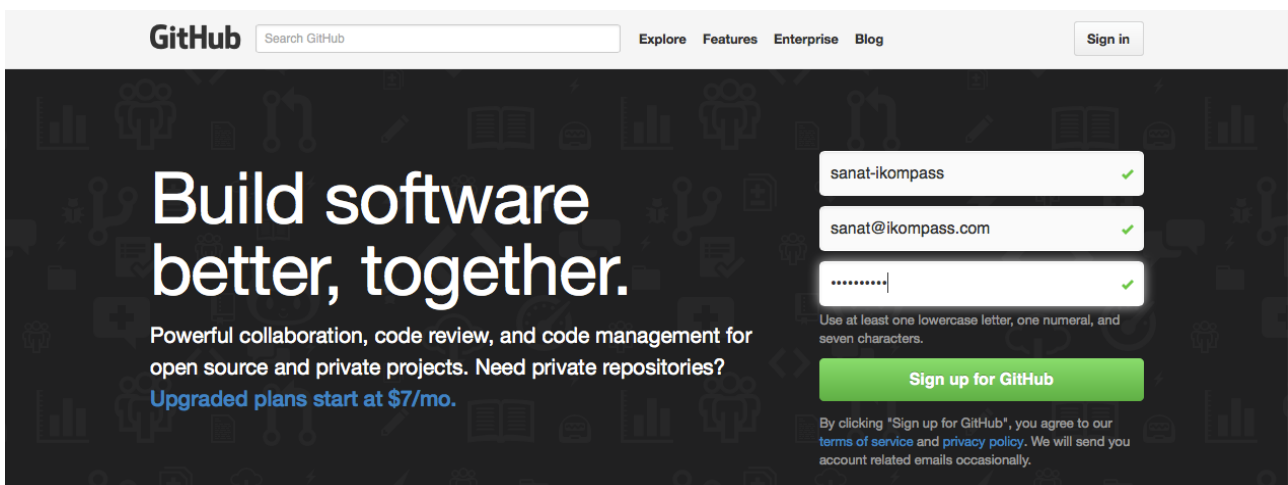


# ANNEXTURE - 1

**Disclaimer :-** The screenshots provided in this tutorial may not be the same when you access the actual webpage as they are frequently updated.

## Sign up for Free GitHub account

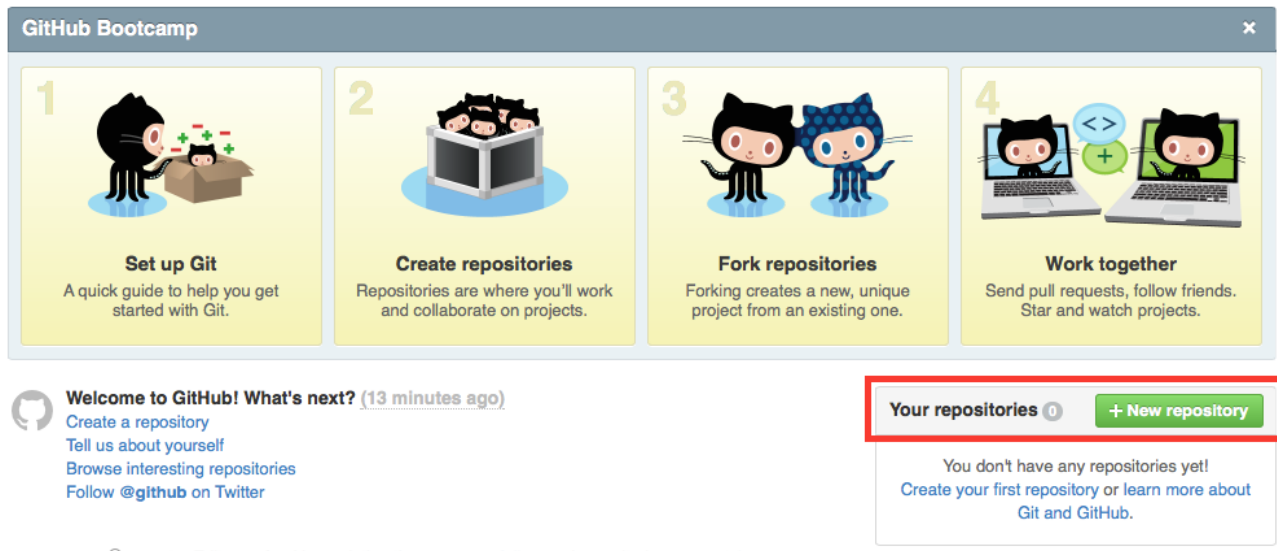
1. Log on to <https://github.com> and enter your details and click on “Sign up for GitHub” button, to create a GitHub account.

