

The background of the slide is a vibrant, abstract network pattern. It features a dense web of thin, light-colored lines connecting various circular nodes of different sizes. The color gradient transitions from a bright, warm orange and yellow on the left side to a deep, rich purple on the right side. The nodes and lines are semi-transparent, creating a layered, ethereal effect.

Book Connect Website

Zamavundla Samkelisiwe Duze

Agenda

Introduction

What was wrong

What was fixed

Summary





Introduction

- Book connect is an online platform with various books for a reader from fiction to non fiction.
- It is accessible to every user and it allows you to preview the book, before you can dive into it and see if it is what you want...

What was wrong

- ❖ It was challenging to fix the final layout of the app because the first developer had not initially commented this code. Function comments must have descriptions, arguments and logic to make the code more readable and facilitate future debugging.
- ❖ Functions must be declared with the `const` or `let` keyword, which makes it simpler to find bugs and ensures proper operation. Using a function might be preferable when writing some types of code, such as that which creates drop-down menu items. Functions were used to construct HTML fragments, which had the advantage of being reusable.
- ❖ As seen on `originalcode.js`, code was randomly ordered, with code for related functions placed far apart.
- ❖ The code has been reorganized and divided into sections based on functions (by using event listeners and also extracting the information from `html`) to increase readability.
- ❖ The dark/light mode symbol is deceptive, according to reviewers. For users who are using the app for the first time, it may be perplexing because it seems more like a login button.

The way to get
started is to quit
talking and
begin doing.

Walt Disney



What was fixed

READABILITY OF THE CODE

- ❖ ARRANGE CODE BY FUNCTIONALITY AND REMOVE UNUSED/UNNECESSARY CODE. .
- ❖ SEPARATE CODE INTO FILES, LONGER CODE IS MOVED TO ITS OWN FILE SO THAT THE PERSON READING IT, WON'T BE CONFUSED
- ❖ MAKE DESCRIPTIVE VARIABLE NAMES SINCE INITIALLY THERE WERE JUST RANDOM WRITING.
- ❖ MAKES IT EASIER TO TRACK THE PURPOSE OF VARIABLES BY USING DOM ELEMENTS.
- ❖ USE RESTRUCTURINGS TO ACCESS OBJECTS & ARRAYS.
- ❖ ALSO BREAK UP CODE USING THE RUNNING PROCESS STRUCTURE



CODE BREAK UP FOR FUNCTIONING

PAGE STRUCTURE

- ❖ View books and previews with title, author, year & summary.
- ❖ See 36 books at a time.
- ❖ Scroll down using a button to show more.

DISPLAY SETTINGS

- ❖ Load page with system dark & light mode settings.
- ❖ Toggle dark & light mode setting on page.

BOOK SEARCH

- ❖ Search for books by title, author & genre.
- ❖ Do a strict search using the following parameters (title, author & genre.)

Maintainability

❖ Add JSDoc comments for all functions.

❖ Add comments for variables with type.

```
/**
 * Creates a slice of specified length from the database of books.
 *
 * @param {array} array is an object array of all books with properties.
 * @param {number} start is a number denoting where to start extract the array number of books.
 * @param {number} end is a number denoting where to end extracting the array number of books.
 * @return {HTMLElement}
 */
export const createPreviewsFragment = (array , start, end) => {

  const booksExtract = array.slice(start, end);

  let previewFragment = document.createDocumentFragment();

  for (let i = 0; i < booksExtract.length; i++){

    let { author, image, title, id } = booksExtract[i];
    author = authors[author];

    const preview = {
      author,
      id,
      image,
```


Variables Names

- This helps with selecting the right CSS style when getting the properties to work on Event listeners to click between the themes and toggling around the website

```
120 // This code checks darkmode/lightmode settings of user's system and assign them to the
121 html.display.settingsTheme.value = window.matchMedia && window.matchMedia('(prefers-color
    'night' : 'day';
122
123 let v = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches?
124
125 html.display.settingButton.addEventListener('click', () => {
126     html.display.settingsOverlay.showModal();
127 });
128
129 html.display.settingsCancel.addEventListener('click', () => {
130     html.display.settingsOverlay.close();
131     html.display.settingsSubmit.reset();
132 });
133
134 const css = {
135     day : ['255, 255, 255', '10, 10, 20'],
136     night: ['10, 10, 20', '255, 255, 255']
137 };
138
139 html.display.settingsSubmit.addEventListener('submit', (event) => {
140     event.preventDefault();
141
142     const formData = new FormData(event.target);
143
144     const selected = Object.fromEntries(formData);
145
146     if (selected.theme === 'night') {
147         document.documentElement.style.setProperty('--color-light', css[selected.theme][0]);
148         document.documentElement.style.setProperty('--color-dark', css[selected.theme][1]);
149     } else if (selected.theme === 'day') {
150         document.documentElement.style.setProperty('--color-light', css[selected.theme][0]);
151         document.documentElement.style.setProperty('--color-dark', css[selected.theme][1]);
152     }
153 });
```

