PORTFOLIO

강의계획서

아시아 직업교육 허브대학

2021 학년도 2학기		전공	컴퓨터정보공학과		학부	컴퓨터공학부	
과목명		졸업작품(종합설계)(2015085-PA)					
강의실 과 강의시간		화:5(3-218),6(3-218),7(3-218),8(3-218) 학점 4					
교과분류		실습			시수	4	
담당 교수	강환수 + 연구실: 2호관-706 + 전 화: 02-2610-1941 + E-MAIL: hskang@dongyang.ac.kr + 면담가능기간: 화 11:00~11:50						
학과 교육목표							
과목 개요	○ 수업운영방안: 비대면(코로나19 방역상황 호전시 대면으로 전환) ○ 온라인 수업 운영방법: 녹화형/실시간 화상 수업으로 병행하여 진행 ○ 과목 개요: 컴퓨터정보공학과에서 학습한 내용들을 종합적으로 활용하여 실제로 프로젝트를 구현해 보는 과목으로서 팀별로 하나의 프로젝트를 완성하는 것을 목표로한다. 이러한 프로젝트 개발 활동은 졸업 후 사회에 진출했을 때 실제적으로 행하게 될 일의 실전준비가 될 것이다. 또한 하나의 프로젝트를 처음부터 시작하여 끝까지 스스로 완성하여 봄으로써 학습내용들을 체계화하고 실용적인 실력을 키우는 데 그 목적이 있다. 팀별로 완성된 작품은 졸업작품 최종 발표와 학술대회 등을 통해 공개되며 평가된다.						
학습목표 및 성취수준	대학 교육목표 및 학과 교육목표를 달성하기 위하여 본 교과목은 컴퓨터정보공학과 교육과정에서 배 운 다양한 기술을 이용하여 팀별로 하나의 프로젝트를 수행해 프로젝트 작품을 개발하는 것을 목표로 한다. 본 과목을 통해 다음을 성취할 수 있다. - 팀별로 자신의 역할을 정하고 팀 프로젝트 작품을 기획하고 설계할 수 있다. - 작성된 설계서를 기반으로 지금까지 배운 과목별 학습내용을 이용해서 팀 프로젝트 작품을 개발할 수 있다.						
			도서명	저자		출판사	비고
주교재	유인물			-		-	
수업시 사용도구	각 팀별로 프로젝트 개발 시 필요한 재료 및 도구						
평가방법	출석 20% + 중간고사(수행과정 평가) 30%(프로젝트보고서 1차발표/5, 2차발표/10, 중간개인/15) + 기말고사(과정 및 최종발표 평가) 30%(프로젝트 최종발표/15, 최종보고서/5, 개인 팀내기여도/10) + 교내외 학술대회 및 작품전 참가 및 수상 10% + 취업활동 및 기타 10%						
수강안내	강의완료 후 학생들은 팀 프로젝트의 일원으로서 하나의 작품을 개발하기 위해 작품의 기획부터 개발, 그리고 발표까지 프로그램의 개발 프로세스 모든 과정을 경험할 수 있으며 자신의 역할을 충분히 수행 할 수 있다. 선수과목으로 1학기의 시스템분석설계 과목을 학습한 학생들이 수강해야 한다						

