

# Portfolio

## <개인 신상>

	한글 김 정 후		한자 金 定 厚	영문 Jung hu Kim
	생년월일	1993년 11월 24일		
	최종학력	University of Wisconsin-Madison 생물학 전공		
	주소	경기도 의왕시 포일동 676-10 201호		
	연락처	전화번호	전화 010-7304-6388	휴대전화
		E-mail	dabltk@naver.com	

Skill Set	Programming Language : JAVA(주), C/C++, Python
	OS : Ubuntu, CentOS
	RDBMS : MySQL, Oracle
	NoSQL : MongoDB
	Framework : Spring Framework, Tensorflow
	Web : HTML5, CSS, JavaScript, JSP, NodeJS, Apache Tomcat
자격증, 수상 등 특기사항	IT 자격증 : 정보처리기사, 리눅스마스터2급
	어학 자격증 : TOEIC 965, OPIc AL, JLPT N1, JPT 930
	기타 자격증 : 한자급수 2급, 한국사능력검정 1급
	수상경력 : 2019.03.15. 비전설계 실천사례 공모전 육군 참모총장상 수상

## <수행 프로젝트>

순번	주제	수행 기간	비고
1	Session 및 토큰 기반 인증, Ajax 기반 예약시스템, 계층형 게시판을 가진 리조트 웹서비스 풀스택 개발	20.1.13~현재	<ul style="list-style-type: none"> <li>• Bootstrap</li> <li>• JSP</li> <li>• HTML5</li> <li>• MySQL</li> <li>• Java</li> <li>• JQuery/Ajax</li> </ul>
2	NodeJS와 MongoDB를 활용한 비디오 호스팅 서비스 구축	20.1.1~20.1.6	<ul style="list-style-type: none"> <li>• Oracle</li> <li>• VMware/CentOS</li> </ul>
3	JSP Model1과 Model2 아키텍처를 적용한 성적 조회 웹 서비스 구축 및 비교	19.12.12~19.12.15	<ul style="list-style-type: none"> <li>• MySQL</li> <li>• JSP</li> <li>• JS/JQuery</li> <li>• Apache Tomcat</li> </ul>
4	자바로 구현한 멀티스레드 기반 TCP Socket 통신 오목 게임	19.11.25~19.11.27	<ul style="list-style-type: none"> <li>• Java</li> <li>• TCP/Socket</li> <li>• Thread</li> </ul>
5	Chart.js와 리눅스 셸스크립트를 활용한 실시간 시스템 자원 모니터링 웹 애플리케이션 개발	19.09.10~19.09.11	<ul style="list-style-type: none"> <li>• Ubuntu</li> <li>• JavaScript</li> <li>• HTML5</li> </ul>
6	Oracle DBMS 서버 구축 및 기업 매출 실적 시각화	19.11.15~19.11.17	<ul style="list-style-type: none"> <li>• Oracle</li> <li>• VMware/CentOS</li> </ul>
7	네이버 NMT API 기반 OCR 번역기 안드로이드 앱 개발	18.10.23~18.10.26	<ul style="list-style-type: none"> <li>• Java</li> <li>• Android Studio</li> </ul>
8	AWS 서비스를 활용한 서버리스 웹 애플리케이션 개발	2020.02.05.~2020.02.06	<ul style="list-style-type: none"> <li>• AWS</li> </ul>

# 김 정 후 의 프로젝트 #1

## <프로젝트 개요>

주제	Session 및 토큰 기반 인증, Ajax 기반 예약시스템, 계층형 게시판을 가진 리조트 홈페이지 풀스택 개발
개요	JSP,MySQL,HTML5,CSS3,JQuery,Java 지금까지 배운것들을 활용하여 기획, 프론트엔드, 백엔드 개발을 전부 수행한 반응형 리조트 웹사이트
주요 수행 내용	<ul style="list-style-type: none"> <li>- Bootstrap 컴포넌트를 활용한 UI 개발</li> <li>- DB와 연동한 계층형 게시판 구현</li> <li>- Session 및 토큰 기반 사용자 인증 기능 구현</li> <li>- Ajax로 예약중복을 비동기적으로 체크하는 예약시스템 구현</li> <li>- Session을 이용한 로그인/로그아웃/관리자 권한 구현</li> </ul>

