

IERG 4210 - Tutorial 2

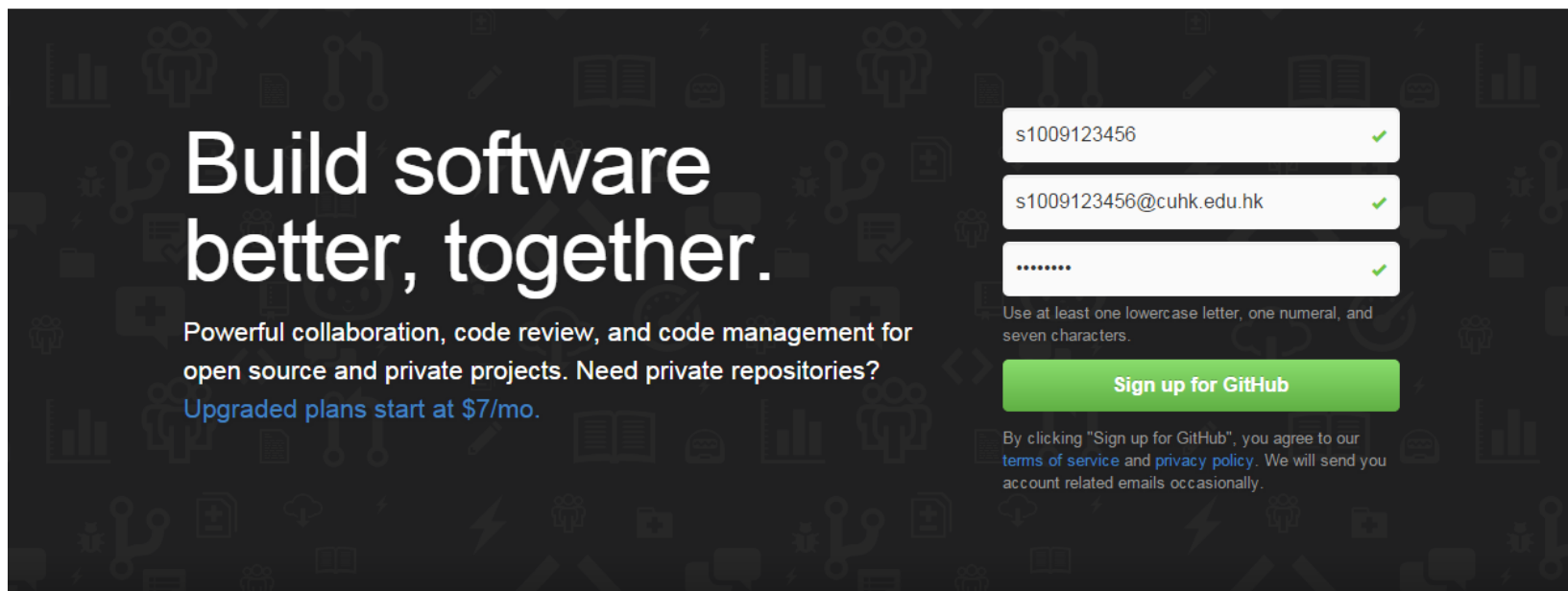
Benedict Mak

19 Jan 2015

GitHub - Registration

- Go to <https://github.com/>
- Use your SID and cuhk email as the username and email account
- (e.g. Username: s1009123456

Email: s1009123456@cuhk.edu.hk)


A screenshot of the GitHub registration page. The background is dark with a pattern of faint icons. On the left, the text "Build software better, together." is prominently displayed in white. Below it, a paragraph describes GitHub's features: "Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo." On the right side, there is a registration form with three input fields. The first field contains the username "s1009123456" and has a green checkmark. The second field contains the email "s1009123456@cuhk.edu.hk" and also has a green checkmark. The third field contains masked characters "....." and has a green checkmark. Below these fields, a green button labeled "Sign up for GitHub" is visible. At the bottom right, there is a small disclaimer: "By clicking 'Sign up for GitHub', you agree to our terms of service and privacy policy. We will send you account related emails occasionally."


GitHub - Registration


- Sign up! and Choose the Free Plan
- Don't check "Help me set up an organization next"

Welcome to GitHub

You've taken your first step into a larger world, @s1009635373.

 **Completed**
Set up a personal account

 **Step 2:**
Choose your plan

 **Step 3:**
Go to your dashboard

Choose your personal plan

Plan	Cost (view in HKD)	Private repos	
Large	\$50/month	50	<button>Choose</button>
Medium	\$22/month	20	<button>Choose</button>
Small	\$12/month	10	<button>Choose</button>
Micro	\$7/month	5	<button>Choose</button>
Free	\$0/month	0	<button>Chosen</button>

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an *estimate* based on *current* exchange rates. Local prices will change as the exchange rate fluctuates.

Don't worry, you can cancel or upgrade at any time.

- ☐ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
[Learn more about organizations.](#)

Finish sign up

Each plan includes:

Unlimited collaborators
Unlimited public repositories

- ✓ Free setup
- ✓ SSL Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

GitHub - Registration

- Check your CUHK email for the confirmation email.
- Fill in the “Github Account” survey in eLearn to let us know your account ID before Tuesday (20 Jan 2015) ends
- Check your email for invitation of ierg4210]
- You will see a shopXX for you (XX is a number)

Confirm Email Verification

Please confirm verification of **s1009635373@cuhk.edu.hk**.



s1009635373@cuhk.edu.hk

Confirm

Github setup

- Download and install Git CLI (<https://git-scm.com/downloads>)
- Follow these steps to setup git for your assignment
- Change shop00 and s1009123456 to your shop and SID.
- Open terminal / cmd (Command is in RED)
- Create a folder for your shopXX (`mkdir shop00`)
- Go to the folder shopXX (`cd shop00`)
- Initial git (`git init`)
- Configure username and email
(`git config user.name s1009123456`
`git config user.email s1009123456@cuhk.edu.hk`)

