

온라인 강의 사이트

강동욱

Tel : 010-5220-4331

E-mail : ebyul96428@naver.com

목차

1. 프로젝트 목표
2. 개발 환경
3. 구현 기술
4. 그 외 기술
5. 후기

1. 프로젝트 목표

목표 1 : React 와 SpringBoot를 사용하여 수강신청 사이트 제작(서버는 MySql 사용)

목표 2 : 수강 신청 사이트에서 강의 클릭 시 해당 강의의 ID에 따른 데이터를 서버에 요청한 뒤 강의 상세 페이지로 출력

목표 3 : 수강 신청사이트에 로그인 기능 추가 / 회원가입 기능 추가

목표 4 : 강의 목록 페이지 제작 및 검색기능 포함

목표 5 : 로그인 상태에서 장바구니 기능 활성화 / 강의 상세페이지에서 장바구니 담기 기능 추가

목표 6 : 장바구니에 담긴 강의 결제 기능 추가 / 결제 내역은 서버에 추가

2. 개발 환경

개발 기간 : 2023.01. ~ 2023.02 (2개월)



| 온라인 교육 사이트

플랫폼 : Windows

개발 인원 : 5명(팀 프로젝트)

담당 역할 : 교육 상세 페이지 – 웹 디자인 설정(개인작업 기여도 100%),
상세페이지-커리큘럼(개인작업 기여도 100%),
상세페이지 – 수강평(개인작업 기여도 100%),
장바구니 페이지 개발(팀 작업 기여도 5%) – 전체 선택 기능

개발 환경

언어 : Java (JDK 11), HTML/CSS, TypeScript

프레임 워크 : Spring Boot, React

데이터베이스(DB) : MySQL

IDE : Eclipse 2022-12, Visual Studio Code, POSTMAN, HeidiSQL, Gradle

API : RESTful API

2. 개발 환경 - 툴

IDE



언어

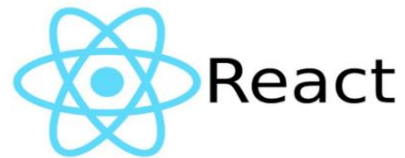


API : RESTful API

빌드 도구



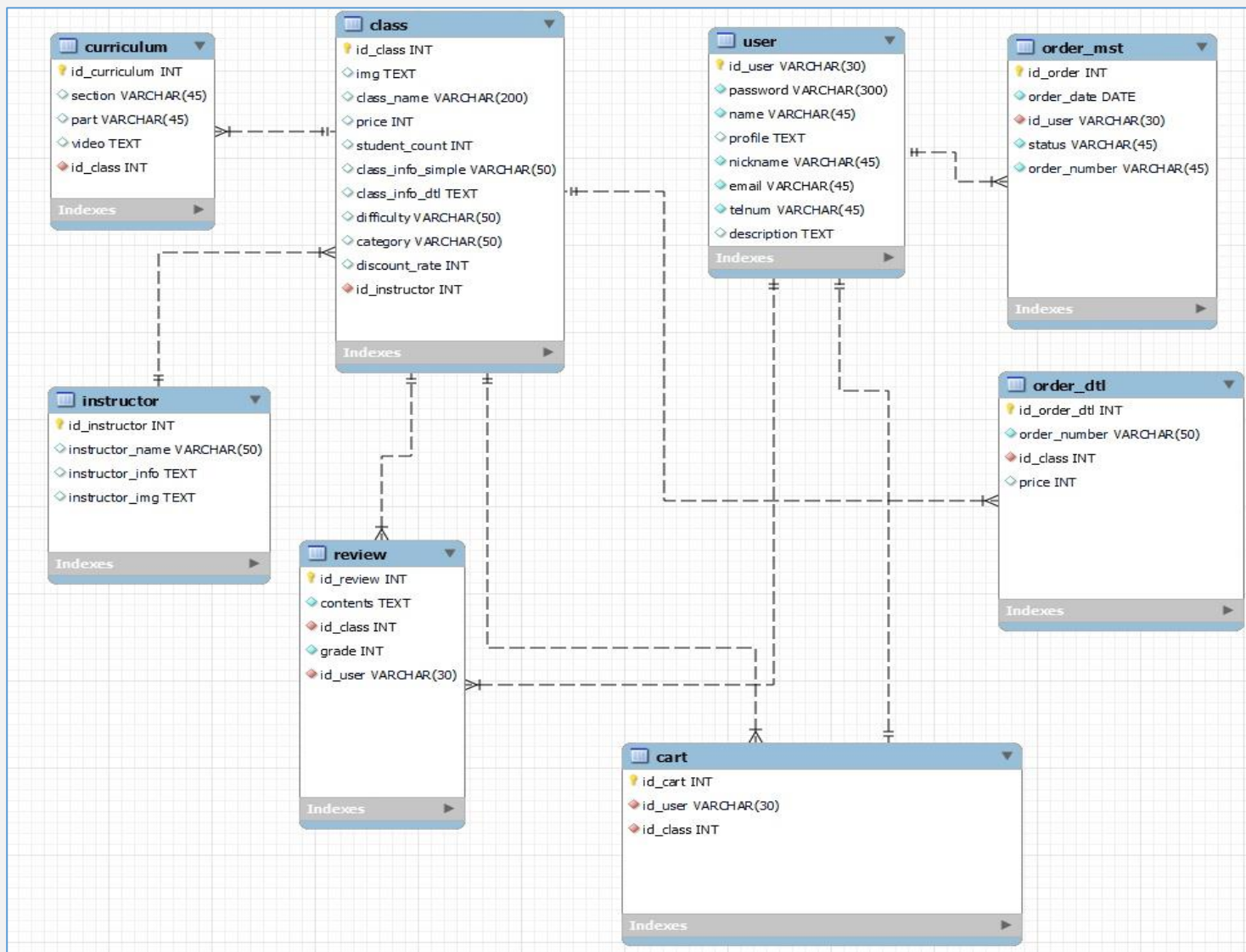
프레임 워크



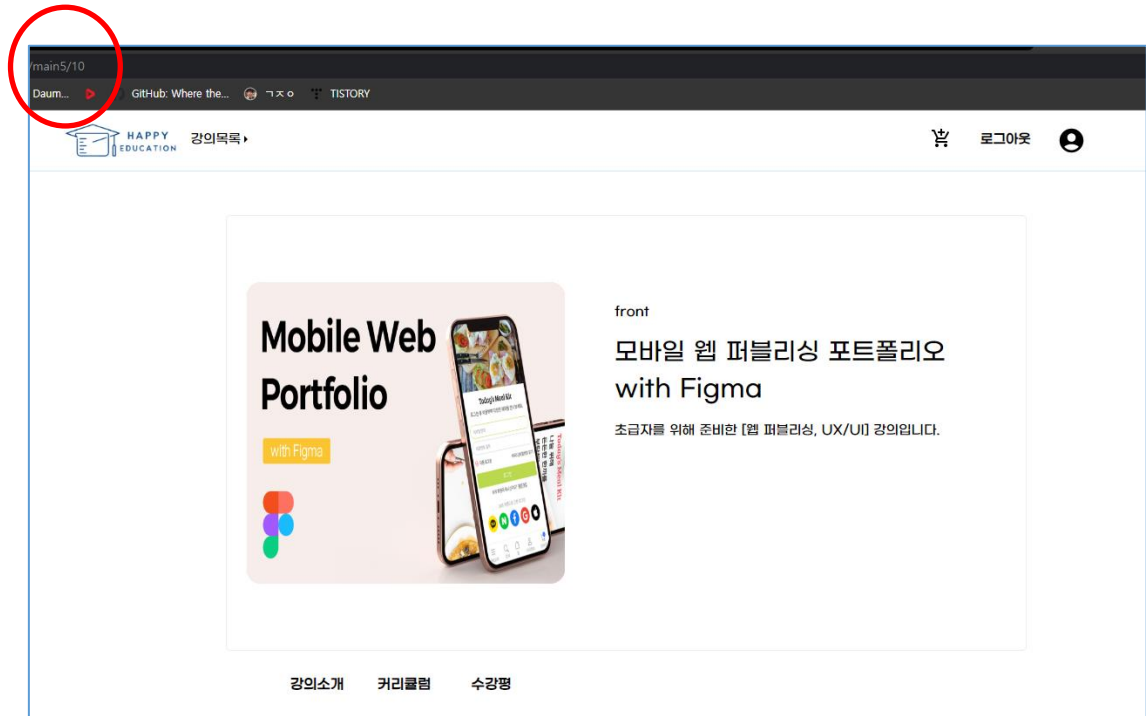
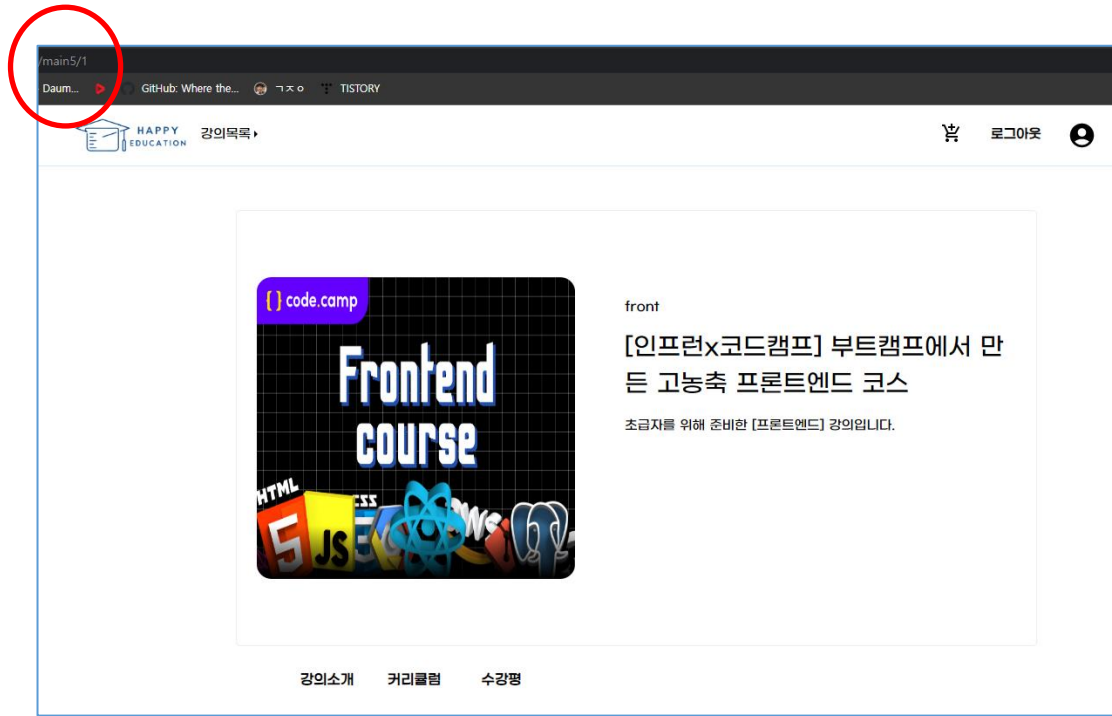
서버



