

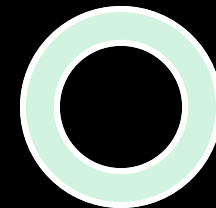
넷플릭스 서비스 클론 코딩

풀스택 14회차 나니가스키~?(3팀)

유현진

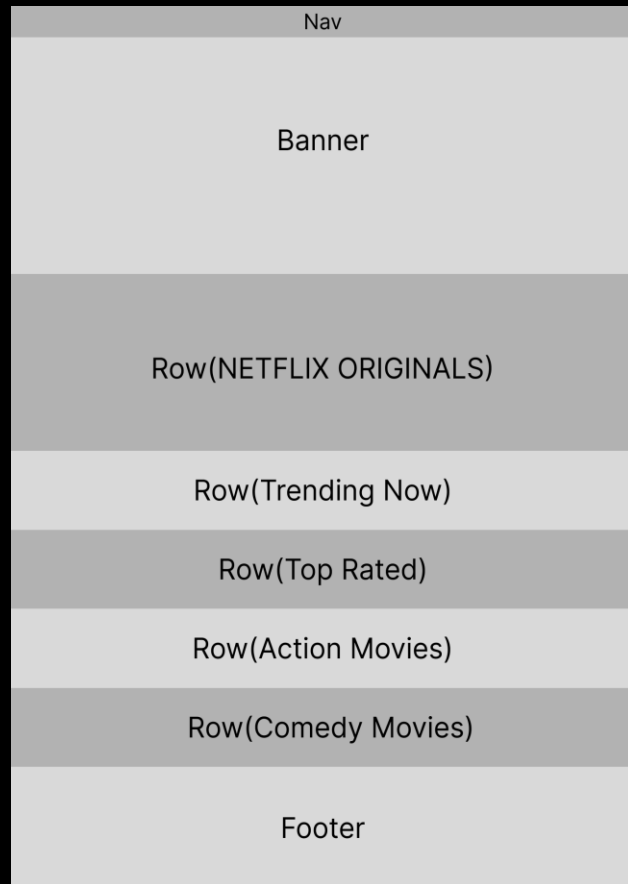
구현 목표

- React 기반 웹 개발
- TypeScript 활용
- 도커(Docker)를 활용한 배포
- AWS 환경 이해 및 활용

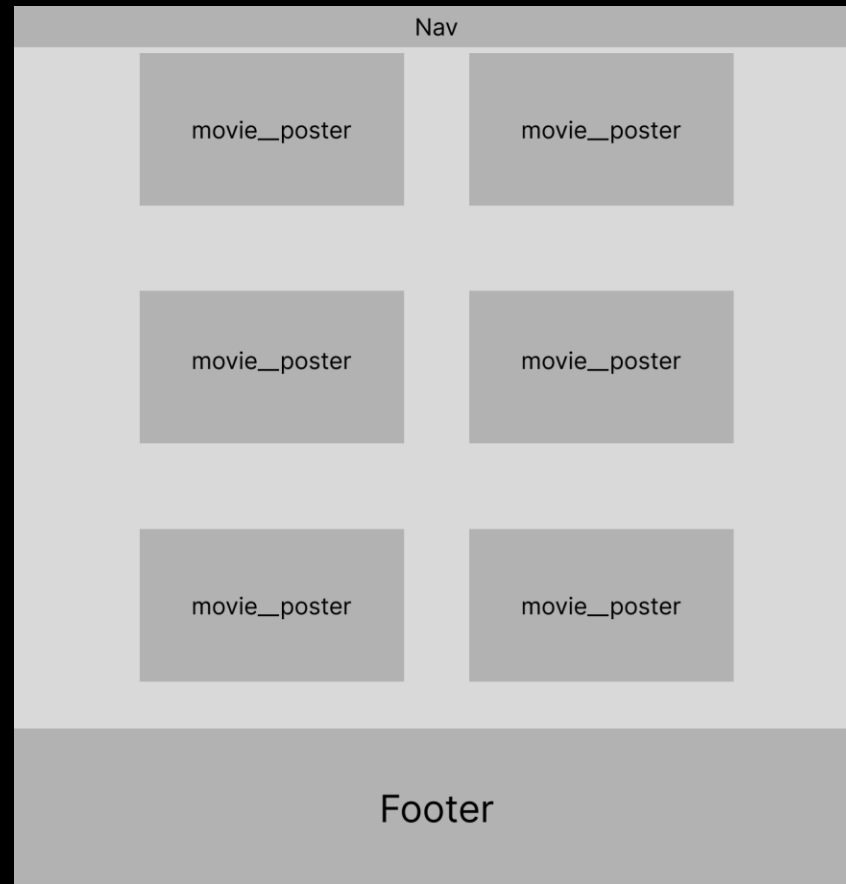




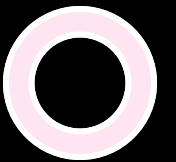
와이어프레임



MainPage



SearchPage

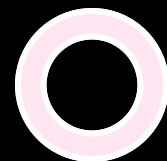




React + TypeScript 기반 웹 개발

```
src > TS movie.ts > ...
1  export interface Movie {
2      id: number;
3      name?: string;
4      title?: string;
5      poster_path?: string;
6      backdrop_path?: string;
7      media_type?: string;
8      original_name?: string;
9      overview?: string;
10     videos?: {
11         results: { key: string }[]
12     };
13     [key: string]: any;
14 }
15
```

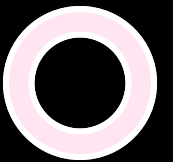
- API 응답 데이터를 타입으로 정의
- 정확한 타입 보장과 안정성 확보



React Router 활용

```
10 const Layout: FC = () => {
11   return (
12     <div>
13       <Nav />
14       <Outlet />
15       <Footer />
16     </div>
17   );
18 };
19
20 const App: FC = () => {
21   return (
22     <div className="app">
23       <Routes>
