TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÁO CÁO QUÁ TRÌNH 1**

**CÔNG NGHỆ JAVA**

**User Management**

*Người hướng dẫn*: **ThS.** **PHẠM THÁI KÌ TRUNG**

*Người thực hiện*: **Hồ Hữu An – 521H0489**

*Khoá:* **25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo, company name

Description automatically generated

**BÁO CÁO QUÁ TRÌNH 1**

**CÔNG NGHỆ JAVA**

**User Management**

*Người hướng dẫn*: **ThS.** **PHẠM THÁI KÌ TRUNG**

*Người thực hiện*: **Hồ Hữu An – 521H0489**

*Khoá*  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

# LỜI CẢM ƠN

Trước khi học môn học này, em chưa biết gì về Công nghệ java. Em chỉ biết rằng đó là một trong những công nghệ nổi tiếng và phổ biến nhưng em không biết nó như thế nào. Sau khi học môn học này, em đã có cái nhìn toàn diện hơn về Công nghệ java và tầm quan trọng của nó trong cuộc sống hiện đại. Em đã học được về các loại JDBC, Hibernate, JSP Servlet, Spring boot.

Em đặc biệt ấn tượng với cách giảng dạy của thầy. Thầy rất nhiệt tình và có kiến thức sâu rộng về bảo mật thông tin. Thầy đã truyền tải kiến thức một cách dễ hiểu và thú vị. Thầy cũng luôn sẵn sàng giải đáp thắc mắc của em và giúp em hiểu rõ hơn về những gì mình đang học.

Nhờ sự hướng dẫn của thầy, em đã có thể hoàn thành bài báo cáo của mình một cách xuất sắc. Em đã học được rất nhiều điều từ bài báo cáo này và em tin rằng những kiến thức này sẽ giúp em trong tương lai.

Một lần nữa, em xin gửi lời cảm ơn sâu sắc tới thầy Phạm Thái Kì Trung. Em hy vọng rằng thầy sẽ tiếp tục giảng dạy môn học này và giúp đỡ nhiều sinh viên khác như em.

**BÁO CÁO ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Em xin cam đoan đây là sản phẩm đồ án của riêng chúng em và được sự hướng dẫn của TS. Bùi Quy Anh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng em xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 8 tháng 10 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*



*Hồ Hữu An*

# PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc147846481)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN 3](#_Toc147846482)

[MỤC LỤC 1](#_Toc147846483)

[CHƯƠNG 1: Tổng quan và giới thiệu chức năng 2](#_Toc147846484)

[Account admin 2](#_Toc147846485)

[Công nghệ sử dụng 2](#_Toc147846486)

[JSP 2](#_Toc147846487)

[Servlet 2](#_Toc147846488)

[JDBC (Java Database Connectivity) 2](#_Toc147846489)

[Dependencies 3](#_Toc147846490)

[Chức năng 4](#_Toc147846491)

[CHƯƠNG 2: Source Code 8](#_Toc147846492)

[Backend 8](#_Toc147846493)

[Model 8](#_Toc147846494)

[Database 12](#_Toc147846495)

[User 12](#_Toc147846496)

[Manager 18](#_Toc147846497)

[DAO 26](#_Toc147846498)

[Frontend 34](#_Toc147846499)

[Component 34](#_Toc147846500)

[Manager 37](#_Toc147846501)

[User 45](#_Toc147846502)

[CHƯƠNG 3: Database 53](#_Toc147846503)

[CHƯƠNG 4: WebApp 56](#_Toc147846504)

# CHƯƠNG 1: Tổng quan và giới thiệu chức năng

## Account admin

Username: admin

Password: 123qwe

## Công nghệ sử dụng

### JSP

JavaServer Pages (JSP) là một công nghệ phát triển ứng dụng web Java, cho phép lập trình viên kết hợp mã Java và HTML để tạo ra các trang web động.

JSP cho phép nhúng mã Java trong các thẻ <% ... %> và sử dụng các thẻ đặc biệt (thẻ JSP) để thực hiện các tác vụ như lặp lại danh sách, điều kiện và truy cập cơ sở dữ liệu.

JSP được biên dịch thành các Servlet trước khi chạy trên máy chủ ứng dụng web. Kết quả được trả về từ Servlet sẽ được hiển thị trên trình duyệt của người dùng.

### Servlet

Servlet là một lớp Java được sử dụng để xử lý các yêu cầu và phản hồi trong mô hình ứng dụng web.

Servlet nhận yêu cầu từ trình duyệt và tạo ra phản hồi tương ứng. Phản hồi có thể là các trang HTML động, dữ liệu JSON hoặc bất kỳ loại dữ liệu nào khác.

Servlet cung cấp các phương thức để xử lý yêu cầu GET và POST, truy cập thông tin yêu cầu (ví dụ: thông tin về tham số, tiêu đề, cookie) và gửi phản hồi trở lại trình duyệt.

### JDBC (Java Database Connectivity)

JDBC là một API (Application Programming Interface) cho phép các ứng dụng Java kết nối và tương tác với các cơ sở dữ liệu quan hệ như MySQL, Oracle, SQL Server, v.v.

JDBC cung cấp các lớp và giao diện để thực hiện các thao tác như thiết lập kết nối đến cơ sở dữ liệu, thực thi câu truy vấn SQL, xử lý kết quả truy vấn và quản lý giao dịch.

JDBC sử dụng các trình điều khiển (driver) cụ thể cho mỗi cơ sở dữ liệu để thiết lập kết nối và tương tác với cơ sở dữ liệu.

### Dependencies

1. javax.servlet:jstl:1.2

groupId: javax.servlet

artifactId: jstl

version: 1.2

Mô tả: Đây là một thư viện JSTL (JSP Standard Tag Library) để sử dụng các thẻ tiêu chuẩn trong JSP. JSTL cung cấp các thẻ để thực hiện các tác vụ như lặp lại danh sách, điều kiện, định dạng số, truy cập cơ sở dữ liệu và xử lý các biểu thức.

1. javax.servlet.jsp:javax.servlet.jsp-api:2.3.3

groupId: javax.servlet.jsp

artifactId: javax.servlet.jsp-api

version: 2.3.3

Mô tả: Đây là một API (Application Programming Interface) cho JSP (JavaServer Pages). JSP API cung cấp các lớp và giao diện để phát triển và sử dụng các trang JSP trong ứng dụng web.

1. javax.servlet:javax.servlet-api:4.0.1

groupId: javax.servlet

artifactId: javax.servlet-api

version: 4.0.1

Mô tả: Đây là một API cho Servlet. Servlet API cung cấp các lớp và giao diện để phát triển và triển khai các servlet trong ứng dụng web.

