Task 7: Creating CSS Animations for Your Profile Page

Objective:

Create keyframe animations and transitions to enhance the user experience with interactive and dynamic effects. Apply animations to various elements on your profile page, such as buttons, images, and text.

Pre-requisites:

- Basic understanding of HTML elements and CSS properties
- Familiarity with a code editor like Visual Studio Code

Concepts Covered:

- Keyframe Animations
- CSS Transitions
- Combining Animations and Transitions

Concepts:

1. **Keyframe Animations:**

   Keyframe animations allow you to create smooth transitions between different states of an element. You define the styles for various keyframes, and the browser interpolates between them.

   ```css
   @keyframes bounce {
       0%, 20%, 50%, 80%, 100% {
           transform: translateY(0);
       }
       40% {
           transform: translateY(-30px);
       }
       60% {
           transform: translateY(-15px);
       }
   }
   .animated-element {
       animation: bounce 2s infinite;
   }
   ```

2. **CSS Transitions:**

   CSS transitions allow you to change property values smoothly (over a given duration) whenever a specified CSS property changes.

```
.transition-button {
    background-color: #333;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    transition: background-color 0.3s ease, transform 0.3s ease;
}
.transition-button:hover {
    background-color: #ff5733;
    transform: scale(1.1);
}
```

3. **Combining Animations and Transitions:**

   Combining animations and transitions on a single element can enhance the interactive experience by providing both continuous and triggered effects.

```
.combined-element {
    animation: bounce 2s infinite;
    transition: color 0.3s ease;
}
.combined-element:hover {
    color: #ff5733;
}
```

Setup:

1. **Install Visual Studio Code (VS Code):**

   Download and install VS Code from [Visual Studio Code](#).

2. **Web Browsers:**

   Use Google Chrome or Mozilla Firefox for viewing your webpage and utilizing their developer tools for debugging.

Tasks:

1. **Keyframe Animations (20 minutes):**
   - Create a simple keyframe animation (like a bounce or fade-in effect) and apply it to elements on your webpage, such as profile images or headlines.
   - Example:

```
<h1 class="animated-element">John Doe's Profile</h1>
```

```css
@keyframes fadeIn {
    from {
        opacity: 0;
    }
    to {
        opacity: 1;
    }
}
.animated-element {
    animation: fadeIn 2s ease-in-out;
}
```

2. **CSS Transitions (20 minutes):**
   - Implement CSS transitions to enhance interactive elements like buttons and links. For instance, change color, background, or transform properties on hover.
   - Example:

```html
<button class="transition-button">Hover Over Me</button>
```

```css
.transition-button {
    background-color: #333;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    transition: background-color 0.3s ease, transform 0.3s ease;
}
.transition-button:hover {
    background-color: #ff5733;
    transform: scale(1.1);
}
```

3. **Combining Animations and Transitions (20 minutes):**
   - Integrate both animations and transitions on a single element to observe how they work together. For example, make a button slowly change color over time and also react instantly when hovered over.
   - Example:

```html
<p class="combined-element">Hover and Watch Me Bounce!</p>
```

```css
@keyframes bounce {
    0%, 20%, 50%, 80%, 100% {
        transform: translateY(0);
    }
    40% {
        transform: translateY(-30px);
    }
    60% {
        transform: translateY(-15px);
    }
}
.combined-element {
    animation: bounce 2s infinite;
    transition: color 0.3s ease;
}
.combined-element:hover {
    color: #ff5733;
}
```

Instructions:

1. Write the required code in `index.html` and `styles.css`.

2. Open the `index.html` file in your web browser to ensure the code displays correctly.

3. Use the browser's developer tools to debug and inspect the elements.

Resources:

- **animation - CSS: Cascading Style Sheets | MDN**
  The animation shorthand CSS property applies an animation between styles. It is a shortha...
  https://developer.mozilla.org/en-US/docs/Web/CSS/animation

- **transition - CSS: Cascading Style Sheets | MDN**
  The transition CSS property is a shorthand property for transition-property, transition-dura...
  https://developer.mozilla.org/en-US/docs/Web/CSS/transition

- Get started with
  Visual Studio Code
  Documentation for Visual Studio Code
  Find out how to set-up and get the most from Visual Studio Code. Optimized for building and d...
  https://code.visualstudio.com/docs

Videos:

CSS ANIMATION TUTORIAL #1

INTRODUCTION

▶ 🥷 **Net Ninja**   **3:46**min   **339,966** Views   **4,002** Likes

---

GitHub Instructions:

1. **Open in Visual Studio Code:**

   After clicking on the "Open in Visual Studio Code" button from the GitHub Classroom confirmation page, VSCode will open the repository directly. If prompted, select "Open" or "Allow" to open the repository in VSCode.

2. **Open the Terminal in VSCode:**

   In VSCode, open a terminal by selecting Terminal > New Terminal from the top menu.

3. **Complete the Task:**

   In VSCode, write your solution in the `index.html` and `styles.css` files.

4. **Run and Test Your Code:**

   Open your `index.html` file in a web browser to ensure it works correctly. Use the following command:

   ```
   open index.html
   ```

5. **Commit Your Changes:**

   In the VSCode terminal, add your changes to git:

   ```
   git add index.html styles.css
   ```

   Commit your changes with a meaningful message:

```
git commit -m "Completed task 11"
```

6. **Push Your Changes to Your Repository:**

   Push your changes to your forked repository:

```
git push origin main
```

7. **Create a Pull Request:**

   Go to your repository on GitHub.

   Click on the "Pull Requests" tab.

   Click the "New Pull Request" button.

   Ensure the base repository is the original template repository and the base branch is `main`.

   Ensure the head repository is your forked repository and the compare branch is `main`.

   Click "Create Pull Request".

   Add a title and description for your pull request and submit it.

Summary of Commands:

```
# Open in Visual Studio Code

# Open terminal in VSCode

# Complete the task by editing index.html and styles.css

# Navigate to the directory containing index.html
cd path/to/your/index.html

# Run your code
open index.html

# Add, commit, and push your changes
git add index.html styles.css
git commit -m "Completed task 7"
git push origin main

# Create a pull request on GitHub
```