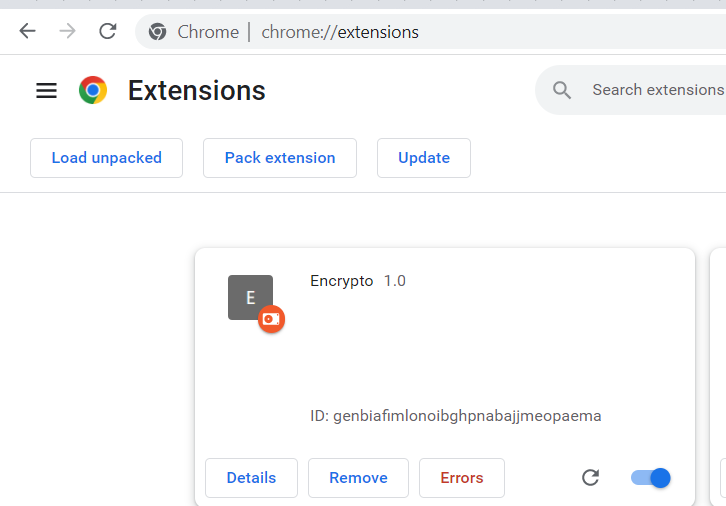
**Goal** : In this task, we ask you to prototype a chrome extension that will encrypt data before it is uploaded to a website. You can assume the web uploader has a meta tag in the webpage source that says it uses our service, e.g., “”. The chrome extension should see this tag and send it (POST) along with the user’s API-key to some Oblivious API (you can mock this however way you like) and receive back a 256bit encryption key. This key must then encrypt the data before uploading it to the server. In practice, each “data-obliv-encrypt” ID would be linked to a specific Docker container

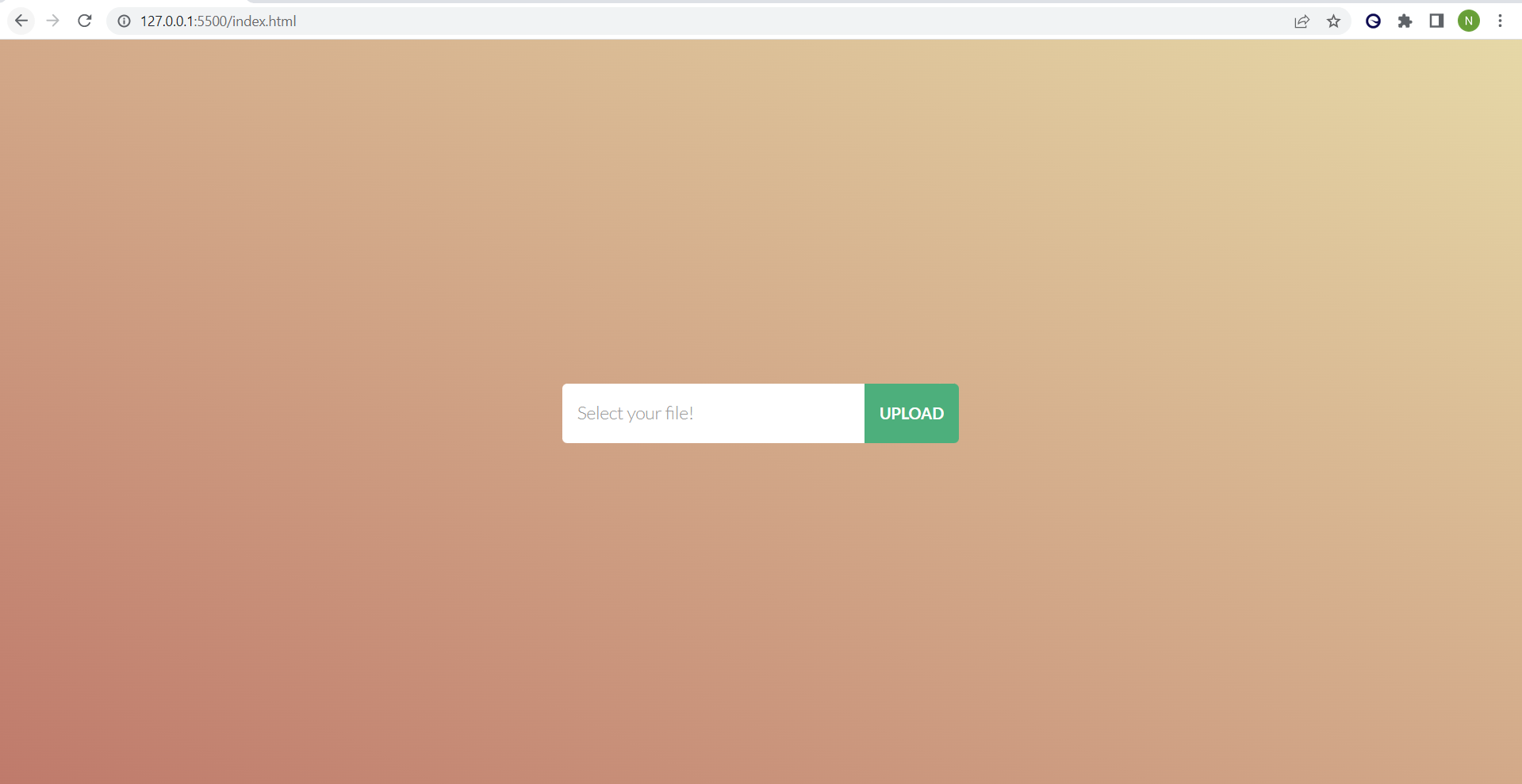
**Solution** : I have developed a browser addon called ‘Encrypto 1.0’. It can be activated by following steps -

