

당근 지원 준비 발표 대본

대전대학교 컴퓨터공학과 박상돈 · 취창업역량강화

진행 표시는 읽지 않고 본문만 그대로 읽습니다. 1~18장은 지원자 역할의 모범 발표, 19~22장은 교수자의 과제 안내입니다. 뒤의 포트폴리오 설명은 PDF를 열어 두고 별도로 읽을 때 사용합니다.

수업 시작 안내

여러분, 이번 발표 과제는 실제 채용공고 하나를 골라서, 그 회사에 지원하려면 무엇을 준비해야 하는지 설명하는 것입니다. 오늘은 제가 당근의 신입 백엔드 개발자에 지원한다고 가정하고 예시를 보여드리겠습니다. 공고를 읽고 계획을 세우는 데서 끝내지 않고, 실제로 실행되는 작은 프로젝트와 제출할 포트폴리오까지 준비했습니다. 지금부터 지원자 입장에서 발표하겠습니다. 뒤에서 보여드리는 구현과 측정 결과는 이 수업 예제를 위해 만든 자료입니다. 여러분은 같은 틀에 자신의 경험과 결과를 넣으면 됩니다.

01 당근 백엔드 개발자 지원 준비

안녕하세요. 대전대학교 컴퓨터공학과 박상돈입니다. 저는 당근의 신입 백엔드 개발자 공고를 골랐습니다. 오늘은 이 공고에서 어떤 역량을 요구하는지, 그 요구를 제 프로젝트로 어떻게 보여줄지 말씀드리겠습니다.

핵심 프로젝트는 교내 중고거래 관심 피드입니다. 클릭 기록을 받아 피드에 반영하고, 중복 입력을 처리했습니다. 테스트 결과와 조회 성능을 측정한 기록을 묶어서 네 쪽의 포트폴리오로 정리했습니다.

02 제가 고른 공고

제가 선택한 공고의 정확한 이름은 소프트웨어 엔지니어 백엔드, 피드 ML 데이터 플랫폼 신입입니다. 당근의 공식 채용 사이트에서 신입 정규직 공고라는 점을 확인했습니다.

화면은 2026년 9월 6일에 직접 확인한 공고입니다. 본문에는 마감일이 적혀 있지 않았고, 지원서 페이지도 열렸습니다. 실제 지원 시점에는 공고가 계속 유효한지 다시 확인하겠습니다. 이 직무를 고른 이유는 전공 수업에서 배운 내용을 작은 서비스로 구현해 보여줄 수 있다고 판단했기 때문입니다.

03 이 팀이 해결하는 문제

이 팀은 추천에 필요한 데이터를 만들고 저장해서, 요청이 왔을 때 빠르게 제공하는 일을 합니다. 사용자가 무엇을 봤는지 기록이 빠지거나 늦게 반영되면, 추천에 쓰는 정보도 달라질 수 있습니다.

저는 이 일을 학부 수준에서 다룰 수 있는 범위로 줄였습니다. 사용자의 클릭을 저장하고, 관심 카테고리를 계산해 게시글의 순서를 바꾸는 기능입니다. 회사의 큰 시스템 전체를 훑내 내기보다는, 데이터의 정확성과 조회 속도를 직접 확인할 수 있는 부분을 구현했습니다.

04 필수 조건과 우대 경험

공고를 읽을 때는 필수 조건과 우대 경험을 구분했습니다. 필수 조건에서는 컴퓨터공학 기초와 하나 이상의 언어로 문제를 해결하는 능력을 확인했습니다. 우대 경험에서는 프로젝트를 끝까지 완성한 경험과 측정으로 확인한 개선을 찾았습니다.

그래서 준비 방향을 정했습니다. 기술 이름을 많이 나열하기보다, 제가 고른 방법이 왜 필요한지 설명하고 코드와 테스트 결과를 붙이겠습니다. 오른쪽 문장은 공고를 그대로 옮긴 것이 아니라, 공고를 읽고 제가 정한 준비 방향입니다.

05 공고와 결과물을 연결하기

이 표는 공고의 요구와 이번 예제의 결과물을 연결한 것입니다. 전공 기초는 인덱스와 트랜잭션을 선택한 이유로 설명합니다. 언어 활용 능력은 실제 파이썬 코드와 입력 검증으로 보여줍니다.

