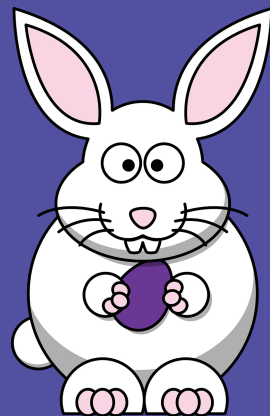


/*elice*/

엘리스 개인 프로젝트

레이서 포트폴리오



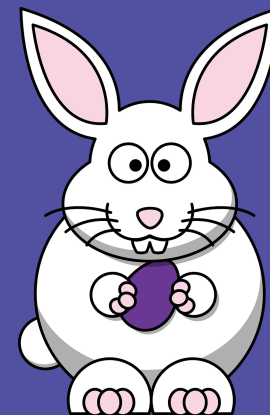
정희재 레이서

목차

- 01. 프로젝트 수행 과정
- 02. 라이브 데모
- 03. 프로젝트를 진행하며 배운 점
- 04. 어려웠던 점
- 05. 나만의 팁
- 06. Q&A

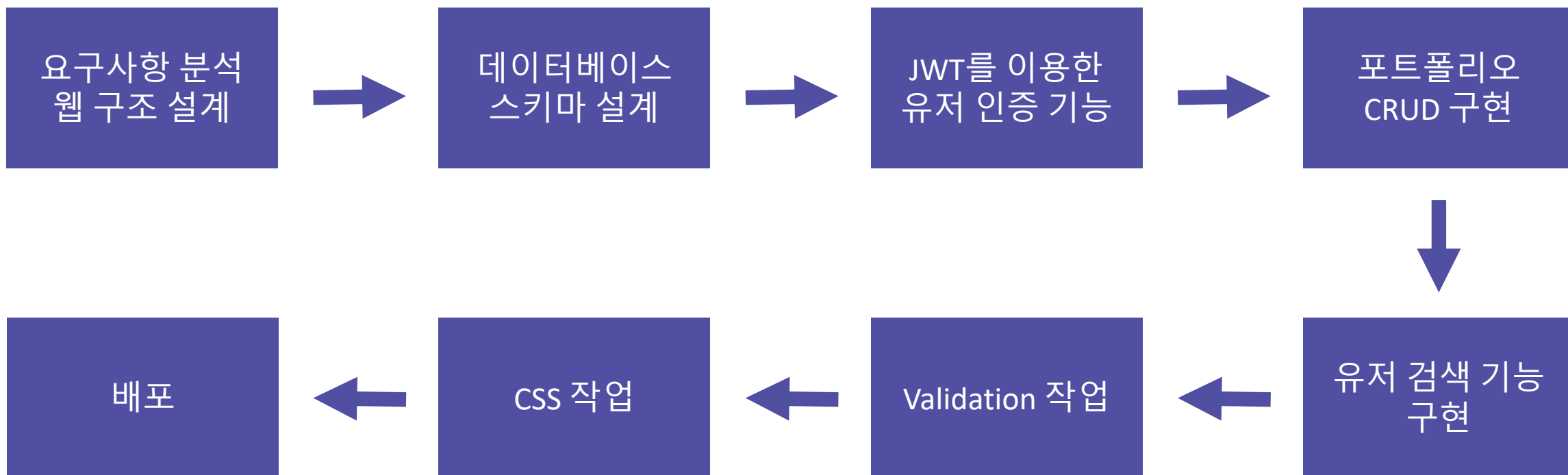
01

프로젝트 수행 과정



`/*elice*/`

✓ 프로젝트 수행 과정



✓ 프로젝트 관리

문서 관리

스크럼&회고

Week 1

- 8월 16일 월요일
- 8월 17일 화요일
- 8월 18일 수요일
- 8월 19일 목요일
- 8월 20일 금요일
- 8월 21일 토요일

Week 2

- 8월 23일 월요일
- 8월 24일 화요일
- 8월 25일 수요일
- 8월 26일 목요일
- 8월 27일 금요일

프로젝트 문서

- 기술 스택
- DB Diagram
- API

개발 노트

진짜 노트에 그린 그림...

