

Django웹 개발 미니 프로젝트

Django 기반 기능구현 미니 프로젝트



목차

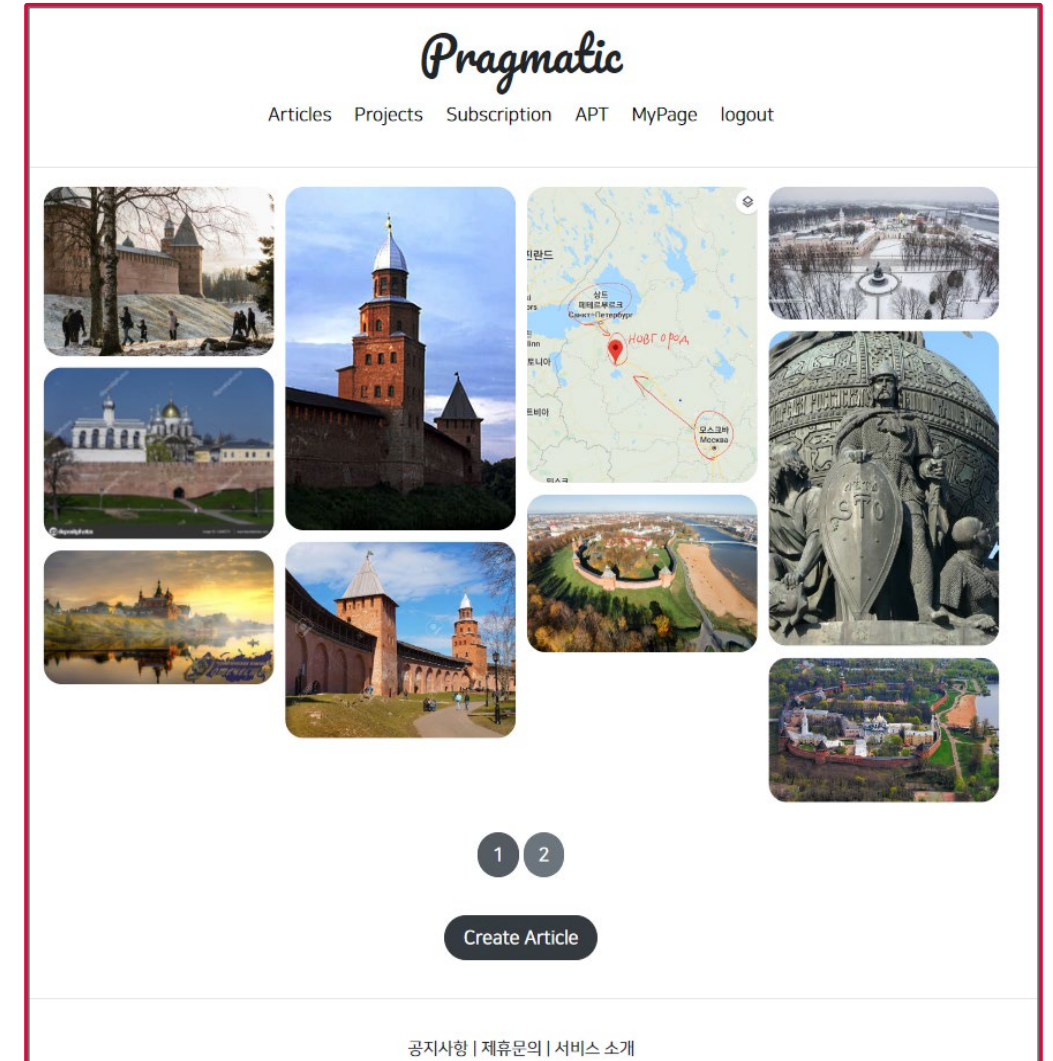


1. 프로젝트 배경
2. 프로젝트 소개
3. 구현 기능 목록
4. 테이블정의서
5. 기능 소개
6. FlowChart/WBS
7. 프로젝트 한계점 및 느낀점

Django를 활용한 기본 웹 개발

+ 대시보드 구현 역량 Up

- 웹 개발자 지망생으로서 데이터직군에서 협업하기 위한 기본적인 대시보드 구현 능력 및 배포하기 위한 웹을 구현하는 역량을 키우고자 함
- 수집 및 전처리된 데이터를 가지고 의도에 따라 출력하고 CRUD기반의 여러 기능들을 구현하며 장고 프레임워크에 대한 이해도를 높이하고자 함

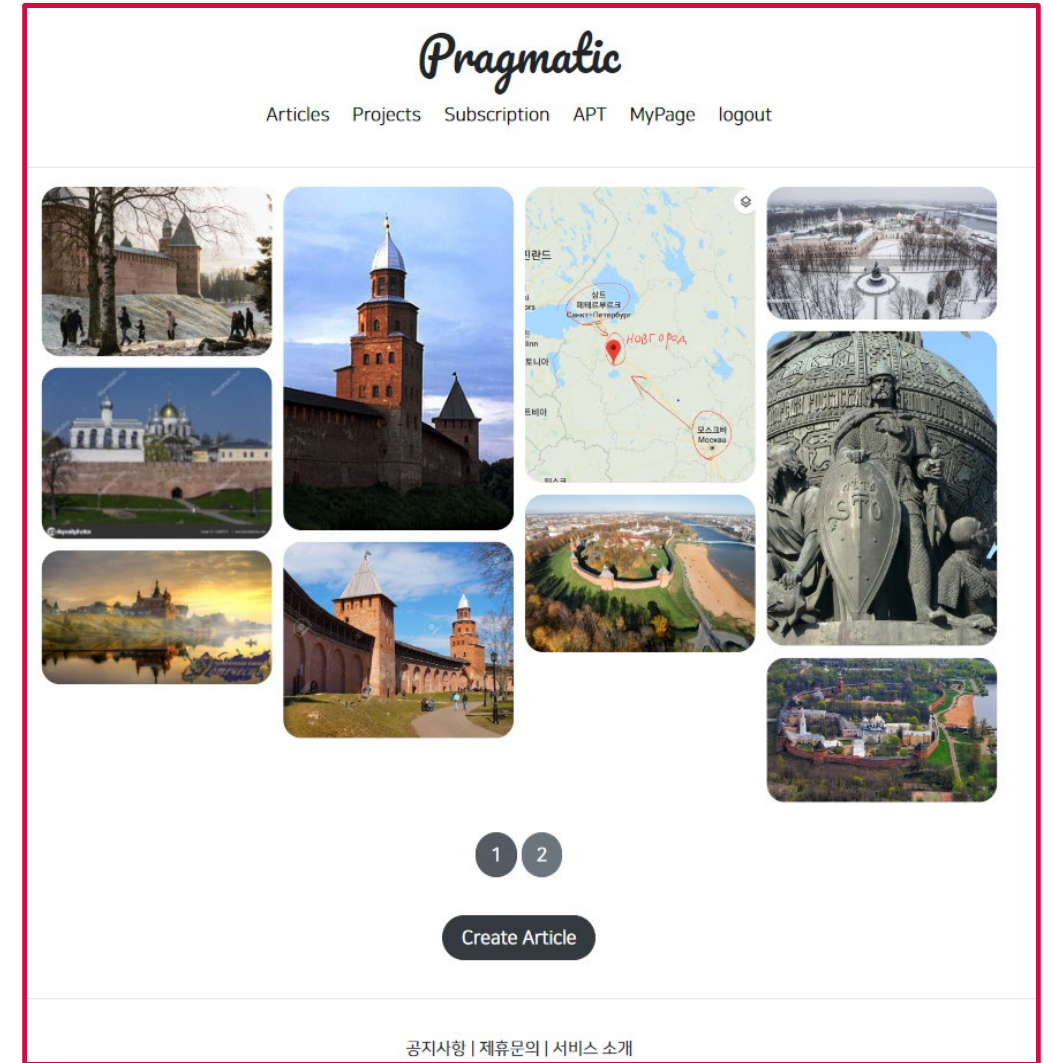


<메인 페이지>

Django를 활용한 기본 웹 개발

+ 서울 열린데이터 공공데이터 활용
보증금 예측 모델페이지 구현

- 장고에서 제공하는 CRUD 특화 방식인 Class Based View(CBV) 방식으로 코드를 작성하여 각 기능 구현



<메인 페이지>



곽정근

담당 업무: 장고기능구현/웹 개발

rhkrcjswo@gmail.com

Git Hub : <https://github.com/alpha8108/PinterestDjango2>

회원 가입(Account)

1. 회원 조회
2. 회원 정보 수정
3. 회원 탈퇴
4. 로그인/로그아웃

댓글(Comment)

1. 댓글 작성
2. 댓글 삭제 + decorator

프로필(Profile)

1. 프로필 생성
2. 프로필 조회
3. 프로필 수정
4. 프로필 삭제(회원 탈퇴시)

게시판(Project)/구독

1. 게시판 생성
2. 게시판 조회
3. 게시판 구독

게시글(Article)

1. 게시글 생성
2. 게시글 조회
3. 게시글 수정 + decorator
4. 게시글 삭제 + decorator
5. 게시글 페이지(리스트뷰)

서울 보증금 예측모델 페이지

1. 서울 열린 데이터 광장 공공 데이터를 활용
2. 입력한 값에 따라 예측 보증금 출력

게시글(Article)

속성	자료형	설명
id	Integer (Primary Key)	게시글의 고유 식별자
writer	ForeignKey(User)	게시글 작성자 (User 테이블과 연결)
project	ForeignKey(Project)	게시글이 속한 프로젝트 (Project 테이블과 연결)
title	CharField(최대길이:200)	게시글의 제목
image	ImageField	게시글의 이미지 파일
content	TextField	게시글의 내용
created_at	DateField(auto_now_add=True)	게시글 생성 일자

댓글(Comment)

속성	자료형	설명
id	Integer (Primary Key)	댓글의 고유 식별자
article	ForeignKey(Article)	댓글이 속한 게시글 (Article 테이블과 연결)
writer	ForeignKey(User)	댓글 작성자 (User 테이블과 연결)
content	TextField	댓글 내용
created_at	DateTime (auto_now=True)	댓글 작성 일시

프로필(Profile)

속성	자료형	설명
id	Integer (Primary Key)	프로필의 고유 식별자
user	OneToOneField(User)	해당 프로필에 연결된 사용자 (User 테이블과의 일대일 관계)
image	ImageField	프로필 이미지
nickname	CharField(max_length=20, unique=True)	닉네임 (최대 길이 20자, 고유해야 함)
message	CharField(max_length=100)	메시지 (최대 길이 100자)

게시판(Project)

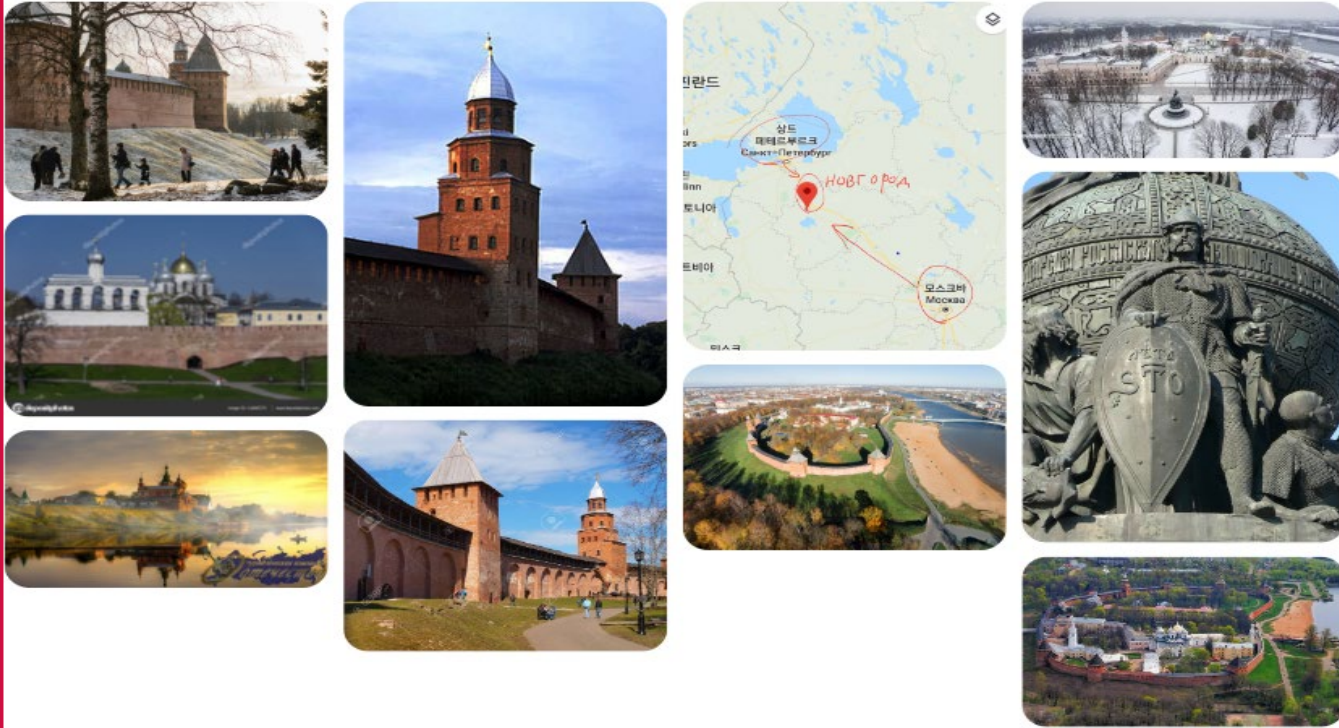
속성	자료형	설명
id	Integer (Primary Key)	프로젝트의 고유 식별자
title	CharField(max_length=20, null=False)	프로젝트 제목 (최대 길이 20자, null 값 허용 안 함)
description	CharField(max_length=200, null=True)	프로젝트 설명 (최대 길이 200자, null 값 허용)
image	ImageField	프로젝트 이미지
created_at	DateTimeField (auto_now_add=True)	프로젝트 생성 일시

구독(Subscription)

속성	자료형	설명
id	Integer (Primary Key)	구독의 고유 식별자
user	ForeignKey(User)	구독한 사용자 (User 테이블과 연결)
project	ForeignKey(Project)	구독한 프로젝트 (Project 테이블과 연결)

Pragmatic

Articles Projects Subscription APT MyPage logout



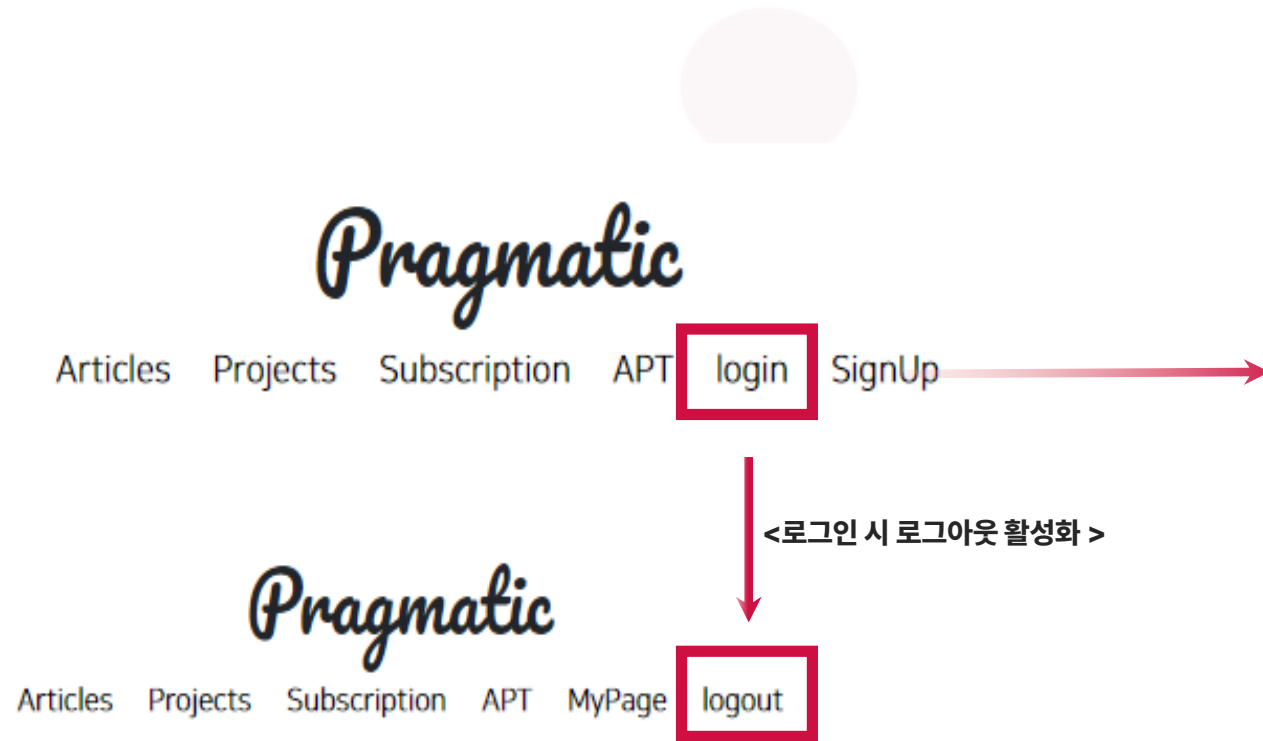
1 2

Create Article

메뉴 탭

게시글 목록

게시글 생성 버튼



The login form is titled 'Login'. It contains two input fields: 'Username' and 'Password'. Below these fields is a dark blue button labeled 'Login'.

<로그인 폼>



SignUp

Username

Username

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password

Password

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation

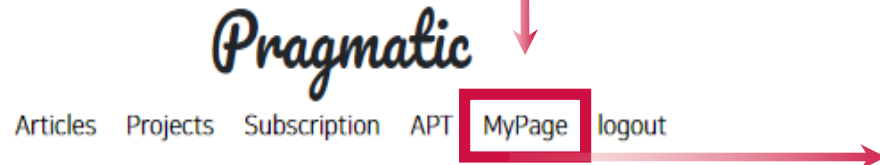
Password confirmation

Enter the same password as before, for verification.

가입

<회원가입 폼>

<회원 가입 후>



Pragmatic

Articles Projects Subscription APT MyPage logout

March 22, 2024, 8:31 a.m. 에 가입함

Create Profile

⚙️ ⓧ

No Article YET!

1

Create Article

<회원 정보 조회>

프로필 생성 버튼

프로필 수정 및 회원탈퇴 버튼

Pragmatic

ArticlesProjectsSubscriptionAPTMyPagelogout

March 22, 2024, 8:31 a.m. 에 가입함

Create Profile

No Article YET!

1

Create Article

<회원 정보 화면>

Update Profile

Image

Currently: profile/15세기_노브고로드_중세_크레믈_벽_ODMOMIW.jpg

Change:

파일 선택

선택된 파일 없음

Nickname

나의 닉네임

Message

나의 대화명

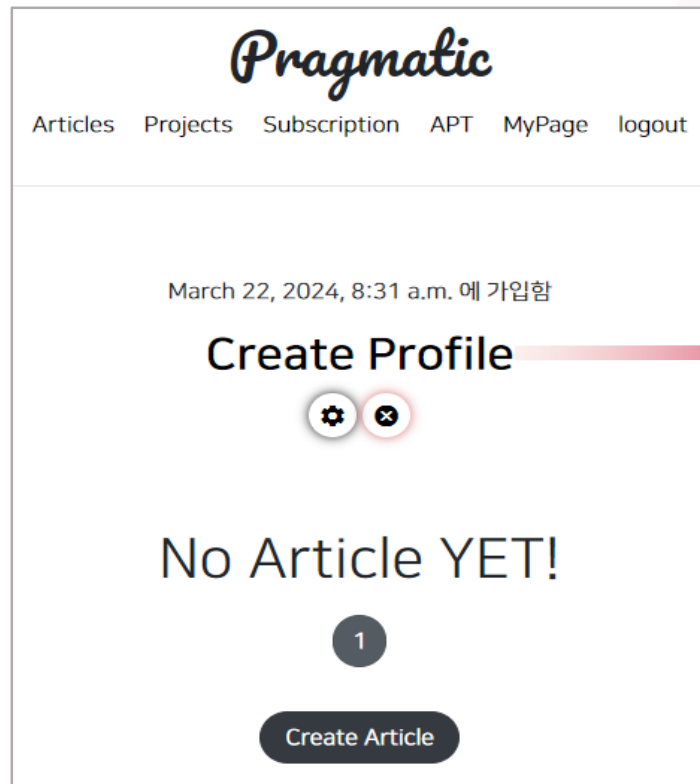
프로필 업데이트

<프로필 생성 및 수정 폼>

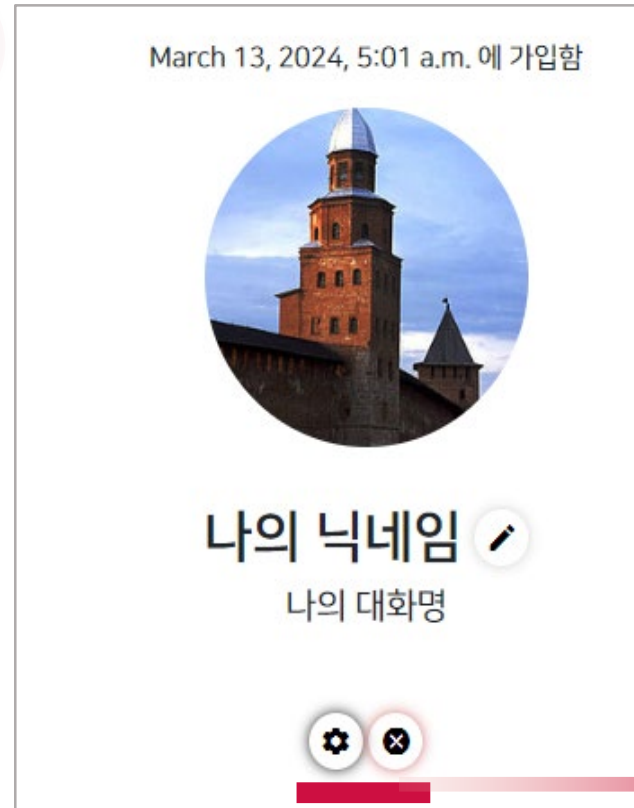
프로필 사진 설정

닉네임 설정

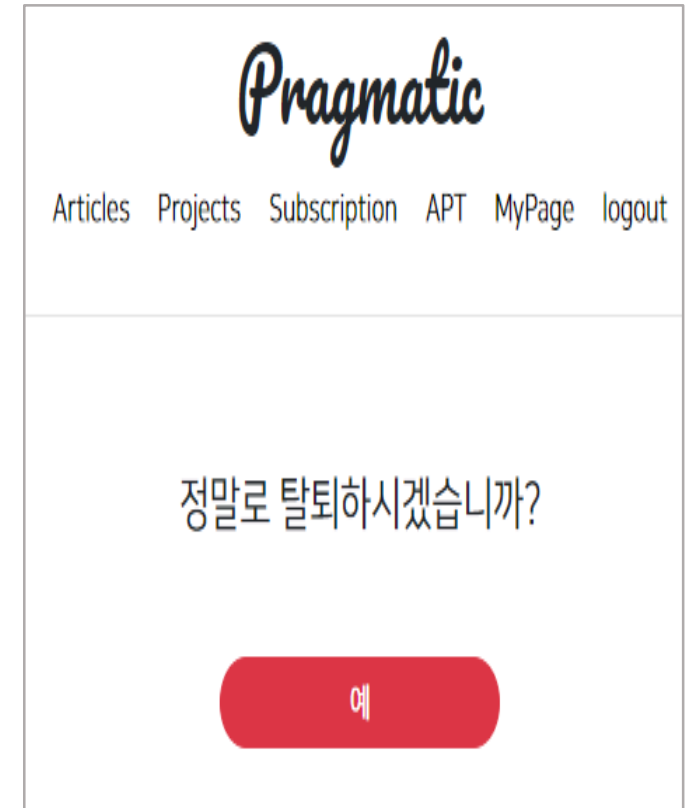
대화명 설정



<회원 정보 화면>



<만들어진 프로필 화면>

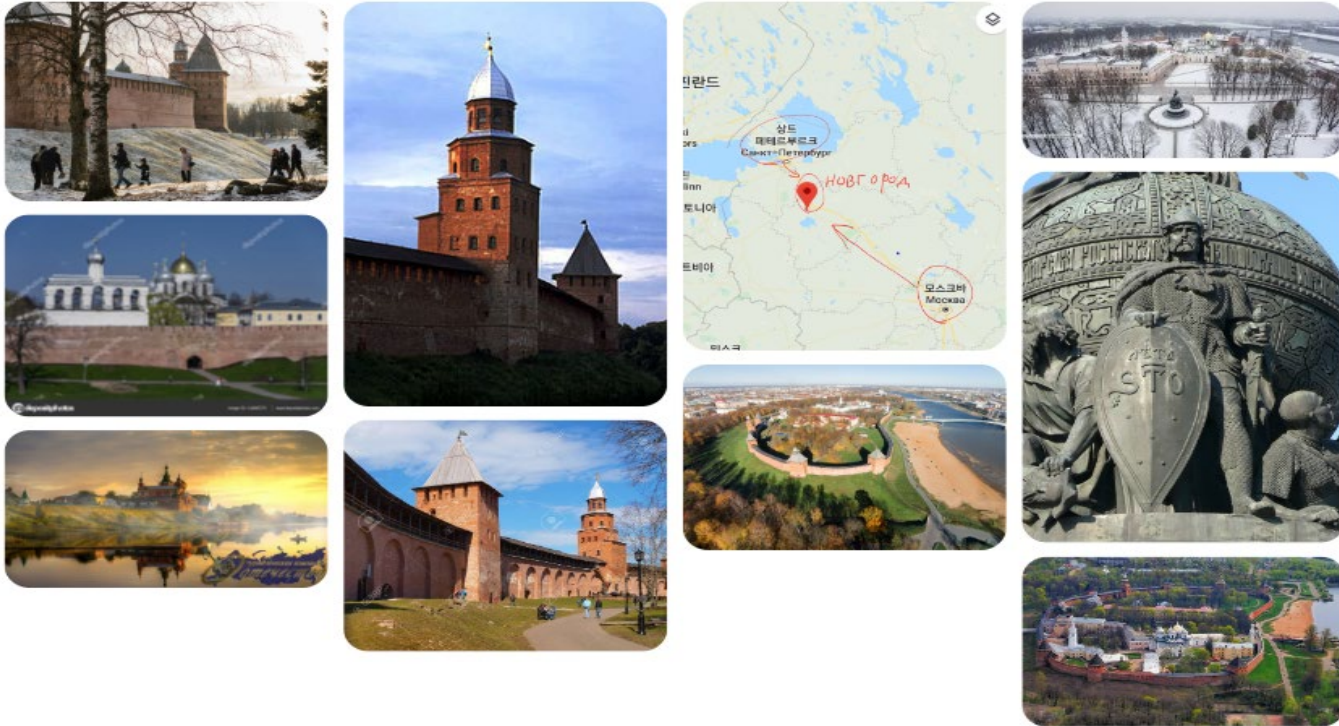


<회원 탈퇴 화면>

Pragmatic

Articles Projects Subscription APT MyPage logout

게시글 탭



게시글 목록

1 2

Create Article

게시글 생성 버튼

The diagram illustrates the 'Article Create' form with the following fields and their corresponding data labels:


- Title**: Labeled as **게시글 제목** (Article Title).
- Image**: Labeled as **게시글 이미지** (Article Image).
- Project**: Labeled as **게시할 게시판** (Board to post).
- Content**: Labeled as **게시글 내용** (Article Content).

The form also includes a '파일 선택' (File Selection) button, a '선택된 파일 없음' (No files selected) message, and a '게시' (Post) button.

<게시글 생성및 수정 폼>

첫 게시글

작성자 : 나의 닉네임



첫 게시글 내용

UpdateDelete

<보여지는 게시글>

수정/삭제 버튼

Pragmatic

Articles Projects Subscription APT MyPage logout

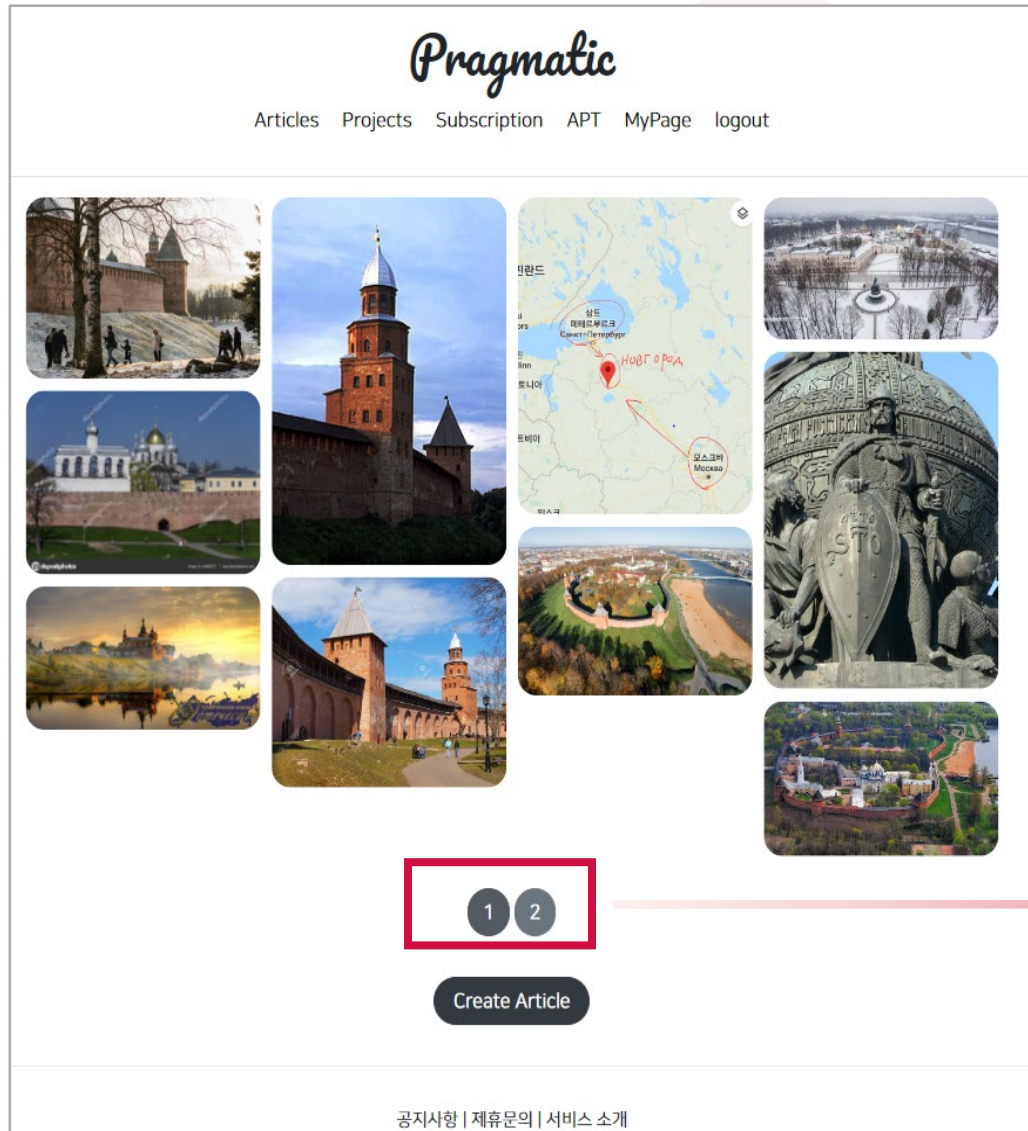
Delete Article : 첫 게시글

삭제

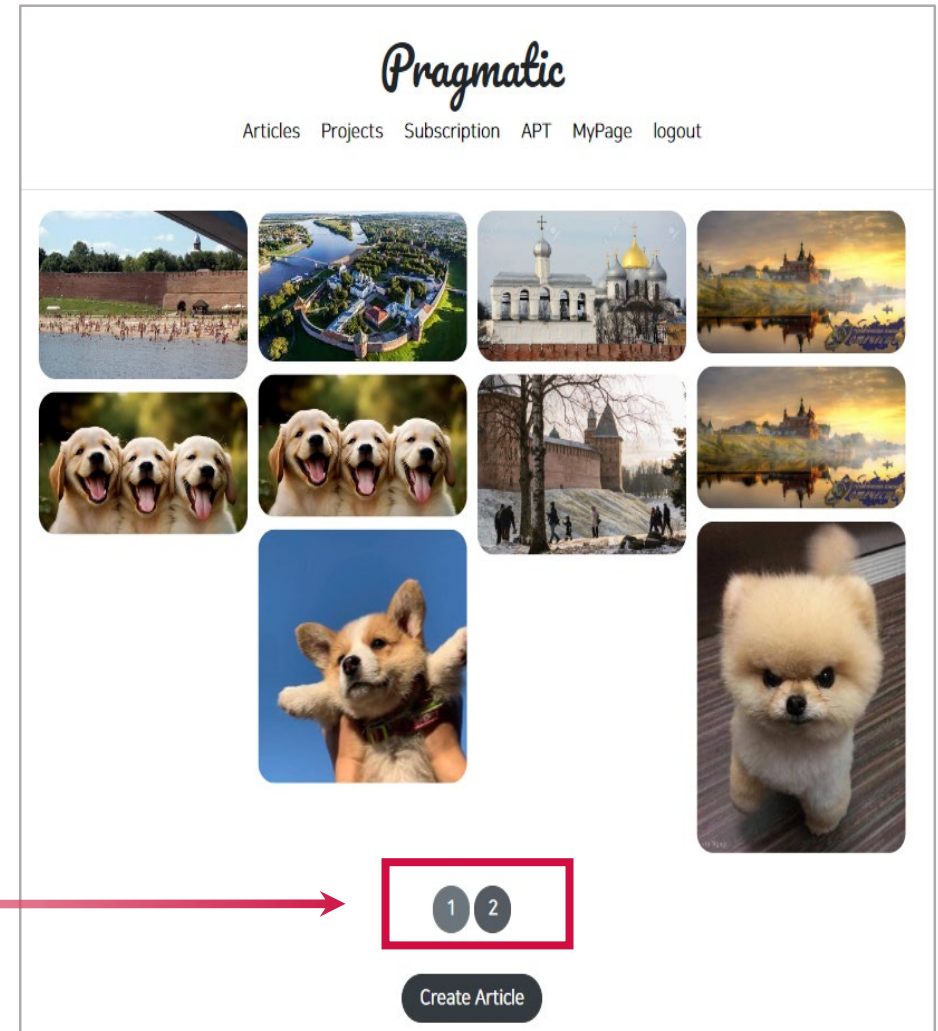
공지사항 | 제휴문의 | 서비스 소개

Pragmatic

<게시글 삭제화면>



<첫 페이지>



<다음 페이지>

강아지3

작성자 : 수정한 닉네임



강아지 게시판에 올리는 용도입니다.

