

GitHub Copilot – HTML/CSS/JS

Website Development With HTML/CSS/JS and Copilot

- Part 01: Folder Setup
- Part 02: Home Page Structure
- Part 03: Registration Page
- Part 04: Login Page
- Part 05: Weight Entry Page
- Part 06: Weight History Page
- Part 07: Top Three and Admin Page
- Appendix

Environment:

Since we are going to work with HTML, CSS, JavaScript, there is no setup to begin with. You just need to decide on a folder to create your files.

You do not need GitHub Copilot Professional to get a lot from this bootcamp but it does help to have it installed. There is a free *limited* version. Here are the instructions for that version:

<https://docs.github.com/en/copilot/managing-copilot/managing-copilot-as-an-individual-subscriber/getting-started-with-copilot-on-your-personal-account/getting-started-with-a-copilot-plan#accessing-copilot-free>

Although the environment that I am using is a HTML/CSS/JS/web development one, the ideas presented can be adapted to any programming environment. In the beginning we will use the Ask mode of Copilot.

Part 01: Folder Setup

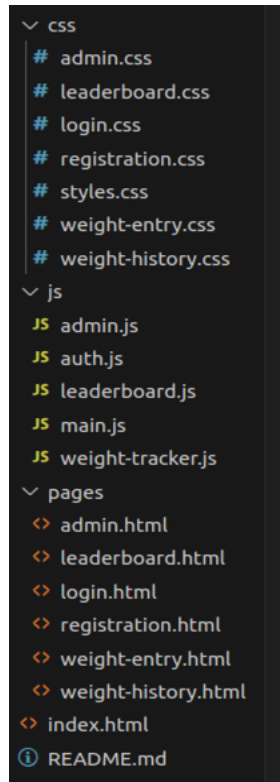
Decide on a folder to start from, e.g. Documents. Open VS Code, you don't have to navigate to a folder yet. Once in VS Code sign in to your GH Copilot account. Sign in to Copilot.

The website we are going to build is based on a friendly competition we usually have in the office where I work. The idea is to find the employee who can lose the most weight during a certain period, usually around Christmas time in the west. Each competitor will sign up and each day they come into the office, the employee will weigh herself and enter that weight and date into the form. At the end of the period, an administrator will do some calculations and figure out the top three weight losers.

Type in your first prompt in the Chat window. For this prompt, we will use the Ask mode of GitHub Copilot as well as the GPT 4.1 if available. Note, the actual model you use will vary based on your subscription level and environment. If you are not sure, just use the default. Here is the prompt:

Help me develop a website for a weight loss competition using only HTML, CSS and JavaScript. This is a company friendly competition we have among the employees. We want a home page, a registration page, a login page, a page to enter weights based on login information, a page for each employee to view a history of weights they entered and a page to show the top three employees who have lost the most weight during the period. There should also be an admin page that would handle the calculations.

Here is the response that I got on my Linux VM.



As you can see, the entire website is already build, just from one prompt. This is what the site looks like now. Remember there will be several versions of this image, almost each participant trying this will get a slightly different result:



We can start right away by fixing the menu, here is my prompt:

Restructure the menu, I want a horizontal menu with this order: Enter Weight, Weight History, Leaderboard, Login, Register and Admin

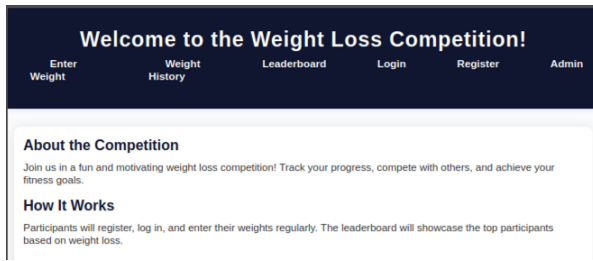
Now I will try to get the site to look like one of the companies in the group of companies that Skillsoft owns:

Add a general style and theme to the website, follow the theme on this website: <https://www.codecademy.com/>

Lets also make the site a bit more responsive:

Make the site responsive.

This is the result so far:



Part 02: Home Page

Lets now try to enhance the home page.

Add an aside to the index.html page to show links to 10 healthy recipes. Place the aside on the right side of the web page. Give it rounded borders and a background color to separate it from the main part of the home page.

