**React 프로젝트 GitHub → Vercel 배포 순서**

**✅ 1️⃣ GitHub에 프로젝트 올리기**

git init git add . git commit -m "first commit" git branch -M main git remote add origin https://github.com/본인아이디/레포이름.git git push -u origin main

→ GitHub에 코드가 올라갔는지 확인 (main 브랜치)

**✅ 2️⃣ Vercel 접속**

1. 👉 https://vercel.com
2. **“Continue with GitHub”** 클릭 (GitHub 계정으로 로그인)
3. 로그인 후 **“Add New Project”** 클릭

**✅ 3️⃣ GitHub 연결하기**

* Vercel이 자동으로 GitHub 저장소 목록을 보여줍니다.
* 방금 올린 레포를 선택 → **“Import”** 클릭

**✅ 4️⃣ 빌드 설정 자동 감지**

Vercel이 자동으로 React를 인식해서 아래처럼 표시됩니다 👇

| **설정 항목** | **값** |
| --- | --- |
| Framework Preset | React |
| Build Command | npm run build |
| Output Directory | build |

👉 별도로 수정할 필요 없습니다.

**✅ 5️⃣ “Deploy” 버튼 클릭**

* 이제 자동으로 빌드 → 배포됩니다 🚀
* 잠시 후 URL이 생성됩니다.  
  예:
* https:*//my-react-app.vercel.app*

✅ **끝났습니다!**

**💡 이후부터는 자동 배포**

한 번 연결하면,  
GitHub에 코드를 수정하고 push만 해도  
Vercel이 자동으로 감지해서 새 버전을 배포합니다.

예:

git add . git commit -m "update home section" git push

→ 몇 초 후 자동으로 새 버전이 배포됩니다 🎯

**⚙️ 추가 팁 (중요!)**

| **상황** | **해결 방법** |
| --- | --- |
| 라우터(/about, /project) 새로고침 시 404 | vercel.json 파일에 rewrite 설정 추가 |
| 백엔드 API 따로 있음 | .env 파일에 API\_URL 넣고 fetch 시 사용 |
| 환경 변수 필요 (예: API\_KEY) | Vercel Dashboard → “Environment Variables” 탭에서 설정 |

예시 – vercel.json (리액트 라우터 에러 방지용)

{ "rewrites": [ { "source": "/(.\*)", "destination": "/" } ] }

**✅ 정리**

| **단계** | **설명** |
| --- | --- |
| 1 | GitHub에 코드 push |
| 2 | Vercel 로그인 (GitHub 계정) |
| 3 | “Add New Project” → 내 저장소 선택 |
| 4 | 자동 설정 확인 (React 감지) |
| 5 | “Deploy” 클릭 → 완료 🎉 |