



SKKU-DM

Service for the double majors in SKKU

성균관대학교 복수 전공 학생들을 위한 서비스

GROUP

MG세대 (Team G)

TEAM MEMBER

권우진 김재현 박지수 이채은

CONTENTS

1 Progress Update

2 Repository Setup

3 Backend Details

4 Frontend Details

5 Infra Details

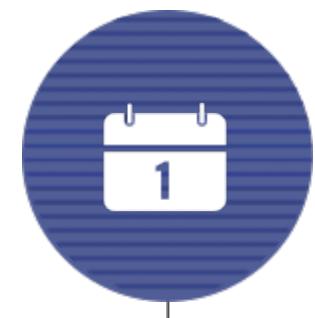
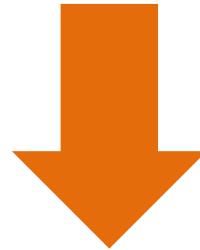
CHAPTER

01

Progress Update

Progress Update

...



WEEK 1 - 4

Project Topic Refinement
and Proposal Writing



WEEK 5 - 11

Application Development



WEEK 12

Testing



WEEK 13

Application Deployment and
Preparing Final Presentation

CHAPTER 01

Progress Update



WEEK 5 - 11

Application Development

Frontend (Chaeun & Woojin)

- Setup Flutter Projects
- Implement Some Pages

Backend (Jisoo & Jaehyeon)

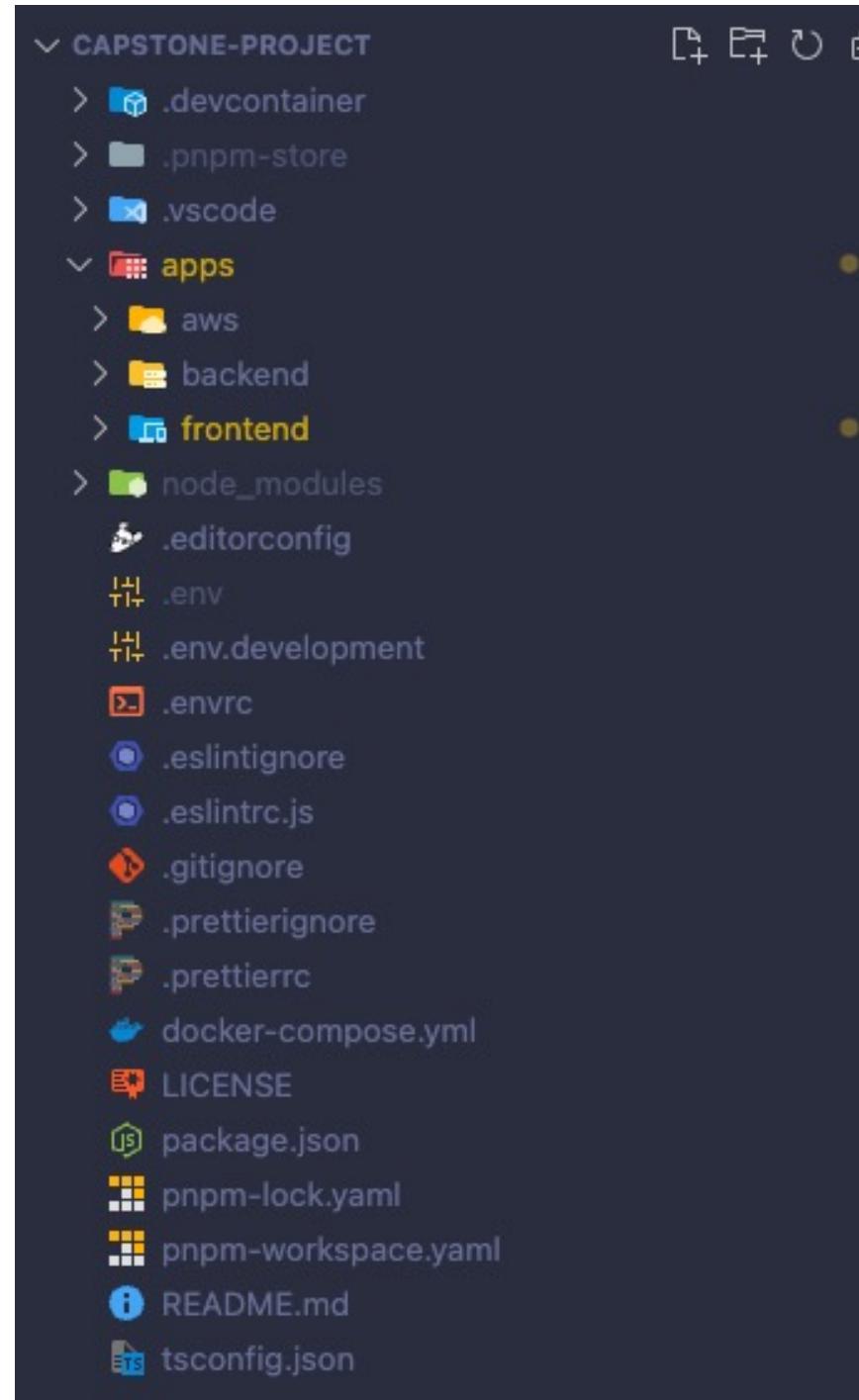
- Migrate Designed Schema to Prisma ORM
- Setup NestJS Projects
- Prepare Midterm Presentation

