

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 01 / 대전대 컴공에서 취업을 준비하는 방법

오늘은 여러분이 컴퓨터공학과에서 무엇을 공부하고, 그 공부를 어떻게 취업 준비로 연결할지 설명하겠습니다. 채용공고를 하나 읽고, 그 공고에서 요구하는 능력을 우리 학과 과목과 연결해 보겠습니다. 이어서 프로젝트를 어느 수준까지 완성해야 하는지, AI를 사용하면서도 본인의 실력을 어떻게 쌓을지 살펴보겠습니다. 마지막으로 여러분이 이번 학기에 실행할 준비 계획과 발표 과제를 정리하겠습니다. 함께 제공하는 제 실제 포트폴리오는 경험을 설명하는 방식의 참고 자료로 사용합니다.

슬라이드 02 / 오늘 정할 네 가지

취업 준비를 시작하라고 하면 무엇부터 해야 할지 막막할 수 있습니다. 오늘은 네 가지를 정하면 됩니다. 먼저 어떤 일을 하는 개발자가 되고 싶은지 공고 하나로 구체화합니다. 다음으로 그 일을 준비하는 데 필요한 전공 과목 두 개를 고릅니다. 그리고 이번 학기에 끝낼 프로젝트 기능 하나를 정합니다. 마지막으로 완성했다고 말할 수 있는 검사 방법과 증거를 정합니다. 아직 직무가 확실하지 않아도 괜찮습니다. 이번 학기에는 하나를 골라 직접 해 보고, 그 경험으로 다음 선택을 하면 됩니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 03 / 개발 직무마다 만드는 결과가 다릅니다

컴공을 졸업한다고 모두 같은 일을 하는 것은 아닙니다. 백엔드는 화면 뒤에서 요청을 처리하고 데이터를 저장합니다. 프론트엔드는 사용자가 직접 보는 화면과 조작 흐름을 만듭니다. AI와 데이터 분야에서는 데이터를 처리하고 결과의 품질을 확인합니다. 게임이나 시스템 분야에서는 규칙과 성능, 실행 환경의 제약을 다룹니다. 표의 구분은 진로를 탐색하기 위한 예시이며 실제 회사에서는 역할이 겹치기도 합니다. 오늘은 백엔드 공고를 중심으로 설명하되, 다른 분야에서도 요구사항을 경험으로 증명하는 방법은 같습니다.

슬라이드 04 / 실제 공고: 당근 피드 백엔드 신입

오늘의 사례는 당근 피드 팀의 백엔드 신입 공고입니다. 이 직무는 추천 화면에 필요한 데이터를 저장하고 제공하는 시스템을 개발합니다. 예를 들어 어떤 글을 봤는지, 추천에 쓸 정보가 잘 만들어졌는지, 요청에 빠르게 응답하는지를 다룹니다. 공고에 머신러닝이라는 말이 있지만 모델 학습만 하는 직무로 보면 안 됩니다. 공고는 머신러닝 전문성을 필수로 요구하지 않는다고 설명하고, 소프트웨어공학 역량을 핵심으로 제시합니다. 따라서 지원 준비의 중심은 전공 기초와 서비스를 끝까지 만드는 경험입니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 05 / 공고에서 정한 공부의 우선순위

공고의 모든 기술 이름을 동시에 공부하려고 하면 시작하기 어렵습니다. 이 공고에서 먼저 확인할 것은 전공 기초, 한 가지 이상의 언어 숙련, 그리고 끝까지 완성한 개발 경험입니다. 예를 들어 파이썬을 주력 언어로 정했다면 기능을 만드는 것뿐 아니라 오류를 읽고 테스트를 작성할 수 있어야 합니다. 이후 프로젝트에서 메시지 처리나 배포가 필요해졌을 때 해당 기술을 추가합니다. 이 순서는 공고를 읽고 제안하는 학습 우선순위입니다. 특정 도구 몇 개를 배웠다는 이유만으로 지원 요건을 모두 충족했다고 볼 수는 없습니다.

슬라이드 06 / 백엔드가 하는 일: 클릭 한 번의 처리

화면에서 전공책을 클릭했다고 생각해 봅시다. 화면은 서버에게 어떤 글을 선택했는지 알려 줍니다. 서버는 요청이 올바른지 확인하고 데이터베이스에 기록합니다. 그 기록을 읽어 관심 분야가 반영된 목록을 만들고 화면으로 돌려줍니다. 백엔드 개발자는 이 가운데 요청 확인, 저장, 조회, 응답을 책임집니다. 응답이 늦어서 같은 요청이 다시 들어오거나, 저장 도중 문제가 생길 때도 결과가 잘못되지 않도록 만들어야 합니다. 화면에 보이는 프로그램은 이 과정을 설명하기 위해 만든 실제 수업용 실행 예제입니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 07 / 백엔드 프로젝트의 최소 준비 범위

