



EzoBank

유승국

목차



01

EzoBank 소개

02

EzoBank 기획의도

03

EzoBank 개발 환경

04

EzoBank 기능 소개

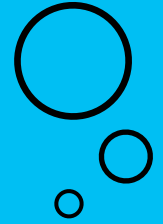
05

EzoBank 보완할 점





EzoBank 소개



EzoBank 소개



계좌 생성, 입출금, 이체, 공지사항, 게시판 등의 기능을 구현한 은행 웹 사이트로, 실제 은행 사이트에서는 입출금 기능이 없지만 **능력 향상에** 의의를 두고 은행에서 할 수 있는 **최소한의 기능들을** 구현하였습니다.





EzoBank 기획의도

EzoBank 기획의도

개발 목적

JAVA, JSP, Spring을 수강하면서 배웠던 기능들을 다시 구현해 보고 각자의 부족한 부분을 알고, 오류, 정보공유, 의견 등을 통해 팀 협업 능력을 성장시키는데, 목표를 두고 있습니다.

담당 업무

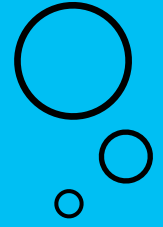
로그인, 회원가입,
회원 정보 수정,
메인, CSS, 풋터

개발 기간

2020.12.21 ~
2021.01.08



EzoBank 개발환경



EzoBank 개발환경

개발환경

FRONT-END

- HTML5
- CSS
- JAVASCRIPT

BACK-END

- JAVA
- JSP/SERVLET

FRAMEWORK

- SPRING MVC

DBMS

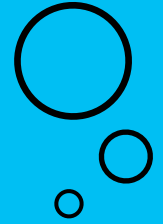
- ORACLE

WEB-SERVER

- APACHE-TOMCAT 8.5 VERSION



EzoBank 기능 설명



TABLE

01 회원가입을 위한 TABLE

```
create table bank(  
  idx NUMBER(15) primary key, -- 일련번호  
  name VARCHAR2(100) not null, -- 사용자  
  ssn VARCHAR2(15) not null unique, -- 주민번호  
  tel VARCHAR2(50) not null, -- 전화번호  
  email VARCHAR2(100) not null, -- 이메일  
  addr VARCHAR2(200), -- 주소  
  id VARCHAR2(100) not null unique, -- 아이디  
  pwd VARCHAR2(100) not null, --비밀번호  
  account VARCHAR2(150) not null unique, -- 계좌번호  
  balance NUMBER(38) default 0, -- 잔액  
  ip VARCHAR2(50) not null, -- ip  
  regdate DATE -- 날짜
```

02 거래내역을 위한 TABLE

```
create table detail(  
  account VARCHAR2(150),  
  content VARCHAR2(150),  
  deposit NUMBER(38),  
  withdraw NUMBER(38),  
  balance NUMBER(38) default 0,  
  usedate DATE,  
  det Number(5)  
);  
create table det(  
  det Number(5) unique,  
  detname VARCHAR2(50)
```

TABLE

03 Q&A TABLE

```
create table board(  
  idx NUMBER(3) primary key, --게시글 일련번호  
  name VARCHAR2(100) not null, --작성자  
  subject VARCHAR2(200) not null, --제목  
  content CLOB, --내용  
  pwd VARCHAR2(50) not null, --비밀번호  
  ip VARCHAR2(50), --ip  
  regdate DATE, --작성일  
  readhit NUMBER(5) default 0, --조회수  
  --어떤 게시판이든 무조건 필요한 column  
  ref NUMBER(3), --기준글 번호  
  step NUMBER(3), --댓글순서(수직적)  
  depth NUMBER(3), --댓글의깊이(수평적)  
  del_info NUMBER(1) default 0
```

04 공지사항 TABLE

```
create table notice(  
  idx NUMBER(3) primary key, --게시글 일련번호  
  subject VARCHAR2(200) not null, --제목  
  content CLOB, --내용  
  ip VARCHAR2(50), --ip  
  regdate DATE, --작성일  
  readhit NUMBER(5) default 0, --조회수  
  del_info NUMBER(1) default 0  
);
```

메인

소상공인 금융지원 상품 안내

코로나19 피해를 입은 소상공인 여러분 응원합니다.



출금



이체



잔액조회



EzoBank



계좌생성



공지사항(이벤트)



Q&A

각 이미지, 롤링 배너 클릭 시 해당 페이지로 이동
(Session에 값이 없다면 로그인 페이지로 이동)

```
//슬라이드
var slideIndex = 1;

var i;

var slides = document.getElementsByClassName("mySlides");
var dots = document.getElementsByClassName("dot");

function plusSlides() {
    slideIndex++;

    if (slideIndex > slides.length) {slideIndex = 1}

    for (i = 0; i < slides.length; i++) {
        slides[i].style.display = "none";
    }

    for (i = 0; i < dots.length; i++) {
        dots[i].className = dots[i].className.replace(" active", "");
    }

    //도트무늬 클릭
    function currentSlide(n) {

        slideIndex = n;

        if (slideIndex > slides.length) {slideIndex = 1}

        if (slideIndex < 1) {slideIndex = slides.length}

        for (i = 0; i < slides.length; i++) {
            slides[i].style.display = "none";
        }
        for (i = 0; i < dots.length; i++) {
            dots[i].className = dots[i].className.replace(" active", "");
        }
        slides[slideIndex-1].style.display = "block";
        dots[slideIndex-1].className += " active";
    }

    setInterval("plusSlides()", 2000);
}
```

EzoBank 회원가입

* Please enter your account information.

