

Team project portfolio

OPICK SHOP



Team : 5Ping Mall

Production : 성득명 빙승현 지수정 김태연

Production date : 25.09.22 ~ 25.10.14

Categories

Introduction ^

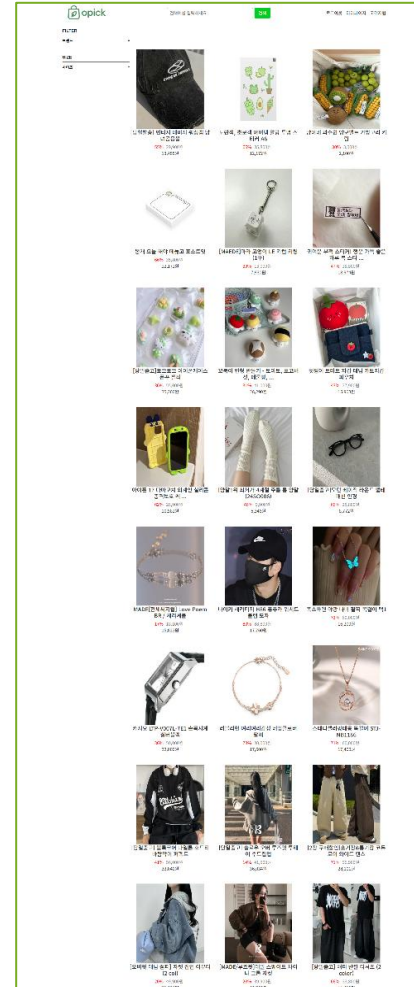
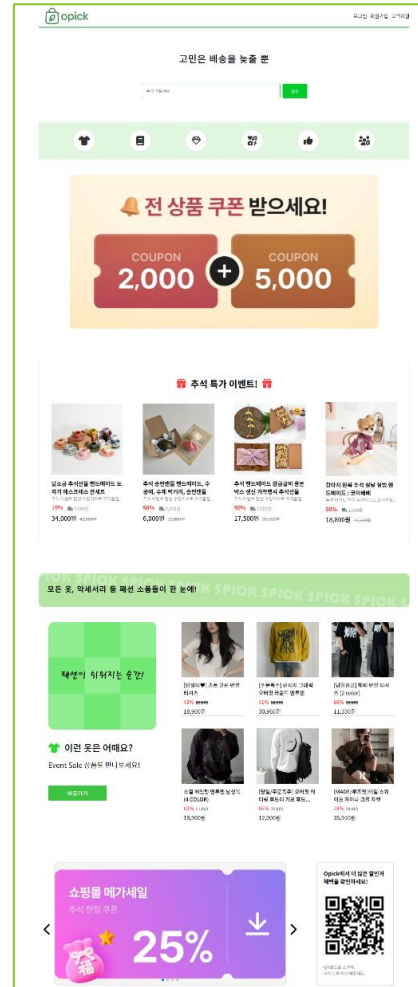
- [Select a Website](#)

Project Plan v

Implementation v

Evaluation v

Finish



Opick shop

- 직접 제작한 본인의 작품을 거래하는 온라인 쇼핑몰
- 심플하지만 함축적으로 필요한 정보들을 담고 있는 레이아웃 형태로 간단히 표현
- 아이디어스, 숨고, 텀블벅 등의 여러 거래 사이트 참고

