**Pusan National University Computer Science and Engineering Technical Report 2021-02**

**이더리움 기반 학습 인증 서비스 “바름” 개발**



이승윤

김유미

심재영

지도교수 김호원

목 차

[1. 서론 1](#_Toc105090444)

[1.1. 연구 배경 1](#_Toc105090445)

[1.2. 기존 문제점 2](#_Toc105090446)

[1.3. 필요성 3](#_Toc105090447)

[1.4. 연구 목표 4](#_Toc105090448)

[2. 연구 배경 5](#_Toc105090449)

[2.1. 배경 지식 5](#_Toc105090450)

[2.1.1. 블록체인(BlockChain) 5](#_Toc105090451)

[2.1.2. 이더리움(Ethereum) 5](#_Toc105090452)

[2.1.3. 트러플(Truffle) 7](#_Toc105090453)

[2.1.4. 가나슈(Ganache) 7](#_Toc105090454)

[2.1.5. QR코드 7](#_Toc105090455)

[2.1.6. Node.js 7](#_Toc105090456)

[2.1.7. Express 8](#_Toc105090457)

[2.1.8. React 8](#_Toc105090458)

[2.1.9. Next.js 8](#_Toc105090459)

[2.2. 요구사항 9](#_Toc105090460)

[2.2.1. 계정 9](#_Toc105090461)

[2.2.2. 강의 검색 9](#_Toc105090462)

[2.2.3. 강의 수강 10](#_Toc105090463)

[2.2.4. 강의 생성 10](#_Toc105090464)

[2.2.5. 블록체인을 통한 학습 진행 과정 저장 10](#_Toc105090465)

[2.2.6. 블록체인을 통한 수업 수료 인증 10](#_Toc105090466)

[2.2.7. QR 코드로 수료 확인 11](#_Toc105090467)

[2.3. 개발 환경 12](#_Toc105090468)

[3. 연구 내용 13](#_Toc105090469)

[3.1. 시스템 구조도 13](#_Toc105090470)

[3.2. 웹 14](#_Toc105090471)

[3.2.1. 웹 기능 15](#_Toc105090472)

[3.2.2. 웹 사용 기술 19](#_Toc105090473)

[3.3. 서버 20](#_Toc105090474)

[3.3.1. 데이터베이스 22](#_Toc105090475)

[3.3.2. API 26](#_Toc105090476)

[3.4. 블록체인 34](#_Toc105090477)

[3.4.1. 스마트 컨트랙트 35](#_Toc105090478)

[3.4.2. 스마트 컨트랙트 컴파일 및 마이그레이션 39](#_Toc105090479)

[3.4.3. 상호작용 43](#_Toc105090480)

[3.5. 서비스 사용 과정 45](#_Toc105090481)

[3.5.1. 주요 기능 플로우 45](#_Toc105090482)

[3.5.2. 서비스 사용 시나리오 46](#_Toc105090483)

[4. 연구 결과 분석 및 평가 48](#_Toc105090484)

[4.1. 설계 변경 내역 48](#_Toc105090485)

[4.2. 연구 결과 분석 49](#_Toc105090486)

[4.3. 평가 61](#_Toc105090487)

[5. 결론 및 향후 연구 방향 62](#_Toc105090488)

[5.1. 결론 62](#_Toc105090489)

[5.2. 향후 연구 방향 63](#_Toc105090490)

[6. 개발 일정 및 역할 분담 64](#_Toc105090491)

[6.1. 개발 일정 64](#_Toc105090492)

[6.2. 역할 분담 65](#_Toc105090493)

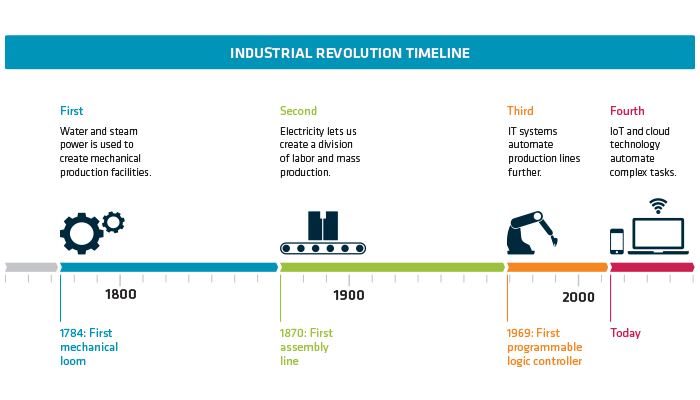
[7. 산업체 멘토링 결과 반영 66](#_Toc105090494)

# 서론

## 연구 배경

인간의 삶은 학습의 연속이다. 매일 사용하는 휴대 전화 사용 방법부터 언어, 수학, 그림, 노래, 지켜야 할 예절까지. 새로운 것을 탐구하여 배우고 반복해서 익혀 능숙하게 할 수 있게 된다. 고로 우리는 태어나서 죽을 때까지 평생을 배우며 살아가야 한다.

공자의 논어 학이편 1장에는 “학이시습지 불역열호(배우고 때때로 익힌다면 또한 기쁘지 아니한가)”가 가장 먼저 등장한다. 예부터 강조되어 온 학습의 중요성은 앞으로도 지속될 뿐만 아니라 더욱 강조되고 있다. 현대 사회의 인간은 넘쳐나는 정보통 속에 살고 있다. 그렇기에 방대한 정보 속에서 어떤 교육을 듣고 학습하며 어떤 능력을 갖추었는지를 분별하는 것 또한 매우 중요해지고 있다. 이로써 세상은 인간에게 더 많은 정보를 더 빠르게 학습하는 능력을 원하고 있다.

종래 변화의 속도는 점차 빨라져 이제는 1-2년 내에 모든 것이 바뀌는 시대다. 빠르게 변화하고 발전하는 기술과 유행에 대응해야 하는 개발자에게 있어서 학습 능력이 매우 중요하게 다가온다. 따라서 우리는 다양한 사람들이 질 좋은 교육 콘텐츠와 마주하고 학습을 통해 갖춘 자신의 능력을 체계화된 방법으로 증명할 수 있는 학습 서비스를 만들고자 한다.

[24]

## 기존 문제점

학습 분야에서 중요한 것은 학습 데이터이며 특히 데이터의 안전성과 투명성이 중요하다. 특정 서버에 저장된 데이터는 서버의 안정성과 보안성 수준에 따라 데이터 관리 수준이 직결된다. [1] 중앙 집중형 서버에서는 데이터 유실, 악의적 조작에 대한 위험을 간과할 수 없다. 특히 증명 관련 데이터와 학습 활동 데이터를 업체 또는 개인이 해킹하거나 이익을 위해 조작할 수 있는 가능성이 있다. [2]

인증서를 매번 발급하고 위변조를 확인해야 하는 문제를 해결할 수 없을까? 우리는 때때로 학습한 내용을 증명해야 할 필요가 있을 때가 있다. 이런 경우 수료증, 성적표, 졸업장같이 다양한 종류의 인증서를 매번 새로 발급해서 회사 또는 기관에 제출해야 한다. 인증서를 제출 받은 곳에서는 인증서의 위변조 여부를 발급 기관을 통해 별도로 확인하여 인증서를 다시 검증해야 한다. 또 발급 기관에서는 위변조 방지를 위해 통상적으로 일정 기간 이후에 만료되도록 설정하므로 미리 발급받은 인증서를 사용할 수 없는 경우도 있다.

결과만 나와있는 인증서, 학습 진행 과정은 알 수 없을까? 인증서를 발급받으면 수료한 과정(또는 과목)의 결과만 나와있어 학습 진행 과정을 파악하기 어렵다. 진행 과정을 파악할 수 있다면 학습자가 학습을 완료한 후 내려진 평가가 공정하였는지 투명하게 확인하고 비교할 수 있어 기존 평가 시스템의 공정성 문제를 해결할 수 있다. 학습 진행 중에도 각종 평가와 진행 사항을 손쉽게 나타낼 수 있으며 다른 사람과 공유할 수 있게 되는 장점도 있다.

온라인 학습 사이트에서 발행된 인증서 믿을 수 있을까? 우리는 인터넷을 통해 누구나 원하는 강의를 수강할 수 있다. 디지털 전환 속도가 빨라지고 비대면 교육의 필요성이 증가하면서 전문 온라인 교육 기관을 찾는 사람 역시 많아지고 있다. 영세한 사이트의 경우 홍보 또는 금전적인 목적으로 결과를 조작하여 인증서를 발급하는 문제가 발생할 수 있다. 심지어 사이트가 없어지는 경우 발급과 증명이 불가능한 상황이 발생한다. 특정 기업 또는 제작자가 대부분의 이익을 가져가고 참여를 원하는 소수가 공평하게 참여할 수 없는 형평성의 문제 [1]도 간과할 수 없는 이슈이다.

## 필요성

기존 학습 과정에서 발견한 문제 해결 방법은 없을까? 인증서를 매번 새로 발급하고 위변조 여부를 확인해야 하는 과정을 보다 안전하게 간단히 처리하고, 학습 과정을 확인하고 증명할 수 있도록 하여 평가의 공정성 문제를 해결하며, 누구나 강의를 만들 수 있고 인증서 발급과 학습 증명이 가능하도록 하는 기술. 우리는 블록체인에서 그 답을 찾을 수 있을 것이라 확신한다.

블록체인의 중요한 특징으로는 기록이 담긴 원장을 제3자에게 맡기지 않고, 참여자들이 직접 검증과 승인 등의 활동을 하며 만들고 관리할 수 있는 탈중앙성, 블록의 거래 기록을 참여자들 누구나 볼 수 있는 투명성, 연결된 블록은 수정하거나 삭제하기 어려운 불변성이 있다. [3] 이런 특징을 이용한다면 현재 학습 시스템이 가진 문제점을 획기적으로 해결할 수 있다.

먼저 수료증, 졸업장, 성적표 등 다양한 인증서를 필요할 때마다 매번 새로 발급받지 않아도 된다. 네트워크에 이미 연결된 어느 한 블록을 수정하거나 삭제하려면 모든 참여자의 컴퓨터에 있는 해당 블록의 모든 내용을 한꺼번에 바꾸어야 하기 때문에 내용 위변조가 거의 불가능하다. 학습 내용이 추가되어 내역이 변경되는 경우에도 블록체인 네트워크의 원장에 반영되어 최신 상태를 확인 가능하다. 따라서 인증서를 새로 발급하고 위변조 여부를 감정하는 과정을 없앨 수 있다.

두번째로 수료한 과정(또는 과목)의 결과만 나와있는 인증서와 달리 학습을 진행하면서 받은 평가와 진행 사항을 블록에 기록하여 학습 진행 과정을 명확하게 파악할 수 있다. 학습자가 어떤 순서로 과정을 수강했는지, 활동을 했다면 어떤 역할을 맡았는지, 또 어떤 평가를 받았는지를 투명하게 확인할 수 있다. 이를 통해 학습자는 강의자의 평가(성적 부여)가 타당한지 확인할 수 있다. 예비학습자와 검증자는 평가 시스템을 믿을 수 있는지 판단할 수 있다.

마지막으로 누구나 강의를 만들고 믿을 수 있는 학습 증명이 가능하다. 특정 업체와 제작자가 독점하여 큰 이익을 가져가는 것이 아니라 참여를 원하는 강의자가 공평하게 참여하여 강의를 제작하는 생태계를 구축할 수 있다. 악의적인 온라인 업체가 돈만 받고 인증서를 발급해주거나 수강자의 성적을 우수하게 조작하여 교육 효과가 큰 것처럼 실적을 조작하는 것이 불가능해진다. 또 중앙 집중형 서버에서 인증서를 발급하고 증명해주는 것이 아니라 참여자 간에 직접 기록을 확인할 수 있어 온라인 강의 사이트가 사라져서 인증을 받을 수 없는 상황을 원천적으로 차단할 수 있다.

## 연구 목표

개발할 서비스는 쉬운 학습 과정 인증, 투명한 평가 시스템, 신뢰받는 열린 학습 시스템으로 블록체인 기술을 활용하여 기존 학습 시스템을 획기적으로 개선한다. 학습 플랫폼 이름은 ‘바름’으로 정하였다. 표준국어대사전을 보면 “바르다”에는 3가지 의미가 있다.

1. 사실과 어긋남이 없다, 겉으로 보기에 비뚤어지거나 굽은 데가 없다. [4]

* 블록체인을 통한 학습 진행 과정 저장, 과정(수업) 수료 인증으로 투명하고 정확한 학습 절차 인증 과정을 구현한다.

1. 껍질을 벗기어 속에 들어 있는 알맹이를 집어내다. [4]

* 학습 결과만 나오는 껍데기 인증서 대신 알맹이인 학습 진행 과정을 살펴볼 수 있어 학습자와 검증자 모두가 믿을 수 있는 평가 시스템을 구현한다.

1. 다른 물건(물체)의 표면에 고르게 붙인다. [4]

* 특정 업체의 독점 없이 강의를 원하는 사람이라면 고르게 누구나 참여할 수 있는 학습 플랫폼을 구현한다.

다양한 블록체인 플랫폼 중 이더리움을 기반으로 과제를 수행할 것이다. 이더리움은 퍼블릭 블록체인이고 스마트 컨트랙트가 동작하는 탈중앙화 어플리케이션을 개발할 수 있어 학습 데이터의 안전성과 투명성을 보장하여 문제를 해결하는데 가장 적합하다고 판단하였다. 과제를 수행하면서 이더리움으로 학습 데이터를 블록에 추가하고 스마트 컨트랙트를 통해 수업과 과정 수료 인증 기능을 구현하는 것이 목표이다.

매번 발급하고 위변조를 확인해야 하는 인증서, 결과만 나와있는 인증서, 다양한 기관에서 발행된 인증서, 과연 믿을 수 있을까? 우리 PLMS팀은 블록체인을 이용하여 자신의 능력을 체계화된 방법으로 증명하고 다양한 사람들이 질 좋은 교육 콘텐츠와 만날 수 있는 “바름” 플랫폼을 개발하였다.

|  |
| --- |
| 요구조건 |
| 강의 수강  – 웹을 통해 강의를 수강할 수 있는 시스템 |
| 강의 수강 인증  - 학습 데이터를 이더리움 블록체인 상에 저장, 읽기  - 스마트 컨트랙트로 강의 수료 처리 |
| 강의 생성  – 웹을 통해 강의를 생성할 수 있는 시스템 |

# 연구 배경

## 배경 지식

### 블록체인(BlockChain)

블록체인은 여러 거래내역을 묶어 블록을 구성하고 해시를 이용하여 여러 블록을 체인처럼 연결하여 다수의 사람들이 복사하여 분산 저장하는 알고리즘이다. [5] 비즈니스 네트워크에서 거래내역을 기록하고 자산을 추적하는 프로세스를 효율화하는 변하지 않는 공유 원장이다. [6] 블록체인 기술은 신뢰성, 안전성, 탈중앙화의 특성을 가진다. 데이터의 위변조가 불가능하여 중개기관 없이 신뢰 가능한 안전한 거래, 데이터 처리가 가능하다. [5]

교육 분야에서는 데이터의 안정성과 투명성, 형평성 문제를 [2] 해결하기 위해 블록체인 도입이 필요하다. 먼저 매번 새로 인증서를 발급받지 않아도 되고 위변조 감별 절차가 간소화된다. 블록체인 네트워크에 학습 데이터가 지속적으로 반영되므로 항상 최신 상태의 학습 내역을 확인할 수 있다. 데이터 조작이 불가능하므로 기록된 학습 데이터를 신뢰할 수 있다. 다음으로 학습 진행 과정을 볼 수 있어 평가 시스템의 신뢰성을 높이고 강의의 질을 높인다. 학습자는 강의자의 평가가 적절한지 확인할 수 있으며 다른 학습자를 통해 학습 기간과 강의 품질을 확인할 수 있다. 검증자는 학습자가 강의에 적극적으로 참가했는지 강의와 학습자의 수준은 어느 정도되는지 자세하게 파악할 수 있다. 학습 과정이 공개되므로 질 낮은 강의와 자격증으로 교육 효과가 큰 것처럼 광고하거나 [7] 돈만 받고 인증서를 발급해주는 악의적인 사례를 예방할 수 있다. 마지막으로 누구나 강의를 만들고 참여할 수 있다. [1] 참여를 원하는 강의자가 공평하게 참여하여 강의를 제작하고 인증해 줄 수 있다. 강의가 사라지더라도 학습 데이터는 그대로 보존되므로 영구적인 학습 인증이 가능하다.

### 이더리움(Ethereum)

이더리움은 암호 화폐 이더와 분산형 어플리케이션을 구동하는 커뮤니티 운영 기술(플랫폼)이다. [8] 이전의 블록체인 기술은 화폐에 특화된 기능으로 구현되어 암호 화폐 이외 영역에서 사용되기에는 어려움이 있었다. 그러나 이더리움은 프로그래밍이 가능하여 스마트 컨트랙트가 적용된 탈중앙화 어플리케이션(이하 디앱)을 개발할 수 있다. 디앱은 이더리움에 배포되면 항상 프로그래밍된 대로 동작하므로 완전히 신뢰가능하다. [8] 이더리움에서 스마트 컨트랙트를 도입함으로써 블록체인을 이용하여 위변조 방지가 필요한 계약서 작성 기능을 구현할 수 있게 되었다.

이더리움의 스마트 컨트랙트는 자동 실행, 예측 가능한 결과, 공공 기록, 개인 정보 보호, 공개된 조건의 특징을 가진다. [9] 스마트 컨트랙트는 계약 조건이 만족되면 결과가 자동으로 실행된다. 예를 들어 강의자가 등록한 영상을 모두 시청하면 해당 수업을 수료한 것으로 처리할 수 있다. 스마트 컨트랙트는 예측 가능한 결과를 생성한다. 말 또는 글로 표현된 계약은 사람마다 서로 다르게 해석할 가능성이 존재한다. 그러나 스마트 컨트랙트는 코드로 작성된 조건에 따라 정확하게 실행되며 동일한 상황에서 동일한 결과를 생성한다. 스마트 컨트랙트는 공적인 기록에도 적합하다. 학습 데이터는 공개된 블록체인에 있기 때문에 검증자가 학습자의 학습 정보를 정확하게 추적, 검증할 수 있다. 이더리움은 가명 네트워크로 거래에는 거래자의 신원이 아닌 고유한 암호화 주소가 사용된다. 따라서 관찰자로부터 개인 정보를 보호할 수 있다. 스마트 컨트랙트의 계약 조건을 미리 확인할 수 있다. 계약 조건이 모두에게 투명하게 공개되어 있기 때문에 신용할 수 있다. 예를 들어 모든 영상을 수료하고 평균 점수가 60점 이상일 때 수료를 하는 스마트 컨트랙트가 있다면 해당 스마트 컨트랙트의 적용을 받는 수업의 수료 인증을 신뢰할 수 있다.

