

FINAL PJT

금융 상품 비교 애플리케이션

챕터의 포인트

- 목표
- 개발 환경 및 버전
- 필수 요구사항
- 제출

목표

| 프로젝트 목표

- 싸피 출신 개발자 최차윤은 드디어 취업에 성공 했습니다.
- 결혼 후 자가마련을 위해 재테크 계획을 세우려고 합니다.
- 그런데, 신입사원이라 은행에 갈 시간이 없어 금융 정보를 한 곳에 알 수 있는 웹 애플리케이션을 직접 만들어 보려고 합니다.

차윤이가 필요한 메인 기능 네 가지

1. 예금 & 적금 금리 비교
2. 신혼 여행을 위한 환율 계산기
3. 내 집 주변 은행 검색
4. 나에게 맞는 상품 추천(가장 중요)

test | 로그인

Bankbook

예금비교 환율계산기 은행 지도 게시판

예금비교

정기예금
금액 조건을 입력하세요

은행을 선택하세요

전체

예치기간

전체기간

확인

정기예금 | 정기적금

공시 제출월	금융회사명	상품명	6개월	12개월	24개월	36개월
202306	우리은행	WON플러스예금	3.67	3.73	3.36	3.33
202306	한국스탠다드차타드은행	e-그린세이버예금	3.9	4.1	-	-
202306	대구은행	DGB주거예우대예금(첫만남고객형)	3.11	3.85	3.87	3.89
202306	대구은행	DGB행복파트너예금(일반형)	2.87	3.61	3.63	3.65
202306	대구은행	DGB합계예금	-	3.75	-	-
202306	대구은행	IM스마트예금	3.05	3.75	3.45	3.45
202306	대구은행	안녕,축도야예금	-	3.61	-	-
202306	부산은행	LIVE정기예금	3.5	3.65	2.8	2.85
202306	부산은행	더(The) 특판 정기예금	3.85	4	3.65	3.35
202306	광주은행	미즈물리리창기예금	-	3.71	3.87	3.97

화면 예시

| 차윤이가 필요한 메인 기능 네 가지

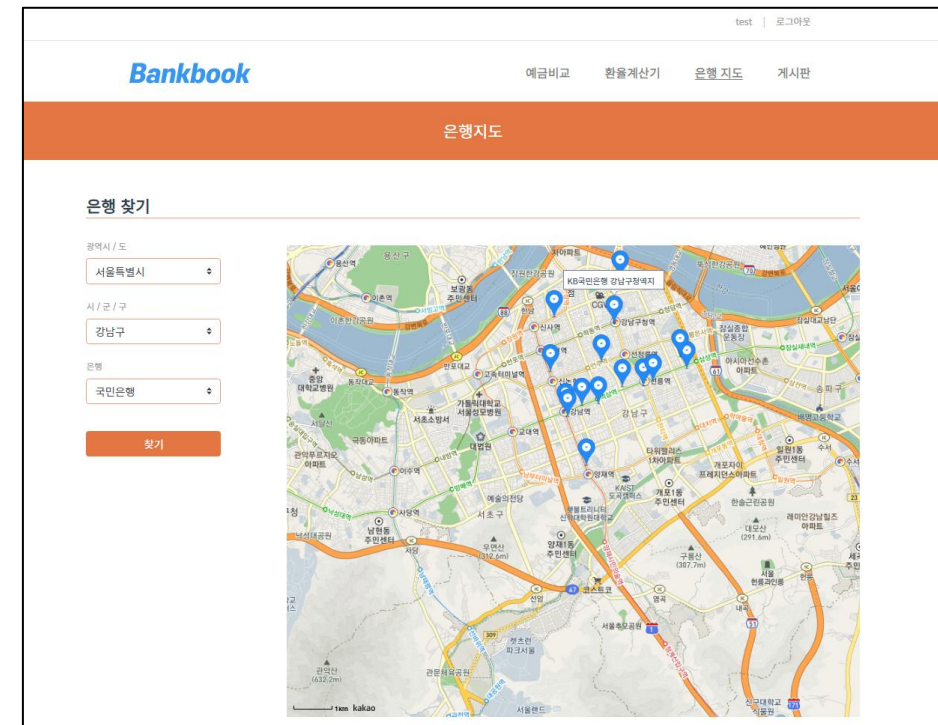
1. 예금 & 적금 금리 비교
2. **신혼 여행을 위한 환율 계산기**
3. 내 집 주변 은행 검색
4. 나에게 맞는 상품 추천(가장 중요)

The screenshot shows a web application interface for 'Bankbook'. At the top, there is a navigation bar with the logo 'Bankbook' and links for '예금비교', '환율계산기', '은행 지도', and '게시판'. Below the navigation bar is a green header with the text '환율'. The main content area is titled '환율 계산기' and contains a form with two dropdown menus. The first dropdown menu is set to '미국 달러' and the second is set to '대한민국(원화)'. Below the form, there is a small note: '* 엔화 / 인도네시아 루피아는 100단위, 나머지는 모두 1단위'.

화면 예시

| 차윤이가 필요한 메인 기능 네 가지

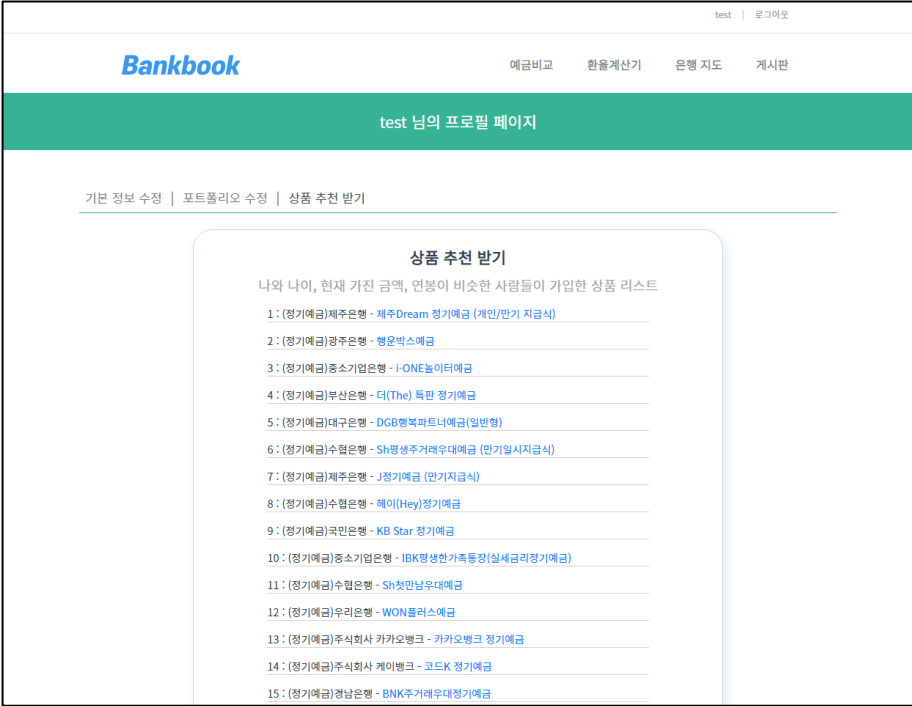
1. 예금 & 적금 금리 비교
2. 신혼 여행을 위한 환율 계산기
3. 내 집 주변 은행 검색
4. 나에게 맞는 상품 추천(가장 중요)



화면 예시

| 차윤이가 필요한 메인 기능 네 가지

- 1. 예금 & 적금 금리 비교
- 2. 신혼 여행을 위한 환율 계산기
- 3. 내 집 주변 은행 검색
- 4. 나에게 맞는 상품 추천(가장 중요)



화면 예시

| 프로젝트를 완성하기 위해 필요한 지식

- **사용 API**
 - 한국수출입은행 환율정보 API
 - 카카오맵 API
 - 금융감독원 API
- **주요 라이브러리**
 - Bootstrap
 - chartjs
 - Django REST Framework

| 프로젝트를 임하는 마음가짐

- 이 프로젝트는 여러분들의 취업에 활용할 수 있도록 나만의 애플리케이션을 만드는 데 도움을 줍니다.
- 필수 내용을 최대한 빨리 구현해 주세요.
- 이후에는 **나만의 프로젝트로 변경**해주시면 됩니다.
- 디자인도 변경해보고, 더 넣고 싶은 기능들을 마음껏 넣어봅니다.
 - 추가하고 싶은 API 도 마음껏 넣어봅니다.
- 남들과는 다른 나만의 애플리케이션을 만들어보세요!

개발 환경 및 버전

| 필수 개발 환경

- Python 3.9.x
- Django 4.2.x
- Node.js LTS
- Vue 3 & Pinia

| 사용 아키텍처

- 자유롭게 구성합니다. (아래 예시 참조)
 1. Django & Vanilla JavaScript
 2. Django REST Framework & Vue3
- 각 팀의 아키텍처를 설명할 수 있는 그림 혹은 설명을 README.md에 포함시켜 작성합니다.

필수 요구사항

| 개요

- 필수 요구사항은 프로젝트 내에 반드시 포함되어야 합니다.
- **구현을 시작하기 전에 반드시 설계 및 역할 분담 과정을 먼저 수행해야 합니다.**
- 이 외 추가적인 기능 및 반응형 디자인 등은 팀 별로 자유롭게 수행합니다.

1. 메인 페이지

2. 회원 커스터마이징

3. 예적금 금리 비교

4. 환율 계산기

5. 근처 은행 검색

6. 커뮤니티(게시판)

7. 프로필 페이지

8. 금융 상품 추천 알고리즘

9. 생성형 AI

10. README

11. 기타

| 1. 메인 페이지

- 본 프로젝트의 목표는 '금융 상품 비교 애플리케이션' 서비스를 완성하는 것입니다.
- 각 팀의 서비스를 소개할 수 있도록 메인 페이지를 구성합니다.
- [추천] Carousel 및 적절한 이미지를 활용하여 화면을 구성합니다.

| 2. 회원 커스터마이징

- 회원 관리를 할 수 있는 기능(회원 가입, 로그인, 로그아웃 등)을 구성합니다.
 - 회원 관리 라이브러리(allauth & dj-rest-auth) 등을 자유롭게 사용할 수 있습니다.
- Django의 기본 User 를 상속받아 커스텀 User 를 구현합니다.
 - 커스텀 User 를 활용하기 위해 적절한 Serializer 를 구성합니다.
- 필수적으로 포함되어야 할 필드는 다음과 같습니다.
 - **유저이름, 이메일, 가입한 상품 목록을 저장할 수 있는 필드**(blank=True, null=True 옵션을 적절히 활용합니다.)
 - [힌트] 가입한 상품 목록을 저장할 수 있는 필드
 - 쉼표(,)로 구분된 텍스트, 리스트 등 관리하기 편한 데이터 타입과 규칙으로 구성합니다.

| 3. 예적금 금리 비교 - 데이터 저장

- 본 프로젝트를 완성하기 위해서는 '금융 상품 데이터' 가 필수적으로 필요합니다.
- 적절한 API 를 활용하여 금융 상품 정보를 가져옵니다.
 - 예시: 금융상품통합비교공시 API
- 가져온 데이터를 DB 에 저장합니다.
- 이미 DB 에 존재하는 데이터는 새로 저장하지 않도록 구성합니다.
 - [힌트] Django 의 filter(Qureyset API) 를 활용합니다. (이미 존재하는 상품이라면 통과, 없다면 저장)

3. 예적금 금리 비교 - 전체 조회

- 상품 목록을 볼 수 있도록 적절한 화면을 구성합니다.
- 가입 기간에 따라 다른 금리 및 기타 금융 상품 특성을 고려하여 화면을 구성합니다.
- 은행을 선택하여 목록을 필터링 할 수 있도록 추가적인 기능을 구성합니다.
- 은행 선택 시, 해당 은행의 금융 상품 목록만 조회하도록 화면을 구성합니다.
- 특정 상품 클릭 시 상세 정보를 볼 수 있도록 적절한 화면을 구성합니다.

정기예금 정기적금

정기예금

검색하기
검색 조건을 입력하세요

은행을 선택하세요
전체은행

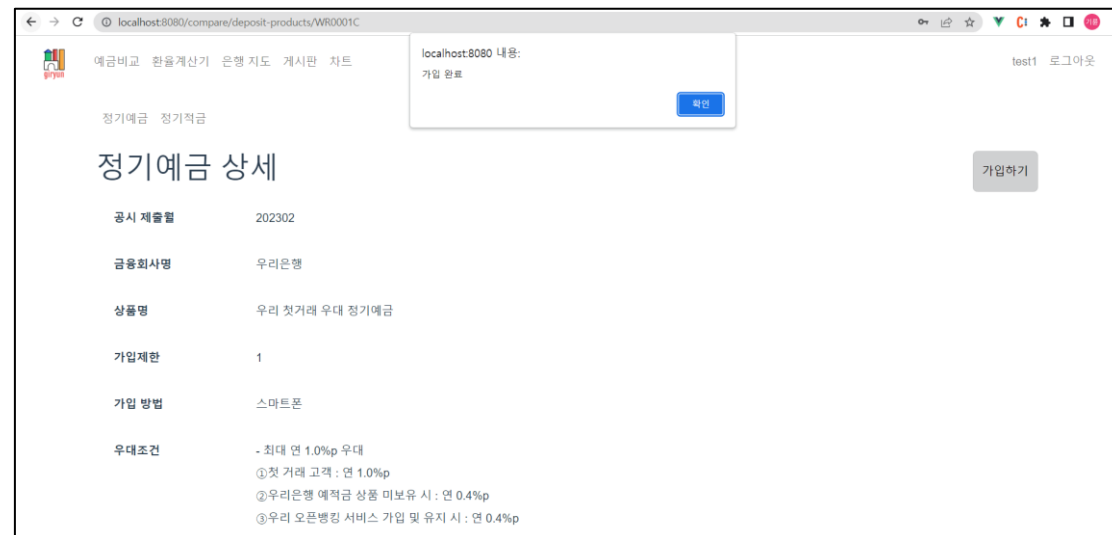
예치기간
전체기간

확인

공시 제호 일	금융회사명	상품명	6개월 (Click to sort ascending)	12개월 (Click to sort ascending)	24개월 (Click to sort ascending)	36개월 (Click to sort ascending)
202302	우리은행	WON플러스예금	3.55	3.8	3.64	3.66
202302	우리은행	우리 찾기예 후대 정기예금	3.9	4	4	4
202302	한국스탠다드차타 드은행	◆그린세이브예금	3.7	3.9	-	-
202302	대구은행	DGB주거자유대예금(일반납고 형)	3.31	4.05	4.07	4.09
202302	대구은행	DGB행복파트너예금(일반형)	3.12	3.86	3.88	3.9
202302	대구은행	DGB행예금	-	4.05	-	-
202302	대구은행	IM스마트예금	3.05	3.75	3.8	3.85
202302	대구은행	안녕, 독도야예금	-	3.61	-	-
202302	부산은행	LIVE정기예금	2.55	2.75	2.8	2.85

3. 예적금 금리 비교 - 상세 조회

- 해당 금융 상품에 대한 자세한 정보를 출력할 수 있도록 적절한 화면을 구성합니다.
- 가입 기간에 따라 다른 금리 및 기타 금융 상품 특성을 고려하여 화면을 구성합니다.
- [가입하기] 버튼을 추가하여, 클릭 시 해당 회원의 [가입한 상품 목록]에 상품 ID를 추가합니다.
- 로그인 된 사용자에게만 해당 버튼을 출력합니다.



3. [도전과제] 예적금 금리 비교 - 상세 조회

- 관리자는 금리 정보를 수정할 수 있도록 구현합니다.
- 상품의 금리 정보가 수정되면, 가입한 유저의 이메일로 메일이 전송되도록 구현합니다.
- Django 에 내장된 모듈을 활용합니다.(django.core.mail)
- 메일을 전송해주는 메일 호스트 서버는 네이버, Twilio 등을 활용합니다.
- 서비스의 과금 정책에 주의하여 구현합니다.

메일 예시



| 4. 환율 계산기

- 적절한 API 를 활용하여 현재 환율에 대한 정보를 가져옵니다.
 - 예시: 한국수출입은행 환율정보 API
- 국가를 선택할 수 있도록 구성합니다.
- 2가지 입력을 받도록 구성합니다.
 - 원화 입력 시: 선택한 국가의 통화로 변환된 값을 출력합니다.
 - 타국 통화 입력 시: 해당 통화를 원화로 변환한 값을 출력합니다.

환율계산기

미국 달러 ▾

1

\$

1317

₩

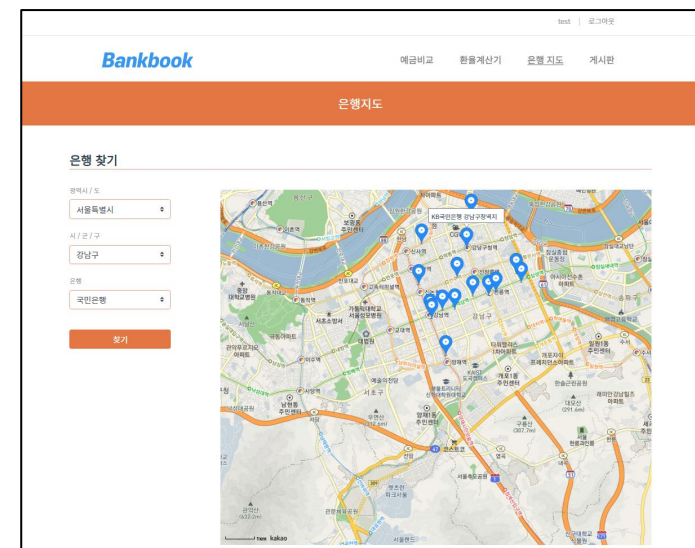
* 엔화, 인도네시아 루피아는 100 단위, 나머지는 모두 1 단위

| 4. 환율 계산기(참고)

- 프론트(Vue) 에서 환율 API 에 요청을 보내면 CORS 에러가 발생합니다.
- 백엔드에서 데이터를 가져와 프론트로 전달할 수 있도록 구성합니다.
- 로직 예시
 - [프론트] 화면 접속 시 백엔드로 데이터 요청
 - [백엔드] 환율 API 에 데이터 요청
 - [백엔드] 환율 API 를 통해 받아온 데이터를 프론트로 전송
 - [프론트] 백엔드를 통해 받아온 데이터를 화면에 출력

5. 근처 은행 검색

- 적절한 API 를 활용하여 지도를 표시합니다.
- 예시: [Kakao Maps API](#)
- 위치와 은행을 선택(입력)할 수 있도록 구성합니다.
- 선택 시, 해당 위치 근처의 은행 정보를 적절하게 출력합니다.



| 6. 커뮤니티(게시판)

- 회원간 소통 할 수 있는 커뮤니티 기능(게시판)을 구현합니다.
 - 게시판이 아닌 다른 종류의 소통 공간을 구성해도 괜찮습니다.
- 회원의 권한에 따라 다른 동작을 하도록 구성합니다.
 - 예시: 본인이 작성한 게시글 및 댓글만 삭제, 수정 가능하도록 구성합니다.
- 소통 방식은 자유롭게 구성합니다.
 - 예시: 금융 상품 리뷰 게시판, 내가 가입한 상품 자랑 게시판 등

7. 프로필 페이지

- 회원의 기본 정보를 출력할 수 있도록 적절한 화면을 구성합니다.
- 회원의 정보를 수정하기 위한 적절한 화면을 구성합니다.
- 내가 가입한 금융 상품 리스트를 출력할 수 있도록 화면을 구성합니다.
- 차트 라이브러리를 활용하여 가입한 상품 금리 정보를 그래프로 출력합니다.

예금비교

환율계산기

은행 지도

계시판

차트

test1 로그아웃

test1 님의 프로필 페이지

기본 정보 수정

포트폴리오 수정

상품 추천 받기

회원번호

10001

ID

test1

Email

이메일을 설정해주세요

수정하기

Nickname

test1

수정하기

나이

50

수정하기

현재 가진 금액

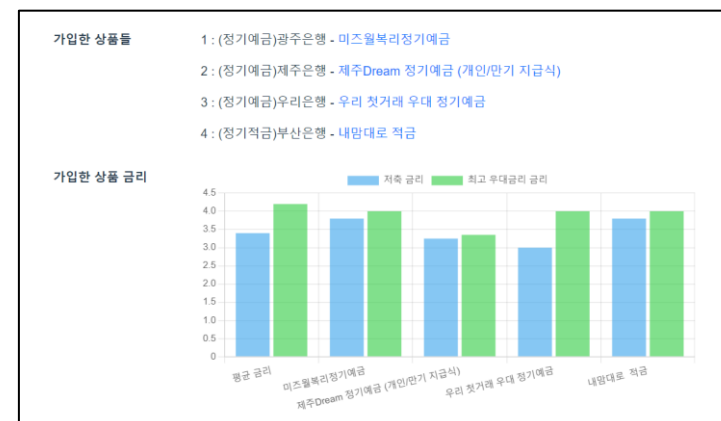
1,333,333

수정하기

연봉

32,432

수정하기



| 8. 금융 상품 추천 알고리즘

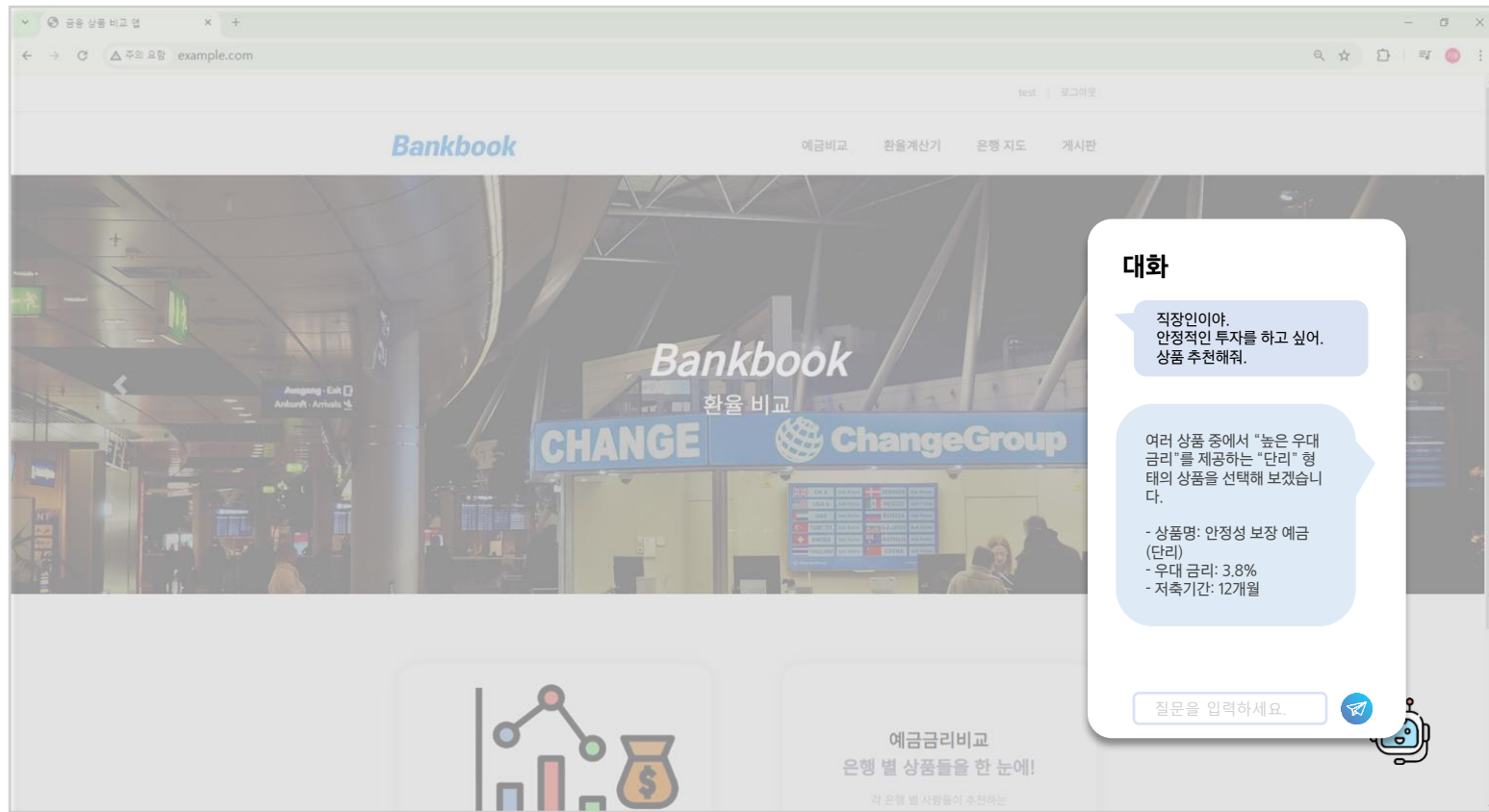
- 추천 방식은 자유롭게 구성합니다.
- 반드시 최소 1개 이상의 금융 상품을 추천 받을 수 있도록 구성합니다.
- 어떠한 방식으로 추천 시스템을 구현 했는지 기술적으로 설명할 수 있어야 합니다.

