Grand ICT 연구센터

재학생 포트폴리오 작성 및 관리

충북대학교 산업인공지능학과

2020, 10, xx



내용

- 포트폴리오에 대한 이해
- □ 포트폴리오의 구성
- ☐ GitHub 사용법
- GitHub를 이용한 포트폴리오 작성 및 관리

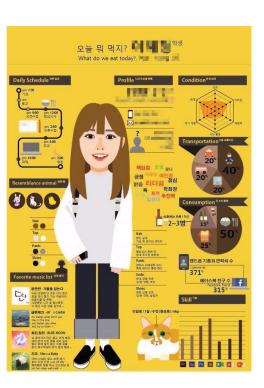
포트폴리오에 대한 이해

🔲 포트폴리오 ?

◆모든 직종에서 자신의 경력, 능력을 증명할 수 있는 파일이나 프로그램 등을 의미







포트폴리오의 구성

- 포트폴리오에 포함되는 내용
 - ◆ 자기 소개
 - ⊙ 간략하게 기술
 - ◆ 이력 및 학력
 - 인턴, 외주, 산학, 수상경력 등
 - ◆ 연락처
 - ◉ 이메일, 핸드폰, SNS, LinkedIn, GitHub
 - ◆ 프로젝트
 - 프로젝트 이름 및 기간
 - 큰 기술 위주의 적용 기술
 - 프로젝트 내용(되도록 자세히 서술)
 - 사진, 동영상, 웹사이트 형식의 프로젝트 데모

- ☐ Git
 - ◆ 버전을 관리해 주는 시스템으로서 파일의 변경 사항을 추적
 - 변경 사항을 기록하고, 특정 시점의 버전으로 언제든지 돌아갈 수 있도록 해줌
- ☐ GitHub
 - ♦ 웹사이트!
 - git으로 관리하는 자료를 다른 사람과 공유하거나 백업



- ☐ Git 기본구조
 - ◆ 개인컴퓨터 + internet 망 + 원격서버로 구성
 - 개인컴퓨터를 Working Directory, Staging Area, Local Repository로 구분



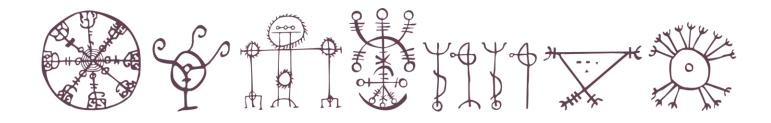
- Working Directory
 - 작업 영역으로서 파일의 변화를 감지
 - Git에 의한 관리를 받고자 하는 경우 Staging Area로 옮겨줌
- Staging Area
 - Git을 이용한 관리를 수행
 - 파일의 내용 추가, 삭제, 수정 등을 git이 알게됨



- Local Repository
 - ⊙ 저장소
 - commit 명령어에 의해 Staging Area에서 Local Repository로 이동
 - 파일 변경 시 **이전 생성 파일은 그대로 유지**되고 **그 위에 변경 파일 저장**
 - ▶ 2개의 별도 파일 생성
 - ▶ 사용자가 이전 파일과 변경된 파일의 다른 점을 쉽게 파악 가능
- Remote Repository
 - 원격 저장소로서 내 컴퓨터가 아닌 다른 장소에 저장
 - 컴퓨터 고장이나 데이터 유실 시 작업 내용을 원격으로 가져올 수 있음

- GitHub 사용
 - ◆ Git 설치
 - ◆ GitHub 가입
 - ◆ Git 초기 설정
 - ◆ GitHub 온라인 저장소 만들기
 - ◆로컬 파일을 온라인 저장소에 업로드





☐ Git 설치

◆http://git-scm.com



Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.





About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

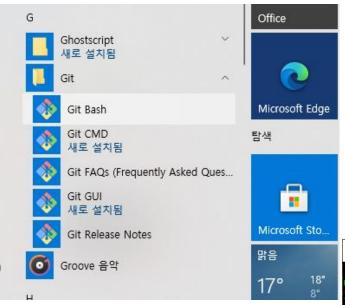
GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.