Categories

Introduction ▾

Project Plan ▲

- Persona

- Prototype

Implementation ▾

Evaluation ▾

Finish



이름 : 윤소연
직업 : 조향사
취미 : 향수 제작

요구 사항

- 본인의 작품 판매 및 다른 작가의 작품 구매를 위한 사이트
- 기업보단 개인 작품 판매의 비중이 큰 사이트 요청



이름 : 이승연
직업 : 도자기 공방 원데이클래스 운영
취미 : 도자기 공예

요구 사항

- 본인의 작품 판매가 쉽게 가능한 사이트
- 카테고리가 명확히 나뉘져있는 사이트 요청

Categories

Introduction

v

Project Plan

^

- Persona

- Prototype

Implementation

v

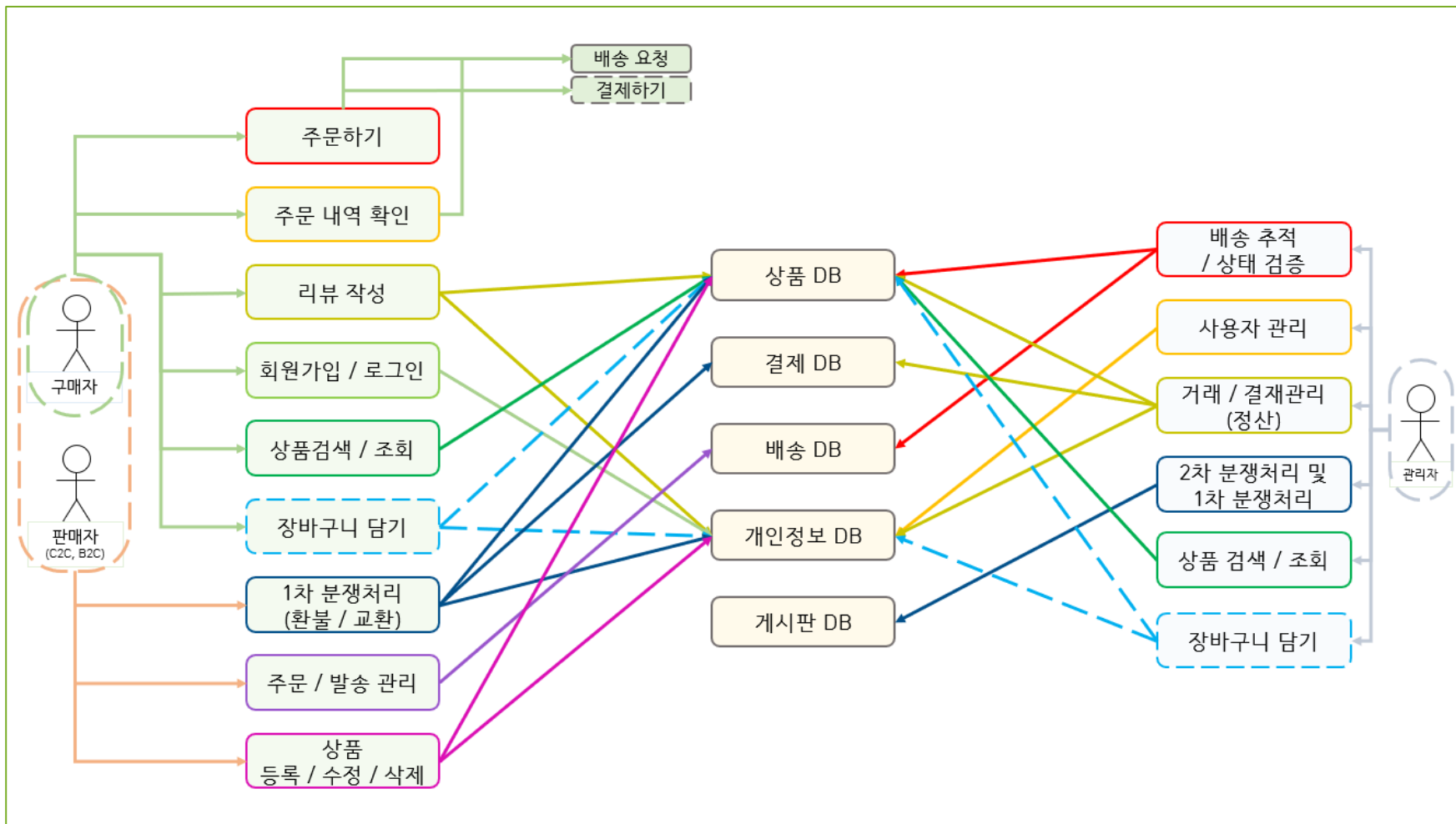
Evaluation

v

Finish

1차 구상 (Diagram)

- UseCase



Categories

Introduction

v

Project Plan

^

- Persona

- Prototype

Implementation

v

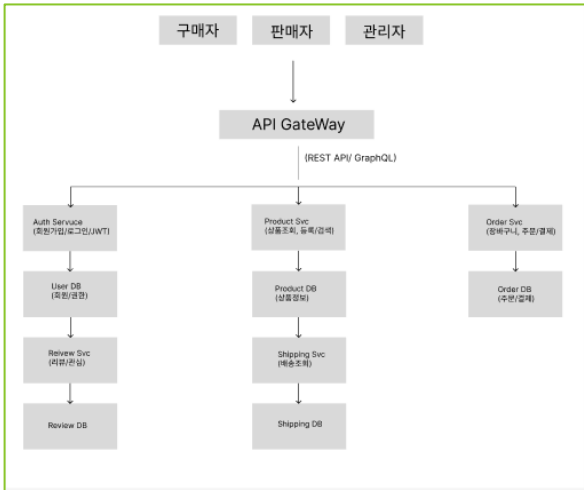
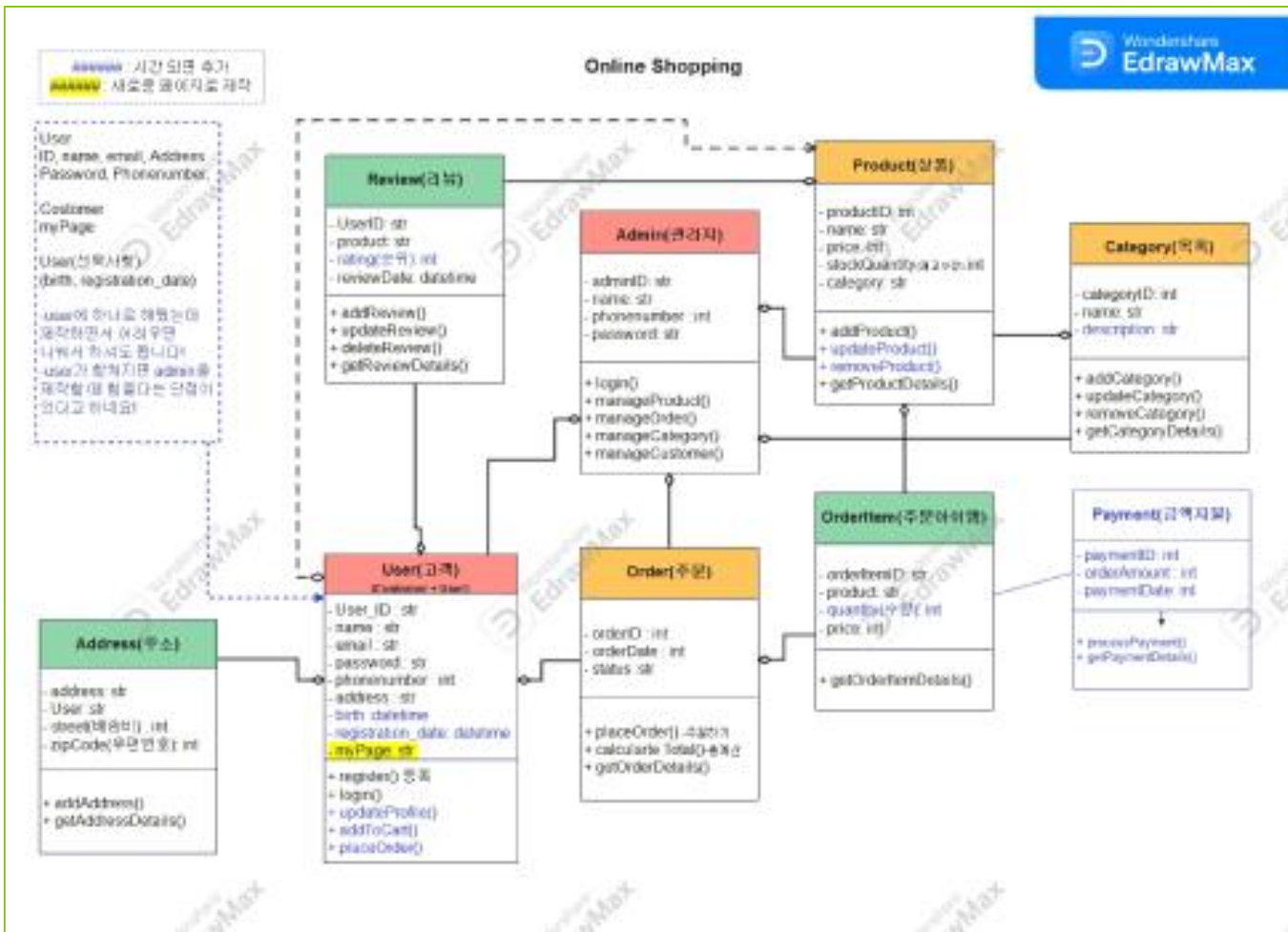
Evaluation

v

Finish

1차 구상 (Diagram)

- Class



Categories

Introduction

v

Project Plan

^

- Persona

- Prototype

Implementation

v

Evaluation

v

Finish

1차 구상 (Diagram)

- other

DB 구상 리스트								
User			Product			Comment		
컬럼명	데이터 타입	nullable	컬럼명	데이터 타입	nullable	컬럼명	데이터 타입	nullable
id	int	지정X	id	int	지정X	id	int	지정X
userid	str	F	title	str	F	content	str	F
password	str	F	content	str	F	create_date	datetime	F
username	str	F	create_date	datetime	F	modify_date	datetime	T
email	str	F	modify_date	datetime	T			
phone	str	F						
address	str	F						
admin	boolean	F						

필수 기능:

1. 회원가입/로그인
2. 상품검색/조회 (기본)
3. 장바구니 담기 (결제 전)
4. 주문하기 (매출 핵심)
5. 주문 내역 확인 (불만방지)
6. 배송조회/관리 (배송DB관련/주문이후 핵심)
7. 결제처리 (거래 성립)
8. 관리자-상품등록/삭제/조회 (상품DB관리/쇼핑몰 운영)
-거래내역 관리(정산)

구매자(User) : 회원가입/로그인 → 상품검색 → 장바구니(?) → 주문 → 결제 → 배송 조회 → 리뷰작성
관리자(Admin): 회원/상품/정산 관리, 분쟁 처리

판매자 : 상품 등록/수정/삭제, 주문관리, 리뷰 확인
관리자에 판매자 기능까지 포함시킬건지...?

선택 기능:

1. 리뷰작성
2. 관심상품 담기 (편의기능)
3. 1차/2차 분쟁처리 (환불/교환)
4. 관리자-사용자 관리 (규모 커질때 필요)
5. 상품 등록 특수(검사(심사))
6. 배송추적 세부 알림/처리 상세 기능
(기본 배송조회만 있으면 됨-옵션)

쇼핑몰

개인 / 관리자

개인 로그인 회원가입 마이페이지 상품등록 댓글
관리자 회원관리

<필수>

기능 구현할 명세서
로그인 회원가입 연동 기능
검색
채팅기능(q&a)

카테고리(상품위주)

-상품 등록 (일자순)

신규/인기 이름, 가격, 추천순 조회(?)순 구현 가능하다면
글자 수 사진 크기 제한

-메인페이지 물건((슬라이더))

고객지원 클릭 시 게시판 이동
질문 → 답변

자주 묻는 질문 FAQ

예상질문 답변 써놓기(게시판 최상단) 디비사용x
공지사항(?)

Categories

Introduction

✓

Project Plan

^

- Persona

- Prototype

Implementation

✓

Evaluation

✓

Finish

2차 기본 코드 작성 및 main, sub 와이어프레임 제작

- Figma, PyCharm, Flask, SQLite, vscode(html, css, javascript) 사용

```

# 보통 1차 설정에 따라 사용함
# 공백, NULL, 불가

# unique = 이미 설정 값이 데이터베이스에 존재하면 안됨
# 이메일, 주민등록번호 같은 데이터에 주로 사용

# nullable = NULL 값을 허용할지 결정할지
# False를 설정시 이미 불가능함
# 허용할 수 있는 데이터에서 사용함

# autoincrement = 증가할 값에서 주로 사용, 세 번호들이 들어갈 때 자동으로 증가함
# 주로 1차항에서 사용함

# onDelete="CASCADE" = 종속성이 있을 때 종속성이 없으면 삭제
# 그날 후엔 오류가 발생할 수 있음

=====
# 유저모델 설정
class User(db.Model):
    """유저 모델"""

    id = db.Column(db.Integer, primary_key=True, autoincrement=True) # db에 설정한 순서로 부여되는 id
    user_id = db.Column(db.String(50), unique=True, nullable=False) # user 가 회원 등록시 작성하는 로그인용 아이디 / 필수입력
    password = db.Column(db.String(200), nullable=False) # 비밀번호 / 필수입력
    username = db.Column(db.String(50), nullable=False) # 닉네임 / 필수입력
    email = db.Column(db.String(100), unique=True, nullable=False) # 이메일 / 필수입력
    phone = db.Column(db.String(20), unique=True, nullable=False) # 전화번호 / 필수입력
    address = db.Column(db.String(200), nullable=False) # 주소 / 필수입력
    is_admin = db.Column(db.Boolean, nullable=False, default=False) # 이진값 여부 / 필수 입력

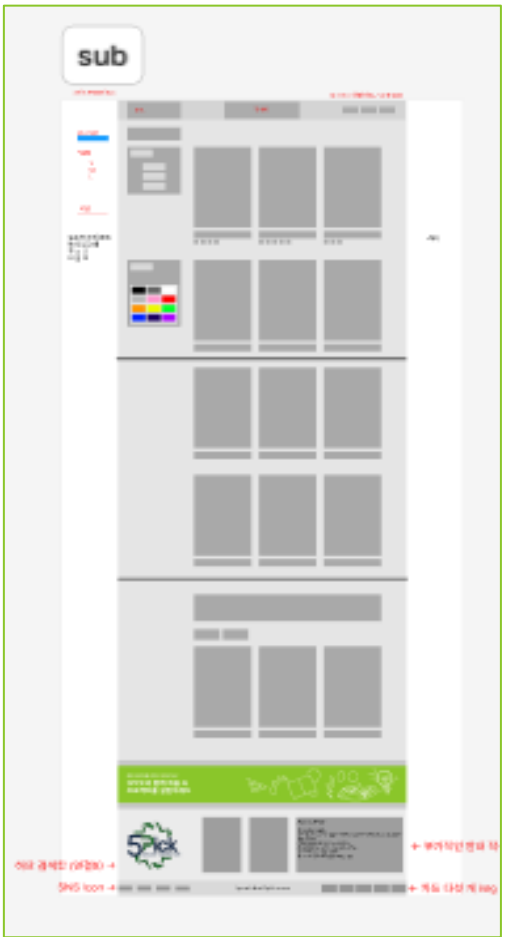
# 상품모델 설정
class Product(db.Model):
    """상품 모델"""

    id = db.Column(db.Integer, primary_key=True, autoincrement=True) # db에 설정한 순서로 부여되는 id
    title = db.Column(db.String(200), nullable=False) # 제목 / 필수입력
    content = db.Column(db.Text(), nullable=False) # 글 내용 / 필수입력
    create_date = db.Column(db.DateTime, nullable=False) # 작성일, 시간 / 자동설정
    modify_date = db.Column(db.DateTime, nullable=True) # 글 수정일 수정하지 않으면

# 작성자를 참조
user_id = db.Column(db.Integer, db.ForeignKey('user_id', ondelete='CASCADE'), nullable=False) # 사용자 아이디 참조
user = db.relationship('user', backref=db.backref('product_set')) # Product에서 user(작성자를) 참조

# 댓글모델 설정
class Comment(db.Model):
    """댓글"""

    id = db.Column(db.Integer, primary_key=True, autoincrement=True) # db에 설정한 순서로 부여되는 id
    content = db.Column(db.Text(), nullable=False) # 댓글 내용 / 필수입력
    create_date = db.Column(db.DateTime, nullable=False) # 댓글 시간 / 자동설정
    modify_date = db.Column(db.DateTime, nullable=True) # 댓글 수정일 시간 설정
    product_id = db.Column(db.Integer, db.ForeignKey('product_id', ondelete='CASCADE')) # 상품 글에 속해있는 상품의 id를 참조
    product = db.relationship('Product', backref=db.backref('comment_set')) # 댓글 글이 속해있는 상품의 id를 참조
    user_id = db.Column(db.Integer, db.ForeignKey('user_id', ondelete='CASCADE'), nullable=False) # 댓글을 작성한 사람의 id를 참조
    user = db.relationship('user', backref=db.backref('comment_set')) # 댓글 작성자를 참조
    
```