이더리움은 프로그래밍을 통해 화폐에 기능과 목적을 추가할 수 있으며 튜링 완전성을 지원하는 언어를 내장하여 어플리케이션을 위한 보안 및 다른 어플리케이션과의 효율적인 상호작용이 중요한 상황에 주안점을 두고 있다. [10] 이더리움 계좌에는 외부 소유 계좌와 컨트랙트 계좌가 있다. 외부 소유 계좌는 사용자의 계좌로 별도의 프로그래밍 코드 없이 거래 진행 시 트랜잭션을 발생시킨다. 컨트랙트 계좌는 코드를 활성화시키고 명령을 수행하게 된다. [11] 다시 말해 스마트 컨트랙트 관련 트랜잭션을 외부 소유 계정에서 만들고 컨트랙트 계정에서 이를 처리한다.

이더리움 트랜잭션은 논스, 가스 가격, 가스 한도, 수신자, 값, 데이터, 서명으로 구성된다. 논스는 발신하는 외부 사용자 계정에서 발행되어 메시지를 재사용하는 것을 방지하는데 사용된다. 가스 가격은 지급할 가스 가격으로 연산마다 가스 가격이 달라진다. 가스 한도는 트랜잭션에서 사용할 가스 최대치이다. 악의적으로 반복해서 수행되는 코드를 방지하기 위해 수수료를 지급하도록 하여 연산에 제한을 둔다. 수신자는 수신자의 이더리움 주소이다. 값은 발신자가 보내는 이더이다.

이더리움 메시지는 컨트랙트 계정간 주고받는 일종의 객체이다. [10] 발신처, 수신처, 값, 데이터, 가스 한도로 구성된다. 값은 메시지와 함께 전달되는 이더이다. 트랜잭션과 마찬가지로 가스 한도가 있어 공격자로부터 악의적인 연산을 막는다.

### 트러플(Truffle)

트러플은 이더리움 가상 머신을 사용하는 블록체인을 위한 개발 환경, 테스트 프레임워크, 에셋 파이프라인을 제공한다. [12] Truffle boxes는 dApp을 빠르게 만들 수 있도록 하는 여러 템플릿으로 구성된다. 이 중 Express-Box를 참고하여 API 서버를 우선적으로 제작한 뒤 기존 서버 코드와 스마트 컨트랙트 관련 코드의 통합 작업을 수행하였다.

### 가나슈(Ganache)

가나슈는 이더리움 개발에 사용할 수 있는 개인용 블록체인으로 스마트 계약을 배포, 테스트해볼 수 있는 간이 블록체인을 제공한다. 네트워크 연결 필요 없이 로컬에서 작동시킬 수 있다. [13] Truffle은 Ganache를 통한 테스트를 기본적으로 지원하므로 로컬 환경에서 자유롭게 작성한 스마트 컨트랙트를 배포하여 테스트할 수 있다.

### QR코드

QR코드는 흑백 격자무늬 패턴 코드로 정보를 나타내는 매트릭스 형식의 이차원 코드이다. 기존에 사용되던 바코드는 1차원으로 숫자, 문자 정보가 가능하였다. QR 코드는 2차원 형태를 가져 바코드보다 더 많은 정보를 담을 수 있으며 더 작은 공간을 차지한다. 오염과 손상에 강하며 어느 방향에서도 인식 가능하다는 특징이 있다. 최대 16분할하여 데이터를 표현하여 좁고 긴 표면에도 큰 데이터를 인쇄할 수 있다. [14]

인증서 검증을 위해 QR코드를 스캔할 경우 사이트로 이동하여 학습 과정을 볼 수 있도록 구현할 수 있다. 또는 암호화된 사용자 주소를 삽입, 이를 이용하여 학습 정보를 추적, 검증할 수 있도록 할 수 있다.

### Node.js

Node.js는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 엔진이다. 비동기 이벤트 기반 자바스크립트 런타임으로 확장성 있는 네트워크 어플리케이션 설계에 유용하다. 다수의 연결을 동시에 처리할 수 있으며 실행할 작업이 없는 경우 대기한다. [15] 특히 I/O 처리에서 우수한 성능을 보여 학습 플랫폼에 적합하다고 판단되어 도입하였다. 아래에서 설명할 Express와 React 모두 노드 기반으로 동작하는 프레임워크이다.

### Express

Express는 Node.js를 위한 빠르고 개방적인 간결한 웹 프레임워크이다. 유틸리티 메소드와 라우팅 기능, 다양한 미들웨어를 사용하여 쉽고 빠르게 강력하게 API를 작성할 수 있다. [16] 계정, 강의 찾기, 강의 수강, 학습 진행 과정 저장, 과정/수업 수료 인증 같은 기능 수행시 데이터베이스 추가/수정/제거/저장, 이더리움 블록체인 네트워크에 기록 및 조회하는 API 서버 역할을 수행한다.

### React

사용자 인터페이스를 만들기 위한 자바스크립트 라이브러리이다. 노드 서버에서 렌더링한다. 리액트는 데이터가 변경되면 변경이 필요한 컴포넌트만 갱신하여 랜더링하기 때문에 효율적이고 상호작용이 많은 UI를 만들 때 생기는 어려움을 줄여준다. 컴포넌트는 스스로 상태를 관리하는 캡슐화된 자바스크립트 코드이다. 컴포넌트를 조합하여 더 복잡한 UI를 설계할 수 있다. DOM과는 따로 상태를 관리할 수 있고 다양한 형식의 데이터를 앱 내부에서 쉽게 전달 가능하다. [17] 사용자 인터페이스는 리액트를 통해 표시되며 익스프레스로 제작된 API 서버와 연동하여 데이터를 표시한다.

### Next.js

Next.js는 생산(프로덕션)을 위한 리액트 프레임워크이다. 별도 구성 없이 하이브리드 정적/서버 랜더링, 스마트 번들링, 라우트 프리패칭 기능을 사용 가능하다. [18] 기존 React가 클라이언트 브라우저에서만 컨텐츠를 렌더링할 수 있었는데 서버 측에서 렌더링된 어플리케이션이 포함되도록 기능을 확장하였다. [19] React가 자바스크립트에 대한 액세스 권한이 없거나 비활성화된 사용자를 수용하지 않는 문제, 잠재적인 보안 문제, 페이지 로드 시간 연장, 검색 엔진 최적화에 악영향을 끼치는 점을 해결하기 위해 Next.js는 초기 전송 시 서버 측에서 렌더링 되도록 하였다. [20]

## 요구사항

### 계정

학습 시스템 접근과 사용자 구분을 위해 계정 기능이 필요하다. 세부적으로 로그인, 로그아웃, 회원가입 기능이 요구된다.

학습을 하거나 강의를 등록하기 원하는 사용자는 회원가입을 통해 기능 접근 권한을 부여 받아야한다. 회원가입을 위해 이름, 아이디(이메일), 비밀번호, 휴대전화 번호를 입력 받는다. 사용자 아이디는 다른 사용자와 중복해서 사용될 수 없으며 8자리 이상을 사용해야 한다. 회원가입을 완료한 사용자는 로그인 화면으로 이동한다.

회원가입한 사용자는 로그인하여 학습 또는 강의를 등록할 수 있다. 회원가입시 입력한 아이디와 비밀번호를 통해 로그인 가능하다. 로그인에 성공할 경우 메인 화면으로 이동한다. 실패한 경우 다시 로그인해야 한다.

로그인한 사용자는 로그아웃하여 컴퓨터를 사용하는 다른 사용자가 강의를 수강하거나 등록하지 못하게 할 수 있다. 로그아웃에 성공할 경우 메인 화면으로 이동한다.

### 강의 검색

강의 수강을 위해 강의 검색으로 강의를 찾아 강의 정보(상세)를 확인할 수 있는 기능을 제공한다.

사용자는 강의 조회(검색) 기능을 이용하여 과정과 강의를 검색할 수 있다. 검색 상자에 원하는 검색어를 입력하면 강의 제목, 강의 요약, 강의자에 해당 검색어 키워드가 포함되어 있는 경우 목록으로 표시된다. 검색 결과 목록의 각 항목은 강의 사진, 강의 제목, 강의 요약, 강의자 정보로 구성된다. 사용자가 원하는 강의를 선택하면 강의 정보(상세)가 표시된다.

강의 정보(상세) 기능으로 강의 정보를 살펴볼 수 있다. 강의 제목, 강의 요약, 강의자, 강의 목록을 표시한다. 강의 진도는 블록을 이용하여 학습한 순서대로 표시한다. 완료한 순서대로 영상, 시험 종류에 따라 별도로 표시하고 완료 날짜와 점수를 함께 제공한다. 강의 공지사항은 최신 게시글을 기준으로 5개까지 목록으로 표시한다. 공지사항 게시글은 강의별로 구분되며 제목, 내용, 작성자, 작성일자로 구성된다.

### 강의 수강

강의 수강 기능으로 강의를 학습할 수 있다. 좌측에는 강의에서 학습할 내용이 순서대로 목록으로 표시된다. 영상, 시험 종류 별로 다른 화면을 사용한다. 수강(영상)은 수강 필요, 수강 완료 상태가 존재한다. 시험은 응시 필요, 응시 완료, 채점 완료 상태가 존재한다. 채점 완료 상태인 경우 점수를 표시한다. 영상, 시험 항목에는 최종 상태 변화 시간을 추가하며 수강(영상)은 수강 완료, 시험은 채점 완료 상태가 되면 네트워크 상의 블록에 기록된다.

### 강의 생성

로그인한 사용자는 강의를 생성할 수 있다. 강의명, 강의 설명, 카테고리, 강의 대표 이미지를 입력해야 한다. 강의 내용을 추가할 때에는 영상의 경우 제목과 url을 입력해야 하며 시험의 경우 제목과 질문, 답을 입력해야 한다. 강의명, 설명, 카테고리, 이미지는 수정, 삭제가 가능하다. 단, 한 번 등록한 강의 내용은 수정은 가능하나 삭제는 불가능하다.

### 블록체인을 통한 학습 진행 과정 저장

강의 수강 과정에서 수행한 학습 활동은 블록체인에 저장된다. 수강 완료, 시험 채점 완료 상태가 되었을 때 원장에 기록한다. 이 때 강의 번호, 컨텐츠 번호, 학습자 이메일, 날짜, 상태, 점수, 피드백이 저장된다. 학습자는 강의 정보 화면에서 블록 형태로 정리된 학습 진행 사항을 확인할 수 있으며 QR 코드를 이용하여 학습 진행 과정을 공유할 수 있다. 인증서에는 각 진행 과정 별로 제목, 수강 완료일, 점수, 블록 해시와 트랜잭션 해시가 표시된다.

### 블록체인을 통한 수업 수료 인증

블록체인에 저장된 학습 진행 과정 내용을 바탕으로 스마트 컨트랙트를 통해 조건이 갖춰지면 자동으로 수업을 수료한 것으로 기록한다. 학습자는 수료한 수업에 대해 인증서를 발급할 수 있다. 인증서에는 학습자 이름과 QR 코드가 표시되며 각 완료한 강의별로 강의명, 강의자, 강의 요약, 분류, 수강 완료일, QR코드가 표시되며 QR코드를 통해 강의별로 인증서를 찾아볼 수 있다.

### QR 코드로 수료 확인

검증자 또는 학습자로부터 공유 받은 자는 QR코드를 통해 학습 진행 과정과 수료 여부를 확인할 수 있다. QR코드에 링크를 삽입하여 학습 진행 과정 및 수료 여부 확인 화면으로 연결된다. 해당 화면에서는 블록 형태로 학습 진행 과정이 표시된다. 학습 과정의 강의명(또는 시험명), 종류, 점수, 완료 날짜, 피드백이 함께 표시된다. 아래에는 상세 내용으로 학습 과정이 한 번 더 표시되며 수강 순서, 강의명, 완료 날짜, 점수, 블록 해시와 트랜잭션 해시가 표시된다.

## 개발 환경

|  |  |
| --- | --- |
| 개발 언어 | JavaScript, Solidity |
| 개발 도구 | Visual Studio Code, IntelliJ IDEA, Git  Node.js, Express, React, Next.js  Web3.js, Truffle Suite, Ganache |
| 실행 환경 | Windows 10/11, Ubuntu 20.04.2  Chrome, Edge |
| 데이터베이스 | MySQL |

|  |  |  |
| --- | --- | --- |
| 웹 | 서버 | 블록체인 |
| Next.js | Express | Ethereum |
| @emotion/react  @emotion/styled  @mui/material  axios  classnames  grommet  mui  next  next-images  react  react-dom  react-icons  react-loader-spinner  react-player  react-qr-code  sass  styled-components | bcrypt,  cookie-parser,  cors,  csv-parse,  debug,  ejs,  express,  express-session,  http-errors,  morgan,  mysql2,  node-sass-middleware,  passport,  passport-local,  truffle-contract,  web3 | Truffle Suite  Ganache |
| JavaScript | JavaScript | Solidity |

# 연구 내용

## 시스템 구조도

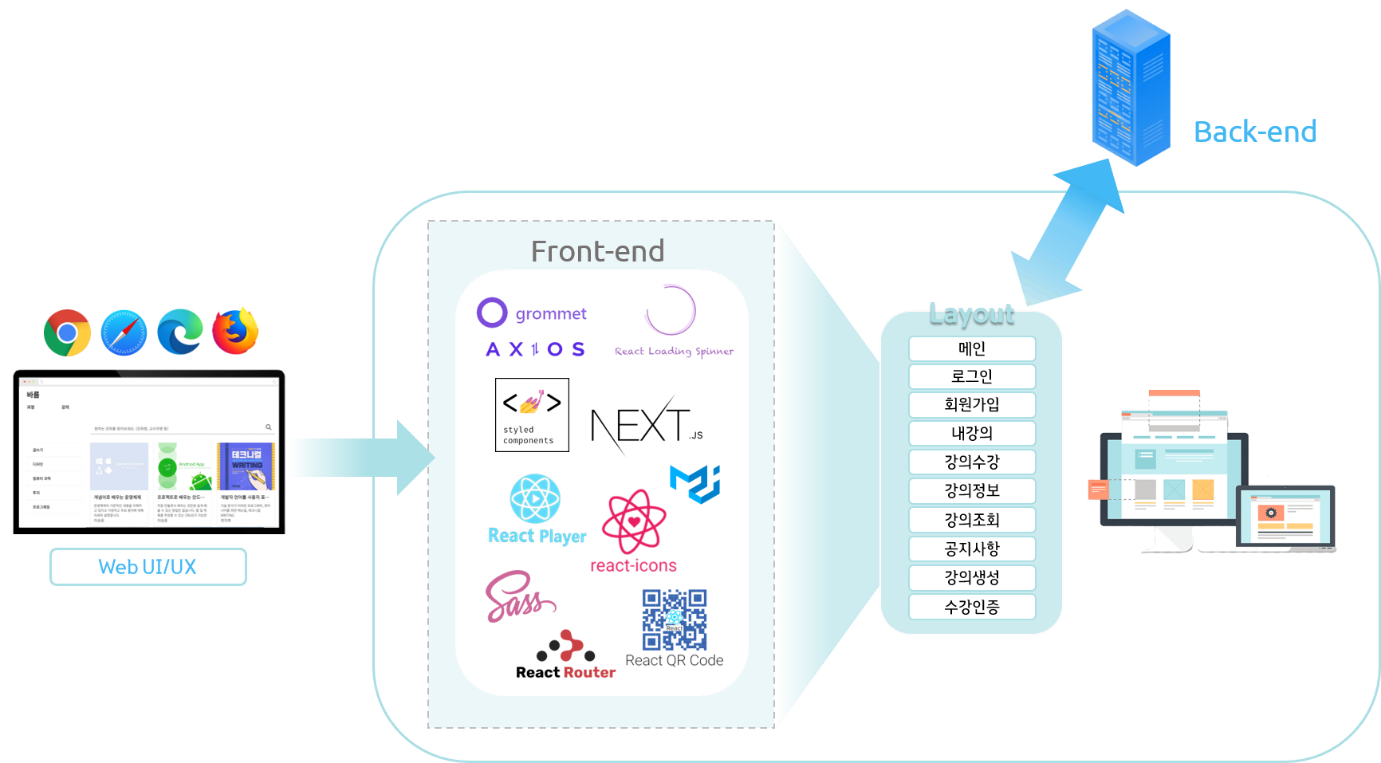
텍스트, iPod, 주차장, 전자기기이(가) 표시된 사진

자동 생성된 설명

바름 서비스는 크게 **웹, 서버, 블록체인**으로 구분할 수 있다

* Web : 사용자와 직접 상호작용하며 서버와 API를 통해 통신
* Server : 웹에서 받은 사용자 요청을 처리하고 데이터베이스에 기록, 블록체인과 웹을 연결해주는 기능도 수행
* Blockchain : 이더리움 네트워크를 선택. Solidity로 작성한 스마트 컨트랙트를 트러플로 컴파일 및 마이그레이션

