**FPT UNIVERSITY**

**FPT POLYTECHNIC COLLEGE**

****

**JAVA 5 - SPRING BOOT**

**Hướng dẫn lab 7**

**NHÓM 5: PHẠM NGUYỄN MINH TRIẾT**

**NGUYỄN HOÀNG HẢI**

**HOÀNG VIỆT ANH**

**NGUYỄN ĐỨC PHÁT**

**LỚP: IT16303**

**GIẢNG VIÊN: LÊ ANH TÚ**

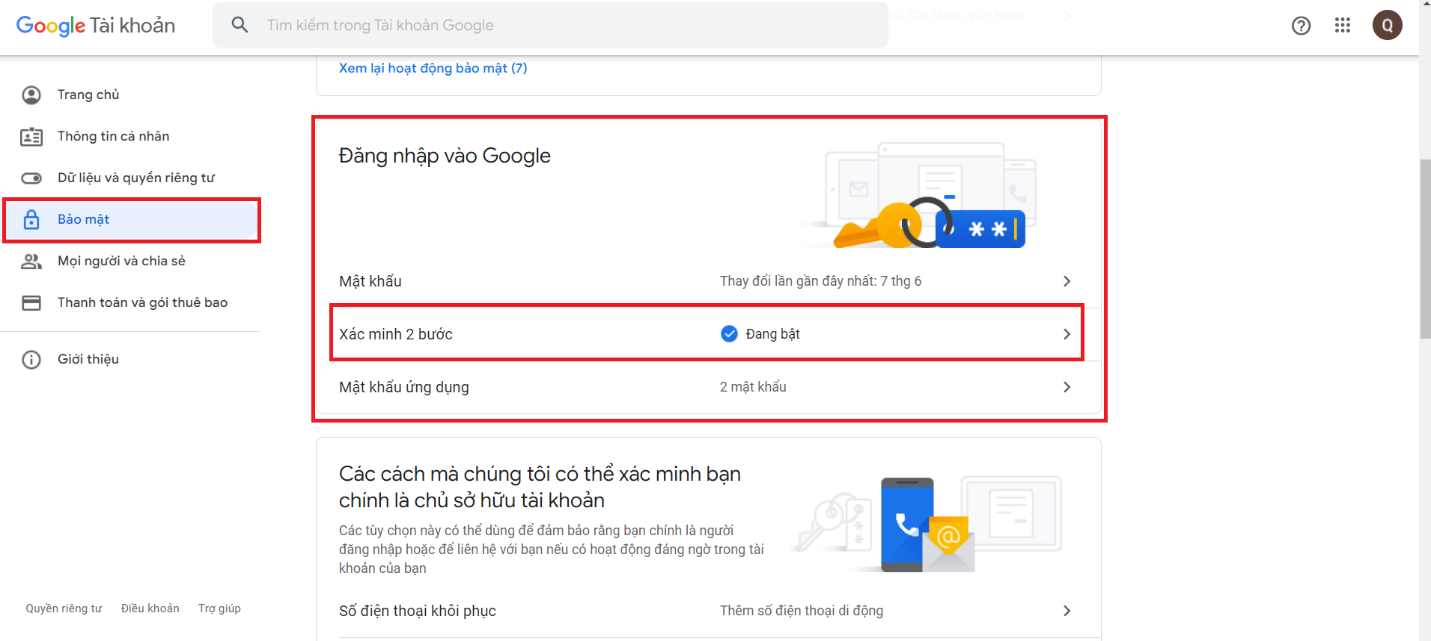
***Ho Chi Minh City, February 2021***

**Chuẩn bị**

**1. Truy cập vào trang web** [**https://myaccount.google.com**](https://myaccount.google.com)

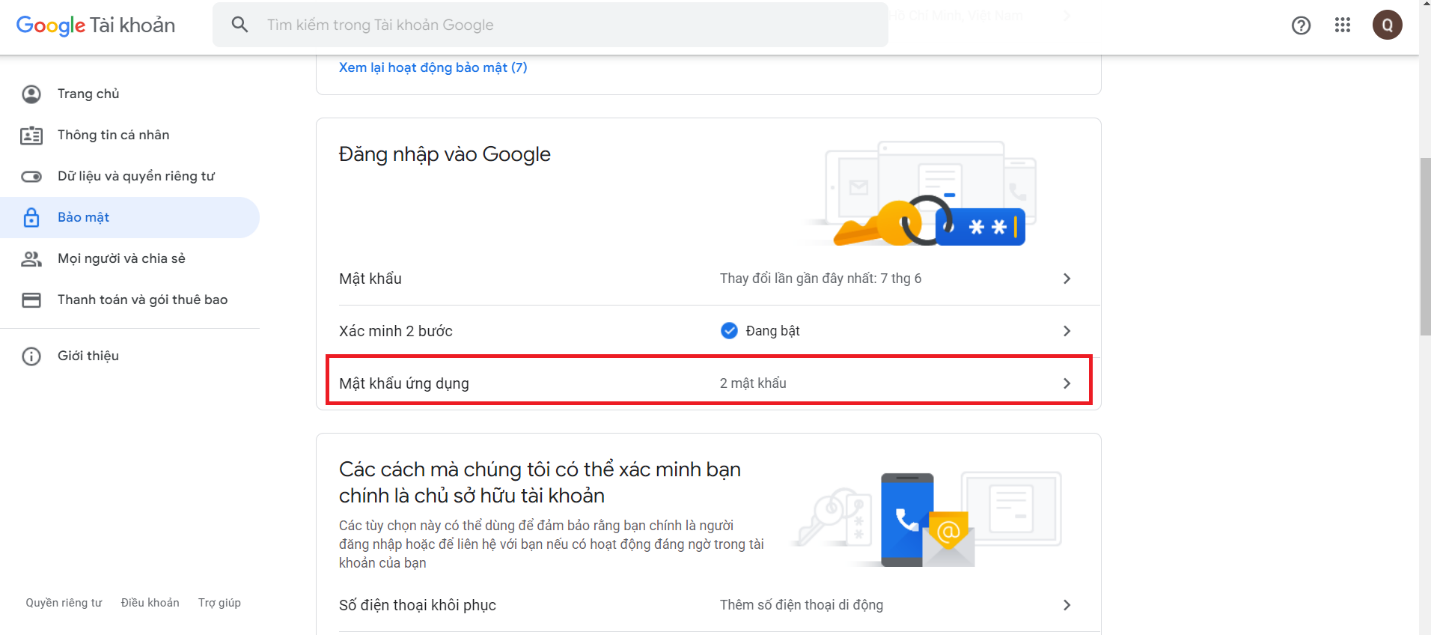
**2. Đăng nhập tài khoản google của bạn**