Categories

Introduction

v

Project Plan

^

- Persona

- Prototype

Implementation

v

Evaluation

v

Finish

3차 로고 디자인 및 sign in, sign up, my page 와이어프레임 제작



최종



Categories

Introduction

v

Project Plan

v

Implementation

^

- Main page

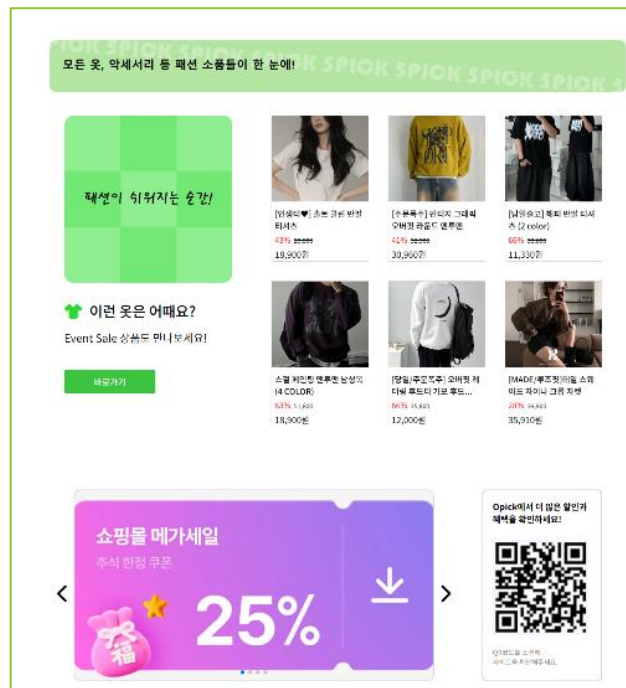
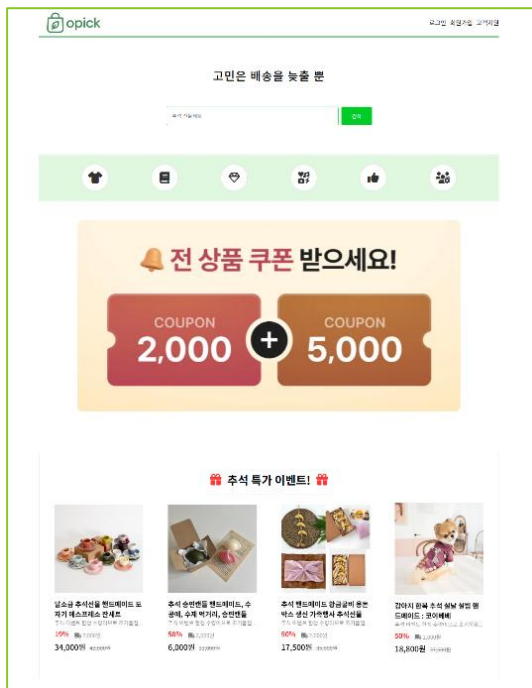
- Sign in / Sign up

- Sub page

Evaluation

v

Finish



Header - 로고, 로그인/회원가입/고객지원(-> 로그아웃/마이페이지/고객지원), 카테고리

Slider - 메인 슬라이드

Section - 이벤트, 카테고리 각각의 상품, 쿠폰팩 등의 기능과 자주 묻는 질문

Footer - 페이지 설명과 회사 소개

Categories

Introduction

V

Project Plan

V

Implementation

 \wedge

[- Main page](#)

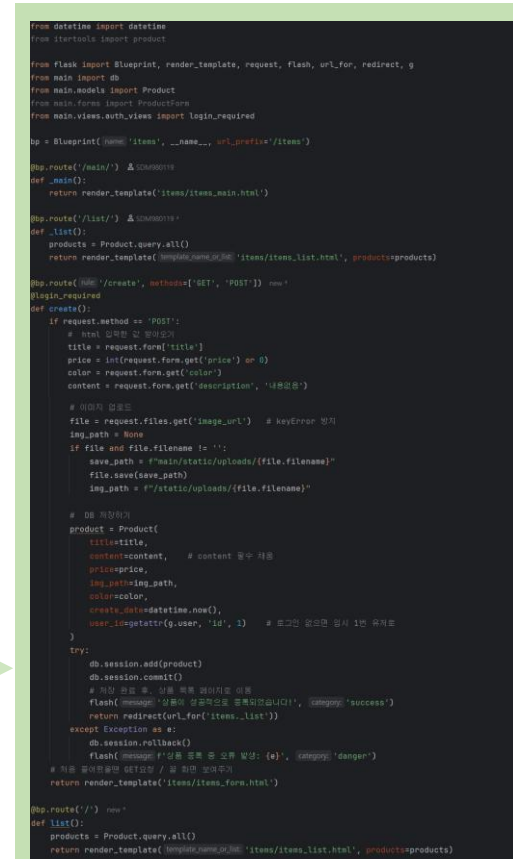
- Sign in / Sign up

- Sub page

Evaluation

V

Finish



마이페이지

- ✓ 로그인 후 오른쪽 상단의 '마이페이지' 클릭 시 본인의 개인정보 변경과 상품 등록, 댓글 기록 조회가 가능하도록 기능 구현.
- ✓ 상품 등록 버튼을 누르면 상품 등록 페이지로 넘어가도록 설정.

상품등록 페이지

- ✓ 제목, 할인율, 가격, 카테고리 선택, 이미지 업로드 기능 구현.
- ✓ 상품 등록을 하면 '마이페이지' -> '상품 등록 기록 조회' 칸 추가.