## 웹

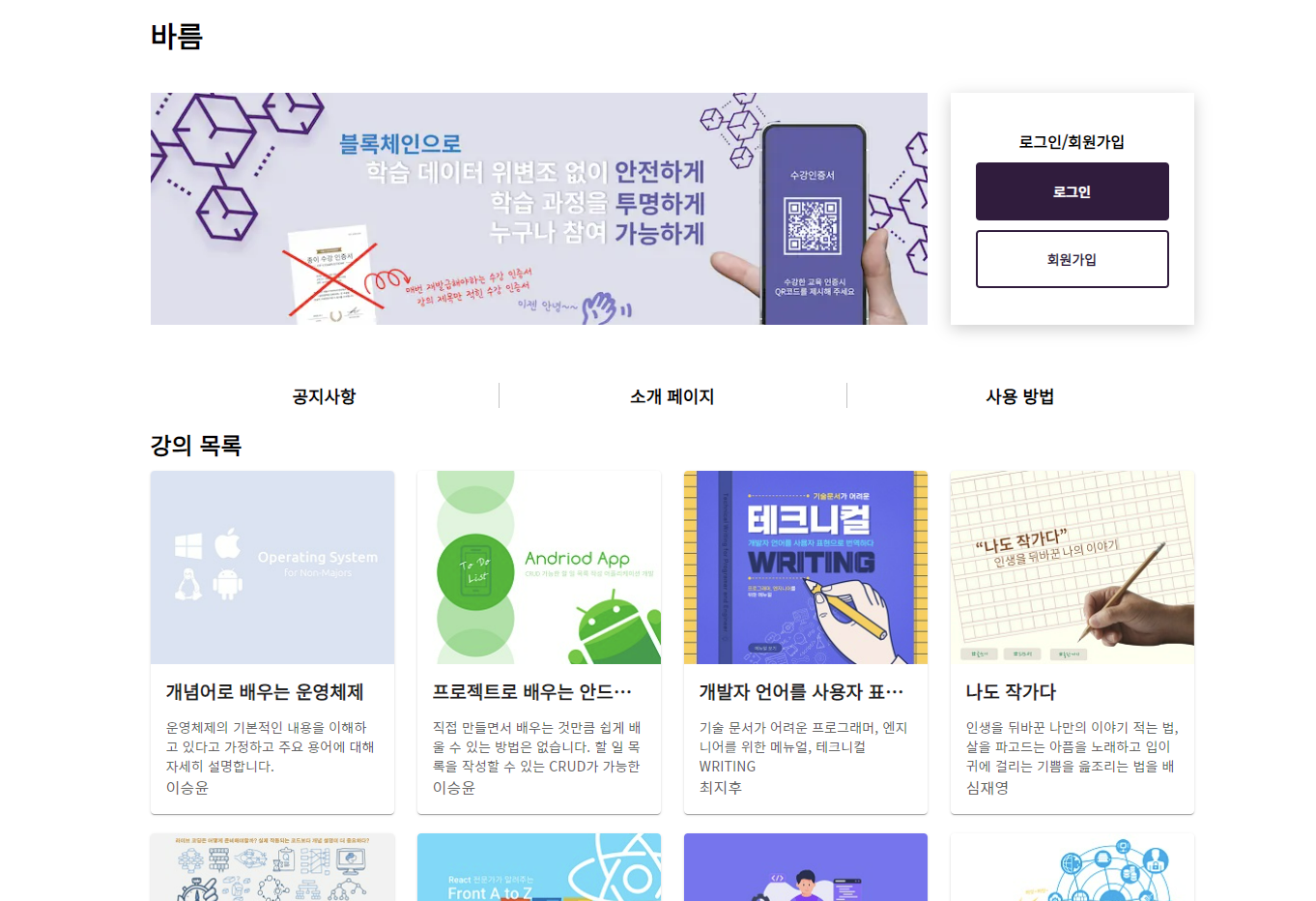


웹은 서버 API에서 가져온 데이터를 출력하고 사용자가 입력한 데이터를 서버로 전송하며 사용자와 서비스를 연결해주는 역할을 수행한다. node.js 기반으로 구축된 오픈 소스 웹 개발 프레임워크인 next.js를 사용하였다.

### 웹 기능

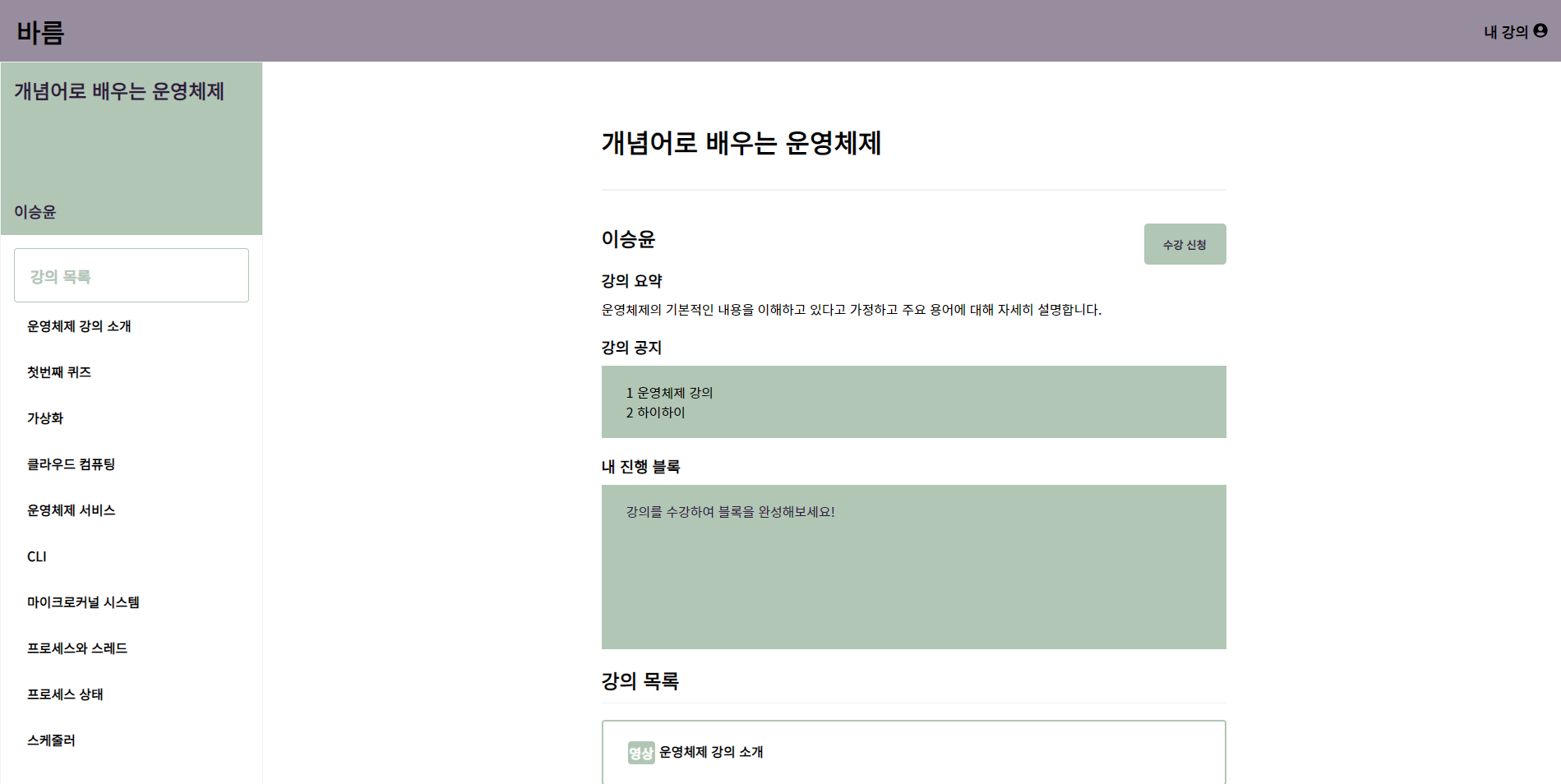
웹 페이지는 크게 메인, 강의 수강, 강의 생성, 수강 인증서로 구성된다.

* 메인 페이지



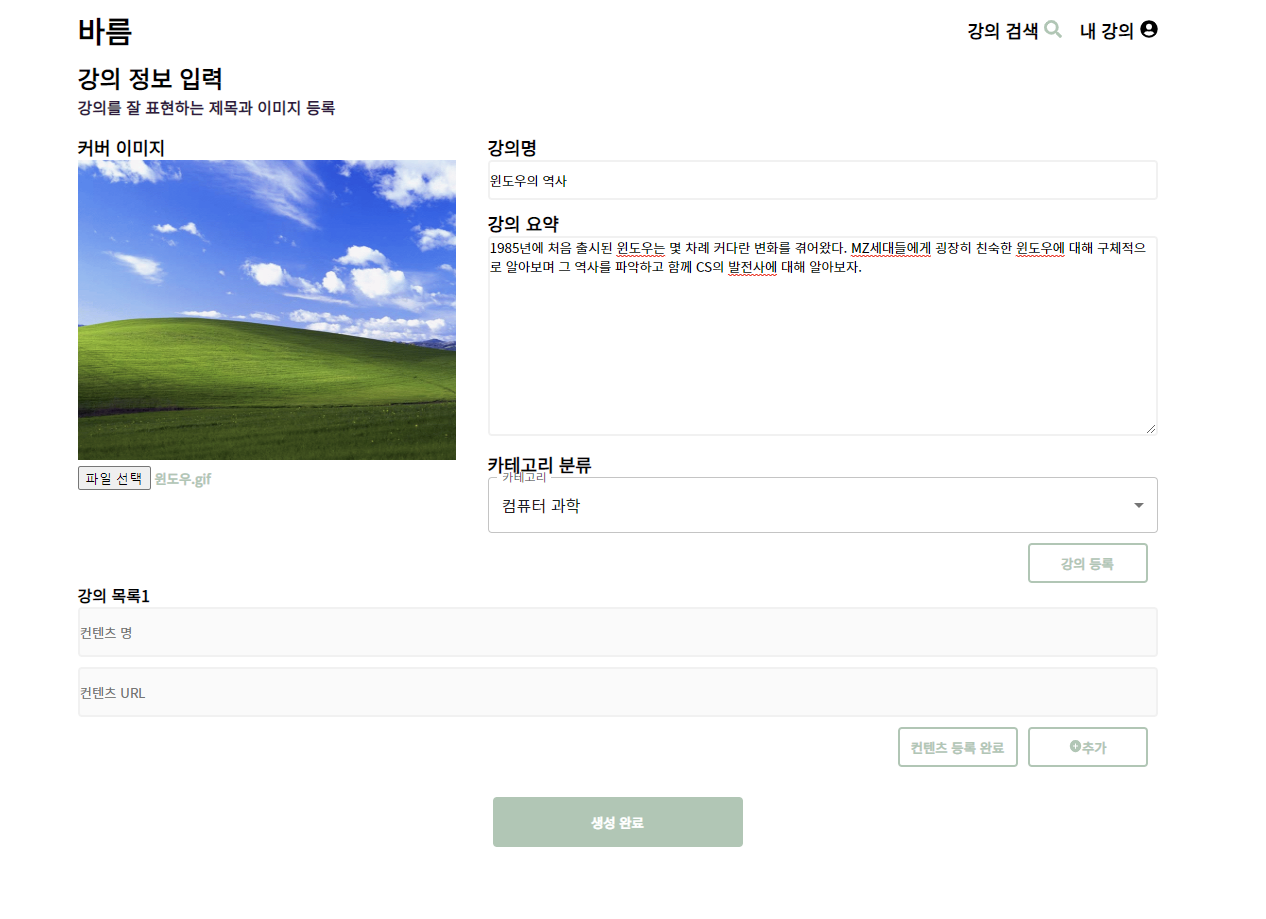
메인 페이지에는 로그인, 로그아웃, 회원가입, 강의 검색, 강의 목록 확인 기능이 있다. 회원가입을 통해 사용자 정보를 등록하고 로그인하여 사용자 인증을 할 수 있다. 로그아웃으로 연결을 끊을 수 있다. 강의 목록을 통해 현재 데이터베이스에 저장된 강의 중 8개를 볼 수 있고, 강의 검색으로 원하는 카테고리, 강의명, 교수자명을 가진 강의를 확인할 수 있다.

* 강의 수강 페이지



강의 수강 페이지는 수강 완료/수강 중/생성 강의, 강의 공지, 강의 정보, 강의 수강, 진행 상태 확인, 수강 신청 기능을 가지고 있다. 내 강의 목록에서 수강 완료/수강 중/생성 강의들을 확인하고 강의 정보를 확인하여 진행 상태를 알 수 있다. 교수자는 강의 공지를 수정하고 삭제할 수 있다. 아직 수강하지 않은 강의라면 수강 신청 기능을 사용하여 강의를 수강할 수 있다.

* 강의 생성 페이지



강의 생성 페이지는 강의 생성, 강의 수정, 강의 삭제, 컨텐츠 추가 기능이 있다. 강의 생성 기능을 통해 커버 이미지, 강의명, 강의 요약, 카테고리를 입력하여 새로운 강의를 생성할 수 있다. 컨텐츠 추가 기능을 사용하여 컨텐츠 명과 URL을 입력해 컨텐츠를 추가할 수 있다. 강의는 자유롭게 수정, 삭제할 수 있다.

* 수강 인증서 페이지

텍스트이(가) 표시된 사진

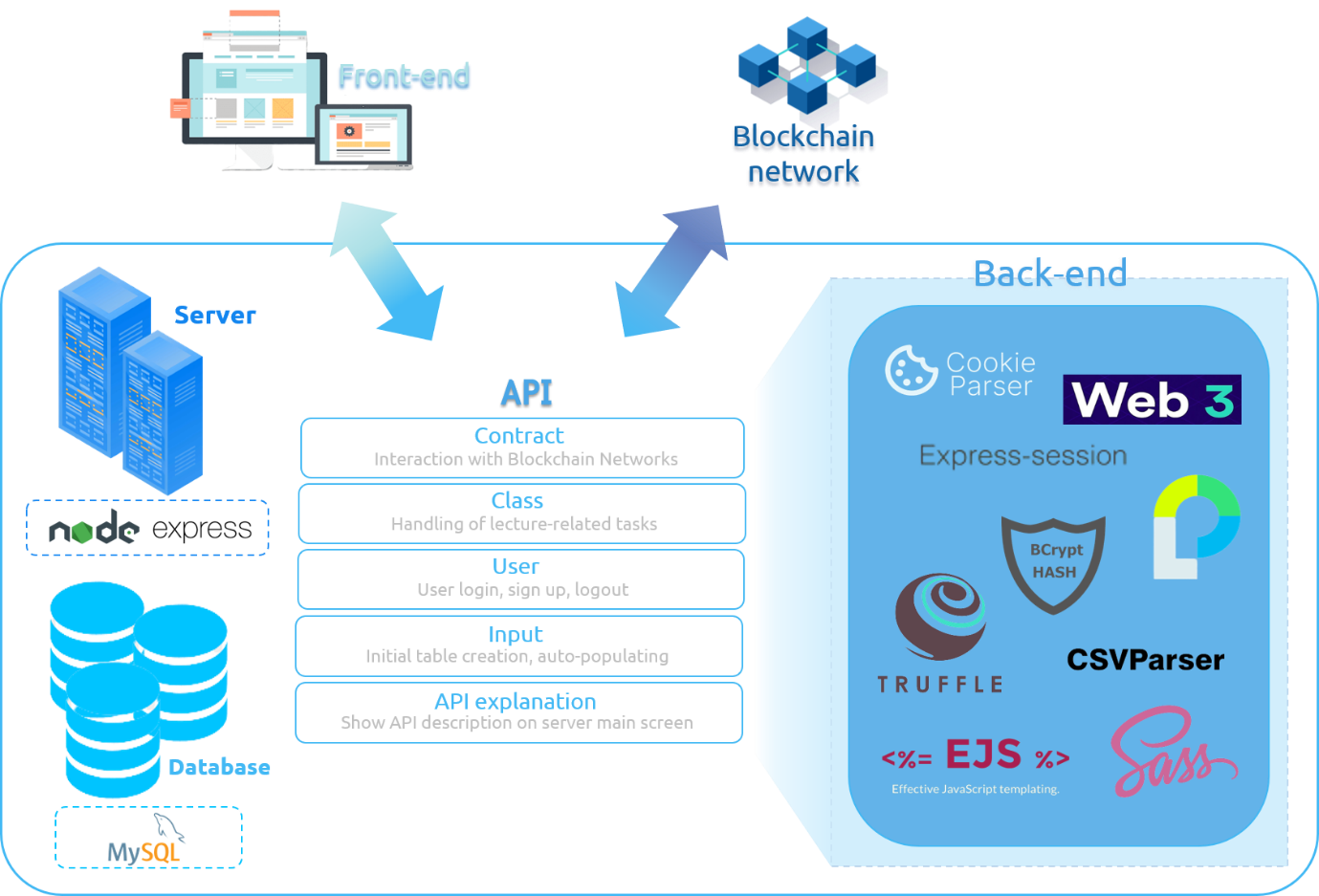
자동 생성된 설명

수강 인증서 페이지는 수강완료한 개별 강의와 강의 목록을 확인할 수 있는 인증서로 구성되어 있다. 수강완료 강의 목록 인증서를 통해 사용자가 수강완료한 모든 강의 목록을 확인할 수 있고, 수강완료 개별 강의 인증서를 통해 사용자의 구체적인 학습 정보와 진행 상황을 확인할 수 있다. 인증서는 QR코드를 통해 누구든지 접속이 가능하다.

### 웹 사용 기술

|  |  |
| --- | --- |
| MUI v4에서 v5로 이사 준비 | DevelopersIO | MUI는 리액트 개발에서 쉽게 사용할 수 있는 UI Framework으로 Google이 제안한 디자인 시스템으로 높은 퀄리티의 디지털 경험을 제공하는 것을 목표로 한다. 각 컴포넌트의 레이아웃 디자인에서 사용하였다. |
| post-thumbnail | Axios는 브라우저와 Node.js를 위한 promise 기반 HTTP 클라이언트이다. 확장 가능한 인터페이스를 가진 작은 패키지로 사용하기 쉬운 라이브러리를 제공한다. 웹과 서버간의 통신을 위해 사용하였다. |
|  | React Player는 파일, YouTube, FaceBook, Twitch 등 다양한 URL을 재생할 수 있도록 지원하는 React Component이다. 강의 영상을 재생하고 수강 완료시 이벤트가 실행되도록 하기 위해 사용하였다. |
| 간단한 ⚛️ QR 코드 생성 및 스캔을 위한 React 앱 - wenyanet | React QR Code는 URL으로 이동할 수 있는 QR코드를 생성해주는 react와 react native 라이브러리이다. 누구든지 인증서를 확인할 수 있게 하기 위해 인증서에 사용하였다. |
| Base64 Encoder/Decoder APK - Windows 용 다운로드 - 최신 버전 2.2 | React-file-base64는 파일을 base64로 변환해주는 react component이다. 신뢰할 수 없는 통신 채널을 통해 binary 데이터를 안전하게 전송할 수 있다. |
| Sass(SCSS) 완전 정복! | HEROPY | Sass는 CSS를 확장한 스타일 시트 언어로 CSS로 컴파일된다. 변수, 중첩 규칙, 믹스인, 함수 등을 사용할 수 있어 스타일 시트를 잘 정돈하여 프로젝트에 적용할 수 있다. |

## 서버



서버는 블록체인 네트워크와 웹을 연결해주며 웹에서 사용할 API를 제공하는 역할을 수행한다. Node.js 기반으로 Express 프레임워크를 사용하여 구성하였으며 데이터베이스는 MySQL을 사용하였다.