1. com.mysql:mysql-connector-j:8.1.0

groupId: com.mysql

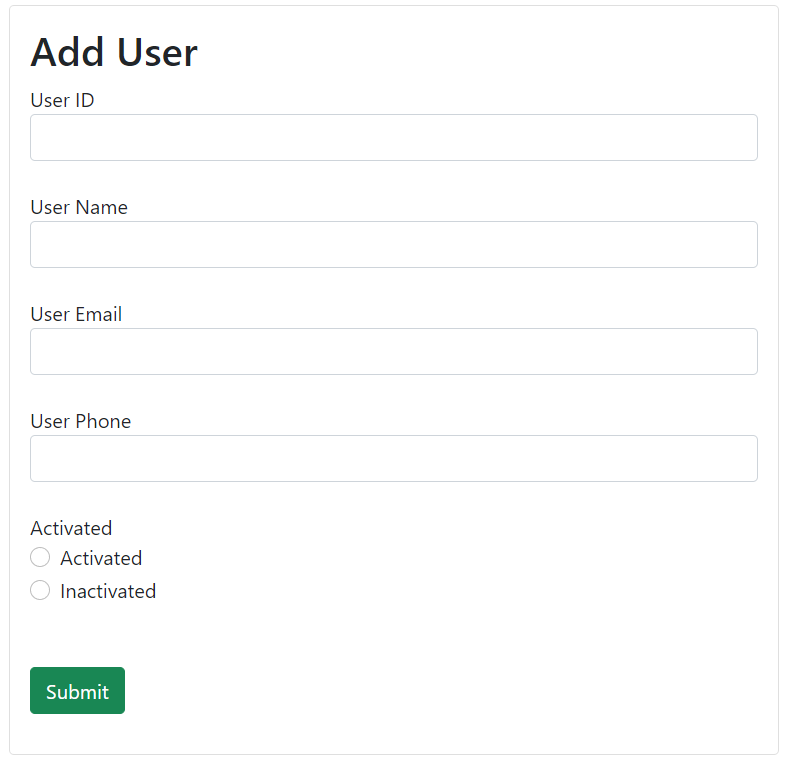
artifactId: mysql-connector-j

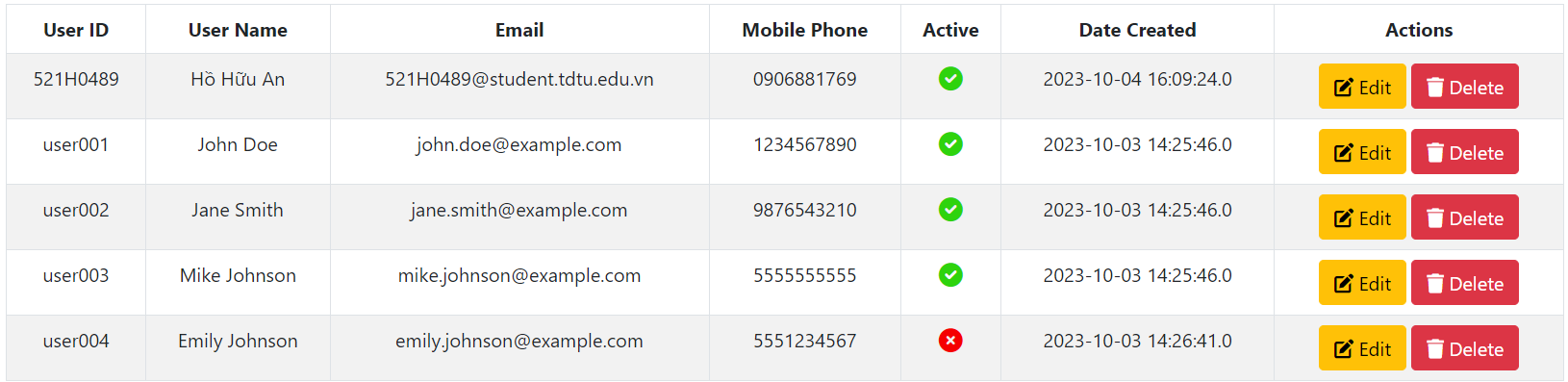
version: 8.1.0

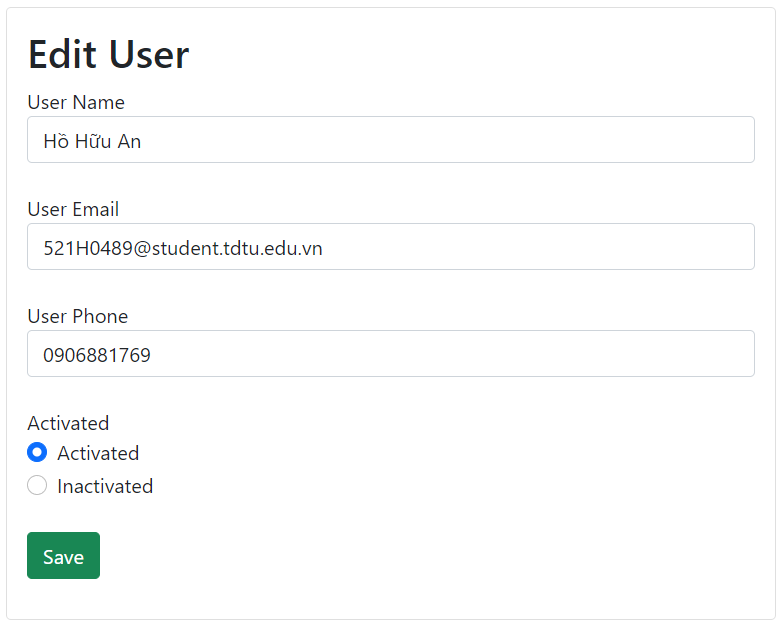
Mô tả: Đây là một trình điều khiển JDBC (Java Database Connectivity) để kết nối và tương tác với cơ sở dữ liệu MySQL từ ứng dụng Java.

## Chức năng

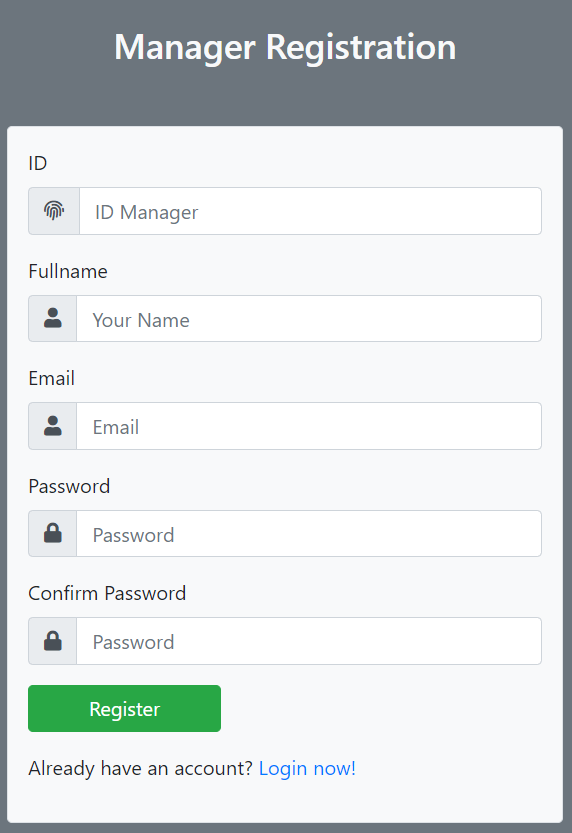
CRUD: Creat – Read – Update – Delete



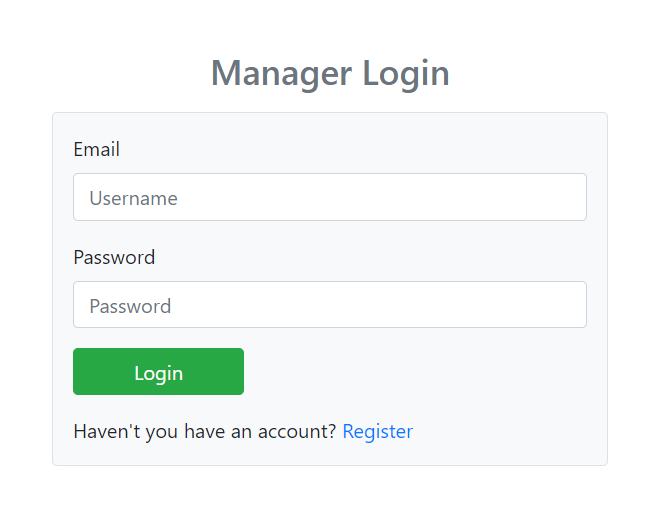




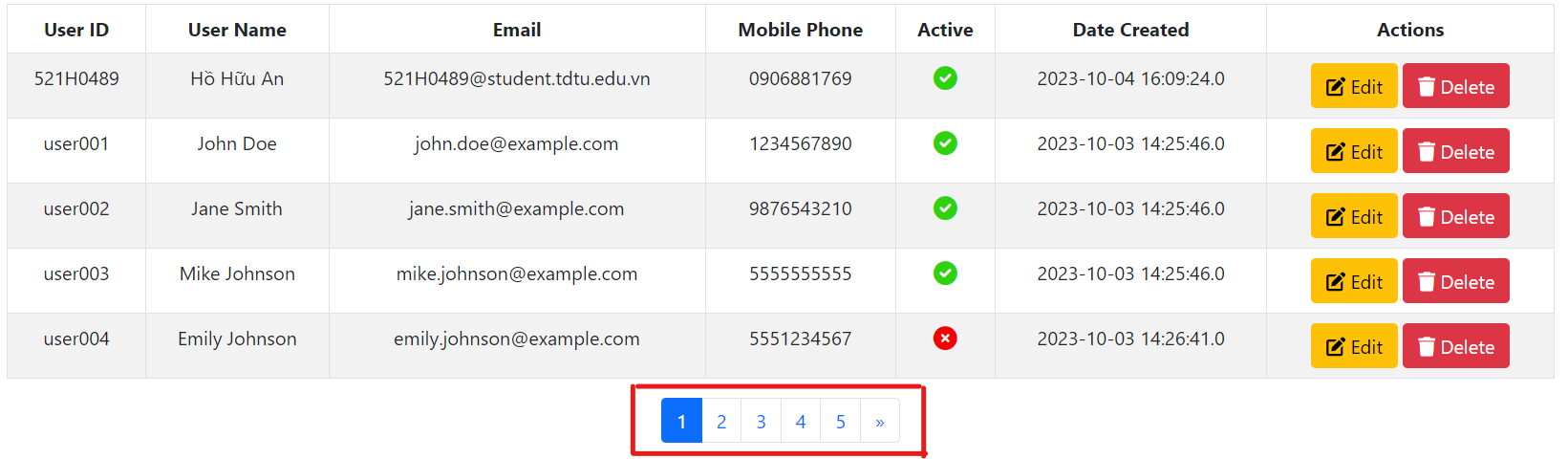
Đăng kí:



Đăng nhập:



Phân trang:



Phân quyền: admin và manager

# CHƯƠNG 2: Source Code

## Backend

### Model

**User.java**

package usermanagement.model;

import java.util.Date;

public class User {

    public String userID;

    public String userName;

    public String email;

    public String mobilePhone;

    public int isActive;

    public Date dateCreated;

    public User() {

        // Default constructor

    }

    public User(String userID, String userName, String email, String mobilePhone) {

        this.userID = userID;

        this.userName = userName;

        this.email = email;

        this.mobilePhone = mobilePhone;

    }

    public User(String userID, String userName, String email, String mobilePhone, int isActive, Date dateCreated) {

        this.userID = userID;

        this.userName = userName;

        this.email = email;

        this.mobilePhone = mobilePhone;

        this.isActive = isActive;

        this.dateCreated = dateCreated;

    }

    public User(String id, String name, String email, String country, int active) {

        this.userID = id;

        this.userName = name;

        this.email = email;

        this.mobilePhone = country;

        this.isActive = active;

    }

    public String getUserID() {

        return userID;

    }

    public void setUserID(String userID) {

        this.userID = userID;

    }

    public String getUserName() {

        return userName;

    }

    public void setUserName(String userName) {

        this.userName = userName;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getMobilePhone() {

        return mobilePhone;

    }

    public void setMobilePhone(String mobilePhone) {

        this.mobilePhone = mobilePhone;

    }

    public int getActive() {

        return isActive;

    }

    public void setIsActive(int isActive) {

        this.isActive = isActive;

    }

    public Date getDateCreated() {

        return dateCreated;

    }

    public void setDateCreated(Date dateCreated) {

        this.dateCreated = dateCreated;

    }

    @Override

    public String toString() {

        return "User{" +

                "userID='" + userID + '\'' +

                ", userName='" + userName + '\'' +

                ", email='" + email + '\'' +

                ", mobilePhone='" + mobilePhone + '\'' +

                ", isActive=" + isActive +

                ", dateCreated=" + dateCreated +

                '}';

    }

}

**Manager.java**

package usermanagement.model;

public class Manager {

        private String id;

        private String fullName;

        private String email;

        private String password;

    public Manager() {

    }

    public Manager(String id, String fullName, String email, String password) {

        this.id = id;

        this.fullName = fullName;

        this.email = email;

        this.password = password;

    }

    public Manager(String fullName, String password) {

        this.fullName = fullName;

        this.password = password;

    }

    public String getId() {

        return id;

    }

    public void setId(String id) {

        this.id = id;

    }

    public String getFullName() {

        return fullName;

    }

    public void setFullName(String fullName) {

        this.fullName = fullName;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public String getPassword() {

        return password;

    }

    public void setPassword(String password) {

        this.password = password;

    }

    @Override

    public String toString() {

        return "Manager{" +

                "id=" + id +

                ", fullName='" + fullName + '\'' +

                ", email='" + email + '\'' +

                ", password='" + password + '\'' +

                '}';

    }

}

### Database

**DBConnect.java**

package usermanagement.db;

import java.sql.Connection;

import java.sql.DriverManager;

public class DBConnect {

    private static Connection conn;

    public static Connection getConn() {

        try {

            Class.forName("com.mysql.cj.jdbc.Driver");

            conn = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/process1?user=root");

        } catch (Exception e) {

            e.printStackTrace();

        }

        return conn;

    }

}

### User

**AddUser.java**

package usermanagement.user;

import usermanagement.dao.UserDAO;

import usermanagement.model.User;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

import java.sql.SQLException;

import java.util.Objects;

@WebServlet("/add")

public class AddUser extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        RequestDispatcher dispatcher = req.getRequestDispatcher("user\_add.jsp");

        dispatcher.forward(req, resp);

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String id = req.getParameter("id");

        String name = req.getParameter("name");

        String email = req.getParameter("email");

        String phone = req.getParameter("phone");

        int activatedValue = (Objects.equals(req.getParameter("activated"), "true"))?1:0;

        HttpSession session = req.getSession();

        RequestDispatcher dispatcher = req.getRequestDispatcher("user\_add.jsp");

        //send to jsp

        req.setAttribute("user", new User(id, name, email, phone, activatedValue));

        UserDAO userDAO = new UserDAO();

        if(userDAO.selectAllId().contains(id)){

            session.setAttribute("error", "ID was exist");

            dispatcher.forward(req, resp);

        } else if (id == null || id.isEmpty()){

            session.setAttribute("error", "Please provided id");

            dispatcher.forward(req, resp);

        } else if (name == null || name.isEmpty()) {

            session.setAttribute("error", "Please input name");

            dispatcher.forward(req, resp);

        } else if (email == null || email.isEmpty()) {

            session.setAttribute("error", "Please input email");

            dispatcher.forward(req, resp);

        } else if (phone == null || phone.isEmpty()) {

            session.setAttribute("error", "Please input phone");

            dispatcher.forward(req, resp);

        } else {

            try {

                if( userDAO.insertUser(new User(id, name, email, phone, activatedValue)) == 1 ){

                    // Set success message

                    req.getSession().setAttribute("toastMessage", "User information added successfully!");

                    req.getSession().setAttribute("toastType", "success");

                } else {

                    // Set error message

                    req.getSession().setAttribute("toastMessage", "Failed to add user information. Please try again.");

                    req.getSession().setAttribute("toastType", "error");

                }

                resp.sendRedirect("/");

            } catch (SQLException e) {

                throw new RuntimeException(e);

            }

        }

    }

}

**DeleteUser.java**

package usermanagement.user;

import usermanagement.dao.UserDAO;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.SQLException;

@WebServlet("/delete\_user")

public class DeleteUser extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String id = req.getParameter("id");

        UserDAO userDAO = new UserDAO();

        try {

            if( userDAO.deleteUser(id) ){

                // Set success message

                req.getSession().setAttribute("toastMessage", "User information deleted successfully!");

                req.getSession().setAttribute("toastType", "success");

            } else {

                // Set error message

                req.getSession().setAttribute("toastMessage", "Failed to delete user information. Please try again.");

                req.getSession().setAttribute("toastType", "error");

            }

            resp.sendRedirect("/");

        } catch (SQLException e) {

            throw new RuntimeException(e);

        }

    }

}

**ListUser.java**

package usermanagement.user;

import java.io.IOException;

import java.util.List;

import usermanagement.dao.UserDAO;

import usermanagement.model.Manager;

import usermanagement.model.User;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

@WebServlet("/")

public class ListUser extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        HttpSession session = req.getSession();

        Manager loginManager = (Manager) session.getAttribute("manager");

        if( loginManager == null ){

            resp.sendRedirect("/login");

            return;

        }

        int page = 1;

        int recordsPerPage = 5;

        if (req.getParameter("page") != null)

            page = Integer.parseInt(req.getParameter("page"));

        UserDAO userDAO = new UserDAO();

        List<User> listUser = userDAO.selectAllUsers((page - 1) \* recordsPerPage, recordsPerPage);

        int noOfRecords = userDAO.getNoOfRecords();

        int noOfPages = (int)Math.ceil(noOfRecords \* 1.0 / recordsPerPage);

        req.setAttribute("listUser", listUser);

        req.setAttribute("noOfPages", noOfPages);

        req.setAttribute("currentPage", page);

        RequestDispatcher dispatcher = req.getRequestDispatcher("user\_list.jsp");

        dispatcher.forward(req, resp);

    }

}

**Update.java**

package usermanagement.user;

import usermanagement.dao.UserDAO;

import usermanagement.model.User;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.SQLException;

import java.util.Objects;

@WebServlet("/update\_user")

public class UpdateUser extends HttpServlet {

    String id = null;

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        id = req.getParameter("id");

        UserDAO userDAO = new UserDAO();

        User user = userDAO.selectUser(id);

        // to jsp

        req.setAttribute("user", user);

        RequestDispatcher dispatcher = req.getRequestDispatcher("user\_edit.jsp");

        dispatcher.forward(req, resp);

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        req.setCharacterEncoding("UTF-8");

        String name = req.getParameter("name");

        String email = req.getParameter("email");

        String phone = req.getParameter("phone");

        int activatedValue = (Objects.equals(req.getParameter("activated"), "true"))?1:0;

        UserDAO userDAO = new UserDAO();

        try {

            if (userDAO.updateUser(new User(id, name, email, phone, activatedValue))) {

                // Set success message

                req.getSession().setAttribute("toastMessage", "User information updated successfully!");

                req.getSession().setAttribute("toastType", "success");

            } else {

                // Set error message

                req.getSession().setAttribute("toastMessage", "Failed to update user information. Please try again.");

                req.getSession().setAttribute("toastType", "error");

            }

            resp.sendRedirect("/");

        } catch (SQLException e) {

            throw new RuntimeException(e);

