

포트폴리오 작성가이드

멀티캠퍼스 취업지원센터

- 01 포트폴리오 작성가이드
- 02 참고자료

I 포트폴리오란

- 포트폴리오는 희망하는 직무를 수행하기 위해 준비한 내용(기술력 등)을 한 눈에 보여줄 수 있는 내용으로 자기소개서나 면접에 대비하기 위해 미리 정리해두는 것이다.

I 포트폴리오 작성

1.자가진단 및 직무 선택하기

: 자신의 역량과 흥미, 적성, 성향들을 파악하고 그에 적합한 직무를 선택한다.

2. 직무 분석하기

: 선택한 직무에 대한 정보를 수집하고, 그 직무에서 강조되고 있는 핵심 역량을 파악한다.
직무 분석 후, 자신이 갖춘 역량이 무엇인지 정리하고 부족한 역량을 보완할 계획을 세운다.

3. 경험과 활동을 연대기로 정리하기

: 현재까지 자신이 갖춘 경험과 활동을 구체적으로 정리한다.
직무의 핵심 역량과 자신이 갖춘 역량 간의 관계를 어필 할 수 있도록 서술하는 것이 중요하다.

4. 기업 분석으로 나만의 전략 세우기

: 내가 지원할 직무를 채용하는 기업들에 대해 알아보고 지원할 만한 기업들을 미리 선정한다.
기업들의 인재상도 미리 파악해두고, 지원동기 및 아이디어 제안도 구상해두는 것이 좋다.

5. 작성된 취업 포트폴리오 활용하기

: 작성된 취업 포트폴리오를 바탕으로 내가 보완해야 할 직무 역량에 대한 계획을 세울 수 있다.
또한 수 많은 취업 기회가 동시에 몰려드는 채용 시즌에 신속하게 취업 성공 확률이 높은
입사지원서를 작성할 수 있다.

I 포트폴리오 구성 Tip

No	항목	참고사항
1	표지	취업에 대한 목표 포트폴리오 소제목을 부제목으로 작성하여도 무방
2	목차	포트폴리오 구성 순서를 목차로 작성
3	목표	취업목표에 대해 작성하도록 하며, 희망직무와 기업을 선택한 이유를 구체적으로 작성 * 단, 포트폴리오 제출 목적에 따라 가감적용 필요
4	프로젝트	직무에 해당되는 프로젝트 개발 경험을 구체적으로 작성함 직무와 연계성이 높은 프로젝트를 순차적으로 한 눈에 쉽게 볼 수 있도록 작성
5	기타사항	프로젝트 개발 경험 외 공모전 참여 경력, 자격사항 등 직무와 관계 있는 이력사항을 기재하여 성장가능성을 명시 * 단, 포트폴리오 제출 목적에 따라 가감적용 필요
6	마무리	현재의 나의 상태와 기업이 요구하는 인재상에 대한 분석을 통해 향후 수행해야 하는 과업중심으로 결론을 제시 * 단, 포트폴리오 제출 목적에 따라 가감적용 필요

- 포트폴리오 양식은 **명시된 기준이 없기에 본인의 실력을 보여줄 수 있는 작품집으로 이해하면 쉽다.**
- 자기소개서, 이력서와는 차별된 무기로 면접 전 서류심사에서 경쟁력이 될 수 있으므로 지원분야와 관련된 본인의 보유기술 및 스킬, 프로젝트 경험, 대외활동, 공모전 경력 등을 사전에 준비하여 작성할 수 있도록 한다.
- 포트폴리오의 핵심은 **구체적 성과를 정리하여 단순히 나열하는 것이 아닌 시각적인 구성요소를 반영하여 나를 표현하는 것 중요하다.**
- 시각적인 구성 적용 시 텍스트에 적합한 이미지, 그래프, 인포그래픽을 활용하며 색상이 화려하기 보다는 포인트가 되는 컬러 또는 디자인을 적용하여 정돈된 느낌을 줄 수 있도록 한다.

