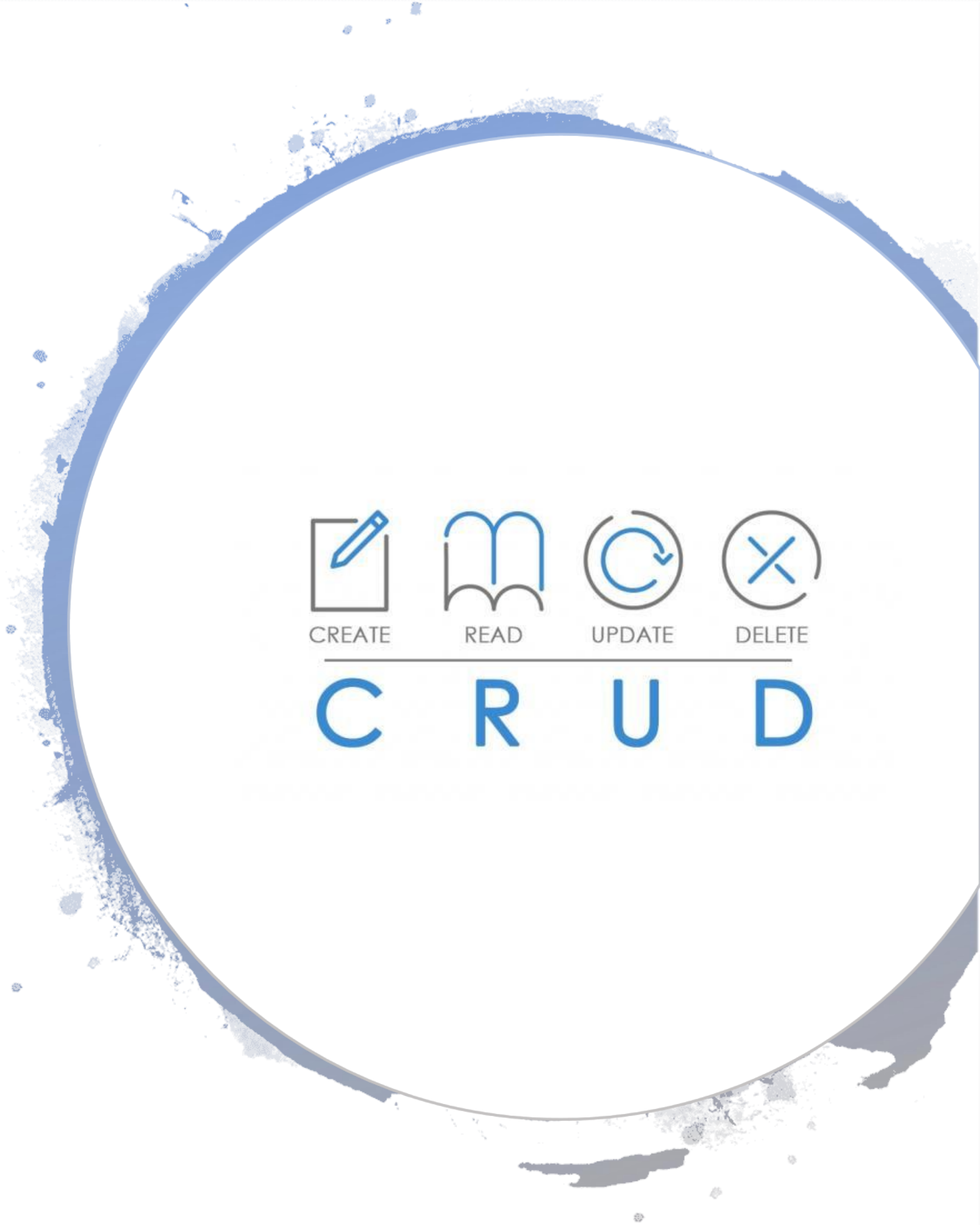
2BeConnected

By : Kaushik Patil (001056490)

Objective

The aim of my final project is to make a contact application website using Spring Web MVC Framework with Create, Read, Update and Delete (CRUD) Functionalities.



