Sale Management

# Yêu cầu

## Mô tả

Tạo website quản lý bán hàng với các chức năng chính như sau:

* Đăng nhập / đăng ký tài khoản. Các tài khoản mới đăng ký phải được kích hoạt qua email.
* Quản lý thông tin cá nhân
* Quản lý danh sách Category, cho phép thêm/xóa/sửa
* Quản lý danh sách Product, cho phép thêm/xóa/sửa/thay đổi trạng thái (ACTIVE/INACTIVE) và thay đổi hình ảnh đại diện.
* Quản lý Cart (giỏ hàng), hiển thị danh sách sản phẩm có trạng thái ACTIVE và cho phép:
  + Xem giỏ hàng
  + Thêm sản phẩm vào giỏ hàng
  + Thay đổi số lượng sản phẩm trong giỏi hàng
  + Xóa sản phẩm ra khỏi giỏ hàng

## Site Map (sơ đồ di chuyển giữa các trang web)

# Hướng dẫn thực hiện

## Thiết kế Database

Tạo database với tên là **sale\_db**

### Table **account** (Tài khoản)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **code** | int | Auto increment |
| email | nvarchar(100) | Email |
| password | nvarchar(50) | Mật khẩu đăng nhập |
| full\_name | nvarchar(1000) | Họ tên |
| birth\_date | date | Ngày sinh |
| address | nvarchar(200) | Địa chỉ |
| gender | char(1) | Giới tính (M: Nam, F: Nữ) |
| image | nvarchar(50) | Ảnh đại diện |
| active | bit | True: tài khoản đã kích hoạt thành công  False: tài khoản chưa được kích hoạt |

### Table **category** (Nhóm sản phẩm)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **code** | int | Auto increment |
| name | nvarchar(100) | Tên category |
| description | nvarchar(1000) | Mô tả (html) |

### Table **product** (Sản phẩm)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **code** | int | Auto increment |
| name | nvarchar(100) | Tên product |
| image | nvarchar(50) | Tên file hình ảnh |
| price | bigint | Giá bán |
| *category\_code* | int | Khóa ngoại, liên kết với category.code |
| description | nvarchar(1000) | Mô tả (html) |
| status | varchar(10) | Trạng thái: ACTIVE/IN-ACTIVE |

### Table **cart** (Giỏ hàng)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **code** | int | Auto increment |
| *account\_code* | int | Mã tài khoản |
| total | bigint | Thành tiền = price \* quantity |
| created\_date | DateTime | Ngày tạo |

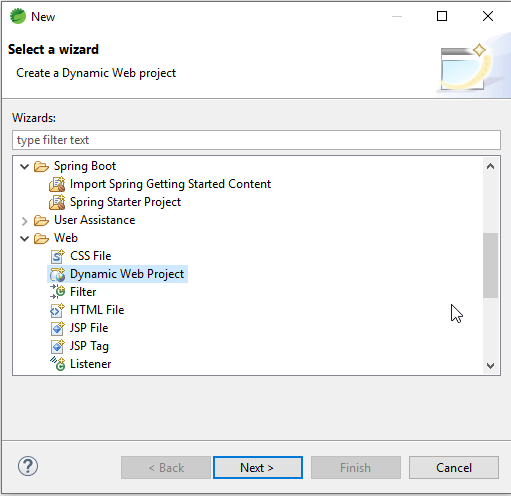
### Table **cart\_detail** (chi tiết giỏ hàng)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **code** | int | Auto increment |
| *cart\_code* | int | Mã giỏ hàng |
| *product\_code* | int | Mã product |
| quantity | int | Số lượng mua |
| price | bigint | Đơn giá |
| total | bigint | Thành tiền = price \* quantity |

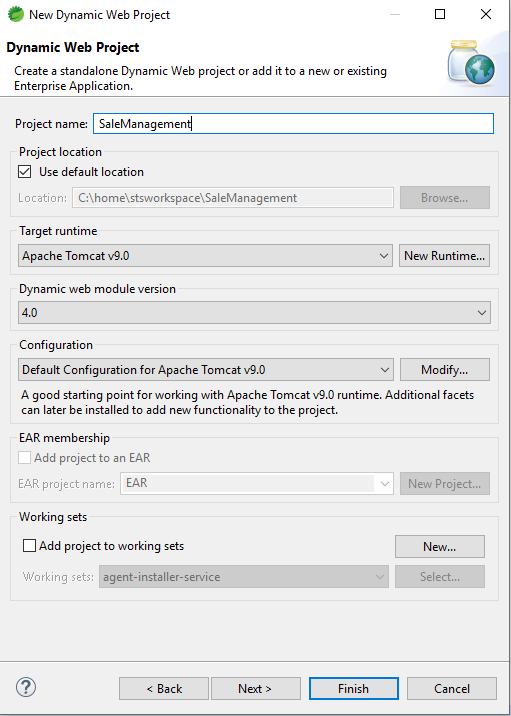
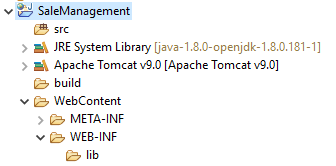
## Tạo Project

Tạo mới Dynamic Web Project bằng cách click vào menu: **File\New\Other..**

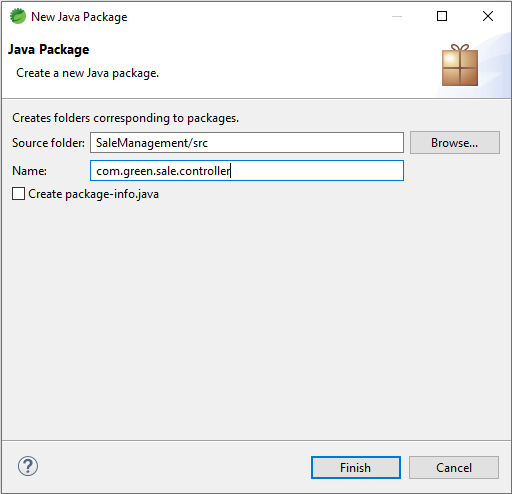
Chọn **Dynamic Web Project** và nhấn **Next**



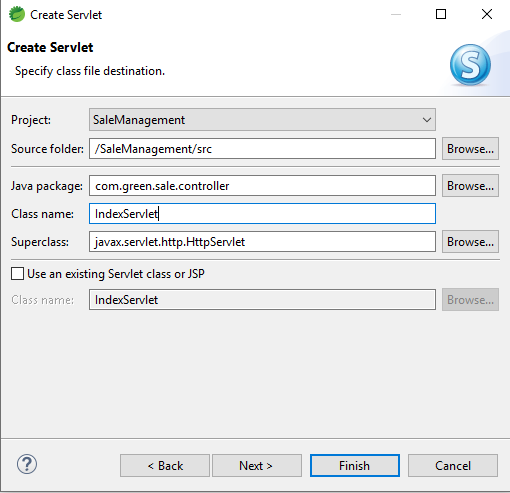
Nhập tên project là **SaleManagement** và nhấn nút **Finish**

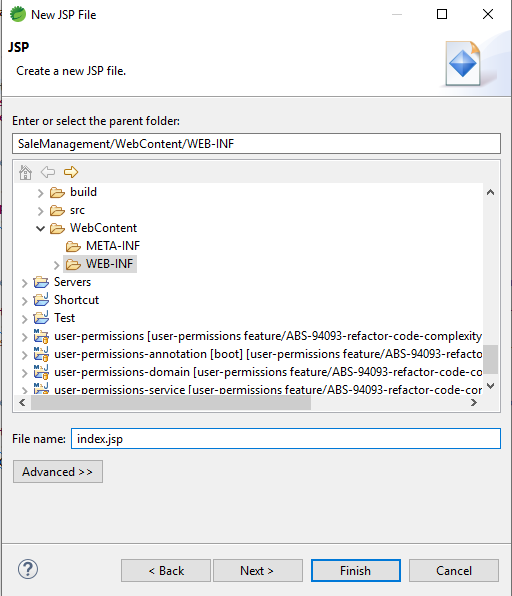
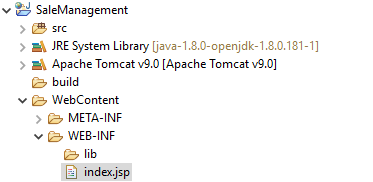
Tạo package với tên là **com.green.sale.controller**



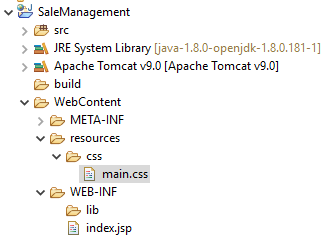
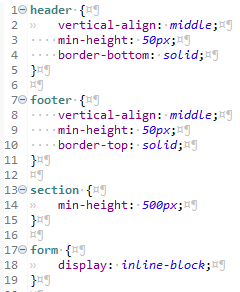
Tạo servlet với tên là **IndexServlet**



Tạo file **index.jsp** nằm trong thư mục **WEB-INF**

Tạo thư mục **resources** và file **main.css** nằm trong thư mục **WEB-INF**

Cập nhật file **index.jsp**, thay đổi encoding thành **UTF-8** và import file **main.css**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Sale Management</title>

