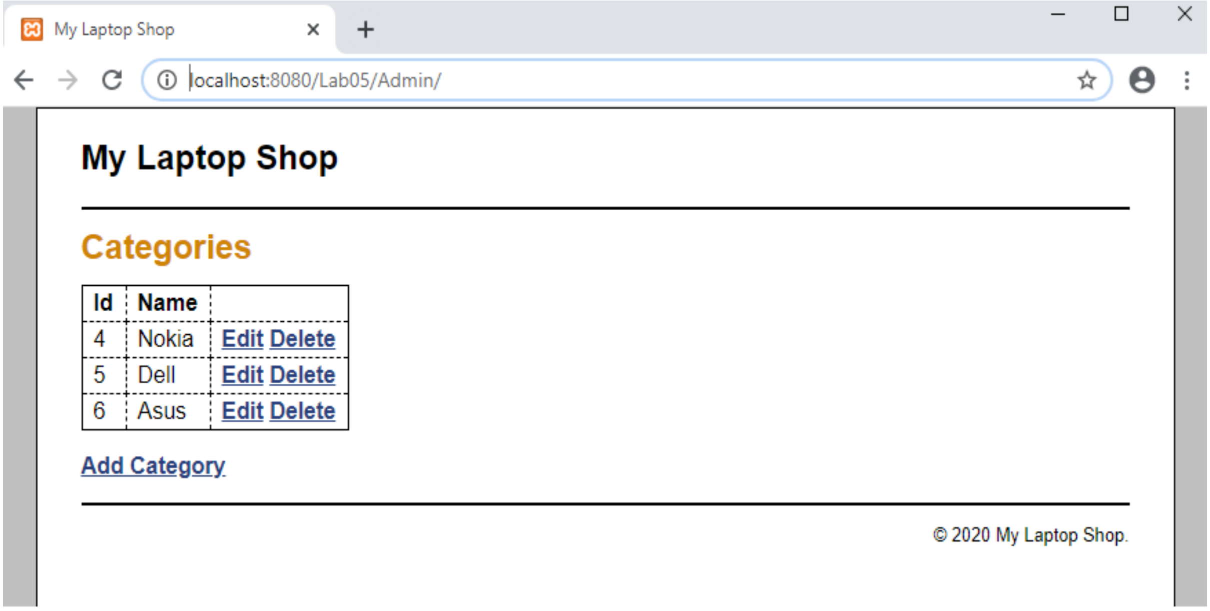
**Lab 05.02.** Xây dựng ứng dụng web theo mô hình MVC. Thao tác hiển thị, thêm, xóa sửa dữ liệu với bảng **Categories** của cơ sở dữ liệu **Lab05**. Giao diện ứng dụng website như sau:

+ Giao diện ứng dụng web



My Laptop Shop

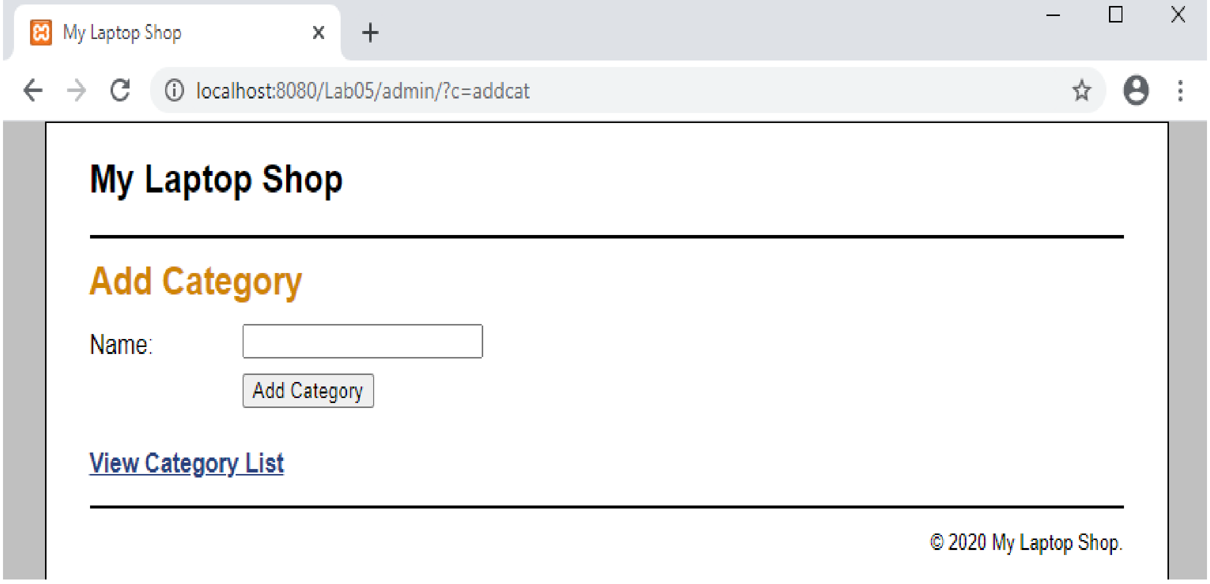
Categories

| Id | Name | |
|---|---|---|
| 4 | Nokia | Edit Delete |
| 5 | Dell | Edit Delete |
| 6 | Asus | Edit Delete |

