TOON PICK

웹툰 랭킹 웹프로젝트

portfolio Side Project

gitHub
https://github.com/Yangwonho

Blog
https://blog.naver.com/zizonwanna

Phone
010-4015-8685

목차

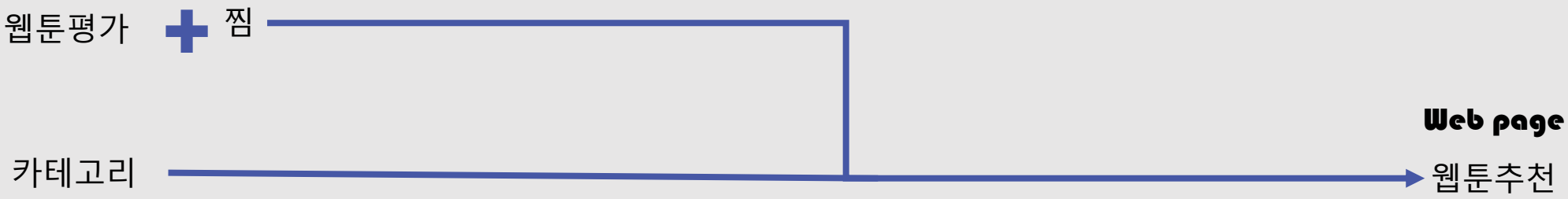
1. 기획 및 설계 - 핵심기능

평가 및 랭킹

웹툰 평가를 기준으로 순위를 매기기 위한 평가 방식을 도입합니다.
간단한 별점 체계(1~5점) 로 이루어지며, 사용자는 한줄평 및 리뷰를 작성할 수 있습니다.
평가 방식은 유저 랭킹과 웹툰 순위, 두 가지 랭킹으로 구분됩니다.

찜 및 추천

웹툰 이용자의 개인 맞춤형 경험을 강화하기 위해 찜 및 추천 기능을 도입합니다.
이 기능을 통해 사용자는 관심 있는 웹툰을 쉽게 관리하고, 취향에 맞는 새로운 웹툰을 추천받을 수 있습니다.



1 기획 및 설계 - 컨셉정의

타겟 유저

기능에 대한 방안 / UX,UI / 디자인 컨셉을 선정하기 위해 타겟 유저를 정의하였습니다. 선정한 주요 타겟은 웹툰 주 소비층인 20대 남성입니다.(콘텐츠진흥원 발췌)

간편 & 직관

디자인 컨셉은 간편함과 직관적임을 중점적인 컨셉으로 잡았습니다.
웹툰 평가 시스템에 있어서도 잘 어울리며 주 타겟층에도 잘 맞는 컨셉이라고 생각하여 선정하였습니다.

LOGO

컨셉

웹툰 을 Pick 한다는 뜻의 툰 픽(Toon Pick)
둥글둥글한 이미지로 만화와 관련됨을
직관적으로 보이도록 하였고 기존 다른 웹툰
사이트와 다른 주조색을 사용하였습니다.

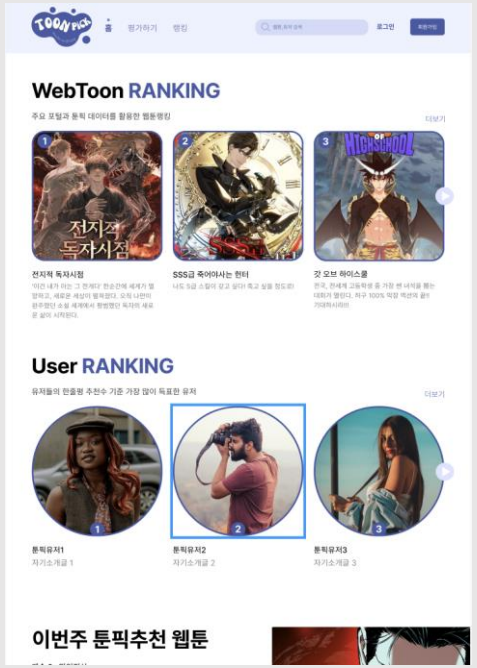
기능적으로도 간편함을 중점적으로 기능구현을
할 예정입니다.

