

이커머스 웹 사이트 땡켓 제작

지도 교수 : 이병천 교수님

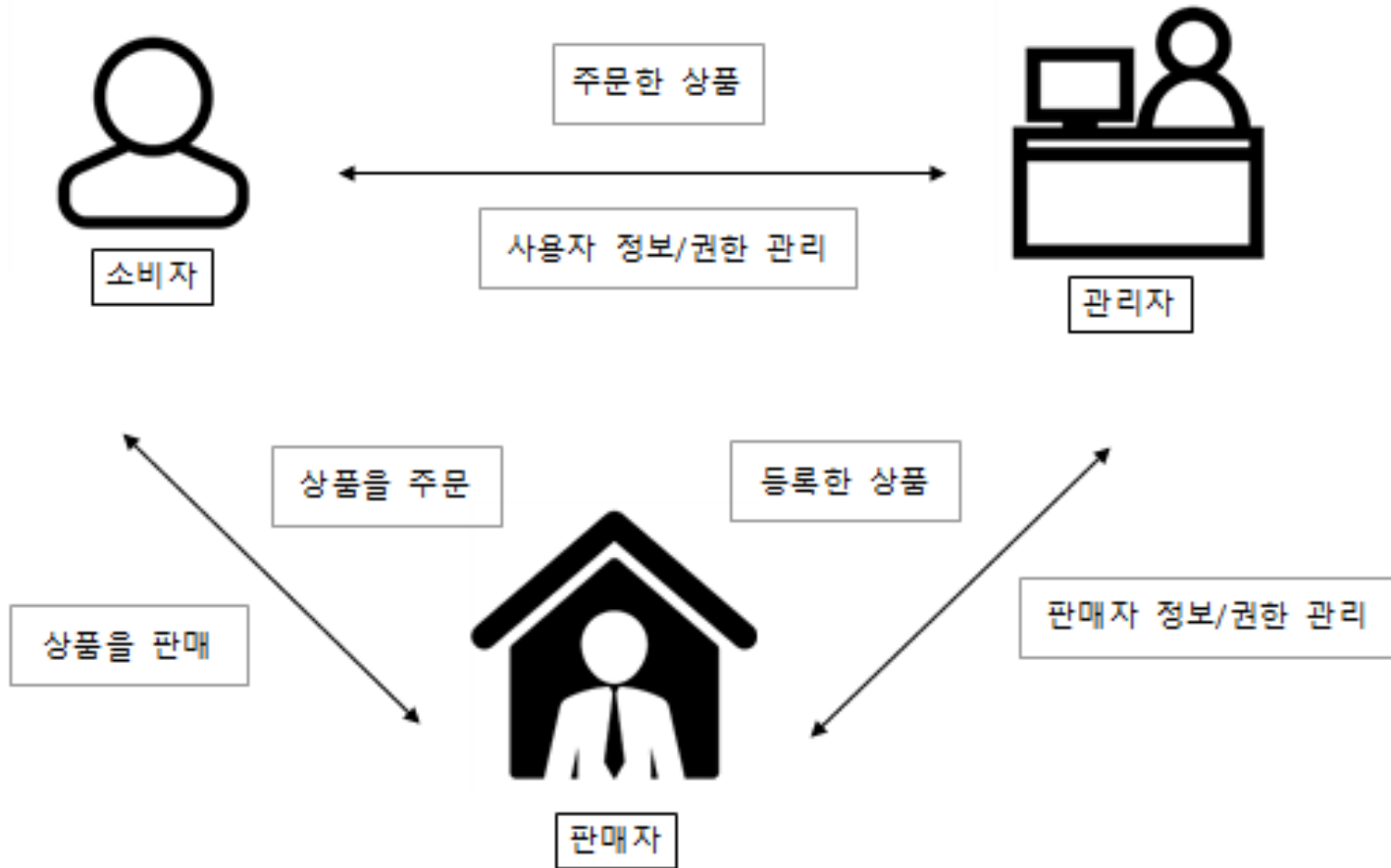
팀장 : 91812816 이다원

팀원 : 91812804 이나원

목차

- **구상도**
- **개발 환경 및 개발 내용**
- **개발 시스템 운영**
- **결론 및 기대효과**

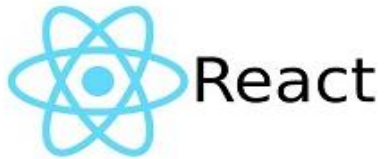
구상도



개발 환경 및 개발 내용

개발 환경

Frontend



Backend



OS



DB



Deploy



개발 환경 및 개발 내용

개발 내용(1/6)