2. 개발환경 - DB ERD



3. 구현기술 - 강의 상세 페이지[페이지 출력]



메인 페이지나 목록 페이지에서 강의를 클릭 할 시, 해당 강의의 클래스ID에 따라 상세 페이지에서 출력되는 데이터가 달라지도록 구성했습니다.

```
// idClass 입력시 파라미터로 받도록 설정
const { idClass } = useParams<string>();
```

```
axios
.get('http://localhost:4040/api/main5/${idClass}/')
.then((response) => {
  setDetailItems(response.data.data);
  setInstruct(response.data.data.instructor);
});
```

클래스 ID는 URL에서 사용될 수 있도록 useParams 을 사용하여 파라미터 데이터로 받을 수 있도록 구성하였습니다

```
<>
{detailItems && instruct && (
  <div className="detail5_main" key={detailItems.idClass}>
    <div className="detail5_topbanner">
      <div className="detail5_top_body">
        <img className="detail5_img" src={detailItems.img} />
        <div className="detail5_title">
          <p>{detailItems.category}</p>
          <p id="item5_title">{detailItems.className}</p>
          <p id="item5_teacher">{detailItems.instructor}</p>
          <p>{detailItems.classInfoSimple}</p>
        </div>
      </div>
    </div>
  </div>
)
```

3. 구현기술 - 강의 상세 페이지[페이지 출력 - DTO]

Name	Type	Description	Required
Id_class	Int	강의 번호	0
img	String	강의 이미지	0
class_name	String	강의 이름	0
Class_info_simple	String	강의 간단 설명	0
Class_info_dtl	String	강의 설명	0
price	Int	수강 가격	0
Category	String	카테고리	0
Difficulty	String	난이도	0
Instructor	List	강사 명	0
Curriculum_list	List	커리큘럼	0
Review_list	List	리뷰	0

3. 구현기술 - 강의 상세 페이지[페이지 출력]



front

[인프런x코드캠프] 부트캠프에서 만든
고능축 프론트엔드 코스

초급자를 위해 준비한 [프론트엔드] 강의입니다.

강의소개 **커리큘럼** 수강평

강의소개 **커리큘럼** 수강평

강의소개 강의소개 클릭

1년간 300여명의 개발자를 배출한 '진짜' 부트캠프의 고능축 커리큘럼입니다. 이 강의 하나로 여러분은 현업 주니어 개발자 수준까지 성장 가능하며, 프론트엔드 기술스택의 활용 능력과 지식을 얻게 될거예요. 누구나 사회적, 경제적, 교육적 배경에 상관없이 커리어를 쌓을 수 있도록, [인프런x코드캠프]가 준비했습니다 :)



Try anything. Try everything! 코드캠프는 유능한 개발자를 양성하며 개발의 미래를 바꾸고자 하는 목표를 가지고 있습니다. 사회적, 경제적, 교육적 배경에 상관없이 누구에게나 커리어 전환의 기회를 제공하기 위해 인프런에 나타났습니다. 코랩과 함께 커리어 점프에 도전하세요!

강의소개 **커리큘럼** 수강평

커리큘럼

강의 1

강의 2

강의 3

커리큘럼 클릭

강의소개 커리큘럼 **수강평**

수강평 수강평 클릭

aaa

★★★★☆

1

bbb

★★★★☆

1

<

1

>

작성하기

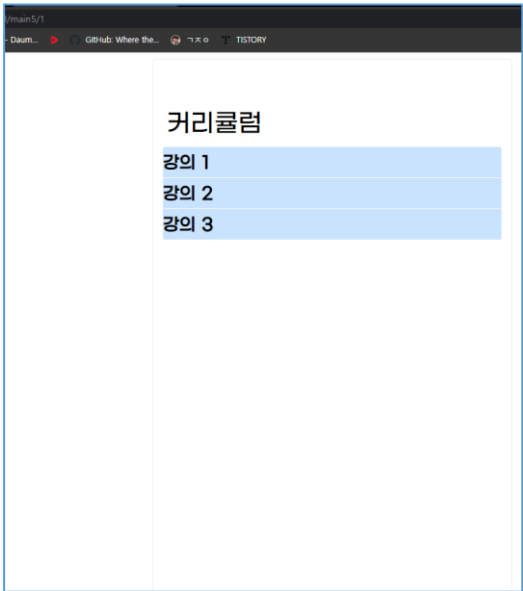
```
// 클릭 스크롤
const [select, setSelect] = useState<string>('');

const onScrollClick = (id: string) => {
  if (id === 'content1')
    idRef1.current?.scrollIntoView({ behavior: 'smooth' });
  if (id === 'content2')
    idRef2.current?.scrollIntoView({ behavior: 'smooth' });
  if (id === 'content3')
    idRef3.current?.scrollIntoView({ behavior: 'smooth' });

  setSelect(id);
};
```

각 항목별로 id를 할당하여 해당 항목을 클릭 시, id가 일치하는 영역으로 스크롤 될 수 있도록 네비게이션바를 구축하였습니다.

3. 구현기술 - 강의 상세 페이지[커리큘럼]



영상 출력은 Iframe 태그를 사용하여 출력했습니다.