Add Category

© 2020 My Laptop Shop.

+ Giao diện thêm category



My Laptop Shop

Add Category

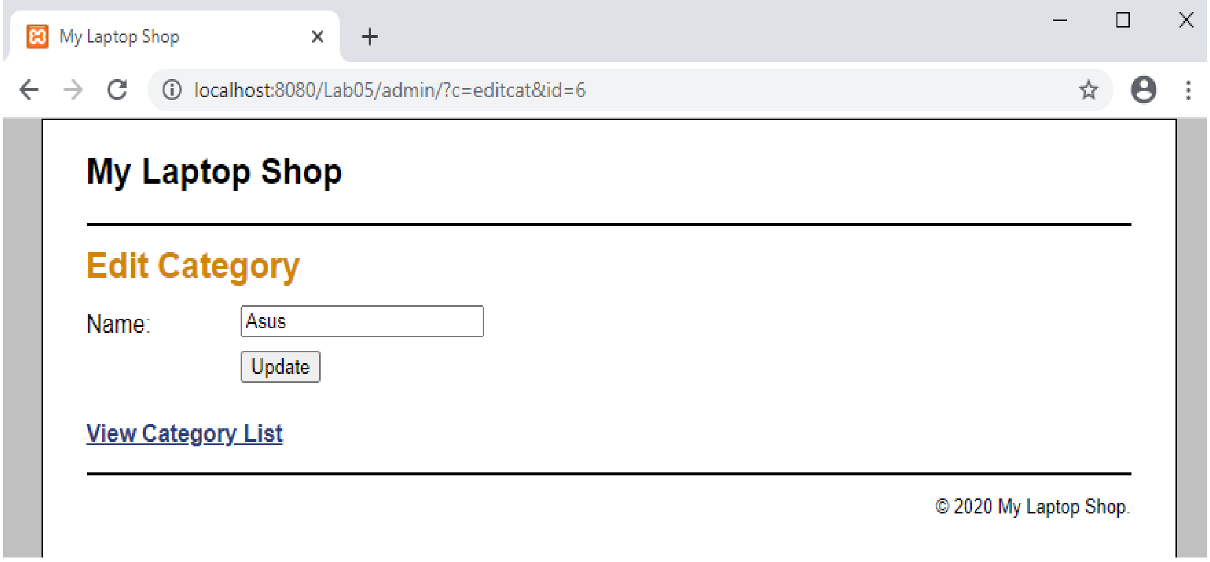
Name:

Add Category

View Category List

© 2020 My Laptop Shop.