- 컴포넌트 구조
- 인증 절차
- JWT

To do List 관리

로그인

필수

- ✓ 유저로부터 아이디(이메일)와 비밀번호 정보를 입력받아 로그인 합니다.
- ✓ 아이디와 비밀번호는 필수 입력 사항 입니다.
- ✓ 로그인한 유저에 대해 session(Token)으로 관리해야 합니다.

선택

- ✓ 비밀번호는 다음의 링크1, 링크2에 맞추어 최소 8자리 이상의 길이로 입력 받아야 합니다.
- ✓ 아이디는 이메일 형식으로만 입력 받아야 합니다.

회원가입

필수

- ✓ 유저로부터 아이디(이메일), 비밀번호, 이름 정보를 입력받아 회원가입합니다.
- ✓ 비밀번호와 비밀번호 확인의 값이 일치해야 합니다.

선택

- ✓ 아이디는 이메일 형식으로만 정보를 입력 받아야 합니다.
- ✓ 이름은 한글, 영문으로만 입력 받아야 합니다.
- ✓ 비밀번호는 다음의 링크1, 링크2에 맞추어 영문, 숫자, 특수문자 중 2종류 이상을 조합하여 최소 10자리 이상 또는 3종류 이상을 조합하여 최소 8자리 이상의 길이로 구성 합니다.

로그아웃

필수

- ✓ 현재 로그인한 유저에 대해 로그아웃 합니다.
- ✓ 로그아웃한 유저를 현재 session에서 제거해야 합니다.

내 포트폴리오 보기, 수정, 업로드, 삭제

필수

- ✓ 학력: 학교이름, 전공 정보를 text 형식으로 입력받습니다. 학위에 대한 사항은 radio button을 통해 입력받습니다.
- ✓ 수상 이력: 수상 내역과 상세내역을 text 형식으로 입력받습니다.
- ✓ 프로젝트: 프로젝트 이름과 상세내역을 txt 형식으로 입력받습니다. 프로젝트 수행기간은 datepicker를 활용해 날짜(년월일)를 입력받습니다.
- ✓ 자격증: 자격증 이름, 발급기관을 text 형식으로 입력받습니다. 취득일자는 datepicker를 활용해 날짜(년월일)를 입력받습니다.

디렉토리 분리

Name	Last commit
Client	서버 주소 변경
Server	API 경로 수정
.gitignore	Update .gitignore
README.md	Update README.md