1. 기획 및 설계 - 웹페이지 설계도

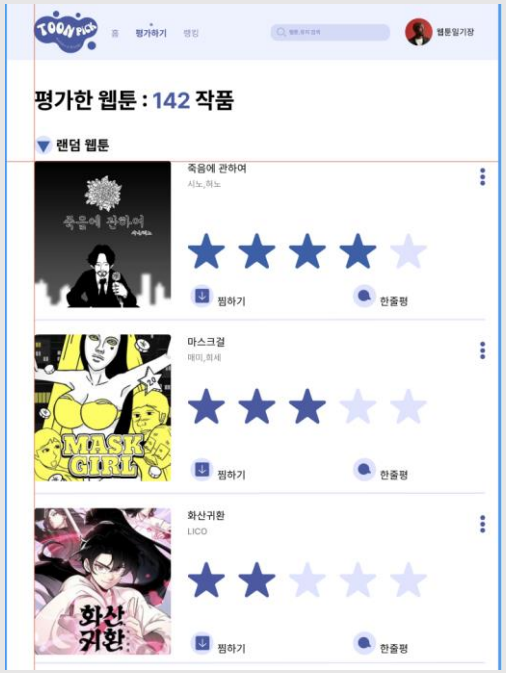
웹페이지 설계도

디자이너와의 협업을 고려하여 피그마로 제작한 웹페이지 설계도 입니다.
더 보기 혹은 자세히 보기를 원하시면 링크를 통해 확인 할 수 있습니다.
링크 : https://www.figma.com/design/PrEJUNoWKnzuqw0QBNuqa5/Untitled?node-id=0-1&t=ox1bVguCSuA71gJJ-1

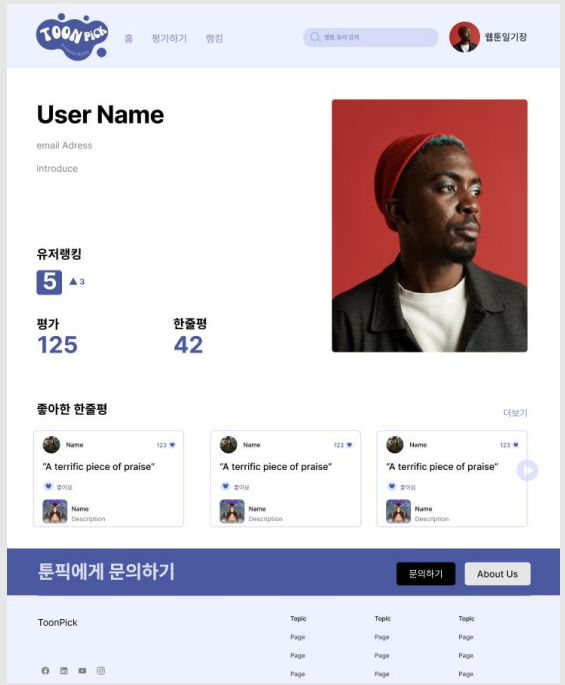
메인페이지



평가하기

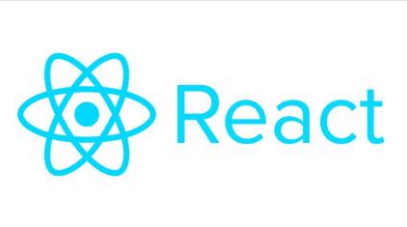


마이페이지



2. 기술 스택 - Tech Stack

FRONT-END



React
React Router (페이지 이동)
Axios (API 통신)
Tailwind CSS (스타일링)

BACK-END

Spring Boot
Spring Security
(로그인/회원 인증)
JPA (DB 연동)
REST API 설계

DATA-BASE

ER-Diagram
Query

TOOLS

IntelliJ (백엔드)
VSCode (프론트엔드)
Figma (UI 설계)
GitHub (버전 관리)
Postman (API 테스트)

2. 기술 스택

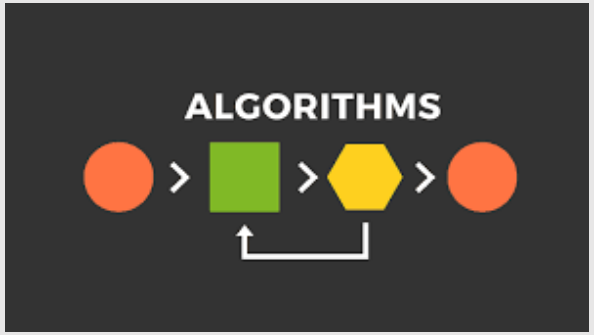
- 구현 포인트

FRONT-END

1 모바일 UI 최적화
2 Tailwind CSS로 일관된 스타일
3 TypeScript 도입
4 SPA 기반 라우팅 처리

BACK-END

1 소셜 로그인 연동 (OAuth2)
2 REST API 설계 및 계층 분리
3 랭킹 계산 알고리즘 구현
4 사용자 맞춤 추천 시스템



3 데이터베이스 - MySQL

선정이유

- 속도 빠르고 경량화됨 (특히 SELECT 성능 우수)
- 문서와 커뮤니티 방대 (에러 해결 쉽고 자료 많음)
- PHP/LAMP 환경에 최적화
- 다양한 호스팅과 클라우드에서 기본 지원

기존에 사용하던 oracle , MariaDB, PostGreSQL 과의 차이점을 경험

ERD

ER-Diagam

비즈니스 에서는 ERWIN 을 사용하였으나 – 사이드 프로젝트에서 사용하기에 물리모델 기준인 dbdiagram.io 사용하였습니다.

