

KSBI-BIML 2026

Bioinformatics & Machine Learning(BIML)
Workshop for Life Scientists

생명정보학 & 머신러닝 워크샵(온라인)



AI-assisted Web-based Bioinformatics Tool Development

박정빈 _ 부산대학교



KSBI
KOREAN SOCIETY FOR
BIOINFORMATICS

| 한국생명정보학회



본 강의 자료는 한국생명정보학회가 주관하는 BIML 2026 워크샵을 목적으로
제작된 것으로 해당 목적 이외의 다른 용도로 사용할 수 없음을 분명하게 알립니다.

이를 다른 사람과 공유하거나 복제, 배포, 전송할 수 없으며 만약 이러한 사항을 위반할 경우
발생하는 **모든 법적 책임은 행위자 본인에게 있음**을 알립니다.

KSBI-BIML 2026

Bioinformatics & Machine Learning (BIML) Workshop for Life Scientists

한국생명정보학회가 주최하는 BIML-2026 동계 Bioinformatics & Machine Learning 교육 워크숍에 여러분을 초대합니다.

BIML 워크숍은 생명정보학 연구자들이 최신 AI바이오 분야의 인공지능 기반 분석 기술과 바이오 데이터 분석 기법을 이론과 실습을 통해 체계적으로 배울 수 있는 전문 교육 프로그램입니다. 2015년에 시작된 BIML 워크숍은 올해로 12년 차를 맞이하며, 국내 생명정보학 분야의 최초이자 최고 수준의 교육 프로그램으로 자리 잡았습니다. 이번 워크숍은 크게 인공지능바이오(AI바이오) 분야와 디지털바이오 분야, 두 분야로 구성됩니다.

AI바이오 분야에서는 생명정보 분석에 폭넓게 응용되고 있는 다양한 인공지능 기반 자료 모델링 기법을 다룰 예정입니다. 특히, 인공지능 심층학습을 활용한 단백질 구조 예측, 유전체 분석, 신약 개발에 대한 이론 및 실습 강의를 진행됩니다.

또한 디지털바이오 분야에서는 단일세포오믹스, 공간오믹스, 멀티오믹스, 메타오믹스에 대한 강의도 마련되어 있어, 연구자들의 분석 역량 강화에 실질적인 도움을 줄 것으로 기대됩니다.

또한 2024년부터 추가된 의료정보 자료 분석을 다루는 강의를 올해도 지속해서 운영하고자 합니다. 이는 최근 의료정보 자료 분석에 관한 연구 수요 증가를 반영한 것으로, 관련 연구를 수행하는 의과학자 및 의료정보 연구자들에게 유용한 지침을 제공할 것입니다.

또한, 올해도 생명정보학 기술의 다양화에 발맞춰 온라인 강좌를 대폭 확대했습니다. 올해는 무료 강좌 10개를 포함한 총 40개 이상의 강좌가 개설되며, 연구 주제에 맞는 강좌 추천과 강연료 할인 혜택도 제공합니다.

BIML-2026는 국내 주요 연구 중심 대학의 전임 교수 및 각 분야 최고 전문가들의 강의로 구성되어 있으며, 기초 이론부터 최신 연구 동향까지 아우르는 심도 있는 교육의 장이 될 것으로 확신합니다.

여러분의 많은 관심과 참여를 기대합니다!

2026년 2월

한국생명정보학회장 류 성 호

AI-assisted Web-based Bioinformatics Tool Development

과거에는 생명정보학 웹 도구를 개발하기 위해 프론트엔드, 백엔드, 데이터베이스 등 광범위한 웹 기술을 모두 익혀야 했다. 그러나 생성형 AI(ChatGPT, Claude, Gemini 등)의 등장으로 코딩의 장벽이 낮아졌으며, 이제는 아이디어와 논리적 설계 능력만 있다면 누구나 자신만의 분석 도구를 만들 수 있는 시대가 되었다.

이에 본 강의는 복잡한 코드를 직접 한 줄씩 작성하는 전통적인 방식에서 벗어나, AI 코딩 어시스턴트를 활용하여 바이오 데이터 분석 로직을 구현하고 이를 웹 어플리케이션으로 빠르게 서비스화하는 방법을 학습한다. 최신 웹 프레임워크와 AI의 코드 생성 능력을 결합하여, 실무에서 즉시 활용 가능한 수준의 Bioinformatics Tool을 제작하고 배포하는 것을 최종 목표로 한다.

강의는 다음의 내용을 포함한다:

- AI-assisted Coding 환경 구축
- AI를 활용한 간단한 생명정보 데이터 분석 스크립트 작성
- 웹 기반 GUI (Web-based GUI) 개발
- 배포 및 서버 운영

* 교육생준비물:

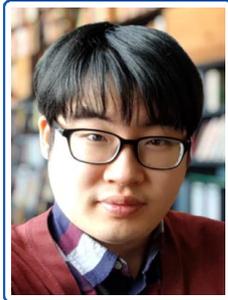
- 노트북 (메모리 8GB 이상, 디스크 여유공간 30GB 이상)
- 유료로 구독한 Anthropic Claude 계정
- GitHub 계정

* 강의 난이도: 초급

* 강의: 박정빈 교수 (부산대학교 의생명융합공학부)

Curriculum Vitae

Speaker Name: Dr. Jeongbin Park



► Personal Info

Name Jeongbin Park
Title Assistant Professor
Affiliation Pusan National University

► Contact Information

Address Room# 411, Kyung-Ahm Engineering Bldg.,
Busandaehak-ro 49, Mulgeum-eup, Yangsan-si,
Gyeongsangnam-do, Republic of Korea
Email jeongbin.park@pusan.ac.kr

Research Interest

Translational bioinformatics, Machine learning and computational genomics

Educational Experience

2012 B. Sc. in Physics, Pusan National University, Korea
2014 M. Sc. in Physics, Seoul National University, Korea
2020 Dr. rer. nat. in Bioscience, Heidelberg University, Germany

Professional Experience

2018-2020 Researcher, Berlin Institute of Health (BIH) at Charite, Germany
2020-2022 Postdoctoral Researcher, German Cancer Research Center (DKFZ), Germany
2022- Assistant Professor, Pusan National University, Korea

Selected Publications (3 maximum)

1. Abyot Melkamu Mekonnen, Kang Seong, et al., Variant-aware Cas-OFFinder: web-based in silico variant-aware potential off-target site identification for genome editing applications, *Nucleic Acids Research* 53(W1), W118-W124
2. Jeongbin Park, et al., Cell segmentation-free inference of cell types from in situ transcriptomics data, *Nature Communications* 12, 3545, 2021
3. Jeongbin Park, et al., Digenome-seq web tool for profiling CRISPR specificity, *Nature Methods* 14 (6), 548-549, 2017

KSBi-BIML 2026

AI-assisted Web-based Bioinformatics
Tool Development

Jeongbin Park
Pusan National University

서론

- 생명정보학에서 왜 웹 개발이 중요한가?
 - 대부분의 생명정보학 기반 도구들은 리눅스 환경을 염두에 두고 개발되며 그 사용법이 어려운 편
 - 따라서 이에 대한 접근성을 높이는 것은 전반적인 과학 발전에 중요한 영향을 끼치게 됨
 - 따라서 도구나 알고리즘을 개발하는 경우, 웹 인터페이스를 추가한다면 논문의 임팩트가 크게 강화됨

서론

- 왜 웹 개발인가?
 - **AI는 웹 개발을 잘 함** (경험상 다른 코딩 작업보다 훨씬 더 잘 하는 듯함)
 - AI는 반복적이고 인간이 지루함을 느낄 만한 작업을 특히 잘 수행하는데, 웹 개발의 경우 코딩에 반복적인 패턴이 존재하며, 이를 AI가 매우 잘 학습하여 인간을 수월하게 대체하는 편
 - 따라서 **기본적인 웹 개발 이론**과 자주 사용되는 **코딩 패턴의 명칭을 이해**하면 대부분의 작업을 자동화할 수 있음