기능은 크게 API 설명, input, user, class, contract로 나눌 수 있다. 먼저 API 설명은 서버 메인 화면에 접속하였을 때 API 설명을 표시하는 역할을 수행한다. 이를 위해 csv-parse, ejs, sass를 사용하였다. 다음으로 input 부분은 테이블 생성과 데이터 입력 기능을 담당한다. SQL을 직접 입력하지 않아도 자동으로 초기 작업을 수행할 수 있도록 하였고 테이블을 초기화할 수 있도록 구성하였다. user 부분은 사용자 로그인, 회원가입 등을 수행하는 부분으로 passport를 사용하여 session 기반으로 작동하도록 구현하였다. 또한 회원가입시 비밀번호 암호화를 위해 bcrypt를 사용하였다. class 부분은 강의, 강의 종류, 강의 내용, 강의 내용 종류, 강의 공지, 학습 과정, 학습 과정 상태, 수강 중인 강의 등 강의 관련 데이터베이스와 상호 작용하여 강의 관련 작업을 처리한다. 마지막으로 contract는 블록체인 네트워크와 상호 작용하여 스마트 컨트랙트를 이용한다. 이더리움 블록체인에 학습 과정과 수강 완료한 강의를 기록하고 조회하는 API를 제공한다. 이를 위해 truffle과 web3를 사용하였다.

|  |  |
| --- | --- |
| API 설명 | 서버 메인 화면에 API 설명 표시  csv-parse, ejs, sass |
| Input | 초기 테이블 생성, 데이터 자동 입력  mysql2, cookie-parser |
| User | 사용자 로그인, 회원가입, 로그아웃  passport, session, bcrypt, mysql2 |
| Class | 강의 관련 작업 처리  mysql2 |
| Contract | 블록체인 네트워크와 상호 작용  truffle, web3, mysql2 |

### 데이터베이스

* User 사용자

|  |
| --- |
| id VARCHAR(64) PRIMARY KEY, |
| pw VARCHAR(32) NOT NULL, |
| name VARCHAR(32) NOT NULL, |
| detail VARCHAR(256) NOT NULL, |
| phone VARCHAR(16) NOT NULL |

사용자 테이블에는 필드로 이메일, 비밀번호, 이름, 정보, 전화번호가 있다. 사용자 정보를 저장하여 로그인, 회원가입 등에 사용한다.

* ClassCategory 강의 카테고리

|  |
| --- |
| name VARCHAR(32) PRIMARY KEY |

강의의 카테고리 필드로 사용될 강의 카테고리 목록이다. 강의 테이블에서 enum으로 구현하면 새로운 카테고리가 추가될 때마다 데이터베이스를 직접 수정해야 하는 문제가 있다. 카테고리를 별도로 분리하여 추가, 제거를 용이하게 하였다. 강의 데이터에는 컴퓨터 과학, 프로그래밍, 글쓰기, 디자인, 투자 항목이 존재한다.

* Class 강의

|  |
| --- |
| id INT PRIMARY KEY AUTO\_INCREMENT, |
| name VARCHAR(128) NOT NULL, |
| detail VARCHAR(1024), |
| userId VARCHAR(64) NOT NULL, |
| category VARCHAR(32) NOT NULL, |
| image MEDIUMTEXT, |
| FOREIGN KEY (userId) REFERENCES user (id), |
| FOREIGN KEY (category) REFERENCES classCategory (name) |

강의 정보를 저장하는 테이블이다. 강의id, 강의명, 강의 정보, 강의자id, 카테고리, 대표 이미지를 필드로 가진다. 강의자id는 사용자 테이블의 id로 외래키로 지정하였고 카테고리는 앞서 언급한대로 강의 카테고리의 이름으로 외래키로 참조하였다.

* TakingClass 수강 신청한 강의

|  |
| --- |
| userId VARCHAR(64), |
| classId INT, |
| completedDate DATETIME, |
| isCompleted BOOLEAN DEFAULT FALSE, |
| isSaved BOOLEAN DEFAULT FALSE, |
| blockHash VARCHAR(66), |
| transactionHash VARCHAR(66), |
| PRIMARY KEY (userId, classId), |
| FOREIGN KEY (userId) REFERENCES user (id), |
| FOREIGN KEY (classId) REFERENCES class (id) |

사용자가 수강 중인 또는 완료한 강의를 저장하는 테이블이다. 사용자id와 강의id를 키로 하여 구분한다. 완료 날짜, 완료 여부, 블록체인 상에 저장되었는지 여부, 블록체인 상에 저장된 경우 블록 해시와 트랜잭션 해시 값을 저장한다.

* ContentType 강의 내용 종류

|  |
| --- |
| name VARCHAR(16) PRIMARY KEY |

강의 내용 종류를 저장하는 테이블이다. 강의 내용의 종류로 현재는 영상, 시험이 있다. 향후 다른 종류의 강의 내용이 등장한다면 편리하게 추가/제거 가능하다.

* Content 강의 내용

|  |
| --- |
| classId INT, |
| contentId INT, |
| type VARCHAR(16) NOT NULL, |
| title VARCHAR(128) NOT NULL, |
| url VARCHAR(128), |
| PRIMARY KEY (classId, contentId), |
| FOREIGN KEY (classId) REFERENCES class (id), |
| FOREIGN KEY (type) REFERENCES contentType (name) |

강의 차시별 내용을 저장하는 테이블이다. 강의id와 컨텐츠id를 키로 구분된다. 내용의 종류, 제목, 영상인 경우 url을 저장한다.

* Question 문제

|  |
| --- |
| classId INT, |
| contentId INT, |
| questionId INT, |
| title VARCHAR(128) NOT NULL, |
| answer VARCHAR(1024) NOT NULL, |
| PRIMARY KEY (classId, contentId, questionId), |
| FOREIGN KEY (classId, contentId) REFERENCES content (classId, contentId) |

강의 내용 중 문제를 저장하는 테이블이다. 강의id와 내용id, 문제id로 항목을 구분한다. 한 강의 내용(차시)에 여러 문제를 사용할 수 있도록 구성하였다.

* Test 시험

|  |
| --- |
| classId INT, |
| contentId INT, |
| questionId INT, |
| userId VARCHAR(64), |
| answer VARCHAR(1024) NOT NULL, |
| PRIMARY KEY (classId, contentId, questionId, userId), |
| FOREIGN KEY (classId, contentId, questionId) REFERENCES question (classId, contentId, questionId) |

문제에 대한 사용자가 입력한 답을 저장하는 테이블이다. 문제에서처럼 강의id와 내용id, 문제id로 구분되나 사용자가 입력한 답을 저장해야하므로 사용id까지 4개를 키로하여 각 레코드를 구분한다.

* ProcessState 학습 과정 상태

|  |
| --- |
| name VARCHAR(32) PRIMARY KEY |

학습 과정 상태를 저장하는 테이블이다. 현재 수강 필요, 수강 완료, 제출 필요, 제출 완료, 채점 완료 상태가 있다. 나중에 강의가 추가되고 강의 내용 종류가 달라짐에 따라 학습 과정에 여러 상태를 쉽게 추가/제거할 수 있도록 열거형을 사용하지 않고 학습 과정 상태 테이블을 별도로 분리하였다.

* Process 학습 과정

|  |
| --- |
| classId INT, |
| contentId INT, |
| userId VARCHAR(64), |
| date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP, |
| state VARCHAR(32) NOT NULL, |
| score INT NOT NULL, |
| feedback VARCHAR(1024), |
| isSaved BOOLEAN NOT NULL DEFAULT FALSE, |
| blockHash VARCHAR(66), |
| transactionHash VARCHAR(66), |
| PRIMARY KEY (classId, contentId, userId), |
| FOREIGN KEY (state) REFERENCES processState (name) |

사용자가 학습한 과정을 저장하는 테이블이다. 강의id와 내용id, 사용자id를 키로 구분한다. 학습일자와 상태, 점수, 피드백, 블록체인에 저장 여부, 블록체인에 저장된 경우 블록 해시와 트랜잭션 해시를 저장한다.

* Notice 공지사항

|  |
| --- |
| id INT PRIMARY KEY AUTO\_INCREMENT, |
| classId INT, |
| title VARCHAR(128) NOT NULL, |
| contents VARCHAR(1024) NOT NULL, |
| userId VARCHAR(64), |
| date DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP, |
| modificationDate DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP, |
| FOREIGN KEY (classId) REFERENCES class (id), |
| FOREIGN KEY (userId) REFERENCES user (id) |

강의별 공지사항을 저장하는 테이블이다. 강의id와 제목, 내용, 작성자id, 작성일, 수정일로 구성된다.

### API

#### User

* 로그인

|  |  |
| --- | --- |
| 방식 | POST |
| 주소 | /users/signin |
| 파라미터 | id, pw |
| 반환값 | {user, result: Boolean} |

* 로그아웃

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /users/signout |
| **파라미터** |  |
| **반환값** | {result: Boolean} |

* 회원가입

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /users/signup |
| **파라미터** | id, pw, name, detail, phone |
| **반환값** | {result: Boolean} |

* 사용자 이름

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /users/name/:userId |
| **파라미터** |  |
| **반환값** | {result: Boolean, name: String} |

#### Class

* 강의 정보

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/info/:classId |
| **파라미터** |  |
| **반환값** | [{class, result: Boolean}, ...] |

* 강의 목록(메인)

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/main |
| **파라미터** |  |
| **반환값** | [{class, result: Boolean}, ...] |

* 강의 목록(검색 페이지, 16개)

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/all |
| **파라미터** |  |
| **반환값** | [{class, result: Boolean}, ...] |

* 강의 종류(카테고리) 목록

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/category/list |
| **파라미터** |  |
| **반환값** | [{name: String, result: Boolean}, classCategory...] |

* 강의 종류별 강의 목록

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/category/:classCategory |
| **파라미터** |  |
| **반환값** | [{class, result: Boolean}, ...] |

* 강의 조회 (검색)

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/search/:query |
| **파라미터** |  |
| **반환값** | [{class, result: Boolean}, ...] |

* 수강 신청

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/enrol/:classId |
| **파라미터** | /login required/ |
| **반환값** | {result: Boolean} |

* 수강 중인 내 강의

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/my |
| **파라미터** | /login required/ |
| **반환값** | [{class, result: Boolean}, ...] |

* 내가 만든 강의

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/my/teach |
| **파라미터** | /login required/ |
| **반환값** |  |

* 새 강의 생성

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/create |
| **파라미터** | name, detail, category, image(String), /login required/ |
| **반환값** | {result: Boolean, classId: Int} |

* 기존 강의 정보 변경

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/update/:classId |
| **파라미터** | name, detail, category, image(String), /login required/ |
| **반환값** | {result: Boolean} |

* 강의 삭제

|  |  |
| --- | --- |
| **방식** | DELETE |
| **주소** | /class/delete/:classId |
| **파라미터** | /login required/ |
| **반환값** | {result: Boolean} |

* 수강 신청 전인 강의인지 확인

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/isBefore/:classId |
| **파라미터** | /login required/ |
| **반환값** | {result: Boolean} |

* 학습 과정 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/process/:userId/:classId |
| **파라미터** |  |
| **반환값** | [{process, result: Boolean}, ...] |

* 강의별 공지사항

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/notice/class/:classId |
| **파라미터** |  |
| **반환값** | [{notice, result: Boolean}, ...] |

* 강의 공지 작성

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/notice/create |
| **파라미터** | classId, title, contents, /login required/ |
| **반환값** | {result: Boolean} |

* 강의 공지 수정

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/notice/update/:id |
| **파라미터** | title, contents, /login required/ |
| **반환값** | {result: Boolean} |

* 강의 공지 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/notice/read/:id |
| **파라미터** |  |
| **반환값** | {notice} |

* 강의 공지 삭제

|  |  |
| --- | --- |
| **방식** | DELETE |
| **주소** | /class/notice/delete/:id |
| **파라미터** | /login required/ |
| **반환값** | {result: Boolean} |

* 강의 내용(컨텐츠) 추가

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/contents/:classId/add |
| **파라미터** | [{contentId, type, title, url}, ...], /login required/ |
| **반환값** | {result: Boolean} |

* 강의 내용 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/contents/:classId/:contentId |
| **파라미터** |  |
| **반환값** | {content, result: Boolean} |

* 강의 내용 목록

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/contents/:classId |
| **파라미터** |  |
| **반환값** | [{content}, ...] |

* 강의 영상 수강 완료

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/done/video |
| **파라미터** | classId, contentId, /login required/ |
| **반환값** | {result: Boolean} |

* 강의 시험 제출

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/done/test |
| **파라미터** | classId, contentId, answer[], /login required/ |
| **반환값** | {result: Boolean} |

* 강의 시험 채점

|  |  |
| --- | --- |
| **방식** | POST |
| **주소** | /class/done/test/score |
| **파라미터** | classId, contentId, score, feedback, /login required/ |
| **반환값** | {result: Boolean} |

* 완료된 강의 리스트 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/complete/list/:userId |
| **파라미터** |  |
| **반환값** | [{class, result: Boolean}, ...] |

* 완료된 강의인지 확인

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /class/complete/:userId/:classId |
| **파라미터** |  |
| **반환값** | {isCompleted: Boolean, result: Boolean} |

#### Contract

* 블록체인에 저장된 모든 학습 과정 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /contract/process/all |
| **파라미터** |  |
| **반환값** | [{classId, contentId, userId, date, state, score, feedback}, ...] |

* 블록체인에 저장된 학습 과정 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /contract/process/read/:classId/:contentId/:userId |
| **파라미터** |  |
| **반환값** | {classId, contentId, userId, date, state, score, feedback} |

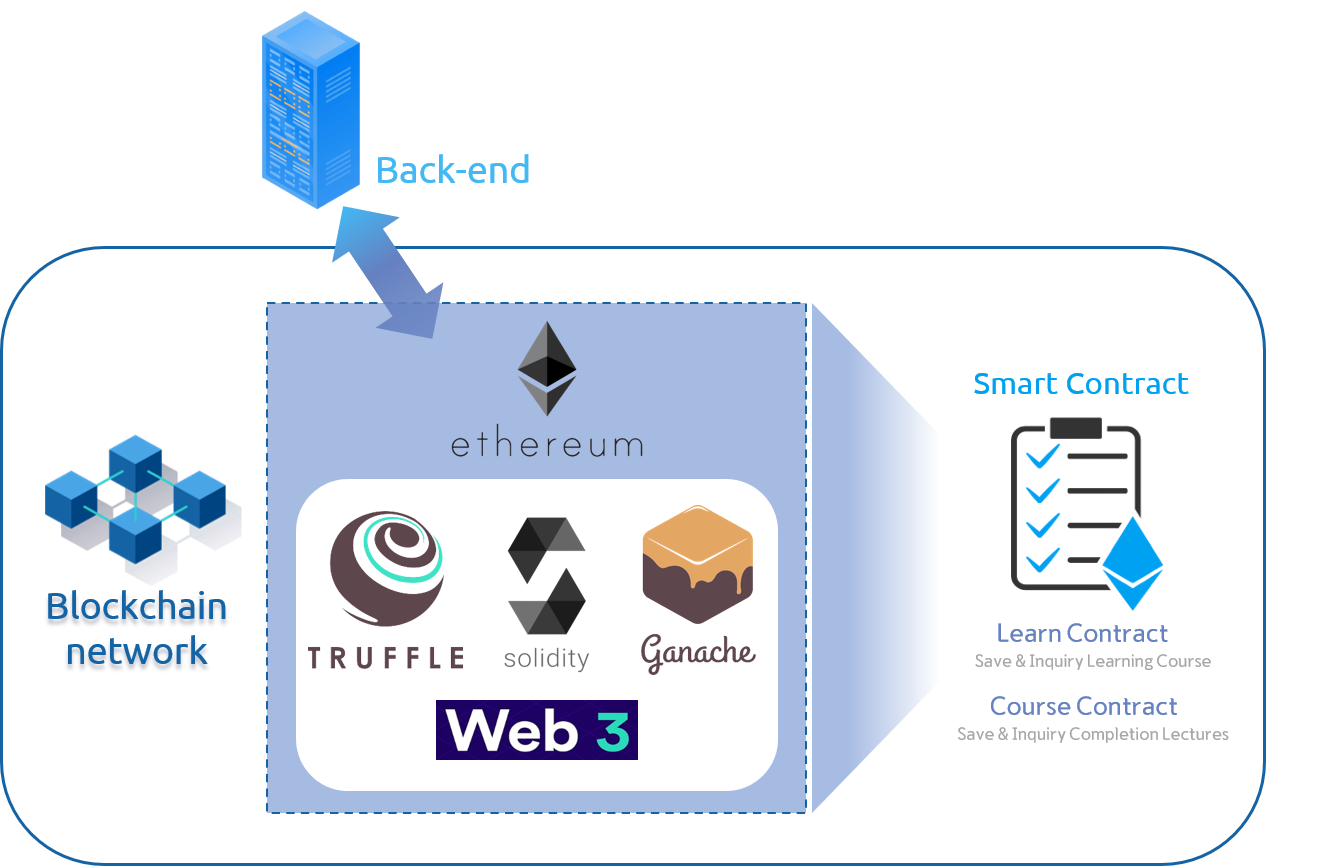
* 블록체인에 저장된 모든 수료 강의 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /contract/class/all |
| **파라미터** |  |
| **반환값** | [{userId, classId, completedDate, name, detail, teacher, category}, ...] |

* 블록체인에 저장된 수료 강의 조회

|  |  |
| --- | --- |
| **방식** | GET |
| **주소** | /contract/class/read/:userId/:classId |
| **파라미터** |  |
| **반환값** | {userId, classId, completedDate, name, detail, teacher, category} |

## 블록체인



블록체인은 이더리움 네트워크를 선택하였다. 이더리움에서는 분산형 어플리케이션(dApp)을 구동할 수 있다. solidity로 작성한 스마트 컨트랙트를 트러플로 컴파일 및 마이그레이션한다. 서버에서는 트러플과 web3로 이더리움 블록체인에 학습 과정과 수강 완료한 강의를 기록한다.

