

The flow of Registration and Signin:

The information of the user will be stored in Firestore Database, the connection will happen inside the expressBackendServer and displayed in the ReactFrontEnd

Backend:

Express:

Route:

```
route.post( '/signin', async(res, req) =>{  
  await pull data from Firestore  
  res.(sign in successful ? fail)  
})  
route.post( '/register', (res, req) =>{  
  await pull data from Firestore  
  res.(register + sign in successful ? fail)  
})
```

DB

FirestoreDB:

Connect to FireStore:

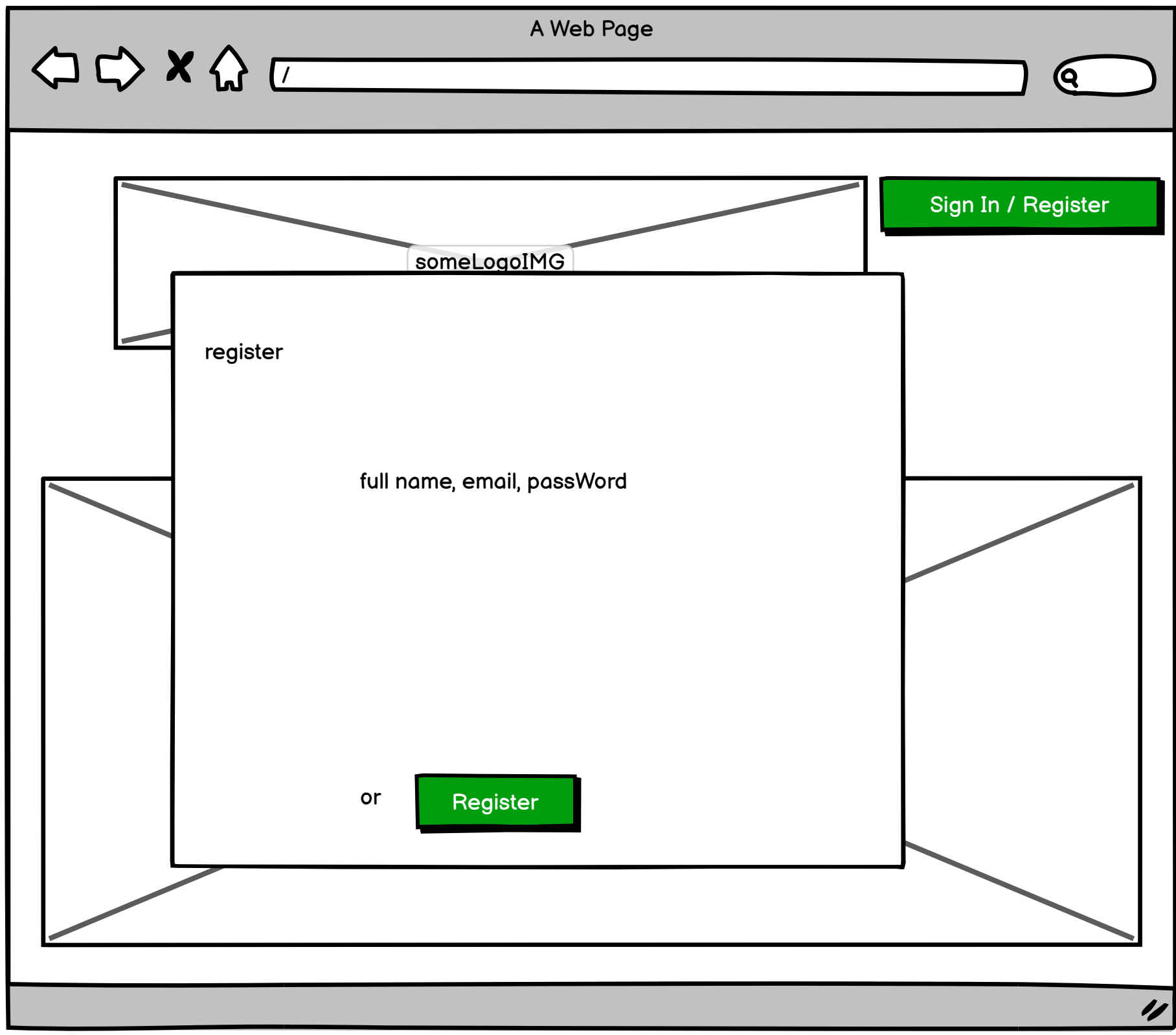
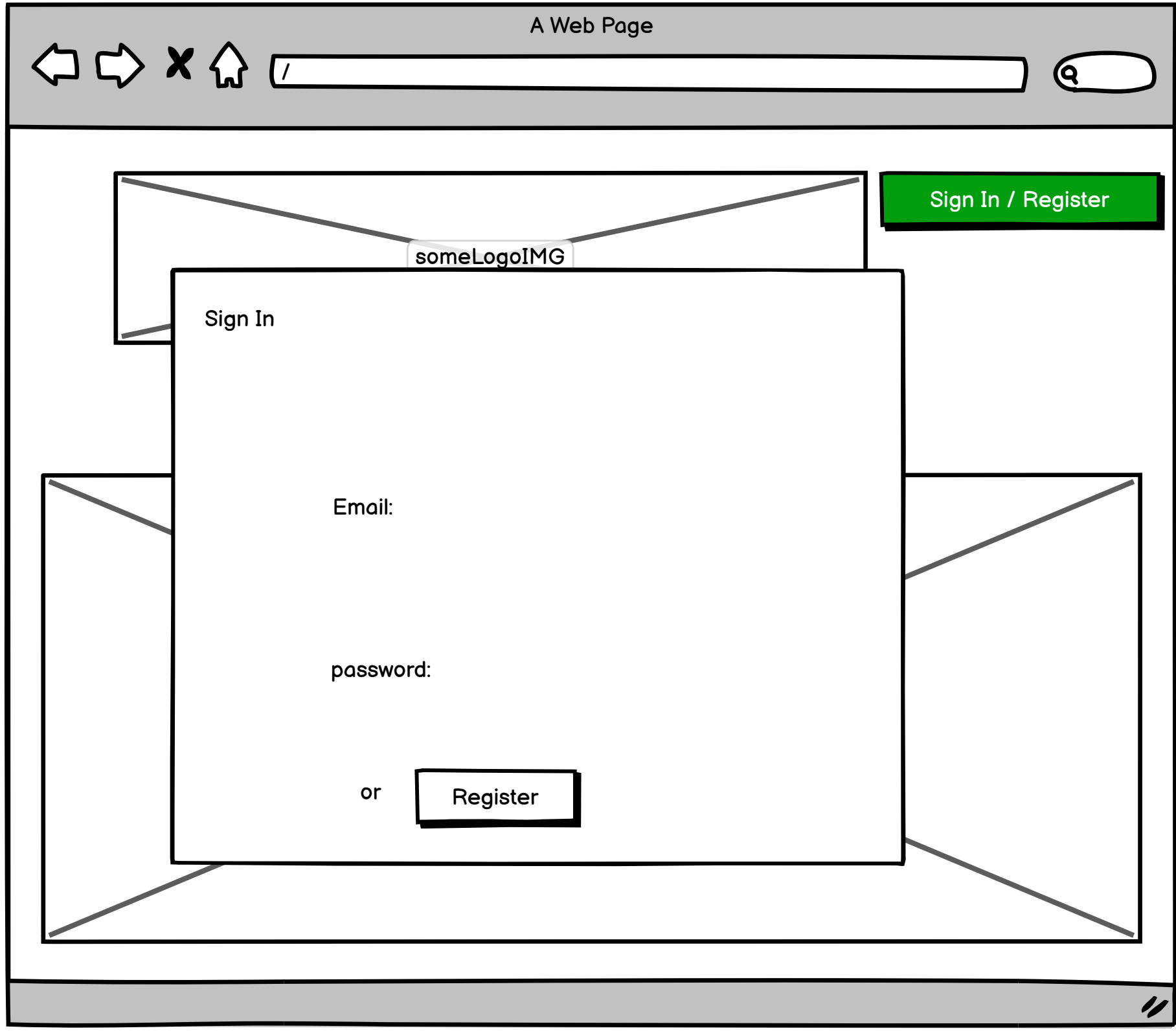
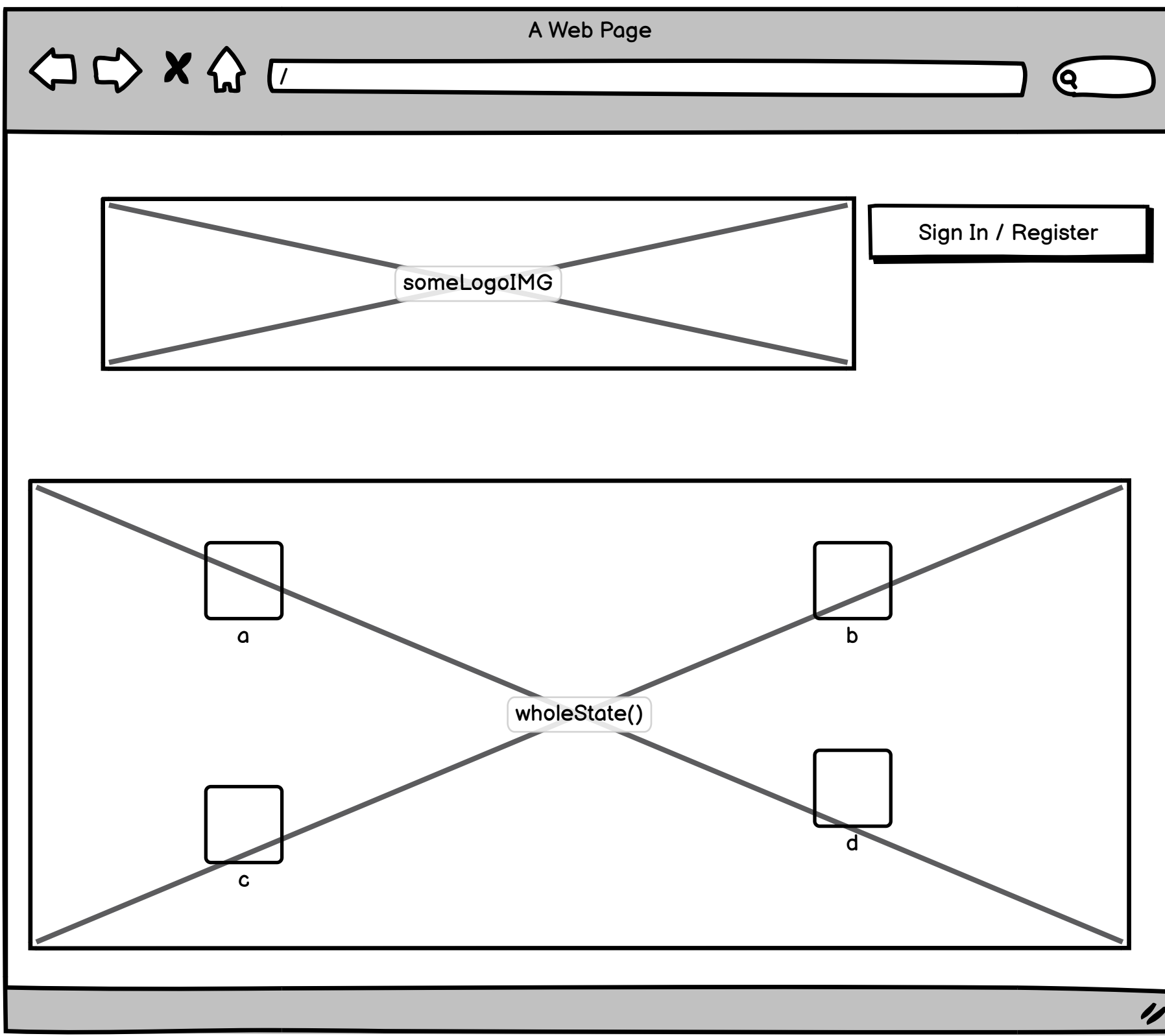
docs: <https://firebase.google.com/docs/firestore/quickstart#node.js>

