

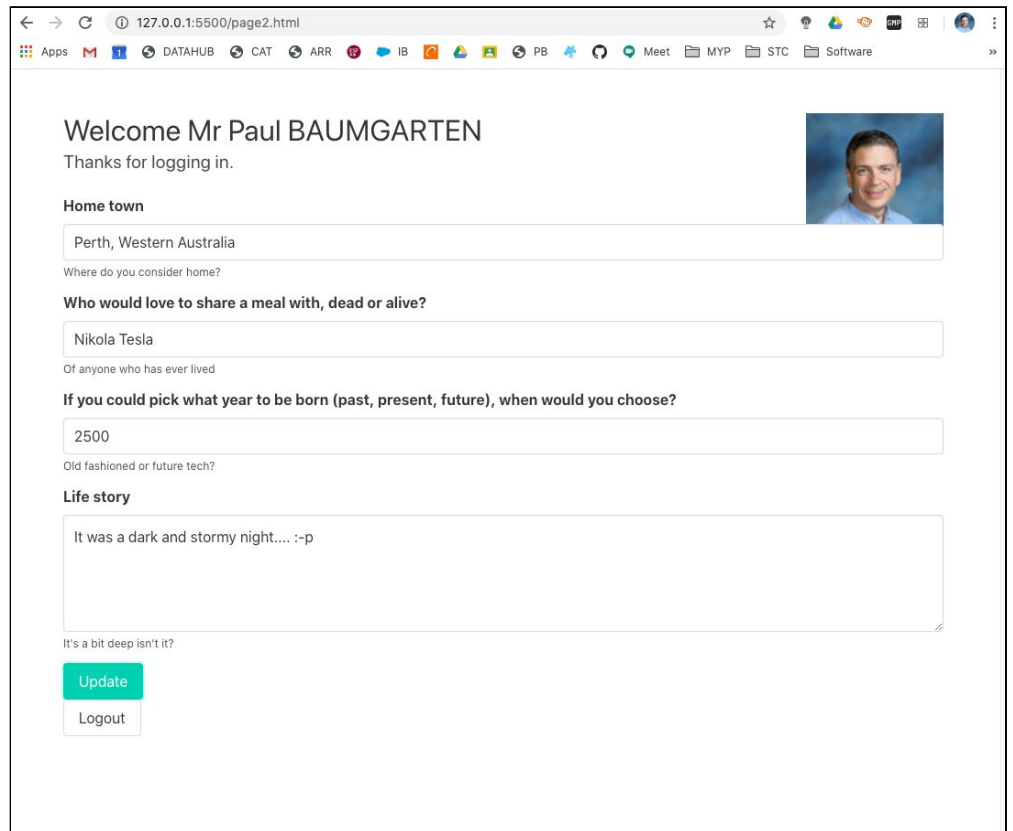
Firestore database (Cloud Firestore) guide

As of 01/05/2020

End goal

On page load, it will retrieve the values for the logged in user from Firestore.

On clicking the save button, it will save the values in the form fields to the Firestore.



The screenshot shows a web browser window with the URL 127.0.0.1:5500/page2.html. The page displays a user profile for 'Mr Paul BAUMGARTEN' with a profile picture. The form includes the following fields and labels:

- Welcome Mr Paul BAUMGARTEN**
Thanks for logging in.
- Home town**
Perth, Western Australia
- Where do you consider home?**
- Who would love to share a meal with, dead or alive?**
Nikola Tesla
- Of anyone who has ever lived**
- If you could pick what year to be born (past, present, future), when would you choose?**
2500
- Old fashioned or future tech?**
- Life story**
It was a dark and stormy night.... :-p
- It's a bit deep isn't it?**
- Update** (button)
- Logout** (button)

Setup Firestore database

This assumes you have completed the Firestore authentication steps already

1. In the firestore console, open your project and go to Database / Create
2. Start in production mode
3. Set location (note: for my students, asia-east2 is HK)
4. Open the rules tab... set a rule (see examples below)

Firestore database permission rules

Permissions rules tend to be the most frustrating part of getting this to work properly. While learning, I recommend just using the first example one for at least write permissions. It does mean userA could potentially overwrite something from userB, but at least they are logged in. When you are ready to secure further you can study the docs further.

Some example rules for the most common scenarios are provided below. Make sure you replace the "some_collection" text with the name of your collection you want the rule to apply to.

For more detail refer to the Firestore rules documentation], which is where these examples have come from.

- <https://firebase.google.com/docs/rules>

This rule allows any user who is signed in to read and write any record.

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if request.auth.uid != null;
    }
  }
}
```

This rule restricts access to the authenticated owner of the content only. The data is only readable and writable by one user, and the data path contains the user's ID.

```
service cloud.firestore {
  match /databases/{database}/documents {
    // Allow only authenticated content owners access
    match /some_collection/{userId}/{documents=**} {
      allow read, write: if request.auth.uid == userId
    }
  }
}
```

This rule allows anyone (public) to read a data set, but restricts the ability to create or modify data at a given path to the authenticated content owner only.

```
service cloud.firestore {
  match /databases/{database}/documents {
    // Allow public read access, but only content owners can write
    match /some_collection/{document} {
      allow read: if true
      allow create: if request.auth.uid == request.resource.data.author_uid;
      allow update, delete: if request.auth.uid == resource.data.author_uid;
    }
  }
}
```

This rule allows anyone signed in (authenticated) to read a data set, but restricts the ability to create or modify data at a given path to the authenticated content owner only.