24         <Route path="/" element={<Layout />}>
25           <Route index element={<MainPage />} />
26           <Route path=":movieId" element={<DetailPage />} />
27           <Route path="search" element={<SearchPage />} />
28         </Route>
29       </Routes>
30     </div>
31   );
32 };
33
34 export default App;
35
```

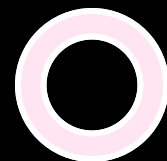
- React Router를 활용해 SPA 방식의 라우팅 구현
- 공통 레이아웃을 유지한 채 각 페이지(Main, Search, Detail)를 동적으로 렌더링





React + Vite 앱 AWS EC2에 Docker로 배포하는 순서

1. GitHub에 코드 업로드
2. Dockerfile 작성
3. AWS EC2 인스턴스 생성 (키 페어 생성)
4. EC2에 SSH 접속
5. Docker 설치
6. Git 설치 및 코드 클론
7. Docker 이미지 빌드 및 컨테이너 실행
8. 브라우저로 접속해 배포 확인



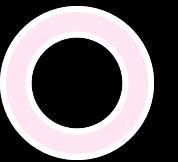


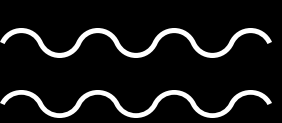
Dockerfile

Dockerfile

```
1 FROM node:18 AS builder
2 WORKDIR /app
3
4 COPY package*.json ./
5 RUN npm ci
6
7 COPY . .
8 RUN npm run build
9
10 FROM nginx:alpine
11 COPY --from=builder /app/dist /usr/share/nginx/html
12 EXPOSE 80
13 CMD ["nginx", "-g", "daemon off;"]
14
```

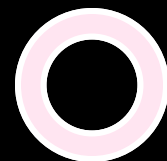
- 멀티 스테이지 빌드를 통해 빌드 환경과 실행 환경 분리
- npm ci 사용 더 빠르고 일관된 의존성 설치 구현