## <프로젝트 결과>

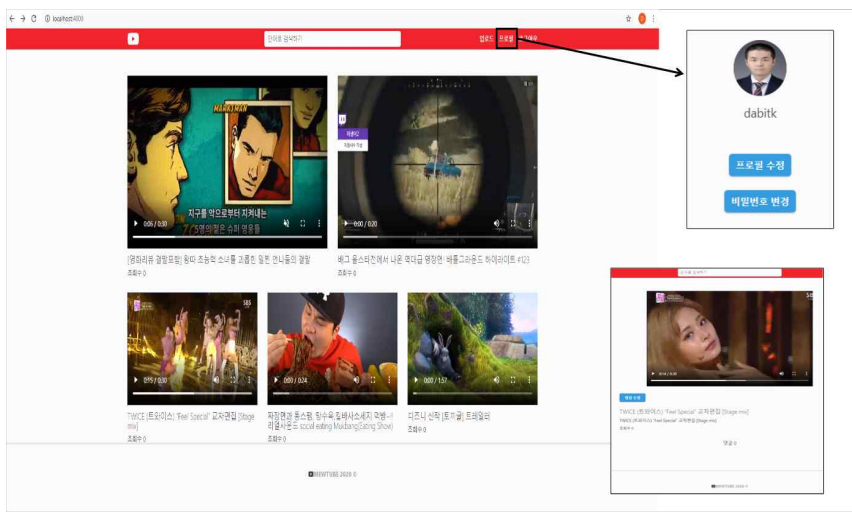
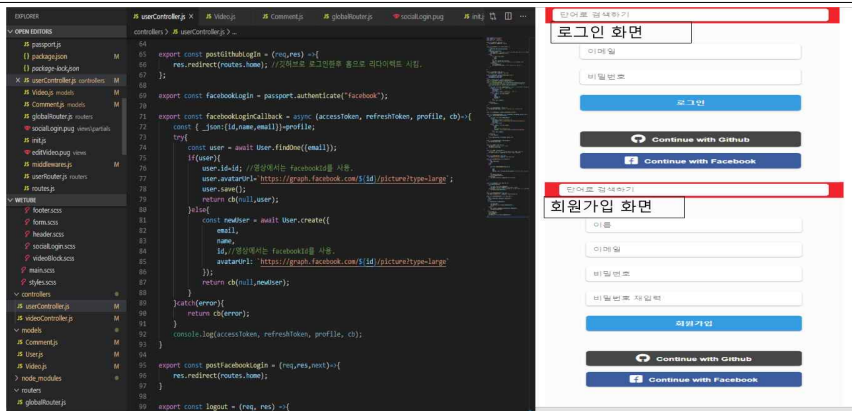
<p>1. Bootstrap, JQuery, TMap API를 활용한 프론트엔드 개발</p> <p>2. JSP, MySQL을 활용한 백엔드 HTML 렌더링 및 로직 처리</p>	
<p>1. 세션 및 토큰 (OAuth)을 이용한 사용자 인증</p> <p>2. 계층형 게시판 구현</p> <p>3.</p>	
GitHub	<a href="https://github.com/dabitk/LuxResort">https://github.com/dabitk/LuxResort</a>

## 김 정 후 의 프로젝트 #2

### <프로젝트 개요>

주제	NodeJS와 MongoDB를 활용한 비디오 호스팅 서비스 구축
개요	NodeJS와 MongoDB를 이용하여 동영상 파일의 업로드, 수정, 삭제, 검색이 가능한 웹 애플리케이션 개발
주요 수행 내용	<ul style="list-style-type: none"> <li>-Express 프레임워크를 이용, NodeJS 서버 구축 및 미들웨어를 이용한 프로그램 흐름 제어</li> <li>-Mongoose 패키지를 이용하여 NodeJS와 MongoDB를 연결, 이후 동영상 데이터에 대한 CRUD 수행</li> <li>- Passport 패키지를 이용하여 GitHub와 Facebook OAuth 구현</li> <li>- Pug 패키지를 이용한 HTML페이지 렌더링</li> <li>- Model(DAO, DTO), View(Pug, SCSS), Controller(route.js와 controller.js)를 완전히 분리한 웹 애플리케이션 구현</li> </ul>