| 예시1 - OpenAI API 활용

- OpenAI API를 활용한 상품 추천 챗봇을 구현합니다.
- 예시 시나리오
 - [사전 준비] OpenAI API를 활용하여 “현재 존재하는 금융 상품 종류와 금리”를 사전에 학습 시킵니다.
 - [유저] “성향”, “현재 가입된 상품 종류” 등 본인의 상황을 설명합니다.
 - [챗봇] 사용자의 상황에 맞는 상품을 추천 해줍니다.
- 주의사항
 - 과금에 주의하여 사용해야 합니다.

예시1 - OpenAI API 활용

- 화면 예시 (챗봇 모양 클릭 시 대화창 출력)



| 예시2 - 나와 상황이 비슷한 사람들이 가입한 상품 10개 추천

- 회원 필드에 나이, 현재 가진 금액, 연봉 필드를 추가합니다.
- 10,000 명의 랜덤한 회원 데이터를 생성하고, DB에 저장합니다.
- 나와 나이, 현재 가진 금액, 연봉의 분포가 비슷한 사람들이 가입한 상품을 10개 추천합니다.
- 파이썬 데이터 분석 라이브러리(Numpy & Pandas)를 활용합니다.

test1 님의 프로필 페이지

기본 정보 수정

포트폴리오 수정

상품 추천 받기

상품 추천 받기

나와 나이, 현재 가진 금액, 연봉이 비슷한 사람들이 가입한 상품 리스트

나이, 금액, 연봉 정보를 입력해주세요.

test1 님의 프로필 페이지

기본 정보 수정

포트폴리오 수정

상품 추천 받기

상품 추천 받기

나와 나이, 현재 가진 금액, 연봉이 비슷한 사람들이 가입한 상품 리스트

1 : (정기예금)제주은행 - 제주Dream 정기예금 (개인/만기 지급식)

2 : (정기예금)광주은행 - 행운박스예금

3 : (정기예금)중소기업은행 - I-ONE놀이터예금

4 : (정기예금)부산은행 - 디(The) 특판 정기예금

5 : (정기예금)대구은행 - DGB행복파트너예금(일반형)

6 : (정기예금)수협은행 - Sh행생주거대출대예금 (만기일시지급식)

7 : (정기예금)제주은행 - J정기예금 (만기지급식)

8 : (정기예금)수협은행 - 해이(Hey)정기예금

| 예시3 - 포트폴리오를 활용한 상품 추천

- 회원 ID를 외래키로 하는 포트폴리오 테이블을 추가로 생성합니다.
- 저축성향, 최애 은행 등 개인을 나타낼 수 있는 데이터를 필드에 추가합니다.
- 데이터를 적절히 활용하여 자신에게 맞는 상품들을 추천해 줍니다.

