**BÀI 3: ADD TEMPLATE BOOTSTRAP**

**Web dạng Business:**

+ Phần web: Là giao diện khi người dùng truy cập vào

+ Phần admin: Là các giao diện dành cho người quản trị

**Trang dowload template:** [Free Bootstrap Themes & Templates - Start Bootstrap](https://startbootstrap.com/themes)

+ Phần administrative: [SB Admin - Free Bootstrap Admin Template - Start Bootstrap](https://startbootstrap.com/template/sb-admin)

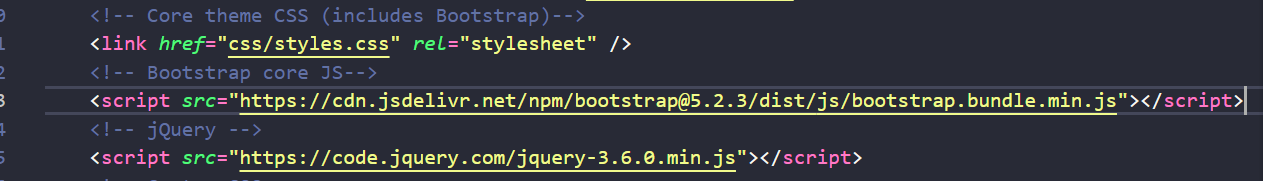
+ Phần user:

**Các bước:**

1. Trong webapp, tạo 1 folder template

+ Nếu muốn apply giao diện bootstrap thì cần tối thiểu 3 file: bootstrap.min.css, bootstrap.min.js và một đường link dẫn tới jQuery.js

+ Ngoài ra thì cần đưa thư mục jQuery vào và truyền đường dẫn tới các file chung trong decorator để có thể code cú pháp jQuery



Ngoài ra ta có thể custom lại css nếu muốn

+ Đưa các file cần thiết cho vào template theo hình

A screenshot of a computer

Description automatically generated

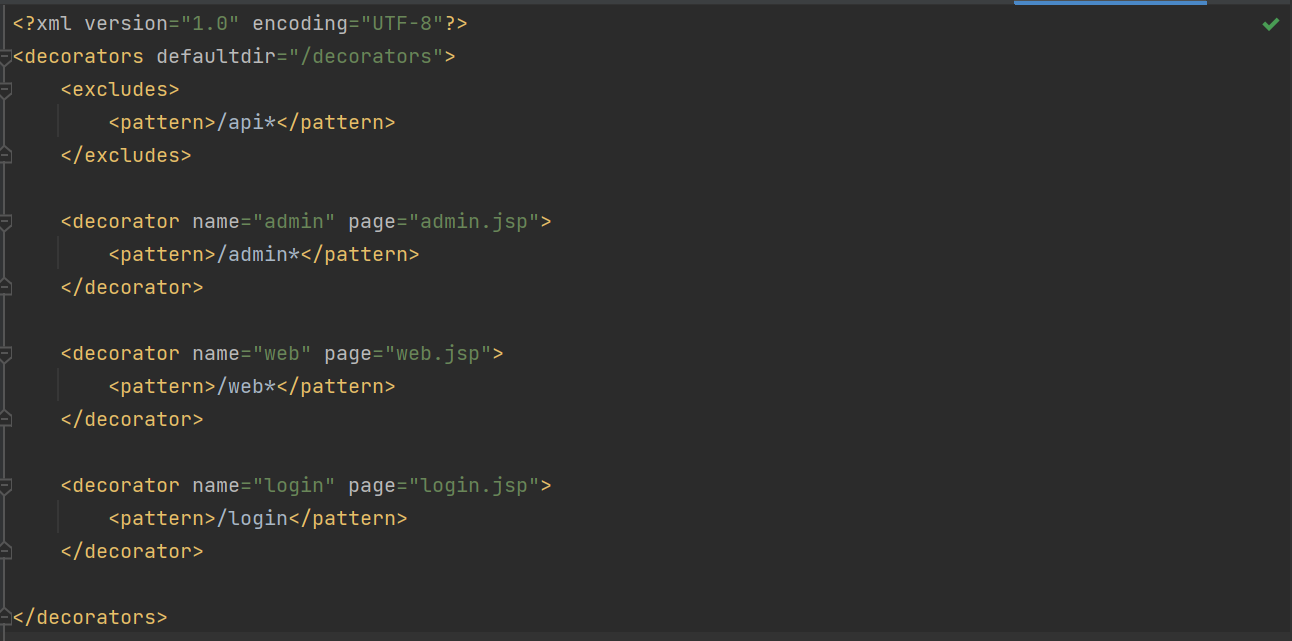
2. Sử dụng sitemesh decorator để tiến hành phân chia giao diện:

