

[Riiid] Santa Backend Engineer 사전과 제

📅 최근 수정일

2023년 5월 22일

▼ Type

A

📅 적용 시작일

2023년 5월 23일

🟠 J 댓글 추가

[과제전형 안내](#)

[기한](#)

[요구사항](#)

[인터뷰 내용](#)

[기본 시스템](#)

[관리자](#)

[학생](#)

[수강 신청](#)

[기술 스택](#)

[Language](#)

[Framework / Library](#)

[Database](#)

[Build](#)

[Test](#)

[Other Tools](#)

[제출 및 평가](#)

[제출 방법](#)

[포맷](#)

[평가](#)

[주의사항](#)

과제전형 안내

본 과제 전형은 주어진 기술 스택을 사용하여 요구사항을 만족하는 서비스를 직접 구현하는 것으로 진행됩니다. 본 문서에 명시된 요구사항 및 제약사항은 반드시 만족해야 하며 명시되지 않은 부분에 대해서는 지원자가 자유롭게 판단 및 결정하여 진행할 수 있습니다(이런 부분 또한 평가에 반영됩니다).

본 문서의 모든 내용은 평가에 직접적으로 반영되는 요구사항이거나 과제전형 진행에 도움이 되는 힌트일 수 있으니 불이익이 없도록 문서 전체를 꼼꼼히 읽고 과제 전형을 진행하시기 바랍니다.

기한

기한은 기본적으로 1주일이 주어지며, 최대 7일 연장이 가능합니다.

- 연장을 희망하는 경우 과제를 시작한 시점으로부터 **48시간** 내에 희망하는 기한을 회신하여 주시기 바랍니다. 1주일 또는 협의된 기한 내에 제출되지 않은 경우 미제출로 간주합니다.

요구사항

! 본 과제에 언급된 내용은 모두 창작된 것이며 실제와 관련이 없습니다.

퀴즈에서는 한 대학교와 제휴를 맺고 학생들에게 산타를 통한 온라인 영어 강의(이하 '강의'라고 한다)를 제공하기로 했습니다. 제공되는 강의에는 필수 강의와 선택 강의를 있으며 모든 학생은 필수 강의를 수강해야 졸업을 할 수 있게 됩니다. 강의 수강신청 기간이 얼마 남지 않은 시점에, 지원자는 수강신청 서비스를 개발하는 팀의 유일한 백엔드 개발자로 배치되었습니다.

이 서비스는 실제로 교직원과 학생이 사용할 것이므로 서비스의 품질은 실제로 운영 가능한 수준이어야 합니다. 즉, 서비스에는 오류가 없어야 하고 서비스가 충분히 견고해야 합니다. 잘못된 데이터가 등록되거나 시스템이 올바르게 작동하지 않는 상태를 갖지 않도록 지원자가 판단하기에 필요한 validation도 구현되어야 합니다. 수강신청 서비스의 특성상 짧은 시간 동안 많은 사용자가 동시에 접속하여 큰 트래픽을 발생시키므로 이런 특성에 대응할 수 있어야 합니다.

또, 앞으로도 이 서비스를 지속적으로 운영하고 개선해나가야 하므로 유지보수가 용이해야 하며 확장에도 열려 있어야 합니다. 그리고 지금은 지원자가 유일한 백엔드 개발자이지만 앞으로 총원될 다른 개발자와의 협업도 고려해야 합니다.

아래에는 제휴 비즈니스 담당자를 인터뷰하여 알게 된 정보들이 있습니다. 잘 정리된 요구사항은 아니지만 이 정보를 바탕으로 요구사항을 정리하고 요구사항을 바탕으로 수강 신청 서비스를 구현해야 합니다.

기한 내에 수강신청 서비스를 개발해 이 제휴 사업이 무사히 진행될 수 있도록, 그리고 학생들이 무사히 졸업할 수 있도록 도와주세요.

! API에 대해 인증은 구현하지 않아도 불이익이 없으니 선택적으로 구현하시면 됩니다. 만약 인증을 구현하지 않는다면, 요청 시 학생 ID 등을 함께 전송하는 등의 방법으로 구현해도 됩니다.

인터뷰 내용

기본 시스템

- 과목
 - 과목은 필수 과목과 선택 과목으로 구분됩니다.
 - 모든 과목에는 과목명과 고유한 과목 코드를 갖습니다.
 - 한 과목에 대해 한 학기에, 또는 서로 다른 학기에 서로 다른 여러 강의가 개설될 수 있습니다.
 - 모든 과목의 성적은 A, B, C, D, F로 구분됩니다.
- 강의
 - 모든 강의는 과목과 수강 정원을 갖습니다.
 - 수강 정원은 무제한일 수 없습니다.

