# Bio-FormulaGPT 개발자 가이드 (상세)

## 1. 플랫폼 아키텍처

Bio-FormulaGPT는 AI 모델(OpenAI GPT-4)을 활용하여 수식을 분석, 해석하고 시각적으로 제공하는 플랫폼입니다. 전체 아키텍처는 다음과 같은 계층으로 구성됩니다:  
  
[사용자] ⇄ [Frontend(Vue/React)] ⇄ [Backend(NestJS)] ⇄ [OpenAI API / 외부 API(PubChem, NCBI)]  
  
- 프론트엔드는 사용자 입력(수식, 질문)을 수집하고 시각화(UI + MathJax/Katex)를 담당합니다.  
- 백엔드는 수식 API 관리, GPT 연동, 사용자 인증 및 외부 API 연동을 포함합니다.

## 2. 주요 모듈 설명 및 흐름도

🧩 핵심 모듈:  
- Formula API (생물학/화학 수식 CRUD 및 설명 요청)  
- GPT Service (explain, quiz, code, tutor 대화)  
- External API (PubChem: 화합물 정보 / NCBI: 유전자 정보)  
  
📊 순서도 예시 (수식 해설 요청):  
1. 사용자가 수식을 선택하고 "해설 요청" 클릭  
2. 프론트에서 `/gpt/explain` 엔드포인트에 POST 요청  
3. NestJS가 OpenAI API에 수식 설명 prompt 요청  
4. 응답을 받아 프론트에 전달 → MathJax로 렌더링  
  
🔁 알고리즘 개요 (수식 추천):  
- 사용자가 자주 본 수식의 태그를 추적  
- 태그 기반 유사 수식을 DB에서 추출  
- 추천 수식 목록 반환

## 3. 설계 방식 (확장성/모듈화)

- NestJS의 모듈 기반 아키텍처 활용 → bio, chem, gpt, auth 등으로 분리  
- 모든 GPT 기능은 `gpt.service.ts` 내에서 공통 처리 → Prompt만 변경  
- 외부 API 연동은 별도 모듈화 (PubChem, NCBI)  
- 수식 데이터는 Tag + Category 기반 분류 → 향후 커리큘럼 확장 가능

## 4. 유지보수 및 배포 방법

✅ 유지보수 팁:  
- GPT Prompt 버전이 변경되면 gpt.service.ts 내부 Prompt들을 업데이트  
- 수식 DB에 추가/수정은 관리자 페이지 또는 SQL로 처리  
- 외부 API가 변경될 경우 `/external/` 내 서비스만 수정하면 됨  
  
✅ 배포 방식:  
- Vercel: frontend 자동 배포 (push → CI/CD 실행)  
- Render: backend + PostgreSQL 연결 자동화  
- Hugging Face: app.py 파일만 있으면 챗봇 데모 실행 가능  
  
✅ 테스트:  
- Postman으로 API 단위 테스트  
- GPT 기능은 key 분리 환경에서 테스트 권장  
- 커리큘럼은 JSON 파일 단위 테스트 후 DB화 가능