✓ 개발 툴

VS Code



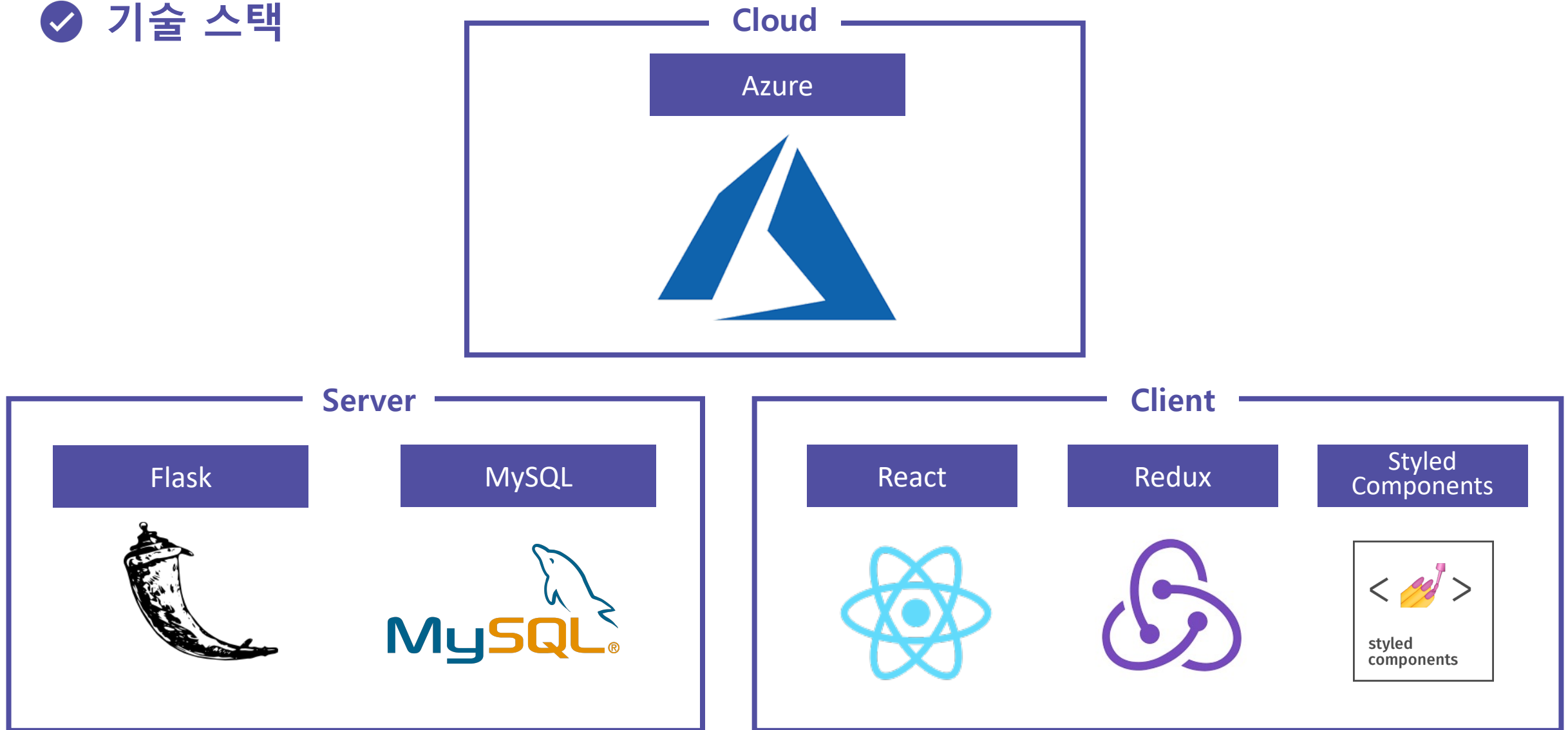
POSTMAN



Whale

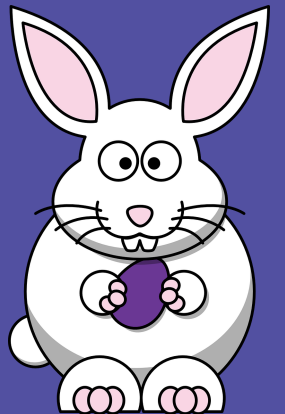


✓ 기술 스택



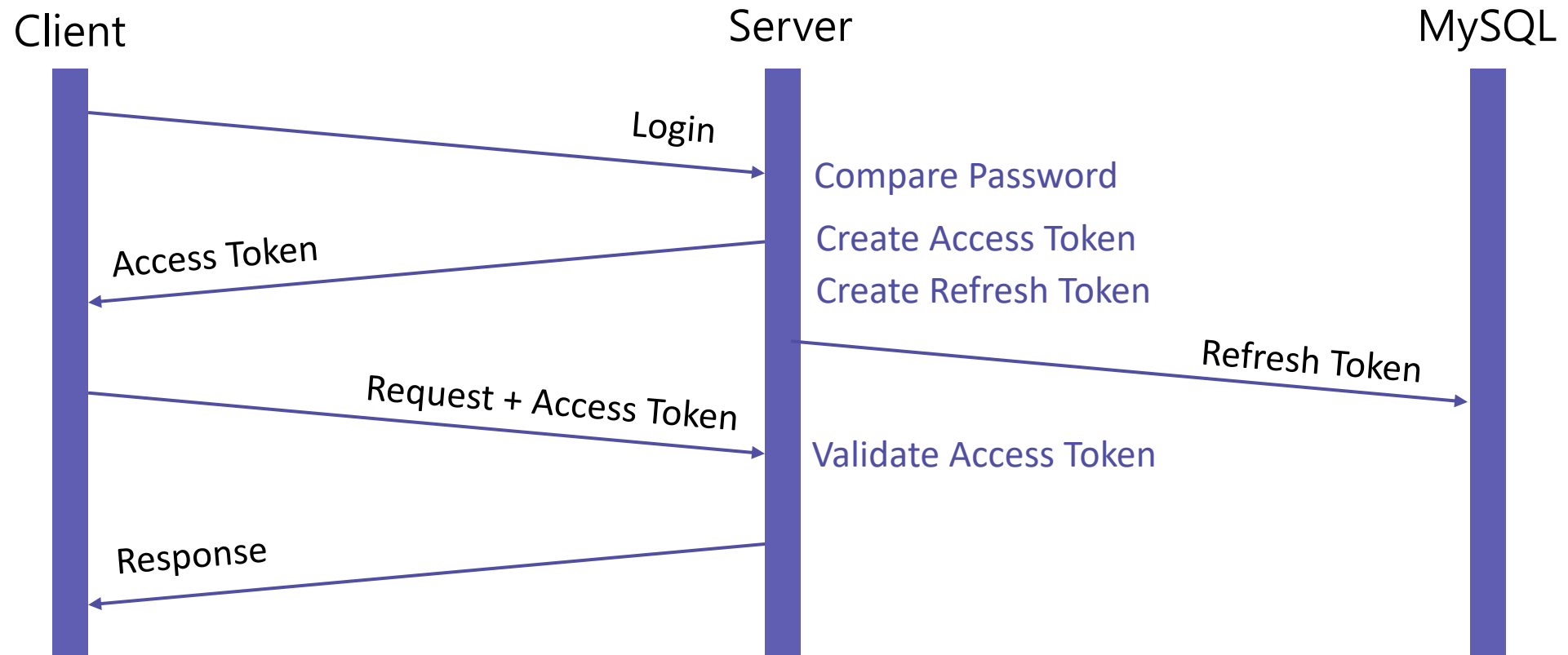
02

라이브 데모

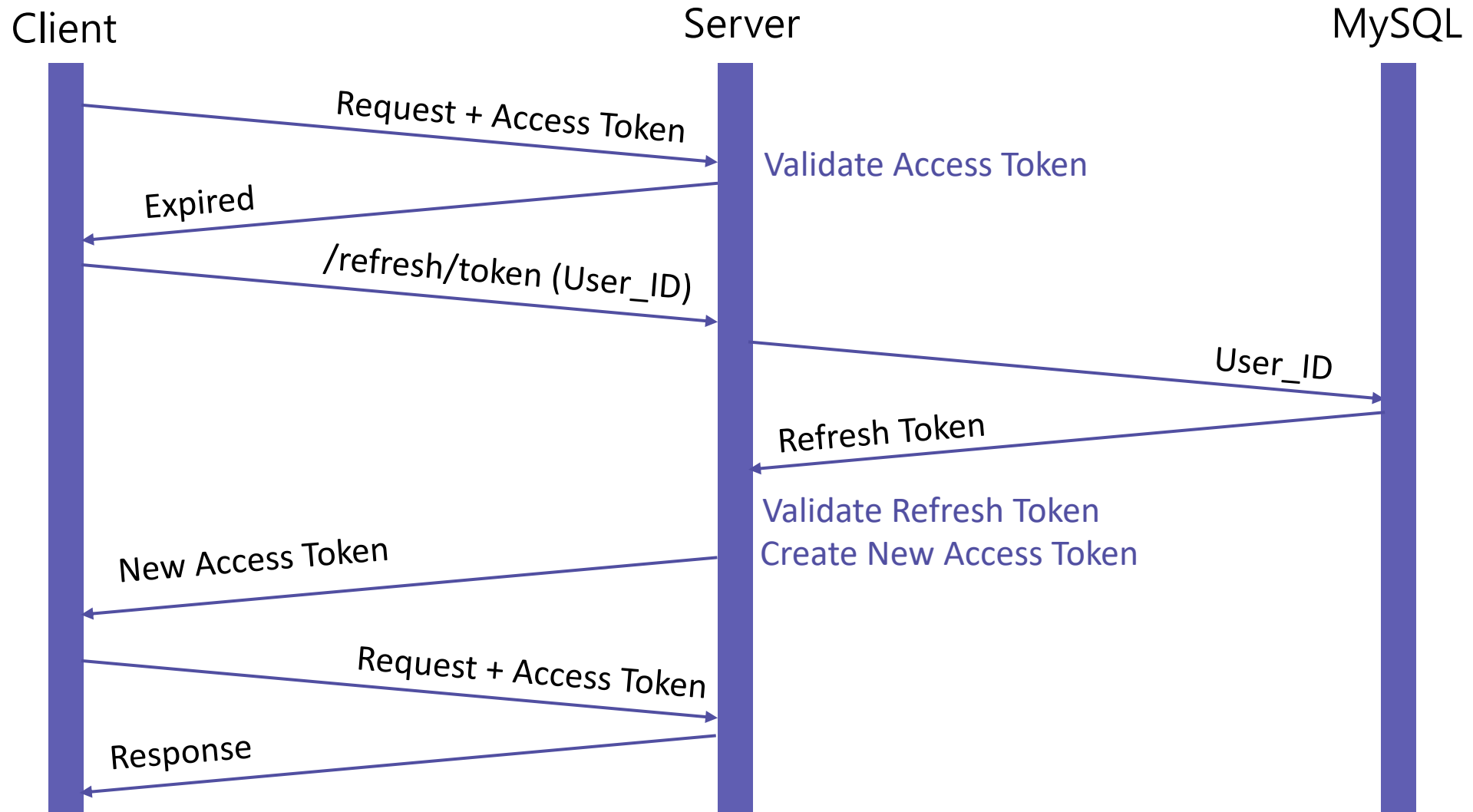


`/*elice*/`

✓ JWT (Access Token)



✓ JWT (Refresh Token)



✓ CRUD

학력

☐ 재학 ☒ 졸업 ☐ 졸업예정 ☐ 중퇴

☒ 재학 ☐ 졸업 ☐ 졸업예정 ☐ 중퇴

기존에 있던 학력

UPDATE

새로 추가한 학력

INSERT

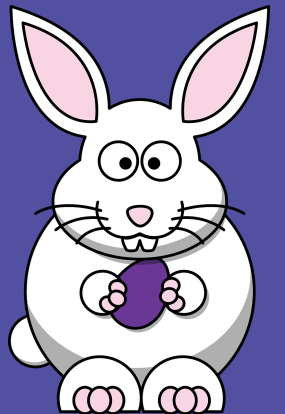
클릭시 한번에 저장

Data

```
[
  { id : 1,
    name : 한양대학교 ERICA캠퍼스,
    major : 소프트웨어학부,
    edu_type : 졸업
  },
  { id : -1,
    name: 한양대학원,
    major : AI전공,
    edu_type : 재학
  }
]
```

03

프로젝트를 진행하며 배운 점



`/*elice*/`

✓ 설계를 잘 하자 (코드의 중복)