- 학생
 - 모든 학생은 학번, 이수 학기 수, 과거 수강 내역 정보를 갖습니다.
 - 모든 학생은 8학기를 이수하면 졸업할 수 있습니다(8학기가 아닌 학생은 없다고 가정).
 - 단, 필수 과목을 수강하지 않았거나 F 성적이 있는 등의 이유로 졸업을 하지 못했을 수 있으므로 이수 학기는 8보다 클 수 있습니다.
 - 이수 학기가 8보다 같거나 큰 경우 졸업까지 남은 학기는 0으로 간주합니다.
 - 과거 수강 내역에는 수강한 강의와 성적 정보가 포함됩니다.
 - 학생은 이미 수강한 과목을 다시 수강할 수 있습니다.
 - 단, 이미 B 이상의 성적을 받았던 과목은 다시 수강할 수 없습니다.
 - 동일한 과목을 여러 번 수강한 경우 과거 수강 내역에는 가장 최근 수강한 강의와 성적만 포함됩니다. 즉, 과거 수강 내역에 같은 과목이 두 개 이상일 수는 없습니다.
- 학기
 - 학기는 이름(2023년 1학기, 2023년 여름계절학기, ...)을 갖습니다.
- 수강 신청
 - 수강 신청은 학기 생성 → 학기 확정 → 수강 신청 시작 → 수강 신청 종료 순서로 진행됩니다.
 - 수강 신청을 시작하기 위해서는 학기를 확정해야 합니다.
 - 학기가 확정되면 해당 학기에 개설되는 강의를 추가할 수 없습니다.
 - 수강 신청이 종료되는 시점에 학생들의 수강 신청 내역을 바탕으로 강의를 수강할 학생이 확정됩니다.
 - 수강 신청한 학생 수가 수강 정원보다 적거나 같다면 모든 학생의 수강이 확정됩니다.
 - 수강 신청한 학생 수가 수강 정원보다 많다면 수강 우선순위에 따라 우선순위가 높은 학생부터 순서대로 수강이 확정됩니다.

관리자

- 관리자는 과목을 생성할 수 있습니다.
- 관리자는 학생을 생성할 수 있습니다.
- 관리자는 학기를 생성할 수 있습니다.
- 관리자는 생성된 학기에 해당 학기에 개설되는 강의를 등록할 수 있습니다.
- 관리자는 학기를 확정할 수 있습니다.
- 관리자는 확정된 학기에 대해 수강 신청을 시작할 수 있습니다.
- 관리자는 수강 신청중인 학기에 대해 수강 신청을 종료할 수 있습니다.

학생

- 학기가 확정되기 전에는 학생은 해당 학기에 대한 어떠한 정보도 볼 수 없습니다.
- 학생은 확정된 학기에 대해 해당 학기에 개설되는 강의의 목록을 조회할 수 있습니다.
- 수강 신청이 시작되었을 때 학생은 해당 학기에 개설되는 강의를 수강 신청할 수 있습니다.
- 수강 신청이 종료되었을 때 학생은 자신의 수강 신청 결과를 확인할 수 있습니다.
- 수강 신청 결과에는 해당 학기에 수강 신청한 강의와 확정 여부가 포함됩니다.

수강 신청

- 수강 신청을 한 학생들은 수강 신청이 종료되는 시점에 수강 우선순위에 따라 수강이 확정됩니다.

- 학생들은 졸업을 하기 위해 필수 과목을 반드시 수강하여 F가 아닌 성적을 받아야 합니다. 그러므로 필수 강의의 수강 우선순위는 다음과 같습니다.
 1. 졸업까지 남은 학기가 적은 학생일수록 우선 순위가 높음.
 2. 졸업까지 남은 학기가 같다면 이수 학기가 많은 학생일수록 우선 순위가 높음.
 3. 단, 이미 동일한 과목에서 D 이상의 성적을 받은 학생은 우선 순위가 가장 낮으며 이들 간에는 졸업까지 남은 학기나 이수 학기를 고려하지 않음.
- 학생들은 선택 과목을 선택적으로 수강할 수 있으며 수강하지 않더라도 졸업할 수 있습니다. 그러나 F 성적을 받은 과목이 있는 경우에는 졸업할 수 없으므로 만약 F 성적을 받은 과목이 있다면 그 과목을 다시 수강하여 D 이상의 성적을 받아야 합니다. 그러므로 선택 강의의 수강 우선순위는 다음과 같습니다.
 1. 동일한 과목을 수강하여 F 성적을 받은 사람(동일한 과목을 2번 이상 수강한 경우 최근 성적이 반영됨).
 - a. 졸업까지 남은 학기가 적은 학생일수록 우선 순위가 높음.
 - b. 졸업까지 남은 학기가 같다면 가장 많은 학기를 이수한 사람일수록 우선 순위가 높음.
 2. 동일한 과목을 수강한 이력이 없는 사람.
 3. 동일한 과목을 수강한 이력이 있는 사람.
- 필수 강의와 선택 강의 모두에 대해서 수강 우선순위가 같은 학생들 간의 우선순위를 고려할 필요가 있는 경우 수강 신청을 먼저 한 학생일수록 우선 순위가 높습니다.
 - 이 과제에서는 수강 신청 시각까지 완전히 동일한 경우가 없다고 가정하며 이를 보장하기 위해 별도로 구현을 할 필요는 없습니다.

