**Advanced Web Programming – Routing, Controller, dan View**

**Week 2**



**Baskoro Seno Aji**

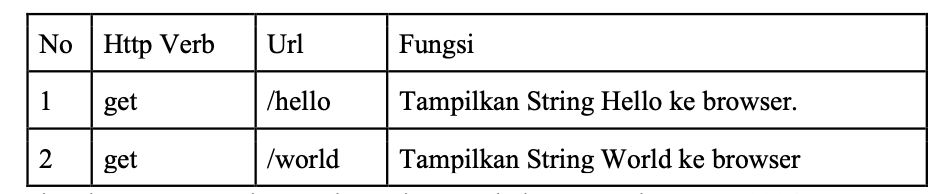
**2341720063 / 05 / TI-2I**

**Github Link :** [**Github**](https://github.com/Anothaatype/University/tree/main/Semester-4/Pemrograman-Web-Lanjut/Week-2)

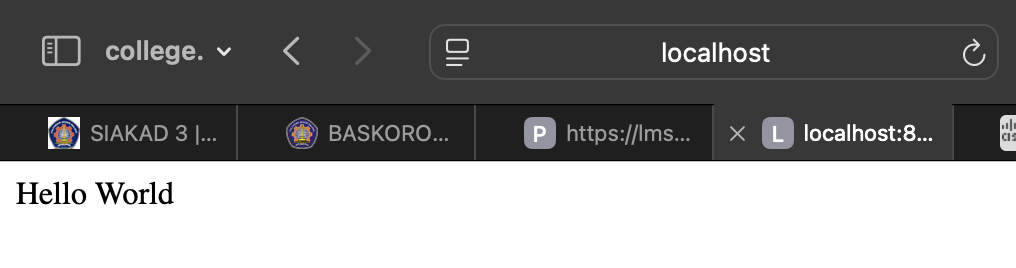
**Polytechnic State of Malang**

**Basic Routing**

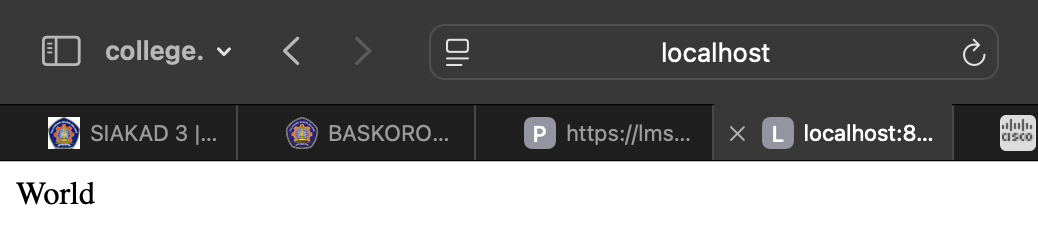
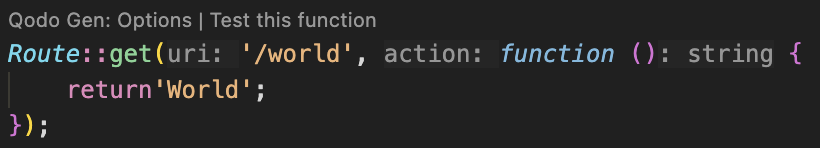
**Langkah – Langkah Praktikum 1 :**

****

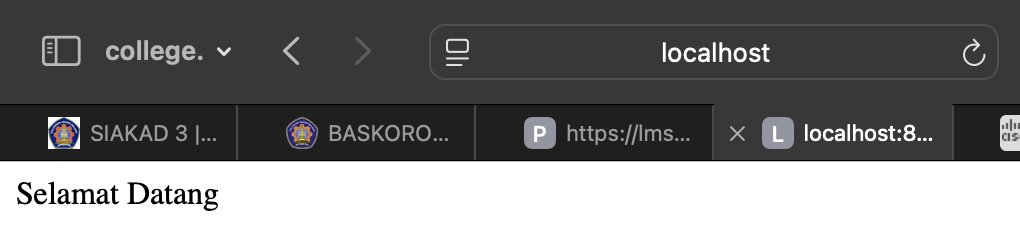
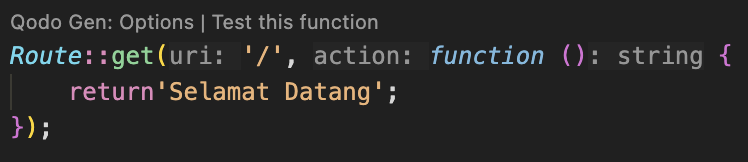
1. This Defines the GET routes where it will be direct to the /hello url by stating it into the route, this functions will immediately return the String ‘ Hello World “ and display it on the web browser

****

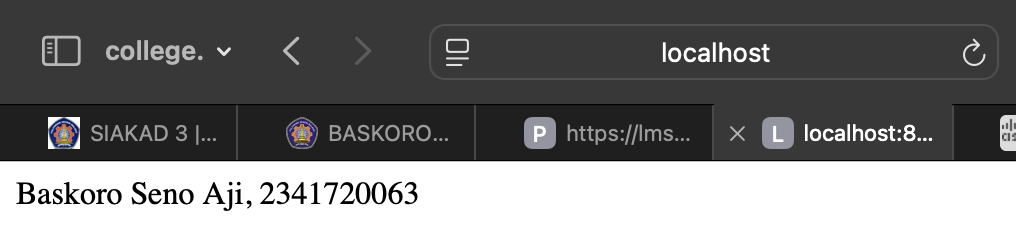
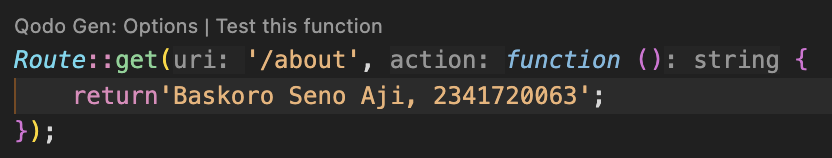
1. This Defines the GET routes where it will be direct to the /world url by stating it into the route, this functions will immediately return the String ‘ World ‘ and display it on the web browser



1. This Defines the GET routes where it will be direct to the / url by stating it into the route, this functions will immediately return the String ‘ Selamat Datang ‘ and display it on the web browser

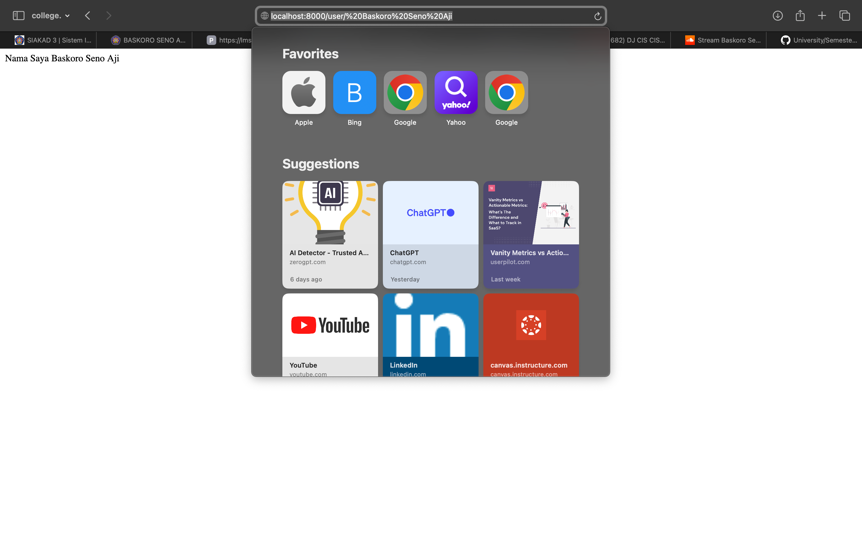
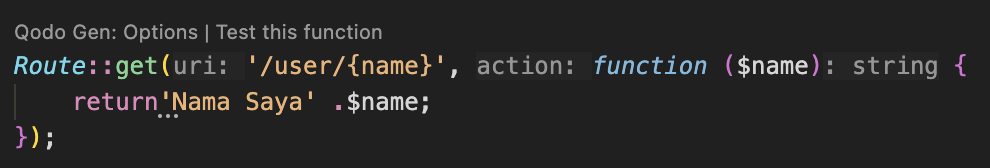
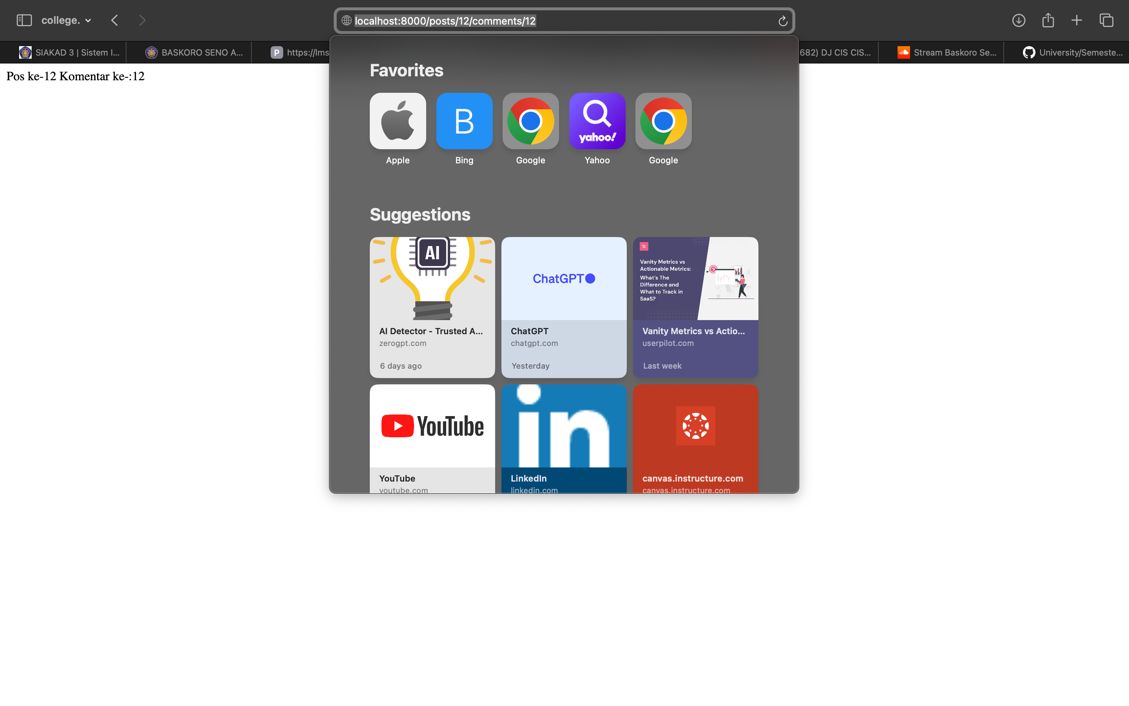


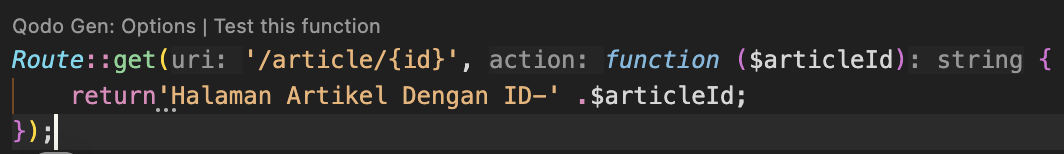
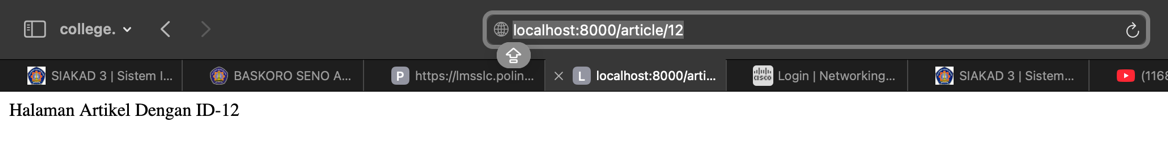
1. This Defines the GET routes where it will be direct to the /about url by stating it into the route, this functions will immediately return the String ‘ Baskoro Seno Aji, 2341720063 ‘ or Name and Nim and display it on the web browser



**Route Parameters**

**Langkah-Langkah Praktikum 2 :**

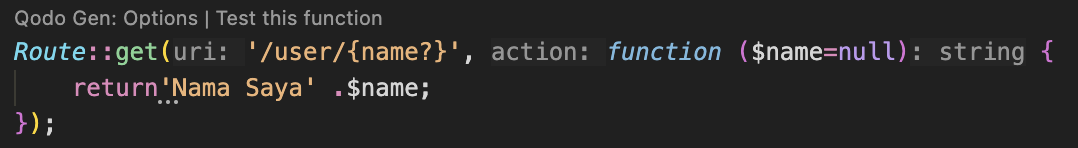
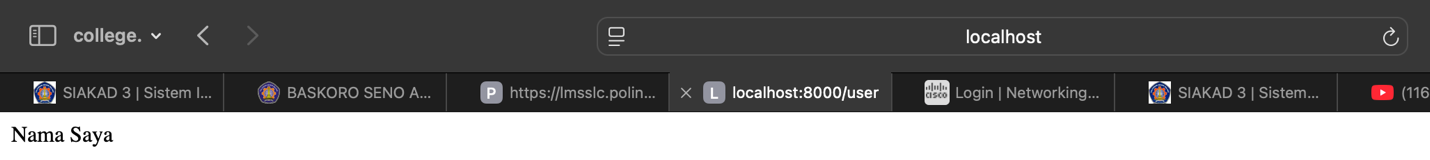
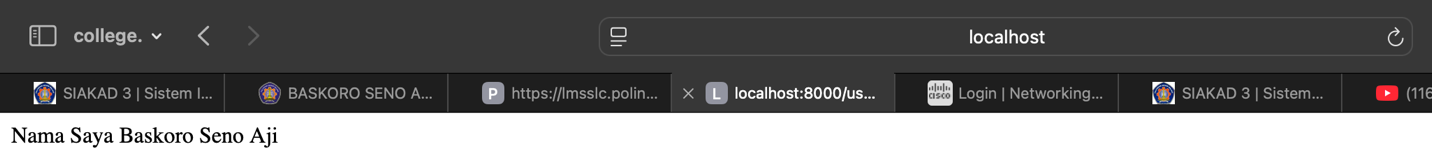
1. This defines a GET route that send parameters in the form of username $name, this function return the strings that we type in the link url as state by that $name 
2. This defines a GET route with two Dynamic Parameters: {post} and {comment}, this parameters used are $postId and $commentId, where it will take the parameters in the link url that we’ve type.
3. This defines a GET route request that accept a dynamic parameters in the form of {id}, and passes it to the function as $articleId.



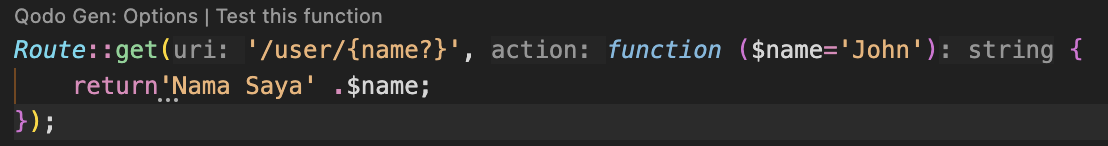
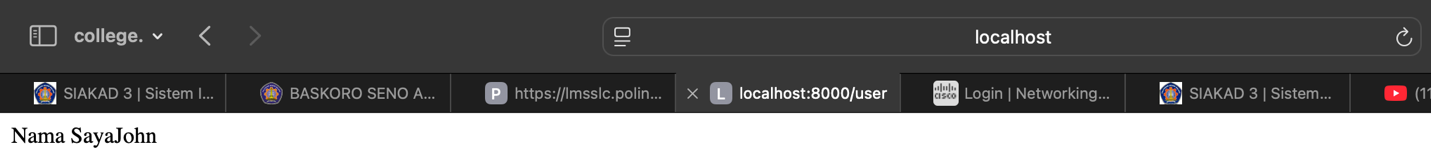
**Optional Parameters**

**Langkah-Langkah Praktikum 3 :**

1. This defines a GET route that can create an option for the output name between inserting the name or not in the url route.

* Code : 
* Without Inserting : 
* With Inserting : 

1. This defines a GET route that instantiate the value of the $name parameter into John so the $name parameter have the auto define value

**Route Name**

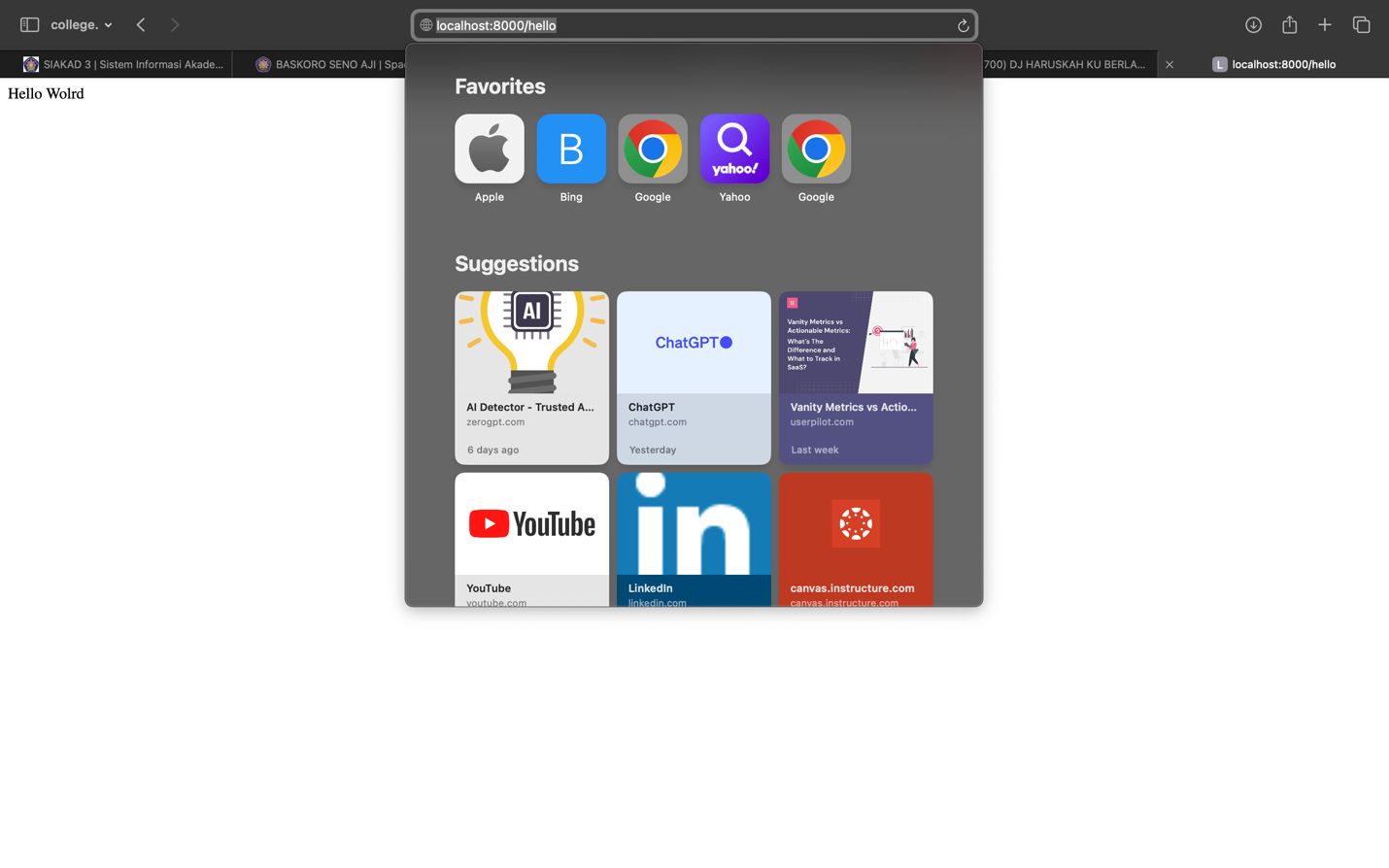
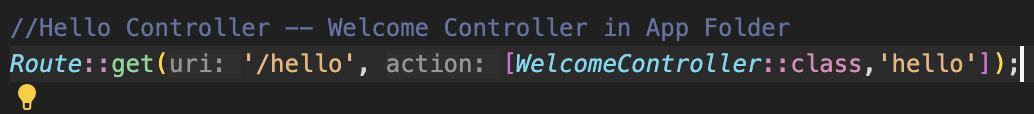
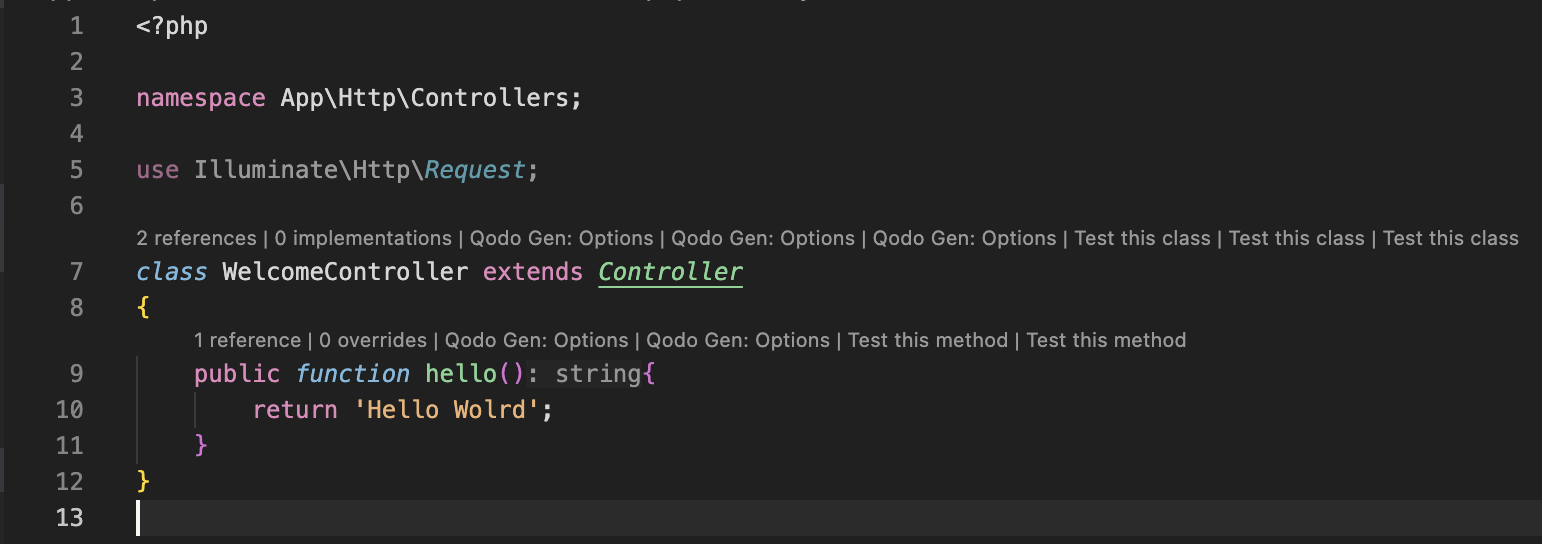
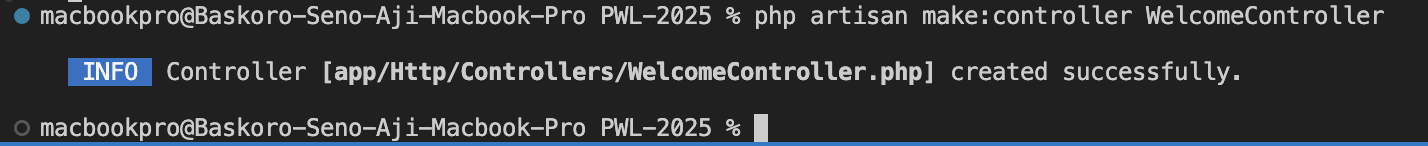
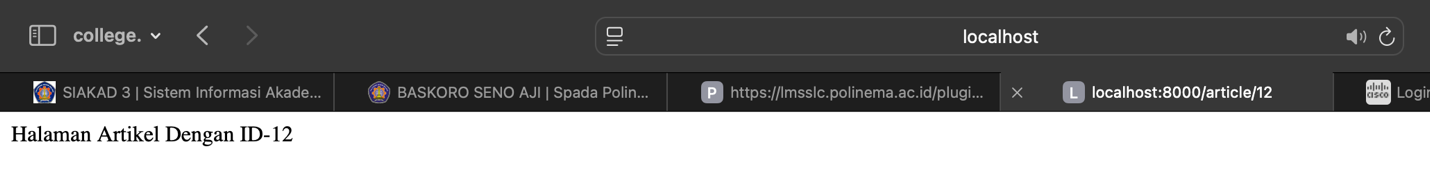
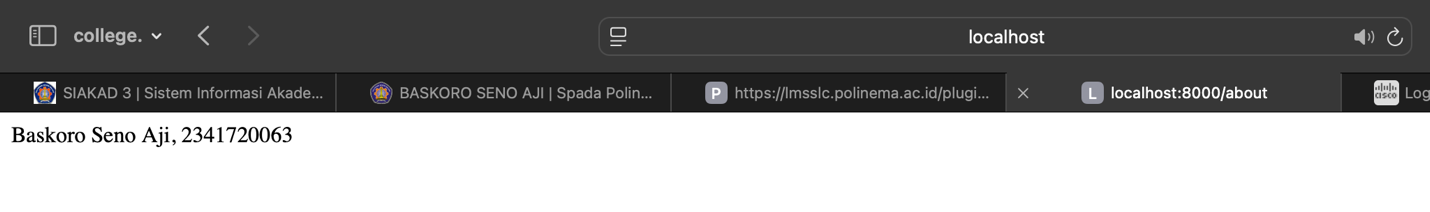
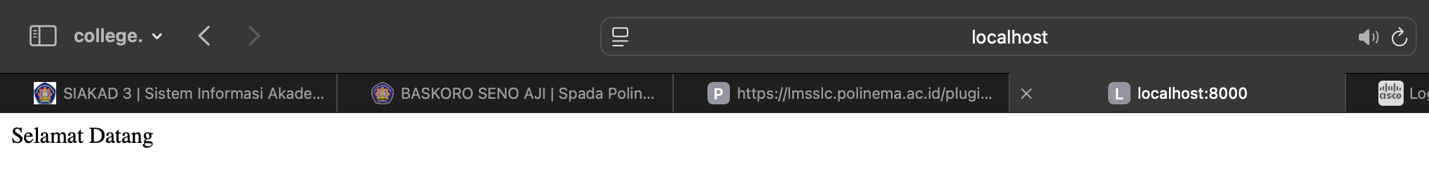
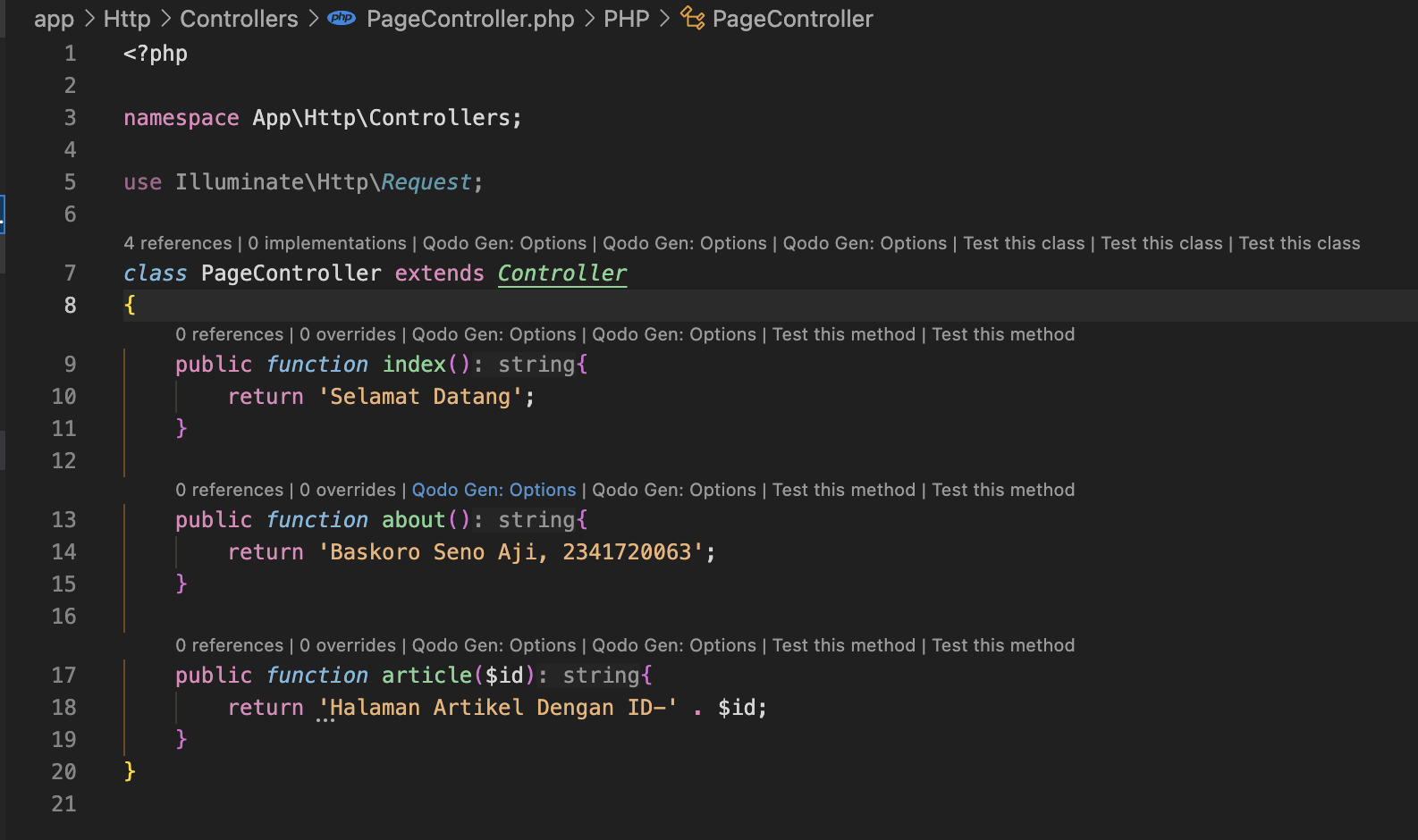
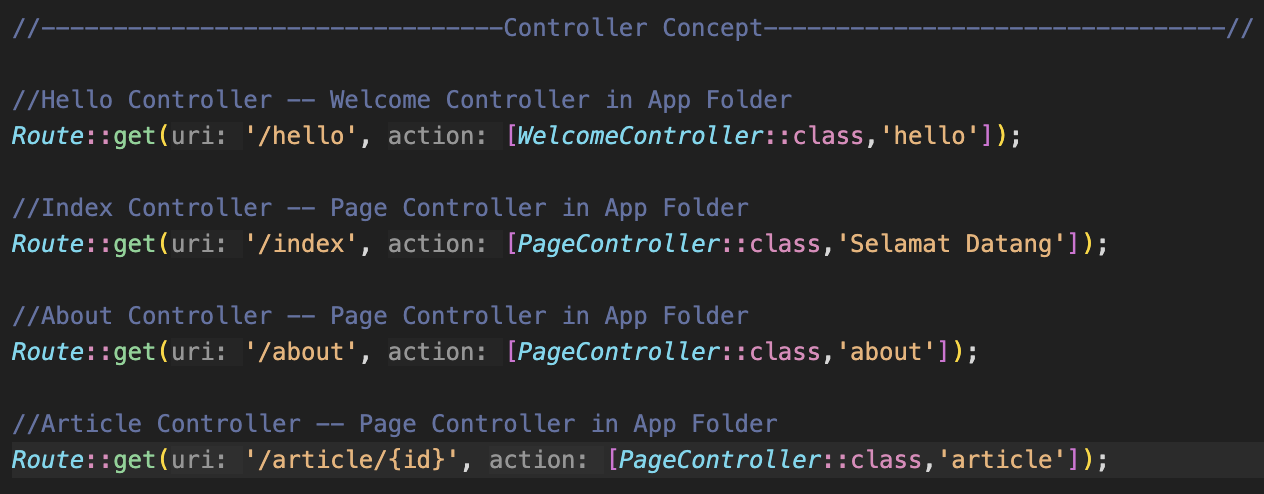
**Langkah – Langkah Praktikum 4 :**

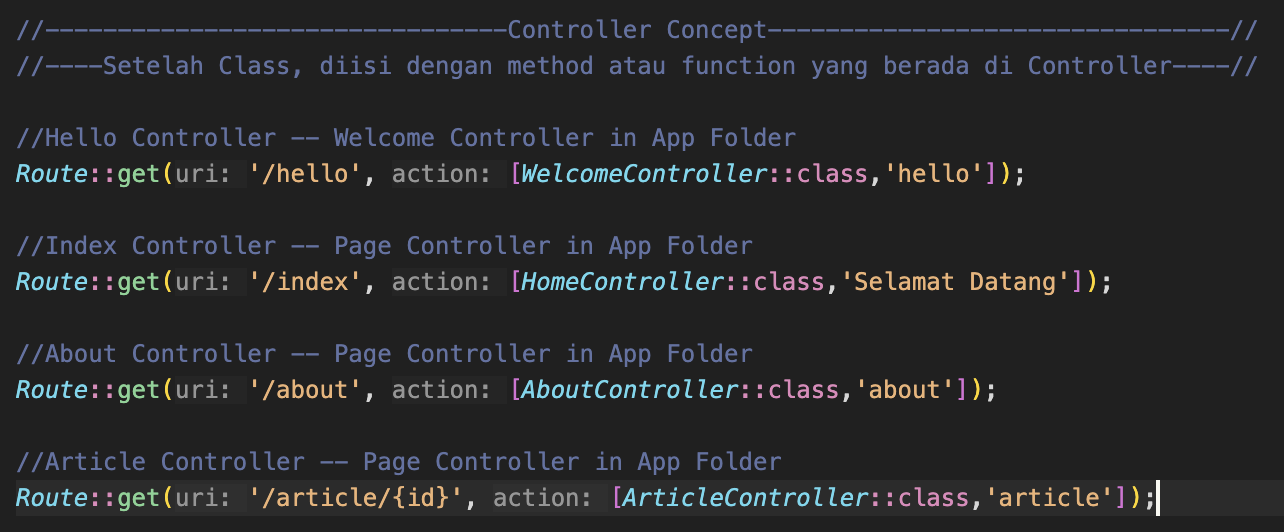
1. This route defines a URL /user/profile that executes an empty function when accessed. It is also given the name ‘profile’

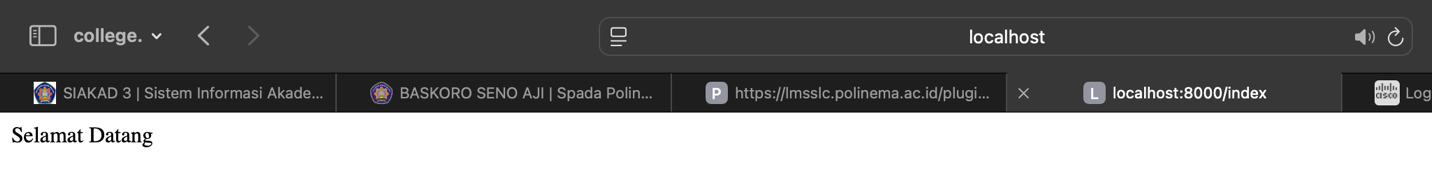
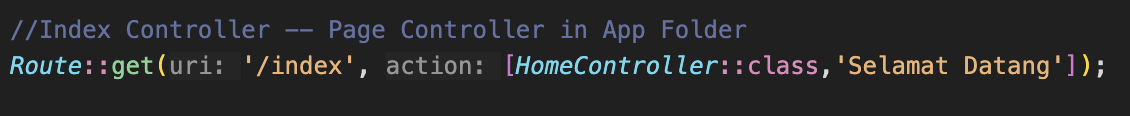
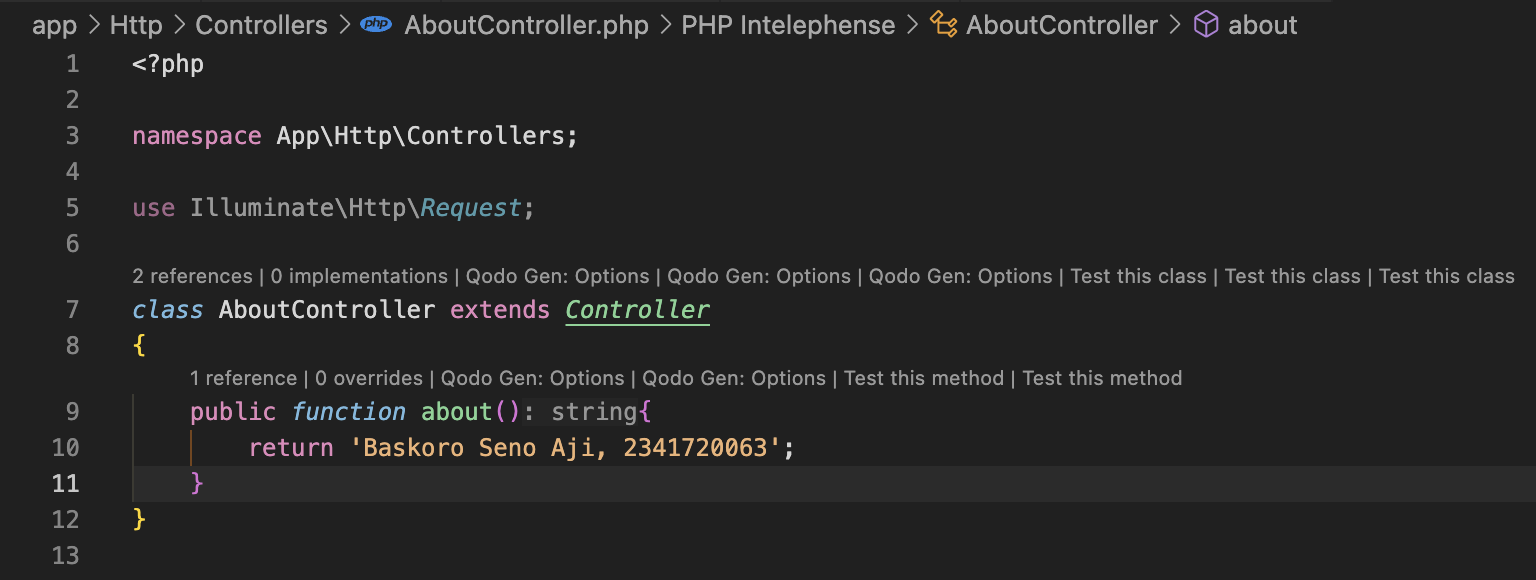
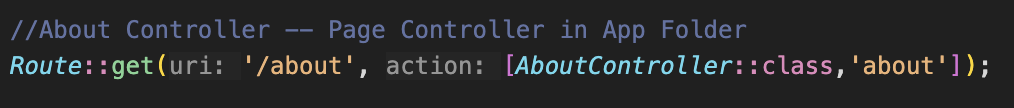
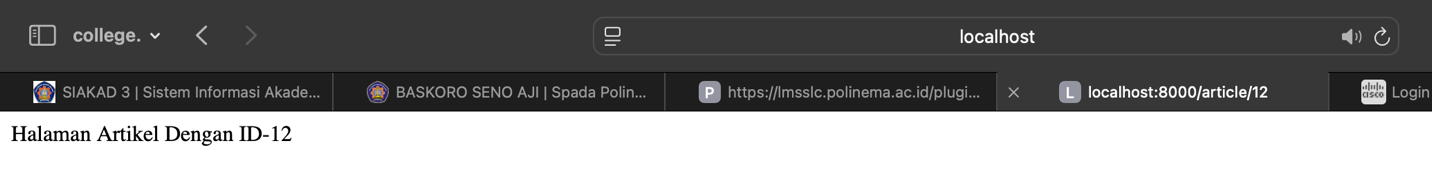
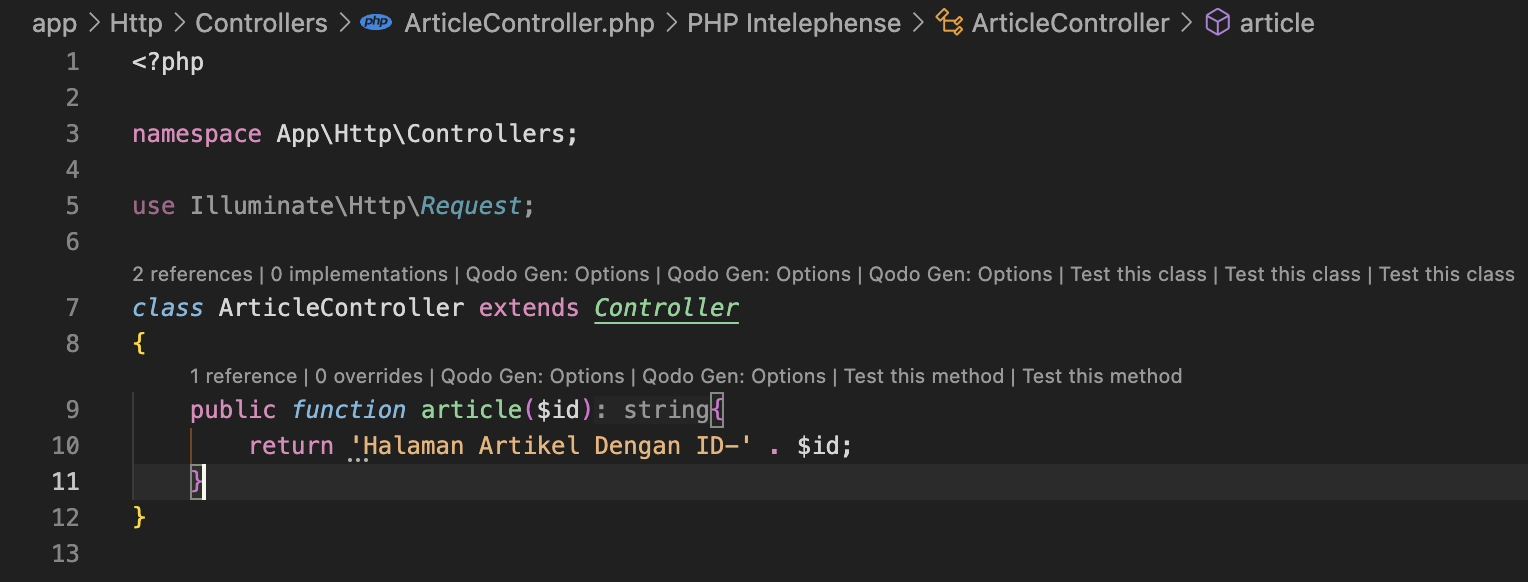
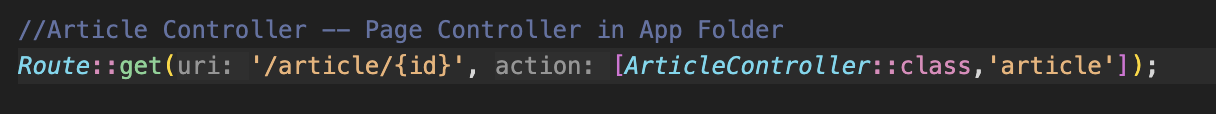


**Membuat Controller**

**Langkah – Langkah Praktikum 5 :**

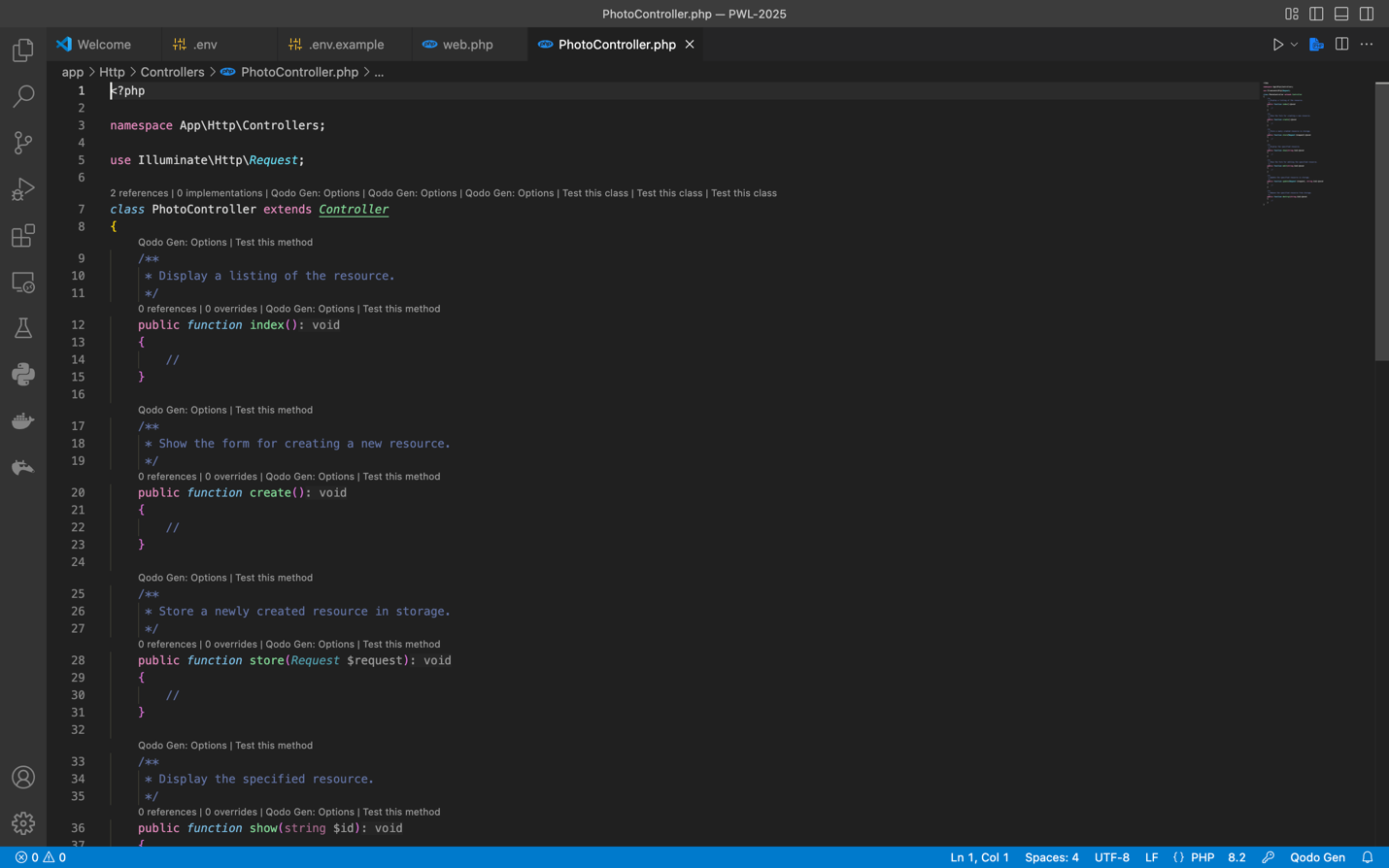
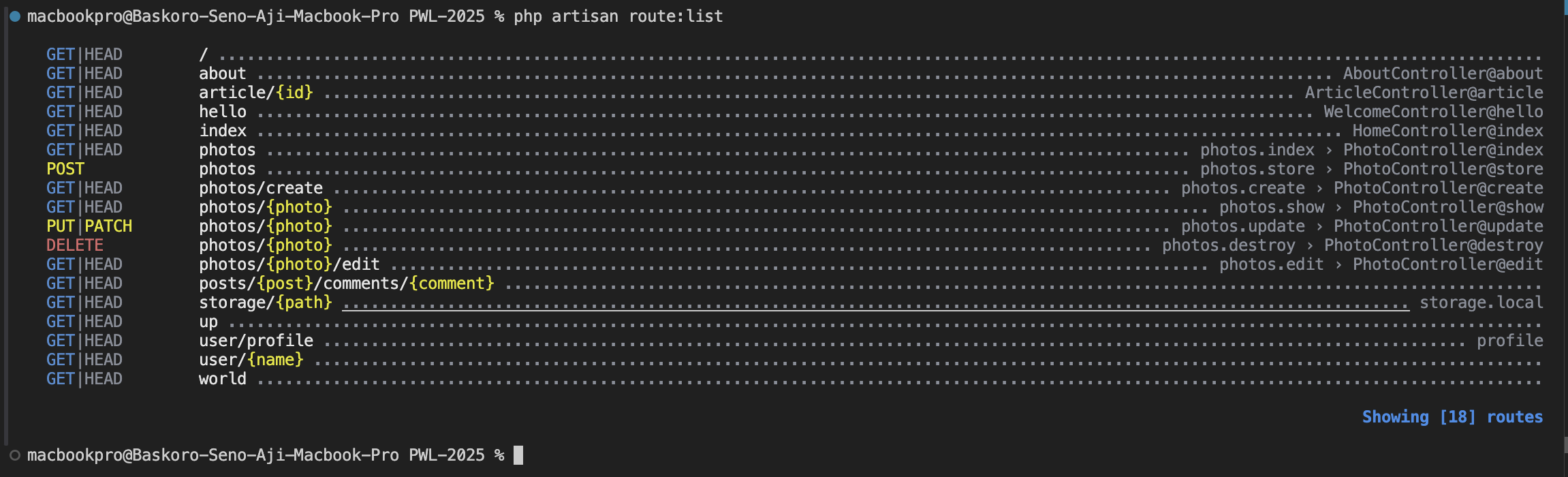
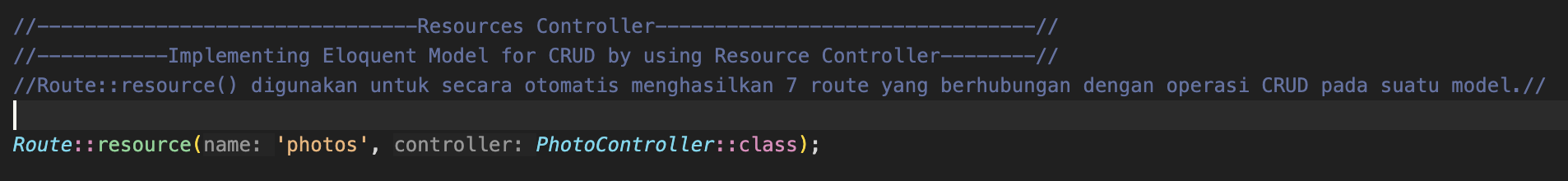
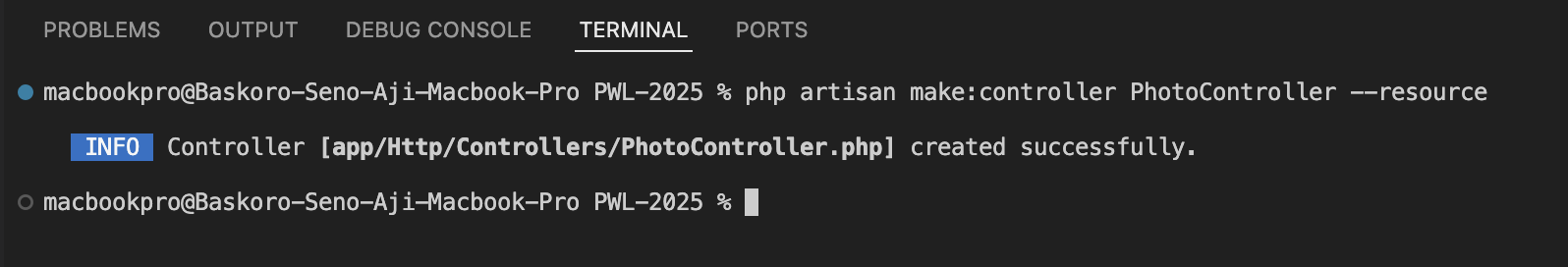
1. The code defines a WelcomeController class within the App\Http\Controllers. This route give responds for the GET requests at the /hello URL and triggers or call the hello() method of the WelcomeController, which responds by displaying the “Hello World” message
2. The code defines a PageController class within the App\Http\Controllers. This route give responds for the GET requests at the /index, /about and also the /article URL and triggers or call the function or method within the PageController, which responds by displaying every value inside the “……” message
3. This Section we’re modifying the code based on every Controller Section where we create it one by one where it still having the same function and consisting the same things, it’s only different on the route to connect it into the controller

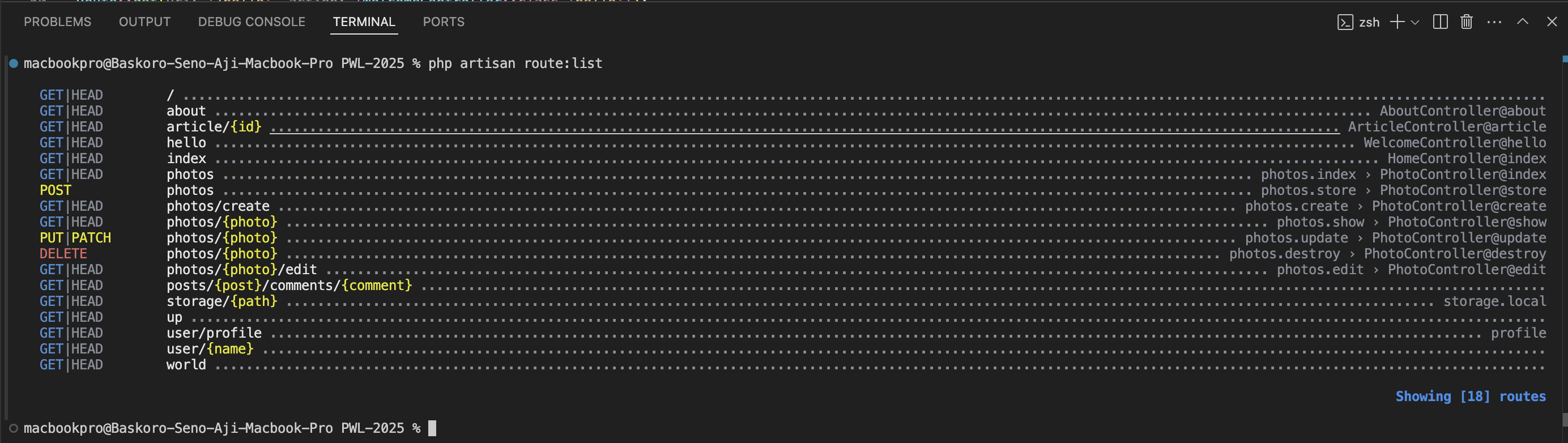
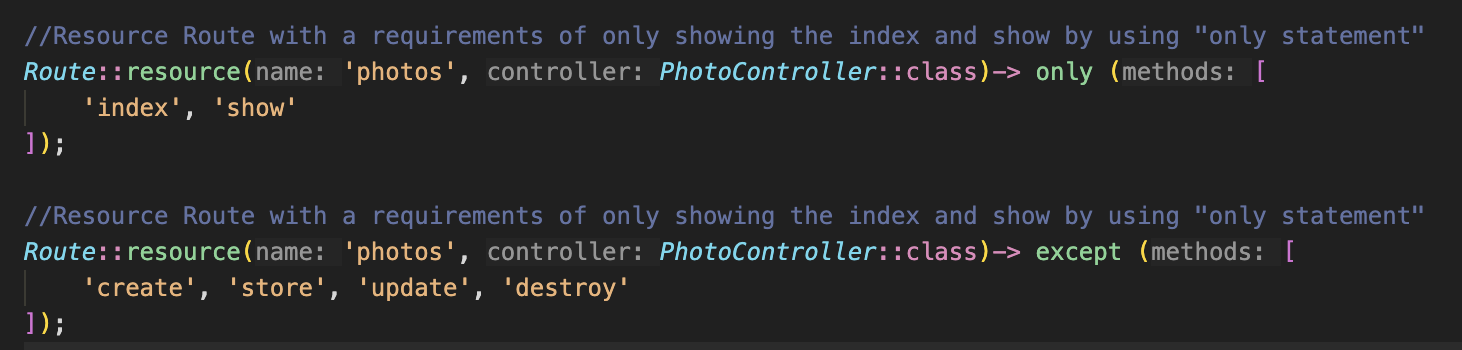


* index Route – HomeController : 
* about Route – AboutController : 
* article Route – ArticleController : 

**Resource Controller**

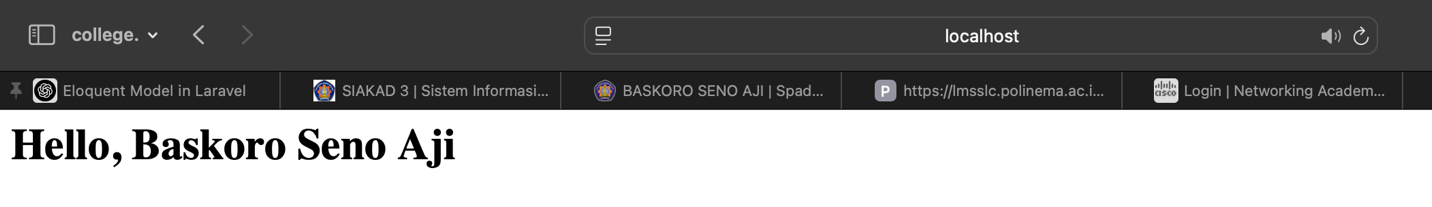
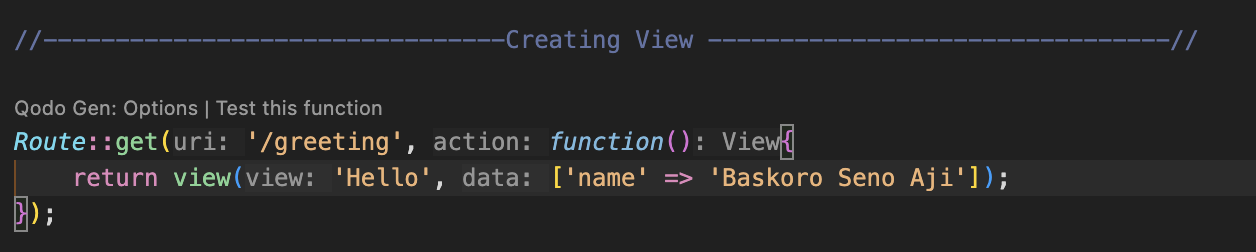
**Langkah-Langkah Praktikum 6 :**

1. The command **php artisan make:controller PhotoController** –resource generates a eloquent model in resource controller with methods for handling 7 CRUD operations.
2. This specific command of ONLY and EXCEPT are a reqirement for the resource model to shown the specific Operations



