SPORTS TRIVIA WEBAPP

Samantha Tamakloe-Bonaca

Finny Chen

AGENDA

01

App
Overview
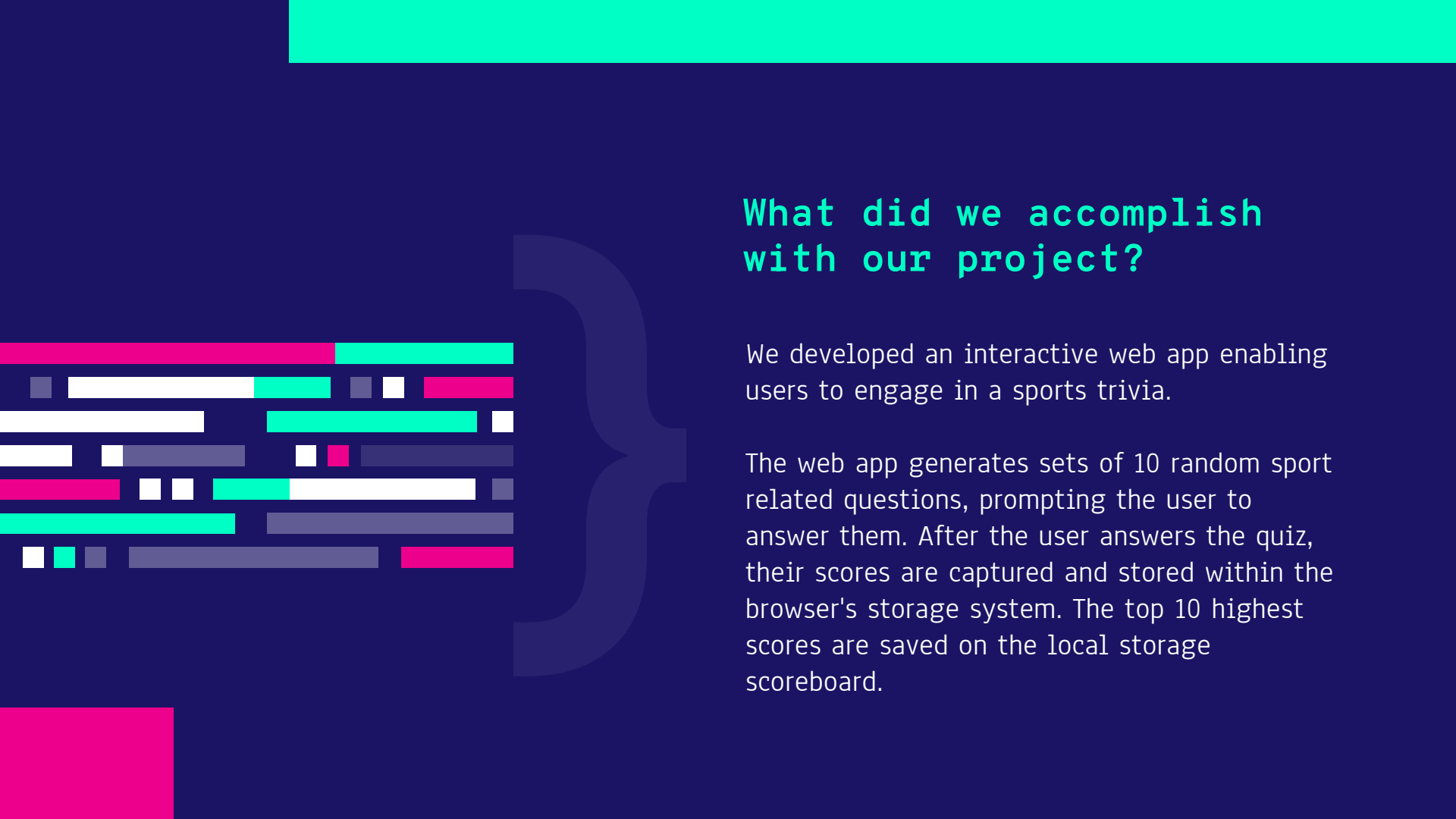
02

App Demo

03

Moscow
Method

04

Project
Experience

01

App Overview

What did we accomplish with our project?