기술 스택

지원자는 다음 기술 스택을 사용할 수 있습니다.

- 별도로 명시되지 않은 도구, 기술 등에 대해서는 지원자가 임의로 결정할 수 있습니다.

Language

- Kotlin

Framework / Library

- Spring Boot 2.7
- Spring MVC
- JPA(Spring Data JPA)
 - QueryDSL 사용 가능
 - MyBatis, iBatis, jOOQ 등 사용 불가
- 로깅 필요시 Logback 사용

Database

- PostgreSQL 11+
- Flyway

Build

- Gradle
 - Gradle Kotlin DSL을 사용하여야 함(Groovy 사용 불가).

- Docker & Docker Compose
 - 서비스 실행을 위해 필요한 외부 환경(데이터베이스 등)은 docker compose를 사용해 구성할 수 있도록 한다.

Test

- JUnit5를 사용하여 작성
- Kotlin을 사용하는 경우 Kotest 사용 가능
- Testcontainers 사용 가능

Other Tools

- 설계 및 구현이 AWS 서비스에 의존하는 경우 LocalStack 사용 가능

제출 및 평가

제출 방법

- 하기에 언급된 문서 및 파일은 "파일명 **[Santa Backend Engineer_사전과제] 제출자명** " 으로 저장하여, **recruit@riiid.co** 로 제출해주세요.
1. 작업한 프로젝트 전체를 압축한 것
 - 빌드된 파일을 제외하기 위해 **gradle clean** 후에 제출해주세요.
 2. API 명세
 3. 제출한 코드를 실제로 실행 및 테스트할 수 있는 방법이 기재된 문서
 - 가급적 프로젝트 내 **README.md** 파일에 작성해주시기 바랍니다.
 4. 기타 각종 문서들을 압축한 것
 - 설계와 관련된 각종 문서, ERD 등 평가에 도움이 될 수 있는 자료가 있다면 함께 제출해주세요.

포맷

- 압축 파일은 **zip** 을 사용해 주세요.
- 문서 파일은 **pdf** 로 제출해 주시기 바라며 예외적으로 스프레드시트가 필요한 경우 엑셀 포맷(**xls** , **xlsx**)을 사용할 수 있습니다.
- 그 외 포맷은 가급적 특별한 뷰어 없이 열어볼 수 있는 것으로 제출해주시기 바라며 열리지 않는 경우 평가하지 않습니다.
 - 한글 파일(**hwp** 등)로 제출된 문서는 평가하지 않습니다.

평가

- macOS에서 평가하며 프로젝트 코드는 IntelliJ로 확인합니다.
- 제출한 모든 파일은 평가에 포함됩니다.
- 코드 한 줄 한 줄의 구현부터 전체 설계까지 모든 부분을 고려합니다. 모든 부분을 지원자가 생각하는 최선의 방식으로 진행해주시기 바랍니다.
- 평가는 약 1주일 정도 소요되며 제출한 자료의 분량에 따라 지연될 수 있습니다.
- 결과는 합격과 불합격 모두 안내해 드리며 별도로 불합격 사유나 코드리뷰 등을 제공해드리지 않습니다.

주의사항

- 지원자는 과제의 내용과 과제의 수행 결과 및 만들어낸 모든 산출물을 비밀로 유지하여야 합니다.
- Git 등의 버전 관리 도구를 사용하는 경우 로컬 레포지터리만을 사용해야 하며 GitHub 등 어떤 원격 레포지터리에든 코드가 업로드되지 않아야 합니다.
- 과제와 관련한 비밀 엄수 의무를 위반한 사실이 적발된 경우, 최종 합격 및 수습 기간 종료 이후라도 채용이 취소될 수 있으며 이에 대한 법적 책임 또한 질 수 있습니다.