+ Giao diện sửa nội dung Category



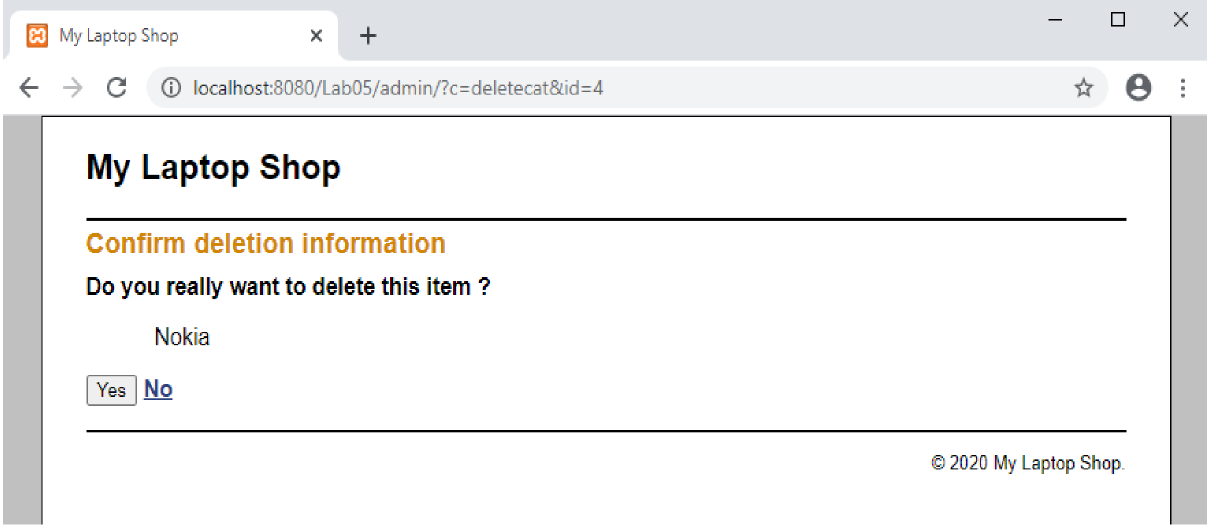
My Laptop Shop

Edit Category

Name: Asus

© 2020 My Laptop Shop.

+ Giao diện xóa Category



My Laptop Shop

Confirm deletion information

Do you really want to delete this item ?

Nokia

© 2020 My Laptop Shop.

<u>**Gợi ý:**</u>

- **Bước 1.** Trong MySQL, tạo cơ sở dữ liệu tên Lab05 với bảng Categories.

**CREATE DATABASE** IF NOT EXISTS Lab05 CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci';

**CREATE TABLE** `categories` (

 `categoryID` int(11) NOT NULL primary key AUTO_INCREMENT,

 `categoryName` varchar(255) NOT NULL

);

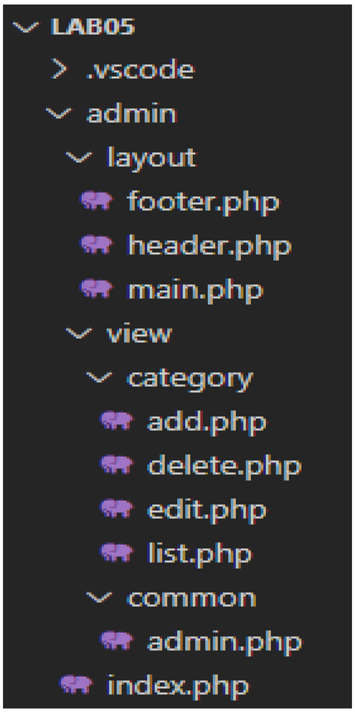
**INSERT INTO** `categories` (`categoryName`) VALUES

('Dell'),

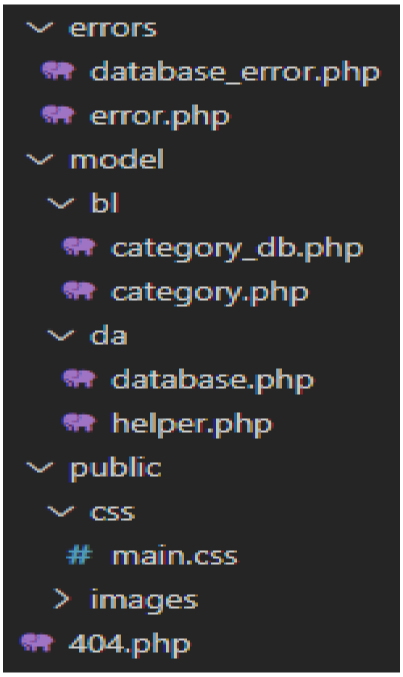
('Asus'),

('Hp');

- **Bước 2**. Tạo cấu trúc ứng dụng web như hình.

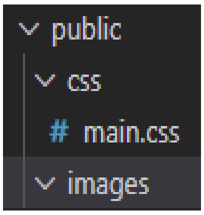


```
∨ LAB05
  > .vscode
  ∨ admin
    ∨ layout
        footer.php
        header.php
        main.php
    ∨ view
      ∨ category
          add.php
          delete.php
          edit.php
          list.php
      ∨ common
          admin.php
    index.php
```



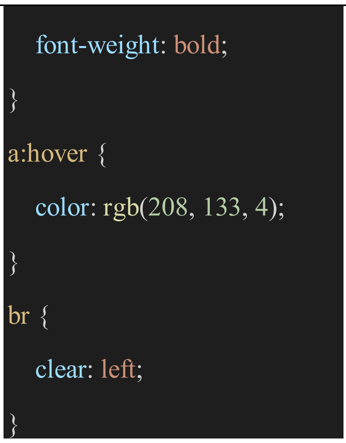
```
∨ errors
    database_error.php
    error.php
∨ model
  ∨ bl
      category_db.php
      category.php
  ∨ da
      database.php
      helper.php
∨ public
  ∨ css
    # main.css
  > images
  404.php
```