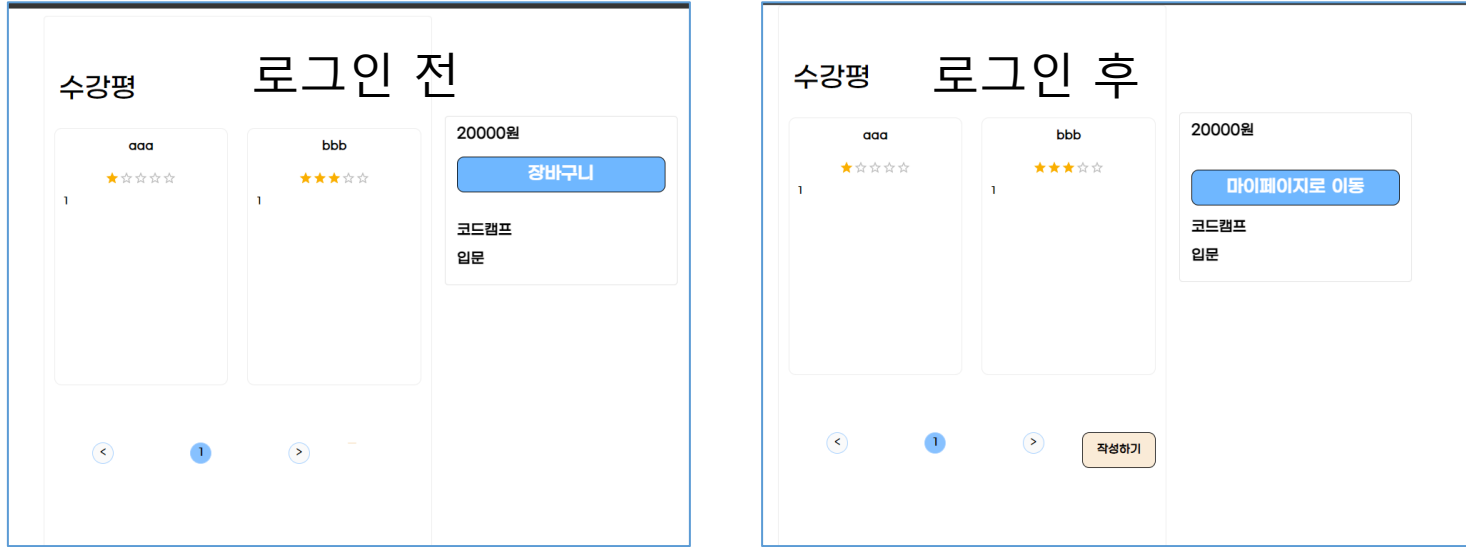
커리큘럼에는 해당 강의의 Youtube 영상 URL을 탑재하였으며, 로그인 여부와 구매 여부에 따라 표시되도록 설정했습니다.

```
<ul>
  className="car5_title"
  onClick={() => setMenu((isOpen) => !isOpen)}
  key={section}
  >
  <h4>강의 {section}</h4>
  {itemList.map((item) => (
    <li className={isOpen ? 'car5_item' : 'hide_item'}>
      강의 {item.part}
      <iframe
        className={
          {
            idUser:
              ? status
              ? isOpen
              ? 'car5_video'
              : 'hide_video'
              : 'hide_video'
              : 'hide_video'
          }
        }
        width="560"
        height="315"
        src={item.video}
        title="YouTube video player"
        allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
        allowFullScreen
      ></iframe>
    </li>
  ))}
</ul>
```

3. 구현기술 - 강의 상세 페이지[커리큘럼 - DTO]

Name	Type	Description	Required
Id_Curriculum	Int	커리큘럼 번호	0
Section	String	강의 이미지	0
Part	String	강의 이름	0
Video	String	강의 간단 설명	0
Id_class	Int	강의 번호	0

3. 구현기술 - 강의 상세 페이지[수강평]



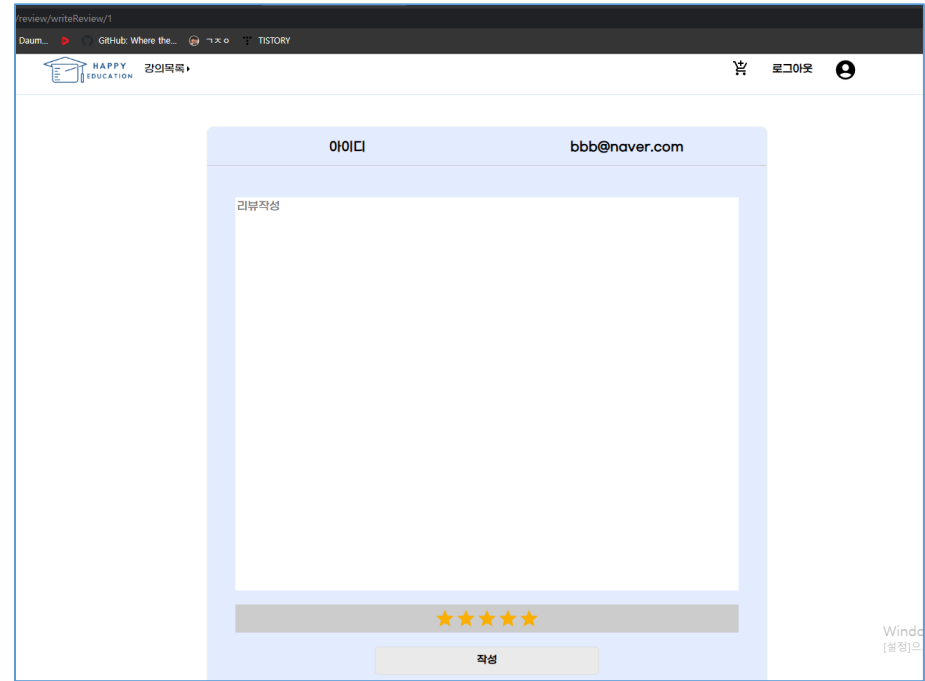
기본적으로 각 상세페이지별 수강평 확인은 가능하나, 수강평 작성, 수정 및 삭제 기능은 로그인인 된 상태에서만 가능하도록 설정했습니다.

또한 수정 및 삭제 기능은 로그인인 되어있고 해당 수강평의 작성자 (UserId)가 일치해야 활성화 되도록 조건을 따로 지정하였습니다.

```
<div className="reviewSubmit">
  <button
    type="button"
    className={
      idUser === CONTENT.idUser ? 'loginEnable' : 'loginDisable'
    }
    onClick={() => UpdateReview(CONTENT.idReview)}
  >
    수정 하기
  </button>
  <button
    type="button"
    className={
      idUser === CONTENT.idUser ? 'loginEnable' : 'loginDisable'
    }
    onClick={() => DeleteReview(CONTENT.idReview)}
  >
    삭제 하기
  </button>
</div>
```

```
return (
  <div className="review5_list">
    { /* 보여지는 페이지 */ }
    {ReviewItem}
    <div className="review5_change">
      <div className="reviewCollect5">...
    </div>
    <button
      type="button"
      className={idUser ? 'writeReview5' : 'writeReview5 disabled'}
      onClick={WriteReview}
    >
      작성 하기
    </button>
  </div>
</div>
);
```

3. 구현기술 - 강의 상세 페이지[수강평 작성]



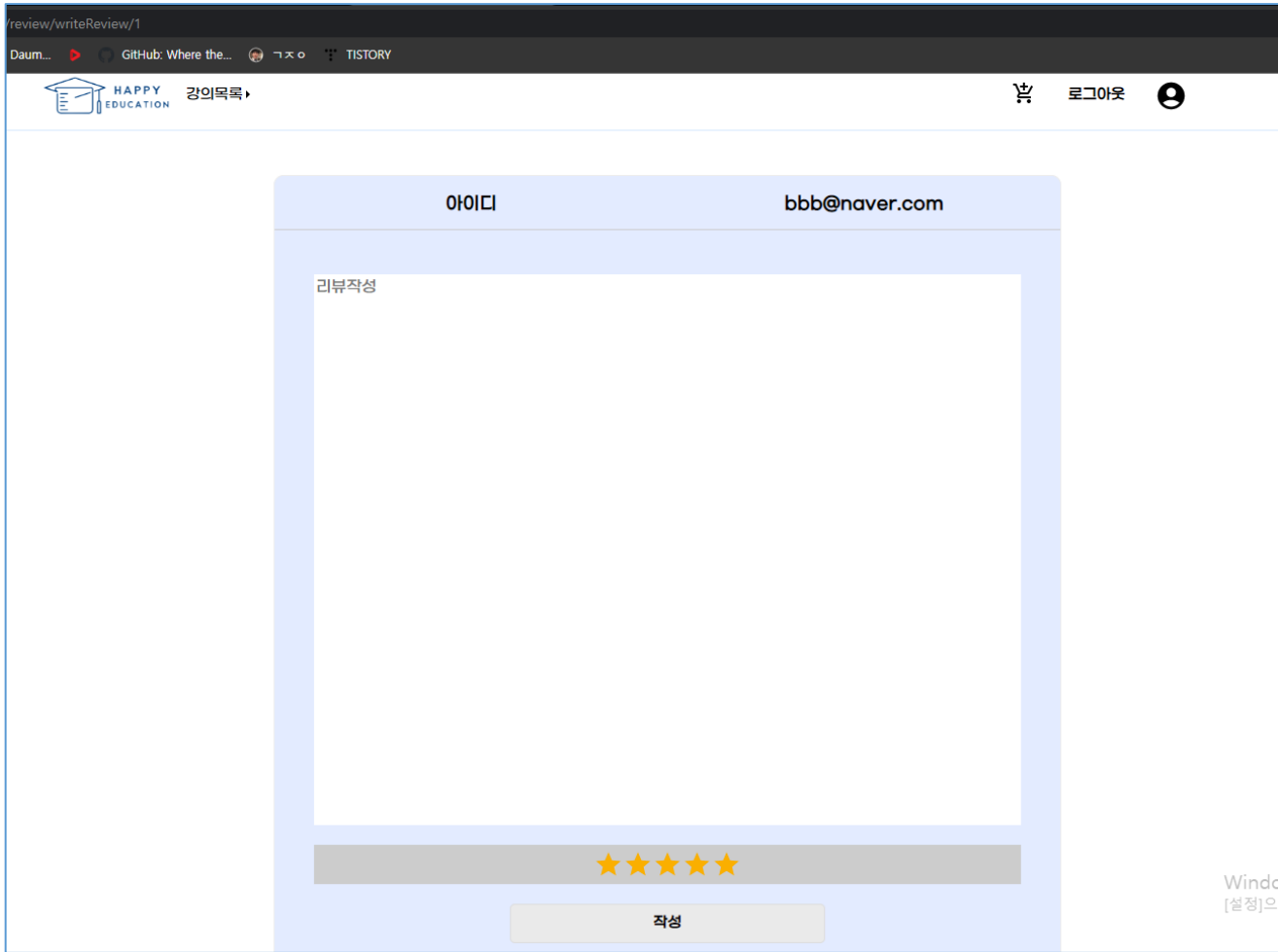
해당 강의를 구매했을 때, 작성하기 버튼을 클릭하여 수강평을 작성 할 수 있습니다.