링크 : https://dbdiagram.io/d/67ef97394f7afba1845695b0

4 기능정의 - 기능정의서

웹툰 랭킹 시스템

웹툰 랭킹 시스템 기능 정의

기본 설계 / 웹페이지 설계도 / ERD 를 취합하고
기능을 추출하여 웹 페이지 기능을 정의
하였습니다.

1. 회원 관련 기능
2. 웹툰 관련 기능
3. 평가 및 랭킹 기능
4. 리뷰 기능
5. 찜 및 추천 기능
6. 문의(Q&A) 기능
7. 파일 관리 기능

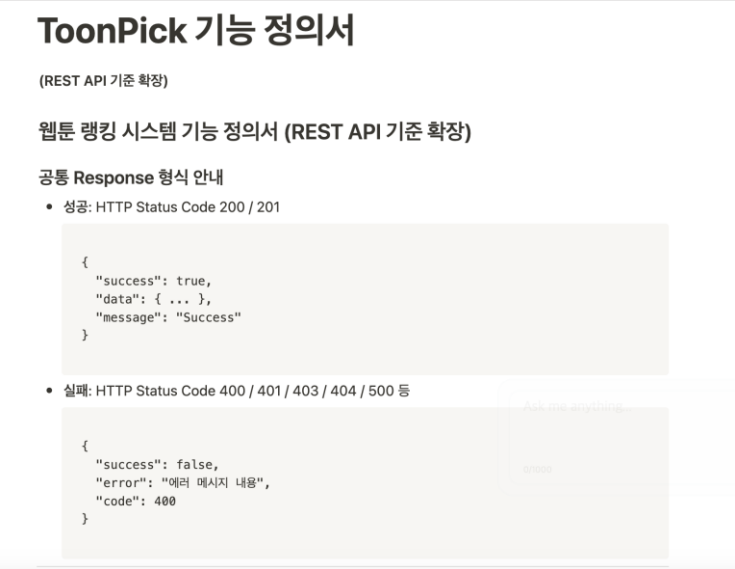
```
웹툰 랭킹 시스템 ToonPick 기능 정의서
================================================================================
1. 회원 관련 기능
--------------------------------------------------------------------------------
1.1 회원가입

설명: 사용자는 이메일/비밀번호 또는 소셜 로그인(카카오, 네이버,
페이스북) 방식으로 회원가입 가능

관련 테이블: user

함수명: signUp

request: { email: String, password: String, loginType:
String, socialId: String }

response: 성공 { userInfo }, 실패 { errorMessage }

--------------------------------------------------------------------------------
1.2 로그인
```

 # 상세한 기능은 기능정의서에서 표현 (txt파일)
 # Link :
https://github.com/Yangwonho/Yangwonho/blob/main/docs/ToonPick_Feature_TechDoc.pdf

4 기능정의 - 기능정의서

웹툰 랭킹 시스템

Notion 기능정의서



ToonPick 기능 정의서

(REST API 기준 확장)

웹툰 랭킹 시스템 기능 정의서 (REST API 기준 확장)

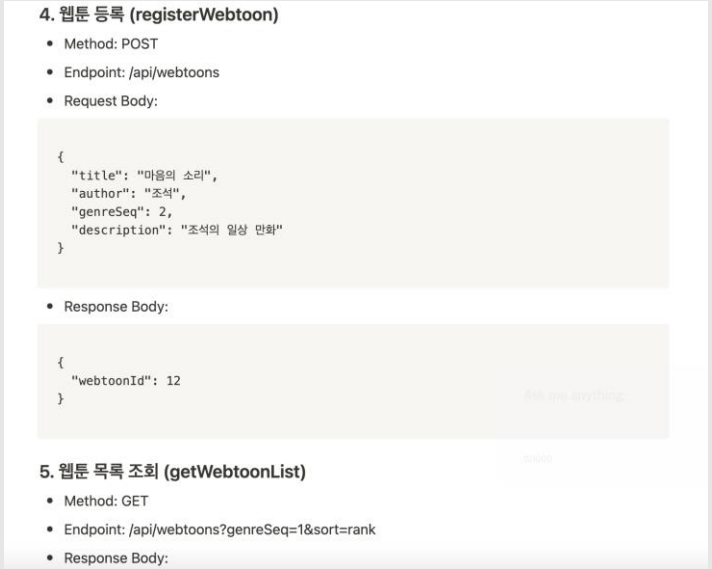
공통 Response 형식 안내

- 성공: HTTP Status Code 200 / 201

```
{
  "success": true,
  "data": { ... },
  "message": "Success"
}
```

