1. **技術主題**

**1.Manage Users in Firebase**

# Create a user

You create a new user in your Firebase project by calling the [createUserWithEmailAndPassword](https://firebase.google.com/docs/auth/web/password-auth" \l "create_a_password-based_account) method or by signing in a user for the first time using a federated identity provider, such as [Google Sign-In](https://firebase.google.com/docs/auth/web/google-signin) or [Facebook Login](https://firebase.google.com/docs/auth/web/facebook-login).

You can also create new password-authenticated users from the Authentication section of the [Firebase console](https://console.firebase.google.com/), on the Users page, or by using the [Admin SDK](https://firebase.google.com/docs/auth/admin/manage-users#create_a_user).

## Send a user a verification email

You can send an address verification email to a user with the sendEmailVerification method. For example:

var user = firebase.auth().currentUser;  
  
user.sendEmailVerification().then(function() {  
  // Email sent.  
}).catch(function(error) {  
  // An error happened.  
});

You can customize the email template that is used in Authentication section of the [Firebase console](https://console.firebase.google.com/), on the Email Templates page. See [Email Templates](https://support.google.com/firebase/answer/7000714) in Firebase Help Center.

It is also possible to pass state via a [continue URL](https://firebase.google.com/docs/auth/web/passing-state-in-email-actions) to redirect back to the app when sending a verification email.

Additionally you can localize the verification email by updating the language code on the Auth instance before sending the email. For example:

firebase.auth().languageCode = 'fr';  
// To apply the default browser preference instead of explicitly setting it.  
// firebase.auth().useDeviceLanguage();

**2.Firebase UI**

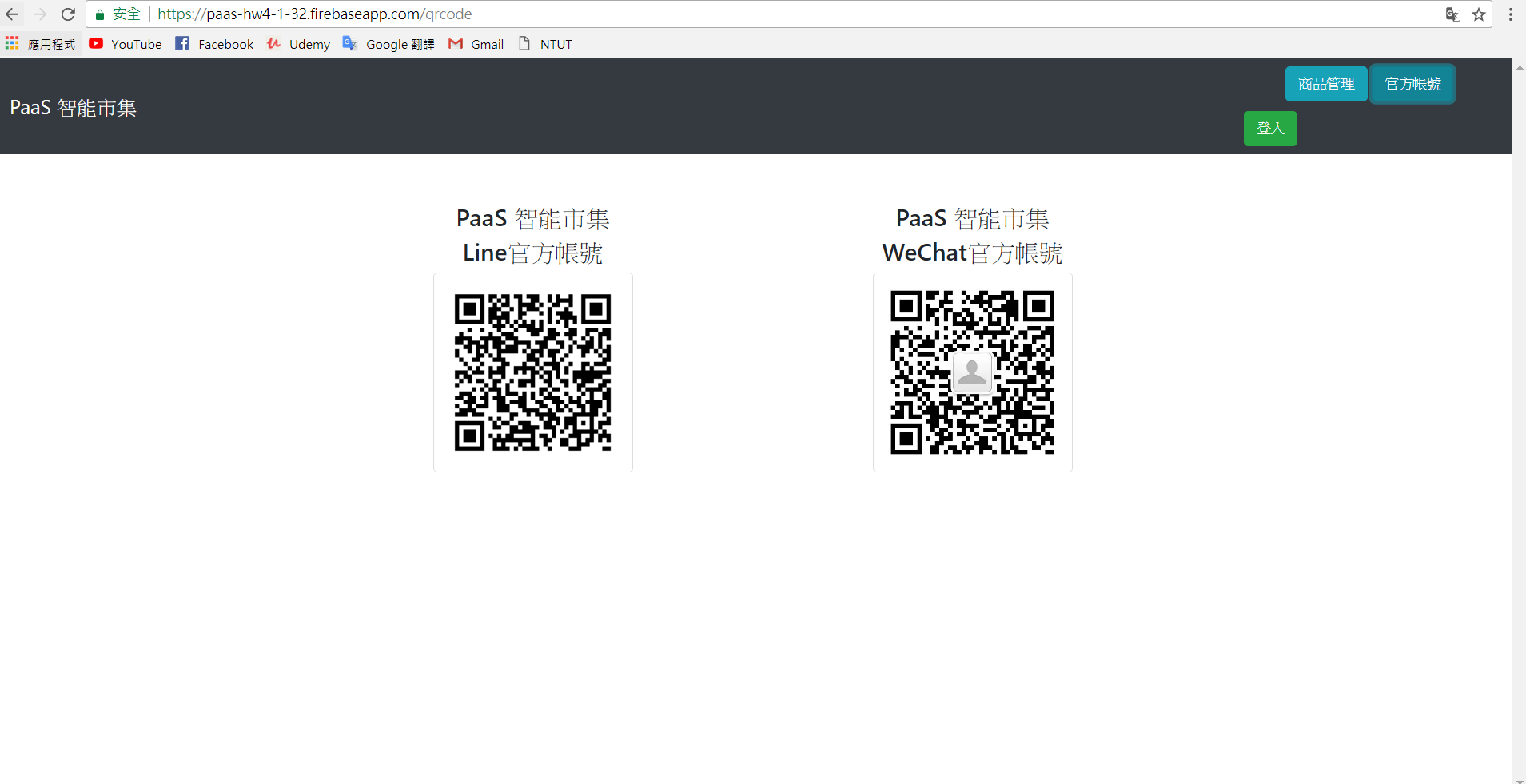
##### A UI library for Firebase, including binding for the realtime database, authentication and storage.