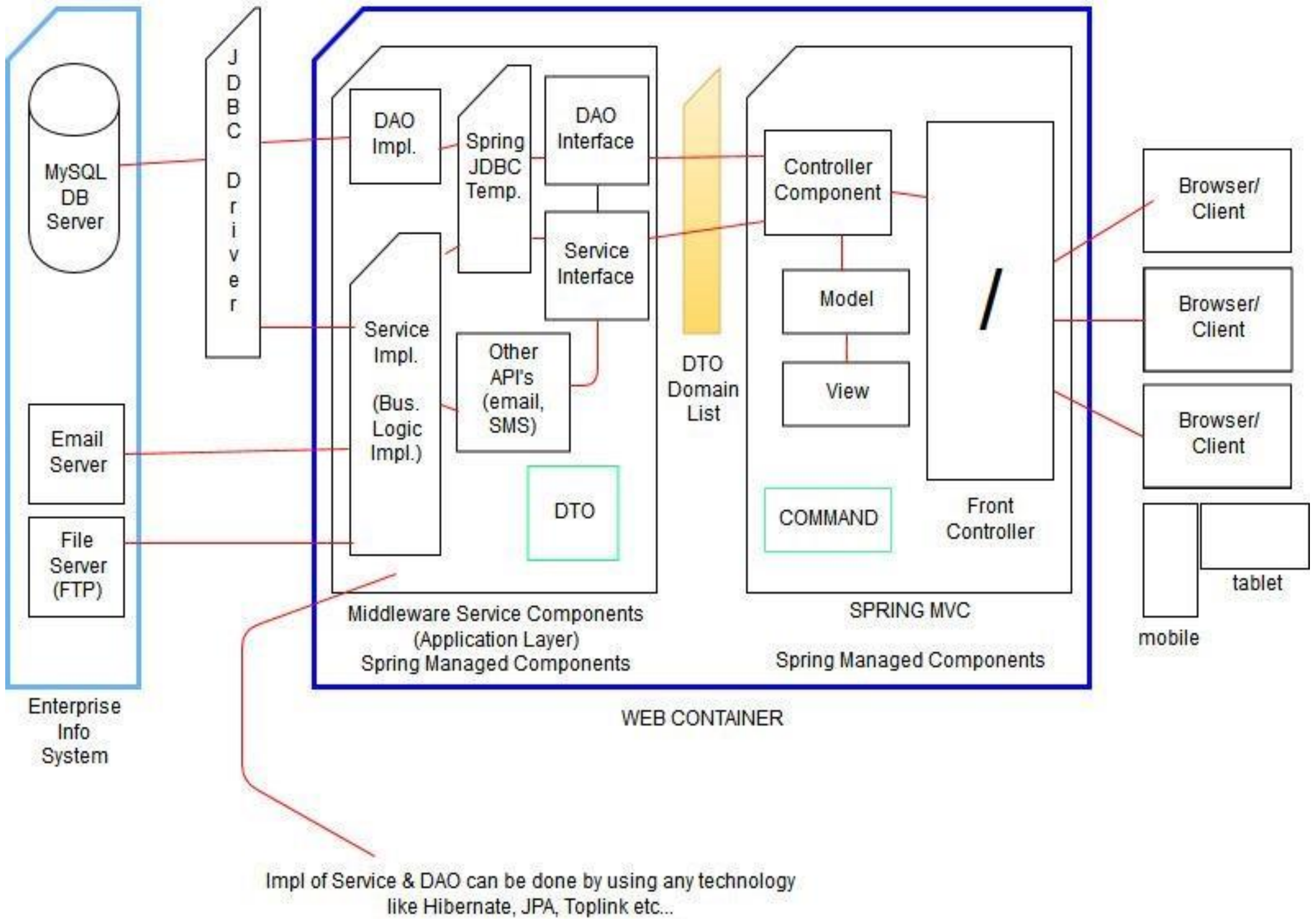
Technologies Used

- Java EE – For JSP and Servlet
- Framework – Spring Web MVC & Hibernate for Mapping
- Database and Management – MySQL and MySQL Workbench
- Integrated Development Environment (IDE) - NetBeans
- Server – Apache Tomcat
- UI Tech – HTML, CSS
- JavaScript library – AJAX throw jQuery
- Build Tool - Apache Maven

Project Requirements and Characteristics

- Multiuser
- Two Roles : User and Admin
- Systematic User Registration
- Login and Logout for User Account
- Add Contacts Feature
- Update Contacts Feature
- Delete Contact Feature
- Bulk Delete Feature
- Search Contact Feature with keyword
- Contact List with all contact details
- Admin can see all the registered users
- Admin has functionality to activate or block the user account
- PDF generator in admin account

Project Architecture



Database Entities

Contact

- **Contact ID**
- **Name**
- **Phone**
- **Email ID**
- **Address**
- **Remark**

User

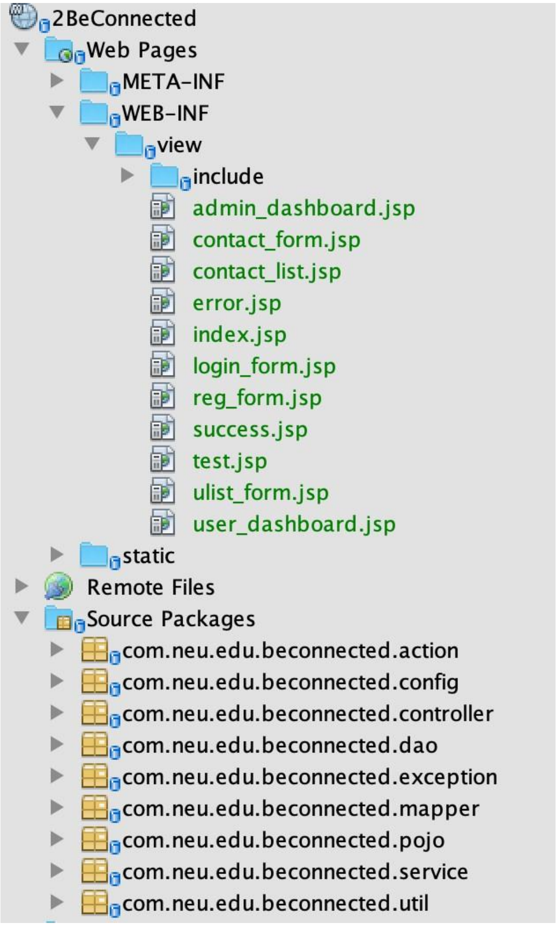
- **User ID**
- **Name**
- **Phone**
- **Email ID**
- **Address**
- **Login Name**
- **Password**
- **Role**
- **Login Status**

SQL Commands

```sql
CREATE TABLE `contact` (
  `contactId` int(11) NOT NULL AUTO_INCREMENT,
  `userId` int(11) DEFAULT NULL,
  `name` varchar(50) DEFAULT NULL,
  `phone` varchar(50) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `address` varchar(150) DEFAULT NULL,
  `remark` varchar(150) DEFAULT NULL,
  PRIMARY KEY (`contactId`),
  KEY `user_fk_idx` (`userId`),
  CONSTRAINT `user_fk` FOREIGN KEY (`userId`) REFERENCES `user`
(`userId`) ON DELETE CASCADE ON UPDATE NO ACTION
);
```

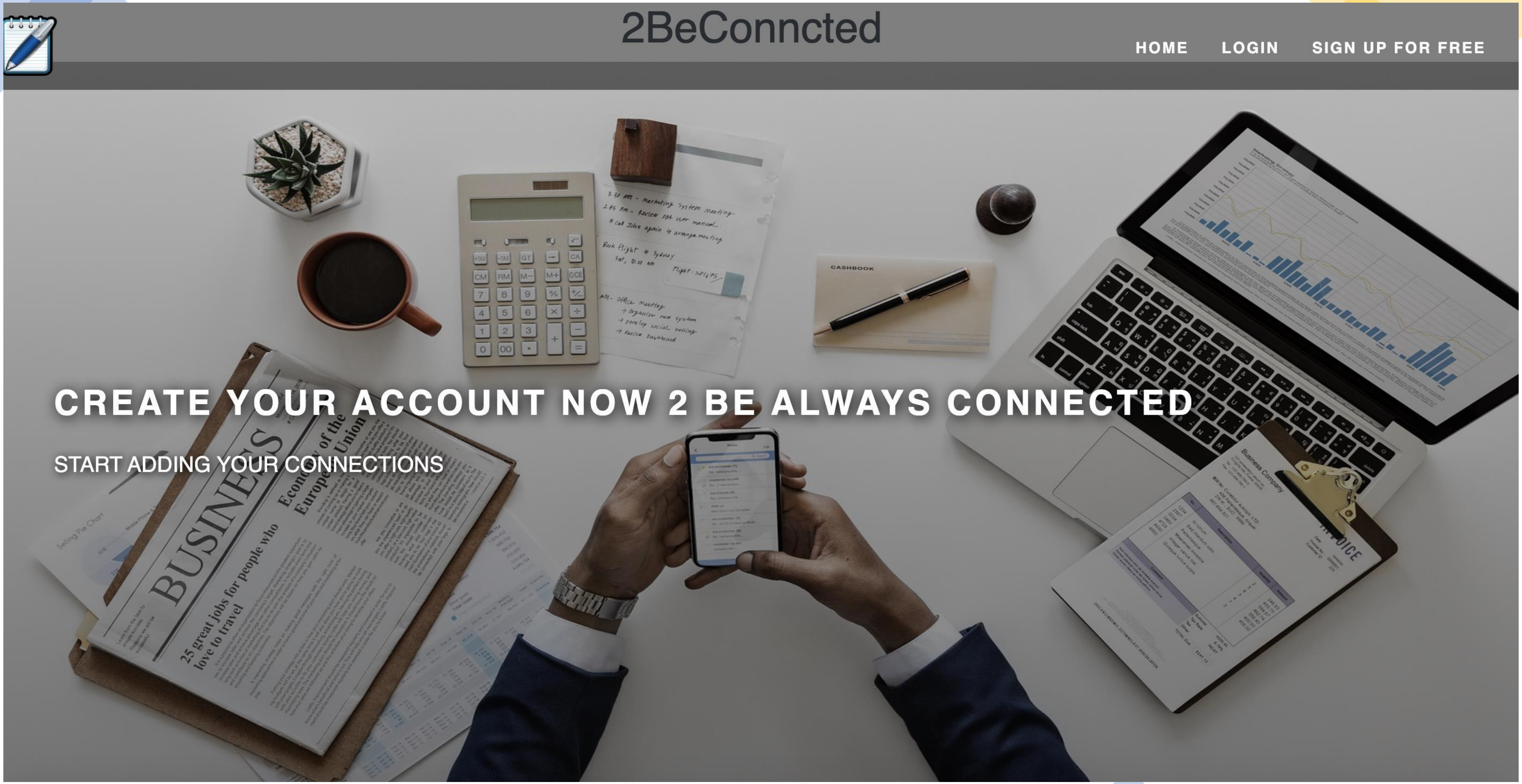
```sql
CREATE TABLE `user` (
  `userId` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) DEFAULT NULL,
  `phone` varchar(50) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `address` varchar(150) DEFAULT NULL,
  `loginName` varchar(45) NOT NULL,
  `password` varchar(50) DEFAULT NULL,
  `role` int(1) NOT NULL DEFAULT '2',
  `loginStatus` int(1) NOT NULL DEFAULT '1',
  PRIMARY KEY (`userId`),
  UNIQUE KEY `loginName_UNIQUE` (`loginName`)
);
```

