**Ⅰ. 프로젝트 전제**

| **항목** | **내용** |
| --- | --- |
| **스택** | Vite + React 18, TypeScript, TailwindCSS 3.x |
| **루트 구조** |  |

text

복사편집

src/

├─ assets/

├─ components/

├─ modules/ # utils, api, hooks

├─ pages/

└─ styles/

| **디자인 토큰** |

scss

복사편집

:root{

--accent:#88F0FF;

--accent-glow:0 0 6px rgba(136,240,255,.8);

--motion-ease:cubic-bezier(.25,.8,.5,1);

--motion-base:.3s;

}

**Ⅱ. Task Break‑Down (Ready for JIRA / Notion)**

**❶–❻은 P1(이번 스프린트), ❼–⓰은 P2~P3**

| **ID** | **User Story & Acceptance Criteria** | **File / 위치** |
| --- | --- | --- |
| **❶ HERO‑LAYOUT‑REF** | *As a user on 1024 px 이하 노트북, I want the hero section to stack vertically so that 텍스트와 일러스트가 모두 보인다.* ☑ Breakpoint 1024 px에서 .hero {flex-direction: column-reverse} ☑ .illust{height:40vh} ☑ Lighthouse CLS < 0.1 | src/pages/Home.tsx, styles/hero.css |
| **❷ STICKY‑MININAV** | *As a user, I can jump between 캐릭터/키보/보스 섹션 and see which tab is active.* ☑ <nav data-scroll-spy> 구현 ☑ Mobile(≤576 px)에서는 footer‑tabbar로 렌더 | components/MiniNav.tsx |
| **❸ CAROUSEL‑UPGRADE** | *As a user, I can swipe 썸네일을 드래그·터치로 이동하고 활성 썸네일이 확대된다.* ☑ scroll-snap-type:x mandatory ☑ Actived 썸네일 transform:scale(1.1) + glow | components/Carousel.tsx |
| **❹ SAFE‑DOM‑UPDATE** | *No innerHTML injection remains in code.* ☑ 모든 innerHTML = 제거 → textContent or JSX ☑ DOMPurify 적용 위치 0건 | 전역 |
| **❺ TOAST‑ALERT** | *Alerts use non‑blocking Snackbar.* ☑ useToast() hook 도입 (headlessui or radix) ☑ 기존 alert/confirm 100 % 제거 | modules/useToast.ts, replace points |
| **❻ IMAGE‑WEBP‑LAZY** | *Hero and list images load as .webp and lazy‑load.* ☑ <img srcSet="xxx.webp" loading="lazy"> ☑ Lighthouse LCP < 2.5 s on 4G simulated | assets, components |
| **❼ COLOR‑CONTRAST** | WCAG AA 4.5:1 이상 준수, 모든 컬러 토큰 업데이트 | styles/tokens.scss |
| **❽ MOTION‑VARIABLE** | 모든 transition/animation에 var(--motion-base) var(--motion-ease) 사용 | 전역 |
| **❾ MINIGAME‑FLASHCARD** | *User plays flashcard quiz; 점수 저장(LocalStorage).* ☑ Route /minigame ☑ 데이터 JSON {id,name,img} ☑ Flip animation framer‑motion | pages/mini-game/FlashCard.tsx |
| **➓ CI‑LIGHTHOUSE** | *Every PR triggers Lighthouse‑CI > 95.* | .github/workflows/lighthouse.yml |

**Ⅲ. 코드 Snippet 모음 (즉시 복붙 가능)**

**1. Hero 반응형**

tsx

복사편집

// Home.tsx

export const Hero:FC = () => (

<section className="hero flex md:flex-row flex-col-reverse items-center">

<InfoCard className="md:w-1/2 w-full"/>

<img

className="illust md:w-1/2 w-full h-[40vh] object-contain"

src="/assets/luna.webp"

alt="루나 일러스트"

loading="eager"

fetchpriority="high"