백엔드를 준비하려면 구체적으로 무엇을 만들어야 할까요. 처음에는 물품을 등록하고 조회하는 작은 서비스면 충분합니다. API는 화면이나 다른 프로그램이 서버에 요청하는 약속입니다. 어떤 주소에 어떤 정보를 보내면 어떤 응답이 오는지 정합니다. 데이터베이스에는 사용자와 게시글을 저장합니다. 로그인한 사용자만 자기 글을 수정하도록 권한을 확인하고, 잘못된 입력에는 오류를 돌려줍니다. 마지막에는 다른 사람이 실행할 수 있도록 배포나 실행 절차를 준비합니다. 이 표는 앞으로 완성할 학습 목표이며, 현재 클릭 예제가 이 모든 기능을 갖춘 것은 아닙니다.

슬라이드 08 / 대전대 컴공의 공개 교과과정

화면은 대전대학교 컴퓨터공학과 공식 교과과정 페이지입니다. 이 표에서 자료구조, 알고리즘, 운영체제, 데이터베이스, 네트워크와 같은 과목을 확인할 수 있습니다. 앞으로 설명하는 학년과 학기는 이 공개 표의 편성 기준입니다. 이 페이지에 적용 교육과정 연도가 명확하게 표시되어 있는 않으므로, 실제 수강신청 때는 본인 학번에 적용되는 교육과정과 해당 학기 개설 과목을 다시 확인해야 합니다. 여기서는 수강신청을 대신 정하는 것이 아니라, 각 과목을 공부해야 하는 이유와 남길 결과물을 연결하겠습니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 09 / 1·2학년: 코드를 이해하고 고치는 기초

일 학년과 이 학년에서는 코드를 직접 읽고 고칠 수 있는 기초를 만드는 것이 중요합니다. C프로그래밍에서는 입력부터 결과까지 프로그램이 어떻게 움직이는지 설명해 보세요. 자료구조에서는 해시나 트리를 외우는 데서 끝내지 말고, 같은 문제에 서로 다른 구조를 사용해 보세요. 알고리즘에서는 입력이 커졌을 때 시간이 왜 늘어나는지 설명해야 합니다. C++, 자바, 파이썬을 모두 같은 깊이로 공부하려고 하기보다 수업을 통해 비교하고 주력 언어를 정하세요. 언어 수보다 스스로 오류를 찾고 구현을 설명하는 능력이 먼저입니다.

슬라이드 10 / 3학년 이후: 서버가 믿을 만하게 동작하도록

삼 학년 이후에는 서비스가 실패하는 상황을 다루게 됩니다. 운영체제와 네트워크를 배우면 여러 요청이 함께 들어오거나 통신이 끊기는 상황을 이해하는 데 도움이 됩니다. 데이터베이스에서는 저장을 한 덩어리로 처리하는 트랜잭션과 조회를 빠르게 하는 인덱스를 프로젝트에 적용해 보세요. 자바 서버 프로그래밍과 리눅스에서는 요청을 처리하고 서버를 실행하며 로그로 문제를 찾습니다. 소프트웨어공학과 캡스톤에서는 요구사항을 정리하고 팀과 검토하는 경험을 남깁니다. 과목별 과제를 하나의 프로젝트로 이어 가면 준비가 누적됩니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 11 / 관심 분야에 따라 더할 과목

관심 분야가 정해지면 관련 과목을 더합니다. AI와 데이터에 관심이 있다면 파이썬, 데이터 마이닝, 기계학습 과목을 연결하고 평가 자료를 남기세요. 게임 분야에서는 그래픽스와 게임 프로그래밍을 활용해 실행 가능한 게임을 만들고 저장이나 성능 문제를 설명할 수 있어야 합니다. 화면 개발에 관심이 있다면 전공 과목과 함께 HTML, CSS, 자바스크립트를 별도로 학습해 실제 화면을 만들어 보세요. 인프라와 보안 분야에서는 리눅스, 네트워크, 클라우드 지식을 연결합니다. 이 표는 모두 들어야 할 목록이 아니라 진로에 따른 선택 예시입니다.