Github setup

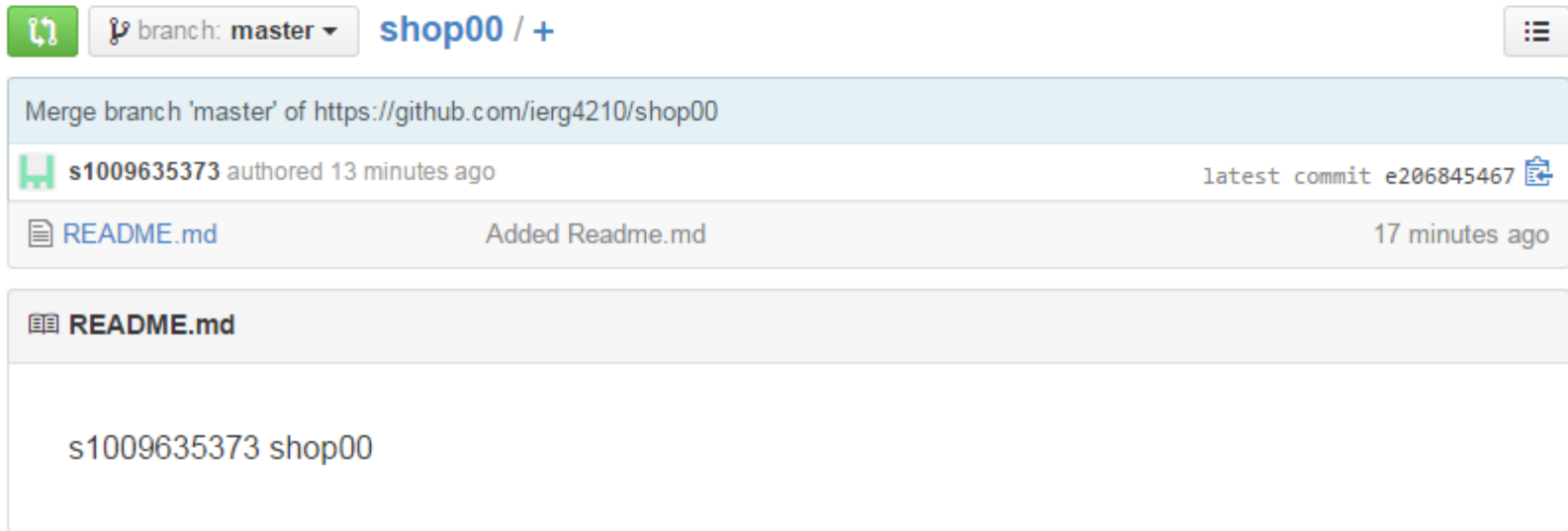
- Setup alias for remote repositories
(`git remote add origin https://github.com/ierg4210/shop00.git`
`git remote add shop-samples https://github.com/ierg4210/shop-samples.git`)

(Optional)

- Cache your credential
Windows - (`git config credential.helper wincred`)
Linux - (`git config credential.helper cache`)
- Confirm what you just set (`git config -l`)

First New File

- Create a README.md file with your SID and shop ID
Windows – notepad
Linux (`echo "s1009123456 shop 00" > README.md`)



The screenshot shows the GitHub interface for a repository named 'shop00'. At the top, there's a green 'fork' button, a dropdown menu showing 'branch: master', and the repository name 'shop00' with a plus icon. Below this is a light blue header bar indicating a merge of the 'master' branch from 'https://github.com/ierg4210/shop00'. The commit history shows a commit by user 's1009635373' made 13 minutes ago, with the latest commit hash 'e206845467'. A file named 'README.md' is listed as 'Added Readme.md' 17 minutes ago. The content of the README.md file is displayed below, showing the text 's1009635373 shop00'.

branch: master shop00 / +

Merge branch 'master' of https://github.com/ierg4210/shop00

s1009635373 authored 13 minutes ago latest commit e206845467

README.md Added Readme.md 17 minutes ago

README.md

s1009635373 shop00

Commit your first modification

- Add your change to your first commit (`git add -A`)
(`-A` = all changes)
- Commit your changes (`git commit -m "Added README.md"`)
(`-m "X"` = your comment)
- Push your changes to master branch (`git push -u origin master`)

Download Sample Code

- Switch to master branch (**git checkout master**)
 - See all the branches in shop00 (**git branch -a**)
(* represents which branch are you in)
- Download phase 1 sample code (**git pull shop-samples phase1**)
- Save your commit comment

- There may be some merge conflict
 - Message: CONFLICT (content): Merge conflict in index.html
 - The conflict file will be marked like this

<<<<<< HEAD

for phase1

Place your HTML here for the main page.

← Your file

=====

For phase1, Place your HTML here for the main page.

>>>>>> 40c0bb67006c8c4811b2ec066ac3672520d8065c

← Shop-samples
file

Download Sample Code

- Delete all the markers (e.g. <<<< =====) and decide how do you merge the changes
- Add all the changes and commit (`git add -A` and `git commit -m "message"`)
- Push to your shop (`git push -u origin master`)

Assignment Submission

- Keep all your updates in the master branch
- When you are going to submit your assignment
 - Open a new branch for every phases (`git checkout -b phase1`)
 - Push it to the remote (`git push -u origin phase1`)
- Update Phase1
 - Go to phase1 (`git checkout phase1`)
 - Merge the master branch with phase 1 (`git merge master`)
- Always check the status to see any new messages (`git status`)