1 주차	[개강일(9/1), 수강정정(9/6 ~ 9/7)]
학습주제	<강의 소개> 첫 시간의 수강 학생들 확인과 이 과목에서 학습하게 되는 졸업작품에 대한 개요 및 강의 방법, 평가 방법 등에 대해 설명한다. 특히, 시스템분석설계 과목과 연계된 졸업작품의 진행 방법과 완성된 졸업작품을 평가하기 위한 졸업작품평가회와 전산학부학술대회의 출품에 대해서 소개한다. 더불어 강의 목표와 강의 시 유의사항에 대해서도 설명한다.
목표및 내용	학습대상자 출석 및 선수과목 이수 여부 확인 졸업작품 강의 내용 및 목표 소개 학습 방법 및 학습 평가 소개 팀구성 및 확인
미리읽어오기	1. 팀별 작품 진행 상황 파악 2. 방학 중 팀별 활동 보고 준비
과제,시험,기타	
2 주차	[2주]
학습주제	<탐별 작품진척도 점검 및 지도> 각 팀별로 현재 진행 주인 팀 프로젝트 작품에 대한 작품 진행 상황 점검과 방학 중 팀별 활동 내 역 파악
목표및 내용	1. 각 팀별 프로젝트 구성 인원을 확인한다. 2. 방학 중 팀별 활동 내역을 발표하여 각 팀별 프로젝트 진행상황을 파악한다. 3. 각 팀별 프로젝트 진행 상황에 따라 앞으로의 프로젝트 개발 계획을 수립한다. 4. 팀별 프로젝트 진행 상황에 따라 자신이 맡은 역할의 세부 계획을 수립한다.
미리읽어오기	
과제,시험,기타	졸업작품 프로젝트보고서 1차발표
3 주차	[3주, 추석연휴(9/20~9/21) -> 보강(9/30,12/14)]
학습주제	<탐별 작품 변경 사항 확인> 팀별 프로젝트 작품의 기획과 달리 변경 또는 수정된 기능들을 확인하고 변경된 사용자 인터페이 스 등을 점검한다. 이를 통해 최종적으로 개발해야 하는 팀 프로젝트 사용자 인터페이스 및 기능 등을 확정한다.
목표및 내용	1. 팀 프로젝트 시작 시 기획 및 설계했던 내용과 변경된 기능 및 구현 내역을 확인한다. 2. 변경 및 수정된 부분에 대한 사용자 인터페이스와 기능에 대해 재설계를 진행한다. 3. 이에 따라 팀별 또는 개인별 개발 계획을 변경하고 수정한다.
미리읽어오기	
과제,시험,기타	변경된 최종 설계서
4 주차	[4주, 추석연휴(9/22) -> 보강(12/8)]
학습주제	<팀별 작품 개발을 위한 관련 기술 학습> 수정된 팀별 작품을 위해 필요한 관련 작품 개발 자료 및 기술들을 수집하고 이를 학습한다.
목표및 내용	1. 수정 및 추가된 팀 프로젝트 기능들을 위해 관련 기술 자료들을 수집한다. 2. 수집된 작품 관련 소스들을 분석하고 이를 작품에 반영할 수 있도록 학습한다. 3. 작품을 작품개발서버에 설치하기 위한 관련 자료들을 학습하고 이해할 수 있다.
미리읽어오기	
과제,시험,기타	

DOMOYANG MIRAE UNIVERSITY				
5 주차	[5주, 개천절(10/3) -> 보강(10/4)]			
학습주제	<팀별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝 트의 각 역할을 학습한다.			
목표및 내용	전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 다. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.			
미리읽어오기				
과제,시험,기타	졸업작품 프로젝트보고서 2차발표			
6 주차	[6주, 한글날(10/9) -> 보강(12/13)]			
학습주제	<팀별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝 트의 각 역할을 학습한다.			
목표및 내용	1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 4. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.			
미리읽어오기				
과제,시험,기타				
7 주차	[7주]			
학습주제	<팀별 작품 서버 설치> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 개발된 작품을 서버에 설치하여 동작여부를 확인한다.			
목표및 내용	 현재까지 개발된 작품을 서버에 설치하고 이를 동작시킬 수 있다. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다. 			
미리읽어오기				
과제,시험,기타	각 팀 프로젝트 구성원별 기능별 개발 진행 사항 전공동아리 킨텐스작품전 참가			
8 주차	[8주]			
학습주제	<중간고사> 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 개발 내용을 시연 및 발표한다.			
목표및 내용	1. 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 개발 내용을 발표 자료로 만들고 이를 프리젠테이션한다. 2. 지금까지 개발한 팀 프로젝트 작품의 중간 평가를 위해 작품의 동작 과정을 시연한다.			
미리읽어오기				
과제,시험,기타	중간고사 또는 보고서 발표			
취제,작품,기의				

