

## 개발 콘텐츠 숏영상 공유 플랫폼 - 숏데브

### 프로젝트 소개 (제안배경 및 주요내용)

· 개발자들은 개발 관련 정보를 영상이나 유튜브를 통해서 정보를 많이 얻고 있다. 최근 full 영상 보다는 15초 내의 틱톡 및 short 컷 영상이 인기가 많다. 이에 따라 개발 관련 내용으로만 구성된 15초 내의 short컷 영상을 제공하는 서비스를 만들어본다. 개발 관련 최신 트렌드나 개발 기술에 대해 짧은 영상을 통해 빠르고, 신속하게 정보를 얻을 수 있는 플랫폼이 된다. 빠른 정보를 얻고 싶거나, 개발관련 흥미로운 영상을 접하고 싶다면 숏데브 플랫폼을 많이 이용할 것이다. 많이 이용하게 되면 개발 관련 교육, 도서, 등 사용자들을 위한 광고도 연재할 수 있는 플랫폼으로 수익을 낼 수 있다

### 활용장비 · AWS 서버 이용

· 기획안 리뷰 , 코드 리뷰, 기술 stack 논의

-front-end : 어드민 - react , App - react-native

### React-Admin

React 기반의 관리자 페이지 제공 서비스이다.

### React-native

페이스북이 개발한 오픈 소스 모바일 애플리케이션 프레임워크이다.

## -Bactk-end : java, Springboot, redis, jwt를 이용한 백엔드 서버 구현

### Redis

Redis는 오픈 소스로서 NoSQL로 분류되기도 하고, Memcached와 같이 인 메모리 솔루션으로 분류되기도 한다. 성능은 Memcached에 버금가면서 다양한 데이터 구조체를 지원함으로써 DB, Cache, Message Queue, Shared Memory 용도로 사용될 수 있다.

한편, Redis는 Remote Dictionary Server의 약자로 외부에서 사용 가능한 Key-Value 쌍의 해시 맵 형태의 서버라고 생각할 수 있다. 그래서 별도의 쿼리 없이 Key를 통해 빠르게 결과를 가져올 수 있다.

또한, 디스크에 데이터를 쓰는 구조가 아니라 메모리에서 데이터를 처리하기 때문에 작업 속도가 상당히 빠르다.

개인적으로 Redis를 한 줄로 정의하라고 하면, Redis는 고성능 키-값 저장소로서 String, list, hash, set, sorted set 등의 자료 구조를 지원하는 NoSQL 라고 할 것 같다.

### Redis의 특징

1. 영속성을 지원하는 인 메모리 데이터 저장소
2. 다양한 자료 구조를 지원함.
3. 싱글 스레드 방식으로 인해 연산을 원자적으로 수행이 가능함.
4. 읽기 성능 증대를 위한 서버 측 리플리케이션을 지원
5. 쓰기 성능 증대를 위한 클라이언트 측 샤딩 지원
6. 다양한 서비스에서 사용되며 검증된 기술

### JWT

Json Web Token 약자로 모바일이나 웹의 사용자 인증을 위해 사용하는 암호화된 토큰을 의미합니다. JWT 정보를 request에 담아 사용자 정보 열람, 수정 등 개인적인 작업 등을 수행할 수 있게한다.

### JWT는 어떤 정보를 담아서 전달하나

JWT는 세 파트로 나누어지고, 각 파트는 .(점)으로 구분하여 aaaaa.bbbbbb.cccccc 이런식으로 표현됩니다. JWT는 URL에서 파라미터로 사용할 수 있도록 URL\_Safe 한 Base64url 인코딩을 사용합니다.

## 기존의 인증 / 인가 방식과 다른점?

서버 기반 인증 기존의 인증 시스템에서는 서버측에서 유저들의 정보를 세션에 기억하고 있어야 합니다. 이 세션을 유지하기 위해서는 여러가지 방법이 사용되는데 메모리/디스크/데이터베이스 시스템에 이를 담곤 합니다.

토큰 기반 시스템은 stateless합니다. 유저의 인정 정보를 서버나 세션에 담아두지 않기 때문에 인정정보를 서버에 담아둠으로써 발생하는 많은 문제점들이 해소됩니다.

