

가계부 및 소비 패턴 분석 앱

이조가짱이조

글로벌미디어학부 20251661 최지훈
전자정보공학부 20251526 박지훈
전자정보공학부 20252668 김시오

주제 선정 이유

적절한 난이도 + 일상생활 + 학생사용 = 돈관리앱?

프로젝트 소개(주요 기능)

예산 설정: 이번달 예산 항목 입력

지출 등록: 카테고리별 지출 내역 입력 및 저장

캘린더 기능: 날짜별 소비 내역 조회

예산 분석: 현재까지의 지출을 반영한 남은 예산 및 일일 사용 가능 금액 계산

소비 패턴 비교: 지난달과 이번 달 소비 내역 비교 그래프 제공

자료구조 설계

budget data
{ "종류" : "금액" }

예) { "식비" : "120000", "옷" : "200000" ... }

dated_spending
{ "날짜" : { "종류" : "금액", "종류" : "금액" } }

예) { "2025-05-02": { "식비": 7094, "기타": 4972 }, .. }

앱을 다시 켜도 데이터 수정값이 저장되도록
txt파일에 구조 저장

≡ budget_data.txt

≡ dated_spending.txt

알고리즘 설계 - main_frame

[앱 실행 시작]



[app.py 초기화]



[폰트 로딩 / Tkinter 창 설정]



[Frame 구성 (main_frame 포함)]



[build_main_screen 호출]



[메인 화면 버튼 선택]



[데이터 저장/불러오기]

< _____

저장은 global_state로 통합



함수 설계 - 매개변수 설명

매개변수 : root,

show_frame_callback,

main_frame,

font_regular,

font_bold

리턴값 : frame

tk 기본 프레임

화면 전환을 위한 함수

뒤로가기를 위해 메인화면 프레임

기본 폰트 스타일 (SUITE)

제목등에 사용될 폰트 스타일 (SUITE_BOLD)

함수 설계 - budget_frame

달 초에 입력하는 예산

예산 입력

수입:
(현재: 3,000,000)

식비:
(현재: 500,000 / 사용: 207,900)

옷:
(현재: 200,000 / 사용: 51,500)

고정:
(현재: 1,000,000 / 사용: 200,000)

교통:
(현재: 100,000 / 사용: 15,600)

기타:
(현재: 200,000 / 사용: 14,900)

