

# 열정으로 무장한 개발자 김인규입니다

PORTFOLIO

CONTACT  
sample@naver.com  
010 1234 5678





# 성장하는 개발자 김인규입니다

개발이 좋아 개발을 시작했고,  
이제 일로 사람들에게 서비스를 해보고 싶습니다.

## 김인규 / Inkyu Kim

1993.08.30 / 서울특별시

Tel. 010-1234-5678

Email. sample@naver.com

서울특별시 서초구

## GRADUATION

2022 서울대학교 컴퓨터공학과 입학

2028 서울대학교 컴퓨터공학과 졸업

## SKILL

언어 : Java, Javascript, HTML, CSS, SQL

프레임 워크 : JQuery, bootstrap, Spring

boot, Mybatis

Tool : Tomcat, Mysql

## AWARDS

2024 무슨대회 무슨무슨상

2027 무슨공모전 어떤어떤상

2027 롤루랄라공모전 눈누난나상

## PROJECT

2022 첫 번째 프로젝트

2022 두 번째 프로젝트

2028 세 번째 프로젝트

2028 네 번째 프로젝트

## ABOUT

열정 가득

만수무강!!

PROJECT.1

# 마론달 그램

## SNS copy 프로젝트

---

# 프로젝트 소개

## 마론달 그램

웹 서비스에 대한 이해를 높이기 위해서 최근 가장 많이 사용되고 있는 **instagram을 copy 한 프로젝트**  
해당 서비스의 사용성들을 기반으로 **기획, 프로젝트 설계, DB 설계를 직접한 1인 프로젝트**이다.  
로그인을 기반으로 서비스가 운용되며, timeline 기능을 중점으로 개발 하였다.  
주요 기능은 **글쓰기, 댓글, 좋아요** 기능이다.

## 프로젝트 진행 느낀점

혼자서 처음 시작해보는 하나의 완전한 프로젝트이다 보니 완성이 될지 걱정이 많이 되었습니다.

배운대로 각각의 기능을 하나의 독립된 **모듈형태**로 생각하고, 작은 단위부터 완전하게 만들어 가자라고 생각했습니다.

처음 **설계 과정에서** 큰 그림을 그릴 때 힘든 점이 있었지만, 기능 하나 하나를 쪼개서 구상 하다보니 조금씩 큰 그림이 완성되어 가는 것을 느꼈습니다.

특히, Spring 프레임워크의 **흐름을 파악**하는 것에 가장 큰 중점을 두었습니다. 서브릿을 통해서도 웹서비스를 경험해 보았지만, Spring을 통한 프로젝트 경험은 또 다른 영감을 주었습니다. Controller Service Reposotory 로 이어지는 구조는 단순히, 기능을 개발하는 것뿐만 아니라 구조적인 부분이 개발에 어떠한 이점을 주는지 체감 할 수 있었습니다.

또한, 여러 Database 의 여러 테이블의 데이터를 조합하여, 하나의 독립된 데이터로 만들어 가는 과정도 많은 경험을 얻었습니다. 데이터를 어떤 형태의 Class로 설계해서 구상해야 하는지, **도메인을 설계** 하는 과정을 익히는 좋은 경험이었습니.

실제 사용자가 사용한다는 생각으로 예외처리를 최대한 많이 대응 하였습니다. 이를 고쳐나가는 과정에서 예상치 못한 **사이드 이팩트**를 만나게 되면서 힘든 점이 있었습니다. 그때 빛을 발한 것이 처음 프로그래밍 언어를 배우면서 했던 제어문과 관련된 문제들을 풀면서 겪었던 경험이었습니.

코딩 테스트 문제와 여러 자료를 통해서 풀어본 문제들에서 고민했던 여러 **알고리즘**들이 유용하게 사용되었습니다.

에러상황을 만났을때는 **에러 메시지를** 집중해서 분석하였습니다. 처음에는 에러 메시지가 무엇을 뜻하는지 이해하기 너무 힘들고 외계어 같았습니다. 하지만, 여러 에러와 해결 과정을 겪으면서, 에러 메시지만큼 **명확한 증거가** 없다는 것을 점점 알게 되었고, 에러 메시지만을 통해서 해결 할수 있는 에러들이 점점 늘어 났습니.