아이디 :

비밀번호 :

비밀번호 확인 :

이름 :

주민번호 : -

전화번호 : () - -

이메일 : @

우편번호

주소 :

상세 주소

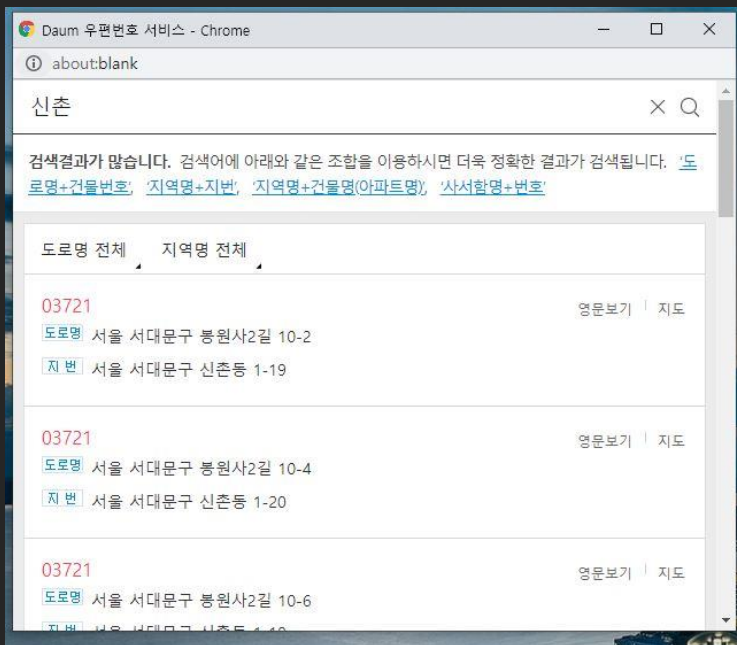
[홈페이지 회원가입 서비스 이용약관, 제1장 총칙, 제1조(목적), 이 약관은 회원이 ㈜Ezo_Bank(이하 "회사")에서 제공하는 인터넷 관련 서비스(이하 "서비스")를 이용함에 있어 회원과 회사간의 권리, 의무 및 책임사항, 이용조건 및 절차 등, 기본적인 사항을 규정함을 목적으로 합니다., 제2조(정의), ①이 약관에서 사용하는 용어의 정의는 다음과 같습니다., 1. 이용자 : 사이트에 접속하여 이 약관에 따라 서비스 주체가 제공하는 서비스를 받는 회원 및 비회원, 2. 홈페이지 회원 : 사이트에 접속하여 이 약관에 동의하고, ID(고유번호)와, Password(비밀번호)를 발급 받은 고객(이하 "회원"), 3. 비회원 : 회원에 가입하지 않고 회사가 제공하는 서비스를 이용하는 고객, 4.

■동의

가입

취소

회원가입



```
//회원 가입
@RequestMapping("/insert.do")
@ResponseBody
public String member_insert(BankVO vo, String ssn, String tel, String email) {

    vo.setSsn(ssn);
    vo.setTel(tel);
    vo.setEmail(email);

    //vo에 ip 등록
    String ip = request.getRemoteAddr();
    vo.setIp(ip);

    //vo에 계좌 생성 및 등록
    String str = String.valueOf(System.currentTimeMillis());

    String account1 = str.substring(0,3);
    String account2 = str.substring(4,6);
    String account3 = str.substring(7,12);

    String account = account1 + "-" + account2 + "-" + account3;
    vo.setAccount(account);

    int res = bank_dao.insert(vo);

    String result = "no";

    function DaumPostcode_API() {
        new daum.Postcode({
            oncomplete: function(data) {

                var extraRoadAddr = '';

                if(data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                    extraRoadAddr += data.bname;
                }

                if(data.buildingName !== '' && data.apartment === 'Y'){
                    extraRoadAddr += (extraRoadAddr !== '' ? ', ' + data.buildingName : data.buildingName);
                }

                if(extraRoadAddr !== ''){
                    extraRoadAddr = '(' + extraRoadAddr + ')';
                }

                document.getElementById('postcode').value = data.zonecode;
                document.getElementById("roadAddr").value = data.roadAddress + extraRoadAddr;
                document.getElementById("jibunAddr").value = data.jibunAddress + extraRoadAddr;

                if(roadAddr !== ''){
                    document.getElementById("extraAddr").value = extraRoadAddr;
                } else {
                    document.getElementById("extraAddr").value = '';
                }
            }
        }).open();
    }
}
```

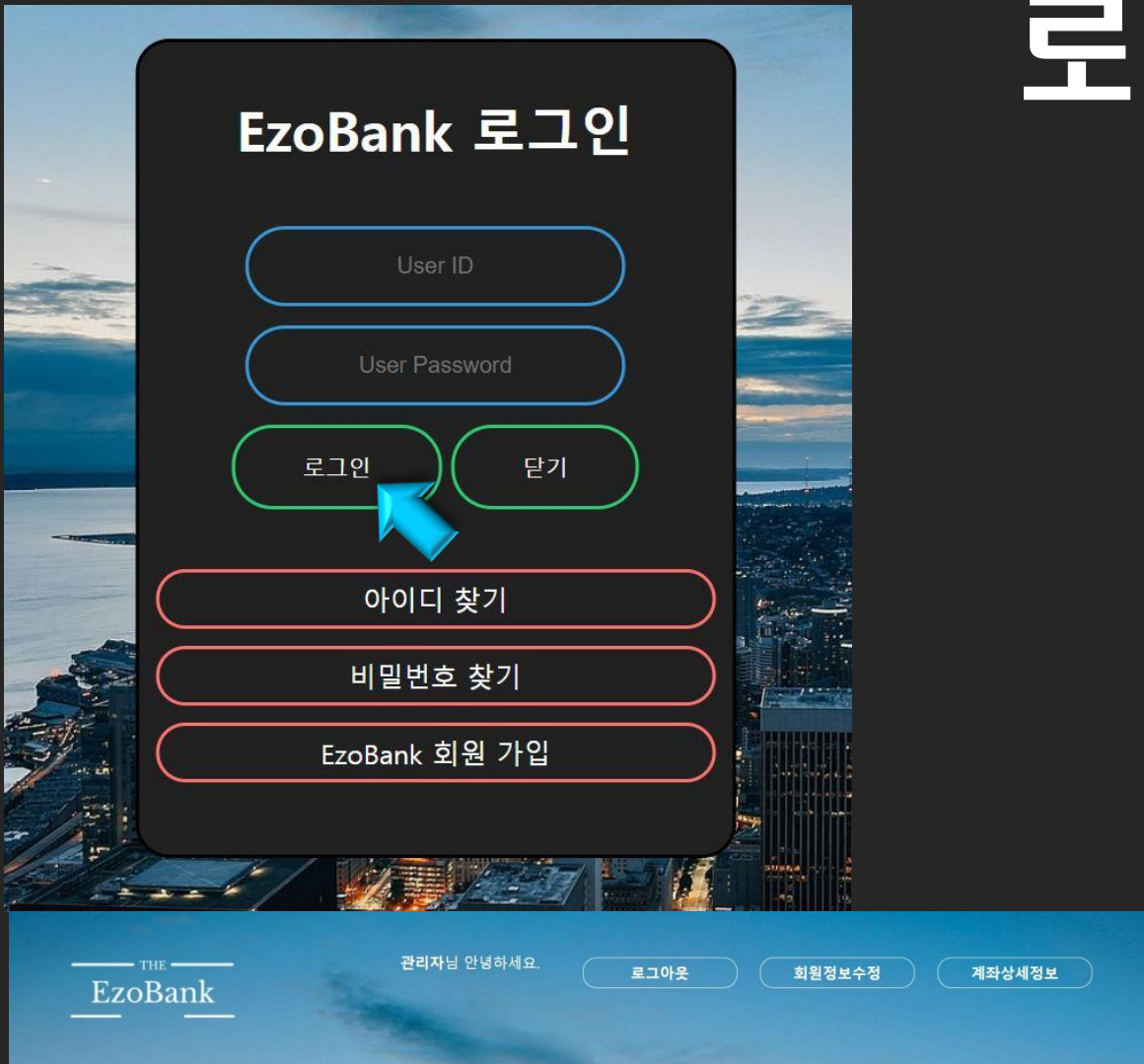
아이디 중복 버튼 -> 아이디 중복 확인을 할 수 있는 버튼

password 버튼 -> password type을 text type으로 변경하여 비밀번호를 볼 수 있는 버튼

주민번호 버튼 -> DB에 해당하는 주민번호가 있다면 중복 가입 불가

우편 번호 검색 버튼 -> 다음 우편번호 API를 사용하여 정확한 주소 검색 가능

로그인



The image shows a login form for EzoBank. The form is titled "EzoBank 로그인" and is set against a background of a city skyline at night. The form contains several input fields and buttons. A blue arrow points to the "로그인" button.

EzoBank 로그인

User ID

User Password

로그인 닫기

아이디 찾기

비밀번호 찾기

EzoBank 회원 가입

THE EzoBank

관리자님 안녕하세요.

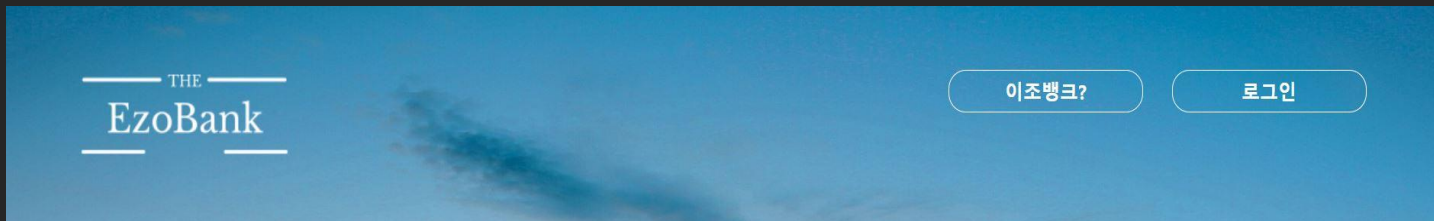
로그아웃 회원정보수정 계좌상세정보

```
//로그인 하기!  
@RequestMapping("/sign_up.do")  
@ResponseBody  
public String sign_up(Model model, BankVO vo) {  
  
    String param = "";  
    String res = "";  
  
    BankVO bankvo = bank_dao.login(vo);  
  
    //DB검색 결과 해당 ID가 없는 경우  
    if(bankvo == null){  
        param = "no_id";  
        res = String.format("[{'param':'%s'}]",param);  
        return res;  
    }  
    //DB검색 결과 해당 ID는 있지만 PWD가 없는 경우  
    if(!bankvo.getPwd().equals(vo.getPwd())) {  
        param = "no_pwd";  
        res = String.format("[{'param':'%s'}]",param);  
        return res;  
    }  
  
    //DB검색 결과 해당 ID와 PWD가 존재하는 경우  
    //1시간 동안 세션 저장!(로그인 상태 유지..)  
    session = request.getSession();  
    session.setAttribute("user",bankvo);  
    session.setMaxInactiveInterval(60*60);  
  
    param = "clear";  
    res = String.format("[{'param':'%s'}]",param);  
    return res;  
}
```

로그인 -> 로그인시 로그인 정보가 Session에 60분간 기록



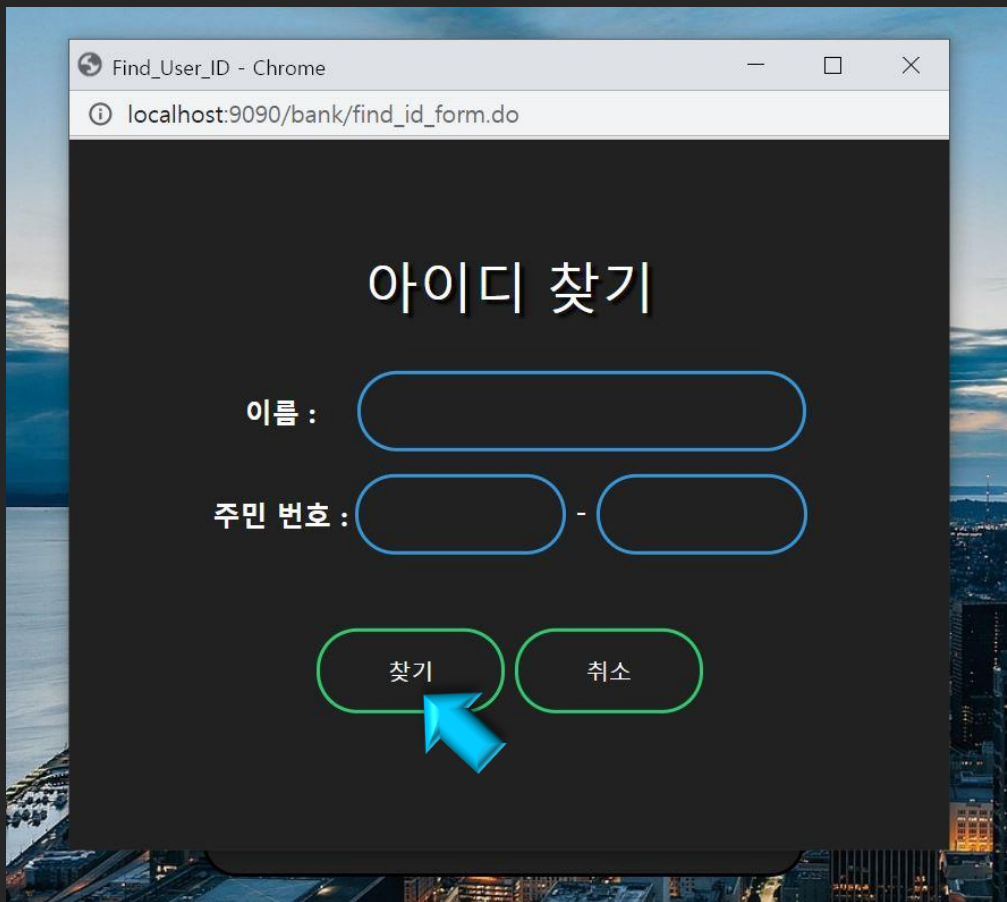
로그아웃



```
//로그아웃
@RequestMapping("/logout.do")
public String logout() {
    //저장된 세션 지우기
    session.removeAttribute("user");
    return "redirect:login.do";
}
```