Object Restructuring

- ❖ Restructuring by using handlers.js folder helps with creating fragments which will be imported to script.js and avoid long code which will end up getting lost in it whilst debugging

```
//  
export const createPreviewsFragment = (array , start, end) => {  
  
  const booksExtract = array.slice(start, end);  
  
  let previewFragment = document.createDocumentFragment();  
  
  for (let i = 0; i < booksExtract.length; i++){  
  
    let { author, image, title, id } = booksExtract[i];  
    author = authors[author];  
  
    const preview = {  
      author,  
      id,  
      image,  
      title,  
    };  
    previewFragment.appendChild(createPreview(preview));  
  }  
}
```

```
// handlers.js  
import { createPreviewsFragment } from './script.js';  
  
const books = [...];  
const start = 0;  
const end = 10;  
  
const previewFragment = createPreviewsFragment(books, start, end);  
  
const container = document.querySelector('.container');  
container.appendChild(previewFragment);
```

Performance

- ❖ Using the import & export tags to access code between files..
- ❖ Use const or let to declare variables, to avoid variables to be overwritten, resulting in bugs.
- ❖ Remove unused/unnecessary code for better debugging.
- ❖ Store DOM elements in object for better calling when it comes to running the website.
- ❖ Use defer to load JavaScript last.
- ❖ Remove unused variables.
- ❖ Test code to check if there are no errors, use the console log. Some error could have been fixed by looking console log errors which makes it easier to point them out and debug.

Performance (Selecting elements)

- ❖ Store DOM elements in object as stored `.querySelector` variables rather than calling `.querySelector` for the same element multiple times.

```
// this code deals with the view of the app (which includes search button and descriptions of books to be visible) by
// selecting data from html and makes the viewing and scrolling have a smoother experience//
export const html = {
  view: {
    mainHtml: document.querySelector('[data-list-items]'),
  },
  scroll: {
    moreButton: document.querySelector('[data-list-button]'),
  },
  preview: {
    summaryList: document.querySelectorAll('[data-preview]'),
    summaryOverlay: document.querySelector('[data-list-active]'),
    summaryBlur: document.querySelector('[data-list-blur]'),
    summaryImage: document.querySelector('[data-list-image]'),
    summaryTitle: document.querySelector('[data-list-title]'),
    summarySubTitle: document.querySelector('[data-list-subtitle]'),
    summaryDescription: document.querySelector('[data-list-description]'),
    summaryClose: document.querySelector('[data-list-close]'),
  },
  display: {
    settingsOverlay: document.querySelector('[data-settings-overlay]'),
    settingButton: document.querySelector('[data-header-settings]'),
    settingsTheme: document.querySelector('[data-settings-theme]'),
    settingsCancel: document.querySelector('[data-settings-cancel]'),
    settingsSubmit: document.querySelector('[data-settings-form]'),
  },
  search: {
    searchCancel: document.querySelector('[data-search-cancel]'),
    searchButton: document.querySelector('[data-header-search]'),
    searchOverlay: document.querySelector('[data-search-overlay]'),
    searchTitle: document.querySelector('[data-search-title]')
  }
}
```


Performance (JavaScript Loading)

- ❖ This helps with loading JavaScript at the end and importing data from a different script and make it perform a certain way on another script for full user-friendly experience to the user.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Book Connect</title>

    <link rel="apple-touch-icon" sizes="180x180" href="/meta/apple-touch-icon.png">
    <link rel="icon" type="image/png" sizes="32x32" href="/meta/favicon-32x32.png">
    <link rel="icon" type="image/png" sizes="16x16" href="/meta/favicon-16x16.png">
    <link rel="manifest" href="/meta/manifest.json">
    <link rel="mask-icon" href="/meta/safari-pinned-tab.svg" color="#0096ff">
    <link rel="shortcut icon" href="/meta/favicon.ico">
    <meta name="msapplication-TileColor" content="#0a0a14">
    <meta name="msapplication-config" content="/meta/browserconfig.xml">
    <meta name="theme-color" content="#0a0a14">

    <link rel="stylesheet" href="/styles.css" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

    <link
      href="https://fonts.googleapis.com/css2?family=Roboto:wght@500;700&display=swap"
      rel="stylesheet"
    />

    <script src="/Interactive-Web-Apps/IWA_19/book-connect-main/scripts.js" defer type="module"></script>
  </head>
  <body>
    <!--
    * /
    // Import data from data.js and handlers.
    > import { BOOKS_PER_PAGE, authors, genres, books } from "/data.js"
    import { html, } from "/handlers.js";
    import { createPreviewsFragment, updateRemaining, showBookDescription, showPreview, cre

    const range = [0, 36]
    const matches = books; // Try to convert matches to an indepenedent copy.
    let page = 1;

    /* This code checks if a book is not empty/undefined, and if it is an array.
    * Therefore it must be changed '&&' to || because both are invalid.
    */
    if (!books || !Array.isArray(books)) throw new Error('Source required');

    /* range is an array to check if range is within 0 - 36.
    * Change "< 2" to "=== 2" to avoid future errors.
    */
    if (!range && range.length === 2) throw new Error('Range must be an array with two numb
```



Summary

<https://www.loom.com/share/400b94d4343e4eb0bd9d7cd9818148f5?t=166>

This video shows that the user story was met.



Thank You

Zamavundla Samkelisiwe Duze

zsdouzee8@gmail.com

GitHub for this project IWA_I9

https://github.com/Zamavundla/ZAMDUZI20_FTC230I_NWABISA_ZamavundlaDuze_IWA