Project Structure



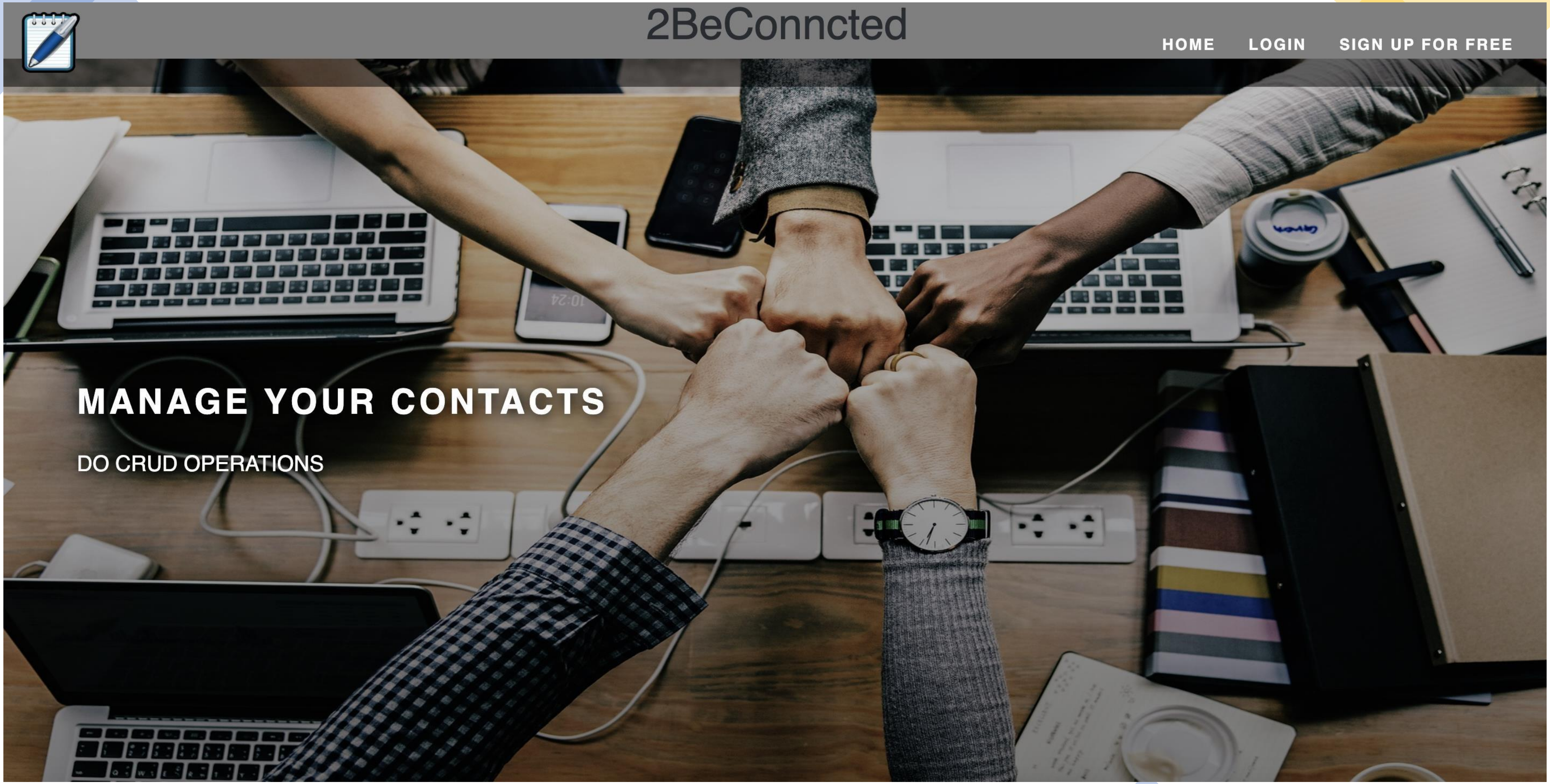
```
2BeConnected
▼ Web Pages
  ▶ META-INF
  ▼ WEB-INF
    ▼ view
      ▶ include
        admin_dashboard.jsp
        contact_form.jsp
        contact_list.jsp
        error.jsp
        index.jsp
        login_form.jsp
        reg_form.jsp
        success.jsp
        test.jsp
        ulist_form.jsp
        user_dashboard.jsp
  ▶ static
▶ Remote Files
▼ Source Packages
  ▶ com.neu.edu.beconnected.action
  ▶ com.neu.edu.beconnected.config
  ▶ com.neu.edu.beconnected.controller
  ▶ com.neu.edu.beconnected.dao
  ▶ com.neu.edu.beconnected.exception
  ▶ com.neu.edu.beconnected.mapper
  ▶ com.neu.edu.beconnected.pojo
  ▶ com.neu.edu.beconnected.service
  ▶ com.neu.edu.beconnected.util
```