Categories

Introduction

v

Project Plan

v

Implementation

^

- Main page

- Sign in / Sign up

- Sub page

Evaluation

v

Finish

Slider / Section

- ✓ 페이지 처음과 중간 부분에 이벤트/쿠폰팩 슬라이드.
- ✓ 추석 특가 이벤트 슬라이드로 할인상품 정리.
- ✓ 카테고리 상품 슬라이드가 자동으로 한 열씩 넘어가도록 설정하고 각각의 상품도 2-3장씩 자동 슬라이드 되도록 구현.

```

/* 패션소품 카테고리 */
var swiper5 = new Swiper(".mySwiper5", {
  loop: true,
  slidesPerView: 3,
  grid: {
    rows: 2,
  },
  autoplay: {
    delay: 2500,
    disableOnInteraction: false,
  },
  spaceBetween: 40,
  navigation: {
    nextEl: ".swiper-button-next5",
    prevEl: ".swiper-button-prev5",
  },
});

const swiperInner1 = new Swiper(".mySwiperInner1", {
  loop: true,
  autoplay: {
    delay: 2000,
    disableOnInteraction: false,
  },
  slidesPerView: 1,
  effect: 'fade',
});

```

```

/* 옷 카테고리 */
var swiper3 = new Swiper(".mySwiper3", {
  loop: true,
  slidesPerView: 3,
  grid: {
    rows: 2,
  },
  autoplay: {
    delay: 2500,
    disableOnInteraction: false,
  },
  spaceBetween: 40,
});

const swiperInner = new Swiper(".mySwiperInner", {
  loop: true,
  autoplay: {
    delay: 2000,
    disableOnInteraction: false,
  },
  slidesPerView: 1,
  effect: 'fade',
});

```

```

/* Main Slider */
const swiper1 = new Swiper(".mySwiper1", {
  loop: true,
  autoplay: {
    delay: 2500,
    disableOnInteraction: false,
  },
  navigation: {
    nextEl: ".swiper-button-next1",
    prevEl: ".swiper-button-prev1",
  },
  allowTouchMove: false,
});

/* 추석 이벤트 상품 */
var swiper2 = new Swiper(".mySwiper2", {
  loop: true,
  navigation: {
    nextEl: ".swiper-button-next2",
    prevEl: ".swiper-button-prev2",
  },
  breakpoints: {
    1024: {
      slidesPerView: 4,
      spaceBetween: 35,
    },
  },
  allowTouchMove: false,
});

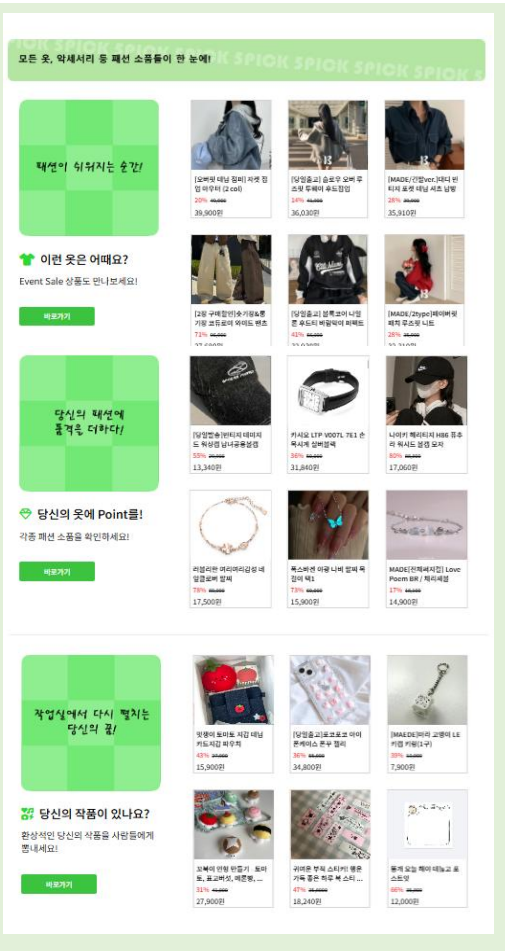
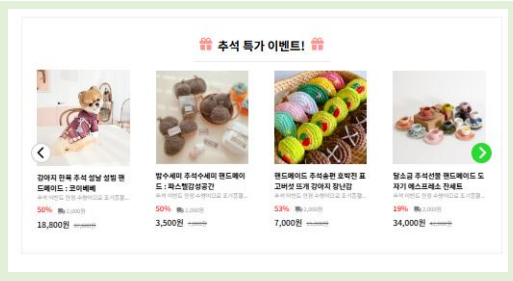
```

```

/* 옷 카테고리 */
var swiper3 = new Swiper(".mySwiper3", {
  loop: true,
  slidesPerView: 3,
  grid: {
    rows: 2,
  },
  autoplay: {
    delay: 2500,
    disableOnInteraction: false,
  },
  spaceBetween: 40,
});

const swiperInner = new Swiper(".mySwiperInner", {
  loop: true,
  autoplay: {
    delay: 2000,
    disableOnInteraction: false,
  },
  slidesPerView: 1,
  effect: 'fade',
});

```



Categories

Introduction

Project Plan

Implementation

[- Main page](#)

- Sign in / Sign up

- Sub page

Evaluation

Finish

로그인

회원가입

고객지원

opick

유저아이디

아이디

비밀번호

비밀번호

이메일

이메일 주소

전화번호

휴대폰 번호

주소

주소

비밀번호

비밀번호

비밀번호확인

비밀번호 확인

회원가입

이미 계정이 있으신가요? 로그인

✓ 기본적인 개인정보와 아이디 비밀번호를 입력하면
회원가입이 진행되는 기능을 구현.

✓ 또한 비회원 상태로 상품 클릭 시 회원가입 창으로 이동되도록 설정.

```
class UserCreateForm(FlaskForm):
    4 usage: 2 SDM +1
    userid = StringField(labe: '유저아이디', validators=[DataRequired('아이디를 입력해주세요'), Length(min=3, max=25)])
    username = StringField(labe: '닉네임', validators=[DataRequired('닉네임을 입력해주세요'), Length(min=3, max=25)])
    password1 = PasswordField(labe: '비밀번호', validators=[DataRequired('비밀번호를 입력해주세요'), EqualTo(fieldname: 'password2', message='비밀번호가 일치하지 않습니다.')])
    password2 = PasswordField(labe: '비밀번호확인', validators=[DataRequired('비밀번호를 다시 입력해주세요')])
    phone = StringField(labe: '전화번호', validators=[DataRequired('000-0000-0000')])
    email = EmailField(labe: '이메일', validators=[DataRequired('이메일을 입력해주세요'), Email()])
    address = StringField(labe: '주소', validators=[DataRequired('주소를 입력해주세요')]) # 추가 고민 필요 주소를 얼마나 조절지
    image = FileField(labe: '프로필사진', validators=[FileAllowed(['jpg', 'jpeg', 'png', 'gif'])])
```

[illegible]

```
# 특정 페이지에 로그인한 유저가 아니면 로그인페이지로 보냄
def login_required(view): 2 usages  ⚡ SDM980119 +1
    @functools.wraps(view)  ⚡ SDM980119
    def wrapped_view(*args, **kwargs):
        if g.user is None:
            _next = request.url if request.method == 'GET' else ''
            return redirect(url_for(endpoint='auth.login', next=_next))
        return view(*args, **kwargs)
    return wrapped_view
```

```
@bp.route('/signup', methods=['GET', 'POST'])
def signup():
    form = UserCreateForm()
    if request.method == 'POST' and form.validate_on_submit():
        user = User.query.filter_by(username=form.username.data).first()
        if not user:
            user = User(userid=form.userid.data,
                        username=form.username.data,
                        password=generate_password_hash(form.password1.data),
                        email=form.email.data,
                        phone=form.phone.data,
                        address=form.address.data
                        )
            db.session.add(user)
            db.session.commit()
            return redirect(url_for('main.index'))
        else:
            flash('이미 존재하는 사용자입니다.')
    return render_template('template_name_or_list:auth/signup.html', form=form)
```