**Membuat View**

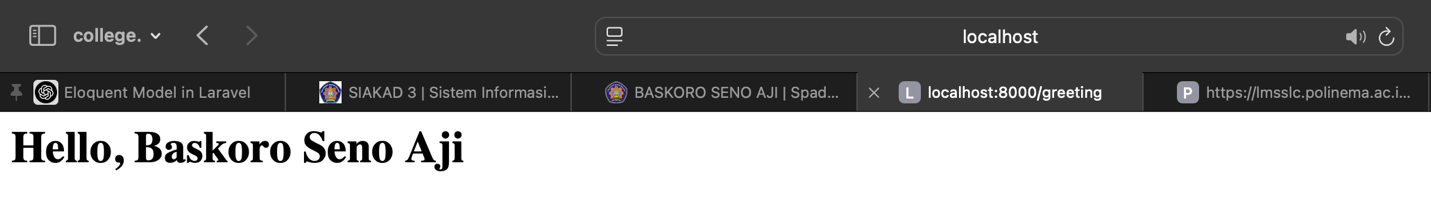
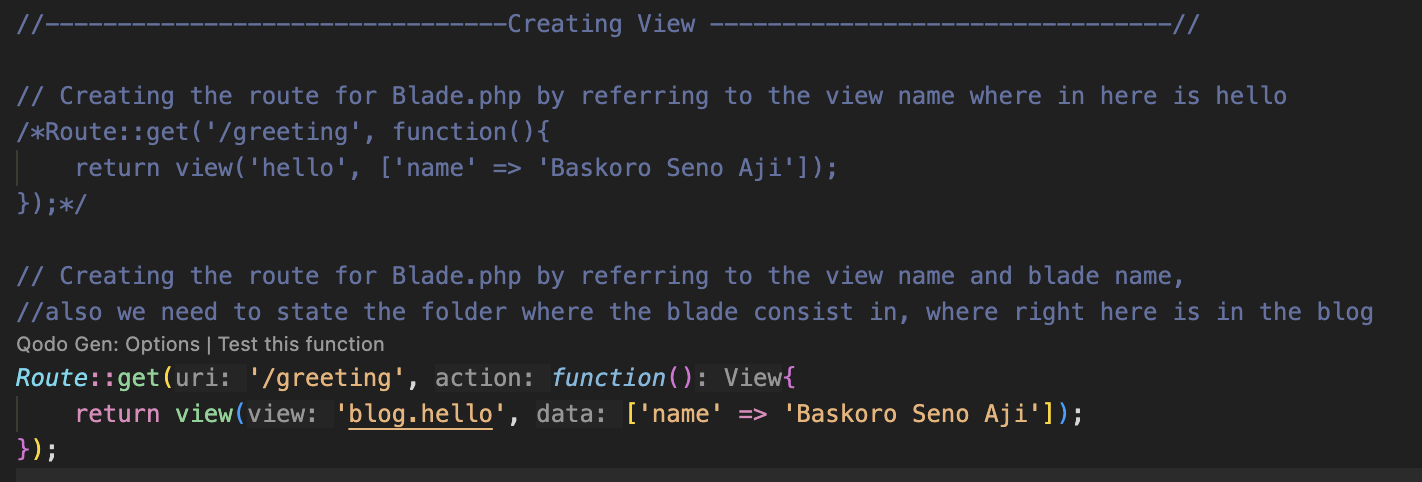
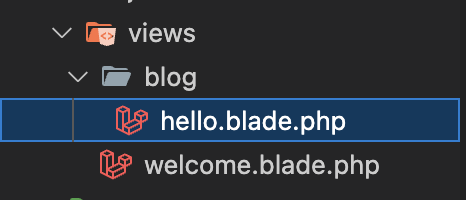
**Langkah-Langkah Praktikum 7 :**

1. In this section, hello.blade.php have responsibility to show interface and it will running with route where route will call view in accordance with file name without blade.php in the statement in that case view have function to show interface to the user. This (view file.blade,php) will search the view name and show the output based on the blade.php file structure. 

**View Dalam Direktori**

**Langkah-Langkah Praktikum 8 :**

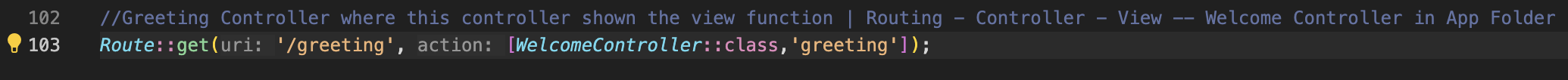
1. Same as before but in this section we’re inserting the hello.blade.php inside the blog folder so in this case we need to state the folder name then the file whereas ‘blog.hello’ in the route



**Menampilkan View Dengan Controller**

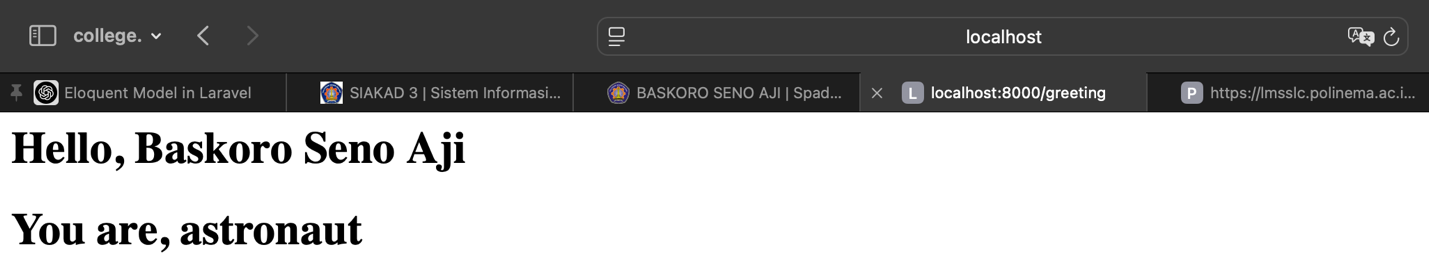
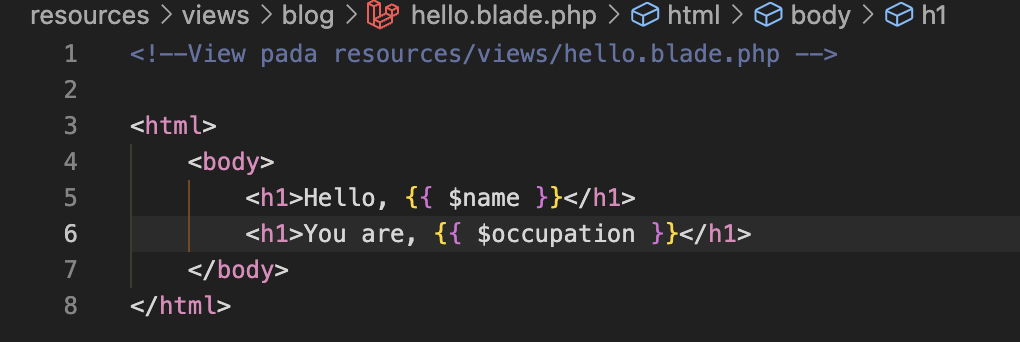
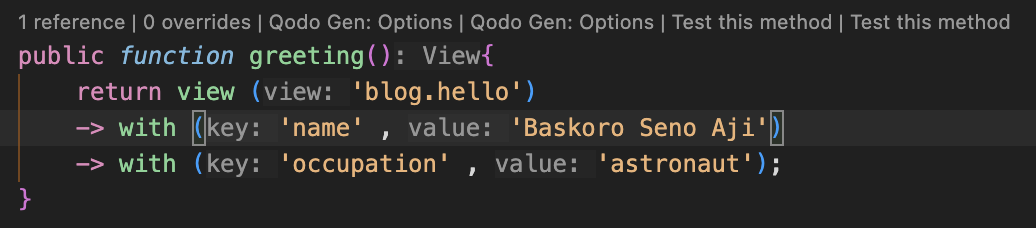
**Langkah-Langkah Praktikum 9 :**

1. In this section we’re trying to call the greeting function with the controller, where the routing will call the controller and the controller will give an output of view whereas greeting in this section.

