**GitHub and Xcode Integration**

Step I – Adding GitHub account to Xcode

1. Open Xcode.

**Text

Description automatically generated with low confidence**

1. At the top-left corner, select Xcode > Preferences.

**A picture containing text

Description automatically generated**

1. Click on the Xcode preferences and it will open a window. Now, click on the Accounts tab.

**Graphical user interface, application

Description automatically generated**

1. At the bottom-left corner in the Accounts tab, you could see a + icon. Click on it to see a list of account options.

**Graphical user interface, application

Description automatically generated**

1. Scroll down the list of account options and select GitHub. Click Continue.

**Graphical user interface, application, chat or text message

Description automatically generated**

1. A new window pops up. In this window, provide your GitHub username in the Account field, and the token you generated online on your GitHub account in the Token field, respectively. Click on Sign In.

**Graphical user interface, text, application

Description automatically generated**

Step II – GitHub token generation

To generate a GitHub token, follow the below steps:

1. Open your GitHub account online in any web browser. At the top-right corner, click on your profile icon and select Settings in the dropdown options.

**Graphical user interface, text, application

Description automatically generated**

1. On the left-hand side, scroll down and select Developer settings.

**Graphical user interface, text, application, email

Description automatically generated**

1. Click on Personal access tokens option.

**Graphical user interface, text, application, email

Description automatically generated**

1. Click the Generate new token button on the top-right corner. It redirects to a new webpage that is like the below shown figure.

**Graphical user interface, text, application, email

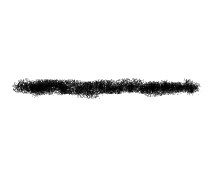
Description automatically generated**

1. Give an appropriate name in the Note field and pick the Expiration date of your choice.
2. Select the following scopes:
   1. repo
   2. admin:public\_key
   3. user
   4. write:discussion
3. Finally, click Generate token at the bottom to generate your token.

**Graphical user interface, application

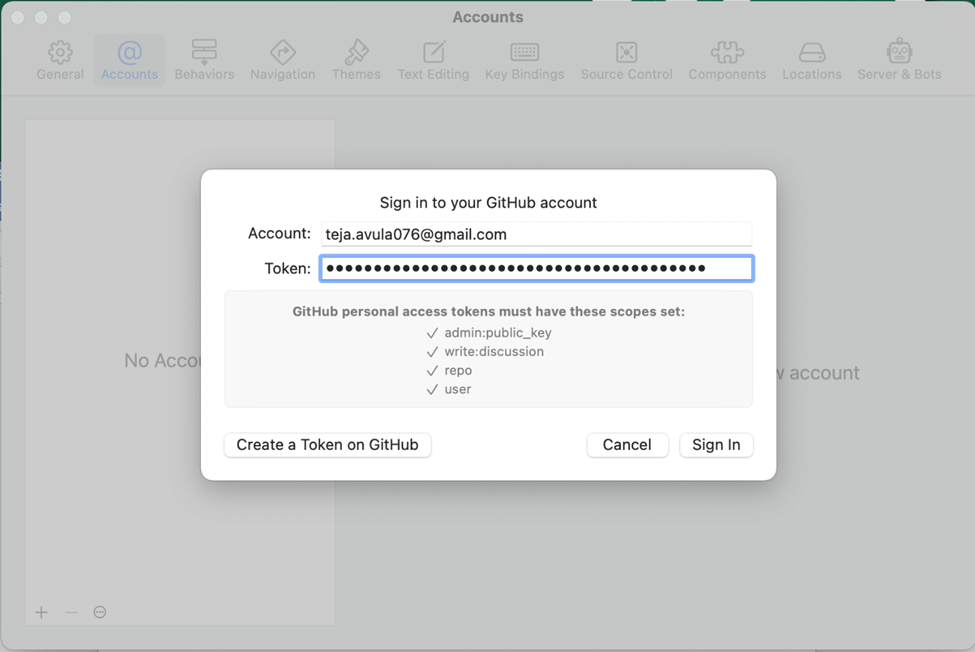
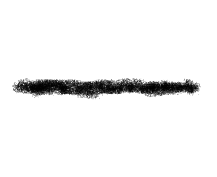
Description automatically generated**

1. Copy the generated new token. Make sure you save this token somewhere else. DO NOT LOSE IT.

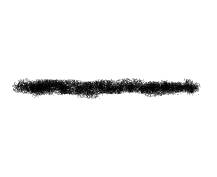
**Graphical user interface, text, application, email

Description automatically generated**

1. Now, go back to Xcode > Preferences > Accounts and give your account name (i.e., GitHub username) and the token value you generated. Click Sign In.

****

1. You will see that your account is now added into the Accounts list.

**Graphical user interface, application

Description automatically generated**

Step III – Initializing a local git repo

1. Create a new folder in the Documents and give it a name of your choice (avoid spaces between words). In the below screenshot, “IOS” is given as folder name.

**Graphical user interface, application

Description automatically generated**

1. Open Terminal by pressing command + space bar and enter search word as terminal.

**Graphical user interface, text, website

Description automatically generated**

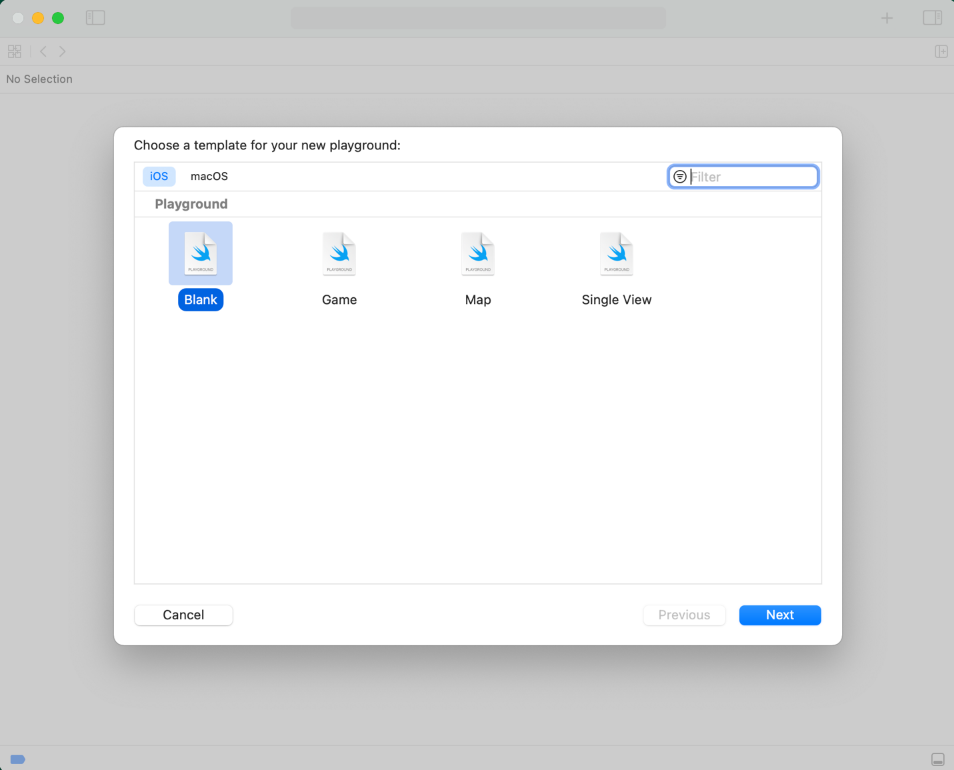
1. In the terminal, use command cd Documents to navigate to Documents directory.
2. Type command git init to initialize a local git repo.
3. Exit the terminal.

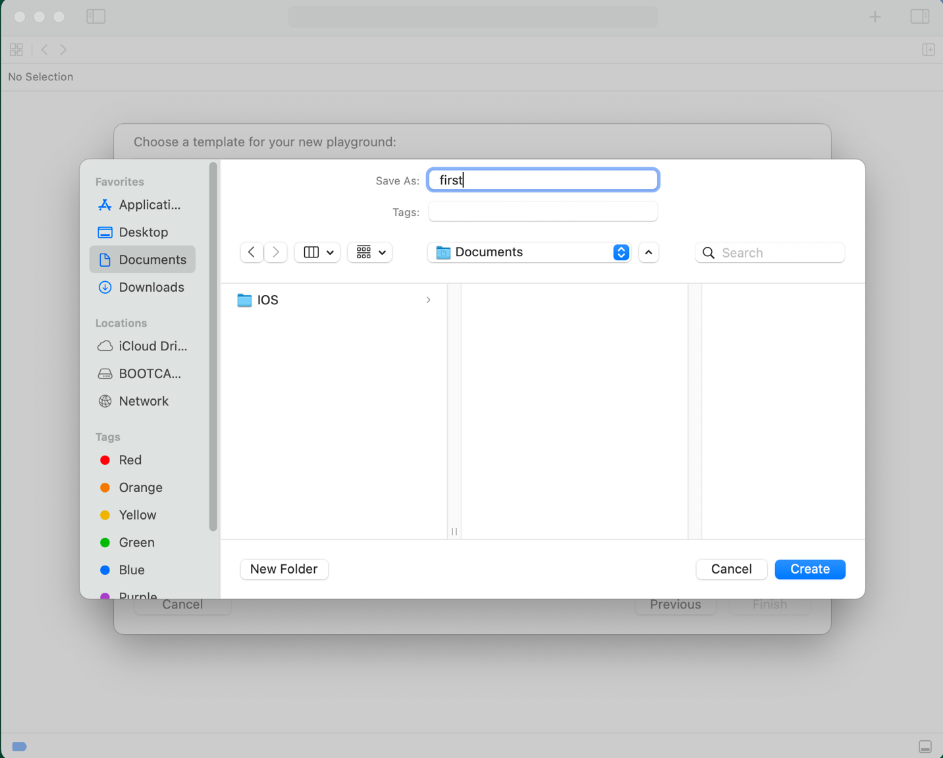
Step IV – Creating a new repository

1. Now create a new playground file on Xcode by going to File > New > Playground.

**Graphical user interface

Description automatically generated**

1. Leave platform to iOS and select Blank playground file and click Next.****
2. Give it an appropriate name and save it into the folder that you have created in the Documents directory by clicking Create.

****

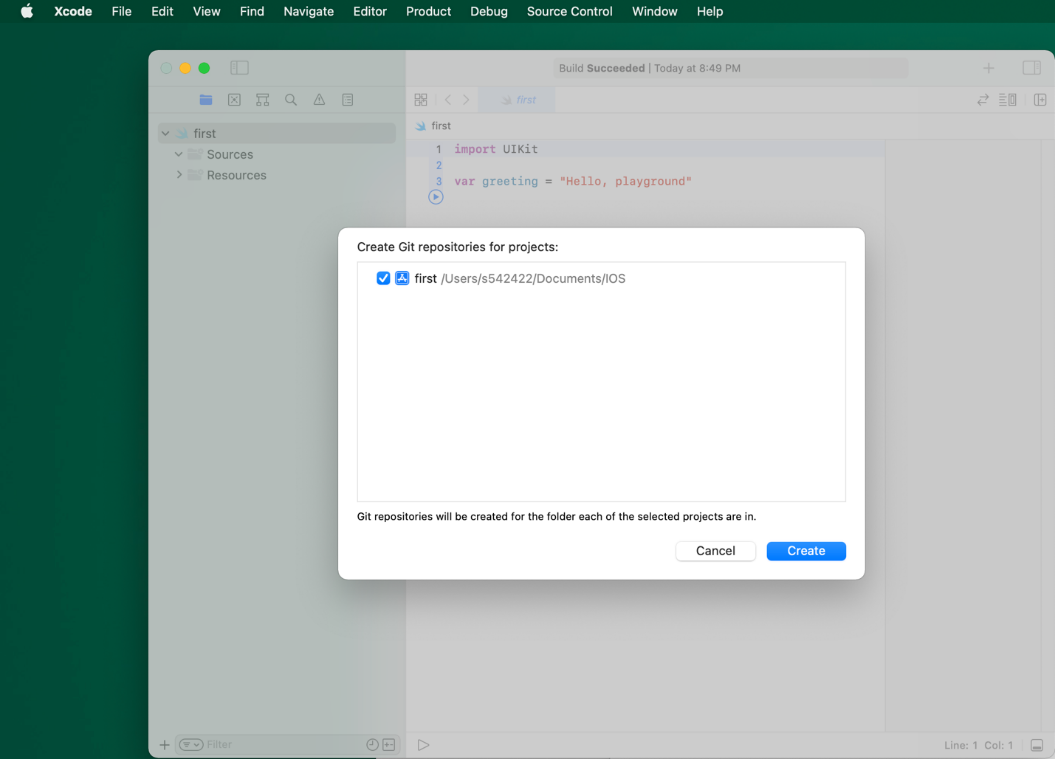
1. You will see the.playground file opens in the Xcode like below.**Graphical user interface, application, Word

   Description automatically generated**
2. Now, click on the Source Control > New Git Repositories.

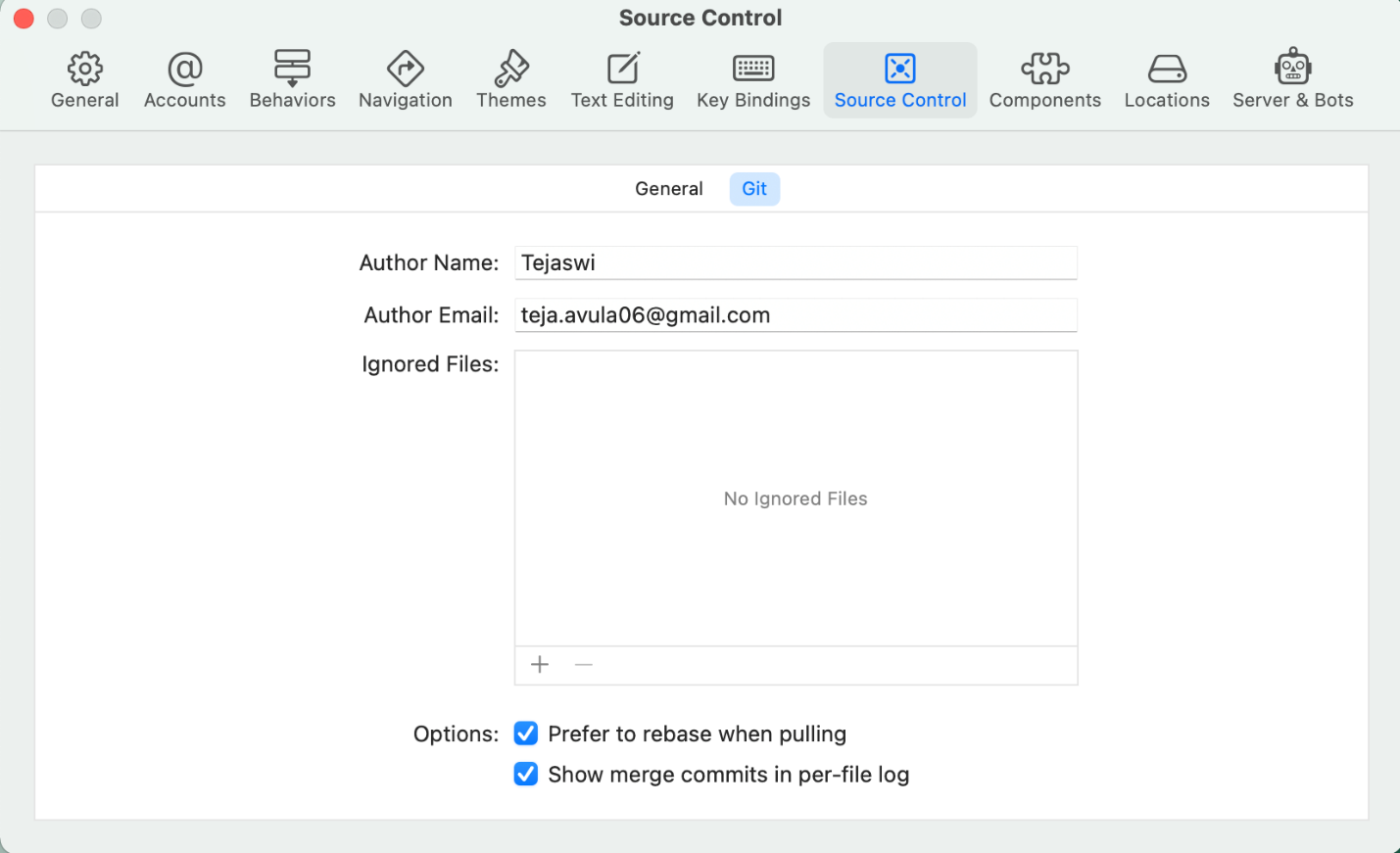
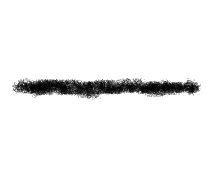
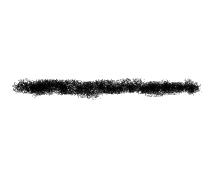
**Graphical user interface, application