FrontEnd:

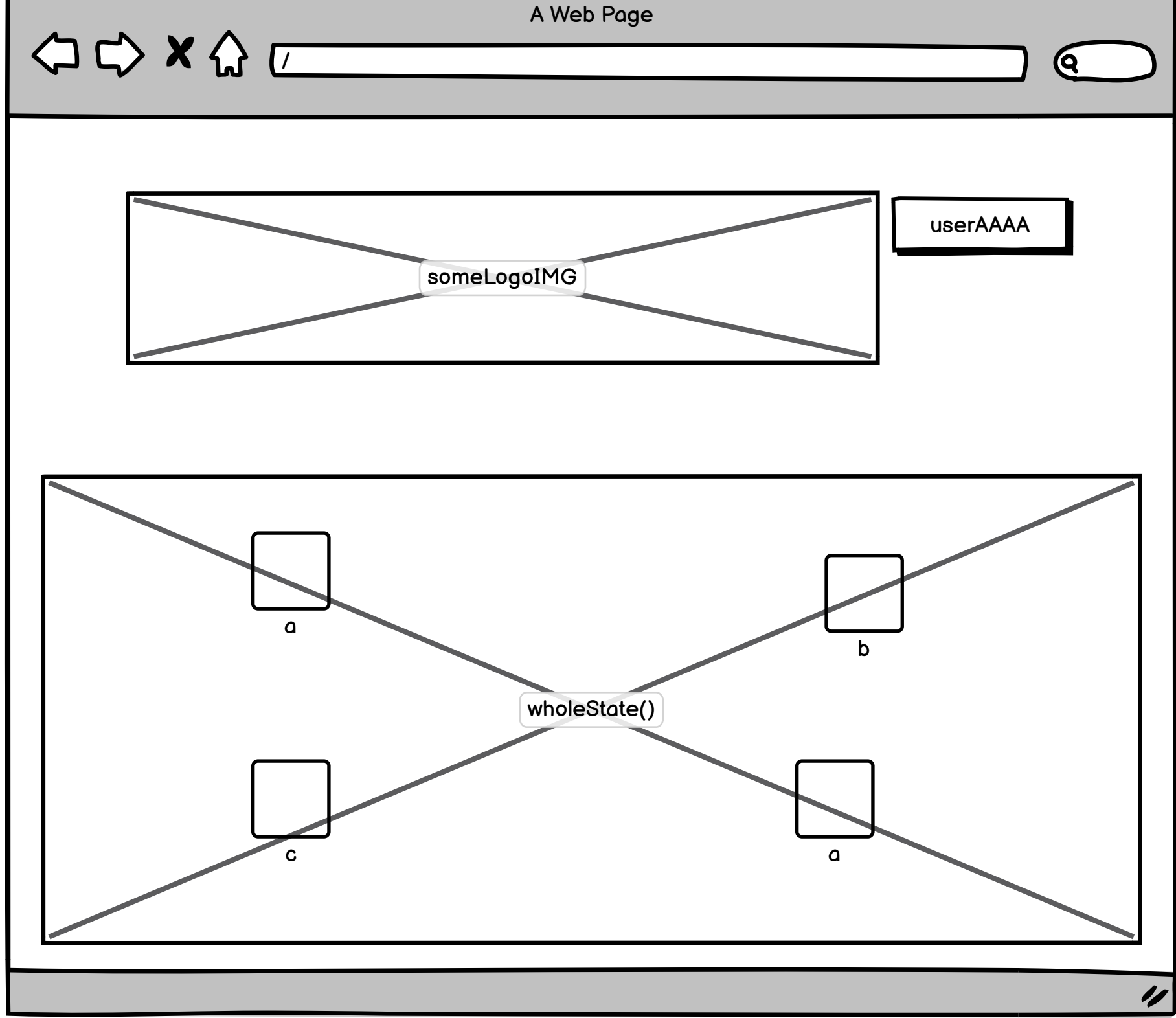
React:

module:

- sign-in Form
logic: onSubmit(form.post('url/signin'))
style: React Boot Strap Form Module
- registration Form
logic: onSubmit(form.post('url/register'))
style: React Boot Strap Form Module



To Store login info / register info, db will be FireStore



This is the flow of how a logged-in user can read posts, make posts, edit, and delete their post.

Each user has 'unique_id' from the Firestore that becomes a query parameter in the MongoDB Collection

```
query = { user_id: "firebase.mongoID" }
```

Backend:

Express:

Routes:

```
route.get('/.firebaseMongoID/mypost', async(req, res) =>{
  await mongoDBInterface.create(req.body)
})
route.post('/.firebaseMongoID/create', async(req, res) =>{
  await mongoDBInterface.create(req.body)
})

route.put('/.firebaseMongoID/update/:postId', async(req, res) =>{
  await mongoDBInterface.updatePost(req.body)
})

route.delete('/.firebaseMongoID/delete/:postId', async(req, res) =>{
  await mongoDBInterface.deletePost(req.body)
})
```

FrontEnd:

React:

Modules:

myPost.jsx

style: React Boot Strap Cards

logic: fetch(url/mongoID/mypost).then(populate using react bootstrap card)

createPost.jsx

style: React Boot Strap Cards

logic: docs:

<https://jasonwatmore.com/post/2020/02/01/react-fetch-http-post-request-examples>

deletePost.jsx

style: Material Ui buttons

logic: doc:

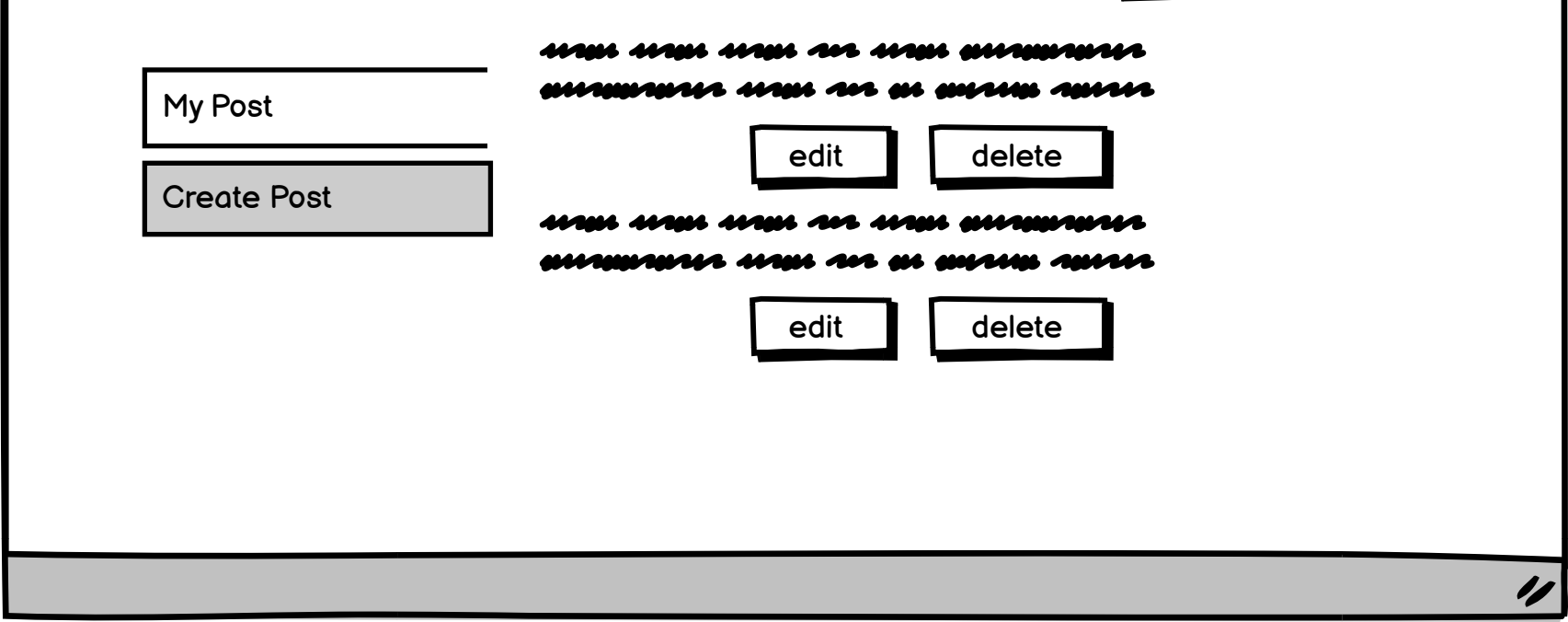
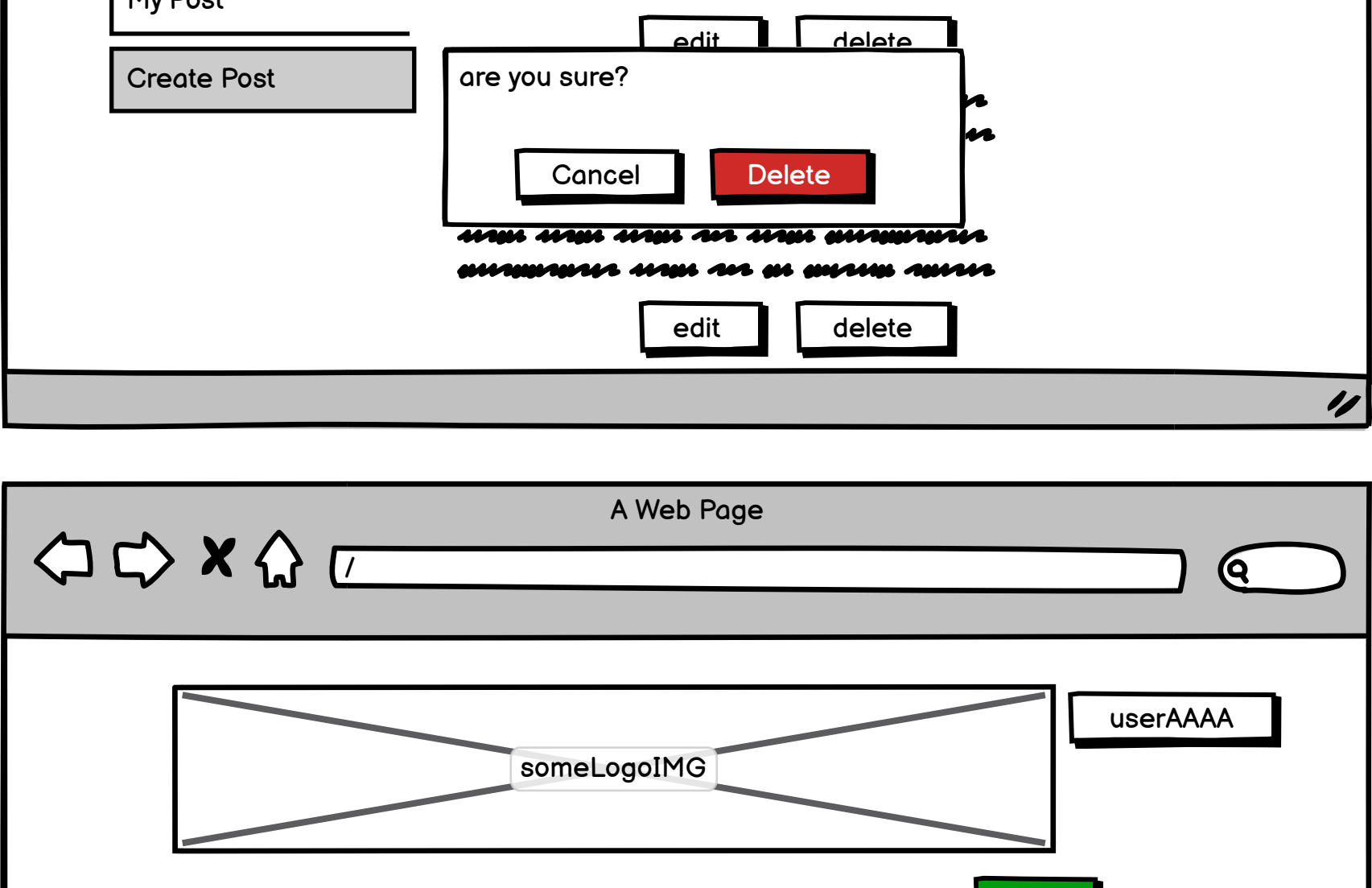
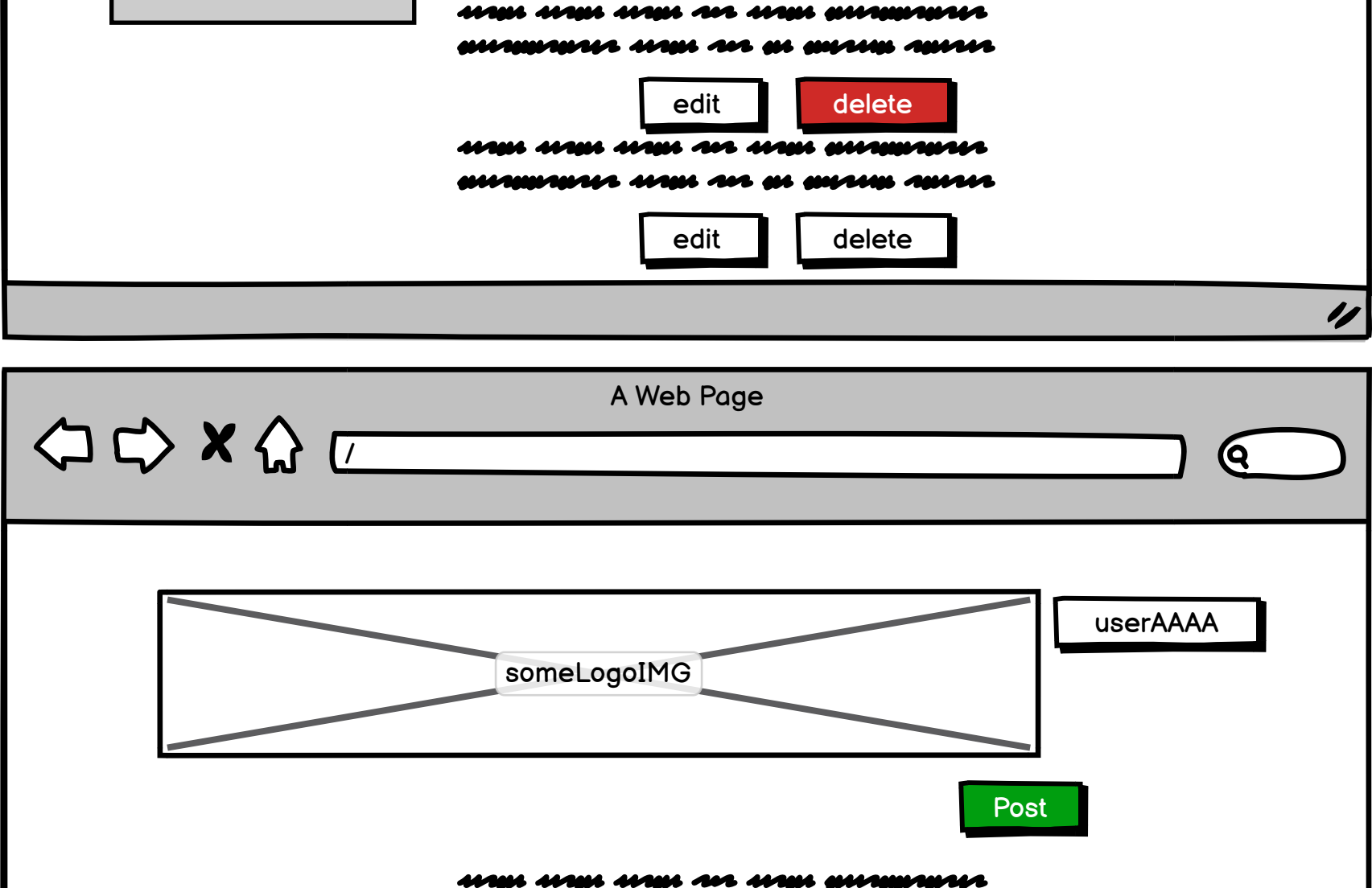