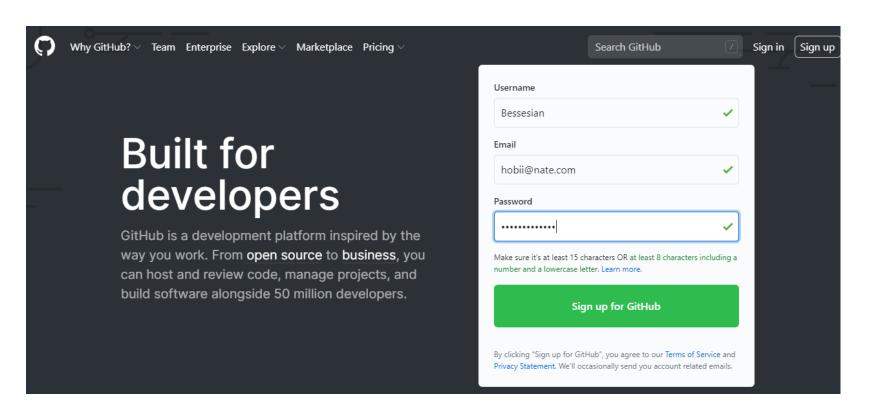


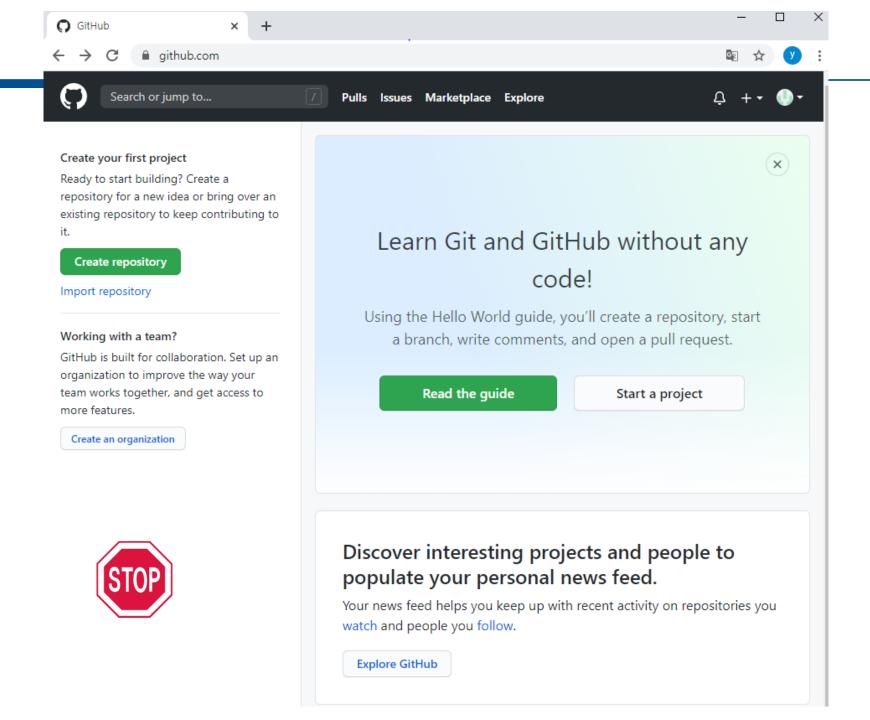


ls / ls -al / cd / . / ..
pwd / mkdir / rmdir
cp / mv / rm / rm -rf

```
MINGW64:/c/Users/user
                                                                                user@CBNU MINGW64 ~
```

- ☐ GitHub 가입
 - https://github.com/





□ Git 초기 설정

```
파일 홈 공유 보기

← → ▼ ↑ → 내 PC → 로컬 디스크 (D:) → Utility → Git → Bessesian

↑ 이름 ↑ 수정한 날짜 유형

바로 가기
● 바탕 화면 ★
● 다운로드 ★
```

☐ Git과 현재 프로젝트 연결

```
user@CBNU MINGW64 /c/Users
$ mkdir /d/Utility/Git/Bessesian/test_project

user@CBNU MINGW64 /c/Users
$ cd /d/Utility/Git/Bessesian/test_project

user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project
$ pwd
/d/Utility/Git/Bessesian/test_project
```

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project

§ git init
Initialized empty Git repository in D:/Utility/Git/Bessesian/test_project/.git/

user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)

§ ls -al
total 4
drwxr-xr-x 1 user 197121 0 10월 5 18:41 ./
drwxr-xr-x 1 user 197121 0 10월 5 18:41 ./
drwxr-xr-x 1 user 197121 0 10월 5 18:41 ./
```

◆ Git과 test_project가 연결됨



- master branch가 생성되고 git bash 현재 폴더 명에 branch 이름(master)
 이 추가됨
- Git은 test_project에서 파일이 생성되거나 삭제되는 것을 감지할 수 있음

◆ test_project 폴더에 test.txt 파일 생성

Test - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

And he thought of those he angered,
For he was not a violent man,
And he thought of those he hurt
For he was not a cruel man
And he thought of those he frightened
For he was not a evil man,

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git status
On branch master

No commits yet

Untracked files:
   (use "git add <file>..." to include in what will be committed)
        test.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- git가 test.txt가 생성된 것은 알고있지만 추적은 하고 있지 않음
- test.txt는 Working Directory에 위치함

- ◆ git에서 관리 혹은 추적
- add 명령어를 통해 Working Directory에서 Staging Directory로 이동



◆ Staging Area에서의 파일 관리

```
Test - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

And he thought of those he angered,
For he was not a violent man,
And he thought of those he hurt
For he was not a cruel man
And he thought of those he frightened
For he was not a evil man,
```

And he understood. He understood himself.

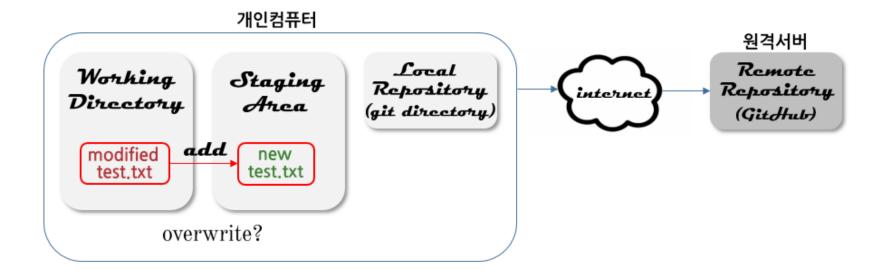
```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git status
On branch master

No commits yet

Changes to be committed:
   (use "git rm --cached <file>..." to unstage)
        new file: test.txt

Changes not staged for commit:
   (use "git add <file>..." to update what will be committed)
   (use "git restore <file>..." to discard changes in working directory)
        modified: test.txt
```

- Unstaged file에 test.txt가 추가됨: 이전에 add된 test.txt와는 다른 파일
- ⊙ 현재 작업 중인 test.txt는 Working Directory에 존재
 - ▶ 작업이 끝난 마지막 시점을 기준으로 추적할 경우 add를 수행



- Git commit
- Staging Area에서 Local repository로 이동
- Local repository로 이동된 파일은 반영구적으로 보존 가능
 - ▶ commit된 시점에 따라 따로 보관됨

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git commit -m "first committed file"
[master (root-commit) d3af9e6] first committed file
1 file changed, 9 insertions(+)
    create mode 100644 test.txt

user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git status
On branch master
nothing to commit, working tree clean
```



```
🥘 test - Windows 메모장
```

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

And he thought of those he angered,
For he was not a violent man,
And he thought of those he hurt
For he was not a cruel man
And he thought of those he frightened
For he was not a evil man,

And he understood. He understood himself.

