TravelMate AI - Git 브랜치 전략 설명서

# 1. 브랜치 전략 개요

TravelMate AI 프로젝트는 협업 중심의 팀 프로젝트로, Git 버전관리를 명확하게 하기 위해 다음과 같은 브랜치 전략을 사용합니다.

기본 구조는 Git Flow를 단순화하여 개발 효율성과 코드 통합을 수월하게 관리합니다.

# 2. 브랜치 구조

- main: 최종 배포되는 안정적인 코드가 저장되는 브랜치입니다.

- dev: 개발자들이 기능을 통합하기 위한 테스트 브랜치입니다.

- feature/\*: 새로운 기능 개발 시 사용하는 브랜치입니다. 예: feature/login-api, feature/chat-ui 등

- hotfix/\*: 배포 중 발견된 긴급 버그를 수정할 때 사용하는 브랜치입니다.

# 3. 작업 흐름

1. 각 개발자는 feature/ 브랜치를 생성하여 기능 개발을 수행합니다.

2. 개발 완료 후 dev 브랜치에 Pull Request를 생성하고 코드 리뷰를 요청합니다.

3. 팀장이 dev에 merge한 뒤, 기능이 안정화되면 main으로 병합합니다.

4. 배포는 항상 main 브랜치 기준으로 진행합니다.

# 4. 브랜치 명명 규칙

- 기능 개발: feature/{기능이름} → 예: feature/schedule-generation

- 디자인: feature/{ui-설명} → 예: feature/plan-card-ui

- 버그 수정: hotfix/{이슈설명} → 예: hotfix/fix-login-redirect

# 5. 커밋 메시지 규칙

- [Feat] 기능 추가

- [Fix] 버그 수정

- [Refactor] 리팩토링

- [Docs] 문서 작성

- [Chore] 기타 설정 및 빌드 관련

# 6. 기타 관리 규칙

- 매일 작업 시작 전 git pull

- 작업 종료 후 push + PR 생성

- conflict 발생 시 직접 해결 또는 리뷰어와 협의

- PR 병합 전 반드시 동작 테스트 수행