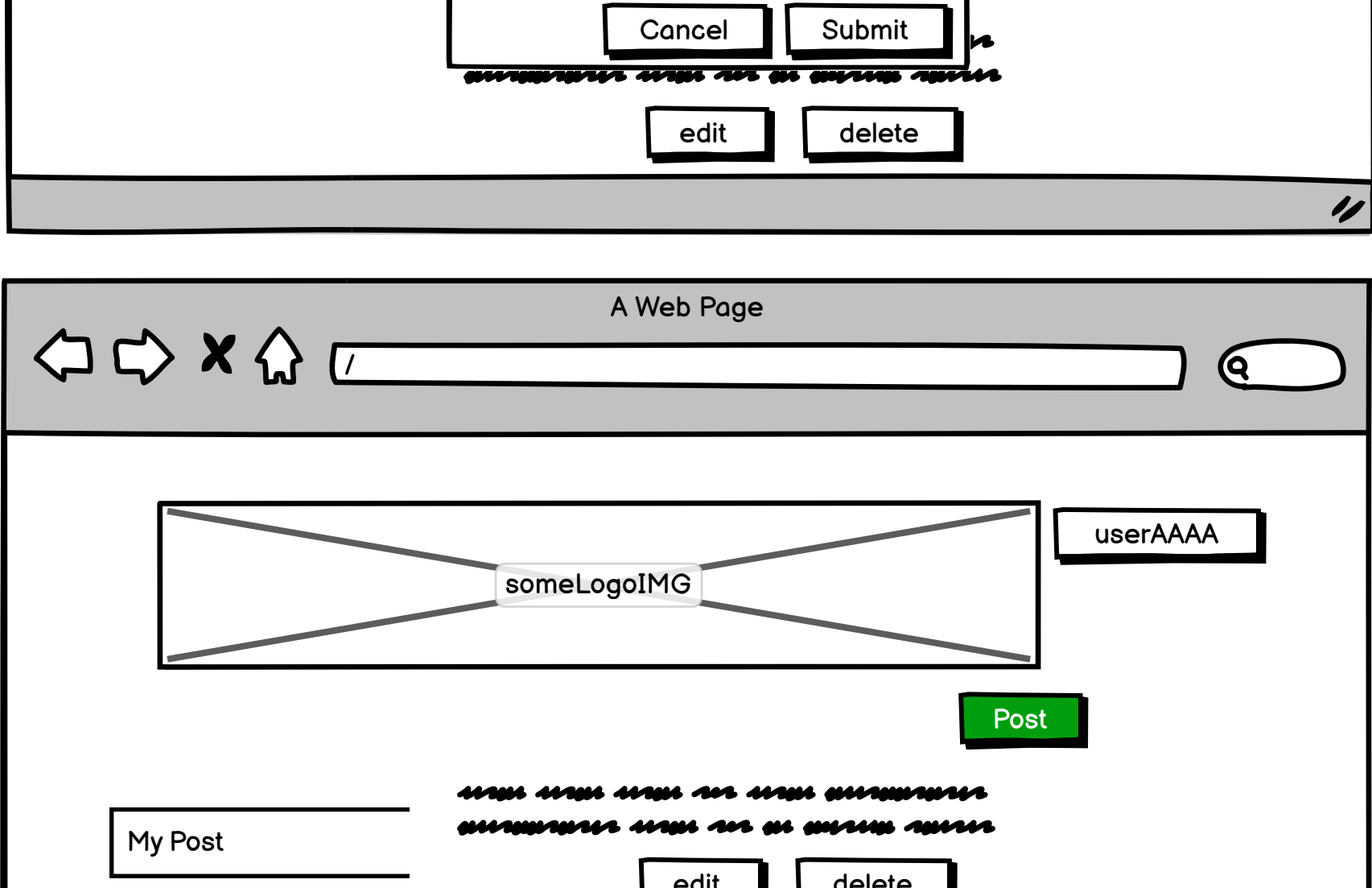
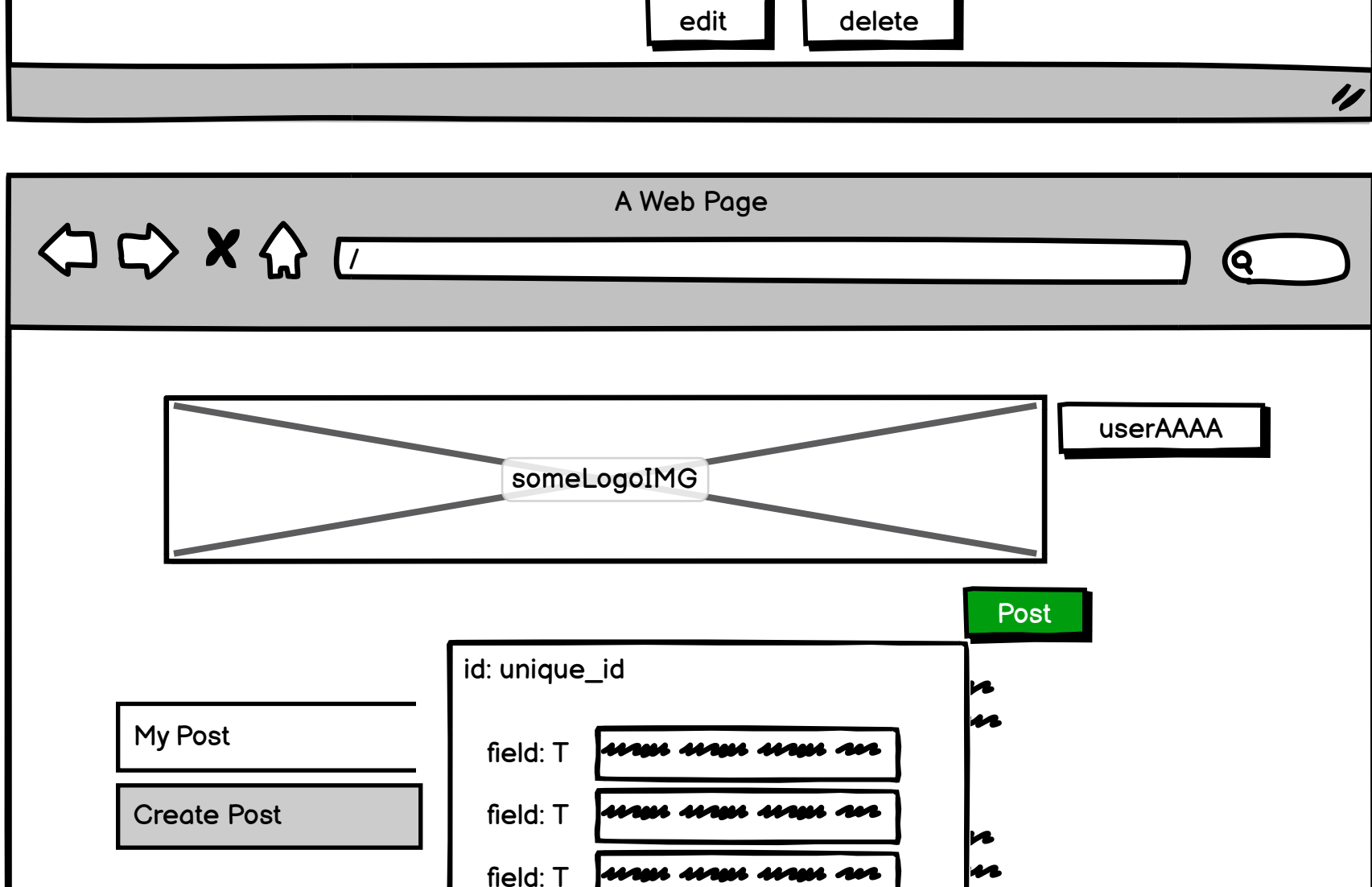
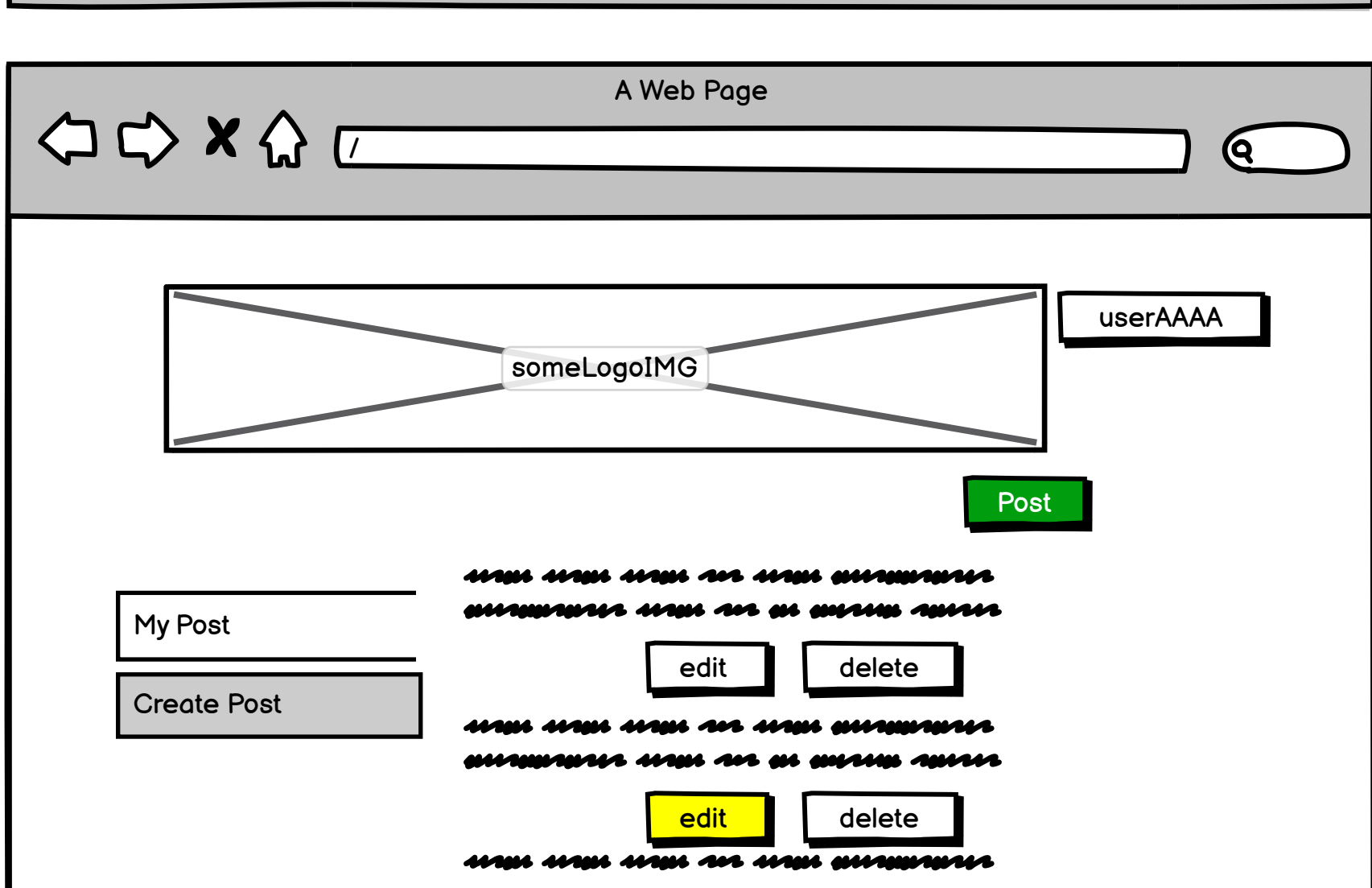
