캡스톤 디자인 I 최종결과 보고서

프로젝트 제목(국문): 웹 기반 메타버스 구축 플랫폼 구현
프로젝트 제목(영문):Implementation of a web-based metaverse building platform

프로젝트 팀(원): 학번: 20191792 이름: 최진아
프로젝트 팀(원): 학번: 20191750 이름: 이혜진
프로젝트 팀(원): 학번: 20191740 이름: 유선아

1. 중간보고서의 검토결과 심사위원의 '수정 및 개선 의견'과 그러한 검토의견을 반영하여 개선한 부분을 명시하시오.

- **test가 왜 필요한지에 대한 이유를 보충** :
  사용자가 사용할 수 있는 프로그램을 만들기 위해서 알파 테스트를 통해 개선점과 추가적으로 프로그램에 대한 의견을 받아서 요구사항에 적용시켜야 한다.
- **3d Object를 만든 이유를 보충** :
  1. 최종적으로 구현하고 싶은 부분은 링크를 통해 다른 Scene으로 이동하고자 하는 것이고, 3D object는 링크를 구현하기 위한 기초 도형으로 사용된다.
  2. 2d 화면 방식으로는 3축일때에 비해서 표현할 수 있는 범위가 적다. 사용자들로 하여금 더 자유롭게 표현할 수 있도록 하기 위해서 3d object를 사용하였다.
- **발표 내용에 대한 전개 방식 수정이 필요함** :
  단순 내용 전달만이 아닌 해당 프로젝트에 대한 배경, 필요성, 기대효과들에 대해서 논리적으로 연결시켜 전개했다.

2. 기능, 성능 및 품질 요구사항을 충족하기 위해 본 개발 프로젝트에서 적용한 주요 알고리즘, 설계방법 등을 기술하시오.

- 메뉴 버튼을 누르면 여러 가지 도형을 생성할 수 있다. 생성되는 도형의 종류는 circle, square, cylinder, 등 이다. 생성되는 도형은 현재 보이는 화면의 중앙에 생성되며, 색상은 랜덤으로 생성된다.

```
AFRAME.registerComponent('pressed-gui-button', {

    init: function() {
        //var sceneEl = document.querySelector('a-scene');
        var objectEl = document.querySelector('#objects');
        var markerEl = document.querySelector('#marker');

        var COLORS = [
            'pink',
            'blue',
            'yellow',
            'red',
            'peachpuff',
            '#2EAFAC',
            '#BAE'];

        // Add boxe when spacebar is pressed.
        this.el.addEventListener('click', function (e) {

            var newEl = document.createElement('a-entity');
            newEl.setAttribute('id', 'boxtest');
            newEl.setAttribute('geometry', {primitive: this.id});
            newEl.setAttribute('material', 'color', COLORS[Math.floor(Math.random() * COLORS.length)]); //random color
            newEl.setAttribute('loc', this.id + new Date().getTime() / 100);

            objectEl.appendChild(newEl);

            //newEl.classList.add('clickable');
            //newEl.addEventListener('click', handleClickEvent);

            var position = markerEl.object3D.getWorldPosition();
            //position.y = 0.5;
            newEl.setAttribute('position', position);
        });
    }
});
```