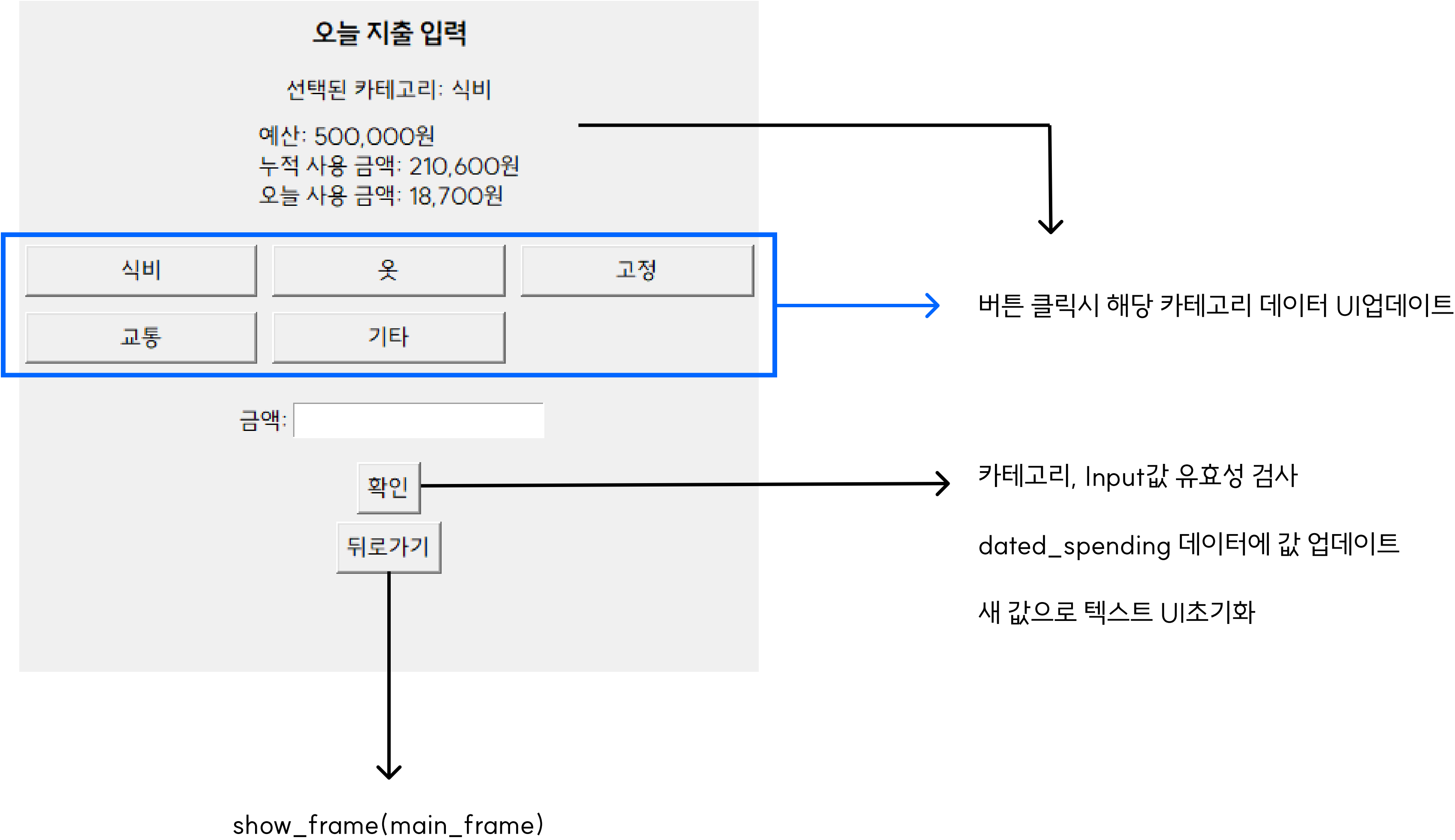
-Input 값들의 유효성 검사

-budget_data(예산 데이터 딕셔너리)에 저장

-라벨 UI 즉시 업데이트
(for문으로 모든객체에 접근)

show_frame(main_frame)