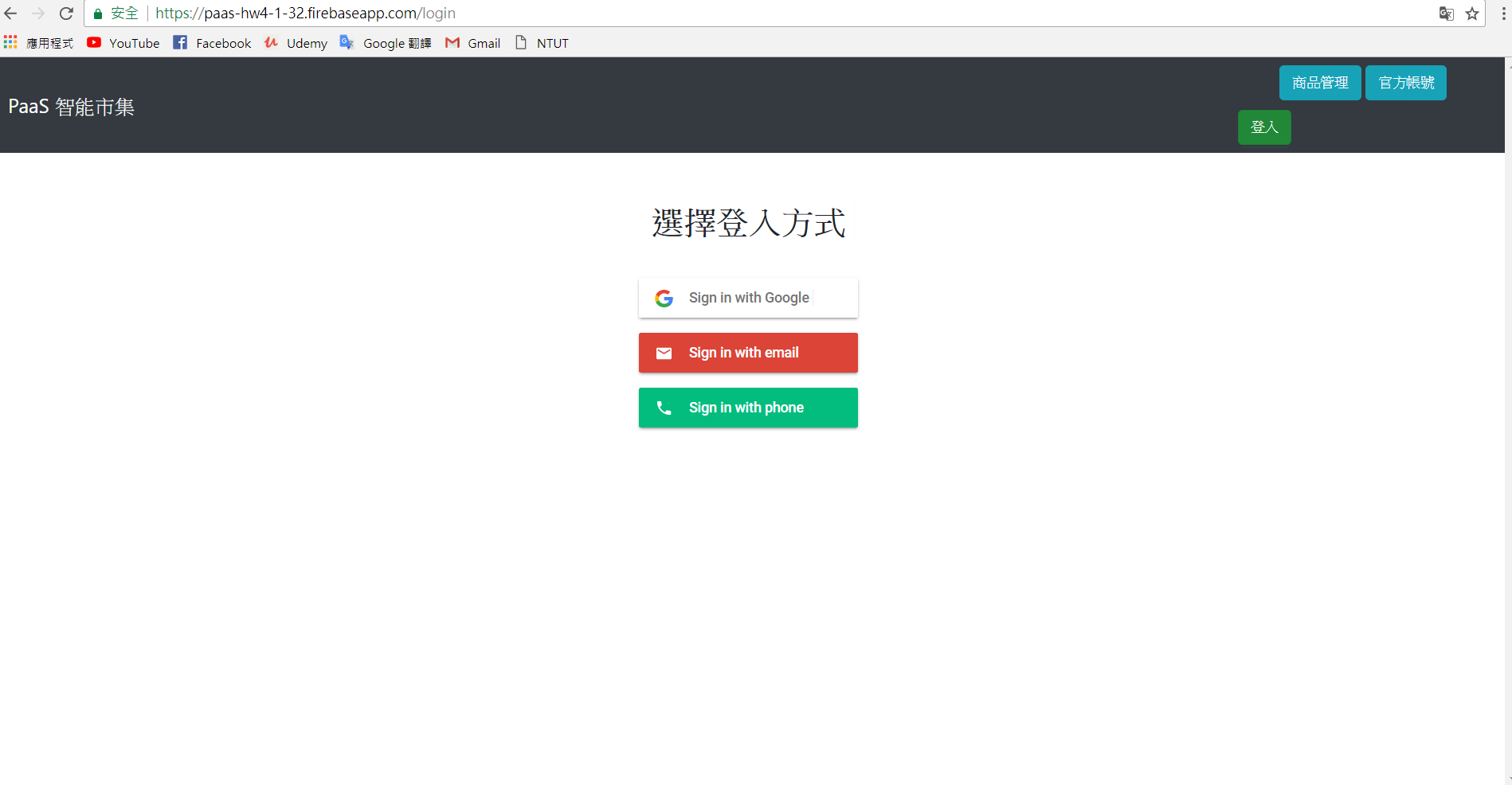
FirebaseUI is a set of open-source libraries for [Firebase](https://firebase.google.com/) that allow you to quickly connect common UI elements to the Firebase database for data storage, allowing views to be updated in realtime as they change, and providing simple interfaces for common tasks like displaying lists or collections of items.

