BrainStation
1-800-903-5159

Software Engineering Diploma Program

Student Transcript

A minimum passing grade of 70% is required for completion of the program.

Unit	Weight	Grade
Unit 1: Introduction to Web Development	**15%**	**89.4%**
Unit 2: Development Fundamentals	**25%**	**90.0%**
Unit 3: Programming with JavaScript	**25%**	
Unit 4: Collaborative Development	**15%**	
Unit 5: Professional Development	**20%**	
Total	100%	

Cumulative Grade: **90%**

Student ID

Student ID:	526847
Chandni	Melwani

Software Engineering Diploma Program	
Start Date:	November 26, 2024
End Date:	February 28, 2025

Program Completion	
Status:	In Progress
Transcript Issued:	December 19, 2024
Withdrawal Date (if Applicable)	

Attendance Grade

Below is a weekly breakdown of your attendance record. A minimum attendance grade of 90% is required for completion of the program. Please notify your TA of your absence with a reason ahead of time.

Week	Mon	Tue	Wed	Thur	Fri	Grade
1	DAY OFF	7.0	7.0	7.0	7.0	100%
2	7.0	7.0	7.0	7.0	7.0	100%
3	7.0	7.0	7.0	7.0	7.0	100%
4	7.0	7.0	7.0	7.0		100%
5						
6						
7						
8						
9						
10						
11						
12						

Cumulative Grade: 100%

Grading System

The BrainStation grading system employs a numerical marking system. Below is a description of grade meanings.

Grade Meanings	Numerical Scale of Marks
Excellent	90-100%
Very Good	80%-90%
Good	70%-80%
Developing	60%-70%
Limited	0-60%

BrainStation
1-800-903-5159

Unit 2: Development Fundamentals 25%

Student ID

Student ID:	526847
Chandni	Melwani

Deliverable One	25%	Performance Rating						
		Weight	Exemplary	Very Good	Satisfactory	Developing	Limited	Incomplete	Days Late
Categories	Requirements/Functionality	30%		x					
	Comprehension/Execution	30%		x					
	Code Quality	20%		x					
	Visual Design	20%		x					

Grade: **90.0%**

Deliverable Two	25%	Performance Rating							
		Weight	Exemplary	Very Good	Satisfactory	Developing	Limited	Incomplete	Days Late
Categories	Requirements/Functionality	30%							
	Comprehension/Execution	30%							
	Code Quality	20%							
	Visual Design	20%							

Grade:

Deliverable Three	50%	Performance Rating							
		Weight	Exemplary	Very Good	Satisfactory	Developing	Limited	Incomplete	Days Late
Categories	Requirements/Functionality	30%							
	Comprehension/Execution	30%							
	Code Quality	20%							
	Visual Design	20%							

Grade:

Overall Grade

Summary of Unit Grade	Weight	Grade
Sprint 1	25%	90%
Sprint 2	25%	
Sprint 3	50%	

Current Unit Grade: **90.0%**

Additional Comments

Done Well
- Overall great work with Sprint 1 of the BandSite Project!
- Nice work styling the project, the site looks very close to the provided mockups and works well at the specified breakpoints as well as in between
- Good work following the BEM methodology, the class names look great, also if you test your compiled css with https://jonassebastianohlsson.com/specificity-graph/ the graph looks great, a fairly flat line which is one thing BEM helps to create. The flat line represents how specific each selector is and generally you want to avoid spikes in the graph which is what your code is doing, great work
- Nice use of scss features, mixins, variables and partials

Opportunities
- Nice work creating the hero__overlay with absolute positioning. That being said there's another way this could be done without adding an extra div tag which would be to use a linear-gradient with multiple background images for example: https://codepen.io/jimbennett/pen/ZYzOeEL That being said the way you've created this is also valid, you don't have to refactor, mainly just for awareness of other techniques that could be used.
- Nice use of Mixins. That being said note that you can also pass args to mixins, similar to functions, which could make some of your mixins more re-usable, leading to more DRY code.
  There is some redundancy, particularly in font mixins. For example, the text-footer mixin is identical to text-body. Consider using the same mixin or a base mixin for shared styles:

```
@mixin text-base($size, $lineHeight, $weight, $tabletSize, $tabletLineHeight, $tabletWeight: $weight) {
    font-family: $font-primary;
    font-size: $size;
    line-height: $lineHeight;
    font-weight: $weight;

    @include tablet-and-up {
        font-size: $tabletSize;
        line-height: $tabletLineHeight;
        font-weight: $tabletWeight;
    }
}

@mixin text-footer {
    @include text-base(0.8125rem, 1.125rem, 400, 0.875rem, 1.375rem);
}
```
- Not a requirement for this project although it's always good to consider what your site may look like on larger screens. At the moment content would continue to stretch on larger screens which could be adjusted by adding a max-width and margin auto, something like a "wrapper" https://css-tricks.com/best-way-implement-wrapper-css/