CHAPTER

02

Repository Setup

Monorepo



Apps/backend: NestJS

Apps/frontend: Flutter

Apps/aws: Terraform

CHAPTER
03

Backend
Detail

Configure Environments

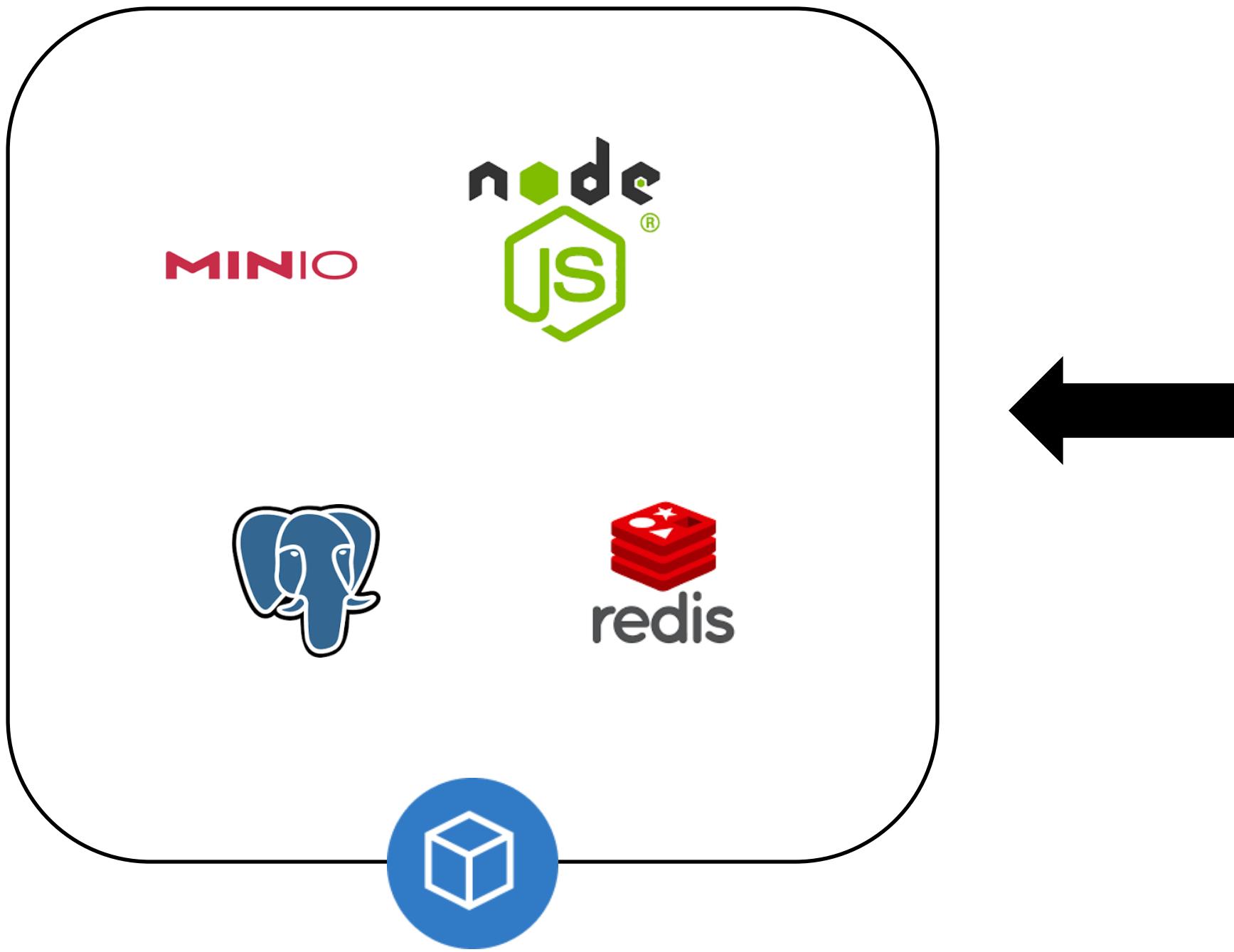
Challenges

It's difficult to install and match versions among team members for development environment setup including Node, databases, caches, object storage, etc.