- DB 저장 : req.body에서 id가 boxtest인 entity를 저장하는 부분이다. position, rotation, scale, id, location, material, color를 순서대로 받아서 item.boxtest에 저장된다.

```
141     // ################ DB에 entity 저장하기 ##############
142     if("boxtest" in req.body)
143     {
144       // boxtest에 이미 저장된 요소들 undefined가 될 때까지 비우기
145       while(item.boxtest.shift() !== undefined)
146
147       console.log('boxtest try save');
148       // console.log('length' + String(req.body.boxtest.length));
149
150       // boxtest value에 entity 하나씩 push하기
151       for(let i = 0; i < (req.body.boxtest.length); i++){
152         item.boxtest.push({
153           x: req.body.boxtest[i][0]['x'], // position
154           y: req.body.boxtest[i][0]['y'],
155           z: req.body.boxtest[i][0]['z'],
156           yaw: req.body.boxtest[i][1]['x'], // rotation
157           pitch: req.body.boxtest[i][1]['y'],
158           roll: req.body.boxtest[i][1]['z'],
159           xscale: req.body.boxtest[i][2]['x'], //scale
160           yscale: req.body.boxtest[i][2]['y'],
161           zscale: req.body.boxtest[i][2]['z'],
162           id: req.body.boxtest[i][2], // id
163           loc: req.body.boxtest[i][3],  // loc
164           geometry: req.body.boxtest[i][4],  // material
165           color: req.body.boxtest[i][5] // color
166         });
167       }
168       console.log('boxtest finish save');
169     }
```

3. 요구사항 정의서에 명세된 기능 및 품질 요구사항에 대하여 최종 완료된 결과를 기술하시오.

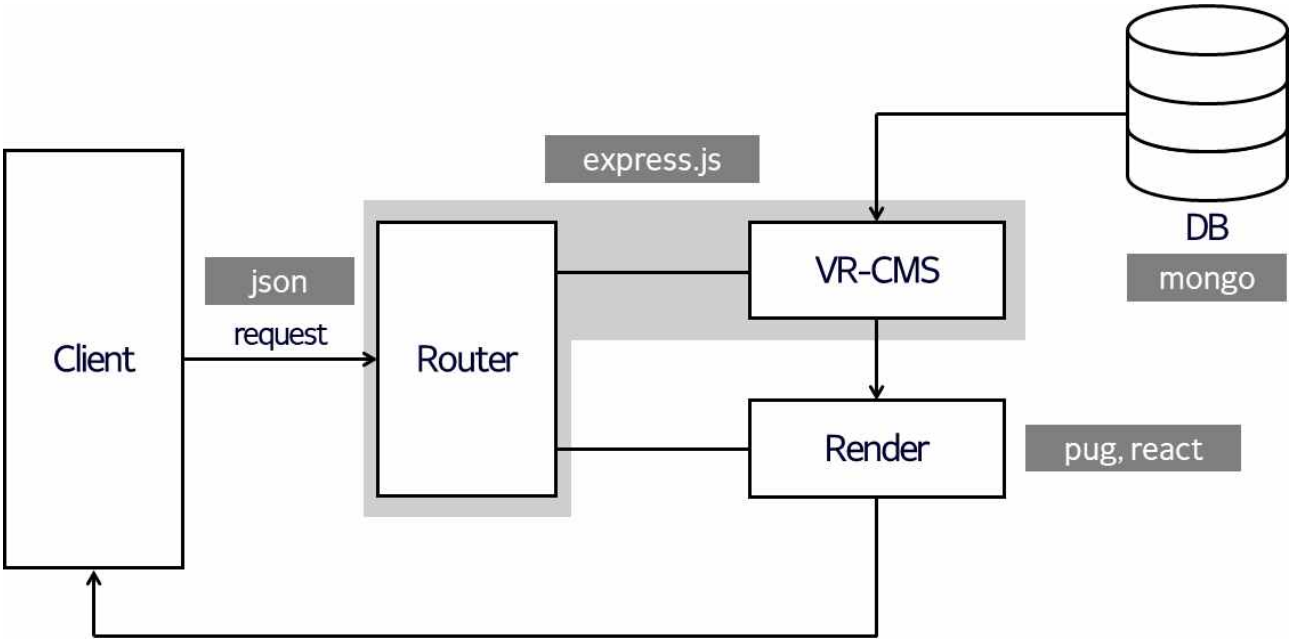


그림 3 플랫폼 시스템 전체 아키텍처

- 클라이언트는 request를 통해서 라우터로 json형식의 데이터를 전송함
- 라우터는 각각 플랫폼인 VR-CMS, 화면을 보여주는 부분인 Render와 연결되어 있으며 플랫폼 VR-CMS는 MongoDB에서 데이터를 가져옴
- Router, VR-CMS는 express.js로, render는 pug, react를 사용하여 구성함

