

CMM Project

소셜 가계부



2017 July 12

저자: 최진헌

CMM Project

소셜 가계부

목차

1 프로젝트 구성	11
1.1 프로젝트 개요	11
1.2 프로젝트 목적	12
1.3 프로젝트 정의	12
1.3.1 프로젝트 이름	12
1.3.2 프로젝트 소개	12
1.3.3 프로젝트 기능	12
1.4 프로젝트 일정	13
1.5 개발환경 및 도구	13
1.5.1 개발서버	13
1.5.2 개발클라이언트	13
1.5.3 데이터베이스	13
1.5.4 버전관리	13
1.5.5 개발툴	13
1.6 기대효과	14
2 프로젝트 개발	15
2.1 시장분석	15
2.1.1 네이버가계부	15
2.1.2 모네타 미니가계부	16
2.2 DJANGO 란?	17
2.3 BOOTSTRAP 이란?	18
3 화면목록 및 흐름도	19
3.1 메인	19
3.1.1 화면 목록	19
3.1.2 화면 흐름도	19
3.2 회원관리	20
3.2.1 화면 목록	20
3.2.2 화면 흐름도	20

3.3 통장관리	21
3.3.1 화면 목록	21
3.3.2 화면 흐름도	21
3.4 통장 데이터관리	22
3.4.1 화면 목록	22
3.4.2 화면 흐름도	22
3.5 자유게시판	23
3.5.1 화면 목록	23
3.5.2 화면 흐름도	23
3.6 자유게시판 댓글	24
3.6.1 화면 목록	24
3.6.2 화면 흐름도	24
3.7 상담게시판	25
3.7.1 화면 목록	25
3.7.2 화면 흐름도	25
3.8 상담게시판 댓글	26
3.8.1 화면 목록	26
3.8.2 화면 흐름도	26
4 DB 설계서	27
4.1 회원관리	27
4.2 계좌관리	27
4.3 데이터관리	28
4.4 자유게시판	28
4.5 자유게시판 댓글	29
4.6 상담게시판	29
4.7 상담게시판 댓글	29
5 URL 설계	30
5.1 회원관리	30
5.2 계좌관리	30
5.3 데이터관리	30
5.4 자유게시판	30
5.5 자유게시판 댓글	31
5.6 상담게시판	31

5.7 상담게시판 댓글	31
6 기능정의	32
6.1 회원관리	32
6.2 계좌관리	33
6.3 데이터관리	34
6.4 자유게시판	35
6.5 자유게시판 댓글	35
6.6 상담게시판	36
7 결과화면	37
7.1 로그인	37
7.1.1 로그인화면	37
7.1.2 로그인	37
7.1.3 회원가입	38
7.1.4 이메일찾기	39
7.1.5 비밀번호찾기	40
7.2 메인	41
7.2.1 메뉴	41
7.4 마이페이지	42
7.4.1 비밀번호변경	42
7.4.2 회원탈퇴	43
7.4.3 계좌추가	43
7.4.4 계좌정보확인	44
7.4.5 계좌정보수정	44
7.4.6 계좌정보삭제	45
7.3 데이터관리	45
7.3.1 데이터추가	47
7.3.2 데이터확인	48
7.3.3 데이터수정	48
7.3.4 데이터삭제	49
7.4 자유게시판	49
7.4.1 자유게시판 글쓰기	50
7.4.2 자유게시판 글보기	50
7.4.3 자유게시판 글수정	51

7.4.4 자유게시판 글삭제	51
7.4.5 자유게시판 댓글보기	52
7.4.6 자유게시판 댓글쓰기	52
7.4.7 자유게시판 댓글수정	53
7.4.8 자유게시판 댓글삭제	53
7.5 상담게시판	54
7.5.1 상담게시판 글쓰기	54
7.5.2 상담게시판 글보기	55
7.5.3 상담게시판 글수정	55
7.5.4 상담게시판 글삭제	56
7.5.5 상담게시판 댓글보기	56
7.5.6 자유게시판 댓글쓰기	57
7.5.7 상담게시판 댓글수정	57
7.5.8 상담게시판 댓글삭제	58

표 목차

[표 1] 대표적인 가게부 서비스	12
[표 2] 메인 화면목록.....	19
[표 3] 회원관리 화면 목록	20
[표 4] 통장관리 화면목록	21
[표 5] 통장 데이터관리 화면 목록	22
[표 6] 자유 게시판 화면 목록	23
[표 7] 자유게시판 댓글 화면 목록	24
[표 8]상담게시판 화면 목록	25
[표 9] 상담게시판 댓글 화면 목록	26
[표 10] 회원관리 DB 설계표	27
[표 11] 계좌관리 DB 설계표	27
[표 12] 데이터관리 DB 설계표.....	28
[표 13] 자유게시판 DB 설계표.....	28
[표 14] 자유게시판 댓글 DB 설계표	29
[표 15] 상담게시판 DB 설계표.....	29
[표 16] 상담게시판 댓글 DB 설계표	29
[표 17] 회원관리 URL 설계표	30
[표 18] 계좌관리 URL 설계표	30
[표 19] 데이터관리 URL 설계표.....	30
[표 20] 자유게시판 URL 설계표.....	30
[표 21] 자유게시판 댓글 URL 설계표	31

[표 22] 상담게시판 URL 설계표.....	31
[표 23] 상담게시판 댓글 URL 설계표	31
[표 24] 회원관리 기능정의표	32
[표 25] 계좌관리 기능정의표	33
[표 26] 데이터관리 기능정의표.....	34
[표 27] 자유게시판 기능정의표.....	35
[표 28] 자유게시판 댓글 기능정의표	35
[표 29] 상담게시판 기능정의표.....	36
[표 30] 상담게시판 댓글 기능정의표	36

그림목차

[그림 1] 가게부의 필요성	11
[그림 2] 프로젝트 일정	13
[그림 3] 점점 늘어나는 가게부 서비스	14
[그림 4] 네이버 가게부	15
[그림 5] 모네타 미니가게부	16
[그림 6] 일반적인 웹서버의 역할	17
[그림 7] 장고 웹서버의 역할	18
[그림 8] 메인 화면 흐름도	19
[그림 9] 회원관리 화면 흐름도	20
[그림 10] 통장관리 화면 흐름도	21
[그림 11] 통장 데이터 추가 화면 흐름도	22
[그림 12] 자유게시판 화면 흐름도	23
[그림 13] 자유게시판 댓글 화면 흐름도	24
[그림 14] 상담게시판 화면 흐름도	25
[그림 15] 상담게시판 화면 흐름도	26
[그림 16] 로그인 화면	37
[그림 17] 다른 정보 입력	37
[그림 18] 로그인 완료	37
[그림 19] 회원가입 화면	38
[그림 20] 회원가입 빈칸입력	38
[그림 21] 이메일 찾기 화면	39

[그림 22] 다른 정보를 입력.....	39
[그림 23] 이메일 찾기 완료.....	39
[그림 24] 비밀번호 찾기 화면	40
[그림 25] 다른 정보 입력	40
[그림 26] 패스워드 변경 폼.....	40
[그림 27] 메인화면	41
[그림 28] 메뉴화면	41
[그림 29] 마이페이지 화면	42
[그림 30] 비밀번호변경 화면	42
[그림 31] 비밀번호변경 유효성 검사	42
[그림 32] 회원탈퇴 화면.....	43
[그림 33] 탈퇴완료	43
[그림 34] 계좌추가 화면.....	43
[그림 35] 계좌추가 완료.....	43
[그림 36] 계좌정보확인 화면	44
[그림 37] 계좌정보수정 화면	44
[그림 38] 계좌정보수정 완료	44
[그림 39] 계좌정보삭제 화면	45
[그림 40] 계좌정보삭제 완료	45
[그림 41] 데이터관리 버튼	45
[그림 42] 데이터관리 화면	45
[그림 43] 수입/지출별 그래프	46
[그림 44] 그래프의 검색범위 변경	47

[그림 45] 그래프의 파일저장	47
[그림 46] 데이터 추가 선택화면.....	47
[그림 47] 데이터 추가화면	47
[그림 48] 데이터 선택 화면.....	48
[그림 49] 데이터확인 화면	48
[그림 50] 데이터수정 선택 화면.....	48
[그림 51] 데이터수정 화면	48
[그림 52] 데이터삭제 선택 화면.....	49
[그림 53] 데이터삭제 확인	49
[그림 54] 자유게시판 선택	49
[그림 55] 자유게시판 화면	49
[그림 56] 자유게시판 글쓰기 선택 화면.....	50
[그림 57] 자유게시판 글쓰기 화면	50
[그림 58] 본인글 확인	50
[그림 59] 다른 사용자의 글 확인	50
[그림 60] 본인글 확인	51
[그림 61] 다른 사용자의 글 확인	51
[그림 62] 본인글 삭제 화면.....	51
[그림 63] 삭제 확인	51
[그림 64]댓글보기 화면	52
[그림 65] 댓글이 없는 화면.....	52
[그림 66] 댓글쓰기 버튼 화면	52
[그림 67] 댓글쓰기 화면.....	52

[그림 68] 댓글수정 화면.....	53
[그림 69] 댓글수정 확인.....	53
[그림 70] 댓글삭제 버튼 확인	53
[그림 71] 댓글삭제 확인.....	53
[그림 72] 상담게시판 선택	54
[그림 73] 상담게시판 화면	54
[그림 74] 상담게시판 글쓰기 선택 화면.....	54
[그림 75] 상담게시판 글쓰기 화면	54
[그림 76] 본인글 확인	55
[그림 77] 다른 사용자의 글 확인.....	55
[그림 78] 본인글 수정버튼 확인.....	55
[그림 79] 상담게시판 글수정	55
[그림 80] 본인글 삭제버튼 확인.....	56
[그림 81] 삭제 확인	56
[그림 82] 다른 사람의 댓글 화면	56
[그림 83] 자신의 댓글 화면.....	56
[그림 84] 댓글쓰기 버튼 화면	57
[그림 85] 댓글쓰기 화면.....	57
[그림 86] 댓글수정 화면.....	57
[그림 87] 댓글수정 확인.....	57
[그림 88] 댓글삭제 버튼 확인	58
[그림 89] 댓글삭제 확인.....	58

1 프로젝트 구성

1.1 프로젝트 개요

가계부: 가게의 수입과 가게 지출을 중심으로 재산의 증가와 감소를 일정한 형식에 맞추어 회기동안 기록하는 장부. [링크](#)

[실전! 금융세상 체험기] 똑똑한 소비 첫걸음 '가계부 쓰기'...반
년만에 지출 30% 줄어

스마트폰 앱 활용이 대세

신용카드 사용내역 자동 입력 계산도 알아서 척척 해줘

어디에 얼마만큼 썼는지 그래프로 일·월 연도별 소비내역 분석도

[그림 1] 가계부의 필요성

해당기사에 따르면 어렸을 때 쓰던 용돈 기입장, 사회 생활을 하며 사용하던 종이 가계부가 스마트폰 앱이나 용돈 기입장으로 진화하고 있는 모습을 볼 수 있다. 기존의 종이 가계부는 지출과 소비를 할 때마다 수첩을 들고 다니거나 집에서 영수증을 가지고 확인하는 번거로움이 있었다면 현재 모바일 가계부의 장점은 언제 어디서든 휴대전화만 있으면 지출과 소비를 기록할 수 있고 자동으로 자신이 얼마나 어느정도 소비하는지 한눈에 볼 수 있는 그래프까지 지원하고 있으니 훨씬 간편하고 자신의 소비성향을 파악하여 지나친 지출을 줄일 수 있다는 장점을 가지고 있어 이용자들이 점점 늘어나고 있는 추세이다.

1.2 프로젝트 목적

	
네이버 가계부	모네타

[표 1] 대표적인 가계부 서비스

현재 포털사이트에 ‘가계부’를 검색하면 수많은 가계부를 서비스하는 앱&웹 등이 검색이 되고 있으며 CMM은 대표적인 이 두개의 서비스를 분석하고 모티브로 삼아 개발하려고 한다.

디바이스에 얽매이지 않기 위하여 모바일웹 서비스를 선택하게 되었고, 2달이라는 짧은 개발 기간으로 인해, 시간대비 개발의 생산성이 높은 Python의 Django, Bootstrap을 선택하게 되었다.

1.3 프로젝트 정의

1.3.1 프로젝트 이름

CMM: Control My Money

1.3.2 프로젝트 소개

Control My Money는 현재 나와있는 가계부 포털과 서비스의 불편하고 어려운 조작법을 보완하여 가계부 뿐만 아니라 다른 사람들과 지식도 공유하고 함께 하는 소셜 가계부이다.

1.3.3 프로젝트 기능

계좌관리: 자신이 가지고 있는 통장들의 계좌번호, 통장이름, 한 줄 메모, 통장설명을 입력하여 사용자가 원하는 대로 추가, 수정, 삭제하여 관리할 수 있다.

계좌데이터 추가: 사용자의 수입과 지출을 날짜, 금액, 카드/현금, 방식에 따라 입력할 수 있으며, 데이터들은 통장별로 정리되어 전체, 수입별, 지출별 표와 그래프로 확인할 수 있다.

또한, ‘적금’ 파트는 자동으로 전체 사용자중 저축왕 랭킹을 도입하여 사용자의 절약 심리를 더욱 자극시킬 수 있다.

자유게시판: 회원들의 가입인사, 또는 친목 도모를 위해 마음껏 사용하는 게시판이며, 닉네임을 사용하기 때문에 누가 어떤 글을 작성하였는지 확인할 수 있다.

상담게시판: 자신의 익명성을 보장받을 수 있는 게시판이며, 자신의 수입, 지출 등이 공개되길 꺼려하는 사용자를 위한 상담게시판이다. 또한 자신이 작성한 글을 메인 페이지에 자신에게만 노출 되기 때문에 손쉽게 확인할 수 있다.

1.4 프로젝트 일정

	TaskName	기간	3월	4월	5월	6월
1	계획수립	4주				
2	자료조사	2주				
3	공부	16주				
4	설계	6주				
5	구현	10주				
6	테스트	2주				

[그림 2] 프로젝트 일정

1.5 개발환경 및 도구

1.5.1 개발서버

Django(1.11)sub & Python(3.6.0)

1.5.2 개발클라이언트

Google Chrome (59.0.3071.115)

1.5.3 데이터베이스

SQLite3

1.5.4 버전관리

Github

1.5.5 개발툴

Sublime Text3 build 3126, Pingendo(Bootstrap)

1.6 기대효과



[금융정보 쓱쓱 모아주는 똑똑한 가계부 앱](#) 머니투데이 | 📄 16면3단 | 2017.06.24. | 네이버뉴스 | [🔗](#)

보여주는 **가계부 앱**이다. 단 한 번만 공인인증서 또는 금융회사 계정 정보를 연동하면 자신의 금융정보를 실시간 확인 및 관리할 수 있다. 지난해 구글 플레이스토어가 선정한 '올해를 빛낸 혁신적인 앱'으로 선정된 바...



[\[글로벌 스타트업\] ④AI 기반 가계부 앱 선보인 獨 스타트업 '짜이트골드'](#)
IT조선 | 2017.06.17. | [🔗](#)

하지만 샤커씨는 **가계부 정리 앱** '짜이트골드(Zeitgold)'를 사용한 후 비용을 지불하기 쉬워졌다. 그는 "짜이트골드 앱은 언제 어디에 얼마를 납부해야 하는지 1센트도 틀리지 않고 제 때 알려주기 때문에 자금 운용하기가..."

[뉴스 더보기 >](#)

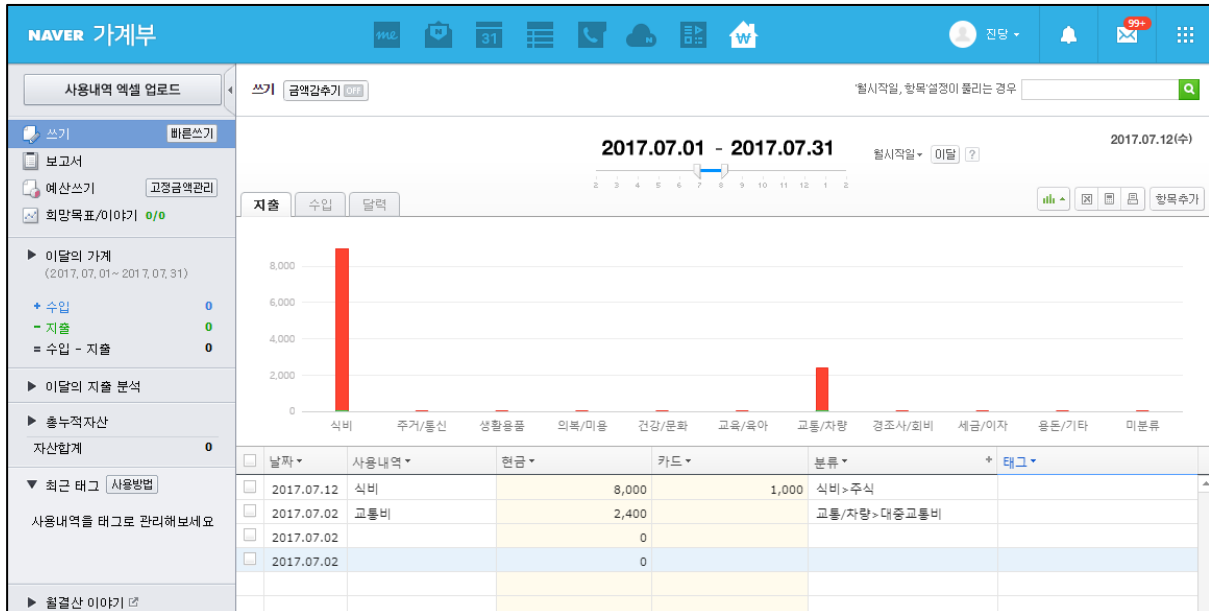
[그림 3] 점점 늘어나는 가계부 서비스

현재도 가계부 앱과 웹서비스는 점차 증가하는 추세이며 앞으로도 편리하고 금전 절약에 확실한 효과가 있는 스마트 가계부의 사용자 또한 증가할 것으로 보인다. 그에 발맞춰 CMM 프로젝트는 기존의 가계부 포털에 있는 소셜 기능과 간편하게 원클릭으로 입력할 수 있는 편리한 가계부 입력기능을 결합하여 보다 더 편한 가계부 사용을 사용자들이 이용할 수 있다고 본다.

2 프로젝트 개발

2.1 시장분석

2.1.1 네이버가계부



[그림 4] 네이버 가계부

초기 디지털 가계부인 엑셀형식을 기반으로 만들어진 것으로 보이며 웹페이지 화면에 입력, 출력하는 형식으로 입력하는 즉시 그래프로 업데이트되며 자신이 원하는 기간을 선택하여 수입과 지출을 확인할 수 있으며 또한 무료로 사용 가능하다.