Additionally, FirebaseUI simplifies Firebase authentication by providing easy to use auth methods that integrate with common identity providers like Facebook, Twitter, and Google as well as allowing developers to use a built in UI for ease of development.

1. **作業輸入、輸出畫面**

4-1:



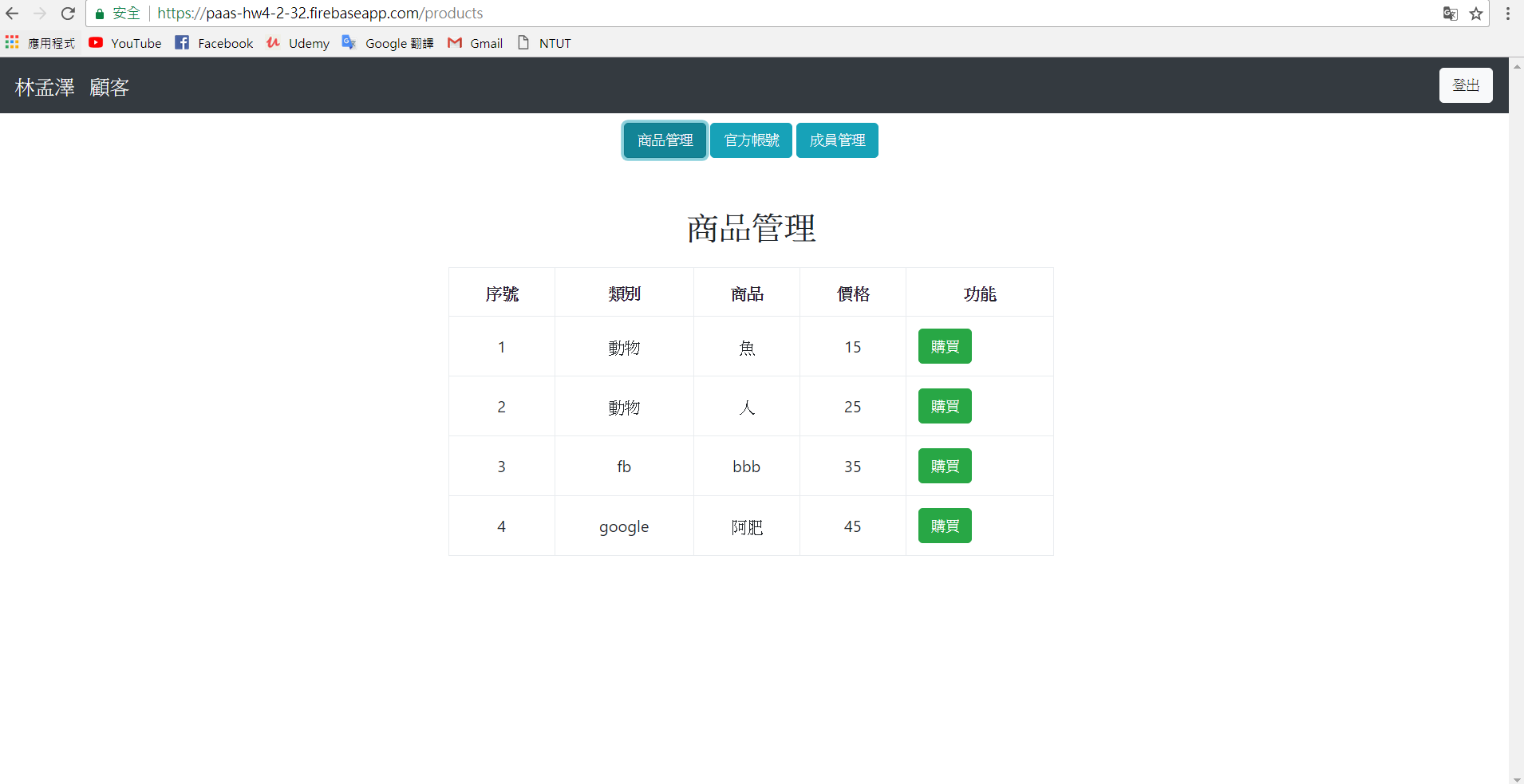
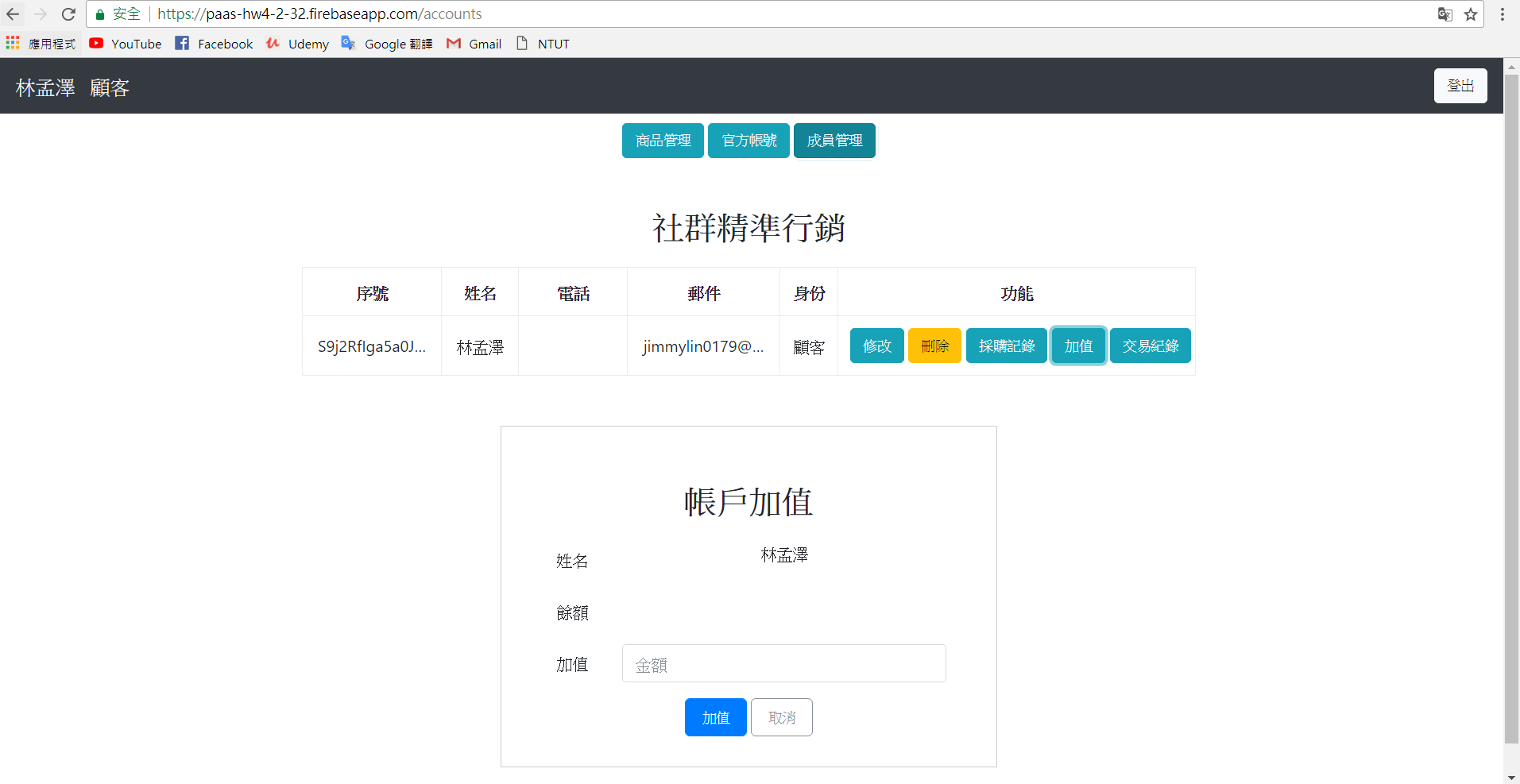






4-2





1. **參考資料**
2. **Auth User Profile**

<https://firebase.google.com/docs/auth/web/manage-users>

1. **Firebase UI**

<https://opensource.google.com/projects/firebaseui>

**3.Intorducing Firebase Authentication 教學影片**

<https://youth.be/8sGY55yxicA>

1. **學習心得**

在這一次的作業當中，老師可以說是首次留下比較多的code給我們自己去發揮，跟以往的作業比較不同，所以老師也直接說這次的作業，就會有許多同學在此陣亡，腦海裡也不斷地浮現老師講的那張路線圖，非常希望自己能夠走到最後面。

這次的作業當中，在4-1當中，主要是把三個button移到上面，再去設定一下，ngif的東西，就大致上完成了，不過也花了我們三個大概一個晚上多一點的時間去研究、互相討論。

在4-2的作業當中，由於本周末又去台中玩耍，但經過了漫長的討論、不斷的嘗試code當中，只有做出了半成品，也就是只有做出了表皮、還有一點點的功能，雖然沒有全部做完，但也對js認識了許多，感覺還有很多需要學習的概念。