### <프로젝트 결과>

동영상 파일의 업로드, 검색, 수정, 삭제 기능 구현	
회원가입/로그인 기능 구현, GitHub와 Facebook OAuth 기능 구현	
GitHub	<a href="https://github.com/dabitzk/MewTube">https://github.com/dabitzk/MewTube</a>

# 김 정 후 의 프로젝트 #3

## <프로젝트 개요>

주제	JSP Model1과 Model2 아키텍처를 적용한 성적 조회 웹 서비스 구축 및 비교
개요	MySQL과 연동하여 CRUD를 수행하는 성적조회 게시판을 Model1과 Model2로 각각 구현
주요 수행 내용	-Model2에서는 Controller 역할로 서블릿 파일을 생성, xml파일에 url패턴 등록 후 Controller에서 dispatcher를 이용하여 클라이언트 요청에 대한 라우팅 적용 -DAO객체에 싱글톤 패턴을 적용하여 웹 애플리케이션 전체에서 단 하나의 객체를 공유하도록 강제 -View에 해당하는 JSP파일에서는 JSTL을 사용하여 코드의 간결함과 가독성 향상을 도모 -Javascript(jQuery)를 활용하여 폼 입력값에 대한 검증을 수행

## <프로젝트 결과>

● 모델 1 구조의 프로젝트 설계도

```

    graph LR
        WB[웹 브라우저] <--> C[Controller]
        C <--> M[Model]
        M <--> V[View]
        V <--> DB[(MySQL DBMS)]
        C <--> DB
    
```

Model (JavaBean):  
 - name VARCHAR(20)  
 - id INT(11) UNSIGNED  
 - AUTO INCREMENT  
 - sex CHAR(1)  
 - mat no. VARCHAR(10)  
 - PRIMARY KEY(id)

MySQL DBMS:  
 - name VARCHAR(20)  
 - id INT(11) UNSIGNED  
 - AUTO INCREMENT  
 - sex CHAR(1)  
 - mat no. VARCHAR(10)  
 - PRIMARY KEY(id)  
 - INDEX (id)

● 모델 2 구조의 프로젝트 설계도

```

    graph LR
        WB[웹 브라우저] <--> C[Controller]
        C <--> M[Model]
        M <--> V[View]
        V <--> DB[(MySQL DBMS)]
        C <--> DB
    
```

Controller (Servlet):  
 - ExamController.java  
 - (이름, 학번, 성별, 학점, 수강, 합계)를 받아서 DB에 저장하고, DB에서 데이터를 받아서 View에 전달한다.

Model (JavaBean):  
 - name VARCHAR(20)  
 - id INT(11) UNSIGNED  
 - AUTO INCREMENT  
 - sex CHAR(1)  
 - mat no. VARCHAR(10)  
 - PRIMARY KEY(id)

MySQL DBMS:  
 - name VARCHAR(20)  
 - id INT(11) UNSIGNED  
 - AUTO INCREMENT  
 - sex CHAR(1)  
 - mat no. VARCHAR(10)  
 - PRIMARY KEY(id)  
 - INDEX (id)