슬라이드 12 / 2026년 2학기 수업을 연결하는 방법

이번 학기 제 공개 강의 자료를 예로 들어 보겠습니다. 알고리즘에서는 복잡도, 분할정복, 탐욕법, 동적계획법 등을 배우고 방법을 선택한 이유를 설명합니다. 생성형 AI 소프트웨어공학에서는 문제 정의, 요구사항, 설계, 검증과 책임을 다룹니다. 캡스톤 프로젝트가 있다면 여기서 배운 내용을 같은 프로젝트에 적용하세요. 예를 들어 검색 기능에서 자료구조를 선택하고, AI에게 맡길 작업의 조건을 적고, 완성된 기능을 테스트하는 방식입니다. 이것은 학습을 연결하는 예시이며 각 과목의 제출 규정과 평가 기준은 해당 강의 안내를 따릅니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 13 / DB 수업 과제를 포트폴리오 경험으로

DB 수업에서 테이블을 만들었다면 그 다음 단계로 이어 가 보세요. 먼저 사용자와 게시글의 관계를 그림이나 표로 정리합니다. 다음으로 어떤 조회가 느린지 같은 데이터와 같은 요청으로 시간을 측정합니다. 인덱스를 추가했다면 실행 계획과 결과가 어떻게 달라졌는지 확인합니다. 그리고 빨라진 만큼 저장 공간이나 쓰기 비용이 늘어나는지도 살펴봅니다. 마지막에는 왜 이 방법을 선택했는지 한 문단으로 적습니다. 이것이 수강한 과목을 실제 역량으로 바꾸는 방법입니다. 측정값은 직접 실행해 얻고, 아직 하지 않은 검사는 계획으로 남기세요.

슬라이드 14 / 1·2학년의 준비 계획

저학년이라면 거창한 서비스를 목표로 잡기보다 작은 프로그램을 끝내는 습관을 먼저 만드세요. 일 학년에는 수업 언어로 기능을 구현하고, 핵심 함수의 흐름을 스스로 설명할 수 있으면 좋습니다. 이 학년에는 자료구조와 알고리즘을 공부하면서 주력 언어를 정하고 작은 공동 프로젝트를 해 보세요. 방학에는 새 프로젝트를 계속 추가하기보다 이전 프로젝트의 오류와 문서를 정리해도 좋습니다. 친구가 내 설명서만 보고 실행할 수 있는지 확인해 보세요. 학년별 계획은 예시이므로 현재 수준에 맞춰 앞 단계부터 시작하면 됩니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 15 / 3·4학년의 준비 계획

고학년에는 하나의 프로젝트를 완성하면서 실제 지원 준비도 병행해야 합니다. 삼 학년에는 서버, 데이터베이스, 네트워크에서 배운 내용을 한 프로젝트에 적용합니다. 방학에는 현장실습이나 인턴 공고를 확인하고, 기회가 맞지 않으면 팀 프로젝트에서 리뷰와 협업 경험을 쌓을 수 있습니다. 사 학년 캡스톤에서는 시연하고 피드백을 반영해 보세요. 지원할 때는 이 경험을 공고에 맞게 골라 설명합니다. 프로젝트가 완벽해질 때까지 공고를 보지 않는 것이 아니라, 공고를 읽으면서 부족한 경험을 보완하는 방식으로 진행합니다.

슬라이드 16 / AI를 써도 사람이 설명해야 하는 것

AI를 사용하면 코드 초안을 빨리 만들 수 있습니다. 하지만 만들어진 코드가 우리 문제를 제대로 해결하는지까지 자동으로 보장되지는 않습니다. GitHub의 공식 안내도 생성한 코드를 사람이 검토하고 테스트해야 한다고 설명합니다. 그래서 여러분은 네 가지를 설명할 수 있어야 합니다. 무엇을 해결하려 했는지, 왜 이 구조를 선택했는지, 어떤 조건으로 맞는지 확인했는지, 그리고 남아 있는 한계가 무엇인지입니다. AI를 사용한 사실보다 이 판단을 직접 할 수 있는지가 학습에서 중요합니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 17 / AI에게 요청하기 전에 요구사항부터 씁니다

AI에게 게시판을 만들어 달라고만 하면 우리가 필요한 조건이 빠질 수 있습니다. 먼저 사용자가 원하는 동작을 쓰고, 변경 범위와 완료 조건을 적어 보세요. 화면의 예시는 같은 클릭이 다시 전송되어도 한 번만 기록하는 기능입니다. 같은 요청 번호와 같은 내용이면 기존 결과를 돌려주고, 같은 번호인데 내용이 다르면 충돌을 알려야 합니다. AI에게는 먼저 현재 코드를 읽고 변경 계획을 설명하도록 요청할 수 있습니다. 생성된 테스트가 구현과 함께 같은 실수를 반복할 수 있으므로, 기대 결과는 사람이 먼저 정해 두는 것이 좋습니다.

