

2025년 상반기 K-디지털 트레이닝

KB 스켈레톤 가이드

[KB] IT's Your Life

1 프로젝트 개요

- ✓ 프로젝트 일정 - 2025.04.07(월) ~ 2025.04.11(금), 5일
- ✓ 프로젝트 주제 - 가게부 서비스 앱
- ✓ 프로젝트 목표
 - 수업시간에 배운 기술 요소를 실제 서비스에 접목해 봄으로써 각 개인의 프로그래밍 능력 향상
 - 서비스(프로젝트) 개발 시 거치게 되는 기획, 설계, 코딩, 테스트, 배포의 각 단계를 실제 경험해 볼 수 있는 기회가 됨
 - 팀 프로젝트 진행 시 팀원 간의 협력 및 충돌 시 조정 과정 경험
 - 프로그램 소스의 형상 관리(버전 관리) 기법 배양
 - 향후 진행될 파이널 프로젝트의 진행 절차를 사전에 경험해 봄

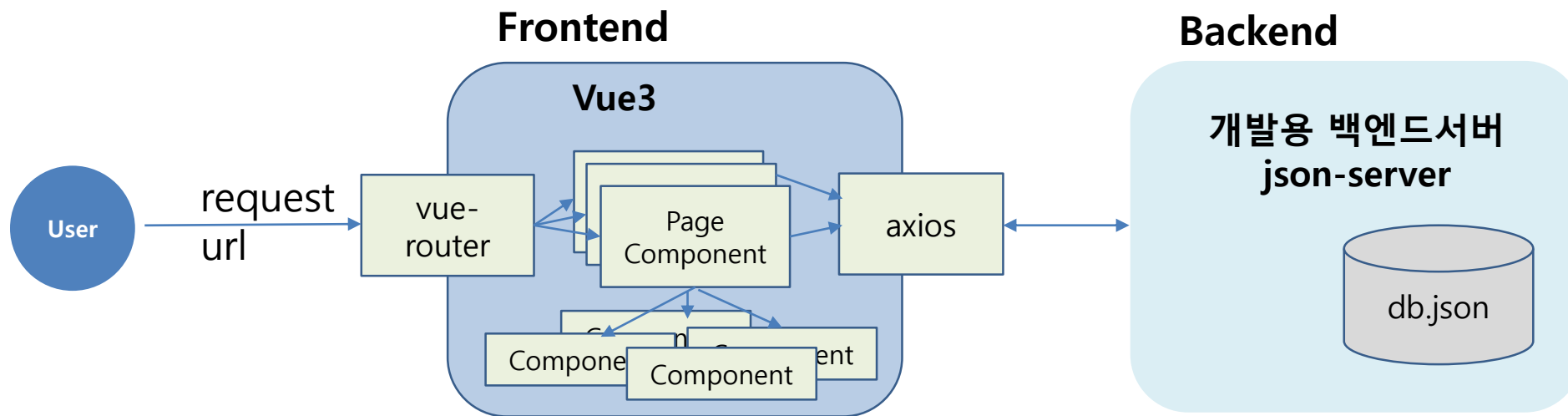
✓ 프로젝트 진행 방식 및 일정

- 팀 구성 방식 - 랜덤구성(제비뽑기, 사다리타기 등 활용)
- 일정(팀 별로 조정)
 - 1일차 - 조 구성, 참조 사이트 조사(벤치마킹) 및 구현 기능 기획
 - 1일차 - UI 설계(url 결정, 화면 스토리보드), 백엔드(데이터 json 구조 결정), 프로젝트 골격 구성, 깃 연동
 - 2일차 - 구현
 - 3일차 - 구현
 - 4일차 - 구현
 - 5일차 - 구현 마무리, 문서화, 깃 정리, 발표

✓ 기능 명세

구분	항목	상세 내용
기능 요구 사항	수입/지출 기록	날짜, 금액, 카테고리, 메모 등의 세부 정보를 입력하여 기록
	거래 내역 조회	필터 기능을 통해 특정 조건에 맞는 거래 내역을 조회 날짜/주간/월별/기간별, 카테고리 별 등 필터링 항목에 따른 요약 내역 조회
	월별 재정 요약	각 월의 수입, 지출, 순이익을 요약 표시 그래픽(차트) 처리는 선택 사항
	데이터 저장	데이터는 모두 json-server에 저장되어야 함
디자인 요구 사항	반응형 디자인	다양한 디바이스와 화면 크기에 맞게 최적화된 디자인(선택)
	사용자 인터페이스	사용자의 편리한 사용을 위한 직관적인 UI 설계
기술적 요구 사항	프론트엔드 구현	Vue 3를 사용하여 모던 웹 애플리케이션 구축
	상태 관리	Composition API를 통한 앱 상태 관리

○ 팀의 여력에 따라 기능 추가 가능!