        }

    }

}

### Manager

**DeleteManager.java**

package usermanagement.manager;

import usermanagement.dao.ManagerDAO;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.SQLException;

@WebServlet("/delete\_manager")

public class DeleteManager extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String id = req.getParameter("id");

        ManagerDAO managerDAO = new ManagerDAO();

        try {

            if( managerDAO.deleteManager(id) ){

                // Set success message

                req.getSession().setAttribute("toastMessage", "Manager information deleted successfully!");

                req.getSession().setAttribute("toastType", "success");

            } else {

                // Set error message

                req.getSession().setAttribute("toastMessage", "Failed to delete Manager information. Please try again.");

                req.getSession().setAttribute("toastType", "error");

            }

            resp.sendRedirect("/managerlist");

        } catch (SQLException e) {

            throw new RuntimeException(e);

        }

    }

}

**ListManager.java**

package usermanagement.manager;

import usermanagement.dao.ManagerDAO;

import usermanagement.model.Manager;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.util.List;

@WebServlet("/managerlist")

public class ListManager extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        int page = 1;

        int recordsPerPage = 5;

        if (req.getParameter("page") != null)

            page = Integer.parseInt(req.getParameter("page"));

        ManagerDAO managerDAO = new ManagerDAO();

        List<Manager> listManager = managerDAO.selectAllUsers((page - 1) \* recordsPerPage, recordsPerPage);

        int noOfRecords = managerDAO.getNoOfRecords();

        int noOfPages = (int)Math.ceil(noOfRecords \* 1.0 / recordsPerPage);

        req.setAttribute("listManager", listManager);

        req.setAttribute("noOfPages", noOfPages);

        req.setAttribute("currentPage", page);

        RequestDispatcher dispatcher = req.getRequestDispatcher("manager/manager\_list.jsp");

        dispatcher.forward(req, resp);

    }

}

**Login.java**

package usermanagement.manager;

import usermanagement.dao.ManagerDAO;

import usermanagement.model.Manager;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

import java.sql.SQLException;

@WebServlet("/login")

public class Login extends HttpServlet {

    ManagerDAO managerDAO = new ManagerDAO();

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        HttpSession session = req.getSession();

        Manager loginManager = (Manager) session.getAttribute("manager");

        if(loginManager != null){

            resp.sendRedirect("/");

            return;

        }

        resp.setContentType("text/html");

        RequestDispatcher dispatcher = req.getRequestDispatcher("manager/manager\_login.jsp");

        dispatcher.forward(req, resp);

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String username = req.getParameter("username");

        String password = req.getParameter("password");

        HttpSession session = req.getSession();

        RequestDispatcher dispatcher = req.getRequestDispatcher("manager/manager\_login.jsp");

        //send to jsp

        req.setAttribute("username",username);

        req.setAttribute("password",password);

        if(username == null || password == null){

            session.setAttribute("error", "Please input username and password");

            dispatcher.forward(req, resp);

        } else if (username.isEmpty()||password.isEmpty()) {

            session.setAttribute("error", "Username and password must not be empty");

            dispatcher.forward(req, resp);

        } else if (password.length() < 6) {

            session.setAttribute("error", "Password must be greater than 6 characters");

            dispatcher.forward(req, resp);

        } else if (username.equals("admin") && password.equals("123qwe")) {

            session.setAttribute("manager", new Manager("admin", "123qwe"));

            req.getSession().setAttribute("toastMessage", "Admin Login successfully!");

            req.getSession().setAttribute("toastType", "success");

            resp.sendRedirect("/");

        } else {

            try {

                Manager manager = managerDAO.get(username,password);

                if (manager != null) {

                    session.setAttribute("manager", manager);

                    req.getSession().setAttribute("toastMessage", "Manager Login successfully!");

                    req.getSession().setAttribute("toastType", "success");

                    resp.sendRedirect("/");

                } else {

                    session.setAttribute("error", "Invalid username or password");

                    dispatcher.forward(req, resp);

                }

            } catch (SQLException e) {

                throw new RuntimeException(e);

            }

        }

    }

}

**Logout.java**

package usermanagement.manager;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

@WebServlet("/logout")

public class Logout extends HttpServlet {

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        HttpSession session = req.getSession();

        session.removeAttribute("manager");

        req.getSession().setAttribute("toastMessage", "Logout successfully!");

        req.getSession().setAttribute("toastType", "success");

        resp.sendRedirect("/login");

    }

}

**Register.java**

package usermanagement.manager;

import usermanagement.dao.ManagerDAO;

import usermanagement.model.Manager;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

import java.sql.SQLException;

@WebServlet("/register")

public class Register extends HttpServlet {

    ManagerDAO managerDAO = new ManagerDAO();

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        HttpSession session = req.getSession();

        Manager loginManager = (Manager) session.getAttribute("manager");

        if(loginManager != null){

            resp.sendRedirect("/");

            return;

        }

        resp.setContentType("text/html");

        RequestDispatcher dispatcher = req.getRequestDispatcher("manager/manager\_register.jsp");

        dispatcher.forward(req, resp);

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String id = req.getParameter("id");

        String fullname = req.getParameter("fullname");

        String email = req.getParameter("email");

        String password = req.getParameter("password");

        String passwordConfirm = req.getParameter("passwordConfirm");

        HttpSession session = req.getSession();

        RequestDispatcher dispatcher = req.getRequestDispatcher("manager/manager\_register.jsp");

        req.getParameterMap().forEach((key, values) -> {

            req.setAttribute(key, values[0]);

        });

        if (id == null ||fullname == null || email == null || password == null || passwordConfirm == null){

            session.setAttribute("error", "Please provide enough information");

            dispatcher.forward(req, resp);

        } else if (id.isEmpty() || fullname.isEmpty() || email.isEmpty()|| password.isEmpty() || passwordConfirm.isEmpty()){

            session.setAttribute("error", "Please fill out all the form fields");

            dispatcher.forward(req, resp);

        } else if (fullname.equals("admin")) {

            session.setAttribute("error", "Please choose another full name");

            dispatcher.forward(req, resp);

        } else {

            try {

                if (managerDAO.getAllId().contains(id)) {

                    session.setAttribute("error", "ID was existed");

                    dispatcher.forward(req, resp);

                } else if (password.length() < 6 ){

                    session.setAttribute("error", "Password must be or more than 6 letters");

                    dispatcher.forward(req, resp);

                } else if (!password.equals(passwordConfirm)) {

                    session.setAttribute("error", "Password does not match");

                    dispatcher.forward(req, resp);

                } else {

                    Manager account = new Manager(id, fullname, email, password);

                    try {

                        if (managerDAO.add(account)){

                            req.getSession().setAttribute("toastMessage", "Register successfully!");

                            req.getSession().setAttribute("toastType", "success");

                            resp.sendRedirect("/login");

                        } else {

                            session.setAttribute("error", "Creat account Failed");

                            dispatcher.forward(req, resp);

                        }

                    } catch (SQLException e) {

                        throw new RuntimeException(e);

                    }

                }

            } catch (SQLException e) {

                throw new RuntimeException(e);

            }

        }

    }

}

**UpdateManager.java**

package usermanagement.manager;

import usermanagement.dao.ManagerDAO;

import usermanagement.model.Manager;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.sql.SQLException;

@WebServlet("/update\_manager")

public class UpdateManager extends HttpServlet {

    String id = null;

    @Override

    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        id = req.getParameter("id");

        ManagerDAO managerDAO = new ManagerDAO();

        Manager manager = managerDAO.selectManager(id);

        // to jsp

        req.setAttribute("manager", manager);

        RequestDispatcher dispatcher = req.getRequestDispatcher("manager/manager\_edit.jsp");

        dispatcher.forward(req, resp);

    }

    @Override

    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String fullName = req.getParameter("fullName");

        String email = req.getParameter("email");

        String password = req.getParameter("password");

        ManagerDAO managerDAO = new ManagerDAO();

        try {

            if (managerDAO.updateManager(new Manager(id, fullName, email, password))) {

                // Set success message

                req.getSession().setAttribute("toastMessage", "Manager information updated successfully!");

                req.getSession().setAttribute("toastType", "success");

            } else {

                // Set error message

                req.getSession().setAttribute("toastMessage", "Failed to update manager information. Please try again.");

                req.getSession().setAttribute("toastType", "error");

            }

            resp.sendRedirect("/managerlist");

        } catch (SQLException e) {

            throw new RuntimeException(e);

        }

    }

}

### DAO

**ManagerDAO.java**

package usermanagement.dao;

import usermanagement.db.DBConnect;

import usermanagement.model.Manager;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

public class ManagerDAO {

    Connection conn;

    private int noOfRecords;