슬라이드 18 / AI가 만든 코드를 확인하는 순서

AI로 기능을 만들 때는 네 단계를 반복하세요. 먼저 현재 프로그램을 실행해서 기준 상태를 확인합니다. 다음으로 AI가 바꾼 파일을 읽고, 필요하지 않은 변경이나 새 의존성이 들어갔는지 확인합니다. 그다음 정상 입력뿐 아니라 빈 입력, 중복 요청, 권한이 없는 요청처럼 실패해야 하는 경우를 검사합니다. 마지막으로 핵심 로직을 자기 말로 설명하고 반영합니다. 설명이 막히는 부분은 추가 학습이 필요한 부분입니다. AI가 테스트를 통과했다고 말한 것과 내가 실행 결과를 확인한 것은 구분해서 기록하세요.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 19 / AI 시대의 개발 공부 루틴

AI 시대에도 혼자 생각하는 시간이 필요합니다. 한 번의 학습 시간을 칠십오 분으로 잡는다면, 처음 이십 분은 AI 없이 문제와 기존 코드를 읽어 보세요. 다음 삼십 분은 막힌 부분의 설명이나 대안을 AI에게 요청하며 구현합니다. 이어서 십오 분 동안 직접 실행하고 실패하는 입력을 넣어 봅니다. 마지막 십 분은 화면을 덮고 핵심 동작과 선택 이유를 설명합니다. 이 시간 배분은 예시이므로 조정해도 됩니다. 중요한 것은 결과를 복사하는 데서 끝내지 않고, 다음에는 스스로 할 수 있는 부분을 늘리는 것입니다.

슬라이드 20 / AI 기능을 넣는 프로젝트의 준비

AI를 개발 도구로 사용하는 것과 제품 안에 AI 기능을 넣는 것은 다릅니다. 학과 안내 문서에서 답을 찾는 서비스를 만든다고 해 보겠습니다. 먼저 공개 가능한 문서를 정하고 기준 날짜를 기록합니다. 그다음 평가 질문 스무 개 정도를 직접 만들고, 검색만 했을 때와 AI 답변을 비교합니다. 문서에 없는 질문에는 어떻게 대응하는지, 답변의 출처가 실제로 맞는지도 확인해야 합니다. 응답 시간과 비용을 측정하고 비밀키는 코드에 넣지 않습니다. 표의 스무 개는 평가를 시작하는 예시이며, 지금 이 서비스의 정확도를 검증했다는 뜻은 아닙니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 21 / 팀에서 필요한 역량도 결과물로 남깁니다

취업 준비에는 혼자 코드를 작성하는 능력만 필요한 것이 아닙니다. 팀원이 내 변경 내용을 이해할 수 있도록 작은 단위로 커밋하고 변경 이유를 적어 보세요. 질문할 때는 안 된다는 말만 하기보다 재현 방법, 예상한 결과, 실제 결과를 함께 전달합니다. 기술 문서를 읽을 때는 예제를 복사하기 전에 사용 조건과 제한을 확인합니다. 영어 문서는 AI 번역을 활용해도 좋지만 함수 이름과 조건은 원문에서 다시 확인하세요. 이런 기록은 협업 역량을 설명할 때 실제 근거가 됩니다.

슬라이드 22 / 실제 프로젝트 사례: Floppy Survivors

제 실제 프로젝트 가운데 플로피 서바이버즈를 예로 들겠습니다. 이 프로젝트에서는 작은 실행 파일 용량이라는 제약을 두고 게임을 개발했습니다. 확인된 빌드의 실행 파일은 백만 육천오백구십이 바이트였고, 기준은 백사십칠만 사천오백육십 바이트였습니다. 빌드 과정에는 크기를 넘으면 실패하도록 하는 검사가 있습니다. 이 수치는 실행 파일만의 크기이며 운영체제나 닷넷 실행 환경까지 포함한 크기는 아닙니다. 학생 여러분은 같은 규모를 따라 할 필요가 없습니다. 자기 프로젝트에도 지켜야 할 기준 하나를 정하고 자동으로 검사하는 경험을 만들면 됩니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 23 / 수업 예제: 같은 클릭이 두 번 도착하면

이번에는 학생이 따라 볼 수 있는 수업 예제입니다. 사용자가 클릭한 뒤 응답을 받지 못하면 같은 요청을 다시 보낼 수 있습니다. 두 요청을 모두 새 클릭으로 저장하면 관심도 기록이 부풀어 오릅니다. 예제는 요청 번호를 함께 저장해서 같은 번호와 같은 내용이 다시 오면 기존 결과를 돌려주도록 했습니다. 내용이 달라진 충돌은 별도로 거절합니다. 실제 검사에서는 같은 요청을 열여섯 번 보냈을 때 새 저장 한 번과 중복 응답 열다섯 번을 확인했습니다. 이처럼 문제, 처리 규칙, 검사 결과가 연결되어야 무엇을 준비했는지 전달됩니다.