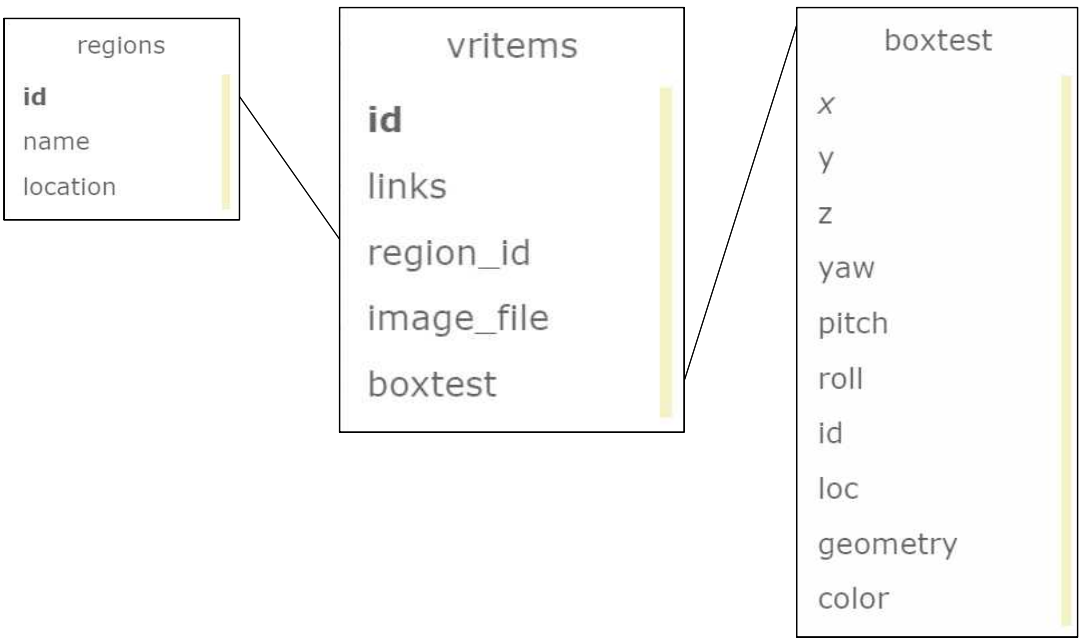


그림 4 DB구조

- 웹 기반 메타버스 플랫폼 캡쳐화면

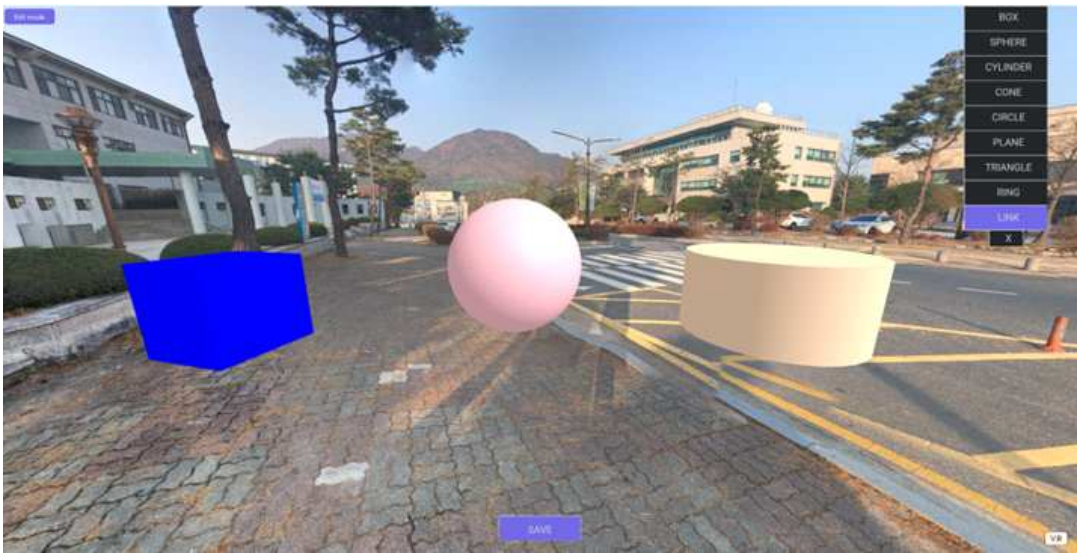


