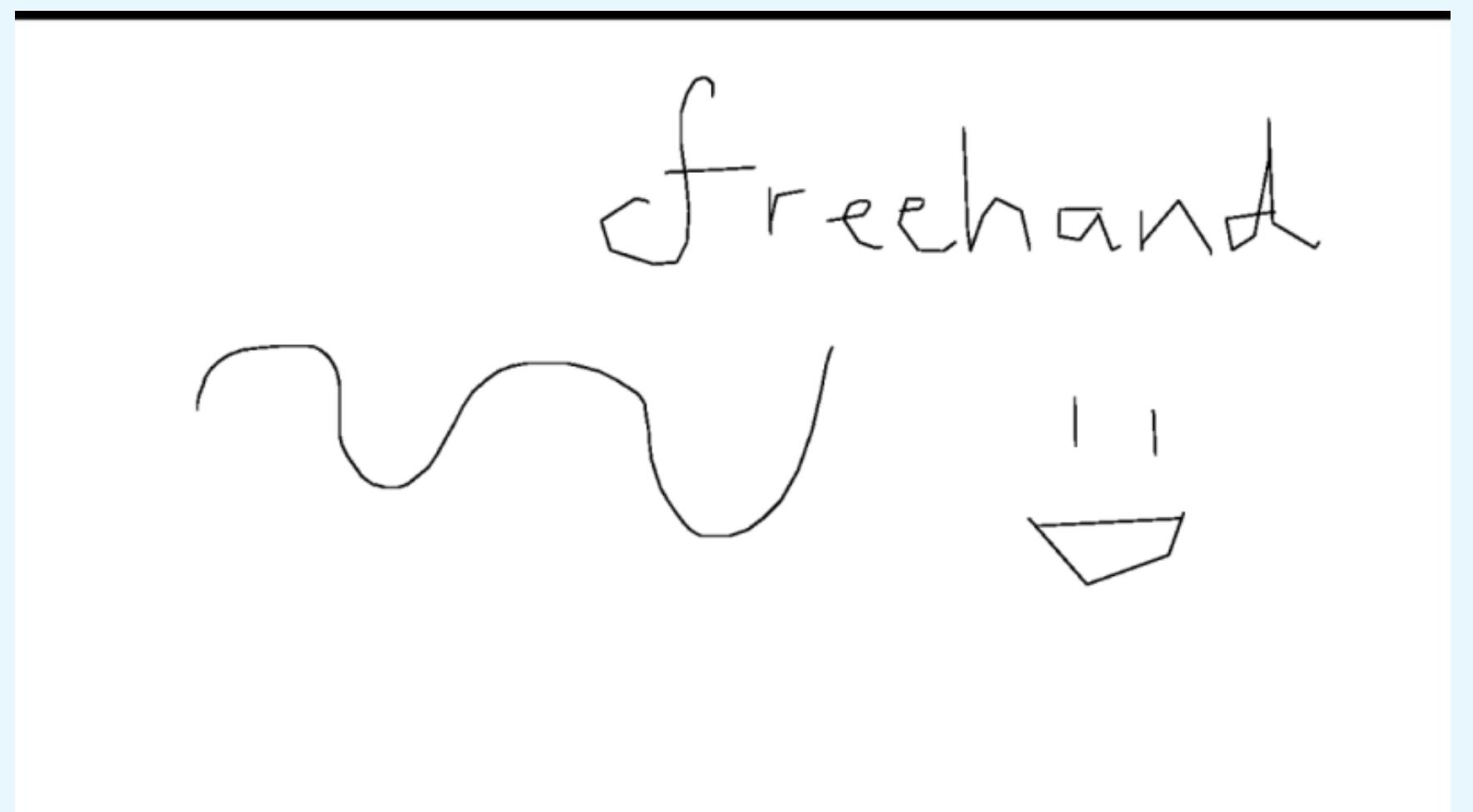
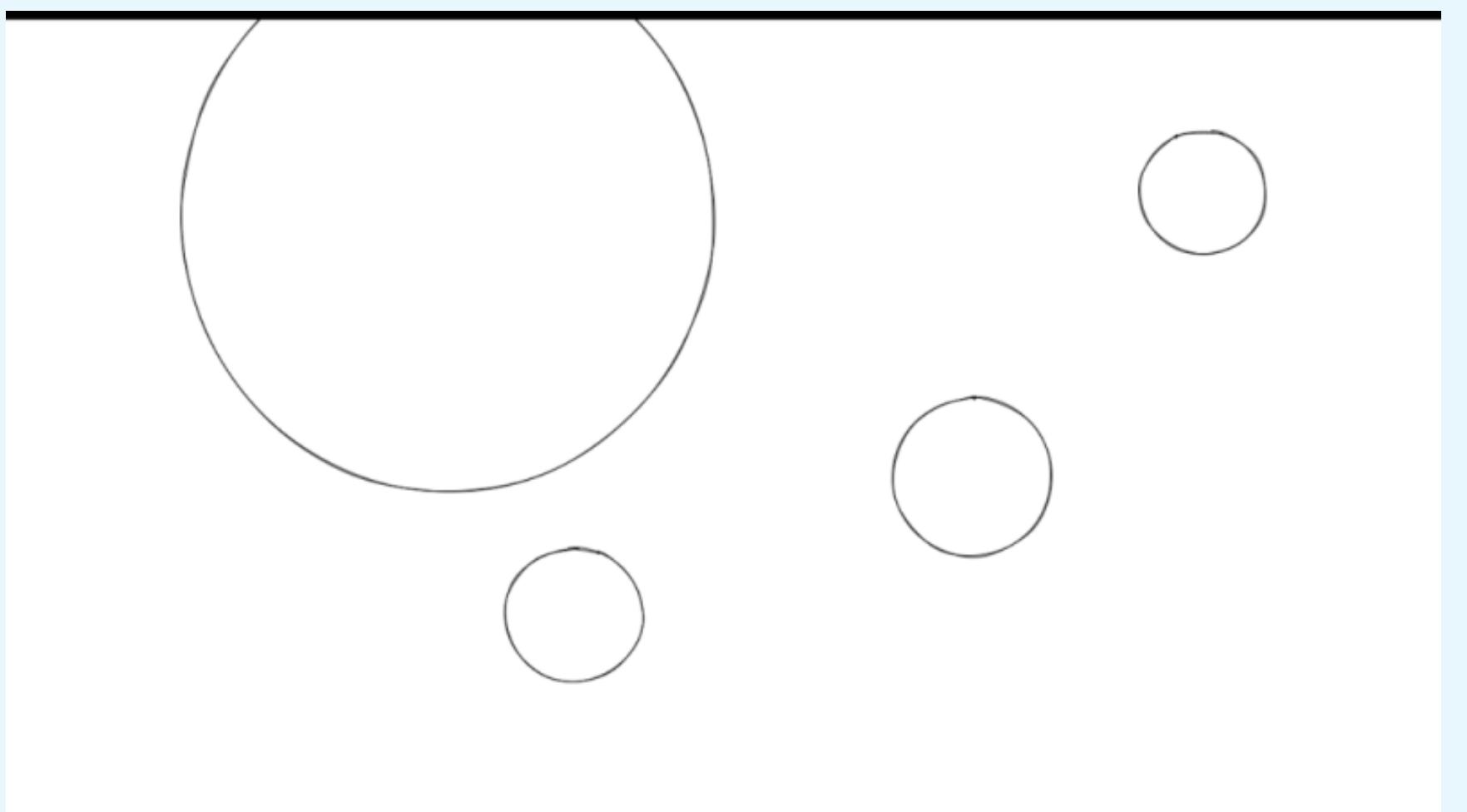
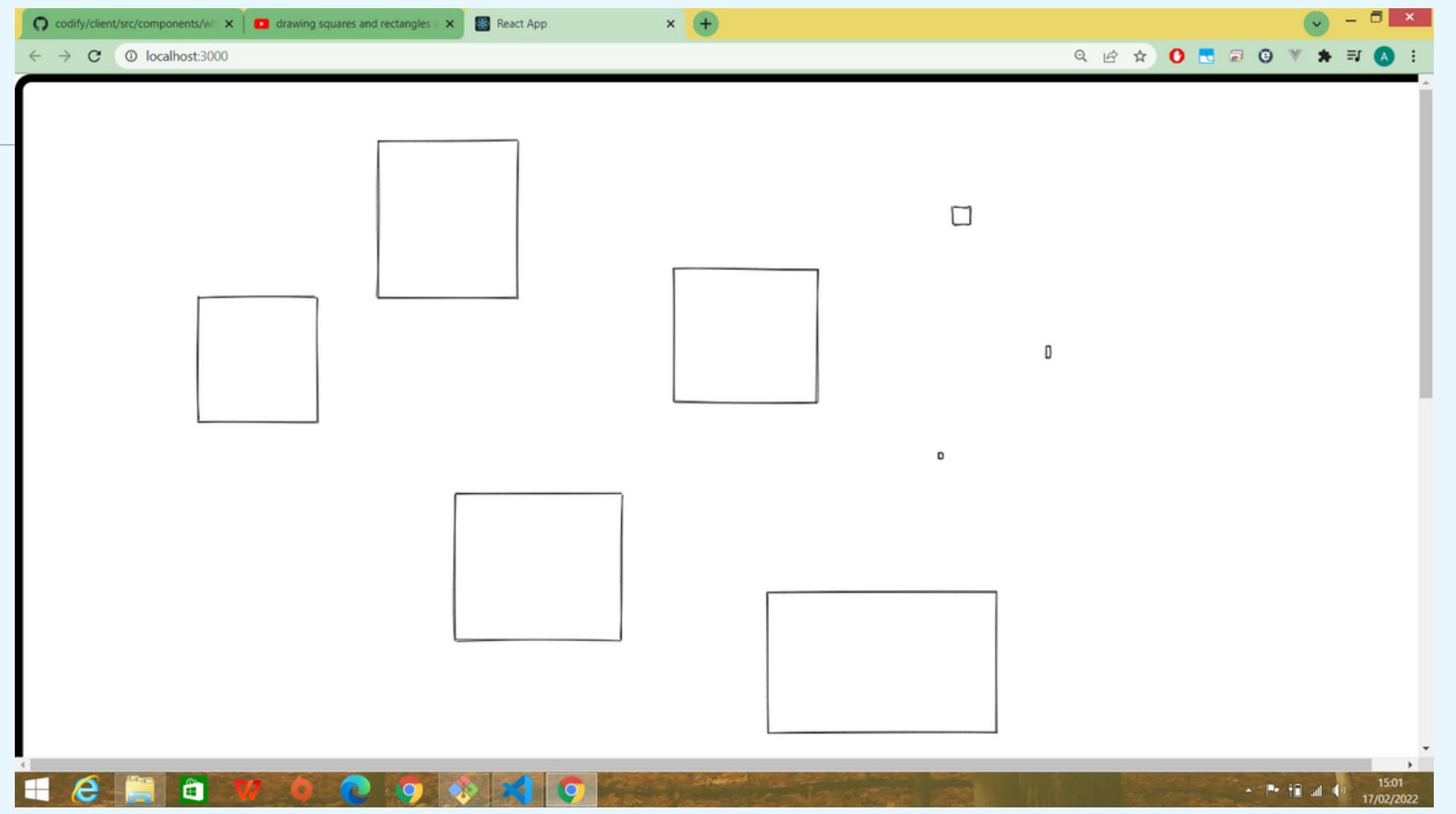
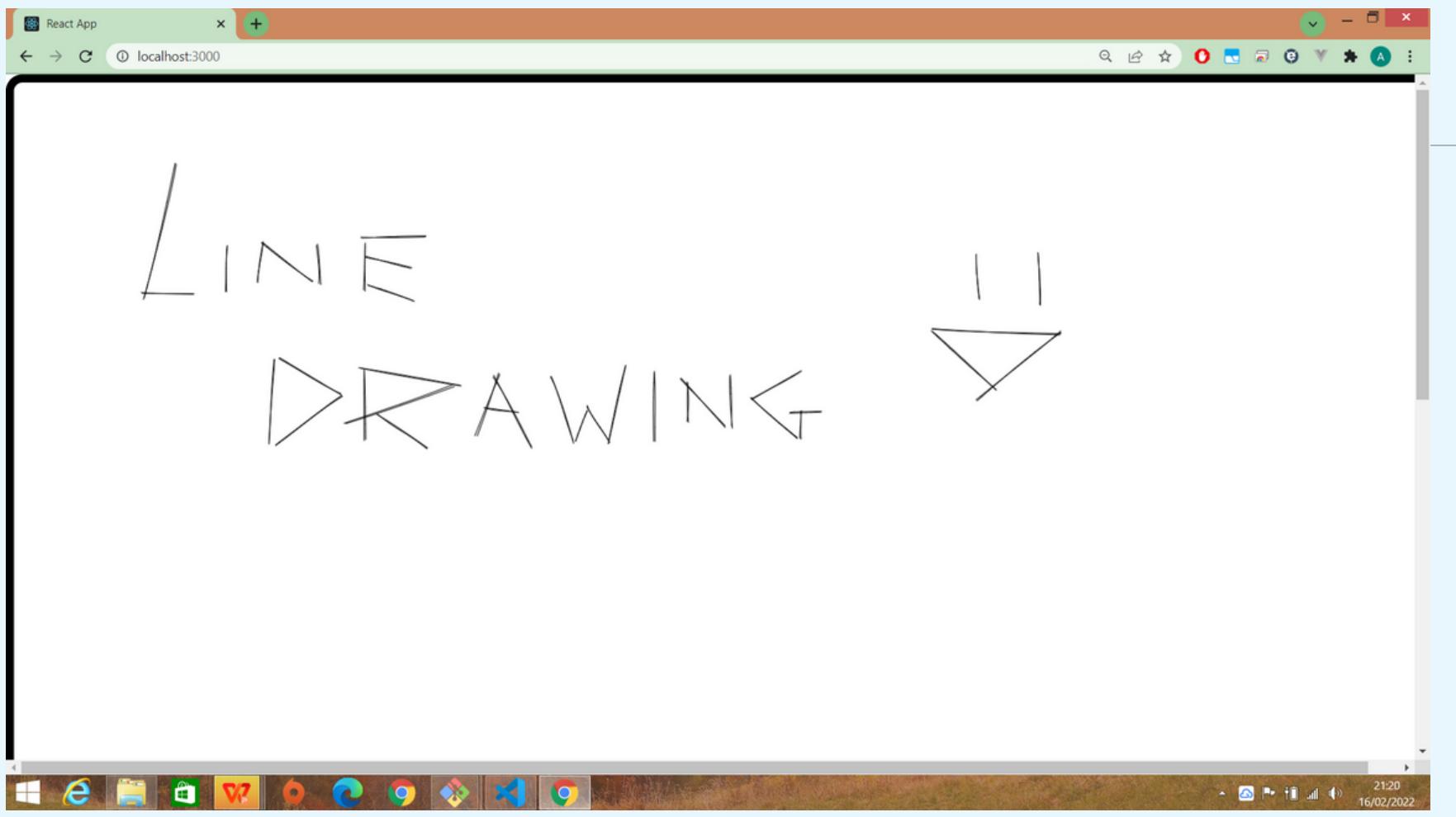
Soon realised what would happen if anything was placed above or below the canvas...