완성 경험은 실행 가능한 프로젝트와 README로, 문제 해결 경험은 열두 개의 테스트로 보여주겠습니다. 측정 가능한 개선에는 같은 SQL을 인덱스 추가 전후에 비교한 기록을 사용합니다. 이렇게 요구사항 옆에 확인할 파일을 붙이면, 포트폴리오의 각 내용이 왜 필요한지 분명해집니다.

06 현재 결과와 남은 준비

이번 예제에서는 API와 화면을 구현했고, 테스트와 SQL 성능 실험까지 마쳤습니다. 여기까지는 실제 파일로 확인할 수 있는 결과입니다.

다만 실제 학생들이 쓰는 서비스로 운영해 본 것은 아닙니다. 로그인과 권한 관리도 없고, 여러 사용자가 동시에 접속하는 HTTP 부하 실험도 하지 않았습니다. 그래서 현재 완성한 부분과 앞으로 준비할 부분을 나눠 적었습니다. 작은 예제에서 확인한 결과를 곧바로 대규모 서비스 운영 경험이라고 설명하지 않겠습니다.

07 기술 글에서 가져온 준비 방향

공고에 연결된 기술 글도 읽었습니다. 이 글은 당근 개발자들이 함께 작성한 자료로, 추천에 쓰는 데이터를 저장하고 빠르게 조회하는 구조를 설명합니다. 캐시와 데이터 갱신 문제도 다룹니다.

제가 가져온 질문은 단순합니다. 지금 느린 부분이 어디인지, 어떤 데이터를 얼마나 자주 읽는지를 먼저 보자는 것입니다. 이번 예제에서는 캐시를 추가하기 전에 SQL의 실행 계획과 인덱스를 확인했습니다. 회사의 구조를 그대로 복사한 것이 아니라, 문제를 확인하는 방법을 제 예제에 적용했습니다.

08 제출할 프로젝트

프로젝트의 이름은 교내 중고거래 관심 피드입니다. 전공책을 확인한 학생에게 책 관련 게시글을 먼저 보여주는 상황을 가정했습니다. 실제 학생들의 불편을 조사한 결과는 아니고, 기능을 설명하기 위해 정한 사용 상황입니다.

합성 게시글 여섯 개를 준비했고, 클릭한 카테고리의 횟수를 기준으로 피드 순서를 바꿉니다. 클릭 저장과 피드 조회, 중복 입력 처리까지 구현했습니다. 결제나 채팅은 범위에 넣지 않았습니다. 이번 프로젝트에서 보여주려는 역량은 추천 모델의 정확도가 아니라 백엔드의 데이터 처리입니다.

09 구현 범위와 기술 선택

구현에는 파이썬과 FastAPI, SQLite를 사용했습니다. FastAPI는 요청 형식을 정의하고 API 문서를 제공하는 데 사용했습니다. SQLite는 학생이 별도 DB 서버 없이 예제를 실행할 수 있도록 선택했습니다.

이벤트 저장과 사용자 요약 갱신은 하나의 트랜잭션으로 묶었습니다. 조회에는 사용자, 카테고리, 발생 시각 순서의 복합 인덱스를 추가했습니다. 원래 계획에 있던 PostgreSQL이나 Redis, 별도 작업 프로세스는 이번 완성본에 넣지 않았습니다. 실제로 구현한 기술과 이후에 검토할 기술을 구분했습니다.

10 정확성 실험

같은 클릭이 두 번 도착하는 상황을 보겠습니다. 첫 번째와 두 번째 입력은 이벤트 ID가 모두 e01입니다. 서버는 첫 요청을 저장하고, 같은 내용의 재전송에는 중복이라는 응답을 돌려줍니다. 그래서 전공책 클릭은 한 번만 집계됩니다.

세 번째 입력은 나중에 도착했지만 실제 발생 시각은 9시 59분입니다. 처음 전공책을 클릭한 10시보다 이전이므로, 최신 관심은 전공책으로 유지돼야 합니다. 이 동작을 테스트와 화면에서 확인했습니다. 중간 실패의 롤백, DB 재접속, 동시 재전송을 포함해 열두 개의 테스트가 통과했습니다.