The task was complete but I had to nudge it a bit, I wanted the aside to appear on the right side of the main content area:

Make the aside occupy the right side of the page like a column

That worked but I had to ask it to remove the bullet points next to the recipes

I also had to prompt the tool to just repeat the first image since a few of the images were of people, not food

Lets also try to fill up the main content area with dummy text:


Fill up the main-content area with three paragraphs of lorem ipsum text and include a larger image of food

Finally lets add a message to join the competition:
below the last paragraph add a message to sign up and add a link to the register page

Welcome to the Weight Loss Competition!

[Enter Weight](#)[Weight History](#)[Leaderboard](#)[Login](#)[Register](#)

Welcome!



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque euismod, nisi eu consectetur consectetur, nisl nisi consectetur nisi, eu consectetur nisl nisi euismod nisi. Etiam euismod, urna eu tincidunt consectetur, nisl nisi aliquam nunc, eget aliquam nisl nisi euismod.


Morbi facilisis, sapien ac cursus commodo, enim erat dictum urna, nec dictum enim erat nec urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Integer facilisis, enim eu cursus commodo, enim erat dictum urna, nec dictum enim erat nec urna.


Aliquam erat volutpat. Suspendisse potenti. Proin euismod, urna eu tincidunt consectetur, nisl nisl aliquam nunc, eget aliquam nisl nisi euismod. Nullam ac urna eu felis dapibus condimentum sit amet a augue.


Ready to join the competition?


[Sign up here!](#)


Healthy Recipes


[Avocado Egg Toast](#)


[Greek Salad Wraps](#)


[Veggie Hummus Sandwich](#)


[Chicken Quinoa Bowl](#)


[Roasted Veggie Bowl](#)

[Salmon Spinach Salad](#)

[Black Bean Tacos](#)

[Cauliflower Fried Rice](#)

[Turkey Chili](#)

[Overnight Oats](#)

In my case, I did have some problems with the menu. I asked Copilot to diagnose and fix the problem.

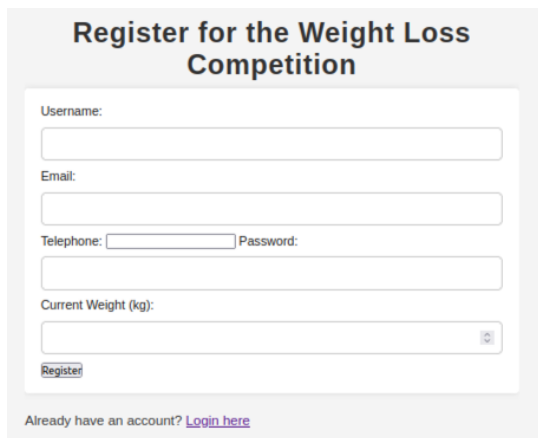
Part 03: Registering

Lets now try to build the register page. Since this is just a browser based application at the moment, there will not be a database involved.

Here is my first prompt:

Create the register page, we want to collect the following data: username, email, telephone, password and current weight in kilograms.

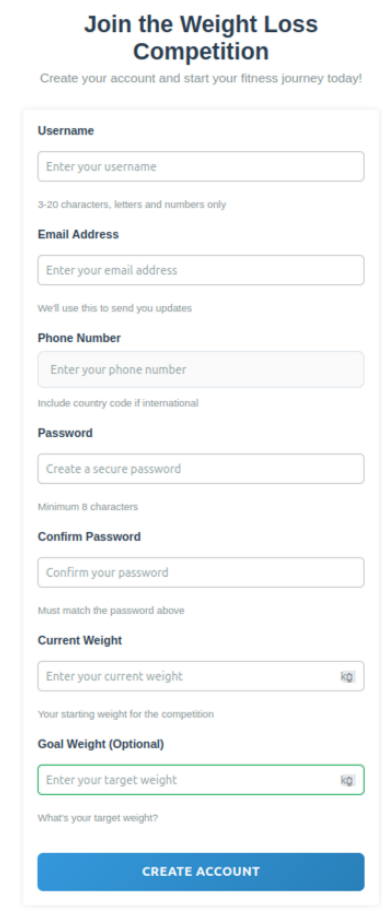
In my case the initial prompt did create a register page so instead of using the word *Create*, use the word *change* in the prompt. This is the result, as you can see, we need to fix the telephone and password fields:



The initial register form is titled "Register for the Weight Loss Competition". It contains five input fields: "Username:", "Email:", "Telephone:" (with a small dropdown menu), "Password:", and "Current Weight (kg):" (with a unit selector). A "Register" button is at the bottom. Below the form, there is a link: "Already have an account? [Login here](#)".