수강평은 상세페이지에서 작성하는 것이 아닌 수강평 작성 페이지로 넘어가서 작성합니다.

```
const WriteReview = () => {  
  navigator(`/review/writeReview/${idClass}`, { state: idUser });  
};
```

작성하기 버튼을 클릭 시 idClass를 Param 으로 넘기고, idUser를 통해 아이디 정보를 같이 넘겨준다.

3. 구현기술 - 강의 상세 페이지[수강평 작성]



```
<div className="boardBody5">
  <div className="boardHead5">
    <div className="board_ID">아이디</div>
    <div className="input_ID">[idUser]</div>
  </div>
  <div className="boardText5">
    <textarea
      className="reviewArea"
      placeholder="리뷰작성"
      onChange={handleTextArea}
    />
  </div>
  </* 별점기능 */>
  <Box sx={{ '& > legend': { mt: 2 } }} className="starRating">
    <Rating
      className="WriteRating"
      value={grade}
      size="large"
      onChange={(event, newgrade) => {
        setGrade(newgrade);
      }}
    />
  </Box>
  <div className="boardFooter">
    <button type="button" className="writeBtn5" onClick={WriteBtn}>
      작성
    </button>
  </div>
</div>
);
```

별점 기능은 Material-UI(mui)에서 Rating기능을 찾아 해당 프로젝트에서 구축하였습니다.

Ex) 작성 페이지를 따로 제작하여 기존의 코드가 더 길어지지 않도록 하였습니다.

3. 구현기술 - 강의 상세 페이지[수강평 작성]

아이디 bbb@naver.com

수강평 작성 페이지 입니다.

★★★★☆

작성

표시되는 아이디는 작성하기 버튼을 클릭하고, 해당 페이지로 이동될 때 아이디 정보를 함께 넘겨 표시합니다.

별점 기능을 추가하여 1점부터 5점까지 원하는 점수를 매겨 수강평으로 등록할 수 있습니다.



강의소개 커리큘럼 수강평

bbb@naver.com

★★★★☆

수강평 작성 페이지 입니다.

수정하기 삭제하기

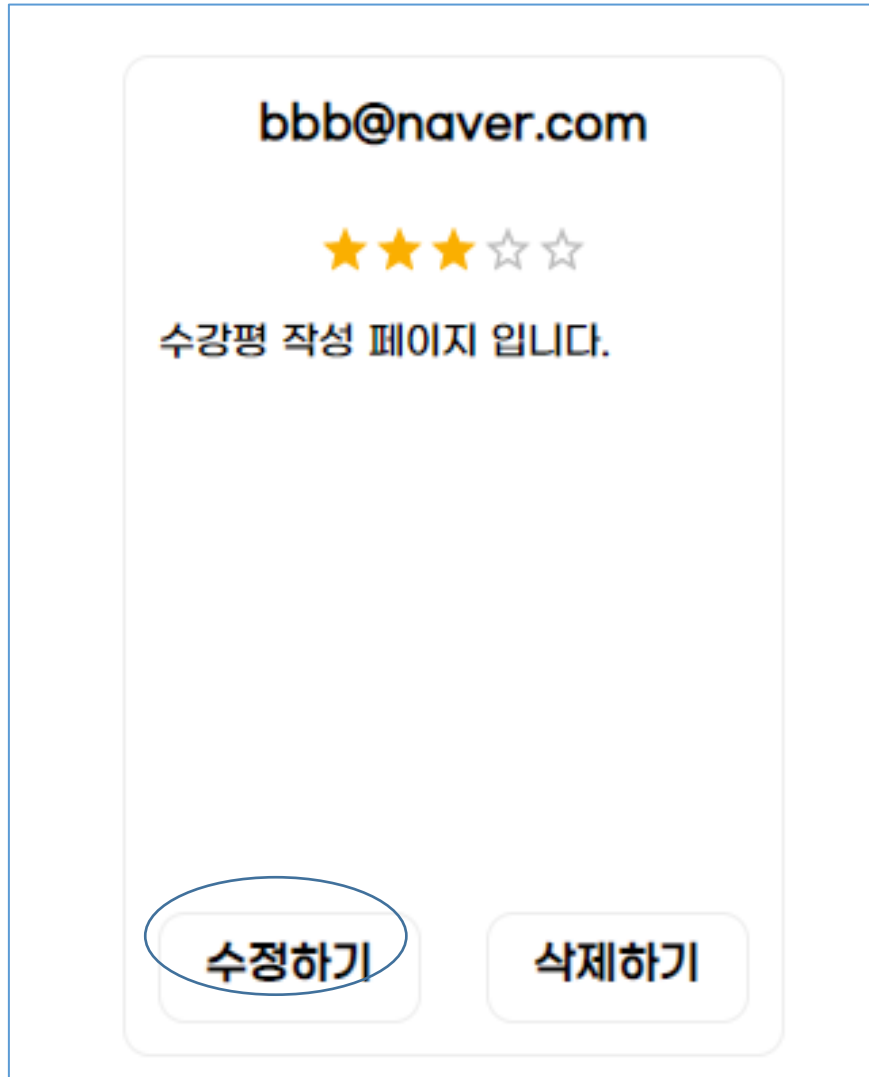
작성 버튼을 눌러 서버에 해당 수강평을 저장하고, 저장된 수강평을 바로 확인할 수 있습니다.

```
const WriteBtn = () => {  
  const postReview = { idUser, contents, idClass, grade };  
  axios  
    .post('http://localhost:4040/review/writeReview', postReview)  
    .catch((error) => console.log(error.message));  
  navigator('/main5/${idClass2.idClass}');  
};
```

3. 구현기술 - 강의 상세 페이지[수강평 작성 - DTO]

Name	Type	Description	Required
Id_Review	Int	리뷰 번호	0
idUser	String	로그인 아이디	0
Contents	String	수강평 내용	0
Id_class	Int	강의 번호	0
Grade	Int	별 점	0

3. 구현기술 - 강의 상세 페이지[수강평 수정 및 삭제]



수정하기 버튼 및 삭제하기 버튼의 활성화는 로그인 상태여야 하며, 수강평이 등록된 테이블에 있는 아이디 정보와 로그인 되어있는 아이디의 정보가 일치해야 활성화되도록 설정했습니다.

```
<div className="reviewSubmit">
  <button
    type="button"
    className={
      idUser === CONTENT.idUser ? 'loginEnable' : 'loginDisable'
    }
    onClick={() => UpdateReview(CONTENT.idReview)}
  >
    수정하기
  </button>
  <button
    type="button"
    className={
      idUser === CONTENT.idUser ? 'loginEnable' : 'loginDisable'
    }
    onClick={() => DeleteReview(CONTENT.idReview)}
  >
    삭제하기
  </button>
</div>
```

3. 구현기술 - 강의 상세 페이지[수강평 수정 및 삭제]

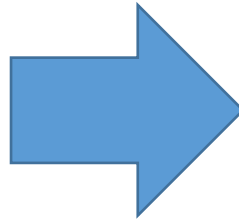
아이디

bbb@naver.com

수정된 내용입니다.

★★★★★

작성



bbb@naver.com

★★★★★

수정된 내용입니다.

