Shadman Rahman

Professor Fried

CSCI 355

Project 2 Grey paper

1.

```
 const result = await chat.sendMessage(userInput);
 const response = result.response;
 return response.text();
}
```

This is a portion where the script waits for user input before generating a response, Once the response is generated, it is converted to text.

2.

3.

```
app.use(express.static('./Homepage'))
```

Allows us to navigate through website, all webpages get posted because they all are located within the Homepage folder.

3.

```
app.post('/chat', async (req, res) => {
 const userInput = req.body.userInput;
 try {
   const responseText = await runChat(userInput);
   res.json({ message: responseText });
 } catch (error) {
   console.error(error);
   res.status(500).json({ message: "Error occurred" });
 }
});
```

Async function where input is received and converted to a json file.

4.

```
const nav = document.querySelector(".nav")
window.addEventListener('scroll', fixNav)

function fixNav(){
   if(window.scrollY > nav.offsetHeight  + 150){
       nav.classList.add('active')
   }
   else{
       nav.classList.remove('active')

   }
```

A function where once we scroll the whole first portion of the viewport, a class .active gets added changing the styling of .nav.

5.

```
   index++;
   if(index > text.length){
       index = 1
   }
   setTimeout(write,speed)
}
```

Index gets incremented and reset from if condition.setTimeout is used as an infinite loop, running based on the speed variable.

6.

```
   textEl.innerText = text.slice(0,index)
```

Select the text of the element and only show the first to index letter, used in conjunction with set timeout to have a typewriter.

7.

```
       setTimeout(() => ci.remove(),250)
```

In 250 ms remove the class/ object ci.

8.

```
const response = await fetch('/chat', {
   method: 'POST',
   headers: { 'Content-Type': 'application/json' },
   body: JSON.stringify({ userInput }),
 });

 const responseData = await response.json();
 responseDiv.textContent = responseData.message;
```

Convert json to string and output the image to the screen. Stringify is used to convert and textContent is used to place the string on the screen.

9.

```
b.addEventListener('click', function(e){
    const x = e.clientX;
    const y = e.clientY;
```

Find the exact location a click is done, an event listener check for clicks, and its x and y coordinates are saved.

10.

```
imgs.style.transform =`translateX(${-index * 500}px)`
```

Transform the image to the left 500px based on the index, overtime we will shuffle through all images, but will repeat infinitely.

A few struggles I have had throughout this project are the routes for the different webpages. At first I had multiple folders for each page, but I would always get a post error. So putting it all in one folder was the solution. Also, connecting a database to my project was a big struggle.