1. What is the difference between UI and UX? Explain in your own words?

User Interface (UI):

1. Looks and Design: UI is all about how things in an app or website look.
   - Think about it like picking cool colors, nice buttons, and arranging stuff on the screen to make it pretty.

2. Visual Appearance: It's about making stuff look good and attractive to users.
   - For example, making your game app's main menu look fun with colorful buttons.

3. User Interaction: UI is what you touch, click, or see on your screen.
   - Like tapping on "Play" to start a game or clicking "Like" on a social post.

4. Design Parts: It's the elements you see, like buttons, menus, and pictures.
   - Imagine the "Send" button in your chat app; that's part of the UI.

User Experience (UX):

1. Overall Feel:
UX is how you feel while using an app or website.
   - It's like when using a navigation app that takes you smoothly to your destination without confusion.

2. User's Happiness:
It's about making sure users are happy and can easily do what they want.
   - For instance, a food delivery app that helps you order quickly and accurately.

3. Whole Journey:
UX covers the whole trip from the moment you open an app until you finish what you came for.
   - From choosing a movie on a streaming app to watching it comfortably - that's UX.

4. Research and Testing:
UX designers talk to users and do tests to understand what people like and what they find easy to use.
   - For example, asking people how they feel about a new feature in a shopping app.

2. What is a front-end developer? Explain the roles and responsibilities in your own words?
   Certainly, here's the explanation without any bold formatting:

Frontend Developer:

A frontend developer is like the architect and builder of the digital world. They're the creative minds behind the part of a website or application that users see and interact with directly. Here are their roles and responsibilities:

1. Creating User Interfaces (UI):
Frontend developers design and build the user interface (UI) of websites and web applications. They decide how everything looks and feels. It's like they're responsible for the layout, colors, fonts, buttons, and images you see on a website.

2. Writing Code:
They use languages like HTML, CSS, and JavaScript to bring designs to life. HTML is like the skeleton, CSS is the skin, and JavaScript is the brain of a web page. They write code to make sure everything is in the right place and looks great on different devices like phones, tablets, and computers.

3. Responsive Design:
Front-end developers make sure websites and apps work well on all sorts of screens and browsers. They ensure that a webpage looks just as awesome on a tiny phone screen as it does on a big computer screen.

4. User Experience (UX):
They work closely with UX designers to make sure users have a smooth and enjoyable experience. This means making sure buttons are clickable, forms are easy to fill out, and animations are smooth.

5. Testing and Debugging:
Front-end developers test everything to make sure it works correctly. They also find and fix any problems, like broken links or buttons that don't work.

6. Collaboration:
They often collaborate with designers to turn design concepts into reality. They also work with backend developers to connect the frontend with the server and databases.

3. Who is a backend developer? Explain the roles and responsibilities in your own words?
   Certainly, here's the explanation without any bold formatting:

   Backend Developer:

   A backend developer is like the behind-the-scenes magician of a website or app. They work on the hidden parts that users don't see, but are essential for everything to work smoothly. Here are their roles and responsibilities:

   1. Server Management:
   Backend developers manage the server, which is like a powerful computer that stores all the data and files for a website or app. They make sure the server runs smoothly so that when you click on something, it happens fast.

   2. Database Handling:
   They deal with databases, which are like digital filing cabinets where all the information is stored. They create, update, and retrieve data from these databases. For example, when you log into a website, the backend fetches your username and password from the database to check if it's correct.

   3. Security:
   They're responsible for the safety of data. They put locks and alarms on the server and databases to keep bad people out. They also make sure your data, like passwords, is encrypted so it can't be easily stolen.

   4. APIs:
   Backend developers create APIs (Application Programming Interfaces) that allow the frontend (the part you see) to talk to the backend. It's like a bridge that helps different parts of a system communicate.