학력 수정 API

학력을 수정하는 코드

Access Token 을 재발급하는 코드

수상이력 수정 API

수상이력을 수정하는 코드

Access Token 을 재발급하는 코드

프로젝트 수정 API

프로젝트를 수정하는 코드

Access Token 을 재발급하는 코드

자격증 수정 API

자격증을 수정하는 코드

Access Token 을 재발급하는 코드

프로필 수정 API

프로필을 수정하는 코드

Access Token 을 재발급하는 코드

✓ 가상 환경을 사용하자

로컬 환경

Flask
Flask-Bcrypt
Flask-SQLAlchemy
Flask-Migrate
Jinja2
Jupyter
Matplotlib
Numpy
Scikit-learn
Scipy
seaborn
Pymongo
PyMySQL

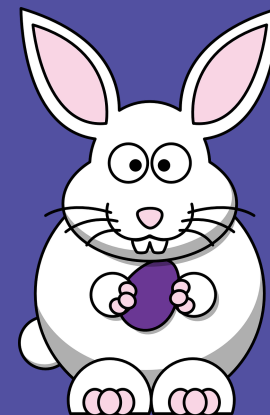
....

배포 환경



04

어려웠던 점



`/*elice*/`

✓ Refresh Token을 어떻게 관리할까?

브라우저
스토리지에
저장하시면 됩니다.



쿠키에 저장해서
사용 해야합니다.



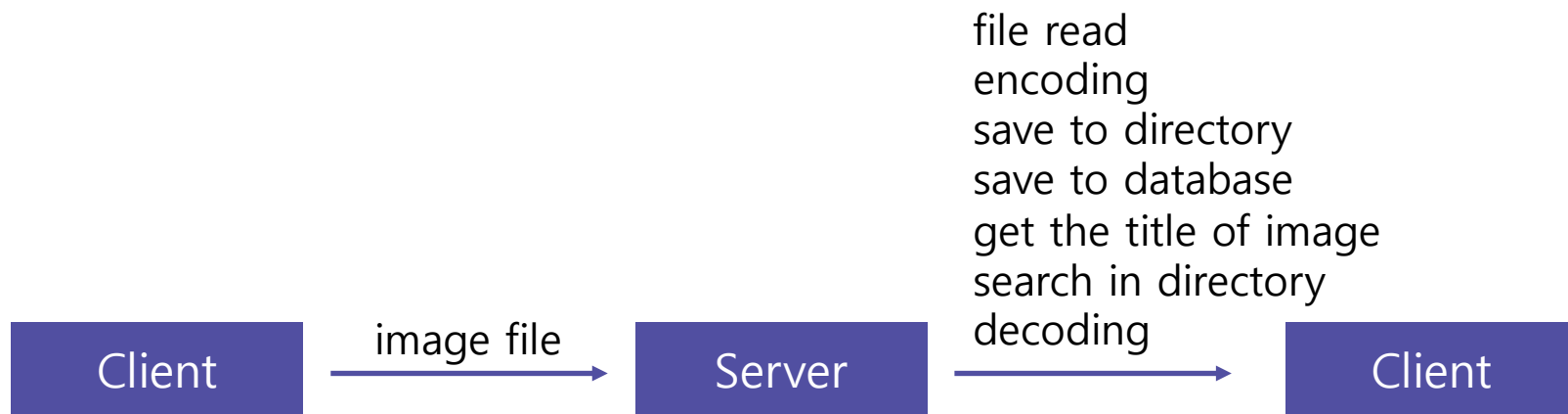
데이터베이스에
저장하고 유저는
인덱스 값을 들고
있으면 됩니다.