Project UI Design



CREATE YOUR ACCOUNT NOW 2 BE ALWAYS CONNECTED

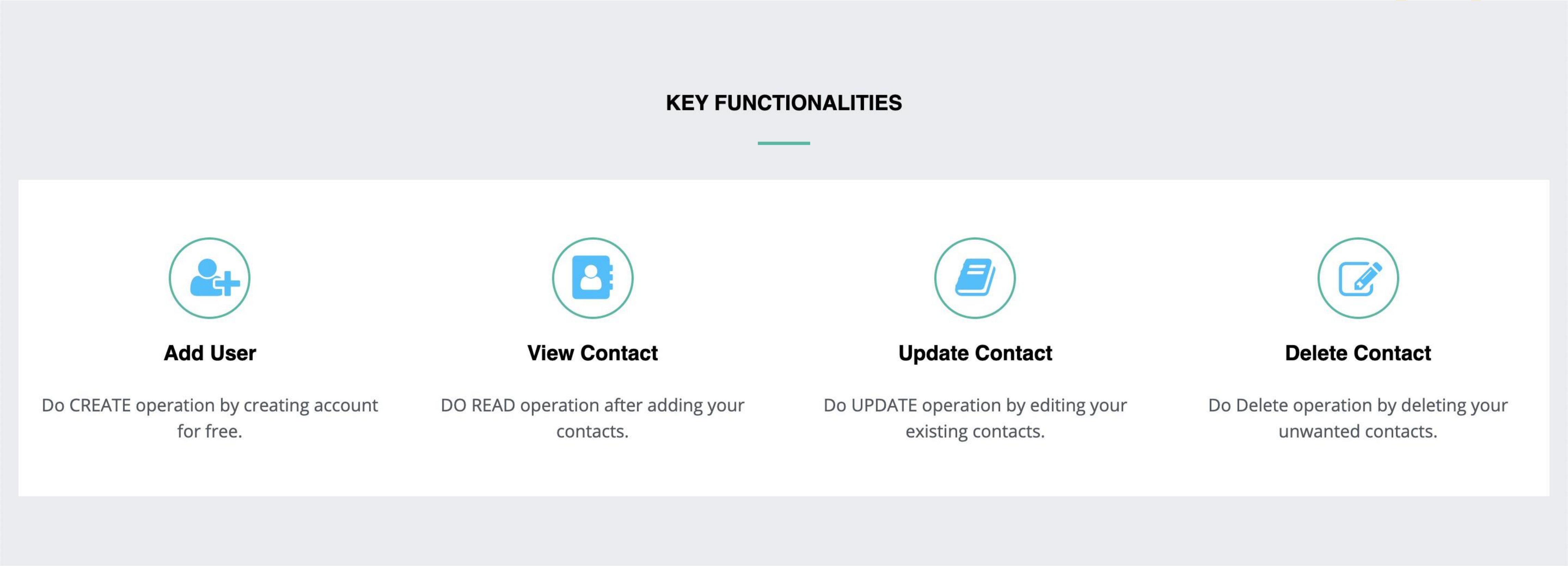
START ADDING YOUR CONNECTIONS



2BeConncted

MANAGE YOUR CONTACTS

DO CRUD OPERATIONS



KEY FUNCTIONALITIES

Add User

Do CREATE operation by creating account for free.

View Contact

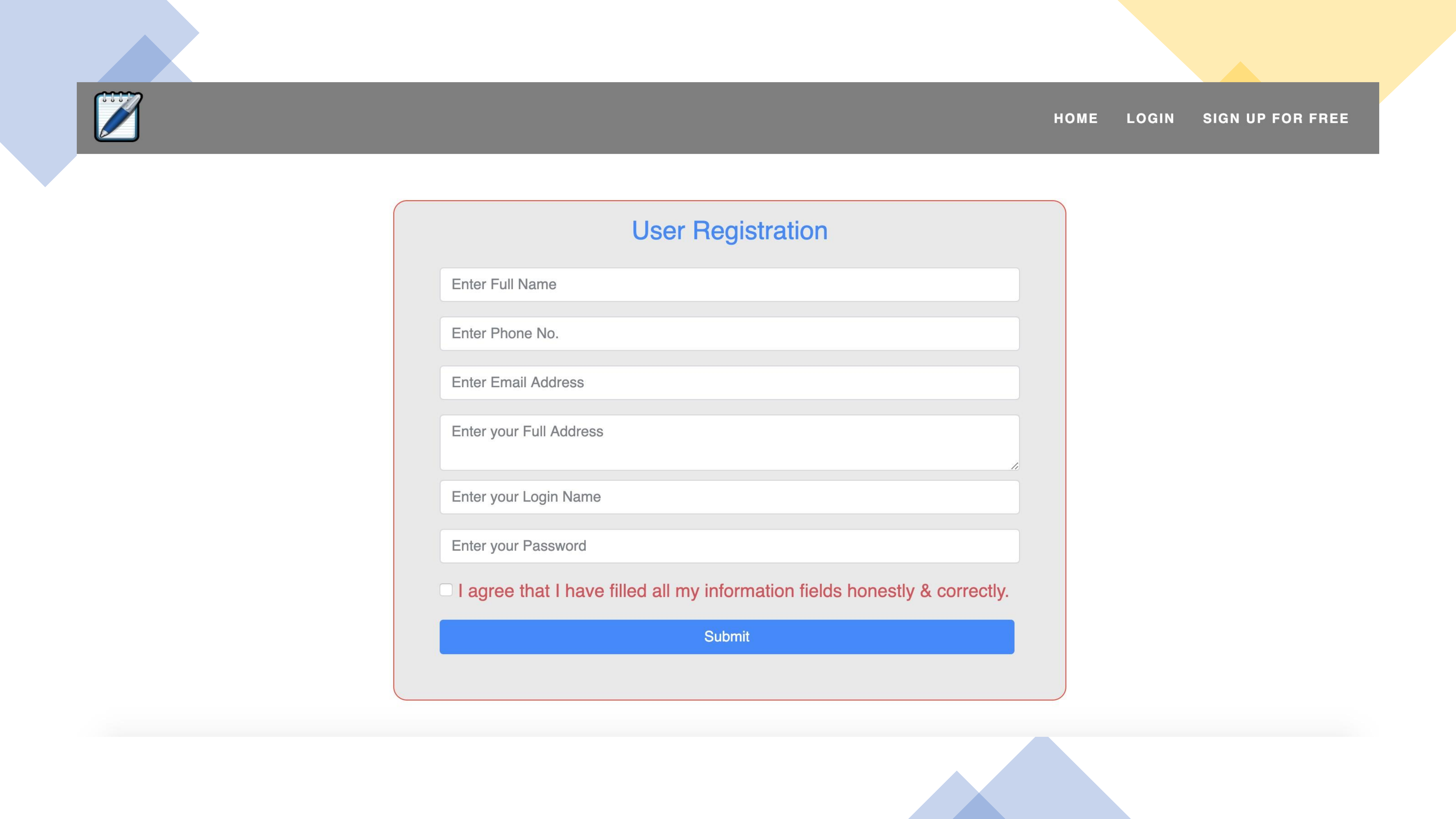
DO READ operation after adding your contacts.

Update Contact

Do UPDATE operation by editing your existing contacts.

