**코틀린 프로그래밍과 응용**

**설계프로젝트 과제제안서**

과목명 : **코틀린 프로그래밍과응용**

제출일자 : 2024년 12월 24일

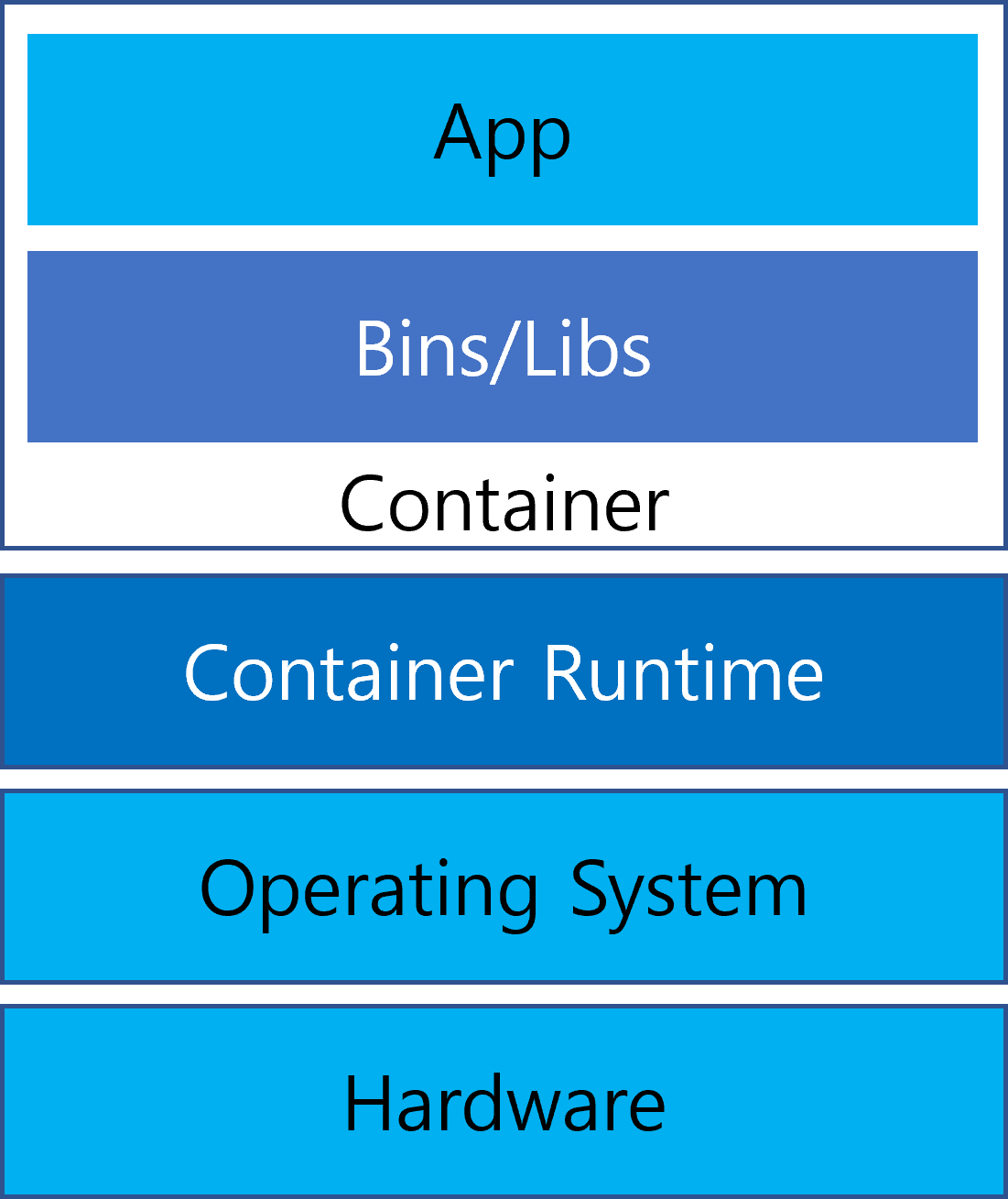
성 명/학번 : 김 동 하 (22012231)