...

CHAPTER
03
Backend
Detail

Devcontainers



Vscode devcontainers

```
...  
services:  
  app:  
    profiles: ['devcontainer']  
    container_name: skkudm-dev  
    build:  
      context: .  
      dockerfile: .devcontainer/Dockerfile  
    volumes:  
      - ./workspace:cached  
    depends_on:  
      - database  
    stdio: true  
    tty: true  
    env_file: .env.development  
    environment:  
      DEVCONTAINER: 1  
    network_mode: host  
  
  local-storage:  
    profiles: ['devcontainer']  
    container_name: skkudm-local-storage  
    image: minio/minio  
    ports:  
      - 9000:9000  
      - 9001:9001  
    command: server /data --console-address ":9001"  
    environment:  
      MINIO_ROOT_USER: ${MINIO_ROOT_USER}  
      MINIO_ROOT_PASSWORD: ${MINIO_ROOT_PASSWORD}  
    volumes:  
      - skkudm-storage:/data  
  
  database:  
    container_name: skkudm-database  
    image: postgres:16-alpine  
    ports:  
      - 5433:5432  
    environment:  
      POSTGRES_USER: postgres  
      POSTGRES_PASSWORD: 1234  
      POSTGRES_DB: skkudm  
    volumes:  
      - skkudm-database:/var/lib/postgresql/data  
...  
...
```

docker-compose.yaml

CHAPTER**03****Backend****Detail****Prisma**

Prisma



```
model User {  
    id      Int      @id @default(autoincrement())  
    email   String   @unique  
    name    String?  
    posts   Post[]  
}  
  
model Post {  
    id       Int      @id @default(autoincrement())  
    title    String  
    content  String?  
    published Boolean @default(false)  
    author   User     @relation(fields: [authorId], references: [id])  
    authorId Int  
}
```

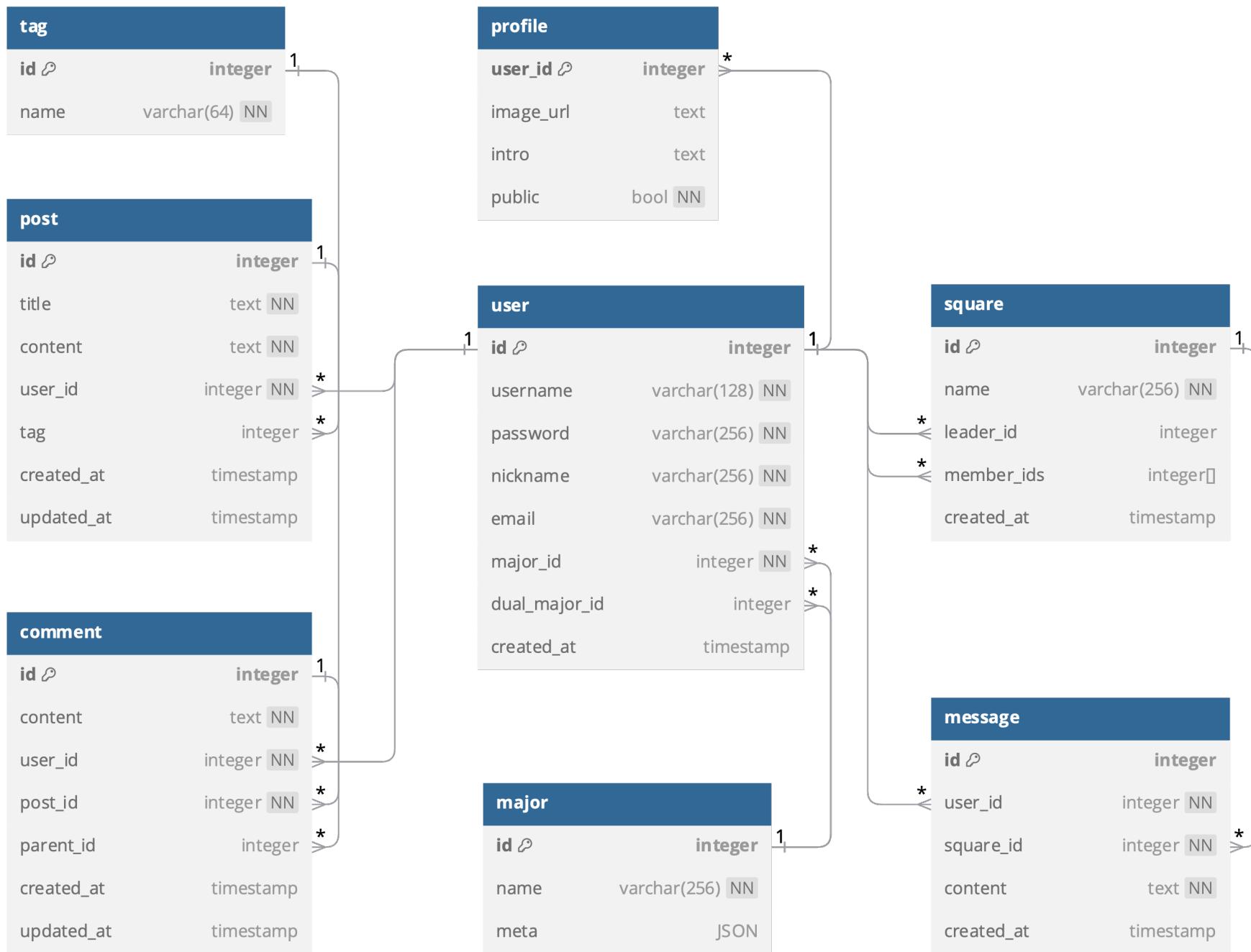


```
import { PrismaClient } from '@prisma/client'  
  
const prisma = new PrismaClient()  
  
async function main() {  
    const user = await prisma.user.create({  
        data: {  
            name: 'Alice',  
            email: 'alice@prisma.io',  
        },  
    })  
    console.log(user)  
}
```