I 프로젝트 구성 Tip

- 1. 프로젝트 진행 목적과 결과, 사용기술, 역할/기여분야 등을 구체적으로 작성
- 2. 항목별 근거에 따른 결과 중심의 프로젝트 이미지 등을 적절히 적용
- 3. 프로젝트 소개 필수 항목

- 프로젝트명, 프로젝트 기획 의도, 프로젝트 수행 기간, 프로젝트 사용기술, 프로젝트 참여 인원 (역할구분), 프로젝트 참여도/기여도, 프로젝트 결과 (URL 기재)

○ IT 포트폴리오 구성 사례 모음

구분	목차순서
사례1	표지 - 목차 - 주제 - 기획의도 및 기대효과 - 주요기능 - 사이트맵 - 주요 화면 설계 - 요구사항 명세서 - DB설계 (ERD) - DB설계 (테이블 명세서) - 개발환경 - 개발일정 - 팀원 별 역할(업무) - 주요화면 (메인화면, 마이페이지, 게시판, 게시글 작성 등) - 주요 소스코드 - 느낀점
사례2	표지 - 목차 - 필요성 - 차별성 - 프로젝트 기획 - 프로젝트 진행 현황 - 프로젝트 현황 별 상세 내용 - 프로젝트 결과 (도메인 주소) - 프로젝트 추후 계획 방향 - 프로젝트 참여자/역할 구분 - 사용기술 및 참고문헌 - 느낀점 - Q&A
사례3	프로젝트명 - 프로젝트 기간 - 프로젝트 구성인원 - 역할 기여도 - 프로젝트 내용 (주제) - 프로젝트 구현단계 - 담당부분 - 최종결과 (도메인주소) - 결과 시연 내용
사례4	표지 - 콘텐츠 - 개발 기획 의도 - 개발 선정 이유 및 필요성 - 주제선정 - 개발환경&사용언어 - 프로젝트 시연 - 프로젝트 시연(사용자기능) - 프로젝트 시연(URL) - 웹 구조도 - 웹 구조도(사이트맵) - 요구사항 명세서 - DB설계(테이블/ERD) - 주요기능 설명 - 주요기능 상세(팀원 업무/기능) - 주요기능 상세(주요화면) - 프로젝트 분석 - 프로젝트 개별 소감

I 포트폴리오 점검하기 (체크리스트)

(출처 - Pxd Story (<https://story.pxd.co.kr/1276>))

해당 글은 Pxd Story 에서 제공하는 비전공자 대상의 포트폴리오 작성을 위한 내용을 기반으로 체크리스트 내용을 제공합니다.

* Pxd Story – UX 관련 블로그 서비스

no	체크리스트	
1	나만의 브랜드가 포트폴리오에 잘 녹여져 있는가?	<input type="checkbox"/>
2	면접관의 입장에서 포트폴리오가 구성되어 있는가? (심플하지만 흥미로운 구성, 페이지의 강약조절, 보는 데 불편함이 없도록)	<input type="checkbox"/>
3	가장 중요한 순서(혹은 스토리 순)로 포트폴리오가 구성되어 있는가?	<input type="checkbox"/>
4	간결하고 핵심 정보가 잘 갖추어져 있는가?	<input type="checkbox"/>
5	매력적인 부분(개성)이 잘 드러나 있는가?	<input type="checkbox"/>
6	자신의 강점이 잘 드러나 있는가?	<input type="checkbox"/>
7	포트폴리오에 문제 제기 및 문제 해결방식이 잘 드러나 있는가?	<input type="checkbox"/>
8	디자인적으로 완성도가 있는 포트폴리오인가?	<input type="checkbox"/>
9	장식적인 그래픽 요소가 적고 정돈된 포트폴리오인가?	<input type="checkbox"/>
10	오타자, 맞춤법에 대한 고려가 잘 되어있는가?	<input type="checkbox"/>
11	프로젝트 기여도, 작업 기간 등 구성요소가 잘 들어가 있는가?	<input type="checkbox"/>
12	주변 실무자 등 인맥을 활용하여 포트폴리오의 피드백을 받았는가?	<input type="checkbox"/>
13	지원하려는 회사에 맞춤화된 포트폴리오를 구성하였는가?	<input type="checkbox"/>
14	완성된 포트폴리오 파일 용량이 너무 과하지는 않은가?	<input type="checkbox"/>

(출처 - <https://velog.io/@chris>)

해당 글은 개발 유튜브 채널 Traversy Media 의 영상 중 '웹 개발자 포트폴리오 웹사이트를 위한 10가지 팁'이라는 이라는 영상의 내용을 정리한 내용으로 velog에 등록된 번역본을 제공합니다.

