

1. 게시판 등록 기능:

- Django에서 `Board` 모델을 정의합니다. 이 모델에는 게시판의 이름, 설명, 생성일 등의 필드를 포함합니다.
- Django Admin에 `Board` 모델을 등록하여 관리자가 게시판을 생성, 수정, 삭제할 수 있도록 합니다.
- Django REST Framework를 사용하여 `Board` 모델에 대한 API 엔드포인트를 구현합니다. 이를 통해 Next.js에서 게시판 목록을 가져오고 게시판 정보를 표시할 수 있습니다.
- 게시판 내의 글쓰기, 삭제, 댓글, 추천, 좋아요 기능은 별도의 모델(`Post`, `Comment`, `Like` 등)을 정의하고, 해당 모델에 대한 API 엔드포인트를 구현합니다.
- 사진 업로드 기능을 위해 `django-storages`와 AWS S3를 연동합니다. `Post` 모델에 이미지 필드를 추가하고, S3에 이미지를 업로드하도록 설정합니다.
- 게시판마다 배너를 등록할 수 있도록 `Banner` 모델을 정의하고, `Board` 모델과 연결합니다. Django Admin에서 배너를 등록하고 관리할 수 있도록 합니다.

2. 대시보드 배너 등록 기능:

- Django에서 `DashboardBanner` 모델을 정의합니다. 이 모델에는 배너의 이미지, 링크, 위치(위, 아래, 좌, 우) 등의 필드를 포함합니다.
- Django Admin에 `DashboardBanner` 모델을 등록하여 관리자가 배너를 등록, 수정, 삭제할 수 있도록 합니다.
- 배너의 순서를 지정할 수 있는 필드(예: `order`)를 추가하여 수동으로 순서를 조정할 수 있도록 합니다.
- Django REST Framework를 사용하여 `DashboardBanner` 모델에 대한 API 엔드포인트를 구현합니다. 이를 통해 Next.js에서 배너 목록을 가져와 표시할 수 있습니다.

3. SEO 태그 정렬 기능:

- Django에서 `SeoTag` 모델을 정의합니다. 이 모델에는 페이지 URL, 메타 태그, 타이틀 태그 등의 필드를 포함합니다.
- Django Admin에 `SeoTag` 모델을 등록하여 관리자가 SEO 태그를 등록, 수정, 삭제할 수 있도록 합니다.
- Django REST Framework를 사용하여 `SeoTag` 모델에 대한 API 엔드포인트를 구현합니다.
- Next.js에서 각 페이지 컴포넌트의 `getInitialProps` 메소드를 사용하여 해당 페이지의 SEO 태그를 가져옵니다.
- 가져온 SEO 태그를 `<Head>` 컴포넌트 내부에 적용하여 페이지의 SEO 최적화를 수행합니다.

위의 방법을 바탕으로 Next.js와 Django를 연동하여 커뮤니티 사이트를 구축할 수 있습니다. 다음은 추가적인 고려 사항입니다:

- Django에서 모델 간의 관계를 설정할 때는 외래 키(ForeignKey), 일대다(OneToMany), 다대다(ManyToMany) 등의 관계를 적절히 사용합니다.

- Django의 시리얼라이저를 사용하여 모델 인스턴스를 JSON 형태로 직렬화하고, Next.js에서 이를 사용할 수 있도록 합니다.
- Next.js에서는 `fetch` 또는 `axios` 라이브러리를 사용하여 Django API와 통신할 수 있습니다.
- 사용자 인증 및 권한 관리를 위해 Django의 인증 시스템과 JWT(JSON Web Token)를 활용할 수 있습니다.
- 프론트엔드에서 상태 관리가 필요한 경우 Redux나 React Context를 사용할 수 있습니다.

이러한 방법을 통해 Next.js와 Django를 효과적으로 연동하고, 커뮤니티 사이트에 필요한 기능을 구현할 수 있습니다. 구현 과정에서 발생하는 구체적인 문제나 질문이 있다면 추가적으로 문의해 주시기 바랍니다.