앵 ~ 그렇게 하는거
아닌데 ~

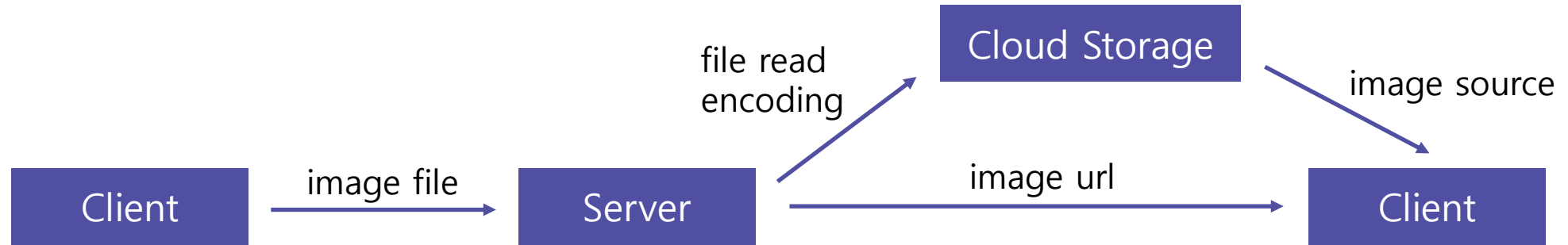


✓ 프로필 이미지 구현



프로필 하나 로딩 하는데 1초
어떻게 방법이 없을까 ??

✓ 프로필 이미지 구현



클라우드 스토리지로
문제 해결 ?!

✓ 프로필 이미지 구현

RuntimeError: Starting with v0.37.0, the 'azure-storage' meta-package is deprecated and cannot be installed anymore. Please install the service specific packages prefixed by `azure` needed for your application.

나한테 왜 그래...

✓ 프로필 이미지 구현

Blob

Blob 업로드

Blob 다운로드

Blob 삭제

Blob 삭제 취소

Blob 속성 가져오기

여러 Blob 삭제

URL에서 Blob 복사

URL에서 Blob 복사 중단

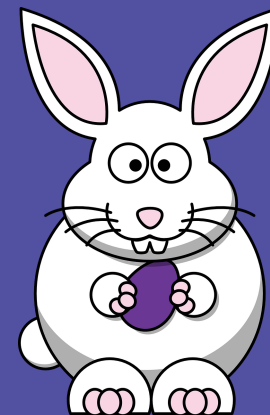
Blob에서 임대 가져오기

```
def stream_block_blob(self):  
  
    import uuid  
    # Instantiate a new BlobServiceClient using a connection string - set chunk size to 1MB  
    from azure.storage.blob import BlobServiceClient, BlobBlock  
    blob_service_client = BlobServiceClient.from_connection_string(self.connection_string,  
                                                                    max_single_get_size=1024*1024,  
                                                                    max_chunk_get_size=1024*1024)  
  
    # Instantiate a new ContainerClient  
    container_client = blob_service_client.get_container_client("containersync")  
    # Generate 4MB of data  
    data = b'a'*4*1024*1024  
  
    try:  
        # Create new Container in the service  
        container_client.create_container()  
  
        # Instantiate a new source blob client  
        source_blob_client = container_client.get_blob_client("source_blob")  
        # Upload content to block blob  
        source_blob_client.upload_blob(data, blob_type="BlockBlob")  
  
        destination_blob_client = container_client.get_blob_client("destination_blob")  
        # [START download_a_blob_in_chunk]  
        # This returns a StorageStreamDownloader.  
        stream = source_blob_client.download_blob()  
        block_list = []
```