/>

</section>

);

**2. Glow Button**

tsx

복사편집

<button

className="btn-primary px-6 py-3 rounded-full

text-white bg-[var(--accent)]

shadow-[var(--accent-glow)] transition-transform

hover:scale-105">

사전예약

</button>

**3. Toast Hook**

ts

복사편집

// modules/useToast.ts

import { create } from "zustand";

type Toast = { id: number; msg: string };

export const useToast = create<{ toasts: Toast[]; push:(msg:string)=>void }>(

set => ({

toasts: [],

push: msg => set(s => ({ toasts:[...s.toasts,{id:Date.now(),msg}] }))

})

);

**4. Scroll‑Spy Mini‑Nav**

tsx

복사편집

const sections = ["character","keyboard","boss"];

export const MiniNav = () => {

const [active,setActive] = useState(sections[0]);

useEffect(()=>{

const io = new IntersectionObserver(

entries => entries.forEach(e=>e.isIntersecting && setActive(e.target.id)),

{threshold:0.6}

);

sections.forEach(id=>io.observe(document.getElementById(id)!));

return ()=>io.disconnect();

},[]);

return (

<nav className="sticky top-0 bg-backdrop/40 backdrop-blur z-50">

{sections.map(id=>(

<a key={id}

className={clsx("px-4",active===id && "text-[var(--accent)]")}

href={`#${id}`}>{id.toUpperCase()}</a>

))}

</nav>

);

};

**Ⅳ. 전달용 “README\_kilocode.md” 샘플**

md

복사편집

# 작업 범위

이번 스프린트(P1)는 ❶~❻ Task 구현이 목표입니다.

- 브랜치: `feature/p1-refactor`

- 코드 스타일: ESLint + Prettier config 포함

## 빌드 & 실행

```bash

npm i

npm run dev

**참고 시안**

* 디자인 레퍼런스: <https://azurpromilia.com/kr/>
* 색상·애니메이션 토큰: styles/tokens.scss 참고

markdown

복사편집

> \*\*이 파일을 첨부해서 kilo code에 넘기면\*\* 각 Task ID별 커밋 또는 PR을 생성할 수 있도록 충분한 정보가 포함됩니다.

---

src/

└─ pages/

└─ minigame/

├─ index.tsx # <MiniGameHub />

├─ flashcard/ # <FlashCardGame />

│ ├─ Game.tsx

│ ├─ Result.tsx

│ └─ hooks.ts

└─ favorite/ # 기존 ‘최애찾기’ 위치 이동

2. Task 목록 업데이트

ID User Story & Acceptance Criteria File/위치

⓫ HUB‑MENU As a user, when I click “미니게임” in the main nav, I see 두 개의 버튼(최애찾기, 플래시카드).

☑ 버튼 hover → Accent Glow

☑ 반응형 Grid(2열 → 1열) pages/minigame/index.tsx

⓬ FLASHCARD‑GAME I play 10 문제를 푼 뒤 점수를 확인하고 기록한다.

☑ 카드 이미지 + 4지선다(랜덤 순서)

☑ 정답 시 1점, 오답 0점 → 10문제 종료 후 Result 화면

☑ 문제·보기 데이터 중복 금지 pages/minigame/flashcard/Game.tsx

⓭ SCORE‑STORAGE Game 결과가 서버에 기록된다.

☑ Supabase table flashcard\_scores(id, alias, score, created\_at)

☑ 익명 사용자는 로컬 alias prompt 1회 저장(LocalStorage)

☑ supabase-js@2 insert 성공 시 토스트 “성공” pages/minigame/flashcard/hooks.ts

⓮ DAILY‑RANK UTC+9 23:59 기준으로 랭킹이 매일 리셋·갱신된다.

☑ Cloud Function (schedule 59 14 \* \* \*) 가

‑ 당일(KST) 최고 점수 TOP10 추출, flashcard\_daily\_rank(date, list) 테이블 저장

☑ 실패 시 Slack Webhook 알림 /supabase/functions/daily\_rank.ts

⓯ RANK‑VIEW 미니게임 페이지 하단에서 오늘 랭킹을 볼 수 있다.

☑ useRank(date=today) 훅으로 데이터 호출

☑ TOP 3에 색상(금/은/동) 뱃지, 4~10은 숫자 표기 pages/minigame/flashcard/Result.tsx

⓬~⓯ : 이번 스프린트 포함(P1) 여부는 PO 판단. 최소 ⓬⓭까지는 필수.

3. 주요 스니펫

3‑1. FlashCard 훅 (문제 로직)

ts

복사

편집

// hooks.ts

import { shuffle } from "@/modules/utils";

import { useState } from "react";

import charList from "@/data/characters.json";

export function useFlashCard(rounds = 10){

const pool = shuffle(charList).slice(0, rounds);

const [step,setStep] = useState(0);

const [score,setScore] = useState(0);

const current = pool[step];

const choices = shuffle([

current,

...shuffle(charList.filter(c=>c.id!==current.id)).slice(0,3)

]);

function answer(id:number){

if(id === current.id) setScore(s=>s+1);

if(step < rounds-1) setStep(s=>s+1);

else return true; // finished

return false;

}

return {step,score,current,choices,answer,finished:step>=rounds};

}

3‑2. Supabase 삽입

ts

복사

편집

// hooks.ts (add)

import { createClient } from "@supabase/supabase-js";

const supabase = createClient(import.meta.env.VITE\_SUPABASE\_URL,

import.meta.env.VITE\_SUPABASE\_ANON);

export async function saveScore(alias:string, score:number){

const {error} = await supabase

.from("flashcard\_scores")

.insert({alias, score});

if(error) throw error;

}

3‑3. Cloud Function 스케줄 (Supabase Edge Functions)

ts

복사

편집

// functions/daily\_rank.ts

import { createClient } from "https://esm.sh/@supabase/supabase-js";

const supabase = createClient(Deno.env.get("SUPABASE\_URL")!,

Deno.env.get("SUPABASE\_SERVICE\_ROLE")!);

export default async () => {

const today = new Date().toISOString().slice(0,10); // UTC date

const {data} = await supabase

.from("flashcard\_scores")

.select("alias,score,created\_at")

.gte("created\_at", `${today}T14:59:59Z`) // 00:00 KST

.order("score",{ascending:false})

.order("created\_at",{ascending:true})

.limit(10);

await supabase.from("flashcard\_daily\_rank")

.upsert({date: today, list: data});

};

4. 환경 변수 (.env)

env

복사

편집

VITE\_SUPABASE\_URL=https://xxxxx.supabase.co

VITE\_SUPABASE\_ANON=public-anon-key

5. 디자인 토큰 추가

scss

복사

편집

// tokens.scss (추가)

$gold : #FFD700;

$silver:#C0C0C0;

$bronze:#CD7F32;

랭킹 1~3위 뱃지에 사용.

### 결론 ― 핵심 보완 포인트

- \*\*Acceptance Criteria(검수 기준)\*\* 와 \*\*코드 Snippet\*\*을 각 Task에 붙여 \*\*애매함 0%\*\*.

- \*\*토큰·변수·디렉터리 구조\*\*를 먼저 명시해 두면 kilo code가 미리 scaffold 생성 가능.

- 상위 README에 \*\*우선순위·브랜치 전략·스타일 가이드\*\* 세 줄 요약 추가 (이미 예시 기재).

이 정도 수준이면 자동 리팩터 도구든, Junior Dev든 \*“헤매지 않고”\* 바로 작업을 시작할 수 있습니다. 추가 세부가 필요하면 특정 Task ID를 지목해 질문하도록 안내하시기 바랍니다.

::contentReference[oaicite:0]{index=0}