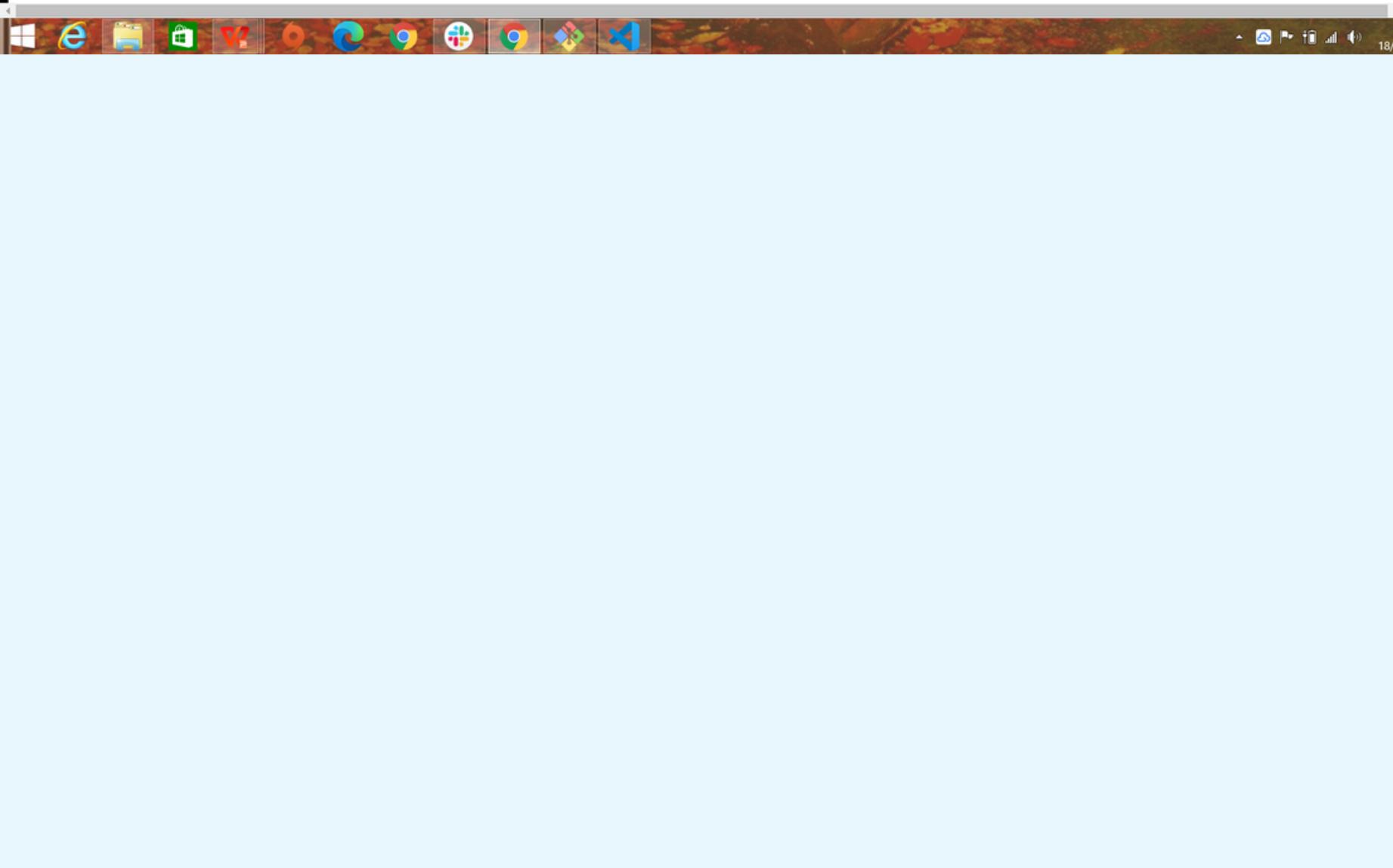
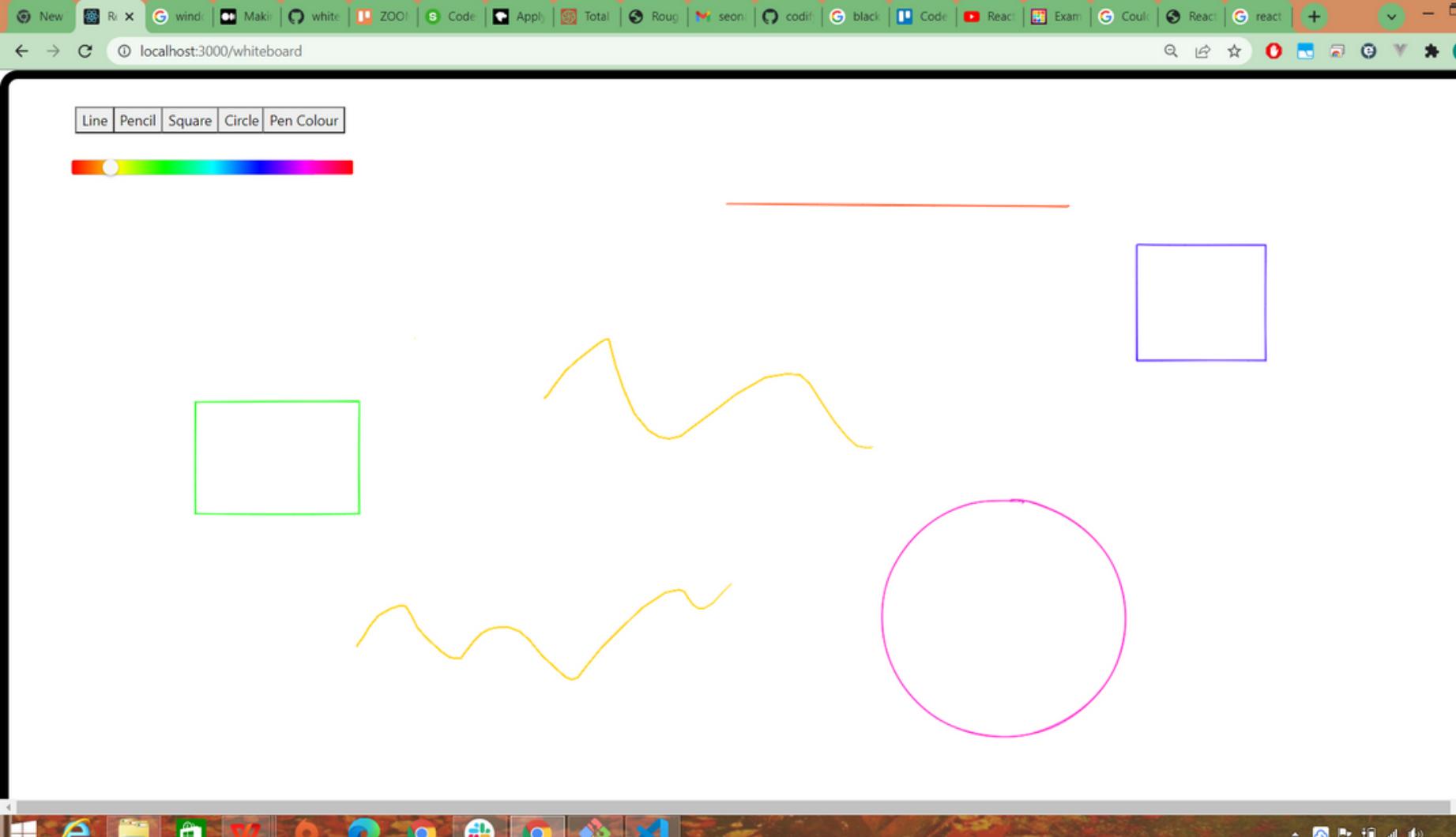


