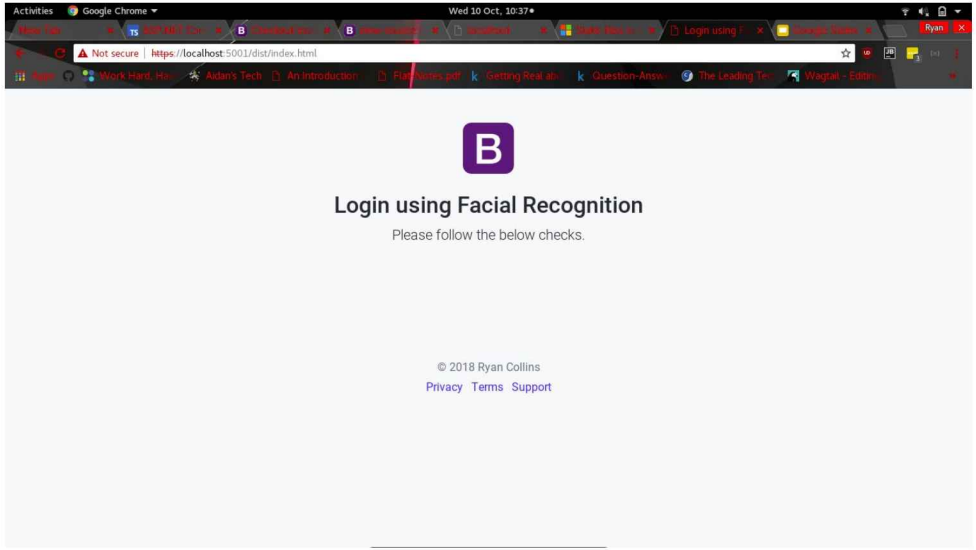
Facial Recognition Auth

For the web

Step 1: Login Portal

- Web application catches a video of user's face.
- Depending on liveness test, might also provide extra data to the user through the UI (e.g. what word to speak/where to look on the screen)
- Video captured, metadata used and hashed.
- All content sent to Step 2



Step 2: Main Service

- Data serviced by our API.
- Data sent to all of the microservices configured in the config file.

In future, confirmation will be using SAML2, so this would allow us to use our system as a login system for other third party systems (as a SSO).

Step 3: User ID

- While we have a face, we don’t have who they are.
- This check uses facial features to generate metrics based on a person.
- Using this, we return the user ID based on the metrics
- Other systems will use different methods of identifying people (e.g. comparing against a reference image).
- Once complete, these all return to part of the main service.

Step 4: Liveness Tests

- Using several different liveness tests, we check that the video is actually fine and hasn't been tampered with, and that the person is alive.
- For tamper detection, we check that the video (including metadata), and the hash match up.
    - We also check the timestamps, to confirm they are all equal.
- For liveness tests, we are expecting to try two methods:
    - Pupil Detection: this has been done before, but the aim is to experiment with tracking eyes based on Computer Vision
    - Speech/Lip reading: we give the user a word/set of words to say. These words are then matched up using lip reading algorithms.
    - Further methods could be added in the future.

Step 5: Back to Main - Confirmation

- After output from all the external services have been confirmed, a definite response is returned to the user, and potential third parties (in the case of SSO auth). The user should see this screen: