

12. 대시보드 프로젝트 생성

새 프로젝트 만들기

Next.js 팀은 `npm` 이나 `yarn` 대신, `pnpm` 사용을 추천합니다.
(`pnpm` 은 더 빠르고 효율적인 패키지 매니저입니다.)

`pnpm` 이 설치되어 있지 않다면, 다음 명령어로 글로벌 설치할 수 있습니다:

터미널

```
npm install -g pnpm
```

Next.js 앱 생성

터미널을 열고, 원하는 폴더로 이동한 다음 다음 명령어를 실행합니다:

터미널

```
npx create-next-app@latest nextjs-dashboard --example  
"https://github.com/vercel/next-learn/tree/main/dashboard/starter-example"  
--use-pnpm
```

이 명령어는:

- `create-next-app` CLI를 사용해 Next.js 프로젝트를 초기화합니다.
- `--example` 플래그로 이 강좌용 스타터 예제를 기반으로 프로젝트를 생성합니다.

프로젝트 구조 탐색하기

이 강의에서는 처음부터 코드를 작성하기보다는
이미 작성된 코드를 기반으로 학습합니다.
(실제 개발 현장처럼 말이죠.)

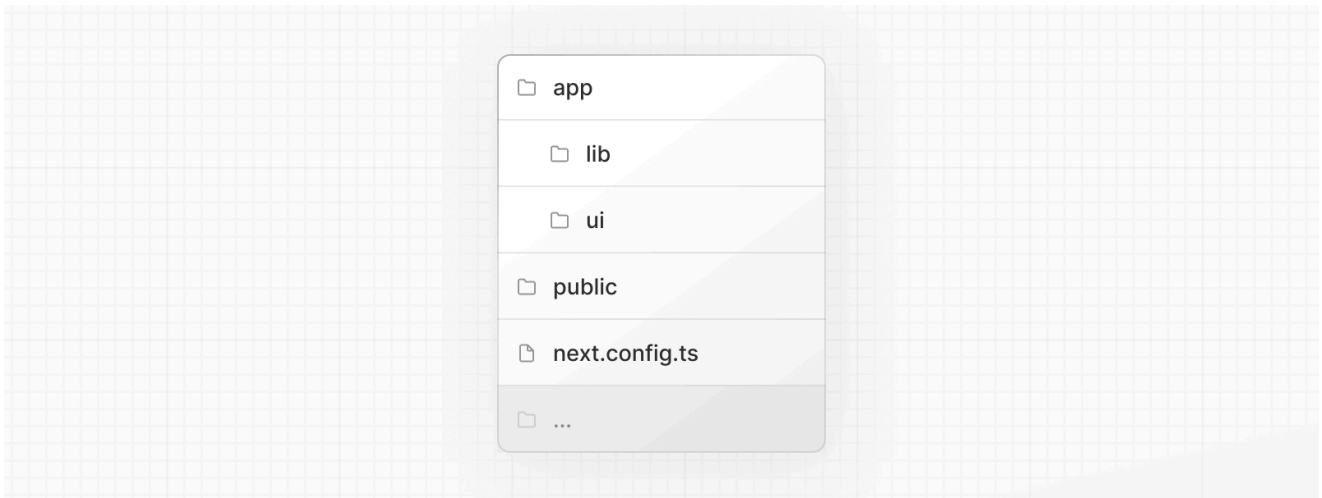
설치가 끝나면 프로젝트를 열고, `nextjs-dashboard` 디렉토리로 이동합니다:

터미널

```
cd nextjs-dashboard
```

폴더 구조

대시보드 프로젝트의 주요 폴더는 다음과 같습니다:



(※ 정상 복구된 이미지)

- `/app`
모든 라우트(route), 컴포넌트, 로직이 포함됩니다. 주로 이 폴더에서 작업합니다.
- `/app/lib`
데이터 가져오기(fetch)나 공통 유틸 함수 같은 기능 코드가 들어있습니다.
- `/app/ui`
카드, 테이블, 폼 같은 UI 컴포넌트가 들어있습니다. (이미 스타일이 적용되어 있습니다.)
- `/public`
정적 자산(이미지 등)이 들어있습니다.
- `next.config.ts` 등 설정 파일
`create-next-app` 이 자동 생성한 파일이며, 강의에서는 수정할 필요 없습니다.

자유롭게 폴더 구조를 탐색해보세요.

아직 코드를 다 이해하지 못해도 괜찮습니다!

플레이스홀더 데이터

실제 데이터베이스나 API가 준비되기 전, 플레이스홀더 데이터로 UI를 구축할 수 있습니다.

- JSON 파일이나 JavaScript 객체로 가짜 데이터를 만들거나
- [mockAPI](#) 같은 3rd-party 서비스를 사용할 수도 있습니다.

이 강의에서는 `app/lib/placeholder-data.ts` 파일에 기본 데이터가 제공됩니다.

예를 들어 송장(invoices) 데이터는 다음과 같습니다:

`/app/lib/placeholder-data.ts`

```
const invoices = [
  {
    customer_id: customers[0].id,
    amount: 15795,
    status: 'pending',
    date: '2022-12-06',
  },
  {
    customer_id: customers[1].id,
    amount: 20348,
    status: 'pending',
    date: '2022-11-14',
  },
  // ...
];
```

추후 데이터베이스 설정 챕터에서 이 데이터를 이용해 초기 데이터를 "시드(seed)"할 것입니다.

타입스크립트 (TypeScript)

이 프로젝트는 **TypeScript**로 작성되었습니다.

파일 확장자가 `.ts` 또는 `.tsx` 인 이유입니다.

TypeScript를 몰라도 괜찮습니다! 필요한 부분은 친절히 안내해줄 예정입니다.

예를 들어, 데이터베이스에서 반환할 데이터를 다음처럼 타입으로 정의합니다:

/app/lib/definitions.ts

```
export type Invoice = {
  id: string;
  customer_id: string;
  amount: number;
  date: string;
  status: 'pending' | 'paid';
};
```

이렇게 타입을 명확히 정의하면, 잘못된 데이터 형식(예: 문자열 대신 숫자)을 넘기는 실수를 방지할 수 있습니다.

만약 타입 자동 생성을 원한다면,
[Prisma](#)나 [Drizzle](#) 같은 ORM을 사용하는 것도 좋은 방법입니다.

또한 Next.js는 TypeScript 사용을 자동 감지하여 필요한 설정과 패키지를 자동 설치해줍니다.

개발 서버 실행하기

설치한 프로젝트 패키지를 먼저 설치합니다:

터미널

```
pnpm i
```

그 다음, 개발 서버를 실행합니다:

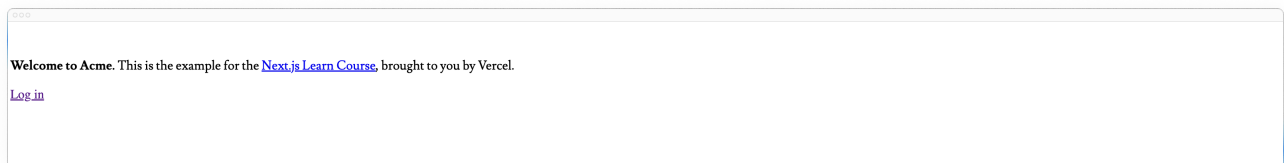
터미널

```
pnpm dev
```

기본적으로 개발 서버는 포트 3000 에서 실행됩니다.

브라우저에서 <http://localhost:3000> 을 열어보세요.

초기 화면은 스타일이 적용되지 않은 상태로 다음과 같이 보입니다:



(※ 정상 복원된 이미지)