하지만 본인 혼자만의 가계부 작성 기능 외에 다른 사람들과 금융이나 재테크에 관련된 정보나, 금융상담을 받을 수 있는 커뮤니티가 없어 아쉬운 점도 있었다.

2.1.2 모네타 미니가계부

The screenshot shows the MONETA Mini-gaebubu website interface. At the top, there's a navigation bar with links like '미가파티', '미니팅', '미가커뮤니티', '버룩시장', and '미가모바일'. Below this, there's a search bar and a list of posts. The main content area displays a table of posts with columns for '번호' (Number), '분류' (Category), '제목' (Title), '필명' (Pseudonym), '작성일' (Date), and '조회' (Views). The posts are sorted by '최신호회원글보기' (Newest member post). On the right side, there are several promotional banners and a sidebar with links to '미가 FAQ', '미가 사용법', '미가인 소식지', and '오른동장'.

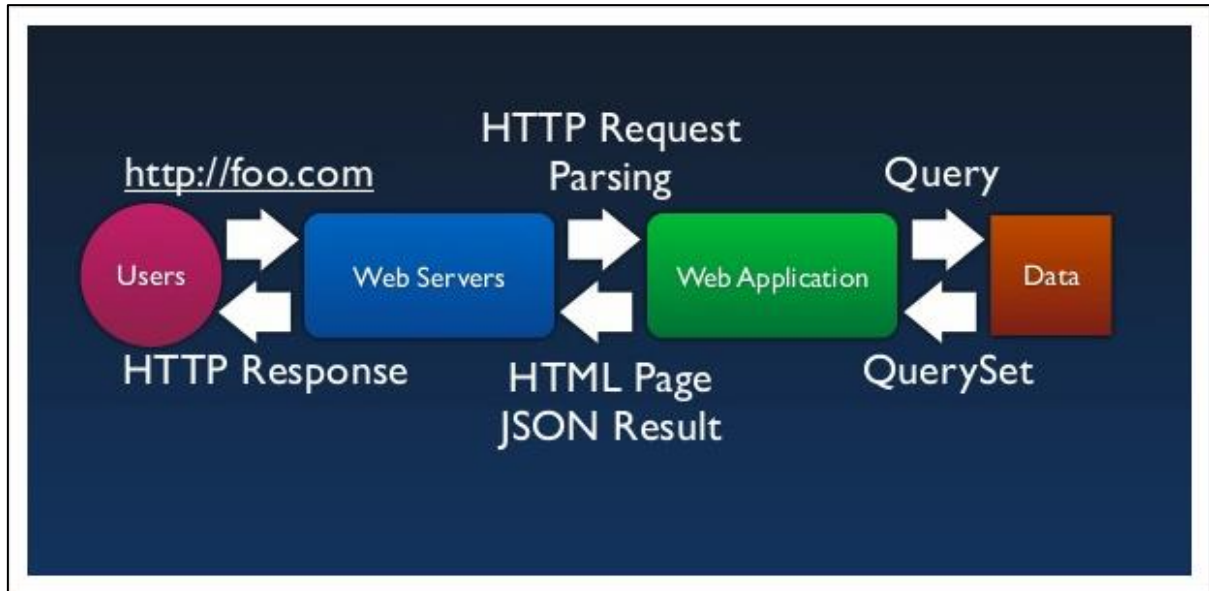
번호	분류	제목	필명	작성일	조회
	싱글	베스트 선정되시면, 필명댓글을 드립니다! [36]	가계부운영자	01/11	11161
36979	30대남	지갑속 현금은 가계부저리를 어떻게 하는건가...	kasagood	07/12	39
36976	분류를 선택하세요	저축만기되면 어떻게 표시해야되나요?(7)	소랑아	07/04	637
36975	가족동거녀	30대 싱글 여자 가계부(6월)[2]	z라이웃z	07/02	938
36974	가족동거녀	서른두번째 가계부 일니다 :[2]	워너비알뜰녀	06/30	577
36970	분류를 선택하세요	전월척자(이월) 처리 방법[2]	신나게야싸	06/25	604
36969	30대녀	36세 싱글 여자 가계부(5월)[4]	z라이웃z	06/24	1280
36968	가족동거녀	[66번째] 44살 2017. 6월 가계부 정산[6]	깨롱이	06/24	860
36967	30대녀	(31살) 새로운 하루하루! 즐겁게! 알차게 살기[7]	김지봉	06/19	1179
36965	30대녀	[30살] 긍정적인 평범녀의 5월가계부 ♥[2]	행위맘	06/13	1334
36960	분류를 선택하세요	가계부 작성관련 질문이 있습니다.[2]	AegisofAtena	06/07	473
36957	30대남	[38] 후원자가 늘었습니다 [15]	세미v	06/04	1507
36956	가족동거녀	올해는 윤 5월인데요?![1]	수줍음	06/02	815
36955	가족동거녀	서른+서른한번째 가계부입니다 :[2]	워너비알뜰녀	06/01	1178
36954	가족동거녀	[65th] 44살 2017. 5월 가계부 정산[12] 회원공개	깨롱이	05/28	1738
36952	30대남	햇색시랑제리 화보 공개..무결정 색시 매력 [1]	ozzy721	05/23	1064
36951	분류를 선택하세요	패이크 삭스 가져가세요^^	아가2	05/23	734
36949	가족동거녀	30대 싱글 여자 가계부[1]	z라이웃z	05/21	1707
36946	30대녀	(31살) 5월은 지출의 달 ^^:[5]	김지봉	05/17	1636
36939	40대남	미니 가계부 저장 문제[2]	au2115	05/10	965
36933	30대남	[38] 싱크대 정수기를 질렀습니다.[5]	세미v	05/07	1545
36931	30대녀	[30살] 긍정적인 평범녀의 4월가계부 ♥[8]	행위맘	05/04	1639
36929	30대녀	30대중반 싱글여성 가계부입니다. [16]	zhfn000000	05/01	2685
36928	30대녀	[31세] 마이너스 양심불량 4월 가계부 --:[6]	휘게라이프	05/01	1382
36927		2017년 4월 나도 드디어 결혼이한걸 하나... [8] [BEST]	topsilve	05/01	2299
36925	30대녀	31세 5월의 신부를 앞두고...[3월 미가인♥][9]	넘치는끼토끼	04/29	1409
36924	가족동거녀	[64th] 44살 2017. 4월 가계부 정산[12] 회원공개	깨롱이	04/24	1787
36923	30대녀	(31살) 일상의 반복, 새로운 시작[10] [BEST]	김지봉	04/17	1995
36921	30대녀	[30살] 긍정적인 평범녀의 3월가계부 ♥[7] [BEST]	행위맘	04/08	2198
36917	가족동거녀	가계부를 구성원별로 구분해서 쓰는 방법 [3] 회원공개	생돌님	04/03	446

[그림 5] 모네타 미니가계부

가계부 보다도 금융, 재테크 관련 포털의 기능이 강하며, 수많은 커뮤니티와 동호회와 노하우, 이벤트, 그 외 금융과는 관련 없는 여러 커뮤니티까지 있어 포괄적인 회원 수용이 가능해 보인다. 절약노하우, 카드노하우 등의 커뮤니티는 장점이지만 가계부를 사용하기 위해서는 PC 버전을 설치하고 이용이 가능한 단점이 있다.

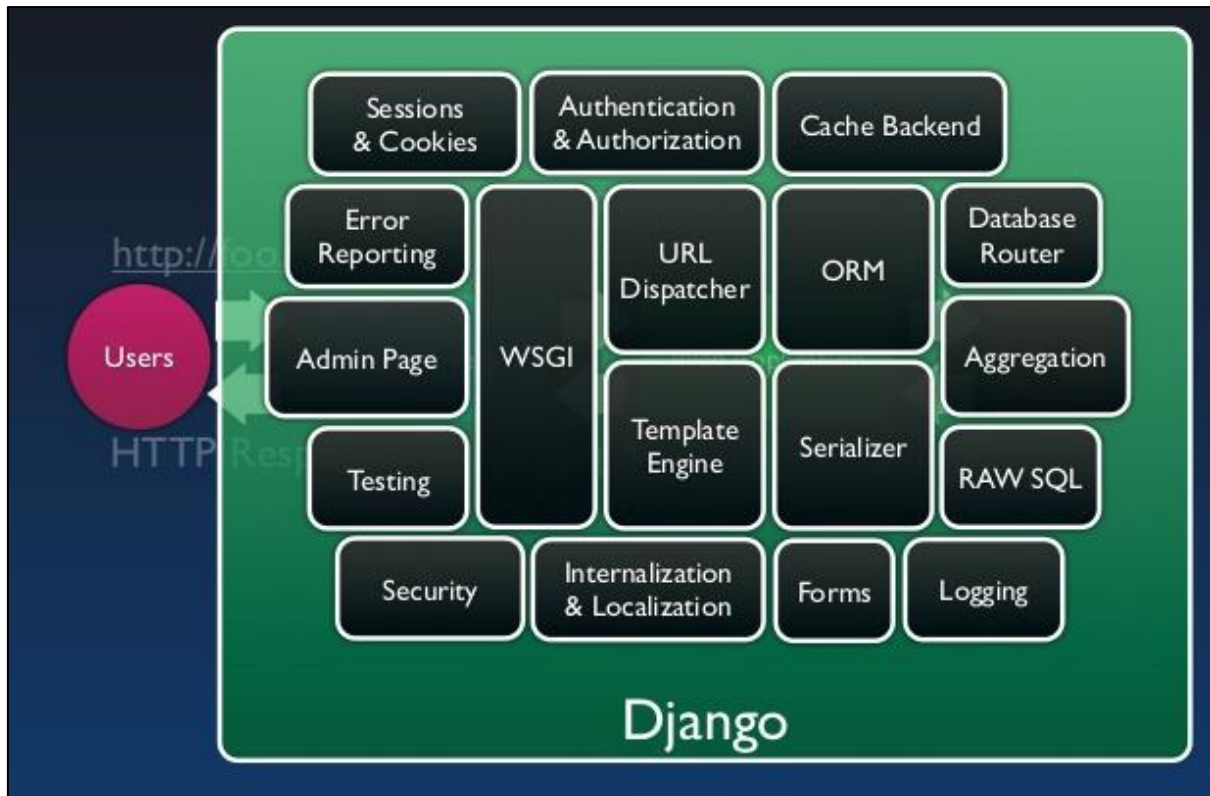
2.2 Django 란?

Django 란 Python 으로 만들어진 무료 오픈소스 웹 어플리케이션 프레임워크 이다. 일반적인 Web Server 는 다음과 같은 역할을 한다.



[그림 6] 일반적인 웹서버의 역할

일반적인 웹 서버와 다르게 Django 는 Full Stack Framework 라고 불리며, Full Stack Framework 란, 하나의 프레임 워크에서 서버가 필요로 하는 대부분의 역할을 혼자 맡아서 하는 것이며 다음 그림은 Django Webserver 의 역할 영역이다.



[그림 7] 장고 웹서버의 역할

이만큼 많은 역할을 하는 장고의 장점은 웹 개발을 쉽게 할 수 있으나 프레임워크의 특성상, 세세한 설정을 변경하거나, 서버와 관련된 자세한 조정을 하는 것은 힘들 수 있다는 단점이 있다.

2.3 Bootstrap 이란?

Bootstrap 은 트위터에서 사용하는 각종 레이아웃, 버튼, 입력창 등의 디자인을 CSS 와 JavaScript 로 만들어 놓은 것이다.

Bootstrap 을 사용하는 이유는 오픈소스이며, 상업적 이용이 가능한 장점이 있기 때문이다. 또한 웹 페이지에서 많이 쓰이는 요소를 전부 내장하고 있고, 편의에 따라 재사용 할 수 있다.

웹 페이지는 Bootstrap 의 CSS 와 JavaScript 관련 이미지만 설치하고 미리 지정된 CSS 클래스나 JavaScript 함수만 불러오면 손쉽게 디자인을 할 수 있으며, 하나의 웹 페이지를 디바이스의 제약을 받지 않는 반응형 웹 디자인을 지원하는 강점이 있다.

3 화면목록 및 흐름도

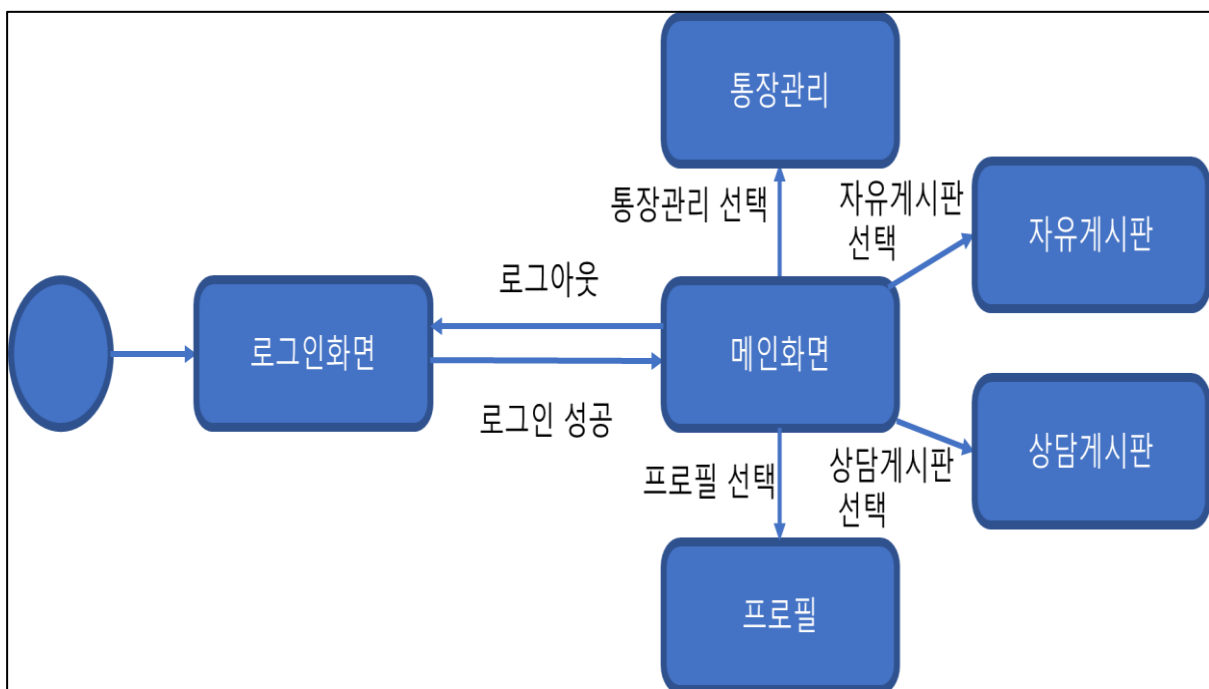
3.1 메인

3.1.1 화면 목록

화면구분	화면명	권한	설명
메인	메인화면	전체	메인에 출력되는 화면으로 최근 게시글, 익명게시판 최근 게시글, 내가쓴 익명 게시판의 글을 확인할 수 있으며 통장관리도 할 수 있는 창

[표 2] 메인 화면목록

3.1.2 화면 흐름도



[그림 8]메인 화면 흐름도

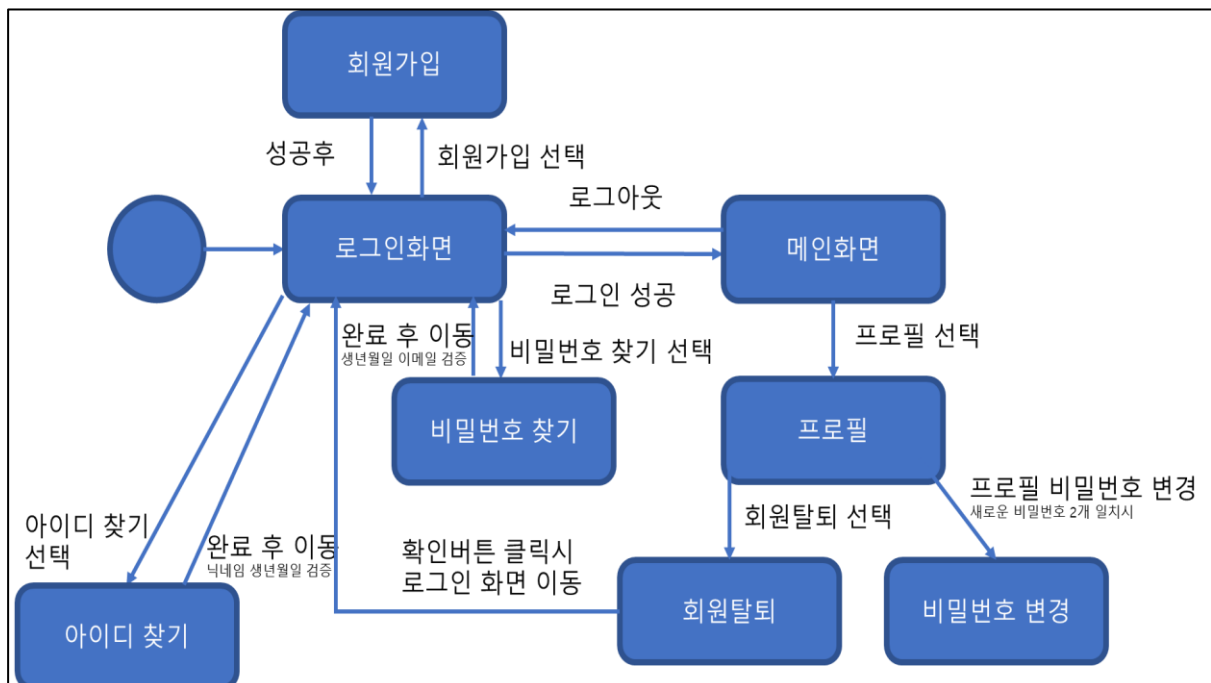
3.2 회원관리

3.2.1 화면 목록

화면구분	화면명	권한	설명
회원관리	회원가입	전체	회원 가입을 할 수 있는 창
	로그인	전체	로그인을 할 수 있는 창
	아이디 찾기	전체	아이디를 찾을 수 있는 창
	비밀번호 찾기	전체	패스워드를 찾을 수 있는 창
	프로필	전체	자신의 개인정보를 볼 수 있는 창
	비밀번호 변경	전체	비밀번호를 변경 할 수 있는 창
	회원 탈퇴	전체	회원정보를 삭제할 수 있는 회원 탈퇴 창