<link href=*"*<%=request.getContextPath() %>*/resources/css/main.css"*

rel=*"stylesheet"* type=*"text/css"*>

</head>

<body>

<h1>Sale Management home page</h1>

</body>

</html>

Cập nhật file **IndexServlet.java**, thay đổi servlet path thành **“”** và thêm xử lý hiển thị file **index.jsp** vào method **doGet**.

package com.green.sale.controller;

import java.io.IOException;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class IndexServlet

\*/

@WebServlet("")

public class IndexServlet extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public IndexServlet() {

super();

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

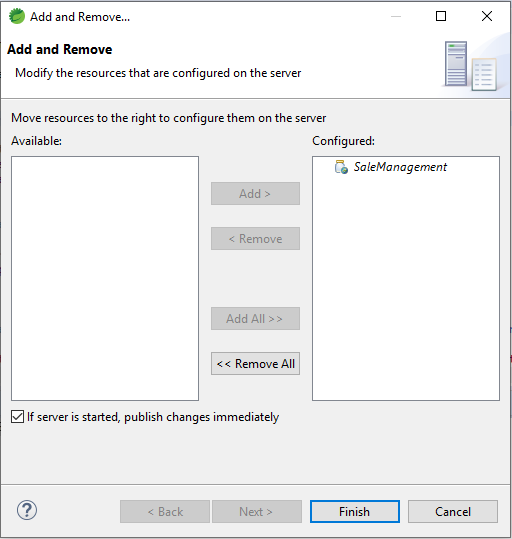
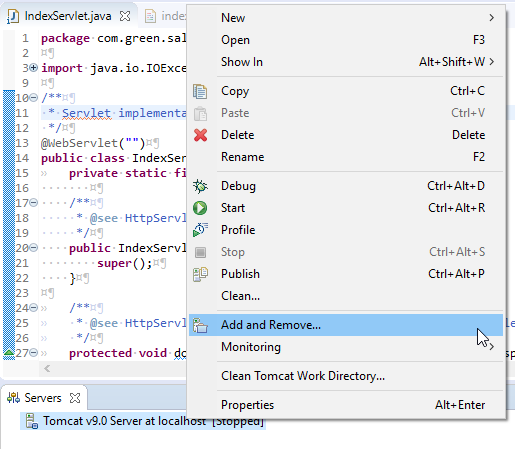
request.getRequestDispatcher("WEB-INF/index.jsp").forward(request, response);

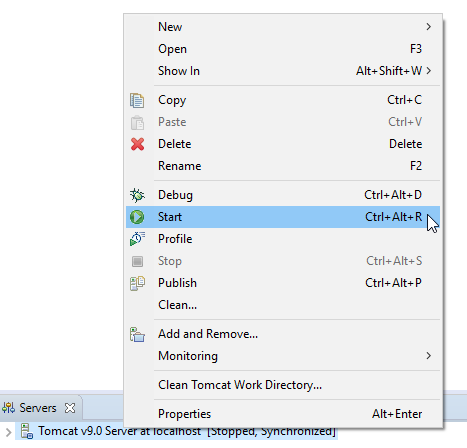
}

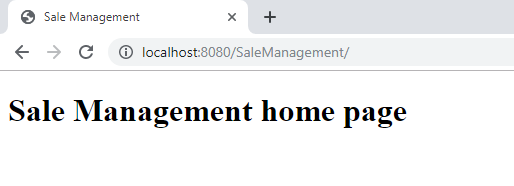
}

Chạy thử trang web vừa được tạo.

* Add project vào Tomcat
* Start Tomcat server
* Mở browser và nhập đường dẫn: **localhost:8080/SaleManagement**

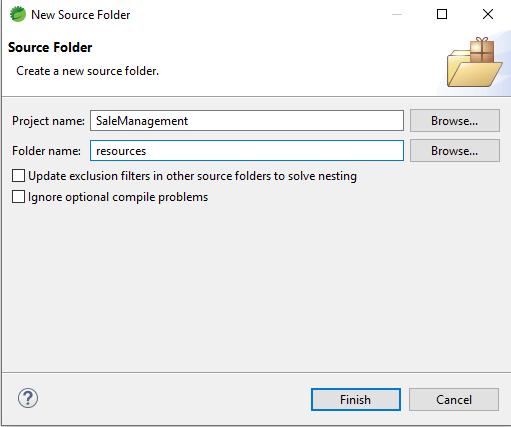
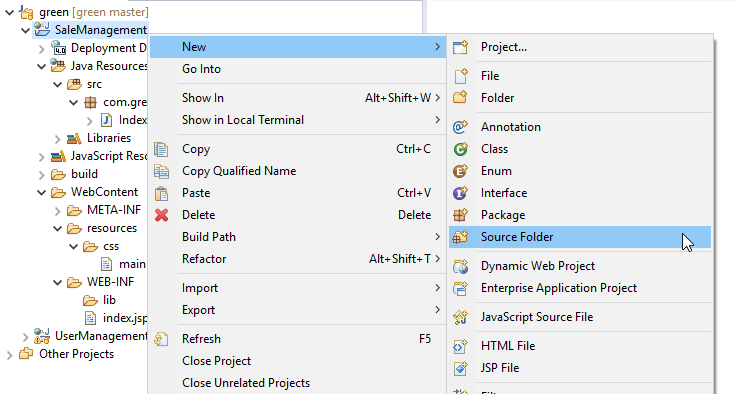




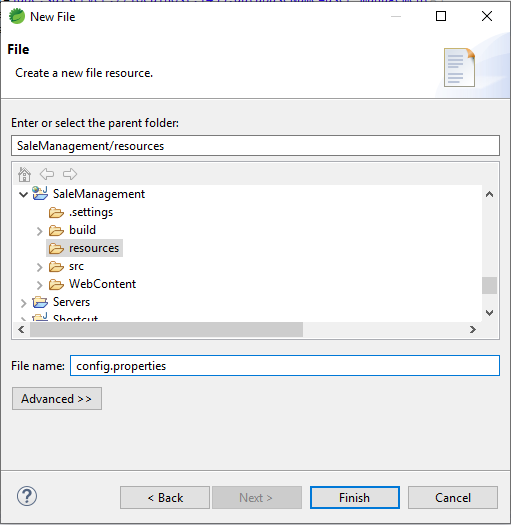
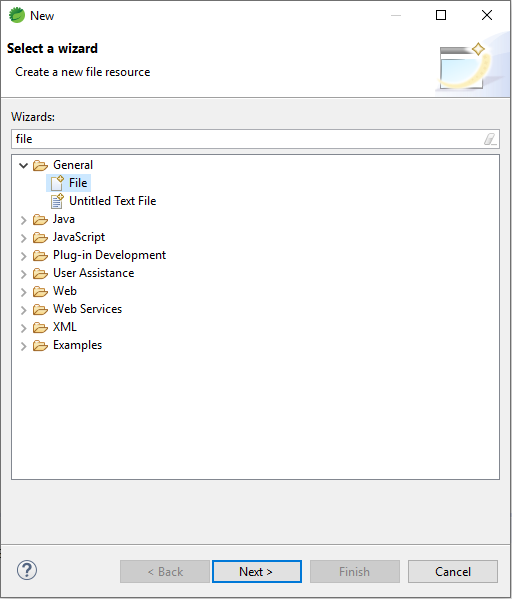


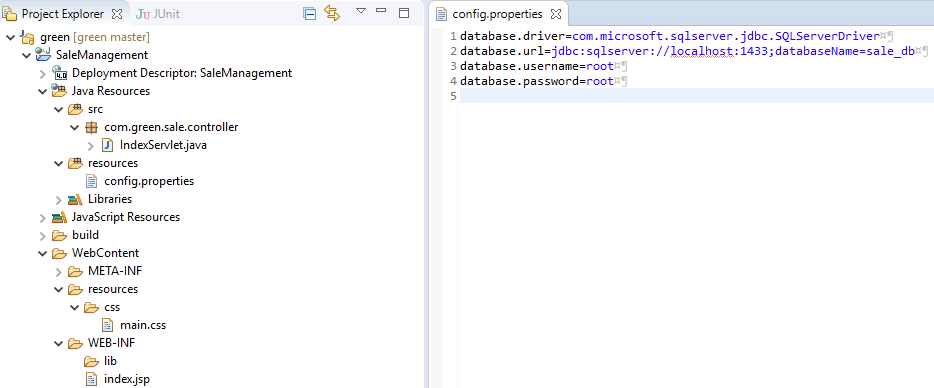
## Import thư viện JDBC để kết nối với Database

Tạo mới Source Folder với tên là **resources**



Tạo file **config.properties**





Tạo class **ApplicationCofig** để đọc nội dung file **resources/config.properties**

**package** com.green.sale.utils;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.util.Properties;

**public** **class** ApplicationConfig {

**private** **static** Properties *prop* = **new** Properties();

**static** {

**try** (InputStream input = ApplicationConfig.**class**.getClassLoader().getResourceAsStream("config.properties")) {

**if** (input == **null**) {

System.***out***.println("Sorry, unable to find config.properties");

} **else** {

// load a properties file from class path, inside static method

*prop*.load(input);

}

} **catch** (IOException ex) {

ex.printStackTrace();