상품 DB

thinket.products

Documents

thinket.products

DOCUMENTS 10 TOTAL SIZE 2.9KB AVG. SIZE 297B INDEXES 2 TOTAL SIZE 64.0KB AVG. SIZE 32.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ↺ ⋮

ADD DATA VIEW ()

Displaying documents 1 - 10 of 10 < > REFRESH

```
1  _id: ObjectId("60780b6e9a153b1bec07da0e")
2  name: "팬티"
3  category: "Pants"
4  image: "/images/product-1.jpg"
5  price: 20000
6  countInStock: 10
7  brand: "Thinket"
8  rating: 5
9  numReviews: 15
10 description: "high quality pants"
11 __v: 0
12 createdAt: 2021-04-15T09:46:22.969+00:00
13 updatedAt: 2021-05-20T09:09:48.206+00:00
14 seller: ObjectId("60778b23eab87d3948211099")
```

ObjectID
String
String
Int32
String
Int32
Int32
String
Int32
Date
Date
ObjectID

CANCEL UPDATE

```
_id: ObjectId("60a62419affc8d170cb5094d")
name: "꽃 자수 월간우스"
image: "/uploads/1621500965430.jpg"
price: 18000
category: "Blouse"
```

```
const productSchema = new mongoose.Schema({
  name: {type: String, required: true, unique: true},
  seller: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
  category: {type: String, required: true},
  image: {type: String, required: true},
  price: {type: Number, required: true},
  countInStock: {type: Number, required: true},
  brand: {type: String, required: true},
  rating: {type: Number, required: true},
  numReviews: {type: Number, required: true},
  description: {type: String, required: true},
  createdAt: {type: Date, required: true},
  updatedAt: {type: Date, required: true},
  seller: {type: mongoose.Schema.Types.ObjectId, ref: 'User'}
});

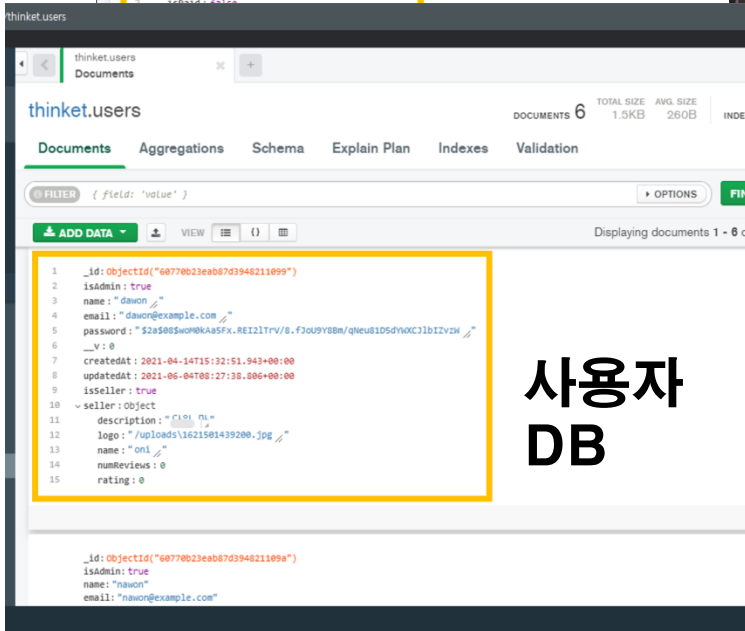
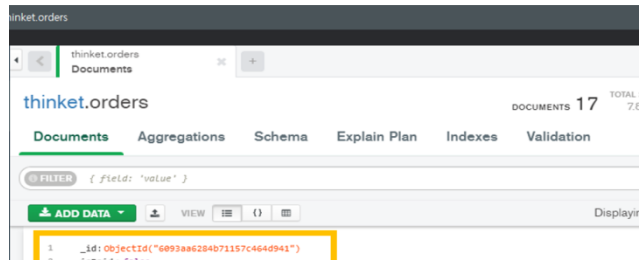
const reviewSchema = new mongoose.Schema({
  product: {type: mongoose.Schema.Types.ObjectId, ref: 'Product', required: true},
  reviewer: {type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true},
  review: {type: String, required: true},
  rating: {type: Number, required: true},
  createdAt: {type: Date, required: true},
  updatedAt: {type: Date, required: true}
});
```