수정하기

삭제하기

PATCH 를 사용하여 수강평 내용
및 별 점만 수정 후 서버에
저장될 수 있도록 구축

```
const UpdateBtn = () => {  
  const patchReview = { contents, grade };  
  axios  
    .patch(...  
    )  
    .then((respo) => { ...  
    })  
    .catch((error) => console.log(error.message));  
  navigator(` /main5/${idClass.idClass}`);  
};
```

3. 구현기술 - 강의 상세 페이지[수강평 수정 - DTO]

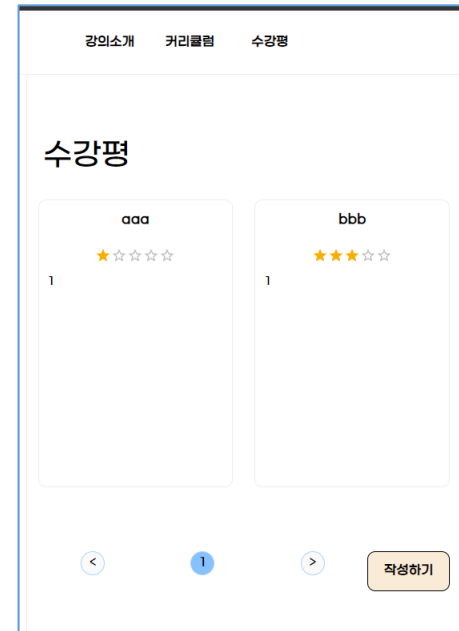
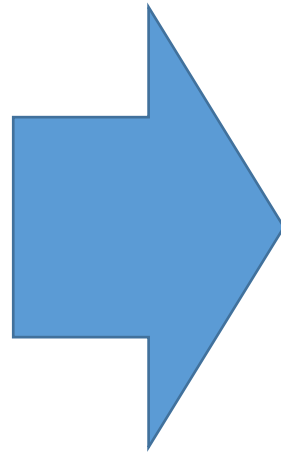
Name	Type	Description	Required
Id_Review	Int	리뷰 번호	0
idUser	String	로그인 아이디	0
Contents	String	수강평 내용	0
Id_class	Int	강의 번호	0
Grade	Int	별 점	0

3. 구현기술 - 강의 상세 페이지[수강평 수정 및 삭제]



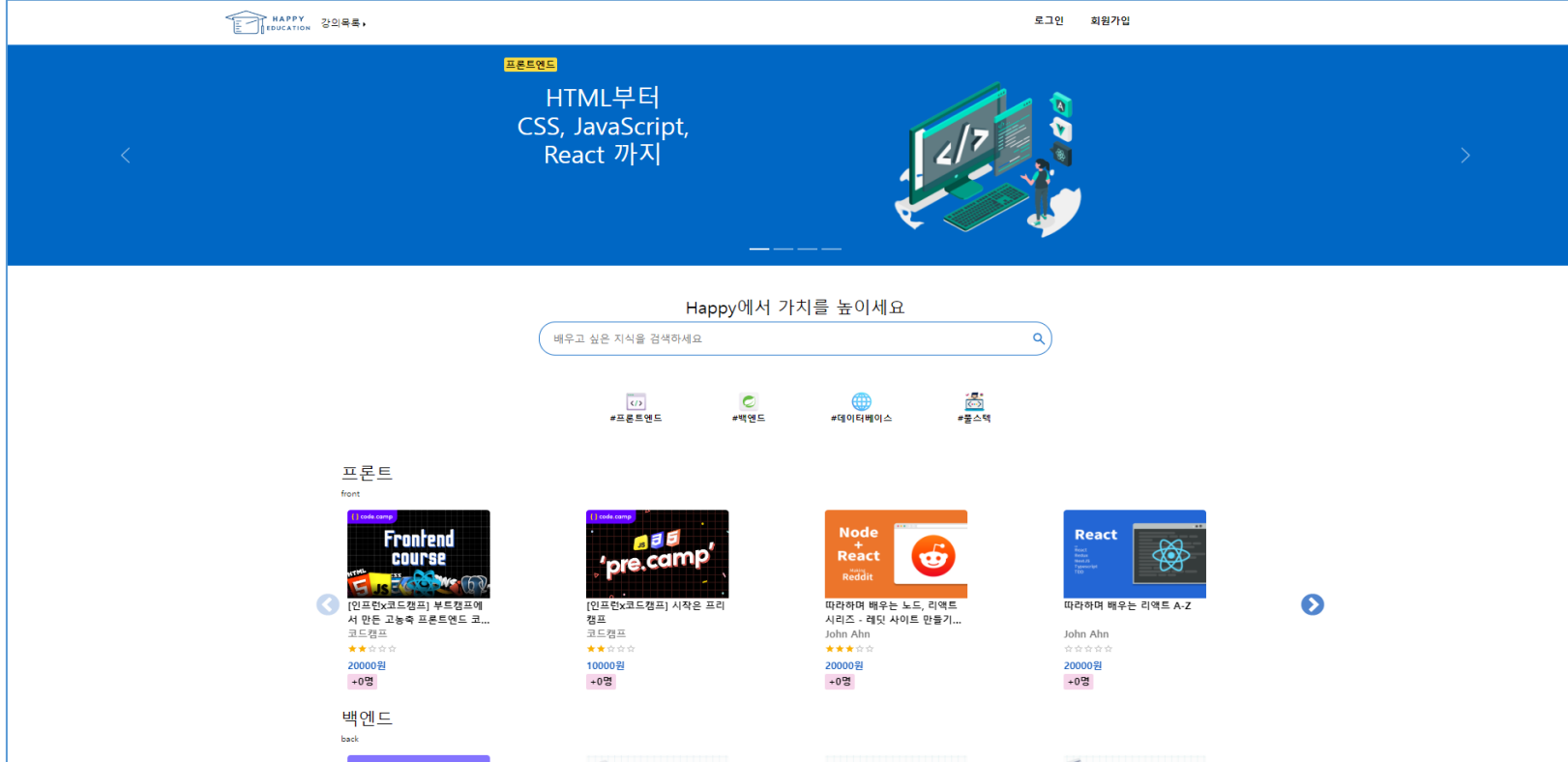
삭제하기 버튼을 눌러 등록된 수강평을 삭제 할 수 있습니다.

```
const DeleteReview = (idReview?: number) => {  
  axios  
    .delete(`http://localhost:4040/review/deleteReview/${idReview}`)  
    .catch((error) => console.log(error.message));  
  window.location.reload();  
};
```



Window.location.reload();
를 사용해서 현재 위치에서
페이지가 새로고침 되도록
구성하였습니다.

4. 그 외-홈페이지 주요기능[메인페이지]



```

{ /*배너*/ }

<Banner />

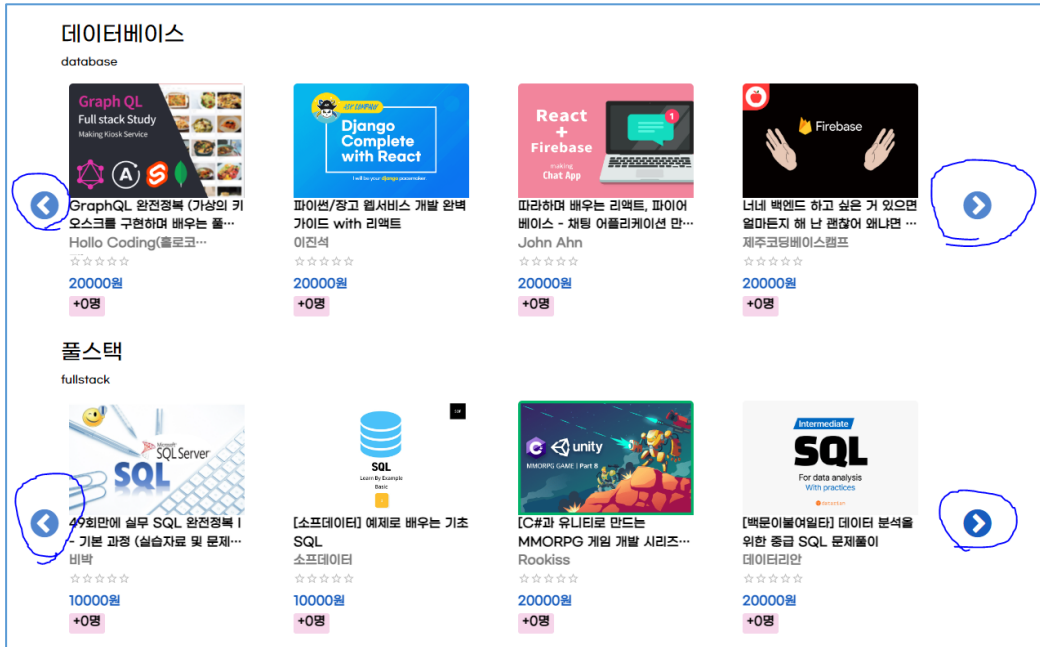
{ /* 검색 */ }
<div className="search_container4"> ...
</div>
{ /* 아이콘 목록 */ }
<div className="lecture_icon_list4"> ...
</div>
{ /* 강의 목록 */ }

<FrontCarousel />

```

메인화면 구성은 배너, 검색란, 각 강의 카테고리별 내용으로 구성되어 있습니다.