DONGYANG MIRAE UNIVERSITY					
9 주차	[중간고사 기간 : 10/27(화)~11/2(화)]				
학습주제	<탐별 작품 개발 진행> 최종 졸업작품 발표회 전까지 기획한 팀 프로젝트 작품의 개발을 진행하며 이를 통해 팀 프로젝트의 각 역할을 학습한다.				
목표및 내용	1. 전체적인 팀별 작품 개발 진도 및 진행 상황을 파악할 수 있다. 2. 팀 프로젝트의 각 역할별 기능별 진행 상황을 파악할 수 있다. 3. 팀 프로젝트 구성원별로 개발 진행 상황을 보고하고 개발시 문제점을 토론하고 해결할 수 있다. 4. 다음주까지 개발 계획을 확인하고 필요한 요구사항을 요청할 수 있다.				
미리읽어오기					
과제,시험,기타	각 팀 프로젝트 구성원별 기능별 개발 진행 사항				
10 주차	[10주]				
학습주제	<학부 학술대회 및 경진대회 준비> 학부에서 개최하는 학술대회 및 경진대회에 각 팀별 프로젝트 작품을 출품하기 위한 작업을 진행 한다.				
목표및 내용	 각 팀별 현재까지 개발 진행 사항을 파악하고 학부 학술대회 및 경진대회 출품여부를 지도교수와 협의한다. 학부 학술대회 및 경진대회 브로셔를 위한 작품 소개 자료를 작성한다. 학부 학술대회 및 경진대회 발표자료 및 데모 동영상 등을 준비한다. 				
미리읽어오기					
과제,시험,기타	학부 학술대회/경진대회 브로셔 학부 학술대회/경진대회 발표자료 학부 학술대회 데모 동영상				
11 주차	[11주]				
학습주제	<작품 설치 및 최종 테스트> 개발된 작품을 서버에 설치하고 각 모듈별 동작 과정과 전체 동작 과정을 테스트한다.				
목표및 내용	1. 개발된 작품을 서버에 최종적으로 설치 완료한다. 2. 각 모듈별(웹, 앱, 서버 등) 동작 과정을 테스트한다. 3. 전체 작품의 동작 시나리오를 만들고 이에 따라 테스트를 완료한다.				
미리읽어오기					
과제,시험,기타	작품 동작 시나리오				
12 주차	[12주]				
학습주제	<최종졸업작품 발표> 학과 교수님들을 평가위원으로 한 최종졸업작품발표회를 개최하여 각 팀별로 개발된 작품을 발 표하고 시연한다. 이를 통해 졸업작품 과목을 평가한다.				
목표및 내용	 각 팀별로 팀 프로젝트 작품 발표자료를 만들어 졸업작품평가전에서 발표하고 개발된 작품의 동작 여부를 시연한다. 평가위원들로부터 지적받은 내용들을 체크리스트로 만들어 수정 보완 사항을 파악한다. 				
미리읽어오기					
과제,시험,기타					

23:20

DONGYANG MIRAE U	NIVERSITY 이시아 역입교육 어느내의
13 주차	[13주]
학습주제	<개발된 작품 보완 작업> 졸업작품발표회에서 평가위원들로부터 지적받은 수정사항 및 추가사항들의 개발을 보완한다. 이 에 대한 체크리스트를 만들어 보완여부를 확인한다.
목표및 내용	 평가위원들로부터 지적받거나 부족한 기능들에 대한 체크리스트를 만든 후 만들어진 체크리스트에 따라 보완사항들을 개발한다. 보완된 내용들을 테스트하여 작품 개발을 완료한다.
미리읽어오기	
과제,시험,기타	
14 주차	[14주]
학습주제	<개발된 작품 보완 작업 확인> 지도교수를 통해 졸업작품발표회에서 지적된 사항들의 수정 및 보완이 완료되었는지 확인한다.
목표및 내용	 작품평가회에서의 지적사항 보완 여부를 확인한다. 아직 부족한 부분에 대해서는 재확인이 필요한지 여부를 판단하고 이에 따라 최종 작품 개발을 완료한다
미리읽어오기	
과제,시험,기타	
15 주차	[기말고사기간 : 12/15(수)~21(화)]
학습주제	<최종 완성된 작품 제출> 졸업 작품 과목을 통해 개발된 모든 내용을 제출하고 개발된 작품을 항시 확인할 수 있도록 서버 에 유지관리 작업을 수행한다.
목표및 내용	1. 최종적으로 완성된 작품을 디지털 형식(CD, USB 등) 으로 제작하여 제출한다. 2. 개발된 작품을 서버에 설치 완료하여 항시 개발된 작품이 동작할 수 있도록 유지한다. 3. 작품 개발을 위해 작성한 관련 자료(기획서,설계서,발표자료 등)을 정리하여 제출한다.
미리읽어오기	
과제,시험,기타	최종작품 관련 일체 자료
수업지원 안내	장애학생을 위한 별도의 수강 지원을 받을 수 있습니다. 언어가 문제가 되는 학생은 글로 된 과제 안내, 확대문자 시험지 제공 등의 지원을 드립니다.

A Table of Contents.

- 1 머리말
- **2** git
- 3 로그인 회원가입
- 4 후기

Part 1, 머리말



Part 1, 머리말

본 교과목 포트폴리오는 버전관리 시스템 git과 졸업작품을 통해 자기주도 학습한 내용에 대해소개하는 포트폴리오다.

본 교과목 포트폴리오는 git에 대한 이해를 도와주고 git의 사용법을 알려주며, 졸업작품을 통해 자기주도 학습한 내용에 대해 소개한다.

본 포트폴리오는 git에 대한 사용방법을 모르는 분들이 봤으면 좋겠다.