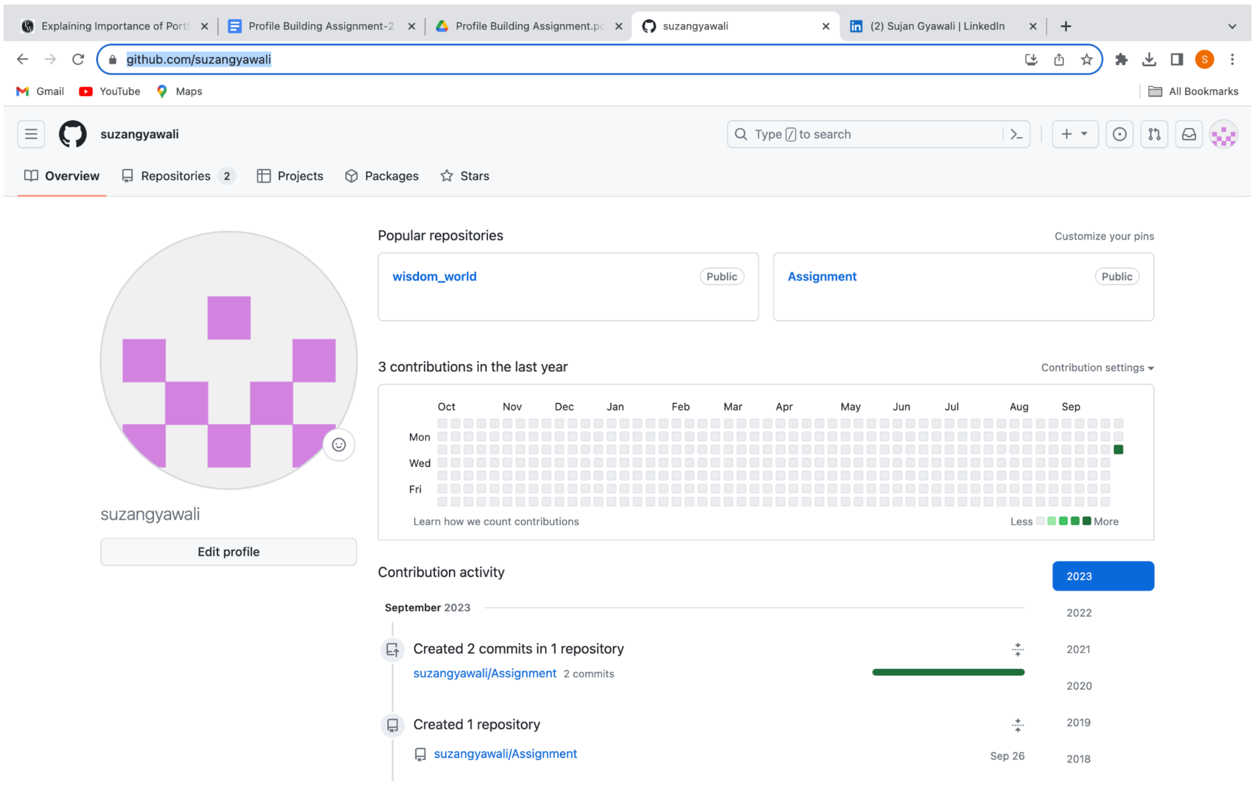
   5. Performance:
   They make sure everything runs super fast and doesn't crash, even if lots of people use the website or app at the same time. It's like making sure a concert runs smoothly without any hiccups.

   6. Testing and Debugging:
   They test all the hidden code to find and fix any issues. It's like checking the engine of a car to make sure it doesn't break down.

   7. Collaboration:
   They work closely with frontend developers and other team members to ensure the whole system works together. It's like a team effort to build a big, complex machine.

4. Create your own GitHub & LinkedIn accounts. Share the screenshot of both of your accounts and share your profile links of GitHub & Linkedin in the solution.

   Github Link: https://github.com/suzangyawali



Explaining Importance of Portf | Profile Building Assignment-2 | Profile Building Assignment.p | suzangyawali | (2) Sujan Gyawali | LinkedIn