그림 5 오브젝트 버튼 및 오브젝트 생성 화면

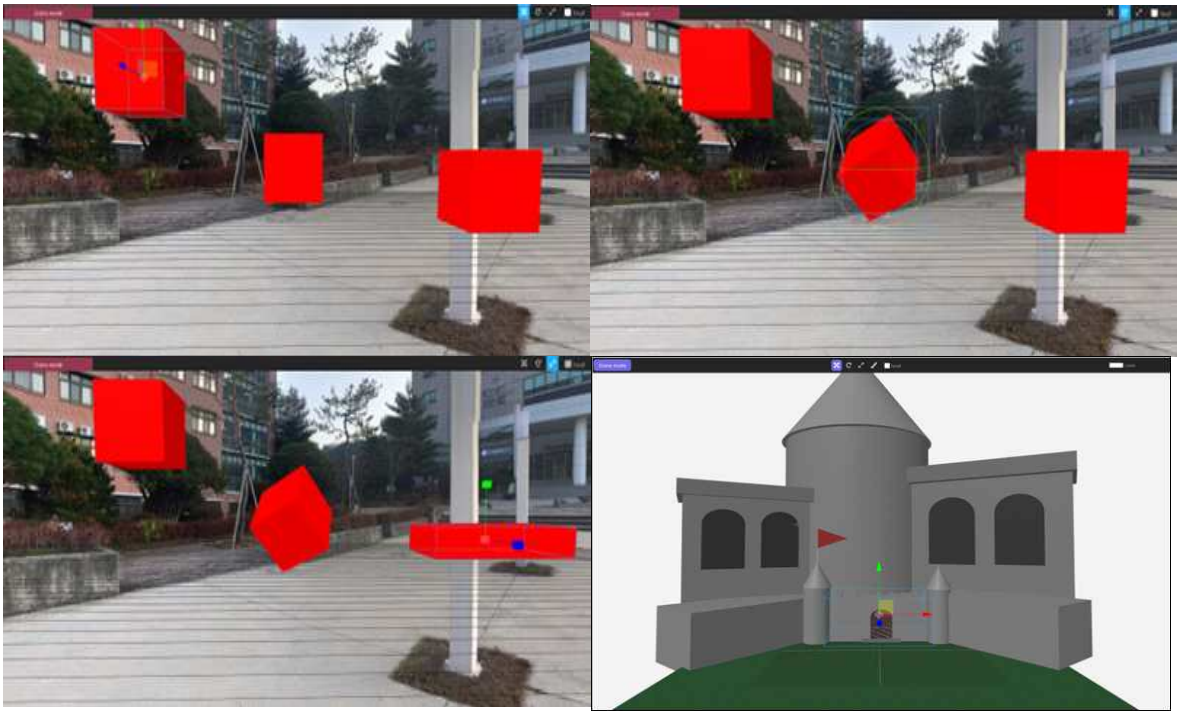


그림 6 3d 오브젝트 편집화면 및 예시

4. 구현하지 못한 기능 요구사항이 있다면 그 이유와 해결방안을 기술하시오.