For Chrome users:

* In the address bar search for this: chrome://extensions.
* Enable Developer Mode and switch into it.
* Click the Load unpacked button and select the extension directory.

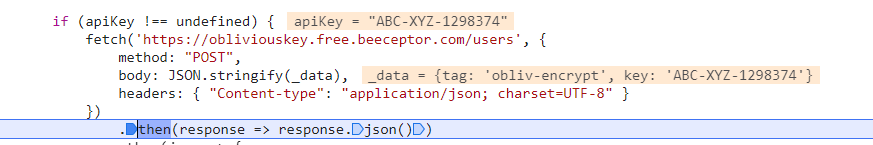


After the extension is activated, Run the index.html file and try to upload any test file using upload as shown in the below image.

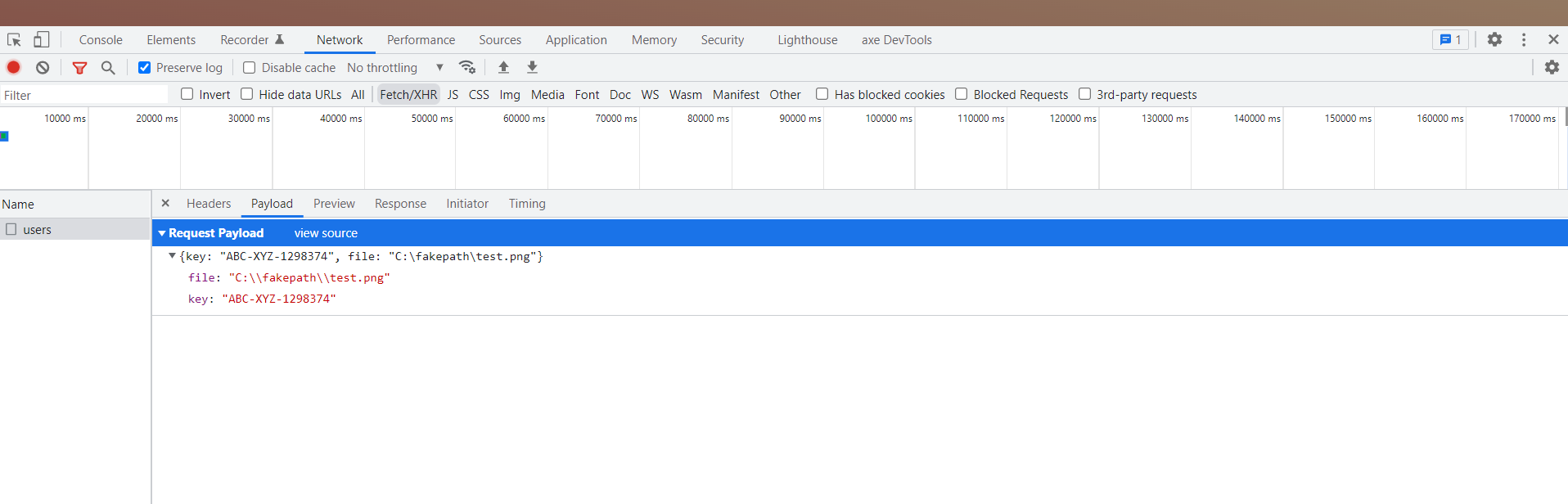


Once you upload a file, following steps happens in in the background:

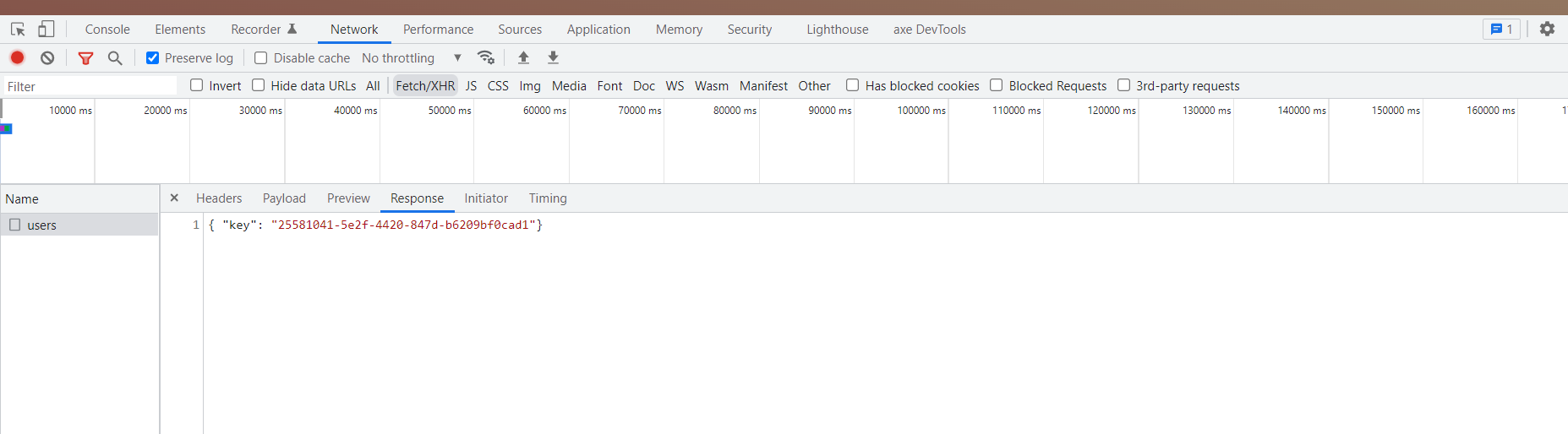
* The browser extension checks for the data attribute “data-obliv-encrypt”.
* If it is available then send that attribute along with its value(API key) to a mock API that would return the encryption key. <https://obliviouskey.free.beeceptor.com/users> making a POST call
* API call that returns secret key based on value passed.



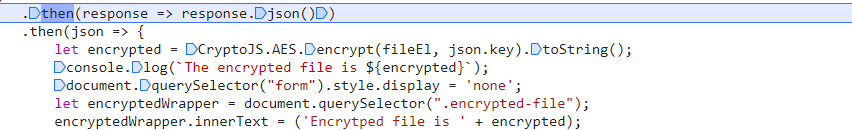
Payload data



Response with the secret key



* Once the fetch is completed based on the response I have included a CryptoJS plugin which encrypts the file taking the response secret key and encrypted value is replaced on the page for viewing it but ideally that would be sent to the server.



* I have included the decryption logic also which would be a follow up task at the endpoint.

.