**3. Vào phần “Bảo mật”, kéo xuống dưới tìm phần xác minh 2 bước, nếu chưa bật hãy thiết lật và bật nó lên (Làm theo hướng dẫn của google)**

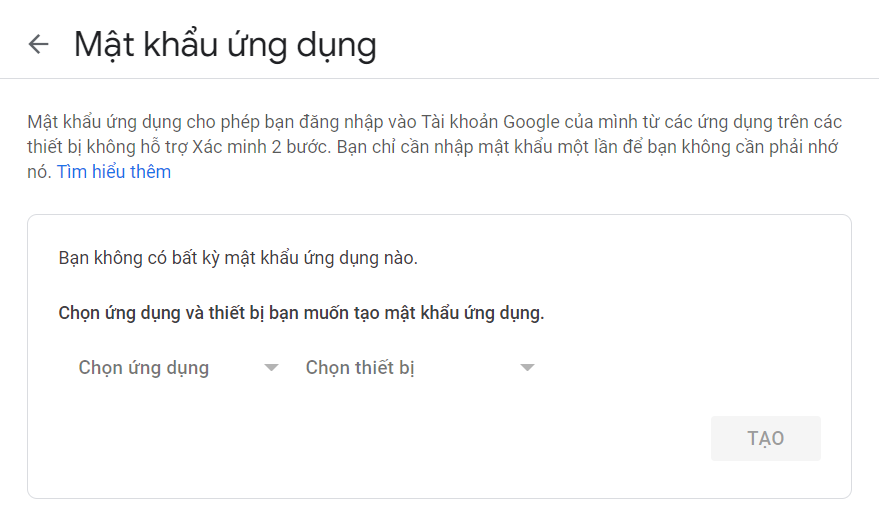
****

**4. Sau khi đã bật xác minh 2 bước, chọn “Mật khẩu ứng dụng”**

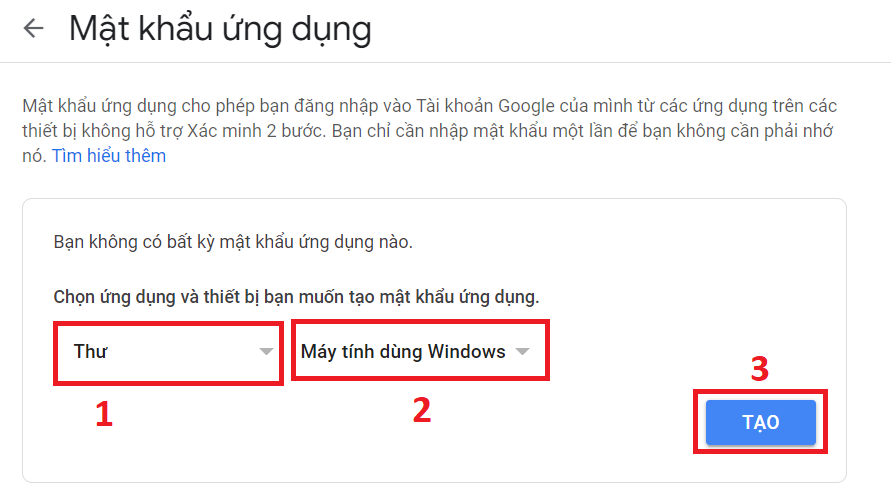
*(Lưu ý: Phần này chỉ xuất hiện khi bạn bật xác minh 2 bước)*

****

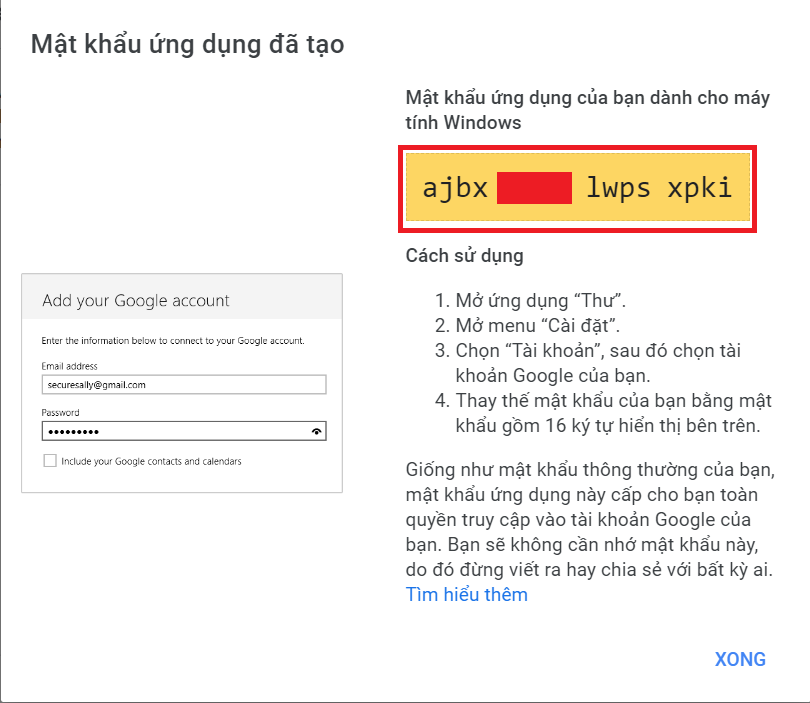
**Sau đó nhập lại mật khẩu google của bạn -> Màn hình hiển thị**

