CFG Kickstarter
JavaScript

August 2024

Budget App

Creating a finance web app

Group Members:

Nina Thomas
Reeshel Rodriguez
Soraya Momoniat

Motivation

“94% of women believe they’ll be personally responsible for their finances at some point in their lives...”

“48 % of women are confident about their finances..””

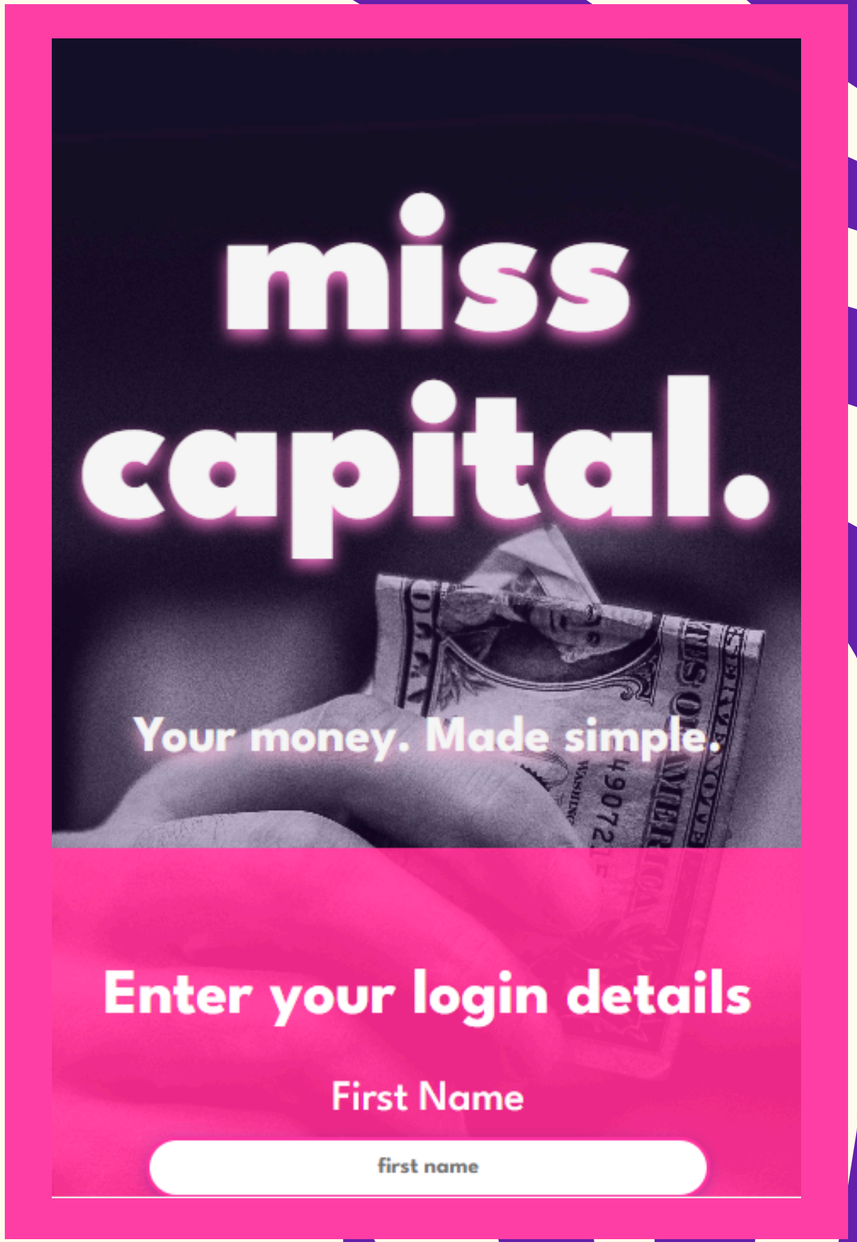
“Only 28% feel empowered to act.
Their biggest regret is not saving and investing sooner in life.”

-Bank of America

LETS TALK

about

- Our team
- Project Overview
- Collaboration Tools
- Code Demo
- Functions Used
- Project Retro: Issues Encountered
- Project Retro: What Went Well
- Future Improvements



miss capitalteam

Nina Thomas
Healthcare

Reeshel Rodrigues
Finance

Soraya Momoniat
Event Operations

project overview

SHORT BRIEF

In this project, we created a budget app called Miss Capital. We endeavoured to do the following:

Project Goals

- Get input from the user on a web page
- Make changes to the HTML and CSS using JavaScript
- Use an event to trigger a change to a web page

Bonus

- Add HTML elements using the DOM
- Remove HTML elements from the webpage using DOM
- Use other DOM events
- Use of arrays, loops and/or conditional logic in JS

collab tools

During our project we used various tools to help us communicate as a group and also progress with our code as most of us were relatively beginners!