Part 2, git



Part 2, git이란?

git이란 **분산형 버전 관리 시스템(Version Control System)** 의 한 종류이며, 빠른 수행 속도에 중점을 둔다.

예를 들어, 우리가 레포트를 제출한다고 가정했을 때, 처음에 저장했을 때 'report.txt'라고 저장을 했다가 수정을 하면서 'report_final.txt'로 저장하고 또 수정을 거치면서 'report_final_final.txt'...로 수정을 하게 되는 경험이 있었을 것이다. 여기서 이 파일들을 복사,백업,저장 등을 하였고, 이러한 것을 버전 관리라고 부른다.

Part 2, git의 장점

1. 분산적인 개발

깃(Git)을 사용하는 전체 개발 내역을 각 개발자의 로컬 컴퓨터로 복사할 수 있습니다. 나중에 서로 수정된 내역을 합치기(Merge)할 수도 있으며 이 때 깃의 고유한 프로토콜을 이용하게 됩니다.

2. 효율적인 개발

깃(Git)은 일반적인 다른 버전 관리 시스템보다 성능이 뛰어나며, 변경 이력이 많더라도 변경된 내용만 처리한다는 점에서 메모리 적인 효율성 또한 뛰어납니다.

3. 변경 이력 보장

작업된 모든 내역(Commit 내역)들은 모두 별도의 영역에서 관리되어 안전하게 프로젝트를 운영할 수 있습니다.

Part 2, git의 영역

Git은 3가지 영역으로 관리된다.

1. 작업영역(Working Directory)

- 실제 프로젝트 디렉터리, git 이력과 관련된 정보가 저장 되어있는 .git을 제외한 모든 영역을 말한다.
- 실제 코드를 수정하고 추가하는 변경이 이루어지는 영역

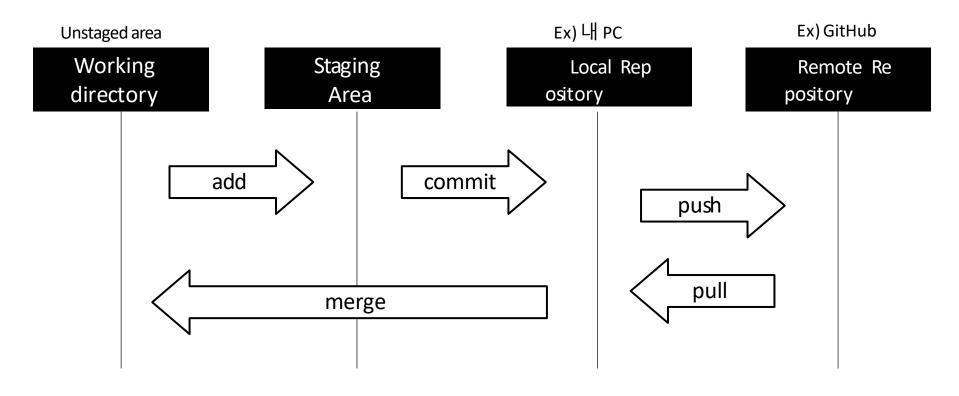
2. Index

- Working Directory에서 Repository로 정보가 저장되기 전 준비 영역
- 파일 상태를 기록, 스테이징 한다고도 표현하며, Staging Area로 불리기도 함.
- .git/index 파일로 관리된다.
- git add 명령어로 Working Directory에서 Index영역으로 정보가 저장된다.
- git commit 명령어로 Index영역에서 Repository로 정보가 저장된다.

3. 저장소(Repository)

- 파일이나 폴더를 변경 이력별로 저장해 두는 곳
- .git 디렉터리 내에 존재한다.
- Local Repository : 내 PC에 파일이 저장되는 개인 저장소
- Remote Repository : 파일이 원격 저장소 전용 서버에서 관리되며 여러 사람이 함께 공유하기 위한 저장소

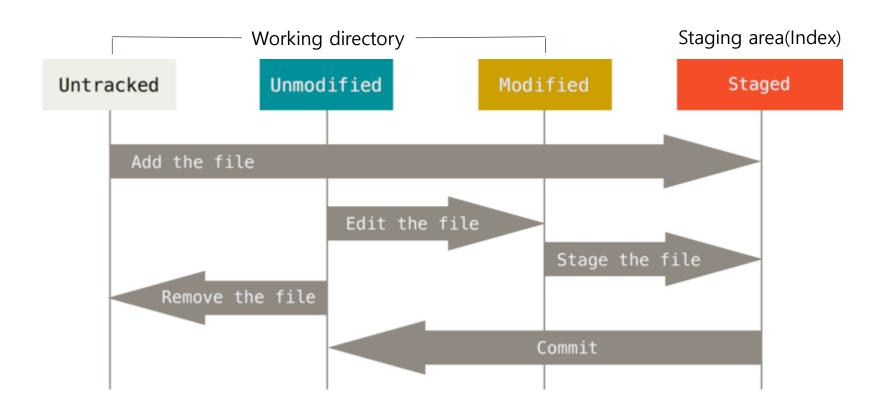
git의 영역(2)