4. 그 외-홈페이지 주요기능[메인페이지]



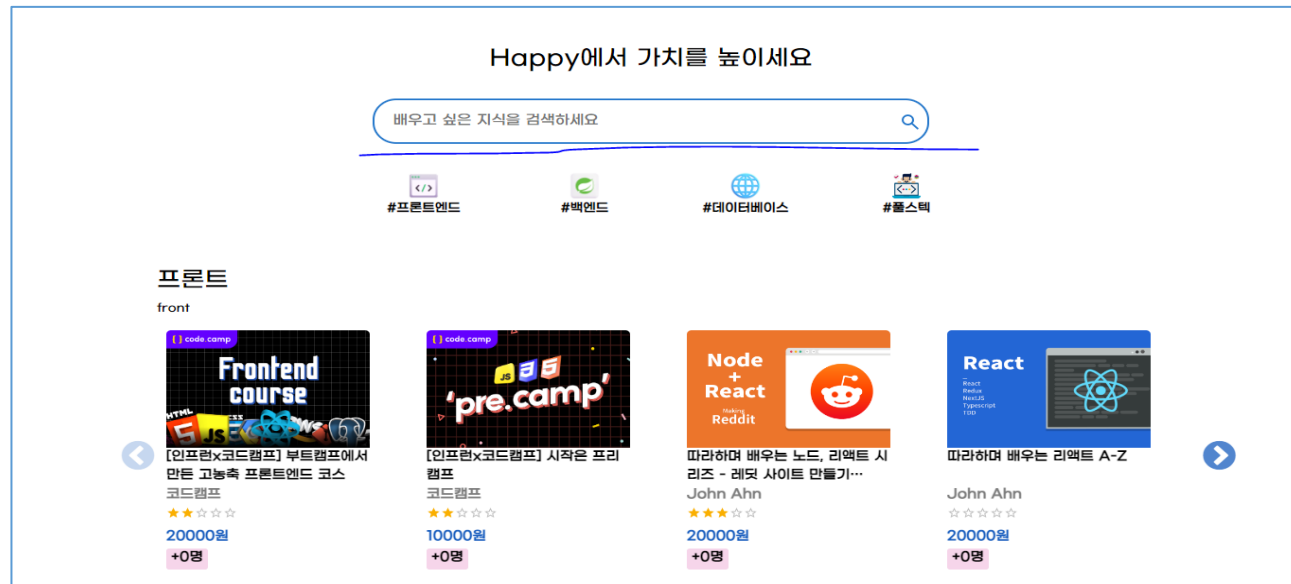
스크롤 기능을 통해 각 과정당 미리보기 강의를 더 찾아 볼 수 있습니다.

```
function Banner() {  
  return [  
    <div className="swipe_container4">  
      <Carousel>  
        <Carousel.Item interval={3000}> ...  
      </Carousel.Item>  
        <Carousel.Item interval={3000}> ...  
      </Carousel.Item>  
        <Carousel.Item interval={3000}> ...  
      </Carousel.Item>  
        <Carousel.Item interval={3000}> ...  
      </Carousel.Item>  
      </Carousel>  
    </div>  
  ]  
}
```

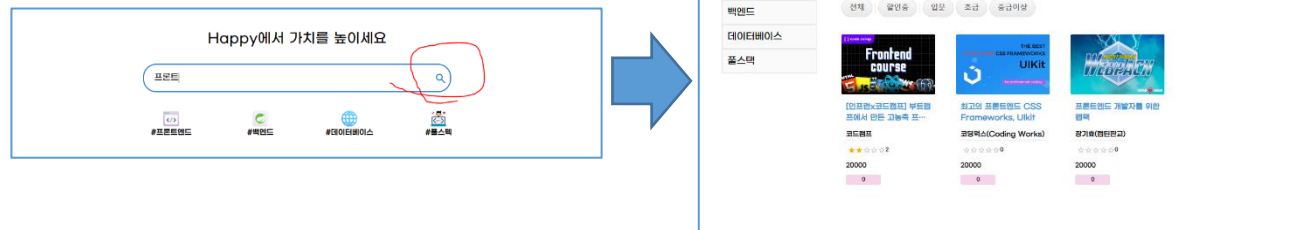
4. 그 외-홈페이지 주요기능[메인페이지]

```
// 검색창에서 검색
const getDataList = () => {
  if (search.trim().length !== 0) {
    navigate(`/list/${search}`);
  }
};
```

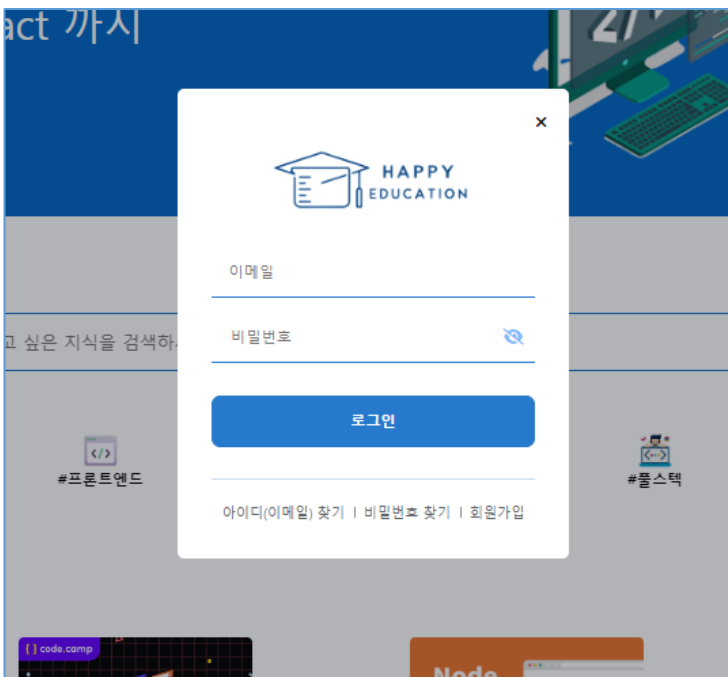
```
{/* 검색 */}
<div className="search_container4">
  <div className="search_container_content4">
    <h3>Happy에서 가치를 높이세요</h3>
    <div className="search_wrapper4">
      <input
        type="text"
        className="m_search_input4"
        maxLength={20}
        placeholder="배우고 싶은 지식을 검색하세요"
        onChange={(e) => setSearch(e.target.value)}
        onKeyDown={handleKeyPress}
      />
      <button ...
      </button>
    </div>
  </div>
</div>
```



검색 기능을 통해 특정 키워드가 포함된 강의를 전부 찾을 수 있습니다.