[표 3] 회원관리 화면 목록

3.2.2 화면 흐름도



[그림 9] 회원관리 화면 흐름도

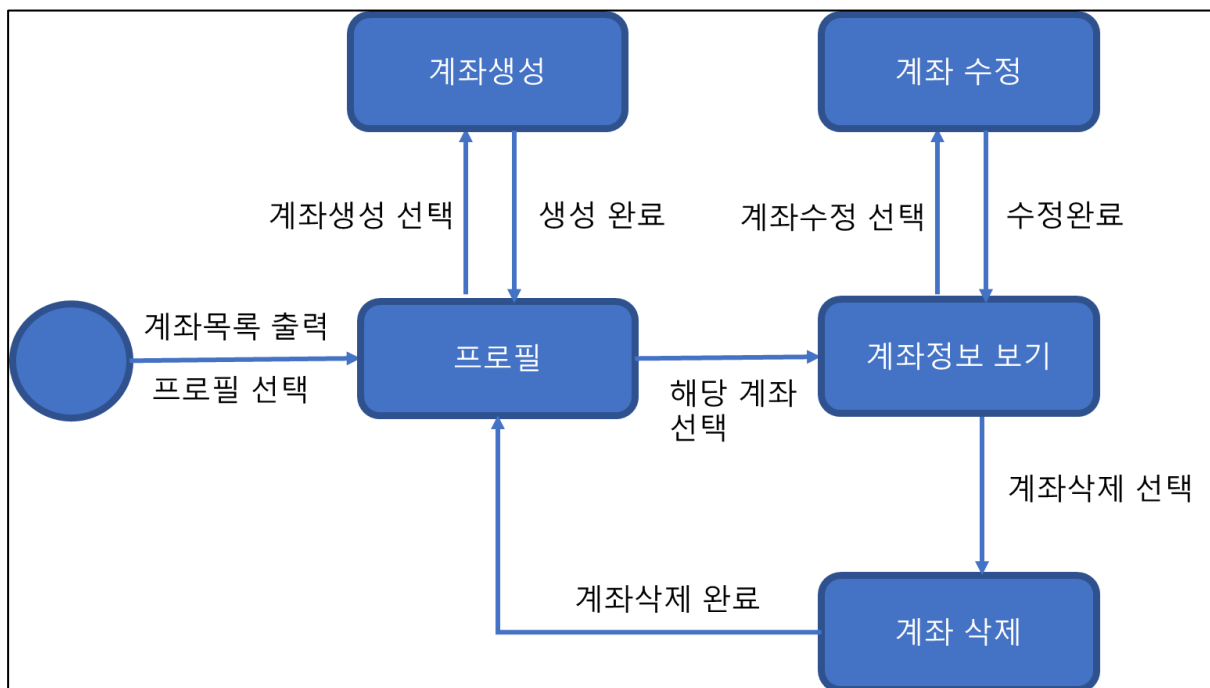
3.3 통장관리

3.3.1 화면 목록

화면구분	화면명	권한	설명
계좌관리	계좌생성	전체	계좌를 생성하는 창
	계좌정보 보기	전체	입력한 계좌 정보를 볼 수 있는 창
	계좌정보 수정	전체	계좌 정보를 수정하는 창
	계좌 삭제	전체	계좌를 삭제하는 창

[표 4] 통장관리 화면목록

3.3.2 화면 흐름도



[그림 10] 통장관리 화면 흐름도

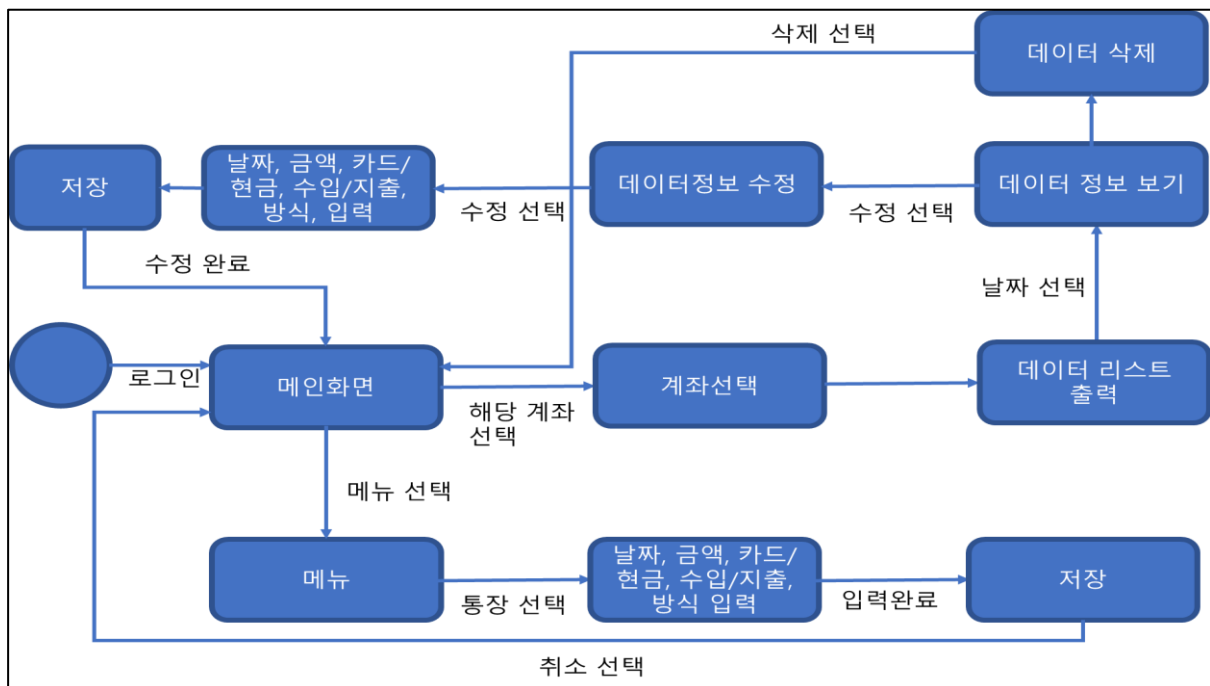
3.4 통장 데이터관리

3.4.1 화면 목록

화면구분	화면명	권한	설명
통장 데이터 추가	데이터생성	전체	데이터를 입력하여 생성하는 창
	데이터 리스트 출력	전체	데이터들을 리스트로 출력하는 창
	데이터정보 보기	전체	데이터의 상세정보를 확인하는 창
	데이터정보 수정	전체	데이터의 정보를 수정하는 창
	데이터 삭제	전체	데이터의 정보를 삭제하는 창

[표 5] 통장 데이터관리 화면 목록

3.4.2 화면 흐름도



[그림 11] 통장 데이터 추가 화면 흐름도

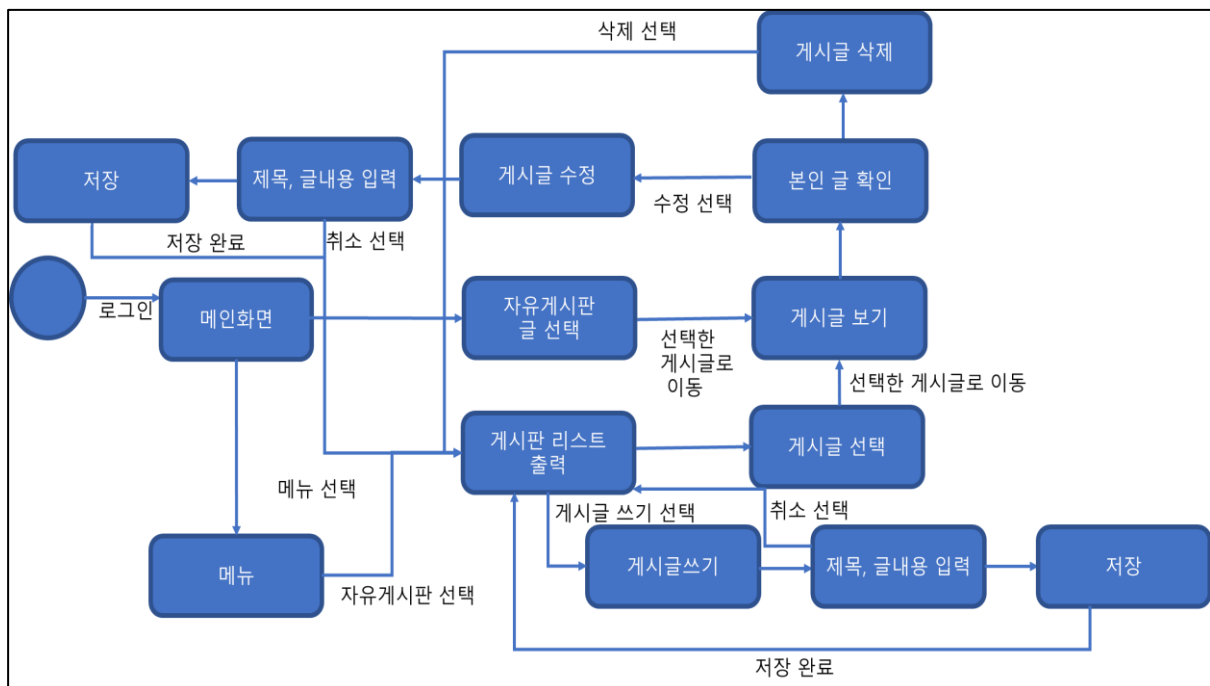
3.5 자유게시판

3.5.1 화면 목록

화면구분	화면명	권한	설명
자유게시판	게시글 리스트	전체	게시글을 리스트로 출력하는 창
	게시글 추가	전체	게시글을 입력하여 글을 추가하는 창
	게시글 출력	전체	게시글을 출력하는 창
	게시글 수정	전체	게시글을 수정하는 창
	게시글 삭제	전체	게시글을 삭제하는 창

[표 6] 자유 게시판 화면 목록

3.5.2 화면 흐름도



[그림 12] 자유게시판 화면 흐름도

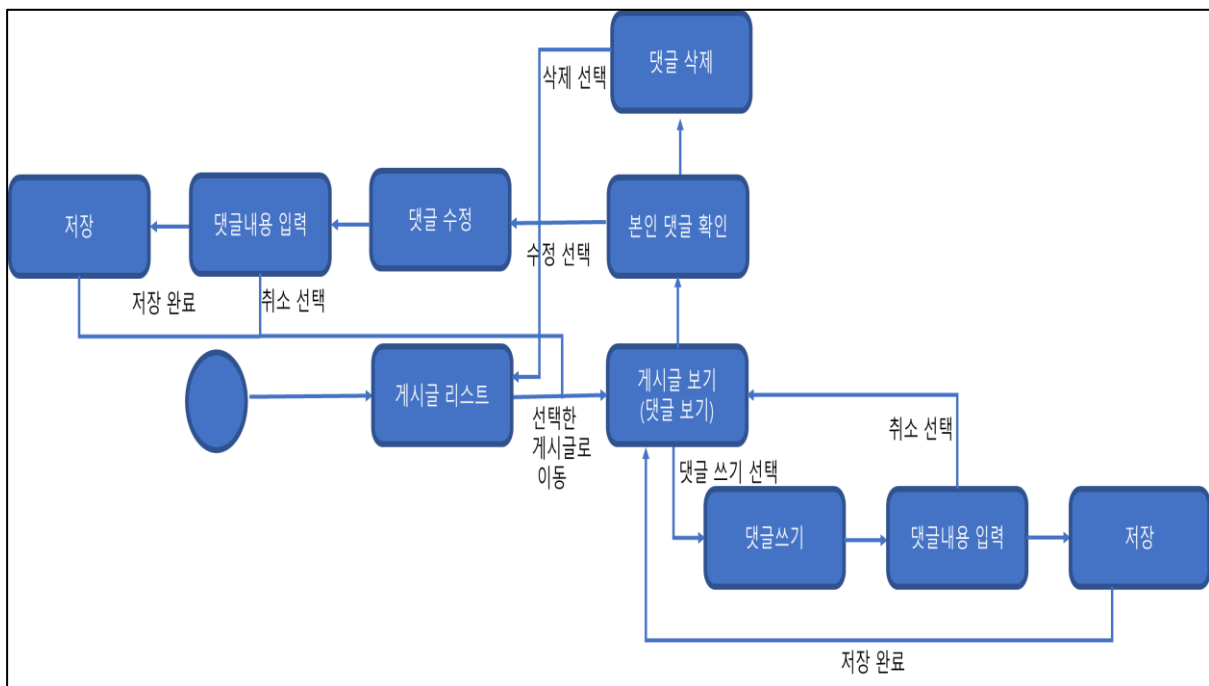
3.6 자유게시판 댓글

3.6.1 화면 목록

화면구분	화면명	권한	설명
자유게시판 댓글	게시글 출력 (댓글 출력)	전체	게시글과 댓글을 한곳에서 출력하는 창
	댓글 추가	전체	댓글을 입력하여 댓글을 추가하는 창
	댓글 수정	전체	댓글을 수정하는 창
	댓글 삭제	전체	댓글을 삭제하는 창

[표 7] 자유게시판 댓글 화면 목록

3.6.2 화면 흐름도



[그림 13] 자유게시판 댓글 화면 흐름도

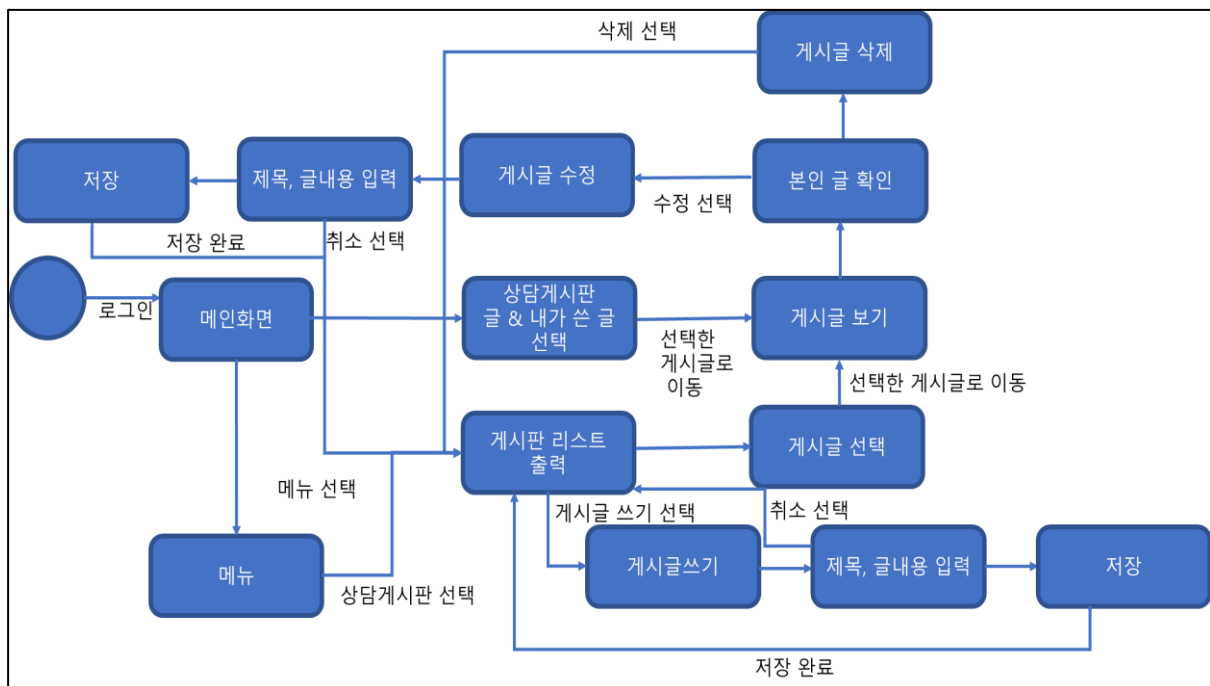
3.7 상담게시판

3.7.1 화면 목록

화면구분	화면명	권한	설명
상담게시판	게시글 리스트	전체	게시글을 리스트로 출력하는 창
	게시글 추가	전체	게시글을 입력하여 글을 추가하는 창
	게시글 출력	전체	게시글을 출력하는 창
	게시글 수정	전체	게시글을 수정하는 창
	게시글 삭제	전체	게시글을 삭제하는 창

[표 8] 상담게시판 화면 목록

3.7.2 화면 흐름도



[그림 14] 상담게시판 화면 흐름도

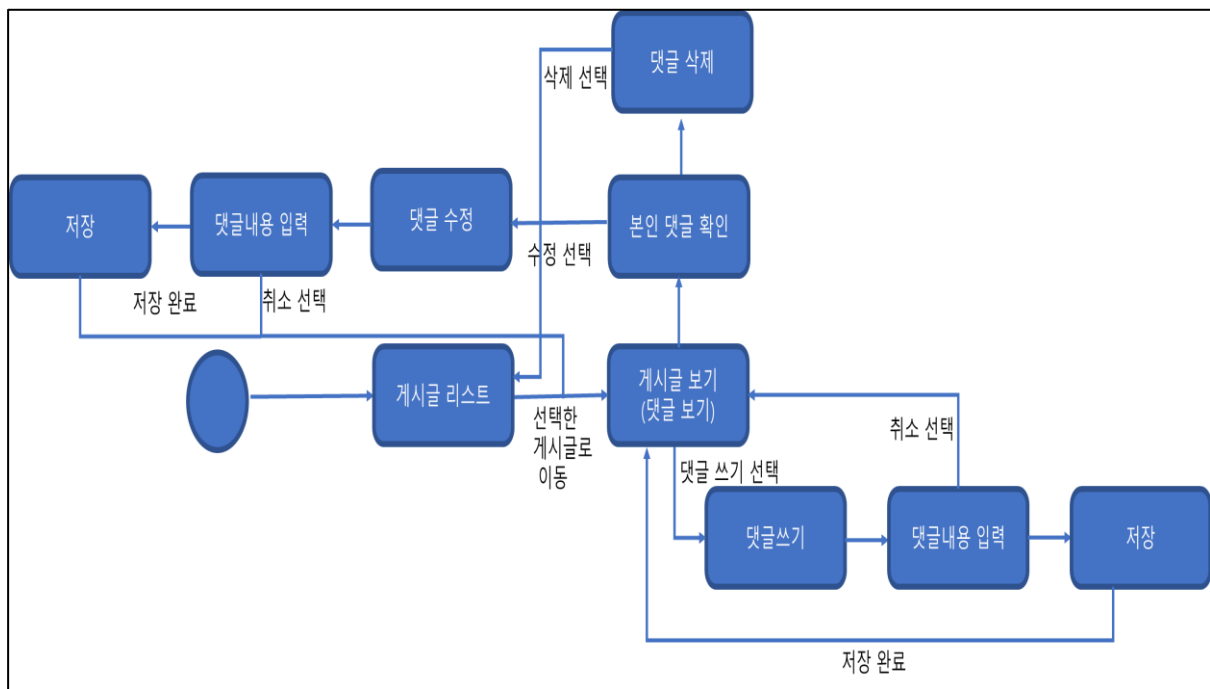
3.8 상담게시판 댓글

3.8.1 화면 목록

화면구분	화면명	권한	설명
상담게시판 댓글	게시글 보기 (댓글 보기)	전체	게시글과 댓글을 한곳에서 출력하는 창
	댓글 쓰기	전체	댓글을 입력하여 글을 추가하는 창
	댓글 수정	전체	댓글을 수정하는 창
	댓글 삭제	전체	댓글을 삭제하는 창

