**Name:** Nathan Johnson  
**Student ID:** 923721692  
**GitHub Username:** Natty2027  
**Assignment Number:** 3

**CSC 317 Assignment Final Submission**

**Name:** Nathan Johnson  
**Student ID:** 923721692  
**GitHub Username:** Natty2027

**Analysis**

My process built the portfolio piece using a hybrid CSS workflow that was developed using handcrafted styles for precision control, utilizing two AI-created theme variations for creative motivation and resulting in a polished theme which adopts CSS variables for theme consistency, media queries which fluidly re-sizes content between mobile, tablet and desktop sizes and easy to read hover and focus states to promote accessibility. On the main page I added a welcome banner, a navigation bar with links to my about, education, experience, projects, skills, extracurricular activities, and contact sections, as well as a to switch CSS themes, and a promotional video. I have also built a budget planner application that allows you to add and filter income or expense transactions right from the statement and to see real-time summaries of your totals and balances for different periods, plus I made an iOS-like calculator with full memory operations, a sign toggling, percentages, arithmetic functions, dynamic theming without page reloading, and with button sizing and contrast optimization. Under the hood, my core stylesheet provides global reset and base styles, grid-based form and button styles, flexbox-powered summaries and lists while being fleshed out with color-coded borders to breakdown how much income and expenses are, and subtle animation (especially within header and project cards) to bring a dynamic experience that fits together like a puzzle. I made the budget calculator in order for people to have some kind of interactive and easy way to instantly keep track of their finances. When you enter a transaction, you just choose whether it’s an income or an expense, enter the amount, select a category (say, “Groceries” or “Salary”), and pick a date. You watch the totals being recalculated in real time instead of finding out after you press the “=” button that you are in the red for this month, with separate subtotals for income and expenses easily visible at a glance. To make finding past entries easy, I incorporated a filter panel: You can switch between income, expenses or see everything at once, and you can also search by category to drill down on specific spending habits. Under the hood, I employed CSS variables to maintain consistent spacing and color cues (green borders for income, red for expenses), and a CSS grid layout to ensure the input form, filter controls and transaction list rearrange gracefully on narrower screens.

This is the theme switcher my webpage, it is very versatile.

// Theme switcher: cycles through stylesheets and remembers the last one

const themes = [

'styles.css',

'ai-theme1.css',

'ai-theme2.css',

'hybrid.css'

];

function switchTheme() {

// Get current index (or default to 0)

let idx = parseInt(localStorage.getItem('themeIndex') || '0', 10);

// Advance to next theme, wrap around

idx = (idx + 1) % themes.length;

// Update the <link> href and store new index

document.getElementById('theme-link').href = themes[idx];

localStorage.setItem('themeIndex', idx);

}

// Attach to button click

document

.getElementById('theme-switcher-btn')

.addEventListener('click', switchTheme);

// On page load, apply stored theme

window.addEventListener('DOMContentLoaded', () => {

const savedIdx = parseInt(localStorage.getItem('themeIndex') || '0', 10);

document.getElementById('theme-link').href = themes[savedIdx];

});