****

**Chọn các bước theo hình:**

****

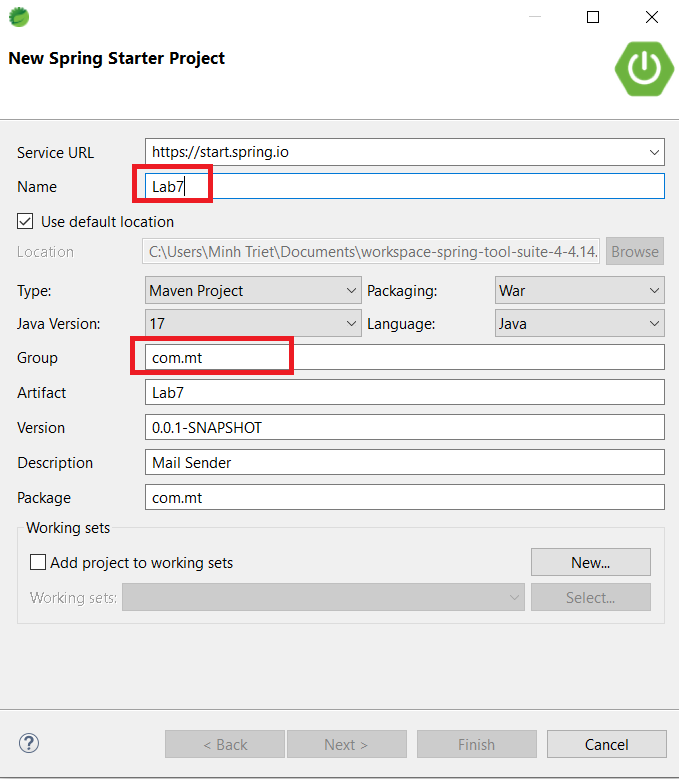
**Màn hình hiện ra mật khẩu ứng dụng**

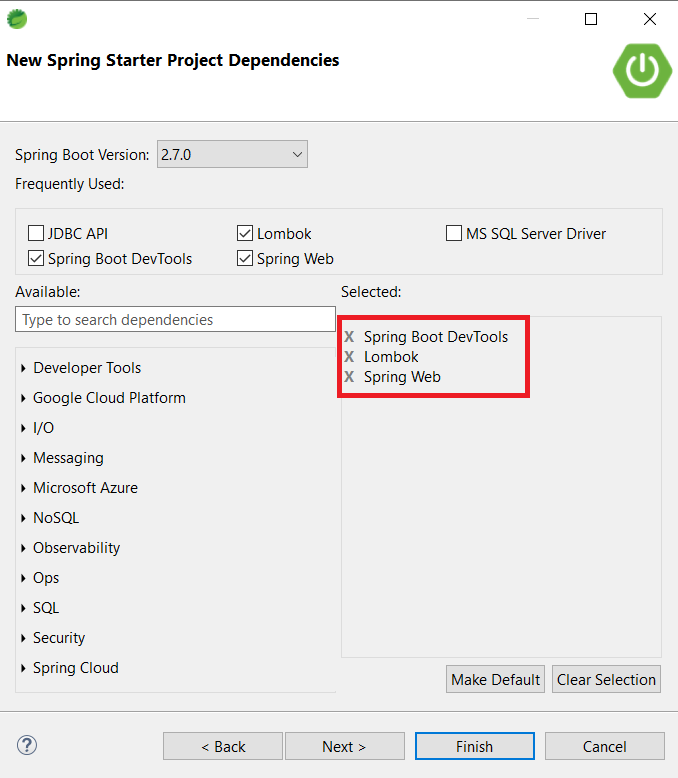
****

**LƯU LẠI MẬT KHẨU ĐÓ (QUAN TRỌNG): ví dụ của mình là:** ajbx\*\*\*\*lwpsxpki

**Hướng dẫn bài 1 : Gửi email đơn giản**

**Tạo Project:**

****

****

**Bổ sung Dependency cho file pom.xml**

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

</dependency>

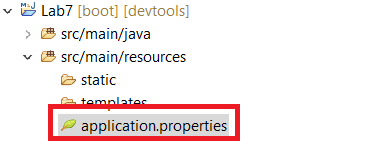
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-mail</artifactId>

</dependency>

**Tìm đến file application.properties**

****

**Thêm vào :**

spring.mvc.view.prefix=/views/

spring.mvc.view.suffix=.jsp

spring.mail.host=smtp.gmail.com

spring.mail.port=587

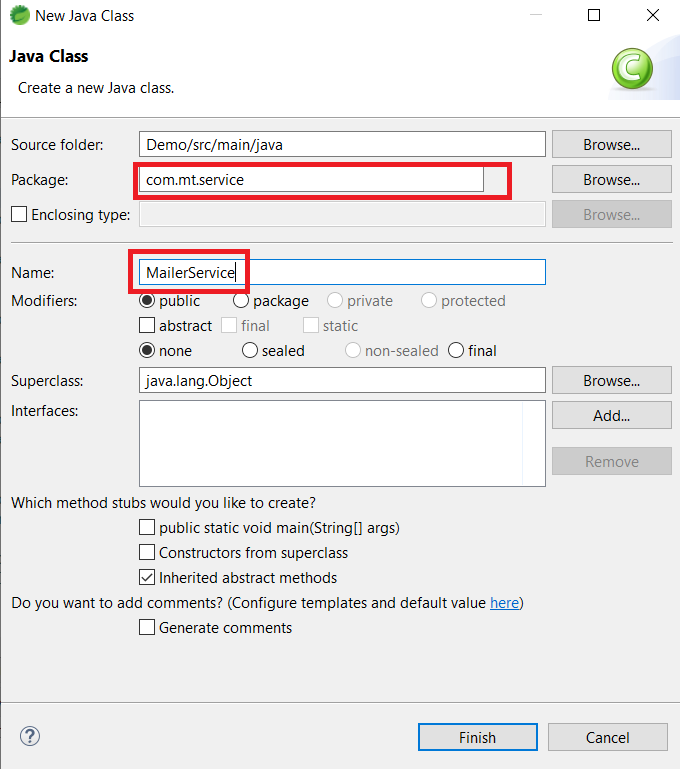
spring.mail.username=Thay bằng email của bạn

spring.mail.password=Thay bằng mật khẩu ứng dụng đã tạo

spring.mail.properties.mail.smpt.auth=true

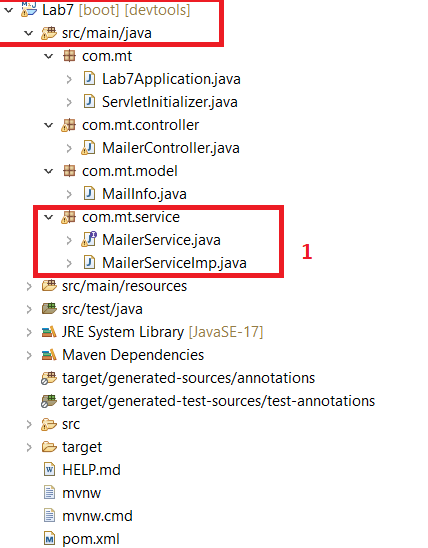
spring.mail.properties.mail.smtp.starttls.enable=true

**Tạo 1 Class nằm trong package như hình:**

****

**Tạo package com.mt.service ->**

1. **Tạo Interface MailerService.java**
2. **Tạo Class MailerServiceImp.java**

****

**Thêm src cho interface MailerService**

**public** **interface** MailerService {

**void** send(MailInfo mail) **throws** MessagingException;

**void** send(String to, String subject, String body) **throws** MessagingException;

}

**Thêm mã nguồn cho class MailerServiceImp**

@Service

**public** **class** MailerServiceImp **implements** MailerService {

@Autowired

JavaMailSender sender;