By MoodyBlues

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git status
On branch master
Changes not staged for commit:
    (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
    modified: test.txt

no changes added to commit (use "git add" and/or "git commit -a")

user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git add test.txt

user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git commit -m "second committed file"
[master aa91232] second committed file
1 file changed, 2 insertions(+)
```



commit ?

❖ 작업공간 안에 있는 모든 파일과 파일의 데이터를 사진 찍듯이 복사해서 저장소에 보존

★ 커밋은 작업공간의 어떤 시점의 스냅샷이라고 할 수 있음

🤏 커밋이란 git commit 명령어로 저장소에 올리는 작업공간의 하나의 변경된 상태를 의미

🥩 하나의 커밋이 더해질 때는 이전 커밋에서 변경된 사항만 추가

🔣 커밋은 서로 연결된 체인 구조를 형성



- Git log
- Git 저장소에 commit한 기록을 보여주는 명령어

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git log
commit aa91232bac9e32a838ddf457f36fa2adab41a9cf (HEAD -> master)
Author: Bessesian <hobii@nate.com>
Date: Tue Oct 6 09:07:33 2020 +0900

    second committed file

commit 6b66944bee006e1fde2e4e8fc72f5ebcef38625e
Author: Bessesian <hobii@nate.com>
Date: Tue Oct 6 08:42:41 2020 +0900

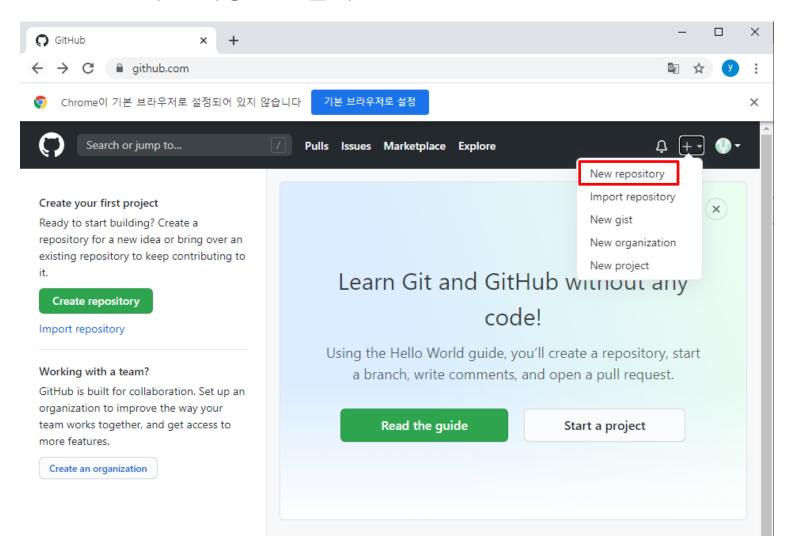
    first committed file
```

- ♦ 실습
- testA.txt, testB.txt, textC.txt 생성 후 한번에 add/commit 수행

```
ls -al
total 8
drwxr-xr-x 1 user 197121
                          0 10월
                                 6 09:17 ./
drwxr-xr-x 1 user 197121
                          0 10월 5 18:41 .../
                          0 10월
                                  6 09:07 .git/
drwxr-xr-x 1 user 197121
                                  6 09:04 test.txt
-rw-r--r-- 1 user 197121 257 10월
-rw-r--r-- 1 user 197121 6 10월
                                 6 09:17 testA.txt
rw-r--r-- 1 user 197121 9 10월 6 09:17 testB.txt
rw-r--r-- 1 user 197121
                          4 10월
                                  6 09:17 testC.txt
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
nothing added to commit but untracked files present (use "git add" to track)
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git add *
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
       new file: testA.txt
       new file: testB.txt
       new file: testC.txt
```

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git commit -m "test, testA, testB, testC file add"
[master 29aa600] test, testA, testB, testC file add
 3 files changed, 3 insertions(+)
 create mode 100644 testA.txt
create mode 100644 testB.txt
 create mode 100644 testC.txt
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git log
commit 29aa6003541d02aef4732303e28132f7ec416c57 (HEAD -> master)
Author: Bessesian <hobii@nate.com>
Date: Tue Oct 6 09:20:58 2020 +0900
    test, testA, testB, testC file add
commit aa91232bac9e32a838ddf457f36fa2adab41a9cf
Author: Bessesian <hobii@nate.com>
Date: Tue Oct 6 09:07:33 2020 +0900
   second committed file
commit 6b66944bee006e1fde2e4e8fc72f5ebcef38625e
Author: Bessesian <hobii@nate.com>
       Tue Oct 6 08:42:41 2020 +0900
Date:
    first committed file
```

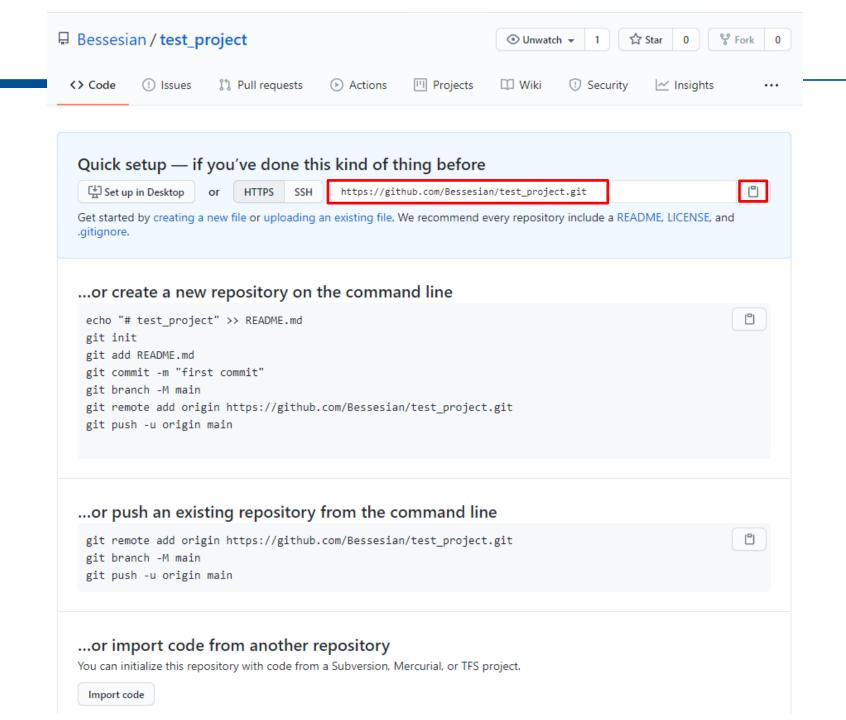
◆ GitHub 온라인 저장소 만들기



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

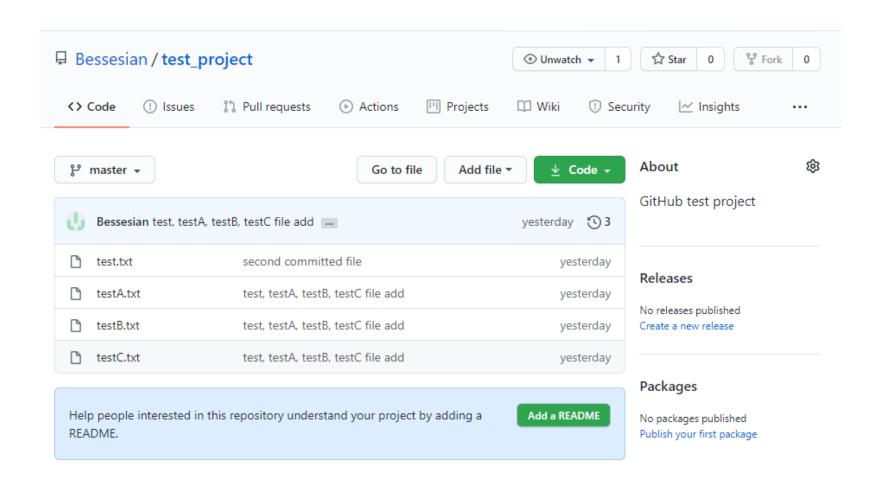
Owner * Re	epository name *		
Bessesian ▼ /	test_project	~	
Great repository names are short and memorable. Need inspiration? How about cautious-octo-funicular? Description (optional)			
Public Anyone on the intern	net can see this repository. You	choose who can o	commit.
O Private You choose who can	see and commit to this reposit	tory.	
Initialize this repository wit Skip this step if you're impo		y.	
Add a README file This is where you can write a long description for your project. Learn more.			
Add .gitignore Choose which files not to tra-	ck from a list of templates. Lea	arn more.	
Choose a license A license tells others what the	ey can and can't do with your	code. Learn more.	
Create repository			



- ◆ 온라인 저장소 등록
 - git bash의 붙여넣기 단축키는 ctrl+v가 아니라 shift+ insert

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git remote add origin https://github.com/Bessesian/test_project.git

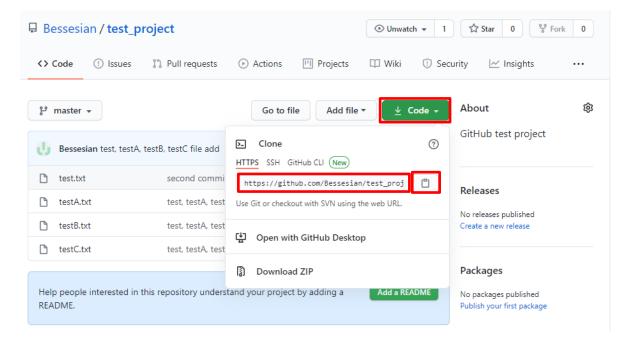
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ git push origin master
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (11/11), 874 bytes | 437.00 KiB/s, done.
Total 11 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Bessesian/test_project.git
* [new branch] master -> master
```