슬라이드 24 / 수업 예제: DB 과목을 조회 개선에 적용

이 예제에서는 데이터베이스 과목에서 배우는 인덱스도 적용했습니다. 같은 사용자의 기록을 찾을 때 전체 기록을 훑는 조회와 인덱스를 이용하는 조회를 비교했습니다. 피구십오는 측정값을 정렬했을 때 구십오 퍼센트 지점의 시간입니다. 이 조건에서 오 점 삼공사삼 밀리초가 영 점 공 팔육육 밀리초로 줄었습니다. 대신 인덱스 공간이 약 삼 점 이 메비바이트 늘었습니다. 합성 데이터와 준비된 로컬 환경에서 SQL을 측정한 결과이므로 실제 사용자 서비스 전체가 같은 비율로 빨라졌다고 말하면 안 됩니다. 결과와 측정 범위를 함께 설명해야 합니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 25 / 실습 예제를 지원용 경험으로 더 발전시키려면

지금 예제에서 한 일과 앞으로 해야 할 일을 분명히 구분하겠습니다. 예제는 클릭 저장, 중복 처리, SQL 조회 실험을 수행했습니다. 지원용 프로젝트로 발전시키려면 실제 사용자 구분과 권한 검사가 필요합니다. 배포 후에는 요청 전체의 지연과 오류도 확인해야 합니다. 다른 환경에서 실행해 보고, 데이터를 어떻게 사용할지 정한 뒤 피드백을 받아 볼 수 있습니다. 이런 항목을 이미 했다고 적는 대신 남은 준비 과제로 적으세요. 현재 상태를 정확히 설명하면서 다음 행동을 구체적으로 제시하는 것도 중요한 능력입니다.

슬라이드 26 / 포트폴리오: 실제 경험을 공고에 맞게 배치

포트폴리오는 나라는 지원자가 무엇을 해 왔는지 보여 주는 문서입니다. 먼저 실제 경험을 모은 기본 포트폴리오를 만드세요. 그다음 지원 공고와 가까운 프로젝트를 먼저 배치하고, 해당 역량을 보여 주는 설명을 보강합니다. 회사마다 경력을 새로 만들거나 프로젝트의 사실을 바꾸는 것이 아닙니다. 함께 제공한 제 포트폴리오는 실제 경력과 연구, 게임, 검색 서비스 사례를 정리한 자료입니다. 여러분은 논문이나 큰 프로젝트가 없어도 됩니다. 수업이나 팀 프로젝트에서 직접 맡았던 부분과 해결한 문제를 같은 방식으로 설명하면 됩니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 27 / 프로젝트 설명은 이 순서로 씁니다

프로젝트를 설명할 때는 어떤 서비스를 만들었는지 한 문장으로 시작하세요. 그다음 팀 구성과 내가 맡은 부분을 밝힙니다. 이어서 해결한 문제 하나를 고르고, 왜 그 방법을 선택했는지 설명합니다. 마지막에는 어떤 조건으로 확인했는지와 남은 한계를 적습니다. 예를 들어 클릭 기록에서 중복 요청 때문에 집계가 늘어나는 문제를 해결했다면, 요청 번호로 중복을 구분한 이유와 동시에 같은 요청을 보낸 검사 결과를 보여 줄 수 있습니다. 기술 이름은 이 설명을 이해하는 데 필요한 만큼 넣으면 됩니다.

슬라이드 28 / 코딩 테스트와 라이브 코딩 준비

코딩 테스트와 라이브 코딩은 프로젝트 준비와 함께 진행하세요. 이 당근 공고에서는 C++, Go, 자바, 파이썬을 라이브 코딩 언어로 안내합니다. 이 가운데 익숙한 언어를 정해 연습할 수 있습니다. 정답 코드를 외우기보다 입력 조건을 확인하고 자료구조를 선택하는 과정을 말해 보세요. 빈 입력이나 중복 값처럼 놓치기 쉬운 조건도 검사합니다. 연습할 때는 실제 평가 상황을 가정해 AI 없이 풀고 설명하는 시간을 갖는 것이 좋습니다. 실제 전형에서 허용되는 도구와 방식은 회사의 최신 안내를 따릅니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 29 / 전공 면접은 내 프로젝트로 설명해 봅니다

전공 면접에서는 개념을 외우는 것과 프로젝트 경험을 연결해 보세요. 자료구조를 물으면 정의를 말한 뒤 내 프로젝트에서 왜 그 구조를 골랐는지 설명합니다. 네트워크에서는 응답이 끊겼을 때 다시 보낸 요청을 어떻게 처리했는지 이야기할 수 있습니다. DB에서는 인덱스의 장점과 비용, 저장 도중 실패했을 때의 처리를 설명합니다. 화면의 질문은 연습 예시이며 실제 회사의 출제 문제를 뜻하지 않습니다. 모르는 질문을 받으면 아는 범위와 확인이 필요한 부분을 구분해서 말하는 연습도 해 보세요.