@Override

**public** **void** send(MailInfo mail) **throws** MessagingException {

MimeMessage message = sender.createMimeMessage();

// Sử dụng Helper để thiết lập các thông tin cần thiết cho message

MimeMessageHelper helper = **new** MimeMessageHelper(message, **true**, "utf-8");

helper.setFrom(mail.getFrom());

helper.setTo(mail.getTo());

helper.setSubject(mail.getSubject());

helper.setText(mail.getBody(), **true**);

helper.setReplyTo(mail.getFrom());

String[] cc = mail.getCc();

**if** (cc != **null** && cc.length > 0) {

helper.setCc(cc);

}

String[] bcc = mail.getBcc();

**if** (bcc != **null** && bcc.length > 0) {

helper.setBcc(bcc);

}

List<File> files = mail.getFiles();

**if** (files.size()>0) {

**for** (File file:files) {

helper.addAttachment(file.getName(), file);

}

}

// Gửi message đến SMTP server

sender.send(message);

}

@Override

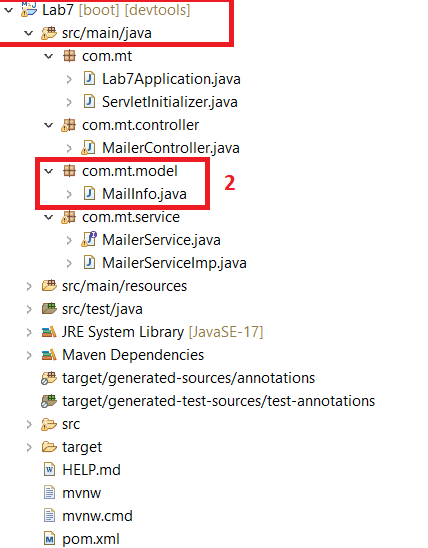
**public** **void** send(String to, String subject, String body) **throws** MessagingException {

**this**.send(**new** MailInfo(to, subject, body));

}

}

**Tạo class package com.mt.model -> Tạo class MailInfo.java**

****

**Thêm mã nguồn cho class MailInfo.java**

@Data

@NoArgsConstructor

@AllArgsConstructor

**public** **class** MailInfo {

String from = "MinhTriet";

String to;

String[] cc;

String[] bcc;

String subject;

String body;

List<File> files=**new** ArrayList<>();

**public** MailInfo(String to, String subject, String body) {

**this**.to = to;

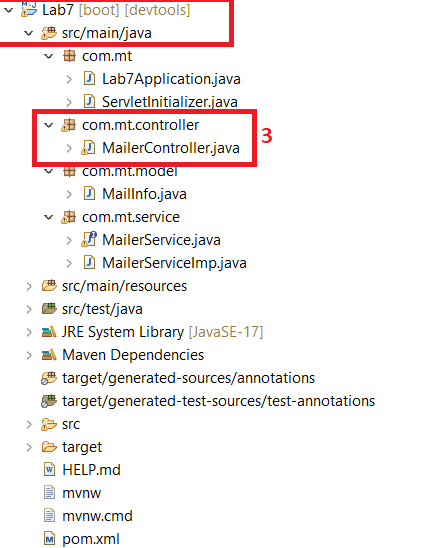
**this**.subject = subject;

**this**.body = body;

}

}

**Tạo package com.mt.controller -> Tạo class MailerController.java**

****

**Thêm mã nguồn cho class MailerController.java**

@Controller

@RequestMapping("mailer")

**public** **class** MailerController {

@Autowired

MailerService mailer;

@GetMapping("")

**public** String mailViewer() {

**return** "index";

}

@ResponseBody

@PostMapping("/send")

**public** String send(Model model,@RequestParam String txtTo,@RequestParam String txtSubject,@RequestParam String txtContent) {

**try** {

MailInfo mail=**new** MailInfo();

mail.setTo(txtTo);

mail.setSubject(txtSubject);

mail.setBody(txtContent);

mailer.send(mail);

**return** "<h1>Gửi email thành công</h1>";

} **catch** (MessagingException e) {

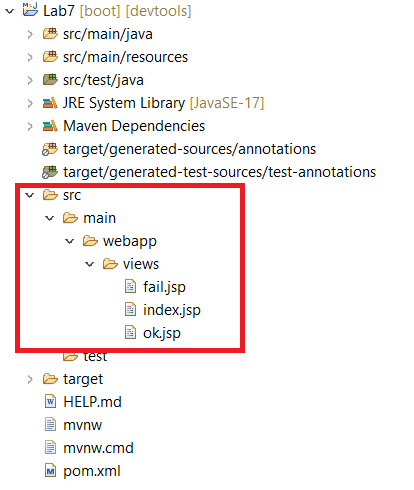
**return** "<h1>Gửi email thất bại</h1>"+e.getMessage();

}

}

}

**Tại thư mục src, tạo các file .jsp theo cấu trúc sau**

****

**File index.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Send mail</title>

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"* rel=*"stylesheet"* integrity=*"sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"* crossorigin=*"anonymous"*>

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"* integrity=*"sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"* crossorigin=*"anonymous"*></script>

<style type=*"text/css"*>

\*{

--formWidth:*460px*;

--btnWidth:*70px*;

--btnCenter:*195px*;

}

*.formMailer*{

width: *var(*-*-formWidth)*;

margin: *auto*;

margin-top: *4rem*;

}

**h2**{

color:*#0000FF*;

}

*.btnSend*{

margin-top: *1.5rem*;

margin-left: *195px*;

width: *var(*-*-btnWidth)*;

}

</style>

</head>

<body>

<form class=*"formMailer"* action=*"/mailer/send"* method=*"post"*>

<h2>Gửi email</h2>

<div class=*"form-group"*>

<label for=*"formControlInput"*>Gửi đến</label> <input

type=*"email"* class=*"form-control"* id=*"formControlInput"* name=*"txtTo"*

placeholder=*"Nhập email người nhận..."*>

</div>

<div class=*"form-group"*>

<label for=*"formControlInput2"*>Tiêu đề</label> <input

type=*"text"* class=*"form-control"* id=*"formControlInput2"* name=*"txtSubject"*