최초 요구사항	구현 여부	이유	해결방안
3D object 색깔 변경	미구현	일정 부족 (시간 내에 구현 못함)	캡스톤 디자인Ⅱ에서 진행
3D link 연결	미구현	일정 부족 (시간 내에 구현 못함)	캡스톤 디자인Ⅱ에서 진행
외부 3D object 추가	미구현	일정 부족 (시간 내에 구현 못함)	캡스톤 디자인Ⅱ에서 진행

5. 요구사항을 충족시키지 못한 성능, 품질 요구사항이 있다면 그 이유와 해결방안을 기술하시오.

분류(성능, 속도 등) 및 최초요구사항	충족 여부	이유
웹 페이지 로딩 시간	충족 (평균 2.5sec 내외)	
동시 사용자 접속 수		일정 부족
데이터 형식 오류 응답 시간	충족 (평균 200ms 이내)	
장애 복구 대책		일정 부족
웹 호환성	충족(Internet Explorer 제외하고 가능)	
정보제공	충족	

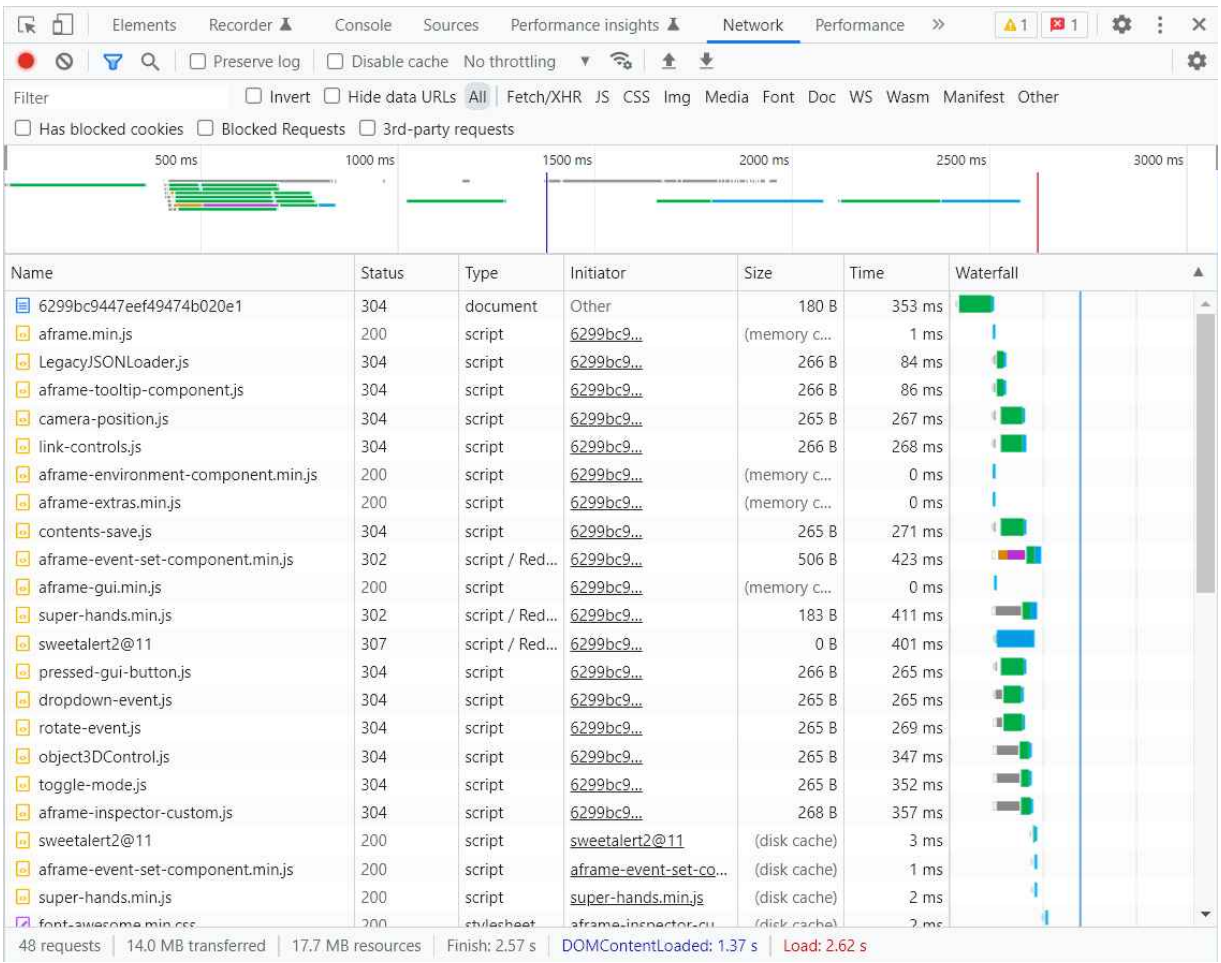
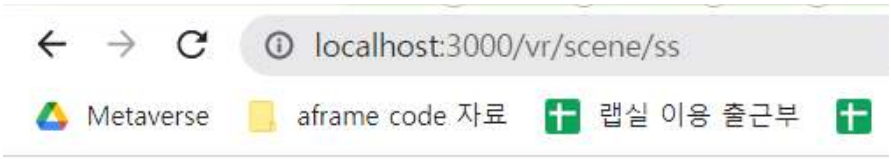


