HCDE 438

Deveshi Modi

Reflection on Learning Outcomes

Building my AI Color Palette Generator taught me a lot about full-stack web development that I hadn't experienced before. This project let me take what we learned in HCDE 438 and actually build something complete, which helped me understand how everything connects in real web development.

The biggest area where I improved was working with React. I had used it before, but not for anything this comprehensive. I got much better at creating components that could be reused, managing state with useState, and using useEffect for things like API calls. One thing that really helped was learning to keep components focused on one thing. Early in the project, I had some components that were doing too much like my Color Card to display the generated colors, which made debugging really difficult. When I broke them down into smaller pieces, everything became much easier to manage and fix.

Building the backend was completely new for me. Instead of calling the Gemini API directly from the frontend, I created an Express server to handle those requests as I could not directly call the API. This was something I had not known about before but learned through debugging the API. This was important for security, but it meant learning how to make the frontend and backend communicate properly. I had to figure out how to send data back and forth, deal with CORS issues, and keep API keys secure. I had never used fetch on the server side before, so that took some time to understand. Tools like curl and Postman were really helpful for testing when things weren't working in the browser.

The Gemini API integration was interesting but challenging. Since it returns natural language responses, I had to be very specific in my prompts to get back the format I needed. Sometimes the API would return extra text or not format things as JSON, so I had to add error handling for those cases. This made me think more carefully about error handling throughout the whole application.

For the interface, I used Material UI, which helped me create a clean design without having to write all the CSS myself. I learned how to use their theming system to keep everything consistent and worked with their layout components to make sure the app worked well on different screen sizes. Making it responsive was important since people might use it on their phones.

Deploying the app was a big step. There were a lot of things to consider when moving from my local development setup to something that would work for other people. I had to handle environment variables properly, make sure everything was bundled correctly, and test the live version thoroughly.

Looking back, I would approach some things differently. I should have planned out the backend structure earlier instead of focusing so much on the frontend first. I didn't realize how complex the API integration would be. I also wish I had set up testing and better logging from the start, which would have saved me time debugging later.

This project connected well with what we covered in HCDE 438. I used React Router for navigation, hooks for state management, and applied the responsive design principles we discussed. Concepts like error handling and user feedback were important parts of my implementation. It felt good to take the theory and actually apply it to build something real.

Overall, this project gave me confidence in full-stack development and showed me how to build, test, and deploy a complete application. I learned how to solve problems I hadn't encountered before, whether that was debugging API issues, organizing my code better, or thinking about what users actually need. It also made me realize I enjoy working on both the frontend and backend, which is something I want to continue exploring.