A screenshot of the GitHub website's sign-up page. The header features the GitHub logo, a search bar, and navigation links for Explore, Features, Enterprise, and Blog, along with a Sign in button. The main content area has a dark background with the text "Build software better, together." and a description of GitHub's capabilities. On the right, there is a sign-up form with three input fields: a username "sanat-ikompas" with a green checkmark, an email "sanat@ikompas.com" with a green checkmark, and a password field with a green checkmark. Below the password field is a note: "Use at least one lowercase letter, one numeral, and seven characters." A green "Sign up for GitHub" button is positioned below the form. At the bottom right, a small disclaimer states: "By clicking 'Sign up for GitHub', you agree to our terms of service and privacy policy. We will send you account related emails occasionally."

2. On the next page, you will be asked to choose a plan. By default, a free plan is chosen for you. Click “Finish sign up” button and you are done.

# Creating your repository

1. A repository is where your files will be placed. On the GitHub account's welcome page, click "New Repository" button.




2. On the "Create Repository" page, enter the following details:

- A. Name of the repository
- B. Description for the repository
- C. Make the repository "Public" by choosing that option. If you want to make it "Private", you may need to upgrade your account by paying some money.
- D. Check the option - Initialize the repository with a README

Owner

Repository name



sanat-ikompass

 / 


ios

Great repository names are short and memorable. Need inspiration? How about [massive-octo-cyril](#).


**Description** (optional)

Repository to store ios projects

---

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---

☒ **Initialize this repository with a README**


This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

---

Create repository

Once done, click the Create repository button. If everything goes well, you should be seeing your repository screen as follows:

 [sanat-ikompass](#) / [ios](#)

Unwatch 1

Star 0

Fork 0

---


Repository to store ios projects — Edit

1 commit


1 branch

0 releases

1 contributor

 branch: master [ios](#) / +

Initial commit

 sanat-ikompass authored just now

latest commit ef011d7ddb

README.md

Initial commit

just now

README.md

# ios

Repository to store ios projects

<> Code

Issues 0

Pull Requests 0

Wiki

Pulse

Graphs

Settings

HTTPS clone URL

<https://github.com>

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Note:- You must verify your email address before you can start using GitHub account.

### Generating SSH keys for your GitHub account

The GitHub account must trust your computer before it can allow files from it to be added to the repository. SSH keys are a way to identify trusted computers, without involving passwords. Follow the steps below to generate a SSH key:

1. On a Unix based system (like Mac OS), the ssh keys are usually stored in “.ssh” folder, within your home directory and path for it looks like: ~/.ssh
2. We can use ssh-keygen tool to generate the ssh key as follows:
  - i. Open terminal and type the following command:

```
ssh-keygen -t rsa -C "sanat@ikompass.com"
```

Forget not to use your own email id in the above command.

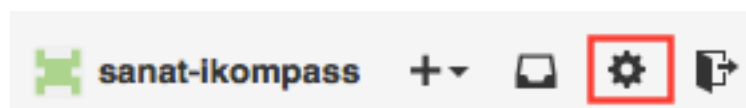
- ii. If asked to choose a location, just hit Enter. We want default settings for this step.
  - iii. After this, you will be asked to provide a passphrase. Please use a strong passphrase at this step and don't leave it blank.