The Building Process

- Canvas API documentation
- Many many articles and tutorials - Youtube, Medium, Stack Overflow
 - Step by step approach / built tool by tool - once one tool was functional, I moved onto the next one
 - Decided to slow down and start again from the start to really understand the code
 - Styling --> other tools on the page, e.g. header buttons, whiteboard toolbar, had to be laid over the canvas using CSS positioning to avoid misaligning the cursor and drawings

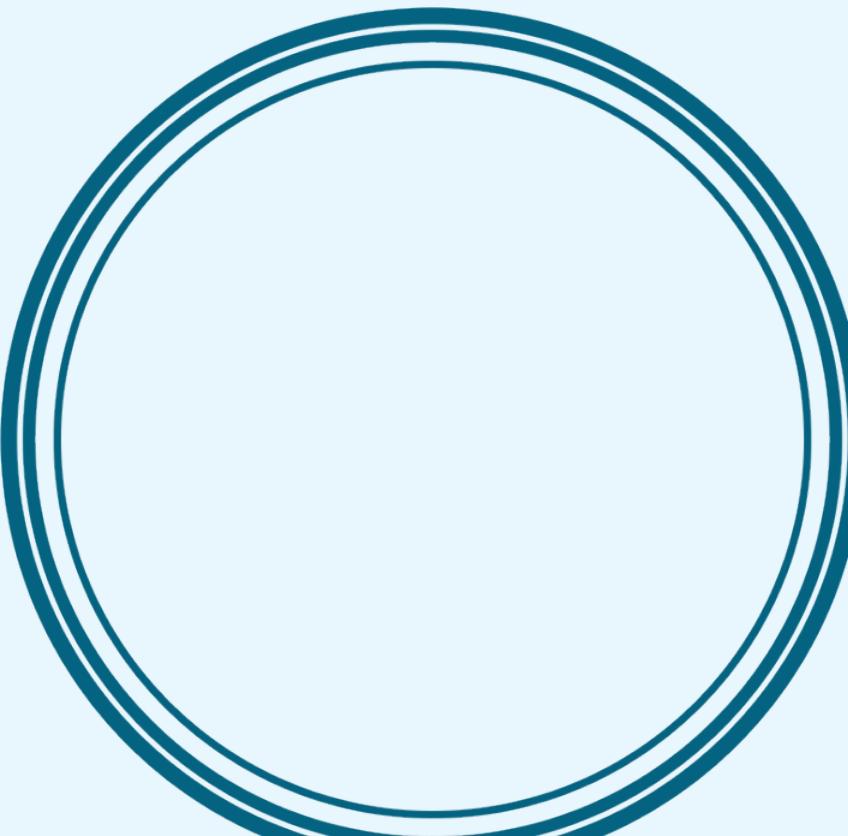






Maths!

- Tool to move and resize the drawings became very complex and mathematical very quickly
- Use of mathematical formulas and parts of my brain I haven't used for over 10 years :D
- Moving circles got the better of me - sounds easier than it is!!





Whiteboard

Quiz

Reflection Area

Personal Documentation

Show/hide toolbar

Download image

Current tool: select



Select background: Choose file Code example.png

Pen Colour:



Fill Colour:

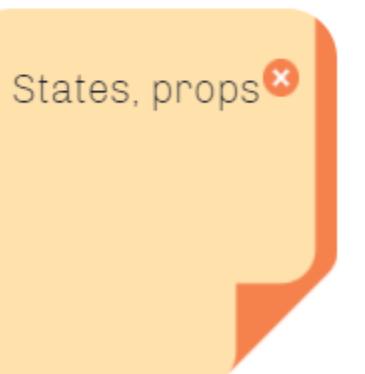


Codify

```
const handleAddProject = (newProject) => {
  setProjects((state) => [...state, newProject]);
};
```



whiteboard, setWhiteboard





Whiteboard

Quiz

Reflection Area

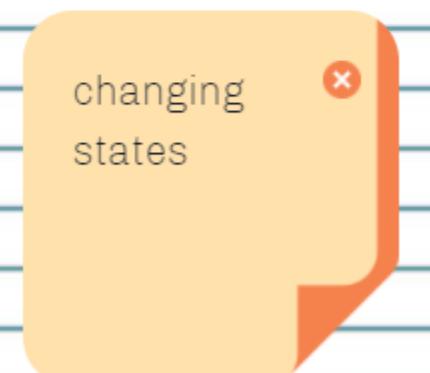
Personal Documentation

Show/hide toolbar

Download image

changing states

States



```
const handleAddProject = (newProject) => {
  setProjects((state) => [...state, newProject]);
};
```



Finished Product

A functional whiteboard with tools including:

- Select tool to move and resize drawings
- Pencil / freehand drawing
- Line and square drawing
- Ability to add typed text
- Sticky notes that can be dragged around the canvas
- Increase/decrease line thickness
- Different background images including lined and grid
- Local image upload
- Pen / shape fill colour pickers
- Undo/redo functions
- Download image function which takes a screenshot of the canvas in .png format
- Button to clear and reset the canvas



With more time...

- Drag and drop area for local files and images
- Re-introduce circles and work out how they can be moved and resized
- Save instances to local storage so users can navigate away from the whiteboard if needed, but not lose their notes
- Styling - change cursor icon per tool, e.g. a pencil for freehand drawing
- Ability to send current instance of whiteboard to the backend resource area