3

강의목표

- **Chapter 1. 웹 개발 이론**
 - HTML, CSS, ECMAScript (JavaScript) 등 용어에 대한 이해
 - 서버/클라이언트 모델에 대한 이해
 - 웹 표준 및 접근성에 대한 이해
 - 프론트엔드/백엔드에 대한 이해
- **Chapter 2. 웹 개발 실전**
 - AI 도우미 (Claude Code) 를 활용한 웹 개발 경험

4

Chapter 1. 웹 개발 이론

인터넷의 탄생

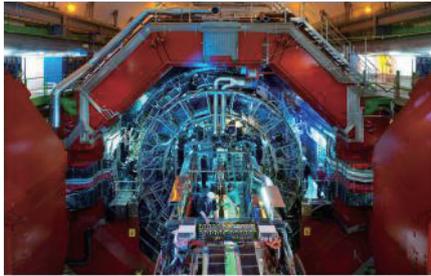
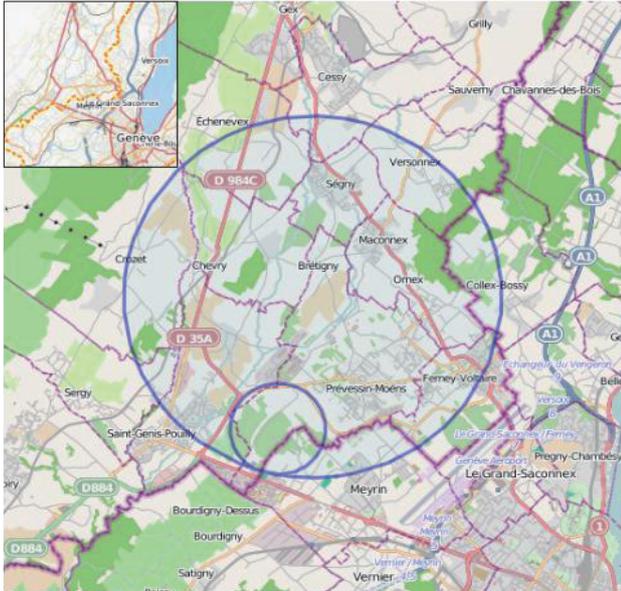


Image from Wikipedia

인터넷의 탄생

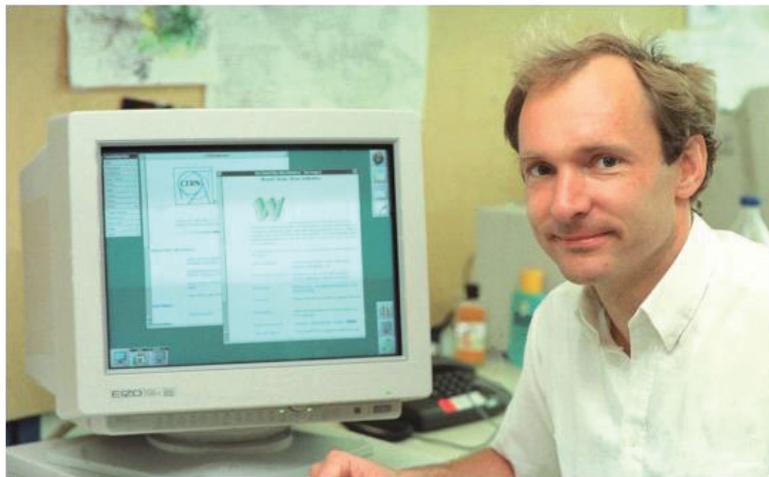


Image from Wikipedia

7

인터넷의 탄생

- 팀 버너스리



<https://home.cern/science/computing/birth-web>

8

인터넷의 탄생

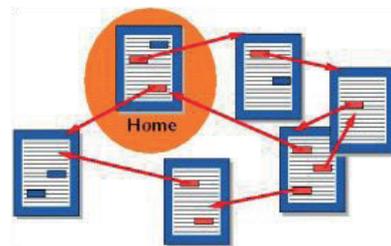


9

하이퍼링크와 하이퍼텍스트

- '하이퍼링크 (Hyperlink)' 또는 '링크 (Link)'

- 문서와 문서를 이어주는 연결고리



- 하이퍼텍스트 (Hypertext)

- 하이퍼링크로 서로 연결된 텍스트 문서

10

Hypertext Markup Language (HTML)

- 1990년경 처음 제안, 1993년 공식 발표
- 당시 CERN에서 사용중이었던 SGML이라는 문서 포맷 기반

```
<!DOCTYPE motd [ <!E
<motd>
<!-- created: 2003-12-12-->
<sentence>Do not throw
out the <keep>baby</>
with the
<refuse>dirty</>,
<refuse>stinky</>,
<refuse>bathwater</>.
</>
<!-- finish this later-->
</motd>
```

SGML

11

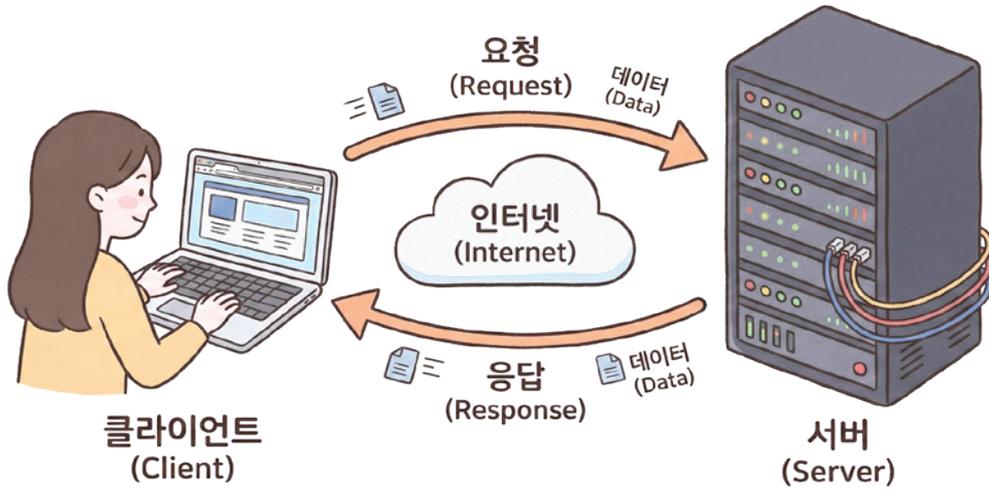
Hypertext Markup Language (HTML)

- 현대의 HTML 문서 구조

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello, world!</title>
  </head>
  <body>
    <p>Hello, world!</p>
  </body>
</html>
```

12

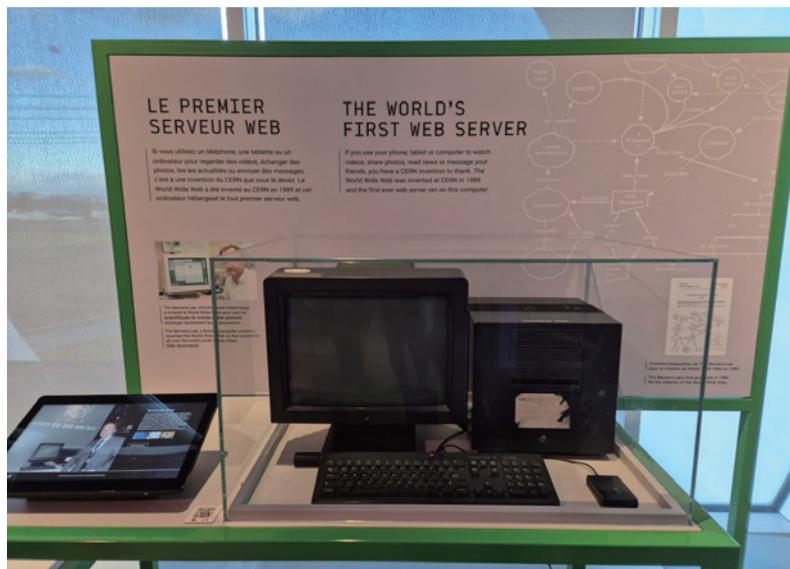
서버와 클라이언트



13

서버와 클라이언트

- 최초의 웹 서버 (CERN)



14

서버와 클라이언트

- 클라이언트의 요청

```
GET / HTTP/1.1
Accept: text/html
```

- 서버의 응답

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close
```

헤더

```
<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

HTML

15

HTTP 응답 코드

- 앞으로 자주 보게 될 코드

- 200: OK
- 404: Not found
- 500: Internal Server Error

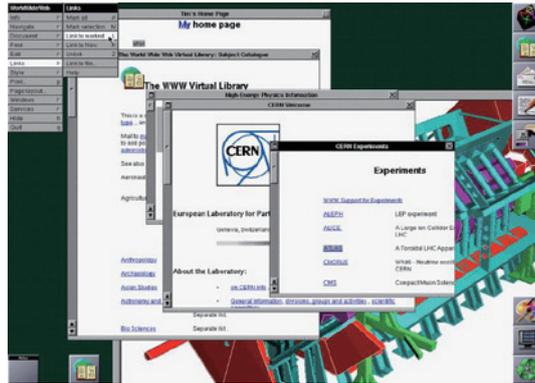
- 일반적 규칙

- 1xx (정보): 요청을 받았으며 프로세스를 계속한다
- 2xx (성공): 요청을 성공적으로 받았으며 인식했고 수용하였다
- 3xx (리다이렉션): 요청 완료를 위해 추가 작업 조치가 필요하다
- 4xx (클라이언트 오류): 요청의 문법이 잘못되었거나 요청을 처리할 수 없다
- 5xx (서버 오류): 서버가 명백히 유효한 요청에 대해 충족을 실패했다

16

월드 와이드 웹 (WWW)

- World Wide Web (WWW)
 - 1990년 발표
 - 서로 복잡하게 연결된 하이퍼텍스트 문서들의 연결 구조를 탐색할 수 있는 정보 시스템
 - 최초의 웹 브라우저 (WorldWideWeb) 등장



<https://www.w3.org/People/Berners-Lee/WorldWideWeb.html>

17

웹 언어의 발전

- JavaScript
 - '넷스케이프'사에서 1995년 발표
 - 정적인 HTML에 동적인 요소를 추가해주는 언어
 - 마케팅 목적으로 당시 가장 유행하던 언어인 Java의 이름을 차용 (그러나 구조적으로는 사실 Java와는 전혀! 아무런! 관련이 없음)
 - `script` 블록 안에 작성 (아래 예시 참조)

```
<script>
  let name = prompt('Input your name!');
  window.alert('Welcome ' + name);
</script>
```

18

웹 언어의 발전

- JavaScript
 - 초창기 자바스크립트는 웹 브라우저마다 표준이 약간씩 전부 달랐음
 - 따라서 모든 웹 브라우저 (인터넷 익스플로러, 파이어폭스, 사파리 등등..)에 맞도록 코드를 각각 별도로 작성해야 했음
 - 그러나 현대에는 모든 브라우저가 준수하기를 권장하는 '표준' 구현이 있으며, 현재 시점에서는 더 이상 브라우저별 차이를 고민하지 않아도 됨
 - 이러한 표준을 'ECMAScript' 라고 부름

19

웹 언어의 발전

- Cascading Style Sheets (CSS)
 - 1996년 발표
 - HTML이 화면에 표현되는 방법을 기술하는 언어

```
<div style="border: 1px solid black; padding: 20px;">
  <span style="color: red; font-weight: bold;">
    WARNING
  </span>
</div>
```



```
<div class="box">
  <span>
    WARNING
  </span>
</div>
```

HTML

```
.box {
  border: 1px solid black;
  padding: 20px;
}

.box span {
  color: red;
  font-weight: bold;
}
```

CSS

20

웹 개발 프로세스

• 초기 웹 개발 프로세스

- 웹 기술 초창기에는 웹 개발이란 HTML, CSS, 자바스크립트를 모두 잘 이해하고 있는 개발자가 디자이너와 협업하여 웹 사이트를 만드는 것을 의미했음

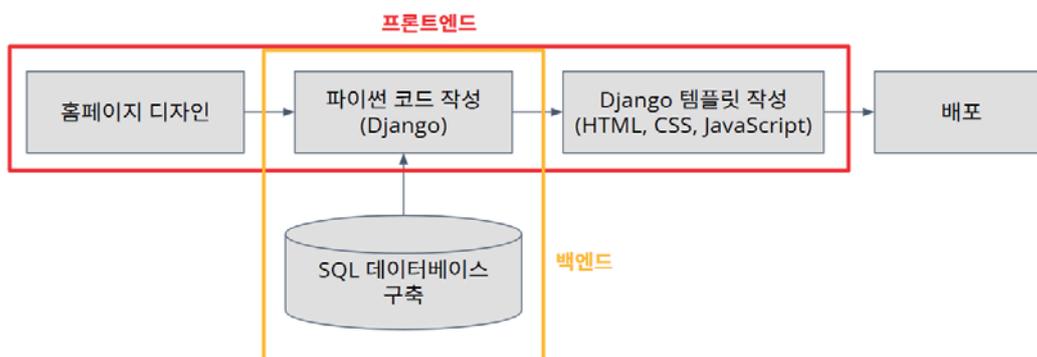


21

웹 개발 프로세스

• 초기 웹 개발 프로세스

- 만약 여기에 데이터를 저장하고 불러와야 할 경우 (회원관리, 게시판 등) 여기에 SQL 데이터베이스를 붙이게 되는데, 이 때 파이썬, 루비 등 범용 언어를 활용해서 데이터에 따라 동적으로 HTML을 생성하는 '웹 프레임워크'가 등장하게 됨
- HTML/CSS/JS인 부분과 아닌 부분을 구분하여 '백엔드', '프론트엔드'라 부르기 시작함



22

웹 개발 프로세스

• 웹 프레임워크

- 한편 세월이 흘러 표준화된 ECMAScript의 구동 성능이 네이티브 C/C++ 에 비견될 정도로 빨라지기 시작함
- 따라서 아예 파이썬 등 언어를 거치지 않고 자바스크립트로 프론트엔드에 해당하는 부분을 완전히 분리하여 작성해버리려는 시도들이 등장

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>HTML Webpage</title>
</head>
<body>
  <header>
    <h1>Hello, World!</h1>
  </header>
  <main>
    <p>This is a basic HTML webpage.</p>
  </main>
  <footer>
    <p>Copyright © 2024. All rights reserved.</p>
  </footer>
</body>
</html>
```



```
<!-- App.svelte -->
<script>
  let name = 'World';
</script>
<header>
  <h1>Hello, {name}</h1>
</header>
<main>
  <p>This is a basic SvelteKit webpage.</p>
</main>
<footer>
  <p>(new Date()).getFullYear() ©. All rights reserved.</p>
</footer>
```

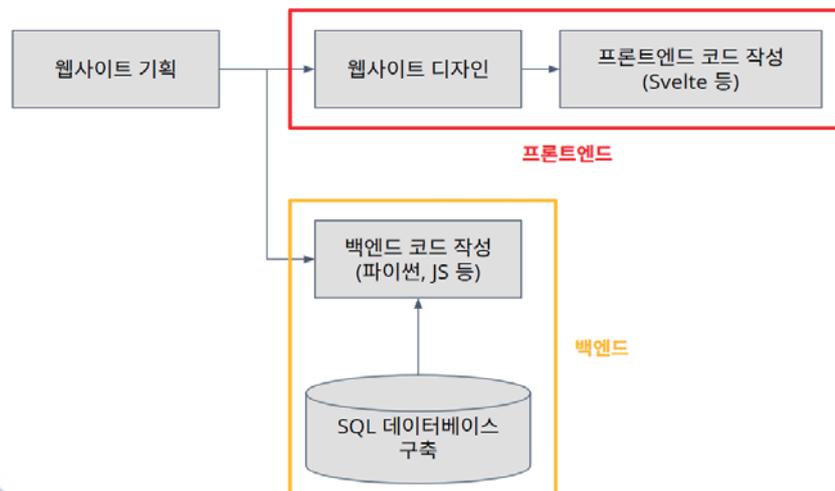
'Svelte'

23

웹 개발 프로세스

• 현대의 웹 개발

- 프론트엔드/백엔드의 완전한 분리
- 현대에는 백엔드 구현 언어로 ECMAScript를 사용하기도 함



24

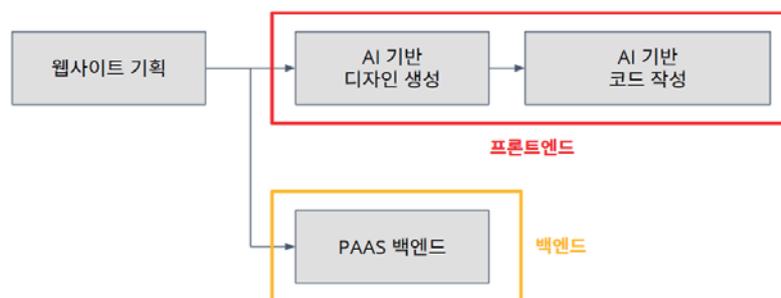
웹 개발 프로세스

- 프론트엔드 엔지니어
 - 화면에 보여지는 부분의 구현 (배치, 애니메이션 등)
 - 백엔드와의 연동 (REST API)
 - HTML, CSS, ECMAScript, 프론트엔드 프레임워크 (React, Svelte 등) 지식
- 백엔드 엔지니어
 - 화면에 보여지지 않는 부분의 구현 (웹 서버, 데이터베이스)
 - 프론트엔드와의 연동 (REST API)
 - 파이썬, 자바 등 기존 언어 + 웹 서버 및 데이터베이스 서버 운영 지식
- 풀스택 엔지니어
 - 둘 다 하는 사람

25

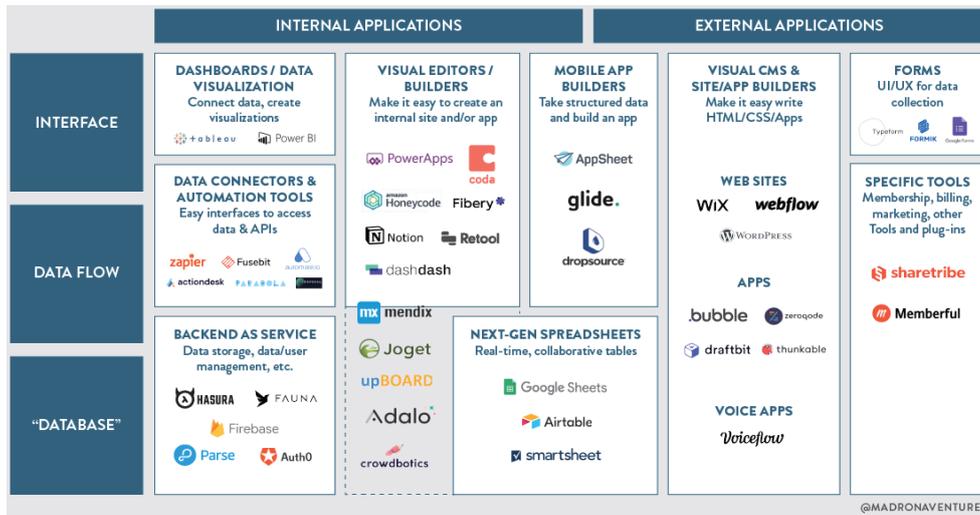
웹 개발 프로세스

- 현대의 웹 개발
 - 이제 프론트엔드 개발은 완전히 AI를 활용하여 개발이 가능
 - 백엔드는 직접 개발할 수도 있지만, 많은 경우 보안 및 데이터 백업 문제 등을 고려하여 백엔드를 직접 개발하지 않고 'Firebase' 등 표준화된 백엔드 플랫폼 (PAAS, Platform as a service) 를 주로 사용함 (PAAS 사용요금은 고객에게 웹사이트 관리비 형태로 징수)



26

웹 개발 프로세스



<https://www.madrona.com/investing-in-low-code-no-code-development-platforms/>

27

웹 사이트 기획

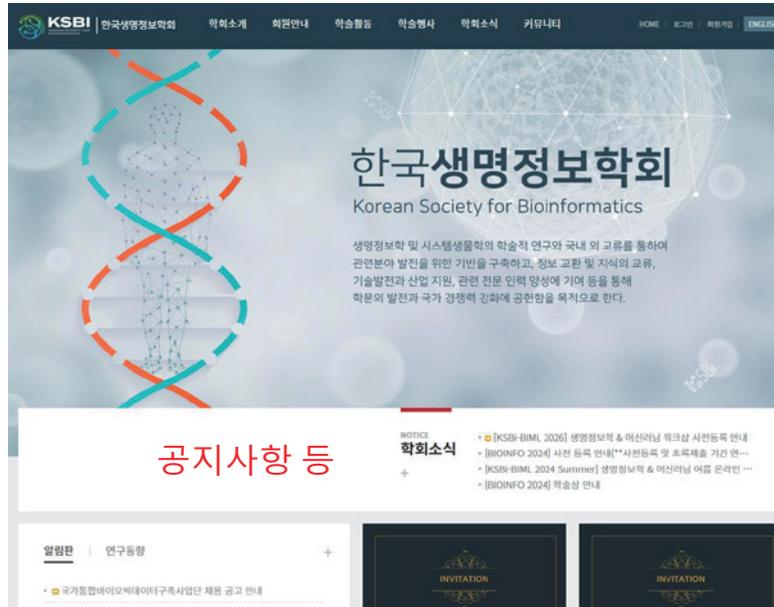
- 랜딩 페이지
 - 웹 사이트의 가장 첫 페이지
 - 방문자의 이목을 끌 수 있을 만한 화려한 디자인
 - Hero section (<https://www.awebco.com/blog/hero-section/>)
 - Carousel (<https://flowbite-svelte.com/docs/components/carousel/>)
 - 기타 다양한 동영상, 사진, 이미지, 애니메이션 등
 - 반드시 첫 페이지에 보여야 하는 필수 정보 (공지사항, 배너 등)

28

웹 사이트 기획

- 랜딩 페이지

거대한
Hero section
(동영상)



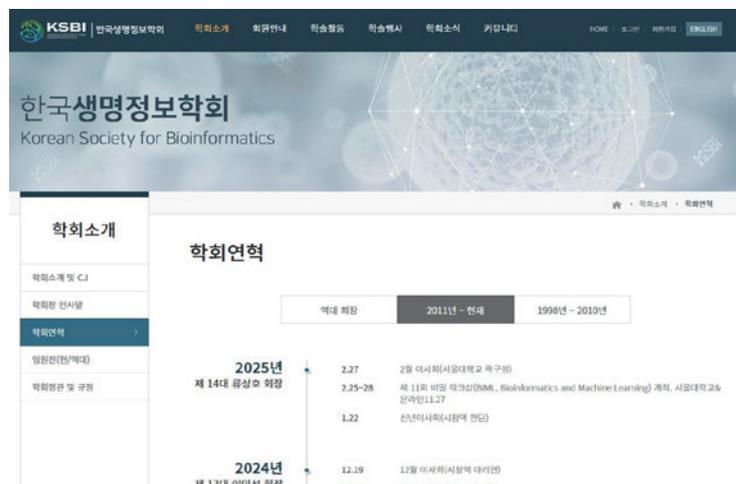
29

웹 사이트 기획

- 일반 페이지

- 랜딩 페이지의 Hero section, Carousel 등의 디자인 요소를 축소하고, 랜딩 페이지를 제외한 모든 페이지에 공통적인 디자인 적용