}

}

**public** **static** String getConfig(String key) {

**return** *getConfig*(key, **null**);

}

**public** **static** String getConfig(String key, String defaultVal) {

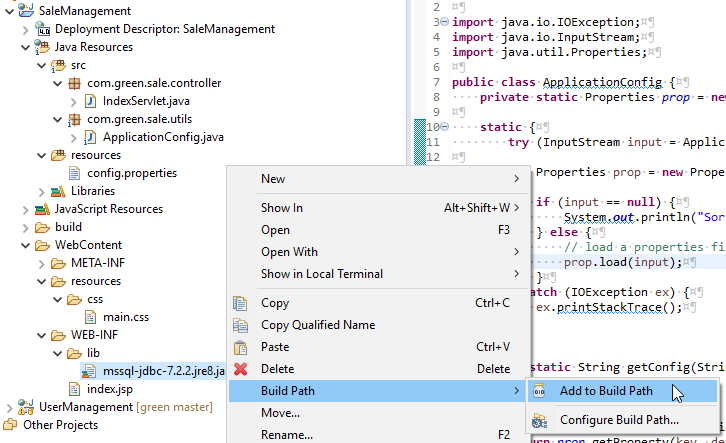
**return** *prop*.getProperty(key, defaultVal);

}

}

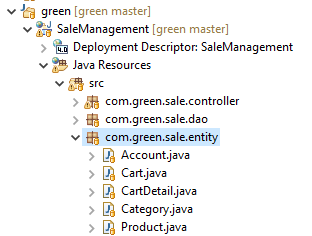
Download thư viện JDBC tại link bên dưới và copy vào thư mục **/WebContent/WEB-INF/lib**

<https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc/7.2.2.jre8>

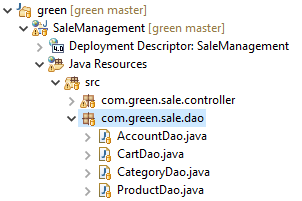


## Tạo entities và DAO

Tạo package **com.green.sale.entity** và các class **Account**, **Category**, **Product**, **Cart, CartDetail** có các thuộc tính tương ứng với các table trong database.



Tạo package **com.green.sale.dao** và các class DAO để xử lý dữ liệu cho các table trong daabase: **AccountDao**, **CategoryDao**, **ProductDao**, **CartDao**



Tạo các phương thức **find**, **findAll**, **insert**, **update**, **delete** tại các class DAO.

Ví dụ cho class AccountDao

**public** **class** AccountDao {

**public** Account find(String code) {

**return** **null**;

}

**public** List<Account> findAll() {

// học viên thực hiện phần này

}

**public** **boolean** insert(Account account) {

// học viên thực hiện phần này

}

**public** **boolean** update(String code, Account account) {

// học viên thực hiện phần này

}

**public** **boolean** delete(String code) {

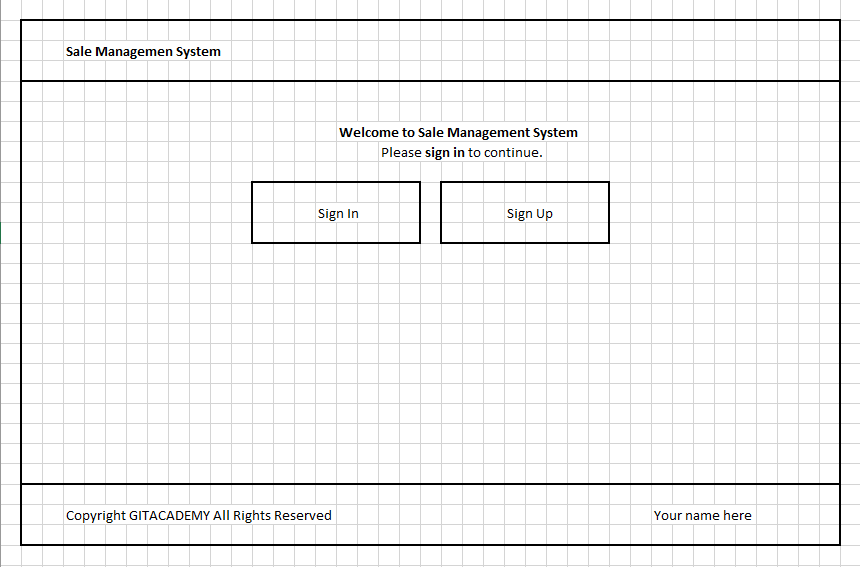
// học viên thực hiện phần này

}

}

## Index Page

### Giao diện



### Yêu cầu

* Khi nhấn button “**Sign In”** sẽ chuyển qua trang đăng nhập
* Khi nhấn button “**Sign Up”** sẽ chuyển qua trang đăng ký

### Hướng dẫn

Cập nhật file **index.jsp** theo cấu trúc như bên dưới.

**Lưu ý:** *Sinh viên tự thêm các html tag cần thiết và sử dụng css để điều chỉnh giao diện.*

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Sale Management</title>

<link href=*"*<%=request.getContextPath() %>*/resources/css/main.css"* rel=*"stylesheet"* type=*"text/css"*>

</head>

<body>

<header>

<div class=*"header-title"*>Sale Management System</div>

</header>

<section>

<p>Welcome to Sale Management System<p>

<p>Please sign in to continue.</p>

<form action=*"*<%=request.getContextPath() %>*/login"* method=*"get"* >

<button>Sign In</button>

</form>

<form action=*"*<%=request.getContextPath() %>*/regis"* method=*"get"*>

<button>Sign Up</button>

</form>

</section>

<footer>

<div style="display: *inline-block*; float: *left*;">Copyright GITACADEMY All Rights Reserved</div>

<div style="display: *inline-block*; float: *right*;">Insert your name here</div>

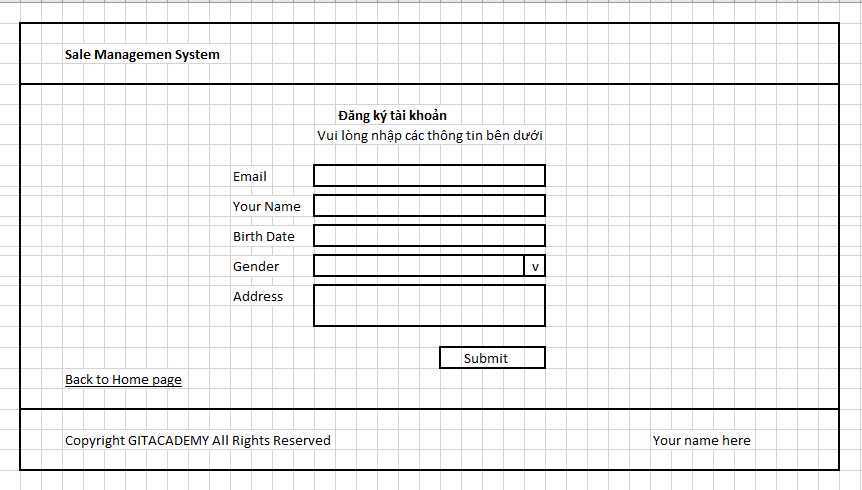
</footer>

</body>

</html>

## Sign Up Page

### Giao diện

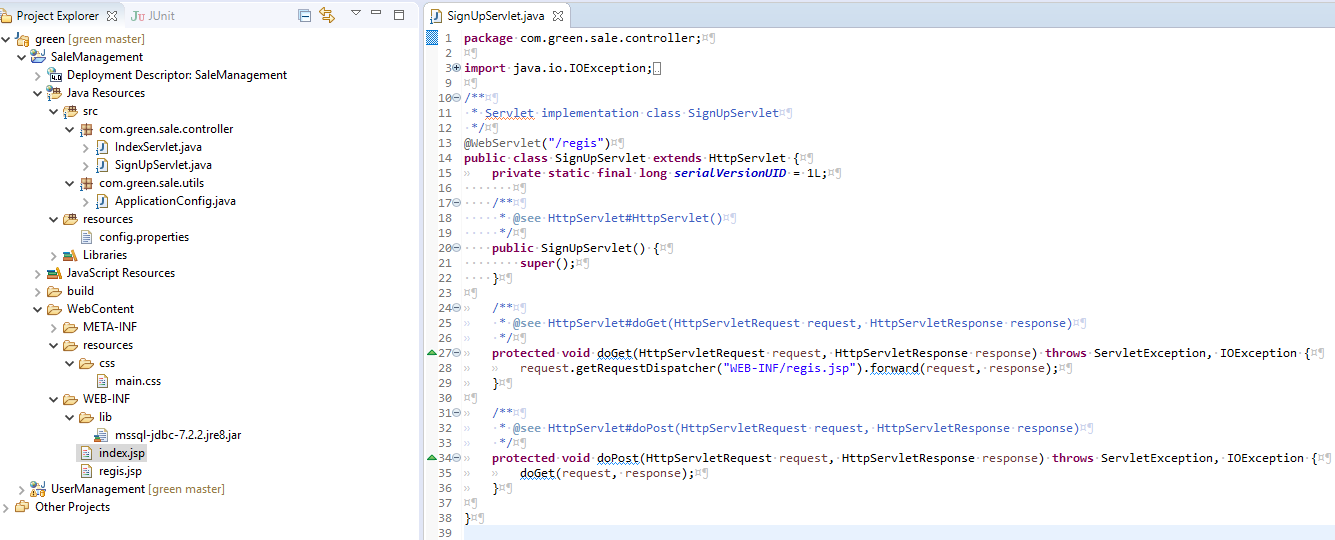


### Yêu cầu

* Khi click link **“Back to Home page”** sẽ quay về trang Index.
* Khi click button **“Submit”** sẽ gửi thông tin đã nhập về server và tiến hành kiểm tra
  + Nếu thông tin được nhập không đúng hoặc bỏ trống thì hiển thị thông báo lỗi tương ứng.
  + Nếu email đã tồn tại trong hệ thống thì hiển thị thông báo lỗi hướng dẫn người dùng chuyển qua trang đăng nhập.
  + Nếu thông tin đúng thì lưu vào database (với mật khẩu được khởi tạo giá trị ngẫu nhiên) và gửi email xác nhận.
  + Nội dung email bao gồm các thông tin: thông báo đăng ký thành công, mật khẩu và các thông tin đã nhập tại trang đăng ký.
* Sau khi xử lý đăng ký thành công sẽ tự động quay về trang Index và hiển thị thông báo yêu cầu người dung kiểm tra email.