디버깅하는 과정에서는 항상 **input 과 output**을 중점을 주었습니다. 어떤 데이터가 전달이 되어서 어떤 결과가 나오는지의 과정을 하나하나 좁혀 가면서, 버그 상황을 찾아 갔습니.. 두리몽실하게 넓게 생각 할 때 보다 훨씬 명확히 해결 할 수 있었습니다. 또한 **logback과 이클립스 디버깅**을 최대한 활용하였습니다. 혼자 진행하다 보니 모든 것을 내가 다 감당해야해서 힘들기는 했지만, 도리어 그만큼 더 얻은 것이 많은 프로젝트 였습니.

## 프로젝트 진행 느낀점 sample

각각의 기능을 하나의 독립된 **모듈형태**로 생각하고, 작은 단위부터 완전하게 만들어 가자라고 생각했습니다. 처음 **설계 과정에서** 큰 그림을 그릴 때 힘든 점이 있었지만, 기능 하나 하나를 쪼개서 구상 하다보니 조금씩 큰 그림이 완성되어 가는 것을 느꼈습니다. 특히, Spring 프레임워크의 **흐름을 파악**하는 것에 가장 큰 중점을 두었습니다. 서브릿을 통해서도 웹서비스를 경험해 보았지만, Spring을 통한 프로젝트 경험은 또 다른 영감을 주었습니다. Controller Service Reposotory 로 이어지는 구조는 단순히, 기능을 개발하는 것뿐만 아니라 구조적인 부분이 개발에 어떠한 이점을 주는지 체감 할 수 있었습니다.

여러 Database 의 여러 테이블의 데이터를 조합하여, 하나의 독립된 데이터로 만들어 가는 과정도 어려움이 있었습니다. 차근차근 필요한 데이터를 정리하여 **도메인을 설계** 하고 이를 Class로 만들어서 사용하였습니다. 이 과정을 반복하다 보니 이제 자연스럽게 자료의 구조가 머리에 그려지는 것을 느낄 수 있었습니다.

실제 사용자가 사용한다는 생각으로 예외처리를 최대한 많이 대응 하였습니다. 이를 고쳐나가는 과정에서 예상치 못한 **사이드 이팩트**를 만나게 되면서 힘든 점이 있었습니다. 그때 빛을 발한 것이 처음 프로그래밍 언어를 배우면서 했던 제어문과 관련된 문제들을 풀면서 겪었던 경험이었습니다. 코딩 테스트 문제와 여러 자료를 통해서 풀어본 문제들에서 고민했던 여러 **알고리즘**들이 유용하게 사용되었습니다.

에러상황을 만났을 때는 **에러 메시지를** 집중해서 분석하였습니다. 처음에는 에러 메시지가 무엇을 뜻하는지 이해하기 너무 힘들고 외계어 같았습니다. 하지만, 여러 에러와 해결 과정을 겪으면서, 에러 메시지만큼 **명확한 증거**가 없다는 것을 점점 알게 되었고, 에러 메시지만을 통해서 해결 할 수 있는 에러들이 점점 늘어 났습니다.

디버깅하는 과정에서는 항상 **input 과 output**을 중점을 주었습니다. 어떤 데이터가 전달이 되어서 어떤 결과가 나오는지의 과정을 하나하나 좁혀 가면서, 버그 상황을 찾아 갔습니다. 두리몽실하게 넓게 생각 할 때 보다 훨씬 명확히 해결 할 수 있었습니다. **logback과 이클립스 디버깅**을 최대한 활용하였습니다.

문제 상황이 발생했을 때, 단순히 눈앞에 보이는 문제를 해결하기 보다는 근본이 되는 문제를 찾기에 주안점을 두었습니다. 눈앞에 문제만 if문 몇 줄로 처리 할 수도 있었지만, 이러한 부분이 쌓이면서 작은 문제에도 수많은 조건문이 필요 했습니다. 그러한 문제점을 깨닫고, 근본이 되는 문제를 찾아 내고 해결하려고 하였습니다. 찾고 해결하는데는 시간이 다소 소요 되었지만, 결과적으로는 전체적인 개발 시간에 도움이 된다는 것을 느꼈습니다.