    Statement stmt;

    public int getNoOfRecords() { return noOfRecords; }

    private static final String INSERT\_MANAGERS\_SQL = "INSERT INTO managers" + "  (id, fullName, email, password) VALUES  (?, ?, ?, ?);";

    private static final String SELECTID\_MANAGERS\_SQL = "SELECT id FROM managers;";

    private static final String DELETE\_MANAGERS\_SQL = "delete from managers where id = ?;";

    private static final String SELECT\_MANAGERS\_BY\_ID = "select fullName,email,password from managers where id =?;";

    private static final String UPDATE\_MANAGERS\_SQL = "update managers set fullName = ?,email= ?, password = ? where id = ?;";

    private static final String SELECT\_ALL\_MANAGERS = "SELECT \* FROM managers;";

    public ManagerDAO(){conn = DBConnect.getConn();}

    public boolean add(Manager manager) throws SQLException {

        PreparedStatement preparedStatement = conn.prepareStatement(INSERT\_MANAGERS\_SQL);

        preparedStatement.setString(1, manager.getId());

        preparedStatement.setString(2, manager.getFullName());

        preparedStatement.setString(3, manager.getEmail());

        preparedStatement.setString(4, manager.getPassword());

        return preparedStatement.executeUpdate() > 0;

    }

    public List<String> getAllId() throws SQLException{

        PreparedStatement preparedStatement = conn.prepareStatement(SELECTID\_MANAGERS\_SQL);

        ResultSet resultSet = preparedStatement.executeQuery();

        List<String> id = new ArrayList<>();

        while (resultSet.next()) {

            id.add(resultSet.getString(1));

        }

        resultSet.close();

        return id;

    }

    public Manager get(String email, String password) throws SQLException {

        PreparedStatement preparedStatement = conn.prepareStatement("SELECT \* from Managers m where m.email = ? and m.password = ?");

        preparedStatement.setString(1, email);

        preparedStatement.setString(2, password);

        ResultSet resultSet = preparedStatement.executeQuery();

        Manager manager = null;

        if (resultSet.next()) {

            manager = new Manager(

                    resultSet.getString(1),

                    resultSet.getString(2),

                    resultSet.getString(3),

                    resultSet.getString(4)

            );

        }

        return manager;

    }

    public List<Manager> selectAllUsers(int offset, int noOfRecords){

        List<Manager> list = new ArrayList<>();

        String query = "select SQL\_CALC\_FOUND\_ROWS \* from managers limit " + offset + ", " + noOfRecords;

        try{

            stmt = conn.createStatement();

            ResultSet rs = stmt.executeQuery(query);

            while (rs.next()) {

                String ID = rs.getString("id");

                String fullName = rs.getString("fullName");

                String email = rs.getString("email");

                String password = rs.getString("password");

                Manager mnager = new Manager(ID,fullName,email,password);

                list.add(mnager);

            }

            rs.close();

            rs = stmt.executeQuery("SELECT FOUND\_ROWS()");

            if (rs.next())

                this.noOfRecords = rs.getInt(1);

        } catch (SQLException e) {

            printSQLException(e);

        }

        finally

        {

            try {

                if (stmt != null)

                    stmt.close();

                if (conn != null)

                    conn.close();

            }

            catch (SQLException e) {

                e.printStackTrace();

            }

        }

        return list;

    }

    private void printSQLException(SQLException ex) {

        for (Throwable e : ex) {

            if (e instanceof SQLException) {

                e.printStackTrace(System.err);

                System.err.println("SQLState: " + ((SQLException) e).getSQLState());

                System.err.println("Error Code: " + ((SQLException) e).getErrorCode());

                System.err.println("Message: " + e.getMessage());

                Throwable t = ex.getCause();

                while (t != null) {

                    System.out.println("Cause: " + t);

                    t = t.getCause();

                }

            }

        }

    }

    public boolean deleteManager(String id) throws SQLException {

        boolean rowDeleted;

        PreparedStatement preparedStatement = conn.prepareStatement(DELETE\_MANAGERS\_SQL);

        preparedStatement.setString(1, id);

        rowDeleted = preparedStatement.executeUpdate() > 0;

        return rowDeleted;

    }

    public Manager selectManager(String id) {

        Manager manager = null;

        try (PreparedStatement preparedStatement = conn.prepareStatement(SELECT\_MANAGERS\_BY\_ID);) {

            preparedStatement.setString(1, id);

            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()) {

                String fullName = rs.getString("fullName");

                String email = rs.getString("email");

                String password = rs.getString("password");

                manager = new Manager(id, fullName, email, password);

            }

        } catch (SQLException e) {

            printSQLException(e);

        }

        return manager;

    }

    public boolean updateManager(Manager manager) throws SQLException {

        PreparedStatement preparedStatement = conn.prepareStatement(UPDATE\_MANAGERS\_SQL);

        preparedStatement.setString(1, manager.getFullName());

        preparedStatement.setString(2, manager.getEmail());

        preparedStatement.setString(3, manager.getPassword());

        preparedStatement.setString(4, manager.getId());

        return preparedStatement.executeUpdate() > 0;

    }

}

**UserDAO.java**

package usermanagement.dao;

import java.sql.\*;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import usermanagement.db.DBConnect;

import usermanagement.model.User;

public class UserDAO {

    private Connection conn;

    private int noOfRecords;

    Statement stmt;

    public int getNoOfRecords() { return noOfRecords; }

    public UserDAO()  {conn = DBConnect.getConn();}

    private static final String INSERT\_USERS\_SQL = "INSERT INTO users" + "  (id, name, email, phone, IsActive) VALUES  (?, ?, ?, ?, ?);";

    private static final String SELECT\_USER\_BY\_ID = "select id,name,email,phone,IsActive from users where id =?;";

    private static final String SELECT\_ALL\_USERS = "select \* from users;";

    private static final String DELETE\_USERS\_SQL = "delete from users where id = ?;";

    private static final String UPDATE\_USERS\_SQL = "update users set name = ?,email= ?, phone = ?, IsActive = ? where id = ?;";

    private static final String SELECT\_ID\_USERS = "SELECT id FROM users;";

    public Integer insertUser(User user) throws SQLException {

         PreparedStatement preparedStatement = conn.prepareStatement(INSERT\_USERS\_SQL);

         preparedStatement.setString(1, user.getUserID());

         preparedStatement.setString(2, user.getEmail());

         preparedStatement.setString(3, user.getEmail());

         preparedStatement.setString(4, user.getMobilePhone());

         preparedStatement.setInt(5, user.getActive());

         return preparedStatement.executeUpdate();

    }

    public User selectUser(String id) {

        User user = null;

        try (PreparedStatement preparedStatement = conn.prepareStatement(SELECT\_USER\_BY\_ID);) {

            preparedStatement.setString(1, id);

            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()) {

                String userName = rs.getString("name");

                String email = rs.getString("email");

                String mobilePhone = rs.getString("phone");

                int isActive = rs.getInt("IsActive");

                user = new User(id, userName, email, mobilePhone, isActive);

            }

        } catch (SQLException e) {

            printSQLException(e);

        }

        return user;

    }

    public List<String> selectAllId() {

        List<String> listID = new ArrayList<>();;

        try (PreparedStatement preparedStatement = conn.prepareStatement(SELECT\_ID\_USERS);) {

            ResultSet rs = preparedStatement.executeQuery();

            while (rs.next()) {

                String id = rs.getString("id");

                listID.add(id);

            }

            rs.close();

        } catch (SQLException e) {

            printSQLException(e);

        }

        return listID;

    }

    public List<User> selectAllUsers(int offset, int noOfRecords) {

        List<User> list = new ArrayList<>();

        String query = "select SQL\_CALC\_FOUND\_ROWS \* from users limit " + offset + ", " + noOfRecords;

        try{

            stmt = conn.createStatement();

            ResultSet rs = stmt.executeQuery(query);

            while (rs.next()) {

                String userID = rs.getString("id");

                String userName = rs.getString("name");

                String email = rs.getString("email");

                String mobilePhone = rs.getString("phone");

                int isActive = rs.getInt("IsActive");

                Date dateCreated = rs.getTimestamp("DateCreated");

                User user = new User(userID,userName,email,mobilePhone,isActive, dateCreated);

                list.add(user);

            }

            rs.close();

            rs = stmt.executeQuery("SELECT FOUND\_ROWS()");

            if (rs.next())

                this.noOfRecords = rs.getInt(1);

        } catch (SQLException e) {

            printSQLException(e);

        }

        finally

        {

            try {

                if (stmt != null)

                    stmt.close();

                if (conn != null)

                    conn.close();

            }

            catch (SQLException e) {

                e.printStackTrace();

            }

        }

        return list;

    }

    public boolean deleteUser(String id) throws SQLException {

        boolean rowDeleted;

        PreparedStatement preparedStatement = conn.prepareStatement(DELETE\_USERS\_SQL);

        preparedStatement.setString(1, id);

        rowDeleted = preparedStatement.executeUpdate() > 0;

        return rowDeleted;

