

최종 보고서



과제명	개인 지출 내역 추적 및 분석기
과목 / 교수	모바일컴퓨팅및응용 / 이현병 교수님
제출일자	2025.12.09
학번 / 이름	203622 / 윤태현(컴퓨터공학)

목차

1. 프로젝트 개요
1.1 프로젝트 명
1.2 개발 배경 및 목적
2. 시스템 아키텍처
2.1 시스템 구조도
2.2 기술 스택
3. 주요 기능 구현
3.1 지출 내역 관리
3.2 캘린더 기반 시각화
3.3 AI 소비 분석 리포트
4. 문제 해결
4.1 Flask 한글 인코딩 문제 해결
4.2 상태 관리와 UI 동기화
4.3 날짜 포매팅 및 비동기 처리
5. 결과물
5.1 메인 화면
5.2 지출 추가 화면
5.3 AI 분석 결과 화면
5.4 상세 화면
6. 결론 및 기대 효과
6.1 프로젝트 성과
6.2 기대 효과
6.3 향후 발전 방향

1. 프로젝트 개요

1.1 프로젝트 명

- 개인 지출 내역 추적 및 분석기

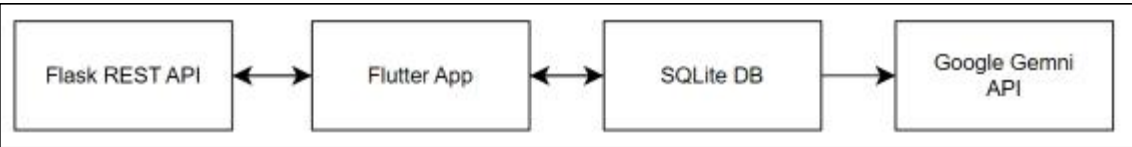
1.2 개별 배경 및 목적

- 배경 : 현대인의 소비 패턴이 다양화됨에 따라 단순한 기록이 아닌 지출 습관 분석의 필요성이 필요하다고 생각되었음.
- 목적 : Fluuter와 Flask 서버를 이용한 앱을 설계 및 구축하여 개발 역량 강화, 완성된 결과물을 통해 앱 사용자가 지출 패턴을 파악하여 소비 습관 개선.

2. 시스템 아키텍처

2.1 시스템 구조도

- Frontend : 사용자 인터페이스(UI) 담당, 사용자 입력을 받아 서버로 요청을 보냄.
- Backend : 비즈니스 로직 처리, 데이터베이스와 AI 서비스 중계.
- AI Service : 지출 데이터를 JSON 형태로 받아 분석 리포트 생성.



2.2 기술 스택

구분	기술 스택	설명
Language	Dart, Python	• 모바일 앱과 서버 개발에 최적화된 언어 설정
Frontend	Flutter	• UI 구성 및 사용자 상호작용 처리 • Provider 패턴을 이용해 상태 관리
Backend	Flask	• RESTful API 서버 구축 • 데이터 처리 및 API 중계
Database	SQLite	• 지출 내역 역구 저장 및 관리
AI Service	Google Gemini 2.5 Flash	• 지출 내역 기반 텍스트 요약 및 조언 생성

3. 주요 기능 구현

3.1 지출 내역 관리

- 기능 : 날짜, 금액, 카테고리, 메모를 포함한 지출 내역을 생성, 조회, 수정, 삭제.
- 구현 : Flask의 routes.py에서 HTTP Method별 엔드포인트를 구현하여 처리.

3.2 캘린더 기반 시각화

- 설계 변경 : 초기 설계의 단순 리스트 형식에서 직관적인 캘린더 뷰 방식으로 변경.
- 구현 : 달력의 날짜 셀 하단에 해당 일자 총 지출액을 표시하여 한눈에 파악, 특정 날짜 선택 시, 하단에 상세 리스트가 비동기로 갱신.

3.3 AI 소비 분석 리포트

- 기능 : 사용자의 지출 내역을 분석하여 총 지출, 주요 카테고리, 소비 습관 조언 제공.
- 구현 : Backend에서 DB 데이터를 JSON 리스트로 변환하여 프롬프트와 함께 전송, 프롬프트 엔지니어링으로 금융 비서 페르소나를 부여하여 자연스러운 답변 유도.

4. 문제 해결

4.1 Flask의 한글 인코딩 문제 해결

- 문제 : Flask jsonify 사용 시 한글 데이터가 ASCII 코드로 변환되어 클라이언트에서 깨짐.
- 해결 : __init__.py의 앱 설정에 app.json.ensure_ascii=false 옵션을 추가하여 인코딩 유지


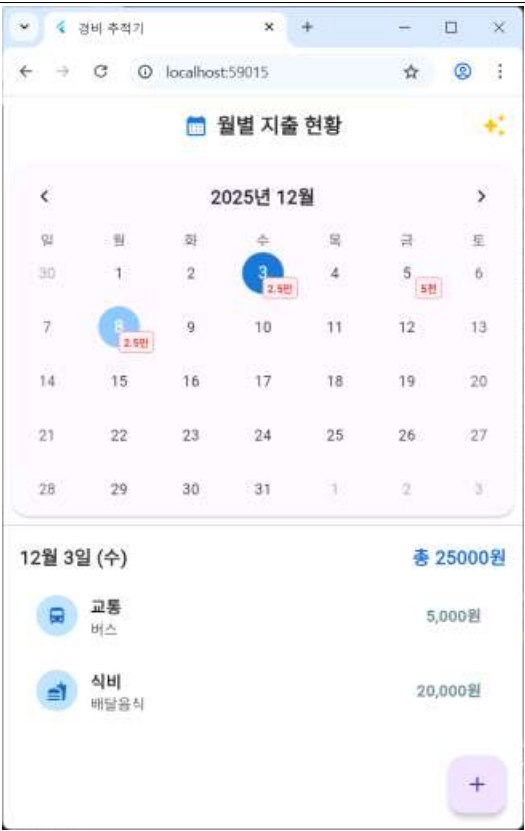
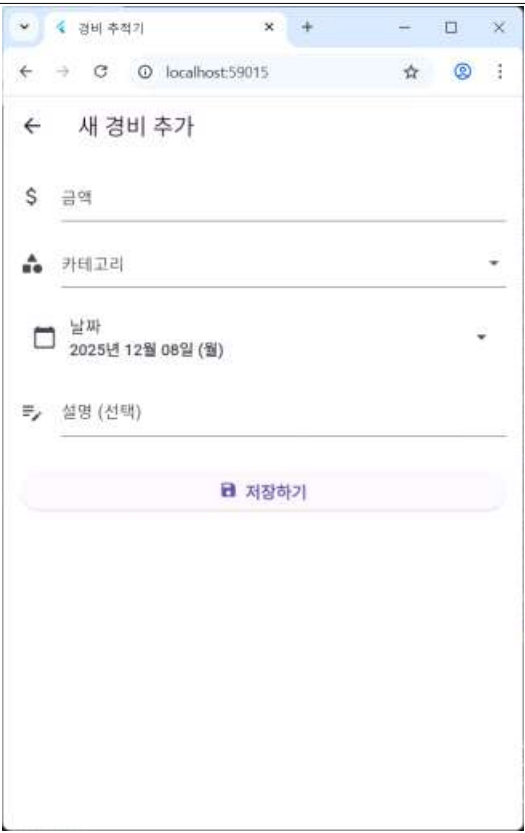
4.2 상태 관리와 UI 동기화

- 문제 : 지출 내역을 추가하거나 삭제했을 때, 화면이 즉시 갱신되지 않아 사용자가 앱을 재시작해야 하는 문제 발생
- 해결 : Flutter의 Provider패턴을 도입하여 데이터 변경이 발생할 때마다 notifyListeners()를 호출하여, 이를 구독하고 있는 UI 위젯들이 자동으로 다시 그려지도록 구현

4.3 날짜 포매팅 및 비동기 처리

- 문제 : Python과 Dart 간의 날짜 객체 형식이 다르고 문자열 파싱 처리가 복잡함.
- 해결 : Backend에서 dateutil.parser를 사용하여 파싱 및 isoformat()으로 표준화하여 응답 Frontend에서 intl패키지를 사용하여 한국어 날짜 형식으로 변환