가장 기본적인 작업의 흐름

Working directory -> Staging area (add) → Staging area -> 지역저장소 (commit) → 지역저장소 -> 원격저장소(push)

Part 2, 파일의 상태



Part 2,

git 명령어

```
$ git config --global user.name "Lee Hye Wook" $ git config --global user.email "dlgpdnr123@naver.com
```

- git 사용자 이름과 이메일 설정

```
$ git config --list
- 리스트를 통해 확인
```

```
leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio
$ git config --global user.name "Lee Hye Wook"

leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio

$ git config --global user.email "dlgpdnr123@naver.com"

leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio

$
```

```
vook@DESKTOP-BM1VPBL MINGW64 ~/portfolio
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs_clean -- %f__
  |ter.lfs.smudge=git-lfs smudge -- %f
|ter.lfs.process=git-lfs filter-process
 ilter.lfs.required=true
nttp.ss|backend=openss|
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocr|f=true
core.fscache=true
core.symlinks=false
core.editor=notepad
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=Lee Hye Wook
user.email=dlgpdnr123@naver.com
gui.encoding=utf-8
```

Part 2,

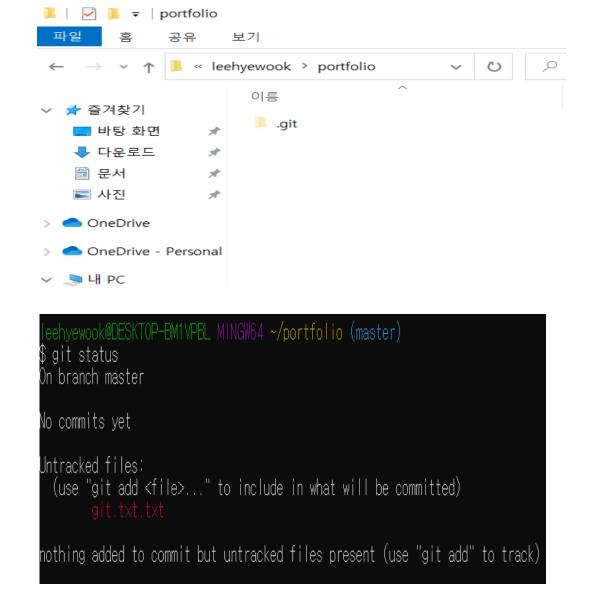
git 명령어

\$ git init

현재 디렉토리에 git 저장소를 생성

\$ git status

git.txt 파일 생성 후 상태 확인



\$ git add

- Untracked 에서 Index(staging area)로, Modified 에서 Index로 파일을 옮김

\$ git commit -m "메세지"

- 메시지와 함께 커밋하기 Index(staging area)에서 깃 레포지토리 저장소로

\$ git commit -am "메세지"

- 메시지와 함께 커밋하기 작업디렉토리에서 index를 거치지 않고 바로 깃 레포지토리 저장소로

```
leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio (master)

git commit -m "example"
[master (root-commit) b90c00d] example

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 git.txt.txt

leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio (master)

git status
On branch master

nothing to commit, working tree clean
```

\$ git log

- 커밋기록(history)보기(author영역의 이름과 이메일 주소)

\$ git log -p

- 파일에 대한 버전 이력 보기

\$ git show

- 현재 브랜치의 가장 최근 커밋 정보를 확인함

\$ git show 커밋해시값(커밋sha-1)

- 특정 커밋 정보를 확인함

\$ git show HEAD

- HEAD포인터가 가리키는 커밋 정보를 확인함

\$ git restore 파일명

- 작업 디렉토리로 되돌리기

\$ git restore --staged 파일명

- add된 파일을 다시 작업 디렉토리로 되돌린다.