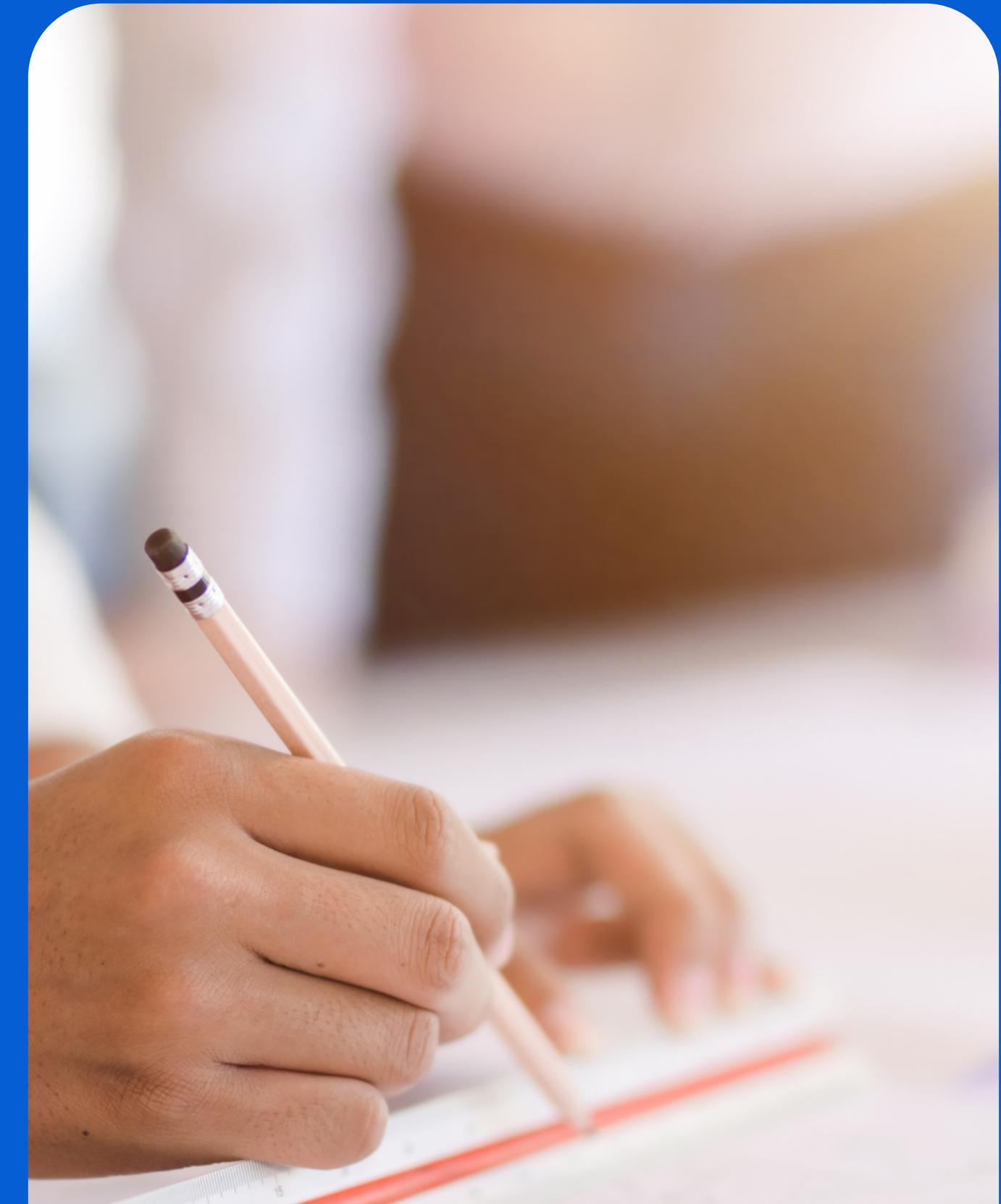
Features



- ✓ Whiteboard
- ✓ Quiz
- ✓ Header & Login
- ✓ Q&As/How-Tos
- ✓ Priorities List

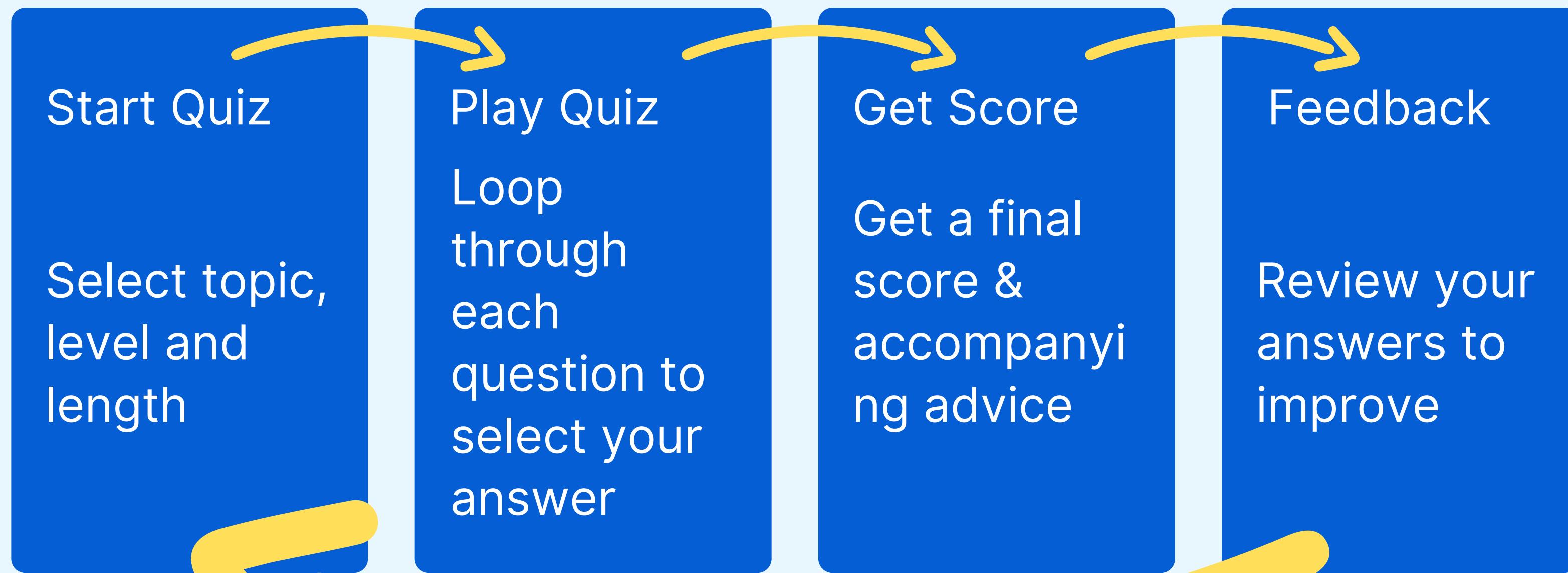


Planning



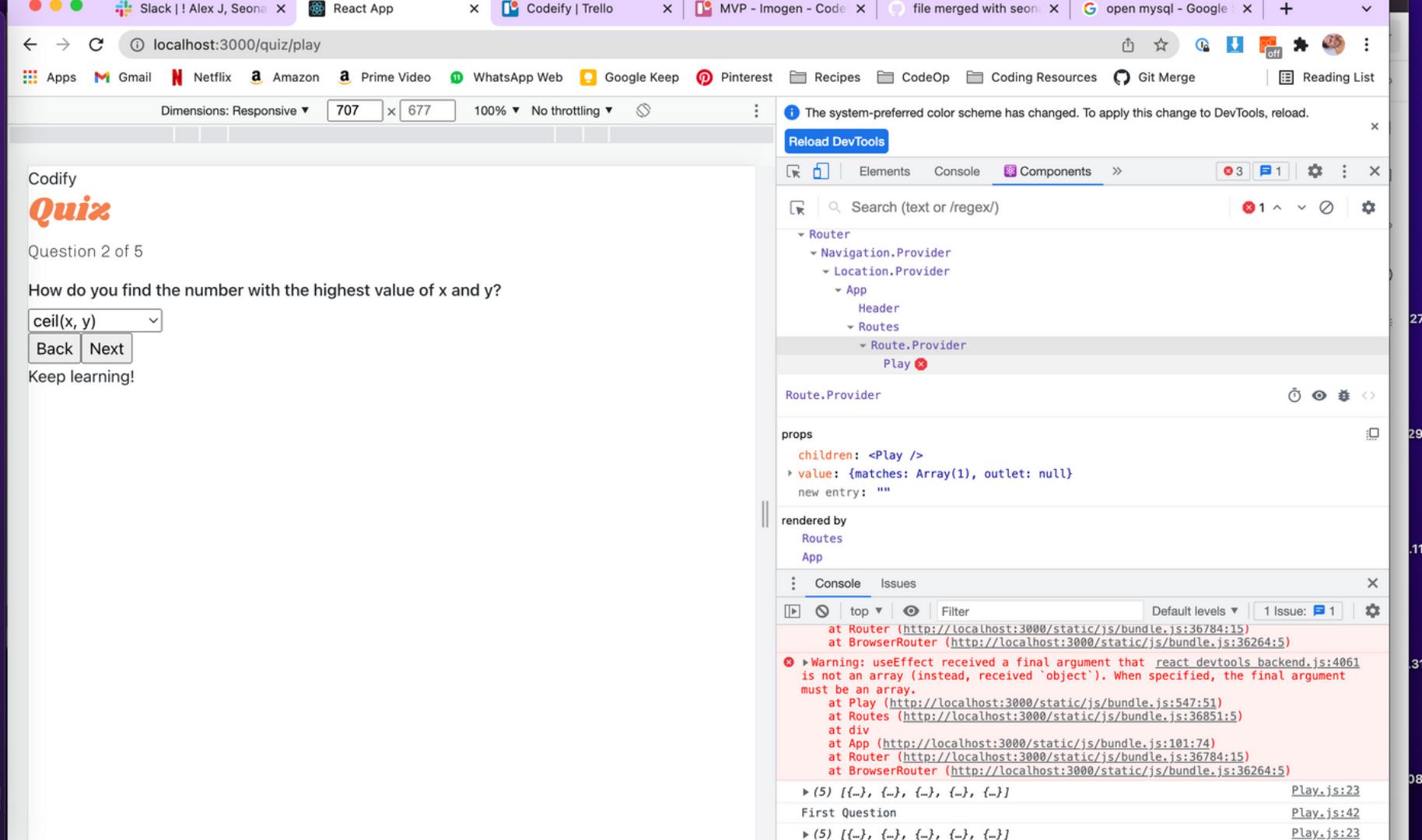
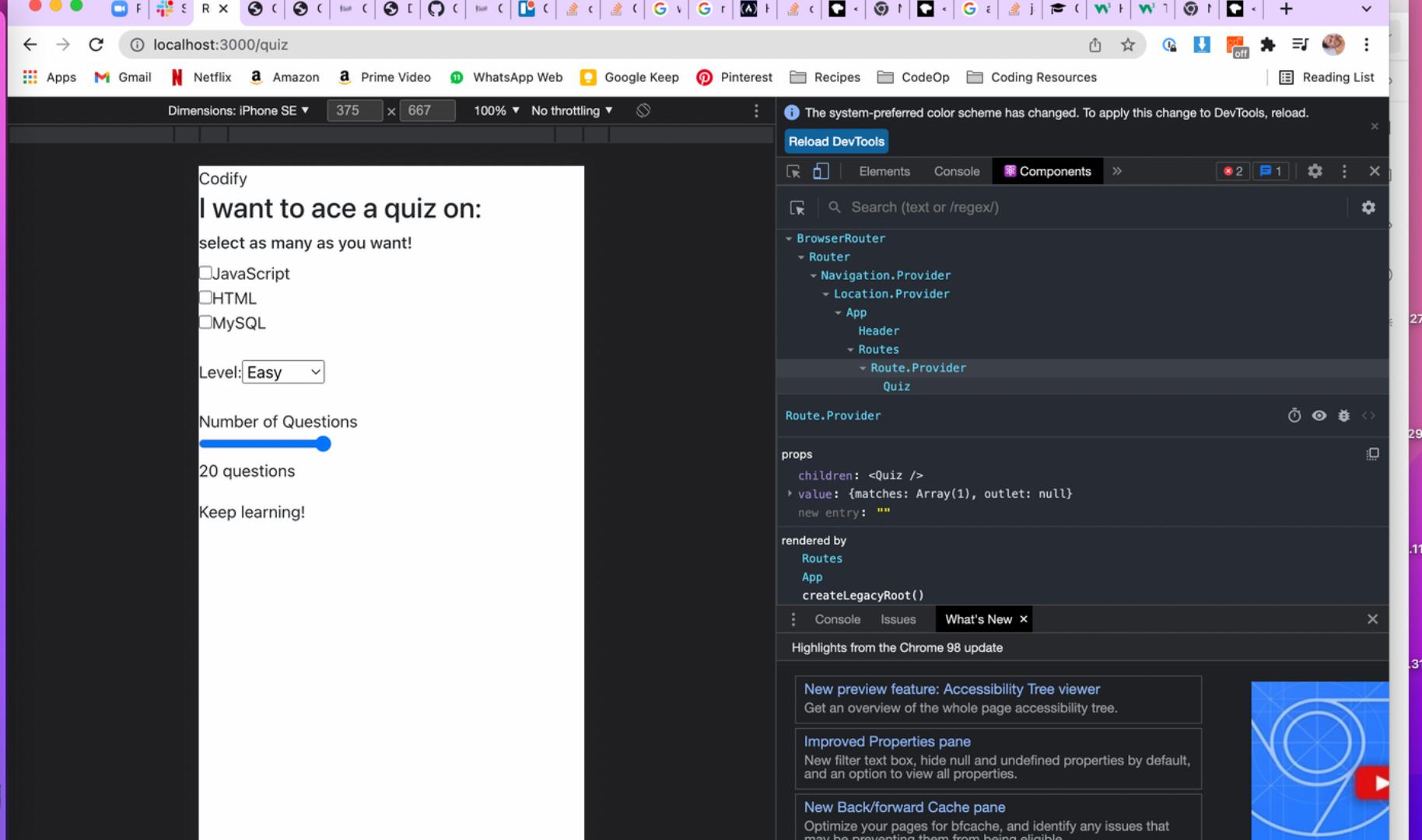
Quiz Concept

A quiz that developers can use to test their knowledge and skills in various topics and get interview ready that is suitable for developers of all levels.



Early Stages

The QuizAPI was relatively easy to use, the data passing between sibling components started to become challenging for me early on.