[표 9] 상담게시판 댓글 화면 목록

3.8.2 화면 흐름도



[그림 15] 상담게시판 화면 흐름도

4 DB 설계서

4.1 회원관리

Object Name:	MyUser	(AbstractBaseUser)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
email	EmailField	unique=True, max_length=50	ID로 사용할 이메일
nickname	CharField	unique=True, blank=False, max_length=20	사용자가 사용할 닉네임
birth	DateField	blank=False	사용자의 생년월일

[표 10] 회원관리 DB 설계표

4.2 계좌관리

Object Name:	ActList	(models.Model)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
act	-	ForeignKey(MyUser)	MyUser 참조
act_num	CharField	unique=True, max_length=25, blank=False	계좌번호
act_name	CharField	max_length=20	계좌이름
act_summary	CharField	max_length=50	계좌 한줄 설명
act_info	TextField	max_length=200	계좌정보

[표 11] 계좌관리 DB 설계표

4.3 데이터관리

Object Name:	BankBook	(models.Model)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
name	-	ForeignKey(ActList)	ActList 참조
act_date	DateField	null=False	날짜
act_price	IntegerField	max_length=15	금액
act_payment	CharField	max_length=6	카드/현금 선택
act_part	CharField	max_length=6	수입/지출 선택
act_total	IntegerField	default=0	계좌 잔여금액, 따로 입력받지 않고 자동 갱신
act_content	CharField	max_length=50	소비내용

[표 12] 데이터관리 DB 설계표

4.4 자유게시판

Object Name:	ActBoard	(models.Model)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
board_name	-	ForeignKey(MyUser)	MyUser 참조
board_title	CharField	max_length=40	글제목
board_nick	CharField	max_length=40	닉네임, 따로 입력받지 않음
board_content	CharField	max_length=3000	글내용
create_time	DateTimeField	auto_now_add=True	작성시간, 자동입력
update_time	DateTimeField	auto_now=True	수정시간, 자동입력
board_hit	IntegerField	default=0	조회수

[표 13] 자유게시판 DB 설계표

4.5 자유게시판 댓글

Object Name:	ActComment	(models.Model)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
act_post	-	ForeignKey(ActBoard)	ActBoard 참조
act_author	-	ForeignKey(MyUser)	MyUser 참조
act_message	TextField	-	댓글내용
create_time	DateTimeField	auto_now_add=True	작성시간, 자동입력
update_time	DateTimeField	auto_now=True	수정시간, 자동입력

[표 14] 자유게시판 댓글 DB 설계표

4.6 상담게시판

Object Name:	Post	(models.Model)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
title	CharField	max_length=100	글제목
writer	TextField	-	닉네임, 따로 입력받지 않음
content	TextField	-	글내용
create_time	DateTimeField	auto_now_add=True	작성시간, 자동입력
update_time	DateTimeField	auto_now=True	수정시간, 자동입력

[표 15] 상담게시판 DB 설계표

4.7 상담게시판 댓글

Object Name:	Comment	(models.Model)	
필드명	타입	제약조건	설명
id	IntegerField	PK, Auto Increment	기본키
post	-	ForeignKey(Post)	Post 참조
author	-	ForeignKey(MyUser)	MyUser 참조
message	TextField	-	댓글내용
create_time	DateTimeField	auto_now_add=True	작성시간, 자동입력
update_time	DateTimeField	auto_now=True	수정시간, 자동입력

[표 16] 상담게시판 댓글 DB 설계표

5 URL 설계

5.1 회원관리

URL	View	Template	설명
/admin/	admin.site.urls	-	관리자 페이지
/join/	views.join	join.html	회원가입 페이지
/	auth_views.login	login.html	로그인 페이지
/logout/	auth_views.logout	-	로그아웃
/changepw/	views.changepw	changepw.html	비밀번호 변경페이지
/searchemail/	SearchEmail.as_view()	search_email.html	이메일찾기 페이지
/findusername/	views.find_username	find_username.html	유효성 검사후 정상시 이메일 반환
/searchpassword/	SearchPassword.as_view()	search_password.html	비밀번호 찾기 페이지
/findpassword/	views.findpassword	search_password.html	유효성 검사해서 정상시 비밀번호 변경 폼 반환
/ForgetChangePw/<email>/	views.forgetchangepw	search_password.html	입력한 두개의 비밀번호가 일치하면 변경 성공
/mypage/	views.mypage	mypage.html	프로필 정보 출력 페이지(계좌 출력, 추가가능)
/user_delete_confirm/	DeleteConfirm.as_view()	my_delete_identification.html	정말 회원을 탈퇴할 것인지 물어보는 페이지
/user_delete/	views.delete_user	Withdrawal	회원탈퇴 처리후 안내메시지

[표 17] 회원관리 URL 설계표

5.2 계좌관리

URL	View	Template	설명
/mypage/<ActList_id>/	views.account_view	account_view.html	계좌 정보 출력 페이지
/mypage/edit/<ActList_id>/	views.account_edit	account_edit.html	계좌 정보 수정 페이지
/mypage/delete/<ActList_id>/	views.account_delete	account_delete.html	계좌 정보 삭제 페이지

[표 18] 계좌관리 URL 설계표

5.3 데이터관리

URL	View	Template	설명
/index/	views.index	index.html	메인페이지
add/<BankBook_id>/	views.bankbook_new	bankbook.html	데이터추가 페이지
index/my_list/<ActList_id>/	views.my_list	my_list.html	계좌에 있는 데이터를 지출별, 수입별 전체 출력
index/my_view/<ActList_id>/	views.my_view	my_view.html	입력한 데이터의 정보 출력 페이지
index/my_edit/<ActList_id>/	views.my_edit	my_edit.html	입력한 데이터의 정보 수정 페이지
index/my_delete/<ActList_id>/	views.my_delete	my_delete.html	입력한 데이터를 삭제하는 페이지

[표 19] 데이터관리 URL 설계표

5.4 자유게시판

URL	View	Template	설명
/board/list/	views.board_list	board_list.html	게시판 리스트 출력 페이지
/board/write/	views.board_write	board_write.html	게시판 게시물 입력 페이지
/board/view/<ActBoard_id>/	views.board_view	board_view.html	게시판 게시물 확인 페이지
/board/edit/<ActBoard_id>/	views.board_edit	board_edit.html	게시판 게시물 수정 페이지
/board/delete/<ActBoard_id>/	views.board_delete	board_delete.html	게시판 게시물 삭제 페이지

[표 20] 자유게시판 URL 설계표

5.5 자유게시판 댓글

URL	View	Template	설명
<ActBoard_pk>/act_comment/write	views.act_comment_write	act_comment_form.html	게시판 댓글 추가 페이지
<ActBoard_pk>/act_comment/<ActComment_pk>/edit/	views.act_comment_edit	act_comment_form.html	게시판 댓글 수정 페이지
<ActBoard_pk>/act_comment/<ActComment_pk>/delete/	views.act_comment_delete	act_comment_confirm_delete.html	게시판 댓글 삭제 페이지

[표 21] 자유게시판 댓글 URL 설계표

5.6 상담게시판

URL	View	Template	설명
/post/list/	views.post_list	post_list.html	상담게시판 리스트 출력 페이지
/post/new/	views.post_new	post_new.html	상담게시판 게시물 입력 페이지
/post/<Post_id>/	views.post_detail	post_detail.html	상담게시판 게시물 확인 페이지
/post/<Post_id>/edit/	views.post_edit	post_edit.html	상담게시판 게시물 수정 페이지
/post/<Post_id>/delete/	views.post_delete	post_delete.html	상담게시판 게시물 삭제 페이지

[표 22] 상담게시판 URL 설계표

5.7 상담게시판 댓글

URL	View	Template	설명
post/<Post_pk>/comment/new/	views.comment_new	comment_form.html	상담게시판 댓글 추가 페이지
post/<Post_pk>/comment/<Comment_pk>/edit/	views.comment_edit	comment_form.html	상담게시판 댓글 수정 페이지
post/<Post_pk>/comment/<Comment_pk>/delete/	views.comment_delete	comment_confirm_delete.html	상담게시판 댓글 삭제 페이지

[표 23] 상담게시판 댓글 URL 설계표

6 기능정의

6.1 회원관리

함수 / 클래스 명	설명	Input	Output
login	#주요 기능은 auth_views.login에게 상속받아 템플릿만 출력	request	-
logout	#주요 기능은 auth_views.logout에게 상속받아 템플릿만 출력	request	-
join	#회원가입을 담당하는 뷰이다. #form: POST로 요청이 들어올 경우 템플릿에서 입력한 UserCreationForm을 request받아 해당 폼의 유효성을 검사한 후 문제가 없으면 저장하고 로그인 페이지로 리다이렉트 GET으로 요청이 들어올 경우 UserCreationForm을 응답	request	form
SearchEmail	#search_email.html로 템플릿 네임을 지정	-	-
find_username	#사용자가 닉네임과 생년월일을 입력하고 유효성검사에 통과하면 email을 반환하는 뷰이다. #POST로 요청이 들어오면 FindUserNameForm을 사용자가 입력한 데이터와 함께 응답이 들어오고 응답받은 닉네임, 생년월일과 DB안에 있는 닉네임과 생년월일이 같을시에 이메일을 entry로 리턴시켜주며 GET으로 요청이 들어온다면 빈 폼을 리턴	request	entry, form
SearchPassword	#search_password.html로 템플릿 네임을 지정	-	-
find_password	#사용자가 이메일 주소와 생년월일을 입력하고 유효성 검사에 통과 하면 비밀번호 변경폼을 리턴시키는 뷰이다. #POST로 요청이 들어오면 사용자가 입력하여 응답받은 이메일과 생년월일과 DB안의 이메일과 생년월일이 일치하는 지 확인하고 일치한다면 유효성 검사 후에 비밀번호를 변경할 폼을 반환 GET으로 요청이 들어온다면 이메일주소와 생년월일을 입력할 폼을 리턴	request	entry, form
changepw	#비밀번호를 변경을 담당하는 뷰이다 #POST로 요청이 들어오면 비밀번호, 비밀번호 확인 두개의 입력이 같은지 확인을 하고 같다면 사용자의 비밀번호를 변경 GET으로 요청이 들어오면 비밀번호를 변경할 비밀번호, 비밀번호확인 이 있는 ChangePWForm을 리턴시킨다.	request	form
DeleteConfirm	#my_delete_indentification.html로 템플릿 네임 지정	-	-
delete_user	#회원 탈퇴를 담당하는 뷰이다 #POST로 요청이 들어오면 입력한 비밀번호가 일치하는지 확인하고 일치한다면 정말로 탈퇴할 것인지 물어보는 폼을 리턴 탈퇴하기 버튼을 클릭하면 유저의 계정정보가 삭제되고 5초뒤 로그인 페이지로 이동 GET으로 요청이 들어오면 본인확인을 하기위한 비밀번호 입력 폼을 리턴	request	-

[표 24] 회원관리 기능정의표

6.2 계좌관리

함수 / 클래스 명	설명	Input	Output
mypage	# 사용자의 프로필정보와 계좌리스트, 추가, 수정, 삭제를 담당하는 뷰이다. #qs_li: 메뉴와 mypage의 계좌들을 목록으로 출력하는 쿼리셋 #form: POST요청이 들어오면 ActListForm을 응답받아 유효성을 검사한 후 저장하며 사용자는 따로 입력받지 않고 현재 사용자로 저장되며 GET으로 요청이 들어오면 ActListForm을 출력 #저장이 완료되면 mypage로 리다이렉트	request	qs_li, form
account_view	#계좌의 상세정보 출력을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #qs_view: 응답받은 id에 매칭되는 데이터를 출력해주는 쿼리셋 #계좌 상세정보 출력페이지 리턴	request, id	qs_li, qs_view
account_edit	#계좌의 정보수정을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: POST로 요청이 들어올 경우 인스턴스와 같이 응답받은 ActListForm을 유효성 검사후에 저장한 후 응답받은 id에 대한 계좌정보로 리다이렉트 한다 GET으로 요청이 들어올 경우 ActListForm을 id와 매칭되는 인스턴스와 함께 응답	request, id	qs_li, form
account_delete	#계좌의 삭제를 담당하는 뷰이다. #요청이 POST로 들어올 경우에 받은 id와 맞게 매칭된 계좌의 데이터를 삭제 #삭제후 mypage로 리다이렉트	request, id	-

[표 25] 계좌관리 기능정의표

6.3 데이터관리

함수 / 클래스 명	설명	Input	Output
index	#메인화면의 출력을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #qs_board_list: 자유게시판의 최신 5개의 글을 출력하는 쿼리셋 #qs_post_list: 상담게시판의 최신 5개글을 출력하는 쿼리셋 #qs_my_post_list: 상담게시판의 자신이 쓴 글만 출력하는 쿼리셋 #qs_income_rank: 저축왕을 출력하는 쿼리셋 #메인 페이지로 리턴	request	qs_li, qs_board_list, qs_post_list, qs_my_post_list, qs_income_rank
my_list	#계좌별 수입/지출 데이터를 출력해주는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #contacts: 해당 계좌의 수입/지출 전체를 출력한 후 페이징처리하여 넘김 #contacts1: 해당 계좌의 수입 전체를 출력한 후 페이징처리하여 넘김 #contacts2: 해당 계좌의 지출 전체를 처리한 후 페이징처리하여 넘김 #qs_income_graph: act_part가 수입인 데이터만 최신날짜순으로 출력하는 쿼리셋 #qs_expenses_graph: act_part가 지출인 데이터만 최신날짜순으로 출력하는 쿼리셋 #qs_total: 해당 계좌의 모든 가격을 계산해서 계좌의 남은돈을 출력하기 위한 쿼리셋 #qs_total_income: act_part가 수입인 데이터만 계산하여 계좌의 총 수입을 계산하는 쿼리셋 #qs_total_exepenses: act_part가 지출인 데이터만 계산하여 계좌의 총 지출을 계산하는 쿼리셋 #계좌 관리 페이지로 리턴	request, id	qs_li, contacts, contacts1, contacts2, qs_income_graph, qs_expenses_graph, qs_total, qs_total_income, qs_total_exepenses
my_view	#해당 계좌에 입력된 데이터의 상세정보를 출력해주는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #qs_view: 받은 id값과 일치하는 BankBook모델의 데이터들을 출력하는 쿼리셋 #데이터 상세정보 출력페이지로 리턴	request, id	qs_li, qs_view
bankbook_new	#해당 계좌에 수입/지출 데이터를 추가하기 위한 뷰이다. #POST로 요청이 들어오면 유효성 검사를 마친후 id(계좌)에 맞게 입력하고, 수입일경우 금액을 total에 + 지출일경우 금액을 total에 -하여 한번에 품을 저장 #저장한 후 메인페이지로 리다이렉트	request, id	
my_edit	#해당 계좌에 입력된 데이터를 수정하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 qs_view: GET으로 요청이 들어오면 BankBookForm에 받은 id에 맞춰 instance를 리턴해주고 POST로 요청이 들어온다면 요청들어온 품의 데이터들을 저장 #저장한후 상세정보를 출력해주는 my_view로 리다이렉트	request, id	qs_li, form
my_delete	#해당 계좌에 입력된 데이터를 삭제하는 뷰이다. #요청이 POST로 들어올 경우에 받은 id와 맞게 매칭된 게시글 데이터를 삭제 #삭제후 해당 계좌의 리스트로 리다이렉트	request, id	-

[표 26] 데이터관리 기능정의표

6.4 자유게시판

함수 / 클래스 명	설명	Input	Output
board_list	#자유게시판의 출력을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #contacts: 게시글들 전부를 로드하여 페이징 처리하여 템플릿으로 넘겨줌 #게시글 리스트 리턴	request	qs_li, contacts
board_view	#자유게시판의 게시글 출력을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #조회수 1 증가시켜 저장시킨 후 #post: 입력받은 id와 맞는 게시글을 get_object_or_404로 반환하여 리턴	request, id	qs_li, post
board_write	#자유게시판의 게시글을 추가를 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 요청이 GET이면 PostForm을 리턴하며 POST요청이 들어올 경우 유효성 검사 후 닉네임필드를 현재 사용자로 먼저 입력한 후 폼을 저장(사용자 임의 변경 방지) #저장 완료후 게시글 리스트로 리다이렉트	request	qs_li, form
board_edit	#자유상담게시판의 게시글 수정을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 요청이 GET이면 id와 일치하는 기존의 저장된 인스턴스들을 폼에 출력하여 리턴시키고 POST로 요청이 오면 유효성 검사 후 저장 #저장후 id게시글로 리다이렉트	request, id	qs_li, form
board_delete	#자유상담게시판의 게시글 삭제를 담당하는 뷰이다. #요청이 POST로 들어올 경우에 받아온 id와 맞게 매칭된 게시글 데이터를 삭제 #삭제후 게시글 리스트로 리다이렉트	request, id	-

[표 27] 자유게시판 기능정의표

6.5 자유게시판 댓글

함수 / 클래스 명	설명	Input	Output
act_comment_write	#자유게시판 게시글의 댓글추가를 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 요청이 GET이면 CommentForm을 리턴하며 POST요청이 들어올 경우 유효성을 검사한 후 사용자는 기존의 사용자로 저장한 후 폼저장(사용자 임의 변경 방지) #저장완료후 해당 post_pk게시글로 리다이렉트	request, post_pk,	qs_li, form
act_comment_edit	#자유게시판 게시글의 댓글수정을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 게시글의 post_pk, 댓글의 pk를 각각 받아온 후 GET요청이 들어오면 instance를 CommentForm에 입력하여 리턴하고 POST요청이 들어오면 유효성 검사 후 저장 #저장완료후 해당 post_pk게시글로 리다이렉트	request, post_pk, pk	qs_li, form
act_comment_delete	#자유게시판 게시글의 댓글삭제를 담당하는 뷰이다. #comment: 게시글의 작성자가 로그인 사용자와 일치하지 않으면 바로 게시글로 리턴시키고 일치하며 POST로 요청이 들어왔을시 get_object_or_404로 받아온 actcomment의 모든 정보를 삭제 #삭제후 해당 post_pk 게시글로 리다이렉트	request, post_pk, pk	actcomment