Comment Create

Content

Content

작성

<댓글 작성 후>

<게시글 밑 댓글 작성 폼>

강아지3

작성자 : 수정한 닉네임



강아지 게시판에 올리는 용도입니다.

나의 닉네임 March 27, 2024, 4:55 a.m.

강아지가 너무 귀엽네요 ㅎㅎ

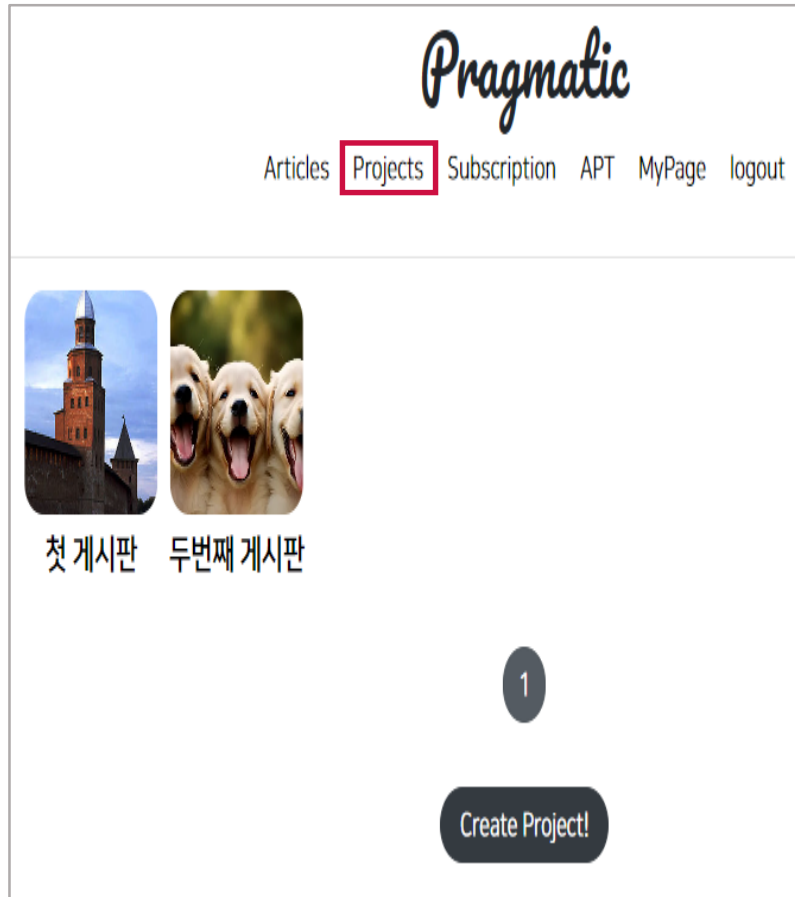
Delete

Comment Create

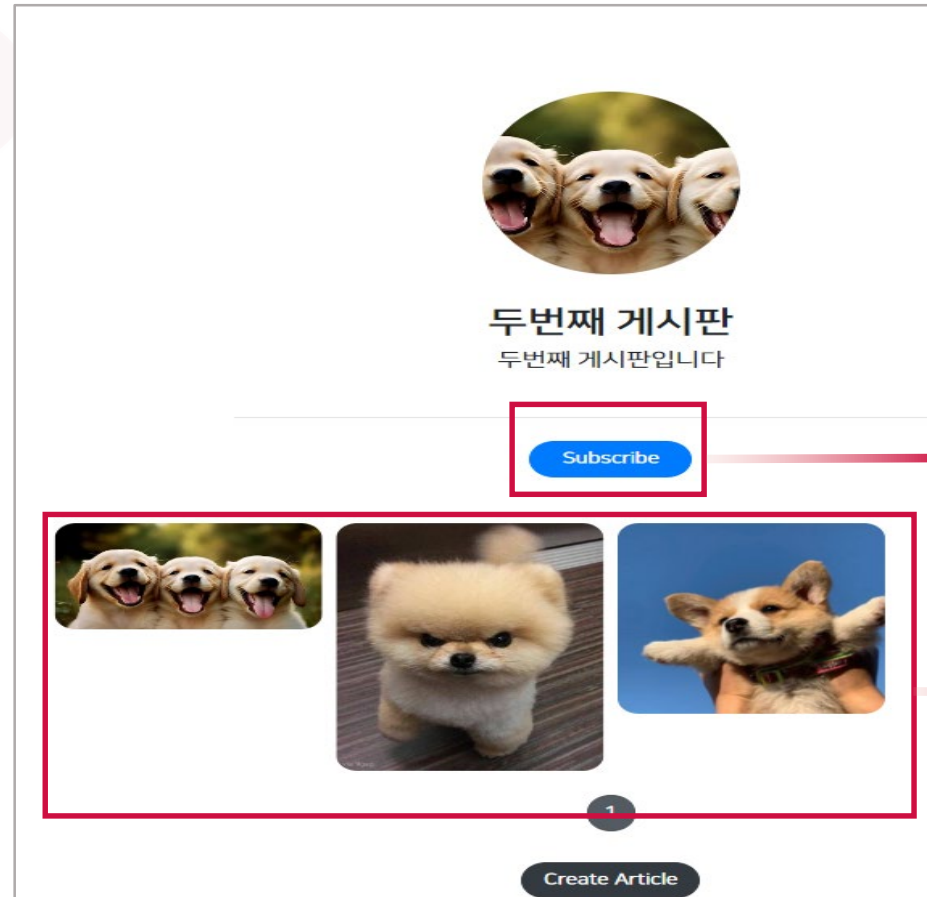
Content

Content

댓글 삭제 버튼



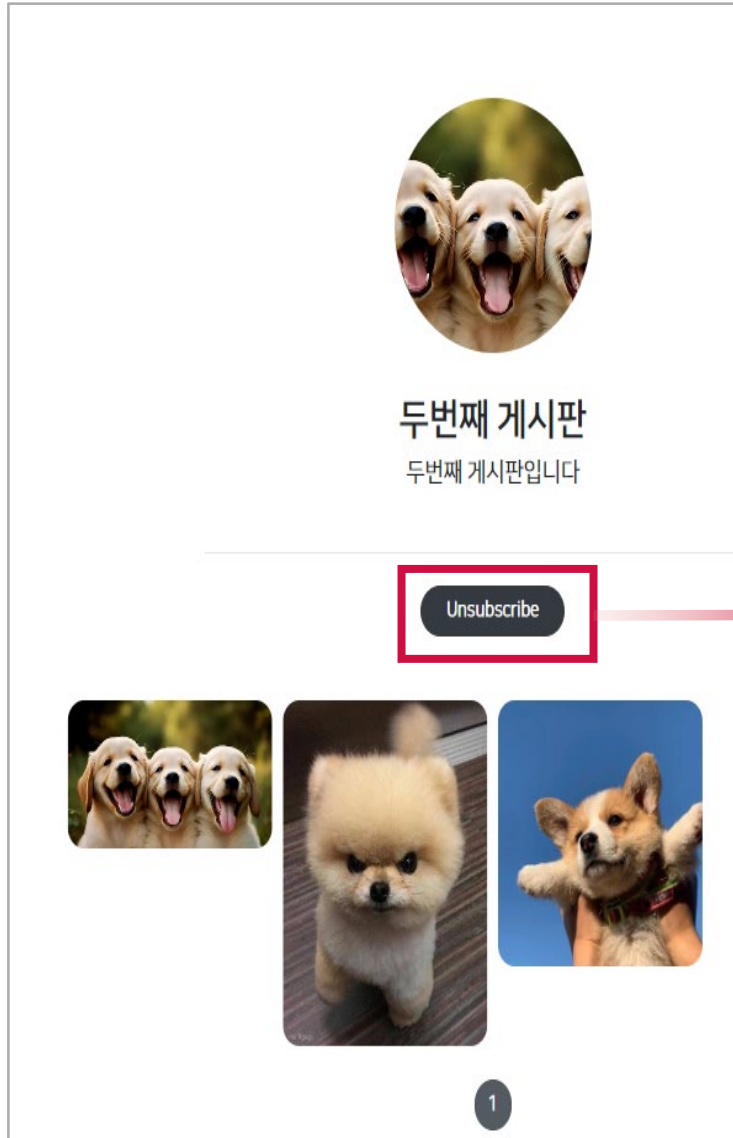
<게시판 목록 화면>



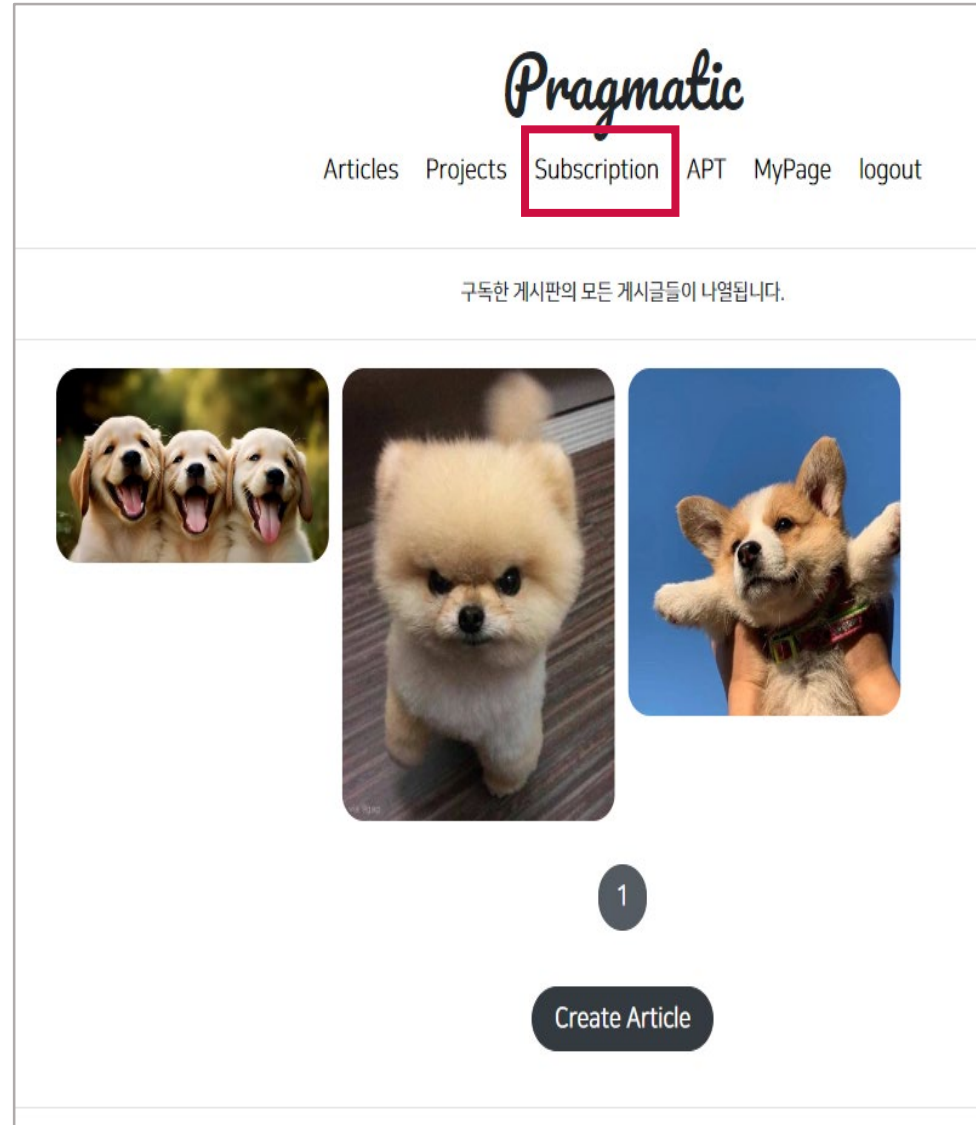
<게시판 조회 화면>

게시판 구독 버튼

<해당 게시판에 올린 게시글 나열>



<게시판 목록 화면>



<구독한 게시판의 모든 게시글 나열>

게시글

1. 게시글 수정
2. 게시글 삭제

```
@method_decorator(article_ownership_required, 'get')
@method_decorator(article_ownership_required, 'post')
class ArticleDeleteView(DeleteView):
    model = Article
```

```
@method_decorator(article_ownership_required, 'get')
@method_decorator(article_ownership_required, 'post')
class ArticleUpdateView(UpdateView):
    model = Article
```

댓글

1. 댓글 삭제

```
@method_decorator(comment_ownership_required, 'get')
@method_decorator(comment_ownership_required, 'post')
class CommentDeleteView(DeleteView):
    model = Comment
```

Decorator 적용

계정

1. 프로필 수정
2. 회원 탈퇴

```
@method_decorator(has_ownership, 'get')
@method_decorator(has_ownership, 'post')
class AccountUpdateView(UpdateView):
    model = User
```

```
@method_decorator(has_ownership, 'get')
@method_decorator(has_ownership, 'post')
class AccountDeleteView(DeleteView):
    model = User
```