placeholder=*"Nhập tiêu đề..."*>

</div>

<div class=*"form-group"*>

<label for=*"formControlTextarea"*>Nội dung</label>

<textarea class=*"form-control"* id=*"formControlTextarea"* name=*"txtContent"*

rows=*"4"* placeholder=*"Nhập nội dung..."*></textarea>

</div>

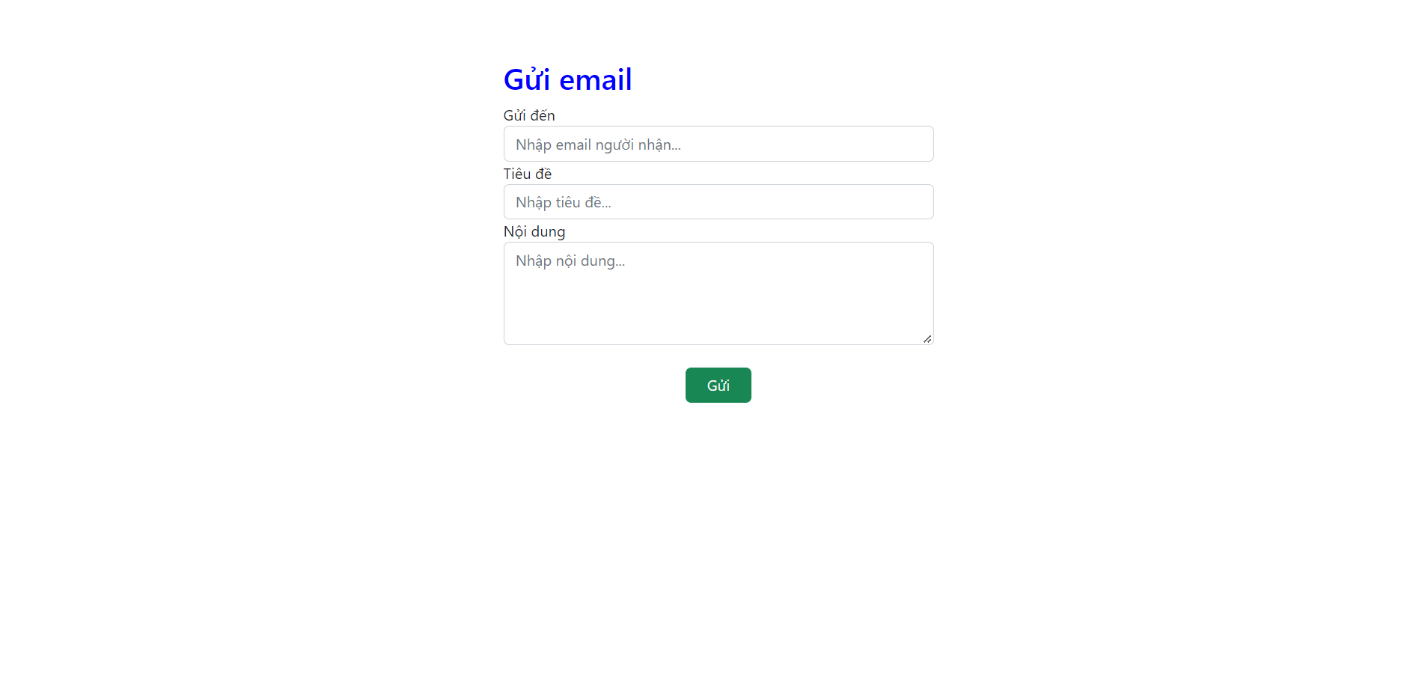
<button class=*"btn btn-success btnSend"*>Gửi</button>

</form>

</body>

</html>

Giao diện Bài 1:



**Hướng dẫn bài 2 : Mục đích: Tối ưu hiệu suất gửi email ở bài 1**

**Trong package com.mt -> Mở class Lab7Application.java**

**Bổ sung** @EnableScheduling

@SpringBootApplication

@EnableScheduling

**public** **class** Lab7Application {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(Lab7Application.**class**, args);

}

}

**Trong package com.mt.service -> Mở interface MailerService.java**

**Bổ sung 2 phương thức sau:**

**void** queue(MailInfo mail);

**void** queue(String to, String subject, String body);

**Trong package com.mt.service -> Mở class MailerServiceImp.java**

List<MailInfo> list = **new** ArrayList<>();

@Override

**public** **void** queue(MailInfo mail) {

list.add(mail);

}

@Override

**public** **void** queue(String to, String subject, String body) {

queue(**new** MailInfo(to, subject, body));

}

@Scheduled(fixedDelay = 5000)

**public** **void** run() {

**while** (!list.isEmpty()) {

MailInfo mail = list.remove(0);

**try** {

**this**.send(mail);

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

**Trong package com.mt.controller -> Mở class MailerController.java**

**Sửa lại phần source:**

@ResponseBody

@PostMapping("/send")

**public** String send(Model model, @RequestParam String txtTo,

@RequestParam String txtSubject, @RequestParam String txtContent) {

MailInfo mail = **new** MailInfo();

mail.setTo(txtTo);

mail.setSubject(txtSubject);

mail.setBody(txtContent);

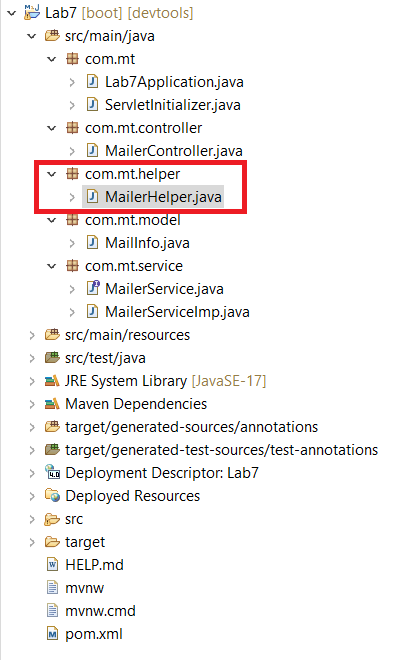
mailer.queue(mail);

**return** "Mail của bạn đã được gửi đi";

}

**Hướng dẫn bài 3 : Xây dựng trang web cho phép người nhập đầy đủ thông tin (from, to, cc, bcc, subject, body, attachments) vào form gửi email, sử dụng MailerService để gửi email**

**Tạo 1 package com.mt.helper -> Tạo class MailerHelper.java**



**Bổ sung source:**

**public** **class** MailerHelper {

**public** String[] parseStringEmailToArray(String emailString) {

String[] arrEmail = **null**;

**if** (emailString.length() > 0) {

emailString = removeSpace(emailString);

arrEmail = emailString.split(",");

}

**return** arrEmail;

}

**private** String removeSpace(String string) {

**return** string.replaceAll(" ", "");

}

**public** File convertMultipartFileToFile(MultipartFile multipartFile) **throws** IOException {

//Tạo 1 thư mục tạm thời để chứa file, sau này có thể xóa thư mục tạm thời đó đi

File convFile = **new** File(System.*getProperty*("java.io.tmpdir") + "/" + multipartFile.getOriginalFilename());

multipartFile.transferTo(convFile);