## JWT의 장점

▶무상태(stateless), 확장성이 있다. 기존 서버에 세션을 저장하는 방식에서 서버 여러대를 사용하여 요청을 분산하였다면 어떤 유저가 로그인했을 때 그 유저는 처음 로그인한 서버에만 요청을 내보내도록 설정해야합니다. 하지만 토큰을 사용하면 토큰 값만 알고 있다면 어떤 서버로 요청이 들어가든 상관이 없습니다. 즉, 세션스토리지 필요가 없다!

▶보안성 쿠키를 전달하지 않아도 되므로 쿠키를 사용함으로써 발생하는 취약점이 사라집니다.

▶여러 플랫폼 및 도메인 어플리케이션 규모가 커지면 여러 디바이스를 호환 시키고 더 많은 종류의 서비스를 제공합니다. 토큰을 사용한다면 그 어떤 디바이스에서도 그 어떤 도메인에서도 토큰만 유효하다면 요청이 정상적으로 처리 됩니다.

## JWT의 단점

▶길이 claim에 넣는 데이터가 많아질 수록 JWT토큰이 길어집니다. API호출 시 매 호출마다 토큰 데이터를 서버에 전달해야 하는데 길이가 길다는 것은 그만큼 네트워크 대역폭 낭비가 심할 수 있습니다.

▶보안 JWT는 기본적으로 Payload에 대한 정보를 암호화 하지 않습니다. 단순히 BASE64로 인코딩만 하기 때문에 중간에 패킷을 가로채거나 기타 방법으로 토큰을 취득했으면 디코딩을 통해 데이터를 볼 수 있습니다. 그래서 JWE(JSON Web Encryption)를 통해 암호화 하거나 중요데이터를 Payload에 넣지 말아야 합니다.

## -github : github를 통한 aws ci/cd 구성

### CI/CD란

CI(Continuous Integration)/CD(Continuous Delivery/Continuous Deployment)란, 어플리케이션 개발에 필요한 여러 단계에 대한 자동화를 통해 어플리케이션을 보다 빠르고 짧은 주기로 고객에게 제공하는 방법입니다. CI/CD의 개념은 지속적인 통합(Continuous Integration), 지속적인 서비스 제공(Continuous Delivery) 및 지속적인 배포(Continuous Deployment)를 통해 새로운 코드의 통합, 테스트, 릴리스, 배포 등의 어플리케이션 라이프사이클 전체에 대한 자동화 과정을 모니터링 가능하도록 하는 것을 말합니다.

### CI(Continuous Integration)

CI는 Continuous Integration의 약자로 지속적인 통합을 의미합니다. 개발자를 위한 자동화 프로세스를 통해 새로운 코드 개발과, 코드의 변경 사항이 정기적으로 빌드 및 테스트되고 공유 레포지토리에 병합되어 여러 명의 개발자가 동시에 어플리케이션 개발과 관련된 코드를 작업할 경우에도 서로 충돌없이 원하는 개발 작업을 수행하고 문제를 해결할 수 있습니다.

일반적인 경우 어플리케이션에 대해 수정/변경된 내용을 반영하는 경우 다른 개발자가 동시에 적용하는 변경 사항과 충돌할 가능성이 있습니다. 하지만 CI를 통해 자동화된 프로세스를 구현한다면, 개발자들이 코드 변경 사항을 메인 레포지토리를 통해 업로드를 수행하면 이를 통해 코드를 병합하는 작업을 더욱 수월하게 자주 수행할 수 있습니다. 개발자가 어플리케이션에 적용한 변경 사항이 병합되면 이러한 변경 사항이 다른 개발자가 수정한 내역이나 프로그램에 영향을 주지 않으며, 어플리케이션에 문제가 발생하지 않도록 병합 작업을 수행합니다. 그리고 각기 다른 레벨의 자동화 테스트(보통 단위테스트 및 통합 테스트) 실행을 통해 변경 사항이 어플리케이션에 제대로 적용되었는지를 확인합니다.

다시 말해, 클래스와 기능에서부터 전체 어플리케이션을 구성하는 서로 다른 모듈에 이르기까지 모든 것에 대한 테스트를 수행합니다. 자동화된 테스트에서 기존 코드와 신규 코드 간의 충돌이 발견되면 CI를 통해 이러한 버그를 더욱 빠르게 확인할 수 있고, 이렇게 확인된 소스를 수정하여 원활하게 배포할 수 있습니다.

