

# 파이널 프로젝트 #1

안드로이드 플루터 프로젝트 - 프로젝트 진행

# Overview

1. 팀 구성 및 역할 분담
2. 주제 선정 및 기획
3. 요구 사항 분석 및 설계
4. 개발 환경 구축
5. 기능 개발
6. 프로젝트 리뷰 및 리팩토링
7. 배포 준비 및 최종 마무리
8. 최종 발표

# 1. 팀 구성 및 역할 분담

- 모든 팀원은 \*\*기획, UI 설계, 개발, 발표까지 전 과정에 함께 참여\*\*합니다.
- 역할은 고정되지 않고, 주요 책임을 나누어 \*\*협업 중심으로 분담\*\*합니다.
- 스크럼 마스터 1인 선정
  - 스프린트 일정 관리, 회의 진행, GitHub 이슈/PR 흐름 정리 등 \*\*프로젝트 진행 조율\*\* 담당
- 예시 역할 분담:
  - 기획 및 화면 설계 : 전체 팀원이 함께 기능 정의 및 화면 플로우 설계 진행
  - UI 개발 책임 : 주로 UI 위젯과 레이아웃 구현을 이끄는 역할
  - 로직 개발 책임 : 상태 관리, API 연동, 데이터 처리 등 핵심 로직 담당
  - 코드 구조 및 품질 관리 : 폴더 구조 설계, 공통 컴포넌트 정리, 리뷰 주도
  - 문서 및 발표 준비 : 회의록, README, 발표 자료 작성 및 발표 리허설 주도

## 2. 주제 선정 및 기획

- 공통 주제인 \*쇼핑몰앱\*을 기반으로 팀별로 어떤 형태의 쇼핑몰을 만들지 논의.  
예) 일반 쇼핑몰, 중고 거래, 마켓플레이스, 특정 상품 전문몰 등
- 앱에 필요한 기능을 함께 정리하고, 각 기능의 \*우선순위\*를 설정합니다.  
예) 회원가입/로그인, 상품 목록, 상품 상세, 장바구니, 주문, 찜하기 등
- 프로젝트 기간과 리소스를 고려하여 \*핵심 기능(필수)\*과 \*부가 기능(선택)\*을 구분.
- 전체 화면 구조 및 사용자 흐름(Flow)을 스케치.  
예) 화이트보드, 종이, Figma, Whimsical 등
- 구현할 화면 목록을 정리하고, 간단한 \*\*와이어프레임\*\* 또는 \*\*UI 시안\*\*을 작성.

# 3. 요구 사항 분석 및 설계

- 기능 명세서 작성(로그인, 상품 조회, 주문 등)
- 화면 명세서, API 정의서(강사 제공), DB 모델링(강사 제공)
- 기술 스택 및 아키텍처 결정 (MVVM / Feature-based / Clean Architecture 등)
- 상태 관리 방식 선정 (예: Riverpod, Bloc)
- 폴더 구조 설계 및 코드 스타일 가이드 설정
- Git 브랜치 전략 및 커밋 규칙 정의

# #기능 명세서 예시

- 기능명 : 사용자 로그인
- 기능 설명 : 사용자가 이메일과 비밀번호로 로그인을 수행한다. 유효한 정보일 경우 토큰을 발급받고 앱 내에서 인증 상태를 유지한다.
- 사용자 : 일반 사용자 / 판매자
- 관련 화면 : 로그인 화면(/login)
- 진입 조건 : 사용자가 앱 실행 후 로그인 버튼 클릭
- 처리 흐름 : 사용자가 이메일, 비밀번호 입력 -> “로그인” 버튼 클릭 -> API 호출:POST /api/auth/login 응답 성공시 accessToken 저장 및 홈화면으로 이동
- 입력 항목 : 이메일, 비밀번호
- 예외 처리 :
  - 이메일 형식이 유효하지 않은 경우 -> 에러 메시지 출력
  - 비밀번호가 틀린경우 -> 에러 메시지 출력
- 성공 조건 : code = “0000” , accessToken, refreshToken 정상 발급되고 홈 화면으로 이동함.
- API 명세 연동 : Swagger : /api/docs/auth/login 참조
- 비고
  - 로그인 유지 기능 포함(SharedPreferences 사용)
  - 로그인 실패 회수 제한(5회 초과시5분 제한)

# #화면 명세서 예시

- 화면ID : LOGIN\_01
- 화면명 : 로그인 화면
- 화면 설명 : 사용자가 이메일과 비밀번호로 로그인할 수 있는 화면
- UI 구성 요소
  - 이메일 입력 필드, 비밀번호 입력 필드, 로그인 버튼, 회원가입 버튼
- 입력 조건 : 이메일 (형식 체크) , 비밀번호 (8자 이상)
- 동작정의
  - 로그인 버튼 -> /api/auth/login 호출
    - 성공시 -> 홈 이동
    - 실패시 -> 메시지 출력
  - 회원 가입 버튼 -> 회원 가입 페이지로 이동 -> SIGNUP\_01
- 상태 처리 : 로딩 상태 표시(버튼 비활성화 및 스피너), 오류 발생시 화면 하단에 SnackBar 출력
- 비고 : 자동 로그인 여부 저장(SharedPreferences 활용)

## 4. 개발 환경 구축

- Flutter 개발 환경 설치 및 테스트
- GitHub 저장소 생성 및 연동
- Flutter 프로젝트 초기화 및 구조 반영
- `.gitignore`, pubspec.yaml 등 기본 세팅 완료



## 5. 기능 개발 (주차별 스프린트 진행)

- UI 화면 구현
- 상태 관리 적용 (Riverpod, Bloc 등)
- API 연동 및 비동기 처리
- 기능별 테스트 및 디버깅

## 6. 프로젝트 리뷰 및 리팩토링

- 코드 리뷰 및 공통 스타일 정리
- 중복 코드 제거 및 구조 개선
- 앱 성능 및 사용자 경험 개선

# 7. 배포 준비 및 최종 마무리

- 앱 빌드(Android APK 또는 Web build)
- 결과물 정리 (스크린샷, 소개 영상 등)
- 발표 자료 작성 및 연습
- GitHub 저장소 정리 및 README 작성

## 8. 최종 발표

- 팀별 발표 (15~20분 내외)
- 시연 중심 + 주요 구현 포인트 설명
- 회고 공유 (좋았던 점, 아쉬웠던 점, 배운 점 등)