Apartmentapp

(보증금예측 모델 페이지)

-APT 페이지를 추가로 생성

1. 서울 열린데이터 전월세가
공공데이터 활용
2. 입력 폼 구현
3. 입력한 값에 따른 예측 보증금 출력

Pragmatic

Articles Projects Subscription **APT** MyPage logout

보증금 예측

지역구:

강남구

임대면적(1평 = 3.3057㎡):

평수x3.3057한 값의 반올림한 값을 입력(정수로 입력)

임대료(만원):

임대료를 입력하세요(숫자만 입력)

건축년도:

1970

건물용도:

아파트

총전보증금(만원):

총전 임대료를 입력하세요(숫자만 입력)

총전임대료(만원):

총전 임대료를 입력하세요(숫자만 입력)

Predict



보증금 예측

지역구:

강남구

임대면적(1평 = 3.3057㎡):

81

임대료(만원):

50

건축년도:

2023

건물용도:

아파트

총전보증금(만원):

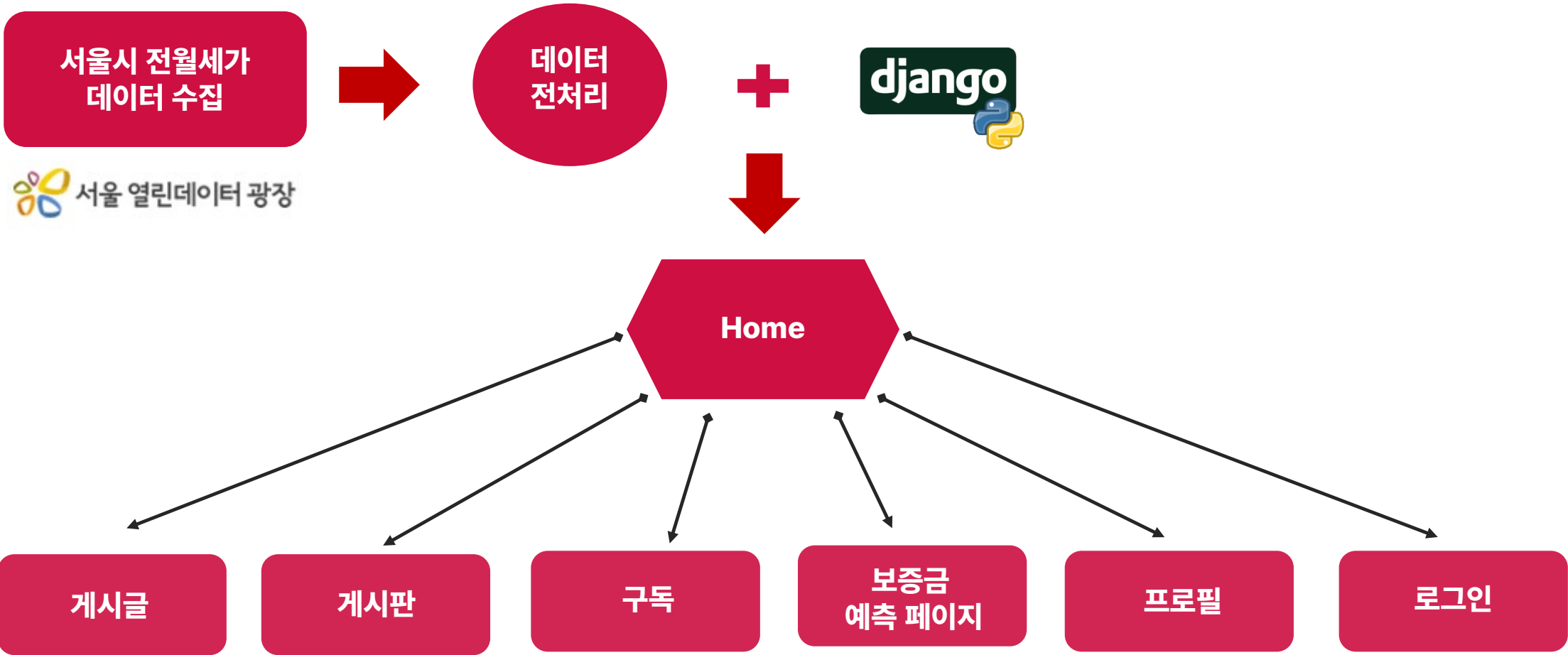
8000

총전임대료(만원):

100

Predict

예측 보증금은
9182.7(만원) 입니다.



Part 4 >> WBS

WBS	03월 18일	03월 19일	03월 20일	03월 21일	03월 22일~27일
기획	기획				
		장고 웹개발 기본 기능 구현 및 보증금 예측페이지 화면단 구성			
데이터수집	서울 열린데이터 광장 데이터 크롤링 및 전처리				
대시보드 구현		(회원가입, 로그인, 게시글, 댓글, 게시판,구독) 기능 구현			
			보증금 예측 페이지 구현		
코드 작성		기능 별 백엔드/프론트엔드 코드작성		머신러닝 코드	
최종					최종 보고

1. 커스터마이징의 한계

- 장고에서 제공되는 폼을 활용하여 프론트를 구성했기에 디자인이 단조로웠음과 동시에 변경이 까다로웠으며 기술적 이슈로 옵션을 변경해보지 못하였음 (ex. 회원가입 입력 폼, 회원정보 수정 폼 등)

→ 향후 UI/UX 디자인 리뉴얼 필요성 있음

<단조로운 디자인>

2. 보증금 예측 단순 기능구현

- 현재 단순히 기능만 구현한 수준으로 개발을 했으며 사용자 경험을 개선할 만한 외부 서비스 통합 요소의 필요성을 느꼈으며 이에 대한 추가 학습이 필요

→ 향후 예측된 결과를 이메일 서비스, 소셜 미디어 등을 연동하여 더 사용자 친화적인 서비스로 구성

3. 확장성 확보

- DB연동이나 서버와 관련된 작업을 아직 학습하지 않은 관계로 전혀 적용하지 못했음 따라서 대용량 데이터 처리나 서버 자원 확장 등을 고려하지 못함

→ 향후 학습하여 차기 프로젝트에 적용할 예정

4. 보안

- 애플리케이션의 보안을 강화하기 위해 취약점을 식별해야 하나 사용자 입력 값의 검증, 보안 헤더 추가, 보안 패치 및 업데이트 적용 등을 고려하지 못했음

5. 코드 리팩터링

- 코드의 가독성을 높이고 유지 보수성을 개선하기 위한 코드 리팩터링을 수행하지 못함.

→ 향후 관련 추가 학습 필요성 느낌

The background is a solid red color. It features several circular shapes: a large blue circle in the top right corner, a medium blue circle in the bottom left corner, and four smaller red circles of varying sizes scattered across the red field. The text "Thank you" is centered in the middle of the image.

Thank you