Lesson 7

Today's focus will be based on the creation of your portfolio website. We will look at:

- Project management and development processes.
- Defining your project requirements.
- Creating a prototype
- The purpose of your portfolio website.

Video Recordings:

https://vimeo.com/715162029/b4cde424a0

https://vimeo.com/715187313/3183a4f94a

https://vimeo.com/715245043/e0f06b8d12

Further Reading:

https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

https://www.aha.io/roadmapping/guide/requirements-management/what-is-a-good-feature-or-user-story-template

Tasks:

1. Create a user story that describes the purpose of your portfolio website.
2. Define user stories for the different types of people who you intend to use your website:
3. Employers
4. Recruiters
5. Developers and creatives
6. Define your elevator pitch.

Questions and Answers:

Q: How can CSS be used to search for page elements that have an attribute that contains a specified word?

CSS provides two types of comparison to identify if an elements attribute contains a specified word. The ***=** comparison allows the word to be identified regardless of whether it's an independent word or part of another word. For example:

```
[data-search="java"]
```

This example will return items that have values containing **java** as well as **javascript** because both words contain **java**.

A more specific comparison is **~=** , which would identify elements that contain **java** as an independent word

```
[data-search~="java"]
```

This example would ignore items set as **javascript** because the **java** text isn't an independent word.

Example:

Today's example initiated the development of a data filter that could be used to allow recruiters and employers to focus on specific content interest. This could be used in your portfolio as a feature to show items where a specific skill has been used.

```
<!DOCTYPE html>
<html>
  <head>


  <body>

    <select onChange="search(this.value)">
      <option value="">All</option>
      <option value="javascript">Javascript</option>
      <option value="html">HTML</option>
      <option value="css">CSS</option>
    </select>

    <ul>
      <li data-search="html css">Item 1</li>
      <li data-search="css">Item 2</li>
      <li data-search="javascript html">Item 3</li>
      <li data-search="javascript">Item 4</li>
    <ul>

<style>
.filtered > *:not(.active){
  display: none;
}
</style>

<script>
function search(value){
  alert(value);
  document.querySelector("ul").classList.add("filtered");

  document.querySelectorAll(`ul > [data-search~="${value}"]`).forEach(function(item){
    console.log(item);
    item.classList.add("active");
  });

}
</script>

  </body>
</html>
```