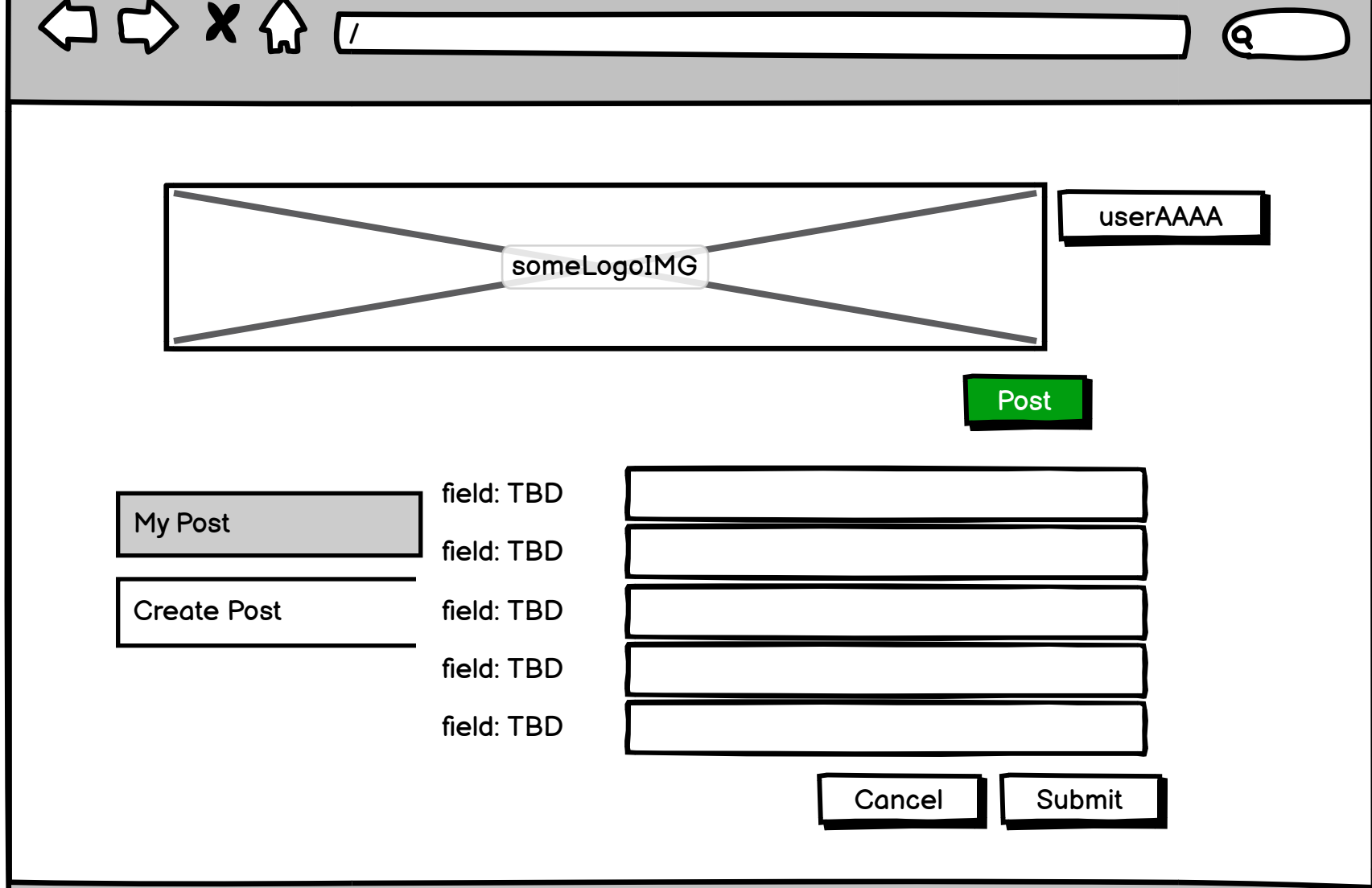
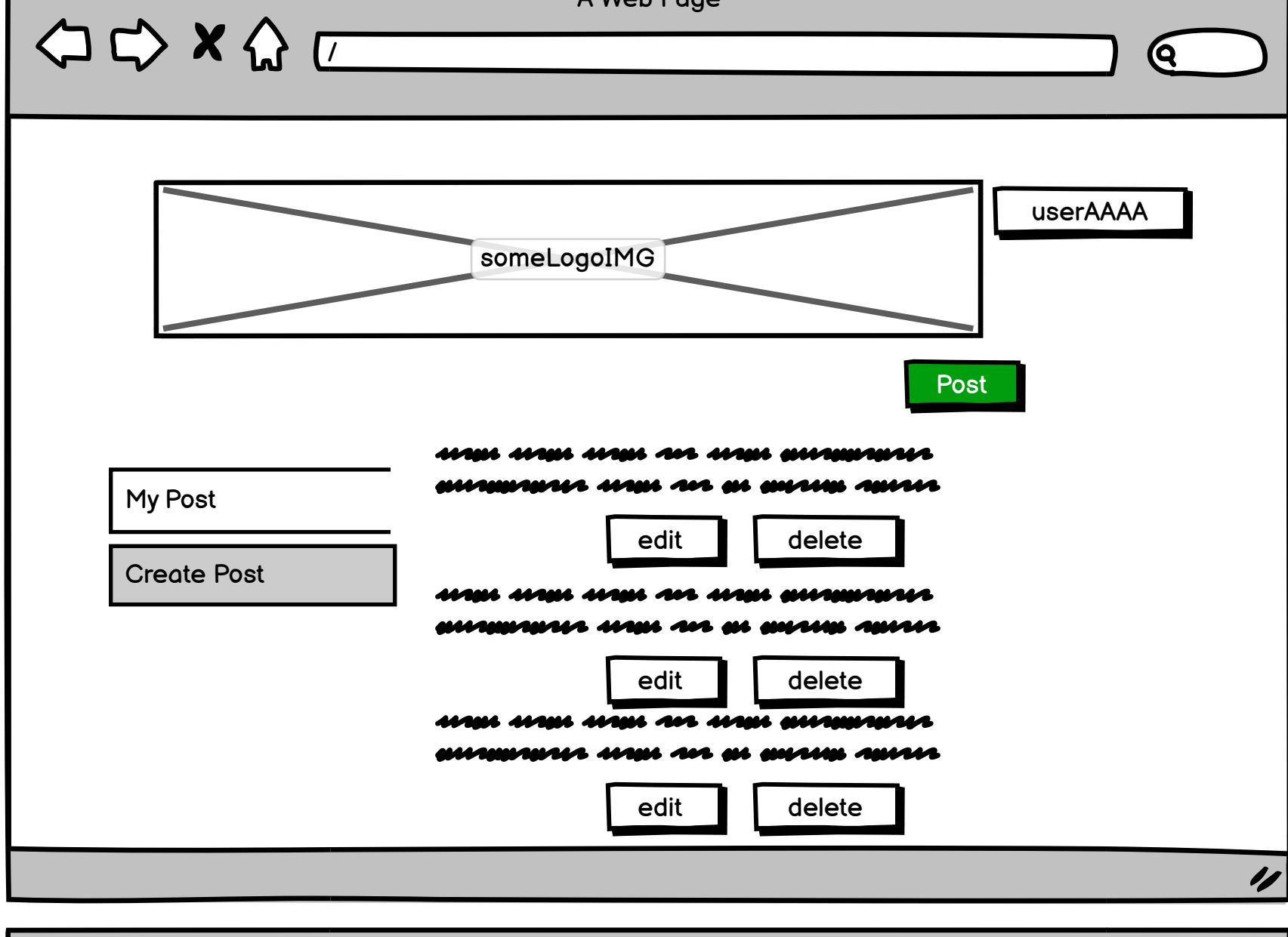
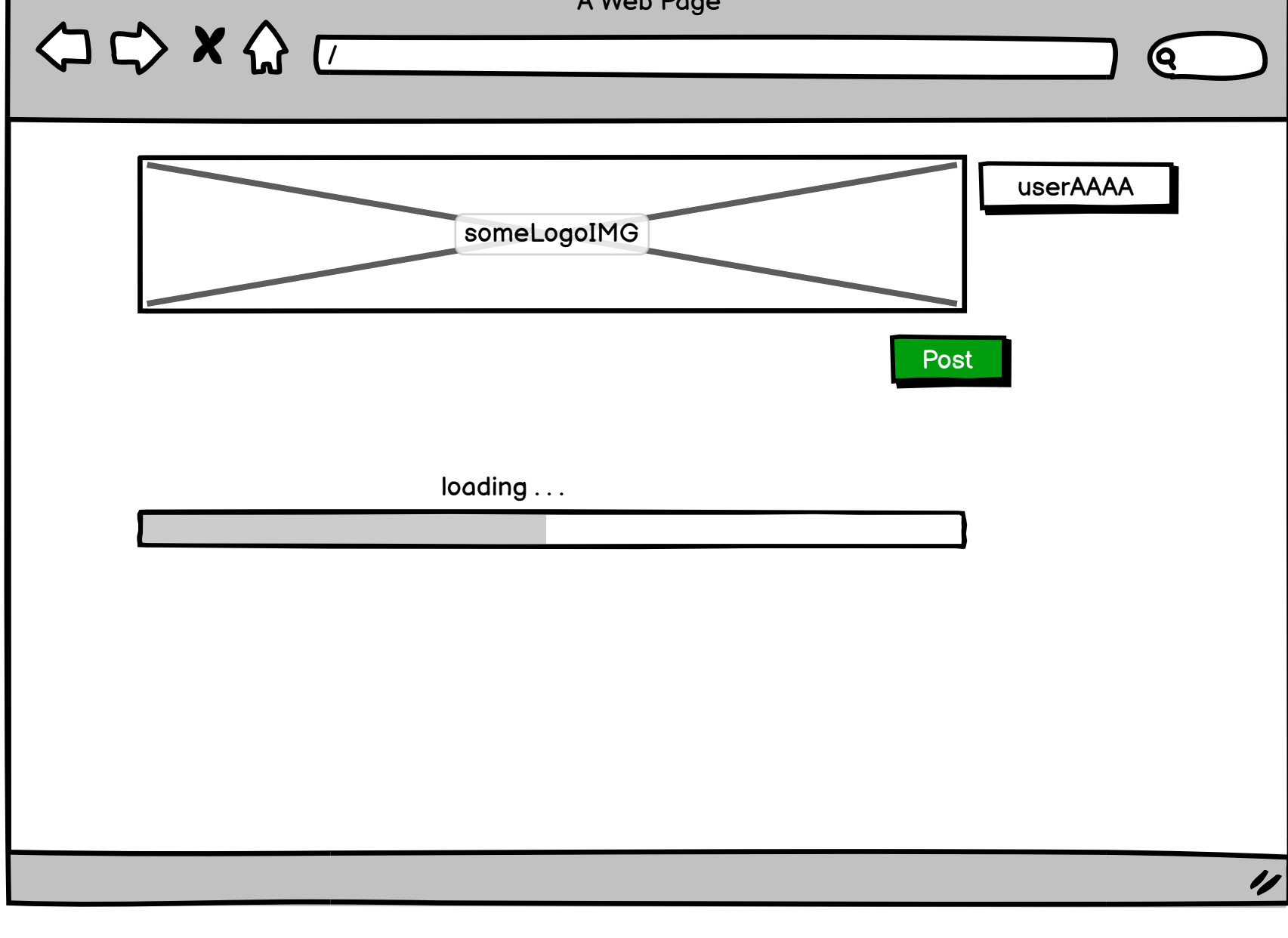