[표 28] 자유게시판 댓글 기능정의표

6.6 상담게시판

함수 / 클래스 명	설명	Input	Output
post_list	#상담게시판의 출력을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #contacts: 게시글들 전부를 로드하여 페이징 처리하여 템플릿으로 넘겨줌 #게시글 리스트로 리턴	request	qs_li, contacts
post_detail	#상담게시판의 게시글 출력을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #post: 입력받은 pk와 맞는 게시글을 get_object_or_404로 반환하여 리턴	request, pk	qs_li, post
post_new	#상담게시판의 게시글을 추가를 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 요청이 GET이면 PostForm을 리턴하며 POST요청이 들어올 경우 유효성 검사후 폼을 저장 #저장후 게시글 리스트로 리다이렉트	request	qs_li, form
post_edit	#상담게시판의 게시글 수정을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 요청이 GET이면 pk와 일치하는 기존의 저장된 인스턴스들을 폼에 출력하여 리턴시키고 POST로 요청이 오면 유효성 검사후 저장 #저장후 게시글 pk로 리다이렉트	request, pk	qs_li, form
post_delete	#상담게시판의 게시글 삭제를 담당하는 뷰이다. #post: 요청이 POST로 들어올 경우에 받아온 pk와 맞게 매칭된 게시글 데이터를 삭제 #삭제후 게시글 리스트로 리다이렉트	request, pk	post

[표 29] 상담게시판 기능정의표

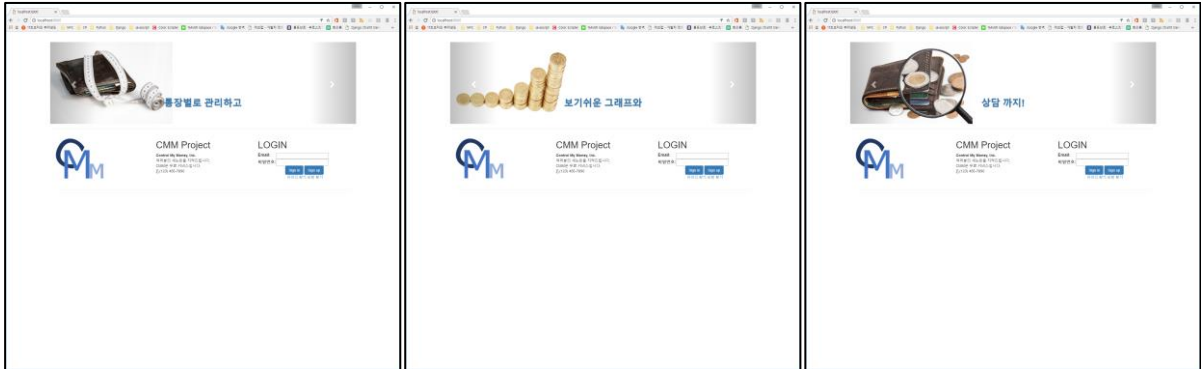
함수 / 클래스 명	설명	Input	Output
comment_new	#상담게시판 게시글의 댓글추가를 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 요청이 GET이면 CommentForm을 리턴하며 POST요청이 들어올 경우 유효성을 검사한 후 사용자는 기존의 사용자로 저장한 후 폼저장(사용자 임의 변경 방지) #저장완료후 해당 post_pk게시글로 리다이렉트	request, post_pk,	qs_li, form
comment_edit	#상담게시판 게시글의 댓글수정을 담당하는 뷰이다. #qs_li: 메뉴의 계좌들을 출력하는 쿼리셋 #form: 게시글의 post_pk, 댓글의 pk를 각각 받아온 후 GET요청이 들어오면 instance를 CommentForm에 입력하여 리턴하고 POST요청이 들어오면 유효성 검사후 저장 #저장완료후 해당 post_pk게시글로 리다이렉트	request, post_pk, pk	qs_li, form
comment_delete	#상담게시판 게시글의 댓글삭제를 담당하는 뷰이다. #comment: 게시글의 작성자가 로그인 사용자와 일치하지 않으면 바로 게시글로 리턴시키고 일치하며 POST로 요청이 들어왔을시 get_object_or_404로 받아온 comment의 모든 정보를 삭제 #삭제후 해당 post_pk 게시글로 리다이렉트	request, post_pk, pk	comment

[표 30] 상담게시판 댓글 기능정의표

7 결과화면

7.1 로그인

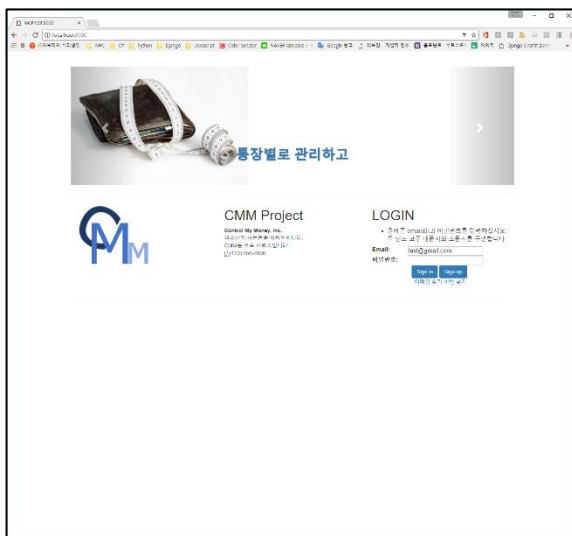
7.1.1 로그인화면



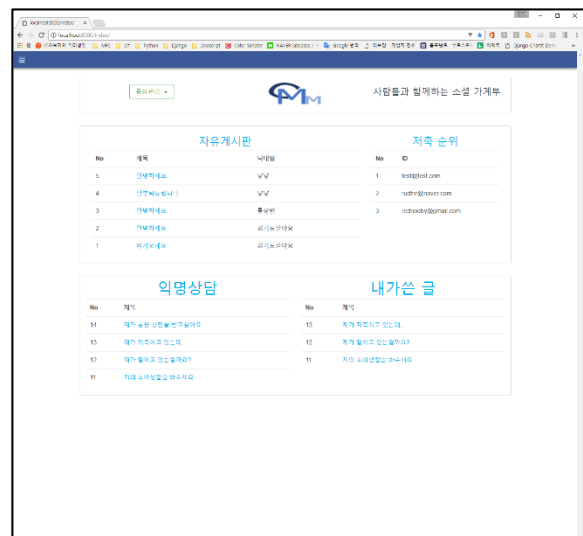
[그림 16] 로그인 화면

[그림 16]은 CMM의 가장 최상위 페이지이다. CMM은 회원가입을 해야만 이용할 수 있으며 캐러셀을 CMM의 광고를 한눈에 볼 수 있는 장점이 있다.

7.1.2 로그인



[그림 17] 다른 정보 입력



[그림 18] 로그인 완료

[그림 17]은 로그인하려는 사용자가 다른 정보를 입력하였을 때 출력되는 에러이고 [그림 18]은 로그인을 정상적으로 성공했을 때만 메인페이지로 넘어간다.

7.1.3 회원가입

회원가입

example@abc.com

닉네임

yyyy-mm-dd

비밀번호

비밀번호 확인

완료 취소

[그림 19] 회원가입 화면

회원가입

example@abc.com

닉네임 이 입력문을 작성하세요.

yyyy-mm-dd

비밀번호

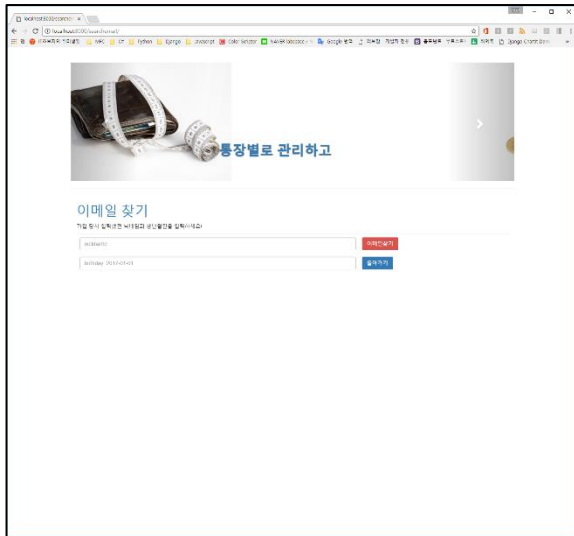
비밀번호 확인

완료 취소

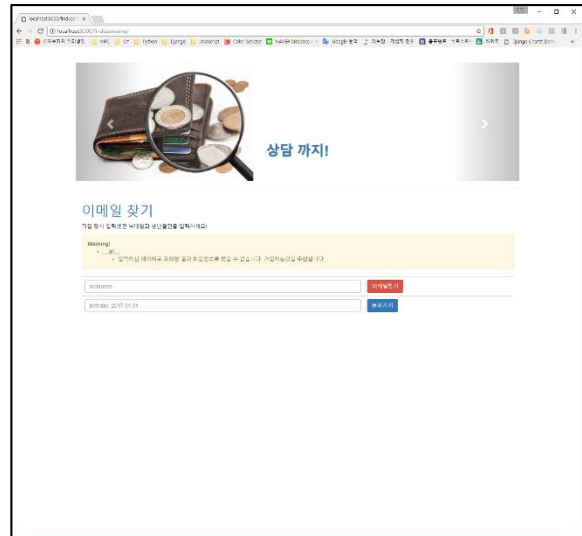
[그림 20] 회원가입 빈칸입력

[그림 19]는 회원가입 화면이며 사용자의 개인정보를 입력한 후 완료 버튼을 누르면 유효성 검사를 마치고 저장이 되지만 [그림 20]과 같이 입력하지 않고 확인 버튼을 누르면 넘어가지 않는다. 또한 사용자가 회원가입을 원하지 않을 시 취소 버튼을 누르게 되면 로그인 페이지로 되돌아 갈 수 있다.

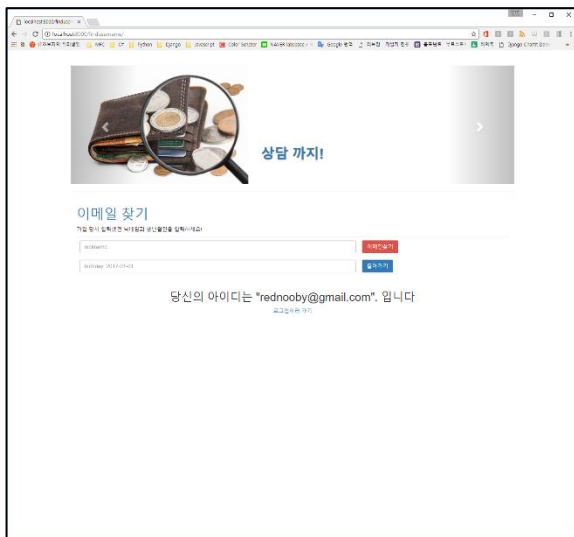
7.1.4 이메일 찾기



[그림 21] 이메일 찾기 화면



[그림 22] 다른 정보를 입력



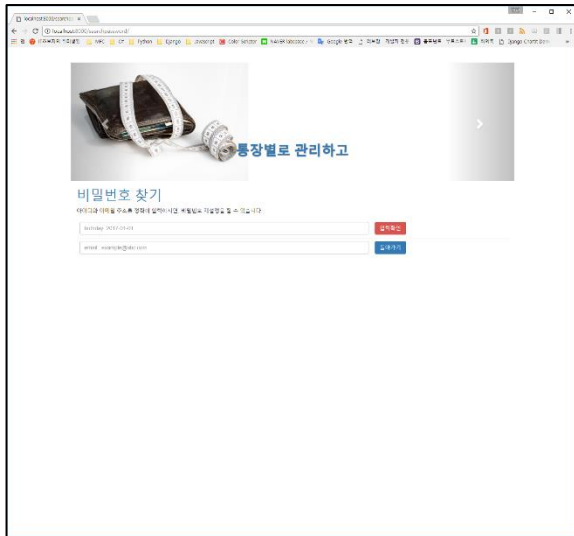
[그림 23] 이메일 찾기 완료

[그림 21]은 사용자가 이메일을 찾기 위한 입력 페이지이며 닉네임과 생년월일을 입력하여 이메일을 찾는 방식이다.

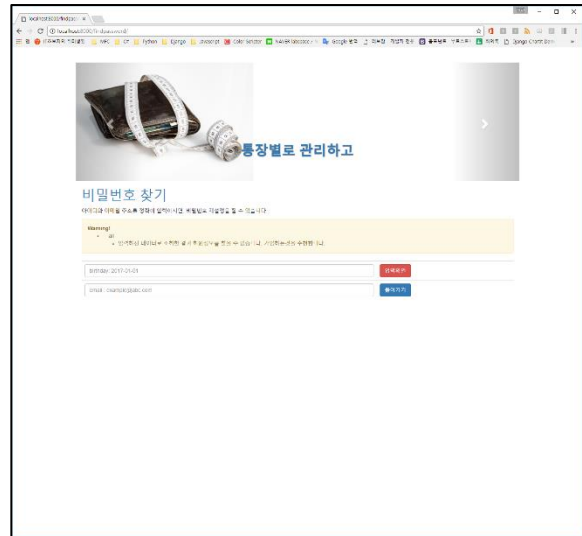
[그림 22]는 입력한 사용자의 정보가 일치하지 않을 경우 나타나는 에러출력문이다.

[그림 23]은 사용자가 올바른 데이터를 입력하여 유효성검사를 마치면 정상적으로 리턴 되는 화면이다.

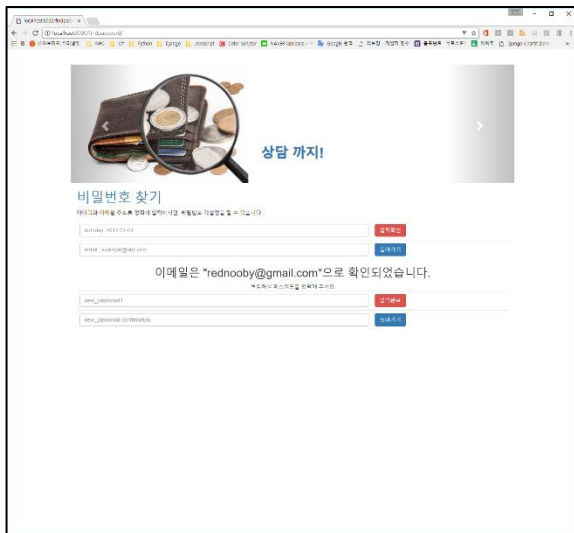
7.1.5 비밀번호찾기



[그림 24] 비밀번호 찾기 화면



[그림 25] 다른 정보 입력



[그림 26] 패스워드 변경 폼

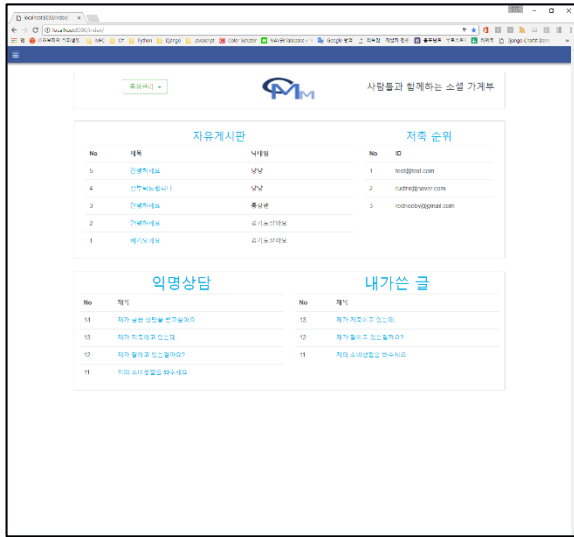
[그림 24]는 사용자가 비밀번호를 찾기 위하여 생년월일과 이메일을 입력하는 페이지이다.

[그림 25]는 다른 정보 입력 시 나오는 에러메시지 이다.

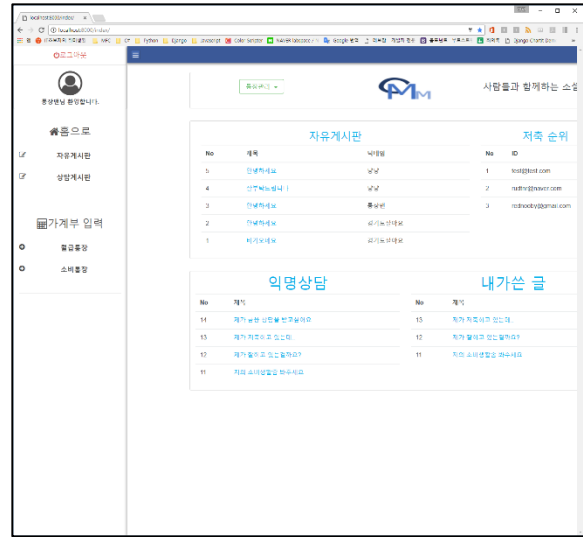
[그림 26]은 올바른 데이터를 입력하면 이메일 리턴과 함께 패스워드를 변경하는 폼이 나오게 되며 정상적으로 패스워드를 변경하면 로그인 페이지로 리다이렉트 된다.

7.2 메인

7.2.1 메뉴



[그림 27] 메인화면



[그림 28] 메뉴화면

[그림 27]은 로그인 성공시 나오는 메인 화면이며 좌측 상단의 메뉴 버튼을 클릭하면 메인화면은 우측으로 밀리면서 [그림 28]과 같은 메뉴가 출력된다.

또한 메인 화면에는 자유게시판의 새 글과, 수입/지출 데이터 등록시 내용을 '적금'으로 등록한 데이터들을 합산하여 저축순위를 출력하고, 익명상담 게시판의 최신글과 익명게시판의 자신이 쓴 글만 출력하여 굳이 자신의 닉네임을 찾지 않고 손쉽게 글을 확인할 수 있다.

[그림 28]은 메뉴 최상단의 로그아웃 버튼을 누르면 로그아웃이 되고, 마이페이지 아이콘과 함께 사용자의 닉네임님 환영합니다 라는 문구가 나오게 되며 가계부 입력 파트에는 사용자가 등록한 계좌만큼 출력이 되어 어느 페이지에 있어도 쉽게 수입/지출 데이터를 저장할 수 있다.