CHAPTER

03

Backend Detail



Database Schema (Not Complete)

```

generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id      Int      @id @default(autoincrement())
  username String  @unique
  // @HideField()
  password String
  nickname String @unique
  email   String

  createdAt DateTime @default(now()) @map("created_at")
  updatedAt DateTime @updatedAt @map("updated_at")

  UserMajor UserMajor[]
  Profile   Profile[]
}

model Major {
  id      Int      @id @default(autoincrement())
  name    String  @unique
  meta   Json

  UserMajor UserMajor[]
}

model UserMajor {
  userId  Int
  majorId Int
  User    User @relation(fields: [userId], references: [id])
  Major   Major @relation(fields: [majorId], references: [id])

  createdAt DateTime @default(now()) @map("created_at")
  updatedAt DateTime @updatedAt @map("updated_at")

  @id([userId, majorId])
  @map("user_major")
}
...

```

Prisma Schema

CHAPTER 04

Frontend Detail



SKKU-DM
Connect with SKKU double majoring students

9:27

Register

ID SKKU Email Address
g.skku.edu
skku.edu

PW

q w e r t y u i ö p
a s d f g h j k l
z x c v b n m

9:27

Register

ORIGIN MAJOR
Multiple Major(s)
Student ID

q w e r t y u i ö p
a s d f g h j k l
z x c v b n m

9:27

Log in

jane@g.skku.edu

q w e r t y u i ö p
a s d f g h j k l
z x c v b n m

...

CHAPTER
04

Frontend
Detail

Bulletin Board

9:27 ■■■■■ 🔍

Bulletin Board

Filter #1 Filter #2 Filter #3 FILTERS

#Math #Classes WRITE

RESULTS

TITLE #1	Contents	1	5

TITLE #2	Contents	3	2

TITLE #3	Contents	17	13



CHAPTER

04

Frontend Detail

Square

DM SQUARE

9:27 9:27

원전공 ▾ 복수전공 ▾ 관심사 ▾

명륜이 ▶ 율전이 ▶

① 비공개
② 소프트웨어학과

산책 드셔 운동 음악

복전러 ▶ 외로운 늙대 ▶

① 경제학과
② 소프트웨어학과

독서 운동 음악 산책

인기 모집글 더 보기 >

운영체제 같이 공부하실 숏트 복전.. (3 / 4)

이번 2학기에 운영체제 같이 공부하실 소프트웨어학과 복전생 있으신가요? 화요일이나 목요일 오후 7시부터 같이 공부하고 싶습니다!! 프로필...