### 스마트 컨트랙트

|  |  |
| --- | --- |
| **스마트 컨트랙트** | |
| Learn 컨트랙트 | **학습 과정을 저장** |
| Course 컨트랙트 | **수료한 강의를 저장** |

스마트 컨트랙트는 학습 과정(process)을 저장하는 Learn 컨트랙트와 수료한 강의를 저장하는 Course 컨트랙트로 구성하였다.

#### Learn 컨트랙트

pragma solidity >=0.4.22 <0.9.0;

contract Learn {…}

먼저 학습 과정을 저장하는 Learn 컨트랙트를 설명한다.

struct process {  
 uint classId;  
 uint contentId;  
 string userId;  
 string date;  
 string state;  
 uint score;  
 string feedback;  
 }  
  
 struct processData {  
 string date;  
 string state;  
 uint score;  
 string feedback;  
 }

* 전체 학습 과정 목록을 저장하는데 사용되는 process 구조체와 classId, contentId, userId를 키로 구조체를 값으로 가지도록 할 때 사용되는 processData 구조체를 선언하였다.

process[] pcs;  
 mapping(bytes32 => processData) public processes;

* pcs는 process 구조체 배열로 전체 학습 목록을 저장하는데 사용된다. processes는 classId, contentId, userId를 이용해 processData 구조체에 접근할 수 있다. bytes32 자료형을 사용한 이유는 classId, contentId, userId를 합쳐서 SHA3 해시로 암호화하여 하나의 키로 사용하기 때문이다.

function save(  
 uint classId,  
 uint contentId,  
 string memory userId,  
 string memory date,  
 string memory state,  
 uint score,  
 string memory feedback) public returns (bytes32){  
 pcs.push(process(classId, contentId, userId, date, state, score, feedback));  
 bytes32 key = keccak256(abi.encode(classId, contentId, userId));  
 processes[key] = processData(date, state, score, feedback);  
 return key;  
 }

* save 함수는 classId, contentId, userId, date, state, score, feedback을 받아 pcs와 processes에 저장한다. string 변수의 memory 키워드는 해당 변수가 임시적으로 저장될 변수임을 의미한다. pcs는 배열이므로 push를 통해 저장할 학습 과정을 배열에 추가한다. processes는 매핑이므로 키와 값 형태로 저장한다. 이 때 키로 classid, contentId, userId 3가지 조합이 사용되므로 세가지를 SHA3 해시로 암호화하여 하나의 키로 사용한다.

function readAll() public view returns (process[] memory) {  
 return pcs;  
 }

* readAll 함수는 모든 학습 과정 목록을 반환하는 함수이다. 블록체인에 저장된 값을 변환하지 않고 읽기만 수행하므로 view 키워드를 추가하여 트랜잭션 없이 조회할 수 있도록 한다.

function read(  
 uint classId,  
 uint contentId,  
 string memory userId) public view returns (processData memory) {  
 bytes32 key = keccak256(abi.encode(classId, contentId, userId));  
 return processes[key];  
 }

* read 함수는 키 값인 classId, contentId, userId로 해당하는 학습 과정을 조회하는 함수이다. 세가지 키 값은 유일한 조합이므로 매핑에서 학습 과정을 바로 가져올 수 있다.

#### Course 컨트랙트

pragma solidity >=0.4.22 <0.9.0;

contract Course {…}

다음으로 학습 완료한 강의를 저장하는 Course 컨트랙트를 작성하였다.

struct class {  
 string userId;  
 uint classId;  
 string completedDate;  
 string name;  
 string detail;  
 string teacher;  
 string category;  
 }  
  
 struct classData {  
 string completedDate;  
 string name;  
 string detail;  
 string teacher;  
 string category;  
 }

* class는 전체 학습 완료한 강의 목록을 저장하는 용도로 사용될 구조체이다. classData는 userId와 classId를 키로 하여 매핑에 저장될 구조체이다.

class[] cls;  
mapping(bytes32 => classData) public classes;

* cls는 구조체 배열로 학습 완료한 강의 목록을 저장한다. classes는 userId와 classId를 키로 강의 정보를 저장할 매핑이다. 여기에서 bytes32를 사용한 이유는 userId와 classId를 합쳐서 SHA3 해시로 암호화하여 하나의 키로 사용하기 때문이다.

function readAll() public view returns (class[] memory) {  
 return cls;  
 }

* readAll 함수는 학습 완료한 강의 목록이 저장된 배열을 반환한다. 블록체인에 저장된 값을 변환하지 않고 읽기만 수행하므로 view 키워드를 추가하여 트랜잭션 없이 조회할 수 있도록 한다.

function readAll() public view returns (class[] memory) {  
 return cls;  
 }

* read 함수는 userId, classId를 키로 학습 완료한 강의 정보를 조회한다. userId와 classId는 유일한 키로 저장된 강의 정보를 매핑에서 바로 조회할 수 있다.

function save(  
 string memory userId,  
 uint classId,  
 string memory completedDate,  
 string memory name,  
 string memory detail,  
 string memory teacher,  
 string memory category) public returns (bytes32){  
 cls.push(class(userId, classId, completedDate, name, detail, teacher, category));  
 bytes32 key = keccak256(abi.encode(userId, classId));  
 classes[key] = classData(completedDate, name, detail, teacher, category);  
 return key;  
 }

* save 함수는 userId, classId, completedDate, name, detail, teacher, category 정보를 받아 cls와 classes에 저장한다. cls는 모든 완료한 강의 정보를 저장하는 배열이므로 push를 통해 class 구조체를 추가해주고 classes는 userId와 classId를 키로 하는 매핑이므로 SHA3로 해시로 암호화하여 이를 키로 하여 classData 구조체를 저장한다.

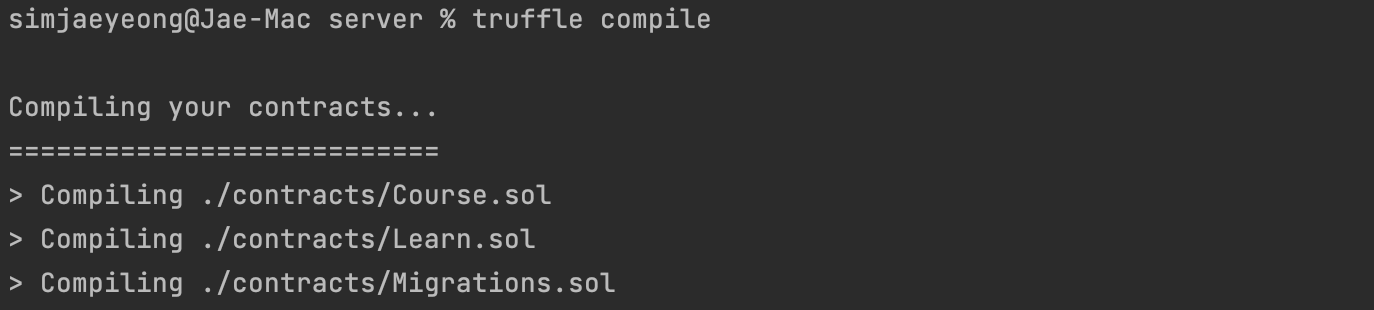
### 스마트 컨트랙트 컴파일 및 마이그레이션

트러플 디렉토리 구조는 contracts, migrations, test, truffle-config.js 파일로 구성된다. 앞에서 작성한 스마트 컨트랙트는 contracts 디렉토리에 저장하며 migrations에는 스마트 컨트랙트 배포를 위한 마이그레이션 시스템을 저장한다. 마이그레이션은 변경 사항을 추적할 수 있는 특수한 스마트 계약이다. test에는 스마트 컨트랙트에 대한 자바스크립트와 솔리디티 테스트 코드를 저장한다. truffle-config.js는 Truffle 설정 파일이다. 설정 파일에서 솔리디티 컴파일러 버전과 네트워크를 정의해야 한다.

module.exports = {networks: {  
 development: {  
 host: "127.0.0.1", // Localhost   
 port: 7545, // Standard Ethereum port   
 network\_id: "\*", // Any network   
 },  
 test: {  
 host: "127.0.0.1",  
 port: 7545,  
 network\_id: "\*"  
 },   
 },   
 compilers: {  
 solc: {  
 version: "0.8.13", }  
 }  
 },  
};

* 네트워크는 가나슈와의 연결을 위해 로컬 호스트, 7545번 포트를 사용하도록 하였다. 솔리디티 컴파일러는 0.8.13 버전을 사용하였다.

솔리디티는 컴파일이 필요한 언어이다. EVM(이더리움 가상 머신)에서 실행하기 위해 바이트 코드로 솔리디티 코드를 컴파일해야 한다. 아래 명령을 입력하여 작성한 솔리디티 코드를 컴파일할 수 있다.



스마트 컨트랙트 컴파일이 성공적으로 완료되었으므로 블록체인에 마이그레이션할 차례이다. 마이그레이션은 한 상태에서 다른 상태로 이동하여, 어플리케이션 컨트랙트의 상태를 변경하는 것을 의미하는 스크립트를 배포한다. 첫번째 마이그레이션에서는 새 코드를 배포할 수 있다. 그러나 이미 배포된 경우라면 다른 마이그레이션에서 데이터를 이동하거나 계약을 새로운 계약으로 교체할 수 있다.

migrations 디렉토리에는 1\_initial\_migration.js 파일이 있다. 이 파일은 다음 스마트 컨트랙트 마이그레이션을 관찰하기 위해 Migrations.sol 계약을 배포하고 향후 변경되지 않은 계약을 중복해서 마이그레이션하지 않도록 한다.

우리가 만든 계약을 배포하기 위해 마이그레이션 스크립트를 추가로 작성해야 한다. migration 디렉토리에 2\_deploy\_contracts.js 파일을 새로 만들었다.

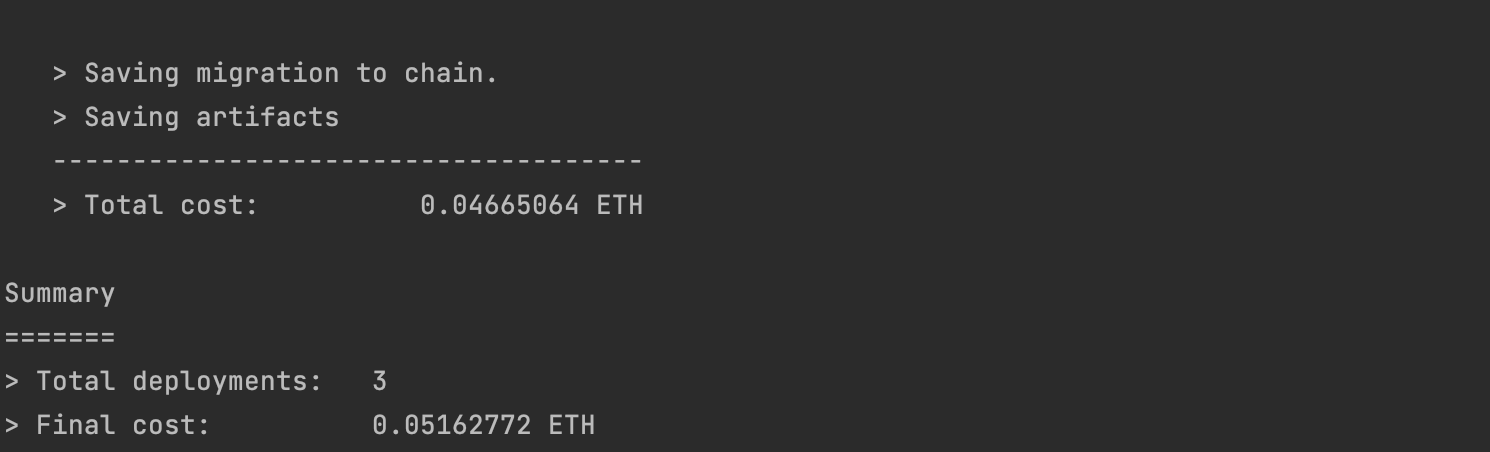
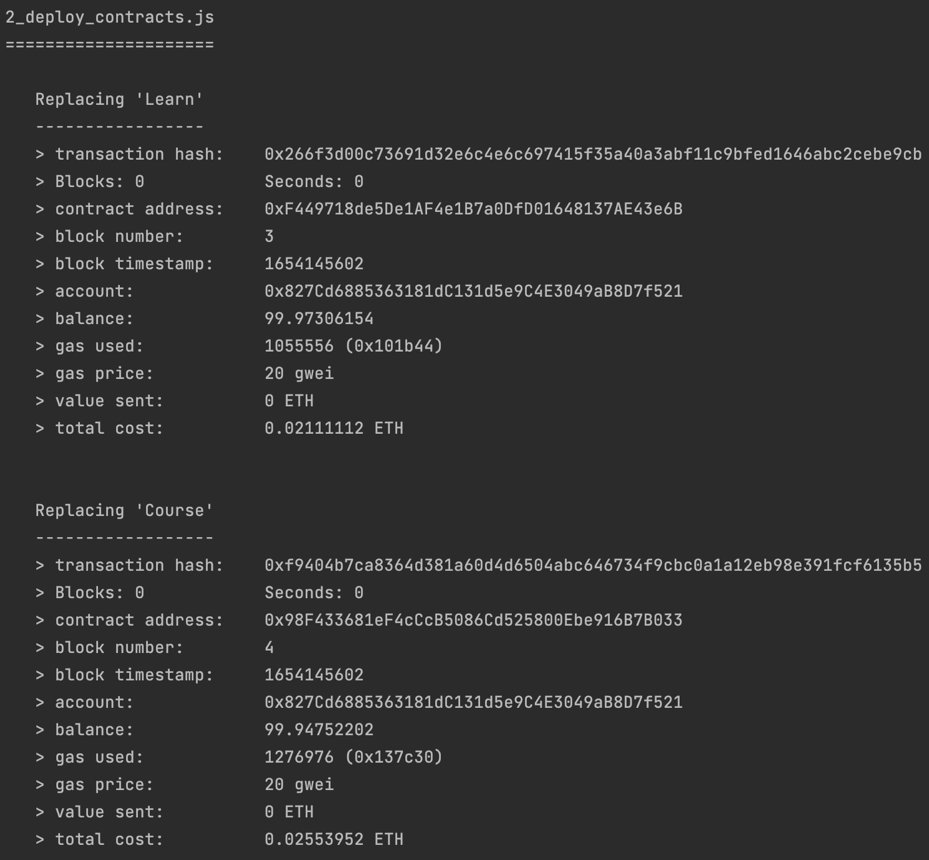
const Learn = artifacts.require("Learn");  
const Course = artifacts.require("Course");  
  
module.exports = (deployer) => {  
 deployer.deploy(Learn);  
 deployer.deploy(Course);  
};

스마트 컨트랙트를 블록체인에 마이그레이션하기 전에 블록체인을 실행해야 한다. 우리는 가나슈를 이용하여 이더리움 개발용 개인 블록체인을 만들고 컨트랙트 배포, 어플리케이션 개발, 테스트에 사용하였다.

아래 명령으로 블록체인에 마이그레이션할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명



테이블이(가) 표시된 사진

자동 생성된 설명마이그레이션이 순서대로 진행되고 관련 정보가 함께 출력되는 것을 확인할 수 있다. 가나슈에서도 블록체인 상태가 바뀐 것을 확인할 수 있다. 현재 블록이 5로 변경되고 0.05이더 정도가 감소한 것을 확인할 수 있다.