11 조회 성능 측정 결과

성능은 합성 이벤트 십만 건과 사용자 천 명을 기준으로 측정했습니다. 같은 SQL에 같은 사용자 목록을 넣고, 복합 인덱스가 있을 때와 없을 때를 비교했습니다. 각 조건에서 삼백 번씩 세 회차를 실행했습니다.

합친 측정값의 피구십오는 인덱스가 없을 때 약 5.304밀리초, 있을 때 약 0.087밀리초였습니다. 피구십오는 측정한 조회의 95퍼센트가 그 시간 안에 끝났다는 뜻입니다. SQL 실행과 결과 읽기만 측정했기 때문에 이 수치를 웹서비스 전체 응답 시간이라고 부르지는 않겠습니다.

실행 계획도 전체 스캔에서 인덱스를 이용한 조회로 바뀌었습니다. 세 회차의 개별 결과와 원본 기록을 함께 남겼고, 인덱스가 있어도 조회 결과 자체는 같다는 점을 확인했습니다.

12 제출용 포트폴리오 네 쪽

지금까지 설명한 내용을 실제 제출용 포트폴리오 네 쪽으로 묶었습니다. 첫 쪽에는 지원 직무, 프로젝트 목적, 실행 화면과 담당 범위를 적었습니다. 둘째 쪽에는 중복 입력과 실패 상황을 어떻게 처리했는지 정리했습니다.

셋째 쪽에는 인덱스 전후의 측정값과 실험 조건을 넣었습니다. 넷째 쪽에는 파일 위치, 실행 방법, 남은 한계와 공고의 요구사항을 연결했습니다. 읽는 사람이 제 설명을 믿는 데서 끝나지 않고, 어떤 파일을 열어 확인하면 되는지 알 수 있도록 구성했습니다.

13 직접 남긴 화면과 파일

포트폴리오에 들어간 화면은 실제로 실행한 예제를 캡처한 것입니다. 클릭을 세 번 보냈을 때 서버의 응답과 저장된 이벤트 수가 어떻게 바뀌는지 확인할 수 있습니다.

화면만으로는 모든 동작을 검증하기 어려워서 테스트 코드도 함께 넣었습니다. 테스트 결과는 verification 파일에서, 성능 실험의 모든 조회 시간은 latency CSV 파일에서 확인할 수 있습니다. README에는 설치와 실행, 테스트, 측정 명령을 적었습니다. 아직 만들지 않은 GitHub 주소나 공개 배포 주소는 넣지 않았습니다.

14 예제 다음의 8주 준비 일정

작은 예제가 완성됐다고 해서 지원 준비가 끝난 것은 아닙니다. 이후 여덟 주는 실제 서비스에 가까운 경험을 보완하는 데 쓰겠습니다. 먼저 사용자 상황을 확인하고, 인증과 권한을 설계한 뒤 PostgreSQL로 옮기는 실험을 하겠습니다.

그다음 배포와 HTTP 부하 측정을 진행하고, 확인한 병목 하나를 개선하겠습니다. 마지막으로 실제 결과로 이력서와 포트폴리오를 정리하고 코딩 면접을 연습하겠습니다. 각 주차에는 공부할 주제와 함께 남길 산출물을 적었습니다. 이 일정은 제 준비 계획이며 채용 일정에 맞춰 조정할 수 있습니다.

15 면접 준비

공고에는 화상 인터뷰의 라이브 코딩 테스트가 안내돼 있습니다. 사용할 수 있는 언어도 확인했고, 이번 준비에서는 파이썬을 선택했습니다. 문제를 푸는 것뿐 아니라 접근 방법과 시간 복잡도를 말하면서 설명하는 연습을 하겠습니다.

프로젝트 질문도 준비하겠습니다. 왜 SQLite를 선택했는지, 같은 이벤트가 다른 내용으로 다시 오면 어떻게 하는지, 인덱스의 단점은 무엇인지 설명할 수 있어야 합니다. 이 질문들은 실제 기출문제가 아니라 제 프로젝트를 기준으로 만든 연습 질문입니다.