함수 설계 - spending_frame



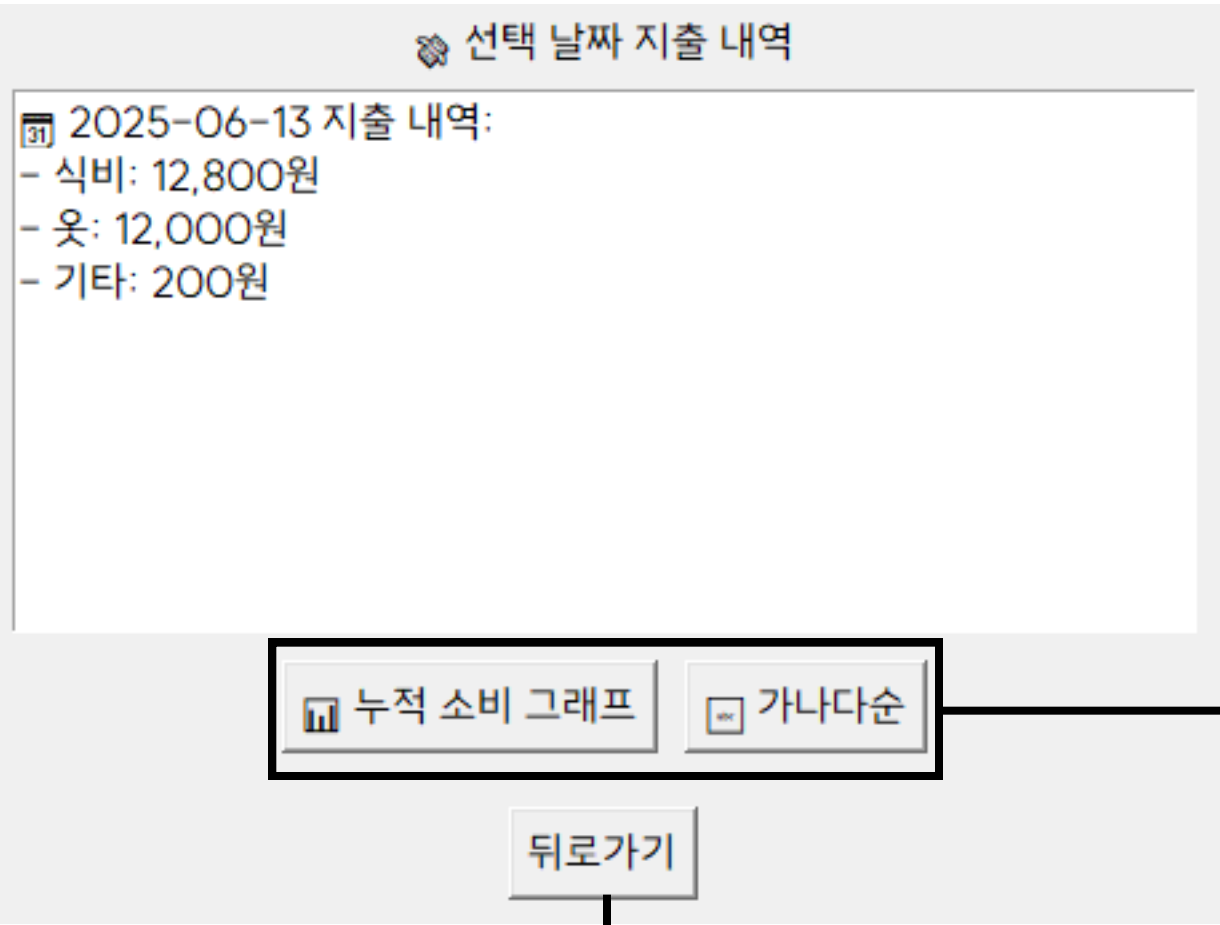
함수 설계 - calendar_frame



tkcalendar 기능으로 캘린더 생성

오늘날짜 강조표시
(background="#ffeb99")

날짜 클릭시 이벤트 발생
(오늘이라면? 분기점도 생성)