## CD(Continuous Delivery/Continuous Deployment)

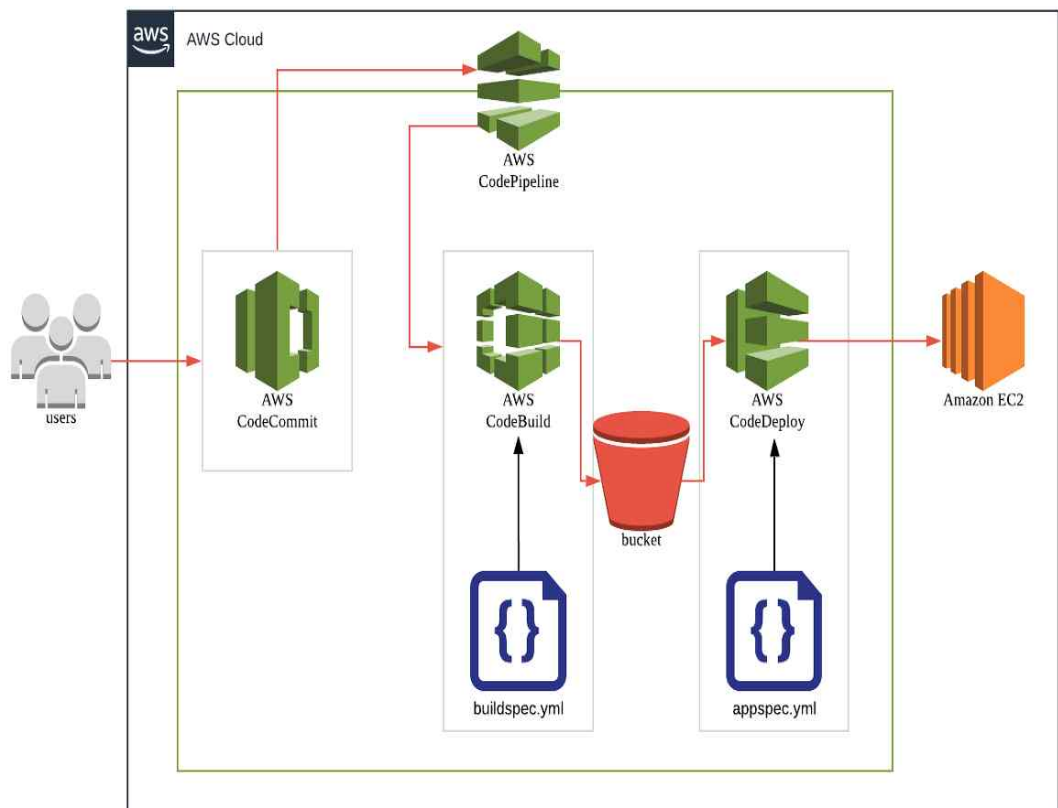
CD는 지속적인 전달(Continuous Delivery) 및 지속적인 배포(Continuous Deployment)의 의미를 가지고 있는데, 두가지 용어를 혼용하여 사용합니다. 개발자가 지속적인 서비스 전달과 배포를 통해 새로운 코드에 대한 배포를 자동화할 수 있으며, 이로써 신속한 어플리케이션을 제공할 수 있도록 도움을 주는 역할을 합니다.

CI 를 통한 빌드의 자동화 및 유닛의 통합 테스트 수행 이후 이어지는 CD 프로세스는 유효한 소스코드를 레포지토리에 자동으로 전달합니다. 따라서 효과적인 CD의 프로세스를 실현하기 위해서는 개발 파이프라인 CI가 먼저 구축되어야 합니다. 이러한 지속적 제공의 목표는 운영 환경으로 배포할 준비가 되어 있는 코드베이스를 확보하는 것에 있습니다.

이러한 CD 프로세스를 수행하면 보다 빠르고 손쉽게 어플리케이션을 운영 환경으로 배포할 수 있습니다.

지속적 배포인 Continuous Deployment 는 파이프라인의 마지막 단계로, 운영 환경으로 배포 준비가 완료된 빌드를 레포지토리에 자동으로 릴리스하는 Continuous Deployment는 어플리케이션을 운영 환경으로 릴리스하는 작업을 자동화합니다. 운영환경으로 빌드를 배포하기 이전의 파이프라인 단계에는 수작업이 없기 때문에 지속적인 배포는 자동화된 테스트에 많은 부분 의존하게 됩니다.

