**HOME PAGE**

|  |
| --- |
| Index.php |

Because the controller and action are not set then by default they are going to be

controller=page

action=page

|  |
| --- |
| Views  layout.php |

This is where bootstramp and the design is..

Routes.php is included in this file

|  |
| --- |
| routes.php |

1. Includes the controllers/page\_controller.php file
2. In the file that is included there’s a class called PagesController
3. A new object is created called page.. So page= new PagesController
4. In that class we are going to call a method.. page->home
5. Home is a methods name

|  |
| --- |
| Controllers  page\_controller.php |

1. The home function has a include file which is views/pages/home.php

|  |
| --- |
| Views  Pages  home.php |

Home.php has all the text that my home page should have.

**READALL**

|  |
| --- |
| Index.php |

Because there’s no values set yet then the home page is going to show.

|  |
| --- |
| Views  layout.php |

By clicking on the product you are assigning two get values

controller=product

action=readall

|  |
| --- |
| routes.php |

1. Calling a function called “call” which has controller and action as arguments
2. include controllers/product\_controller.php
3. Include models/product.php
4. controllers/product\_controller.php has a class called productcontroller which will be instantiated so now product = new productcontroller
5. The class above will call a function called readall like this product->readall

|  |
| --- |
| Controllers  Product\_controller.php |

As mentioned above.. This file has a method called readall().. Inside this method there’s another method called from models/product.php called all()

|  |
| --- |
| Models  product.php |

Function all() is called from the controllers.

This is where the connection is made and the query is made

|  |
| --- |
| Views  Products  readAll.php |

This is included from the Controllers/Product\_controller.php

This is where we could see the list of the products

**Add Product**

|  |
| --- |
| Index.php |

Because there’s no values set yet then the home page is going to show.

|  |
| --- |
| Views  layout.php |

By clicking on add product you are assigning two get values

controller=product

action=create

|  |
| --- |
| routes.php |

1. Calling a function called “call” which have controller and action as arguments
2. include controllers/product\_controller.php
3. Include models/product.php
4. controllers/product\_controller.php has a class called productcontroller which will be instantiated so now product = new productcontroller
5. The class above will call a function called create like this product->create

|  |
| --- |
| Controllers  Product\_controller.php |

As mentioned above.. This file has a method called create().. There’s an if statement.. In our case we have a GET not a POST so include views/products/create.php

|  |
| --- |
| views/products  create.php |

Here we have a form where we can type the info of our new product then (ADD PRODUCT)

When we click the button a POST request has occurred.

|  |
| --- |
| Controllers  Product\_controller.php |

In the create function we have a POST now so the else will run ..which will call the all function from models and then include the 'views/products/readAll.php’

|  |
| --- |
| Models  product.php |

Function add() is called from the controllers.

This is where the connection is made and the query is made.

if(isset($\_POST['name'])&& $\_POST['name']!="") DON’T GET

|  |
| --- |
| Views  Products  readAll.php |

This is included from the Controllers/Product\_controller.php

This is where we could see the list of the products

So after you add a product you go back to all list products.

So if its a GET a form is shown and after you submit the form and you have a POST where the model takes place and runs the query (inserts info into DB).

**See Product Info**

|  |
| --- |
| Index.php |

Because there’s no values set yet then the home page is going to show.

|  |
| --- |
| Views  layout.php |

By clicking on the product you are assigning two get values

controller=product

action=readall

|  |
| --- |
| routes.php |

1. Calling a function called “call” which has controller and action as arguments
2. include controllers/product\_controller.php
3. Include models/product.php
4. controllers/product\_controller.php has a class called productcontroller which will be instantiated so now product = new productcontroller
5. The class above will call a function called readall like this product->readall

|  |
| --- |
| Controllers  Product\_controller.php |

As mentioned above.. This file has a method called readall().. Inside this method there’s another method called from models/product.php called all()

|  |
| --- |
| Models  product.php |

Function all() is called from the controllers.

This is where the connection is made and the query is made

|  |
| --- |
| Views  Products  readAll.php |

This is included from the Controllers/Product\_controller.php

This is where we could see the list of the products

All of the above is the same as the read all

When we press on the button (see product info)

controller=product

action=read

id=30

|  |
| --- |
| Controllers  Product\_controller.php |

This file has the function which as the action (read)

If GET has an id value then the code will run..

A find() method from models will run .. and an include

views/products/read.php

|  |
| --- |
| Models  product.php |

In this file a find() method will run..

Connect to the db..a select statement ..

|  |
| --- |
| Views  Products  read.php |

Images will show here

**Update product**

SAME AS READALL THEN WHEN YOU PRESS UPDATE

A GET METHOD WILL BE SET

controller=product

action=update

id=30

|  |
| --- |
| Controllers  Product\_controller.php |

This file has the function which as the action (read)

If GET has an id value then the code will run..

A find() method from models will run .. and an include

views/products/read.php

|  |
| --- |
| Models  product.php |

In this file a find() method will run..

Connect to the db..a select statement ..

|  |
| --- |
| Views  Products  read.php |

Images will show here