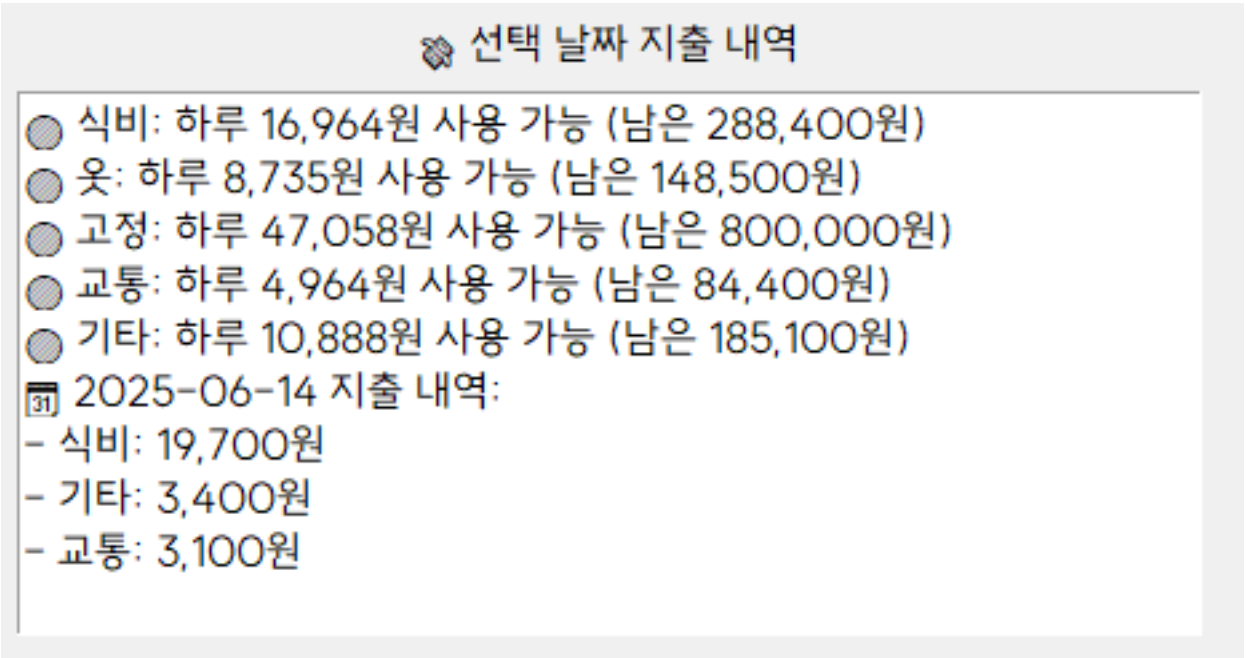


dated_spending
딕셔너리에서 데이터를 읽어

선택된 날짜에 해당하는
지출내역을 표기

show_spending_graph
(sort_by="amount")

show_frame(main_frame)



오늘이라면?