github.com/suzangyawali

suzangyawali

Overview | Repositories 2 | Projects | Packages | Stars

Popular repositories

wisdom_world Public

Assignment Public

3 contributions in the last year

Contribution activity

September 2023

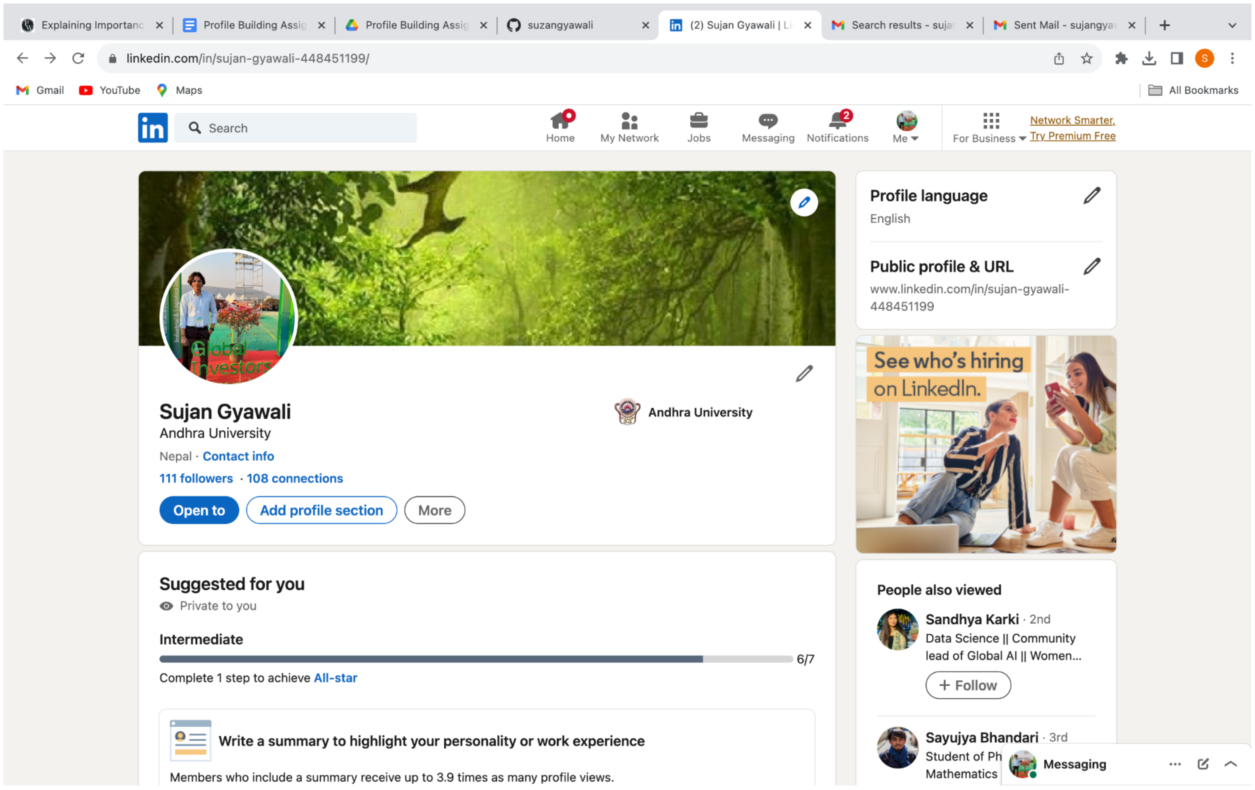
Created 2 commits in 1 repository

suzangyawali/Assignment 2 commits

Created 1 repository

suzangyawali/Assignment Sep 26

   Linkedin Link: https://www.linkedin.com/in/sujan-gyawali-448451199/



Sujan Gyawali
Andhra University
Nepal · Contact info
111 followers · 108 connections
Open to | Add profile section | More

Andhra University

Profile language
English

Public profile & URL
www.linkedin.com/in/sujan-gyawali-448451199

Suggested for you
Private to you
Intermediate
6/7
Complete 1 step to achieve All-star

Write a summary to highlight your personality or work experience
Members who include a summary receive up to 3.9 times as many profile views.

People also viewed
Sandhya Karki · 2nd
Data Science || Community lead of Global AI || Women...
+ Follow

Sayujya Bhandari · 3rd
Student of Ph
Mathematics

Messaging

5. Write your first blog on your favorite platform and share the link to the blog. (-you can choose any topic for writing your 1st blog)

Platform: Medium
Topic: Selectors in CSS.

https://medium.com/@sujangyawali177/selectors-in-css-455fe3e4c9b3

DAY-6

Are you curious about how developers create stunning website designs? *CSS selectors* play a crucial role in web development, allowing developers to *target specific HTML elements* and apply styling rules. In this blog post, we'll explore the five categories of CSS selectors, from simple selectors to pseudo-element selectors, and provide examples to help you master CSS styling.

Why Selector in CSS?

1. They allow developers to target specific HTML elements and apply styling rules to them.
2. They provide a way to customize the appearance of a website and make it visually appealing and user-friendly.