Description automatically generated**

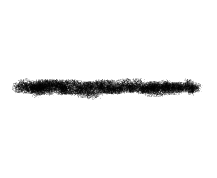
1. A new window pops up like the one shown below.

****

1. Click Create button.
2. If you find any issues,
3. Click on the fix button. It will redirect to Xcode > Preferences > Source Control tab.
4. Where you can select the Git option and enter your full name and email address that is associated to your GitHub account.
5. At last, you can go back and click Create button in the playground.

****

1. Now, you find a new repo that has been created.

**Graphical user interface, application, Word

Description automatically generated**

1. At the bottom, in the playground file, click on the highlighted icon to create a New “IOS” Remote.

**Graphical user interface, application

Description automatically generated**

1. Now, create your repo by clicking on create button.

**Graphical user interface, text, application

Description automatically generated**

1. You find a new repo that has been created in your online GitHub account.

**Graphical user interface, text, application

Description automatically generated**

Graphical user interface, text, application, email

Description automatically generated

Step V – Commit and Push

1. Go back to the playground file you created on Xcode. Make some changes to it.

Graphical user interface, application, Word

Description automatically generated

1. Commit your changes by clicking Source Control > Commit.

Graphical user interface, application, Word

Description automatically generated

1. Select the playground file you want to commit and give a comment in the message box. Make sure you checked the Push to remote option and click on Commit and Push.

Graphical user interface, text, application

Description automatically generated

1. Now, you can see the updated file in your online GitHub account.

Graphical user interface, text, application, email

Description automatically generated

1. In the same way, repeat Steps IV and V for the Xcode projects as well.

**Screen Resolution:**

[**https://www.ios-resolution.com/**](https://www.ios-resolution.com/)

**Public APIs:**

[**https://github.com/public-apis/public-apis#entertainment**](https://github.com/public-apis/public-apis#entertainment)

**SDWebImage framework/library:**

[**https://github.com/SDWebImage/SDWebImage**](https://github.com/SDWebImage/SDWebImage)