AWS에서 제공하는 서비스를 이용한 CI/CD 구성은 다음과 같다.



AWS 서비스명	설명
CodeCommit	먼저 기존 코드를 Github에서 AWS CodeCommit으로 마이그레이션 한다. AWS CodeCommit은 AWS에서 호스팅하는 버전 제어 서비스로 클라우드에서 자산을 비공개로 저장하고 관리하는 데 사용한다.
CodeBuild	애플리케이션 코드를 빌드하기 위해 CodeBuild를 구성한다. AWS CodeBuild는 소스 코드를 컴파일하고 테스트를 실행하며 배포 준비가 된 소프트웨어 패키지를 생성하는 완전히 관리된다.
CodeDeploy	코드를 EC2 서버에 배포한다. AWS CodeDeploy는 Amazon EC2 인스턴스, 온 프레미스 인스턴스 또는 서버리스 Lambda 기능에 대한 애플리케이션 배포를 자동화하는 배포 서비스이다.
CodePipeline	코드를 지속적으로 제공하는 파이프 라인을 구축한다. AWS CodePipeline은 소프트웨어 배포에 필요한 단계를 모델링, 시각화 및 자동화하는 데 사용할 수 있는 지속적인 제공 서비스이다. 코드를 프로덕션에 전달하기 전에 파이프 라인에 승인 프로세스를 통합한다.

## 관련 기술 멘토링

### □ 주요기능 및 예상결과물

실시간 예약 · webRTC를 이용하여 실시간 비동기 영상 서비스

Web Real-Time Communication의 약자로 웹/앱에서 별다른 소프트웨어 없이 카메라, 마이크 등을 사용하여 실시간 커뮤니케이션을 제공해주는 기술이다.

Peer to Peer 통신은

위 그림과 같이 하나의 컴퓨터와 하나의 컴퓨터가 데이터를 주고 받는 형식이다.

즉, 동등 계층 간 클라이언트/서버의 개념이 없이 동등한 노드들로 구성되어 데이터를 주고 받는다.

회원 가입/로그인 · jwt token을 이용한 로그인 기능

App 화면 · pigma를 이용한 화면 기획, react native를 이용한 하이브리드 앱 개발

현황 모니터링 · react를 이용한 admin 관리 페이지 개발

## □ 핵심기술

SpringBoot -java17 적용, spring framework 이용, gradle 이용, github를 통한 형상관리

React -react18을 이용한 front-end 구성

devops -AWS codepipeline을 위한 CI/CD 구성

aws -elasticbeanstalk, ec2 , RDS , Route53, S3, Cloudfront 적용

### EB(Elastic Beanstalk)

- 1.AWS 클라우드에서 애플리케이션을 신속하게 배포하고 관리할 수 있는 서비스
- 2.애플리케이션을 업로드만 하면 용량 프로비저닝, 로드 밸런싱, 조정, 모니터링 등 자동 처리
- 3.EB console, AWS CLI를 이용해 상호작용

### EC2(Elastic Compute Cloud)란?

EC2는 AWS에서 제공하는 클라우드 컴퓨팅 서비스다.

이 서비스를 통해서 아마존이 각 세계에 구축한 데이터 센터의 서버용 컴퓨터들의 자원을 원격으로 사용할 수 있다. 쉽게 말해, 아마존으로 부터 한 대의 컴퓨터를 임대하는 것이다. AWS가 제공하는 URL(Public DNS)를 통해 이 컴퓨터에 접근할 수 있다.

### EC2의 장점

용량을 늘리거나 줄일 수 있다. (탄력성)

사용한만큼 지불하므로 저렴하다.

사용자가 인스턴스를 완전히 제어할 수 있다.

보안 및 네트워크 구성, 스토리지 관리 효과적이다.



## **RDS(Relational Database Service)란 무엇인가?**

간단히 말하면 관계형 데이터베이스를 제공하는 AWS의 서비스이다. 유저가 사용하기 쉽도록 인프라 등을 자동화 시켜주고 유저들은 엔드포인트로 접속할 수 있는 데이터베이스를 제공받는다.

