

소통으로 완성되는 게임 커뮤니티

1팀 : 이태주, 정현두, 김창회

Contents

01	주제
02	데이터 수집 및 전처리
03	게임 가격 정보 및 댓글
04	문제점 및 개선 방안
05	결론 및 향후 계획

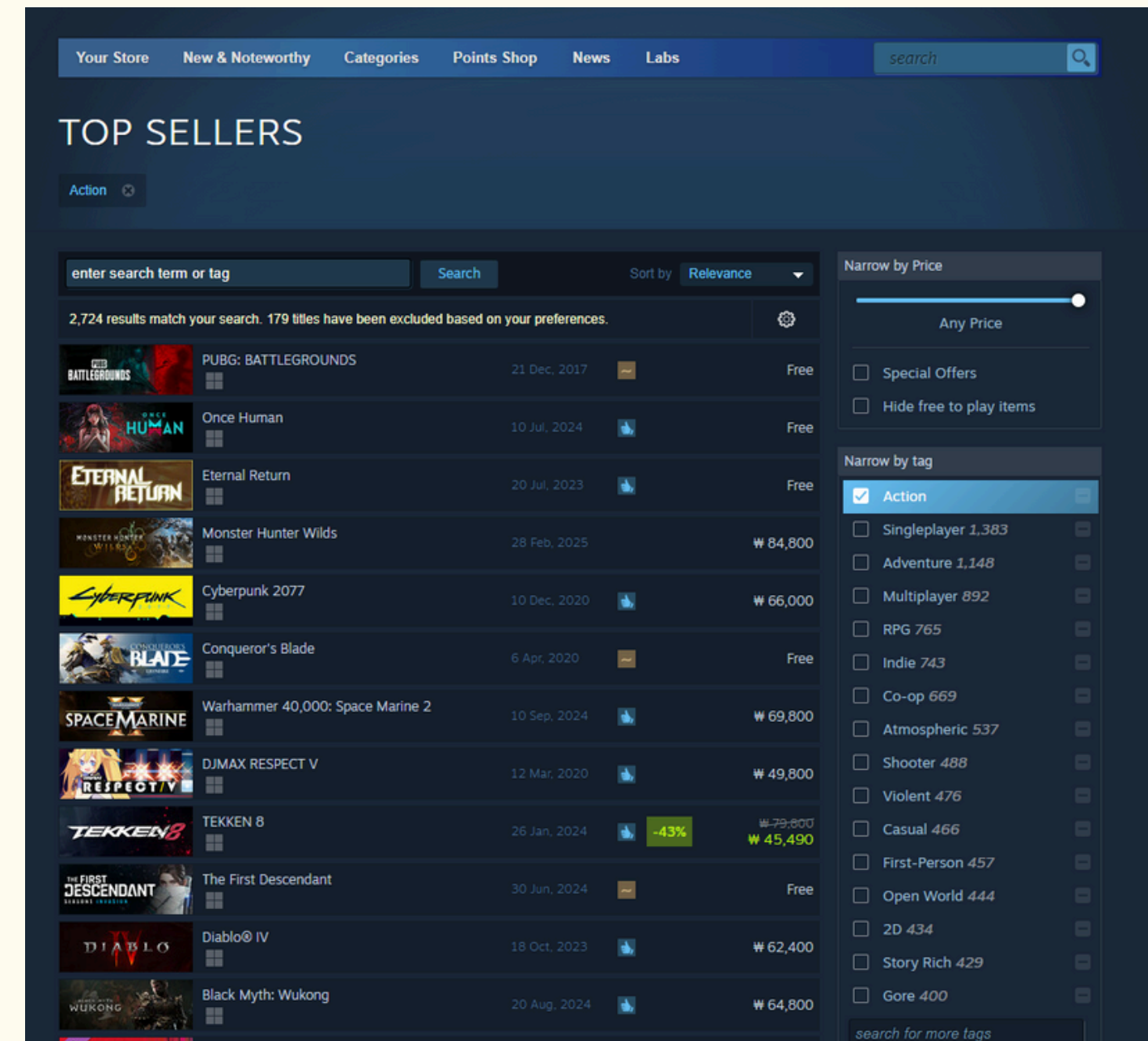
01 주제

주제 소개

판매 순위가 높은 각 장르의 게임에 대한 사용자 댓글을 보여주는 웹페이지를 소개합니다. 또한, 사용자들이 자유롭게 의견을 나누고 토론할 수 있는 게시판 기능을 추가하여 커뮤니티 소통을 강화했습니다.

문제 제기

사용자 간 소통이 원활하지 않아 의견 공유에 한계가 있습니다. 이를 해결하기 위해 게시판 기능을 추가하여 사용자들이 자유롭게 의견을 나누고 토론할 수 있는 공간을 제공합니다.



02 데이터 수집 및 전처리

데이터 수집

게임 id 크롤링

스팀 api

데이터 전처리

댓글 및 댓글 날짜

게임 출시 날짜

게임 가격

게임 플레이 시간

02 데이터 수집 및 전처리

```
import re

# 통화 기호에 따라 숫자를 변환하는 함수
def extract_numbers(price_str):
    if isinstance(price_str, str): # price_str이 문자열인지 확인
        # 통화 기호에 따라 환율을 설정
        if '₩' in price_str:
            multiplier = 1 # 원화: 그대로 사용
        elif 'CDN$' in price_str:
            multiplier = 950 # 캐나다 달러 기준, 예: 1 CAD = 950 KRW
        elif '$' in price_str:
            multiplier = 1300 # 미국 달러 기준, 예: 1 USD = 1300 KRW
        elif '€' in price_str:
            multiplier = 1400 # 유로 기준, 예: 1 EUR = 1400 KRW
        elif '¥' in price_str:
            multiplier = 10 # 엔화 기준, 예: 1 JPY = 10 KRW
        elif 'R$' in price_str:
            multiplier = 300 # 브라질 헤알 기준, 예: 1 BRL = 300 KRW
        else:
            multiplier = 1 # 통화 기호가 없는 경우 그대로 사용

        # 소수점이 포함된 숫자만 추출 (십표와 공백 제거)
        numbers = re.sub(r'^\d.', '', price_str)

        # 추출한 숫자 문자열을 소수로 변환한 후 환율 적용
        return float(numbers) * multiplier if numbers else 0 # 숫자에 통화 단위 곱하기
    return 0 # 문자열이 아닐 경우 0 반환

# 'game_price' 열 변환
Example1['game_price'] = Example1['game_price'].replace('가격 정보 없음', '0') # 가격 정보가 없는 경우 0으로 대체
Example1['game_price'] = Example1['game_price'].apply(extract_numbers) # 각 가격 문자열에 대해 함수 적용
```

게임 가격 전처리

게임 출시 날짜 전처리

```
# 날짜 변환 함수 정의
def parse_date(date_str):
    # 이탈리아어 월 이름을 영어로 변환
    for italian_month, english_month in month_map.items():
        if italian_month in date_str:
            date_str = date_str.replace(italian_month, english_month)

    # 여러 가지 날짜 형식 처리
    try:
        # 'Sep 19, 2024' 형식
        return pd.to_datetime(date_str, format='%b %d, %Y')
    except ValueError:
        pass
    try:
        # '19 Sep, 2024' 형식
        return pd.to_datetime(date_str, format='%d %b, %Y')
    except ValueError:
        pass
    try:
        # '9/set./2024' 형식 (이탈리아어 월 변환 후 처리)
        return pd.to_datetime(date_str, format='%d/%b/%Y')
    except ValueError:
        pass

    return pd.NaT # 변환 실패 시 NaT 반환

# 데이터프레임의 날짜 열 변환
Example1['game_release_date'] = Example1['game_release_date'].apply(parse_date)
Example1['game_release_date'].isnull().sum()
Example1['game_release_date'].info()
```

02 데이터 수집 및 전처리

```
def is_english(text):  
    if isinstance(text, str):  
        return bool(re.match(r'^[\x00-\x7F]*$', text))  
    return False  
  
indices_to_drop = Example1[~Example1['review_text'].apply(is_english)].index  
  
Example1.drop(indices_to_drop, inplace=True)  
  
Example1
```

댓글 날짜

```
Example1['playtime_at_review'].value_counts()  
Example1['playtime_at_review'].isnull().sum() # 417  
Example1['playtime_at_review'].value_counts().sum() # 6581  
Example1['playtime_at_review']=Example1['playtime_at_review']/60  
Example1['playtime_at_review']=Example1['playtime_at_review'].fillna(0)  
Example1['playtime_at_review'].isnull().sum()  
Example1['playtime_at_review'].astype(int)
```

사용자 플레이 시간

```
Example1['timestamp_created'].isnull().sum() # 0  
Example1['timestamp_created'].value_counts().sum() # 6581
```

Unix 타임스탬프

```
import pandas as pd
```

```
Example1['timestamp_created']  
Example1['timestamp_created']=pd.to_datetime(Example1['timestamp_created'], unit='s')
```

게임 댓글 전처리

댓글 날짜 및 플레이시간 전처리

04 문제점 및 개선 방안

game_release_date	game_price	review_text	review_score	language	author	playtime_at_review	timestamp_created
Dec 21, 2017	가격 정보 없음	its a learning curve, but worth it. hands down...	True	english	76561198150855629	1597	1727240369
Dec 21, 2017	가격 정보 없음	good	True	english	76561199006157895	3008	1727235379
Dec 21, 2017	가격 정보 없음	My first game, I was smacked in the head with ...	True	english	76561198856507936	155	1727233525

문제점

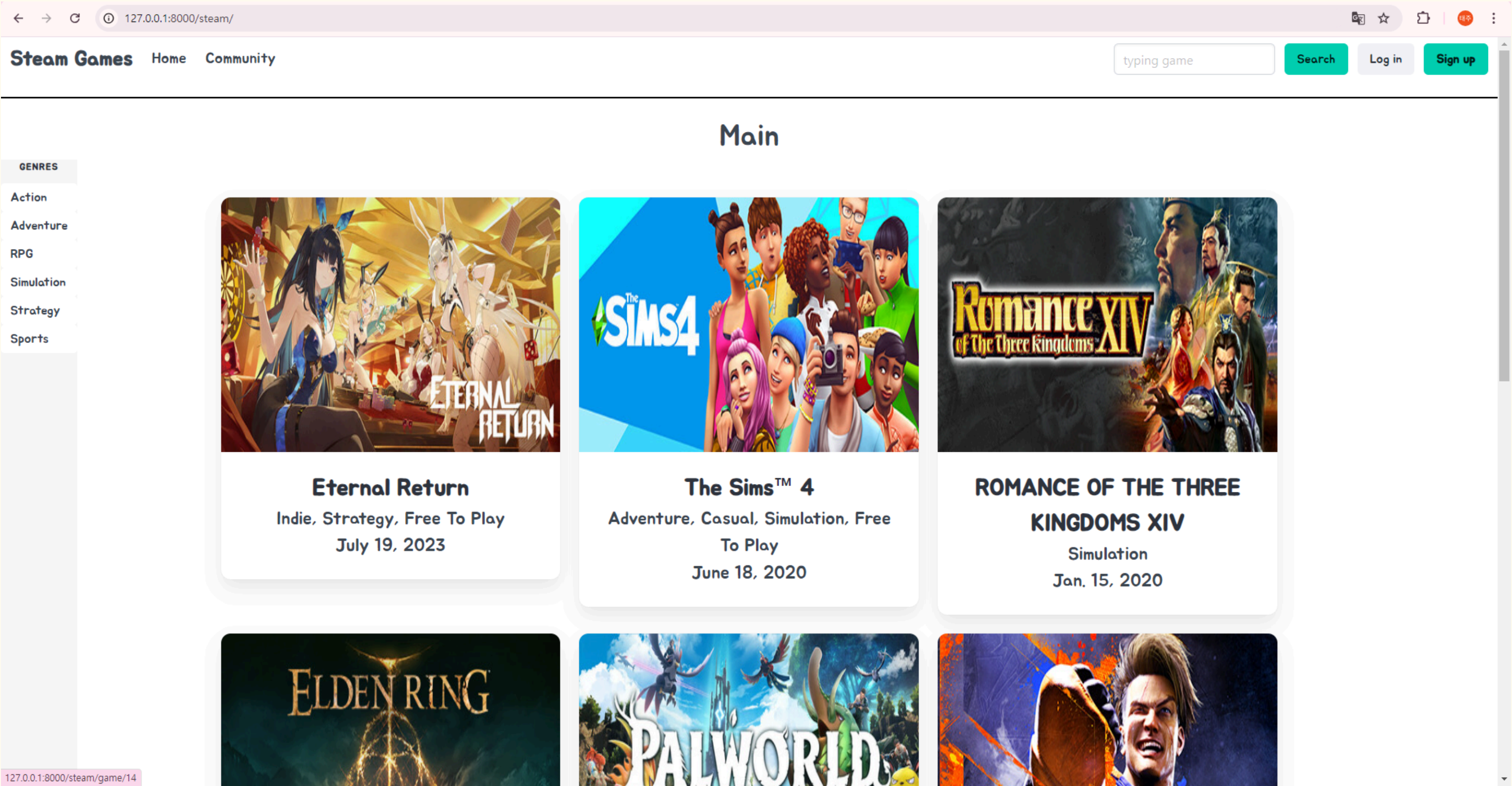
데이터 결측치, 이상치 발견, 데이터 형식 불일치

game_release_date	game_price	review_text	review_score	language	author	playtime_at_review	timestamp_created
2017-12-21	0.0	its a learning curve, but worth it. hands down...	True	english	76561198150855629	26.616667	2024-09-25 04:59:29
2017-12-21	0.0	good	True	english	76561199006157895	50.133333	2024-09-25 03:36:19
2017-12-21	0.0	My first game, I was smacked in the head with ...	True	english	76561198856507936	2.583333	2024-09-25 03:05:25

문제점 개선

이상치 제거, 결측치 처리, 데이터 변환을 통해서 해결

03-0 홈 화면

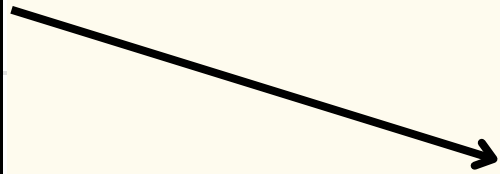


03-1 게임 데이터 스키마 구성

```
# 게임 정보 테이블
class GameInfo(models.Model):
    id = models.AutoField(primary_key=True) # integer - pk - auto inc
    title = models.CharField(max_length=255) # 게임 이름
    genres = models.CharField(max_length=255) # 게임 장르
    image = models.URLField() # 게임 이미지 URL
    price = models.DecimalField(max_digits=10, decimal_places=2) # 게임 가격
    release_date = models.DateField() # 게임 출시 날짜
    mean_playtime = models.FloatField(null=True, blank=True) # 평균 플레이 시간

# 게임 리뷰 테이블
class GameReview(models.Model):
    game_info = models.ForeignKey(GameInfo, on_delete=models.CASCADE)
    content = models.TextField() # 리뷰 텍스트
    recommend = models.IntegerField() # 리뷰 점수
    playtime = models.FloatField() # 리뷰 작성 당시 플레이 시간
    created_at = models.DateTimeField() # 리뷰 작성 시간
```