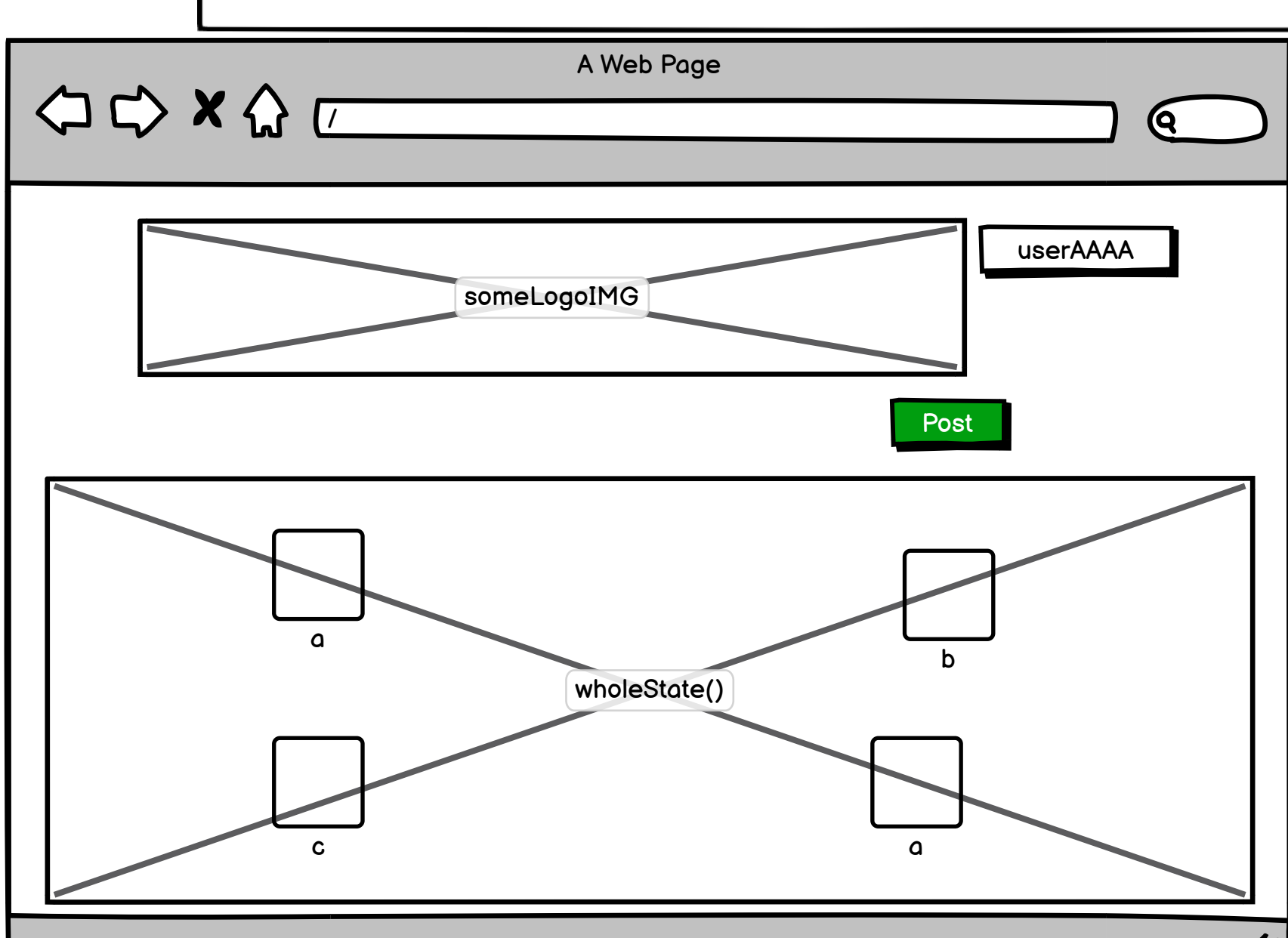
<https://jasonwatmore.com/post/2020/11/11/react-fetch-http-delete-request-examples>

