

Exercises for Session 6

Goals

This session will be all about Firebase. We'll read the user-entered data from our database, upload files and implement authentication.

Cloud Firestore: Reading data

Add another button to your website (build in Session 6) that reads alls data from your collection:

```
<button onclick="loadAllData()">Load all entries</button>
```

To unconditionally load all data we're going to use `get` :

```
database
  .collection(collectionName)
  .get()
  .then(snapshot => {
    snapshot.forEach(doc => {
      // create a `pre` element
      // set the innerText to `JSON.stringify(doc.data())`
      // append the element to the body
    })
  })
```

More information: https://firebase.google.com/docs/firestore/quickstart#read_data

Cloud Storage for Firebase

Let's start by creating a new storage bucket through the Firebase web UI. You can follow this tutorial: <https://firebase.google.com/docs/storage/web/start>

Extend your form with another `<input>` element. This element should have the `type="file"` so the user will be prompted to upload a file.

When the user clicks on the submit button:

- First upload the image to your Cloud Storage
- Then get the download URL of the image
- Lastly create a database entry with the form values and the download url

When reading the data make sure to also display the images (if available).

Firestore Authentication

Please create a new HTML file for the Authentication exercise. We won't be building much of our own UI but use everything that Firebase UI has to offer.

First, add the following dependencies to your `head` :

```
<script src="https://www.gstatic.com/firebasejs/7.0.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.0.0/firebase-auth.js"></script>
<script src="https://cdn.firebase.com/libs/firebaseui/4.2.0/firebaseui.js"></script>
<link type="text/css" rel="stylesheet"
href="https://cdn.firebase.com/libs/firebaseui/4.2.0/firebaseui.css" />
```

Initialize your firebase app as you did in the previous exercise. Then initialize Firebase Authentication and the Firebase UI:

```
const auth = firebase.auth()
const ui = new firebaseui.auth.AuthUI(auth)
```

Create an empty with the id `auth-container` and a button that calls an `onSignIn` function when clicked.

```
function onSignIn() {
  ui.start('#auth-container', {
    signInOptions: [firebase.auth.EmailAuthProvider.PROVIDER_ID],
    signInSuccessUrl: window.location.href
  })
}
```

Before you can test it we need to implement a listener for the login:

```
auth.onAuthStateChanged(function(user) {
  if (user !== null) {
    console.log('Signed in:', user)
  } else {
    console.log('Signed out')
  }
})
```

Make sure your Firebase project supports email registrations. More information:

<https://firebase.google.com/docs/auth/web/firebaseui>

The next functionalities we want to have are:

- Show a greeting for the logged in user
- Replace the login button with a logout button
- On logout: show the login button again and hide the greeting