- ◆ GitHub에서 저장소 내려 받기
- ⊙ 저장소가 없는 폴더로 이동

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/test_project (master)
$ cd ..
user@CBNU MINGW64 /d/Utility/Git/Bessesian
$ |
```

⊙ 저장소 주소 복사

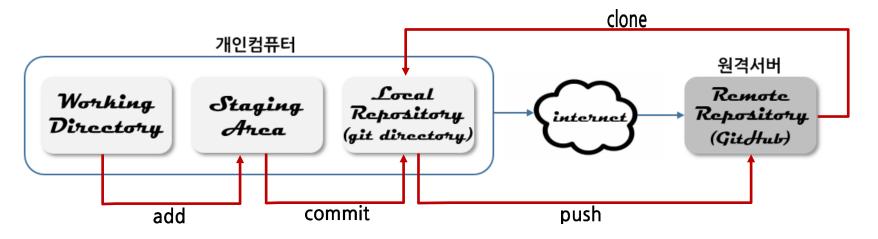


clone

● 온라인 저장소를 local 컴퓨터에 저장하는 명령어

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian
$ git clone https://github.com/Bessesian/test_project.git cloned_project
Cloning into 'cloned_project'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 11 (delta 1), reused 11 (delta 1), pack-reused 0
Unpacking objects: 100\% (11/11), 854 bytes | 2.00 KiB/s, done.
user@CBNU MINGW64 /d/Utility/Git/Bessesian
$ 1s -a1
total 0
drwxr-xr-x 1 user 197121 0 10월 7 11:59 ./
drwxr-xr-x 1 user 197121 0 10월 5 17:38 ../
drwxr-xr-x 1 user 197121 0 10월 7 11:59 cloned_project/
drwxr-xr-x 1 user 197121 0 10월
                                6 09:17 test_project/
```

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian
$ cd cloned_project/
user@CBNU MINGW64 /d/Utility/Git/Bessesian/cloned_project (master)
$ 1s
test.txt testA.txt testB.txt testC.txt
user@CBNU MINGW64 /d/Utility/Git/Bessesian/cloned_project (master)
$ ls -al
total 8
drwxr-xr-x 1 user 197121
                          0 10월 7 11:59 ./
drwxr-xr-x 1 user 197121
                          0 10월 7 11:59 .../
                        0 10월 7 11:59 .git/
drwxr-xr-x 1 user 197121
-rw-r--r-- 1 user 197121 257 10월 7 11:59 test.txt
-rw-r--r-- 1 user 197121 6 10월 7 11:59 testA.txt
-rw-r--r-- 1 user 197121 9 10월 7 11:59 testB.txt
-rw-r--r-- 1 user 197121
                         4 10월 7 11:59 testC.txt
```



♦ 새로운 저장소 industrial-Al 생성

```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI (master)
$ ls -al
total 4
drwxr-xr-x 1 user 197121 0 10월 13 10:14 ./
drwxr-xr-x 1 user 197121 0 10월 13 10:11 ../
drwxr-xr-x 1 user 197121 0 10월 13 10:30 .git/
drwxr-xr-x 1 user 197121 0 10월 13 10:23 programming/
drwxr-xr-x 1 user 197121 0 10월 13 11:02 projects/