\$ git diff

- 파일의 어떤 내용이 변경되었는지 차이점을 비교 (working directory와 staging area간의 비교도 가능)

\$ git diff --staged

- commit된 파일 상태와 add된 파일 상태 비교

```
| leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio (master)
$ git diff
| diff --git a/git.txt.txt b/git.txt.txt
| index e69de29..a9d28ba 100644
| --- a/git.txt.txt
| +++ b/git.txt.txt
| @@ -0,0 +1 @@
| +포트폴리오 입니다.
# No newline at end of file
```

\$ git reset

- 커밋을 아예 취소

\$ git reset --hard <돌아가고 싶은 커밋>

- 돌아가려는 이력 이후의 모든 내용을 삭제 (head랑 index랑 working directory 셋 다 이동시킴)

\$ git reset --mixed <돌아가고 싶은 커밋>

- 돌아가려는 이력 이후의 모든 내용을 삭제하지 만 인덱스는 초기화됨 결국 다시 복원 하려면 add와 commit 필요(working directory로감)

\$ git reset --soft <돌아가고 싶은 커밋>

- 돌아가려 했던 이력으로 되돌아 갔지만, 이후의 내용이 지워지지 않고, 해당 내용의 인덱스 또 한 그대로 있음 바로 다시 커밋 할 수 있는 상 태로 남아있는 것(commit만 필요)

```
leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio (master)
$ git reset --hard HEAD^*
HEAD is now at b90c00d example

leehyewook@DESKTOP-BM1VPBL MINGW64 ~/portfolio (master)
$ git log
commit b90c00d451fcb4f602e51f45511d350ec034b557 (HEAD -> master)
Author: Lee Hye Wook <dlgpdnr123@naver.com>
Date: Mon Nov 29 22:22:25 2021 +0900

example
```

Part 3, 로그인 회원가입



Firebase라?



Firebase는 웹과 모바일 개발에 필요한 기능을 제공하는 서비스이다. 기능으로는 실시간 데이터베이스, 간편한 사용자 인증, 클라우드 저장소, 호스팅, 앱 테스트와 수익 창출을 도와주는 등 다양한 기능을 제공해준다.

로그인 회원가입 파트를 구현하기 위해 여러가지 기능중 Authentication(사용자 로그인 및 ID 관리)과 실시간 데이터베이스(클라우드에 호스팅 된 실시간의 비관계형 데이터베이스)를 사용하였다.

Part 3,

Firebase와 Android studio 연동



Firebase gives you the tools and infrastructure from Google to help you develop, grow and earn money from your app. Learn more

Analytics

Measure user activity and engagement with free, easy, and unlimited analytics. More info

Authentication

Sign in and manage users with ease using popular login providers like Google Sign-In, Facebook, and others. You can even use a custom authentication system. More info

Realtime Database

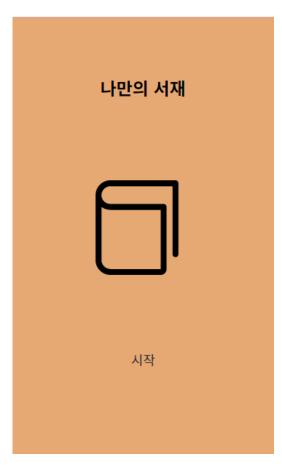
Store and sync data with this cloud-hosted NoSQL database. Data is synced across all clients in realtime and remains available when your app goes offline. More info

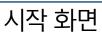
- 1 Connect your app to Firebase
 - ✓ Connected
- Add the Firebase Authentication SDK to
 - ✓ Dependencies set up correctly

NOTE: After adding the SDK, here are some other helpful configuration

Do you want an easier way to manage library versions?
 You can use the Firebase Android BoM to manage your Firebas

Part 3, 레이아웃







로그인 화면



회원 가입 화면

MainActivity(시작화면)

시작화면에 "시작"이라는 텍스트뷰를 클릭할 때 LoginActivity로 이동

```
package com.example.login_project;
import ...
public class MainActivity extends AppCompatActivity {
    @Override
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView start = findViewById(R.id.start);
        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(),LoginActivity.class);
                startActivity(intent);
        });
```

SignUpActivity(회원가입화면)