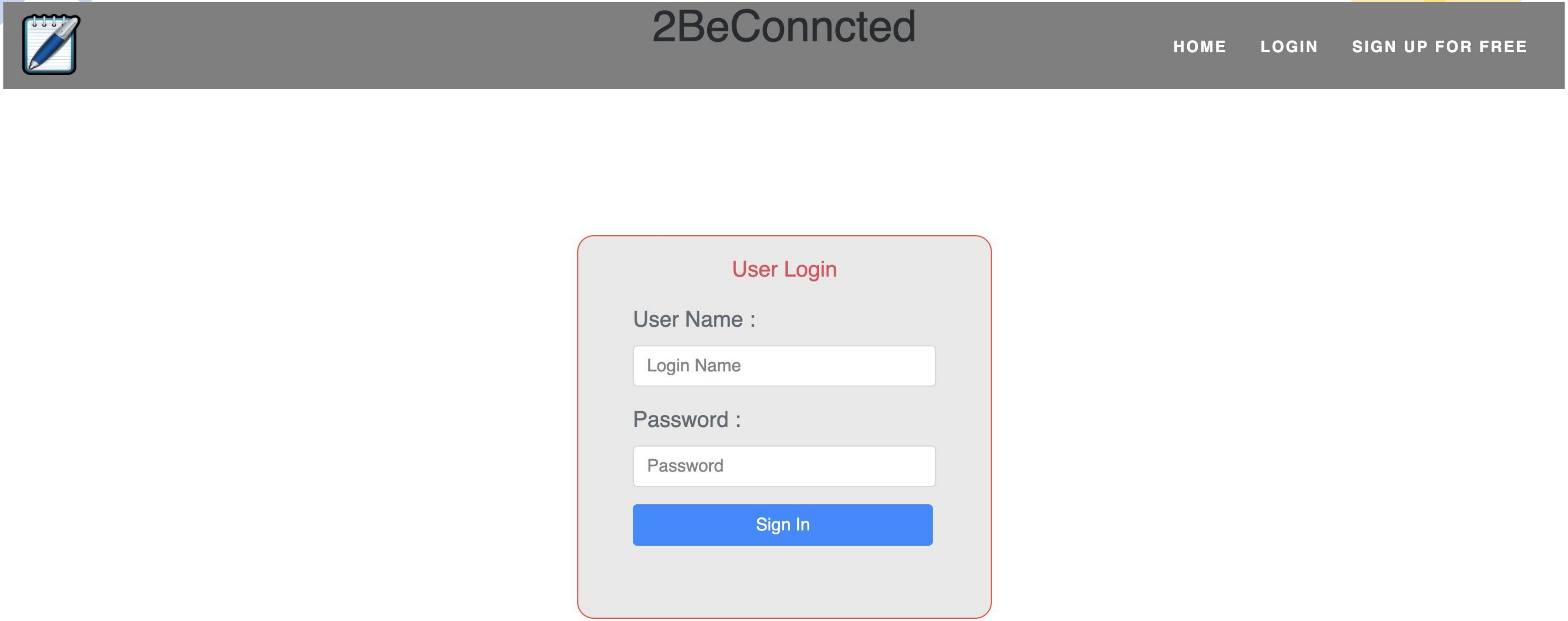
Delete Contact

Do Delete operation by deleting your unwanted contacts.



User Registration

□ I agree that I have filled all my information fields honestly & correctly.

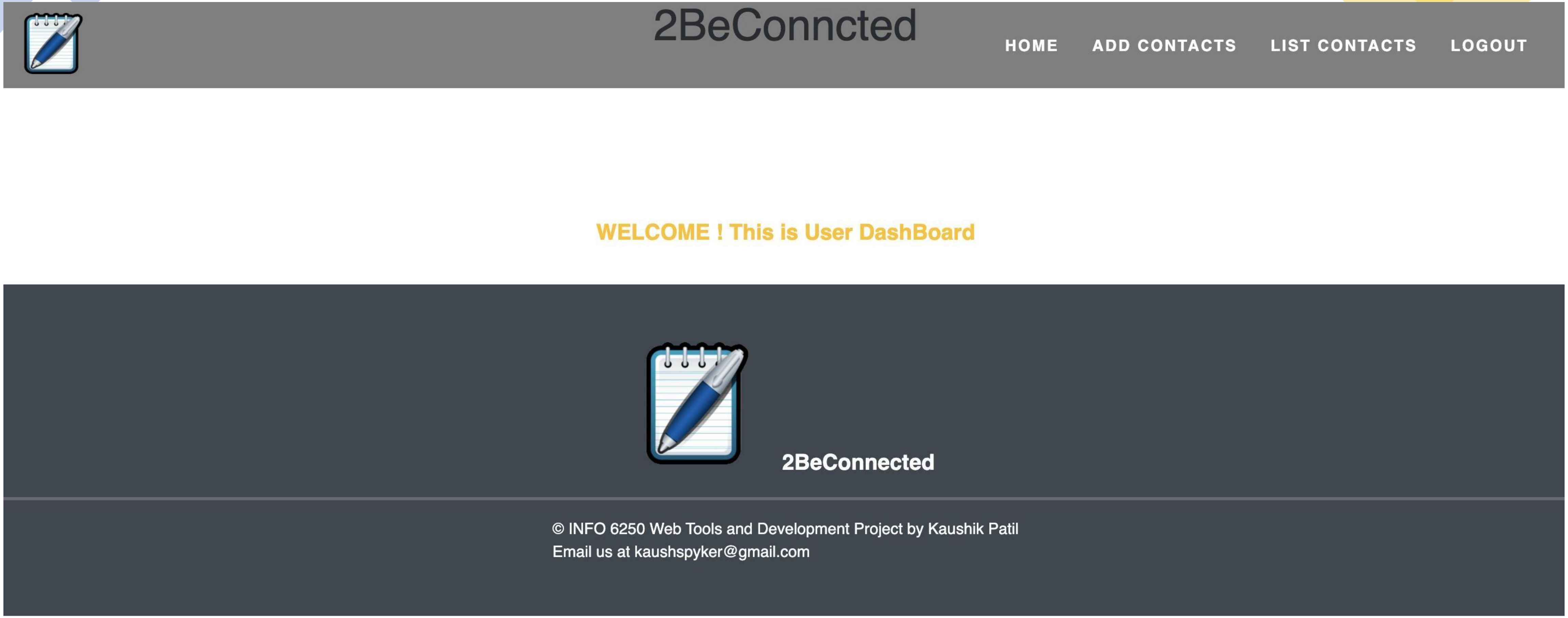


2BeConncted

User Login

User Name :

