

# 학습 내용 보고서

대전 1반 9팀 이상호

# 1. React 학습 내용

React 를 효과적으로 사용하기 위해 아래의 주요 개념과 기능들을 학습하였습니다:

## 1.1 React 환경설정

- 개발 환경 구축: Node.js 설치, npm 활용.
- 프로젝트 생성: create-react-app 또는 Vite 사용.
- 개발 및 빌드 환경 최적화.

## 1.2 State

- State 의 정의와 역할 이해.
- useState 훅을 사용한 상태 관리.
- 상태 변화에 따른 UI 업데이트 구현.

## 1.3 Component

- 함수형 컴포넌트와 클래스형 컴포넌트의 차이점 학습.
- 재사용 가능한 컴포넌트 작성.
- 부모-자식 간 데이터 전달.

## 1.4 Map

- 배열 데이터를 기반으로 동적인 리스트 생성.
- JSX 내에서 map() 함수 활용법.
- Key 속성의 중요성 이해.

## 1.5 Props

- Props 를 통한 컴포넌트 간 데이터 전달.
- Props 의 불변성.
- 디폴트 Props 설정.

## 1.6 React Router

- React Router 를 활용한 SPA 구현.
- 동적 라우팅 및 URL 파라미터 처리.
- useNavigate 와 Link 를 활용한 페이지 이동.

## 1.7 Styled Components

- CSS-in-JS 의 개념 이해.

- Styled Components 를 사용한 컴포넌트 스타일링.
- 테마 설정 및 props 를 활용한 동적 스타일링.

## 1.8 LifeCycle 및 useEffect

- React 컴포넌트 생명주기 이해.
- useEffect 를 사용한 사이드 이펙트 처리.
- 의존성 배열을 활용한 최적화.

## 1.9 Ajax

- axios 또는 fetch 를 사용한 API 호출.
- 비동기 데이터 처리 및 로딩 상태 관리.
- 에러 처리 및 데이터 반영.

## 1.10 Redux

- 전역 상태 관리의 필요성 이해.
- Redux 의 구성 요소: Store, Reducer, Action.
- Redux Toolkit 을 활용한 간소화된 상태 관리.

## 1.11 Async

- 비동기 작업 처리와 Promise 이해.
- async/await 구문 활용.
- Redux 와 Thunk 를 사용한 비동기 로직 구현.

---

## 2. GitLab & Jira 협업 툴 학습

효율적인 협업을 위해 GitLab 과 Jira 를 연동하여 사용하며, 다음 개념과 전략을 학습하였습니다:

### 2.1 스토리, 백로그, 스프린트 개념 학습

- Jira 에서 스토리 작성 및 작업 분할.
- 백로그 관리를 통한 작업 우선순위 결정.
- 스프린트를 통한 단기 목표 설정 및 관리.

### 2.2 GitLab 및 Jira 연동 학습

- GitLab 과 Jira 연동을 통한 이슈 관리.
- 커밋 메시지 및 Merge Request 에 이슈 번호 연결.

- 작업 흐름의 가시성 향상.

## 2.3 Git Branch 전략 수립

- Git Branch 구조 설계:
  - main: 배포 가능한 코드.
  - develop: 통합 개발 브랜치.
  - Feature 브랜치: 기능별 브랜치 생성.
- 브랜치 명명 규칙 설정 (예: feature/, hotfix/ 등).

## 2.4 Merge Request 전략 수립

- Merge Request 생성 시 체크리스트:
  - 코드 리뷰 요청.
  - 테스트 통과 여부 확인.
  - Jira 이슈 연결.

## 2.5 Approve 규칙 수립

- 최소 2 명의 리뷰어 승인 필요.
- 코드 품질과 테스트 기준 만족 확인.
- Merge 전 충돌 해결 필수.