How does it work

The screenshot shows a code editor with a dark theme. On the left is the Explorer sidebar, which lists the project structure under 'CODIFY'. The 'src' folder contains 'components', 'quiz', and 'public' folders. 'components' has files like 'Backup Brain.js', 'Personal Reflect.js', and 'Quiz.css'. 'quiz' contains 'Play.js', 'Quiz.js', and 'Results.js'. Other files listed include 'Dashboard.js', 'Footer.js', 'Header.js', 'Home.js', 'Login.js', 'Logout.js', 'SignUp.js', 'App.css', 'App.js', 'App.test.js', and 'index.css'. The 'Results.js' file is open in the main editor area, showing its code.

```
client > src > components > quiz > JS Results.js > [e] Results > all_answers
13
14 const Results = (props) => {
15   let points = 0;
16   let result = "";
17   let message = "";
18   let emoji = "";
19   let score = 0;
20   let answers = [
21     {
22       question: "",
23       user_answer: "",
24       formatted_user_answer: "",
25       correct_answer: "",
26       formatted_correct_answer: "",
27       all_answers: [],
28     },
29   ];
30
31   function getKeyByValue(object, value) {
32     return Object.keys(object).find((key) => object[key] === value);
33   }
34
35   for (let i = 0; i < props.questions.length; i++) {
36     let value = props.questions[i].correct_answers;
37     let correctAnswer = getKeyByValue(value, "true").slice(0, 8);
38     console.log(correctAnswer);
39     if (props.userAnswersArray[i] === correctAnswer) {
40       points++;
41     }
42     if (i == 0) {
43       answers.splice(0, 1, {
44         question: props.questions[i].question,
45         user_answer: props.userAnswersArray[i],
46         formatted_user_answer: props.userAnswersArray[i].slice(-1),
47         correct_answer: correctAnswer,
48         formatted_correct_answer: correctAnswer.slice(-1),
49         all_answers: []
50           props.questions[i].answers.answer_a,
51           props.questions[i].answers.answer_b,
52           props.questions[i].answers.answer_c,
53           props.questions[i].answers.answer_d,
54           props.questions[i].answers.answer_e,
55         });
56       props.questions[i].answers.answer_a,
57       props.questions[i].answers.answer_b,
58       props.questions[i].answers.answer_c,
59       props.questions[i].answers.answer_d,
60       props.questions[i].answers.answer_e,
```

- 1 The user selects the quiz criteria
- 2 This inputted data is used to send the API request
- 3 The data comes back and is stored
- 4 The quiz loops through each question and stores the answer
- 5 The userAnswer is checked against the correctAnswer, score and feedback is given



Personal Documentation

Dashboard

Whiteboard

Quiz

Reflection Area

Not registered yet?

[Sign Up](#)

Email

email@address.com

Password

Enter the email you wish to login with.

[Login](#)[Logout](#)

Dashboard

Whiteboard

Quiz

Reflection Area

*Time to ace a quiz!***Topic:**

- HTML
- JavaScript
- MySQL

Level:

- Easy
- Medium
- Hard

Header & Login Concept

● Header: Tabs

The plan was to make Alex's idea for tabs come to life, so the user is filing away their skills, knowledge and plans for reference later.

● All features locked

User must be signed in before accessing any of the features, this was enabled using conditional rendering.

● Sign Up & Login

User must register when using the website for the first time and can then login. User stays logged in based on localStorage.

● Logout & Security

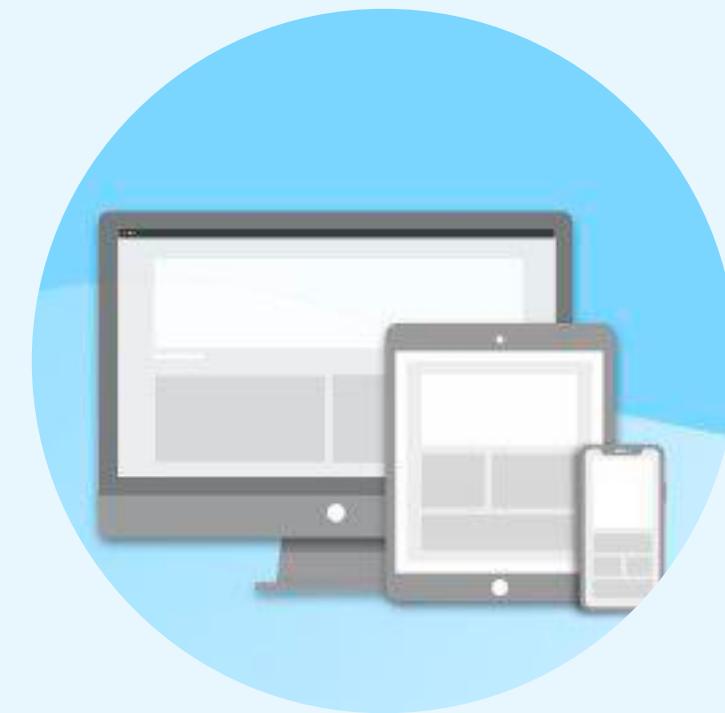
User can easily log out at any time via button that is featured in the header for peace of mind. Login is also secure using Bcrypt & JWT.

Overall Design



Chakra UI

The use of a front end package helps to optimise the user experience and minimise use of CSS.



Responsive

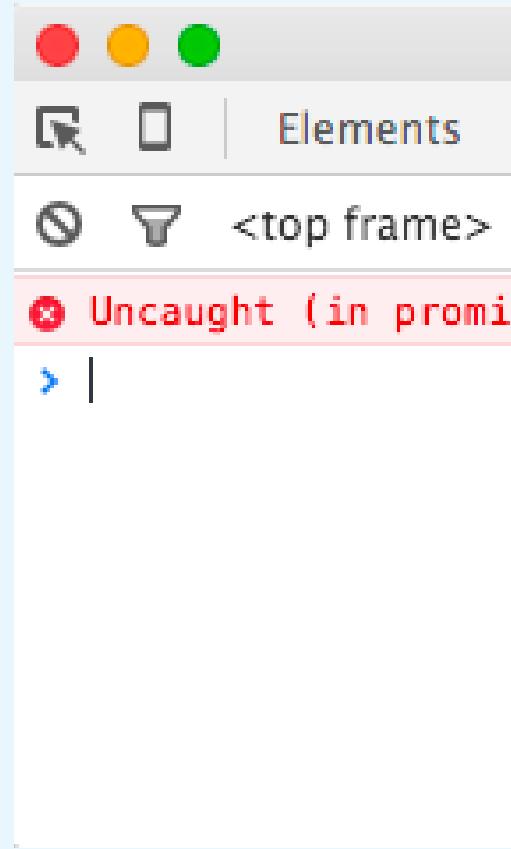
The quiz being responsive allows the user to test themselves on the go at any convenient time.



Clean & Simple

The design does not reinvent the wheel or feature any fancy design, it's just clear and easy to use.

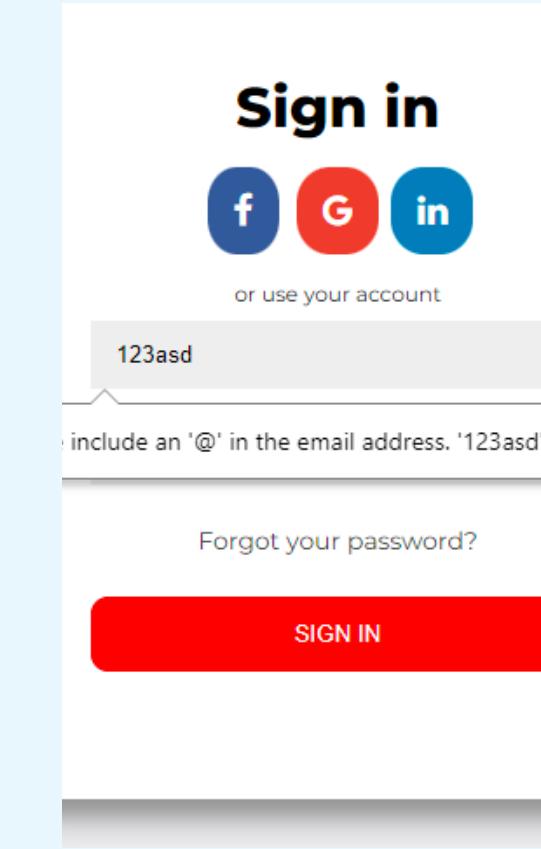
Future Improvements



Error Handling:
Some StartQuiz variations return 0 questions - this needs to be handled better.

A screenshot of the browser developer tools Elements tab showing a portion of the 'after.php' file. The code includes several if statements checking user scope and authentication status, returning true or false based on the conditions.

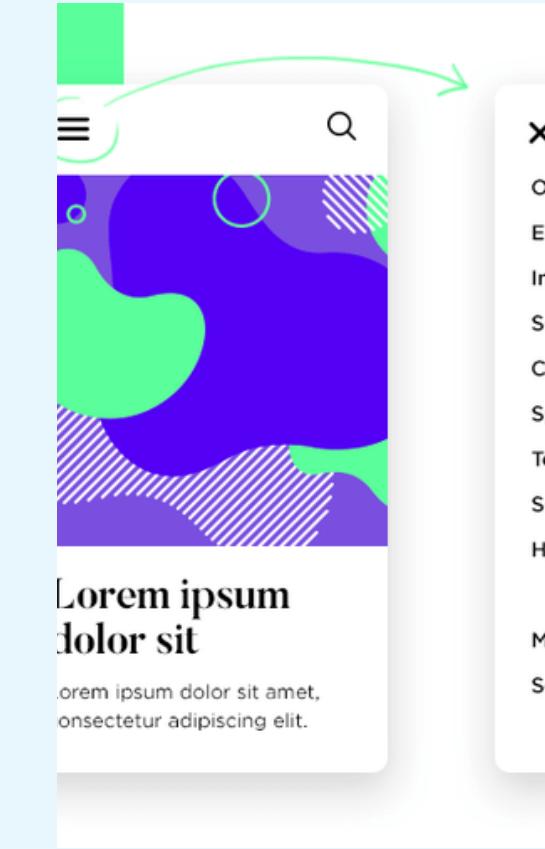
Cleaner Code:
Code could be cleaned up extensively, particularly the Results.js



Register/Login:
Better error handling, email address on register should be unique.

	Reputation Points	Badge
erfall	189	
a Landeta	115	
JRS	77	
ohnson	61	
ng	37	
	20	
	18	
	11	

Results:
Save quiz results so you can go back and look how you did in the past.