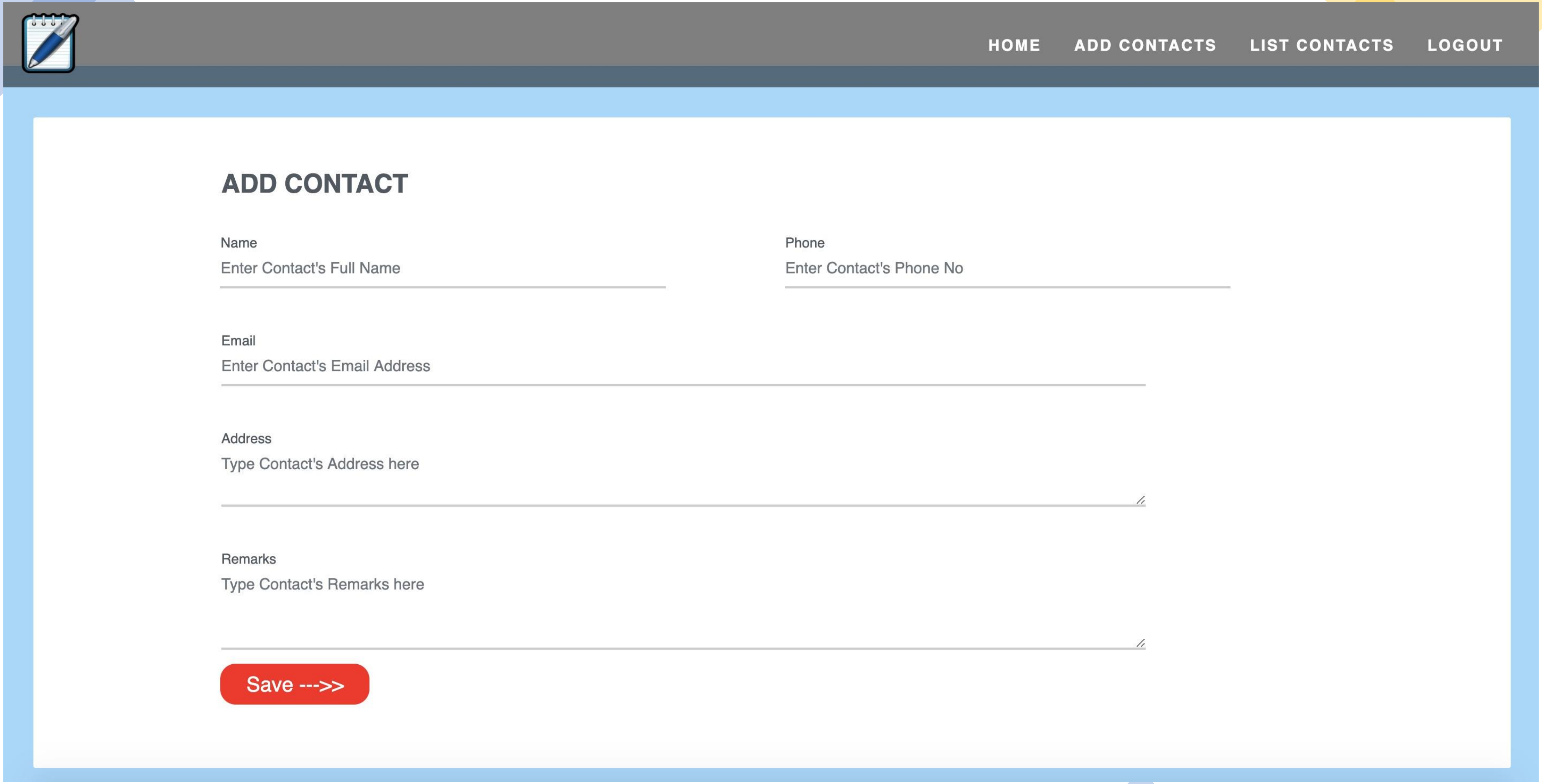
Password :



WELCOME ! This is User DashBoard

2BeConnected

© INFO 6250 Web Tools and Development Project by Kaushik Patil
Email us at kaushspyker@gmail.com