원전공 ▾ 복수전공 ▾ 관심사 ▾

24년 1학기 소프트웨어 복수전공... (1 / 4)

안녕하세요, 1학기에 소프트웨어학과 복수전공 합격하게 된 사람입니다! 똑같이 이번에 합격하신 분들 중에 함께 복전 시작하실 분 구합니다!! 제 프로필 확인해보고 연락주세요(:)

24년 1학기 경영학과 복수전공 합.. (2 / 6)

이번 경영학과 복수전공 합격하게 된 사람입니다! 마케팅 관련된 진로에 관심이 있고 독서나 영화 감상 등이 취미입니다!! 자세한 사항은 제 프로필 확인해보고 연락주세요(:)

운영체제 같이 공부하실 숏트 복전.. (3 / 4)

게시판 광장 프로필

CHAPTER

04

Frontend Detail

Profile

9:27

● ● ●

PROFILE

내 프로필



명륜이
① 비공개
② 소프트웨어학과
산책 독서

졸업 요건

원전공	경제학과
복수전공	소프트웨어학과

계정 MYEONG@G.SKKU.EDU

앱 설정

- [다크모드](#)
- [알림 설정](#)
- [암호/지문 잡금](#)

[!\[\]\(bc5430ca513ddf443e2314911d11fdd1_img.jpg\) 게시판](#)
[!\[\]\(1b43a7b7c06611b40ba0f7b5329c86ff_img.jpg\) 광장](#)
[!\[\]\(78030b36076754627eb0c254b9f03113_img.jpg\) 프로필](#)

9:27

● ● ●

PROFILE

내 프로필



명륜이
① 비공개
② 소프트웨어학과
산책 독서

졸업 요건

복수전공	소프트웨어학과
------	---------

19학번 ▾

● 오류 신고

전공핵심	전공일반	실험실습
27학점	16학점	8학점

바로가기

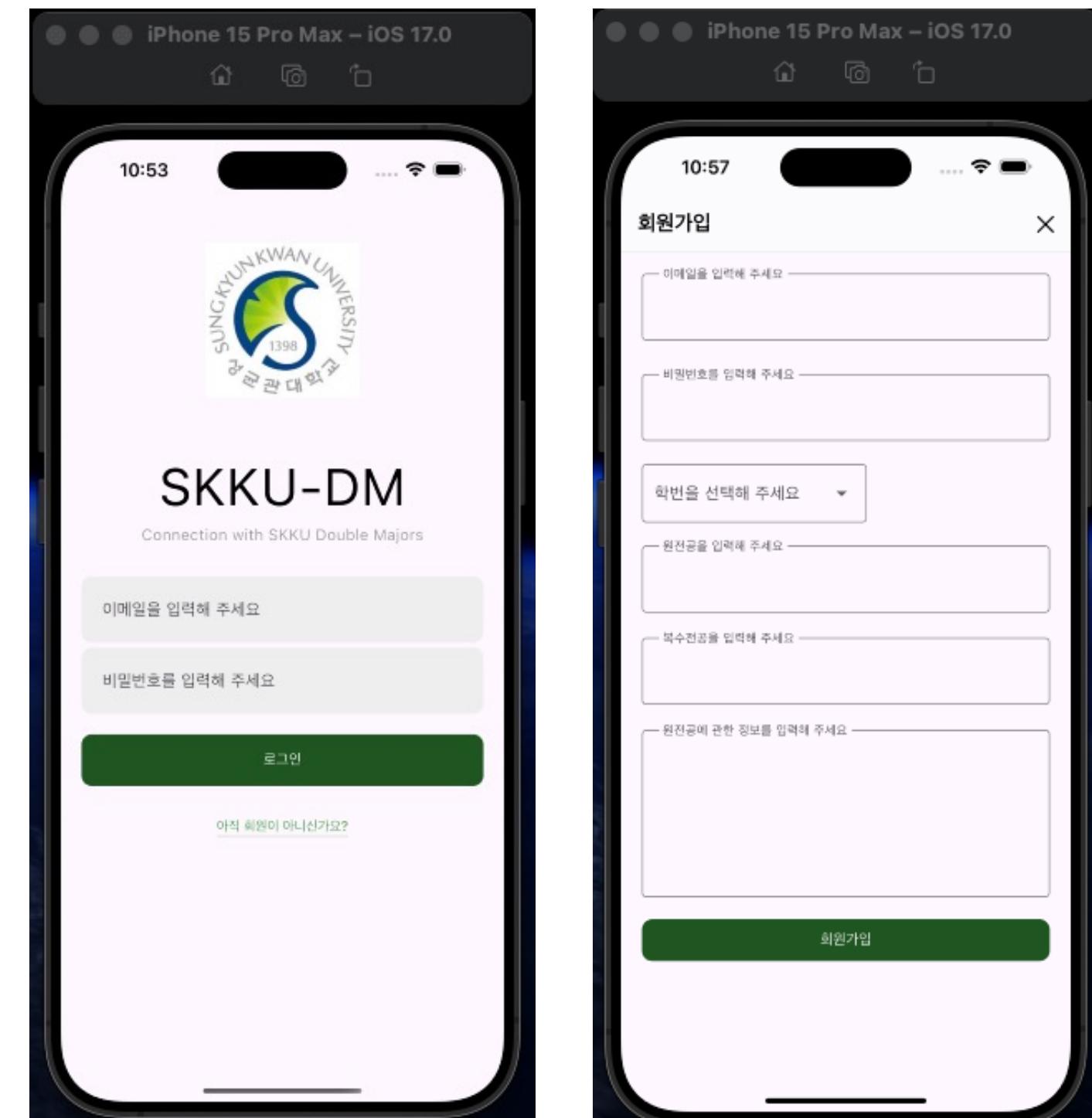
- [▶ 소프트웨어학과 교육과정](#)
- [▶ 소프트웨어학과 로드맵](#)
- [▶ 소프트웨어학과 졸업요건](#)