소속 학과 : 디지털융합대학 컴퓨터학부 정보통신공학전공

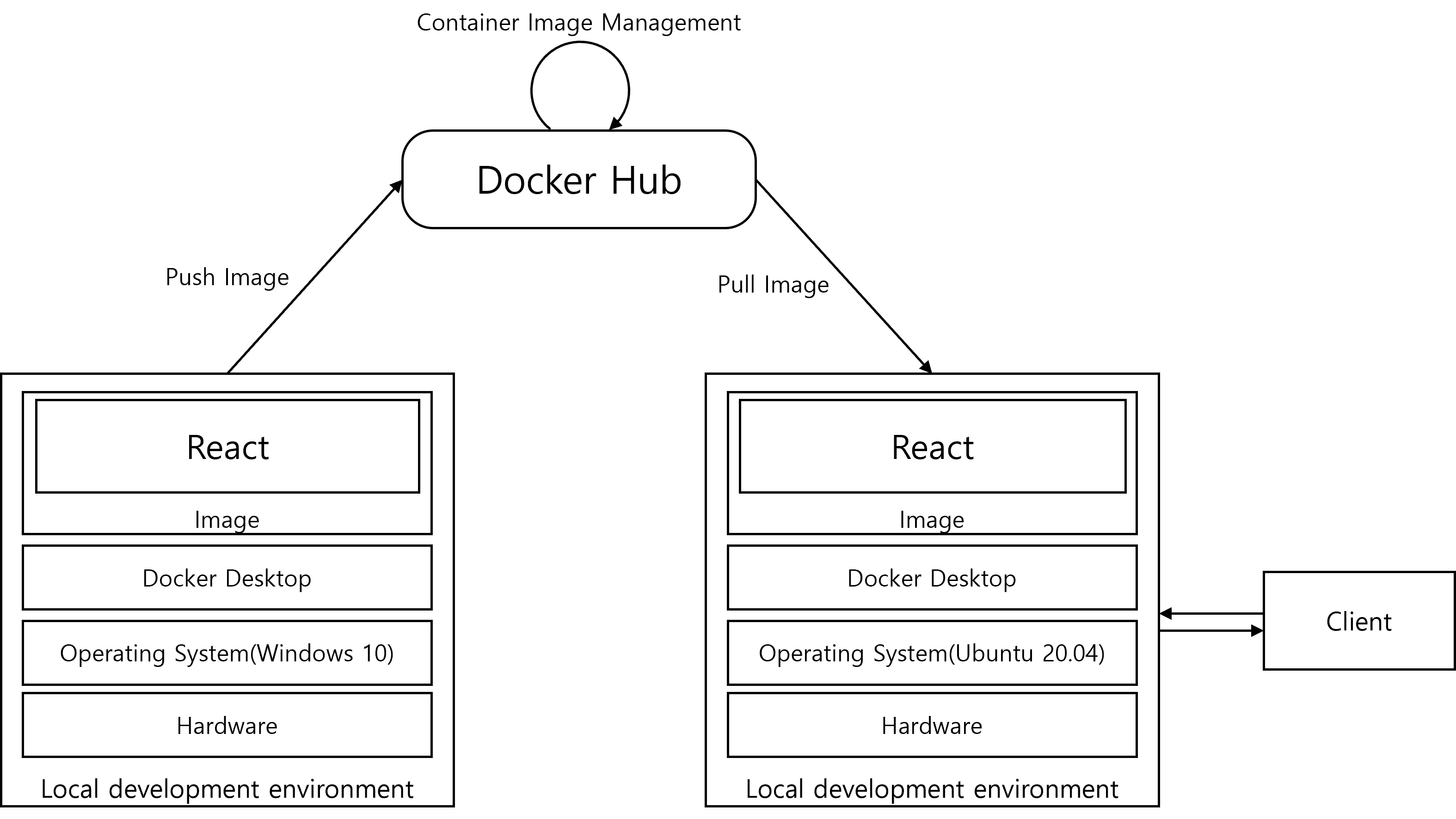
**과제 제안서**

|  |  |
| --- | --- |
| 과 제 명 | Kotlin/JS 및 Spring Boot 기반 프로젝트 관리 시스템 |
| 과제기간 | 2024. 11. 29 (금) ~ 2024. 12. 26 (목) |
| 1. 프로젝트 결과물 제목 | |
| * ProManage (프로젝트 관리 시스템 Manager.시점) | |
| 2. 주요 기능 | |
| * Kotlin Spring Boot 서버 통신 기반 로그인 및 프로젝트 정보 불러오기 * React , Kotlin/JS 기반 UI 디자인 * 여러 프로젝트를 관리하는 회사의 매니저를 위한 프로젝트 관리 시스템 | |
| 3. 팀 구성 | |
| 김동하 (22012231) | |
| 4. 세부 기능블럭도 | |
|  | |
| 5. 기능 순서도 (Sequence Diagram) | |
| 로그인 기능 순서도  프로젝트 등록 기능 순서도 | |
| 6. 관련 Database 구성, Table 별 세부 속성 (attribute) | |
| 데이터 베이스 ERD 구성도 | |
| 7. 주요 Kotlin 클래스, 데이터 멤버, 멤버함수 | |
| Company\_employee (user\_id name specialty user\_email)    Project(project\_id project\_name description start end finish)    Projectmember(user\_id project\_id task\_name task\_description) | |
| 8. 핵심 알고리즘 | |
| 1. 프로젝트 조회 및 관리 알고리즘   * 프로젝트 상세 정보: 프로젝트에 대한 상세정보를 확인할 수 있는 알고리즘 * 프로젝트 관리: 프로젝트를 효과적으로 관리하기위한 알고리즘   2. 테스크 조회 및 관리 알고리즘   * 테스크 관리: 직원의 id로 담당하는 테스크를 등록하는 알고리즘 * 테스크 조회: 관리자가 화면에서 테스크를 선택해 더 자세한 테스크 정보를 볼 수 있게 해주는 해주는 알고리즘   3. 직원 등록 및 관리 알고리즘   * 직원 등록 : 프로젝트에 참여할 수 있는 직원을 등록 할수있게 해주는 알고리즘 * 직원 삭제 : 필요없는 직원에 해당하는 자는 삭제 | |
| 9. Server 구성 | |
| Server 구성: | |
| 10. Server Dependencies | |
|  | |
| 11. Client 화면 구성 | |
| 1. Main.page : 유저 관리 or 매니저 관리중 하나를 선택 2. Member.page : 유저를 관리하는 페이지           2 Project.page : 회사 유저의 목록을 삭제 및 추가 가능       1. Projectmember.page : 프로젝트를 누르면 Project.page로 넘어감 이 페이지에는 프로젝트의 상세부분가 그 프로젝트에 참여하는 멤버를 추가할 수 있음 | |