JSP Database Setup 2

성적입력추가완료

성적입력추가

성적 조회 후 정정 / 삭제

성적 조회 후 정정 / 삭제

JDBC를 통해 MySQL과 Java 사이에 커넥션 생성

모든 Embedded SQL은 DAO Java파일 에서 관리하게 함으로써 뷰와 로직의 분리를 추구함

성적입력추가

성적입력추가

성적입력추가

문자열에 대한 최대길이 설정  
 - 숫자 필드에는 0부터 100 까지만 허용  
 - 정규식을 활용하여 입력값 안의 특수문자, 공백 필터링

150.168.2.3.14.8080 says  
 150.168.2.3.14.8080 says

MySQL과  
연동하는  
성적조회  
게시판을  
Model1과  
Model2로 각각  
구현

Javascript와  
HTML5 태그를  
활용한 폼 입력값  
검증

GitHub

[https://github.com/dabtk/JspModel1\\_2](https://github.com/dabtk/JspModel1_2)

## 김 정 후 의 프로젝트 #4

### <프로젝트 개요>

주제	자바로 구현한 멀티스레드 기반 TCP Socket 통신 오목 게임
개요	Thread와 TCP Socket을 이용하여 서버-클라이언트 구조의 1:1 온라인 오목 게임을 구현
주요 수행 내용	<ul style="list-style-type: none"> <li>- 자바를 이용하여 기본적인 오목의 게임 로직 클래스를 구현</li> <li>- 자바로 서버와 클라이언트를 구축하여 TCP Socket을 통해 데이터를 주고 받음</li> <li>- 서버상에 싱글턴으로 게임판 객체를 구현.</li> <li>- 옵저버 패턴을 적용, 서버가 게임의 상태변화를 클라이언트들에게 브로드캐스트 하도록 메소드 구현. (Synchronized 키워드를 사용하여 스레드 흐름 제어)</li> </ul>

### <프로젝트 결과>

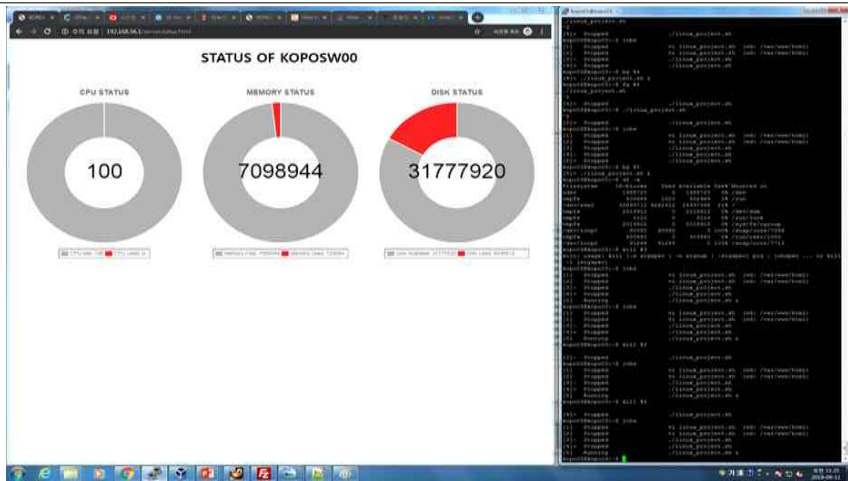
<p>1) 자바로 서버와 클라이언트를 구축.</p> <p>2) 클라이언트는 서버에게 좌표값만을 보내며 서버는 싱글턴으로 구현한 게임판 객체를 갱신함.</p>	<p>1) 클라이언트는 서버로부터 직렬화된 보드객체를 인력스트림으로 받아서 화면에 출력함</p> <p>2) 클라이언트는 서버에게 "X,Y:Color" 형식의 문자열을 출력스트림을 통해 보냄</p> <p>초기상태</p> <p>test3 : x:y:color 형식의 문자열을 성공적으로 받았음</p> <p>test4 : (:) 기준으로 request 값 파싱 성공</p> <p>test5 : x,y,color 값: 0, 4, 1</p> <p>BLACK WON</p> <p>test6 : 직렬화한 보드를 보냈습니다.</p> <p>OMOK</p> <p>JRE System Library [JavaSE-1.8]</p> <p>src</p> <p>OMOK</p> <ul style="list-style-type: none"> <li>Board.java</li> <li>CLIENT.java</li> <li>ClientThread.java</li> <li>SERVER.java</li> </ul>
<p>1) 옵저버 패턴을 적용하여 게임상태 변화를 클라이언트에게 브로드캐스트</p> <p>2) Synchronized 키워드를 활용한 멀티스레드 흐름 제어</p>	<pre> public void broadcast(Board b) throws Exception{     //옵저버 패턴 적용: 게임상태가 변한 것을 클라이언트들에게 브로드캐스트로 알린다     synchronized (playerList) {         //멀티스레딩 환경에서는 자원에 동시에 접근하는 것을 막기위해 synchronized 키워드를 사용한다         //이 메소드가 실행중에는 다른 스레드는 이 메소드를 호출할 수 없다.         for(ObjectOutputStream p : playerList) {             p.writeObject(board); //데이터 직렬화             p.reset();         }     } } </pre> <p>1) 서버는 클라이언트로 부터 받은 좌표값으로 자신의 보드와 게임상태를 업데이트</p> <p>2) 상태 업데이트 후 직렬화한 보드객체를 브로드캐스팅하여 클라이언트에게 전달함</p> <p>보드 상태가 바뀌었습니다.</p> <p>이 정보를 클라이언트들에게 전달합니다.</p>
Git	<a href="https://github.com/dabitk/TCP_OMOK">https://github.com/dabitk/TCP_OMOK</a>

