

FocusQuest 개발 1회차 보고서

보고 일자: 2025년 11월 3일 작성자: 한상범

1. 개요

- 본 보고서는 'FocusQuest' 프로젝트의 첫 번째 개발 세션 진행 상황을 요약하고 향후 계획을 공유하기 위해 작성되었음.
- 기존 'focus-sprint-coach' 프로젝트를 기반으로, 게임화(Gamification) 메커니즘을 적용하여 사용자 참여와 동기 부여를 강화하는 것을 목표로 함.
- 금일, 프로젝트 초기 설정 및 스프린트 1의 핵심 게임화 기능(레벨, XP, 스트릭, 배지)의 기반을 구현하고, 빌드 및 버전 관리를 완료함.

2. 완료된 작업

가. 프로젝트 초기 설정

- **focusquest** 프로젝트 생성:
 - **focus-sprint-coach**의 전체 코드를 **focusquest** 디렉토리로 복사하여 새 프로젝트의 기반을 마련함.
 - **npm install**을 실행하여 프로젝트에 필요한 모든 종속성을 설치함.

나. 스프린트 1: 핵심 게임화 기능 구현

- **Zustand** 스토어 확장:
 - **sessionStore.ts**에 **gamification** 상태 객체를 추가하여 레벨, XP, 크리스탈, 배지, 스트릭 등 게임화 관련 데이터를 관리할 수 있도록 확장함.
 - **types/gamification.ts** 파일을 신설하여 관련 데이터 모델(GamificationState, Badge, Skill)을 정의함.
- **XP** 및 레벨 시스템 구현:
 - 세션 완료 시 시간에 비례하여 XP를 획득하고, 누적 XP에 따라 레벨이 상승하는 로직을 **sessionStore.ts**에 구현함.
 - **utils/gamification.ts** 파일을 신설하여 레벨 계산(**calculateLevel**) 및 다음 레벨까지 필요한 XP 계산(**xpToNextLevel**) 유틸리티 함수를 구현함.
- 스트릭 카운터 구현:
 - 연속 접속일을 추적하는 스트릭(streak) 시스템 로직을 **sessionStore.ts**에 구현함.

- 기본 배지 시스템 구현:
 - `constants/badges.ts` 파일을 신설하고, '첫 퀘스트', '워크 워리어' 등 5개의 기본 배지 정의 및 획득 조건을 구현함.
 - 세션 완료 후 배지 획득 조건을 자동으로 확인하는 `checkAchievements` 함수를 `sessionStore.ts`에 구현함.
- UI 연동:
 - `TimerPanel.tsx` 컴포넌트를 수정하여 사용자의 현재 레벨과 XP를 표시하도록 UI를 업데이트함.

다. 빌드 및 버전 관리

- `npm run build` 명령을 실행하여 프로젝트 전체의 타입스크립트 오류를 해결하고 빌드에 성공함.
- `context.md` 파일과 `focusquest/` 디렉토리의 모든 변경 사항을 Git에 추가하고, "feat: Initialize FocusQuest project with gamification core" 메시지로 커밋함.
- `git push origin main` 명령을 통해 모든 변경 사항을 원격 저장소에 푸시 완료.

3. 현재 진행 상황

- 구현 현황: 스프린트 1의 핵심 목표였던 게임화 코어 기능의 기반 구현이 완료되었으며, 애플리케이션이 오류 없이 빌드되는 안정적인 상태임.
- 발생 이슈: 개발 초기 단계에서 다수의 타입스크립트 오류가 발생했으나, 타입 정의 수정, `import type` 사용, 로직 수정 등을 통해 모두 해결 완료함.
- 조치 사항: 모든 작업 내용을 Git을 통해 안전하게 버전 관리하고 원격 저장소에 백업함.

4. 향후 계획

가. 1순위 목표: 스프린트 2 개발 착수

- `context.md`에 정의된 로드맵에 따라, 다음 스프린트인 '시각적 시스템(Visual Systems)' 개발에 착수할 예정임.
 - 게임화 패널(`GamificationPanel.tsx`) 컴포넌트 신설.
 - 크리스탈 수집 및 스킬 잠금 해제 UI 구현.
 - 배지 그리드 패널 및 배지 획득 시 토스트 알림 기능 구현.

나. 중기 목표: 테스트 및 고도화

- 신규 게임화 기능에 대한 단위 및 통합 테스트 코드를 작성하여 기능의 안정성을 확보함.

- 번들 크기 경고 등 빌드 과정에서 나타난 최적화 필요 사항에 대해 검토하고 개선 작업을 진행함.
-

5. 특이사항 및 이슈

- 현재 특별한 블로커(**Blocker**)는 없음. 프로젝트가 계획대로 순조롭게 진행 중임.