Slack - to communicate during sessions and to arrange huddles.

Google Docs to make meeting notes for those who couldn't attend. Place updates on what was discussed

GitHub Kanban to keep track of and prioritize tasks

YouTube - to grasp some concepts better such as DOM manipulation
Scrimba - interactive JS course for hands on examples

CodePen/VS Code to work individually.
Replit to collaborate on code in real time as a group

Stack Overflow,W3 Schools and ChatGPT to help fix errors and find methods to help us add functionality

needs

For our budgeting app we had a few basic criteria:

- Input values for income and expenses
- Label those values
- Add and substract these values and create a new total
- List each item in a table format

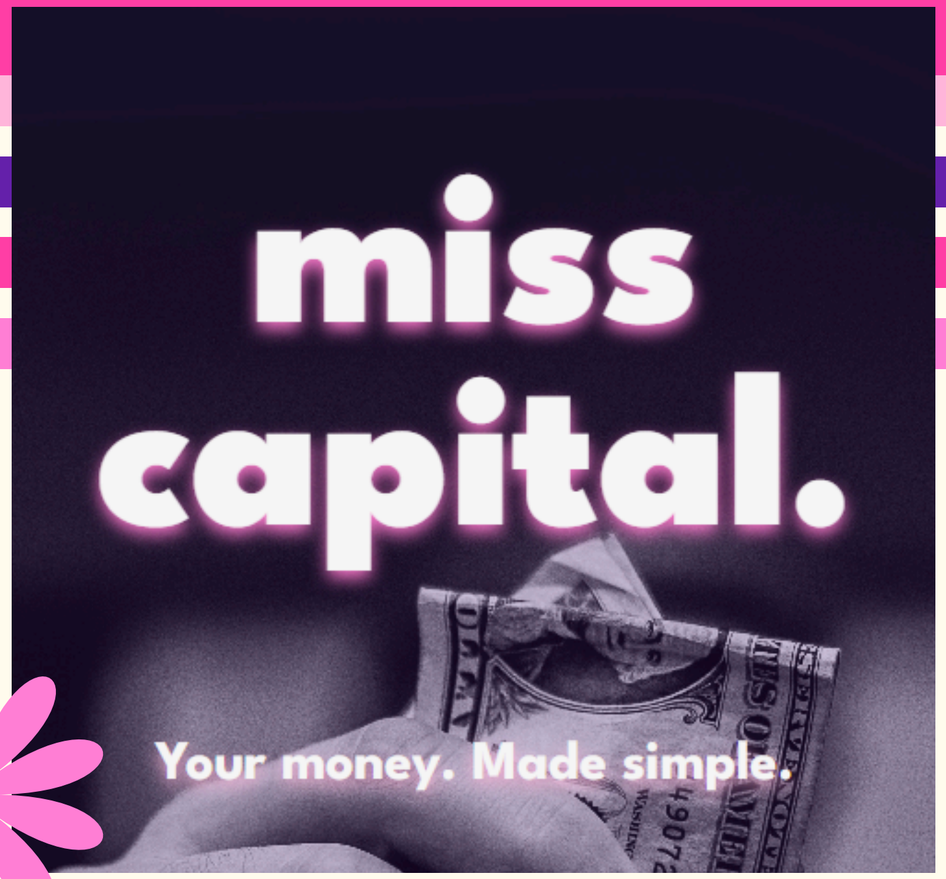
wants

Some of the features we wanted:

- Login feature
- Include a personalised greeting on entry
- Edit and delete the items listed
- Include the current date to each entry
- User friendly GUI

app demo!