## 김 정 후 의 프로젝트 #5

### <프로젝트 개요>

주제	Chart.js와 리눅스 셸스크립트를 활용한 실시간 시스템 자원 모니터링 웹 애플리케이션 개발
개요	1초마다 페이지가 새로고침 되면서 Tomcat서버가 위치한 Ubuntu 가상머신의 시스템 자원이 브라우저상에 출력됨
주요 수행 내용	<ul style="list-style-type: none"> <li>- 로컬 Windows OS에서 Ubuntu 가상머신의 톰캣서버에 접근하기 위해 8080포트에 대한 네트워크 포트포워딩 설정</li> <li>- sar명령어를 이용한 CPU 모니터링, free명령어를 이용한 메모리 모니터링, 그리고 df명령어를 이용한 디스크 용량 모니터링을 구현</li> <li>- 1초마다 serverstatus.html이라는 정적 페이지의 변수값을 셸스크립트 while문을 통해 갱신하도록 설정</li> </ul>

### <프로젝트 결과>

1초마다 serverstatus.ht ml파일을 갱신하도록 리눅스 스크립트 파일 작성	<pre>#!/bin/sh  while: do     cat t1 &gt;&gt; tt     sar 1 1 grep Average  awk '{ print "CPU_FREE=" \$8 }' &gt;&gt; tt     sar 1 1 grep Average  awk '{ print "CPU_USED=" 10-\$8 }' &gt;&gt; tt     free grep Mem  awk '{ print "MEM_FREE=" \$4 }' &gt;&gt; tt     free grep Mem  awk '{ print "MEM_USED=" \$3 }' &gt;&gt; tt     df -k grep -V Filesystem  awk '{ sum + \$4 } END { print "DSK_FREE=" sum }' &gt;&gt; tt     df -k grep -V Filesystem  awk '{ sum += \$3 } END { print "DSK_USED=" sum }' &gt;&gt; tt     cat t2 &gt;&gt; tt     cp tt /var/www/serverstatus.html     sleep 1 done</pre>
Chart.js 라이브러리를 활용한 도넛 차트 작성	
GitHub	<a href="https://github.com/dabitk/Linux201909">https://github.com/dabitk/Linux201909</a>