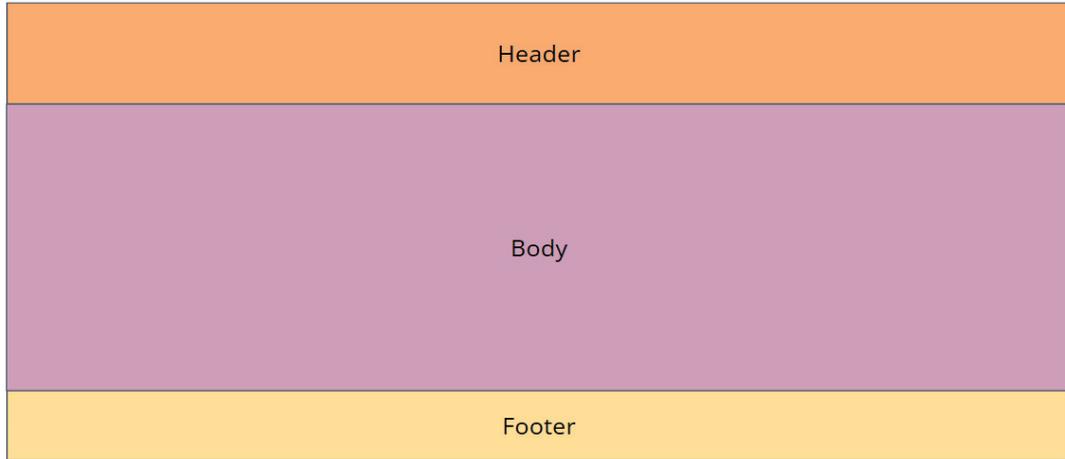
축소된
Hero section



30

웹 사이트 구조

- Header, Body, Footer



31

웹 사이트 구조

- Header, Body, Footer

Header

Body

Footer

The screenshot shows the homepage of the Korean Society for Bioinformatics (KSBI). The header (orange border) contains the KSBI logo, navigation menu, and search bar. The body (purple border) features a main content area with the title '한국생명정보학회' and 'Korean Society for Bioinformatics', a sidebar with a '학회소개' menu, and a main content area with the title '학회소개 및 C.I.' and a large logo. The footer (yellow border) contains contact information, a copyright notice, and a '개인정보처리방침' link.

32

웹 사이트 구조

- Header, Body, Footer 에는 무엇이 들어가야 하는가?
 - Header: 로고, 메인 메뉴 (Navigation Bar), Hero 섹션
 - Body: 주 콘텐츠
 - Footer: 로고 (주로 흑백), 하단 메뉴, 연락처, 주소, 법적 정보, 저작권 정보 등
- 실제 구현 시 각각을 별도의 파일로 나누어서 구현함

33

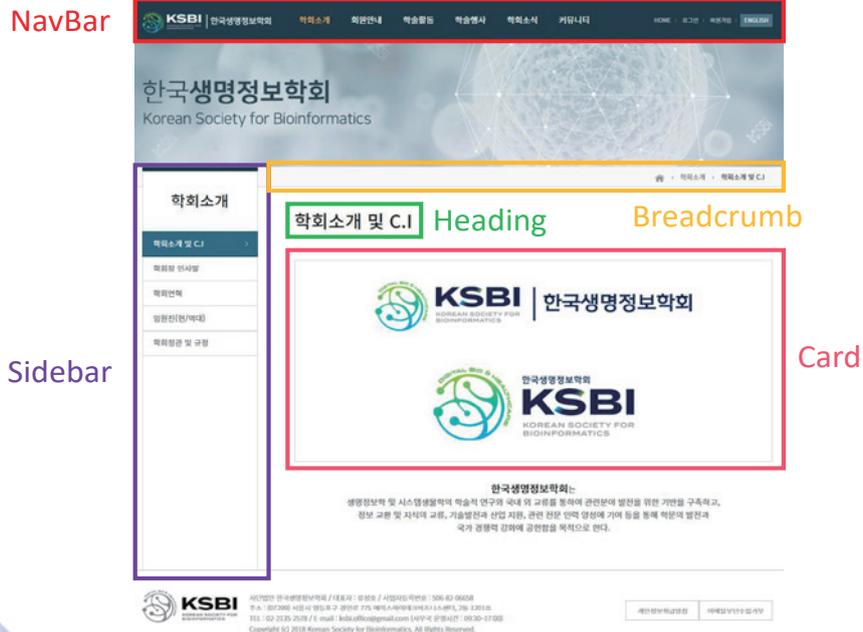
웹 사이트 구조

- UI 컴포넌트
 - 자주 사용되는 웹 구성요소를 패턴화하여 반복 코딩을 줄임
 - 2011년 등장한 '트위터 부트스트랩' 프레임워크에서 유래된 것들이 많음
 - <https://getbootstrap.com/docs/4.0/components/>
 - 웹 도구 제작에 자주 사용되는 구성요소들
 - Navbar: 주로 Header나 Footer에 들어가는 구성요소
 - Card, Heading, Breadcrumb: 본문에 들어가는 구성요소
 - Spinner: 로딩 아이콘 구성요소
 - Input, Label, Checkbox, Radio, Button: 입력 폼 구성요소
 - Pagination: 페이지 네비게이션
 - 기타 필요한 경우 커스텀 컴포넌트를 제작해서 써도 무방

34

웹 사이트 구조

- UI 컴포넌트 설정 예시



35

웹 디자인

- 전통적인 전문 디자인 도구 활용 (Photoshop, Illustrator 등)
 - 가격이 비쌌 (GIMP, Inkscape 등 자유 소프트웨어의 활용도 가능)
 - 상당한 트레이닝이 필요하나 일단 숙련되면 최고의 퀄리티를 낼 수 있음
- 온라인 기반 디자인 도구 활용 (Figma 등)
 - 개인에게 무료, 여러 사람이 동시에 작업 가능
 - 괜찮은 퀄리티, 현업에서도 많이 활용됨

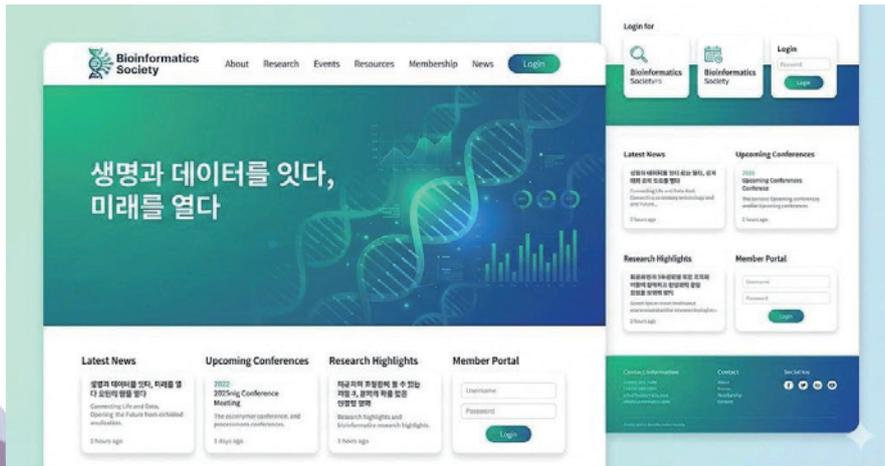
36

웹 디자인

• AI 기반 웹 디자인

- Google Gemini 등을 통해 나노바나나로 디자인 목업을 제작할 수 있음 (아래 예시 참고)

생명정보학회 홈페이지 디자인 목업을 그려줘. 색깔은 전체적으로 녹색과 파란색 계열을 섞어서 사용하고, 모던하고 세련되게 디자인해줘.



