Calorie Counter App

Documentation Package

Vincent Bruno

Spring 2022

CIS 234

Table of Contents

Application Overview / Summary 2

Privacy of Data Stored by the Application 2

Application Limitations 2

Detailed Description of Site Pages 3

Technical (Code) Discussion 4

Site Map and Application Flowchart 4

Project File Hierarchy 4

Wireframes 5

Prototype Images 6-7

References (links) 8

Application Overview / Summary

This Calorie Counter App allows the user to input food/calories and track the total amount of calories they’ve consumed in a day. Without requiring the user to login to an account, the user’s information that is typed in the fields are stored on the user’s web browser until the user clicks on the “Clear” button.

There are a total of four files associated with this website: two HTML files (index.html, about.html), one CSS file (calStyles.css), and one JavaScript file (storage.js) that controls functions which store, clear, and total the user’s inputs. This website also utilizes Bootstrap version 5.1.3.

Two buttons exist on the index/home page of the website. The first button, Add, allows the user to add a food and its respective caloric value to the section titled “Your Meals”. The second button, labeled Clear, functions as a way for the user to clear or restart the calorie counter by clearing the data stored in their local storage.

The total number of individual meals/food that can be stored is dependent on the storage capacity of the browser’s implementation of the HTML-5 localStorage standard. This means that some users will have larger storage areas than others, and the exact size of storage is subject to change over time. Typical local storage area is from 2 – 10 MB with 5 MB being average.

Privacy of Data Stored by the Application

This Calorie Counter application does not save any information onto the cloud (or online), but instead uses your local web via HTML5 localStorage to save your input.

Application Limitations

This is a free app that totals input. It assumes the first value you input is a type of food/meal (i.e. Pizza, ice cream, breakfast) and the second input typed is a number. Inputting names of meals in the Calorie bar and vice versa will result in a “NaN” result as the total.

Also, if a user tries to copy and run the code locally, users might find that this app may change, disappear or fail without notice in the future.

Detailed Description of Site Pages

**Quick Summary of Site Pages**

index.html Home page which lists all saved food, calories, and total

about.html Description and Contact Page for the application

calStyles.css Style sheet for application

storage.js JavaScript document that oversees application functionality

**Home Page (index.html)**

This page introduces the user to the website, has a Bootstrap 5 nav bar and buttons, and prompts them to enter a type of food, its caloric value, and to add it to a list via the “Add” button: localStorage.setItem(key, value).

The list is shown below the “Add” button and will display the contents of the localStorage in the <div> with the ID of “lsOutput”. This can be reset by clicking the “Clear” button: localStorage.clear();

Below the localStorage output display is the “Total Calorie” fieldset. Within this fieldset is a <div> with an ID called “Total” that utilizes this JavaScript code to add the total number of calories of all foods the user input: total += parseInt(localStorage.getItem(key));

**About Notepad (about.html)**

This page is available on the main site navigation bar and provides the user with details on how to use the app. The page also describes the nature of the limitations and expectations that should be known about data stored in the app.

**API of functions in JavaScript Document (storage.js)**

main() – Nests the entire JS code and is called in the opening body tag of the index.html file using the “onload” method.

btnInsert.onclick – “Add” button that creates variables named “key” and “value” and stores the user’s input as objects in the localStorage called “key” and “value”, then reloads the page. Also only adds user input to localStorage if both fields are populated.

btnClr.onlclick – “Clear” button that clears localStorage and reloads the page when user clicks it.

Technical (Code) Discussion

**Technology Stack**

HTML, CSS, JavaScript, Bootstrap

Browsers supporting Local Storage (Chrome, Firefox, Edge)

HTML-5:LocalStorage (see description below)

**Local and Session Storage functions used in this application (storage.js)**

namespace for Local Storage: localStorage

Save local: localStorage.setItem(key, value);

Read local: localStorage.getItem(key);

Clear local (all): localStorage.clear();

Site Map and Project Flowchart

Calorie Counter App: Site Navigation

Home Page

Index.html

About Page

About.html

Navigation Menu: Home, About

Project File Hierarchy

/CalorieCounter/

/css/

calStyles.css

/js/

storage.js

about.html

index.html

Calorie Counter Wireframe

Wireframe: Home (index.html)

Calorie Counter App

Total Calories

Total: 1100

Your Meals

Pizza: 600

Breakfast: 500

Clear

Add

Calories

Food

About

Home

Wireframe: About (about.html)

About

Calorie Counter App

Home

Info about app

Images of the Application

Index.html (part 1 of 4)

Graphical user interface, text, application, email

Description automatically generated

Index.html (part 2 of 4)

A picture containing graphical user interface

Description automatically generated

Index.html (part 3 of 4)

Graphical user interface, text, application, chat or text message

Description automatically generated

Index.html(part 4 of 4)

Shape

Description automatically generated with low confidence

About.html

Graphical user interface, text, application, email

Description automatically generated

Web References used in Calorie Counter app development:

W3Schools (html)

<https://www.w3schools.com/html/default.asp>

W3Schools (css)

<https://www.w3schools.com/css/default.asp>

W3Schools (Bootstrap)

<https://www.w3schools.com/bootstrap5/bootstrap_buttons.php>

YouTube (JavaScript)

<https://www.youtube.com/watch?v=k8yJCeuP6I8>