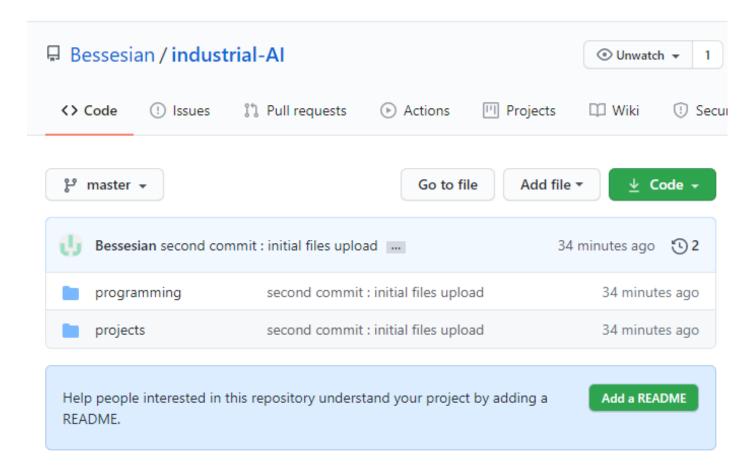
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI/projects (m
$ ls -al
```

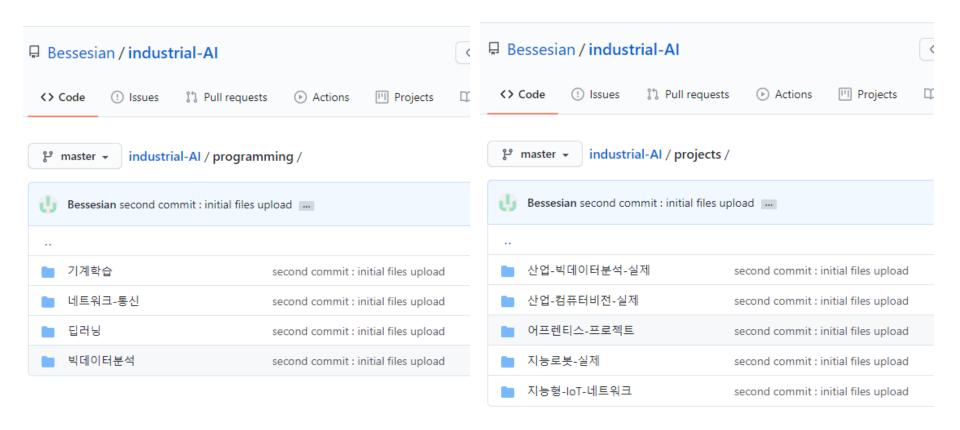
```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI/projects (master) $ ls -al total 0 drwxr-xr-x 1 user 197121 0 10월 13 11:07 ./ drwxr-xr-x 1 user 197121 0 10월 13 10:14 .../ drwxr-xr-x 1 user 197121 0 10월 13 10:12 산 업 -빅 데 이 터 분 석 -실 제 / drwxr-xr-x 1 user 197121 0 10월 13 10:12 산 업 -컴 퓨 터 비 전 -실 제 / drwxr-xr-x 1 user 197121 0 10월 13 10:11 어 프 렌 티 스 -프 로 젝 트 / drwxr-xr-x 1 user 197121 0 10월 13 10:12 지 등 로 봇 -실 제 / drwxr-xr-x 1 user 197121 0 10월 13 10:13 지 등 형 -IoT-네 트 워 크 /
```

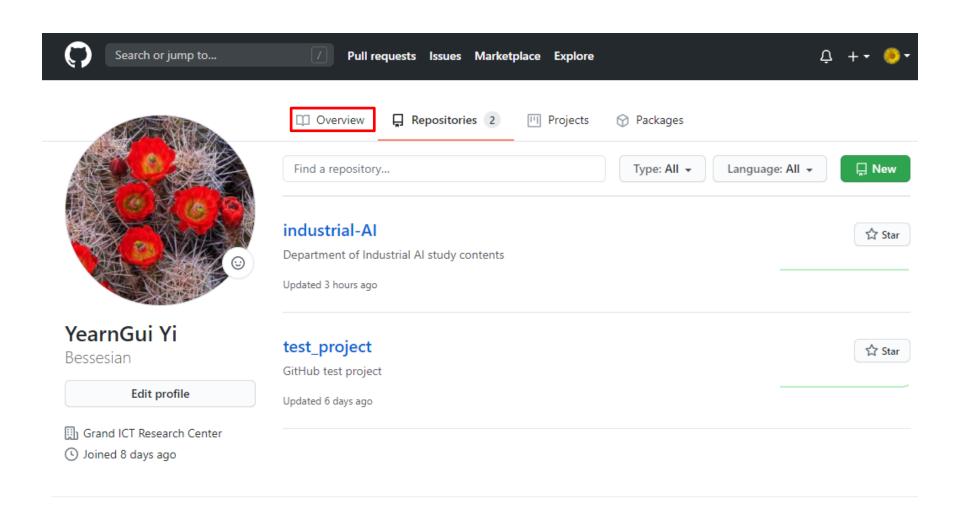
```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI/programming (master) $ ls -al total 0 drwxr-xr-x 1 user 197121 0 10월 13 11:08 ./drwxr-xr-x 1 user 197121 0 10월 13 10:14 ../drwxr-xr-x 1 user 197121 0 10월 13 10:29 기계학습/drwxr-xr-x 1 user 197121 0 10월 13 10:13 네트워크-통신/drwxr-xr-x 1 user 197121 0 10월 13 10:13 데트워크-통신/drwxr-xr-x 1 user 197121 0 10월 13 10:13 데데이터분석/drwxr-xr-x 1 user 197121 0 10월 13 10:13 빅데이터분석/
```

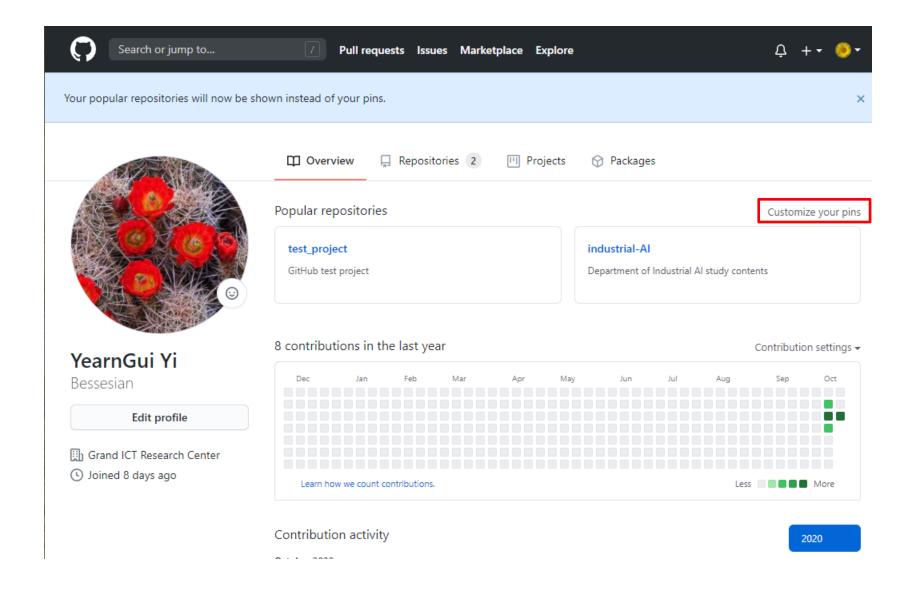
```
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI/programming (master)
$ touch ./네트워크-통신/iot_comm.src