### Hướng dẫn

Tạo servlet **SignUpServlet**, thay đổi servlet path thành **“/regis”** và cập nhật method **doGet**



Tạo file regis.jsp tại thư mục **/WEB-INF/regis.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Sign Up</title>

<link href=*"*<%=request.getContextPath() %>*/resources/css/main.css"* rel=*"stylesheet"* type=*"text/css"*>

</head>

<body>

<header>

<div class=*"header-title"*>Sale Management System</div>

</header>

<section>

<div>

<p>Đăng ký tài khoản<p>

<p>Vui lòng nhập các thông tin bên dưới</p>

<form action=*"*<%=request.getContextPath() %>*/regis"* method=*"post"* >

<table>

<tr>

<td>Email</td>

<td><input name=*"email"* type=*"text"* ></td>

</tr>

<tr>

<td>Your Name</td>

<td><input name=*"fullName"* type=*"text"* ></td>

</tr>

<tr>

<td>Birth Date</td>

<td><input name=*"birthDate"* type=*"date"* ></td>

</tr>

<tr>

<td>Gender</td>

<td>

<select name=*"gender"*>

<option value=*"M"*>Male</option>

<option value=*"F"*>FeMale</option>

</select>

</td>

</tr>

<tr>

<td>Address</td>

<td><textarea name=*"address"* rows=*"3"*></textarea></td>

</tr>

</table>

<button type=*"submit"*>Submit</button>

</form>

</div>

<a href=*"*<%=request.getContextPath()%>*"*>Back to Home page</a>

</section>

<footer>

<div style="display: *inline-block*; float: *left*;">Copyright GITACADEMY All Rights Reserved</div>

<div style="display: *inline-block*; float: *right*;">Insert your name here</div>

</footer>

</body>

</html>

Tạo phương thức **insert** trong class **AccountDao** dùng để lưu thông tin account vào database

**package** com.green.sale.dao;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.text.SimpleDateFormat;

**import** sql.date;

**import** com.green.sale.entity.Account;

**import** com.green.sale.utils.ApplicationConfig;

**public** **class** AccountDao {

**public** **boolean** insert(Account account) {

**int** count = 0;

**try** {

Class.*forName*(ApplicationConfig.*getConfig*("database.driver"));

Connection con = DriverManager.*getConnection*(ApplicationConfig.*getConfig*("database.url"),

ApplicationConfig.*getConfig*("database.username"),

ApplicationConfig.*getConfig*("database.password"));

String sql = "INSERT INTO account(email, password, full\_name, birth\_date, address, gender, image)"

+ " VALUES(?, ?, ?, ?, ?, ?, ?)";

PreparedStatement st = con.prepareStatement(sql);

st.setString(1, account.getEmail());

st.setString(2, account.getPassword());

st.setString(3, account.getFullName());

st.setString(4, new Date(account.getBirthDate()));

st.setString(5, account.getAddress());

st.setString(6, account.getGender().toString());

st.setString(7, account.getImage());

count = st.executeUpdate();

st.close();

con.close();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** count > 0;

}

}

Tạo phương thức **isExist** trong **AccountDao** để kiểm tra email đã tồn tại trong database hay chưa.

**public** **boolean** isExist(String email) {

**boolean** result = **false**;

**try** {

Class.*forName*(ApplicationConfig.*getConfig*("database.driver"));

Connection con = DriverManager.*getConnection*(ApplicationConfig.*getConfig*("database.url"),

ApplicationConfig.*getConfig*("database.username"),

ApplicationConfig.*getConfig*("database.password"));

String sql = "SELECT email FROM account WHERE email = ?";

PreparedStatement st = con.prepareStatement(sql);

st.setString(1, email);

ResultSet rs = st.executeQuery();

result = rs.next();

st.close();

con.close();

} **catch** (Exception e) {

e.printStackTrace();

}

**return** result;

}

Hướng dẫn gửi email xác nhận sau khi kiểm tra thông tin đăng ký thành công (tham khảo <https://www.mkyong.com/java/javamail-api-sending-email-via-gmail-smtp-example/>)

* Download và import thư viện java mail (javax.mail-1.6.2.jar) vào project <https://mvnrepository.com/artifact/com.sun.mail/javax.mail/1.6.2>
* Chuẩn bị tài khoản email dùng làm sender email.

<https://www.drivereasy.com/knowledge/what-are-the-gmail-imap-settings-and-how-to-enable-and-use-them/>

* Khai báo các thông tin sender và email server trong file **config.properties**
* Tạo class EmailUtils và cài đặt phương thức gửi email.

Nội dung class **com.green.sale.utils.EmailUtils**

**package** com.green.sale.utils;

**import** java.util.Properties;

**import** javax.mail.Message;

**import** javax.mail.PasswordAuthentication;

**import** javax.mail.Session;

**import** javax.mail.Transport;

**import** javax.mail.internet.InternetAddress;

**import** javax.mail.internet.MimeMessage;

**public** **class** EmailUtils {

**public** **static** **boolean** send(String to, String subject, String htmlMessage) {

**try** {

String username = ApplicationConfig.*getConfig*("mail.username");

String password = ApplicationConfig.*getConfig*("mail.password");

Properties prop = **new** Properties();

prop.put("mail.smtp.host", ApplicationConfig.*getConfig*("mail.smtp.host"));

prop.put("mail.smtp.port", ApplicationConfig.*getConfig*("mail.smtp.port"));

prop.put("mail.smtp.auth", ApplicationConfig.*getConfig*("mail.smtp.auth"));

prop.put("mail.smtp.starttls.enable", ApplicationConfig.*getConfig*("mail.smtp.starttls.enable")); // TLS

Session session = Session.*getInstance*(prop, **new** javax.mail.Authenticator() {

**protected** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(username, password);

}

});

Message message = **new** MimeMessage(session);

message.setFrom(**new** InternetAddress(ApplicationConfig.*getConfig*("mail.from")));

message.setRecipients(Message.RecipientType.***TO***, InternetAddress.*parse*(to));

message.setSubject(subject);

message.setContent(htmlMessage, "text/html");

Transport.*send*(message);

**return** **true**;

} **catch** (Exception e) {

e.printStackTrace();

}

**return** **false**;

}

}

* Gọi phương thử gửi email tại phương thức xử lý đăng ký (Phương thức **doPost** của class **SignUpServlet**).

**EmailUtils.send(to, subject, htmlMessage)**

**package** com.green.sale.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/regis")

**public** **class** SignUpServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

request.getRequestDispatcher("WEB-INF/regis.jsp").forward(request, response);

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// Lấy thông tin đăng ký người dùng đã nhập từ request

// kiểm tra thông tin đăng ký

// Kiểm tra email đã tồn tại hay chưa (gọi phương thức isExist của class AccountDao)

// Khởi tạo giá trị ngẩu nhiên cho mật khẩu

// gửi email xác nhận (mật khẩu được gửi kèm trong nội dung email)

// EmailUtils.send(to, subject, htmlMessage)

