# **Module 2 Assignment 2: Exploring JavaScript Topics with EJS, Node.js and Express**

## IFT 544: Middleware Prog & Database Sec (2023 Fall)

Santrupth Sunkari

Prof. Dinesh Sthapit

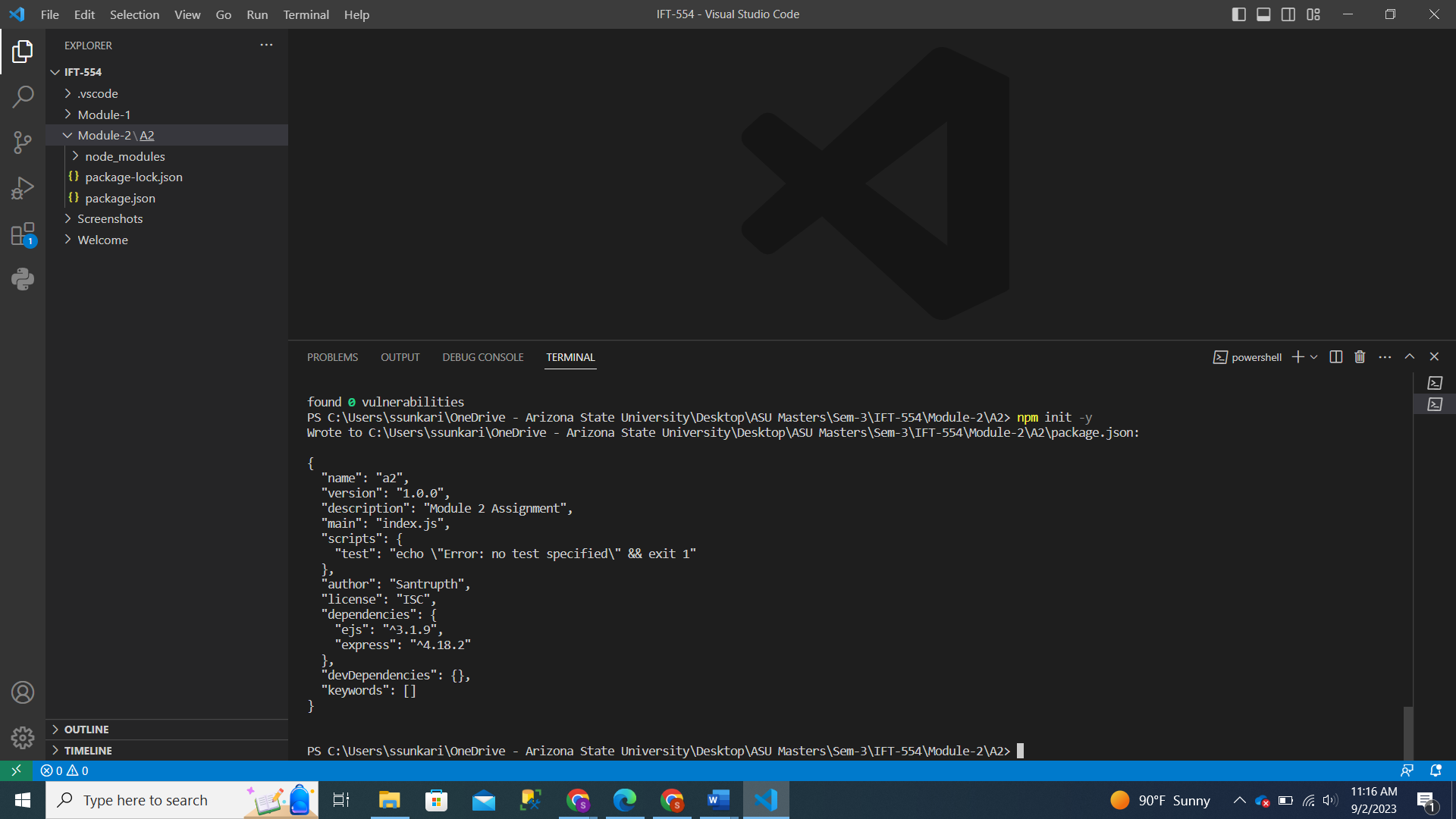
September 3, 2023

1. Setup

A computer screen shot of a black screen

Description automatically generated

1. Creation of package.json



1. Installation of express and body-parser

A screenshot of a computer

Description automatically generated

1. App.js

A screenshot of a computer program

Description automatically generated

1. Index.ejs in the views folder

A computer screen shot of a computer screen

Description automatically generated

1. Result.ejs

A computer screen shot of a program code

Description automatically generated

Running the application

A screen shot of a computer

Description automatically generated

Resulting Page on local host://4000

A screenshot of a computer

Description automatically generated

On entering number 1 as 5 and number 2 as 8

A screenshot of a computer

Description automatically generated

On clicking view source

A screenshot of a computer

Description automatically generated

2. .ejs code Analysis

index.ejs:

* This is the first HTML form where users are prompted to enter 2 numbers.
* The page consists of 2 input fields along with a submit button.
* The form action is yielded to “/Calculate”, indicating when the submit button is clicked it would be handled to a route which could be handled by the URL.
* User is prompted for entering values in both fields.