    }

    public boolean updateUser(User user) throws SQLException {

        PreparedStatement preparedStatement = conn.prepareStatement(UPDATE\_USERS\_SQL);

        preparedStatement.setString(1, user.getUserName());

        preparedStatement.setString(2, user.getEmail());

        preparedStatement.setString(3, user.getMobilePhone());

        preparedStatement.setInt(4, user.getActive());

        preparedStatement.setString(5, user.getUserID());

        return preparedStatement.executeUpdate() > 0;

    }

    private void printSQLException(SQLException ex) {

        for (Throwable e : ex) {

            if (e instanceof SQLException) {

                e.printStackTrace(System.err);

                System.err.println("SQLState: " + ((SQLException) e).getSQLState());

                System.err.println("Error Code: " + ((SQLException) e).getErrorCode());

                System.err.println("Message: " + e.getMessage());

                Throwable t = ex.getCause();

                while (t != null) {

                    System.out.println("Cause: " + t);

                    t = t.getCause();

                }

            }

        }

    }

}

## Frontend

### Component

**Allcss.jsp**

<link

        href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"

        rel="stylesheet"

        integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"

        crossorigin="anonymous">

<link rel="stylesheet"

      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css"

      integrity="sha512-Fo3rlrZj/k7ujTnHg4CGR2D7kSs0v4LLanw2qksYuRlEzO+tcaEPQogQ0KaoGN26/zrn20ImR1DfuLWnOo7aBA=="

      crossorigin="anonymous" referrerpolicy="no-referrer" />

<script

        src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

        crossorigin="anonymous"></script>

<!-- Toastr CSS -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.css">

<!-- jQuery -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<!-- Toastr JavaScript -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

<style>

    html, body {

        height: 100%;

        margin: 0;

        padding: 0;

    }

</style>

**Navbar.jsp**

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<nav class="navbar navbar-expand-lg navbar-dark bg-success">

    <div class="container-fluid">

        <a class="navbar-brand " href="/">

            <i class="fas fa-home"></i> User Management

        </a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"

            data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"

            aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarSupportedContent">

            <ul class="navbar-nav me-auto mb-2 mb-lg-0">

                <li class="nav-item"><a class="nav-link active" href="/">User List</a></li>

                <c:if test="${sessionScope.manager.getFullName().trim() == 'admin'}">

                    <li class="nav-item"><a class="nav-link active" href="/managerlist">Manager List</a></li>

                </c:if>

            </ul>

            <form class="form-inline">

                <div class="bg-light p-1 rounded">

                    <span class="fw-bold">Hi</span>

                    <span class="text-danger fw-bold">

                        <c:out value="${sessionScope.manager.getFullName()}"/>

                        <c:if test="${sessionScope.manager.getFullName().trim() != 'admin'}">

                            (<c:out value="${sessionScope.manager.getId()}"/>)

                        </c:if>

                    </span> |

                    <a href="/logout" class="text-decoration-none text-dark">Logout <i class="fas fa-sign-out-alt"

                            style="color: #000000;"></i></a>

                </div>

            </form>

        </div>

    </div>

</nav>

**toastMessage.jsp**

<!-- Toastr CSS -->

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.css">

<!-- jQuery -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<!-- Toastr JavaScript -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<c:if test="${not empty sessionScope.toastMessage}">

  <script>

    document.addEventListener('DOMContentLoaded', function () {

      toastr.options = {

        closeButton: true,

        progressBar: true,

        timeOut: 3000,

        extendedTimeOut: 2000,

        positionClass: 'toast-top-right',

        preventDuplicates: true,

        showDuration: 300,

        hideDuration: 300

      };

      toastr.${sessionScope.toastType}("<c:out value="${sessionScope.toastMessage}"/>");

    });

  </script>

</c:if>

<c:remove var="toastMessage" scope="session"/>

### Manager

**Manager\_edit.jsp**

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false"%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

    <title>Edit User</title>

    <%@include file="/component/allcss.jsp"%>

</head>

<body>

<header>

    <%@include file="/component/navbar.jsp"%>

</header>

<br>

<div class="container col-md-5">

    <div class="card">

        <div class="card-body">

            <form action="update\_manager" method="post">

                <caption>

                    <h2>Edit Manager</h2>

                </caption>

                <fieldset class="form-group">

                    <label>Manager Name</label> <input type="text"

                                                    value="<c:out value="${requestScope.manager.getFullName()}"/>"

                                                    class="form-control"

                                                    name="fullName"

                                                    required="required">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>Manager Email</label> <input type="text"

                                                     value="<c:out value="${requestScope.manager.getEmail()}"/>"

                                                     class="form-control"

                                                     name="email">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>Manager Password</label> <input type="text"

                                                     value="<c:out value="${requestScope.manager.getPassword()}"/>"

                                                     class="form-control"

                                                     name="password">

                </fieldset>

                <br>

                <button type="submit" class="btn btn-success">Save</button>

            </form>

        </div>

    </div>

</div>

</body>

</html>

**Manager\_list.jsp**

<%@ page pageEncoding="utf-8" %>

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false"%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>User Management</title>

    <%@include file="../component/allcss.jsp"%>

</head>

<body>

<header>

    <%@include file="../component/navbar.jsp"%>

</header>

<%@include file="../component/toastMessage.jsp"%>

<div id="toastContainer"></div>

<div class="container my-4 text-center">

    <h1 class="text-center">List of Manager</h1>

    <!-- Manager List -->

    <table class="table table-striped table-bordered" id="tableId" border="1" cellpadding="5" cellspacing="5">

        <thead>

        <tr>

            <th>Manager ID</th>

            <th>Manager Name</th>

            <th>Email</th>

            <th>Password</th>

            <th>Actions</th>

        </tr>

        </thead>

        <tbody>

        <c:forEach var="manager" items="${requestScope.listManager}">

            <tr>

                <td><c:out value="${manager.getId()}"/></td>

                <td><c:out value="${manager.getFullName()}"/></td>

                <td><c:out value="${manager.getEmail()}"/></td>

                <td><c:out value="${manager.getPassword()}"/></td>

                <td>

                    <a href="/update\_manager?id=${manager.getId()}" class="btn btn-warning">

                        <i class="fas fa-edit"></i> Edit

                    </a>

                    <a href="/delete\_manager?id=${manager.getId()}" class="btn btn-danger deleteButton">

                        <i class="fas fa-trash"></i> Delete

                    </a>

                </td>

            </tr>

        </c:forEach>

        </tbody>

    </table>

    <nav aria-label="Page navigation">

        <ul class="pagination justify-content-center">

            <!-- For displaying Previous link except for the 1st page -->

            <c:if test="${requestScope.currentPage != 1}">

                <li class="page-item">

                    <a class="page-link" href="/managerlist?page=${requestScope.currentPage - 1}" aria-label="Previous">

                        <span aria-hidden="true">&laquo;</span>

                    </a>

                </li>

            </c:if>

            <!-- For displaying Page numbers. The when condition does not display a link for the current page -->

            <c:forEach begin="1" end="${requestScope.noOfPages}" var="i">

                <c:choose>

                    <c:when test="${requestScope.currentPage eq i}">

                        <li class="page-item active" aria-current="page">

                            <a class="page-link" href="#">${i}</a>

                        </li>

                    </c:when>

                    <c:otherwise>

                        <li class="page-item">

                            <a class="page-link" href="/managerlist?page=${i}">${i}</a>

                        </li>

                    </c:otherwise>

                </c:choose>

            </c:forEach>

            <!-- For displaying Next link -->

            <c:if test="${currentPage lt noOfPages}">

                <li class="page-item">

                    <a class="page-link" href="/managerlist?page=${currentPage + 1}" aria-label="Next">

                        <span aria-hidden="true">&raquo;</span>

                    </a>

                </li>

            </c:if>

        </ul>

    </nav>

</div>

</body>

</html>

**Manager\_login.jsp**

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false" %>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Login Page</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</head>

<body>

<header>

</header>

<%@include file="../component/toastMessage.jsp"%>

<div class="container">

    <div class="row justify-content-center">

        <div class="col-md-6 col-lg-5">

            <h3 class="text-center text-secondary mt-5 mb-3">Manager Login</h3>

            <form action="login" method="post" class="border rounded w-100 mb-5 mx-auto px-3 pt-3 bg-light">

                <div class="form-group">

                    <label for="username">Username</label>

                    <input value="<c:out value="${requestScope.username}"/>" id="username" name="username" type="text" class="form-control" placeholder="Username">

                </div>

                <div class="form-group">

                    <label for="password">Password</label>

                    <input value="<c:out value="${requestScope.password}"/>" id="password" name="password" type="password" class="form-control" placeholder="Password">