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI/programming (master)
$ touch ./답러당/cnn_algorithm.pv
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI/programming (master)
$ touch ./빅데이터분석/naive-bayesian.py
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI (master)
$ git add *
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI (master)
$ git commit -m "second commit : initial files upload"
[master 6197386] second commit: initial files upload
9 files changed, 0 insertions(+), 0 deletions(-)
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI (master)
$ git remote add origin https://github.com/Bessesian/industrial-AI.git
user@CBNU MINGW64 /d/Utility/Git/Bessesian/industrial_AI (master)
$ git push origin master
Enumerating objects: 15, done.
Counting objects: 100\% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100\% (6/6), done.
Writing objects: 100% (13/13), 1.12 KiB | 576.00 KiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Bessesian/industrial-AI.git
   483d426..6197386 master -> master
```

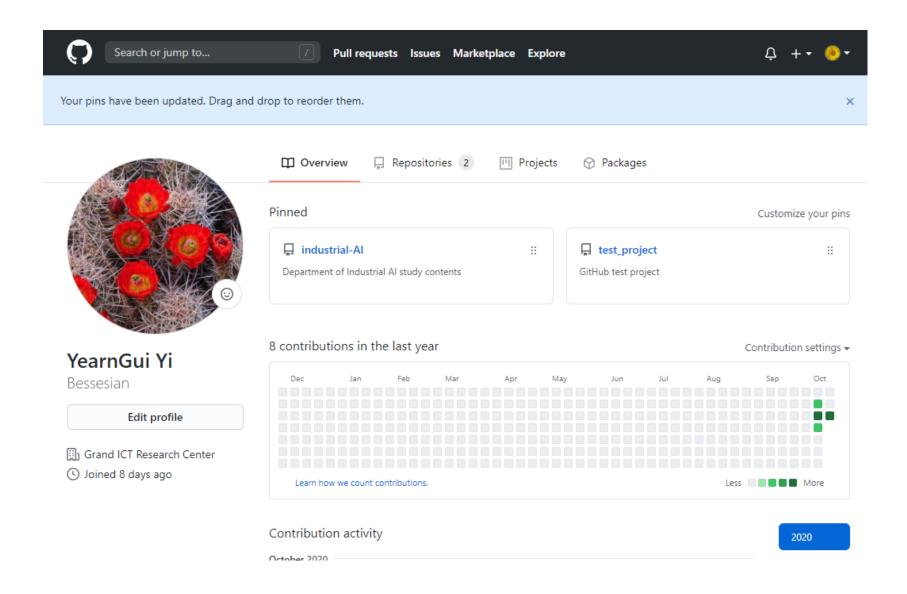
♦ 새로운 저장소 industrial-Al 생성









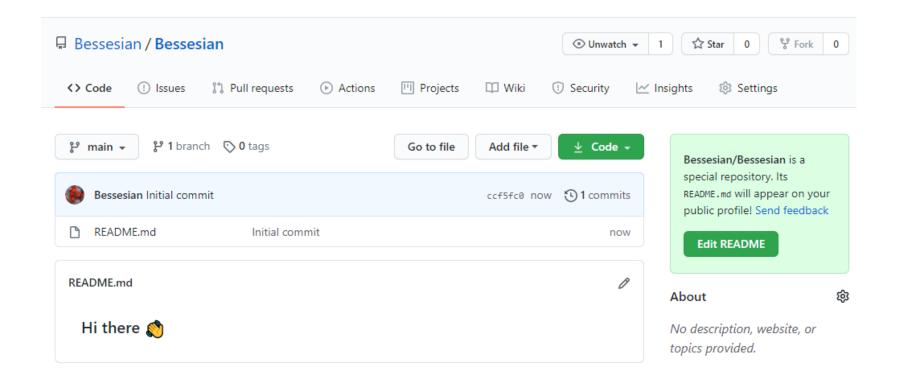


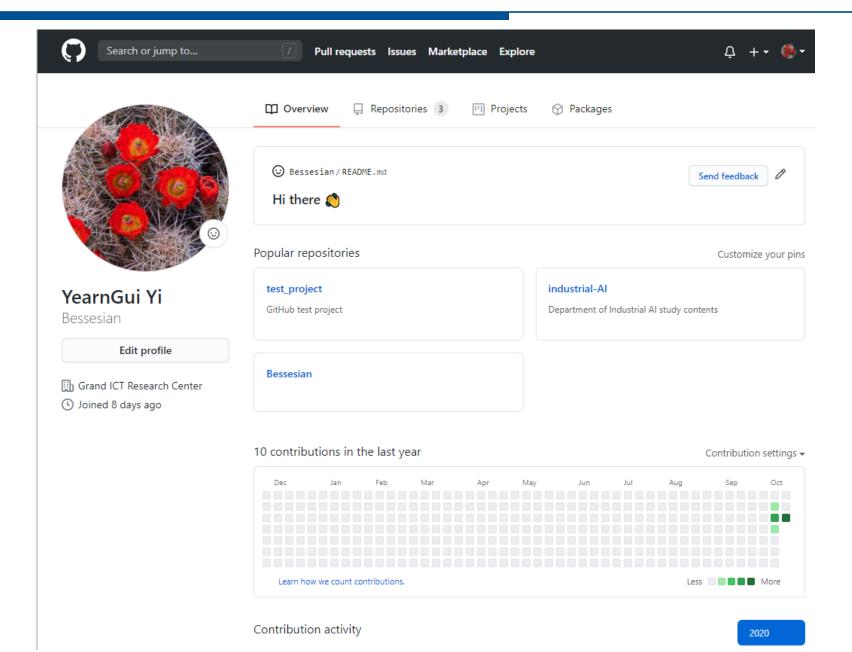
◆ README로 GitHub Profile 꾸미기

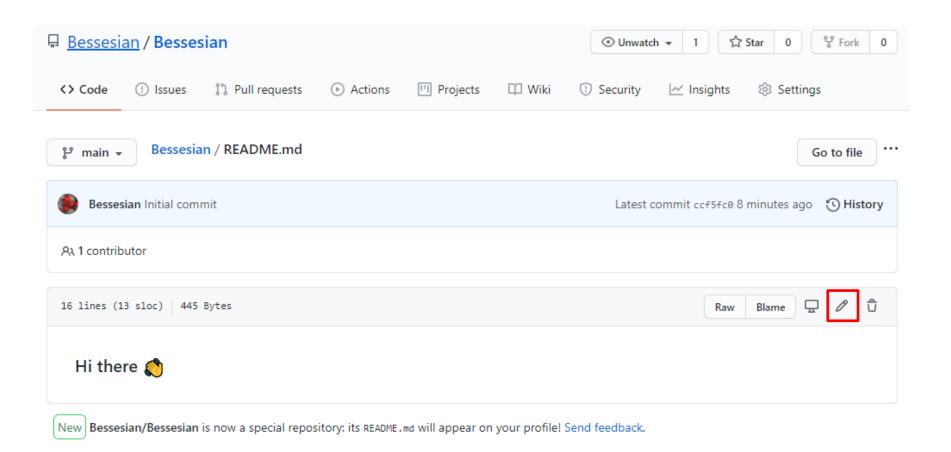