[!\[\]\(9a1b2c75975a424b8a1a69c3ce123520_img.jpg\) 게시판](#)
[!\[\]\(3c78dad572f04f06c8f38cf57be10b36_img.jpg\) 광장](#)
[!\[\]\(711a8baa3c2045cd2ee294425c565046_img.jpg\) 프로필](#)

CHAPTER 04

Frontend Detail

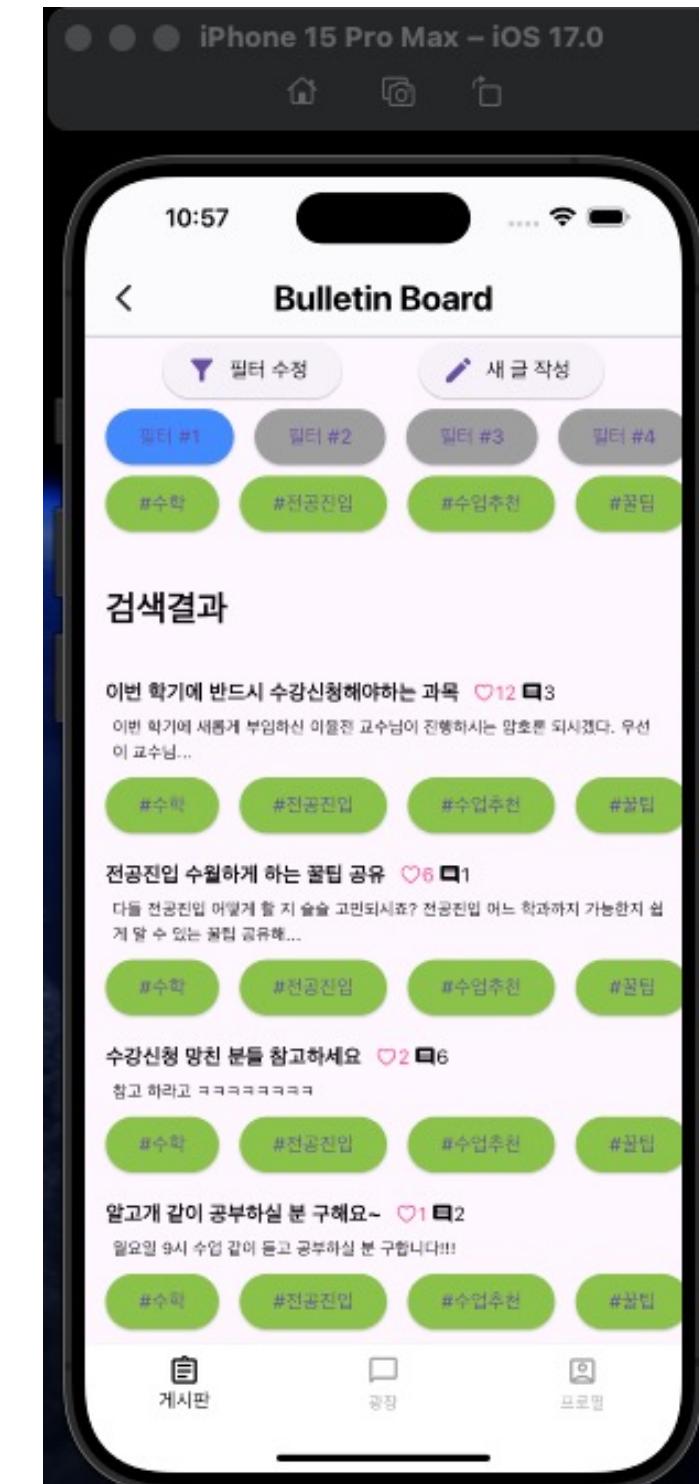
Sign up & Sign in



CHAPTER 04

Frontend Detail

Bulletin Board



CHAPTER
05

Infra Detail