**return** convFile;

}

}

**Sửa source cho method send MailerController.java:**

@ResponseBody

@PostMapping("/send")

**public** String send(Model model,

@RequestParam String txtTo,

@RequestParam String txtCC,

@RequestParam String txtBCC,

@RequestParam String txtSubject,

@RequestParam String txtContent,

@RequestParam("file") MultipartFile multipartFile,

@RequestParam("file1") MultipartFile multipartFile1) **throws** IOException {

MailerHelper helper=**new** MailerHelper();

List<File> files=**new** ArrayList<>();

String[] emailCC=helper.parseStringEmailToArray(txtCC);

String[] emailBCC=helper.parseStringEmailToArray(txtBCC);

MailInfo mail = **new** MailInfo();

mail.setTo(txtTo);

mail.setCc(emailCC);

mail.setBcc(emailBCC);

mail.setSubject(txtSubject);

mail.setBody(txtContent);

//covert MultipartFile to File

**if**(!multipartFile.isEmpty()) {

File file=helper.convertMultipartFileToFile(multipartFile);

files.add(file);

//Set cho MailInfo

mail.setFiles(files);

}

**if**(!multipartFile1.isEmpty()) {

File file=helper.convertMultipartFileToFile(multipartFile1);

files.add(file);

//Set cho MailInfo

mail.setFiles(files);

}

//Gửi mail

mailer.queue(mail);

**return** "<h1>Mail của bạn đã được gửi đi</h1>";

}

**Sửa file index.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Send mail</title>

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"* rel=*"stylesheet"* integrity=*"sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"* crossorigin=*"anonymous"*>

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js"* integrity=*"sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2"* crossorigin=*"anonymous"*></script>

<style type=*"text/css"*>

\*{

--formWidth:*460px*;

--btnWidth:*70px*;

--btnCenter:*195px*;

}

*.formMailer*{

width: *var(*-*-formWidth)*;

margin: *auto*;

margin-top: *1rem*;

}

**h2**{

color:*#0000FF*;

}

*.btnSend*{

margin-top: *1.5rem*;

margin-left: *195px*;

width: *var(*-*-btnWidth)*;

font-size:*25px*;

}

*.btnMore*{

margin-top:*1rem*;

}

</style>

</head>

<body>

<form enctype=*"multipart/form-data"* class=*"formMailer"* action=*"/mailer/send"* method=*"post"*>

<h2>Gửi email</h2>

<div class=*"form-group"*>

<label for=*"formControlInput"*>To</label> <input

type=*"email"* class=*"form-control"* id=*"formControlInput"* name=*"txtTo"*

placeholder=*"Nhập email người nhận..."*>

</div>

<div class=*"form-group"*>

<label for=*"formControlInput2"*>CC</label> <input

type=*"text"* class=*"form-control"* id=*"formControlInput2"* name=*"txtCC"*

placeholder=*"Nhập email(cc) ngăn cách nhau bởi dấu ','"*>

</div>

<div class=*"form-group"*>

<label for=*"formControlInput3"*>BCC</label> <input

type=*"text"* class=*"form-control"* id=*"formControlInput3"* name=*"txtBCC"*

placeholder=*"Nhập email(bcc) ngăn cách nhau bởi dấu ','"*>

</div>

<div class=*"form-group"*>

<label for=*"formControlInput4"*>Tiêu đề</label> <input

type=*"text"* class=*"form-control"* id=*"formControlInput4"* name=*"txtSubject"*

placeholder=*"Nhập tiêu đề..."*>

</div>

<div class=*"form-group"*>

<label for=*"formControlTextarea"*>Nội dung</label>

<textarea class=*"form-control"* id=*"formControlTextarea"* name=*"txtContent"*

rows=*"4"* placeholder=*"Nhập nội dung..."*></textarea>

</div>

<div class=*"form-group"*>

<label for=*"formControlInput4"*>File đính kèm</label>

<input

type=*"file"* class=*"form-control"* id=*"formControlInput4"* name=*"file"*

>

</div>

<div id=*"fileMore"* class=*"form-group"* style="display: *none*;">

<label for=*"formControlInput5"*>File đính kèm</label>

<input

type=*"file"* class=*"form-control"* id=*"formControlInput5"* name=*"file1"*

>

</div>

<div id=*"btnMore"* class=*"btn btn-info btnMore"*>

<svg xmlns=*"http://www.w3.org/2000/svg"* width=*"16"* height=*"16"* fill=*"currentColor"* class=*"bi bi-plus-circle"* viewBox=*"0 0 16 16"*>

<path d=*"M8 15A7 7 0 1 1 8 1a7 7 0 0 1 0 14zm0 1A8 8 0 1 0 8 0a8 8 0 0 0 0 16z"*/>

<path d=*"M8 4a.5.5 0 0 1 .5.5v3h3a.5.5 0 0 1 0 1h-3v3a.5.5 0 0 1-1 0v-3h-3a.5.5 0 0 1 0-1h3v-3A.5.5 0 0 1 8 4z"*/>

</svg>

Thêm tệp</div>

<br/>

<button class=*"btn btn-success btnSend"*>Gửi</button>

</form>

<script type=*"text/javascript"*>

**const** btnMore=document.getElementById('btnMore');

**const** fileMore = document.getElementById("fileMore");

btnMore.addEventListener('click',()=>{

fileMore.style.display="block";