16 지원 동기 답변

당근에서 관심 있는 부분은 사용자 행동을 추천에 반영하는 과정입니다. 이를 이해하기 위해 교내 중고거래 관심 피드 예제를 구현했습니다. 같은 클릭이 다시 도착하는 경우와 늦게 도착하는 기록을 처리하고, SQL 조회 성능을 같은 조건에서 비교했습니다.

이 과정에서 기능을 구현하는 것과 그 기능이 정확하게 동작함을 설명하는 데에는 차이가 있다는 점을 확인했습니다. 코드와 테스트, 측정 조건을 함께 설명할 수 있도록 준비하고 있습니다. 실제 서비스 운영과 부하 대응은 앞으로 보완할 부분입니다.

17 제출 화면과 파일 구성

지원서 화면도 확인했습니다. 이력서는 PDF 형식을 권장하고 있고, 포트폴리오는 최대 50메가 바이트까지 올릴 수 있습니다. 웹 링크를 제출할 때는 PDF나 텍스트 파일에 주소를 적어 첨부하도록 안내돼 있습니다.

제가 준비한 포트폴리오는 네 쪽짜리 PDF입니다. 코드와 실행 방법은 별도 프로젝트 묶음에 정리했습니다. 실제 지원할 때는 이력서의 개인 이력과 프로젝트 설명을 본인 정보로 맞추고, 필요한 자료를 검토자가 접근할 수 있는 방식으로 제공할 것입니다. 네 쪽이라는 분량은 제가 정한 구성이며 회사의 지정 양식은 아닙니다.

18 이번 준비에서 남긴 결과

이번 준비에서는 공고를 분석하고, 그 요구와 연결되는 작은 백엔드를 구현했습니다. 실행 화면과 열두 개의 테스트, 인덱스 전후의 측정값을 남겼고, 이를 제출용 포트폴리오 네 쪽으로 정리했습니다.

현재 확인한 것은 로컬 예제의 정확성과 SQL 조회 성능입니다. 실제 사용자 운영과 HTTP 부하 대응은 다음 준비 과제로 남겨두었습니다. 지원 준비를 설명할 때는 제가 한 일과 앞으로 할 일을 이처럼 구분하겠습니다. 이상으로 발표를 마치겠습니다.

19 학생 발표 안내

여기까지가 제가 보여드린 모범 발표입니다. 이제 여러분의 과제를 안내하겠습니다. 각자 실제 채용공고 하나를 고르고, 왜 그 공고를 선택했는지 설명해 주세요. 공고 화면과 주소, 확인한 날짜를 함께 남기면 됩니다.

핵심 요구를 세 개 이상 골라 자신의 경험이나 준비할 결과물과 연결해 주세요. 이미 만든 프로젝트가 있으면 실제 자료를 사용하고, 아직 없다면 구현 범위와 검증 방법을 구체적으로 설명하면 됩니다. 발표는 여덟 장에서 열 장 정도로, 발표 칠 분과 질문 삼 분을 기준으로 준비해 봅시다.

20 평가 기준 설명

평가에서는 공고를 정확히 읽었는지와 자신을 근거 있게 설명했는지를 가장 많이 보겠습니다. 공고 분석과 자기 진단에 각각 이십오 점, 포트폴리오 설계와 준비 계획에 각각 이십 점, 발표와 질의응답에 십 점을 배정하는 기준을 예로 들었습니다.

제가 여러분에게 묻고 싶은 질문은 이것입니다. 그 역량이 있다는 것을 어떤 결과물로 확인할 수 있나요? 문장이 화려한지보다 코드나 결과, 실행 방법이 설명과 맞는지 보겠습니다. 최종 제출 일정과 평가 방식은 수업 공지에 맞춰 안내하겠습니다.

21 예상 질문 설명

질문을 준비할 때는 자기 발표에서 출발하면 됩니다. 예를 들어 이 공고를 고른 이유, 아직 사용하지 않은 기술을 언제 배울지, 개선 수치를 어떤 조건에서 측정했는지를 스스로 물어보세요.