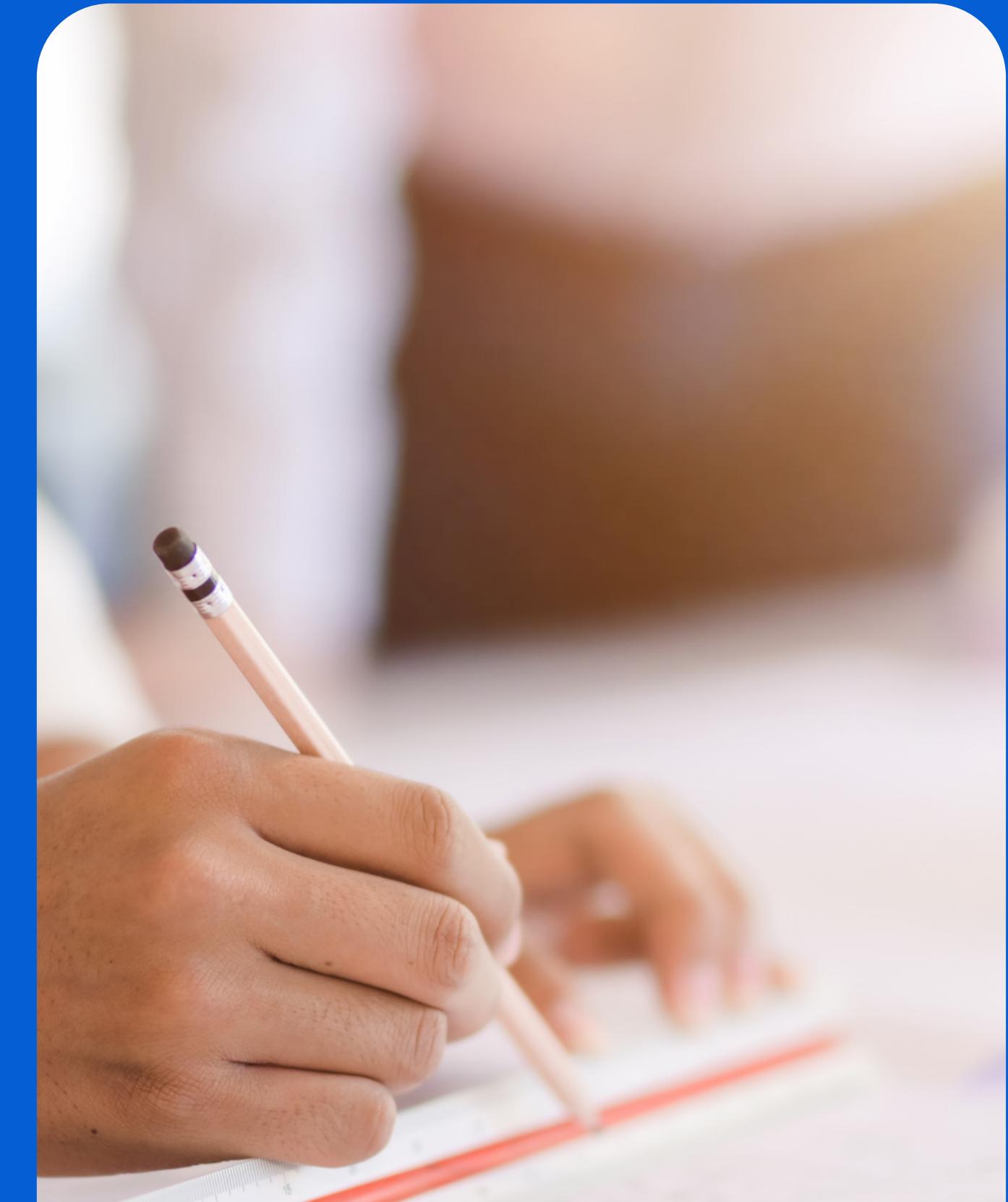


Tabs:
Not fully responsive on smaller screens, tuck away in a hamburger menu.



Features

- ✓ Whiteboard
- ✓ Quiz
- ✓ Header & Login
- ✓ Q&As/How-Tos ➤ Learning & revision
- ✓ Priorities List ➤ Reflecting & planning



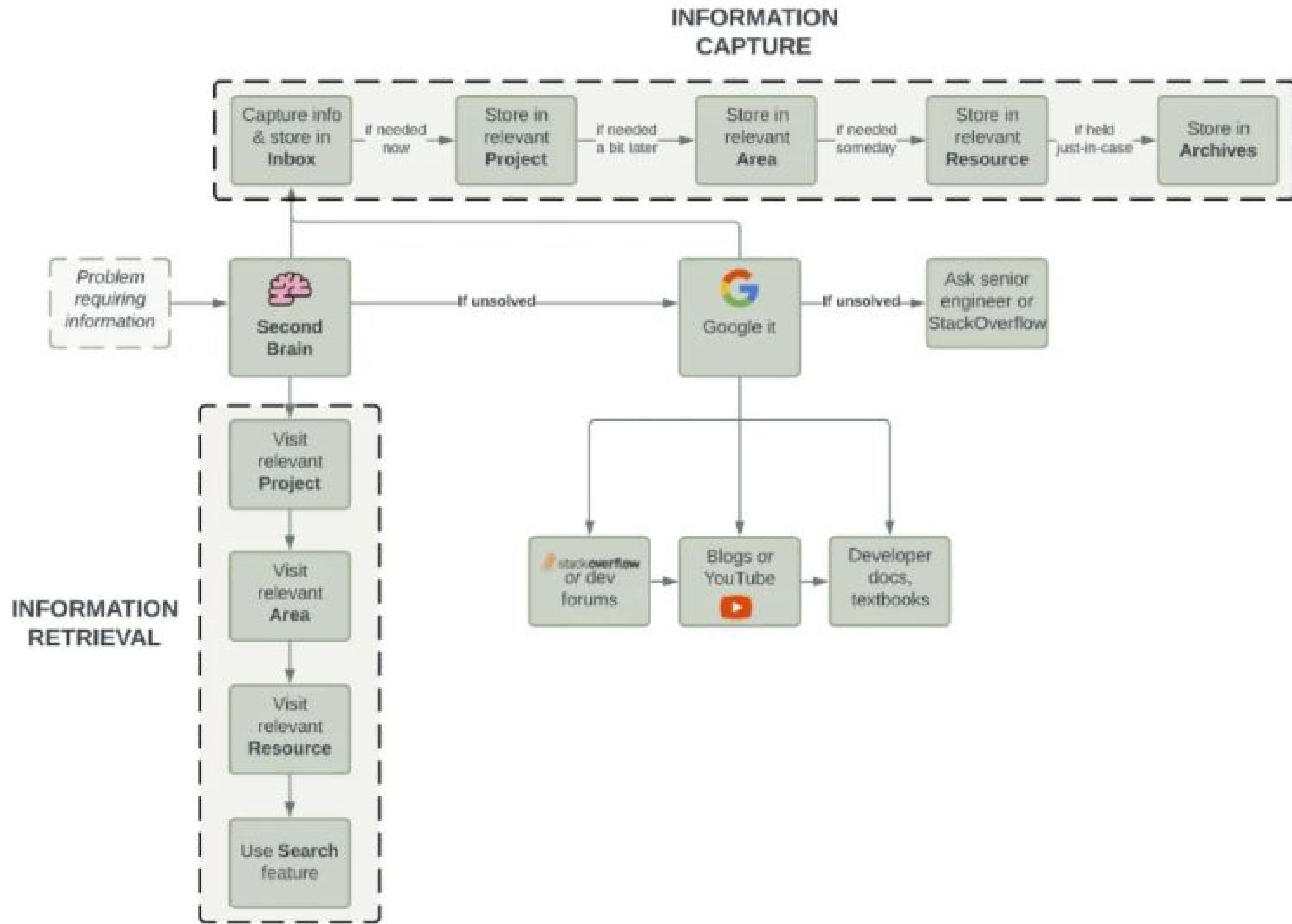
Q&As / How-Tos/ Priority List: The Inspiration

- Personal experience
- "Documentation-driven growth"
- Priorities list: tailored towards career-changers
- How Tos and Q &AS : Learn during the process of creating information, + bonus of having a personal reference library (explanations already in your own words or visual format)

Links to Related Articles

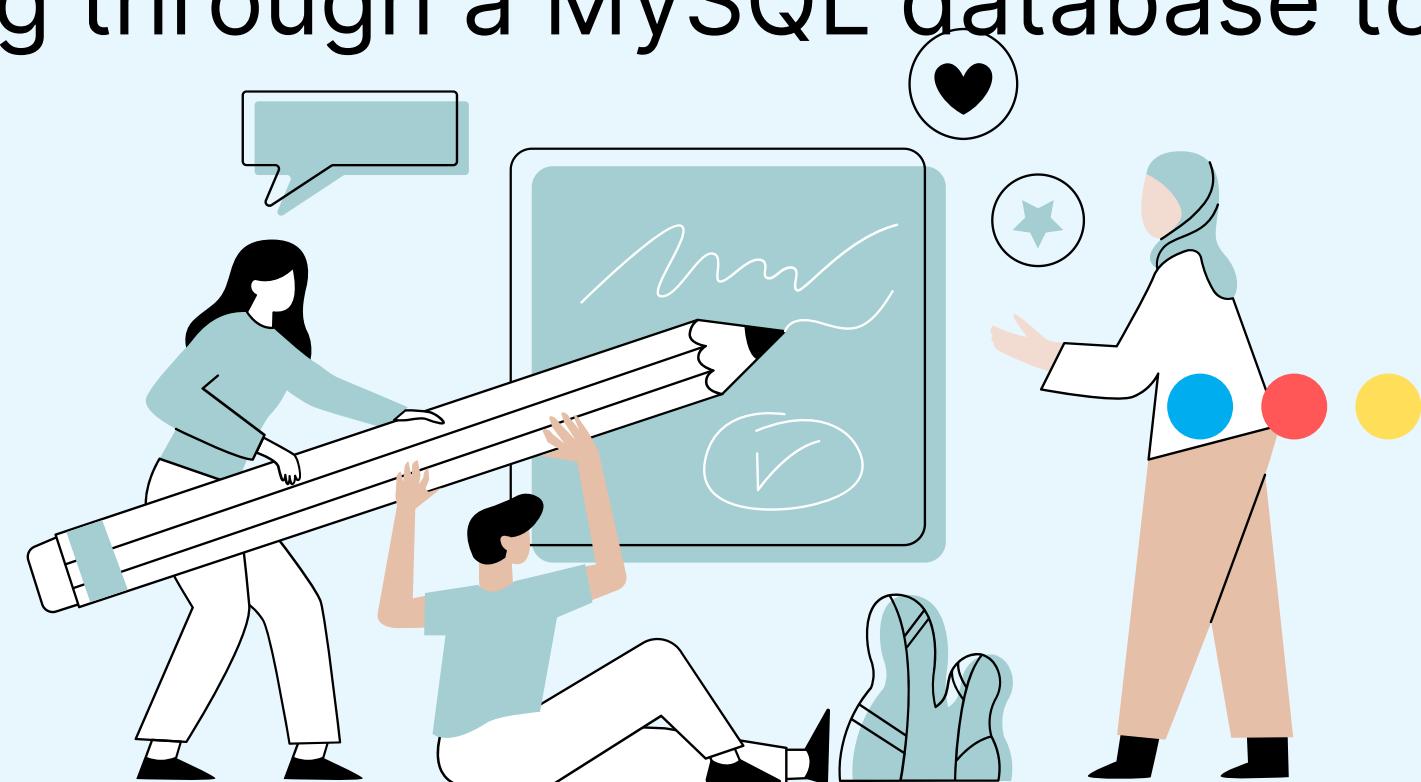
- "[Building Your Personal Documentation](#)" by Monica Powell
- "[How To Build A Second Brain As A Software Developer](#)" by Aseem Thakar
- "[The PARA Method](#)" - Forte Labs