### 상호작용

Web3와 truffle을 이용하여 서버에서 블록체인 네트워크와 상호 작용할 수 있다.

getWeb3: () => {  
 const network = ***truffleConfig***.networks['development']  
 if (network.provider) { // 테스트넷  
 return new web3(network.provider)  
 } else { // 로컬, 가나슈  
 return new web3(new web3.*providers*.HttpProvider(`http://${network.host}:${network.port}`))  
 }  
 }

* truffle-config 파일에 설정된 내용을 바탕으로 테스트넷 또는 로컬 블록체인에 연결된다.

constructor() {  
 this.web3 = ***web3Lib***.getWeb3()  
 this.GAS\_LIMIT = 1000000  
 this.Learn = truffleContract(learnJson)  
 this.Learn.setProvider(this.web3.currentProvider)  
 }

* Learn class를 선언을 살펴보면 생성자에서 web3와 컨트랙트를 초기화한다.

async getAccounts() {  
 let accounts  
 try {  
 accounts = await this.web3.eth.getAccounts()  
 } catch (e) {  
 ***console***.error(e)  
 }  
  
 if (!accounts) {  
 ***console***.error("계정 불러오기 실패, 이더리움 클라이언트 확인")  
 }  
  
 return accounts  
 }

* web3 인스턴스를 가져오고 나면 web3.eth.getAccounts 메소드로 계좌를 불러올 수 있다.

const instance = await this.Learn.deployed()  
pcs = await instance.readAll.call()

또는

process = await instance.read.call(classId, contentId, userId)

* 먼저 getProcess나 getAllProcess 같이 읽기만 수행하는 함수에서는 instance에서 함수.call()로 해당 스마트 컨트랙트의 읽기 함수를 호출할 수 있다. readAll과 read는 앞에서 솔리디티로 작성된 Learn 컨트랙트의 readAll, read 함수를 각각 호출한다. call()을 사용하면 풀 트랜잭션을 보낼 필요 없이 블록체인에서 데이터를 읽을 수 있다. 다시 말해 이더를 소모하지 않는다.

const instance = await this.Learn.deployed()  
const accounts = await this.getAccounts()  
result = await instance.save(classId, contentId, userId, stringDate, state, score, feedback, {  
 from: accounts[0],  
 gasLimit: this.GAS\_LIMIT  
})

* setProcess 함수는 학습 과정을 블록체인 상에 저장하기 위해 Learn 컨트랙트의 save 함수를 호출한다. save 함수는 Learn 컨트랙트의 스토리지(pcs 배열과 processes 맵핑)를 변경한다. 이번에는 call 대신 transaction을 전송한다. 트랜잭션은 from 주소를 요구하며 비용이 발생한다. 이더에서 사용되는 비용은 가스라고 불린다. 가스 비용은 스마트 컨트랙트에서 계산을 수행하거나 데이터를 저장할때 발생한다. 학습 과정 데이터와 계정 주소를 포함하는 객체로 save함수를 실행하면서 트랜잭션을 전송한다.

const instance = await this.Course.deployed()

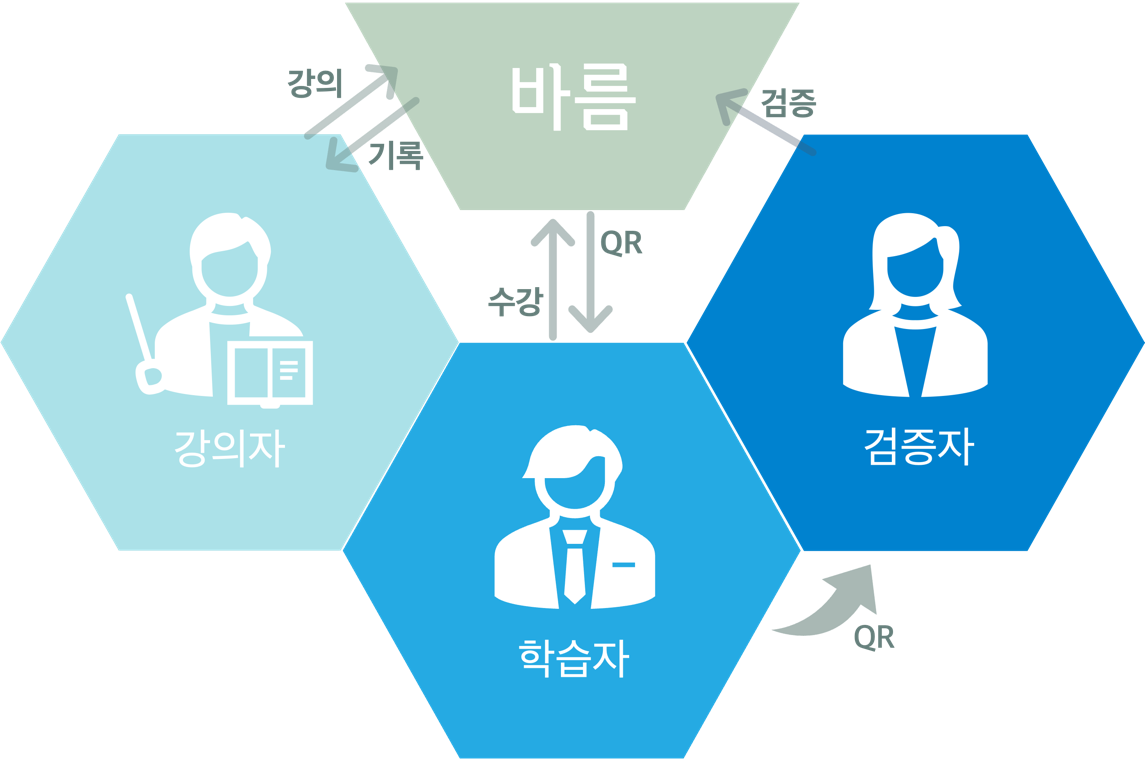
cls = await instance.readAll.call()

class1 = await instance.read.call(userId, classId)  
const accounts = await this.getAccounts()  
result = await instance.save(userId, classId, stringDate, name, detail, teacher, category, {  
 from: accounts[0],  
 gasLimit: this.GAS\_LIMIT  
})

* 이는 Course 클래스에도 유사하게 적용된다. Learn이 학습 과정을 저장한다면 Course는 수강 완료한 강의를 저장하기 때문에 read 함수의 읽기 처리 후 작업과 save 함수 호출에 사용되는 인자만 바뀔 뿐이다.

## 서비스 사용 과정

### 주요 기능 플로우



주요 기능은 크게 2가지로 구성되어 있다.

1. **QR코드로 증명서 조회**

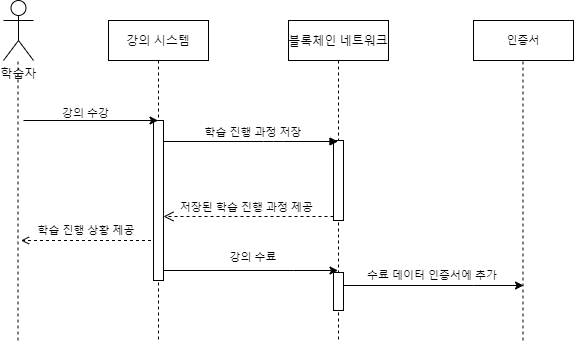
학습자는 강의 증명서의 재발급이나 누락될 여부 없이 QR코드로 수강한 강의를 증명할 수 있다. 강의자는 별도 시스템 구축 없이 강의 수강에 대한 인증을 제공할 수 있다. 검증자는 제출받은 QR코드로 지원자가 수강한 강의를 한꺼번에 조회하고 검증할 수 있다.

1. **학습 과정 블록체인에 저장**

모든 학습 과정은 블록체인에 자동으로 저장되기 때문에 학습자는 별도의 절차 없이 손쉽게 학습 이력을 관리할 수 있다. 검증자는 제출받은 인증서로 위변조의 걱정 없이 학습자가 배운 과정과 수준을 정확히 파악하고 신뢰할 수 있다.

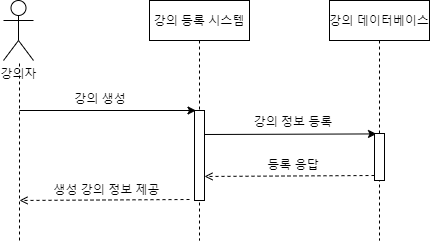
### 서비스 사용 시나리오

* 학습자



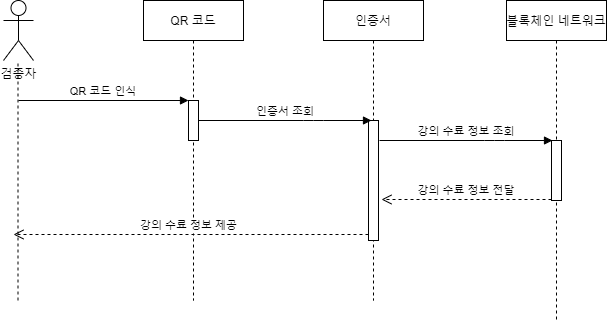
학습자는 바름 시스템을 통해 원하는 강의를 수강할 수 있다. 사용자가 강의를 수강하면 시스템은 자동으로 블록체인에 학습 진행 과정을 저장한다. 저장된 학습 진행 과정이 모두 수업 과정을 수료한다면 블록체인 네트워크에 수업 수료 내용이 기록된다. 학습자는 수강 중 또는 완료 시 학습한 과정을 볼 수 있는 수강 인증서가 자동으로 발급된다. 수강 인증서는 QR 코드 하나로 대체될 수 있으며 QR 코드를 인식하면 인증서 페이지로 연결된다. 따라서 강의 증명서 재발급, 누락 없이 QR 코드 하나만 검증자에 제출할 수 있다. 인증서 페이지에서는 수강한 강의를 QR 코드로 한 번에 인증할 수 있다. 사용자가 수강 완료한 강의가 한꺼번에 표시되며 각 강의 별로 별도의 QR 코드가 존재하여 강의 별로 학습 진도(진행 과정)와 상세 내용을 확인할 수 있다. 학습자는 손쉽게 학습 이력을 관리하고 투명한 평가 절차를 보장받으며 여러 강의를 비교해 볼 수 있다.

* 강의자



누구나 강의를 진행할 수 있다. 별도 시스템 구축 없이 강의 수강에 대한 인증을 바름 서비스가 제공한다. 강의자는 양질의 학습 컨텐츠 제작에만 집중할 수 있다. 또한 강의자는 모든 학습 과정과 평가 정보가 투명하게 공개되므로 학습자에게 정보가 누적되어 강의력을 인정받을 수 있다.

* 검증자



QR 코드 하나로 학습자가 수강한 모든 강의를 한 번에 확인, 검증할 수 있다. 각 강의별로 학습 진행 과정과 사용자의 점수, 피드백을 확인할 수 있으므로 강의 과정과 학습자의 수준을 정확하게 파악할 수 있다. 모든 학습 진행 과정이 블록체인에 저장되므로 검증자는 학습 데이터의 위변조 걱정 없이 안심하고 신뢰할 수 있다.

# 연구 결과 분석 및 평가

## 설계 변경 내역

먼저 블록체인에 기록할 정보를 축약하였다. 모든 정보를 블록체인에 저장한다면 많은 수수료 비용이 발생하고 트랜잭션 처리에 따른 문제가 발생할 수 있다. 가장 중요한 학습 데이터 중 완전히 학습을 완료하여 더 이상 데이터 변경이 발생하지 않는 데이터만 블록체인에 기록하도록 설계하였다.

한 번 업로드한 강의 내용을 임의로 삭제할 수 없도록 변경하였다. 강의 내용을 강의자가 삭제해버리면 기존에 인증된 사용자 데이터와 충돌이 발생하게 된다. 다만 강의 내용을 업데이트 할 수 있도록 하여 내용을 보강할 수 있도록 하였다. 변경 내역 적용을 위해 관련 API를 삭제하고 웹에서 강의 내용을 삭제하는 것이 불가능하도록 변경하였다.

서버의 경우 기존 기능 요구사항에서 중복된 항목에 대한 통합을 진행하였다. 테이블 내부에서 칼럼이 중복되지 않도록 하기 위해 강의 사용자 id와 이메일을 통합하는 등 중복 항목을 삭제하였다. 기능 요구사항에 따라 일부 누락된 테이블과 칼럼을 추가하였다. API와 직관적으로 연동되도록 설계를 변경하였다. 예를 들어 강의 내용 테이블은 클래스ID와 컨텐츠ID를 키로 하여 API 호출시 클래스ID와 컨텐츠ID를 이용한 별도의 테이블 탐색이나 계산없이 바로 해당 레코드에 접근 가능하다.

웹에서는 보라색과 초록색 테마를 적용하였다. 보라색은 믿음을 주는 색상으로 사용자에 신뢰감을 주는 효과를 주며 초록색은 편안한 색으로 사용자에 안락함을 준다. 블록체인으로 쉽고 편안하게 수강한 강의를 인증할 수 있다는 점을 테마 색에 담았다.

수강 인증서의 레이아웃을 변경하였다. 수강 인증서의 QR코드를 크게 교체하였다. 기존 수강 인증서에는 QR 코드가 작게 나와서 시인성이 낮았었는데 QR 코드 크기를 크게 하여 쉽게 찾을 수 있도록 하였다. 또 강의 요약과 상세 내용을 볼 수 있도록 배치하였다. 강의 요약에는 수강한 순서대로 강의 내용의 제목, 종류, 점수, 피드백, 완료 날짜가 블록 형태로 표시되고 상세 내용에서 블록 해시와 트랜잭션 해시를 확인할 수 있도록 하여 신뢰감을 높였다.

## 연구 결과 분석

바름 서비스는 크게 웹, 서버, 블록체인으로 구분할 수 있다. 웹은 사용자와 직접 상호작용하며 서버와 API를 통해 통신한다. 리액트 프레임워크인 Next.js를 이용하여 웹을 구성하였다. 서버는 웹에서 받은 사용자 요청을 처리하고 데이터베이스에 기록한다. 또한 블록체인 네트워크와 상호작용하기 위한 코드도 포함하고 있어 블록체인과 웹을 연결해주는 기능도 수행한다. Node.js와 express를 이용하여 서버를 제작하였으며 데이터베이스는 MySQL을 사용하였다. 블록체인은 이더리움 네트워크를 선택하였으며 solidity로 작성한 스마트 컨트랙트를 트러플로 컴파일 및 마이그레이션한다. 서버에서는 트러플과 web3로 이더리움 블록체인에 학습 과정과 수강 완료한 강의를 기록한다.

