포트폴리오 제작 웹

- 전체 구조

[React] ← REST API → [Spring Boot] ←→ [MySQL / PostgreSQL]

↑

(OpenAI GPT API)

- 1. 프론트엔드: React

주요 페이지 구성

| **페이지** | **설명** |
| --- | --- |
| / | 홈 + 소개 |
| /form | 사용자 정보 입력 폼 (이름, 소개글, 기술 스택, 프로젝트 등) |
| /preview | 선택된 템플릿으로 실시간 미리보기 |
| /portfolio/:id | 완성된 포트폴리오 미리보기 (퍼블릭 공유용) |

**상태 관리**

* 전역 상태: Context API or Zustand 사용
* Form 정보 → 상태 저장 → API로 전송

**프론트 기술 스택 제안**

* React + Vite
* TailwindCSS (빠르고 예쁨)
* Axios (Spring과 통신)
* React Router

- 2. 백엔드: Spring Boot

API 엔드포인트 예시

| **메서드** | **경로** | **설명** |
| --- | --- | --- |
| POST | /api/portfolio | 사용자 포트폴리오 생성 |
| GET | /api/portfolio/{id} | 특정 포트폴리오 조회 |
| POST | /api/gpt/summary | GPT API를 호출하여 설명 자동 생성 (선택) |
| GET | /api/template | 템플릿 목록 조회 |

**기술 스택**

* Spring Boot (Java 17+)
* JPA + Hibernate
* DB: MySQL / PostgreSQL
* Cross-Origin 설정 (CORS 허용 필요)
* GPT API 연동 (RestTemplate or WebClient)

- 3. 배포 흐름 예시

| **구성 요소** | **배포 방법** |
| --- | --- |
| React 프론트 | Vercel, Netlify |
| Spring 백엔드 | Railway, Render, EC2 |
| DB | Railway DB, PlanetScale, RDS |
| 도메인 | 포트폴리오.myname.com 가능 |

- 4. 개발 순서 추천

| **단계** | **내용** |
| --- | --- |
| 1단계 | ERD + API 설계 → Spring Boot 서버 구축 |
| 2단계 | React 입력 폼 구성 및 상태 연결 |
| 3단계 | POST API로 포트폴리오 데이터 저장 |
| 4단계 | /preview에서 템플릿 기반 UI 생성 |
| 5단계 | 퍼블릭 링크 /portfolio/:id 구현 |
| 6단계 | (선택) GPT API 붙여서 설명 자동화 |
| 7단계 | 배포 + PDF 저장 기능 추가 |

**✅ 코드/설계 샘플 도와드릴 수 있음**

필요하시면 다음 중 하나부터 도와드릴 수 있어요:

* React 입력 폼 구조 + 상태관리 코드
* Spring Boot API 컨트롤러 예시 (@RestController)
* GPT API 연동 샘플 (RestTemplate)
* 퍼블릭 포트폴리오 URL 구조 설계
* 프로젝트 전체 디렉토리 구조

전체 디렉토리 구조 (프론트 + 백엔드 분리형)

portfolio-ai-app/

├── frontend/

│ ├── pages/

│ │ ├── Home.tsx

│ │ ├── UploadJobPost.tsx // 채용공고 입력 화면

│ │ ├── AIRebuildResult.tsx // 리모델링된 포트폴리오 미리보기

│ ├── components/

│ │ ├── JobPostInputForm.tsx

│ │ ├── HighlightedPortfolio.tsx

│ └── utils/

│ └── extractKeywords.ts // 키워드 추출용 GPT or rule-based

│

├── backend/ (Java Spring Boot)

│ ├── controllers/

│ │ ├── JobPostController.java

│ │ └── GPTServiceController.java

│ ├── services/

│ │ ├── JobPostAnalysisService.java

│ │ └── PortfolioRebuilderService.java

│ └── models/

│ ├── JobPost.java

│ └── PortfolioData.java

│

├── db/

│ ├── UserPortfolio.json or PostgreSQL

│ └── JobPostKeywords.json

portfolio-generator/

├── backend/ # Spring Boot 백엔드

│ ├── src/

│ │ ├── main/

│ │ │ ├── java/com/example/portfolio/

│ │ │ │ ├── controller/ # REST API 컨트롤러

│ │ │ │ ├── dto/ # 요청/응답 DTO

│ │ │ │ ├── entity/ # JPA 엔티티

│ │ │ │ ├── repository/ # JPA 레포지토리

│ │ │ │ ├── service/ # 비즈니스 로직

│ │ │ │ └── PortfolioGeneratorApplication.java

│ │ │ └── resources/

│ │ │ ├── application.yml # 환경 설정 (DB, CORS, GPT API 등)

│ │ │ └── templates/ # (선택) HTML 템플릿 (PDF용)

│ └── build.gradle / pom.xml

│

├── frontend/ # React 프론트엔드

│ ├── public/ # 정적 자원 (favicon, index.html 등)

│ ├── src/

│ │ ├── assets/ # 이미지, 아이콘 등

│ │ ├── components/ # 재사용 가능한 UI 컴포넌트

│ │ ├── pages/ # Form, Preview, Portfolio 등 페이지별 컴포넌트

│ │ ├── services/ # API 호출 함수 모음 (Axios)

│ │ ├── contexts/ # 상태 관리 (ex. React Context or Zustand)