상품 DB 스키마 작성

```
mongoose.model('Product', productSchema);
mongoose.model('Review', reviewSchema);
```

개발 환경 및 개발 내용

개발 내용(2/6)



```
const orderSchema = new mongoose.Schema({
  orderItems : [{
    name: {type: String, required: true},
    qty: {type: Number, required: true},
    image: {type: String, required: true},
    price: {type: Number, required: true}
  ]
});
```

```
const userSchema = new mongoose.Schema(
{
  name: {type: String, required: true},
  email: {type: String, required: true, unique: true},
  password: {type: String, required: true},
  isAdmin: {type: Boolean, default: false, required: true},
  isSeller: {type: Boolean, default: false, required: true},
  seller: {
    name: String,
    logo: String,
    description: String,
    rating: {type: Number, default:0, required: true},
    numReviews: {type: Number, default:0, required: true},
  },
},
{
  timestamps: true,
});
```

개발 환경 및 개발 내용

개발 내용(3/6)

Router (product)

```
productRouter.get('/:id', expressAsyncHandler(async(req, res) => {
  const product = await Product.findById(req.params.id).populate('seller', 'seller.name seller.logo seller.rating
  seller.numReviews');
  if(product){
    res.send(product);
  }
  else{
    res.status(404).send({message: 'Product Not Found'});
  }
}));

productRouter.post('/', isAuth, isSellerOrAdmin, expressAsyncHandler(async(req, res) => {
  const product = new Product({
    name: 'sample name' + Date.now(),
    seller: req.user._id,
    image: '/images/product-1.jpg',
    price: 0,
    category: 'sample category',
    brand: 'sample brand',
    countInStock: 0,
    rating : 0,
    numReviews: 0,
    description: 'sample description'
  });
  const createdProduct = await product.save();
  res.send({message: 'Product Created', product: createdProduct});
}));
```

특정 요청에 따른 기능을 지정

개발 환경 및 개발 내용

개발 내용(4/6)

메인 페이지 작성

```
return (  
  <div>  
    <h2>베스트 셀러</h2>  
    {loadingSellers ? <LoadingBox></LoadingBox>  
    :  
    errorSellers ? <MessageBox variant="danger">{errorSellers}</MessageBox>  
    :  
    <>  
      {sellers.length === 0 && <MessageBox>No Seller Found</MessageBox>}  
      <Carousel showArrows autoPlay showThumbs={false}>  
        {sellers.map((seller) => {  
          <div key={seller._id}>  
            <Link to={` /seller/${seller._id}`}>  
              <img src={seller.seller.logo} alt={seller.seller.name}></img>  
              <p className="legend">{seller.seller.name}</p>  
            </Link>  
          </div>  
        })}  
      </Carousel>  
    </>  
  }  
  <h2>전체 상품</h2>  
  {loading? <LoadingBox></LoadingBox>  
  :  
  error?<MessageBox variant="danger">{error}</MessageBox>  
  :  
  <>  
    {products.length === 0 && <MessageBox>No Product Found</MessageBox>}  
    <div className="row-sorting center">  
      {  
        products.map((product) => (  
          <Product key={product._id} product={product} />  
        )  
      )  
    }  
  }  
  </div>  
)
```