개발 과정에서 기능에 주안점을 두고 진행하면서 어지럽혀져 있는 코드를 보면서, 이러한 코드들이 추후에 유지 관리에 문제를 일으킬 수 있다고 생각되어서 최대한 간결하고 명확하게 **리팩토링 과정**을 통해서 다시 한번 전체 구조를 다듬는 작업을 하였습니다. 그 결과는 너무나 만족 스러웠고, 이어서 진행하는 부분에서도 개발 속도가 훨씬 빨라지는 것을 확인 할 수 있었습니다.

## 프로젝트 진행 느낀점 sample

구글 검색을 통한 문제 해결 과정도 많이 늘게 되었습니다. 처음에는 한글 블로그 위주로 검색을 하고 찾아 갔습니다. 하지만, 그 과정에서 내가 가진 문제를 해결하기에는 한계가 있었고, 결국 **stackoverflow**의 영어자료도 참조하게 되었습니다. 영어 독해 능력이 뛰어나지 못해서 처음에는 조금 어려움을 겪었지만, 익숙한 **코드 내용과 에러 메시지**를 위주로 분석하다 보니 조금씩 이해하는 시간이 빨라 졌습니다. 확실히 영어 자료까지 포함해서 문제 상황을 파악을 하니 자료가 훨씬 많고, 문제 상황을 해결하는데도 도움이 되었습니다. 또한 인터넷의 자료를 너무 신뢰 하였을때의 문제점도 있는 것을 파악하고, 그 자료를 직접 확인해서 **검증**하고, 여러 검색한 자료로 **크로스 체크**도 하면서 명확하게 검증하면서 사용하였습니다.

기능을 만들어 가는 과정에서 기존에 간단한 실습을 할 때는 하나의 방법으로 해결이 가능했지만, 프로젝트를 진행하면서 하나의 기능을 만드는데 무수히 많은 방법이 있다는 것을 느꼈습니다. 바로 떠오르는 방법으로 해결하려고 했을 때, 나중에 문제가 발생해서 **같은 일을 몇 번 반복해야** 하는 일이 있었습니다. 그 경험 이후로 여러 방법이 있을 것으로 예상 될 때는 쉽게 선택하지 않고 구조와 장단점을 고려해서 그에 맞는 방법을 선택하도록 고민을 많이 했습니다.

여러 기능을 만들다 보니 **오픈소스**를 이용하는 경우가 생겼습니다. 오픈소스를 사용할 때는 일단 **라이선스** 확인을 통해서 사용 가능한 범위를 확인하였습니다. **Apache, MIT, BSD** 등 자주 사용되는 자유로운 라이선스가 있다는 것도 알게 되었습니다. 또한 안정성확보를 위해서 사용자가 많은 것을 위주로 선택하였습니다. 그리고 github를 운영하고 있는 경우에는 **최근 commit 로그**를 확인해서 최신화가 잘되고 있는지 확인하였습니다.

새로운 문법이나, 라이브러리, 프레임워크를 사용할 때는 처음에는 사용법 위주로 사용하였습니다. 최대한 **공식 문서**를 확인해서 간단한 예제를 통해서 결과를 확인하고, 어떤 형태의 **input 과 output**으로 구성이 되는지 기본을 파악하였습니다. 사용법에 익숙해 지고 나서는 **내부 구조**를 이해하려고 노력하였습니다. 어떤 흐름으로 해당하는 결과가 나오고 내부적으로 어떤 로직이 구성되어 있는지 파악하려고 하였습니다. 사용법만 알고 사용할 때 는 문제 상황이 발생하면 아무것도 못하고 생각을 확장할 수 없었던 반면, 내부 구조에 대한 흐름을 파악한 이후로는 해결 할 수 있는 여러 방향이 머리에 떠오르면서 이런 저런 시도를 통해서 문제를 해결할 수 있었습니다. 심지어, 기존 오픈소스에 없던 기능이 있어서 아쉬웠을 때 소스를 분석해서 아주 간단한 수정이었지만, 직접 기능을 만들어서 사용하기도 하였습니다.