37

웹 디자인

• AI 기반 웹 디자인

- 앞서 배운 Header, Body, Footer 및 컴포넌트 개념을 활용하여 더 구체적으로 프롬프트를 작성할 수 있음 (여러 번 생성해보고 마음에 드는 것을 선택)

생명정보학회 홈페이지 디자인 목업을 그려줘. 색깔은 전체적으로 녹색과 파란색 계열을 섞어서 사용하고, 모던하고 세련되게 디자인해줘. 상단 NavBar가 있고, 좌측에는 로고, 가운데에는 "학회소개", "회원안내", "학술활동", "학술행사", "학회소식", "커뮤니티" 메뉴가 있고, 우측에는 로그인 버튼을 만들어줘. NavBar하단에는 "한국생명정보학회", "Korean Society for Bioinformatics" 문구가 들어간 Hero section을 만들어줘. 그 하단에는 왼쪽 Sidebar가 있고, 오른쪽 Body가 있으며 Body의 상단에 오른쪽 정렬된 Breadcrumb가 있고, 그 밑에 Heading이 나오며, 그 아래 본문이 등장하게 해줘. Footer에는 로고와 학회 정보, 주소, 연락처 등이 나오고, 우측에 개인정보 취급방침 링크를 돋보이게 해줘.



38

웹 사이트 배포

- 직접 웹 서버 운영 (self-hosting)
 - 서버 컴퓨터를 구입하여 웹 사이트를 직접 운영
 - 서버 컴퓨터 구축 및 서버 소프트웨어 운영 관련 지식이 필요
 - 일반적인 웹 사이트의 경우 큰 사양이 요구되지 않는 편
 - 심지어 라즈베리 파이로도 가능
 - 트래픽이 많아질 경우 전기세, 회선료를 직접 감당해야 함
 - 수백만명 이상의 사용자가 동시 접속할 경우
 - 일반적으로 생명정보학 알고리즘을 호스팅하는 경우 큰 문제가 되지 않음
 - 많은 경우 유저가 많지 않기 때문

39

웹 사이트 배포

- 외부 서비스 활용
 - 클라우드 서비스 (아마존 웹 서비스, 네이버/카카오 클라우드 등) 활용
 - 몇 번의 클릭만으로 하드웨어를 직접 구축한 것과 같은 효과
 - 사용량에 따른 이용료 지불 (종량제)
- 무료 호스팅 서비스 활용
 - GitHub: 백엔드가 없는 경우 무료 호스팅 지원

40

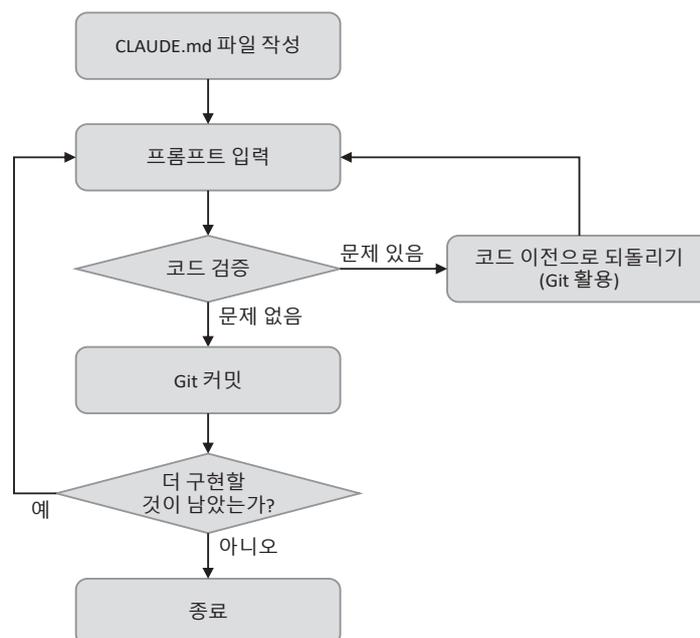
바이브코딩

- AI 에이전트 (Claude Code 등) 을 활용한 코드 작성
 - OpenAI: Codex CLI (GPT Codex 시리즈 기반)
 - Anthropic: Claude Code (Claude 시리즈 기반)
 - Google: Gemini CLI (Gemini 시리즈 기반)
- 본 강의에서는 주로 Anthropic 사의 Claude Code를 사용할 예정
 - Claude Code와 같은 코딩 에이전트는 프로젝트 폴더 안에 있는 'AGENTS.md' 를 참조하여 코딩을 수행하므로, AGENTS.md 파일에 프로젝트의 구체적인 설계를 적어두는 것이 유용함

41

바이브코딩

- AI 에이전트 (Claude Code 등) 을 활용한 코드 작성



42

Chapter 2. 웹 개발 실전

43

웹 도구 기획

- 목표
 - 주어진 FASTA 시퀀스의 Reverse Complement 시퀀스를 출력해주는 웹 도구 만들기
- 기획
 - 랜딩 페이지 없이 일반 페이지로만 구성 (Header/Body/Footer)
 - Header에는 로고 및 메뉴가 들어있는 Navbar로 구성
 - Body에는 시퀀스 입력 필드 및 결과창
 - Footer에는 제작자 정보
- 개발환경
 - Visual Studio Code를 통해 Claude Code 사용
 - React 프론트엔드 웹 프레임워크 활용
 - 개발환경은 리눅스/맥에서 진행
(윈도우 사용자의 경우, 스토어에서 'Ubuntu' 를 검색하여 설치한 후 WSL 환경에서 개발 진행)

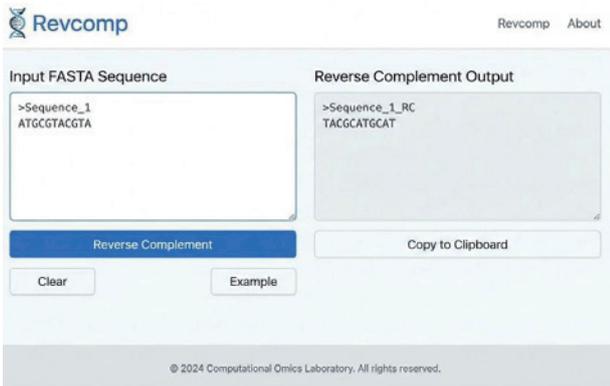
44

웹 도구 기획

- 디자인 생성하기

- Google Gemini에 간단한 명세를 적고 나노바나나를 통해 원하는 디자인을 생성하고, 마음에 드는 형태의 목업을 생성하여 그림 형태로 저장 (아래는 예시)

주어진 FASTA 시퀀스의 Reverse Complement 시퀀스를 출력해주는 웹 도구, Revcomp의 디자인 목업을 그려줘. 상단 Header, 가운데 Body, 하단 Footer로 구성해. Header에는 로고와 메인 메뉴 (Revcomp, About) 로 구성된 Navbar를 넣어. Body에는 시퀀스 입력 필드와 결과창을 동시에 넣어. Footer에는 제작자 정보 (Computational Omics Laboratory) 를 명시해.



Revcomp About

Input FASTA Sequence

>Sequence_1
ATCGGTACGTA

Reverse Complement Output

>Sequence_1_RC
TACCGATGAT

Reverse Complement

Copy to Clipboard

Clear Example

© 2024 Computational Omics Laboratory. All rights reserved.