5. 결과물 예시

<div>5.1 메인 화면</div> <div><div>○ 날짜별 지출 마커</div><div>○ 지출 내역 추가, 수정, 삭제 버튼</div><div>○ 상세 지출 리스트 표시</div><div>○ AI 분석 버튼 </div></div>	 <p>The screenshot shows a web application titled '경비 추적기' (Expense Tracker) running on localhost:59015. The main screen displays a calendar for December 2025. Dates 3, 8, and 5 are marked with blue circles and red labels indicating expenses of 2,500, 2,500, and 5,000 respectively. Below the calendar, a summary for '12월 3일 (수)' shows a total of 25,000원, broken down into '교통 버스' (5,000원) and '식비 배달음식' (20,000원). A purple '+' button is at the bottom right.</p>
<div>5.2 지출 추가 화면</div> <div><div>○ 데이터 입력 : 금액, 카테고리, 날짜, 상세내용</div><div>○ 저장하기 버튼</div></div>	 <p>The screenshot shows the '새 경비 추가' (Add New Expense) form. It includes input fields for '금액' (Amount), a dropdown for '카테고리' (Category), a date picker for '날짜' (Date) set to '2025년 12월 08일 (월)', and a text area for '설명 (선택)' (Description). A purple '저장하기' (Save) button is at the bottom.</p>

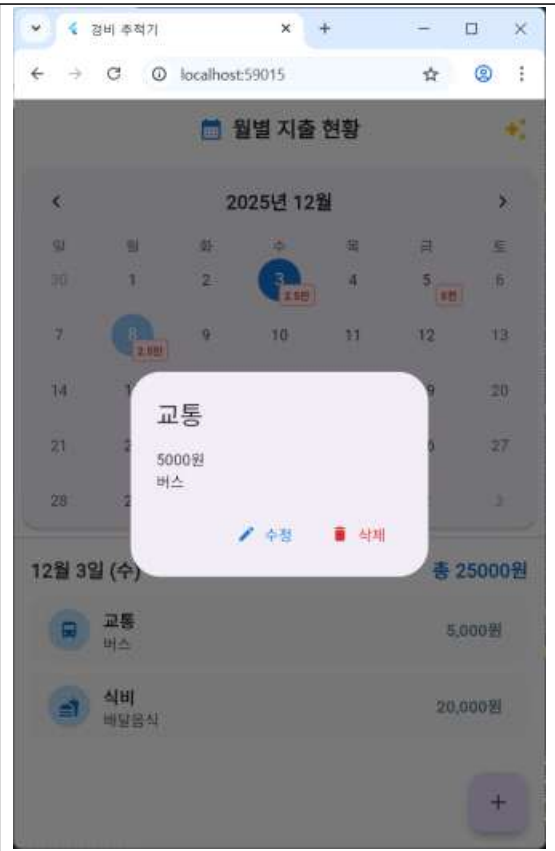
5.3 AI 분석 결과 화면

- 분석 시작 버튼
- 일주일 간 지출액 표시
- 주요 지출 카테고리 TOP3 표시
- 지출 내역 분석 및 조언



5.4 상세 화면

- 지출 내역 클릭 시 수정, 삭제 버튼 활성화



6. 기대 효과

6.1 프로젝트 성과

- 초기 설계했던 CRUD 및 AI 분석 기능을 모두 정상적으로 구현하였으며, 기존 리스트 형식이 아닌 캘린더 뷰 도입으로 사용성 개선.

6.2 기대 효과

- 사용자는 단순히 지출을 기록하는 것이 아닌, AI가 제공하는 피드백을 통해 능동적으로 소비 습관을 개선할 수 있음.
- 직관적인 캘린더 UI를 통해 월별/일별 지출 흐름을 쉽게 파악할 수 있음.

6.3 향후 발전 방향

- 데이터 시각화 : 파이 차트나 그래프를 추가하여 시각적으로 보여주는 기능 추가
- 대화형 AI : 현재의 일방향적인 '요약 리포트'방식을 넘어, 사용자가 “지난달보다 식비가 얼마나 늘었어?”같은 질문을 하면 AI가 대답해주는 형식
- 영수증 기능 : 일일이 금액과 내용을 타이핑하는 번거로움을 줄이기 위해, 영수증 사진을 찍으면 자동으로 날짜, 금액, 상호명 등을 추출하여 입력폼에 채워주는 기능 추가