## **아마존에서 말하는 RDS**

Amazon RDS를 사용하면 클라우드에서 관계형 데이터베이스를 간편하게 설정, 운영 및 확장할 수 있습니다. 하드웨어 프로비저닝, 데이터베이스 설정, 패치 및 백업과 같은 시간 소모적인 관리 작업을 자동화하면서, 비용 효율적이고 크기 조정 가능한 용량을 제공합니다. 사용자가 애플리케이션에 집중해 애플리케이션에 필요한 빠른 성능, 고가용성, 보안 및 호환성을 제공할 수 있도록 지원합니다.

## **Route 53이란?**

AWS에서 제공하는 DNS(Domain Name System)이다.

### **Route 53 기능**

1. 모니터링 - 특정 포트에 대해 모니터링이 가능하다.
2. L4 기능(Failover기능 제공)
3. GSLB(global server load balancing) - 지역에 상관없이 부하를 분산 및 속도 개선이 가능함

## AWS Simple Storage Service(S3)

인터넷용 스토리지 서비스입니다. 이 서비스는 개발자가 더 쉽게 웹 규모 컴퓨팅 작업을 수행할 수 있도록 설계되었습니다.

### AWS S3

Simple Storage Service의 약자로 파일 서버의 역할을 하는 서비스다. 일반적인 파일서버는 트래픽이 증가함에 따라서 장비를 증설하는 작업을 해야 하는데 S3는 이와 같은 것을 대행한다. 트래픽에 따른 시스템적인 문제는 걱정할 필요가 없어진다. 또 파일에 대한 접근 권한을 지정 할 수 있어서 서비스를 호스팅 용도로 사용하는 것을 방지 할 수 있다.

AWS S3(Simple Storage Service)의 특징

- 많은 사용자가 접속을 해도 이를 감당하기 위해서 시스템적인 작업을 하지 않아도 된다.
- 저장할 수 있는 파일 수의 제한이 없다.
- 최소 1바이트에서 최대 5TB의 데이터를 저장하고 서비스 할 수 있다.
- 파일에 인증을 붙여서 무단으로 액세스 하지 못하도록 할 수 있다.
- HTTP와 BitTorrent 프로토콜을 지원한다.
- REST, SOAP 인터페이스를 제공한다.
- 데이터를 여러 시설에서 중복으로 저장해 데이터의 손실이 발생할경우 자동으로 복원한다.
- 버전관리 기능을 통해서 사용자에게 의한 실수도 복원이 가능하다.
- 정보의 중요도에 따라서 보호 수준을 차등할 수 있고,이에 따라서 비용을 절감할 수 있다.  
(RSS - Reduced Redundancy Storage의 약자로 일반 S3 객체에 비해서 데이터가 손실될 확률이 높은 형태의 저장 방식. 대신에 가격이 저렴하다 )

## CloudFront

클라우드프론트는 개발자 친화적 환경에서 짧은 지연 시간과 빠른 전송 속도로 데이터, 동영상, 애플리케이션 및 API를 전세계 고객에게 안전하게 전송하는 고속 콘텐츠 전송 네트워크(CDN) 서비스이다.

CloudFront는 CDN 서비스와 이외에도 기본 보안 기능(Anti-DDoS)을 제공한다.

## CDN 이란?

CDN(Content Delivery Network or Content Distribution Network, 콘텐츠 전송 네트워크)은 콘텐츠를 효율적으로 전달하기 위해 여러 노드를 가진 네트워크에 데이터를 저장하여 제공하는 시스템이다.

인터넷 서비스 제공자(ISP, Internet Service Provider)에 직접 연결되어 데이터를 전송하므로, 콘텐츠 병목을 피할 수 있는 장점이 있다.

우리가 프론트 공부할때 제이쿼리 같은 라이브러리를 링크 src로 불러온적이 있을텐데, 이게 바로 CDN 서비스를 이용하는 것이라고 보면 된다.