Result.ejs:

* This HTML is mainly used for displaying the results.
* This even consists of HTML form which is similar to that of index.ejs, which enables users to perform some other calculations.
* The form displays results using embedded JavaScript in which results are passed as variables when calling the template.
* It mainly displays the operations like sum, difference, product and quotient which are dentoed by <%=sum%>, <%=difference%>, <%=product%> and <%=quotient%>.

JavaScript Code:

* JavaScript code in EJS template is bound by <% and %> tag.
* The tags are mainly used to insert inside the JavaScript logic among HTML template.

Dynamic HTML Generation:

* The JavaScript code that is integrated into the template serves the purpose of creating HTML content that can change dynamically. Specifically, it's in charge of inserting computed values into the HTML structure.

Control Structures:

* The EJS file does not employ explicit control structures such as loops or conditionals; its primary emphasis is on displaying the results of calculations.

Variable Access and Display:

* Variables within the HTML content are both accessed and presented using the <%= variableName %> syntax.
* To illustrate, <%= sum %> retrieves the value of the sum variable, which should be provided to the template during server-side rendering.

EJS-Specific Syntax or Features:

* EJS is primarily employed for variable interpolation, allowing the inclusion of values within HTML content.
* The <%= variableName %> syntax is a common EJS convention for displaying variable values within the HTML.
* Additionally, EJS provides support for <% code %> to embed JavaScript logic without producing any visible output in the rendered HTML.

To summarize, the main objective of this EJS file is to generate dynamic HTML content by incorporating JavaScript code within the template. It utilizes EJS's variable interpolation syntax to exhibit computed outcomes. Although this particular template doesn't make use of control structures, it's important to note that EJS has the capability to accommodate them for more intricate templating requirements.

**3. HTML code analysis**

Rendered HTML code

A screenshot of a computer

Description automatically generated

* The overall structure of the HTML, including elements like <html>, <head>, and <body>, remains consistent between the original result.ejs template and the resulting HTML.
* In the rendered HTML, the dynamic JavaScript code present in the original template has been substituted with fixed values:
  + <%= sum %> is replaced with 13.
  + <%= difference %> is replaced with -3.
  + <%= product %> is replaced with 40.
  + <%= quotient %> is replaced with 0.625.
* The rendered HTML lacks the dynamic aspect found in the original EJS template, where these values would typically be calculated based on server-side data and dynamically inserted into the HTML.

4. **Documentation**

**Analysis of .ejs Template to Rendered HTML**

**Result.ejs code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8"?

    <meta name="viewport" content="width=device-width, initial scale=1.0">

    <title>Document</title>

</head>

<body>

    <form action="/calculate" method="POST">

        <input type="number" name="num1" placeholder="Enter first number" required>

        <input type="number" name="num2" placeholder="Enter second number" required>

        <button type="submit">Calculate</button>

    </form>

    <p>

        <h1>Calculation Results:</h1>

        <p>Sum: <%= sum %></p>

        <p>Difference: <%= difference %></p>

        <p>Product: <%= product %></p>

        <p>Quotient: <%= quotient %></p>

    </p>

</body>

</html>

* + Employs embedded JavaScript enclosed within <% and %> tags to create dynamic HTML content.
  + Exhibits calculation results, including sum, difference, product, and quotient, using the <%= variableName %> syntax.
  + Abstains from incorporating control structures like loops or conditionals within this template.
  + Predominantly tailored for server-side rendering, where dynamic data would substitute the placeholders during the rendering process.

**Browser generated code**

A screenshot of a computer

Description automatically generated

**Things Observed to Differences:**

**Dynamic to Static Content:**

* The .ejs template is intended for producing dynamic content, whereas the rendered HTML is static and devoid of dynamic data.
* Within the HTML code, all instances of placeholders <%= variableName %> have been exchanged for precise values.

**JavaScript Code Alternative:**

* + Within the .ejs template, JavaScript code is implemented to inject dynamic data into the HTML structure.
  + In the resulting HTML, the JavaScript code is replaced with the specific calculated values.

**Use:**

* The .ejs template functions as a framework for generating HTML dynamically, relying on server-side data, user input, or application logic.
* The rendered HTML captures a static representation of the HTML outcome that results when particular data is employed within the .ejs template.

**Advantages of Using .ejs Templates**

**Dynamic Content Generation:**

* .ejs templates enable the dynamic creation of HTML content, simplifying the integration of data-driven elements into web pages.

**Reusability:**

* Templates are reusable with various data sets, decreasing redundancy and bolstering ease of upkeep.

**Modularity:**

* The utilization of templates fosters a division of responsibilities between presentation and logic, enhancing code organization.

**Consistency:**

* Templates contribute to preserving uniform formatting and styling across numerous pages or components.

**Ease of Maintenance:**

* Modifications to the HTML structure can be executed in a single location, the template, impacting all occurrences where that template is employed.

**Examples and Scenarios for Using.ejs Templates**

**Weather Widgets:**

* Developing templates for weather widgets that display current weather conditions and forecasts on websites or mobile apps.

**Real Estate Listings:**

* Designing templates for real estate listings that display property details, images, pricing, and contact information.

**Inventory Management:**

* Creating templates for inventory management systems to track product listings, stock levels, and order processing.

**Booking and Reservation Systems:**

* Crafting templates for booking and reservation systems for hotels, restaurants, flights, or event tickets.

**Personal Blogs:**

* Developing templates for personal blogs where individuals can publish articles, essays, or diaries.