updatePost.jsx:

style: Material Ui buttons + React Bootstrap Cards

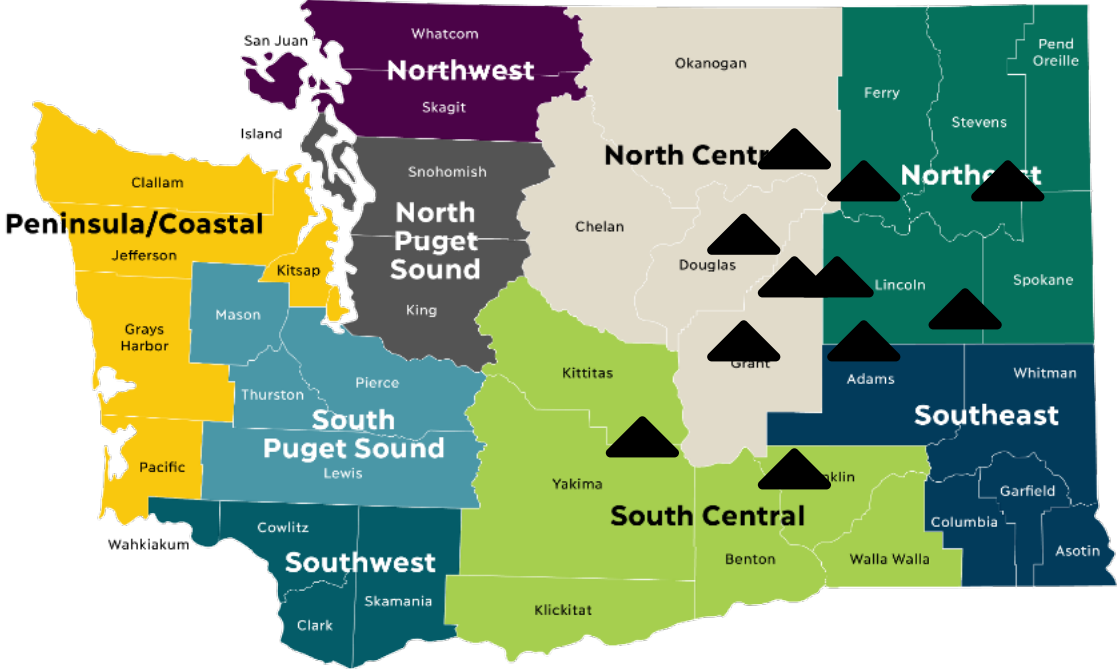
logic: docs:

<https://jasonwatmore.com/post/2020/11/02/react-fetch-http-put-request-examples>





Sign In / Register



.onclick(▲)



Sign In / Register