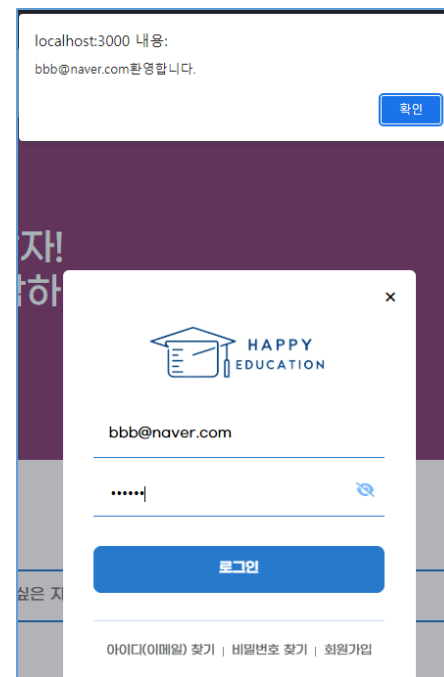


4. 그 외-홈페이지 주요기능[로그인]



```
return [
  <>
    <div className={open ? 'modal-background' : 'modal-hide'}></div>
    <div className={open ? 'modal' : 'modal-hide'}>
      <div className="sign-in-modal">
        <div className="closeButton">
          <div className="header-close-button" onClick={() => handleClose0}>
            <i className="fa-solid fa-xmark"></i>
          </div>
        </div>
        <div className="header--logo">
          <img className="icon-brand-logo" src={mainIcon} alt="메인 아이콘" />
        </div>

        <div className="sign-in-modal--form"> ...
      </div>
      <p className="sign-in-modal_more-action">
        <a href="/findID" className="more-action__text">
          아이디(이메일) 찾기
        </a>
        <span className="more-action-divider"></span>
        <a href="/findPassword" className="more-action__text">
          비밀번호 찾기
        </a>
        <span className="more-action-divider"></span>
        <a href="/signUp" className="more-action__text">
          회원가입
        </a>
      </p>
      <div className="sign-in-modal__social-sign-in">
    </div>
  </div>
</>
]
```



```
const signInHandler = async () => {
  if (userEmail.length === 0 || userPassword.length === 0) {
    alert('이메일과 비밀번호를 입력하세요. ');
    return;
  }
  const data = { ...
};

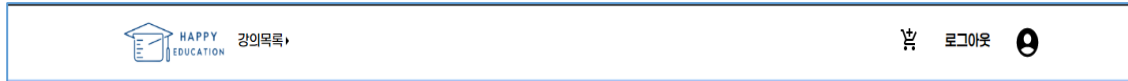
const signInResponse = await signInApi(data);

if (!signInResponse) { ...
}
if (!signInResponse.status) { ...
}
alert(data.userId + ' 환영합니다. ');
const { token, exprTime, user } = signInResponse.data;
const expires = new Date();
expires.setMilliseconds(expires.setMilliseconds + exprTime);

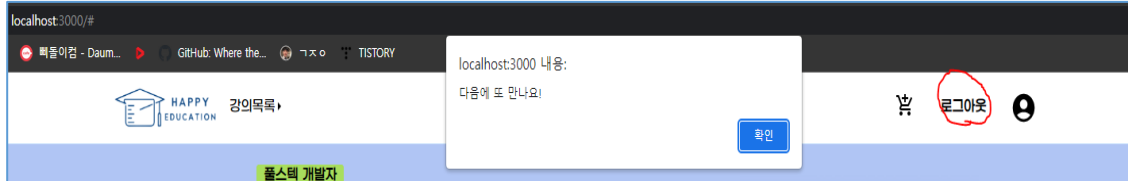
setCookies('token', token, { expires });
setUser(user);
setOpen(false);
};
```


4. 그 외-홈페이지 주요기능[로그인]

로그인 성공 시 상단바



로그아웃 기능 포함



로그인 성공 및 실패 코드

```
const signInHandler = async () => {
  if (userEmail.length === 0 || userPassword.length === 0) {
    alert('이메일과 비밀번호를 입력하세요. ');
    return;
  }
  const data = {
    userId: userEmail,
    password: userPassword,
  };

  const signInResponse = await signInApi(data);

  if (!signInResponse) {
    alert('로그인에 실패했습니다. ');
    return;
  }
  if (!signInResponse.status) {
    console.log(signInResponse);
    alert('로그인에 실패했습니다. ');
    return;
  }
  alert(data.userId + '환영합니다. ');
  const { token, exprTime, user } = signInResponse.data;
  const expires = new Date();
  expires.setMilliseconds(expires.setMilliseconds + exprTime);

  setCookies('token', token, { expires });
  setUser(user);
  setOpen(false);
};
```

4. 그 외-홈페이지 주요기능[회원가입]

회원가입

이메일

example@habby.com

비밀번호

비밀번호

비밀번호 확인

비밀번호 확인

이름

전화번호

회원 가입

회원 가입 성공

localhost:3000 내용:
회원가입이 되었습니다!

확인

회원가입

이메일

ccc@naver.com

비밀번호

.....

비밀번호 확인

.....

이름

ccc

전화번호

010-2222-3333

회원 가입

회원 가입 실패

localhost:3000 내용:
회원가입에 실패했습니다.

확인

회원가입

이메일

ddd

이메일 형식이 올바르지 않습니다.

비밀번호

비밀번호

비밀번호 확인

비밀번호 확인

이름

전화번호

-없이 숫자만 입력하세요

회원 가입

```
const signupHandler = async () => {  
  const data = {  
    userId: userEmail,  
    password: userPassword,  
    passwordConfirm: userPasswordConfirm,  
    name: userName,  
    telNum: userPhone,  
  };  
  
  const signUpResponse = await signUpApi(data);  
  
  if (!signUpResponse) {  
    alert('회원가입에 실패했습니다.');    return;  
  }  
  if (signUpResponse.status) {  
    console.log(signUpResponse);  
    alert('회원가입에 실패했습니다.');    return;  
  }  
  alert('회원가입이 되었습니다!');};  
  
const [passwordView, setPasswordView] = useState<string>('password');  
const [open, setOpen] = React.useState(true);  
const handleOpen = () => setOpen(true);  
const handleClose = () => setOpen(false);  
  
const changePasswordView = () => {  
  if (passwordView === 'password') setPasswordView('text');  
  if (passwordView === 'text') setPasswordView('password');  
};
```

4. 그 외-홈페이지 주요기능[장바구니]

HAPPY
EDUCATION 강의목록

휴

로그아웃

수강바구니

전체선택

선택삭제 X

Frontend
course

[인프런x코드캠프] 부트캠프에서 만든 고난도 프론트엔드 코스

코드캠프

20000원

구매자정보

이름aaaa

이메일aaa@naver.com

전화번호10123456

선택상품 금액0원

할인금액0원

총 결제금액0원

결제하기

```
// 전체 선택
const handleCartCheckAllHandler = (checked: boolean) => {
  let num: number[] = [];
  const idArray: any[] = [];
  let num2 = 0;

  itemList.forEach((el: any) => idArray.push(el.idCart));
  itemList.forEach((el: any) => num.push(el.price));

  num.map((number) => {
    num2 += number;
  });

  console.log(num2);

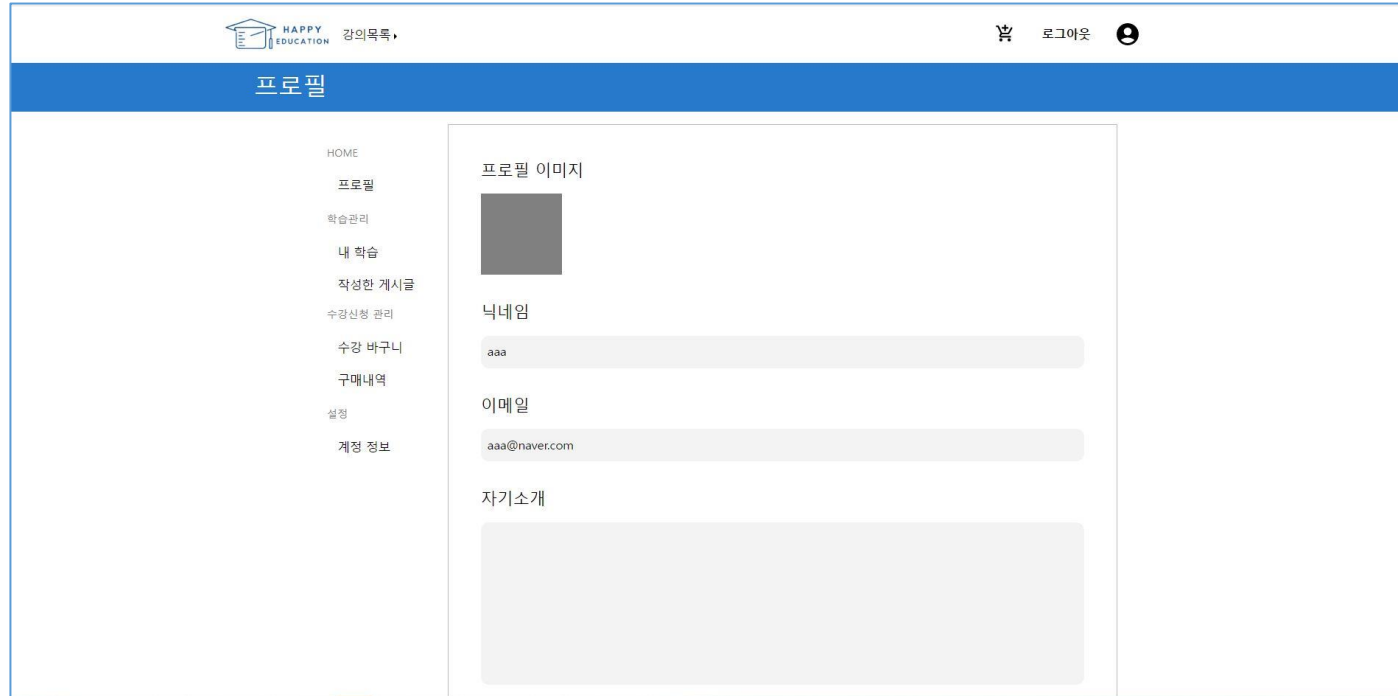
  if (checked) {
    setCheckValue(idArray);
    plusPriceFucntion(num2);
  } else {
    setCheckValue([]);
    setPriceSum(0);
  }
};
```

```

{itemList.map((item) => (
  <div className="cart-list2">
    <input
      type="checkbox"
      value={JSON.stringify(item)}
      onChange={(e) => {
        cartCheckHandler(e.currentTarget.value);
      }}
      checked={checkValue.includes(item.idCart) ? true : false}
      className="course-select2"
    />
    <div className="cart-course-img2">
      <img src={item.img} alt="course-img" />
    </div>
    <div className="cart-course-info2">
      <h3 className="course-title2">
        <a onClick={() => gotoPage(item.idClass)}>{item.className}</a>
      </h3>
      <div>
        <span>{item.instructor}</span>
      </div>
    </div>
    <div className="close2">
      <button className="close-btn2">
        <i className="fa-solid fa-xmark2"></i>
      </button>
    </div>
    <div className="payment2">{item.price}원</div>
  </div>
))