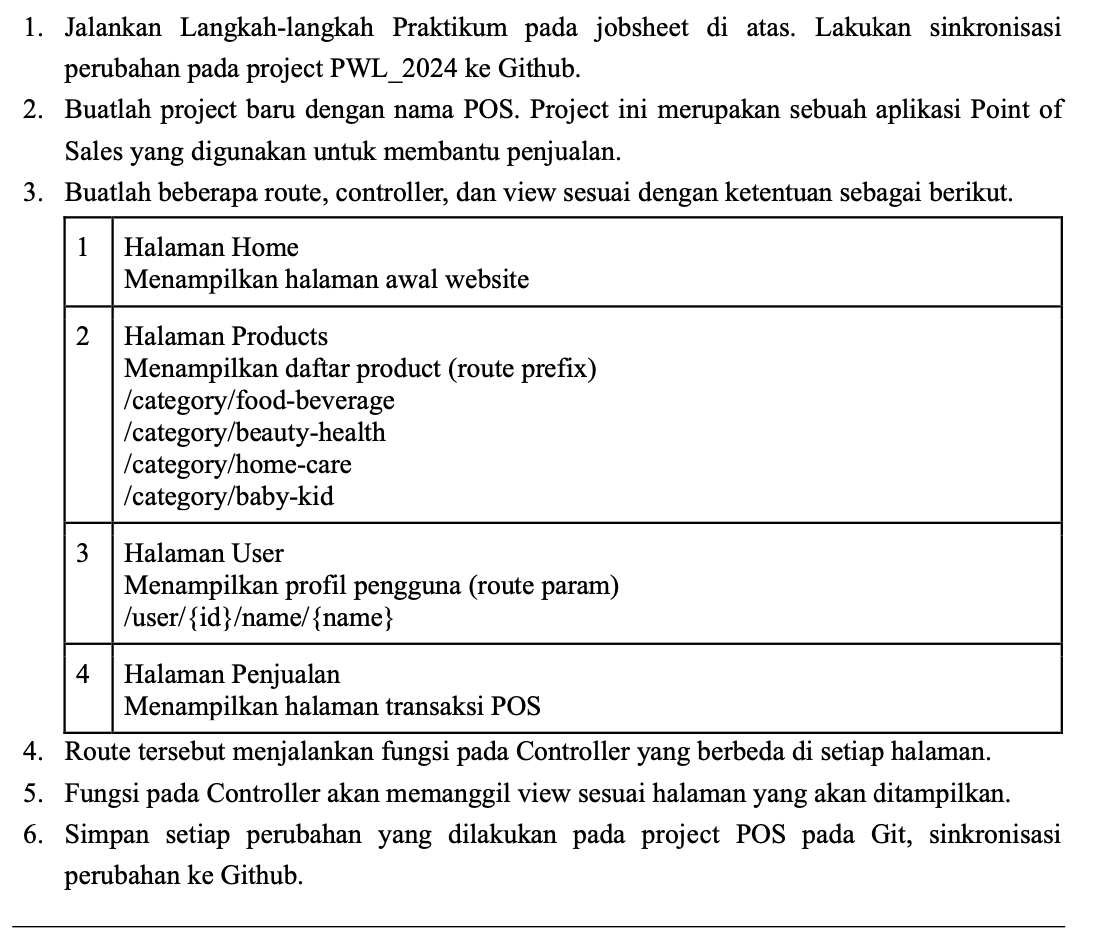


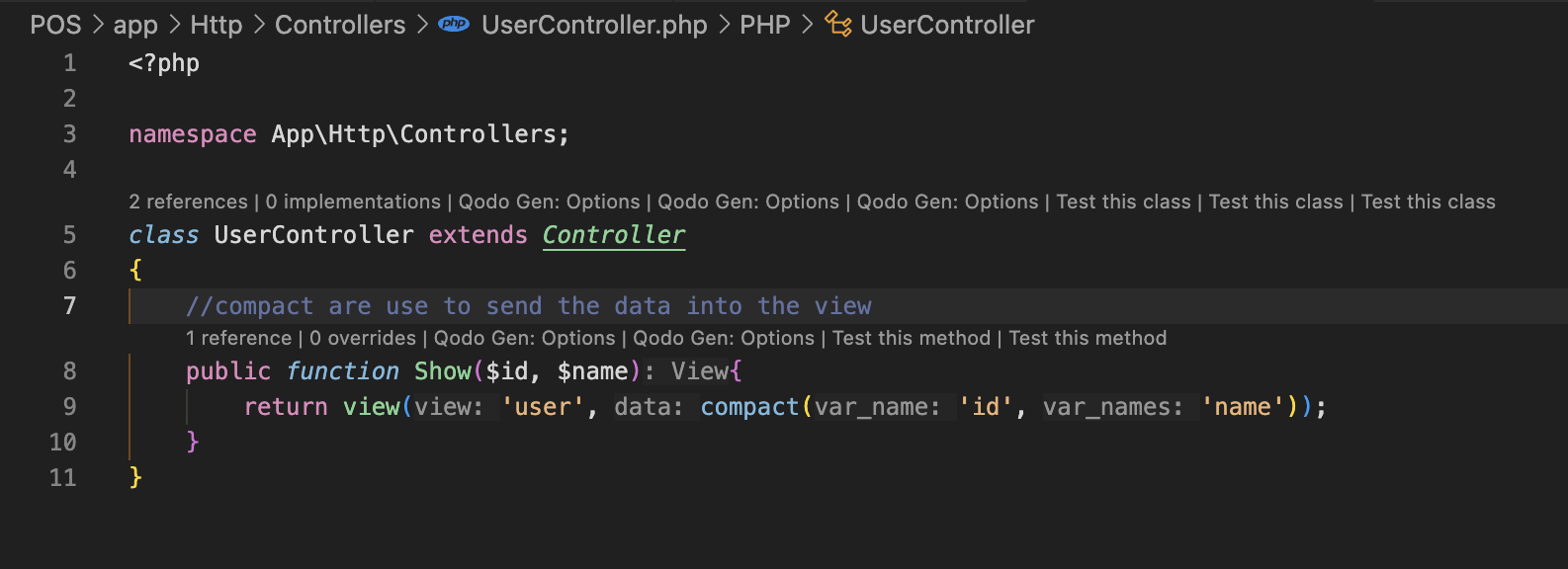
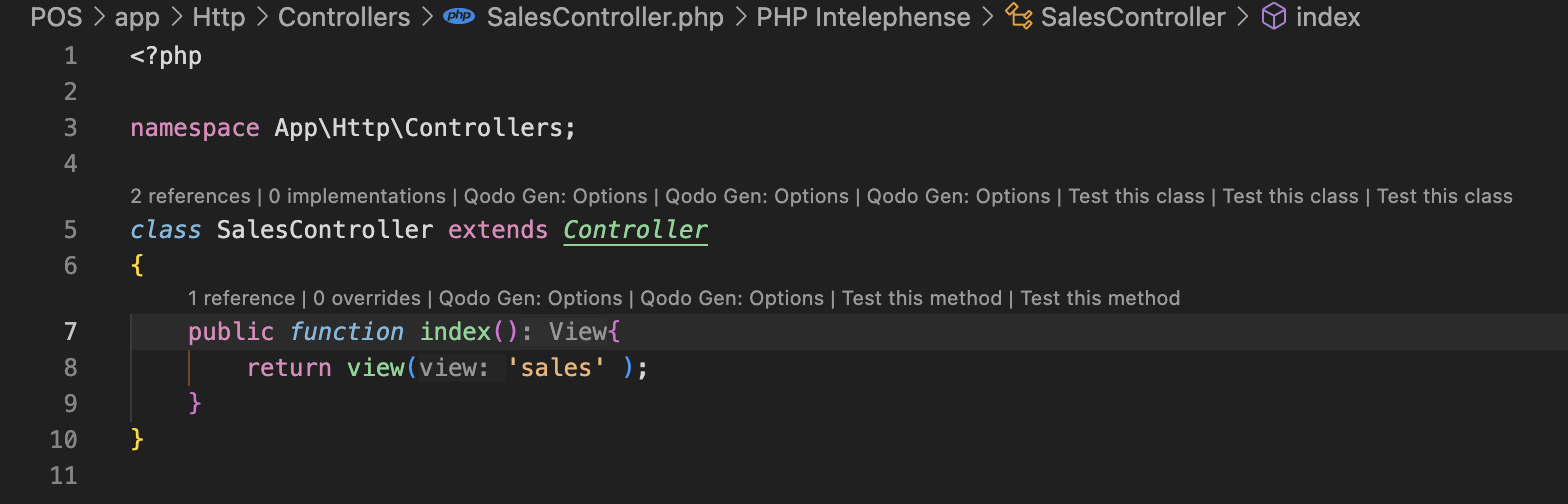
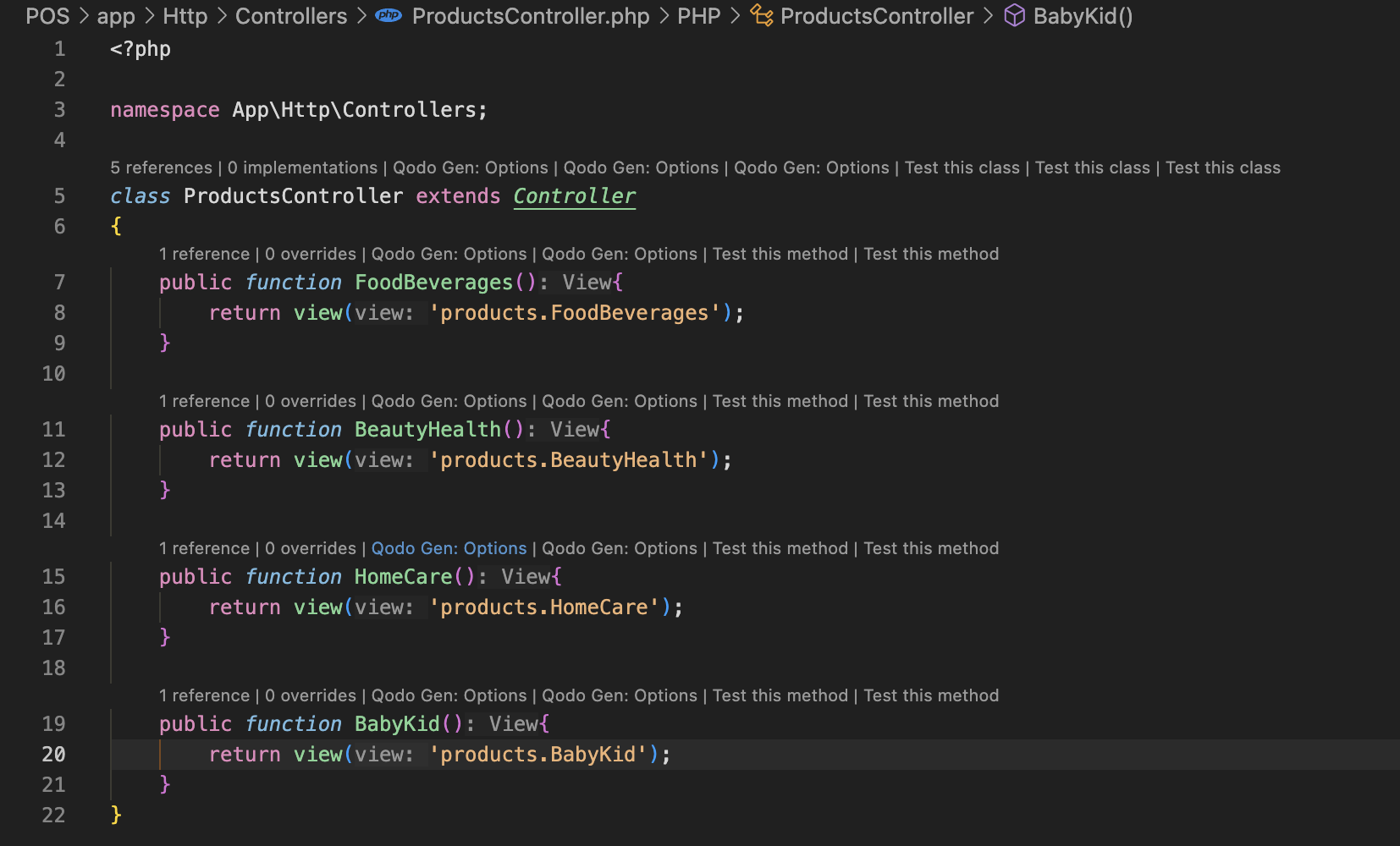
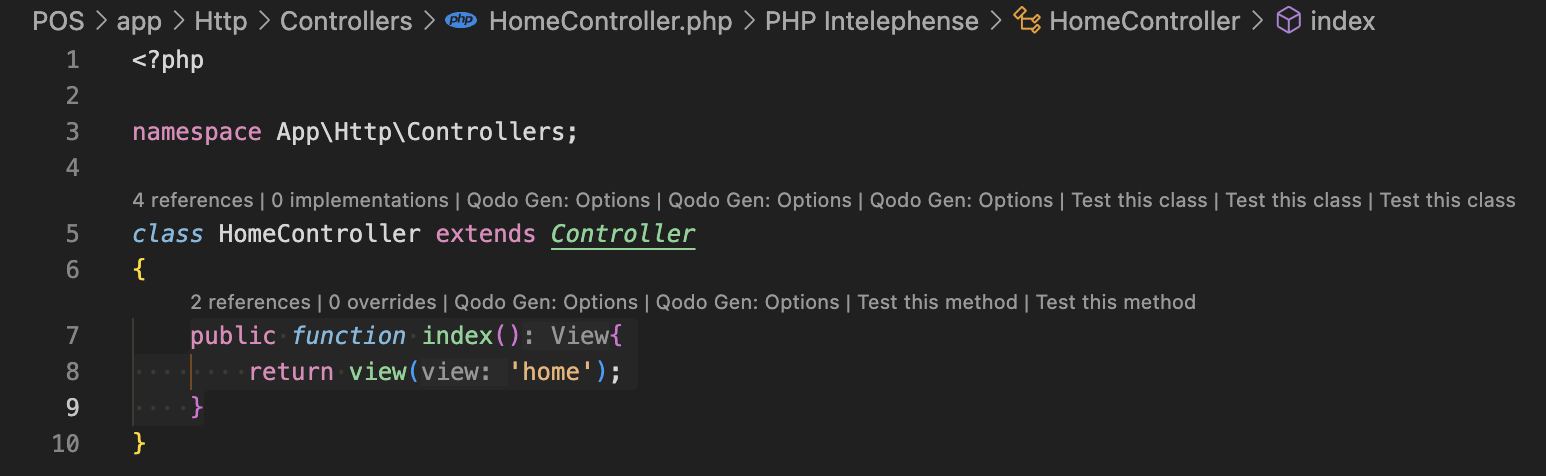
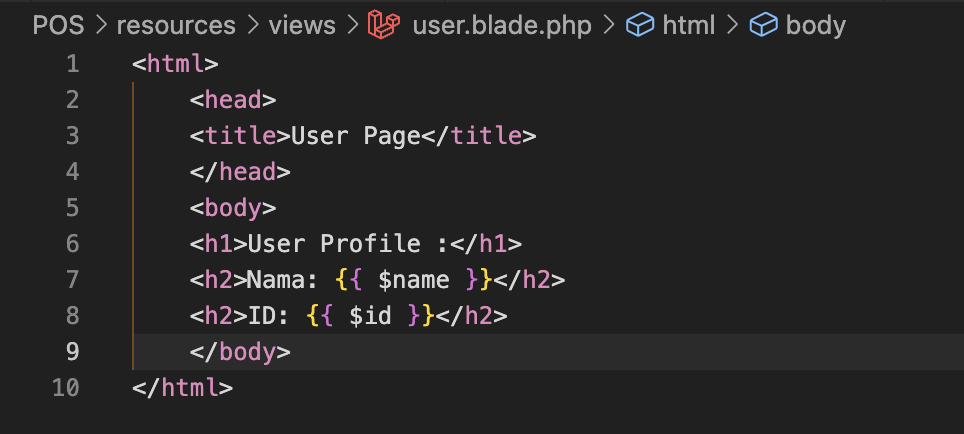
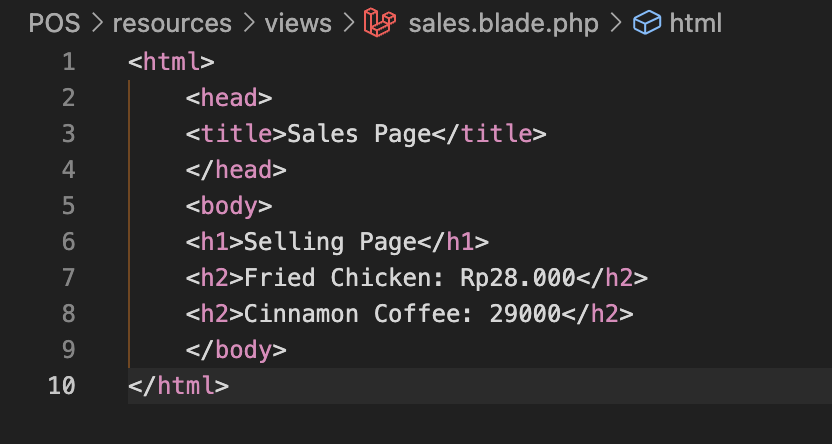
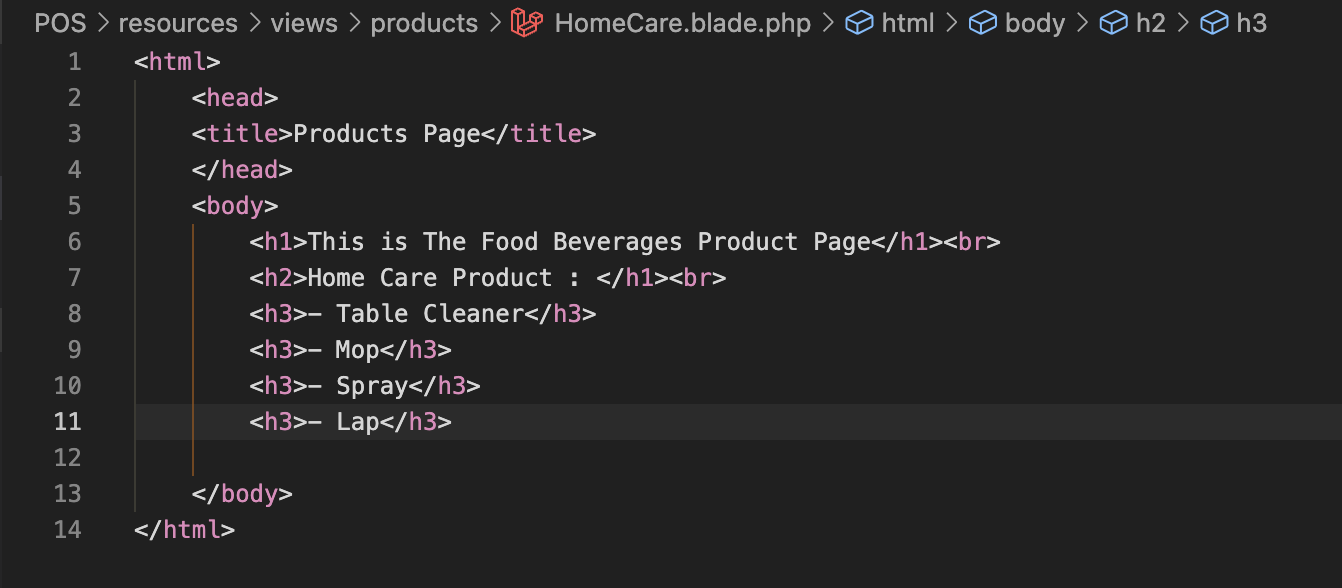
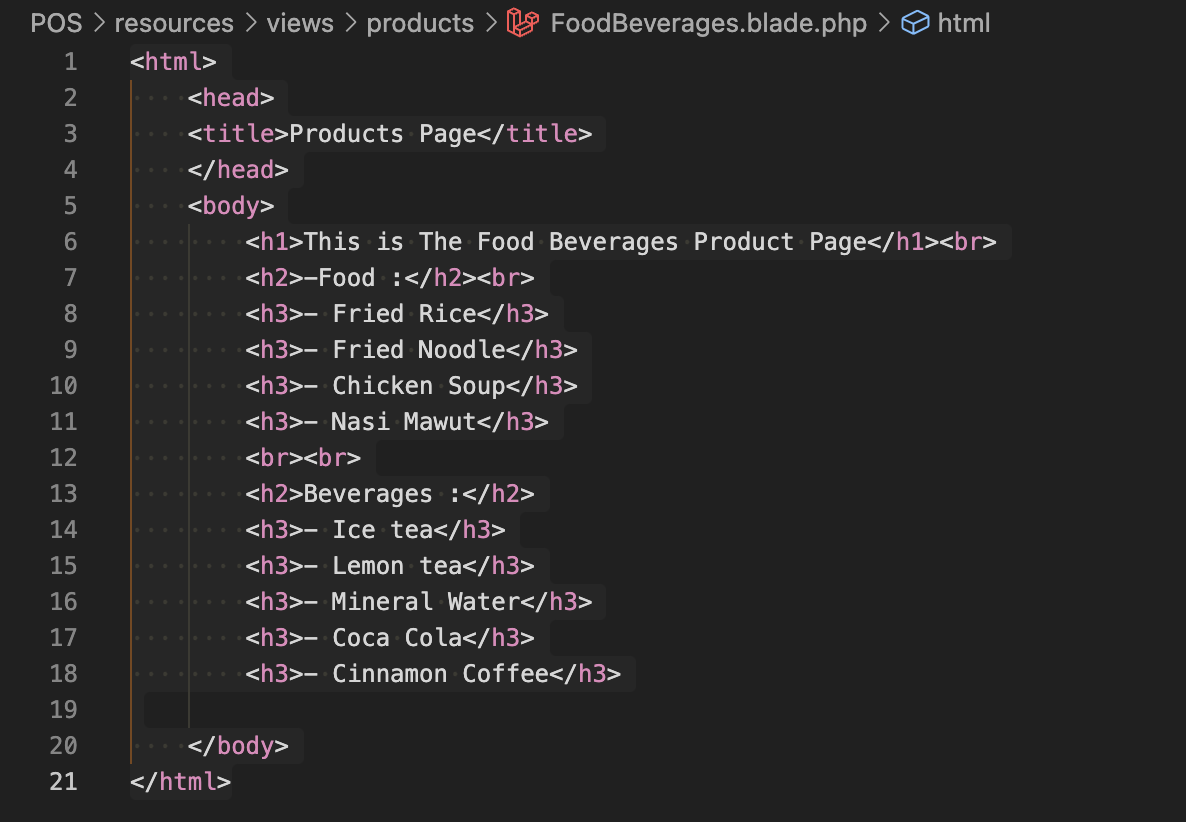
