**미니프로젝트 1 프로젝트 계획서**

1. **프로젝트 개요**

|  |  |
| --- | --- |
| **구분** | **내용** |
| 프로젝트 | 인스타그램 클론 코드 |
| 프로젝트  기간 | 9.15~9.19 |
| 프로젝트  목표 | 친숙한 서비스인 Instagram을 분석해서 구조를 직접설계합니다.  역할을 분담하여 서비스를 코드로 구현해봅니다.  구현하는 과정에서 Front-end와 Back-end가 협업하는 과정을 이해합니다.  자신이 맡은 부분을 남에게 설명할 수 있는 의사소통 능력을 기릅니다.  단순히 구현 후 끝나는 것이 아닌 코드 리뷰와 피드백을 통해 함께 성장합니다.  기존의 코드를 지속적으로 개선하기 위해 리팩토링을 진행합니다.  Spring Boot와 React를 기반으로 다양한 기술 스택을 학습하고 적용합니다. |

1. 프로젝트 수행 계획

|  |  |
| --- | --- |
| 프로젝트  수행 범위 | 1. 사용자 인증 및 관리  2. 게시물 기능  - 게시물 작성, 삭제  - 게시물 목록 / 조회  3. 소셜 기능  - 좋아요, 댓글, 팔로우 기능  4. UI 기능 구현  5. AI 연계  - 사진 등록 시 해시태그 추천 기능 |
| 프로젝트  방법론 및  도구 | 방법록 : 애자일(Agile) - 짧은 주기(데일리 점검 & 피드백)  협업 도구 : GitHub, Notion  개발 환경 : Frontend: React, Axios  Backend : Spring Boot, Spring Security, JPA(Hibernate) DB: MariaDB  Open Ai : ChatGPT |
| 프로젝트  산출물 | 프로젝트 수행계획서, WBS, 요구사항정의서, 단위테스트 결과서, API 명세서 (Swagger 캡처), 서비스 구성도 (이미지), DB 설계도 (이미지), 구축 결과 보고서 (PPT + PDF) |

1. 프로젝트 수행 상세 내용

|  |  |
| --- | --- |
| 상세  프로젝트  수행내용 | 이 프로젝트는 React 기반의 인스타그램 클론 프론트엔드 개발을 목표로 하며, 주요 기능과 UI/UX 구현을 중심으로 진행되었습니다.  1. 피드(Feed) 기능  - 팔로우한 사용자의 게시글을 무한 스크롤 방식으로 조회.  - 각 피드 카드에는 작성자 프로필, 닉네임, 이미지 슬라이더, 본문, 해시태그, 작성일, 좋아요/댓글 UI가 포함됨.  - 좋아요 버튼 클릭 시 상태 및 카운트가 즉시 반영.  - 댓글 및 대댓글(답글) 입력, 삭제, 실시간 목록 갱신 기능 구현.  2. 프로필(Profile) 기능  - 프로필 페이지에서 사용자 정보(프로필 이미지, 닉네임, 이메일, 게시물/팔로워/팔로우 수)와 게시물 목록을 조회.  - 프로필 이미지 클릭 시 파일 선택 창이 열리고, 이미지 업로드 후 즉시 반영.  - 본인 프로필이 아닐 경우 팔로우 버튼 노출 및 팔로우 기능 제공.  - 프로필 정보 변경 시 Navigation의 프로필 이미지도 동기화.  3. 게시물 작성/수정(FeedCreateEditModal)  - 이미지 업로드, 본문 입력, 해시태그 추출 등 게시물 작성 및 수정 기능.  - 게시물 등록/수정 시 Spinner를 통한 로딩 상태 표시.  4. 댓글/대댓글 UI 및 기능  - 댓글 입력자 프로필 이미지와 닉네임, 내용, 작성일, 답글 달기/삭제 버튼이 한 줄에 정렬.  - 대댓글은 부모 댓글보다 오른쪽에 들여쓰기되어 계층 구조가 명확하게 표현됨.  내가 작성한 댓글/대댓글만 삭제 버튼이 노출되며, 삭제 시 실시간 목록 갱신.  5. 공통 컴포넌트 및 UX 개선  - Spinner 컴포넌트 공통화 및 API 호출 시 특정 영역에 로딩 표시.  - IntersectionObserver를 활용한 무한 스크롤 구현.  - 각종 입력 폼, 버튼, 모달 등 styled-components를 활용한 일관된 디자인 적용.  6. 기타  - 로그인/회원가입 페이지, 토큰 관리, API 연동 등 기본 인증 및 사용자 관리 기능  - Git 충돌 및 병합 처리, 커밋/풀/푸시 등 협업 |
| 프로젝트  WBS | 1. 기획  1.1 사전조사  1.2 요구사항 명세서 작성  2. 모델링  2.1 개념/논리 모델링  2.2 물리 모델링  3. 개발  3.1 GitHub 팀 설정  3.2 프론트 구축  3.3 백엔드 구축  3.4 기능구현  4. 테스트  4.1 테스트케이스 작성  4.2 테스트 데이터  4.3 검증  5. 문서화  5.1 프로젝트 수행 계획서  5.2 WBS  5.3 요구사항 정의서  5.4 단위테스트 결과서  5.5 API 명세서  5.6 서비스 구성도  5.7 DB 설계도  5.8 구축 결과 보고서  \* 세부사항 별도 첨부 문서 참고 |
|  |  |