프로젝트를 시작할 때 기획을 하고 구조를 **설계**하는 과정에 큰 도움이 되었습니다. 특히 Databse 와 url 설계 과정에서 최대한 많은 부분을 고려하면서 놓치는 부분이 없도록 노력하였습니다. 그러한 과정에서 자연스럽게, 프로그램의 구조도 머리 속으로 그려지게 되었고 프로젝트를 시작하는 것에 대한 두려움이 많이 사라졌습니다. 또한, 기존에 그때 그때 필요할 때 마다 만들어 썼을때는 나중에 문제가 생겨서 두번 세번 다시 작업해야 하는 경우가 있었습니다. 설계를 통해서 진행했을 때는 이러한 경우가 현격히 줄어 들었고 결국 개발 기간도 확실히 줄어 드는 것을 느꼈습니다.

개발 진행상황을 최대한 기록하기 위해서 애썼습니다. Github와 연계된 **zenhub** 라는 툴을 통해서 개발이 필요한 기능들을 각각의 **이슈**로 만들어서 등록하였습니다. 개발이 필요한 부분을 미리 기록해 놓고 이를 확인하고 처리하는 방식으로 진행했습니다. 또한, 진행 과정에서 있던 문제점이나 막히는 부분들 그리고 해결 방안들을 기록해 놓았더니, 비슷한 상황이었을 때 도움이 많이 되었습니다. 특히, zenhub는 **git의 commit 메시지로** 기록을 할수 있어서 사용성 면에서 매우 좋았습니다.

# 기술 정보

마론달 그램

Github

[https://github.com/dulumary/spring\\_marondalgram\\_0817](https://github.com/dulumary/spring_marondalgram_0817)

## 기술 스택

Tomcat, Mysql

Spring framework, Spring boot, Mybatis, JSTL

Bootstrap, jQuery

Java, javascript, html

## 라이선스

Tomcat ([Apache License 2.0](#)) , Mysql ( GPLv2 or proprietary)

Spring framework ([Apache License 2.0](#)) , Mybatis ([Apache License 2.0](#))

Bootstrap (MIT License), jQuery (MIT License)

이미지 출처 : pixabay

# 프로젝트 진행 단계

마론달 그램

기획

DB 설계

URL 설계

구현



## 기획

## 마흔다섯 번째 그림

## Ovenapp.io 서비스를 사용하여 페이지별 기획서 작성

<https://ovenapp.io/project/MEoDuSsYW8NUBUR7wH788CN5yck53bCz#IMbQe>

## 로그인

The image displays a mobile application interface for 'Marondalgram'. At the top, a grey header bar contains the text 'Marondalgram' in white. Below this, a light grey rounded rectangle serves as a container for the login and sign-up options. Inside this container, there are two input fields: the first is labeled 'Username' with a person icon, and the second is labeled with four dots (password) with a key icon. Below these fields is a blue button with the white text '로그인' (Login). Underneath the button is a blue, underlined link that reads '회원가입' (Sign Up). At the bottom of the screen, a thin grey horizontal line separates the app interface from the footer, which contains the text 'Copyright © marondalgram 2021'.

## 회원가입

# Marondalgram

Username

\*\*\*\*

비밀번호 확인

이름

이메일

회원가입

중복확인

---


Copyright © marondalgram 2021

## 타임라인

Marondalgram

김인규님 환영합니다.로그아웃





내용을 입력해주세요



업로드

hagulu

...

♡ 좋아요 11개

hagulu 동그라미를 그려 봤어요. 심심해요 놀아 주세요. 이라고 살아요.

댓글

marobiana 내가 놀아 줄게 나와라

bada 냐아아앙~

내용을 입력해주세요

게시

## 추가 메뉴

# Database 설계

## 마론달 그램

SpreadSheet 를 통해서 데이터 베이스 테이블 설계  
각 컬럼의 타입, Null 여부, auto\_increment 여부, 설명 항목 등 작성  
이를 기반으로 테이블 생성 쿼리 작성  
[https://docs.google.com/spreadsheets/d/1JVVnXdV3jcEpb\\_xrVFK3EpYsRH0w\\_VbrGuHF317p9dw/edit#gid=1563039968](https://docs.google.com/spreadsheets/d/1JVVnXdV3jcEpb_xrVFK3EpYsRH0w_VbrGuHF317p9dw/edit#gid=1563039968)