Basic Setup

Challenges

When running a Flutter project through an emulator, if you also need to run a devcontainer for backend connectivity, it consumes too much computing power.

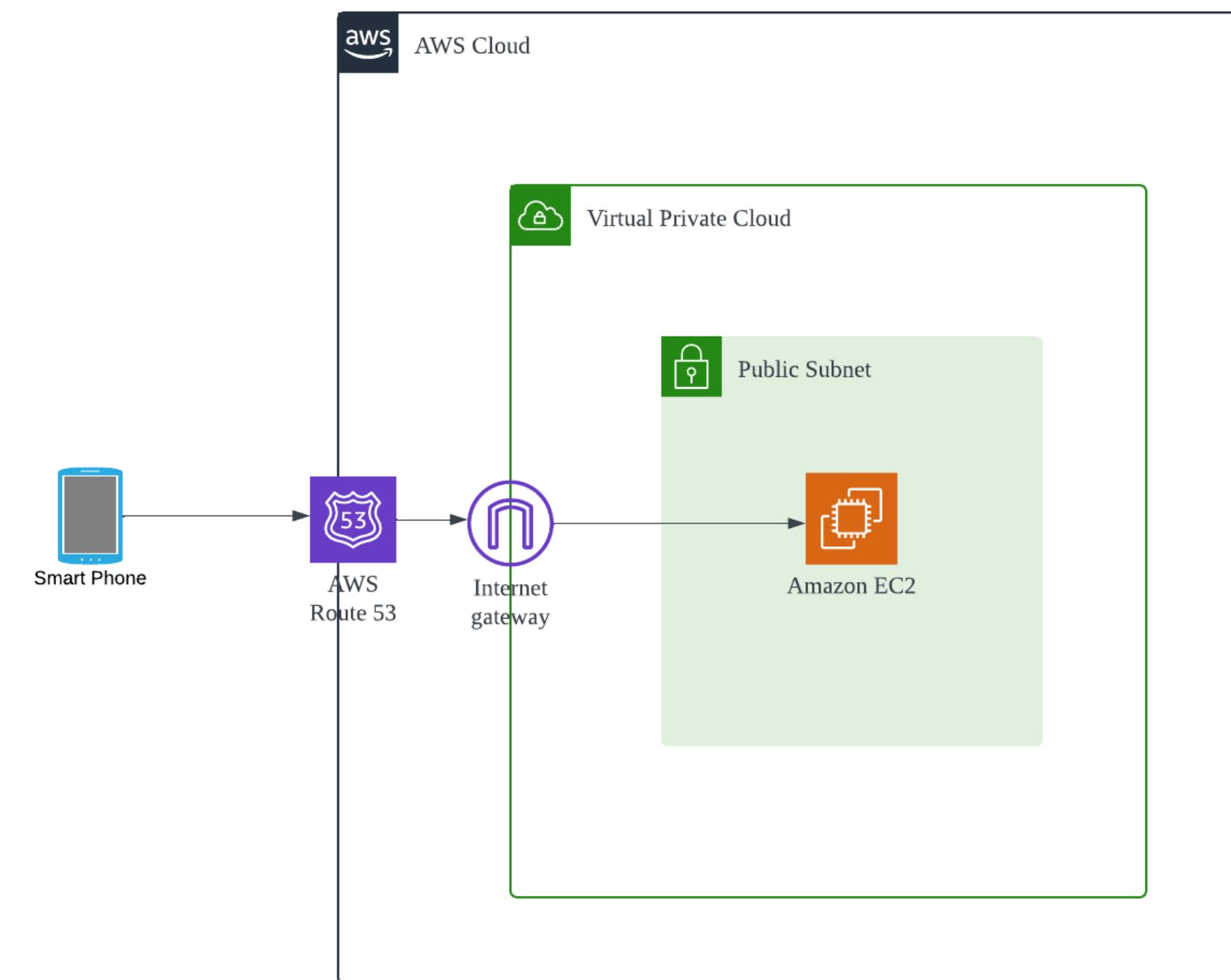
Solution

Reduce local computing power requirements by using a temporarily deployed backend API server during Flutter development