ADD CONTACT

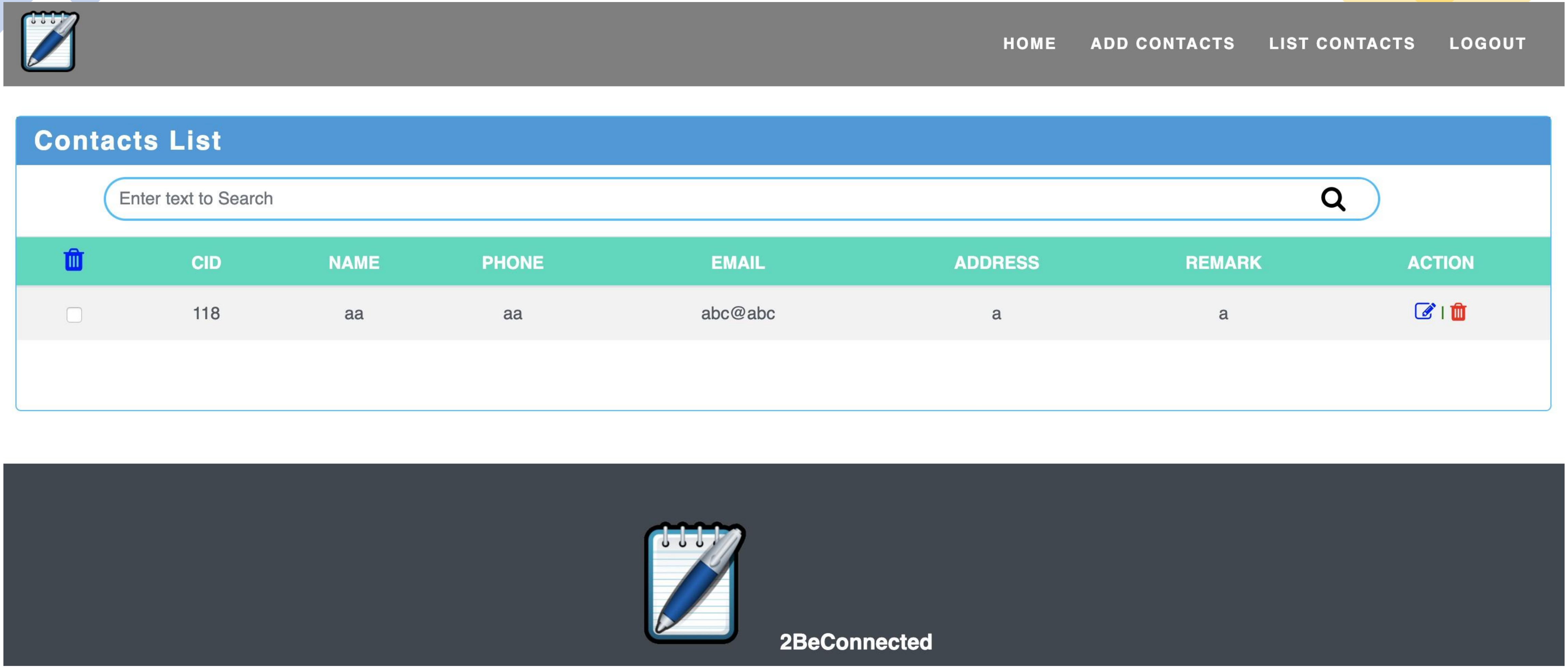
Name
Enter Contact's Full Name

Phone
Enter Contact's Phone No

Email
Enter Contact's Email Address

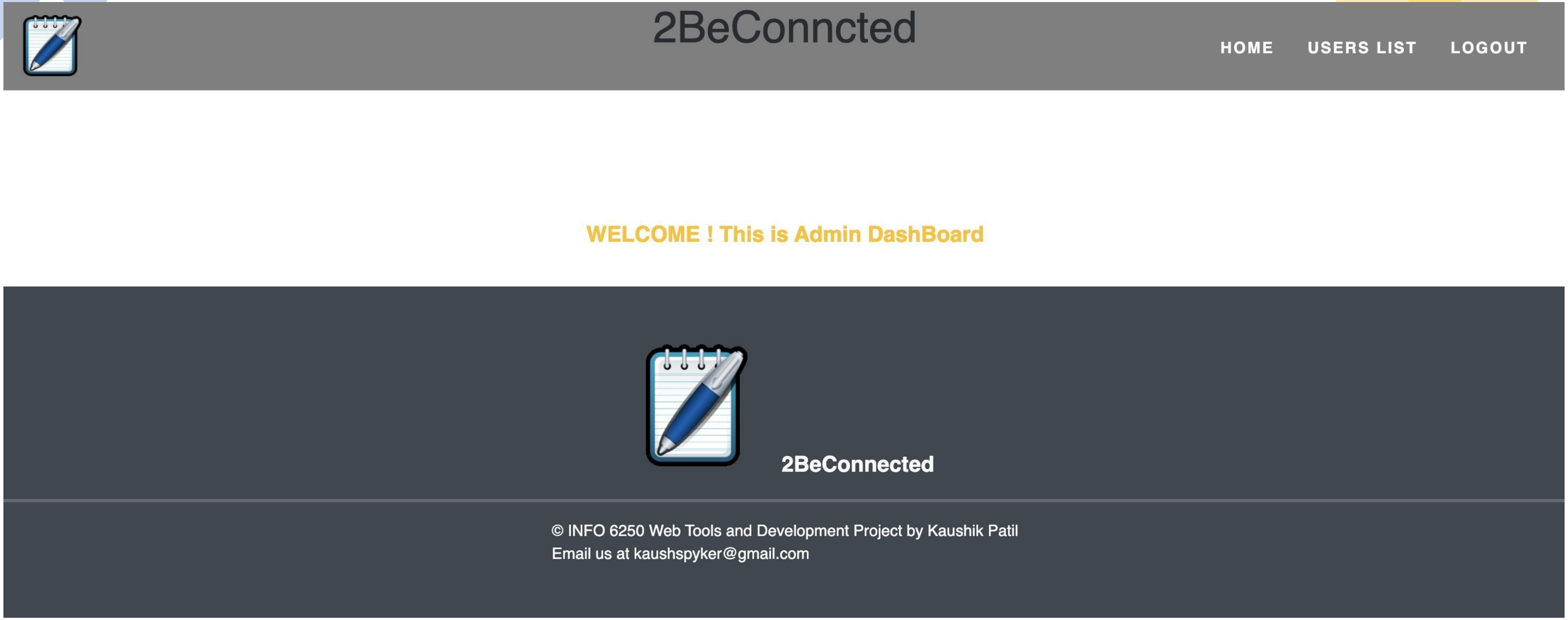
Address
Type Contact's Address here

Remarks
Type Contact's Remarks here



Contacts List

🗑	CID	NAME	PHONE	EMAIL	ADDRESS	REMARK	ACTION
□	118	aa	aa	abc@abc	a	a	✎ | 🗑

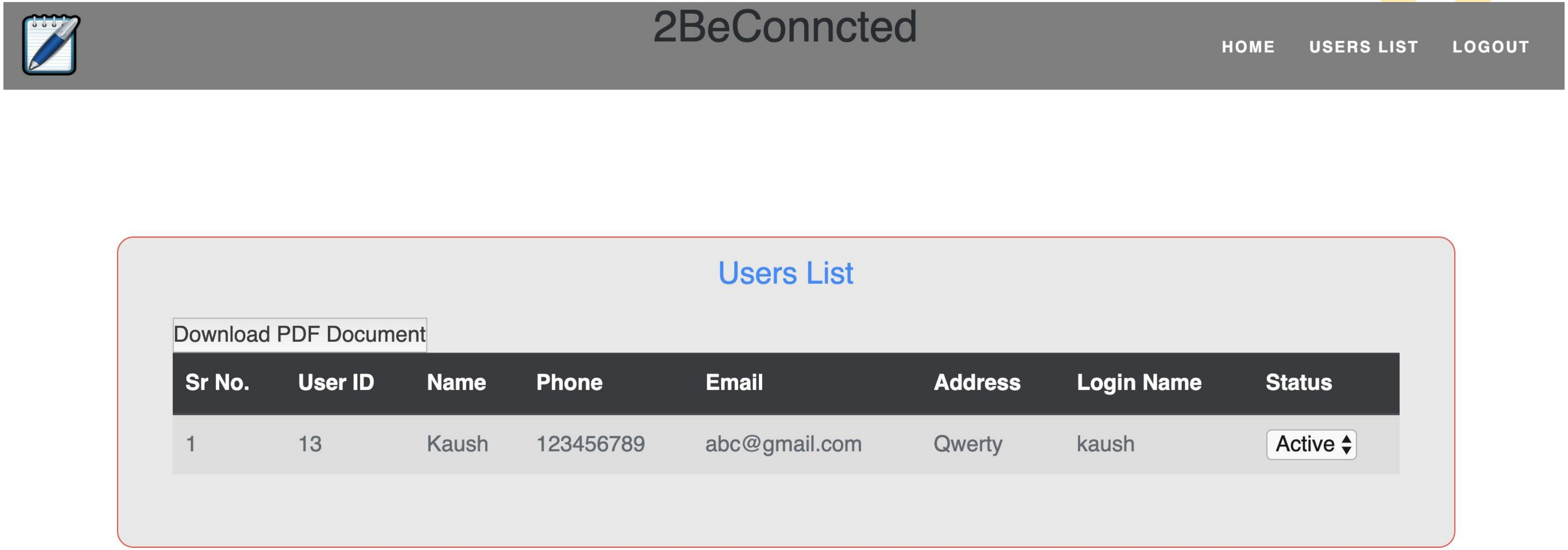


2BeConncted

WELCOME ! This is Admin DashBoard

2BeConnected

© INFO 6250 Web Tools and Development Project by Kaushik Patil
Email us at kaushspyker@gmail.com



Users List

Sr No.	User ID	Name	Phone	Email	Address	Login Name	Status
1	13	Kaush	123456789	abc@gmail.com	Qwerty	kaush	Active

2BeConnected

© INFO 6250 Web Tools and Development Project by Kaushik Patil
Email us at kaushspyker@gmail.com