### User

회원 관리를 위한 테이블

컬럼명	타입	NULL 가능 여부	auto_increament	설명
id	int	N	Y	primary key
loginId	varchar(16)	N	N	사용자 ID
password	varchar(32)	N	N	비밀번호, 암호화한 결과
name	varchar(16)	N	N	사용자 이름
email	varchar(64)	N	N	이메일 주소
createdAt	timestamp	N	N	생성 날짜
updatedAt	timestamp	N	N	수정 날짜

# URL 설계

## 마론달 그램

SpreadSheet 를 통해서 URL 설계  
View URL 과 API URL 을 구분해서 설계  
View URL 은 url 과 parameter 정보 작성  
API URL 은 url, method, parameter, response data 정보 작성  
[https://docs.google.com/spreadsheets/d/1JVVnXdV3jcEpb\\_xrVFK3EpYsRHOW\\_VbrGuHF317p9dw/edit#gid=1563039968](https://docs.google.com/spreadsheets/d/1JVVnXdV3jcEpb_xrVFK3EpYsRHOW_VbrGuHF317p9dw/edit#gid=1563039968)

### View URL

제목	url	parameter	설명
타임라인	/post/timeline		타임라인 화면
로그인	/user/signin_view		로그인 화면
회원가입	/user/signup_view		회원가입 화면

# URL 설계

마론달 그램

## API URL

### 1) 글쓰기 후 저장


URL : /post/create			
method :post			
parameter			
파라미터 명	데이터 타입	NULL 여부	설명
content	String	N	글 내용
file	MultipartFile	Y	첨부 이미지 파일
응답값 (json)			
성공시	{"result":"success"};		
실패시	{"result":"fail"};		

# 타임라인


## 마론달 그램


Marondalgram

김인규님 [로그아웃](#)



업로드

 김인규



♡ 좋아요 11개

김인규 와아앙 바다다

댓글

바다 너 뭐하냐?

김인규 ㄹㄹㅇㄹㄹㅇㄹㄹㅇㄹㄹ

김인규 ㅋㅋㅋㅋㅋ

김인규 댓글 확인

김인규 ...

소개	<ul style="list-style-type: none"><li>• 사용자가 작성한 feed 가 카드 형식으로 보여준다.</li><li>• 상단 입력창을 통해서 내용을 입력하도록한다.</li><li>• 이미지 파일 첨부는 input file 을 활용한다.</li><li>• 단, 파일 첨부 버튼을 예쁘게 하기 위해서 기존 input file은 숨기고 아이콘으로 input file click 이벤트를 받게해서 정리 하였다.</li><li>• Post 마다 의 댓글과 좋아요 여부등을 하나의 객체로 저장하기 위해서 클래스를 설계하고 적용하였다.</li><li>• Ajax를 통해서 글입력, 댓글, 좋아요 기능을 구현하였다.</li><li>• 글을 쓴 사용자만 삭제가 가능하도록 버튼을 노출하였다.</li></ul>
troubleshooting	<ul style="list-style-type: none"><li>• Post와 댓글 정보를 매칭 시키기 위해서 모든 댓글 정보를 다 가져와서 처리 하였다. 이렇게 할 경우 매우 많은 수의 댓글에 대해서 처리에 지연이 생긴다는 것을 인지하였다. 이 문제를 피하기 위해서, Post 마다 댓글을 조회 하도록 변경하고, 이를 묶어서 저장할 수있는 클래스를 만들어서 객체화 하고 이를 이요해서 화면에 보여주도록 하였다.</li></ul>
보완점	<ul style="list-style-type: none"><li>• 실제 instagram 에서 처럼 스크롤을 통한 데이터를 추가하는 방식 적용이 필요</li><li>• 댓글 접기 기능 추가 필요</li><li>• 여러 이미지 업로드및 보여주기 기능</li></ul>



감사합니다!  
잘 부탁드립니다!

2030 - 2035 PORTFOLIO

CONTACT  
Sample@naver.com  
010 1234 5678