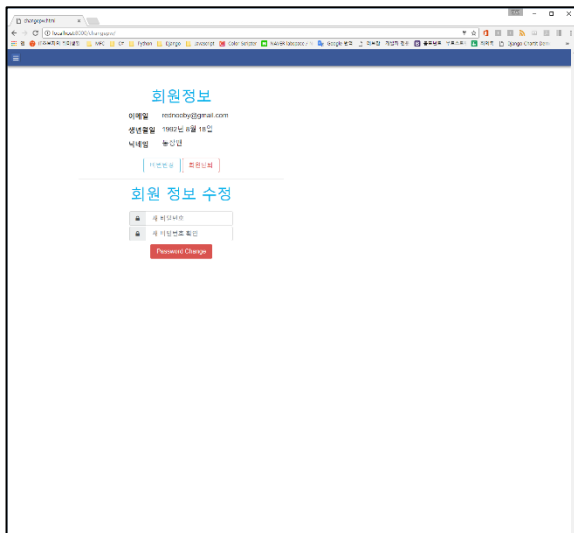
7.4 마이페이지



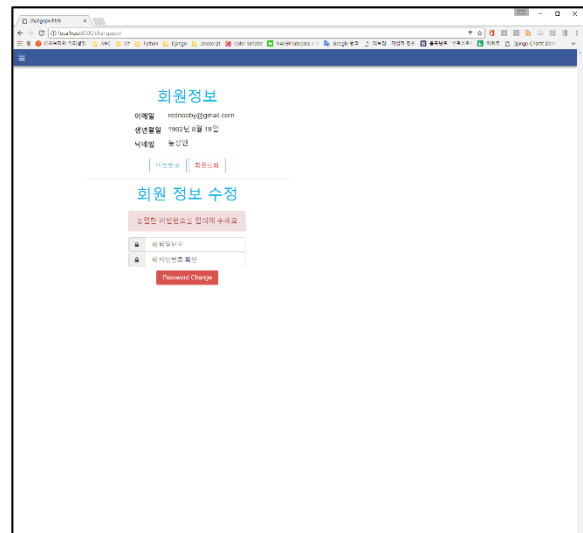
[그림 29] 마이페이지 화면

마이페이지 화면은 [그림 29]와 같고 비밀번호변경, 회원탈퇴를 할 수 있으며 계좌리스트확인, 계좌 정보확인, 계좌추가, 계좌정보수정, 계좌삭제를 할 수 있다.

7.4.1 비밀번호변경



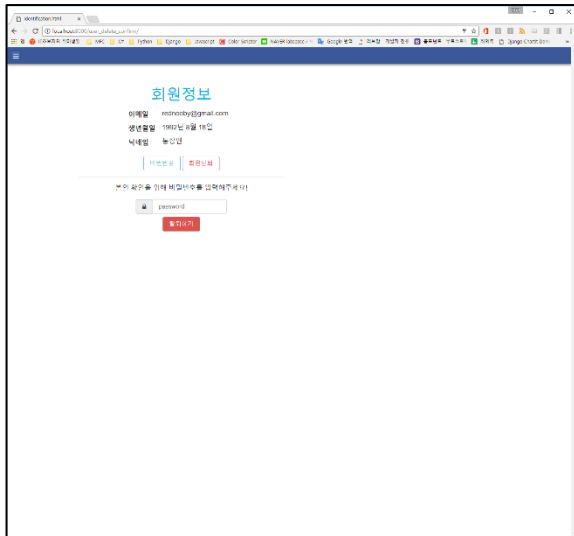
[그림 30] 비밀번호변경 화면



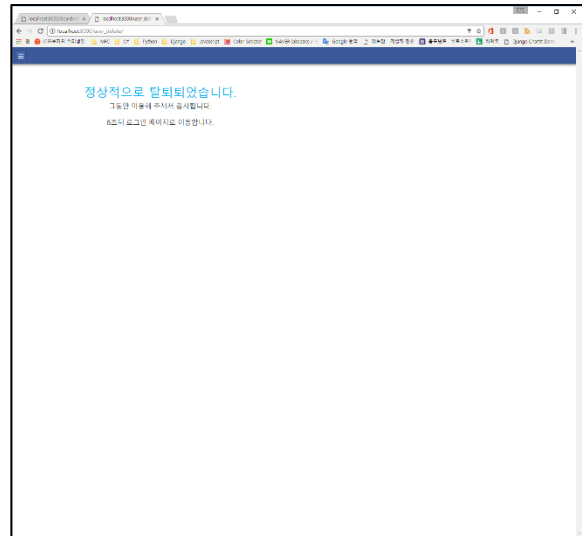
[그림 31] 비밀번호변경 유효성 검사

마이페이지에서 비밀번호변경버튼을 누르면 [그림 30]과 같은 페이지로 이동이 되는데 만약 사용자가 새로운 비밀번호를 일치하지 않게 입력했다면 [그림 31]과 같은 에러메시지를 출력한다.

7.4.2 회원탈퇴



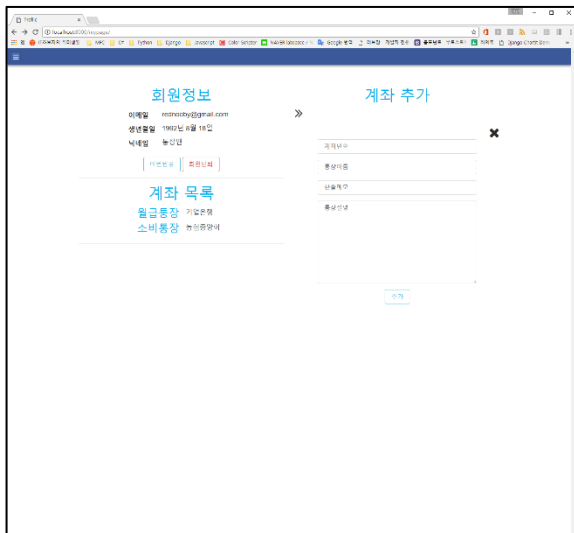
[그림 32] 회원탈퇴 화면



[그림 33] 탈퇴완료

회원 탈퇴 버튼을 누르면 [그림 32]와 같이 본인 인증을 위한 비밀번호 입력창이 뜨는데 비밀번호를 알맞게 입력하면 [그림 33]화면으로 넘어가며 탈퇴가 완료된 것을 확인할 수 있고 5 초뒤 자동으로 로그인페이지로 이동하게 된다.

7.4.3 계좌추가



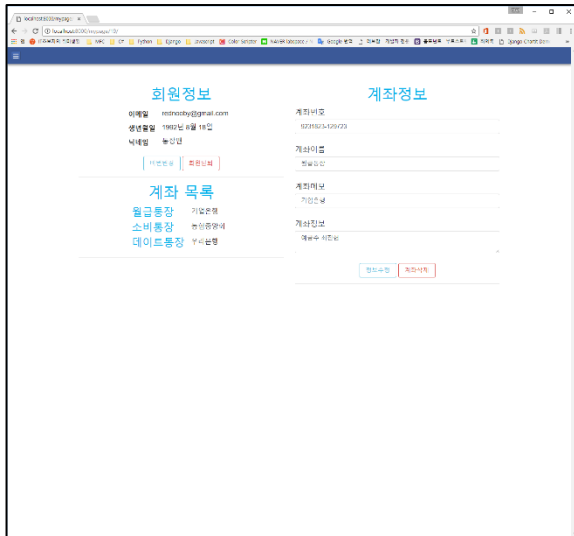
[그림 34] 계좌추가 화면



[그림 35] 계좌추가 완료

[그림 34]는 계좌를 추가하는 화면이다. 사용자가 원하는 데이터를 넣고 추가 버튼을 누르면 [그림 35]와 같이 추가되어 관리할 수 있다.

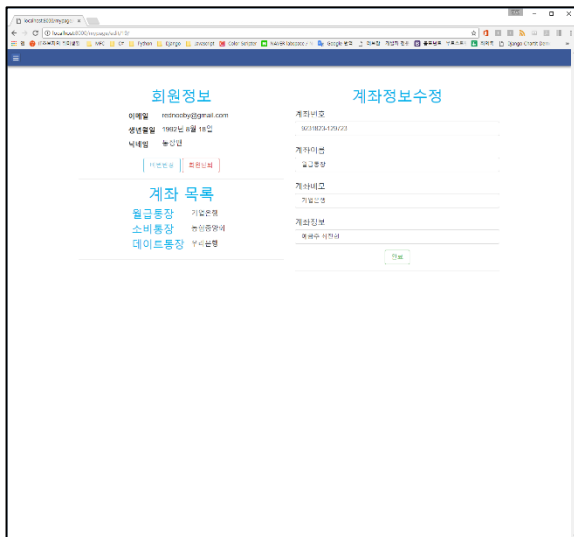
7.4.4 계좌정보확인



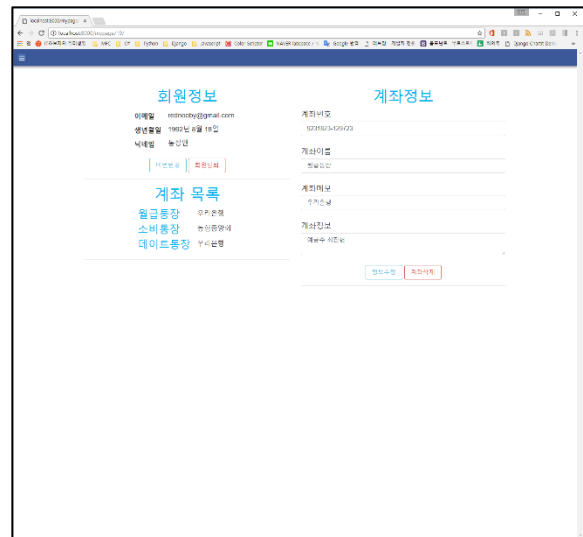
[그림 36] 계좌정보확인 화면

자신이 입력한 계좌의 정보를 확인하고 싶다면 계좌 목록에 있는 계좌를 선택하면 된다. 선택을 하면 [그림 36]과 같이 계좌의 정보가 출력된다.

7.4.5 계좌정보수정



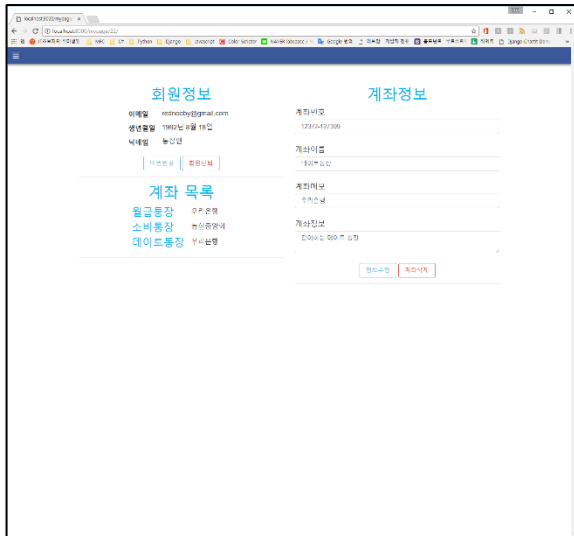
[그림 37] 계좌정보수정 화면



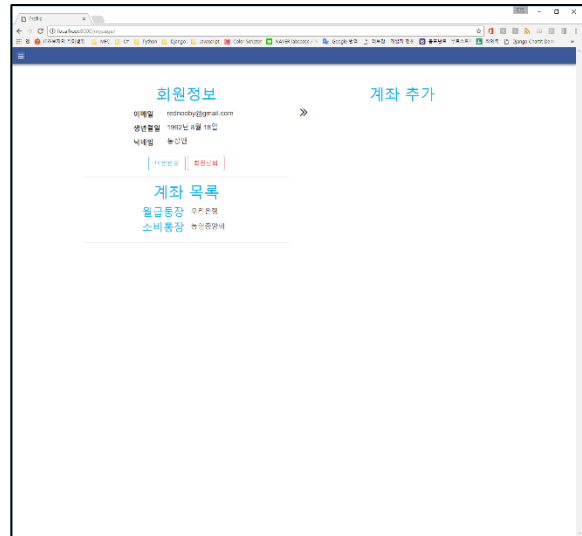
[그림 38] 계좌정보수정 완료

계좌정보 화면에서 정보수정 버튼을 누르면 [그림 37]과 같은 화면이 출력되며 사용자가 수정을 하고 완료버튼을 누르면 [그림 38]과 같이 계좌메모에 기업은행에서 우리은행으로 변경된 것을 확인할 수 있다.

7.4.6 계좌정보삭제



[그림 39] 계좌정보삭제 화면



[그림 40] 계좌정보삭제 완료

사용자가 삭제를 원하면 [그림 39]에 있는 계좌삭제 버튼을 누르면 [그림 40]과 같이 삭제된 것을 확인할 수 있다.

7.3 데이터관리

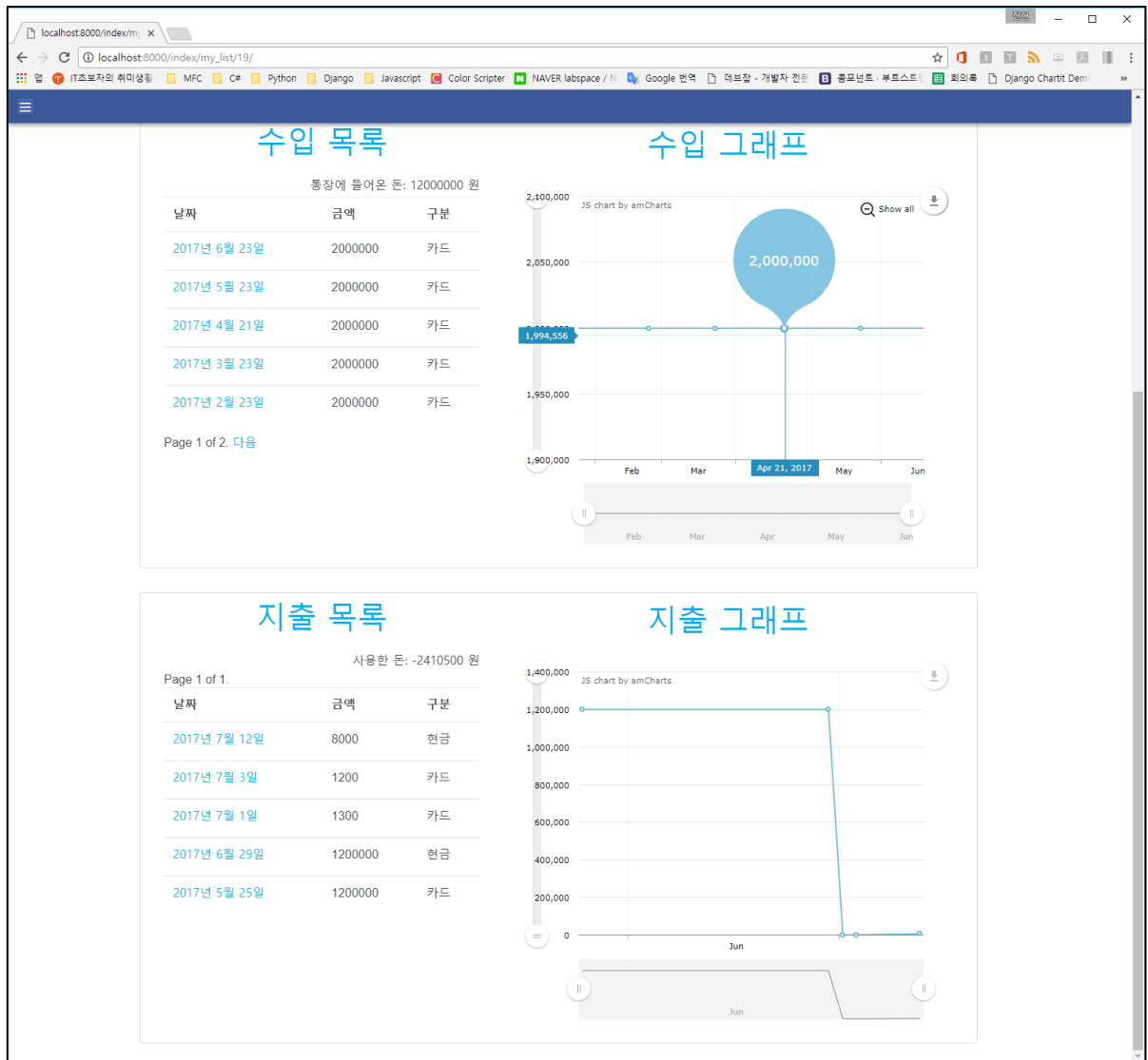


[그림 41] 데이터관리 버튼



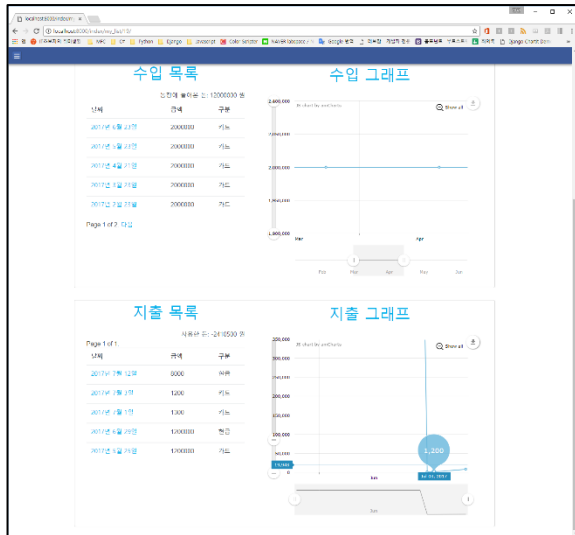
[그림 42] 데이터관리 화면

사용자는 자신의 계좌의 데이터를 관리 할 수 있는데 [그림 41]과 같이 메인화면에서 통장관리 드롭다운 버튼을 누르면 자신의 계좌리스트가 나오며 원하는 계좌를 선택하면 [그림 42]와 같이 전체 계좌 사용내역, 수입별 사용내역&그래프, 지출별 사용내역&그래프 별로 출력된다.

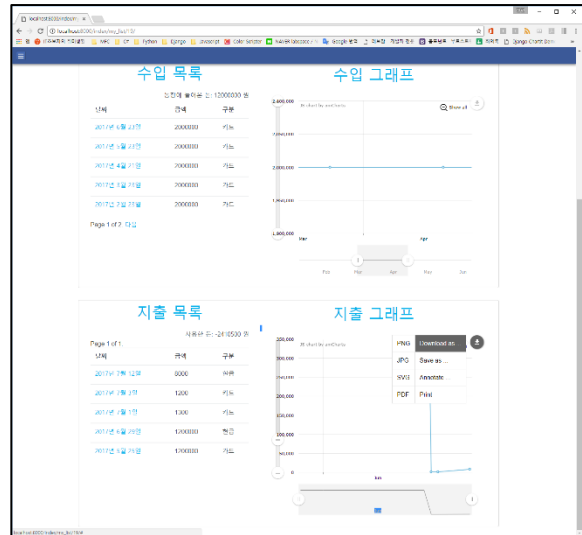


[그림 43] 수입/지출별 그래프

[그림 43]과 같이 사용자는 계좌별로 수입/지출별 목록과 그래프로 확인할 수 있으며 그래프의 축소, 확대 기능을 통하여 아래와 같이 날짜, 금액의 검색범위를 수정하거나 pdf, jpg 등의 파일로 저장할 수 있다. [그림 44][그림 45]



[그림 44] 그래프의 검색범위 변경



[그림 45] 그래프의 파일저장

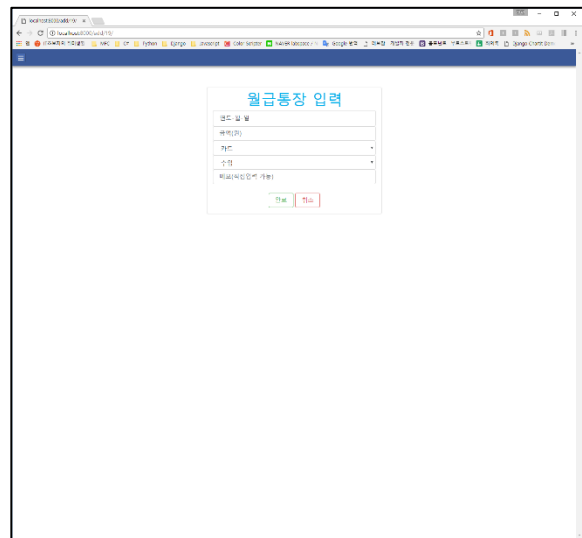
[그림 44]의 수입그래프처럼 사용자는 검색기간 날짜를 마우스 휠로 확대하여 확인하거나, 지출그래프처럼 금액을 확대하여 세밀한 금액구간을 확대할 수 있다.