```
service cloud.firestore {
  match /databases/{database}/documents {
    // Allow public read access, but only content owners can write
    match /some_collection/{document} {
      allow read: if request.auth.uid == userId;
      allow create: if request.auth.uid == request.resource.data.author_uid;
      allow update, delete: if request.auth.uid == resource.data.author_uid;
    }
  }
}
```

page2.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.8.2/css/bulma.min.css">
    <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
    <style>
      /* Styling the photo element*/
      img#profilePic {
        position: absolute; /* Force location on page */
        top: 0;             /* At top of page */
        right: 0;           /* At right of page */
        width: 150px;       /* Force size to 150px */
        height: 150px;
      }
    </style>
  </head>
  <body>
    <section class="section">
      <div class="container">
        <h1 class="title" id="welcome">
          Welcome!
        </h1>
        <p class="subtitle">
          Thanks for logging in.
        </p>
        <img src="" id="profilePic">
        <div class="field">
          <label class="label">Home town</label>
          <div class="control">
            <input name="hometown" class="input" type="text">
          </div>
        </div>
        <div class="field">
          <label class="label">Who would love to share a meal with?</label>
          <div class="control">
            <input name="dinnerguest" class="input" type="text">
          </div>
          <p class="help">Of anyone who has ever lived</p>
        </div>
        <div class="field">
          <label class="label">
            If you could pick what year to be born (past, present, future),
            when would you choose?
          </label>
          <div class="control">
            <input name="yeartobeborn" class="input" type="text">
          </div>
          <p class="help">Old fashioned or future tech?</p>
        </div>
        <div class="field">
          <label class="label">Life story</label>
          <div class="control">
            <textarea name="bio" class="textarea" placeholder=""></textarea>
          </div>
          <p class="help">It's a bit deep isn't it?</p>
        </div>
        <div class="control">
          <button class="button is-primary" onclick="update();">Update</button>
        </div>
        <div class="control">
          <button class="button" id="logout" onclick="logout();">Logout</button>
        </div>
      </div>
    </section>
  </body>
```

```
<script src="https://www.gstatic.com/firebasejs/7.14.2/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.14.2/firebase-auth.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.14.2/firebase-firestore.js"></script>
<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyBmZVNtAW4AtErmHwJrAOSrL6UA858vwMc",
    authDomain: "fir-for-the-video.firebaseio.com",
    databaseURL: "https://fir-for-the-video.firebaseio.com",
    projectId: "fir-for-the-video",
    storageBucket: "fir-for-the-video.appspot.com",
    messagingSenderId: "764304245956",
    appId: "1:764304245956:web:76d41052728870e42d7591"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
<script src="page2.js"></script>
</html>
```

page2.js

```
"use strict";    // Mandate variables to be declared

function loggedIn() {
    /* Get the user profile of our logged in user */
    let user = firebase.auth().currentUser;
    /* We are at page 2, let's display information about our user */
    document.querySelector("#welcome").innerHTML = "Welcome " + user.displayName;
    let img = document.querySelector("#profilePic");
    img.src = user.photoURL;
    // Create our database (Firebase Cloud Firestore) object
    let db = firebase.firestore();
    // Update the database using collection="profile" and documentid=user.uid
    db.collection("profile").doc(user.uid).get().then(function(doc) {
        if (doc.exists) {
            let record = doc.data();
            console.log("loggedIn read document " + user.email + " as:", record);
            document.querySelector("input[name='hometown']").value = record['hometown'];
            document.querySelector("input[name='dinnerguest']").value = record['dinnerguest'];
            document.querySelector("input[name='yeartobeborn']").value = record['yeartobeborn'];
            document.querySelector("textarea[name='bio']").value = record['bio'];
        }
    }).catch(function(error) {
        console.log("loggedIn error reading document " + user.email + " as:", error);
    });
}

function update() {
    // Verify we are logged in
    let user = firebase.auth().currentUser;
    if (user) {
        // Create our JSON object to store
        let record = {};
        record['name'] = user.displayName;
        record['email'] = user.email;
        record['hometown'] = document.querySelector("input[name='hometown']").value;
        record['dinnerguest'] = document.querySelector("input[name='dinnerguest']").value;
        record['yeartobeborn'] = document.querySelector("input[name='yeartobeborn']").value;
        record['bio'] = document.querySelector("textarea[name='bio']").value;
        // Create our database object
        let db = firebase.firestore();
        // Update the database using collection="profile" and documentid=user.uid
        db.collection("profile").doc(user.uid).set(record).then(function() {
            console.log("update successful for " + user.email);
        }).catch(function(error) {
            console.log("update error for " + user.email, error);
        });
    }
}

function main() {
    console.log("index.js/main running...");
    firebase.auth().onAuthStateChanged(function(user) {
        console.log("onAuthStateChanged", user);
        if (user) { /* If we are logged in */
            /* Let the loggedIn() function do the work */
            loggedIn();
        } else { /* If we are not logged in */
            /* Go to the index page if we aren't there already */
            if (document.location.pathname !== "/" ) {
                document.location = "/";
            }
        }
    });
}

/* When the web browser has finished loading everything, run our main() function */
window.onload=main;
```