로그인-아이디 찾기



Find_User_ID - Chrome

localhost:9090/bank/find_id_form.do

아이디 찾기

이름 :

주민 번호 : -

```
//아이디 찾기 폼으로 이동
@RequestMapping("/find_id_form.do")
public String find_form_id() {

    return WEB_PATH + "bank_find_id_form.jsp";
}

//아이디 찾기
@RequestMapping("/find_id.do")
@ResponseBody
public String find_id(BankVO vo, String name, String ssn) {

    vo.setName(name);
    vo.setSsn(ssn);

    BankVO bankvo = bank_dao.find_id_selectOne(vo);

    String result = "no";

    if(bankvo != null) {
        result = bankvo.getId();
    }

    return result;
}
```

이름과 주민번호를 통해 아이디 찾기 가능



로그인-비밀번호 찾기

Find_User_Password - Chrome
localhost:9090/bank/find_pwd_form.do

비밀번호 찾기

이름 :

아이디 :

주민 번호 : -

Find_User_Password - Chrome
localhost:9090/bank/find_pwd_form.do

localhost:9090 내용:
관리자님, 안녕하세요.
새로운 비밀번호를 설정해주세요.

아이디 :

주민 번호 : -

```
//비밀번호 찾기 폼으로 이동
@RequestMapping("/find_pwd_form.do")
public String find_form_pwd() {
    return WEB_PATH + "bank_find_pwd_form.jsp";
}

//비밀번호 찾기
@RequestMapping("/find_pwd.do")
@ResponseBody
public String find_pwd(Model model, BankVO vo, String name, String id, String ssn) {

    vo.setName(name);
    vo.setId(id);
    vo.setSsn(ssn);

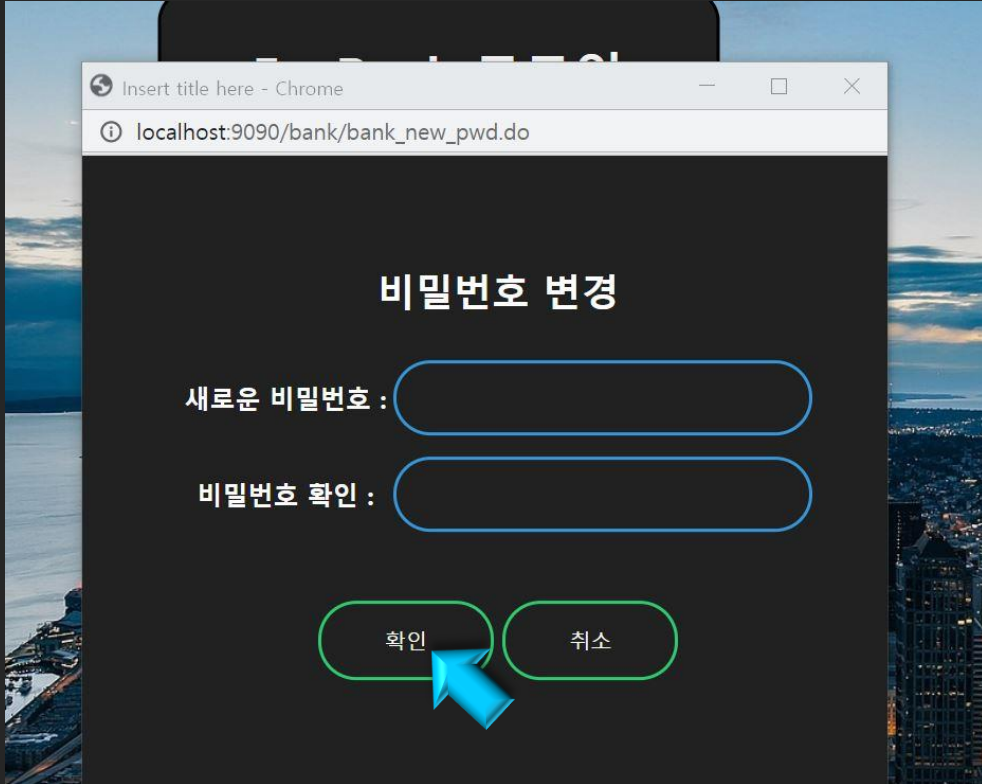
    BankVO bankvo = bank_dao.find_pwd_selectOne(vo);

    String result = "no";

    //DB검색 결과 해당 정보에 대한 조회가 없다면
    if(bankvo != null) {
        result = "yes";
    }
    return result;
}
```