test1 님의 프로필 페이지

기본 정보 수정
포트폴리오 수정
상품 추천 받기

포트폴리오 수정

회원번호

10001

ID

test1

저축성향

☐알뜰형 ☐도전형 ☐성실형

수정하기

최애 은행

우리은행

수정하기

| 9. 생성형 AI

- 서비스의 다양한 방면에서 생성형 AI 를 활용할 수 있습니다.
- 활용 예시
 - 추천 알고리즘 컨셉 아이디어 도출
 - 추천 알고리즘에 사용할 더미 데이터 생성하기 (금융 상품, 서비스 사용자 데이터 등)
 - 추천 알고리즘 구현 (챗봇, 알고리즘 로직 구현 등)
 - 서비스 성능 개선 (컴포넌트 구조 개선, ORM 성능 개선 등)
- 생성형 AI 를 활용한 서비스는 자유롭게 결정합니다.
- 발표 및 README 파일 제출 시 활용한 내용을 포함하여 제출합니다.

| 10. README

- 반드시 아래 내용이 작성되어 있어야 합니다.
 - I. 팀원 정보 및 업무 분담 내역
 - II. 설계 내용(아키텍처 등) 및 실제 구현 정도
 - III. 데이터베이스 모델링(ERD)
 - IV. 금융 상품 추천 알고리즘에 대한 기술적 설명
 - V. 서비스 대표 기능들에 대한 설명
 - VI. 생성형 AI 를 활용한 부분
 - VII. 기타(느낀 점, 후기 등)

| 11. 기타

- 최소한 6개 이상의 URL 및 페이지를 구성해야 합니다.
- HTTP Method 와 HTTP response status code 는 상황에 맞게 적절하게 반환되어야 하며, 필요에 따라 적절한 에러 페이지를 구성해야 합니다.
- .gitignore 파일을 사용하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 프로젝트명
 - final-pjt
 - final-pjt-front, final_pjt_back (두 서버를 모두 사용하는 경우)

참고사항

| Django 환경변수 관리

- API KEY 와 같이 외부에 노출하면 안되는 정보를 따로 관리합니다.
- django-environ 설치 - `(venv) $ pip install django-environ`
- .env 파일 작성

<주의사항> API_KEY와 = 사이에 공백이 있으면 오류가 발생합니다.

```
1 API_KEY='<발급받은 API KEY 입력>'
```

- 발급받은 API KEY 를 따옴표로 묶어서 문자열 형태로 입력합니다.
- 수정 후 서버를 다시 시작해야 반영됩니다.

| Django 환경변수 관리

- settings.py 에 아래 코드를 추가해 줍니다.

```
15  # Build paths inside the project like this: BASE_DIR / 'subdir'.
16  BASE_DIR = Path(__file__).resolve().parent.parent
17
18  √ import os
19  import environ
20
21  # 환경변수를 불러올 수 있는 상태로 설정합니다
22  env = environ.Env(DEBUG=(bool, True))
23
24  # 환경변수를 읽어올 파일을 설정합니다
25  √ env.read_env(
26      |     env_file=os.path.join(BASE_DIR, '.env')
27      | )
28
29  # 환경변수를 읽어옵니다.
30  API_KEY = env('API_KEY')
```

| Django 환경변수 관리

- settings 에 등록된 환경변수는 views.py 에서 다음과 같이 사용합니다.

```
from django.conf import settings  
  
API_KEY = settings.API_KEY
```

- .gitignore 파일에 .env 를 추가하여 API KEY 가 외부에 노출되지 않도록 설정합니다.

제출

| 제출 시 주의사항

- 제출 기한은 학사 시스템에 안내된 제출 기한을 따라야 합니다.
- 제출 위치 ※ 두 곳 모두 제출하며, 팀장이 대표로 제출합니다.
 1. 학사 시스템
 2. <https://lab.ssafy.com/>
 - 프로젝트 명: final-pjt / final-pjt-back, final-pjt-front
 - 반드시 각 반 담당 교수님을 Maintainer로 설정합니다.