그림 7 VR-CMS sample scene 화면 로드 측정 이미지



{"error":"vritem not found"}

그림 8 에러 발생 화면

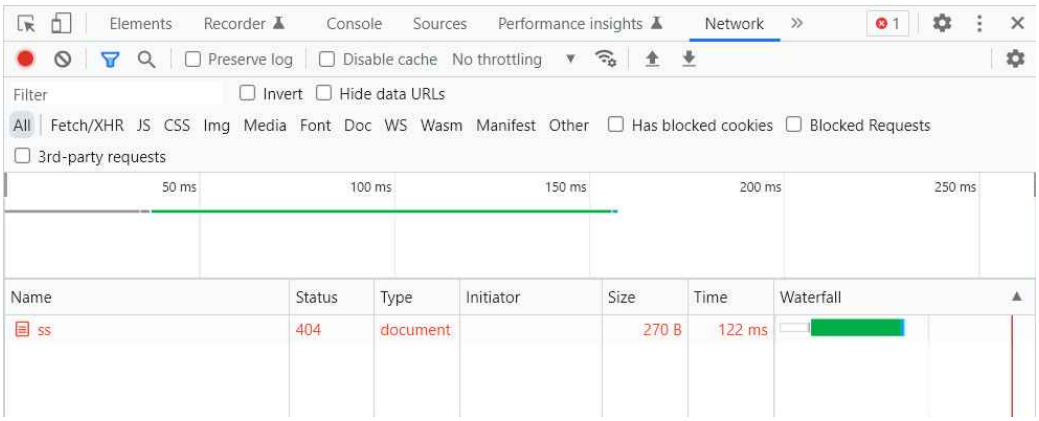


그림 9 link 연결 오류 시 응답 시간

6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

- 사용자 매뉴얼과 실행파일(window 기준)
  1. 시스템 실행 환경 구축 : node.js 최신버전 다운로드 (권장 버전 : 18.3.0)
     mongoDB 최신버전 다운로드 (권장 버전 : 5.0.6)
  2. 자신의 local 컴퓨터에 VR-CMS 소스코드 파일 다운로드
  3. cmd창에서 다음 명령어 실행
     >> npm install
  4. cmd 창에서 다운로드해둔 소스코드 파일위치로 이동(아래 주소는 예시)
     >> cd C: Users\user\Desktop\VR-CMS-main
  5. VR-CMS 폴더로 이동한 상태에서 db 실행
     >> mongod --port 27017 -- dbpath C: Users\user\Desktop\VR-CMS-main
  6. 서버 실행
     >> set DEBUG=express:* & node ./bin/www