CodePen



thefunctions

User Authenitcation (Login)

function login():

get username and password from input fields

convert username to lowercase; password remains case -sensitive

if username is "name" and password is "Password":

unhide the main app

hide the login form

call the **greeting** function

else:

show an alert with "Your username or password is incorrect"

the functions

```
// Add Income to the Income Column
function addIncome() {
    let incomeName = document.getElementById("income").value.toUpperCase();
    let incomeAmount = document.getElementById("amount").value;
    let currentDate = getCurrentDate();
    let errorMessage = document.getElementsByClassName("error")[0];
    let monthYear = getMonthYear();

    if (!incomeName || !incomeAmount) {
        errorMessage.style.display = "block";
    } else {
        errorMessage.style.display = "none";
        if (!isTitleSet) {
      listTitle.style.display = "block"
      listTitle.textContent += ` for ${monthYear}`
      isTitleSet = true;
      }

        let incomeListContainer = document.getElementById("income-list");

        // Create a new list item
        let newItem = document.createElement("div");
        newItem.classList.add("list-item");
        newItem.innerHTML = `<span class="item-date">${currentDate}</span>
                             <span class="item-name">${incomeName}</span>
                             <span class="item-amount">£${incomeAmount}</
                             span>`;

        // Append new item to the income list
        incomeListContainer.appendChild(newItem);
        clearInput();
    }
}
```

Income and Expense Management (addIncome, addExpense):

function addIncome():

get income name and amount from input fields
get the current date
if income name or amount is missing:
show error message
else:
hide error message
if title is not set:
display list title with the current month and year
mark title as set
add income item to income list with name, amount, and date
clear input fields

the functions

Total Money Calculation (totalMoney):

function totalMoney():
get income amount and expense cost from input fields
get the current total balance

if no income/expense name or amount/cost is missing:
show error message
else: hide error message

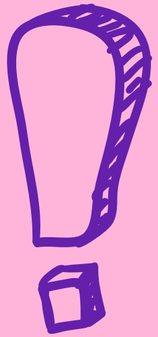
if income amount is provided:
increase current total by income amount
call **addIncome()**
if expense cost is provided:
decrease current total by expense cost
call **addExpense()**
update the total balance display

```
// Display total money when adding items to the list
function totalMoney() {
 let incomeAmount = parseFloat(document.getElementById("amount").
 value) || 0; // Default to 0 if NaN
 let expenseCost = parseFloat(document.getElementById("cost").value) ||
 0;
 let totalEl = document.getElementById("total");
 let currentTotal = parseFloat(totalEl.textContent.replace("£","")) ||
 0;
 let errorMessage = document.getElementsByClassName("error")[0];

 let incomeName = document.getElementById("income")
 let expenseName = document.getElementById("expense")



 if ((!incomeName && !expenseName) || (!incomeAmount && !expenseCost)) {
   errorMessage.style.display="block"
 } else {
   errorMessage.style.display="none"



   if (incomeAmount > 0 && incomeName) {
     currentTotal += incomeAmount
     addIncome()
   }

   if (expenseCost > 0 && expenseName) {
     currentTotal -= expenseCost
     addExpense()
   }

 totalEl.textContent = "£" + currentTotal.toFixed(2);
 }
}
```

issues encountered

A list of our misses during the project. (Sorry!)

While these issues had a negative impact on the project, we learned important lessons from them. We took these as learning opportunities when tackling future projects!



Delays:
due to understanding and applying concepts.

Time restraints:
unable to add edit and delete function.

Bugs:

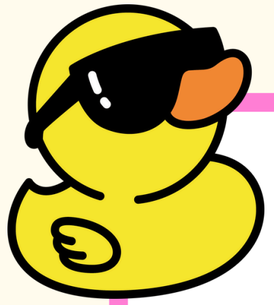
- blank input errors with totals
- unable to input after error message

Inadequate Communication:

- Inability to delegate tasks
- Delays in agreeing on a concept
- Lack of synchronisation when working individually on tasks

what went well

Overall, we're proud of the accomplishments of our team. We learned a great deal both indiviually and also as a team. We shared the same vision for our goals and helped each other understand some of the coding concepts.



Successfully Met Project Goals:

- A functioning budgeting app
- Using JS to manipulate the CSS and HTML elements using DOM
- use of the basic programming concepts learned

Efficient Collaboration:

Our team members worked cohesively and respectfully, despite the challenges of:

- time commitments
- living in different cities
- limitations with technology (poor wifi connections etc)

improvements!

There are a few features that we would like to add to our app!

Points for Improvement	Why?
Fix Bugs!	invalid inputs/ balance total issues
Accounts section	To make it more like a banking app
Edit and Delete functions	to make changes to entries
Pie chart	for visual representation
Local Storage	to save entries the user can refer to later

Thank you for listening!

Any Questions ?

Credits/Contributions

Nina Thomas:
HTML/ CSS / JS on main app/Slide presentation/ initiating Kanban/Replit

Reeshel Rodrigues:
Contributed JS on listing of income/expense functions /Finding bugs in code/Initiated Edit/Delete

Soraya Momoniat:
Username&Password management/Presentation contributions