```

4. 그 외-홈페이지 주요기능[마이페이지]

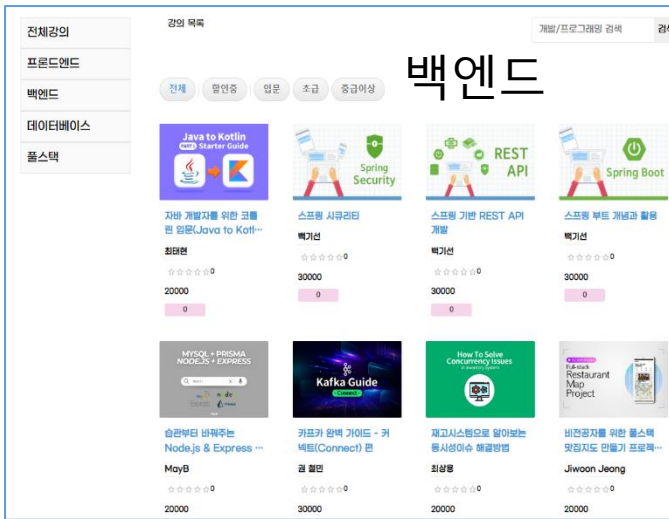
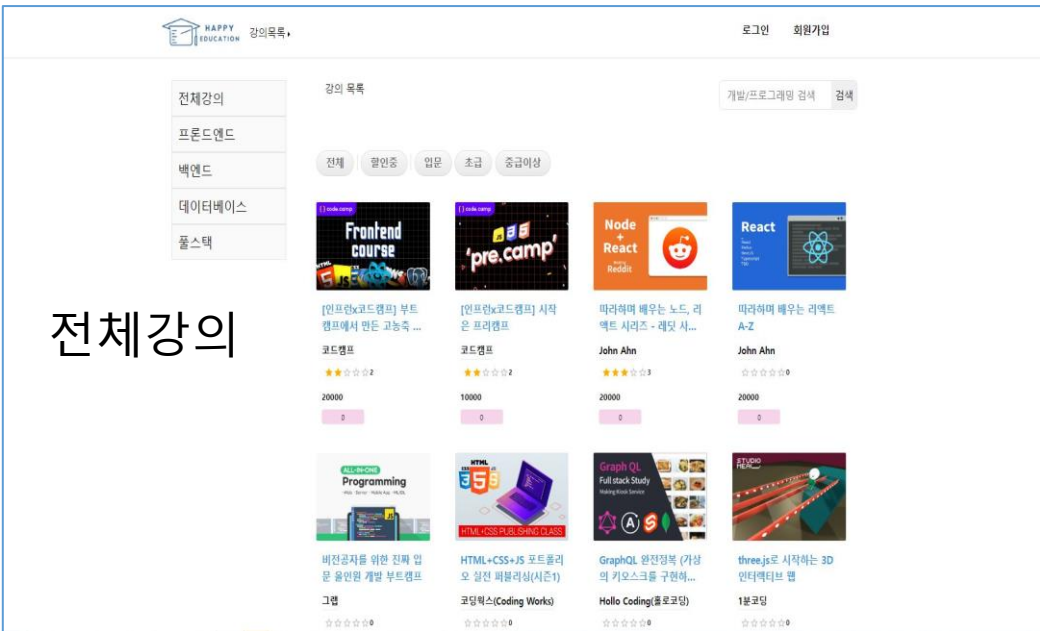


왼쪽 사이드메뉴를 통해 마이페이지에 속해있는 콘텐츠별로 이용할 수 있습니다.

```
<SideBar />
<div className="main-menu1">
  <div className="list1">
    <div className="list01">
      <div className="list11">프로필 이미지</div>
      <div className="list21">
        <div className="image1">
          <img
            src={apiUrl + userProfile?.at(0).profile}
            alt=""
            className="img1"
          />
        </div>
      </div>
    </div>
    <div className="list01">
      <div className="list11">닉네임</div>
      <div className="list31">{userProfile?.at(0).nickname}</div>
    </div>
    <div className="list01">
      <div className="list11">이메일</div>
      <div className="list31">{userProfile?.at(0).email}</div>
    </div>
    <div className="list01">
      <div className="list11">자기소개</div>
      <div className="introduce1">
        {userProfile?.at(0).description}
      </div>
    </div>
  </div>
</div>
```

4. 그 외-홈페이지 주요기능[강의목록]

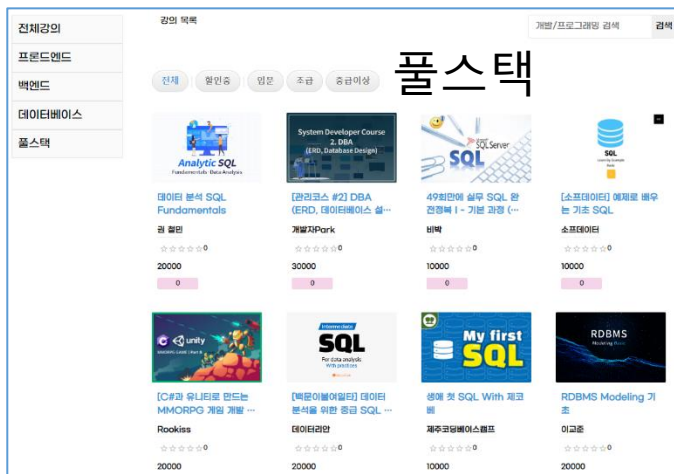
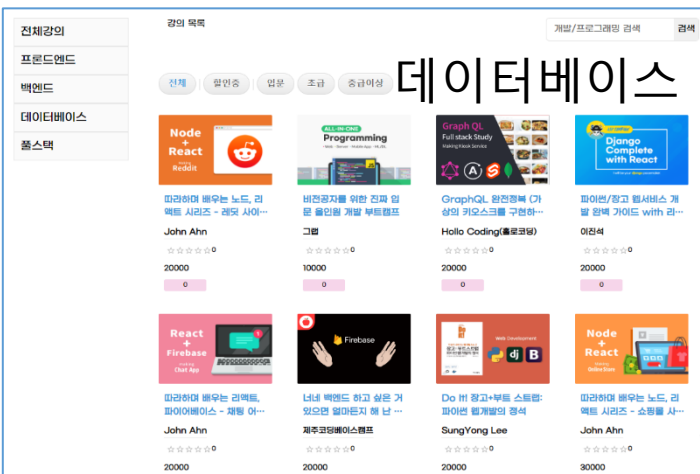
전체강의



```
const allCategoryHandler = (id: string) => {
  setItemList([]);
  setgrade(0);

  axios
    .get('http://localhost:4040/api/list/show/all')
    .then((Response) => {
      const tmp = [];
      setRequestResult('success!');

      for (let i = 0; i < Response.data.data.length; i++) {
        tmp.push({
          idClass: Response.data.data[i].idClass,
          img: Response.data.data[i].img,
          className: Response.data.data[i].className,
          instructor: Response.data.data[i].instructor,
          price: Response.data.data[i].price,
          discountRateRate: Response.data.data[i].discountRateRate,
          studentCount: Response.data.data[i].studentCount,
          grade: Response.data.data[i].grade,
        });
      }
      setItemList(tmp);
      setCategory('all');
      setClickMenu(id);
    })
    .catch((error) => {
      setRequestResult('Failed!');
    });
}
```



5. 후기

아쉬운 점 – 프로젝트를 진행하는 시간이 부족하여 강사별 강의 항목을 보여주는 페이지와 강사 소개 페이지를 구현하지 못하였습니다.

추후 업데이트를 진행하여 강사 소개 페이지와 강사별 강의 항목을 보여주는 페이지를 구현해볼 예정입니다.

힘들었던 점 – 처음에는 JavaScript를 사용하여 React프로젝트를 제작하다가 중간 회의 결과 사용하는 언어를 TypeScript로 변환하여 제작에 들어간 것이 힘들었습니다.

TypeScript 에 대해 모르던 저는 처음에는 당황하였으나, 이내 정신을 차리고 TypeScript가 무엇인지 검색을 통해 확인하였고, 사용하는 방법에 대해 공부한 뒤 기존에 작업하던 코드를 전부 TypeScript 형식으로 변환하는 데에 성공했습니다.