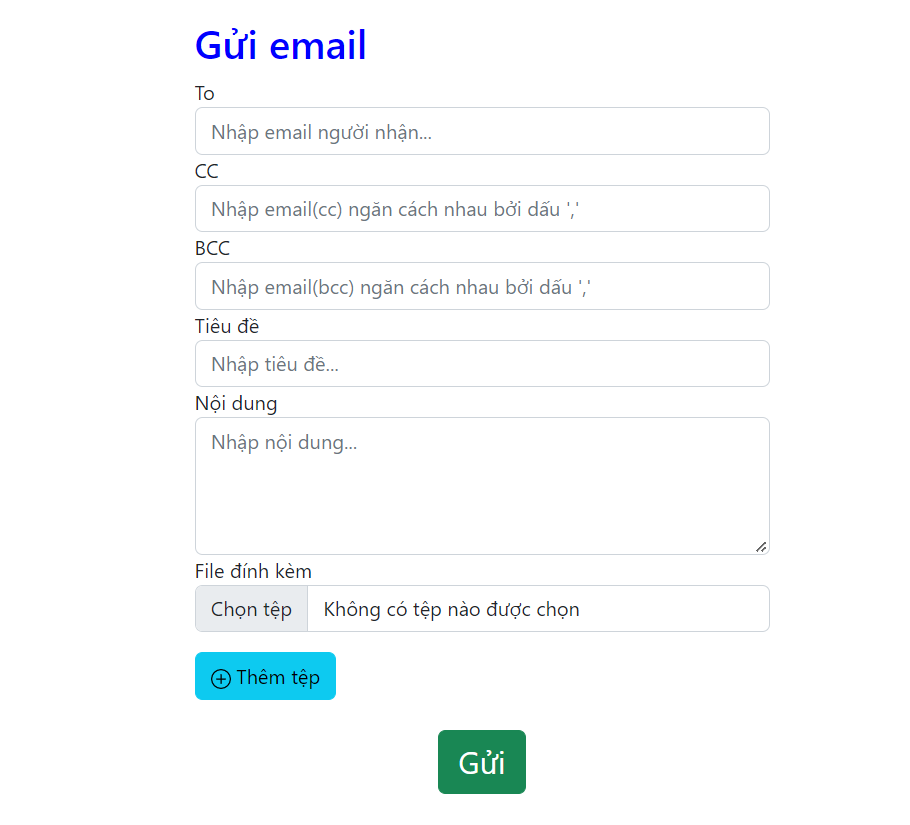
});

</script>

</body>

</html>

**Giao diện bài 3:**

****

**Hướng dẫn bài 4 : Hiển thị tất cả loại hàng và địa chỉ truy cập trên tất cả mọi view của ứng dụng, biết cách cấu hình Interceptor.**

**Tạo 1 project spring mới đặt tên là Lab7\_2 (spring web, loombok, spring boot devtools)**

**Thiết lập file application.properties:**

spring.mvc.view.prefix=/views/

spring.mvc.view.suffix=.jsp

**Bổ sung các depedency còn thiếu ở pom.xml**

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

</dependency>

<dependency>

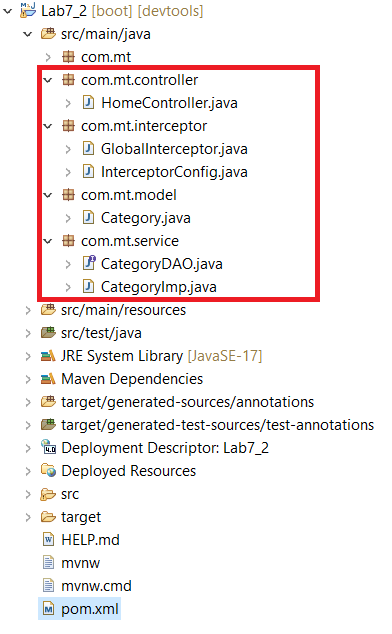
<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

</dependency>

</dependencies>

**Tạo ra các package chứa các class và interface, đặt tên giống như ảnh:**

****

**Sourc cho file InterceptorConfig.java**

@Configuration

**public** **class** InterceptorConfig **implements** WebMvcConfigurer {

@Autowired

GlobalInterceptor global;

@Override

**public** **void** addInterceptors(InterceptorRegistry registry) {

registry.addInterceptor(global)

.addPathPatterns("/\*\*")

.excludePathPatterns("/assets/\*\*");//ngoại trừ URI assets

}

}

**Sourc cho file model Category.java**

@Data

@NoArgsConstructor

@AllArgsConstructor

**public** **class** Category {

**private** **int** id;

**private** String name;

}

**Sourc cho file interface CategoryDAO.java**

**public** **interface** CategoryDAO {

**public** List<Category> findAll();

}

**Sourc cho file model CategoryImp.java**

@Service

**public** **class** CategoryImp **implements** CategoryDAO {

**public** **static** List<Category> *list*=**new** ArrayList<>();

**static** {

*list*.add(**new** Category(1,"Đồng hồ đeo tay"));

*list*.add(**new** Category(1,"Máy tính xách tay"));

*list*.add(**new** Category(1,"Máy ảnh"));

*list*.add(**new** Category(1,"Điện thoại"));

*list*.add(**new** Category(1,"Nước hoa"));

*list*.add(**new** Category(1,"Nữ trang"));

*list*.add(**new** Category(1,"Nón thời trang"));

*list*.add(**new** Category(1,"Túi xách du lịch"));

}

@Override

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

**return** *list*;

}

}

**Sourc cho file GlobalInterceptor.java**

@Service

**public** **class** GlobalInterceptor **implements** HandlerInterceptor {

@Autowired

CategoryDAO dao;

@Override

**public** **boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

**throws** Exception {

request.setAttribute("uri", request.getRequestURI());

**return** **true**;

}

@Override

**public** **void** postHandle(HttpServletRequest req, HttpServletResponse res, Object handler, ModelAndView mv)

**throws** Exception {

req.setAttribute("categories", dao.findAll());

}

}

**Sourc cho file HomeController.java**

@Controller

**public** **class** HomeController {

@RequestMapping("/home/index")

**public** String index() {

**return** "index";

}

@RequestMapping("/home/about")