슬라이드 30 / 이번 학기 12주 실행 계획

이번 학기에 적용할 수 있는 십이 주 계획입니다. 처음 삼 주는 공고를 분석하고 필요한 전공 개념을 복습하면서 프로젝트의 범위를 정합니다. 사 주부터 육 주까지는 핵심 기능과 실패 테스트를 만듭니다. 칠 주부터 구 주까지는 다른 사람이 실행해 보게 하고 피드백을 받아 개선합니다. 마지막 삼 주는 경험을 포트폴리오로 정리하고 발표와 면접을 연습합니다. 처음부터 기능을 많이 넣기보다 각 기간의 제출물을 완성하는 데 집중하세요. 일정이 짧다면 기능 수를 줄이고, 검사와 설명은 남기는 것이 좋습니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 31 / 주당 6시간을 확보한다면

수업 외에 일주일에 여섯 시간을 낼 수 있다면 이렇게 나누어 보겠습니다. 두 시간은 전공 복습과 코딩 문제에 사용합니다. 두 시간은 프로젝트의 작은 기능 하나를 구현합니다. 한 시간은 테스트와 오류 수정에 사용하고, 나머지 한 시간은 기록과 동료 피드백에 사용합니다. 시험 기간에는 시간을 줄여도 되지만 무엇을 했는지 기록을 남기세요. 자격증이나 어학 준비가 공고의 필수요건이라면 별도 시간을 배정해야 합니다. 목적 없이 준비 항목만 늘리기보다 선택한 직무의 조건과 현재 부족한 부분을 기준으로 순서를 정하세요.

슬라이드 32 / 교내 지원을 이용할 때 가져갈 자료

학교의 취업 지원도 활용해 보세요. 대전대학교 취창업진로지원팀 홈페이지에는 상담과 프로그램 신청, 컨설팅, 채용 정보 등의 안내가 있습니다. 상담을 받을 때는 무엇을 해야 할지 모르겠다는 상태에서 한 단계 더 준비해 가면 좋습니다. 관심 공고, 현재 포트폴리오 초안, 부족하다고 생각하는 부분을 정리해 가세요. 전공 프로젝트는 담당 교수나 팀원에게 코드와 재현 방법을 보여주며 피드백을 받을 수 있습니다. 현장실습이나 비교과 프로그램은 공지에서 신청 기간과 참여조건을 확인하고 본인 일정에 맞게 선택하세요.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 33 / 학생 발표 과제: 준비한 일을 보여 주세요

여러분의 발표는 칠 분, 질의응답은 삼 분으로 운영하는 예시를 제안합니다. 첫 일 분에는 공고와 선택 이유를 설명합니다. 다음 일 분에는 관련 전공 과목 두 개와 준비한 내용을 말합니다. 중심이 되는 삼 분에는 본인이 맡은 프로젝트와 문제 해결 사례를 보여 주세요. 마지막 이 분에는 AI를 사용했다면 어디에 사용하고 무엇을 직접 확인했는지, 앞으로 무엇을 준비할지 설명합니다. 프로젝트가 아직 없다면 현재 상태를 밝히고 작은 기능을 직접 만들어 증거를 추가하세요. 계획을 수행 경험으로 적지는 않습니다.

슬라이드 34 / 평가 기준: 경험과 설명이 연결되는가

평가에서는 프로젝트의 규모보다 경험과 설명이 연결되는지를 보겠습니다. 공고가 요구하는 능력을 전공 과목과 연결했는지, 본인이 맡은 코드를 설명할 수 있는지 확인합니다. 문제 해결 사례에서는 결과뿐 아니라 방법을 선택한 이유와 검사 조건을 봅니다. AI 활용 항목은 도구를 많이 사용했는지가 아니라 사용 범위와 직접 검증한 내용을 설명하는지를 봅니다. AI를 사용하지 않았다면 사용하지 않은 이유와 본인의 검증 과정을 설명하면 됩니다. 마지막으로 질문에 답하고 다음에 할 일을 구체적으로 정했는지 확인합니다.

그대로 읽는 발표 대본

진행: 제목은 읽지 않고 본문만 읽습니다. 슬라이드 번호를 보고 넘겨 주세요.

슬라이드 35 / 이번 주 준비 계획서 작성

마지막으로 이번 주 계획을 작성하겠습니다. 먼저 실제 공고 하나를 저장하고 선택 이유를 적으세요. 그다음 필요한 역량 세 개와 그중 부족한 부분을 고릅니다. 부족한 부분에 연결되는 전공 과목 두 개를 적고, 이번 주에 할 행동 하나를 정하세요. 예를 들어 데이터베이스 과목의 인덱스를 복습하고, 내 프로젝트의 조회 하나를 같은 조건으로 측정하는 식입니다. 완료를 확인할 화면이나 테스트 결과도 미리 정합니다. 다음 발표에서는 공부하겠다는 말에서 한 걸음 더 나아가, 실제로 해 본 결과와 배운 점을 들려주면 됩니다.