예산에서 이번 달 누적 지출 금액을 차
감하여 남은 예산을 계산

현재 날짜 기준으로 이번 달에 남은 일
수를 계산

남은 예산을 남은 일수로 나눠 일일 사
용 가능 금액 도출

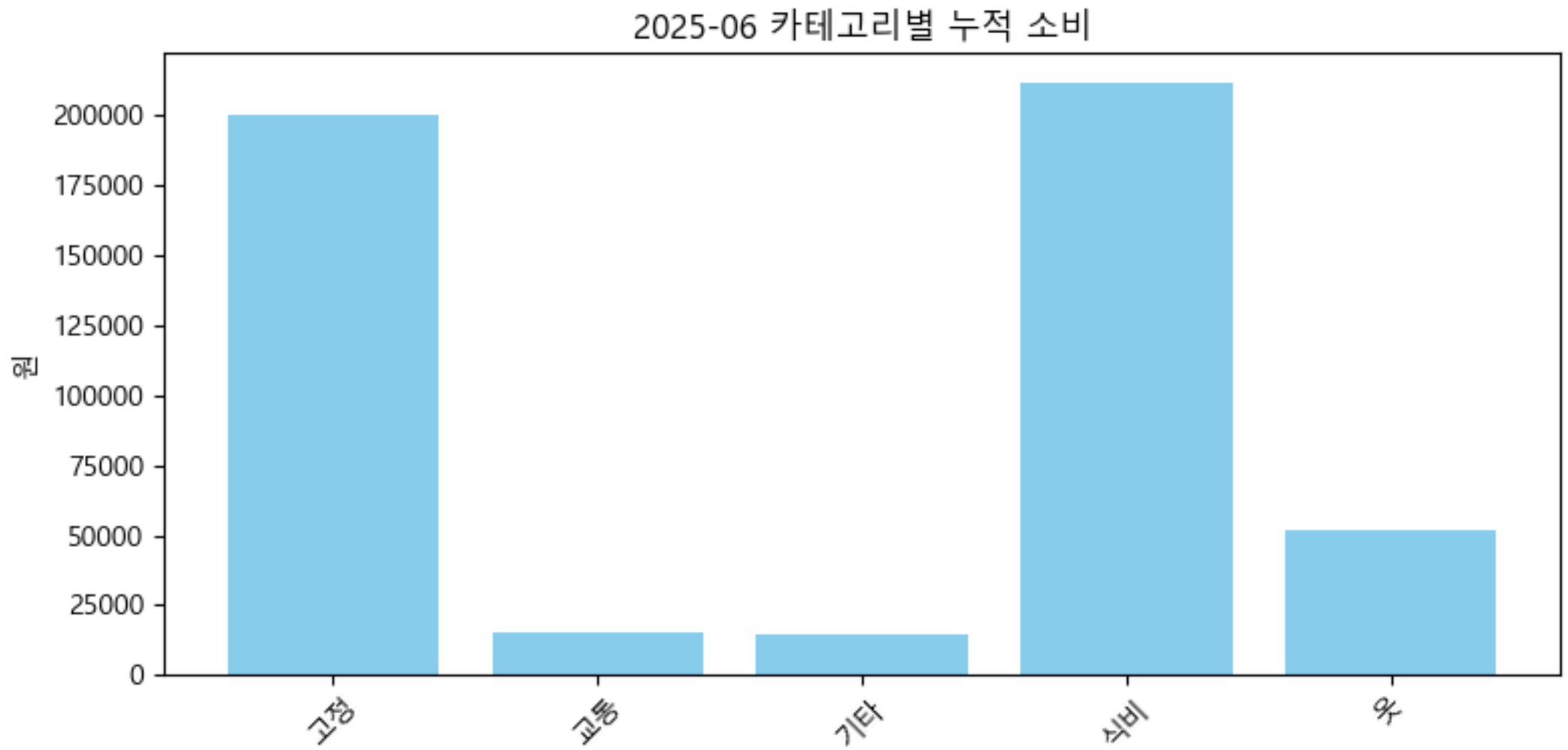
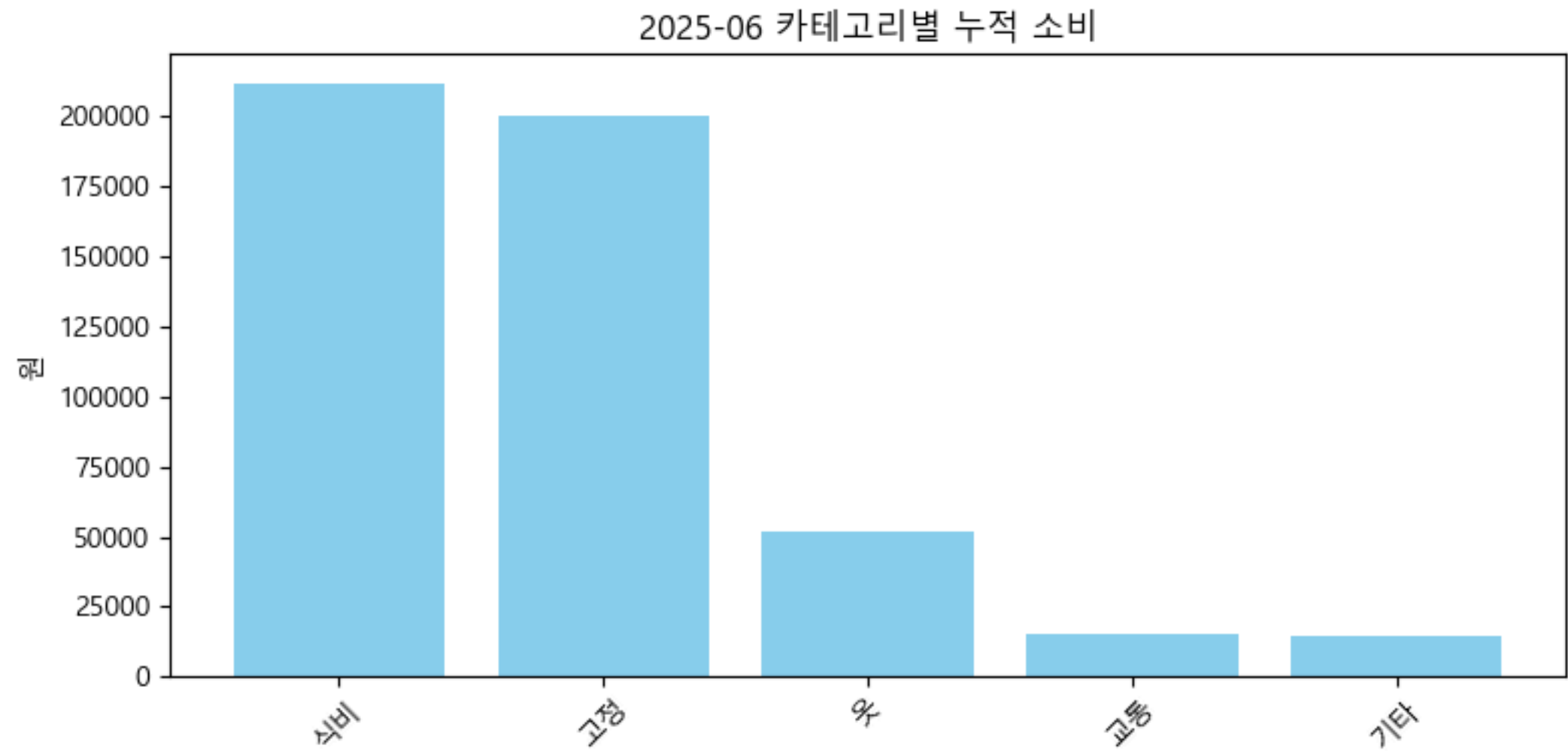
남은 예산 = 전체 예산 - 이번 달 누적
지출
하루 사용 가능 금액 = 남은 예산 ÷ 남
은 일수