베스트 셀러와 전체 상품
보여주는 메인 페이지 개발

개발 환경 및 개발 내용

개발 내용(5/6)

Action, Reducer

```
export const createProduct = () => async(dispatch, getState) => {
  dispatch({type: PRODUCT_CREATE_REQUEST});
  const {userSignin: {userInfo}} = getState();
  try {
    const {data} = await Axios.post('/api/products', {}, {
      headers: {Authorization: `Bearer ${userInfo.token}`}
    });
```

state의 변화에
대한 동작 수행

```
export const productCreateReducer = (state = {}, action) => {
  switch (action.type) {
    case PRODUCT_CREATE_REQUEST:
      return {loading: true};
    case PRODUCT_CREATE_SUCCESS:
      return {loading: false, success: true, product: action.payload};
    case PRODUCT_CREATE_FAIL:
      return {loading: false, error: action.payload};
    case PRODUCT_CREATE_RESET:
      return {};
    default:
      return state;
  }
};
```

```
load: data.product});
```

```
sage ? error.response.data.message : error.message;
d: message});
```

새로운 상태를 반환해주는 reducer

개발 환경 및 개발 내용

개발 내용(6/6)

Component 생성

```
export default function SearchBox(props) {  
  const [name, setName] = useState('');  
  const submitHandler = (e) => {  
    e.preventDefault();  
    props.history.push(`/search/name/${name}`);  
  }  
}
```

```
return (  
  <div>  
    { /* Search Form */}  
    <form className="search" onSubmit={submitHandler}>  
      <div className="row-sorting">  
        <input type="text" onChange={(e) => setName(e.target.value)}></input>  
        <button type="submit" className="primary">  
          <i className="fa fa-search"></i>  
        </button>  
      </div>  
    </form>  
  </div>  
)
```

SearchBox component

```
export default function Product(props) {  
  const {product} = props;  
  return (  
    <div key={product._id} className="product-card">  
      <Link to={` /product/${product._id}`}>  
        { /* Product Image */}  
        <img className="product-image" src={product.image} alt={product.name}/>  
      </Link>  
      { /* Card Content */}
```

```
    <div>  
      <h3>Product</h3>  
      <p>Product ID: {product._id}</p>  
      <p>Product Name: {product.name}</p>  
      <p>Product Price: {product.price}</p>  
      <p>Product Rating: {product.rating} numReviews={product.numReviews} />
```

```
    </div>  
  </div>  
)
```

Product component

```
    <div>  
      <h3>Product</h3>  
      <p>Product ID: {product._id}</p>  
      <p>Product Name: {product.name}</p>  
      <p>Product Price: {product.price}</p>  
      <p>Product Rating: {product.rating} numReviews={product.numReviews} />
```

개발 시스템 운영(1/8)

메인 페이지

The screenshot shows the main page of a web application named 'Thinket'. The header is green and contains a menu icon, the brand name 'Thinket', a search bar, and navigation links for 'Cart', '이달원', 'Seller', and 'Admin'. Below the header, there is a '베스트 셀러' (Best Seller) section featuring a large image of a person wearing a black t-shirt with the text 'NO PARTS GANG'. A light green horizontal bar is positioned below this image. The '전체 상품' (All Products) section displays four product cards: '흑청바지' (Dark Blue Jeans) with a 1 review, '청바지(3color)' (Light Blue Jeans), '청자켓' (Light Blue Jacket), and '화이트 후디' (White Hoodie).

Thinket

Cart 이달원 Seller Admin

베스트 셀러

1 / 2

전체 상품

흑청바지
1 review

청바지(3color)

청자켓

화이트 후디

개발 시스템 운영(2/8)

상품 검색 / 정렬 / 필터링

Thinket

검색결과 : 1개

정렬하기

카테고리

- 모든상품
- Blouse
- Cardigan
- Dress
- Knit
- Outer
- Pants
- Skirt
- Tshirts**
- pants