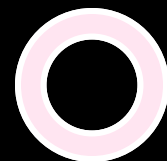
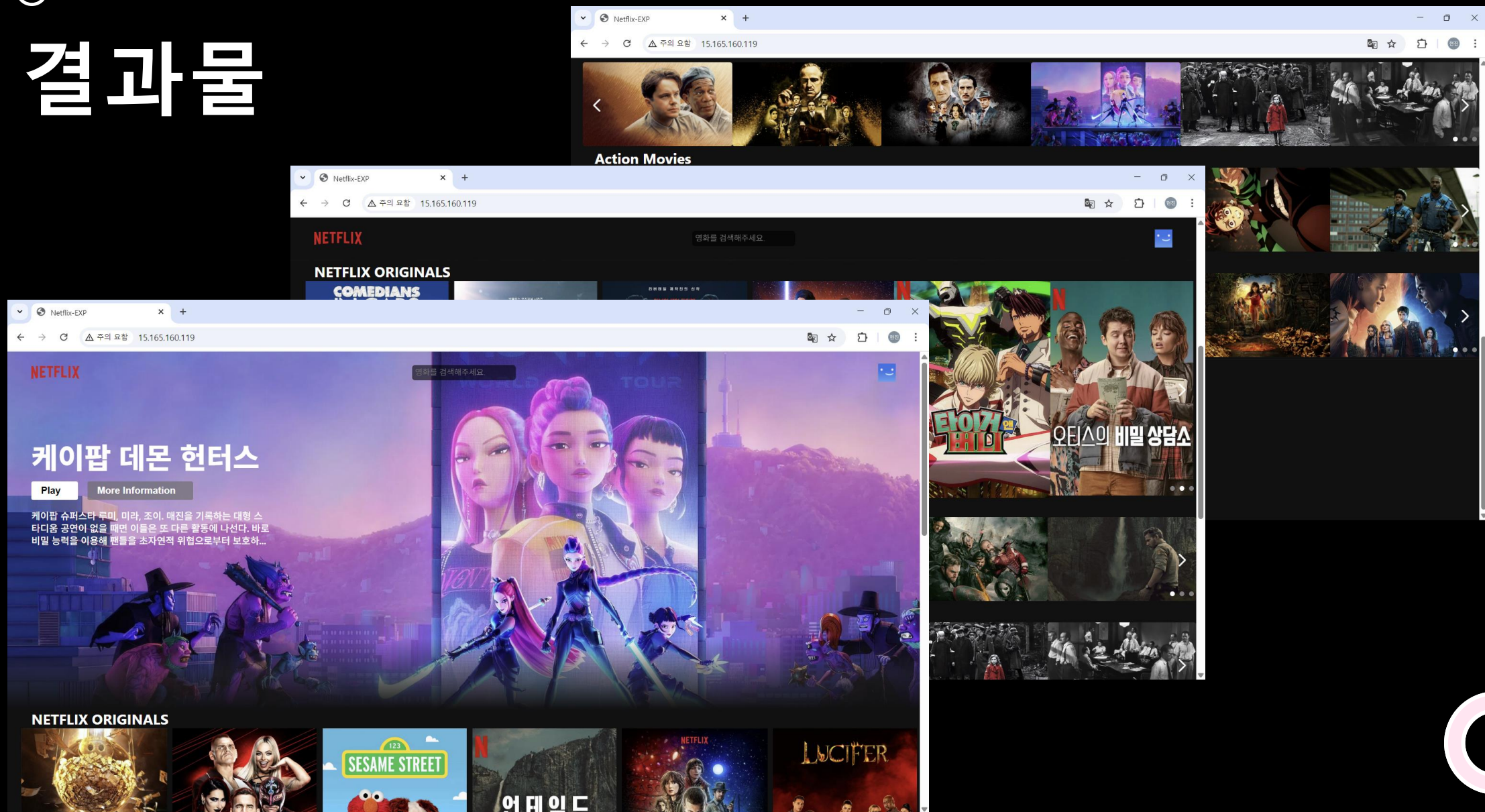


AWS에 배포된 결과물 레포 주소

<http://15.165.160.119/>



결과물



결과물