슬라이드 36 / 자료 출처와 사용 범위

자료의 출처를 정리했습니다. 직무와 전형은 당근 공식 공고를, 과목명과 편성은 대전대 컴공 공개 교과과정표를 참고했습니다. 이번 학기 알고리즘과 생성형 AI 소프트웨어공학의 내용은 제 공개 강의 자료를 연결했습니다. AI 코드 검토의 필요성은 GitHub 공식 안내를 참고했습니다. 과목별 결과물과 학년별 계획, 주당 학습 시간은 이 수업을 위한 제안입니다. 채용공고와 개설 과목, 프로그램 일정은 달라질 수 있으므로 실제 지원이나 수강신청 때 다시 확인하세요.

실제 포트폴리오 설명 대본

진행: 포트폴리오_박상돈.pdf를 해당 쪽에 열고 본문을 읽습니다.

포트폴리오 1쪽 / 소개와 경력

먼저 제 포트폴리오를 예로 보여드리겠습니다. 저는 대전대학교 컴퓨터공학과에서 AI와 게임 시스템을 연구하고 가르치는 박상돈입니다. KAIST에서 수학과 전기전자공학을 공부했고, 연구소와 게임 개발 회사를 거쳐 현재 대학에서 일하고 있습니다. 첫 페이지에는 제가 어떤 사람인지, 어떤 일을 해 왔는지, 어디에서 결과물을 확인할 수 있는지를 적었습니다. 여러분은 이 자리에 전공과 관심 직무, 실제로 해 본 활동, 연락처를 넣으면 됩니다. 경력이 길어야 하는 것은 아닙니다. 읽는 사람이 지원자의 방향을 이해할 수 있으면 됩니다.

포트폴리오 2쪽 / 기술과 프로젝트 연결

두 번째 페이지에서는 기술 이름을 경험과 연결했습니다. Python을 쓴다고만 적으면 어느 정도로 사용할 수 있는지 알기 어렵습니다. 그래서 요청을 만들고, 서버의 대기 상태를 계산하고, 여러 방법을 같은 조건에서 비교한 연구 코드를 함께 적었습니다. C#과 Go도 실제 게임에서 구현한 기능과 연결했습니다. 아래 다섯 사례는 제가 해 온 작업 중에서 설명할 근거가 있는 것들을 골랐습니다. 회사에 지원할 때에는 실제 경험은 그대로 두고, 그 직무와 관련된 사례를 먼저 배치하면 됩니다. 웹 서비스 직무라면 검색 서비스 사례를, 게임 개발 직무라면 게임 시스템 사례를 앞에 놓는 식입니다.

실제 포트폴리오 설명 대본

진행: 포트폴리오_박상돈.pdf를 해당 쪽에 열고 본문을 읽습니다.

포트폴리오 3쪽 / 연구 프로젝트

이 연구는 질문을 어느 컴퓨터에서 처리할지 결정하는 작업입니다. 가까운 컴퓨터는 이동 시간이 짧지만 모델이 작을 수 있고, 클라우드의 큰 모델은 답변 품질이 좋아도 대기과 통신 시간이 들 수 있습니다. 저는 현재 대기 상태와 전송 속도를 반영해서 선택하는 정책과 비교 실험을 구현했습니다. 표에서는 동일한 요청 사천 건을 여러 방법에 넣었습니다. 제안 방법의 종합 목적값은 팔 점 삼오이고, 고정된 대기 시간을 사용하는 방법보다 이십일 점 삼 퍼센트 낮았습니다. 이 수치는 속도 개선율이 아닙니다. 지연과 품질 부족, 비용을 합친 연구 지표의 차이입니다. 또 다섯 단계 앞을 탐색하는 방법은 더 좋은 값을 보였습니다. 이런 비교 조건과 한계를 함께 적어야 결과를 정확하게 설명할 수 있습니다. 이 연구 원고는 현재 심사 중입니다.

포트폴리오 4쪽 / Floppy Survivors

Floppy Survivors는 화면에 보이는 서바이버 게임입니다. 여기서 설명할 개발 문제는 게임을 작은 실행 파일에 넣는 것입니다. 음악 파일과 그래픽 자료를 계속 추가하면 크기가 커집니다. 이 프로젝트에서는 소리의 파형과 음표 데이터를 이용해서 음악과 효과음을 실행 중 만들어 냅니다. 저는 전투와 성장 기능에 더해 이 오디오 처리와 배포 빌드를 작업했습니다. 파일을 작게 만들었다고만 말하지 않고, 실제 실행 파일의 크기와 상한을 표로 넣었습니다. 확인한 파일은 백만 육천오백구십이 바이트이고, 빌드 스크립트는 플로피 용량 상한을 넘으면 실패합니다. 다만 운영체제와 닷넷 실행 환경까지 이 크기에 들어가는 것은 아닙니다. 이 사례는 제약 조건을 정하고, 구현과 빌드 검사에 그 조건을 반영한 경험을 보여 줍니다.