텍스트, iPod, 주차장, 전자기기이(가) 표시된 사진

자동 생성된 설명

* 메인 페이지

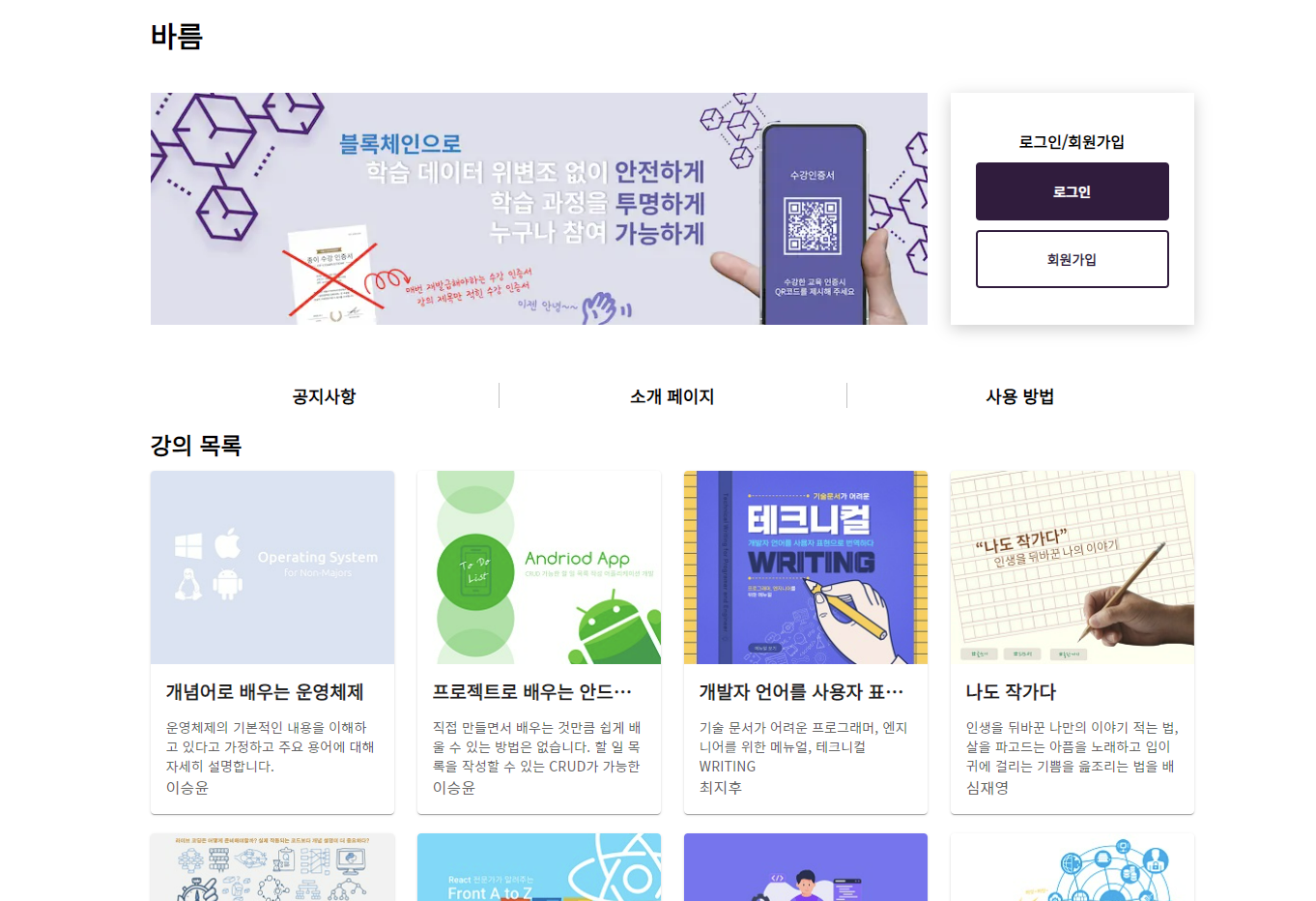


그림 1. 메인

* 로그인/회원가입 페이지

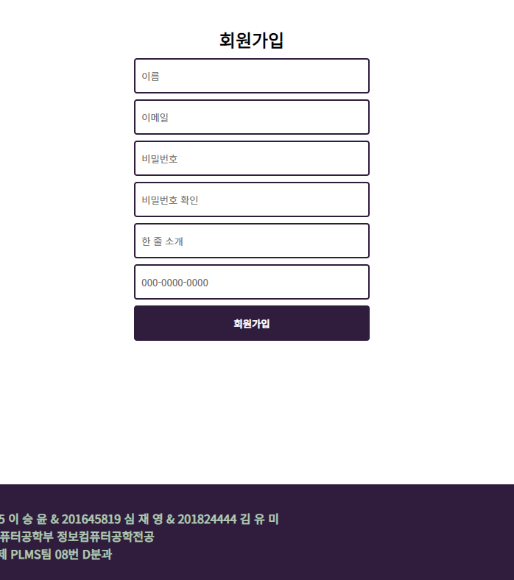


그림 2. 로그인/회원가입

* 강의 수강 페이지

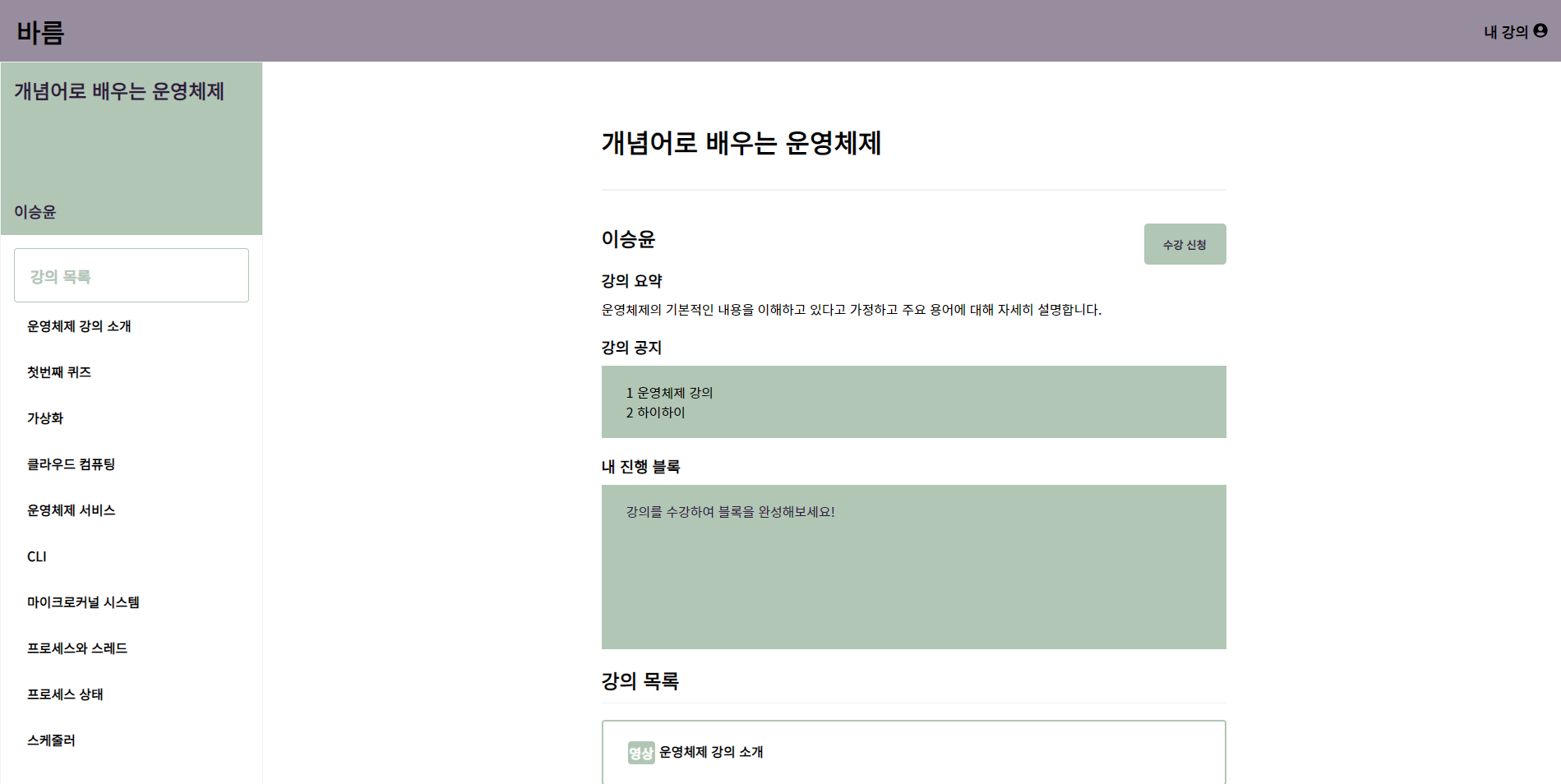


그림 3. 강의 정보(수강 전)

아직 수강하지 않은 강의는 수강 신청 버튼을 누른 후 수강할 수 있다.



그림 4. 강의 정보(수강 시)

내 진행 블록에서 현재 수강한 강의의 진행 상태를 볼 수 있다. 컨텐츠 명, 종류, 점수, 수강 날짜를 확인할 수 있고 블록에 마우스를 올리면 구체적인 평가 내역도 확인할 수 있다.

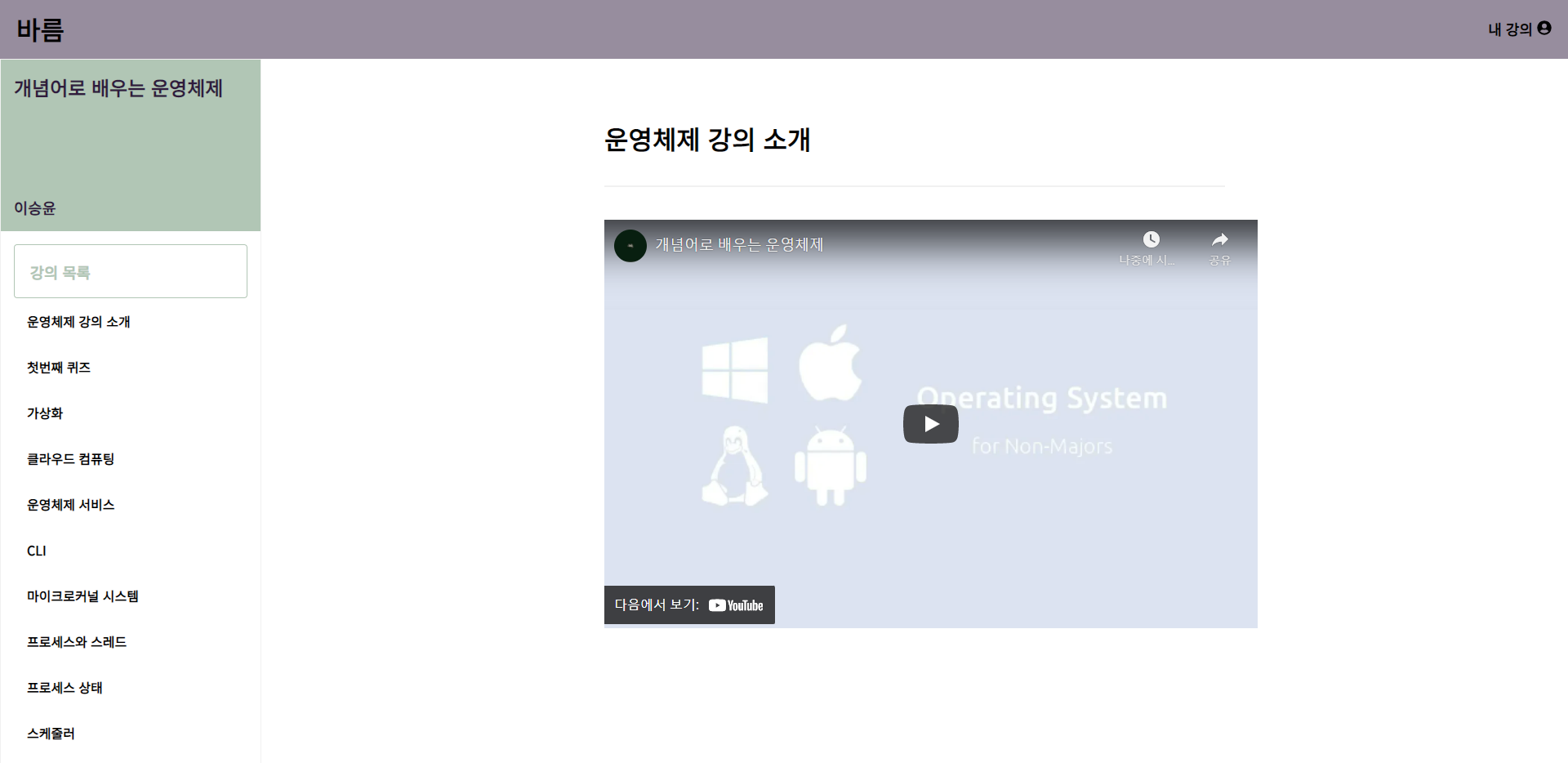


그림 5. 강의 컨텐츠

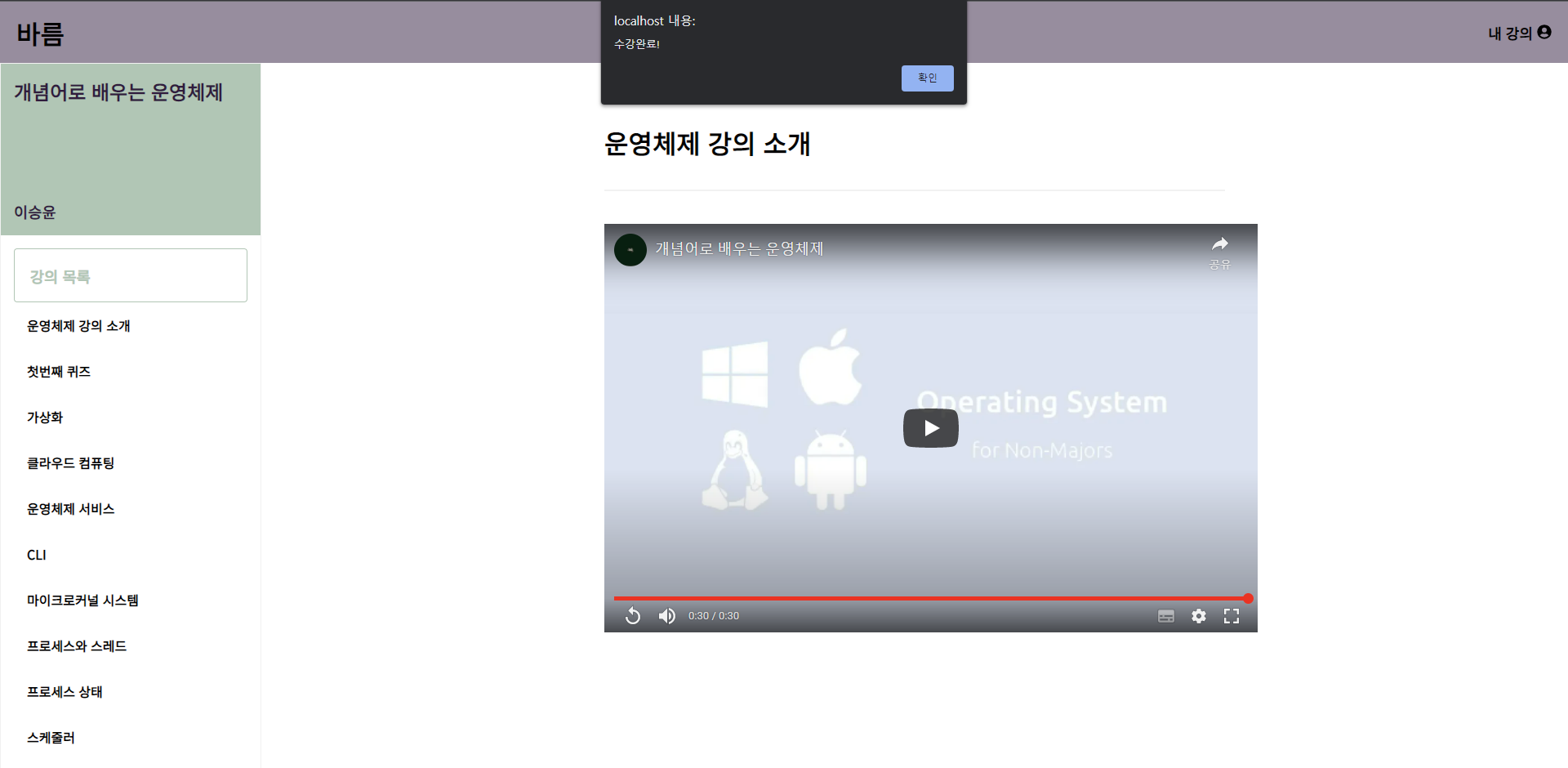


그림 6. 강의 컨텐츠(수강 완료시)

강의 컨텐츠를 끝까지 수강하면 수강 완료 알림이 뜬다. 이 알림이 뜬 후 블록을 확인해보면 수강 상태가 변경된 것을 확인할 수 있다.

* 강의 공지 페이지

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 7. 공지사항

공지사항 읽기는 수강생이라면 누구나 가능하지만 수정, 삭제, 작성은 강의자만이 할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 8. 공지사항 수정, 작성

* 마이 페이지



그림 9. 마이페이지 – 수강 중



그림 10. 마이페이지- 수강 완료

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 11. 마이페이지 – 생성한 강의

마이페이지에서 수강 중, 수강 완료, 생성 강의 탭을 통해 사용자의 강의를 볼 수 있다. 수강 완료 탭의 경우 제일 상단의 QR코드를 통해 전체 강의 목록 인증서(그림 16)로 이동할 수 있다. 각 개별 강의 목록에 있는 QR코드를 통해 개별 강의 인증서(그림 15)로 이동할 수 있다. 생성 강의 상단의 강의 만들기 버튼을 통해 강의를 생성할 수 있고, 강의 수정 버튼을 통해 강의 정보를 변경, 삭제 버튼을 통해 강의를 삭제할 수 있다.

* 강의 생성 페이지

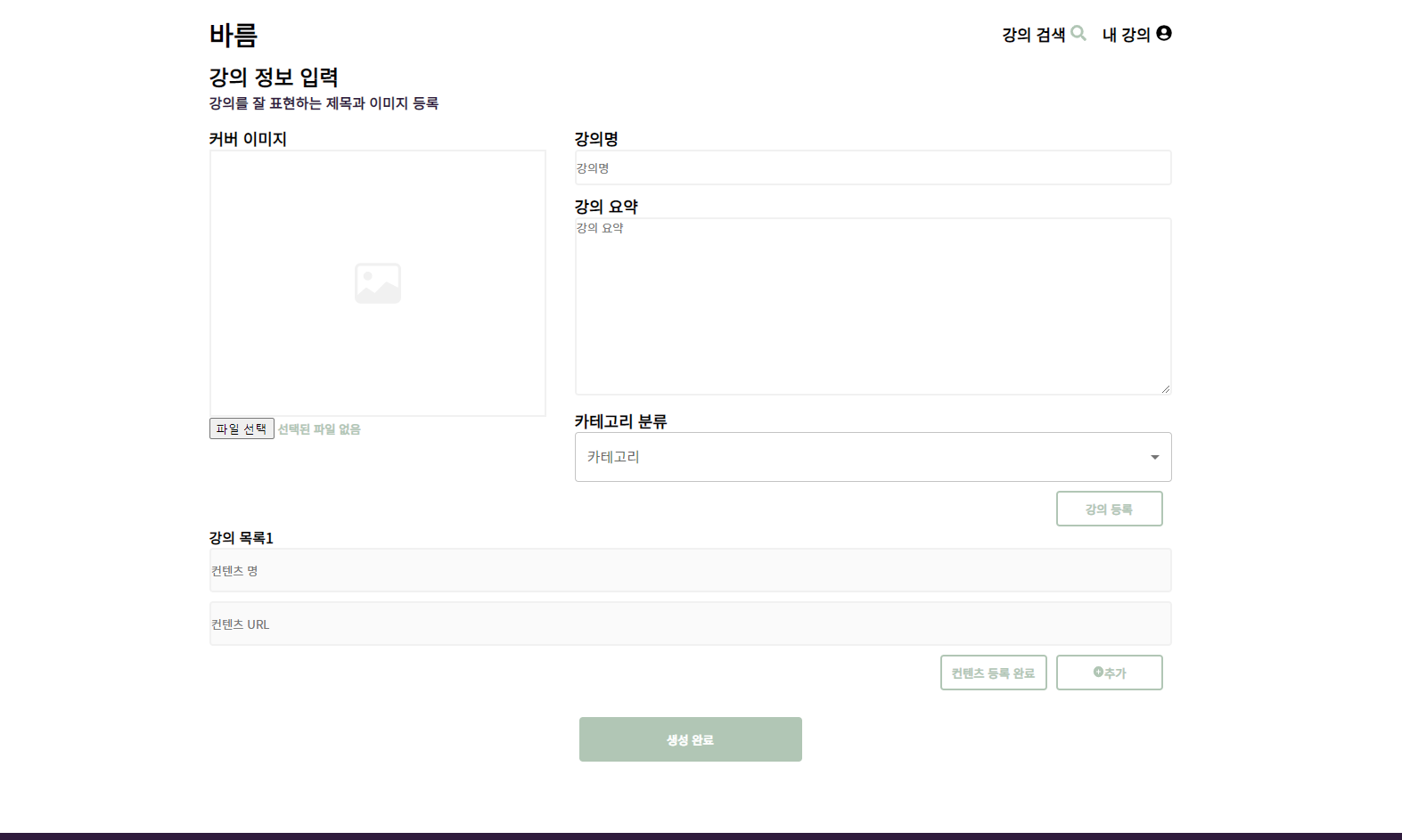


그림 12. 강의 생성

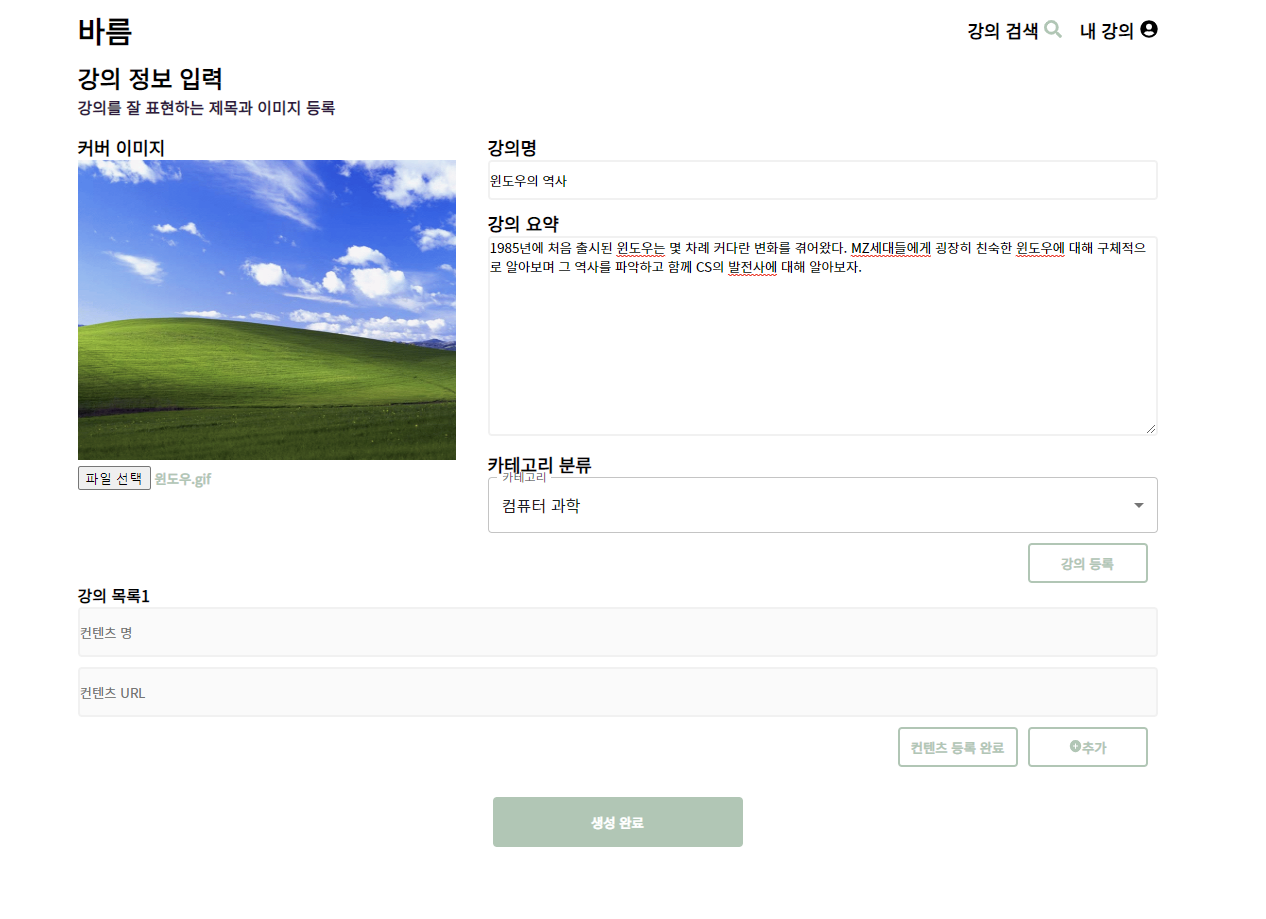


그림 13. 강의 생성 – 강의 입력 시

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 14. 강의 생성 – 컨텐츠 입력

강의를 생성한 후 강의 컨텐츠를 입력한다. 추가 버튼을 눌러 원하는 만큼 강의 컨텐츠를 추가할 수 있다.

