



Firebase Authentication Providers

Weight: 15%;

Due Date: Sunday 5th, December 2021

Submit Before: 11:59pm

Introduction

During class we added the GoogleAuthProvider and the GithubAuthProvider. To complete the lab task add both the providers to your todo app. Note that both these providers will also sign up a user as long as an email address doesn't exist in the user section of the firebase console that matches the Provider email.

Application Pages

Your app currently has just 2pages:

1. **index.js**

The index page currently has the sign in/ sign up with google and github providers and the login with email and password components.

2. **todo.js**

page that renders on successful login or return error.

Task

Index.js Google Provider

Add Google provider to the index page as shown in class. Button should create a new account for a user signing in for the first time with a Google account. A returning user should be automatically signed in to the todo app, then routed to the todo app page.

Task

Index.js Github Provider

In this task you have to add a Github provider to your todo app on the current index page. In class you were shown how to add the Github provider in the firebase authentication panel.

Step1:

Make sure you have a copy of the vercel URL where your todo app is deployed. If you don't have a URL then deploy your site now.

Step 2:


Locate the sign in providers in the authentication section of the firebase console. Add Github from the provider list and copy the callback URL displayed at the bottom of the panel.

ovides you.

Sign-in providers

Add new provider

Provider	Status
Email/Password	Enabled
Google	Enabled


GitHub

Enable

Client ID

Client secret

To complete set up, add this authorization callback URL to your GitHub app configuration. [Learn more](#)

https://dmit-2008-javascript.firebaseio.com/_/auth/handler

Delete provider

Cancel

Save

Task 3



Step 3:

Navigate to your GitHub Developer Settings. <https://github.com/settings/developers>

You should see a button prompting you to create a new OAuth app. Click on the button and create a new app.

GitHub Apps
OAuth Apps
Personal access tokens

No OAuth applications

OAuth applications are used to access the GitHub API. [Read the docs](#) to find out more.

Register a new application

Step 4

Fill out all the information in the form that is provided to you. Make sure to add your deployment url as the homepage pageURL. The callback URL is the link provided by to you in the firebase console when setting up the Github provider. Finally click on the register application button.

Register a new OAuth application

Application name *

Something users will recognize and trust.

Homepage URL *

The full URL to your application homepage.

Application description

Application description is optional

This is displayed to all users of your application.

Authorization callback URL *

Your application's callback URL. Read our [OAuth documentation](#) for more information.

Register application

Cancel



Step 5

You will be provided with a client id, you need to copy this into the firebase console in the tab that says client id. Click the button to generate a new client secret. Copy the secret to the firebase console.

Client ID

[Redacted Client ID]

Client secrets

[Generate a new client secret](#)

Make sure to copy your new client secret now. You won't be able to see it again.



Client secret



Added 1 minute ago by **dominait**

Never used



[Delete](#)

You cannot delete the only client secret. Generate a new client secret first.

Application logo



Drag & drop

[Upload new logo](#)

You can also drag and drop a picture from your computer.

Application name *


to do list

Something users will recognize and trust.



Step 6


Once you copied the client id and secret back into the firebase console click on the client id input. You may have to wait a few seconds before the save button highlights. You are now ready to use Github as an authentication provider.


 GitHub Enable

Client ID

Client secret

To complete set up, add this authorization callback URL to your GitHub app configuration. [Learn more](#)



 Delete provider Cancel Save

Task

Index.js Github Provider

Add the Github provider button to the index page. In class you were shown how to add the required functionality to have a user sign in with their github account. Note that the button will do both sign up and sign in. Firebase checks to see if there is an email matching the one used by Github. Once the user has signed in to your todo app, use the NEXTjs router to switch a valid user to the todo app page.

Submission

Vercel Deployment URL

Github Repo URL



Marking Key

Task	Mark	Task Value
Google Provider Success To get all 3 marks a user must be able to sign in to your app using a google account.		3
Github Provider Setup Success To get all 3 marks a user must be able to sign in to your app using a github account.		3
GitHub Repository & Deployment Success Vercel deployment Github repository		3
		9



Marking Rubric

Marks	5 Marks Criteria
5 [0]	Task was completed with the highest proficiency, adhering to best practices. The tasks demonstrate a high level of proficiency implementing the subject matter and the presentation of the content is to a professional standard.
4 [-1]	Task was completed well, with some minor mistakes. Well above average work, shows good understanding of the implementation details of subject matter. Tasks show a high degree of competence.
3 [-2]	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
2 [-3]	Task completion is below average, the task was poorly completed. Shows understanding of the task and the requirements to implement, but implementation was poorly executed.
1 [-4]	Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution.
0 [-5]	Not completed.

Marks	3 Marks Criteria [minus]
3 [0]	Task was completed well, adhering to best practices, and shows a high degree of proficiency with the subject matter guidelines.
2 [-1]	Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement.
1 [-2]	Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution.
0 [-3]	Shows a little to no degree of competence in completing the task; not completed.

Marks	1 Marks Criteria
1	Task completed satisfactorily
0	Task was not completed