- 실패: HTTP Status Code 400 / 401 / 403 / 404 / 500 등

```
{
  "success": false,
  "error": "에러 메시지 내용",
  "code": 400
}
```



4. 웹툰 등록 (registerWebtoon)

- Method: POST
- Endpoint: /api/webtoons
- Request Body:

```
{
  "title": "마음의 소리",
  "author": "조석",
  "genreSeq": 2,
  "description": "조석의 일상 만화"
}
```

- Response Body:

```
{
  "webtoonId": 12
}
```

5. 웹툰 목록 조회 (getWebtoonList)

- Method: GET
- Endpoint: /api/webtoons?genreSeq=1&sort=rank
- Response Body:

요청 파라미터

기능명	파라미터	타입	필수 여부	설명
회원가입	email	String	✅	이메일 주소
회원가입	password	String	✅	비밀번호
회원가입	loginType	String	✅	로그인 유형 (local, kakao 등)
회원가입	socialId	String	❌	소셜 로그인 ID
로그인	email	String	✅	이메일 주소
로그인	password	String	✅	비밀번호
웹툰 등록	title	String	✅	웹툰 제목
웹툰 등록	author	String	✅	작가 이름
웹툰 등록	genreSeq	Number	✅	장르 고유 번호
웹툰 등록	description	String	❌	설명
별점 평가	webtoonSeq	Number	✅	평가 대상 웹툰 ID
별점 평가	userSeq	Number	✅	평가한 유저 ID
별점 평가	score	Number	✅	1~5점 사이 정수

 # 자세한 기능은 기능정의서에서 표현 notion Online 작업

 # Link : https://www.notion.so/ToonPick-1d1fcb52c0bf805d8894cfa3545de6aa

5 작업방식 - 개발가이드

버전관리 및 이슈트래킹

GitHub

버전 관리 와 이슈 트래킹은 깃허브를 사용할 예정이며 기본적인 작업 지침서를 작성하여 업무상 혼선이 일어나지 않도록 대비 하였습니다.

작업지침서

작업 방식 가이드

✅ 버전 관리 전략: Git Flow

브랜치	용도
`main`	운영 배포용 브랜치 (배포된 안정된 코드만 존재)
`develop`	통합 개발 브랜치 (기능 브랜치들을 병합)
`feature/*`	기능 단위 개발 브랜치 (예: `feature/login`)

- 커밋 메시지 컨벤션 예:
  - `feat` : 기능 추가
  - `fix` : 버그 수정
  - `docs` : 문서 수정
  - `refactor` : 리팩토링

✅ 이슈 관리 방식: GitHub Issues

요소	설명
이슈 유형 라벨	`feature`, `bug`, `refactor`, `question`, `priority` 등
이슈 템플릿	작성자, 내용, 완료 조건 포함

> 이점: 협업 시 투명한 업무 분배, 일정 추적, 작업 히스토리 관리에 용이

✅ 코딩 스타일 가이드

🔷 백엔드 (Spring Boot - Java)

- 네이밍: camelCase
- 클래스 구성: `Controller` → `Service` → `Repository`
- DTO, Entity, Request/Response 구분 철저
- 유효성 검증: `@Valid`, `@NotNull`, `@Size` 적극 활용
- 예외 처리: `@ControllerAdvice` + 전역 예외 핸들러
- DB 트랜잭션: `@Transactional` 필수 사용

 # 프로젝트 툰코 개발가이드는 notion 으로 자세하게 확인 가능합니다.
 # Link : https://www.notion.so/ToonPick-1d1fcb52c0bf805cb96cd057aa4480e3

5 작업방식 - 배포

배포계획

항목	선택 내용
프론트엔드	React (Vercel 또는 Netlify 사용 예정)
백엔드	Spring Boot (✅ Railway 사용 예정)
DB	MySQL (✅ Railway 내에서 함께 구성)
목표	비용 없이 무료 배포 (CI/CD 포함)
상태	배포 계획 수립 완료, 실행은 추후 진행 예정

Front-end

Vercel

Back-end

Railway

자동화

GitHub Actions

사이드 프로젝트이기에 비용발생이 안되는 쪽으로 구현할 예정이며.
로컬 작업후 Vercel 과 Railway 를 통하여 배포 계획을 수립하였습니다.

그리고 빌드/테스트/배포 까지의 CI/CD 파이프라인의 자동화를 위해 GitHub Actions 를 사용할 예정입니다.