[그림 45]는 그래프를 파일로 저장할 수 있는 기능이다.

7.3.1 데이터추가



[그림 46] 데이터 추가 선택화면



[그림 47] 데이터 추가화면

사용자가 데이터 추가를 원할 때 [그림 46]과 같이 메뉴를 클릭하고 가계부 입력에 있는 원하는 계좌를 선택한후 [그림 47]의 화면으로 이동하여 데이터를 추가할 수 있다.

7.3.2 데이터 확인



[그림 48] 데이터 선택 화면

[그림 49] 데이터확인 화면

[그림 48]에서 사용자는 확인하고 싶은 데이터를 먼저 선택하여 날짜를 클릭할 수 있으며 클릭하게 되면 해당 날짜에 대한 내역이 [그림 49]와 같이 화면에 출력된다.

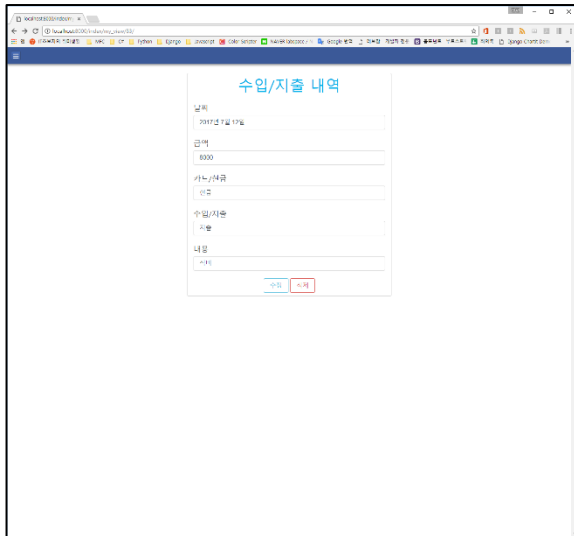
7.3.3 데이터수정

[그림 50] 데이터수정 선택 화면

[그림 51] 데이터수정 화면

사용자가 데이터를 잘못 입력하여 수정하고 싶을 경우 [그림 50]에서 수정 버튼을 눌러 수정페이지로 이동할 수 있으며 [그림 51]의 수정 페이지에서 입력을 완료한 후 완료 버튼을 누르면 수정이 완료된다.

7.3.4 데이터 삭제



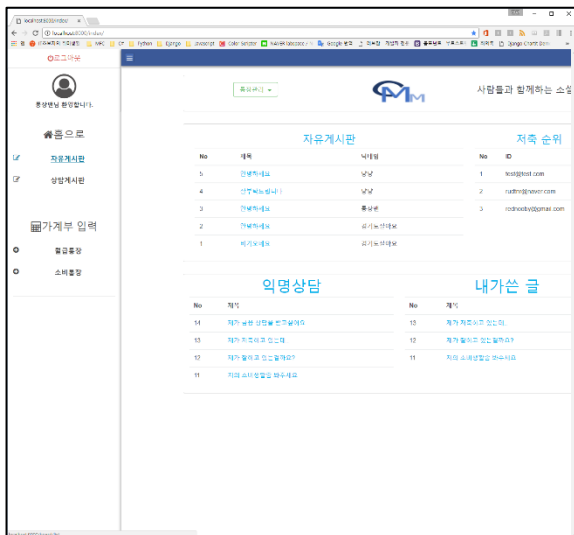
[그림 52] 데이터 삭제 선택 화면



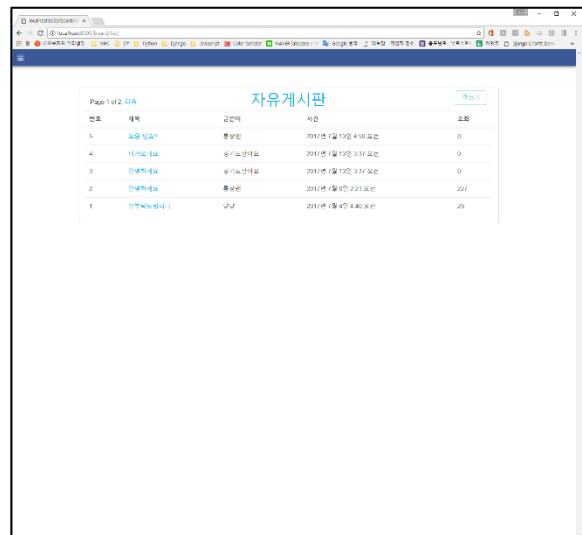
[그림 53] 데이터 삭제 확인

사용자가 데이터의 삭제를 희망할 경우 [그림 52]에서 데이터 삭제를 누르면 삭제가 되며 삭제가 완료 된 것은 [그림 53]에서 확인할 수 있다.

7.4 자유게시판



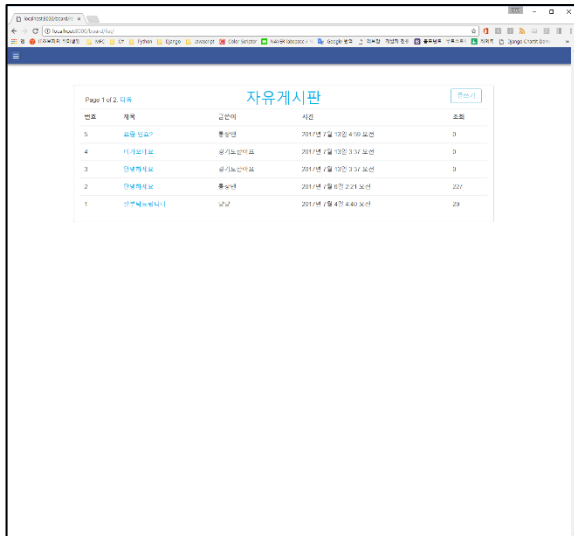
[그림 54] 자유게시판 선택



[그림 55] 자유게시판 화면

사용자가 자유 게시판을 확인하고 싶다면 좌측 상단의 메뉴에서 [그림 54]와 같이 자유게시판을 선택하여 [그림 55]처럼 이동된 것을 확인할 수 있다.

7.4.1 자유게시판 글쓰기



[그림 56] 자유게시판 글쓰기 선택 화면



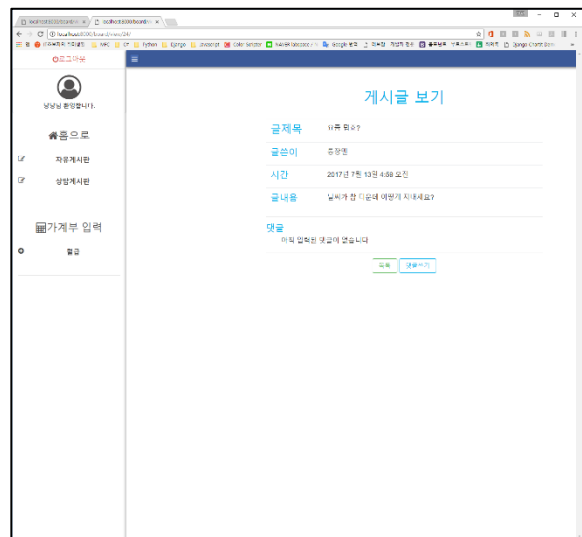
[그림 57] 자유게시판 글쓰기 화면

사용자가 글쓰기를 하고 싶다면 [그림 56]에 있는 글쓰기 버튼을 누르면 [그림 57] 글쓰기 페이지로 넘어가게 되고 원하는 제목과 내용을 입력한 후 완료 버튼을 누르면 글이 저장, 취소 버튼을 누르면 게시글 리스트로 돌아가게 된다.

7.4.2 자유게시판 글보기



[그림 58] 본인글 확인



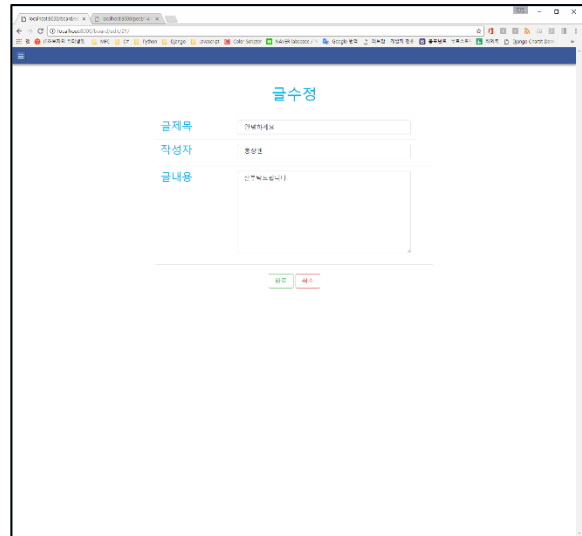
[그림 59] 다른 사용자의 글 확인

자유게시판은 사용자 관계없이 모두 게시글을 확인할 수 있으며 [그림 58]처럼 사용자 본인의 게시글일 경우에만 글수정, 삭제 버튼이 노출되고 다른 사용자의 글은 [그림 59]처럼 노출되지 않는다.

7.4.3 자유게시판 글수정



[그림 60] 본인글 확인



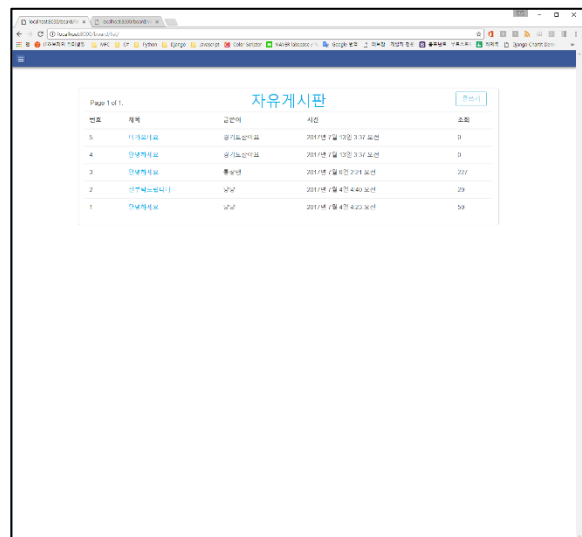
[그림 61] 다른 사용자의 글 확인

자유게시판의 글 수정은 [그림 60]처럼 본인에게만 버튼이 노출되며 버튼을 클릭하면 [그림 61]과 같이 기존의 입력한 데이터를 불러온다. 사용자는 글을 수정하고 완료 버튼을 누르면 수정된 내용이 저장되고 취소버튼을 누르면 게시글로 돌아가게 된다.

7.4.4 자유게시판 글삭제



[그림 62] 본인글 삭제 화면



[그림 63] 삭제 확인

자유게시판의 글 삭제는 [그림 62]처럼 본인의 글만 삭제 할 수 있다. 따라서 삭제 버튼을 누르게 되면 [그림 63]과 같이 게시글이 삭제된다.

7.4.5 자유게시판 댓글보기



[그림 64] 댓글보기 화면



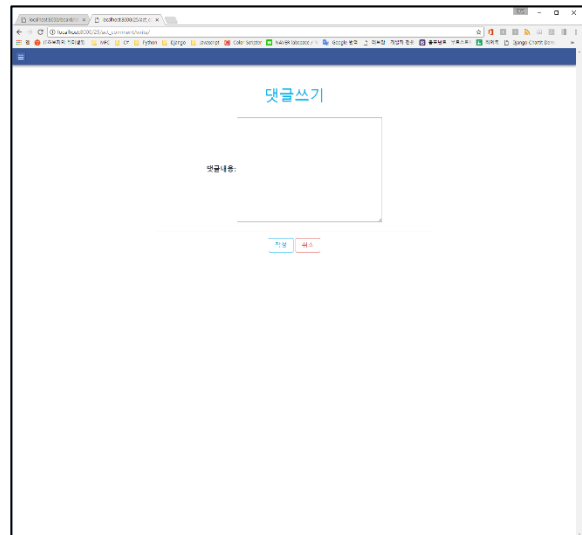
[그림 65] 댓글이 없는 화면

[그림 64]는 게시글 보기에 같이 있는 댓글보기 화면이며 자신의 댓글일 경우에만 수정과 삭제버튼이 노출된다. 만약 게시글에 댓글이 없다면 [그림 65]와 같이 입력된 댓글이 없다고 출력이 된다.

7.4.6 자유게시판 댓글쓰기



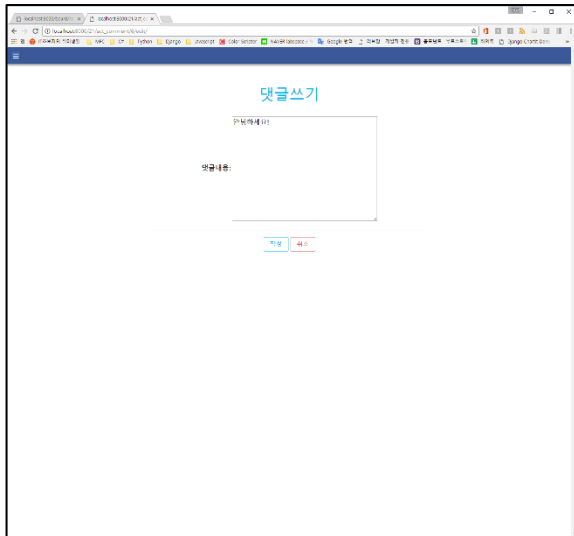
[그림 66] 댓글쓰기 버튼 화면



[그림 67] 댓글쓰기 화면

자유게시판의 댓글은 작성자 상관없이 누구나 사용할 수 있으며 [그림 66]처럼 게시글에 있는 댓글쓰기 버튼을 누르면 [그림 67]의 댓글쓰기 화면으로 넘어간다. 댓글을 입력하고 저장을 누르면 댓글이 저장되고 취소버튼을 누르면 게시글로 돌아간다.

7.4.7 자유게시판 댓글수정



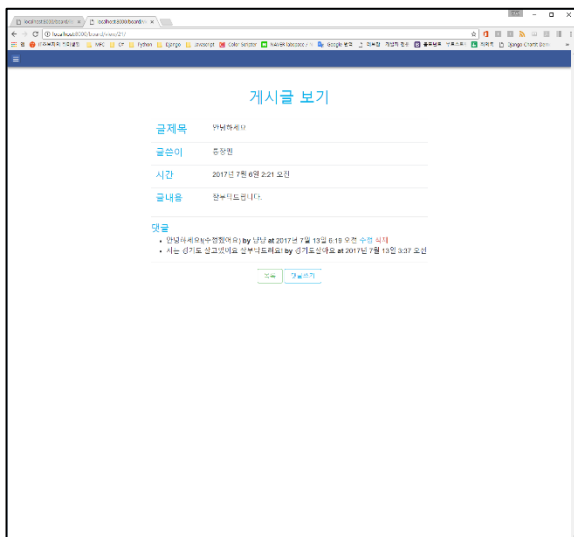
[그림 68] 댓글수정 화면



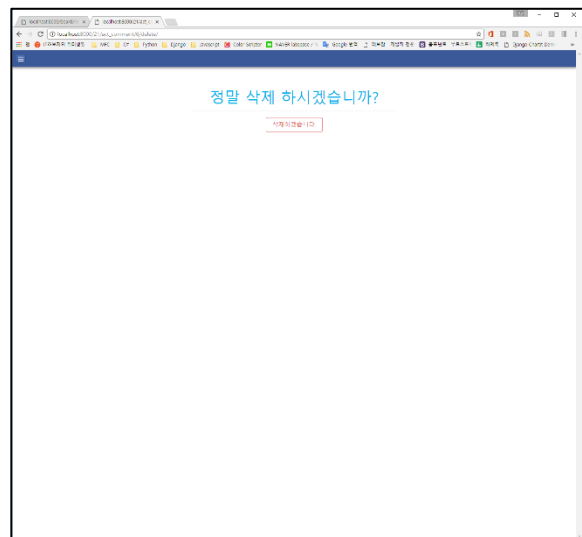
[그림 69] 댓글수정 확인

[그림 68]은 자신의 댓글에 수정버튼을 눌렀을 때 나오는 화면으로 기존의 댓글내용이 출력된다. 수정을 완료하고 완료버튼을 누르면 수정이 완료되지만 취소 버튼을 누르면 해당 게시글로 되돌아가게 된다. 수정된 댓글은 [그림 69]에서 확인할 수 있다.

7.4.8 자유게시판 댓글삭제



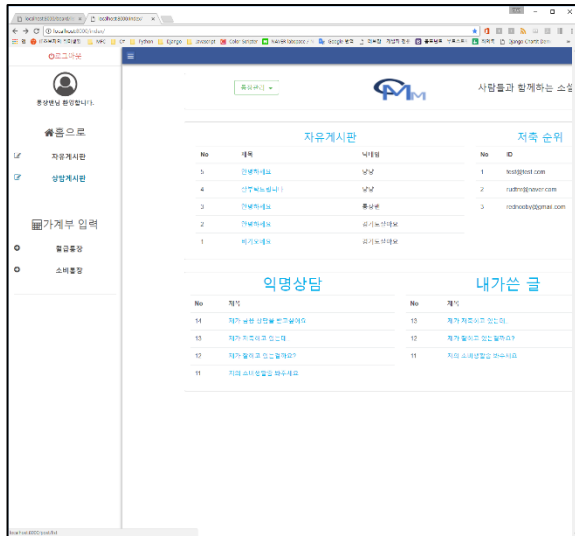
[그림 70] 댓글삭제 버튼 확인



[그림 71] 댓글삭제 확인

[그림 70]에 있는 삭제 버튼을 누르면 [그림 71] 처럼 삭제할 것인지 되묻는 페이지가 나오는데 삭제하겠습니다 버튼을 누르면 댓글이 완전히 삭제된다.