Game_info
id
name
Genres
image
mean_playtime
release_date
price



Game_review
id
game_id(FK)
recommend
score(bool)
playtime
<u>created_at</u>

03-1 게임 페이지

← → ↺ 127.0.0.1:8000/steam/game/14/ ☆ 📁 🔴 149 ⋮

Steam Games Home Community typing game Search Log in Sign up

GENRES

Action

Adventure


RPG

Simulation

Strategy

Sports

상세 페이지



제목 : Eternal Return

장르 : Indie, Strategy, Free To Play


가격 : 0원

출시일 : July 19, 2023

평균 플레이타임 : 47시간

총 리뷰 : 95개

추천 비율 : 89%



게임에 대한 정보와 리뷰, 워드클라우드

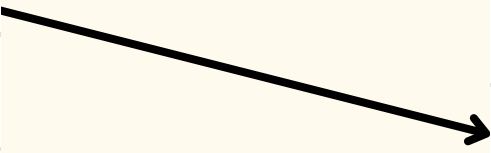
03-2 자유게시판 스키마

```
class community(models.Model):
    id = models.AutoField(primary_key=True) # integer - pk - auto increment
    title = models.CharField(
        "제목",
        max_length=255,
        validators=[validators.MinLengthValidator(1, "제목은 1글자 이상이어야 합니다.")],
    ) # varchar(255)
    content = models.TextField(
        "내용",
        validators=[validators.MinLengthValidator(1, "내용은 1글자 이상이어야 합니다.")],
    ) # text
    nickname = models.CharField(
        "닉네임",
        max_length=50,
        validators=[validators.MinLengthValidator(1, "닉네임 입력하쇼.")],
    )
    created_at = models.DateTimeField(
        auto_now_add=True) # 추가될 때 default로 현재 시간
    updated_at = models.DateTimeField(
        auto_now=True) # 추가 or 업데이트 될 때 default로 현재 시간
    is_deleted = models.BooleanField(default=False)
    hits = models.IntegerField(default = 0)
    recommend = models.IntegerField(default = 0)
    recommended_by = models.ManyToManyField(User, related_name='recommended_posts', blank=True)

class com_comment(models.Model):
    community = models.ForeignKey(community, on_delete=models.SET_NULL, null=True)
    nickname = models.CharField(max_length=50)
    content = models.CharField(max_length=255)
    created_at = models.DateTimeField(auto_now_add=True)
    is_deleted = models.BooleanField(default=False)
```

Community
id
title
content
nickname
created_at
hits
recommend
recommended_by

Com_comment
id
community_id(FK)
nickname
content
created_at
is_deleted



03-2 커뮤니티 페이지(자유게시판)

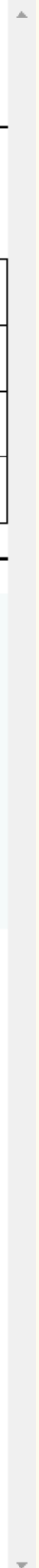
←→↺127.0.0.1:8000/steam/community/🔍☆📁⬇️태주⋮

Steam GamesHomeCommunitySearchLog inSign up

Community

제목	닉네임	등록일	조회수	추천수
기도문	taelee12	Sept. 26, 2024, 8:44 a.m.	52	2
fwefewfw	taelee12	Sept. 27, 2024, 7:28 a.m.	11	1
ㅇㅇㅇㅇ	taeju	Sept. 30, 2024, 5:01 a.m.	7	1

제작자 1팀
https://github.com/Tiaozo/TJ_web



03-2 커뮤니티 페이지(자유게시판)

Steam Games

HomeCommunity

typing game

Search

Logout

ㅎㅎㅎ

글 수정

글 삭제

작성자 : taeju	Sept. 30, 2024, 5:01 a.m.	조회 : 7	추천 : 1
-------------	---------------------------	--------	--------

반갑나

추천 취소

총 댓글 : 1개

Content

댓글 입력

작성

taeju

안녕 좋은 글이네

Sept. 30, 2024, 6:52 a.m.