// Lưu thông tin vào db

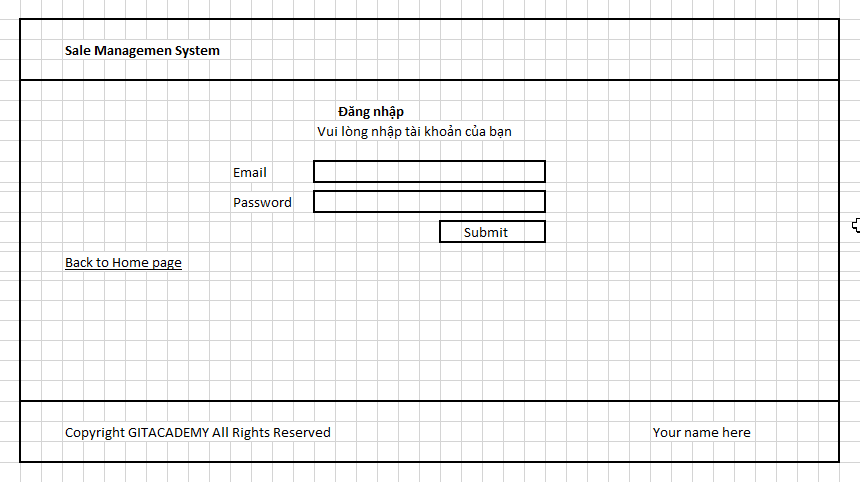
// quay về trang index

}

}

## Sign In Page

### Giao diện

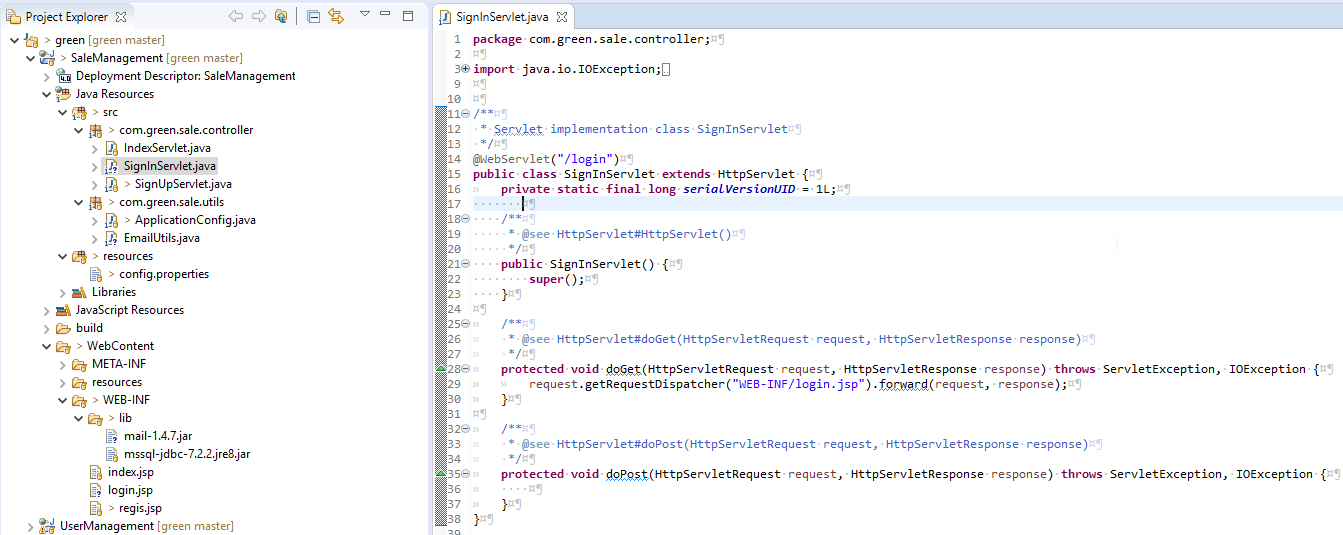


### Yêu cầu

* Khi click link “Back to Home page” sẽ quay về trang Index.
* Khi click button “Submit” sẽ gửi thông tin đã nhập về server và tiến hành kiểm tra
  + Nếu thông tin được nhập không đúng hoặc bỏ trống thì hiển thị thông báo lỗi.
  + Nếu thông tin đăng nhập đúng thì chuyển qua trang Sale

### Hướng dẫn

Tạo servlet **SignInServlet**, thay đổi servlet path thành **“/login”** và cập nhật method **doGet**



Tạo file **login.jsp** tại thư mục **/WEB-INF/login.jsp**

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

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Sign In</title>

<link href=*"*<%=request.getContextPath() %>*/resources/css/main.css"* rel=*"stylesheet"* type=*"text/css"*>

</head>

<body>

<header>

<div class=*"header-title"*>Sale Management System</div>

</header>

<section>

<div>

<p>Đăng nhập<p>

<p>Vui lòng nhập tài khoản của bạn</p>

<form action=*"*<%=request.getContextPath() %>*/login"* method=*"post"* >

<table>

<tr>

<td>Email</td>

<td><input name=*"email"* type=*"text"* ></td>

</tr>

<tr>

<td>Password</td>

<td><input name=*"password"* type=*"password"* ></td>

</tr>

</table>

<button type=*"submit"*>Submit</button>

</form>

</div>

<a href=*"*<%=request.getContextPath()%>*"*>Back to Home page</a>

</section>

<footer>

<div style="display: *inline-block*; float: *left*;">Copyright GITACADEMY All Rights Reserved</div>

<div style="display: *inline-block*; float: *right*;">Insert your name here</div>

</footer>

</body>

</html>

Tạo phương thức **findByEmail** trong class **AccountDao** để tìm thông tin account từ database theo email

**public** Account findByEmail(String email) {

// sinh viên tự thực hiện phần này

}

Xử lý đăng nhập tại phương thức **SignInServlet.doPost**

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// Lấy thông tin đăng ký người dùng đã nhập từ request

// kiểm tra email và password khác rỗng

// kiểm tra email và password đúng với db (gọi phương thức findByEmail của class AccountDao)

// lưu thông tin user vào session với key là CURRENT\_USER

//HttpSession session = request.getSession();

//session.setAttribute("CURRENT\_USER", account);

// redirect qua trang sale

}

Tạo servlet **SaleServlet** nằm trong package **com.green.sale.controller**

**package** com.green.sale.controller;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

@WebServlet("/sale")

**public** **class** SaleServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

request.getRequestDispatcher("WEB-INF/sale.jsp").forward(request, response);

}

}

Tạo file **WEB-INF/sale.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Sale Management</title>

<link href=*"*<%=request.getContextPath() %>*/resources/css/main.css"* rel=*"stylesheet"* type=*"text/css"*>

</head>

<body>

<header>

<div class=*"header-title"*>Sale Management System</div>

<div style="display: *inline-block*; float: *right*">

<div class=*"header-btn-group"*>

<a href=*"*<%=request.getContextPath()%>*\logout"*>Logout</a>

<a href=*"*<%=request.getContextPath()%>*\logout"*>Logout</a>

</div>

</div>

</header>

<section>

<h1>Sale page</h1>

</section>

<footer>

<div style="display: *inline-block*; float: *left*;">Copyright GITACADEMY All Rights Reserved</div>

<div style="display: *inline-block*; float: *right*;">Insert your name here</div>

</footer>

</body>

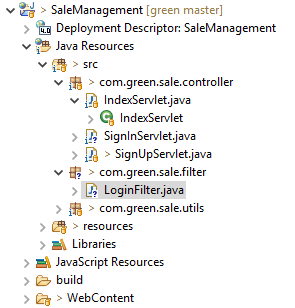
</html>

Tạo filter để ngăn chặn các truy cập trái phép từ người dùng.

|  |  |  |
| --- | --- | --- |
| **Path** | **Mô tả** | **Yêu cầu đăng nhập** |
| /sale/\* | Trang Sale và các trang con | yes |
| /category/\* | Trang category list và các trang con | yes |
| /product/\* | Trang product list và các trang con | yes |
| /profile/\* | Trang user profile và các trang con | yes |
| / | Trang index | no |
| /login | Trang đăng nhập | no |
| /regis | Trang đăng ký | no |
| /resources/\* | Các request download web resource: css, javascript, image,… | no |

Tạo class **LoginFilter** nằm trong package **com.green.sale.filter**

Tham khảo thêm về servlet filter: <https://o7planning.org/vi/10395/huong-dan-java-servlet-filter>



**package** com.green.sale.filter;

**import** java.io.IOException;

**import** javax.servlet.Filter;

**import** javax.servlet.FilterChain;

**import** javax.servlet.ServletException;

**import** javax.servlet.ServletRequest;

**import** javax.servlet.ServletResponse;

**import** javax.servlet.annotation.WebFilter;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

// Yêu cầu login đối với các request có path như bên dưới

@WebFilter(urlPatterns = {

"/sale/\*",

"/category/\*",

"/product/\*",

"/profile/\*"

})

**public** **class** LoginFilter **implements** Filter {

@Override

**public** **void** doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

**throws** IOException, ServletException {

HttpServletRequest htttpRequest = (HttpServletRequest) request;

HttpSession session = htttpRequest.getSession();

Account currentUser = (Account) session.getAttribute("CURRENT\_USER");

// Nếu chưa login thì redirect về trang index

**if** (currentUser == **null**) {

HttpServletResponse httpResponse = (HttpServletResponse) response;

httpResponse.sendRedirect(htttpRequest.getContextPath());

**return**;

}

// thực hiện các filter tiếp theo

chain.doFilter(request, response);

}

}

Tạo Filter tự động chuyển trang khi đã đăng nhập

Tạo class **LoginRedirectFilter** nằm trong package **com.green.sale.filter**

**package** com.green.sale.filter;

**import** java.io.IOException;

**import** javax.servlet.Filter;

**import** javax.servlet.FilterChain;

**import** javax.servlet.ServletException;

**import** javax.servlet.ServletRequest;

**import** javax.servlet.ServletResponse;

**import** javax.servlet.annotation.WebFilter;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** javax.servlet.http.HttpSession;

// Nếu đã login thì không được truy cập vào các trang bên dưới

@WebFilter(urlPatterns = {

"",

"/login",

"/regis"