7.5 상담게시판



[그림 72] 상담게시판 선택



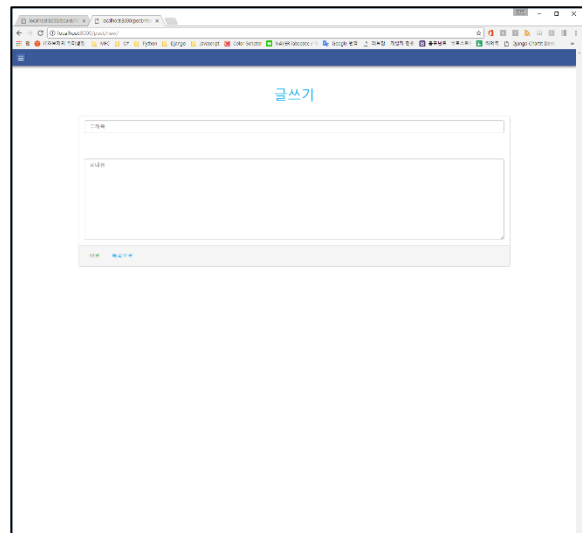
[그림 73] 상담게시판 화면

사용자가 상담게시판으로 가는 방법은 [그림 72]와 같이 메뉴를 통해 가는 방법, 메인화면의 익명상담게시판 게시글 선택, 메인화면의 내가쓴 글 게시글 선택 총 3 가지가 있으며 사용자가 선택하면 [그림 73]과 같은 상담게시판 페이지로 이동한다.

7.5.1 상담게시판 글쓰기



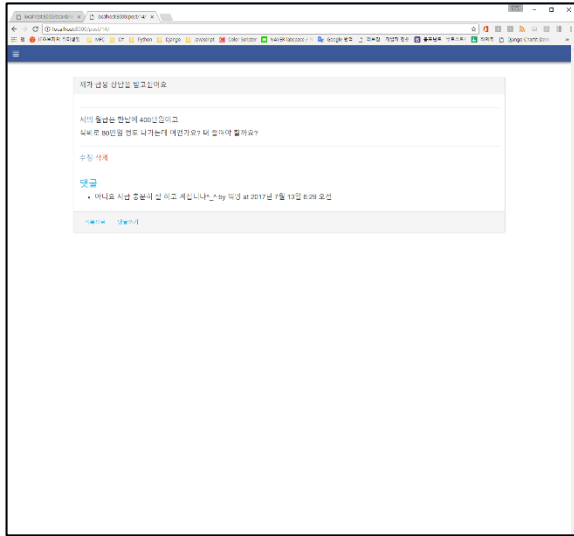
[그림 74] 상담게시판 글쓰기 선택 화면



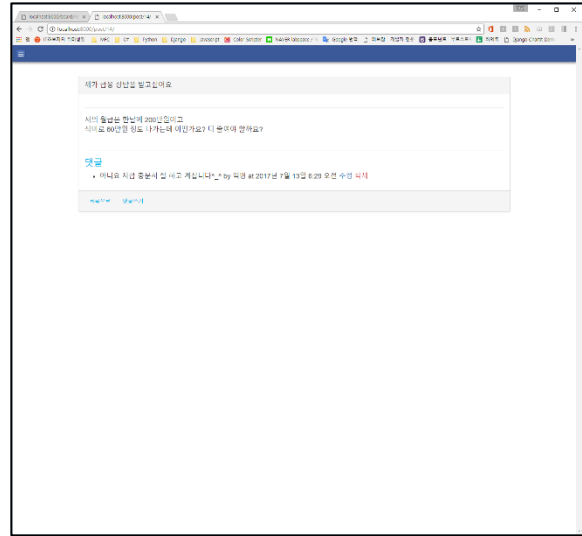
[그림 75] 상담게시판 글쓰기 화면

사용자가 글쓰기를 하고 싶다면 [그림 74]에 있는 글쓰기 버튼을 누르면 [그림 75] 글쓰기 페이지로 넘어가게 되고 원하는 제목과 내용을 입력한 후 완료 버튼을 누르면 글이 저장, 목록으로 버튼을 누르면 게시글 리스트로 돌아가게 된다.

7.5.2 상담게시판 글보기



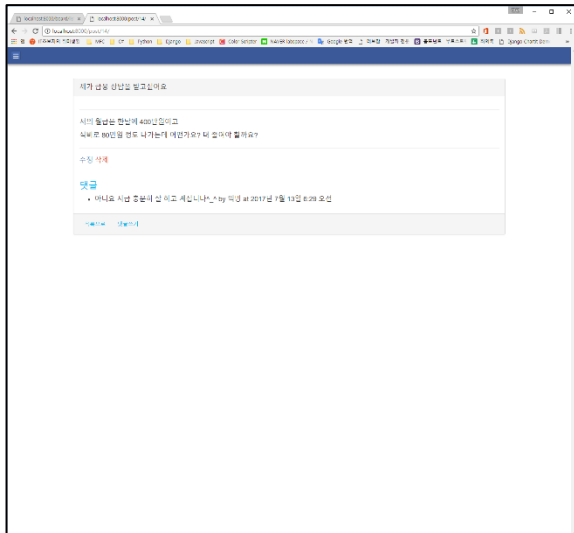
[그림 76] 본인글 확인



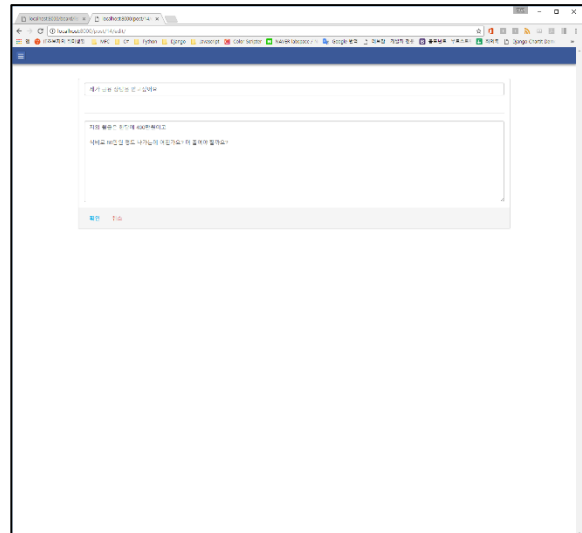
[그림 77] 다른 사용자의 글 확인

상담게시판은 사용자 관계없이 모두 게시글을 확인할 수 있으며 모든 사용자는 익명으로 처리된다. [그림 76]처럼 사용자 본인의 게시글일 경우에만 글수정, 삭제 버튼이 노출되고 다른 사용자의 글은 [그림 77]처럼 노출되지 않는다.

7.5.3 상담게시판 글수정



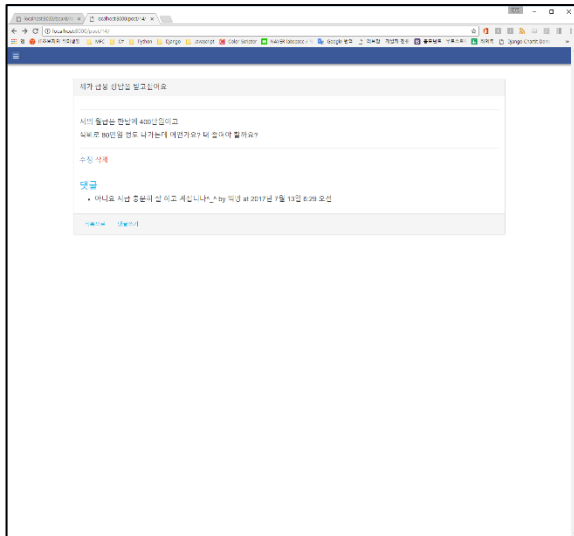
[그림 78] 본인글 수정버튼 확인



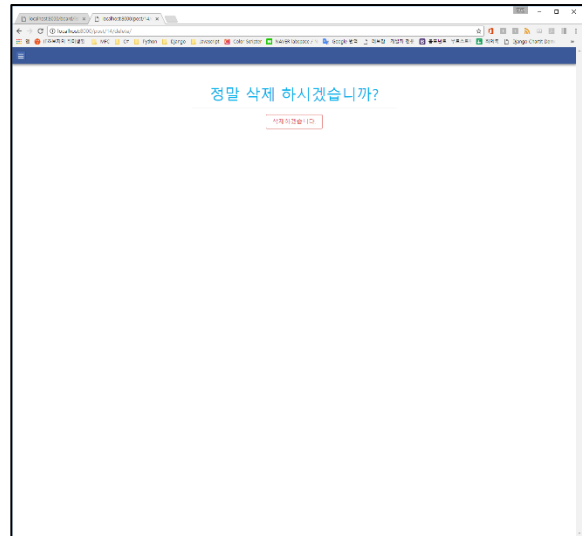
[그림 79] 상담게시판 글수정

상담게시판의 글수정은 글을 쓴 본인만 할 수 있으며 [그림 78]과 같이 버튼이 노출되며 버튼을 누르면 [그림 79]와 같이 수정 페이지로 이동하게 되고 수정을 완료하고 확인 버튼을 누르면 수정완료, 취소 버튼을 누르면 게시글로 돌아가게 된다.

7.5.4 상담게시판 글삭제



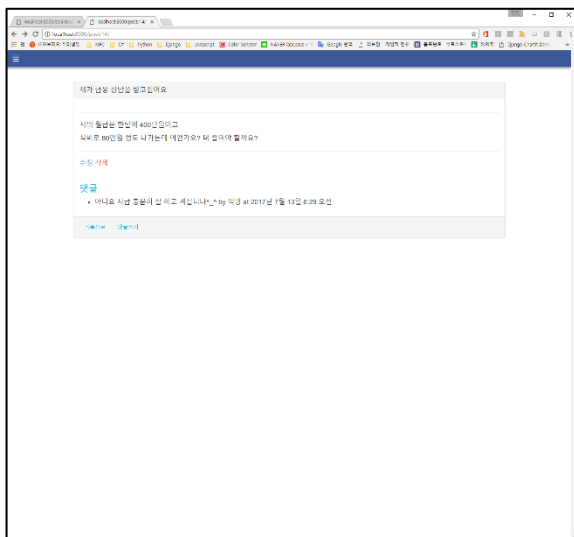
[그림 80] 본인글 삭제버튼 확인



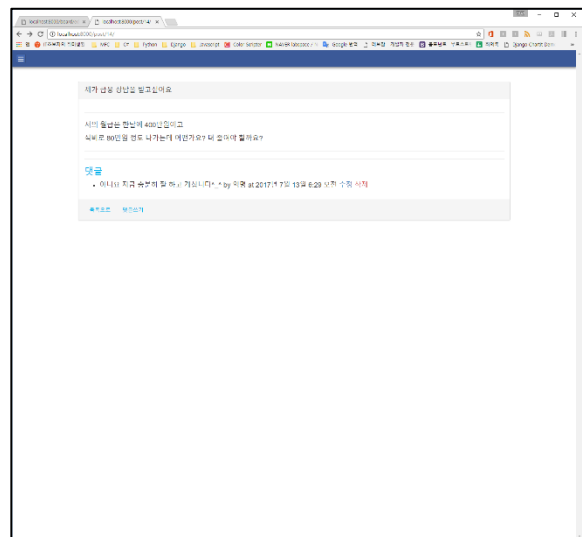
[그림 81] 삭제 확인

자유게시판의 글 삭제는 [그림 80]처럼 본인의 글만 삭제 할 수 있다. 따라서 삭제 버튼을 누르게 되면 [그림 81]와 같이 게시글을 삭제할 것인지 되묻고 삭제하겠습니다 버튼을 누르면 게시글이 완전히 삭제된다.

7.5.5 상담게시판 댓글보기



[그림 82] 다른 사람의 댓글 화면



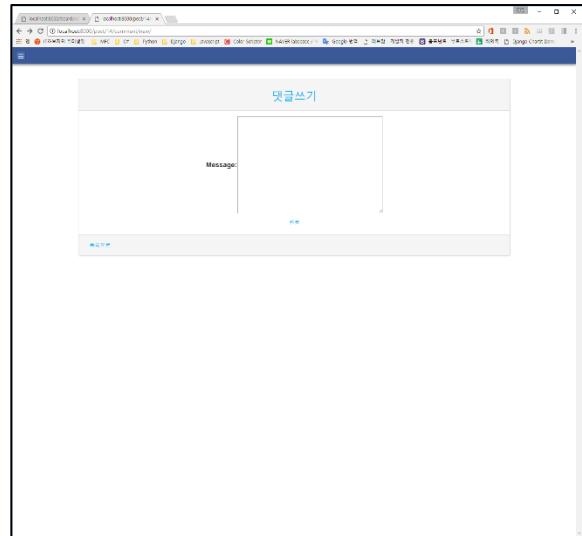
[그림 83] 자신의 댓글 화면

상담게시판의 댓글 글쓴이는 익명으로 저장되며, [그림 82]와 같이 다른 사람의 댓글은 수정과 삭제버튼이 보이지 않지만 [그림 83]과 같이 자신의 댓글은 수정과 삭제 버튼이 보인다.

7.5.6 자유게시판 댓글쓰기



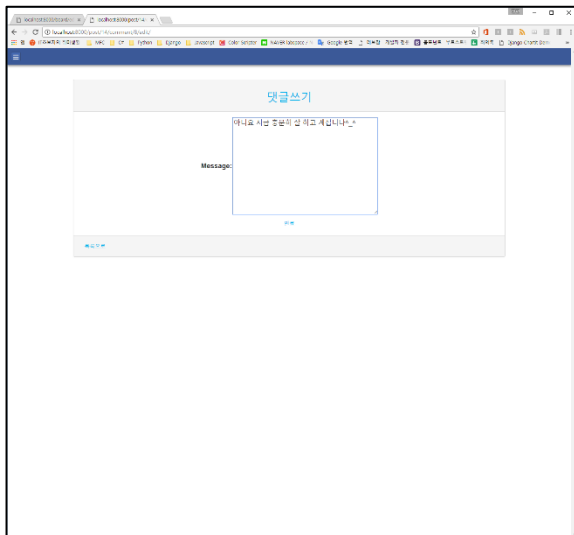
[그림 84] 댓글쓰기 버튼 화면



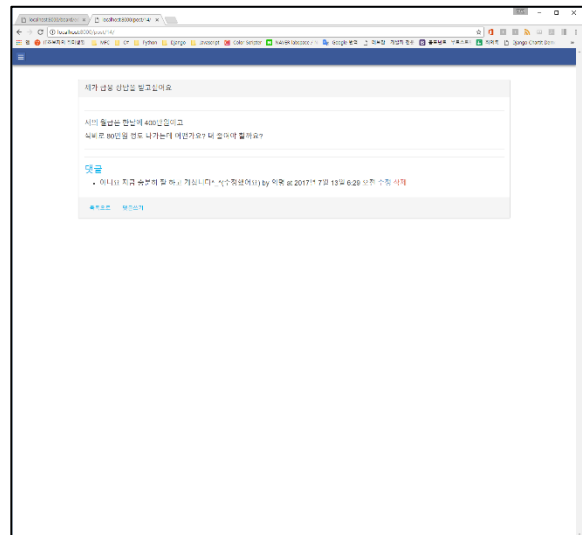
[그림 85] 댓글쓰기 화면

자유게시판의 댓글은 작성자 상관없이 누구나 사용할 수 있으며 [그림 84]처럼 게시글에 있는 댓글쓰기 버튼을 누르면 [그림 85]의 댓글쓰기 화면으로 넘어간다. 댓글을 입력하고 저장을 누르면 댓글이 저장되고 목록으로 버튼을 누르면 게시글로 돌아간다.

7.5.7 상담게시판 댓글수정



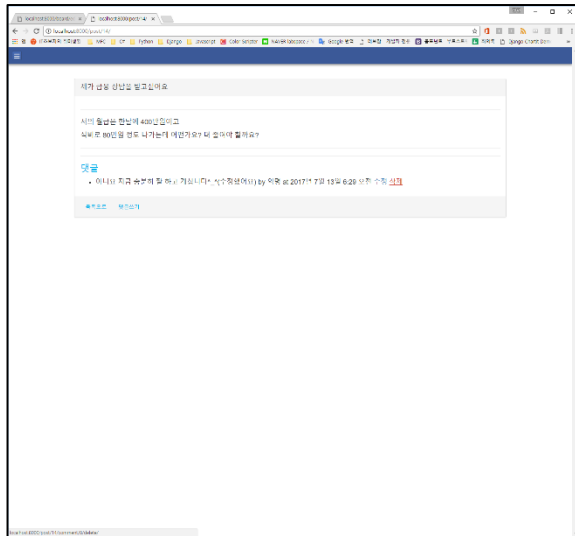
[그림 86] 댓글수정 화면



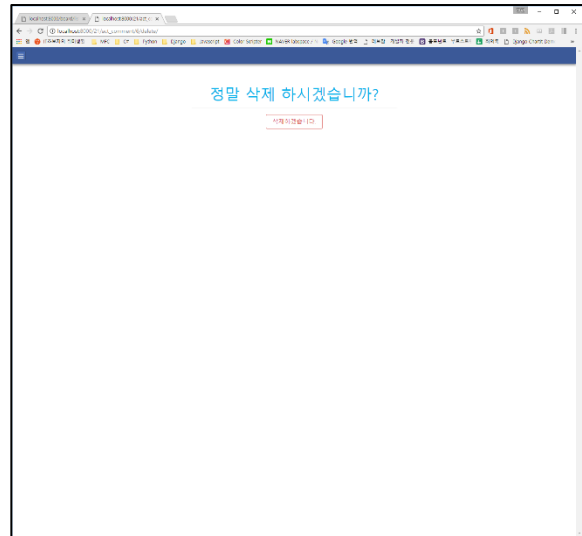
[그림 87] 댓글수정 확인

[그림 86]은 자신의 댓글에 수정버튼을 눌렀을 때 나오는 화면으로 기존의 댓글내용이 출력된다. 수정을 완료하고 완료버튼을 누르면 수정이 완료되지만 목록으로 버튼을 누르면 목록으로 되돌아가게 된다. 수정된 댓글은 [그림 87]에서 확인할 수 있다.

7.5.8 상담게시판 댓글삭제



[그림 88] 댓글삭제 버튼 확인



[그림 89] 댓글삭제 확인

[그림 88]에 있는 삭제 버튼을 누르면 [그림 89] 처럼 삭제할 것인지 되묻는 페이지가 나오는데 삭제하겠습니다 버튼을 누르면 댓글이 완전히 삭제된다.