✓ 사용 기술 스택(필수)

○ 프론트엔드

- vite project
- Composition API 또는 Options API
- vue-router
- axios
- components
- pinia
- 이벤트, 스타일(bs5, fontawesome)

○ 백엔드

- json server
 - db server 역할

✓ 사용 기술 스택(선택)

- figma, draw.io

✓ json-server 개요

- json 파일을 데이터베이스로 하여 REST-API 서버 역할을 제공하는 모듈
- Frontend 개발 시 임시 백엔드 서버로 활용 가능

✓ 설치

- 전역 설치
 - `npm install -g json-server`

✓ 참고 문헌

- <https://www.npmjs.com/package/json-server>

db.json

```
{
  "posts": [
    { "id": "1", "title": "a title", "views": 100 },
    { "id": "2", "title": "another title", "views": 200 }
  ],
  "comments": [
    { "id": "1", "text": "a comment about post 1", "postId": "1" },
    { "id": "2", "text": "another comment about post 1", "postId": "1" }
  ],
  "profile": {
    "name": "typicode"
  }
}
```

○ 주의사항

- id는 문자열 타입으로 운영
 - json-server는 기본적으로 4자리 문자열로 운영함

✓ 실행

○ 전역 설치하지 않고 실행

- `npx json-server db.json`

○ 전역 설치 시

- `json-server db.json`

- `db.json`

```
{  
  "posts": [...],  
  "comments": [...],  
  "profile": {...}  
}
```

JSON Server started on PORT :3000
Press CTRL-C to stop
Watching db.json...

♡(◡ ◡)

Index:
<http://localhost:3000/>

Static files:
Serving ./public directory if it exists

Endpoints:
<http://localhost:3000/posts>
<http://localhost:3000/comments>
<http://localhost:3000/profile>

✓ Routes

- GET /posts
- GET /posts/:id
- POST /posts
- PUT /posts/:id
- PATCH /posts/:id
- DELETE /posts/:id

http://localhost:3000/posts

```
▼ [
  ▼ {
    "id": "1",
    "title": "a title",
    "views": 100
  },
  ▼ {
    "id": "2",
    "title": "another title",
    "views": 200
  }
]
```

http://localhost:3000/posts/2

```
▼ {
  "id": "2",
  "title": "another title",
  "views": 200
}
```

✓ 정렬

- `_sort=필드명1,필드명2`
 - 디폴트: 오름차순
 - `-필드명`: 내림차순

`GET /posts?_sort=id,-views`

✓ 페이지네이션

- `page`: 페이지 번호
- `per_page`(기본값 10): 1 페이지당 데이터 건 수

`GET /posts?_page=1&_per_page=25`

✓ 제출물

- 기획 문서
 - 워드 문서나 notion 등에서 작성한 내용을 담은 doc 또는 pdf 파일
- 화면 설계 문서
 - 스토리 보드
 - ppt, figma, draw.io 이미지 파일 등 실제 작성한 파일
- 소스 코드
 - 소스코드, 이미지 파일 등등
- 발표 자료

✓ 기본 화면 레이아웃

- 상단 바: 날짜 선택기, 설정 및 프로필 접근 버튼
- 요약 카드: 이번 달의 총 수입, 총 지출, 순수익을 보여주는 세 개의 카드
- 최근 거래 목록: 날짜와 함께 표시되는 최근 거래 목록
- 빠른 추가 버튼: 화면 하단에 고정된 버튼으로, 새로운 거래를 신속하게 추가할 수 있음
- 위의 내용을 기본으로 포함하는 화면 구성을 각 팀 기획에 맞게 제작

✓ 메인 대시보드 (홈 화면)

○ 기능

- 최근 거래 내역 요약
- 월별 수입 및 지출 요약
- 빠른 추가 버튼을 통한 새 거래 등록

✓ 거래 내역

○ 기능:

- 거래 내역 전체 보기
- 기간별, 카테고리별 필터링 기능
- 각 거래에 대한 세부 정보 보기 및 수정

○ 구성 요소:

- 필터 바: 날짜 및 카테고리별로 거래를 필터링할 수 있는 옵션
- 거래 목록: 각 거래의 날짜, 카테고리, 금액, 메모가 표시됨
- 수정 및 삭제 옵션: 각 거래 옆에 수정 및 삭제 버튼 제공

✓ 거래 등록 및 수정

○ 기능:

- 새로운 거래 추가 또는 기존 거래 수정
- 거래 날짜, 금액, 카테고리, 메모 입력

○ 구성 요소:

- 입력 폼: 날짜 선택, 금액 입력, 카테고리 선택 드롭다운, 메모 입력 필드
- 저장 버튼: 입력된 정보를 저장하거나 업데이트
- 취소 버튼: 입력을 취소하고 이전 화면으로 돌아감

✓ 설정 및 프로필 관리

○ 기능:

- 사용자 설정 변경
- 프로필 정보 관리

○ 구성 요소:

- 사용자 정보: 이름, 이메일 등의 프로필 정보 표시 및 수정

✓ 1일차: 조 구성, 참조 사이트 조사(벤치마킹) 및 구현 기능 기획

○ 조편성

- 팀장 선출

○ 기존 가계부 앱 또는 사이트 조사

- 프로젝트에 가장 참고할 만한 사이트 1~2개 선정
- 화면 구성 및 전개 흐름 분석
- 각 화면에서 제공하는 기능 및 출력 내용 정리

○ 프로젝트에서 구현할 기능 기획

- 요구 기능 목록을 표로 정리(요구 기능 명세서)

no	구분	기능명	설명	우선순위	기간	담당자

- ✓ 1일차: UI 설계, 백엔드, 프로젝트 골격 구성, 깃 연동
 - 화면 레이아웃 결정
 - 각 화면의 url 및 필요 정보 결정
 - 각 화면 구성 컴포넌트 결정
 - 데이터 저장용 기본 json 파일 결정
 - 테스트 데이터 구축
 - json-server 백엔드 운영 테스트
 - 각 항목별 CRUD 테스트
 - 프로젝트 기본 코드 생성 및 git repository 생성 및 팀원 구성 후 배포

✓ JSON 파일 샘플(팀별 기획에 맞게 재정의)

```
{
  budget: [
    { date, type(income, expense), category, detailCategory, amout, memo },
  ],
  incomeCategory : [ '월급', '용돈', '이자', ... ],
  expenseCategory : [ '식비', '교통비', '유흥', '공과금', ... ],
  periodicExpense: [
    { date, type(expense), category, amout, memo },
  ]
}
```

○ endpoints

- /budget
- /incomeCategory
- /expenseCategory
- /periodicExpense

✓ 2일차: 기본 구현

- 각 페이지 별/컴포넌트별/기능별 개발 담당자 선정
- 라우터 모듈 구성
- 페이지 기본 컴포넌트 제작
 - 각 URL별 라우팅 검증
- AXIOS API 모듈 제작
 - API 별 JSON-SERVER 연동 확인
- 각 페이지별 컴포넌트 제작
- git repository 정리

✓ 3일차, 4일차: 세부 기능 구현

- 각 페이지의 요구 기능들을 컴포넌트와 함께 구현
 - 등록 기능과 출력 기능을 나눠 병렬로 진행
 - 2:3 비율로 팀원 배정
- git repository 정리

✓ 5일차 : 보고서 작성 및 발표

- 프로젝트 문서화
- 발표 준비
- 팀 내 리뷰
- 최종 발표
- 최종 산출물 제출(업로드)

✓ 발표자료 포함 항목

1. 프로젝트 제목
2. 프로젝트 주제 및 구성원
3. 프로젝트 구성도 및 사용 기술
4. 프로젝트 주요 기능
5. 프로젝트 주요 화면 설계
6. 프로젝트 컴포넌트 설계
7. 프로젝트 구현 결과 화면
8. 프로젝트 후기

※팀별 상황에 맞추어 수정 가능

[제출 방법]

모든 산출물은 발표일에
러닝플랫폼(MLP) 개별 업로드
(우측 내용 참고 / 추후 안내 예정)

[포트폴리오 작성요령]

목차를 참조하되,
템플릿/디자인/구성은 변경 및 추가 가능
포트폴리오 바탕으로 프로젝트 발표 진행
(조당 발표시간 10분 ~15분 분량)

※ 강사님께서 별도 공유하신 양식이 있다면 그 양식으로 사용하셔도 됩니다.

01 발표자료 :

스켈레톤 프로젝트 발표 후 MLP에 개별 업로드 진행

- MLP

> 나의학습 > [KB] IT's Your Life - 시험, 설문, 과제

> [제출] 스켈레톤 프로젝트 포트폴리오

- 개인 업로드 진행 (제출 후 수정불가)
- PDF 파일로 제출 (용량제한 : 10MB 이하)
- 파일명 : 00회차_성함_스켈레톤 프로젝트

02 제출기한 : 4월 11일(금) 23시 59분 까지

여러분~ 스켈레톤 프로젝트가 시작됩니다.

최종프로젝트 이전에 프로젝트 가볍게 경험해본다 생각해주시고,

부담없이 팀프로젝트 경험해보주시기 바랍니다.

항상 건강 유의하시고~ 언제나 여러분을 응원합니다♥

- ✓ 비대면반은 프로젝트도 수업의 일환임으로 쉬는 시간을 제외하고 **캠 상시 ON**
(* 캠이 켜지지 않거나 연락이 안되는 경우에는 부정출결로 간주됩니다.)
- ✓ 대면반은 쉬는시간 이외 자리이탈 금지
(* 무단으로 강의장 이탈할 경우 부정출결로 간주됩니다.)
- ✓ 출결이슈(조퇴,결석,외출,공가) 생길 시 매니저, 강사님, 조원들 **미리 공유**하기!
- ✓ 프로젝트 기간동안 훈련생 개인 **건강관리**에 유의합니다!