가격

- 모든 가격
- 1만원 이하
- 1만원 - 5만원
- 5만원 - 10만원

평균 별점

- ★★★★☆ & 이상
- ★★★★☆ & 이상
- ★★★★☆ & 이상

기본 흰 티

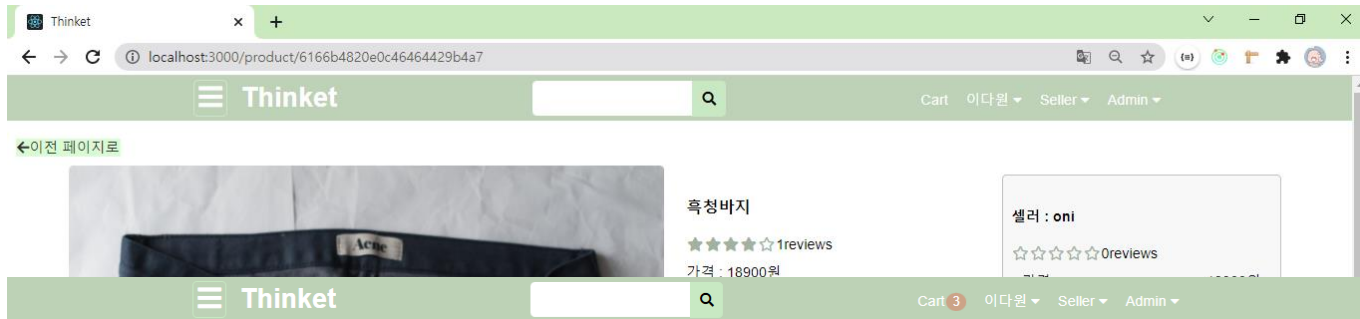
☆☆☆☆☆ 0reviews

8900원

1

개발 시스템 운영(3/8)

상품 상세 페이지 / 장바구니 페이지



개발 시스템 운영(4/8)

주문 과정(1/2)

The image displays two sequential screenshots of the Thinket checkout process. The top screenshot shows the '배송지 주소' (Shipping Address) step, where the user's name '이다원' (Ida-won) is entered in the '이름' (Name) field. The bottom screenshot shows the '결제 수단' (Payment Method) step, where 'PayPal' is selected as the payment method. Both screenshots feature a green header with the Thinket logo, a search bar, and navigation links for '로그인' (Login), '배송지' (Shipping Address), '결제수단' (Payment Method), and '주문확인' (Order Confirmation). The header also includes a 'Cart 3' icon and dropdown menus for '이다원', 'Seller', and 'Admin'. The footer of the page contains the text 'All right reserved'.

Thinket

로그인 배송지 결제수단 주문확인

배송지 주소

이름
이다원

Thinket

로그인 배송지 결제수단 주문확인

결제 수단

PayPal
 Stripe

결제 진행 하기

All right reserved

개발 시스템 운영(5/8)

주문 과정(2/2)

Thinket Cart 3 이다원 Seller Admin

로그인 배송지 결제수단 주문확인


배송지 결제 금액 합계

Thinket Cart nawon Seller Admin

주문 번호 60a4f7ebdde1ad6ea4a4894e

배송지
이름: 이나원
주소: 경기도 고양시, 일산서구, 2222, 덩켓아파트 101동 102호
배송 전

결제
결제 수단: PayPal
결제 전

주문 상품
 Denim Pants 7 x 20000원 = 140000원

결제 금액 합계
상품 금액 140000원
배송비 0원
총 결제 금액 140000원

[PayPal](#)

직불카드 또는 신용카드

계좌 PayPal

All right reserved

개발 시스템 운영(6/8)