                </div>

                <div class="form-group">

                    <button class="btn btn-success px-5">Login</button>

                </div>

                <c:if test="${not empty sessionScope.error}">

                    <div class="form-group">

                        <div class="alert alert-danger"><c:out value="${sessionScope.error}"/></div>

                    </div>

                </c:if>

                <c:remove var="error" scope="session"/>

                <div class="form-group">

                    <p>Haven't you have an account? <a href="/register">Register</a></p>

                </div>

            </form>

        </div>

    </div>

</div>

</body>

**Manager\_register.jsp**

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false" %>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Register</title>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <link rel="stylesheet"

        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.2/css/all.min.css    ">

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

    <style>

        p.item {

            padding: 16px;

            text-align: center;

            border-radius: 8px;

            color: white;

            background-color: green;

        }

    </style>

</head>

<body class="bg-secondary">

    <h3 class="text-center my-5 text-light">Manager Registration</h3>

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md-10 col-lg-8 col-xl-5">

                <div class="border p-3 rounded bg-light">

                    <form method="post" action="register">

                        <div class="form-group">

                            <label for="id">ID</label>

                            <div class="input-group mb-3">

                                <div class="input-group-prepend">

                                    <span class="input-group-text">

                                        <i class="fas fa-fingerprint"></i>

                                    </span>

                                </div>

                                <input  id="id" type="text" class="form-control"

                                        placeholder="ID Manager"

                                        name="id"

                                        value="<c:out value="${requestScope.id}"/>">

                            </div>

                        </div>

                        <div class="form-group">

                            <label for="name">Fullname</label>

                            <div class="input-group mb-3">

                                <div class="input-group-prepend">

                                    <span class="input-group-text">

                                        <i class="fas fa-user"></i>

                                    </span>

                                </div>

                                <input id="name" type="text" class="form-control"

                                        placeholder="Your Name"

                                        name="fullname"

                                        value="<c:out value="${requestScope.fullname}"/>">

                            </div>

                        </div>

                        <div class="form-group">

                            <label for="email">Email</label>

                            <div class="input-group mb-3">

                                <div class="input-group-prepend">

                                    <span class="input-group-text">

                                        <i class="fas fa-user"></i>

                                    </span>

                                </div>

                                <input  id="email" type="text" class="form-control"

                                        placeholder="Email"

                                        name="email"

                                        value="<c:out value="${requestScope.email}"/>">

                            </div>

                        </div>

                        <div class="form-group">

                            <label for="password">Password</label>

                            <div class="input-group mb-3">

                                <div class="input-group-prepend">

                                    <span class="input-group-text">

                                        <i class="fas fa-lock"></i>

                                    </span>

                                </div>

                                <input id="password" type="password" class="form-control"

                                    placeholder="Password"

                                    name="password"

                                    value="<c:out value="${requestScope.password}"/>">

                            </div>

                        </div>

                        <div class="form-group">

                            <label for="password-confirm">Confirm Password</label>

                            <div class="input-group mb-3">

                                <div class="input-group-prepend">

                                    <span class="input-group-text">

                                        <i class="fas fa-lock"></i>

                                    </span>

                                </div>

                                <input id="password-confirm" type="password" class="form-control"

                                    placeholder="Password"

                                    name="passwordConfirm"

                                    value="<c:out value="${requestScope.passwordConfirm}"/>">

                            </div>

                        </div>

                        <c:if test="${not empty sessionScope.error}">

                            <div class="form-group">

                                <div class="alert alert-danger">

                                    <button type="button" class="close"

                                        data-dismiss="alert">&times;</button>

                                    <c:out value="${sessionScope.error}" />

                                </div>

                            </div>

                        </c:if>

                        <c:remove var="error" scope="session" />

                        <div class="form-group">

                            <button class="btn btn-success px-5" type="submit">Register</button>

                        </div>

                        <div class="form-group">

                            <p>Already have an account? <a href="/login">Login now!</a></p>

                        </div>

                    </form>

                </div>

            </div>

        </div>

    </div>

</body>

</html>

### User

**User\_add.jsp**

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false"%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

    <title>User Management Application</title>

    <%@include file="component/allcss.jsp"%>

</head>

<body>

<header>

    <%@include file="component/navbar.jsp"%>

</header>

<br>

<div class="container col-md-5">

    <div class="card">

        <div class="card-body">

            <form action="add" method="post">

                <caption>

                    <h2>Add User</h2>

                </caption>

                <fieldset class="form-group">

                    <label>User ID</label> <input   type="text" class="form-control"

                                                    name="id"

                                                    value="<c:out value="${requestScope.user.getUserID()}"/>">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>User Name</label> <input type="text" class="form-control"

                                                    name="name"

                                                    value="<c:out value="${requestScope.user.getUserName()}"/>">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>User Email</label> <input type="text" class="form-control"

                                                     name="email"

                                                     value="<c:out value="${requestScope.user.getEmail()}"/>">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>User Phone</label> <input type="text" class="form-control"

                                                       name="phone"

                                                       value="<c:out value="${requestScope.user.getMobilePhone()}"/>">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>Activated</label>

                    <div class="form-check">

                        <input type="radio"

                               id="activatedCheckbox"

                               name="activated"

                               value="true"

                               class="form-check-input"

                               <c:if test="${requestScope.user.getActive()  == 1}">checked</c:if>>

                        <label class="form-check-label" for="activatedCheckbox">Activated</label>

                    </div>

                    <div class="form-check">

                        <input type="radio"

                               id="inactivatedCheckbox"

                               name="activated"

                               value="false"

                               class="form-check-input"

                               <c:if test="${requestScope.user.getActive() == 0}">checked</c:if>>

                        <label class="form-check-label" for="inactivatedCheckbox">Inactivated</label>

                    </div>

                </fieldset>

                <br>

                <c:if test="${not empty sessionScope.error}">

                    <div class="alert alert-danger"><c:out value="${sessionScope.error}"/></div>

                </c:if>

                <c:remove var="error" scope="session"/>

                <br>

                <button type="submit" class="btn btn-success">Submit</button>

                <br>

            </form>

        </div>

    </div>

</div>

</body>

</html>

**User\_edit.jsp**

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false"%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<html>

<head>

    <title>Edit User</title>

    <%@include file="component/allcss.jsp"%>

</head>

<body>

<header>

    <%@include file="component/navbar.jsp"%>

</header>

<br>

<div class="container col-md-5">

    <div class="card">

        <div class="card-body">

            <form action="update\_user" method="post">

                <caption>

                    <h2>Edit User</h2>

                </caption>

                <fieldset class="form-group">

                    <label>User Name</label> <input type="text"

                                                    value="<c:out value="${requestScope.user.getUserName()}"/>"

                                                    class="form-control"

                                                    name="name"

                                                    required="required">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>User Email</label> <input type="text"

                                                     value="<c:out value="${requestScope.user.getEmail()}"/>"

                                                     class="form-control"

                                                     name="email">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>User Phone</label> <input type="text"

                                                     value="<c:out value="${requestScope.user.getMobilePhone()}"/>"

                                                     class="form-control"

                                                     name="phone">

                </fieldset>

                <br>

                <fieldset class="form-group">

                    <label>Activated</label>

                    <div class="form-check">

                        <input type="radio"

                               id="activatedCheckbox"

                               name="activated"

                               value="true"

                               class="form-check-input"

                               <c:if test="${requestScope.user.getActive()  == 1}">checked</c:if>>

                        <label class="form-check-label" for="activatedCheckbox">Activated</label>

                    </div>

                    <div class="form-check">

                        <input type="radio"

                               id="inactivatedCheckbox"

                               name="activated"

                               value="false"

                               class="form-check-input"

                               <c:if test="${requestScope.user.getActive() == 0}">checked</c:if>>

                        <label class="form-check-label" for="inactivatedCheckbox">Inactivated</label>

                    </div>

                </fieldset>

                <br>

                <button type="submit" class="btn btn-success">Save</button>

            </form>

        </div>

    </div>

</div>

</body>

</html>

**User\_list.jsp**

<%@ page pageEncoding="utf-8" %>

<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>

<%@ page isELIgnored="false"%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<!DOCTYPE html>

<html>

<head>

    <meta charset="UTF-8">

    <title>User Management</title>

    <%@include file="component/allcss.jsp"%>

</head>

<body>

<header>

    <%@include file="component/navbar.jsp"%>

</header>

    <%@include file="component/toastMessage.jsp"%>

<div id="toastContainer"></div>

    <div class="container my-4 text-center">

        <h1 class="text-center">List of Users</h1>

        <div class="container text-left">

            <a href="<%=request.getContextPath()%>/add" class="btn btn-success m-2">Add

                New User</a>

        </div>

        <!-- User List -->

        <table class="table table-striped table-bordered" id="tableId" border="1" cellpadding="5" cellspacing="5">