45

개발환경 구축

- Node.js 설치

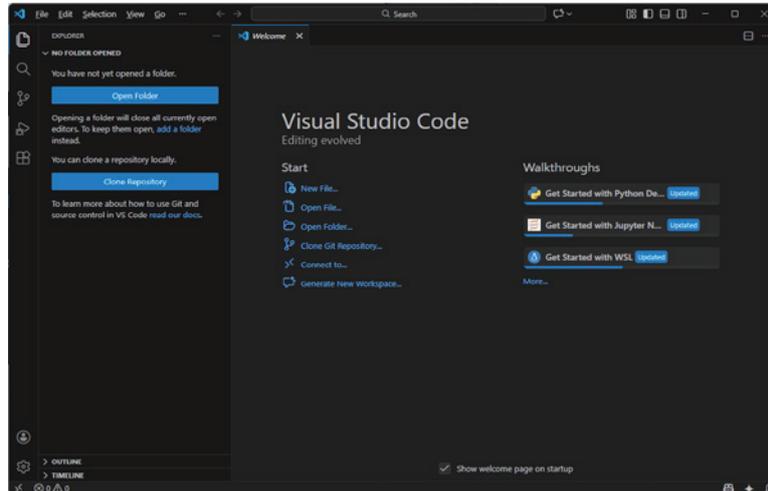
- 다음 웹사이트에 접속하여 Node.js 설치 (nvm, pnpm 선택 후 터미널에 입력)
 - <https://nodejs.org/en/download>

```
Download Node.js®
Get Node.js® v24.13.0 (LTS) for Linux using nvm with pnpm
Info Want new features sooner? Get the latest Node.js version instead and try the latest improvements!
1 # Download and install nvm:
2 curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.3/install.sh | bash
3
4 # in lieu of restarting the shell
5 \. "$HOME/.nvm/nvm.sh"
6
7 # Download and install Node.js:
8 nvm install 24
9
10 # Verify the Node.js version:
11 node -v # Should print "v24.13.0".
12
13 # Download and install pnpm:
14 corepack enable pnpm
15
16 # Verify pnpm version:
17 pnpm -v
Bash
```

46

개발환경 구축

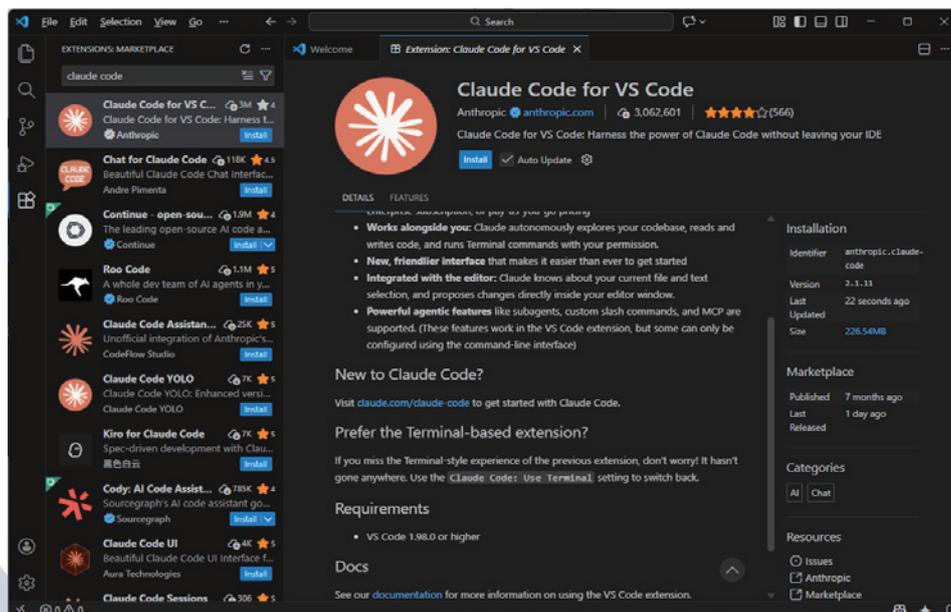
- Visual Studio Code 설치
 - 다음 웹사이트에 접속하여 Visual Studio Code 설치 및 실행
 - <https://code.visualstudio.com/>



47

개발환경 구축

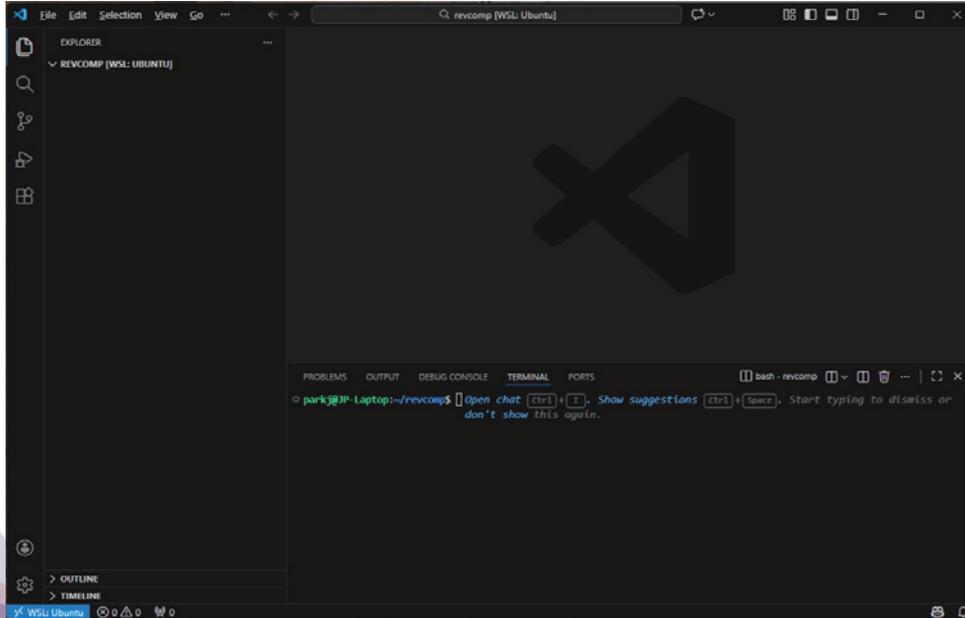
- Claude Code 설치



48

웹 개발 수행

- 프로젝트 폴더 생성 및 열기 (Open Folder) 후 터미널 실행



49

웹 개발 수행

- 터미널에서 다음 명령을 실행하여 React 앱 생성

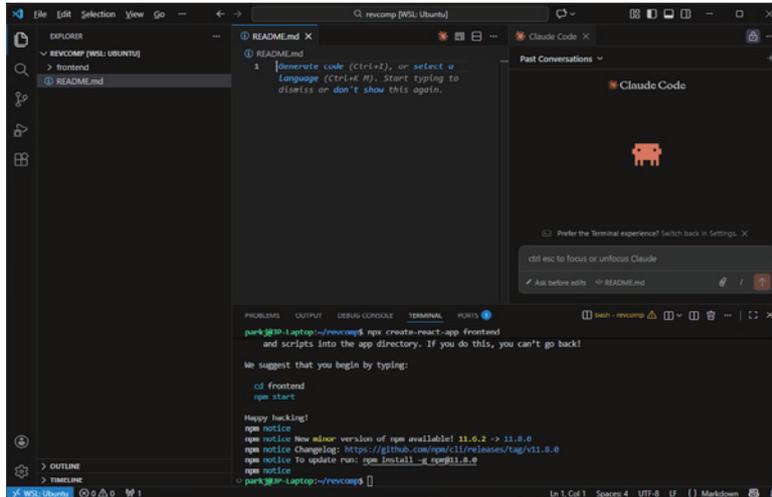
```
npx create-react-app frontend
```

- 현재 AI 에이전트가 초기 앱 생성을 잘 못 하는 경향이 있는데, 이는 AI가 이러한 명령을 활용하기보다는 기존의 코드를 직접 작성해 버리려는 성향에 기인함
- 문제는 기존 코드는 많은 경우 최신 버전이 아니며, 동작하지 않는 경우도 많음
- 따라서 프로젝트 생성 부분은 AI에 맡기기보다는 직접 해 주는 것이 더 나은 편

50

웹 개발 수행

- README.md 파일 생성
 - 좌측의 Explorer 창에서 README.md 파일을 새로 생성한 뒤, 우측 상단의 Claude 버튼을 눌러 Claude Code 실행 후 로그인