We developed an interactive web app enabling users to engage in a sports trivia.

The web app generates sets of 10 random sport related questions, prompting the user to answer them. After the user answers the quiz, their scores are captured and stored within the browser's storage system. The top 10 highest scores are saved on the local storage scoreboard.

02

App Demo

Replit -> https://replit.com/@samanthabonaca/Sports-trivia-App

https://9686dc4c-86ea-49e7-a2b3-2d1e5ab044fd-00-10x7foo54gxdb.worf.replit.dev/

03

Moscow Method

HOW WE PLANNED + EXECUTED OUR PROJECT

Moscow method

A project management technique

Must Have
all the requirements that are necessary for the successful completion of the project

Should Have
Requirements that are important for the project completion but not necessary

Won't Have
Requirements that are recognised as not a priority for the projects timeframe

Could Have
Requirements that are nice to have, but will not have much of an impact if not included

PRIORITISED LIST OF REQUIREMENTS (MoSCoW method)

MUST HAVE
- Question Generation
- User Answer Submission
- Scoring System
- Local Storage

SHOULD HAVE
- User Friendly UI
- Responsive Design

WON'T HAVE
- Multiplayer Mode
- Social Media Integration

COULD HAVE
- Category Selection:
- Timer Functionality:

04

Project Experience

HOW DID YOU SHARE YOUR CODE?

REPLIT

VSC

Initially we started with the sharing extension on **VSC (Live Share)** and then migrated our project to **Replit**, as with the VSC sharing extension we had to work on the code simultaneously which didn't work out with our individual schedules.

BIGGEST CHALLENGE?

Dividing and allocating the work.

Work was split evenly amongst the two of us, but we had areas of coding activities that overlapped and so when one person made changes to the code, it was difficult to retrace our steps.
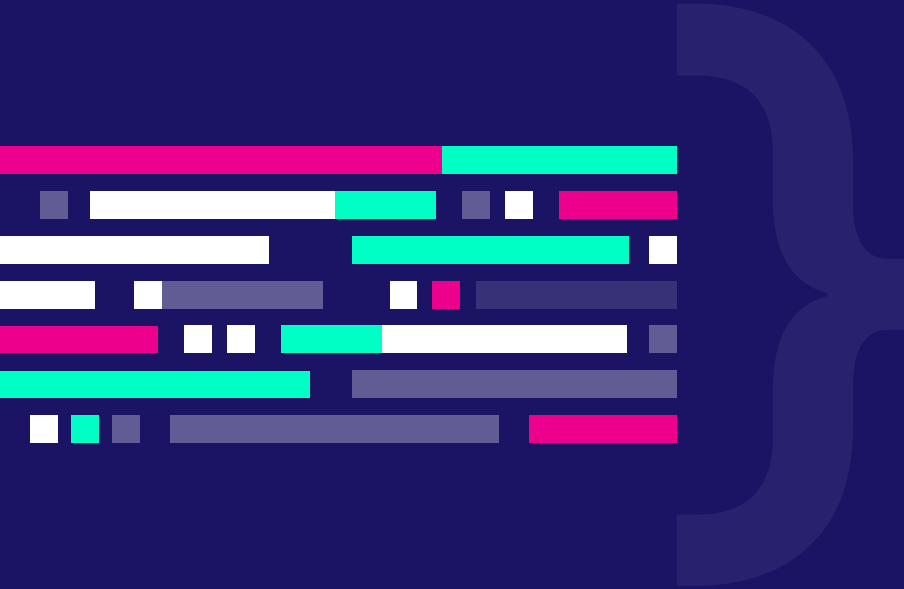
WHAT WENT WRONG?

a. When we first migrated our project to replit, the initial set-up for it was difficult, as if there is no file labelled as HTML file labelled as 'Index.html' replit will not index the project and as a result it cannot be found when trying to preview the project - 404 error

b. Minor coding errors e.g. accidently misspelling a word in the HTML file which means it couldn't be found when using the getElementByID query in Javascript

c. Forgetting to add </> closing tags for HTML elements e.g. forgot to add a </title> tag which resulted in our HTML index file not loading properly in the console.