Categories

Introduction

Project Plan

Implementation

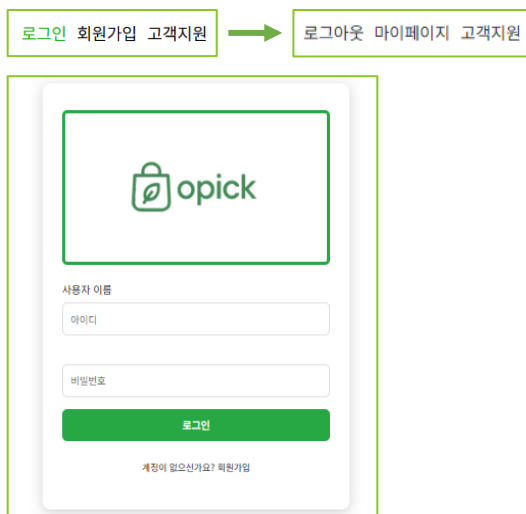
[- Main page](#)

- Sign in / Sign up

- Sub page

Evaluation

Finish



- ✓ 회원가입 진행 후 로그인을 할 경우 '로그인/회원가입'에서 '로그아웃/마이페이지'로 옵션 변경 기능 구현.
- ✓ 로그아웃 시에 모든 session 삭제 후 메인화면으로 이동.
- ✓ 로그인 페이지에서 계정이 없을 경우, 하단의 '회원가입'을 선택해 회원가입 페이지로 이동.

```
class UserLoginForm(FlaskForm): 2 usages 8 SDM
    userid = StringField(label='사용자 이름', validators=[DataRequired('아이디를 입력해주세요'), Length(min=3, max=25)])
    password = PasswordField(label='', validators=[DataRequired('비밀번호를 입력해주세요')])
```

```
% extends "base.html" %}
% block content %}

<div class="auth-box login-shadow">
    <div class="auth-box login-header">
        <div class="auth-box login-border">
            <a href="{url_for('auth.login')}">로그인</a> <a href="{url_for('static', filename='img/pw/login.png')}">로그인 이미지</a>
        </div>
        <div class="auth-box login-text">
            <span>이메일</span>
            <span>비밀번호</span>
            <span>로그인</span>
        </div>
    </div>
</div>

<div class="form">
    <form method="post">
        {{ form.csrf_token }}
        {%- include "form_errors.html" %}

        <div class="form-1">
            {{ form.userid.label(class="form-label") }}
            {{ form.userid(class="form-control", placeholder="이메일주소") }}
        </div>

        <div class="form-2">
            {{ form.password.label(class="form-label") }}
            {{ form.password(class="form-control", placeholder="비밀번호") }}
        </div>

        <button type="submit" class="btn btn-success">로그인</button>
        <a href="{url_for('auth.signup')}">회원가입</a>
    </form>
</div>
</div>
% endblock %}
```

```
# 로그인한 사용자면 g.user에 유저 정보를 받아서 저장
@bp.before_app_request ㄹ SDM980119

def load_logged_in_user():
    user_id = session.get('user_id')
    if user_id is None:
        g.user = None
    else:
        g.user = User.query.get(int(user_id))

# 로그아웃시 session을 삭제하고 메인 으로 보냄
@bp.route('/logout') ㄹ SDM

def logout():
    session.clear()
    return redirect(url_for('main.index'))
```

```
@app.route('/login', methods=['GET', 'POST']) @SOM
def login():
    form = UserLoginForm()
    if request.method == 'POST' and form.validate_on_submit():
        error = None
        user = User.query.filter_by(userid=form.userid.data).first()
        if not user:
            error = '존재하지 않는 사용자입니다.'
        elif not check_password_hash(user.password, form.password.data):
            error = '비밀번호가 올바르지 않습니다.'
        if error is None:
            session.clear()
            session['user_id'] = user.id
            _next = request.args.get('next', '')
            if _next:
                return redirect(_next)
            else:
                return redirect(url_for('main.index'))
        flash(error)
    return render_template('template_name_or_list: 'auth/login.html', form=form) # GET 방식으로 로그인 요청시 html 문서가 나오도록
```