Planning

- Sketched databases
- Sketched user-flow for each page
- Organized component proposals into similar front-end views / BE requirements and classified by difficulty level
- Made list of knowledge gaps
- Began with easiest component (list) to get used to calling & manipulating BE again
- Looked up tutorials to start filling in knowledge gaps (for this project, making a blog-style component, filtering through a MySQL database to return search results, linking DB tables)



Resources/inspo
• Monica Powell
• P.A.R.A method
(forthcoming)

YOUR SECOND BRAIN

→ DATABASE SCHEMA



1 TEACH A TOPIC

title
steps/how-to
resources
tags

2 Q+A'S

your question
answer
tags ✘ id
links ✘
date ✘

3 SNIPPETS

language
type
context
description
snippet
dependencies
tags •

- ie. front-end or back-end
- ie. purpose
- if necessary

4 TEMPLATES

type
location
template
description
tags •

- later iteration?
- ie. README, Client, etc.

4 RESOURCES

name/title
author
Category
URL
Status
tags •
rating

- tweet, vid, image, etc.
- reviewed or pending



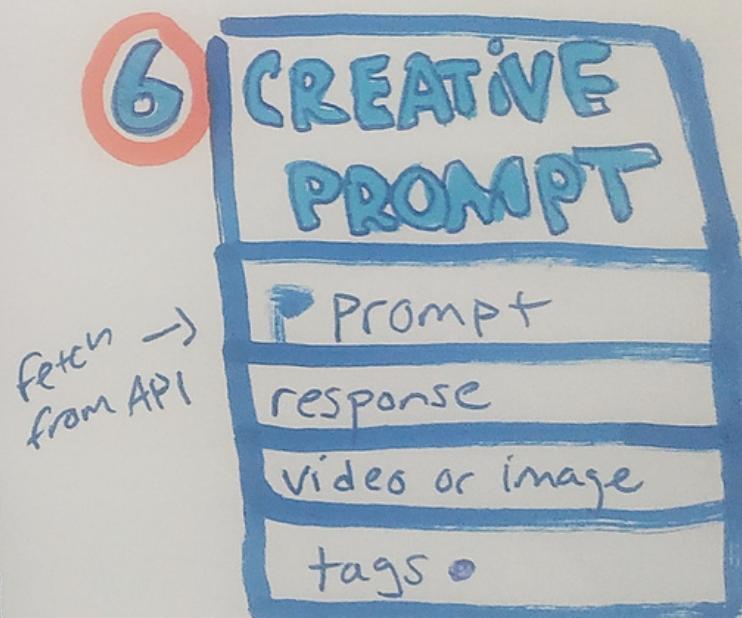
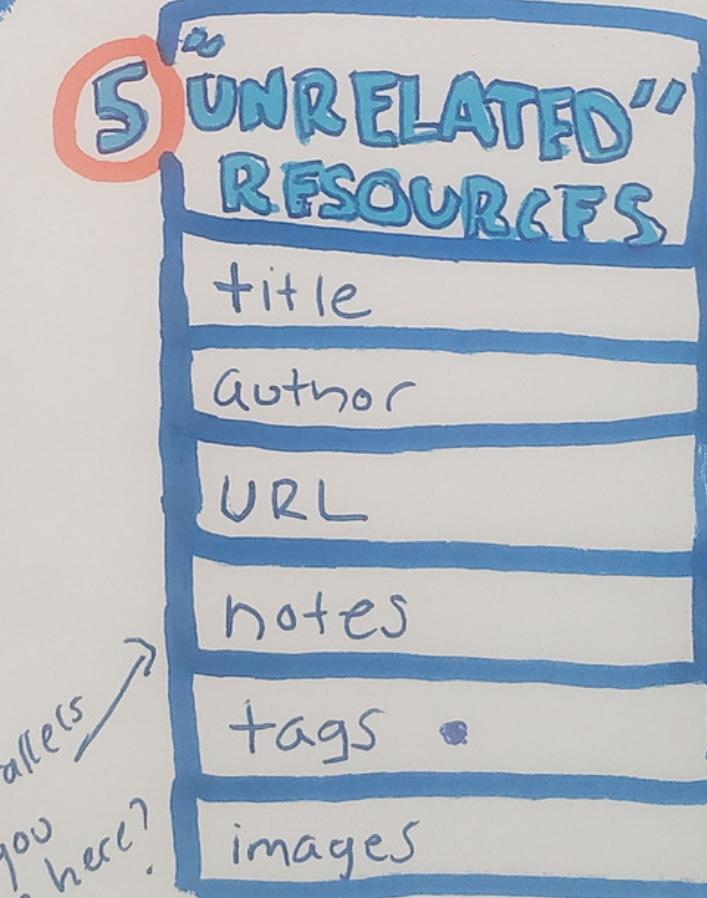
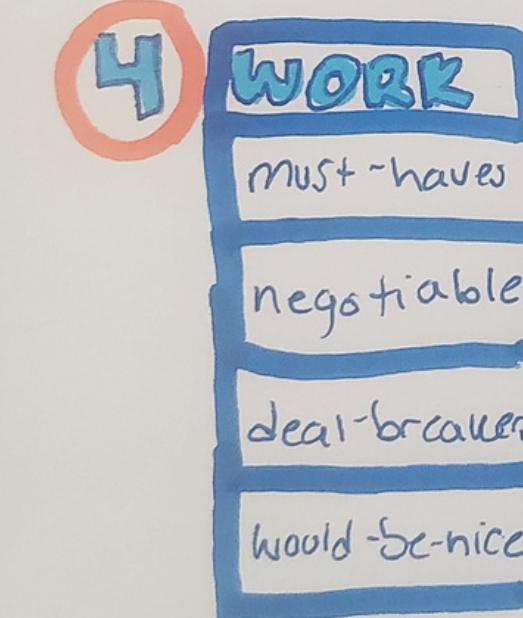
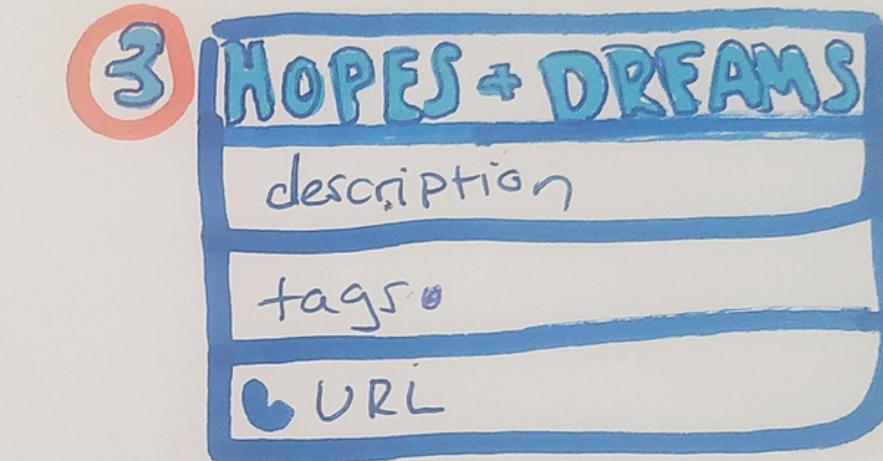
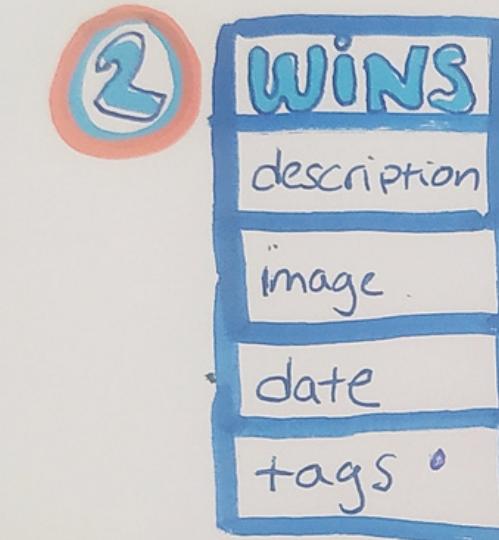
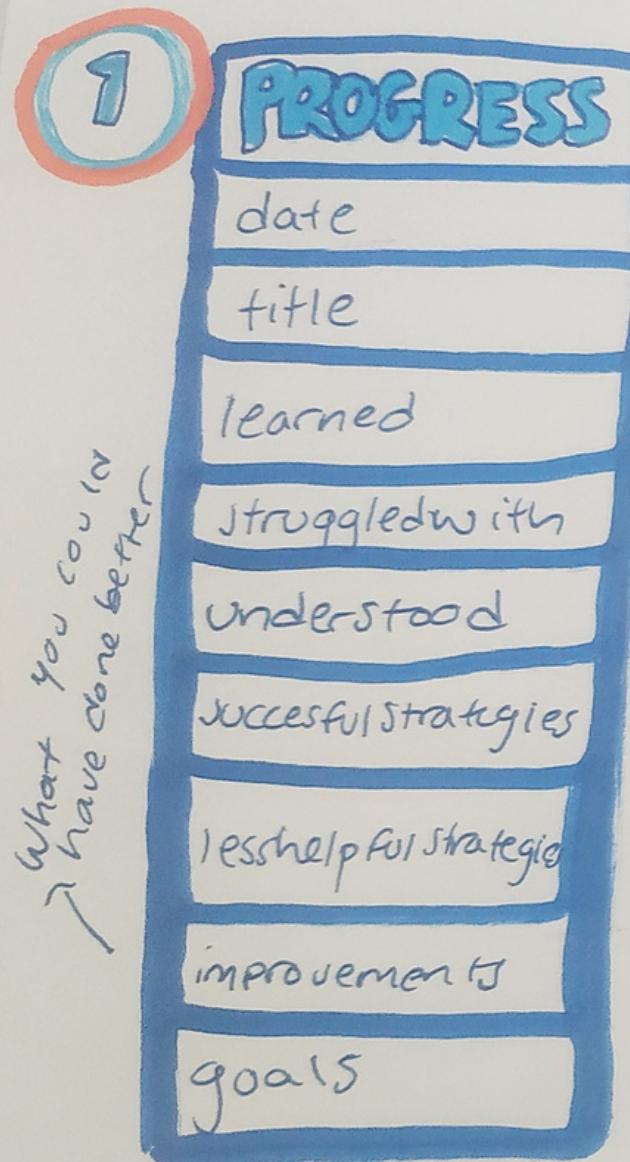
Resources / inspo

- Monica Powell README proj
- Meta cognition

PERSONAL / REFLECTION / CREATIVE AREA

→ DATABASE SCHEMA

• = recurring column



• is this personal
or professional?