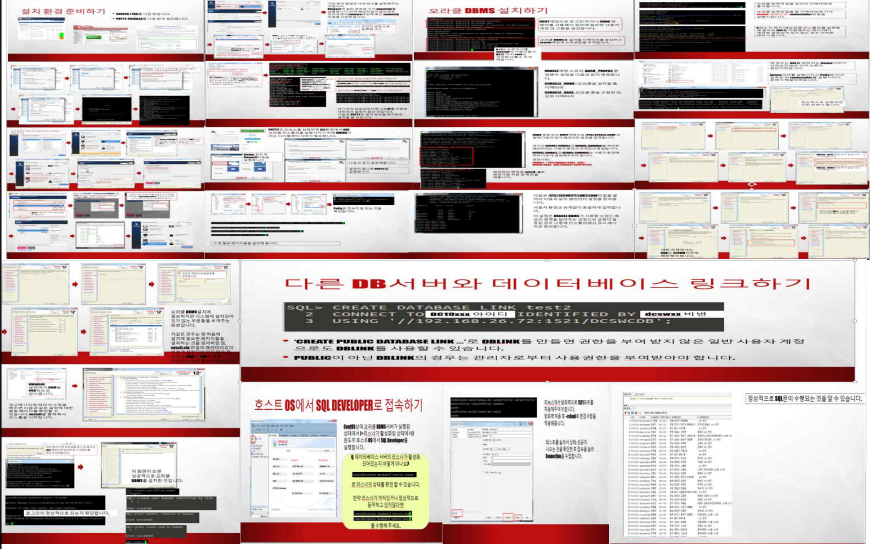
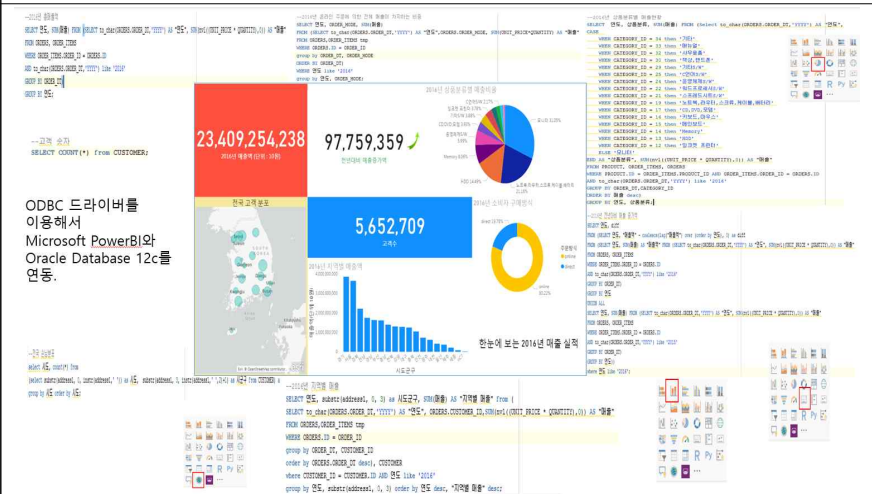


# 김 정 후 의 프로젝트 #6

## <프로젝트 개요>

주제	Oracle DBMS 서버 구축 및 기업 매출 실적 시각화
개요	Oracle 개발환경을 처음부터 구축하고 원격 서버에서 가져온 데이터를 분석하여 이를 시각화
주요 수행 내용	<ul style="list-style-type: none"> <li>- CentOS 7 로컬 가상머신 환경에 Oracle Database 12c 서버 및 클라이언트 환경 구축</li> <li>- 원격 DB서버와 데이터베이스 링크를 구축하고 테이블 복사를 수행</li> <li>- ODBC 드라이버를 이용하여 ORACLE과 Microsoft PowerBI를 연동</li> <li>- SQL쿼리를 이용, 데이터 정제 후 PowerBI를 이용하여 시각화</li> </ul>