Categories

Introduction

✓

Project Plan

✓

Implementation

^

- Main page

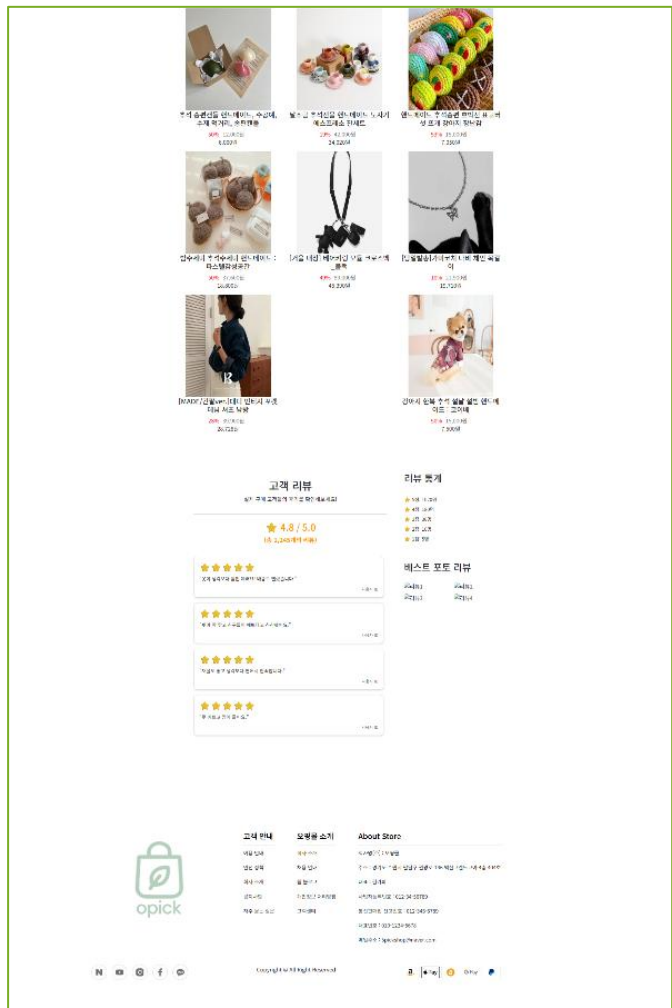
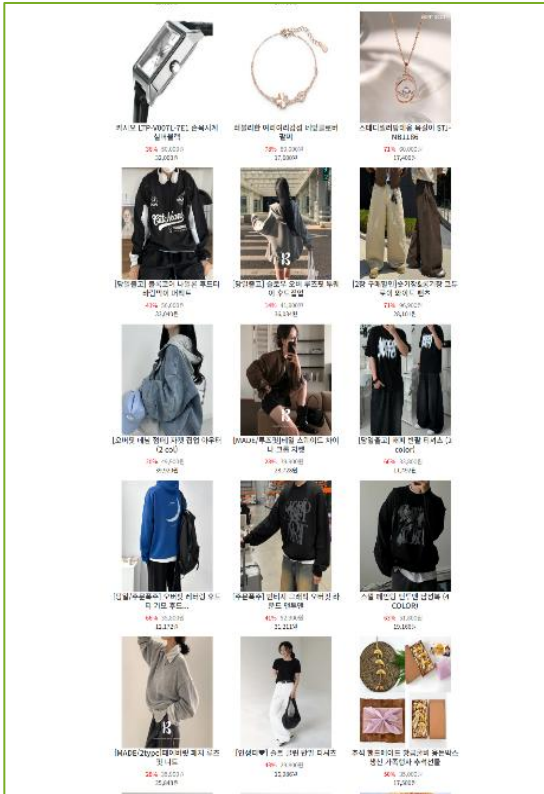
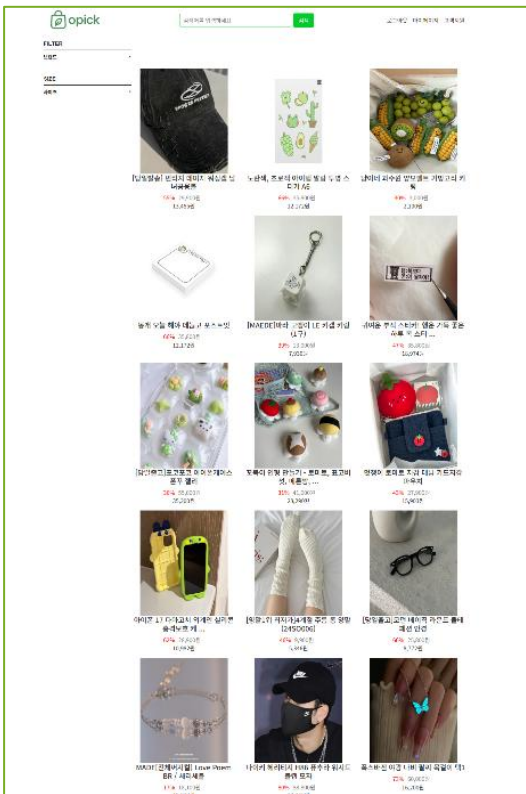
- Sign in / Sign up

- Sub page

Evaluation

✓

Finish



Header - 로고, 로그인/회원가입/고객지원(-> 로그아웃/마이페이지/고객지원), 카테고리

Section - Filter, Size, 상품, 고객 리뷰, 리뷰 통계, 베스트 포토 리뷰

Footer - 페이지 설명과 회사 소개

Categories

Introduction

v

Project Plan

v

Implementation

^

- Main page

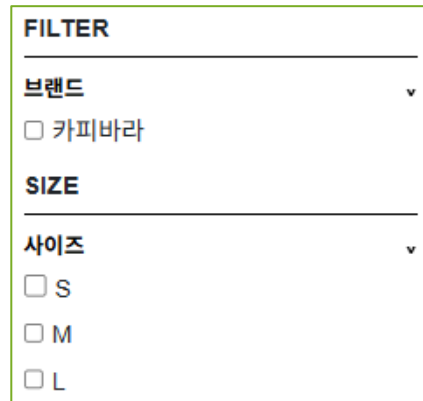
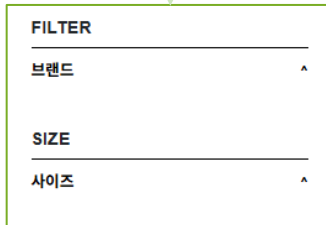
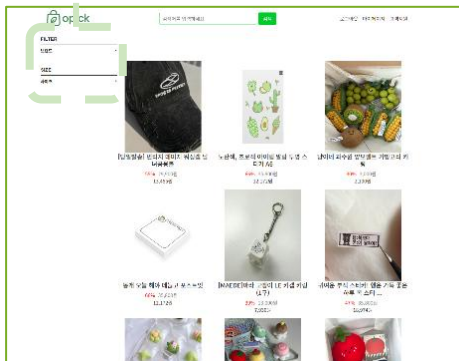
- Sign in / Sign up

- Sub page

Evaluation

v

Finish



```

/* 고정 */
.sidebar {position: sticky;top: 20px;}

/* 필터 */
.filter {width: 1200px; height: 100px; margin: 0 auto; display: flex;}
.filter-line {width: 250px; font-family: Arial, Helvetica, sans-serif; margin-top: 20px;}
.filter-title {font-size: 16px; font-weight: bold; padding-bottom: 8px; border-bottom: 1px solid #000; margin-bottom: 5px;}
.filter-section {display: block;}
.filter-toggle {background: none; border: none; font-size: 14px; font-weight: bold; width: 100%; text-align: center; padding-bottom: 5px;}
.filter-toggle .arrow {font-size: 12px; transition: transform 0.3s;}

.filter-list {list-style: none; margin: 0; padding: 0; display: none;}
.filter-list li {margin: 5px 0;}
.filter-list input {margin-right: 5px;}

/* 열렸을 때 보이게 */
.filter-section.active .filter-list {display: block;}
.filter-section.active .filter-toggle .arrow {transform: rotate(180deg);}
    
```

```

/* 상품1-1 */
.cont1 {width: 1200px; margin: 0 auto; display: flex;}

/* 사이즈 */
.size {width: 250px;}
.size-line {width: 250px; font-family: Arial, Helvetica, sans-serif; margin-top: 20px;}
.size-title {font-size: 16px; font-weight: bold; padding-bottom: 8px; border-bottom: 1px solid #000; margin-bottom: 5px;}
.size-section {display: block;}
.size-toggle {background: none; border: none; font-size: 14px; font-weight: bold; width: 100%; text-align: center; padding-bottom: 5px;}
.size-toggle .arrow2 {font-size: 12px; transition: transform 0.3s;}

#s {width: 15px; height: 15px;}

.size-list {list-style: none; margin: 0; padding: 0; display: none;}
.size-list li {margin: 5px 0;}
.size-list input {margin-right: 5px;}

/* 열렸을 때 보이게 */
.size-section.active .size-list {display: block;}
.size-section.active .size-toggle .arrow2 {transform: rotate(180deg);}
    
```

```

document.querySelectorAll(".filter-toggle").forEach(button => {
  button.addEventListener("click", () => {
    const section = button.parentElement;
    section.classList.toggle("active");
  });
});
    
```

```

document.querySelectorAll(".size-toggle").forEach(button => {
  button.addEventListener("click", () => {
    const section = button.parentElement;
    section.classList.toggle("active");
  });
});
    
```

✓ 브랜드와 사이즈 필터를 이용해 클릭 시 클릭한 필터에 해당되는 품목들만 나오도록 기능 구현.