## CDN 특징

- 웹 페이지, 이미지, 동영상 등의 콘텐츠를 본래 서버에서 받아와 캐싱
- 해당 콘텐츠에 대한 요청이 들어오면 캐싱해 둔 콘텐츠를 제공
- 콘텐츠를 제공하는 서버와 실제 요청 지점 간의 지리적 거리가 매우 먼 경우 or 통신 환경이 안좋은 경우 → 요청지점의 CDN을 통해 빠르게 콘텐츠 제공 가능
- 서버의 요청이 필요 없기 때문에 서버의 부하를 낮추는 효과

## **docker -AWS ECS,ECR , gradle jib을 이용한 docker 배포**

Docker는 애플리케이션을 신속하게 구축, 테스트 및 배포할 수 있는 소프트웨어 플랫폼입니다. Docker는 소프트웨어를 컨테이너라는 표준화된 유닛으로 패키징하며, 이 컨테이너에는 라이브러리, 시스템 도구, 코드, 런타임 등 소프트웨어를 실행하는 데 필요한 모든 것이 포함되어 있습니다. Docker를 사용하면 환경에 구애받지 않고 애플리케이션을 신속하게 배포 및 확장할 수 있으며 코드가 문제없이 실행될 것임을 확신할 수 있습니다.

AWS에서 Docker를 실행하면 개발자와 관리자가 어떠한 규모에서든 매우 안정적이며 저렴한 방식으로 애플리케이션을 구축, 제공 및 실행할 수 있습니다.

### **ECS(Elastic Container Service)**

AWS에서 제공하는 컨테이너 오케스트레이션 서비스입니다.

ECS를 사용하여 Amazon EC2 인스턴스의 관리형 클러스터 혹은 AWS Fargate를 통한 서버리스 환경에서 애플리케이션을 손쉽게 실행할 수 있습니다.

ECS를 사용함으로써 다음과 같은 이점을 얻을 수 있습니다.

- 다양한 AWS의 서비스 연계

AWS에서 제공하고 있는 다양한 서비스들과 옵션들을 연계하여 더욱 효율적으로 컨테이너를 관리할 수 있습니다.

- 제어 플레인 또는 노드 관리 불필요

AWS Fargate를 통해 서버리스 환경을 이용할 수 있으며 제어 플레인 또는 노드를 관리, 인스턴스 패치 및 조정이 불필요하므로 운영에 소요되는 시간이 줄어듭니다.

- 컴퓨팅 비용 절감

클라우드 서비스라서 물리적 서버에 대한 요금을 절약할 수 있습니다. 또한 ECS는 프로비저닝 및 자동 조정을 자율적으로 처리하므로 컴퓨팅 비용을 최대 50%까지 절감할 수 있습니다.

- 보안 및 규제 요구 사항 충족

ECS는 AWS 관리 및 거버넌스 솔루션과 기본적으로 통합됩니다. 따라서 전 세계 거의 모든 규제 기관의 보안 및 규정 준수 표준을 충족할 수 있습니다.

### **ECR이란?**

Docker 컨테이너 이미지를 손쉽게 저장, 관리, 및 배포할 수 있게 해주는 완전 관리형 Docker 컨테이너 레지스트리

### **Amazon ECR의 기능**

- 수명 주기 정책을 통해 이미지의 수명 주기를 관리한다.
- 사용되지 않는 이미지를 정리하는 규칙을 정의하여 리포지토리에 적용할 수 있다.
- 각 리포지토리는 푸시 시 스캔하도록 구성할 수 있으며, 이 과정을 통해 컨테이너 이미지의 소프트웨어 취약성을 식별하는 데 도움을 줄 수 있다.
- 교차 리전 및 교차 계정 복제를 통해 이미지를 필요한 곳에 쉽게 배치할 수 있다.

## 2023년 한이음 ICT멘토링 프로젝트 개요서

주제영역*	<input type="checkbox"/> 생활 <input type="checkbox"/> 업무 <input type="checkbox"/> 공공/교통 <input type="checkbox"/> 금융/핀테크 <input type="checkbox"/> 의료 <input type="checkbox"/> 교육 <input type="checkbox"/> 유통/쇼핑 <input type="checkbox"/> 엔터테인먼트
기술분야	<input type="checkbox"/> SW-AI <input type="checkbox"/> 방송·콘텐츠 <input type="checkbox"/> 블록체인·융합 <input type="checkbox"/> 디바이스 <input type="checkbox"/> 자세·대보안 <input type="checkbox"/> 미래통신·전파
성과목표	<input type="checkbox"/> 논문게재 및 포스터 발표 <input type="checkbox"/> 프로그램등록 <input type="checkbox"/> 특허 <input type="checkbox"/> 기술이전 <input type="checkbox"/> 공모전(공모전명 ) <input type="checkbox"/> 기타( )
프로젝트명	개발 콘텐츠 숏영상 공유 플랫폼 - 숏데브