This is where you can write a long description for your project. Learn more.

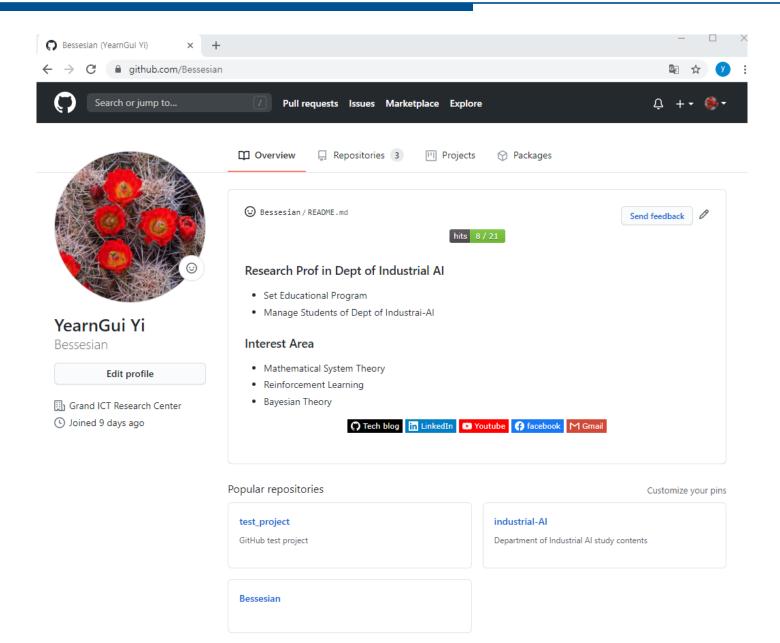
- 자신의 GitHub로 이동 후 Create new repository 선택
 - ► Repository의 이름을 자신의 닉네임으로 설정

Create a new repository A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository. Owner * Repository name * Bessesian * Bessesian You found a secret! Bessesian/Bessesian is a styre repository that you can use to add a README and to your GitHub profile. Make sure it's public and initialize it with a README to get started. Description (optional) Public Anyone on the internet can see this repository. You choose who can commit. You choose who can see and commit to this repository. Initialize this repository with: Skip this step if you're importing an existing repository. Add a README file









◆ Markdown 문서 작성

- 텍스트 기반의 마크업언어로 2004년 존그루버에 의해 만들어 졌음
 - ▶ 특수기호와 문자를 이용한 매우 간단한 구조의 문법을 사용
 - ▶ 웹에서도 보다 빠르게 컨텐츠를 작성하고 보다 직관적으로 인식 가능
- GitHub에서의 README.md 파일
 - ► GitHub 저장소 Repository에 관한 정보를 기록
 √설치방법, 소스코드 설명, 이슈 등을 간단하게 기록하고 가독성을 높일 수 있음

⊙ 장점

- ▶ 간결하고 별도의 도구없이 작성 가능
- ► 다양한 형태로 변환이 가능하고 텍스트(Text)로 저장되기 때문에 용량이 적어 보관이 용이
- ▶ 텍스트파일이기 때문에 버전관리시스템을 이용하여 변경이력을 관리 가능
- ▶ 지원하는 프로그램과 플랫폼이 다양

⊙ 단점

- ▶ 통일된 표준이 없기 때문에 도구에 따라서 변환방식이나 생성물이 다름
- ▶ 모든 HTML 마크업을 대신하지 못함

- ◆ Markdown 문법
 - Header

