Introduction :

Today marked the 11th day of my internship at Surfboard Payments. The focus of the day was on improving my problem-solving skills in JavaScript by working on 15 different JavaScript problems. I also learned how to run my JavaScript code in Visual Studio Code VS Code and how to push my work to GitHub. This was an important milestone for me because it helped me practice JavaScript concepts, improve my coding efficiency, and get familiar with using VS Code and GitHub for project management.

Learning JavaScript Problem-Solving

Problem-solving is one of the most important skills for any programmer. To improve my skills, I worked on 15 JavaScript problems covering different topics such as loops, arrays, objects, functions, and conditional statements.

Some of the problem-solving topics I covered included

1 Variables and Data Types

I worked on problems related to declaring variables using let, const, and var. I also practiced using different data types such as strings, numbers, and booleans.

2 Conditional Statements

I solved problems that involved if, else if, and else statements to make decisions in the code. This helped me understand how to control the flow of a program based on different conditions.

3 Loops for and while

Loops are essential in JavaScript for iterating over data. I worked on problems that required using for loops and while loops to repeat actions multiple times.

4 Arrays and Array Methods

I solved problems related to arrays, such as adding elements, removing elements, and using array methods like map, filter, reduce, and forEach. These methods make it easier to manipulate arrays and process data efficiently.

5 Functions

Functions help in writing reusable code. I worked on problems where I had to create functions, pass parameters, return values, and use arrow functions.

6 Objects and Object Manipulation

I practiced working with objects by creating key-value pairs, accessing properties, updating values, and using Object keys and Object values methods.

7 String Manipulation

Some problems focused on string operations like concatenation, splitting strings, changing case uppercase lowercase, and using string methods such as trim, slice, and replace.

By solving these problems, I improved my ability to think logically and write efficient JavaScript code.

Running JavaScript Code in Visual Studio Code

After solving the problems, I learned how to run my JavaScript code in Visual Studio Code. Here's what I did

1 Installed Node.js

Since JavaScript runs in a browser, I needed Node.js to run JavaScript code in VS Code. I checked if Node.js was installed by running node -v in the terminal.

2 Created a JavaScript File

created a new file named problems.js inside my VS Code workspace and wrote all the JavaScript solutions in it.

3 Running the JavaScript File

I used the terminal in VS Code and executed the command node problems.js. This allowed me to see the output of my JavaScript solutions directly in the terminal.

4 Debugging and Fixing Errors

While running my JavaScript code, I encountered some errors, such as syntax errors and logic errors. I used console.log to debug and understand what was going wrong in the code. Fixing errors helped me learn the importance of testing and debugging code before finalizing it.

Pushing My JavaScript Code to GitHub

Once I successfully ran my JavaScript solutions in VS Code, I learned how to push my work to GitHub. This is an essential skill for developers because GitHub allows version control and collaboration.

Steps I Followed to Push My Code to GitHub

1 Initialized a Git Repository

Inside my project folder, I opened the terminal in VS Code and ran the command git init. This initialized a Git repository in my project directory.

2 Added Files to the Staging Area

I added my JavaScript file to Git by running git add problems.js. This prepared the file to be committed.

3 Committed the Changes

I committed my changes with a meaningful message git commit -m Solved 15 JavaScript problems. This saved my changes in the local repository.

4 Created a GitHub Repository

I logged into GitHub, created a new repository, and copied the repository URL.

5 Linked My Local Repository to GitHub

In the terminal, I linked my local repository to GitHub using the command git remote add origin repository-URL. This connected my project to GitHub.

6 Pushed My Code to GitHub

Finally, I pushed my code to GitHub by running git push -u origin main. My JavaScript solutions were successfully uploaded to my GitHub repository.

By following these steps, I learned how to track changes in my code and share my work with others.

Challenges Faced and How I Overcame Them

1 Debugging Errors in JavaScript

Some of the JavaScript problems had tricky conditions, and I made mistakes in the logic. I used console.log to test different parts of the code and fixed the errors step by step.

2 Understanding Git Commands

Initially, I was confused about the difference between git add, git commit, and git push. I practiced using these commands multiple times, which helped me understand their functions.

3 Pushing to GitHub

I faced an issue where GitHub asked for authentication before pushing my code. I solved this by setting up my GitHub credentials and using a personal access token for authentication.

Key Takeaways from Day 11

1 Problem-solving in JavaScript Solving coding problems helped me improve my logical thinking and understanding of JavaScript concepts.

2 Using VS Code for JavaScript I learned how to write, run, and debug JavaScript code efficiently.

3 Git and GitHub Workflow I understood how to use Git for version control and push my code to GitHub.

4 Debugging Skills Learning how to find and fix errors in JavaScript code was an important lesson.

Conclusion

Day 11 of my internship was a productive day where I strengthened my JavaScript problem-solving skills, learned how to run JavaScript code in VS Code, and successfully pushed my work to GitHub. These are fundamental skills that will help me in future development projects. Moving forward, I aim to solve more coding challenges, improve my efficiency, and explore advanced JavaScript concepts.