EXAMPLES OF CODE ERRORS

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible"
content="ie=edge" />
    <title>Sports Trivia App <title>
    <link rel="stylesheet" href="app.css" />
  </head>
  <body>
</html>
```

Title tag not closed (</title>)

WHAT WENT WRONG?

a. Missing a line of code which resulted in our high score JS element displaying as null on the front-end of the trivia application
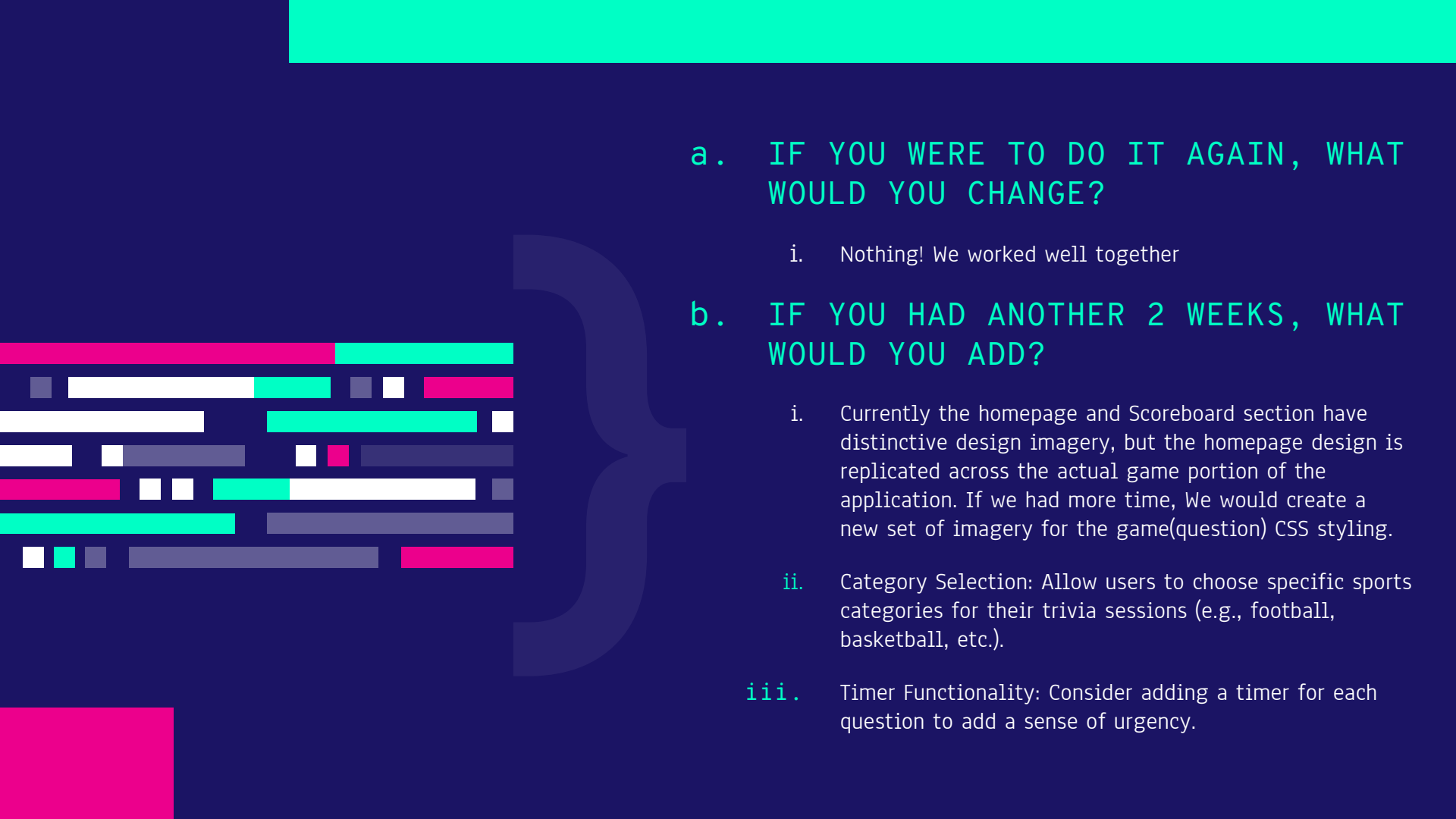
EXAMPLES OF CODE ERRORS