로그인-비밀번호 찾기



Insert title here - Chrome

localhost:9090/bank/bank_new_pwd.do

비밀번호 변경

새로운 비밀번호 :

비밀번호 확인 :

```
// 새로운 비밀번호 설정
@RequestMapping("/change_password.do")
@ResponseBody
public String change_password(BankVO vo, String ssn, String pwd) {

    vo.setSsn(ssn);
    vo.setPwd(pwd);

    int res = bank_dao.password_update(vo);

    String result = "no";

    if(res > 0) {
        result = "yes";
    }
    return result;
}
```

입력한 이름과 아이디, 주민번호가 DB에 존재한다면, 새로운 비밀번호 입력/ 변경



회원정보수정

회원 정보 수정

아이디 :

비밀번호 :

비밀번호 확인 :

이름 :

주민번호 : -

계좌번호 : - -

전화번호 : () - -

이메일 : @

Address :

```
//회원 정보 수정
@RequestMapping("/user_modified.do")
@ResponseBody
public String user_modified(BankVO vo, String ssn, String account, String tel, String email) {

    //수정 폼에 입력된 값들을 해당하는 유저 DB를 수정한다.
    vo.setSsn(ssn);
    vo.setAccount(account);
    vo.setTel(tel);
    vo.setEmail(email);

    int res = bank_dao.user_modified(vo);

    String result = "no";

    if(res > 0) {
        result = "yes";
    }

    return result;
}
```

아이디, 주민번호, 계좌번호를 제외한 모든 정보 수정 가능
(비밀번호를 제외한 모든 정보 자동으로 입력)

비밀번호가 틀리면, 회원수정 불가



입금

:::입금:::

hye님 계좌에 입금하실 금액을 입력해주세요

1	2	3
4	5	6
7	8	9
C	0	del

사용자의 idx 입금 금액 입금:1, 출금:2,
이체:3 전달

```
@RequestMapping("/depositandwithdraw.do")
@ResponseBody
public String depositandwithdraw(int idx,int amount,int check) {
    int res = 0;
    String result = "";

    if(check==1) {
        //입금 mapper
        res = bdao.update_deposit(idx, amount);
        res = adao.account_insert_deposit(idx, amount, check);
    }
    else{
        //출금 mapper
        res = bdao.update_withdrawal(idx, amount);
        res = adao.account_insert_withdrawal(idx, amount, check);
    }

    if(res>0) {
        result = "success";
    }

    return result;
}
```

Bank table

거래내역 table

입금할 금액을 입력하면 해당 금액이 DB에 입금되어 저장!



출금

...출금...

hye님 계좌에서 출금하실 금액을 입력해주세요

1	2	3
4	5	6
7	8	9
C	0	del

출금

나가기

```
@RequestMapping("/depositandwithdraw.do")
@ResponseBody
public String depositandwithdraw(int idx,int amount,int check) {
    int res = 0;
    String result = "";

    if(check==1) {
        //입금 mapper
        res = bdao.update_deposit(idx, amount);
        res = adao.account_insert_deposit(idx, amount, check);
    }
    else{
        //출금 mapper
        res = bdao.update_withdrawal(idx, amount);
        res = adao.account_insert_withdrawal(idx, amount, check);
    }

    if(res>0) {
        result = "success";
    }

    return result;
}
```

사용자의 idx
입금 금액
입금:1, 출금:2,
이체:3 전달

Bank table

거래내역 table

출금할 금액을 입력하면 해당 금액이 DB에서 출금되어 저장!



이체-계좌번호

:::이체:::

hye님 이체하실 계좌번호를 입력해 주세요

1	2	3
4	5	6
7	8	9
C	0	del

'-'를 붙이지 말고 숫자만 입력해주세요

확인

나가기

```
@RequestMapping("/transfer_page.do")
public String transfer_page(int idx, Model model) {
    BankVO vo = bdao.selectone(idx);

    //전체 회원 list 가져오기
    List<BankVO> list = bdao.select();

    //전체 회원의 계좌를 저장하기 위한 list
    List<String> account = new ArrayList<String>();

    //전체 회원의 계좌 저장하기
    for(int i=0;i<list.size();i++) {
        account.add(list.get(i).getAccount());
    }

    model.addAttribute("vo", vo);
}

@RequestMapping("/transfer_account.do")
@ResponseBody
public String transfer_account(int idx,String acc) {
    String result = "";

    BankVO vo = bdao.selectone(acc); //입력받은 계좌 vo받기

    if(vo==null) { // 입력한 계좌가 존재하지 않다면 null을 return
        return null;
    }

    int idxYou = vo.getIdx();
    String name = vo.getName();
    //이체를 위한 return
    result = String.format("[{'idxMe': '%d', 'idxYou': '%d', 'name': '%s'}]", idx, idxYou, name);

    return result;
}
```