Confirm the same passphrase in the next step and hit Enter. An RSA key will be created for you.

```
sanat — bash — 80x24
sanats-MacBook-Pro:~ sanat$ ssh-keygen -t rsa -C "sanat@ikompass.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/sanat/.ssh/id_rsa):
Created directory '/Users/sanat/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/sanat/.ssh/id_rsa.
Your public key has been saved in /Users/sanat/.ssh/id_rsa.pub.
The key fingerprint is:
bb:aa:84:62:7d:6a:04:35:8b:7c:12:96:0d:bf:f8:5a sanat@ikompass.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
| .+
| +.+
| o+.o
| =.o.
| .+. S
| oo .
| ..oE.. .
| ..ooo .
| .....
+-----+
```

3. Every ssh key has a public-private pair. The public key needs to be added to your GitHub account. To do that open the public key file using your text editor and copy the key from there. It is usually named : **id\_rsa.pub** and found in: **~/.ssh** directory

4. In your GitHub account, open the Settings page by clicking its icon (in the top-right corner):



5. In the Settings page, click **SSH keys** :

Personal settings
<b>Profile</b>
<a href="#">Account settings</a>
<a href="#">Emails</a>
<a href="#">Notification center</a>
<a href="#">Billing</a>
<a href="#">SSH keys</a>
<a href="#">Security</a>
<a href="#">Applications</a>
<a href="#">Repositories</a>
<a href="#">Organizations</a>

6. Click “Add SSH key” button

<b>SSH Keys</b>	<a href="#">Add SSH key</a>
There are no SSH keys with access to your account.	

7. In the Title field, add a descriptive label for the new key. In the Key field, paste your public key:

### Add an SSH Key

Title

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDGQfLO+aPX/bJVnwQBglEBveHFOANQiLpdiR9luID4u/RX85Bc9CuypQB
OW314LxEOX9WbehBREGDF3aM4njZvp+uVUSwUvWIWnNt8RHMle5sxSxYCViyx6rnUFcASFgZCulA+WGYYawzQq
hFBcdfGNKGNz03e8A7rH4YDSuHA3I20EYMTth9E8Q5RWsPZ4bJnhrKKgBcVEoNP+KcK9hIcbX25Fq2GN/ABxrTdlZQ
Cn/+GxJMloDgzrjEwCofxGm9gvoJ3TUIJd4bPYG0JNZEwxkOchH4Sd8qO7wEfb1//Wb6YgCLV2cVs+wwDx2Kqi68BK7
hizN8StTZp2xZjMer sanat@ikompass.com
```

8. Click Add key button to confirm your entry. On successful addition of the key, you should see the same in your account as follows:

### SSH Keys Add SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

**sanat's mac**  
bb:aa:84:62:7d:6a:04:35:8b:7c:12:96:0d:bf:f8:5a  
Added on Jan 18, 2015 — Never used

9. Validate the settings by running the following command in your terminal: `ssh -T git@github.com`

You may be prompted with the following message:

```
The authenticity of host 'github.com (192.30.252.130)' can't be
established.
RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)?
```

Hit Enter and wait for the confirmation to come from the github server. If you see a message like below, you are all set:

```
Warning: Permanently added 'github.com,192.30.252.130' (RSA) to the list
of known hosts.
Identity added: /Users/sanat/.ssh/id_rsa (/Users/sanat/.ssh/id_rsa)
Hi sanat-ikompas! You've successfully authenticated, but GitHub does
not provide shell access.
```

## Configuring Xcode to use GitHub account as File Repository

Xcode offers a GUI tool to manage the source control but unfortunately it treats even the Git repository as Subversion (another popular open source SCM tool). I prefer to use terminal for initial initiation of the repository. The steps and commands are very simple but before you execute them first close Xcode. Once your Xcode is closed, follow the commands below:

1. Browse to the project path where the .xcodeproj file is.
2. Execute the following from your command line:

```
git init
git add .
git commit -m 'Committing from local to remote repository'
git remote add origin git@github.com:sanat-ikompas/ios.git
git pull origin master
```

3. This will open the VI editor so that you can enter any comment explaining your commit. Add a comment if you wish and save/close the editor. If you are not aware of VI editor and its commands, then the following snippet may help you:

1. To enter in the add/insert mode hit :i
2. Once in edit mode, you can add your text
3. Once done exit the add/insert mode by hitting :Esc



4. To save your changes and exit the editor press :wq!

w = write

q = quit

! = force

4. Once the VI editor is gone, enter the following command:

```
git push origin master
```

That's all it takes. If you find git command's coming from alien land, worry not ! Git is very well documented and as stated earlier, you should read more about them from their online documentation which is available at : <https://help.github.com>

Finally, open project in XCode and you should be able to see Source Control is configured for GitHub use.