            <thead>

            <tr>

                <th>User ID</th>

                <th>User Name</th>

                <th>Email</th>

                <th>Mobile Phone</th>

                <th>Active</th>

                <th>Date Created</th>

                <th>Actions</th>

            </tr>

            </thead>

            <tbody>

            <c:forEach var="user" items="${requestScope.listUser}">

                <tr>

                    <td><c:out value="${user.getUserID()}"/></td>

                    <td><c:out value="${user.getUserName()}"/></td>

                    <td><c:out value="${user.getEmail()}"/></td>

                    <td><c:out value="${user.getMobilePhone()}"/></td>

                    <c:choose>

                        <c:when test="${user.getActive() == 1}">

                            <td><i class="fas fa-check-circle fa-lg" style="color: #2ed30d;"></i></td>

                        </c:when>

                        <c:otherwise>

                            <td><i class="fas fa-times-circle fa-lg" style="color: #f50000;"></i></td>

                        </c:otherwise>

                    </c:choose>

                    <td><c:out value="${user.getDateCreated()}"/></td>

                    <td>

                        <a href="/update\_user?id=${user.userID}" class="btn btn-warning">

                            <i class="fas fa-edit"></i> Edit

                        </a>

                        <a href="/delete\_user?id=${user.userID}" class="btn btn-danger deleteButton">

                            <i class="fas fa-trash"></i> Delete

                        </a>

                    </td>

                </tr>

            </c:forEach>

            </tbody>

        </table>

        <nav aria-label="Page navigation">

            <ul class="pagination justify-content-center">

                <!-- For displaying Previous link except for the 1st page -->

                <c:if test="${requestScope.currentPage != 1}">

                    <li class="page-item">

                        <a class="page-link" href="?page=${requestScope.currentPage - 1}" aria-label="Previous">

                            <span aria-hidden="true">&laquo;</span>

                        </a>

                    </li>

                </c:if>

                <!-- For displaying Page numbers. The when condition does not display a link for the current page -->

                <c:forEach begin="1" end="${requestScope.noOfPages}" var="i">

                    <c:choose>

                        <c:when test="${requestScope.currentPage eq i}">

                            <li class="page-item active" aria-current="page">

                                <a class="page-link" href="#">${i}</a>

                            </li>

                        </c:when>

                        <c:otherwise>

                            <li class="page-item">

                                <a class="page-link" href="?page=${i}">${i}</a>

                            </li>

                        </c:otherwise>

                    </c:choose>

                </c:forEach>

                <!-- For displaying Next link -->

                <c:if test="${currentPage lt noOfPages}">

                    <li class="page-item">

                        <a class="page-link" href="?page=${currentPage + 1}" aria-label="Next">

                            <span aria-hidden="true">&raquo;</span>

                        </a>

                    </li>

                </c:if>

            </ul>

        </nav>

    </div>

</body>

</html>

# CHƯƠNG 3: Database

--

-- Database: `process1`

--

-- --------------------------------------------------------

--

-- Table structure for table `managers`

--

CREATE TABLE `managers` (

  `id` varchar(255) NOT NULL,

  `fullName` varchar(255) DEFAULT NULL,

  `email` varchar(255) DEFAULT NULL,

  `password` varchar(255) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `managers`

--

INSERT INTO `managers` (`id`, `fullName`, `email`, `password`) VALUES

('M001', 'John Doe', 'johndoe@example.com', 'password1234566785'),

('M002', 'Jane Smith', 'janesmith@example.com', 'password456123'),

('M003', 'Michael Johnson', 'michaeljohnson@example.com', 'password789'),

('M004', 'Emily Davis', 'emilydavis@example.com', 'passwordabc'),

('M005', 'David Wilson', 'davidwilson@example.com', 'passworddef'),

('M006', 'Sarah Anderson', 'sarahanderson@example.com', 'passwordghi'),

('M007', 'Robert Thompson', 'robertthompson@example.com', 'passwordjkl'),

('M008', 'Jessica Lee', 'jessicalee@example.com', 'passwordmno'),

('M009', 'Matthew Clark', 'matthewclark@example.com', 'passwordpqr'),

('M010', 'Olivia Turner', 'oliviaturner@example.com', 'passwordstu'),

('M011', 'Alex Johnson', 'alexjohnson@example.com', 'password123'),

('M012', 'Samantha Davis', 'samanthadavis@example.com', 'password456'),

('M013', 'Daniel Wilson', 'danielwilson@example.com', 'password789'),

('M014', 'Oliver Thompson', 'oliverthompson@example.com', 'passwordabc'),

('M015', 'Sophia Lee', 'sophialee@example.com', 'passworddef'),

('M016', 'Jacob Wilson', 'jacobwilson@example.com', 'password123'),

('M017', 'Isabella Anderson', 'isabellaanderson@example.com', 'password456'),

('M018', 'Ethan Thompson', 'ethanthompson@example.com', 'password789'),

('M019', 'Mia Lee', 'mialee@example.com', 'passwordabc'),

('M020', 'William Johnson', 'williamjohnson@example.com', 'passworddef'),

('M021', 'Ava Davis', 'avadavis@example.com', 'passwordghi'),

('M022', 'James Wilson', 'jameswilson@example.com', 'passwordjkl'),

('M023', 'Charlotte Thompson', 'charlottethompson@example.com', 'passwordmno789'),

('M024', 'Benjamin Lee', 'benjaminlee@example.com', 'passwordpqr789456'),

('M025', 'Harper Clark', 'harperclark@example.com', 'passwordstu123456');

-- --------------------------------------------------------

--

-- Table structure for table `users`

--

CREATE TABLE `users` (

  `id` varchar(50) NOT NULL,

  `name` varchar(50) NOT NULL,

  `email` varchar(100) NOT NULL,

  `phone` varchar(15) NOT NULL,

  `IsActive` tinyint(1) NOT NULL DEFAULT 1,

  `DateCreated` datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `users`

--

INSERT INTO `users` (`id`, `name`, `email`, `phone`, `IsActive`, `DateCreated`) VALUES

('521H0489', 'Hồ Hữu An', '521H0489@student.tdtu.edu.vn', '0906881769', 1, '2023-10-04 16:09:24'),

('user001', 'John Doe', 'john.doe@example.com', '1234567890', 1, '2023-10-03 14:25:46'),

('user002', 'Jane Smith', 'jane.smith@example.com', '9876543210', 1, '2023-10-03 14:25:46'),

('user003', 'Mike Johnson', 'mike.johnson@example.com', '5555555555', 1, '2023-10-03 14:25:46'),

('user004', 'Emily Johnson', 'emily.johnson@example.com', '5551234567', 0, '2023-10-03 14:26:41'),

('user005', 'Michael Brown', 'michael.brown@example.com', '9876543210', 0, '2023-10-03 14:26:41'),

('user006', 'Sarah Davis', 'sarah.davis@example.com', '1234567890', 0, '2023-10-03 14:26:41'),

('user007', 'David Wilson', 'david.wilson@example.com', '5555555555', 1, '2023-10-03 14:26:49'),

('user008', 'Jessica Thompson', 'jessica.thompson@example.com', '4444444444', 0, '2023-10-03 14:26:49'),

('user009', 'Daniel Anderson', 'daniel.anderson@example.com', '9999999999', 1, '2023-10-03 14:26:49'),

('user010', 'Olivia Martinez', 'olivia.martinez@example.com', '5557778888', 1, '2023-10-03 18:28:07'),

('user011', 'Christopher Lee', 'christopher.lee@example.com', '2223334444', 1, '2023-10-03 18:28:07'),

('user012', 'Sophia Taylor', 'sophia.taylor@example.com', '7778889999', 1, '2023-10-03 18:28:07'),

('user013', 'Robert Wilson', 'robert.wilson@example.com', '1112223333', 1, '2023-10-03 18:28:28'),

('user014', 'Jennifer Adams', 'jennifer.adams@example.com', '4445556666', 1, '2023-10-03 18:28:28'),

('user015', 'Andrew Smith', 'andrew.smith@example.com', '7778889999', 1, '2023-10-03 18:28:28'),

('user016', 'Amanda Johnson', 'amanda.johnson@example.com', '2223334444', 1, '2023-10-03 18:28:37'),

('user017', 'Matthew Davis', 'matthew.davis@example.com', '5556667777', 1, '2023-10-03 18:28:37'),

('user018', 'Elizabeth Brown', 'elizabeth.brown@example.com', '8889990000', 1, '2023-10-03 18:28:37'),

('user019', 'William Jones', 'william.jones@example.com', '4445556666', 1, '2023-10-04 12:06:15'),

('user020', 'Sophie Wilson', 'sophie.wilson@example.com', '7778889999', 1, '2023-10-04 12:06:15');

--

-- Indexes for dumped tables

--

--

-- Indexes for table `managers`

--

ALTER TABLE `managers`

  ADD PRIMARY KEY (`id`);

--

-- Indexes for table `users`

--

ALTER TABLE `users`

  ADD PRIMARY KEY (`id`);

COMMIT;

# CHƯƠNG 4: WebApp

**A screenshot of a computer

Description automatically generated**

**A screenshot of a login screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**