```
# 제목 1
## 제목 2
### 제목 3
#### 제목 4
##### 제목 5
###### 제목 6
```

제목 1

제목 2

제목 3

제목 4

제목 5

제목 6

Text

이텔릭체는 *별표*를 사용

두껍게는 **별표두개**를 사용

두꺼운 이탤릭은 ****별표세계****를 사용

취소선은 ~~물결표시~~를 사용

이텔릭체는 *별표*를 사용

두껍게는 **별표두개**를 사용

두꺼운 이탤릭은 *별표세개*를 사용

취소선은 물결표사를 사용

Lists (Ordered/Unordered)

- 1. First Item
- Second Item
- 3. Third Item
- 1. Fourth Item
- </br>
- * First
 - * Second
 - * Third
- </br>
- ▶ *, +, 모두 동일한 효과

- 1. First Item
- 2. Second Item
- 3. Third Item
- 4. Fourth Item
- First
 - Second
 - Third

Code

```
<code>
for i in range(1,2):
  print(i)
</code>
```

for i in range(1,2):
 print(i)

Link

```
[Google](http://google.com)
<http://google.com>
```

Google

http://google.com

• Image

```
![Forest](./images/iForest.PNG)
```



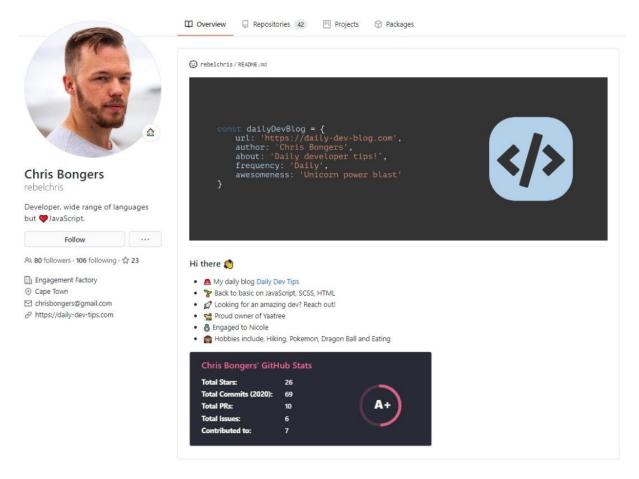
Table

First Header	
Content Cell	Content Cell
Content Cell	Content Cell

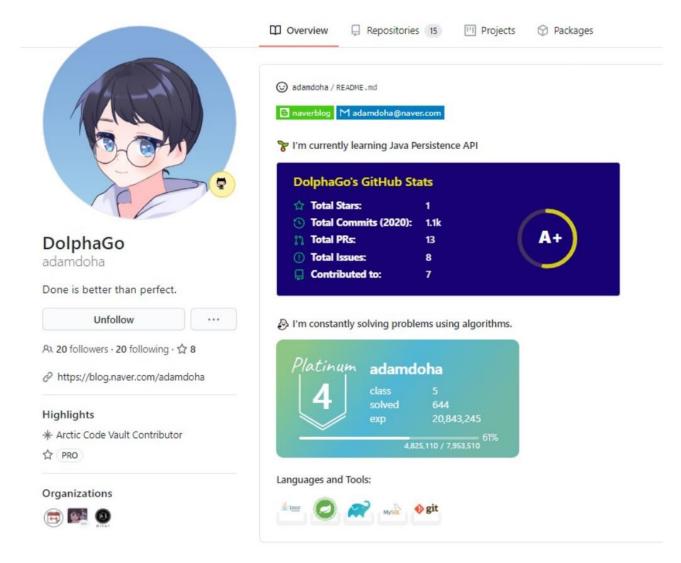
First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

First Header	Second Header	Third Header
Left	Center	Right

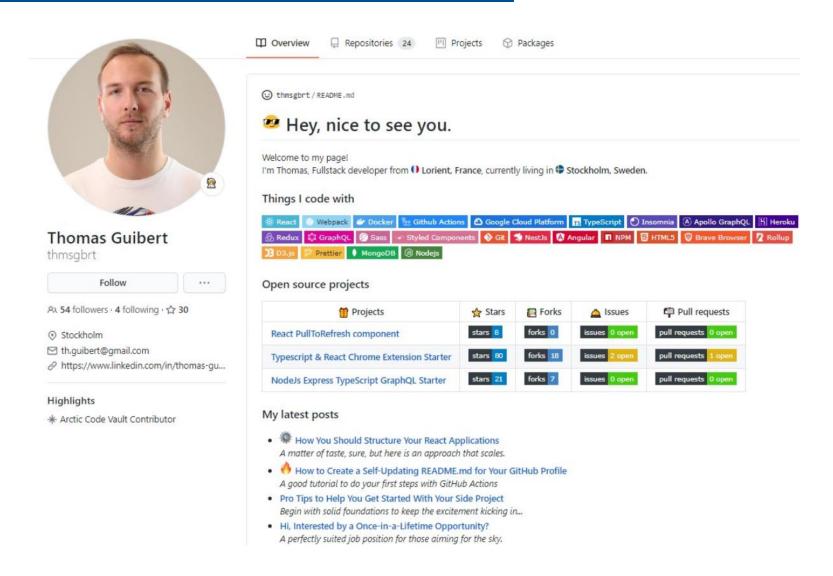
◆ GitHub 포트폴리오 작성 예



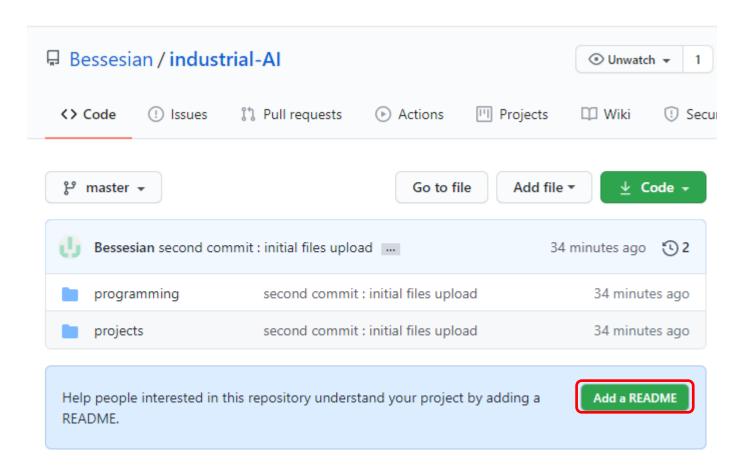
https://github.com/rebelchris



https://github.com/adamdoha



https://github.com/thmsgbrt/thmsgbrt



Grand ICT 연구센터

감사합니다