│ │ ├── utils/ # 템플릿 변환 함수, 헬퍼 함수

│ │ ├── App.tsx

│ │ └── main.tsx

│ ├── tailwind.config.js # 스타일 설정

│ ├── package.json

│ └── vite.config.ts

│

├── README.md

├── .gitignore

└── LICENSE (선택)

주요 연동 포인트

| **연결 요소** | **설명** |
| --- | --- |
| frontend/src/services/api.ts | Spring Boot 백엔드와 Axios 통신 |
| CORS 설정 | application.yml에 프론트 URL 허용 |
| 포트폴리오 데이터 저장 | POST /api/portfolio → DB 저장 |
| 템플릿 렌더링 | 프론트에서 템플릿 선택 후 HTML 구성 |
| 퍼블릭 보기 | GET /api/portfolio/{id} → React에서 /portfolio/:id 페이지로 출력 |
| PDF 출력 (선택) | 백엔드에서 HTML → PDF 변환 (Flying Saucer 등 사용) |

**✅ 📄 입력 폼에 포함할 필드**

**🧍 기본 정보**

* 이름 (name)
* 직무 / 역할 (role)
* 소개글 (bio)
* GitHub, Email, LinkedIn (links)

**🛠 기술 스택**

* 문자열 배열 형태 (techStack[])  
  예: React, Spring Boot, MySQL 등

**📦 프로젝트들 (여러 개 입력 가능)**

* 제목 (title)
* 설명 (description)
* 사용 기술 (techStack[])
* GitHub 주소 (github)
* 배포 주소 (demo)
* 대표 이미지 (선택) (imageUrl)

추가 예정

 🔄 "입력한 항목 수정하기" 기능

 📤 데이터 저장 (예: localStorage 또는 서버 전송)

 🎨 Tailwind로 디자인 커스터마이징

**단계별 개발 순서 추천**

1. **기존 React 입력폼 + 미리보기 구현**
2. Spring Boot로 **REST API 서버 구축**
3. PostgreSQL 또는 MongoDB 연동
4. API 통신 + DB 저장 기능 추가
5. GPT API 연동
6. 사용자 인증/관리 추가
7. 배포 (프론트: Vercel, 백엔드: Render or EC2)

**3. 작동 순서 (Flow)**

1. **사용자 입력**
   * 채용공고 URL 붙여넣기 or 공고 텍스트 직접 입력
2. **공고 분석**
   * [백엔드] JobPostAnalysisService
     + GPT API 또는 크롤링+Rule기반 방식으로 기술 키워드 추출
     + ex) ["React", "Java", "Leadership", "CI/CD", "AWS"]
3. **포트폴리오 리모델링**
   * 사용자의 원래 포트폴리오 데이터 중 공고와 관련 높은 프로젝트 / 기술 우선 정렬
   * GPT로 요약/리프레이징도 가능 (e.g. “Implemented” → “Led implementation for...”)
   * 중요도 낮은 정보는 하단으로 이동 or 생략
4. **사용자 미리보기 제공**
   * AIRebuildResult.tsx에서 재구성된 포트폴리오 출력
   * "원본으로 보기", "수정하기", "이력서로 내보내기" 등 기능 제공

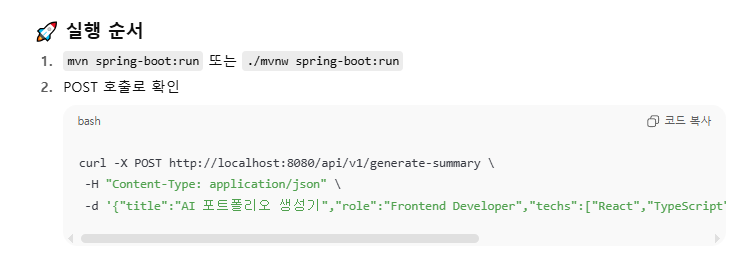
핵심기능

| **기능** |  | **설명** |
| --- | --- | --- |
| 채용공고 URL 또는 텍스트 입력 |  | 링크 붙여넣기 or 텍스트 복사 |
| 키워드 추출 및 가중치화 |  | GPT or Rule로 기술/역할 추출 |
| 포트폴리오 재정렬 |  | 추출된 키워드 기반 필드 우선 정렬 |
| 문장 리프레이징 |  | GPT로 자연스러운 표현 보정 |
| 하이라이팅 기능 |  | 매칭된 기술스택 강조 |
| 다운로드/링크 공유 |  | 맞춤 포트폴리오 내보내기 기능 |

**5. 추가 확장 가능성**

* 📑 채용공고 히스토리 저장 및 추천 기반 모델 훈련
* 🧩 유사 포지션 자동 매칭 (ex. “이런 공고에도 잘 어울립니다”)
* 📄 이력서 자동 생성 (.pdf)
* 🧠 자가 분석 피드백 (부족한 기술스택 or 프로젝트 제안)





curl -X POST http://localhost:8080/api/v1/generate-summary \

-H "Content-Type: application/json" \

-d '{"title":"AI 포트폴리오 생성기","role":"Frontend Developer","techs":["React","TypeScript"],"bullets":["Tailwind UI 구현","OpenAI API 연동"]}'

예시

<https://hyunseob.github.io/resume/>

<https://wonny.oopy.io/>

데비콘

<https://devicon.dev/>

심플아이콘

<https://simpleicons.org/>

백엔드 실행

cd portfolio-backend

mvn -U clean package

mvn spring-boot:run