이체하고자 하는 계좌번호
존재여부

이체하는 계좌번호 idx를
넘겨주는 Ajax

이체할 계좌번호 입력하고 DB에 입력한 데이터에 해당하는 계좌가 있다면, 금액 입력 페이지로 변환



이체-금액

:::이체:::

hye님 rim께 이체하실 금액을 입력해주세요

1	2	3
4	5	6
7	8	9
C	0	del

확인

뒤로가기

```
//이체하기를 완료하기 위한 idxMe : 내 idx , idxYou : 이체하기 위한 상대방 idx
@RequestMapping("/transfer_money_page.do")
public String transfer_money_page(Model model, int idxMe, int idxYou) {
    BankVO voMe = bdao.selectone(idxMe);
    model.addAttribute("voMe",voMe);

    BankVO voYou = bdao.selectone(idxYou);
    model.addAttribute("voYou",voYou);

    return MyCommon.Transfer.VIEW_PATH + "transfer_money_page.jsp";
}

//이체 페이지의 Ajax
@RequestMapping("/transfer_amount.do")
@ResponseBody
public String transfer_amount(int idxMe, int idxYou, int amount) {
    int res = 0;

    res = bdao.update_deposit(idxYou, amount); // 상대방 계좌 입금
    res = bdao.update_withdrawal(idxMe, amount); // 내 계좌 출금
    res = adao.account_insert_transfer(idxMe, idxYou, amount);

    String result = "";

    if(res>0) {
        result = "success";
    }

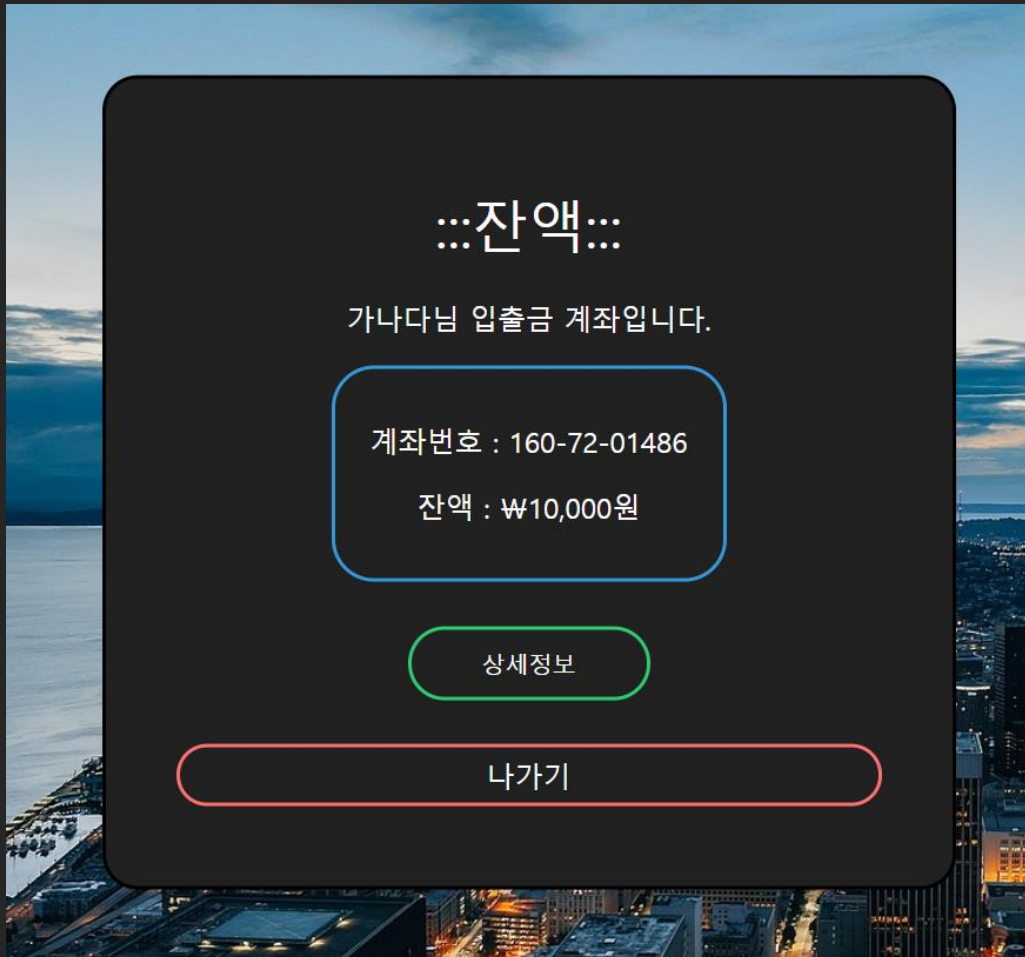
    return result;
}
```

이체 성공여부

이체할 금액을 입력하면 상대방 DB에 기본 금액에서 더해짐



잔액조회



```
@RequestMapping("/aac_view.do")
public String bank_view(int idx, Model model) {

    BankVO vo = bdao.all_selectOne(idx);

    if (vo != null) {
        //idx에 해당되는 객체를 정상적으로 가져온 경우
        model.addAttribute("vo", vo);
    }

    if (vo == null) {
        System.out.println("정보 없음");
    }

    return MyCommon.Detail.VIEW_PATH + "bank_view.jsp";
}
```

해당 유저의 계좌 잔액이 확인 가능



거래내역

잔액

가나다님 입출금 계좌입니다.