})

**public** **class** LoginRedirectFilter **implements** Filter {

@Override

**public** **void** doFilter(ServletRequest request, ServletResponse response, FilterChain chain)

**throws** IOException, ServletException {

HttpServletRequest htttpRequest = (HttpServletRequest) request;

HttpSession session = htttpRequest.getSession();

Account currentUser = (Account) session.getAttribute("CURRENT\_USER");

// Nếu đã login thì redirect về trang sale

**if** (currentUser != **null**) {

HttpServletResponse httpResponse = (HttpServletResponse) response;

httpResponse.sendRedirect(htttpRequest.getContextPath() + "/sale");

**return**;

}

// thực hiện các filter tiếp theo

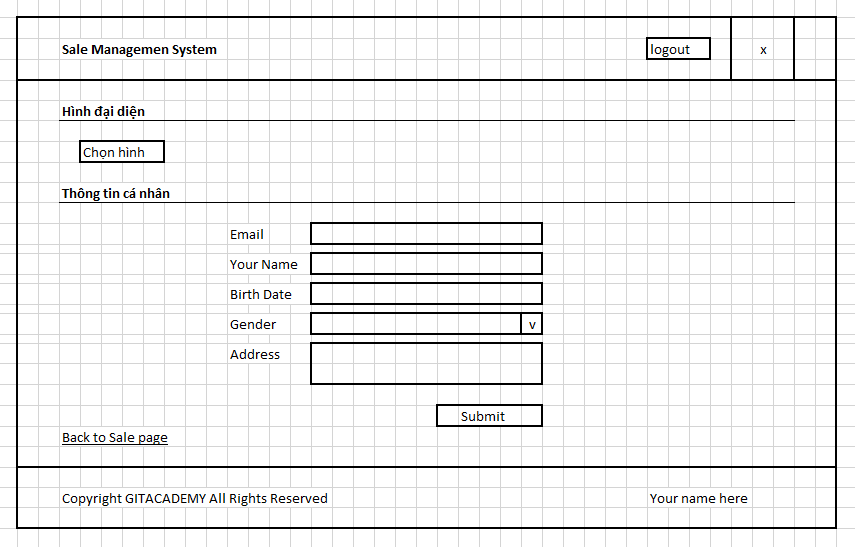
chain.doFilter(request, response);

}

}

## Profile Page

### Giao diện



### Yêu cầu

Tạo trang thông tin cá nhân (path: **/profile**) với các chức năng sau:

* Hiển thị thông tin user đã đăng nhập, và cho phép chỉnh sửa
* Thay đổi hình ảnh đại diện
* Hiển thị hình đại diện ở góc trên bên phải màn hình

### Hướng dẫn

Tạo file **profile.jsp** tại thư mục **WEB-INF/profile.jsp**

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

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Profile</title>

<link href=*"*<%=request.getContextPath() %>*/resources/css/main.css"* rel=*"stylesheet"* type=*"text/css"*>

</head>

<body>

<header>

<div class=*"header-title"*>Sale Management System</div>

<div style="display: *inline-block*; float: *right*">

<div class=*"header-btn-group"*>

<a href=*"*<%=request.getContextPath()%>*\logout"*>Logout</a>

<a href=*"*<%=request.getContextPath()%>*\logout"*>Logout</a>

</div>

<div class=*"header-btn-group"*>

<img src=*"*<%=request.getContextPath() %>*/profile/image"* alt=*"Avatar"* width=*"40"* height=*"40"*>

</div>

</div>

</header>

<section>

<div>

<h2 class=*"section-header"*>Hình đại diện</h2>

<div>

<img src=*"*<%=request.getContextPath() %>*/profile/image"* alt=*"Avatar"* width=*"100"* height=*"100"*>

</div>

<div>

<form action=*"*<%=request.getContextPath() %>*/profile/image"* method=*"post"* enctype=*"multipart/form-data"*>

<input type=*"file"* name=*"file"* />

<button type=*"submit"*>Upload</button>

</form>

</div>

</div>

<div>

<h2 class=*"section-header"*>Thông tin cá nhân</h2>

<div>

<form action=*"*<%=request.getContextPath() %>*/profile"* method=*"post"* >

<table>

<tr>

<td>Email</td>

<td><input name=*"email"* value=*"*${\_account.email}*"* type=*"text"* ></td>

</tr>

<tr>

<td>Your Name</td>

<td><input name=*"fullName"* value=*"*${\_account.fullName}*"* type=*"text"* ></td>

</tr>

<tr>

<td>Birth Date</td>

<td><input name=*"birthDate"* value=*"*${\_account.birthDate}*"* type=*"date"* ></td>

</tr>

<tr>

<td>Gender</td>

<td>

<select name=*"gender"*>

<option value=*"M"* selected=*"*${\_account.gender **eq** 'M'}*"*>Male</option>

<option value=*"F"* selected=*"*${\_account.gender **eq** 'F'}*"*>FeMale</option>

</select>

</td>

</tr>

<tr>

<td>Address</td>

<td><textarea name=*"address"* rows=*"3"*>${\_account.address}</textarea></td>

</tr>

</table>

<button type=*"submit"*>Submit</button>

</form>

</div>

</div>

<a href=*"*<%=request.getContextPath()%>*/sale"*>Back to Sale page</a>

</section>

<footer>

<div style="display: *inline-block*; float: *left*;">Copyright GITACADEMY All Rights Reserved</div>

<div style="display: *inline-block*; float: *right*;">Insert your name here</div>

</footer>

</body>

</html>

Tạo class **ProfileServlet** tại package **com.green.sale.controller**  và xử lý hiển thị thông tin user đang đăng nhập

Xử lý cập nhật thông tin user tại phương thức **doPost**

**package** com.green.sale.controller;

**import** java.io.IOException;

**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** com.green.sale.entity.Account;

@WebServlet("/profile")

**public** **class** ProfileServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\* Xử lý hiển thị trang profile

\*/

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// lấy thông tin user đang đăng nhập

HttpSession session = request.getSession();

Account account = (Account) session.getAttribute("CURRENT\_USER");

// gán thông tin user vào request attribute và hiển thị trên trang profile

request.setAttribute("\_account", account);

request.getRequestDispatcher("WEB-INF/profile.jsp").forward(request, response);

}

/\*\*

\* Xử lý cập nhật thông tin user

\*/

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// Lấy thông tin user người dùng đã nhập từ request

// kiểm tra thông tin đăng ký

// Kiểm tra email đã tồn tại hay chưa

// gọi phương thức update của class AccountDao để lưu thông tin vào db

// Cập nhật thông tin user trong session (CURRENT\_USER)

// hiển thị trang profile với thông tin mới nhất được lấy từ DB

doGet(request, response);

}

}

Tạo class **ProfileImageServlet** tại package **com.green.sale.controller** để xử lý upload/download hình đại diện.

**package** com.green.sale.controller;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** java.util.List;

**import** javax.servlet.ServletContext;

**import** javax.servlet.ServletException;

**import** javax.servlet.ServletOutputStream;

**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** org.apache.commons.fileupload.FileItem;

**import** org.apache.commons.fileupload.FileItemFactory;

**import** org.apache.commons.fileupload.disk.DiskFileItemFactory;

**import** org.apache.commons.fileupload.servlet.ServletFileUpload;

**import** com.green.sale.entity.Account;

**import** com.green.sale.utils.ApplicationConfig;

/\*\*

\* https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload/1.4

\* https://mvnrepository.com/artifact/commons-io/commons-io/2.6

\*

\*/

@WebServlet("/profile/image")

**public** **class** ProfileImageServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** ServletFileUpload uploader;

**private** String PROFILE\_IMAGE\_DIR;

@Override

**public** **void** init() **throws** ServletException {

PROFILE\_IMAGE\_DIR = ApplicationConfig.*getConfig*("image.dir.profile");

File filesDir = **new** File(PROFILE\_IMAGE\_DIR);

**if** (!filesDir.exists()) {

filesDir.mkdirs();

}

FileItemFactory factory = **new** DiskFileItemFactory();

uploader = **new** ServletFileUpload(factory);

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

// lấy thông tin user đang đăng nhập

HttpSession session = request.getSession();

Account account = (Account) session.getAttribute("CURRENT\_USER");

String fileName = PROFILE\_IMAGE\_DIR + File.***separator*** + account.getCode();

ServletContext ctx = getServletContext();

File file = **new** File(fileName);

// nếu không tồn tại thì lấy hình đại diện mặc định

**if** (!file.exists()) {

file = **new** File(ctx.getRealPath("resources/image/avatar.png"));

}

InputStream fis = **new** FileInputStream(file);

String mimeType = ctx.getMimeType(file.getAbsolutePath());

response.setContentType(mimeType != **null** ? mimeType : "application/octet-stream");

response.setContentLength((**int**) file.length());

response.setHeader("Content-Disposition", "inline");

ServletOutputStream os = response.getOutputStream();

**byte**[] bufferData = **new** **byte**[1024];

**int** read = 0;

**while** ((read = fis.read(bufferData)) != -1) {

os.write(bufferData, 0, read);

}

os.flush();

os.close();

fis.close();

}

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

HttpSession session = request.getSession();