CHAPTER
05

Basic Setup

Infra Detail

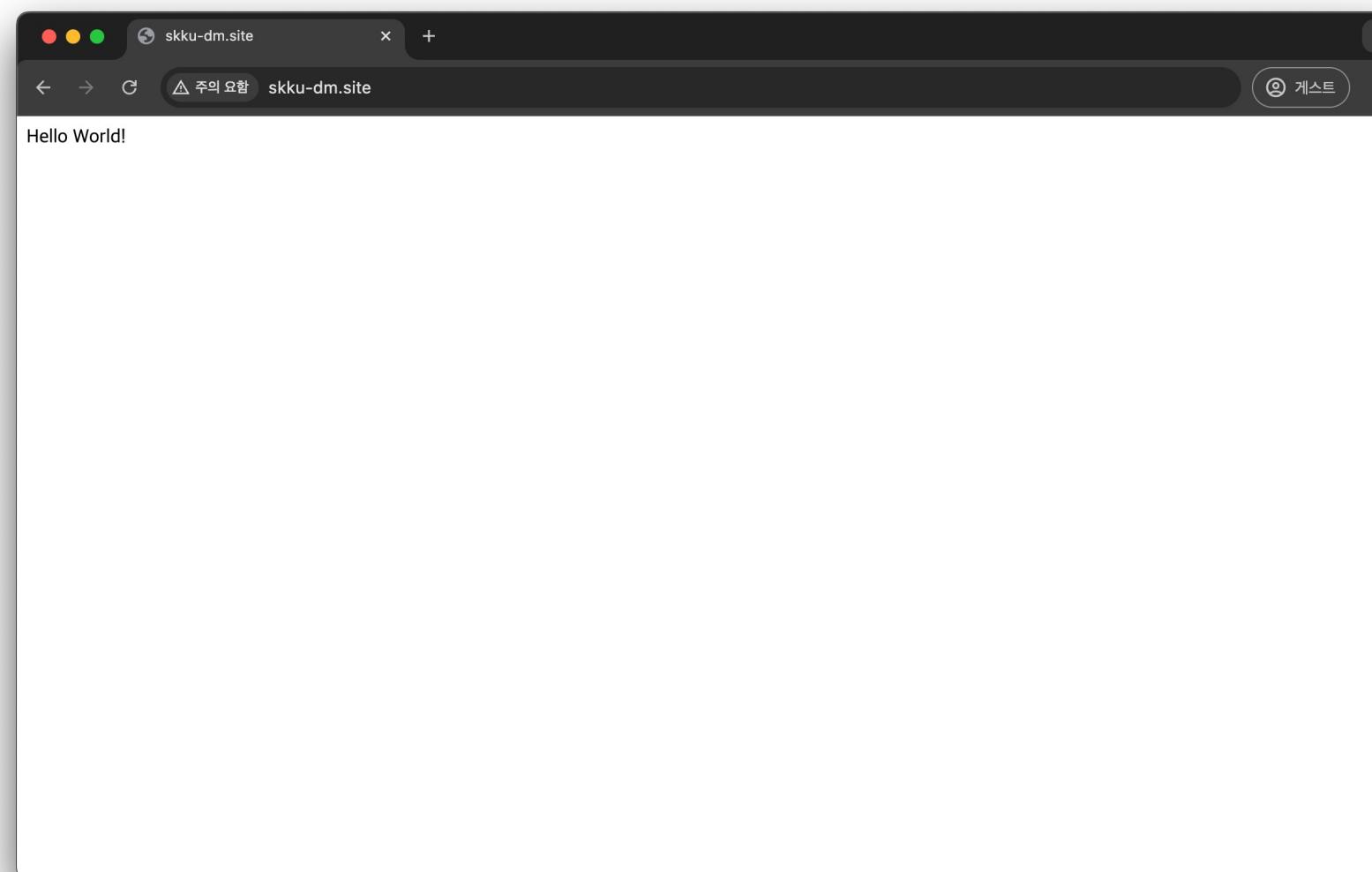


CHAPTER
05

Infra Detail

Basic Setup

<http://skku-dm.site>



THANK YOU!