remaining_days = last_day -
today_dt.day + 1

remaining = max(0, budget -
monthly_spent)

per_day = remaining //
remaining_days

함수 설계 - show_spending_graph(sort by = " ")



matplotlib 라이브러리를 사용하여 막대 그래프(bar chart) 시각화

현재 선택된 날짜를 기준으로 해당 월(yyyy-mm)의 데이터를 필터링

카테고리별로 누적 지출 금액을 합산하여 시각적으로 비교

정렬 기준(금액순 / 가나다순)에 따라 그래프 출력 방식 변경 가능

함수 설계 - build_analysis_screen

14일 기준 소비 비교 분석

● 이번 달(2025-06) vs ● 지난 달(2025-05)

14일 기준 누적 소비 비교

전체 소비:
- ● 이번 달: 420,900원
- ● 지난 달: 554,464원

카테고리별 소비:
- 고정: ● 150,000원 vs ● 300,000원
- 교통: ● 15,600원 vs ● 24,292원
- 기타: ● 14,900원 vs ● 20,149원
- 식비: ● 197,900원 vs ● 162,376원
- 옷: ● 42,500원 vs ● 47,647원

그래프로 비교

뒤로가기

통계 데이터를 필터링하여 수치 비교

14일 기준 카테고리별 소비 비교

카테고리	2025-05 (₩)	2025-06 (₩)
고정	300,000	150,000
교통	24,292	15,600
기타	20,149	14,900
식비	162,376	197,900
옷	47,647	42,500

시연영상

팀 구성 및 역할

최지훈 : 기획, 메인함수, 캘린더 창 함수, 분석 창 함수, 함수 병합,
피피티제작, GUI전환, 발표, 실행영상편집

김시오 : 예산입력 창 함수, 기획

박지훈 : 디자인, 기획, 지출 창 함수, 발표자료 제작

에필로그

텍스트나 콘솔 출력에만 머무르지 않고,
직접 UI로 구현해서 눈에 보이는 결과물이 나온다는 점이 가장 흥미로웠다.

직접 화면을 설계하면서
프로그래밍이 '사용자 입장에서 생각하는 일'이라는 것을 체감하였다.

Q&A