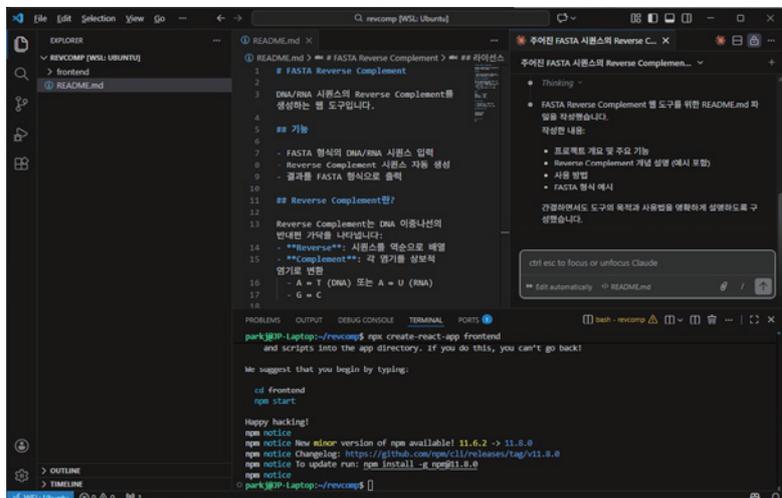


51

웹 개발 수행

- README.md 파일 생성
 - Claude Code 프롬프트 창에 revcomp 앱의 설명을 입력하고 README.md 파일 생성을 요청하고, 'Allow all edits in this session' 선택

주어진 FASTA 시퀀스의 Reverse Complement 시퀀스를 출력해주는 웹 도구를 제작하려 하는데, 여기에 대한 README.md 파일을 간략하게 작성해.



52

웹 개발 수행

- AGENTS.md 파일 생성
 - 코딩 세션이 길어지면 에이전트가 처음 내린 명령을 잊어버리는데, 이를 방지하기 위해 웹사이트의 명세를 정리해둔 파일인 AGENTS.md 파일을 작성
 - 마찬가지로 Claude Code의 프롬프트에 AGENTS.md 파일 작성을 요청하되, 앞서 생성한 frontend 폴더를 활용하고, 또한 디자인 목업 이미지를 복사하여 붙여넣은 뒤 디자인 명세도 같이 생성하라고 요청

이 도구를 개발하기 위한 AGENTS.md 를 작성해줘. 이미지를 참조하여 웹 사이트의 디자인 명세도 함께 상세하게 작성해. frontend 폴더에 미리 초기화해둔 react를 활용해서 개발할 예정이야.



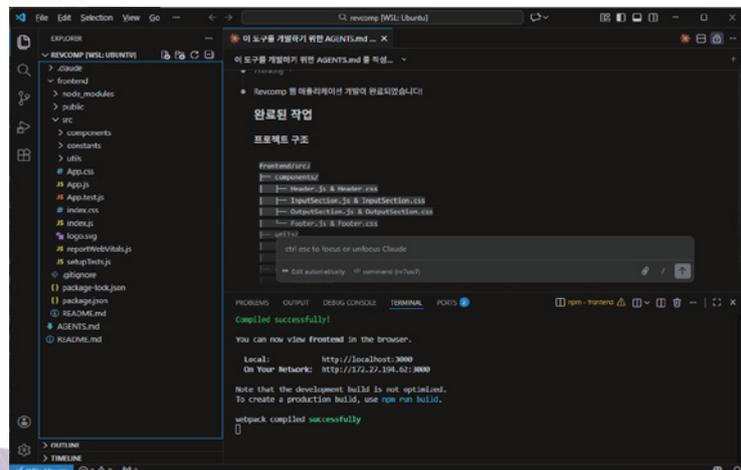
README.md

53

웹 개발 수행

- 개발 시작
 - 이제 Claude Code 에 '개발 시작' 명령을 내리면 개발이 수행됨
 - 개발이 완료된 후 터미널에서 frontend 디렉토리로 이동한 뒤, 다음 명령을 입력하여 개발 서버를 실행

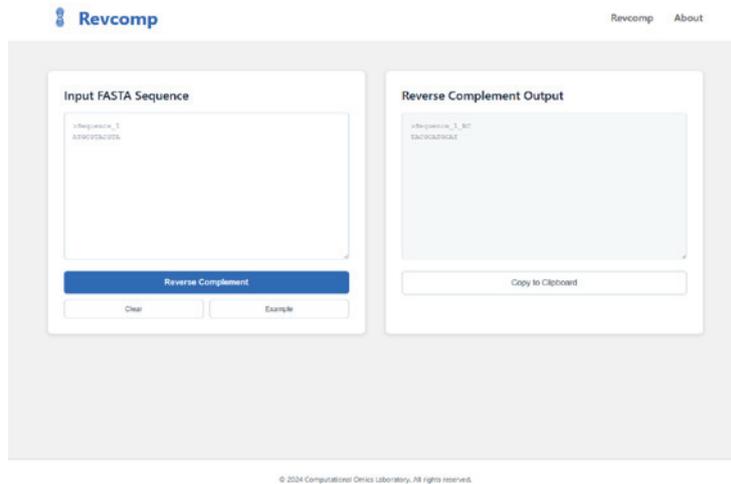
npm start



54

웹 개발 수행

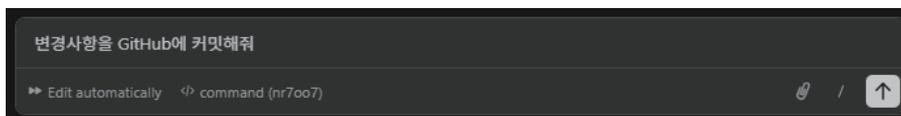
- 결과 확인
 - 웹 브라우저를 통해 실제 개발된 결과를 확인



55

웹 개발 수행

- Git 초기화 및 변경사항 저장
 - Claude Code로 돌아가 GitHub에 변경사항을 커밋하라고 요청



- 만약 Git 초기 설정 (사용자 이름 및 이메일) 이 되어 있지 않은 경우, 다음과 같이 입력하여 설정 (본인의 이름과 이메일을 사용할 것)

```
parkj@JP-Laptop:~/revcomp$ git config --global user.name "Jeongbin Park"
parkj@JP-Laptop:~/revcomp$ git config --global user.email "jeongbin.park@pusan.ac.kr"
parkj@JP-Laptop:~/revcomp$
```

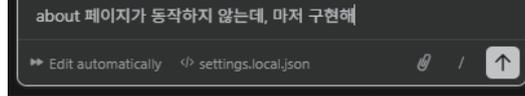
- 만약 GitHub에 Repository를 만들지 않았을 경우, GitHub 사이트에서 Repository 생성 후 커밋 시도

56

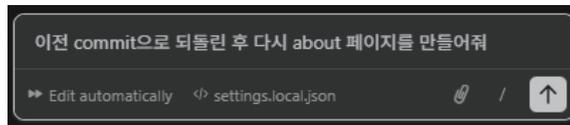
웹 개발 수행

- 추가 수정 및 개선

- 개선할 점이 있다면 Claude Code에 개선을 요청



- 만약 구현 중 문제가 생길 경우, 이전 Commit으로 되돌린 후 다시 수행하라고 명령하면 됨



57

결론

- AI의 발전으로 생명정보학 웹 도구 개발은 더 이상 그렇게 어렵지 않음

- 누구나 기본 원리와 용어만 체득하면 손쉽게 에이전트를 통해 원하는 웹 도구 제작 가능
- 더 복잡한 도구를 제작하는 경우에도 앞서 배운 이론들을 잘 활용하여 검색 및 수정이 가능할 것

- 소스 코드는 아래 GitHub 주소를 통해 확인 가능

- <https://github.com/pnuicolab/bioinfo2026-winter-workshop>

- 질의 응답 이메일

- jeongbin.park@pusan.ac.kr

58



감사합니다

59