createUserWithEmailAndPassword 메소드 를 사용하여 이메일 주소와 비밀번호를 가져와 기존에 중복이 되는지 안되는지 확인후 신규 사용자를 만듬

```
<u>A</u> 11 火 1 ∧
private void signUp(){ //회원가입 절차
   String email = ((EditText)findViewById(R.id.idEditText)).getText().toString().trim();
   String password = ((EditText)findViewById(R.id.passwordEditText)).qetText().toString().trim();
   String passwordCheck = ((EditText)findViewById(R.id.passwordCheckEditText)).getText().toString().trim();
   if(email.length()>0 && password.length()>0 && passwordCheck.length()>0){ // 이메일 또는 패스워드 또는 패스워드 확인 부분이 비어져있을때
       if(password.equals(passwordCheck)){ // 비밀번호와 비밀번호 확인이 같다면
           mAuth.createUserWithEmailAndPassword(email, password) //회원가입 진행
                   .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
                       @Override
                       public void onComplete(@NonNull Task<AuthResult> task) {
                           if (task.isSuccessful()){
                             final FirebaseUser user = mAuth.getCurrentUser();
                               UserAccount account = new UserAccount();
                              account.setIdToken(user.getUid());
                               account.setEmailId(user.getEmail());
                               account.setPassword(password);
                               mDatabadeRef.child("UserAccount").child(user.getUid()).setValue(account);
                              startToast( msg: "회원가입에 성공하였습니다.");
                              Intent intent = new Intent(getApplicationContext(),LoginActivity.class);
                               startActivity(intent);
                           } else {
                               Toast.makeText(getApplicationContext(), text: "실패", Toast.LENGTH_SHORT).show();
                               if(task.getException() != null){
                                   startToast(task.getException().toString());
```

LoginActivity(로그인 화면)

로그인 화면에 있는 회원가입 버튼을 누르면 회원가입 페이지(SignUpActivity) 로 이동

```
public class LoginActivity extends AppCompatActivity {
   private FirebaseAuth mAuth; // 파이어베이스
   private static final String TAG = "SignUpActivity";
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_login);
       mAuth = FirebaseAuth.getInstance();
       findViewById(R.id.imageView).setOnClickListener(onClickListener); // 로그인 버튼
       TextView textView = findViewById(R.id.signGo);
       textView.setOnClickListener(new View.OnClickListener() { //회원가입 버튼 클릭시 회원가입 페이지로 이동
           @Override
           public void onClick(View v) {
               Intent intent = new Intent(getApplicationContext(), SignUpActivity.class);
               startActivity(intent);
       });
```

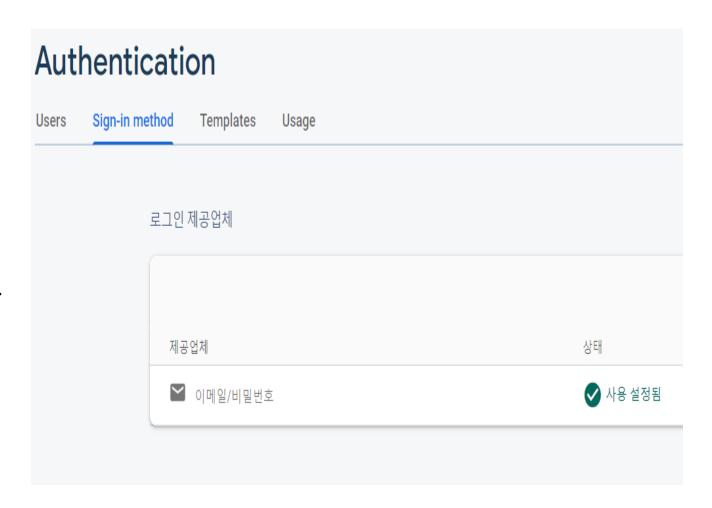
LoginActivity(로그인 화면)

입력된 이메일 주소와 비밀번호를 signInWithEmailAndPassword에 전달 후 이메일 주소와 비밀번호가 맞는지 확인

```
private void signUp(){ // 로그인 진행
   String email = ((EditText)findViewById(R.id.emailEditText)).getText().toString();
   String password = ((EditText)findViewById(R.id.passwordEditText)).getText().toString();
   if(email.length()>0 && password.length()>0){
       mAuth.signInWithEmailAndPassword(email, password)
               .addOnCompleteListener( activity: this, new OnCompleteListener<AuthResult>() {
                   @Override
                   public void onComplete(@NonNull Task<AuthResult> task) {
                       if (task.isSuccessful()) {
                           FirebaseUser user = mAuth.getCurrentUser();
                           startToast( msg: "로그인에 성공하였습니다.");
                           Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                           intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                           startActivity(intent);
                       } else {
                          Toast.makeText(getApplicationContext(), text: "입력된 정보가 옳바르지 않습니다.",Toast.LENGTH_SHORT).show();
    }else {
       startToast( msg: "이메일 또는 비밀번호를 입력해 주세요.");
```