Account account = (Account) session.getAttribute("CURRENT\_USER");

**if** (!ServletFileUpload.*isMultipartContent*(request)) {

**throw** **new** ServletException("Content type is not multipart/form-data");

}

**try** {

List<FileItem> fileItemsList = uploader.parseRequest(request);

FileItem fileItem = fileItemsList.get(0);

**if** (fileItem.getSize() > 0) {

File file = **new** File(PROFILE\_IMAGE\_DIR + File.***separator*** + account.getCode());

file.delete();

fileItem.write(file);

}

} **catch** (Exception e) {

**throw** **new** ServletException("Error to upload file", e);

}

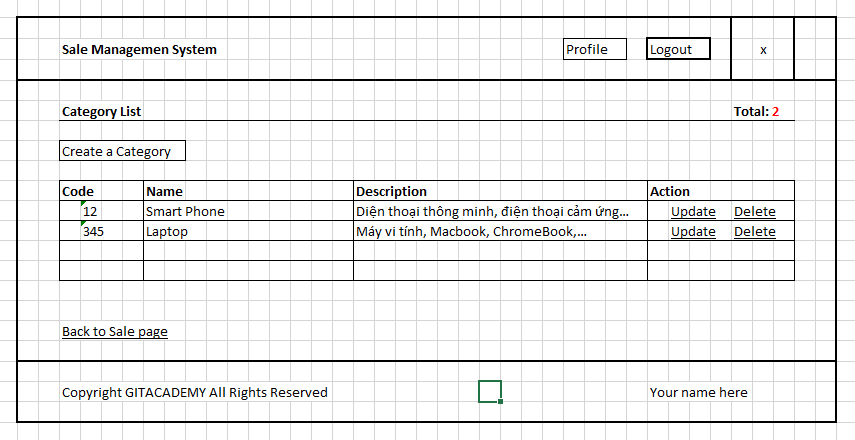
response.sendRedirect(request.getContextPath() + "/profile");

}

}

## Category List Page

### Giao diện



### Yêu cầu

Tạo trang Category List (path: **/category**) hiển thị danh sách category với các chức năng:

* Hiển thị thông tin category (code, name, description)
* Xóa một category
* Tạo liên kết qua trang Category Detail để tạo mới hoặc cập nhật thông tin category.

### Hướng dẫn

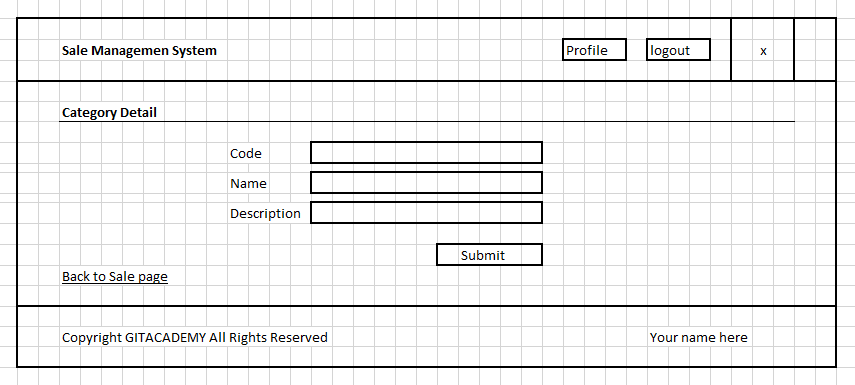
Tạo file **category.jsp** tại thư mục **WEB-INF/category.jsp**

Tạo class **CategoryServlet** tại package **com.green.sale.controller**  và xử lý hiển thị danh sách category

Tạo class **CategoryDetailServlet** tại package **com.green.sale.controller**  và xử lý xóa category tại phương thức **doPost**

## Category Detail Page

### Giao diện



### Yêu cầu

Tạo trang Category Detail để tạo mới hoặc cập nhật thông tin category

* Tạo mới category
  + Textbox category code sẽ không hiển thị
  + Các thông tin phải được kiểm tra trước khi lưu vào database
  + Sau khi thực hiện thành công sẽ quay về trang Category List (/category)
* Cập nhật category.
  + Thông tin category đang được cập nhật phải hiển thị lên các textbox tương ứng (code, name, description) .
  + Textbox category code được hiển thị nhưng không cho phép chỉnh sửa
  + Các thông tin phải được kiểm tra trước khi lưu vào database
  + Sau khi thực hiện thành công sẽ quay về trang Category List (/category)

### Hướng dẫn

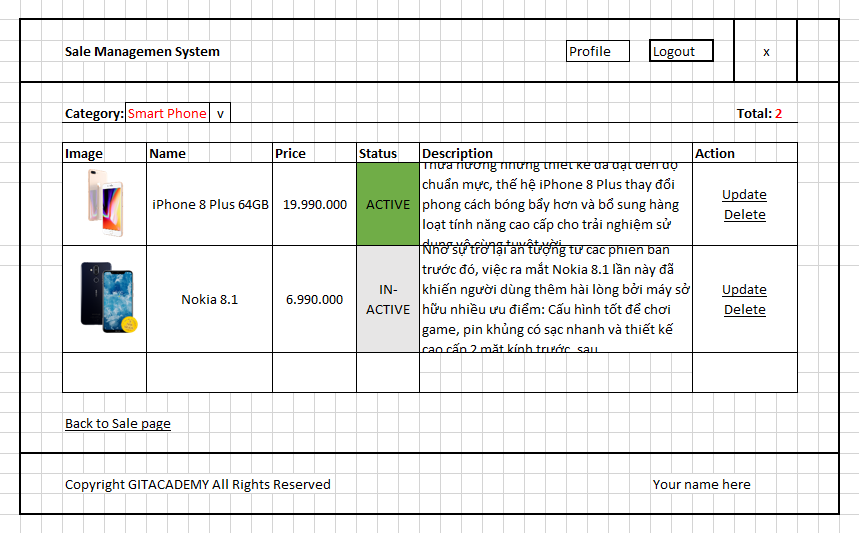
Tạo file **category-detail.jsp** tại thư mục **WEB-INF/category-detail.jsp**

Xử lý lấy thông tin category tại phương thức **doGet** của class **CategoryDetailServlet**

Xử lý tạo mới / cập nhật category tại phương thức **doPost** của class **CategoryDetailServlet**

## Product List Page

### Giao diện



### Yêu cầu

Tạo trang Product List (path: **/product**) hiển thị danh sách product với các chức năng:

* Tìm danh sách product theo category, khi người dùng chọn một category mới thì danh sách product bên dưới phải được cập nhật tương ứng với category được chọn.
* Hiển thị thông tin product (image, name, status, price, description)
* Tô màu cho cột status tương tự như giao diện tham khảo ở trên
* Xóa một product
* Tạo liên kết qua trang Product Detail để tạo mới hoặc cập nhật thông tin.

### Hướng dẫn

Tạo file **product.jsp** tại thư mục **WEB-INF/product.jsp**

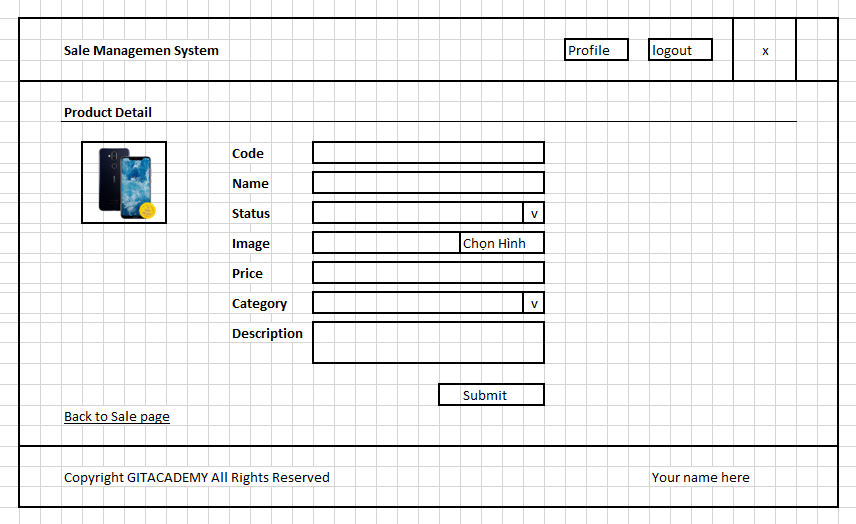
Tạo class **ProductServlet** tại package **com.green.sale.controller**  và xử lý hiển thị danh sách product

Tạo class **ProductImageServlet** tại package **com.green.sale.controller** và xử lý hiển thị hình đại diện product

Tạo class **ProductDetailServlet** tại package **com.green.sale.controller**  và xử lý xóa product tại phương thức **doPost**

## Product Detail Page

### Giao diện



### Yêu cầu

Tạo trang Product Detail để tạo mới hoặc cập nhật thông tin product

* Tạo mới product
  + Textbox code sẽ không hiển thị
  + Hình product sẽ không hiển thị
  + Các thông tin phải được kiểm tra trước khi lưu vào database
  + Sau khi thực hiện thành công sẽ quay về trang Product List
* Cập nhật product.
  + Thông tin product đang được cập nhật phải hiển thị lên các textbox tương ứng (code, name, status, price, category, description) .
  + Hình product phải được hiển thị ở góc trên bên trái màn hình
  + Textbox category code được hiển thị nhưng không cho phép chỉnh sửa
  + Các thông tin phải được kiểm tra trước khi lưu vào database
  + Sau khi thực hiện thành công sẽ quay về trang Product List