  7. 서버 실행 후 웹브라우저에서 localhost:3000으로 웹사이트 접속

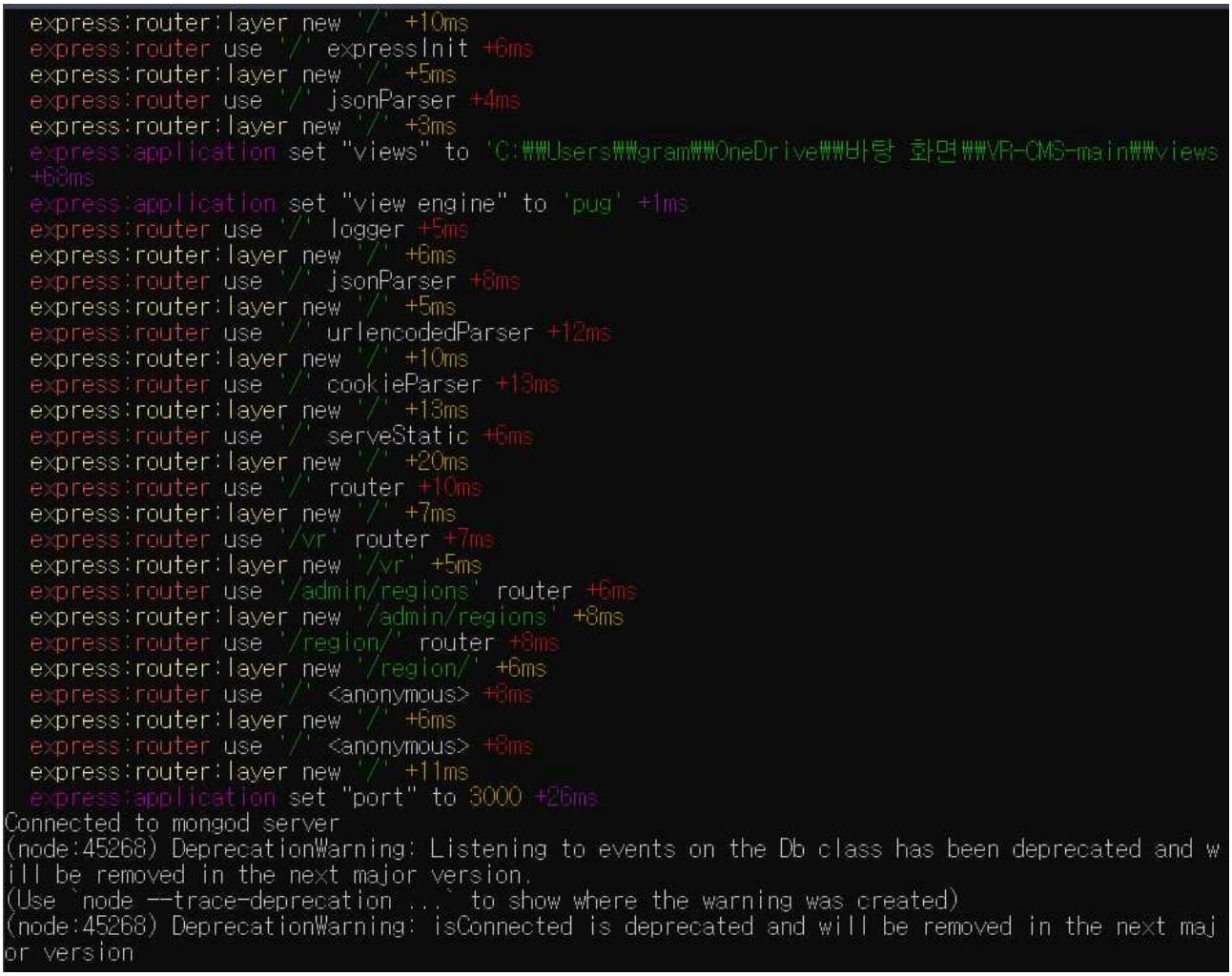


그림 10 정상적으로 서버가 실행되었을 시의 cmd 창 화면

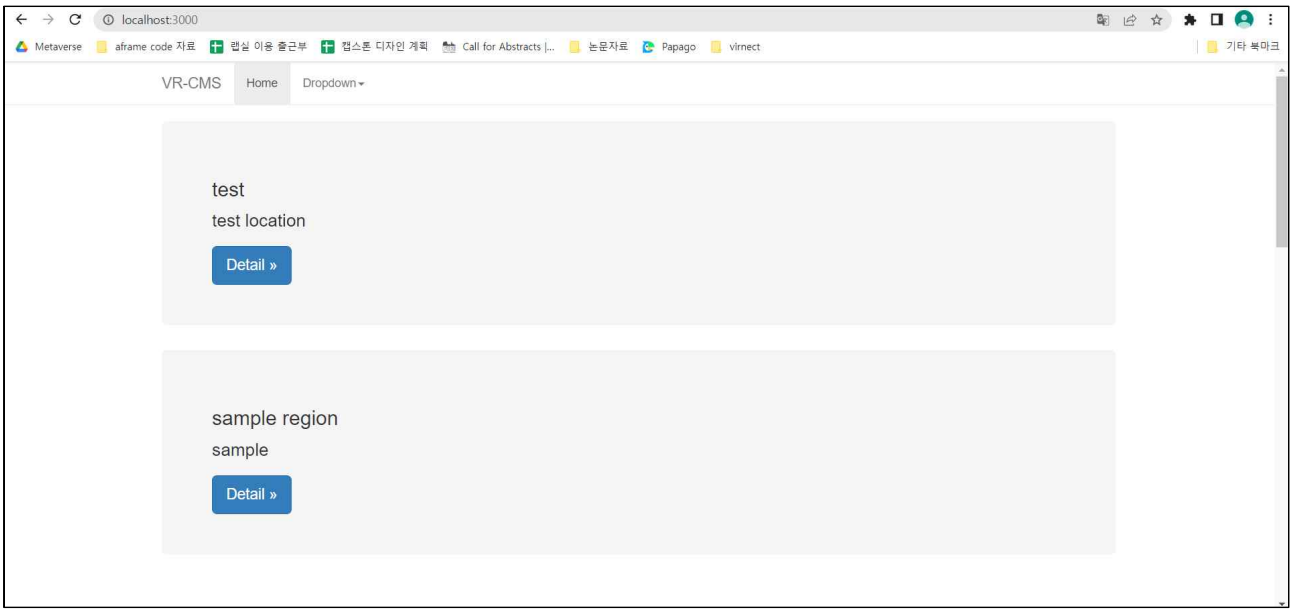


그림 11 정상적으로 웹사이트에 접속한 화면

7. 캡스톤디자인 결과의 활용방안

1) 시장성

- 기존의 메타버스와 달리 웹 기반으로 운영되어 접근이 쉬움
- 단순히 콘텐츠를 즐기는 consumer 뿐만 아니라 콘텐츠를 생산하는 producer의 역할을 통해서 플랫폼을 상호작용적으로 이용할 수 있음
- 플랫폼을 사용해본 사람들을 대상으로 설문조사 진행
  - 87% 이상이 플랫폼을 추천할 의사가 있다고 밝힘
  - 길 안내나 매장 내부 위치 소개, 마케팅, 자기 PR 등 다양한 방면으로 플랫폼을 활용하고 싶다고 밝힘

2) 교육성

- 웹 기반 메타버스를 사용하여 비대면으로 공학 진로 프로그램을 진행
  - 참여자 대다수가 공학 관련 진로에 흥미를 느끼고 플랫폼 사용에도 만족도를 보임