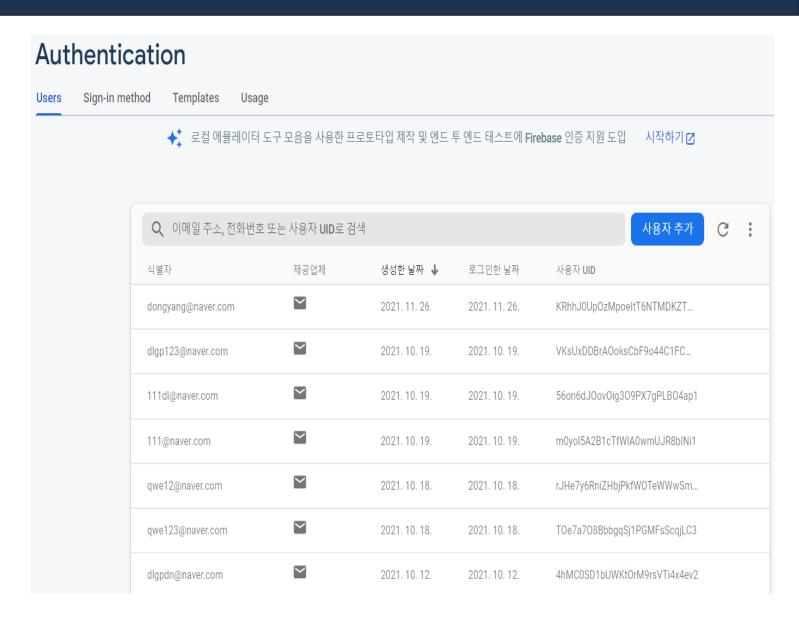
Part 3, Authentication(인증)

Authentication은 회원가입과 로그인을 위한 인증과 관련된 서비스이다. 여러가지 기능중 Email/Password 기반의 회원가입 방식을 사용했다.



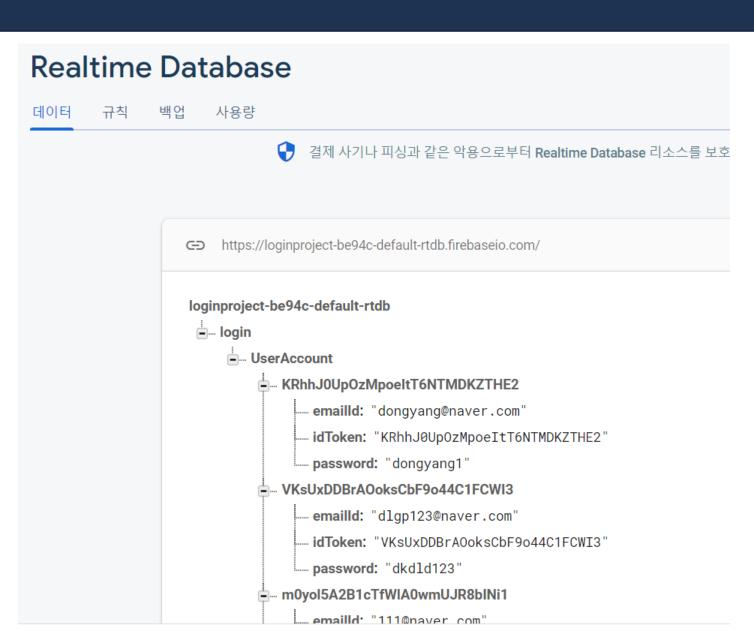
Authentication(인증)

Firebase Auth 로 회원가입이 되면, Authentication 에서 가입한 회원 목 록을 확인 할 수 있다.



Realtime Database

회원가입 된 이메일과 고유id와 비밀번호 를 데이터베이스에 저장



Part 4, 中기



Part 4, 中フ

1년 동안 졸업작품을 하면서 아쉬운 마음이 가장 컸다. 처음 졸업작품이라는 얘기를 들었을 때, 아무것도 모르는 내가 과연 저걸 할 수 있을까? 라는 걱정과 불안함이 가장 컸다. 하지만, 1학기 때 열심히 공부하여 성적장학금을 받아 좋은 작품을 만들 수 있을 것 이라는 자신감이 들었지만, 생각 했던 것 보다 어려웠고 뜻대로 안되는게 많았다. 능동적으로 할 수 있었음에도 수동적인 자세로 임했 던 것 같다. 능동적인 자세를 취하고 어렵다고 포기하지 않는 마음을 임했더라면 좀 더 완성된 작품을 만들 수 있을 생각에 아쉬운 마음이 크다. 다음에 또 작품을 만들 기회가 있다면 포기 하지 않고 좀 더 능동적인 자세를 취해 끝까지 최선을 다해 임할 것이다. 졸업작품이라는 과목을 통해 면접에 대해 자세하게 알게 되었고, git의 기초를 탄탄히 다진 것 같아 뿌듯하다. 앞으로도 하는 모든 일에 대해 수동적인 자세가 아닌 능동적인 자세로 임할 것이다.

감사합니다.