계좌번호 : 160-72-01486

잔액 : ₩10,000원

상세정보

나가기

계좌 입·출금 내역

조회기간

전체 ▼ 검색

거래일자	항목	입금	출금	항목잔액
2021-01-28 01:33:37	출금	0	20,000	49,990,000
2021-01-28 01:33:17	입금	10,000	0	50,020,000
2021-01-28 01:32:53	출금	0	40,000	49,970,000
2021-01-28 01:32:46	입금	25,000	0	50,035,000
2021-01-28 01:32:36	입금	80,000	0	50,090,000

1 2

나가기

```
@RequestMapping("/detail_view.do")
public String detail_view(Model model, String account ) {

    List<DetailVO>list=null;

    if (account == null) {
        list = detail_dao.selectView();
    } else {
        list = detail_dao.selectView(account);
    }

    model.addAttribute("list", list);

    return MyCommon.Detail.VIEW_PATH + "bank_detail.jsp";
}
```

입금하고 출금한 날짜, 금액 조회 가능하며, 1개월, 3개월 전 입출금 내역 확인도 가능



게시판-Q&A

Q&A					
번호	제목	작성자	작성일	조회수	
62	질문	chr	2021-01-06 14:42:24	4	
41	오랜만	손칼국수	2020-12-16 15:07:04	24	
61	ㄴ 새해	chr	2021-01-06 13:16:09	1	
36	Lion Heart	Girls Generation	2020-12-05 09:33:58	5	
	제목	질문	하하호호	2020-12-05 09:29:58	0
	작성자	chr 2 3 ▶			
	작성일	2021-01-06 14:42:24			
	내용	<div>회원가입 질문있습니다. 1717</div>			
	비밀번호	<input type="password"/>			

목록보기

답변

수정

글삭제

등록

비회원도 비밀번호만
지정하면 Q&A 사용가능



게시판-공지사항

관리자님 안녕하세요.

로그아웃

회원정보수정

계좌상세정보

공지사항(이벤트)

번호	제목	작성일	조회수
6	[이벤트]당첨자 발표	2021-01-07 14:09:00	2
5	[이벤트]	2021-01-07 13:57:35	14
4	[공지]온라인상담 실시	2021-01-07 11:53:11	59
3	[이벤트]20201년	2021-01-07 11:03:00	2

◀ 1 2 ▶

등록

제목	[공지]코로나 방지를 위한 영업시간 단축 안내
작성일	2021-01-07 09:28:54
내용	이전 5시까지 운영을 3시로 변경

목록보기 수정 글삭제

관리자만 공지사항
등록,수정,글 삭제가 가능

hye님 안녕하세요.

로그아웃

회원정보수정

계좌상세정보

공지사항(이벤트)

번호	제목	작성일	조회수
6	[이벤트]당첨자 발표	2021-01-07 14:09:00	2
5	[이벤트]	2021-01-07 13:57:35	14
4	[공지]온라인상담 실시	2021-01-07 11:53:11	59
3	[이벤트]20201년	2021-01-07 11:03:00	3

◀ 1 2 ▶

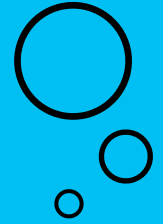
제목	[공지]코로나 방지를 위한 영업시간 단축 안내
작성일	2021-01-07 09:28:54
내용	이전 5시까지 운영을 3시로 변경

목록보기





EzoBank 보완할 점



보완할 점 (로그인)



로그인 상태에서 60분이 경과하면 자동으로 로그 아웃 처리



보완

시간을 표시하고, 시간 연장 버튼 생성, 60분이
경과하기 전에 Session 시간 연장 가능



보완할 점 (계좌 내역)

...계좌 입·출금 내역...

조회기간

전체 ▼ 검색

거래일자	항목	입금	출금	항목잔액
2021-01-28 01:33:37	출금	0	20,000	49,990,000
2021-01-28 01:33:17	입금	10,000	0	50,020,000
2021-01-28 01:32:53	출금	0	40,000	49,970,000
2021-01-28 01:32:46	입금	25,000	0	50,035,000
2021-01-28 01:32:36	입금	80,000	0	50,090,000

◀ 1 2 ▶

나가기

현재 계좌 입,출금 내역만 조회가 가능,
1개월, 3개월 전 내역만 조회 가능



보완

이체 대상과 내역도 함께 조회 가능
캘린더를 통해 해당 월/일 조회 가능





Thank you

EzoBank Code

https://github.com/tmdrnr96/YSK_/tree/master/EzoBank

해당 웹사이트 코드는 위의 깃 허브에서 확인해 보실 수 있습니다.