## <프로젝트 결과>

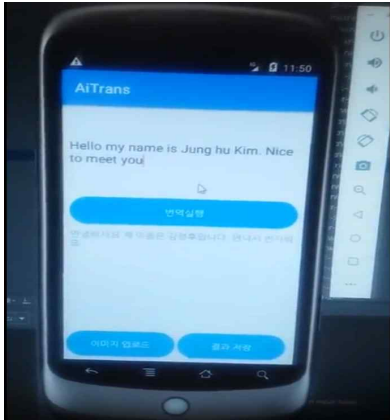
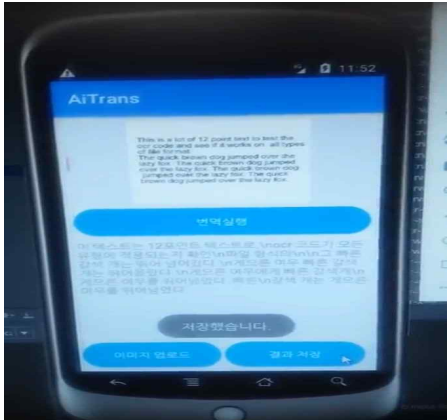
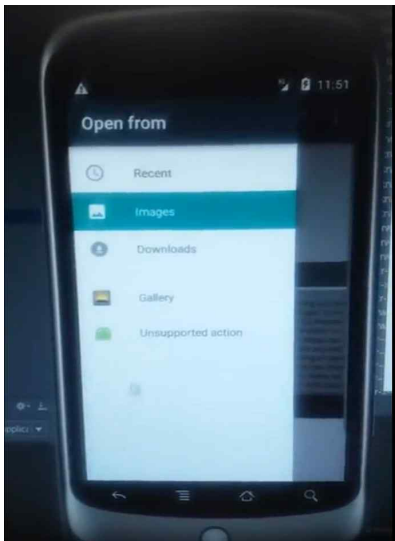
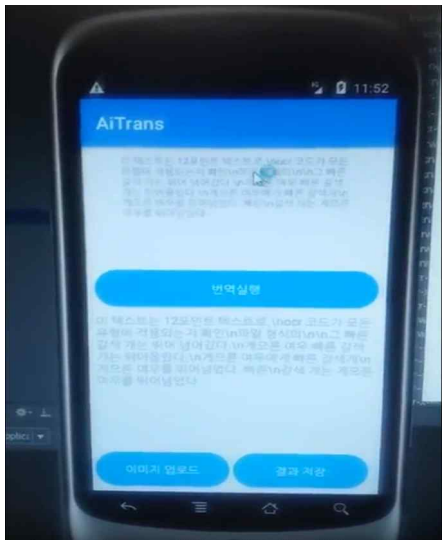
CentOS 환경에 Oracle DBMS 서버 구축 및 원격 서버와의 DB Link 구축	
빅데이터 시각화	
GitHub	<a href="https://github.com/dabtk/Installing-Oracle-12c-on-CentOS-virtual-machine">https://github.com/dabtk/Installing-Oracle-12c-on-CentOS-virtual-machine</a>

## 김 정 후 의 프로젝트 #7

### <프로젝트 개요>

주제	네이버 NMT API 기반 OCR 번역기 안드로이드 앱 개발
개요	네이버 NMT API, 구글 Tesseract OCR 모듈을 활용하여 업로드한 이미지상의 영문을 한글로 번역하고, 결과를 갤러리에 저장할 수 있는 안드로이드 앱.
주요 수행 내용	<ul style="list-style-type: none"> <li>- 안드로이드 스튜디오에서 XML과 자바를 사용, 앱의 레이아웃 구성</li> <li>- Tesseract OCR 오픈소스를 import하여 OCR 모듈 구축</li> <li>- 네이버 REST API를 통해 입력한 문자열, 또는 이미지에서 추출한 문자열을 보내고 번역문을 JSON형태로 받아오는 모듈 구축</li> <li>- JSON 데이터 파싱 수행</li> </ul>

### <프로젝트 결과>

<p>1) 이미지를 OCR모듈을 통해 문자열로 변환</p> <p>2) 네이버 REST API로 번역문을 JSON형태로 받고 이를 파싱하여 출력</p>	 
<p>결과물을 스마트폰의 갤러리에 저장 및 불러오기 기능</p>	 
시연영상	<a href="https://www.youtube.com/watch?v=SriE8RFHoAo">https://www.youtube.com/watch?v=SriE8RFHoAo</a>