사랑합니다
마이크로소프트

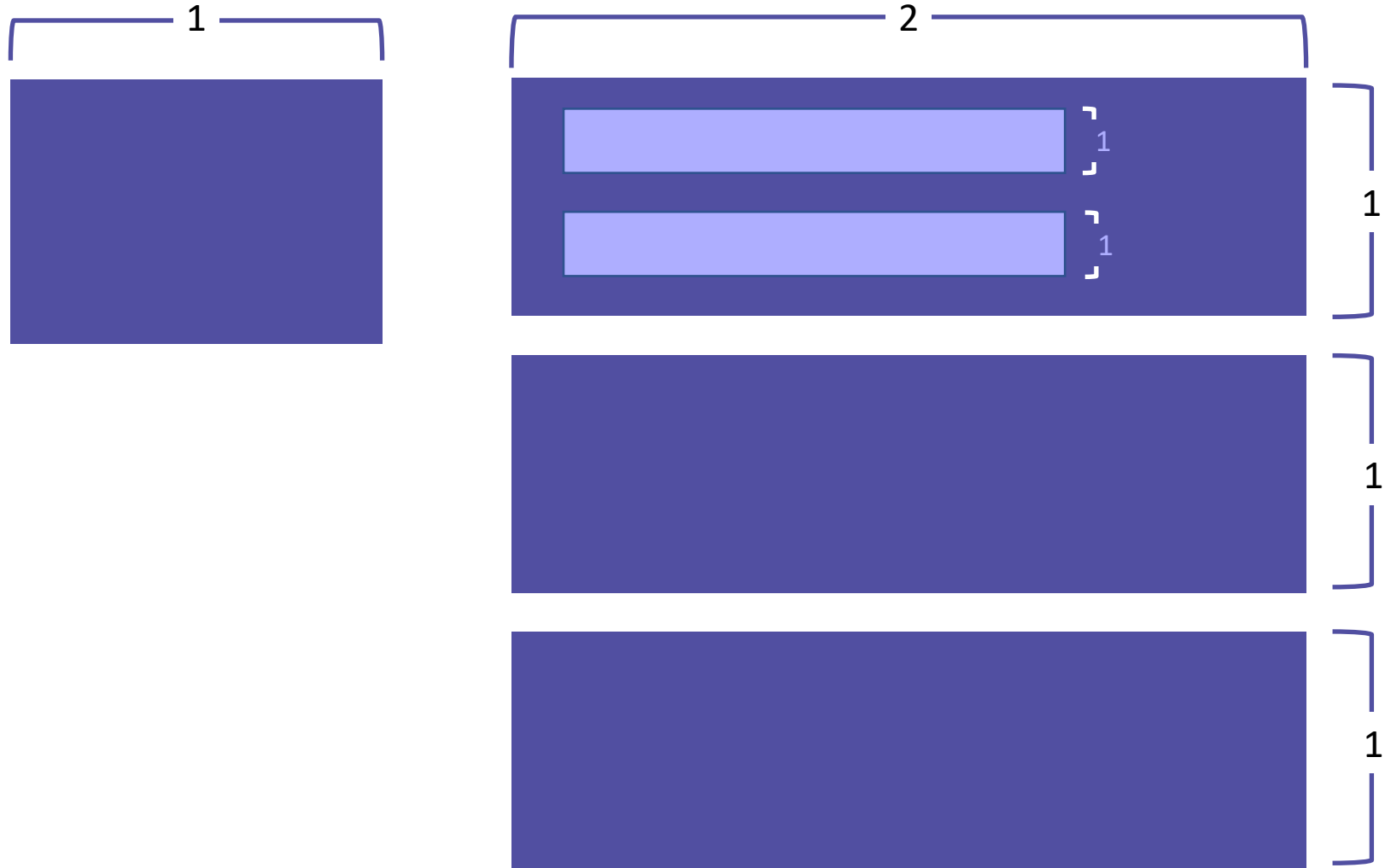
05

나만의 팁



`/*elice*/`

✓ 레이아웃은 Grid가 최고

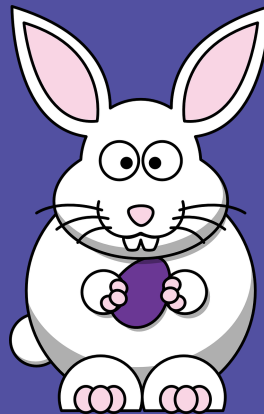


✓ 새로 고침 했을 때 404 에러 해결

/etc/nginx/sites-available/default

```
location / {  
    # First attempt to serve request as file, then  
    # as directory, then fall back to displaying a 404.  
    try_files $uri $uri/ /index.html =404;  
}
```

Q & A



`/*elice*/`