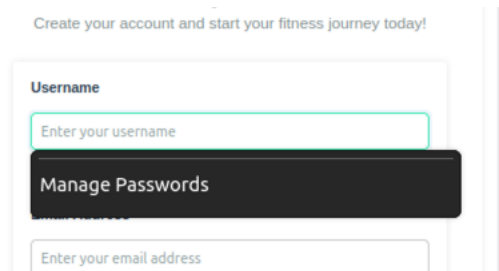
The form fields in the registration file are not consistent, make the form nicer and move it to just below the menu.

The response from Copilot was to update both the html and the css files.
Here is the result:



The improved register form is titled "Join the Weight Loss Competition" with the subtitle "Create your account and start your fitness journey today!". It features a clean, modern design with the following fields: "Username" (with a hint "Enter your username" and a note "3-20 characters, letters and numbers only"), "Email Address" (with a hint "Enter your email address" and a note "We'll use this to send you updates"), "Phone Number" (with a hint "Enter your phone number" and a note "Include country code if international"), "Password" (with a hint "Create a secure password", a note "Minimum 8 characters", and a "Confirm Password" field with a note "Must match the password above"), "Current Weight" (with a hint "Enter your current weight" and a unit selector "kg", and a note "Your starting weight for the competition"), and "Goal Weight (Optional)" (with a hint "Enter your target weight" and a unit selector "kg", and a note "What's your target weight?"). A large blue "CREATE ACCOUNT" button is at the bottom. Below the form, there is a link: "Already have an account? [Sign in here](#)".

After the new form was created, I noticed a problem, when I clicked on the username field, I got this message in a black pop up box.



The screenshot shows the register form with a black pop-up box that says "Manage Passwords" appearing over the "Username" field. The form fields are "Username", "Email Address", and "Current Weight".

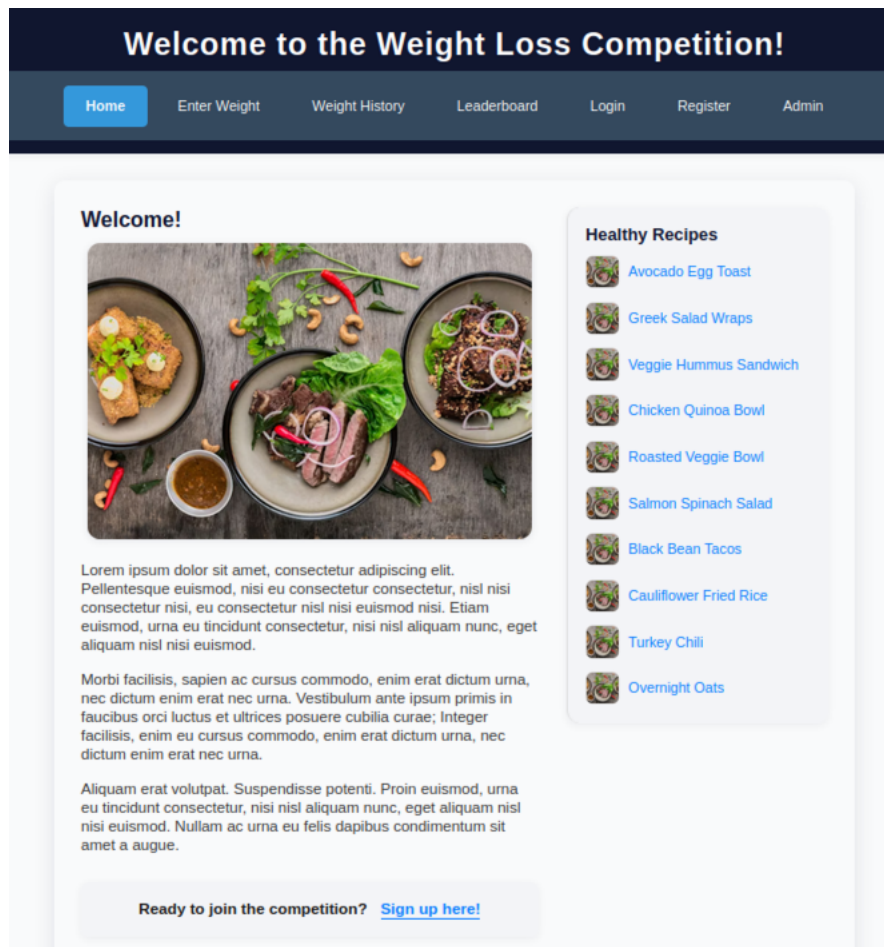
I had to prompt again to fix this issue:

When I click in the username form field on the browser, I get a pop up about manage passwords.

After all those changes I noticed that once we leave the home landing page there is no way to get back to that page, so here is my prompt:

After leaving the home (index.html) page, there is no way for me to see that page again, I only see the five other pages.

The image below is how the site appears with links to all pages:



One last thing we can do with the register page is to add a verify human check.

Current Weight

Enter your current weight

Your starting weight for the competition

Goal Weight (Optional)

Enter your target weight

What's your target weight?

Human Verification

What is 7 + 7?

Enter the answer

Please solve this simple math problem to verify you're human

CREATE ACCOUNT

Add a simple human verification check to the registration page

Notice that you can click a button to get a new question and answer.

Already have an account? [Sign in here](#)

Part 04 – Copilot Features

1. Open the index.html file. Put your cursor at the end of line 20 and hit the Enter key. You will notice that Copilot is trying to insert a new line for pages/contact.html.

```
<li><a href="pages/login.html">Login</a></li>
<li><a href="pages/registration.html">Register</a></li>
<li><a href="pages/admin.html">Admin</a></li>
<li><a href="pages/contact.html">Contact</a></li>
</ul>
nav>
```

This page will be different for different users. For example it may suggest logout.html. This is Copilot Ghost Text and inline suggestion, where it is trying to think for you. You can accept the suggestion or hit the escape key to disregard it.

This feature is also called code completion.

2. Another feature is what they call slash commands. Slash commands act as **shortcuts** to specific tasks.

Instead of describing a task in full natural language, you can use a concise command prefixed by a / to tell Copilot what you want it to do.

Open the auth.js file and in the prompt input box type /. You will then see several shortcuts including the explain shortcut.

3. Copilot has a feature called **chat participants**, which is invoked via the @ symbol in the prompt area.

Chat participants can be considered "domain experts" or specific tools that Copilot can interact with. When you use it, Copilot has a specific context to work with.

For example **@workspace**: This participant has context about all the code in the open workspace or project folder you are currently working in. Use it when you want Copilot to consider other files, so not just the one you're actively working on.

A common prompt is: @workspace what is the attack surface?

This will cause Copilot to analyze any vulnerabilities in your entire project.

Part 05 – Browser APIs

1. Lets try to get Copilot to implement browser storage for the registration page:
Implement browser storage for the username of the registration page. Put all JavaScript in a separate JavaScript file.
2. I had to create a new registration.js file but all the code was provided to me by the tool. Also here are the instructions it gave me:

This implementation provides:

Automatic username saving - Saves the username to localStorage as the user types
Username restoration - Loads the saved username when the page loads
Form handling - Manages form submission with validation
Error handling - Graceful handling of localStorage errors
Registration data storage - Saves complete registration info
Utility functions - Helper functions for managing stored data

It also gave me this information:

The script uses the weightLossApp_ prefix for localStorage keys to avoid conflicts with other applications.

3. In my case the first try did no work. I asked Copilot to analyze the situation and it turned out that the form name was not matching the JS code. Another issue was that the JS was not added to the registration html document.

Part 06 – Agent Mode, Logging In

1. Now that we have the browser storage API working we can now check this object as we login. So lets implement this feature in the login.htm page using Agent mode:

When a user attempts to log in, check if that user is already logged in via the browser storage api. If already logged in, display an appropriate message in the browser in red.

