Hi Capo, I'm recording a video to explain my plan, what actually happened, and what was interesting about the experience when I completed my assignment.

Initially, I planned to create a chess game that would engage the Chess Lover community. Although I'm not part of any specific official Chess Lover club, I do occasionally play chess with my friends. However, I lost interest because chess isn't really my passion, and I wanted to use this assignment to do something that truly engages me. I really enjoy painting and natural landscapes, so I thought about creating a platform where people in my community could generate and display their natural landscape art pieces to each other using Generative AI—a technology that I find both curious and amazing.

So, I planned to use JavaScript to develop a web application that allows users to input prompts and receive images using a text-to-image AI model on the main page. I also wanted to create a gallery page to showcase the art pieces generated by users.

Here's what I did: First, I figured out how to implement the front end for my main page. This required an input text field, a submit button to submit the input, and a gallery button to navigate to the gallery page. After pressing the submit button, the generated image would be displayed below, so I created an empty space for it to appear. Initially, I planned to implement buttons for users to submit their images to the gallery, but I realized this would require hosting a server, which wasn't feasible. Instead, I decided to leave a message saying, "Save the image and send it to me to be featured in the Gallery."

I encountered a few problems with CSS styling but was able to resolve them. Now, the most important part: how can I get AI models to generate images for me? This implementation requires using JavaScript to handle the API call and button press. As you can see in the code here, I need to first retrieve the prompt, then pass it to my API calling function when the submit button is pressed. It sends my prompt to the Dall-E 2 model hosted on the cloud by OpenAI and returns an image URL to me. I then update the src attribute of my placeholder image element on the main page with the returned image URL. I also handle the logic of navigating to the gallery page once the gallery button is pressed.