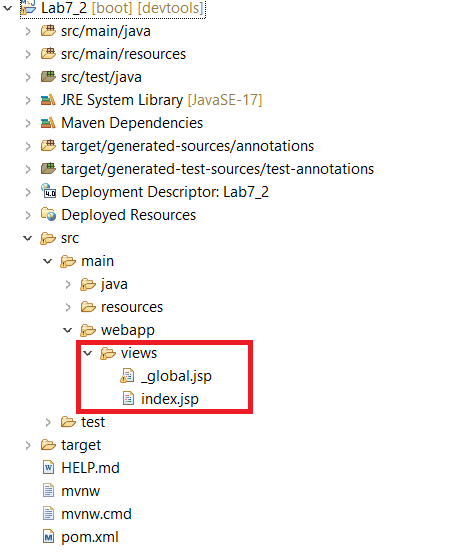
**public** String about() {

**return** "home/about";

}

}

**Tạo 2 file .jsp theo ảnh sau:**

****

**Sourc cho file \_global.jsp**

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

pageEncoding=*"UTF-8"*%>

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

<%@ taglib uri=*"http://java.sun.com/jstl/fmt\_rt"* prefix=*"fmt"*%>

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<b>${uri}</b>

<c:forEach var=*"item"* items=*"*${categories}*"*>

<li>${item.name}</li>

</c:forEach>

</body>

</html>

**Sourc cho file index.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Home</title>

</head>

<body>

<h3>HOME PAGE</h3>

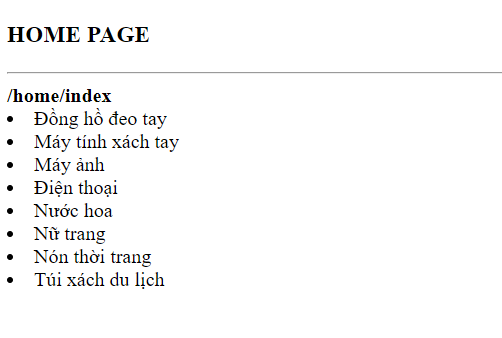
<hr>

<jsp:include page=*"\_global.jsp"*/>

</body>

</html>

**Kết quả:**

****

**Hướng dẫn bài 5 : Xây dựng AuthInterceptor và cấu hình để bảo mật ứng dụng.**

**Thêm các interface và class như hình:**

****

**Thêm sourc vào model Account.java**

@Data

@AllArgsConstructor

@NoArgsConstructor

**public** **class** Account {

**private** String username;

**private** String password;

**private** **boolean** admin;

}

**Thêm sourc vào model Session.java**

@Data

@NoArgsConstructor

@AllArgsConstructor

**public** **class** Session {

**private** String key;

**private** Object value;

}

**Thêm sourc vào AuthInterConfig.java**

@Configuration

**public** **class** AuthInterConfig **implements** WebMvcConfigurer {

@Autowired

AuthInterceptor auth;

@Override

**public** **void** addInterceptors(InterceptorRegistry registry) {

registry.addInterceptor(auth).addPathPatterns("/account/\*\*","/order/\*\*", "/admin/\*\*")

.excludePathPatterns("/assets/\*\*","/account/login", "/admin/index");

}

}

**Thêm sourc vào AuthIntercaptor.java**

@Service

**public** **class** AuthInterceptor **implements** HandlerInterceptor {

@Autowired

SessionService session;

@Override

**public** **boolean** preHandle(HttpServletRequest request, HttpServletResponse response, Object handler)

**throws** Exception {

String uri = request.getRequestURI();

Account user = (Account) session.get("user"); // lấy từ session

String error = "";

**if** (user == **null**) { // chưa đăng nhập

error = "Please login!";

}

// không đúng vai trò

**else** **if** (!user.isAdmin() && uri.startsWith("/admin/")) {

error = "Access denied!";

}

**if** (error.length() > 0) { // có lỗi

session.set("security-uri", uri);

response.sendRedirect("/account/login?error=" + error);

**return** **false**;

}

**return** **true**;

}

}

**Thêm sourc vào AccountDAO.java**

**public** **interface** AccountDAO {

**public** Account getOne(String username);

}

**Thêm sourc vào AccountImp.java**

@Service

**public** **class** AccountImp **implements** AccountDAO {

**public** **static** List<Account> *listAccount* = **new** ArrayList<>();

**static** {

*listAccount*.add(**new** Account("pnmtriet","123",**true**));//Admin

*listAccount*.add(**new** Account("nvtriet","123",**false**));//Customer

}

@Override

**public** Account getOne(String username) {

Account account=**new** Account();

**if**(username.isEmpty()||username=="") {

account=**null**;

**return** **null**;

}

**for**(Account item:*listAccount*) {

**if**(item.getUsername().equalsIgnoreCase(username)) {

account=item;

}

}

**return** account;

}

}

**Thêm sourc vào SessionService.java**

**public** **interface** SessionService {

**public** Object get(String key); //Get theo key trả về type

**public** **void** set(String key, Object value); //set key là string, type tùy ý

}

**Thêm sourc vào SessionImp.java**

@Service

**public** **class** SessionImp **implements** SessionService {

**public** **static** List<Session> *listSession*=**new** ArrayList<>();

@Override

**public** Object get(String key) {

Object value = **null**;

**if**(key.isEmpty()||key=="") {

**return** **null**;

}

**for**(Session item:*listSession*) {

**if**(key.equalsIgnoreCase(item.getKey())) {

value=item.getValue();

}

}

**return** value;

}

@Override

**public** **void** set(String key, Object value) {

**for**(**int** i=0;i<*listSession*.size();i++) {

Session item=*listSession*.get(i);

**if**(key.equalsIgnoreCase(item.getKey())) {

*listSession*.remove(i);

}

}

Session session=**new** Session();

session.setKey(key);

session.setValue(value);

*listSession*.add(session);

}

}

**Thêm source vào AccountController.java**

@Controller

**public** **class** AccountController {

@Autowired

AccountDAO dao;

@Autowired

SessionService session;

@GetMapping("/account/login")

**public** String login() {

**return** "account/login";

}

@GetMapping("/admin/index")

**public** String adminInex() {

**return** "admin/index";

}

@GetMapping("/admin/loginIndex")

**public** String loginIndex() {

**return** "admin/loginIndex";

}

@PostMapping("/account/login")

**public** String login(Model model, @RequestParam("username") String username,

@RequestParam("password") String password) {

**try** {

Account user = dao.getOne(username);

**if** (!user.getPassword().equals(password)) {

model.addAttribute("message", "Invalid password");

} **else** {

session.set("user", user);

String uri = (String) session.get("security-uri");

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

**return** "redirect:" + uri;

} **else** {

model.addAttribute("message", "Login succeed");

}

}

} **catch** (Exception e) {

model.addAttribute("message", "Invalid username");

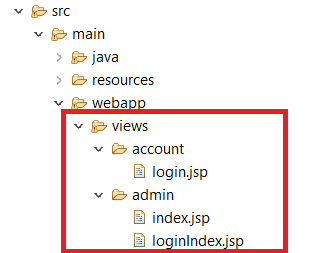
}

**return** "account/login";

}

}

**Tạo ra các file .jsp theo cấu trúc như hình:**

****

**Thêm source cho file account/login.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

</head>

<body>

<h3>LOGIN</h3>

<b><i>${message}${param.error}</i></b>

<form action=*"/account/login"* method=*"post"*>

<input name=*"username"* placeholder=*"Username?"*> <input

name=*"password"* placeholder=*"Password?"*>

<button>Login</button>

</form>

</body>

</html>

**Thêm source cho file admin/index.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Admin index</title>

</head>

<body>

<h1>ĐÂY LÀ TRANG WEB CHÙA (KHÔNG CẦN ĐĂNG NHẬP VẪN XEM ĐƯỢC)</h1>

</body>

</html>

**Thêm source cho file admin/loginIndex.jsp**

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

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

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

<title>Admin</title>

</head>

<body>

<h1>ĐÂY LÀ TRANG PHẢI LOGIN BẰNG ADMIN MỚI ĐƯỢC XEM</h1>

</body>

</html>

HOÀN THÀNH!