### Hướng dẫn

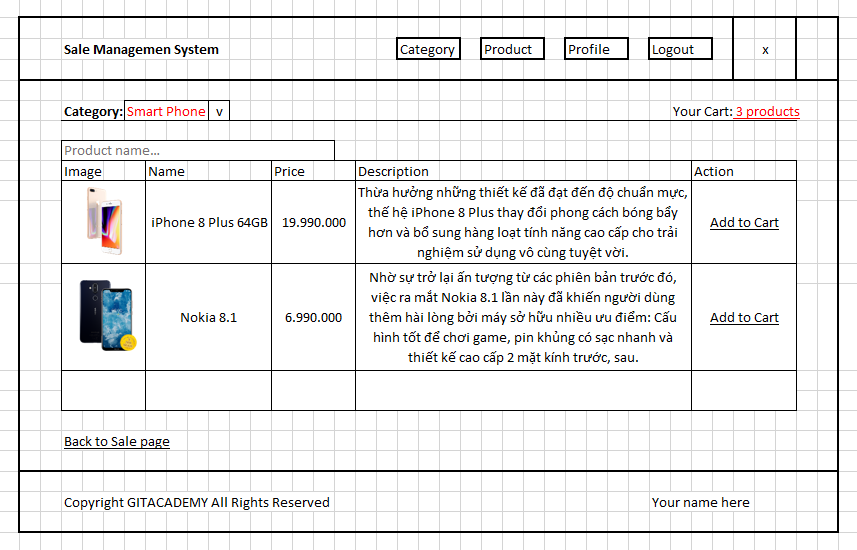
Tạo file **product-detail.jsp** tại thư mục **WEB-INF/product-detail.jsp**

Xử lý lấy thông tin product tại phương thức **doGet** của class **ProductDetailServlet**

Xử lý tạo mới / cập nhật product tại phương thức **doPost** của class **ProductDetailServlet**

## Sale Page

### Giao diện



### Yêu cầu

Tạo trang Sale với các chức năng:

* Hiển thị số lượng product đã chọn trong giỏ hàng
* Cho phép tìm product theo tên và category
* Hiển thị thông tin product (image, name, price, description)
* Cho phép click “Add to Cart” để thêm product vào giỏ hàng
* Di chuyển qua trang web tưng ứng khi click các liên kết trên màn hình

### Hướng dẫn

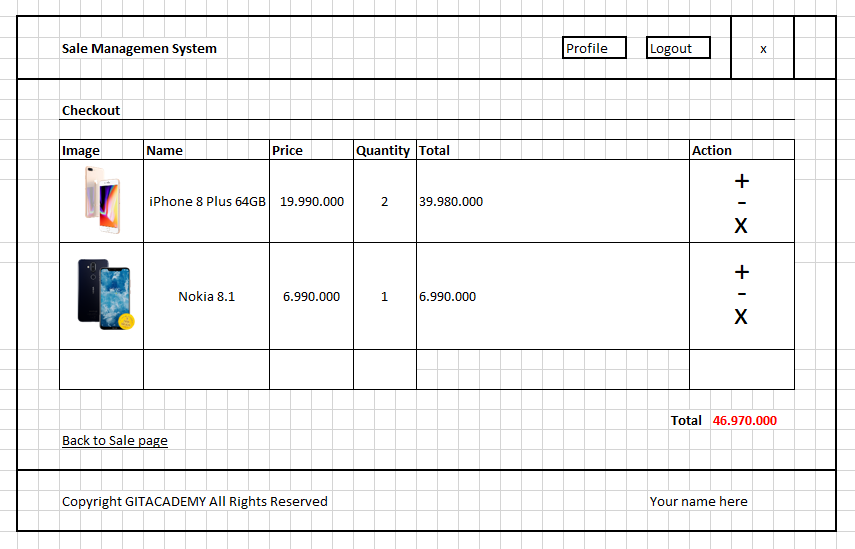
Tạo file **sale.jsp** tại thư mục **WEB-INF/sale.jsp**

Tạo class **SaleServlet** tại package **com.green.sale.controller**  và xử lý hiển thị danh sách product

Tạo class **CartServlet** tại package **com.green.sale.controller**  và xử lý thêm product vào giỏ hàng tại phương thức **doPost**

## Checkout Page

### Giao diện



### Yêu cầu

Tạo trang checkout với các chức năng sau:

* Hiển thị danh sách product đã được thêm vào giỏ hàng
* Tính tổng số tiền dựa theo số lượng product đã chọn và đơn giá
* Tăng / Giảm số lượng product
* Xóa product ra khỏi giỏ hàng

### Hướng dẫn

Tạo file **checkout.jsp** tại thư mục **WEB-INF/checkout.jsp**

Xử lý hiển thị danh sách product trong giỏ hàng tại phương thức **doGet** của class **CartServlet**

Xử lý Tăng / Giảm / Xóa product tại phương thức **doPost** của class **CartServlet**

# Phụ lục

## /WebContent/resources/css/main.css

**header** {

vertical-align: *middle*;

min-height: *50px*;

border-bottom: *solid*;

}

**footer** {

vertical-align: *middle*;

min-height: *50px*;

border-top: *solid*;

}

**section** {

min-height: *500px*;

}

**form** {

display: *inline-block*;

}

## /resources/config.properties

database.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver

database.url=jdbc:sqlserver://localhost:1433;databaseName=sale\_db

database.username=root

database.password=root

mail.from=pqjdev002@gmail.com

mail.username=pqjdev002@gmail.com

mail.password=javadev002

mail.smtp.host=smtp.gmail.com

mail.smtp.port=587

mail.smtp.auth=true

mail.smtp.starttls.enable=true

Mục Lục

[Yêu cầu 1](#_Toc13733288)

[1. Mô tả 1](#_Toc13733289)

[2. Site Map (sơ đồ di chuyển giữa các trang web) 1](#_Toc13733290)

[Hướng dẫn thực hiện 2](#_Toc13733291)

[1. Thiết kế Database 2](#_Toc13733292)

[ Table **account** (Tài khoản) 2](#_Toc13733293)

[ Table **category** (Nhóm sản phẩm) 2](#_Toc13733294)

[ Table **product** (Sản phẩm) 2](#_Toc13733295)

[ Table **cart** (Giỏ hàng) 2](#_Toc13733296)

[ Table **cart\_detail** (chi tiết giỏ hàng) 2](#_Toc13733297)

[2. Tạo Project 3](#_Toc13733298)

[3. Import thư viện JDBC để kết nối với Database 8](#_Toc13733299)

[4. Tạo entities và DAO 10](#_Toc13733300)

[5. Index Page 11](#_Toc13733301)

[ Giao diện 11](#_Toc13733302)

[ Yêu cầu 11](#_Toc13733303)

[ Hướng dẫn 11](#_Toc13733304)

[6. Sign Up Page 12](#_Toc13733305)

[ Giao diện 12](#_Toc13733306)

[ Yêu cầu 12](#_Toc13733307)

[ Hướng dẫn 13](#_Toc13733308)

[7. Sign In Page 17](#_Toc13733309)

[ Giao diện 17](#_Toc13733310)

[ Yêu cầu 17](#_Toc13733311)

[ Hướng dẫn 17](#_Toc13733312)

[8. Profile Page 22](#_Toc13733313)

[ Giao diện 22](#_Toc13733314)

[ Yêu cầu 23](#_Toc13733315)

[ Hướng dẫn 23](#_Toc13733316)

[9. Category List Page 23](#_Toc13733317)

[ Giao diện 23](#_Toc13733318)

[ Yêu cầu 23](#_Toc13733319)

[ Hướng dẫn 24](#_Toc13733320)

[10. Category Detail Page 24](#_Toc13733321)

[ Giao diện 24](#_Toc13733322)

[ Yêu cầu 24](#_Toc13733323)

[ Hướng dẫn 24](#_Toc13733324)

[11. Product List Page 25](#_Toc13733325)

[ Giao diện 25](#_Toc13733326)

[ Yêu cầu 25](#_Toc13733327)

[ Hướng dẫn 25](#_Toc13733328)

[12. Product Detail Page 26](#_Toc13733329)

[ Giao diện 26](#_Toc13733330)

[ Yêu cầu 26](#_Toc13733331)

[ Hướng dẫn 26](#_Toc13733332)

[13. Sale Page 27](#_Toc13733333)

[ Giao diện 27](#_Toc13733334)

[ Yêu cầu 27](#_Toc13733335)

[ Hướng dẫn 27](#_Toc13733336)

[14. Checkout Page 28](#_Toc13733337)

[ Giao diện 28](#_Toc13733338)

[ Yêu cầu 28](#_Toc13733339)

[ Hướng dẫn 28](#_Toc13733340)

[Phụ lục 28](#_Toc13733341)

[1. /WebContent/resources/css/main.css 28](#_Toc13733342)

[2. /resources/config.properties 29](#_Toc13733343)