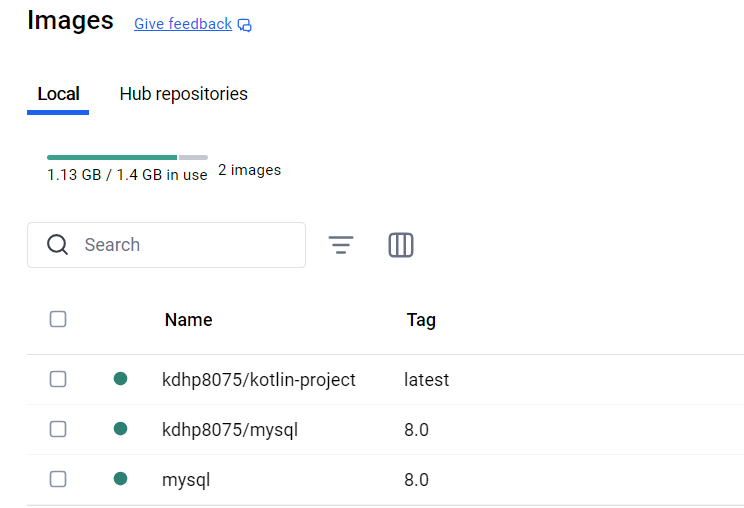




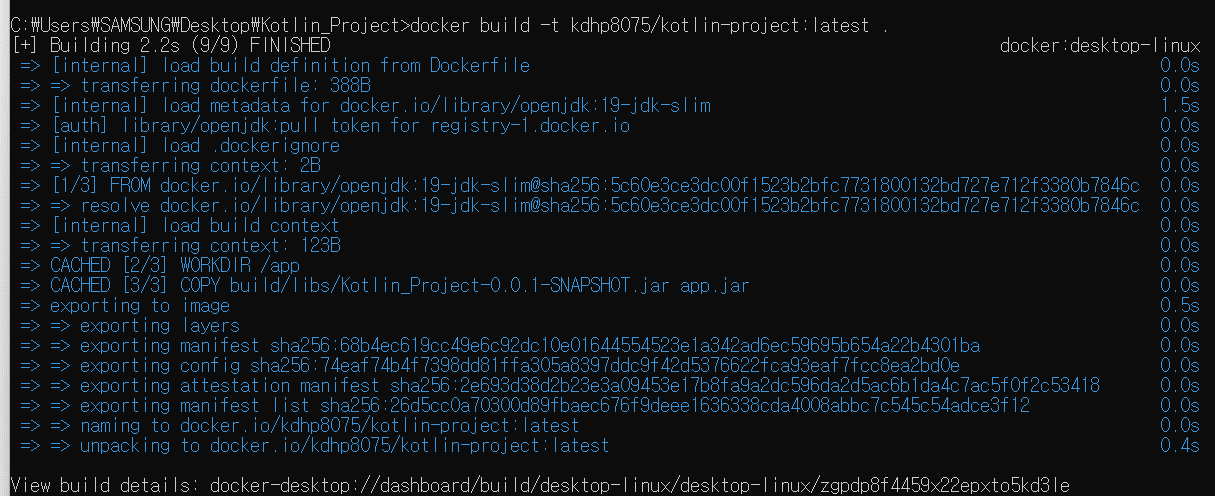
**Docker deployment**



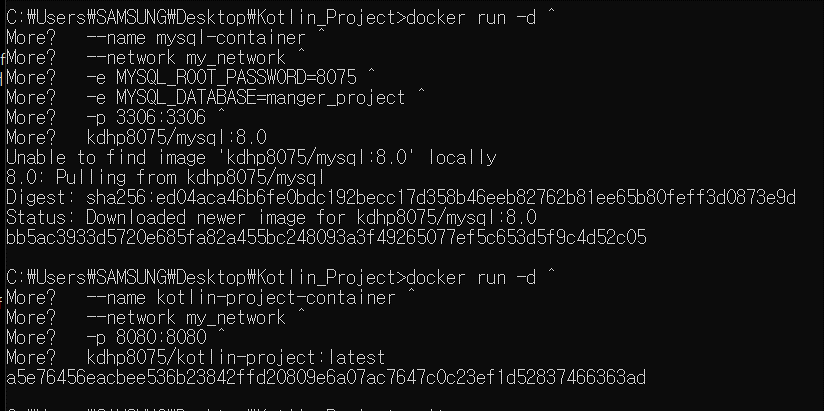
**Docker Sequence**

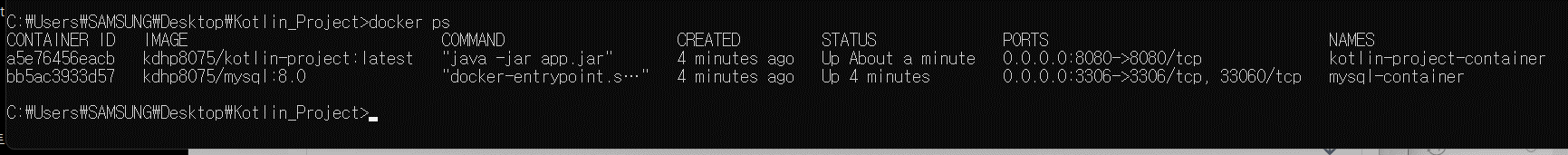


Docker Hub push



Docker 실행 및 작동





|  |
| --- |
| 14 . MSA 구조 |
| **Employee Service**   * 역할: 직원 정보를 관리하고 제공하는 서비스. * 포함된 파일:   + model/Employee.kt: 직원 엔티티.   + repository/EmployeeRepository: 직원 데이터를 관리하는 리포지토리.   + service/EmployeeService: 직원 관련 비즈니스 로직.   + controller/EmployeeController: 직원 API 엔드포인트.   **Project Service**   * 역할: 프로젝트 정보를 관리하고 제공하는 서비스. * 포함된 파일:   + model/Project.kt: 프로젝트 엔티티.   + repository/ProjectRepository: 프로젝트 데이터를 관리하는 리포지토리.   + service/ProjectService: 프로젝트 관련 비즈니스 로직.   + controller/ProjectController: 프로젝트 API 엔드포인트.   **Project Member Service**   * 역할: 프로젝트 멤버 정보를 관리하고 제공하는 서비스. * 포함된 파일:   + model/ProjectMember.kt: 프로젝트 멤버 엔티티.   + repository/ProjectMemberRepository: 프로젝트 멤버 데이터를 관리하는 리포지토리.   + service/ProjectMemberService: 프로젝트 멤버 관련 비즈니스 로직.   + controller/ProjectMemberController: 프로젝트 멤버 API 엔드포인트.   **Home Service**   * 역할: 초기 엔드포인트를 제공하는 서비스. * 포함된 파일:   + controller/HomeController: 메인 엔드포인트를 제공. |