2. This is how Copilot responded:
Now the login functionality has been enhanced with the following features:

Key Features Added:

Login Status Check: When the login page loads, it checks if a user is already logged in via localStorage
Red Error Message: If already logged in, displays a red message saying "You are already logged in as [username]"

Form Disabling: Disables the login form inputs and submit button when already logged in

Logout Option: Adds a logout button when user is already logged in

Session Management: Stores user session data including username, login time, and session ID

Message Display: Clean message system with different styles for error, success, and info messages

How it works:

Page Load: When login.html loads, checkAlreadyLoggedIn() runs automatically

Storage Check: Looks for weightLossApp_currentUser in localStorage

Already Logged In: If found, shows red error message and disables form

Login Attempt: If user tries to login while already logged in, shows additional error message

Successful Login: Stores user data in localStorage and redirects to home page

Logout: Provides logout functionality that clears the session data

Testing:

Visit the login page - should work normally if not logged in

Login with any username/password - will simulate successful login

Visit login page again - should show red "already logged in" message

Use the logout button to clear the session

Try logging in again - should work normally

The red error messages are styled with color: #c33; background-color: #fee; to clearly indicate the already-logged-in status.

3. After logging in and trying again to log in, I get this message:

The screenshot shows the 'Weight Loss Competition' login interface. At the top, a dark blue header contains the title 'Weight Loss Competition' and navigation links: 'Enter Weight', 'Weight History', 'Leaderboard', 'Login' (highlighted with a blue border), and 'Register'. Below the header, a white login card is centered. The card has a heading 'Welcome Back!' and a subheading 'Sign in to continue your weight loss journey'. A red error message box states 'You are already logged in!'. Below this, there are input fields for 'Username or Email' (containing 'Axle') and 'Password' (masked with dots). A 'Sign In' button is at the bottom of the form. At the very bottom of the card, a link says 'Don't have an account? [Create one here](#)'.

Part 07 – Weather API

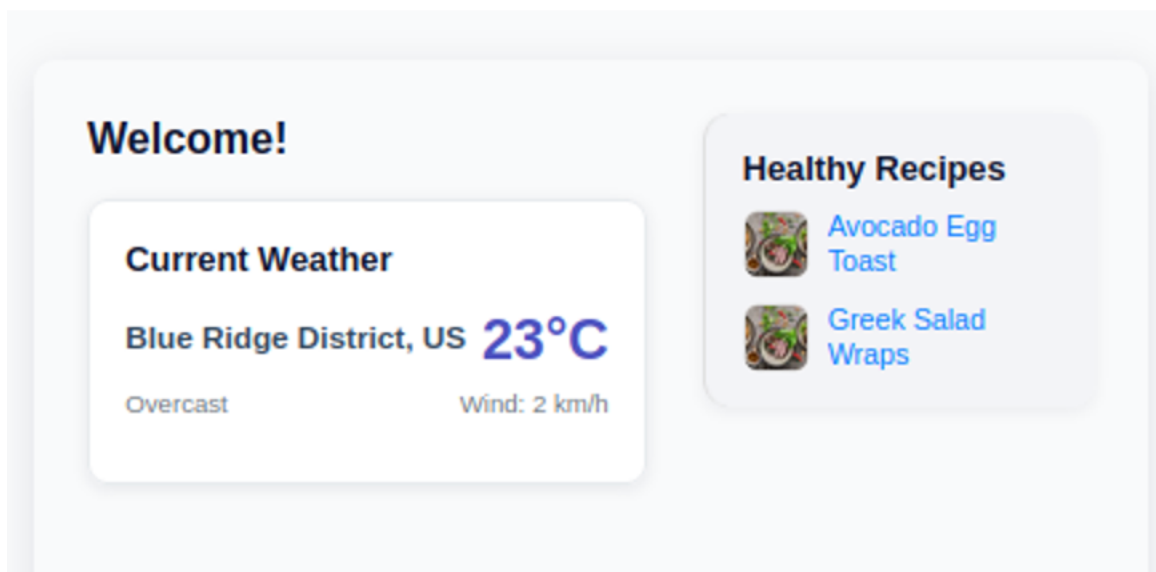
For this part, you can start a new application and prompt for a free weather API. First I simply asked:

Are there any public weather data that I can access for free

Then from that they gave me a sample code, so this would be the next logical prompt:

Implement the example usage using my current html based application

Below is the result:



Appendix A – Local Server (Alternative to Live Server)

In terminal run this command: `python3 -m http.server 8000`