- line if you
read interesting
industry article

COMPONENT UI FORMATS

BOARD STYLE

- Wins.js
- Hopes and dreams.js

Total : 2

Difficulty level : 2 

EDITABLE LIST STYLE

- Work Future.js 

Total : 1

Difficulty level : 1 

CARD VIEW STYLE

- Teach A Topic.js
- Q+A.js 
- Snippets.js
- Personal Coding Resources.js
- Unrelated Resources.js

Total : 5

Difficulty level : 3 

BLOG STYLE

- Learning Progress.js
- Personal Creative Posts.js

Total : 2

Difficulty level : 4 



Coding Process

- Worked feature by feature: created back-end tables in MySQL, created migrate variables, made API routes, tested routes in Postman, then once all of his was ok, followed UI sketch to create front-end
- Designed one component with testing for placement of html elements, but did not get to test other components due to lack of time / knowledge

#MYSQL Q&A— MIGRATE VARS

table	MySQL migrate command	Code Snippet
Q and A	let qAndAsSQL = """DROP TABLE if exists q_and_as; CREATE TABLE AUTO_INCREMENT, question TEXT, answer TEXT, ta con.query(qAndAsSQL, function (err, result) { if (err) throw err; console.log("Table creation `q_and_as` was suc console.log("Closing..."); });	app.get("/lesson-list/:searchTerms", (req, res) => { db(`SELECT * FROM teach_a_topic WHERE MATCH(topic_title, step_by_step) AGAINST ("\${req.params.) .then((results) => { res.send(results.data); }) .catch((err) => res.status(500).send(err)); });



Final Functionality

Q & A Component

- Users can add & delete interactive q & a cards
- Users can search full text of both question and answer cards

Work Priorities Component

- Users can add list items by category
- List is saved to BE and can be converted to checklist view

How Tos Component

- Blog style rendering
- User posts sorted and displayed by date
- All posts are searchable by title and content
- Add new post button that lets user add a new post with rich text editor



The Codebase



```
if ($window.scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) < header1_initialPadding) {
        header1.css('padding-top', '' + $window.scrollTop() - header1_initialDistance);
    }
} else {
    header1.css('padding-top', '' + header1_initialPadding + 'px');
}

if ($window.scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) < header2_initialPadding) {
        header2.css('padding-top', '' + $window.scrollTop() - header2_initialDistance);
    }
} else {
    header2.css('padding-top', '' + header2_initialPadding + 'px');
}
```

Back-End and Front-End

Back-End

- 8 database tables created (6 in use, but could have been 3)
- API routes for get, post, delete, and search added for each

Front-End

- 14 components total
- Tests implemented on one component

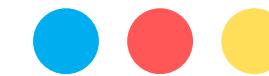
```
//FULL TEXT SEARCH THROUGH THE Q & A CARDS
app.get("/q-and-as-list-search/:searchTerms", (req, res) => {
  db(
    `SELECT * FROM q_and_as WHERE MATCH(question, answer) AGAINST ("${req.params.searchTerms}")`
  )
    .then((results) => {
      res.send(results.data);
    })
    .catch((err) => res.status(500).send(err));
});

/*API ROUTES FOR THE TEACH A TOPIC TABLE */
//GET THE FULL LIST OF TOPIC ENTRIES
app.get("/lesson-list", (req, res) => {
```

```
  <!-- GET THE FULL LIST OF TOPIC ENTRIES -->
  res.send([
    {
      id: 1,
      title: "How to Backup Your Brain"
    },
    {
      id: 2,
      title: "How to Create a Personal Reflection Area"
    },
    {
      id: 3,
      title: "How to Play a Quiz"
    }
  ]);
});
```

- ✓ components
 - ✓ Backup Brain
 - # HowTos.css
 - JS HowTos.js
 - # MyQandAs.css
 - JS MyQandAs.js
 - ✓ Personal Reflection Area
 - # WorkReqsList.css
 - JS WorkReqsList.js
 - JS WorkReqsList.test.js
 - ✓ quiz
 - JS Play.js
 - # Quiz.css
 - JS Quiz.js
 - JS Results.js
 - ✓ Seonaid Components Navigation
 - # DocumentationHomePage.css
 - JS DocumentationHomePage.js
 - # ReflectionAreaHomePage.css
 - JS ReflectionAreaHomePage.js
 - ✓ whiteboard
 - > images
 - JS toolbar.js
 - # Whiteboard.css
 - JS Whiteboard2.js
 - IMG Codify background.png
 - IMG Codify.png
 - JS Footer.js
 - JS Header.js
 - JS Home.js

Review & Reflection





Challenges



Group

- Github conflicts
- Different timings & time commitments
- Deploying
- Random warnings in the console

Imogen

- Passing data between siblings:
Header/tabIndex & Login/loginStatus
- Conditionally rendering components within components
- Centering things
- Keeping code clean
- All help online is about class components

Seonaid

- Being overly-ambitious about the number of database tables
- Linking tables in the back-end
- Search filtering
- Adding a text editor
- Styling

Alex

- Working out HTML5 Canvas and use of mouse coordinates
- Math formulas, moving circles around the canvas
- Misaligned cursor and drawings



Successes



Group

- Effective teamwork!
- Helping eachother debug
- Github / using staging branch

Imogen

- Results page - score/comparison logic and accordion styling
- Full login & sign up functionality
- No Git conflicts
- Debugging
- Design/using front end package
- Prop expert now

Seonaid

- Adding text editor
- Storing html in BE and converting for display
- Planning API routes / implementing search route
- Increasing understanding of database management

Alex

- Combining several different tools into one whiteboard
- Introducing some tools without tutorials/ major Googling!
- Appreciation of what actually goes into coding a whiteboard from scratch - I took MS Paint for granted

Acknowledgements and Thanks

EACH OTHER! 

Easy group dynamic,
very effectively 'divided
and conquered'

SOFIA & ALEX 

Tab fixing MAGIC

CODEOP /
OUR CLASS! 

Proud of all of us!



Future Features

1.

INSTRUCTOR LOGINS

Adding students to cohort list, assigning homework

2.

PUBLIC "CLASSROOM"

An area where students can share information and resources, and contribute to a collaborative cheat-sheet

3.

REFLECTION BLOG

An area where students can regularly upload entries reflecting on what they've learned, what strategies worked for them, and set future learning goals

4.

RESOURCE COLLECTION

A place where students can upload tutorials, videos, articles, and links to resources

Future Features

5.

RESOURCES BOARD

Students can upload articles, videos, etc. of concepts not directly related to programming, but which have deepened or expanded their understanding of the topic

6.

TAGS

Linking back-end tables with tags in order to facilitate a global search of all resources and student entries

7.

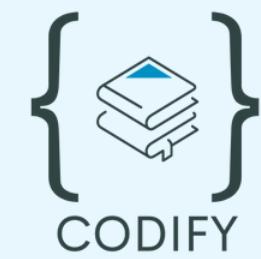
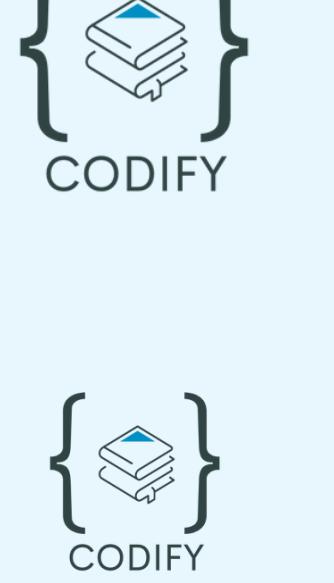
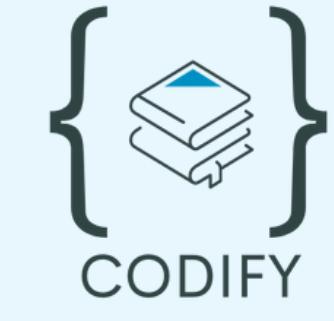
CODE SNIPPETS

An area to save frequently used code snippets and templates

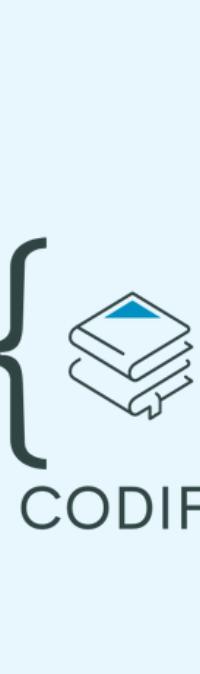
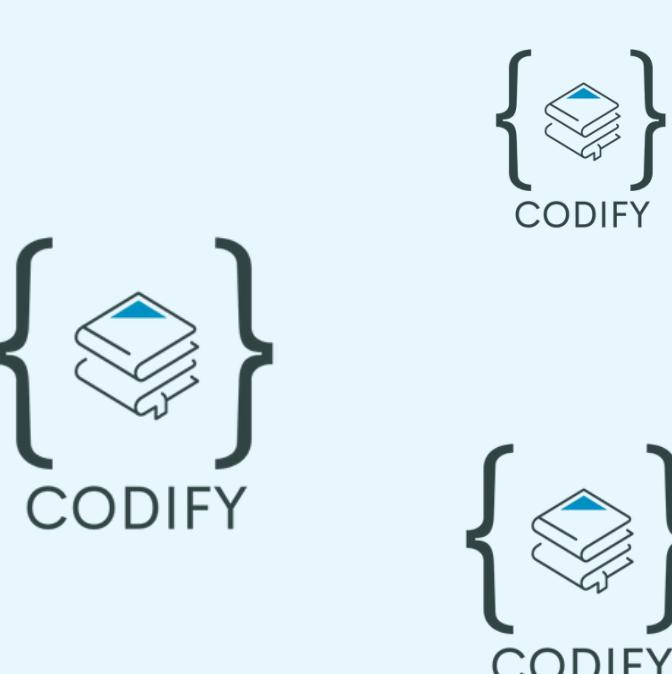
8.

SEARCH FUNCTIONS

More advanced styling and search methods



THANK YOU



Any questions?