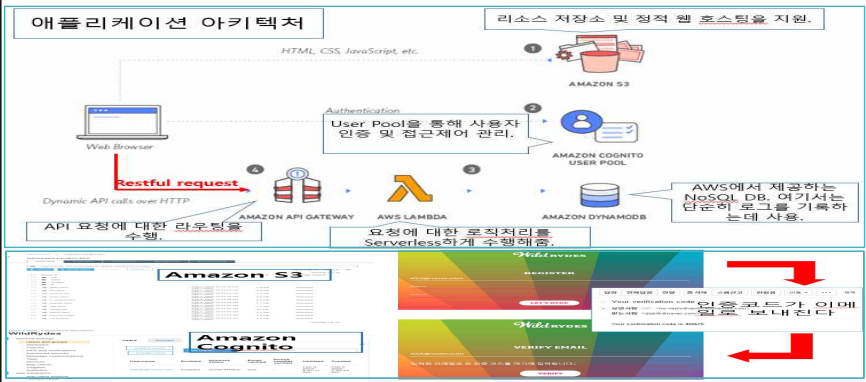
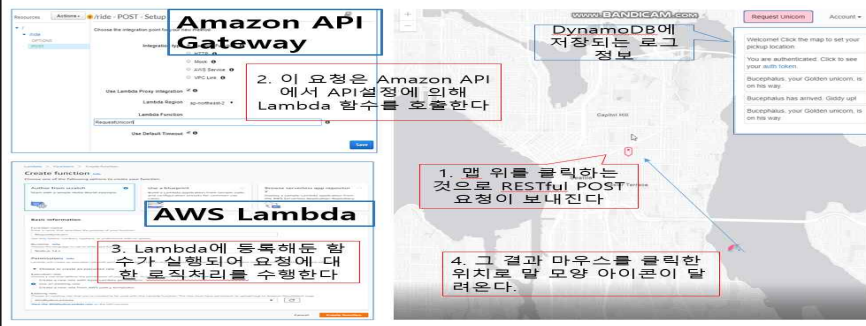


## 김 정 후 의 프로젝트 #8

### <프로젝트 개요>

주제	AWS 서비스를 활용한 서버리스 RESTful 웹 애플리케이션 구현
개요	AWS 홈페이지의 공식문서를 참고하며 Amazon S3, Amazon Cognito, Amazon API Gateway, AWS Lambda, Amazon DynamoDB 서비스를 이용, 1) 사용자 인증 기능과 2) 서버리스 환경에서 RESTful request를 처리하는 웹 애플리케이션을 구현.
주요 수행 내용	<ul style="list-style-type: none"> <li>- Amazon S3에 웹 애플리케이션의 리소스를 저장하고 정적 웹 호스팅 기능을 사용.</li> <li>- Amazon Cognito를 활용하여 다음 두가지 기능을 구현: <ol style="list-style-type: none"> <li>1. 회원가입하면 인증코드를 담은 이메일을 발송하고 이 코드를 이용해서 회원가입을 완료</li> <li>2. Cognito User Pool을 이용한 사용자 권한 관리 및 접근제어</li> </ol> </li> <li>- Amazon API Gateway를 이용하여 POST요청에 대한 REST API 를 설정함.</li> <li>- AWS Lambda를 이용하여 API Gateway로부터 받은 RESTful request를 처리하도록 설정함 (개발자는 서버 환경을 직접 구성하지 않고 구현한 로직을 실행 가능)</li> <li>- Amazon DynamoDB에 AWS Lambda가 request 처리 결과(로그)를 DB테이블에 Insert 할 수 있는 권한을 주기 위해서 AWS IAM 역할 설정.</li> </ul>

### <프로젝트 결과>

1) Amazon S3 정적 웹 호스팅 기능 사용 2) 이메일 인증을 포함한 회원가입 기능 설정	
1) REST API 설정 및 API에 의해 호출되는 Lambda 함수 설정 2) DynamoDB에 결과 로그 저장	
GitHub	<a href="https://github.com/dabtk/AWS_Serveless_WebApp">https://github.com/dabtk/AWS_Serveless_WebApp</a>