### □ 프로젝트 개요

#### 프로젝트 소개 (제안배경 및 주요내용)

· 개발자들은 개발 관련 정보를 영상이나 유튜브를 통해서 정보를 많이 얻고 있다. 최근 full 영상 보다는 15초 내의 틱톡 및 short 컷 영상이 인기가 많다. 이에 따라 개발 관련 내용으로만 구성된 15초 내의 short컷 영상을 제공하는 서비스를 만들어본다. 개발 관련 최신 트렌드나 개발 기술에 대해 짧은 영상을 통해 빠르고, 신속하게 정보를 얻을 수 있는 플랫폼이 된다. 빠른 정보를 얻고 싶거나, 개발관련 흥미로운 영상을 접하고 싶다면 숏데브 플랫폼을 많이 이용할 것이다. 많이 이용하게 되면 개발 관련 교육, 도서, 등 사용자들을 위한 광고도 연재할 수 있는 플랫폼으로 수익을 낼 수 있다.

예상일정	예상팀원(수)	예상난이도
2023. 04. 01. ~ 11. 30.	5 (명)	중
활용장비	· AWS 서버 이용	
지도방법	· 기획안 리뷰, 코드 리뷰, 기술 stack 논의 -front-end : 어드민 - react, App - <b>react-native</b> -Bactk-end : java, Springboot, redis, jwt를 이용한 백엔드 서버 구현 -github : github를 통한 aws ci/cd 구성 관련 기술 멘토링	

### □ 주요기능 및 예상결과물

실시간 예약	· webRTC를 이용하여 실시간 비동기 영상 서비스
회원 가입/로그인	· jwt token을 이용한 로그인 기능
App 화면	· pigma를 이용한 화면 기획, react native를 이용한 하이브리드 앱 개발
현황 모니터링	· react를 이용한 admin 관리 페이지 개발

□ 핵심기술

SpringBoot	-java17 적용, spring framework 이용, gradle 이용, github를 통한 형상관리
React	-react18를 이용한 front-end 구성
devops	-AWS codepipeline을 위한 CI/CD 구성
aws	-elasticbeanstalk, ec2 , RDS , Route53, S3, Cloudfront 적용
docker	-AWS ECS,ECR , gradle jib를 이용한 docker 배포

□ 기대효과 및 활용분야

개발자 지식 플랫폼	-흥미로운 개발 주제로 개발자 직무자들에게 도움이 있을 수 있는 app 개발
개발 트렌드 정보 공유	-최신 트렌드 기술과 상호 기술 사용법을 영상으로 공유 할 수 있는 app 개발
개발 역량 향상	-실제 운영할 수 있는 app을 개발해 봄으로써, 실무에서 적용할 수 있는 오픈소스를 적용해보고, 추후 개발자로서 역량을 키워나가는 데 도움이 될 수 있는 기회를 만들어본다.

□ 참고사항 # 프로젝트 공고자 신분에 따라 작성

희망 멘티	전공분야	무관
	지 역	<input type="checkbox"/> 수도권 <input type="checkbox"/> 충청권 <input type="checkbox"/> 영남권 <input type="checkbox"/> 호남권 <input type="checkbox"/> 강원권 <input type="checkbox"/> 제주 <input type="checkbox"/>
	학 년	<input type="checkbox"/> 4학년 <input type="checkbox"/> 3학년 <input type="checkbox"/> 2학년 <input type="checkbox"/> 1학년 <input type="checkbox"/>
	필요역량 (프로그래밍언어 등)	java 또는 javascript 중 하나 기본 지식 필요
멘티들에게 하고 싶은 말		실제 사용될 수 있는 app을 만들어보고, 이번 프로젝트를 통해서 백엔드 또는 프론트 엔지니어 맡은 한분야에 대해 전문 역량을 키워 나갈 수 있도록 지도 하려 합니다. 좋은 성과물이 나오면 실제 운영할 수 있는 앱으로 발전 시켜보는 것도 좋을거 같습니다. 제 이력을 보시면 알겠지만 자신있게 멘토링 해드릴 수 있으니 믿어주시고, 단순 프로젝트 참여한다는 마인드보다는 개발 역량에 관심이 많고, 열정을로 해주실 있는 멘티분들 신청해주세요.