* Velog - 개발자들을 위한 블로그 서비스

● 웹 개발자 포트폴리오를 위한 10가지 팁

1. Presentable Design/UI

- 디자인에 신경을 많이 써라.
- 포트폴리오에 당신이 개발한 웹사이트를 나열하게 될 텐데, 당연하게도 평가자가 가장 먼저 보게 되는 화면이며, 당신의 포트폴리오 자체도 당신의 프로젝트이기 때문에 가장 중요하다.
- 그 자리에서 바로 보여줄 수 있고 즉시 설명 가능한 디자인과 UI로 만들어라.
- 당신이 UI디자이너가 아니라 프론트엔드 개발자나 백엔드 개발자라면, 혹은 당신의 디자인 스킬이 끔찍하다면, 바닥부터 모두 디자인할 필요는 없다. 이미 잘 만들어진 템플릿들을 잘 활용하는 것도 능력이다.

2. Good Hosting & Domain

- @.herokuapp.com 같은 무료 서브도메인 쓰지 말아라. 비싸지 않고 어렵지 않으니 반드시 커스텀 도메인을 구매해서 연결해라.
- 반드시 https(SSL)를 지원해라. Chrome은 https를 지원하지 않으면 경고를 띄우기 때문에 UX가 좋지 않다.
- 기본적인 것들이라 이런 것들을 하지 않으면 평가자에게 꼼꼼하지 못하다거나 사소한 것을 신경 쓰지 않는다는 느낌을 줄 수도 있다.

3. Keep it simple

- 가장 중요한 몇 개의 프로젝트만 넣고 불필요한 것은 모두 빼고 최소화해라. 가능한 한 글도 최대한 짧게 써라.
- 프로젝트는 항상 가장 중요한 것들을 맨 위에 써라. 너무 길면 읽지 않는다. 너무 많아도 보기 귀찮고 어렵다. 자기소개만 10줄이면 아무도 읽지 않을 것이다. 최대한 짧고 심플하게 유지해라.
- 큰 회사들의 랜딩페이지를 참고해봐라. 10-15페이지의 구구절절한 페이지들보다 하나의 깔끔하고 간결한 싱글 랜딩페이지가 더 매력적이다.

4. Don't exaggerate skills

- 당신의 스킬셋을 과장하지 말아라. 거짓말 하지 말아라. 솔직해져라.
- 루비 온 레일즈 튜토리얼을 따라 해 봤다고 당신이 루비 개발자가 되지 않는다.
- 사용해본 모든 언어와 모든 프레임워크를 나열하지 말아라. 평가자는 멍청하지 않다.
- 모든 분야를 조금씩 두루 아는 것보다 몇 가지에 전문적인 것이 훨씬 좋다.

5. Live Examples/Demos

- 바로 라이브 프로젝트를 볼 수 있는 버튼을 만들어서 직접 들어가볼 수 있도록 해라.
- 해당 프로젝트를 깃허브에 올려놓고 코드도 즉시 볼 수 있도록 링크버튼을 만들어라.
- 화면이 없는 프로젝트(CLI툴, 라이브러리 등)는 예외다. 깃허브에 코드만 올려놓아도 된다. 대신 README는 잘 적자.
- 절대 프로젝트의 이미지만 덩그러니 놓지 말아라. 그런 건 아무나 할 수 있다.
- 이 경우 프로덕션 프로젝트가 아니라면, 커스텀 도메인이나 https지원 등은 하지 않아도 상관없다.

6. Be Selective of Projects

- 많은 사이트 프로젝트를 했더라도 To-do List같은 당신이 한 모든 프로젝트를 나열하지 말아라.
- 3~6개의 최고 프로젝트만 선택해서 넣어라. 2개의 퀄리티 있고 큰 프로젝트가 10개의 To-do List 보다 훨씬 우수하다.
- 즉, 양보단 질이다.

7. Customize Course Projects

- 강의나 튜토리얼을 따라해서 프로젝트를 했다면 절대 그대로 놓지 말아라. 반드시 자신 나름대로 커스터마이징 해라.
- 유명한 튜토리얼들은 이미 모든 사람들이 알고 있어서 평가자는 이미 똑같은 결과물을 많이 보았을 것이다.