Contact Controller

```
package com.neu.edu.beconnected.controller;

import com.neu.edu.beconnected.pojo.Contact;
import com.neu.edu.beconnected.service.ContactService;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

/**
```

```
 *
 * @author kaushikpatil
 */

@Controller
public class ContactController {

    @Autowired
    ContactService contactService;

    @RequestMapping(value="/user/contact_form")
    public String contactForm(Model m){

        Contact contact = new Contact();
        m.addAttribute("command", contact);
        return "contact_form";
    }

    @RequestMapping(value="/user/contact_save")
    public String saveOrUpdateContact(@ModelAttribute("command") Contact c, Model m, HttpSession session) {
        Integer contactId = (Integer) session.getAttribute("aContactId");
        if (contactId == null) {

            try {
                Integer userId = (Integer) session.getAttribute("userId");
                c.setUserId(userId);
                contactService.save(c);
                return "redirect:/user/contact_list?act=csv";
            } catch (Exception e) {
                e.printStackTrace();
            m.addAttribute("err", "Failed to save contact");
            return "contact_form";
        }
    } else {

        try {
            c.setContactId(contactId);
            contactService.update(c);
            session.removeAttribute("aContactId");
            return "redirect:/user/contact_list?act=ed";
        } catch (Exception e) {
            e.printStackTrace();
            m.addAttribute("err", "Failed to Edit contact");
            return "contact_form";
        }
    }
}

@RequestMapping(value = "/user/edit_contact")
public String prepareEditForm(Model m, HttpSession session, @RequestParam("cid") Integer contactId) {
    session.setAttribute("aContactId", contactId);
    Contact c = contactService.findById(contactId);
    m.addAttribute("command", c);
    return "contact_form";
}


@RequestMapping(value="/user/contact_list")
public String contactList(Model m, HttpSession session) {
    Integer userId = (Integer) session.getAttribute("userId");
    m.addAttribute("contactList", contactService.findUserContact(userId));
    return "contact_list";
}

@RequestMapping(value="/user/del_contact")
public String deleteContact(@RequestParam("cid") Integer contactId) {
    contactService.delete(contactId);
    return "redirect:/user/contact_list?act=del";
}

@RequestMapping(value="/user/contact_search")
public String contactSearch(Model m, HttpSession session, @RequestParam("freeText") String freeText) {
    Integer userId = (Integer) session.getAttribute("userId");
    m.addAttribute("contactList", contactService.findUserContact(userId, freeText));
    return "contact_list";
    }


    @RequestMapping("/user/bulk_contact_delete")
    public String deleteBulkContact(@RequestParam("cid") Integer[] contactIds) {
        contactService.delete(contactIds);
        return "redirect:/user/contact_list?val=bdel";
    }
}
```

User Controller

```
package com.neu.edu.beconnected.controller;


import com.neu.edu.beconnected.action.LoginAction;
import com.neu.edu.beconnected.action.UserAction;
import com.neu.edu.beconnected.pojo.User;
import com.neu.edu.beconnected.exception.UserException;
import com.neu.edu.beconnected.service.UserService;
import javax.servlet.http.HttpSession;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.dao.DuplicateKeyException;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;



/**
 *
 * @author kaushikpatil
 */

@Controller
public class UserController {

    @Autowired
```

```
private UserService userService;

@RequestMapping(value={"/","/index"})
public String index(Model m){
   return "index";
}

@RequestMapping("/login_form")
public String loginForm(Model m){
   m.addAttribute("command",new LoginAction());
   return "login_form";
}

@RequestMapping(value= {"/login"}, method = RequestMethod.POST)
public String handleLogin(@ModelAttribute("command") LoginAction lcmd, Model m, HttpSession session){
```

```
try {

  User loggedInUser = userService.login(lcmd.getLoginName(), lcmd.getPassword());

  if(loggedInUser==null){
    m.addAttribute("err", "Login Failed! Please Enter the Valid Credentials");
    return "login_form";
  }
  else{
    if(loggedInUser.getRole().equals(UserService.ROLE_ADMIN)){

      this.addUserInSession(loggedInUser, session);
      return "redirect:admin/dashboard";
    }
    else if(loggedInUser.getRole().equals(UserService.ROLE_USER)){
```

```
                this.addUserInSession(loggedInUser, session);
                return "redirect:user/dashboard";
            }
            else{
                m.addAttribute("err", "Invalid User!");
                return "login_form";
            }
        }

    } catch (UserException ex) {
        m.addAttribute("err", ex.getMessage());
        return "login_form";
    }
}
```

```
@RequestMapping(value={"/logout"})
public String logout(HttpSession session){
   session.invalidate();
   return "redirect:/index?act=lo";
}

@RequestMapping(value={"/user/dashboard"})
public String userDashBoard(){
   return "user_dashboard";
}


@RequestMapping(value={"/admin/dashboard"})
public String adminDashBoard(){
   return "admin_dashboard";
}

@RequestMapping("/user/list")
public String userList(Model m){
   m.addAttribute("usersList", userService.getUserList());

   return "ulist_form";
}

@RequestMapping(value={"/reg_form"})
public String registrationForm(Model m){
   UserAction cmd = new UserAction();
   m.addAttribute("command", cmd);
   return "reg_form";
```

```
}

@RequestMapping(value="/register")
public String registerUser(@ModelAttribute("command") UserAction cmd, Model m){
   try{
      User user=cmd.getUser();
      user.setRole(UserService.ROLE_USER);
      user.setLoginStatus(UserService.LOGIN_STATUS_ACTIVE);
      userService.register(user);
      return "redirect:/index?act=reg";
   }catch(DuplicateKeyException duplicateKeyException){
      duplicateKeyException.printStackTrace();
      m.addAttribute("err","Login Name already Exist! Please try with another Login Name");
      return "reg_form";
   }
```

```
}

@RequestMapping(value={"/user/change_status"})
@ResponseBody
public String changeStatus(@RequestParam("userId") Integer userId,@RequestParam("loginStatus") Integer loginStatus){
  try{
    userService.changeLoginStatus(userId, loginStatus);
    return "Success : Status Changed! ";
  }catch(Exception ex){
    ex.printStackTrace();
    return "Error : Unable To change Status ! Try Again";
  }
}
```

```
    private void addUserInSession(User u, HttpSession session){
        session.setAttribute("user", u);
        session.setAttribute("userId", u.getUserId());
        session.setAttribute("role", u.getRole());
    }
}
```

PDF Controller

```
package com.neu.edu.beconnected.controller;

import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.View;
import com.neu.edu.beconnected.pojo.User;

@Controller
public class PDFController {
@RequestMapping(value = "/report.pdf", method = RequestMethod.GET)
  public ModelAndView downloadPDF(HttpServletRequest request) {
      View view = new PDFView();
      Map<String,Object> userData = new HashMap<String,Object>();
      User user = (User) request.getSession().getAttribute("user");
      userData.put("user", user);
    return new ModelAndView(view, "userData", userData);
  }
}
```

Thank you