* 강의 인증서 페이지

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 15. 개별 강의 인증서

마이페이지의 수강 완료한 강의의 QR코드(그림 10)를 이용하여 개별 강의 인증서를 볼 수 있다. 진행 상태를 수강한 순서대로 배치된 블록을 통해 쉽게 확인할 수 있고, 상세 내용을 통해 해당 컨텐츠의 블록 해시와 트랜잭션 해시 값을 확인할 수 있다. 이를 통해 위변조 위험을 방지할 수 있다.



그림 16. 강의 목록 인증서

마이페이지의 수강 완료 강의의 최상단 QR코드(그림 10)를 이용하여 강의 목록 인증서를 볼 수 있다. 지금까지 사용자가 수강 완료한 강의 목록을 확인할 수 있고, 강의의 QR코드를 통해 개별 강의 인증서(그림 15)를 확인할 수 있다.

* 강의 검색 페이지



그림 7. 강의 검색

강의 검색 페이지에서 강좌명, 교수자명, 카테고리 등을 검색하여 원하는 강좌를 찾고 수강신청을 할 수 있다.

## 평가

4차 산업혁명의 핵심기술로 주목받고 있는 블록체인 기술은 탈중앙화, 투명성, 공유성의 특성이 있다. 신뢰 기반으로 상호작용이 가능하며 조작할 수 없는 디지털 프로세스를 제공하여 블록체인이 교육 분야를 변화시킬 것으로 예측된다. [21] 블록체인 기술을 활용한 플랫폼에서는 중개인의 개입없이 개인간 직접 상호작용할 수 있다.

블록체인 네트워크에 기록된 내용을 바탕으로 기존 교육 기관, 업체와 무관하게 학습 증명이 가능하다. 모든 학습 과정이 블록에 기록되어 있으므로 위변조의 위험 없이 회사 또는 기관 제출용으로 사용 가능하다. 현재에는 인증서의 진위 여부를 판단하기 위해 해당 교육 기관 및 업체에 직접 확인해야 하며 변경사항 반영, 위변조 위험 때문에 매번 재발급해야 하는 문제가 있다. 이런 방식에서 벗어나 인증서를 요청하는 기관에 블록체인 기반의 인정서를 제출하면 신청 받은 기관은 즉시 사실 여부를 검증할 수 있어 종이 인증서 발급으로 인한 시간적, 경제적 비용을 크게 감소시킬 수 있다. [22] 타 플랫폼과 연동하여 다른 플랫폼에서 수강하는 학습 내역을 네트워크에 저장한다면 통합된 인증서 플랫폼으로 확장이 가능하다.

학습 진행 과정을 알 수 있어 투명하고 신뢰할 수 있는 평가 시스템을 제공한다. 온라인 기업 교육이 크게 증가하는 상황에서 학습자의 학습 과정과 평가 내용을 확인하여 채용, 승진의 참고 자료로 활용할 수 있다. 학습자는 진행 과정을 보며 강의자의 평가가 적절한지 확인하여 교육자와 학습자 간에 상호소통을 원활하게 한다. 투명하고 신뢰할 수 있는 평가 시스템은 학습자에게 동기를 부여하고 다른 수강자의 학습 내역을 확인하여 학습 기간과 강의의 질을 예측할 수 있어 더욱 체계적인 학습을 할 수 있을 것으로 기대된다.

탈중앙화된 학습 증명과 투명하고 신뢰할 수 있는 평가 시스템을 통해 누구나 질 좋은 강의를 만들고 수강할 수 있다. 부실한 업체가 짧은 기간 돈만 받고 자격증, 인증서를 발급하는 경우, 매우 초보적인 내용으로만 강의를 구성하여 성적을 우수하게 보여 교육 효과가 큰 것처럼 광고하는 경우를 쉽지 않게 찾아볼 수 있다. [7] 학습자는 수강 전 다른 학습자의 과정과 평가를 미리 찾아볼 수 있어 낮은 수준의 강의 선택을 회피할 수 있다. 회사와 기관에서는 명칭만 전문가, 심화 과정인지 실제 강의가 어떤 수준이고 학습자의 실력은 어느 정도인지 면밀히 파악할 수 있는 효과가 있다. 단순히 수료 인증서만 발급할 때보다 질 높은 강의의 힘이 커지고 더 좋은 강의가 나올 수 있는 원동력으로 작용한다.

# 결론 및 향후 연구 방향

## 결론

따로, 매번, 다시 발급해야 하는 기존 수강 인증서. 기존의 수강 인증서는 수강 강의 증빙을 위해 강의마다 반복해서 인증서 발급이 필요하다. 강의명만 나와 있어 무엇을 배웠는지 과정과 수준을 정확하게 파악하기 힘들다. 수강 인증서 발행 기관을 신뢰할 수 없는 문제가 있다. 인증서를 매번 새로 발급하고 위변조 여부를 확인해야 하는 과정을 보다 안전하게 간단히 처리하고, 학습 과정을 확인하고 증명할 수 있도록 하여 평가의 공정성 문제를 해결하며, 누구나 강의를 만들 수 있고 인증서 발급과 학습 증명이 가능하도록 하는 기술. 우리는 블록체인에서 그 답을 찾았다.

우리 팀은 블록체인을 도입하여 QR코드로 한 번에 수강한 교육 과정을 증명할 수 있도록 서비스를 개발하였다. 간단하게 과정을 나타내면 강의 수강 > 학습 과정 블록체인에 저장 > 증명서 자동 발급 > QR코드로 증명서 조회로 요약할 수 있다. 학습자는 강의 증명서 재발급, 누락 없이 QR 코드 하나만 제출하면 되며 검증자는 QR 코드 하나로 지원자가 수강한 강의를 한꺼번에 확인, 검증할 수 있다. 강의자는 누구나 강의할 수 있으며 별도 시스템 구축 없이 강의 수강에 대한 인증을 제공할 수 있다. 학습 과정이 블록체인에 저장되므로 학습자는 손쉽게 학습 이력을 관리하고 투명한 평가 절차를 경험할 수 있다. 여러 강의를 비교하기도 쉽다. 검증자는 위변조 걱정 없이 학습자가 배운 과정과 수준을 파악하고 신뢰할 수 있다. 강의자는 양질의 학습 컨텐츠 생산에 집중할 수 있으며 강의력을 인정받을 수 있다.

## 향후 연구 방향

이더리움 블록체인 네트워크에 정보를 기록하는 점은 비싸다. 현재 스마트 컨트랙트 배포에 0.05이더가 소모되며 데이터 기록과 기타 테스트 작업을 하고 나니 총 1.7이더 정도를 소모하였다. 1이더는 2022년 5월 13일 기준으로 274 만원 정도이다. 이를 이더리움 외 솔리디티 코드가 호환가능한 트론, 클레이튼 등 좀 더 저렴한 메인넷으로 옮긴다면 클레이튼의 경우 540원, 트론의 경우 95원으로 저렴하게 서비스가 가능할 것으로 예측된다. 또한 어떻게 하면 사용 가스량을 줄일 수 있는지 스마트 컨트랙트 코드를 좀 더 최적화하는 방향도 기대해 볼 수 있다.

강의에 NFT를 적용하여 강의 소유권을 거래하고 수강자에 따라 이득을 분배하는 모델도 생각해볼 수 있다. NFT를 통해 강의에 대한 소유권을 확립하고 거래할 수 있도록 진화할 수 있다. 강의를 거래할 때마다 강의자에게 수익이 돌아오도록 거래 조건을 설정할 수 있으며 수강자가 지불한 금액에 대해서도 조건을 지정할 수 있을 것이라 생각된다. 따라서 NFT를 통해 다양한 비즈니스 모델이 등장할 것으로 기대된다.

# 개발 일정 및 역할 분담

## 개발 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 업무 | 2월 | | | | 3월 | | | | 4월 | | | | | 5월 | | | | 6월 | | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 5 |
| 지도교수 상담 및 멘토 매칭 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 착수보고서 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 이더리움 학습 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 웹 설계 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 서버 설계 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 데이터베이스 설계 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 학습 컨텐츠 제작 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 중간보고서 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 웹 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 서버 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 스마트 컨트랙트 설계 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 기능/통합 테스트 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 스마트 컨트랙트 개발 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 추가 연구 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 최종보고서 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 최종 발표 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| SW 등록 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 역할 분담

|  |  |
| --- | --- |
| 이름 | 역할 |
| 이승윤 | 서버, 데이터베이스, API 설계  로컬 및 원격 서버 구축  데이터베이스 구축  API 제작  API 연동 및 테스트  이더리움 탈중앙화 분산 어플리케이션(dApp) 개발 |
| 심재영 | UI 디자인  학습 데이터 생성  학습 이미지 제작  포스터 제작 |
| 김유미 | UI 디자인  웹 레이아웃 구현  서버와 API 연동 및 테스트 |
| 공통 | 블록체인, 이더리움 학습/연구  웹 페이지 레이아웃 설계  개발 환경 설정  보고서 작성 |

# 산업체 멘토링 결과 반영

착수보고서에 대해서는 기존 학생의 학습 상황에 대한 포트폴리오는 제3자가 신뢰하지 않는 경향이 있는데 이더리움 블록체인을 적용하여 신뢰성을 줄 수 있으므로 현장에 적용 가능하다는 긍정적인 평가를 받았다. 단 일정 및 계획에서 설계, 구현, 검증 일정은 적합하나 기능, 통합 시험 일정을 추가할 것을 요청받았고 이에 따라 테스트 일정을 추가하였다. 실제 개발 과정에서 일정에 맞추어 테스트를 수행하여 웹과 서버 연동할 때 어려움을 줄일 수 있었다.

중간보고서에서는 계획 대비 개발이 약간 늦춰져 약간의 수정 또는 개선이 필요하다는 의견을 받았다. 개발 진행 사항에 맞추어 QR코드를 통한 인증 부분을 추가로 작성하여 제출하였다. 또한 보고서의 시스템 구성도를 좀 더 구체화했으면 좋겠다는 의견을 받아 사용한 기술을 좀 더 집약적으로 표시하여 구성도의 내용을 보강하였다.

멘토링 없이 그냥 진행하였다면 놓쳤을 수 있는 부분, 빈약한 부분을 점검 받을 수 있어서 도움이 되었다. 착수보고서에서는 처음에 개발 일정에 기능, 통합 시험 일정이 별도로 고려하지 않았는데 멘토링을 통해 해당 부분을 수정할 수 있었다. 테스트 일정을 별도로 두어 해당 일정에 맞추어 웹, 서버 개발을 병행하여 수행할 수 있었다. 우선 기능별로 테스트를 진행하여 기능이 제대로 구현되었는지 검증하였기 때문에 API를 통해 연동할 때도 수월하게 진행할 수 있었다. API 연동 이후 통합 테스트를 통해 사용자가 해당 기능을 온전히 사용할 수 있는지 확인할 수 있었다. 중간보고서에서는 개발 진척도가 느리다는 지적을 받았다. 개발 진행 과정에 맞추어 수강 중, 수강 완료, 수강 인증 페이지를 추가하여 보고서를 최신화하였다. 멘토링이 없었더라면 프로젝트가 기간에 맞게 잘 진행되고 있는지 점검하기 어려웠을 것이다. 멘토링을 통해 프로젝트 진행도를 확인받을 수 있었던 점이 좋았다.

참고 문헌

|  |  |
| --- | --- |
| [1] | 김용성, "블록체인, 교육을 바꾸다," 소프트웨어정책연구소, 22 3 2019. [Online]. Available: https://spri.kr/posts/view/22599. [Accessed 19 2 2022]. |
| [2] | Byline Network, "에듀테크에서 블록체인 사용 매뉴얼," Byline Network, 5 11 2018. [Online]. Available: https://byline.network/2018/11/5-30/. [Accessed 19 2 2022]. |
| [3] | 김준수, "[미래 성장동력, 블록체인] 탈중앙성·투명성·불변성 기술적 특징으로 비즈니스 패러다임 변혁할 기술로 부상," *부산발전포럼,* pp. 8-15, 10 2019. |
| [4] | 국립국어원, "국립국어원 표준국어대사전," [Online]. Available: https://stdict.korean.go.kr/search/searchResult.do?pageSize=10&searchKeyword=%EB%B0%94%EB%A5%B4%EB%8B%A4. [Accessed 20 2 2022]. |
| [5] | 해시넷, "블록체인 - 해시넷," 10 8 2021. [Online]. Available: http://wiki.hash.kr/index.php/%EB%B8%94%EB%A1%9D%EC%B2%B4%EC%9D%B8. [Accessed 21 2 2022]. |
| [6] | IBM, "블록체인 기술이란? - IBM 블록체인," IBM, [Online]. Available: https://www.ibm.com/kr-ko/topics/what-is-blockchain. [Accessed 21 2 2022]. |
| [7] | 이소현, "2주 수업받고 메타버스 전문가?…부실 자격증 '우후죽순'," 한국경제, 20 2 2022. [Online]. Available: https://www.hankyung.com/it/article/202202194675i. [Accessed 21 2 2022]. |
| [8] | ethereum.org, "이더리움이란 무엇인가요?," ethereum.org, 19 2 2022. [Online]. Available: https://ethereum.org/ko/what-is-ethereum/. [Accessed 21 2 2022]. |
| [9] | ethereum.org, "Introduction to smart contracts," ethereum.org, 19 2 2022. [Online]. Available: https://ethereum.org/en/smart-contracts/. [Accessed 22 2 2022]. |
| [10] | 조수현, 이정빈, 박재용, 이대건 and 인호, 이더리움 베이직, 북스타, 2017. |
| [11] | ethereum.org, "Ethereum Whitepaper," 11 5 2022. [Online]. Available: https://ethereum.org/en/whitepaper/. [Accessed 12 5 2022]. |
| [12] | Truffle Suite, "Truffle | Overview," 31 3 2022. [Online]. Available: https://trufflesuite.com/docs/truffle/. [Accessed 31 3 2022]. |
| [13] | 해시넷, "가나슈 - 해시넷," 1 9 2019. [Online]. Available: http://wiki.hash.kr/index.php/%EA%B0%80%EB%82%98%EC%8A%88. [Accessed 29 3 2022]. |
| [14] | DENSO WAVE INCORPORATED, "QR코드란?," DENSO WAVE INCORPORATED, [Online]. Available: https://www.qrcode.com/ko/about/. [Accessed 22 2 2022]. |
| [15] | OpenJS Foundation, "About | Node.js," OpenJS Foundation, [Online]. Available: https://nodejs.org/ko/about/. [Accessed 21 2 2022]. |
| [16] | StrongLoop, Inc, "Express - Node.js Web Application Framework," StrongLoop, Inc, [Online]. Available: https://expressjs.com/ko/. [Accessed 22 2 2022]. |
| [17] | Meta Platforms, Inc, "React - A JavaScript library for building user interfaces," Meta Platforms, Inc, 2022. [Online]. Available: https://reactjs.org/. [Accessed 22 2 2022]. |
| [18] | Vercel, Inc, "Next.js by Vercel - The React Framework," [Online]. Available: https://nextjs.org/. [Accessed 11 5 2022]. |
| [19] | Wikipedia, "Next.js - Wikipedia," 9 5 2022. [Online]. Available: https://en.wikipedia.org/wiki/Next.js. [Accessed 11 5 2022]. |
| [20] | M. Thakkar, Adding Server-Side Rendering to Your React Application, Apress, 2020. |
| [21] | 김승진 and 장환영, "기업교육에서의 블록체인 활용방안 연구," *기업교육과인재연구,* vol. 21, no. 4, pp. 43-64, 2019. |
| [22] | 이재우, "블록체인 기반 교육 분야 인증서 발급시스템에 관한 연구," in *동국대학교 석사학위논문*, 2022. |
| [23] | B. G. Jeppesen, "REALIZING THE FOURTH INDUSTRIAL REVOLUTION," Mjolner, 14 1 2015. [Online]. Available: https://mjolner.dk/2015/01/14/realizing-the-fourth-industrial-revolution/. [Accessed 19 2 2022]. |