Categories

Introduction

✓

Project Plan

✓

Implementation

^

- Main page

- Sign in / Sign up

- Sub page

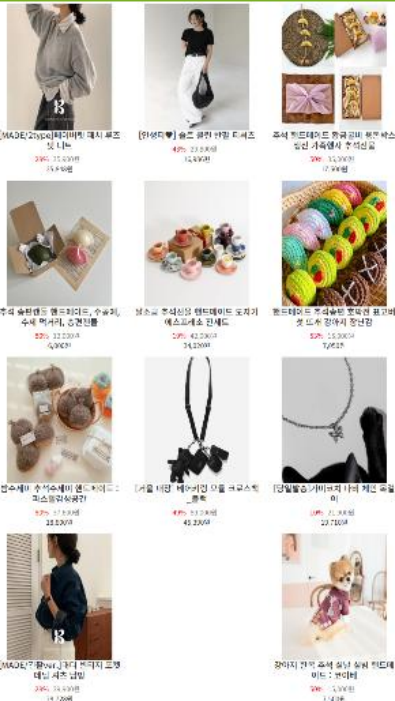
Evaluation

✓

Finish

```
<div class="product-container">
  {% if products %}
  {% for p in products %}
    <div class="product-card-wrapper">
      <a href="{{ url_for('items.detail', product_id=p.id) }}">
        <div class="product-card" data-color="{{ p.color or 'default' }}">
          {% if p.images %}
            <!-- 첫 번째 이미지 -->
            
          {% else %}
            <!-- 이미지 없으면 기본 이미지 -->
            
          {% endif %}
          <h3>{{ p.title }}</h3>
          <div class="disal">
            <span>{{ p.discount }}%</span>
            <p>{{ '{:,.format( p.price) }}원</p>
          </div>
          <p class="discount">{{ '{:,.format( (p.price * (1 - p.discount / 100))|round(0)|int) }}원</p>
        </div>
      </a>
    </div>
  {% endfor %}
  {% else %}
    <p style="text-align: center;">등록된 상품이 없습니다.</p>
  {% endif %}
</div>
```

- ✓ 마이페이지에서 등록한 상품이 최신순으로 나열되도록 설정.
- ✓ 같은 카테고리인 아닌 전체 품목이 나오도록 디자인.
- ✓ 작성한 리뷰가 sub page 하단에 나오도록 front로 제작.
- ✓ 품목 이미지가 없으면 기본 이미지가 나오도록 설정



Categories

Introduction v

Project Plan v

Implementation v

Evaluation ^

- 성득명, 빙승현

- 지수정, 김태연

Finish

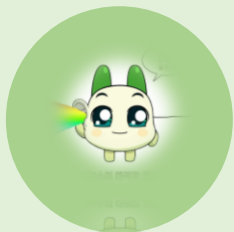


성득명

이번 팀 프로젝트에서는 Python과 Flask를 사용하여 PyCharm 환경에서 웹 서비스를 구현하였으며, 데이터베이스는 SQLite를 사용했습니다.
초기 계획 단계에서 구상했던 기능들을 모두 구현하지 못해 아쉬움이 많이 남는 프로젝트였습니다.

프론트엔드는 기본적인 틀만 잡는 수준에 그쳤지만, 백엔드 구현에 집중하면서 뷰 파일 간의 데이터 전달 방식과 데이터베이스에서 정보를 가져오고 저장하는 과정을 직접 실습해볼 수 있는 뜻깊은 시간이었습니다.

또한 팀 프로젝트를 진행하면서, 팀원들의 기술적인 역량도 중요하지만, 정해진 기간 내에 각자 맡은 역할을 성실히 수행하고 프로젝트에 적극적으로 참여하는 자세가 더욱 중요하다는 점을 깨달을 수 있었습니다.



빙승현

짧은시간으로 인한 최초구상보다는 미구현인 부분이 아쉽지만 프로젝트 진행시에 학습간 배운것 (Flask,Sqlite,Java,Html 등)으로만 쇼핑몰의 필수적인부분은 모두 구현할수있었다는점이 인상깊었습니다.

Categories

Introduction ∨

Project Plan ∨

Implementation ∨

Evaluation ^

- 성득명, 빙승현

- 지수정, 김태연

Finish



지수정

팀 프로젝트인 만큼 정해진 역할을 잘 수행했어야 했는데 많이 미흡하고 이해 못 하는 부분이 많아서 만족할 만한 성과가 나오지 않은 것 같아서 아쉬운 것 같았다.

프론트와 백을 같이 작업해보았는데 이해도가 떨어지다 보니 백을 만드는 과정에서 어려움이 많이 찾아왔지만, 열심히 배우고 물어가며 작성을 해보았다. 하지만, 연결 과정에서 겹치고 오류나는 상황이 발생하여 관련 지식의 이해도와 팀 소통, 협업이 중요한 것 같다는 생각을 하게 되었다.

이번에는 다소 아쉬운 점이 많았지만, 점차 시간이 갈 수록 지금보다 더 발전하고 더 좋은 작업물이 나올 수 있도록 노력할 것이다.



김태연

팀 프로젝트를 진행하면서 기본적으로 잡혀있는 틀이 없었기 때문인지 처음 구상부터 어려움을 많이 느꼈지만, 각자 잘 하는 분야를 내세우니 생기는 각종 문제들을 해결해 나갈 수 있었다.

개인적으로는 백보다 프론트에 집중했었어서 백에 그다지 큰 비중은 없지만 프론트를 두고 봤을 때에 조금 더 디테일을 잡을 수 있었을 텐데 하며 아쉬움이 많이 남는 프로젝트였다. 그럼에도 그만큼 적은 시간에 최대한의 디테일을 잡을 수 있는 방법 또한 배울 수 있었기에 다음 프로젝트를 진행하기 전 조금 더 성장함을 느꼈다.

이번 프로젝트에서도 저번 프로젝트와 다를 바 없이 열심히 참여했고, 문제점을 받아들이며 해결해나가는 마음가짐도 이어졌다. 다음 프로젝트까지 초심 잃지 않고 노력해서 더욱 발전할 것이라 확신한다.

Categories

Introduction ∨

Project Plan ∨

Implementation ∨

Evaluation ^

Finish



좋은 아이디어를 가지고 있으신가요?

5PICK과 함께 마음 속
프로젝트를 실현하세요



Thank you!

Main page link

Sub page link

Opick shop - 5ping mall