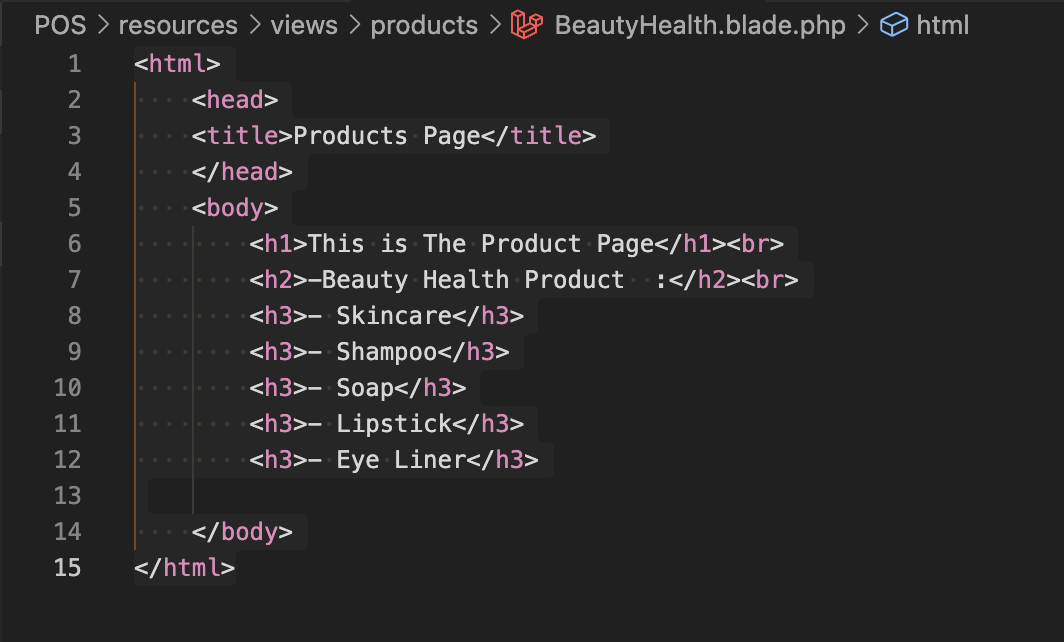
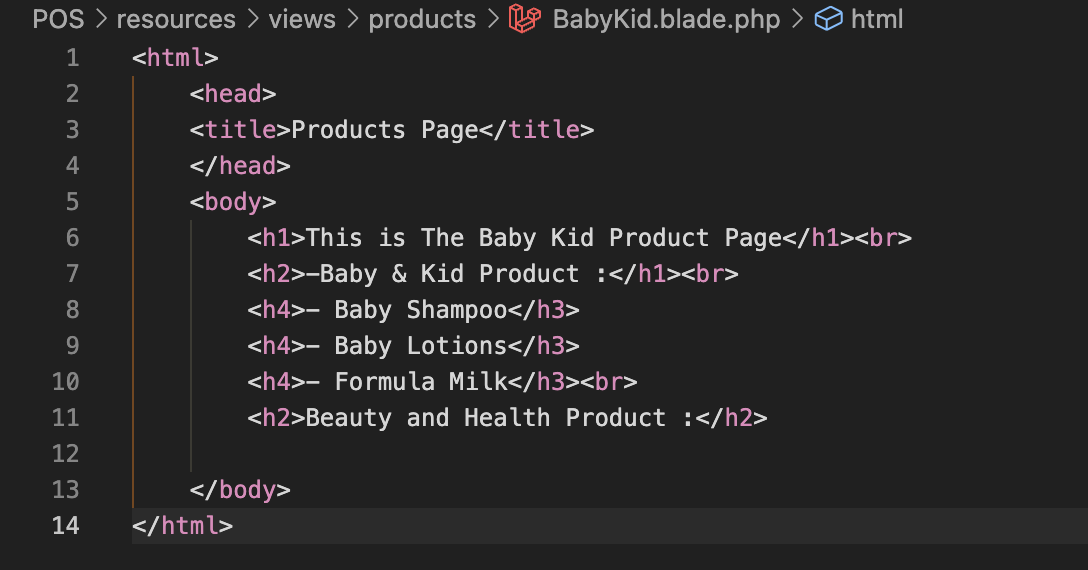
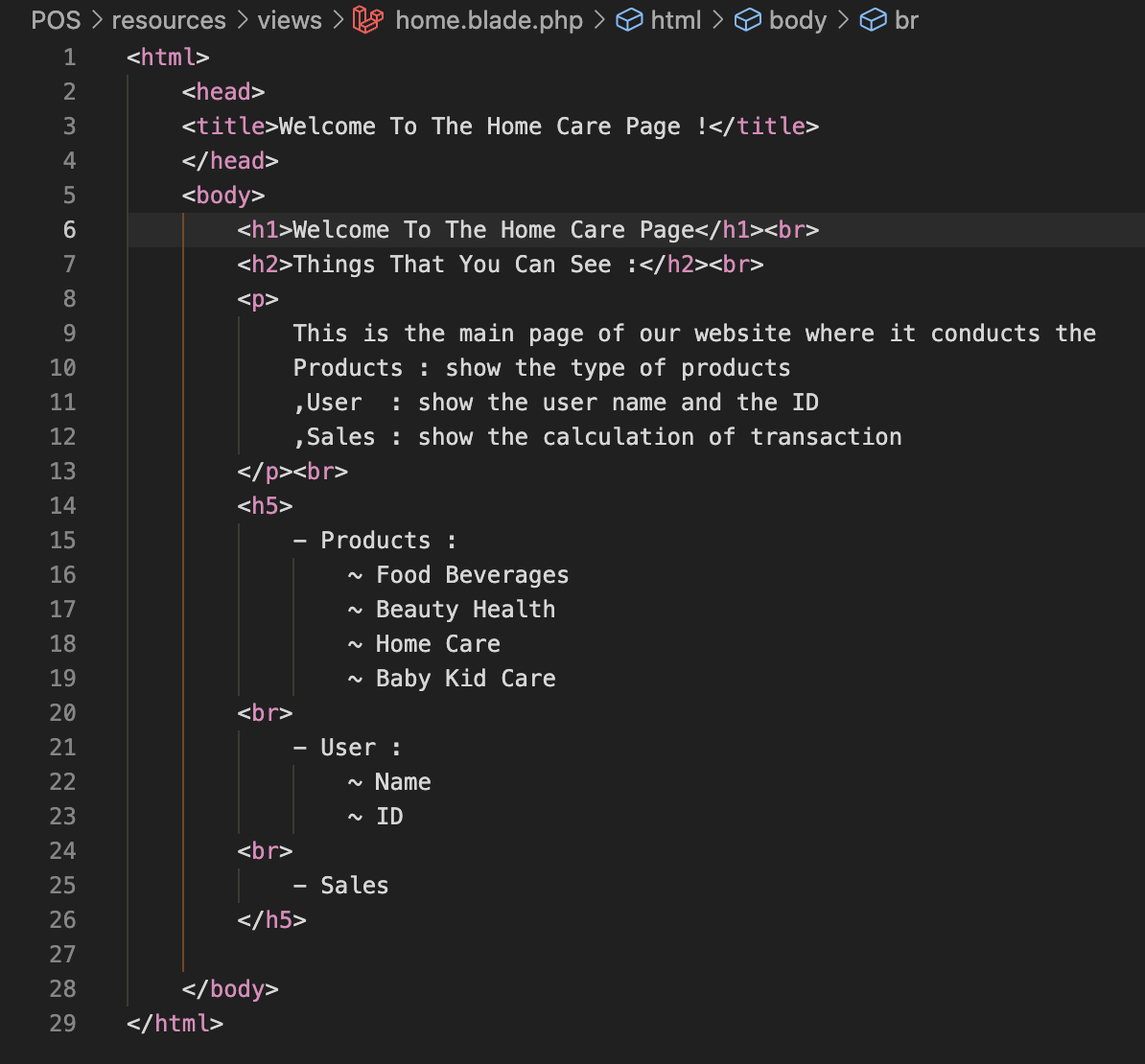
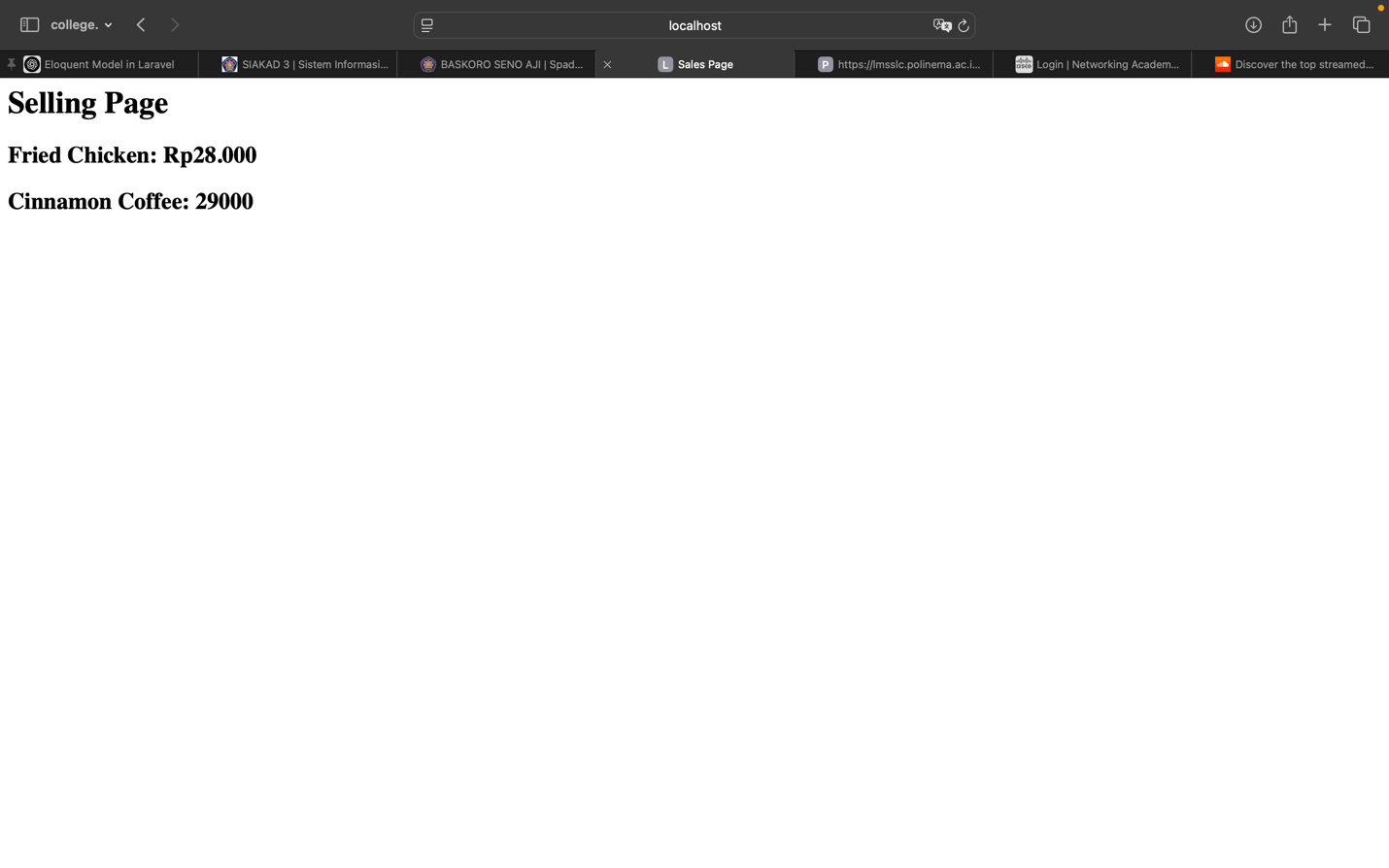
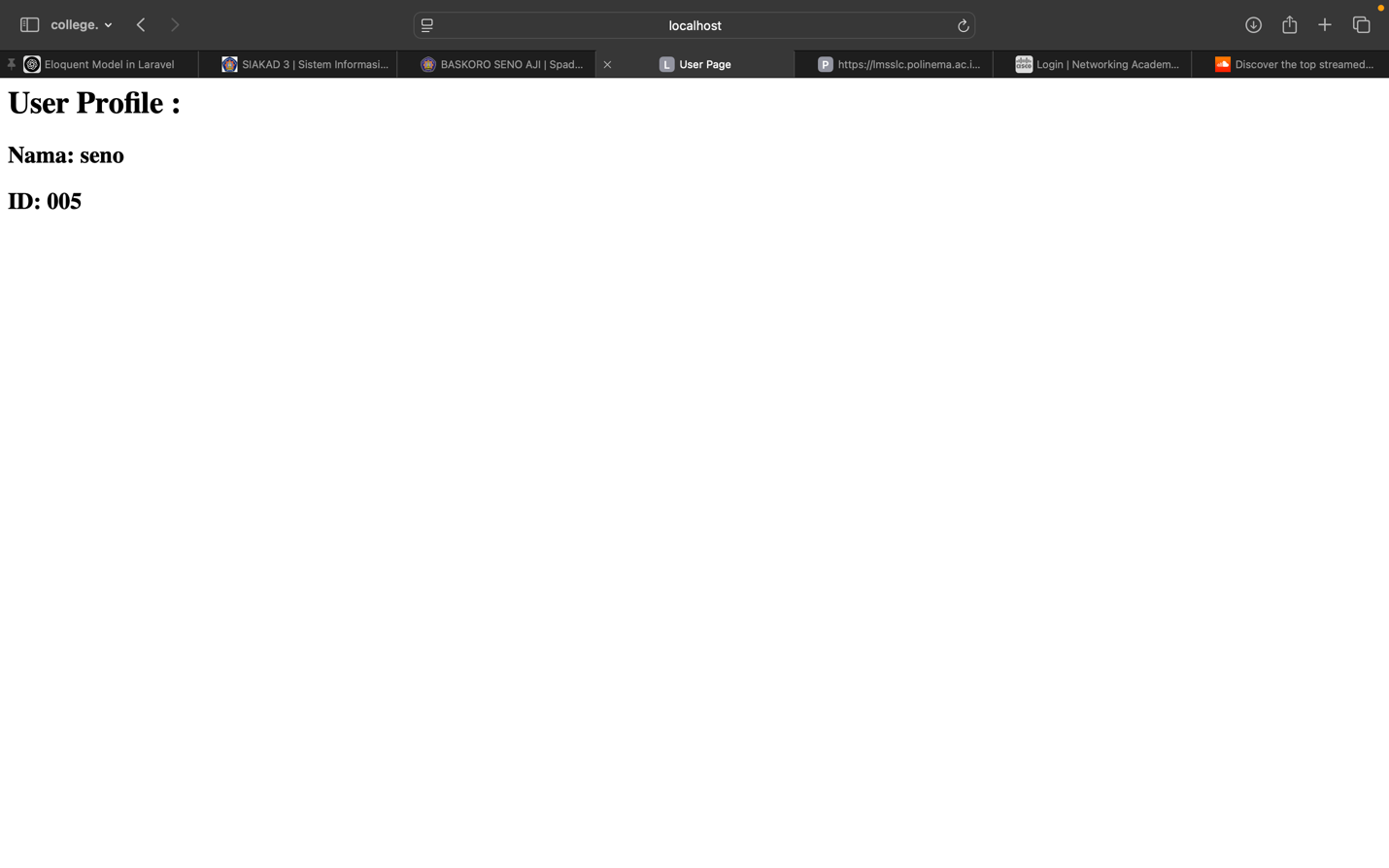
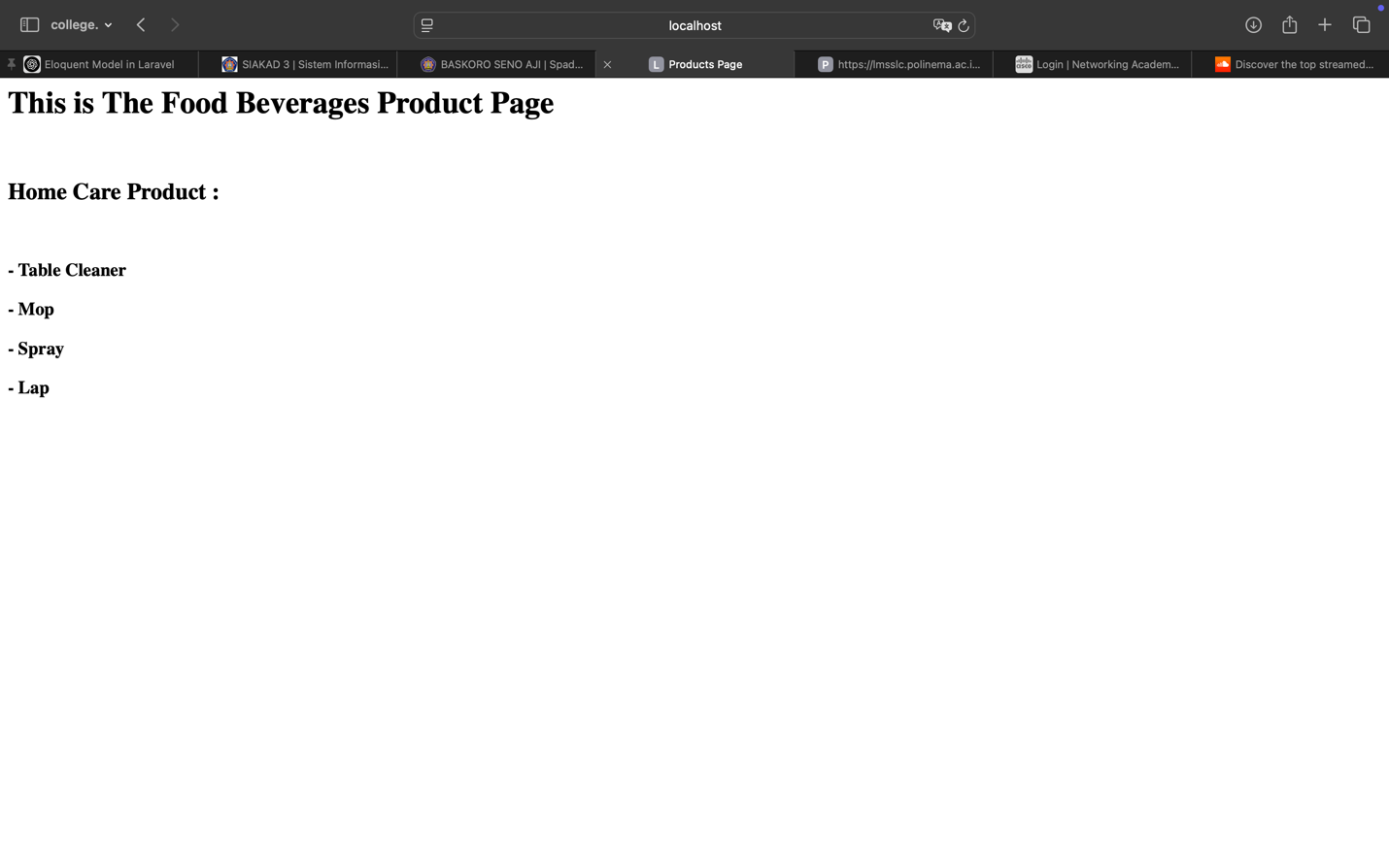
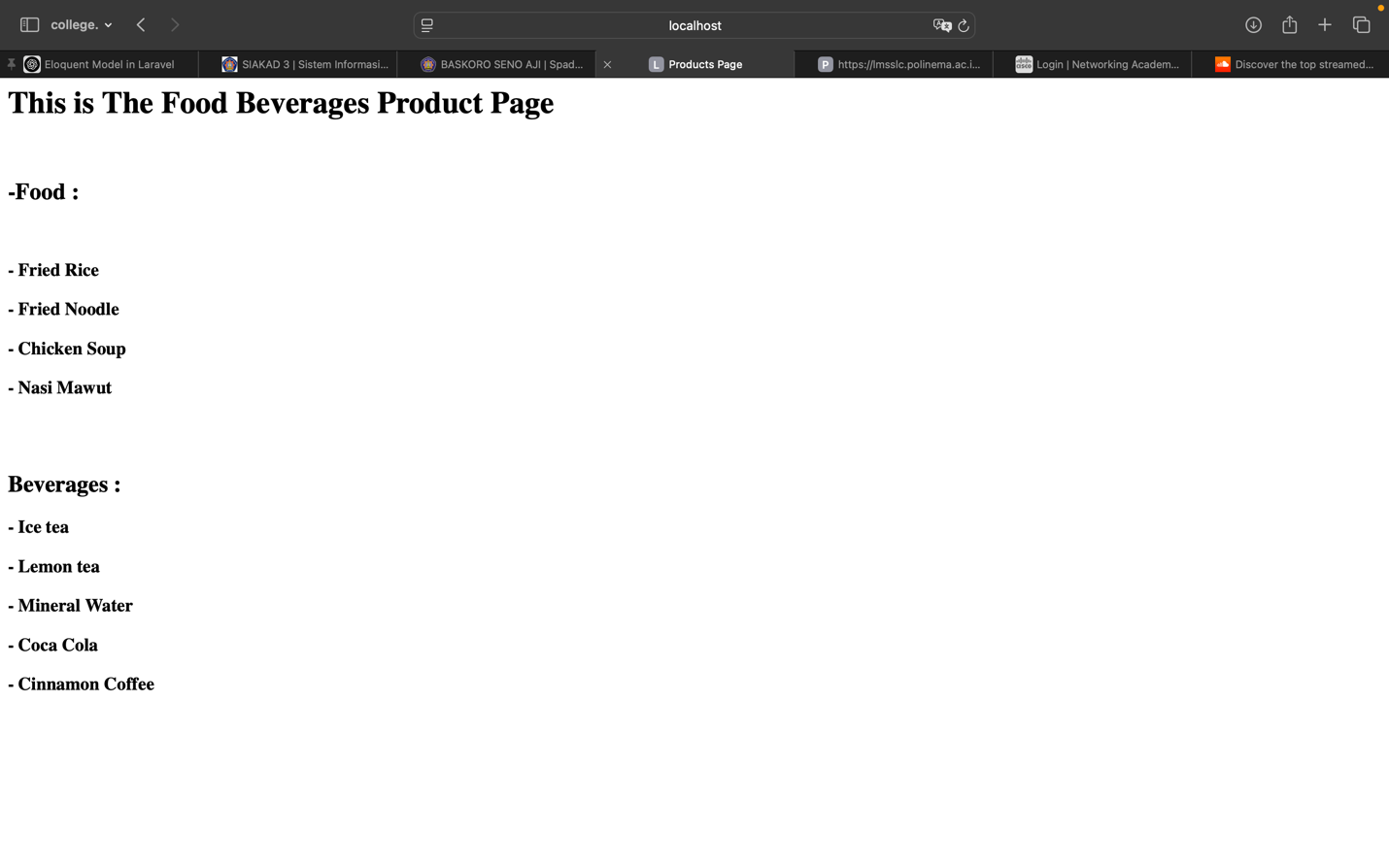
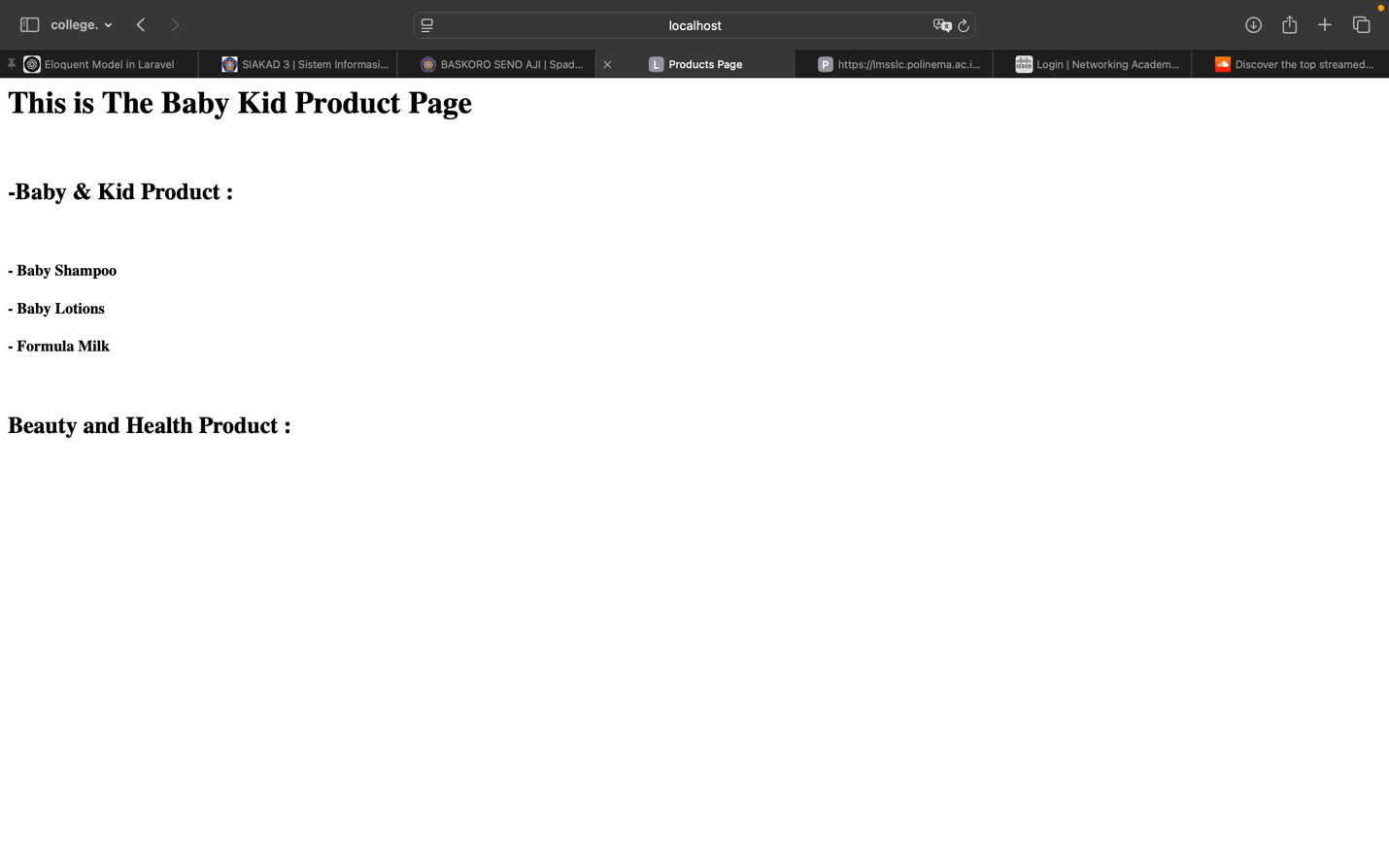
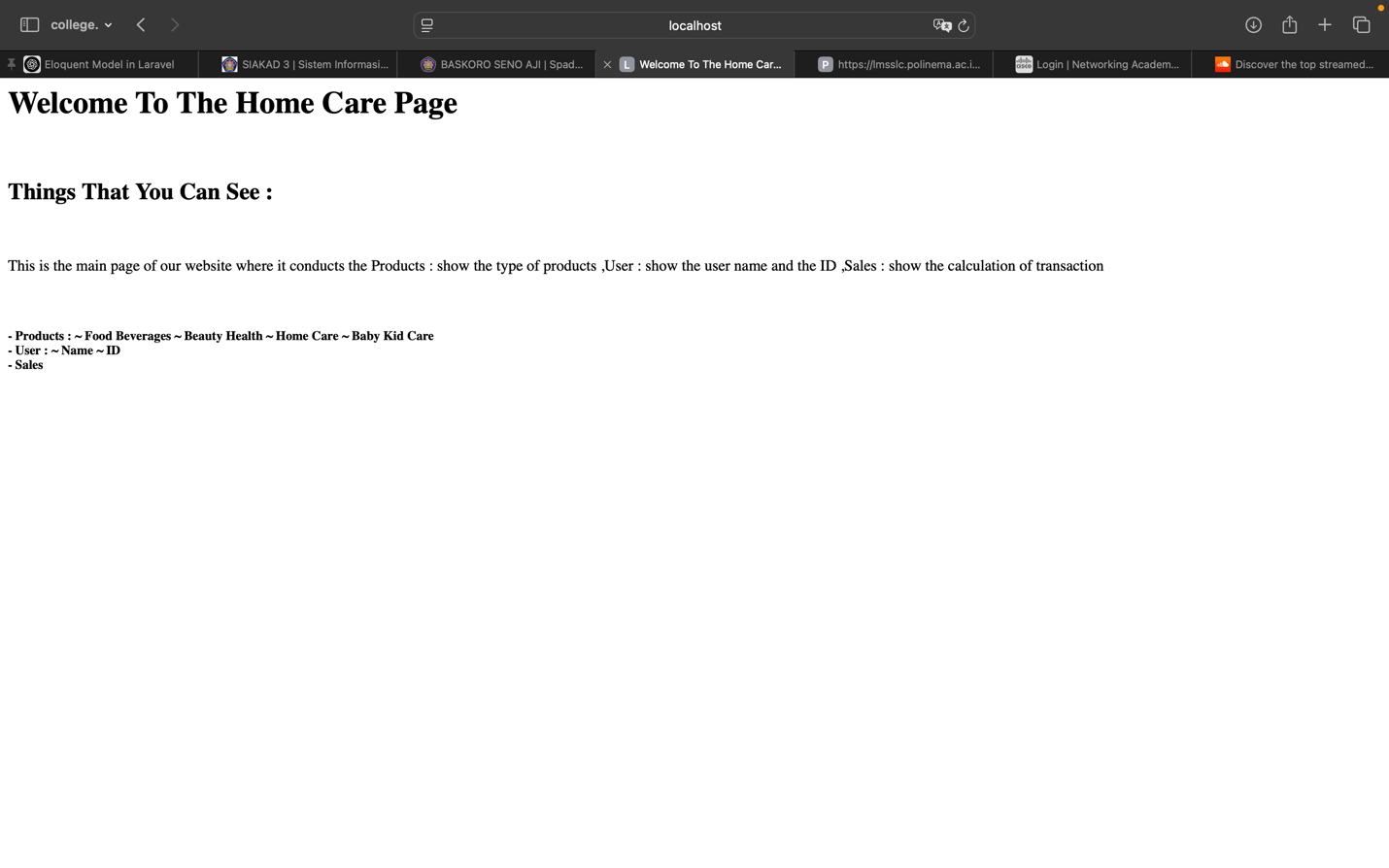
**Meneruskan Data Ke View**

**Langkah-Langkah Praktikum 10 :**

1. In the WelcomeController, the greeting() method uses the with() function to send two variables, name and occupation, to the blog.hello view. The with() function is used to add data to the view consecutively.

**Soal Praktikum**

****

* **Main Route :**
* **Controller : **
* **Views : **
* **Output : **