수정

삭제

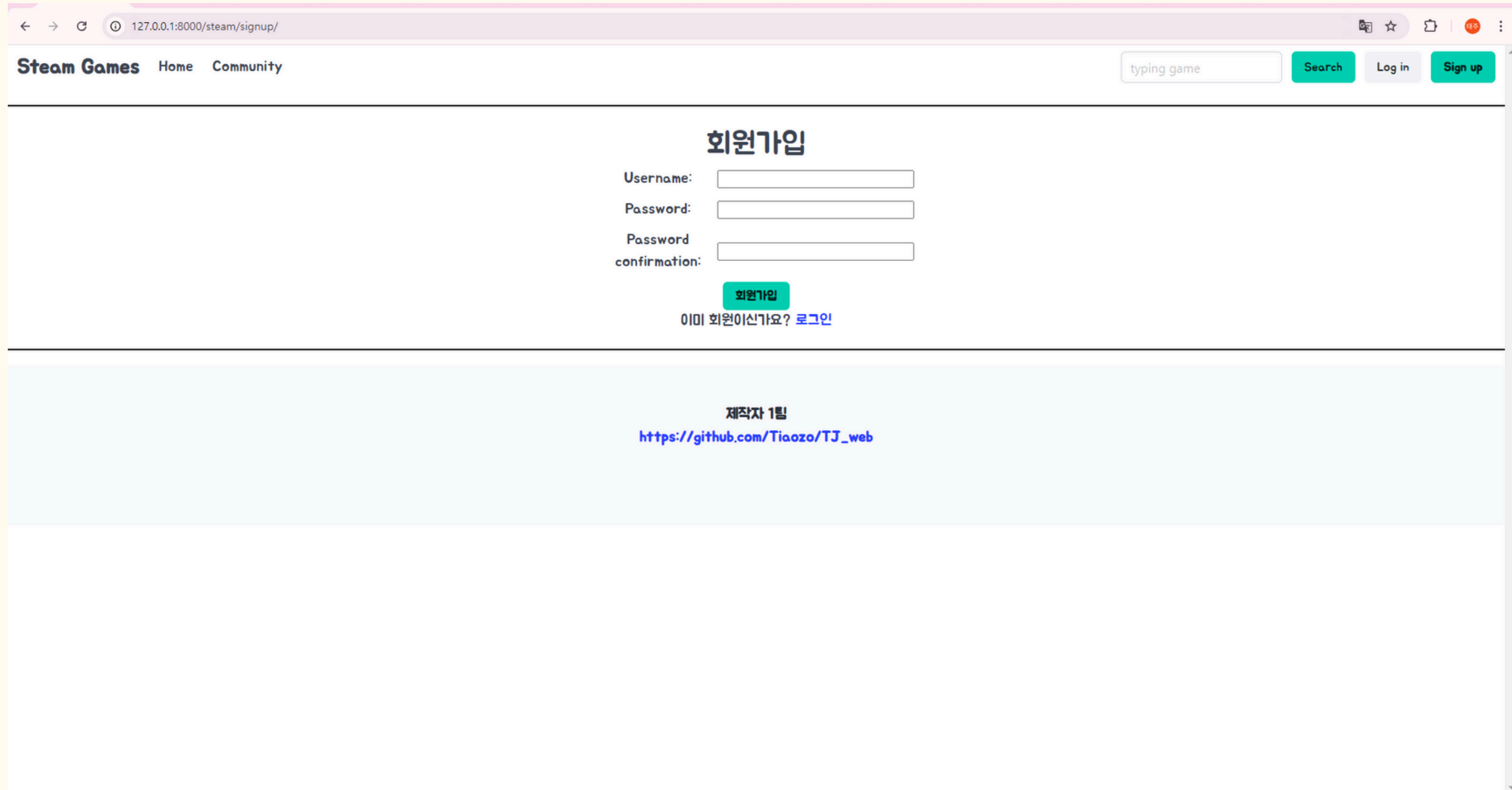
제작자 1팀

https://github.com/Tiaozo/TJ_web

03-3 로그인 페이지



03-3 회원가입 페이지



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/steam/signup/". The page has a header with "Steam Games" and navigation links "Home" and "Community". A search bar contains "typing game" with "Search", "Log in", and "Sign up" buttons. The main content area is titled "회원가입" (Sign Up) and contains a form with fields for "Username:", "Password:", and "Password confirmation:". Below the form is a green "회원가입" button and a link "이미 회원이신가요? 로그인" (Already a member? Log in). The footer area is light blue and contains the text "제작자 1팀" (Developer 1 Team) and the URL "https://github.com/Tiaozo/TJ_web".

Steam Games Home Community

typing game Search Log in Sign up

회원가입

Username:

Password:

Password confirmation:

[회원가입](#)

[이미 회원이신가요? 로그인](#)

제작자 1팀

https://github.com/Tiaozo/TJ_web

05 결론 및 향후 계획

결론

사용자 소통 증진

정보 공유

사용자 경험 향상

향후 계획

장르 추천 챗봇

출시 예정 게임 데이터 수집

출시 예정 게임 게시판