+ Phương thức hoạt động: Khi người dùng request vào một url bất kỳ, url này sẽ đi qua filter của decorator, url được tiến hành kiểm tra phần đầu tiên (ví dụ url có phần mở đầu là \admin-.... thì sẽ được apply giao diện của admin)

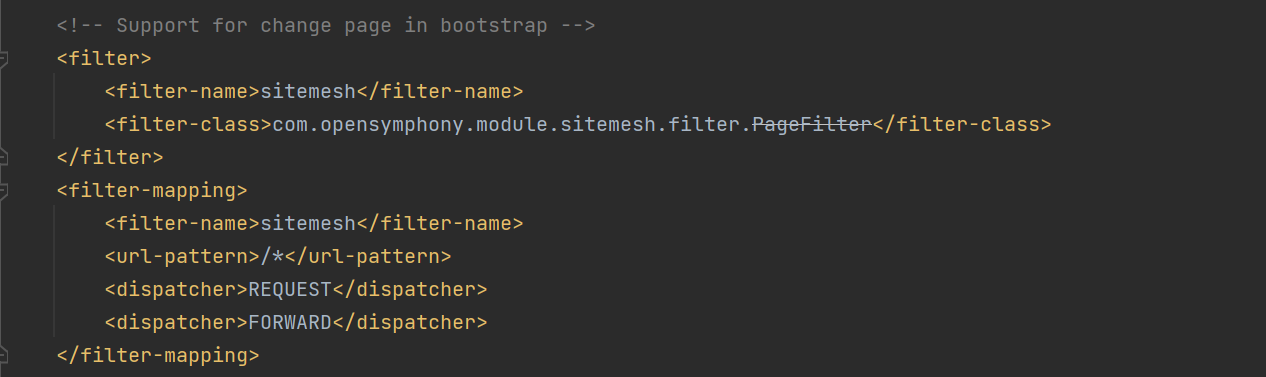
+ Tải dependency đưa vào pom.xml

+ Tạo file decorators.xml trong WEB-INF: File sẽ chứa các đoạn để giúp project có thể xác định và tiến hành filter (được khai báo trong web.xml)

Ví dụ như người dùng request vào admin, thì nó sẽ tự biến tìm đến đường dẫn là admin để truy cập (mà trong /admin thì sẽ dẫn đến trang home đã quy định ở controller)



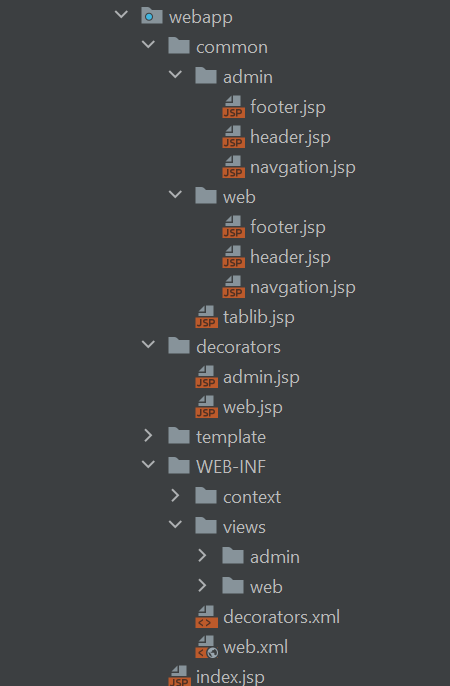
+ Trong webapp.xml đưa khai báo filter vào: Với đoạn filter này, khi apply vào một giao diện thì project sẽ tự động biết được nên apply vào phần nào



+ Tạo thư mục decorators trong webapp, trong đây sẽ chứa 2 master laylout cho admin và user để các phần sau chỉ việc kế thừa 2 cái này. Trong đây định nghĩa cho các phần như header, footer, nav, và body

+ Tiếp theo tạo thư mục common, trong đó tạo các thư mục như admin, user,... để quy định các phần chung (trong mỗi thư mục này sẽ chứa header, footer, navigation-bar). Chỉ có riêng phần body sẽ được tay đổix`

+ Phần con sẽ nằm trong thư mục views. Tại đây tạo 2 folder admin và web để chứa các file jsp con (là body) để một lúc sau ta truyền đường dẫn vào đây



3. Dùng jstl để định nghĩa đường dẫn giao diện

+ Sử dụng cú pháp c:url để định nghĩa cấu trúc cho các đường dẫn css và js

4. Định nghĩa các package controller của admin và web

+ Tạo ra 2 package admin và web để quản lý riêng các controller

+ Để phân biệt class homePage thì thêm giá trị value vào @Controller của mỗi class