8. Easy Contact

- 인사담당자가 당신에게 연락하기 쉽게 하라.
- 이메일, 핸드폰번호 같은 연락처를 Navigation bar, About page, Contact section 등 잘 보이는 곳에 써 놓아라.

9. Web Presence & Brand

- 자신을 브랜딩 할 수 있는 수단을 넣어라.
- 유튜브를 한다면 유튜브 채널, 글을 쓴다면 블로그 주소나 미디엄 주소, 깃허브 주소, SNS주소 등
- 본인 사진이든 로고든, 여러 다른 플랫폼에 비슷하거나 같은 프로필 이미지를 사용해라. 그게 싫다면 primary color라도 일관되게 하라.

10. Be Yourself

- 가장 똑똑한 사람이 되려고 하지 말아라. 똑똑해 보이려고 애쓰지 말아라.
- 그냥 솔직하게 자신을 표현해라.
- 당신의 열정, 취미, 특기 등 당신을 표현할 수 있는 다른 것들을 넣어도 좋지만, Keep it Simple!!

(출처 <https://velog.io/@siders>)

해당 글은 프론트엔드 개발자가 9개월간의 취업 준비 후 입사하여 면접관 경험을 기반으로 작성된 글로 velog에 등록된 '신입 개발자 이력서 잘 쓰는 방법'을 제공합니다.

* Velog – 개발자들을 위한 블로그 서비스

● 신입 개발자 이력서 잘 쓰는 방법

취준생활을 9개월 넘게 하다가 작년에 프론트엔드 신입으로 취업했습니다.

9개월 동안 면접을 보면서 이력서를 고치고 내용이 부족한 것 같아서 활동도 많이 했지만

이러한 노력에도 결과는 생각보다 좋지 않았고 그 이유 또한 잘 몰랐습니다.

하지만 인터뷰 면접관으로 들어가게 되면서 "왜 9개월 동안 잘 안 됐는지"를 깨달았습니다.

TL; DR

1. 최대한 1페이지 안에 되도록 꾸며서 작성하세요.
2. 이력서는 최대한 자세히, 그리고 근거와 함께 작성합니다.
3. 성장가능성, 성실성, 커뮤니케이션 능력, 문제해결능력을 중점에 두고 표현하도록 노력하세요.

사람은 겉모습으로 판단합니다



‘09년에 방영한 인간의 두 얼굴 중에 나오는 장면입니다.

동일한 인물이지만 옷차림에 따라 그 사람에 대한 평가가 확연하게 달라졌습니다. 이력서를 기반으로 지원자를 판단하기 때문에 같은 활동을 했더라도 표현방식에 따라서 결과가 달라질 수 있습니다.

무엇을 어떻게 써야 하는지를 설명하기 전에 먼저 이력서를 검토하는 개발자는 이력서를 어떻게 생각하는지를 알아야 합니다.

이력서를 어떻게 생각할까요?

검토하는 개발자는 이력서를 하루에 수십 개씩 읽습니다.

최근에 어떤 학원이 수료했다면 판박이와 같은 이력서가 쏟아집니다.

내용이 비슷하다면 누구를 뽑아야 할까요?

제일 처음에 본 이력서? 아니면 여러 기술을 사용해본 지원자?

거의 모든 이력서에는 본인이 성실하고 협업을 중요시 하며 동료에게 좋은 영향을 주고 싶은 개발자가 되고 싶다는 등등 그렇게 말합니다.

하지만 그 주장을 뒷받침할 근거가 부족하거나 없습니다. 그리고 지원자가 어떤 분이신지 자세하게 알고 싶지만 부족하거나 없는 경우도 있습니다.

이런 경우 딱히 궁금하거나 물어볼 만한 것이 없기 때문에 보통 그런 이력서가 많이 떨어집니다. 그리고 이력서가 단순 글자로만 이루어져 있다 보니 수십 개를 읽으면 집중력이 떨어지고 지치기가 쉽습니다.

그러기에 되도록 한 페이지분량에, 꾸민 이력서가 눈에 더 잘 들어 옵니다.

기왕이면 포트폴리오 사이트를 만들어서 보여 주는 게 더 좋습니다.

주의할 점

절대로 이력서를 쓰실 때 과장하거나 인터뷰 때 제대로 답변 못할 내용이 있으면 안됩니다.