Types of Selectors

We can divide CSS selectors into 5 categories:

1. Simple selectors.

2. Combinators.

3. Attribute selectors.

4. Pseudo-class selectors.

5. Pseudo-elements selectors.

NOTE: CSS Selectors cannot be used with inline CSS.

We will be using *internal CSS* to demonstrate CSS Selectors.

In this blog, we are going to discuss Simple selectors. A simple selector is the most basic type of selector in CSS and it selects elements based on their tag name, class, or ID.

There are five types of simple selectors in CSS:

1. Universal Selector
2. Element Selector
3. Class Selector
4. ID Selector

5. Selector list

Universal Selector:

The styles applied to the universal selector will be applied to every element on the page.

Asterisk (*): symbol denotes the universal selector in CSS.

```
/* Universal selector */

/* Target all elements on the page */

*{

   margin:0;

   padding:0;
}
```

The universal selector, denoted by the asterisk (*), targets all elements on a web page. In this particular example, the code sets the margin and padding of all elements to 0, effectively removing any default spacing or indentation that may be present.

Element Selector:

The element selector targets elements based on their tag name.

```
/* Element Slector */

/* Target all <h1> elements on the page */

/* h1{

   color:#808081;
```

```css
}
```

An element selector targets all instances of a particular HTML element on a web page. In this example, the code targets all `<h1>` elements and sets their color to `#808081`.

Class Selector:

The class selector targets elements based on given the class value in their class attribute.

To apply a class selector to an element, you need to add a period (.) followed by the class name in your CSS code.

```css
/* Class Selector  */

.para{

color:rgba(255, 38, 0, 0.836);
}
```

```
.hello{
    font-size: 80px;
}
```

The first class selector targets all elements with the class name "*para*" and sets their color to a *shade of red*. To apply this class to an HTML element, you would need to add the class name "para" to the element's class attribute. For example:

```
<p class="para">Lorem ipsum dolor sit amet consectetur adipisicing
elit.</p>
```

The second class selector targets all elements with the class name "hello" and sets their font size to 80 pixels. To apply this class to an

HTML element, you would need to add the class name “hello” to the element’s class attribute.For example:

```
<h1 class="hello">Hello World!</h1>
```

Let’s check your knowledge, shall we? Can you tell output for paragraph element below:

```
<p class="para hello">Lorem ipsum dolor sit amet.</p>
```

Above paragraph element has the class “para” and “hello”, which means it belongs to both the “para” class and the “hello” class.

So the output of this HTML code will be a paragraph element with the following styles applied:

- The text color will be a shade of red, specified by the “color” property in the “.para” class.

- The font size will be 80 pixels, specified by the “font-size” property in the “.hello” class.

These styles are applied to the paragraph element because it has both the “para” and “hello” classes, and the CSS rules defined for those classes will apply to any element that has them.

ID selector:

ID selectors target an element based on its unique ID attribute value, providing **higher specificity** compared to class selectors or other selectors. An element can only have one ID attribute, which must be unique throughout the HTML document.

To apply an ID selector to an element, you need to add a hash (#) followed by the ID name in your CSS code. For example, if you want to target an element with ID=”id1”, you can use the ID selector like this:

```css
/* Target the element with ID="id1" */

#id1{

   color: whitesmoke;

   background-color:black;



}
```

Html code

```html
<p class="id="id1">Lorem ipsum dolor sit amet.</p>
```

Understand Higher specificity through code:

```html
<p class="para hello" id="id1">Lorem ipsum dolor sit amet.</p>
```

Analyze output of above Html code :

The `id` attribute with the value of "id1" is unique to this element, making the selector targeting this ID even more specific than a class selector targeting a class shared among multiple elements.

Styles applied through the ID selector will take precedence over styles applied through the class selector.

Selector list:

A selector list allows you to apply styles to multiple selectors at once. To create a selector list, you need to separate the selectors with a comma. For example, if you want to apply the same style to all <h1> and <h2> elements on the page, you can use a selector list like this:

```
/* Target all <h1> and <h2> elements on the page */

h1, h2 {

 font-weight: bold;
}
```

In conclusion, mastering CSS selectors is crucial for creating effective and efficient styles for your web pages. Now that you understand the five types of simple selectors, it’s time to put this knowledge into practice and start experimenting with your own styles. Keep in mind that using selectors judiciously is key to creating maintainable and performant CSS. To further improve your CSS skills, consider exploring more advanced selectors, learning about cascading and inheritance, and practicing with real-world projects.