주문 내역 확인

Thinket

주문번호

60bf5fec0177dd68d081b095

60c3427a57d7d44580185519

61704d8af3bdcc78d47c3c5b

Thinket

Cart 이다원 Seller Admin

주문내역

판매자

주문번호	고객	주문날짜	총 금액	결제여부	배송여부	상세보기
60bf5fec0177dd68d081b095	고구마	2021-06-08	17000	결제 전	배송 전	상세정보 삭제
60c2f38b33248a44bc5247a3	이다원	2021-06-11	32000	결제 전	배송 전	상세정보 삭제
60c3427a57d7d44580185519	고구마	2021-06-11	16000	결제 전	배송 전	상세정보 삭제
612df1ac3e83cb2a50efff90	이다원	2021-08-31	20000	결제 전	배송 전	상세정보 삭제
612df1dc3e83cb2a50efff92	이다원	2021-08-31	69000	결제 전	배송 전	상세정보 삭제
613992e4dfd3b3711833c8c2	이다원	2021-09-09	80000	결제 전	배송 전	상세정보 삭제
613da67b5d8b9354703c7dd4	이다원	2021-09-12	20000	결제 전	배송 전	상세정보 삭제

Thinket

Cart 이다원 Seller Admin

주문내역

관리자

주문번호	고객	주문날짜	총 금액	결제여부	배송여부	상세보기
60940082f4153050e07acc7e	이다원	2021-05-06	80000	결제 전	배송 전	상세정보 삭제
609402acbc59c6254c439b9d	이다원	2021-05-06	40000	결제 전	배송 전	상세정보 삭제
6094bdf641a80e5ac482dbaa	이다원	2021-05-07	20000	결제 전	배송 전	상세정보 삭제
60a477ebdde1ad6ea4a4894e	nawon	2021-05-19	140000	결제 전	배송 전	상세정보 삭제
60a73c73affc8d170cb50951	이다원	2021-05-21	48000	결제 전	배송 전	상세정보 삭제
60a78d3bbfa7574c40a56399	이다원	2021-05-21	35000	결제 전	배송 전	상세정보 삭제
60b9fbd83f239749940a6a8b	이다원	2021-06-04	20999	결제 전	배송 전	상세정보 삭제
60ba02b01d40715b8063ea5e	이다원	2021-06-04	20000	결제 전	배송 전	상세정보 삭제
60bf5fec0177dd68d081b095	고구마	2021-06-08	17000	결제 전	배송 전	상세정보 삭제
60c2f38b33248a44bc5247a3	이다원	2021-06-11	32000	결제 전	배송 전	상세정보 삭제
60c3427a57d7d44580185519	고구마	2021-06-11	16000	결제 전	배송 전	상세정보 삭제

개발 시스템 운영(7/8)

판매자 - 상품 목록

Thinket Cart 이다원 Seller Admin

상품 수정 61704bc2f3bdcc78d47c3c58

상품명

상품 가격

상품 이미지

상품 이미지 파일
 선택된 파일 없음

상품 카테고리

상품 브랜드

상품 수량

상품 설명

개발 시스템 운영(8/8)

관리자 - 사용자 관리 / 대시보드

☰ Thinket

Q
Cart 이다원 Seller Admin

사용자 목록

ID	이름	이메일	셀러 여부	관리자 여부	상태
60770b23eab87d3948211099	이다원	dawon@example.com	O	O	수정 삭제
60770b23eab87d394821109a	nawon	nawon@example.com	O	O	수정 삭제
6089795b5155f66cf0dedb55	ho	ho@example.com	X	X	수정 삭제
60897bd75155f66cf0dedb5b	1113	1113@ddd.com	X	X	수정 삭제

☰ Thinket

Q
Cart 이다원 Seller Admin

대시보드

총 사용자 수

8

주문 주문 수

24

총 주문 금액

996299원

판매 현황

- Knit
- Blouse
- Pants
- Tshirts
- Dress
- Cardigan
- Skirt
- Outer
- sample category

All right reserved

결론 및 기대효과

결론

- 이커머스 웹사이트가 가져야 하는 기본적인 기능들을 바탕으로 팀만의 웹 사이트를 개발
- 사이트 이용에 있어서 필수적인 기능들을 사용자, 판매자, 관리자에 맞춰서 개발하는데 성공함

기대효과

- 팀 프로젝트의 목표를 달성하기 위해 함께 노력하며 협동심과 책임감의 필요성과 중요성을 느끼게 된 프로젝트
- 웹사이트를 직접 구상하고 주요 기능들을 구현함으로써 개발 능력 향상의 계기로 마련

Q&A

감사합니다.