```
getNewQuestion = () => {
    if (availableQuestions.length === 0 ||
questionCounter >= MAX_QUESTIONS) {
        localStorage.setItem('mostRecentScore', score);
        //go to the end page
        return window.location.assign('/end.html');
    }
```

Missing line of code

WHAT WENT BETTER THAN EXPECTED?

a. Implementing the open trivia API

b. Time management - We both work full-time but we scheduled regular meetings and project checkpoints, and so we managed to complete the project much faster than we originally anticipated

a. IF YOU WERE TO DO IT AGAIN, WHAT WOULD YOU CHANGE?

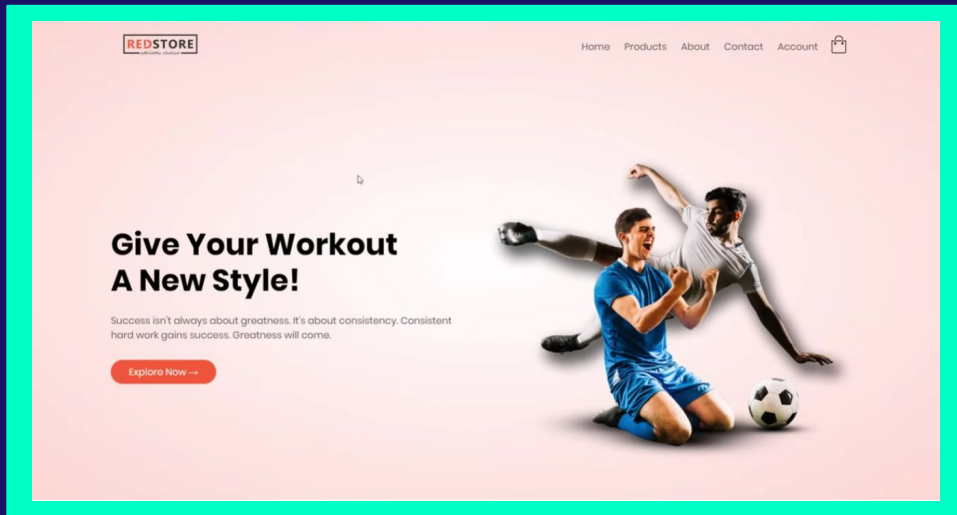 i. Nothing! We worked well together

b. IF YOU HAD ANOTHER 2 WEEKS, WHAT WOULD YOU ADD?

  i. Currently the homepage and Scoreboard section have distinctive design imagery, but the homepage design is replicated across the actual game portion of the application. If we had more time, We would create a new set of imagery for the game(question) CSS styling.

  ii. Category Selection: Allow users to choose specific sports categories for their trivia sessions (e.g., football, basketball, etc.).

  iii. Timer Functionality: Consider adding a timer for each question to add a sense of urgency.

NEXT PROJECT AFTER THE COURSE?

This time around we focused on building a web application.

Next time around, it might be nice to focus on creating a fully functioning website, such as an ecommerce site e.g. Shopify store.

THANKS!