

# PNU SW학습공동체 최종(중간)보고서

## 1. 프로젝트 소개

### 가. 배경 및 필요성

초기에는 은행 내부 직원이 금융상품 정보를 효율적으로 조회할 수 있도록 돕는 AI 챗봇을 기획했으나, 실제 데이터를 확보하는 데 있어 보안과 접근 제한의 현실적 제약이 있었고, 추후 프로젝트의 활용성을 고려하였을 때, 기획 방향을 바꿔야겠다고 판단했습니다.

이에 따라, 청년층을 위한 맞춤형 금융상품 추천 서비스로 방향을 전환하게 되었습니다. 청년들은 카드, 통장, 적금, 청약 등 다양한 금융상품을 접할 기회는 많지만, 연봉, 주거 형태, 소비 성향, 금융 지식 수준이 각기 다르고, 시중의 수많은 상품 중 본인에게 적합한 것을 고르기 어렵다는 어려움을 겪고 있습니다. 특히, 금융기관의 상품 설명은 지나치게 복잡하거나 파편화되어 있어 정보 접근성이 낮은 청년들에게 실질적인 도움이 되지 않는 경우가 많습니다. 따라서, 본 프로젝트는 사용자의 재정 상황과 금융 목표를 기준으로 정보를 정제하고 추천하는 AI 서비스를 통해 청년층이 스스로 금융상품을 이해하고 선택할 수 있도록 돕고자 합니다.

### 나. 개발목표 및 주요내용, 세부내용 등

본 프로젝트의 목표는 청년층의 재정 상황과 금융 목표에 따라 최적의 금융상품을 추천해주는 AI 챗봇을 개발하는 것입니다.

#### ◎ 주요 내용

사용자는 연령, 연소득, 주거 형태, 직업, 금융 목표 총 5가지를 입력합니다. 본 챗봇은 해당 정보를 기반으로 다음과 같은 3단계 과정을 거쳐 사용자에게 걸맞은 금융상품을 추천합니다:

##### [1단계] 사용자 입력값 수집

연령: 20~34세 범위 구간화

연소득: 0원 ~ 4,000만 원 이상 다섯 구간

주거 형태: 부모 동거, 월세, 전세, 자가

직업 상태: 무직, 사회초년생, 정규직, 프리랜서

금융 목표: 단기 유동성, 목돈 마련·저축, 청약·주거마련, 신용관리, 생활비 절약·혜택 극대화

##### [2단계] 상품 필터링

입력값에 따라 조건에 맞지 않는 상품을 제거하여 추천 후보를 줄여나갑니다.

예:

- 무직자는 실적 조건 있는 신용카드 전면 제외
- 청년우대형 통장은 만 34세 이하, 무주택자만 허용
- 연소득이 낮을수록 연회비 및 실적 부담이 낮은 카드만 노출

### [3단계] 상품 점수화 및 우선순위 정렬

필터링된 상품군에 AI 챗봇의 추천 요소를 합산하여 상품을 점수화하고 우선순위를 정렬합니다.

예: 연회비, 금리, 우대 조건 수, 출금 조건, 비대면 가입 가능 여부 등

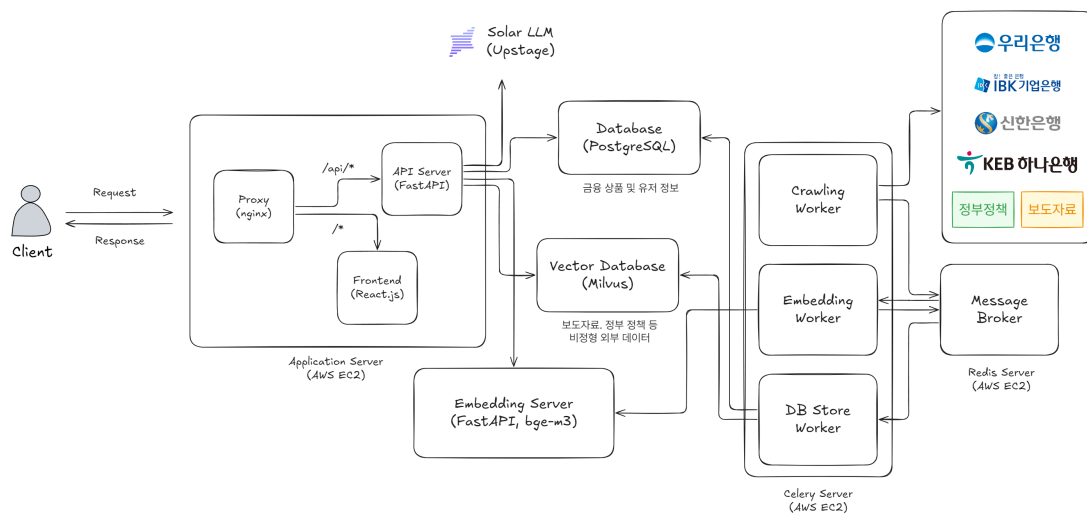
마지막으로, 최종 상위 1~2개 상품을 사용자에게 제안합니다.

## 다. 사회적 가치 도입 계획 등

본 프로젝트는 단순한 정보 제공을 넘어, 청년층의 금융 불균형 해소와 금융정보 접근성 확대를 목표로 합니다. 금융 지식이 부족한 청년층에게 눈높이에 맞는 상품 큐레이션을 제공함으로써 금융상품 선택 과정에서의 스트레스와 실수를 줄입니다. 국가 정책 상품(청년 우대형 통장, 청약, 청년대출 등)을 적절히 매칭하여 정책의 실효성을 높이고 접근성을 향상시키는 데 기여할 수 있습니다. 따라서 챗봇이 축적한 사용자의 선택 데이터를 기반으로 은행·정책금융기관과 협업하여 더 나은 상품 개선 방향도 제안할 수 있습니다. 궁극적으로 본 프로젝트를 통해 청년층의 금융 고민을 해소하고 효율적인 재무 관리를 도모할 수 있도록 고도화할 예정입니다.

## 2. 상세설계

### 가. System Architecture



### 1. Application Server

- \* API Server: Agentic RAG 파이프라인 및 API 엔드포인트
- \* Frontend: React.js 기반의 사용자 인터페이스
- \* Proxy: 요청 라우팅을 위한 프록시 서버

### 2. Database

- \* PostgreSQL(RDB): 은행별 금융 상품 및 유저 정보 저장
- \* Milvus(VectorDB): 보도자료, 정부정책 등 비정형 데이터 저장

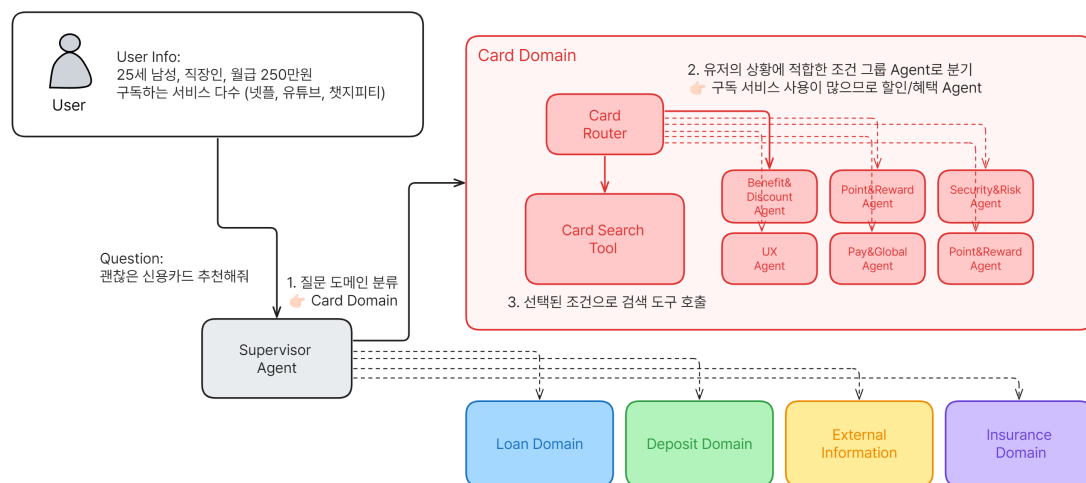
### 3. Embedding Server

- \* FastAPI 및 bge-m3 기반의 텍스트 임베딩 서버
- \* 홈서버를 통해 직접 모델 서빙

### 4. Celery Server

- \* Crawling Worker: 외부 데이터 소스에서 데이터 수집 및 전처리
- \* Embedding Worker: 수집한 데이터 임베딩 요청
- \* DB Store Worker: 최종 데이터 DB에 커밋

## 나. Agentic Flow



- 1) Supervisor Agent: Agent: Domain 분기 처리 및 Planning 담당
- 2) Domain Router: 유저 데이터 및 질문 쿼리 기반으로 Sub Agent 호출
- 3) Domain Sub Agents: 상품 검색을 위한 필터 및 점수화 기준 선택
- 4) Domain Search Tool: 선택한 검색 기준을 적용하여 RDB에서 상품 검색

## 다. 기술 스택

1. DB: PostgreSQL(RDB), Milvus(VectorDB)
2. API Server: Python 3.12 (FastAPI, SQLAlchemy, Alembic)
3. Crawler: Python 3.12 (Celery, Selenium, BeautifulSoup)
4. Text Embedding: BAAI/bge-m3 기반 파생 모델
5. Text Embedding Server: Python 3.10 (FastAPI, llama-cpp-python)
6. Chat Completion: gpt-4.1-mini, solar-pro
7. Frontend & Web: React.js(TypeScript), Tailwind CSS, nginx
8. Deploy: AWS EC2 + ALB(API, Web), 홈서버(DB, Crawler, Text Embedding)

### 3. 팀 소개 (소속, 구성원별 역할)

- PM: 경영학과 이채은
- FE: 정보컴퓨터공학부 이지수
- BE: 기계공학부 강민석
- UI/UX: 디자인학과 김예원