답을 모를 때는 모르는 범위와 확인할 방법을 말하면 됩니다. 특히 팀 프로젝트에서는 자신이 맡은 기능과 동료가 맡은 기능을 구분해 주세요. 이번 예제의 코드를 그대로 자신의 개인 경험이라고 제출하는 것이 과제의 목적은 아닙니다. 여러분이 직접 설명하고 확인할 수 있는 경험을 같은 방식으로 정리하는 것이 목표입니다.

22 출처와 마무리

마지막 장에는 실제 채용공고와 지원서, 참고한 기술 글의 링크를 적었습니다. 기술 구현에서는 SQLite의 공식 문서도 참고했고, 해당 링크는 포트폴리오에 넣었습니다.

오늘 자료에서 공고와 화면, 테스트 결과와 SQL 측정값은 직접 확인한 자료입니다. 실제 학생의 사용자 수요나 회사 시스템 규모를 검증한 것은 아닙니다. 여러분도 사실과 해석, 계획을 구분해 주세요. 공고 한 줄을 자신의 준비와 연결하고, 그 준비를 확인할 수 있는 결과물로 보여주면 됩니다.

포트폴리오를 보여주며 읽는 대본

포트폴리오 PDF를 한 쪽씩 열어 두고 다음 본문을 읽습니다.

1쪽 프로젝트 개요와 실행 화면

첫 쪽은 이 프로젝트가 무엇인지 빠르게 파악하도록 구성했습니다. 지원 직무는 당근의 신입 백엔드 개발자이고, 프로젝트는 교내 중고거래 관심 피드입니다. 클릭 기록으로 관심 카테고리 를 계산하고 게시글의 순서를 바꿉니다.

화면 오른쪽을 보면 클릭을 세 번 보냈지만 고유 이벤트는 두 건입니다. 같은 ID의 클릭을 다시 보냈기 때문입니다. 늦게 들어온 운동용품 기록도 저장하되, 최신 관심은 전공책으로 유지했습니다. 아래에는 실제로 구현한 범위와 사용 기술을 적었습니다.

2쪽 설계와 실패 조건

둘째 쪽은 어떤 문제를 어떻게 처리했는지 설명하는 페이지입니다. 같은 이벤트가 다시 오면 본문까지 비교합니다. 내용이 같으면 중복 응답을 주고, 같은 ID에 다른 내용이 들어오면 충돌로 거부합니다.

이벤트 저장과 사용자 요약은 하나의 트랜잭션으로 묶었습니다. 두 작업 사이에 일부러 오류를 발생시킨 테스트에서는 두 변경이 함께 취소됐고, 다시 보냈을 때 한 번만 반영됐습니다. 이처럼 정상 화면뿐 아니라 실패 조건을 재현한 결과까지 보여주도록 구성했습니다.

포트폴리오 설명 (계속)

3쪽 성능 결과와 조건

셋째 쪽은 성능 결과입니다. 합성 이벤트 십만 건을 넣고, 같은 조회에 인덱스만 추가해 비교했습니다. 각 조건에서 삼백 번씩 세 회차를 측정했고, 합친 결과의 피구십오는 약 5.304밀리초에서 0.087밀리초로 줄었습니다.

다만 이 값은 단일 프로세스의 SQL 실행과 결과 읽기 시간입니다. 네트워크나 HTTP 처리, 동시 접속은 포함하지 않았습니다. 회차별 값도 모두 적었고, 원본 CSV를 첨부했습니다. 인덱스의 추가 공간은 약 3.20메비바이트였으며 쓰기 성능에 미치는 영향은 아직 측정하지 않았습니다.

4쪽 재현 방법과 남은 한계

넷째 쪽에는 검토자가 결과를 확인할 방법을 적었습니다. 설치 후 서버를 실행하는 명령, 테스트 명령, 성능 실험 명령을 순서대로 제시했습니다. 테스트 결과와 원본 조회 시간 파일의 위치도 안내했습니다.

현재 한계도 적었습니다. 합성 데이터로 만든 로컬 예제이며 인증과 실거래 기능이 없습니다. HTTP 부하와 실제 사용자 반응은 검증하지 않았습니다. 이 페이지는 부족한 점을 숨기지 않으면서, 지금까지 확인한 범위를 분명하게 보여주기 위해 넣었습니다.