실제 포트폴리오 설명 대본

진행: 포트폴리오_박상돈.pdf를 해당 쪽에 열고 본문을 읽습니다.

포트폴리오 5쪽 / Beasts of Ephesus

이 게임에서는 감독이 선수를 영입하고 키운 뒤 경기를 관전하고 전술에 개입합니다. 저는 선수와 리그, 전투를 연결하는 시스템을 개발했습니다. 이 페이지에서는 제품을 나눠 배포할 때 생기는 문제를 한 가지 보여 줍니다. 전체판과 체험판을 별도 프로젝트로 복사해서 관리하면 수정 사항이 서로 달라질 수 있습니다. 그래서 같은 코드에서 만들되, 체험판의 진행 제한은 컴파일할 때 결정하도록 했습니다. 저장 폴더도 나눠서 서로의 진행 기록이 섞이지 않게 했습니다. 전체판 설정에 체험판 제한이 섞이면 검사에서 거부하도록 했습니다. 게임 화면만 보여 주는 것보다 이런 설계 문제와 해결 방법을 함께 설명하면, 제가 어떤 부분을 개발했는지가 분명해집니다.

포트폴리오 6쪽 / Hexagon Soup

Hexagon Soup에서는 던전 지도가 실행할 때 만들어지고, 캐릭터가 그 안을 탐색하며 싸웁니다. 저는 Python에서 구현했던 게임 동작을 Go로 옮기고, 원래의 기준 데이터와 비교하는 검사를 작업했습니다. 이 페이지의 사례는 초반 던전의 탐색 구조를 조정한 결과입니다. 같은 조건을 다시 만들기 위해 난수 시드를 고정했습니다. 궁수 캐릭터는 수정 전후 모두 열두 번의 실행에서 목표에 도달했고, 총 이동 수는 이만 천구십에서 만 삼천오백십일로 줄었습니다. 화염 마법사는 열두 번 중 열 번 도달하던 것이 열두 번 모두 도달했습니다. 이것은 자동 플레이의 결과입니다. 실제 사람이 삼십오 점 구 퍼센트 더 재미있게 느낀다는 뜻은 아닙니다. 무엇을 바꿨고 어떤 조건으로 확인했는지까지 적는 것이 결과 설명의 핵심입니다.

실제 포트폴리오 설명 대본

진행: 포트폴리오_박상돈.pdf를 해당 쪽에 열고 본문을 읽습니다.

포트폴리오 7쪽 / 건축 규정 RAG 서비스

이번에는 웹 서비스 사례입니다. 사용자가 질문을 쓰는 화면 뒤에서 어떤 일이 일어나는지 보겠습니다. 서버는 질문을 받으면 관련 조문을 찾습니다. 문장 뜻이 비슷한 자료를 찾는 방법과 검색어가 직접 일치하는 자료를 찾는 방법을 함께 사용합니다. 그 결과를 합쳐 상위 여덟 개 근거를 모델에 보내고, 답변과 출처를 화면에 전달합니다. 제가 개발한 부분은 이 자료 수집과 검색, 질문 처리, 출처 연결입니다. RAG라고 기술 이름만 적는 것보다 이 순서를 설명하면 서버가 하는 일이 이해됩니다. 다만 혼합 검색을 구현했다고 해서 정확도가 자동으로 높아졌다고 말할 수는 없습니다. 정확도를 말하려면 정답이 있는 질문 목록으로 따로 평가해야 합니다. 그래서 여기에는 구현한 흐름과 현재 확인 가능한 범위를 적었습니다.

포트폴리오 8쪽 / 연구 성과와 포트폴리오 활용

마지막에는 대표 논문과 교육 활동, 더 자세한 자료를 볼 수 있는 링크를 넣었습니다. 앞에서 설명한 개발 경험과 제가 해 온 연구가 어떻게 이어지는지 보여 주는 페이지입니다. 여러분도 프로젝트 외에 공개한 코드, 관련 수업이나 활동, 실제로 받은 수상처럼 확인 가능한 경험을 넣으면 됩니다. 회사마다 새 경력을 만드는 것이 아닙니다. 기본 포트폴리오를 만들어 두고, 지원하는 직무와 관련된 사례를 앞에 놓고 설명의 비중을 조절하는 것입니다. 그리고 발표에서는 저는 이것을 할 수 있습니다로 끝내지 말고, 어느 프로젝트에서 무엇을 맡았고 어떤 결과로 확인했는지를 자기 말로 설명해 주세요.