인터뷰 때 이력서를 기반으로 질문을 하기 때문에 만약 대답을 제대로 못하거나

사실과 다르다고 느껴지면 거의 탈락했다고 보시면 됩니다.

어떤 이력서가 매력적일까요?

개발자 된 이유와 과정을 자세히 쓴 지원자

"미래가 어떤지 알고 싶다면 지나온 수레바퀴 길을 보아라" 라는 말이 있습니다.

미래는 지나간 과거의 연장선상에 있기 때문입니다.

신입개발자에게 지금 당장 뛰어들어서 업무 할 실력을 기대하지 않습니다.

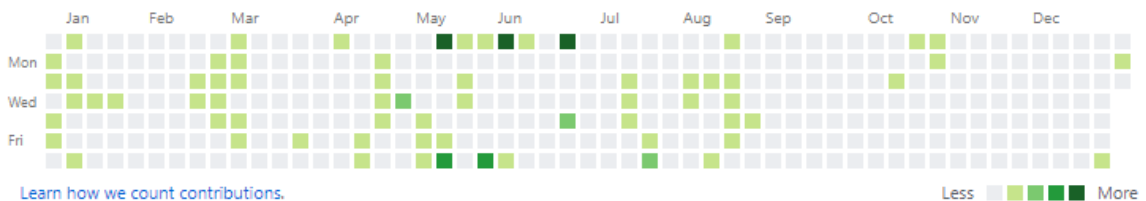
하지만 신입 지원자가 얼마나 성장할 수 있는지는 중요하게 여깁니다.

그러기에 본인이 어떻게 개발자가 되었으며 개발자가 되기 위해 밟았던 과정을 어떻게 성장했는지를
중점에 두고 자세히 서술 하는 게 좋습니다.

Github나 블로그를 꾸준히 운영한 지원자

성장 가능성을 보기 위해 얼마나 꾸준히 공부했는가를 보기도 합니다. 그러기에 블로그나 Github에 공부했던 내용을 꾸준히 작성한다면 성실하다는 인상을 줄 수 있으며 얼마큼 노력했는지 보여줄 수 있는 근거가 됩니다.

여러분이 면접관이라면 두 지원자 중에 어느 분을 만나겠습니까?



협업을 많이 경험해본 지원자

학원에서 했던 프로젝트도 좋지만 해커톤이나 사이드 프로젝트를 하는 IT동아리에서 해본 경험이면 더 좋습니다. 다른 직군 또는 같은 개발자 끼리 일하는데 나와 다른 생각을 가진 동료와 커뮤니케이션을 어떻게 하는지를 보기 위해서 입니다.

그렇기 때문에 협업을 어떻게 진행했으며 만약 갈등이 발생했다면 어떻게 해결했는지, 또한 그러한 과정 속에서 본인이 깨달았거나 고쳐야 할 점이 무엇이였는지를 잘 표현했으면 좋겠습니다.

프로젝트를 상용화 해본 경험이 있는 지원자

상용화 하는 게 쉽지 않지만 전체적인 개발프로세스를 이해할 수 있으며 실무에서 발생할 만한 이슈를 해결할 수 있다는 것을 입증할 수 있기 때문입니다.

실제 업무를 진행할 때 많은 어려운 이슈들이 존재합니다. 그러기에 지원자가 잘 해결할 수 있는지, 문제해결 능력이 있는지 알고 싶기에 이때까지 해온 프로젝트에 대해서 묻는 것입니다.

만약 상용화 해본 경험이 없더라도 해온 프로젝트에서 발생한 이슈에 대해 자세히 설명하고 어떻게 해결했으며, 해결하기 위해 어떤 기술을 썼으며, 왜 그 기술인지를 잘 쓴다면 이 보다 더 좋은 이력서가 없을 거 같습니다.

물론 이외에도 다음과 같은 내용이 있으면 더 좋습니다.

기술에 대한 관심이 많아서 적용해보거나 개발 프로세스를 개선해본 경험 또는 팀원들에게 도움 준 경험을 적으면 좋은 인상을 받을 수 있습니다.

저의 이러한 경험이 새로 시작하는 개발자분들에게 도움이 되기를 바랍니다.

THANK
YOU

www.multicampus.co.kr

서울특별시 강남구 테헤란로 212 2층

취업지원센터 이보라 프로

Tel. 02-3429-5192 E-mail br0809.lee@multicampus.com