- **Bước 3.** Xây dựng file **main.css** cho ứng dụng:

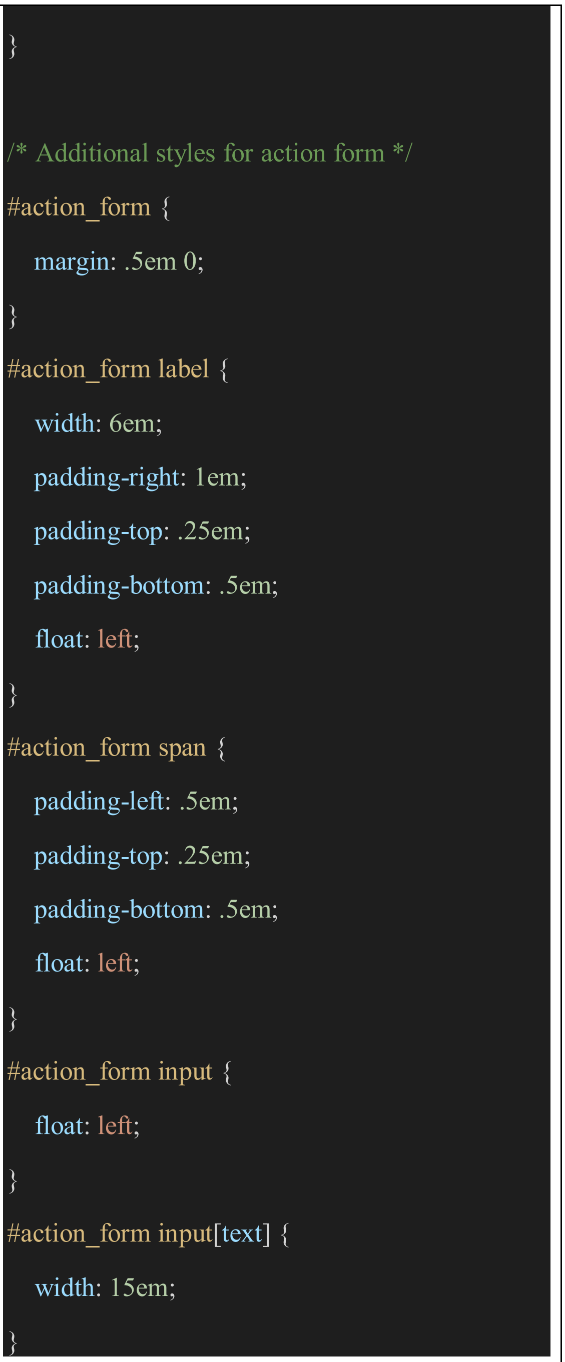


```
∨ public
  ∨ css
    # main.css
  ∨ images
```

```css
/* the styles for the HTML elements */
html {
    background-color: rgb(192, 192, 192);
}
body {
    font-family: Arial, Helvetica, sans-serif;
    width: 760px;
    margin: 0 auto;
    padding: 0 2em;
    background-color: white;
    border: 1px solid black;
}
header {
    border-bottom: 2px solid black;
    padding: .5em 0;
}
header h1 {
    color: black;
}
main {
    margin-bottom: 1em;
}
aside {
    float: left;
    width: 150px;
}
section {
    float: left;
    width: 500px;
}
footer {
    clear: both;
    border-top: 2px solid black;
}
footer p {
    text-align: right;
    font-size: 80%;
}
h1 {
    font-size: 150%;
    margin: .5em 0;
}
h2 {
    font-size: 120%;
    margin: .25em 0 .5em;
}
h1, h2 {
    color: rgb(208, 133, 4);
}
h3{
    font-size: 100%;
    margin: .25em 0 .5em;
}
ul {
    list-style-type: none;
    margin: 0;
    padding-left: 0;
    padding-bottom: 1em;
}
li {
    padding-bottom: 0.5em;
}
a {
    color: rgb(41, 64, 124);
table {
    border: 1px solid black;
    border-collapse: collapse;
    margin-bottom: 1em;
}
td, th {
    border: 1px dashed black;
    padding: .2em .5em .2em .5em;
    text-align: left;
}
/* the styles for classes */
.right {
    text-align: right;
}
.error {
    color: red;
}
.first_paragraph {
    margin-top: 0;
}
.last_paragraph {
    margin-bottom: 2em;
}
/* the styles for the div tags that divide the page in to sections */
#left_column {
    float: left;
    width: 150px;
    text-align: center;
}
#right_column {
    float: left;
    padding-left: 1em;
    padding-bottom: 2em;
```



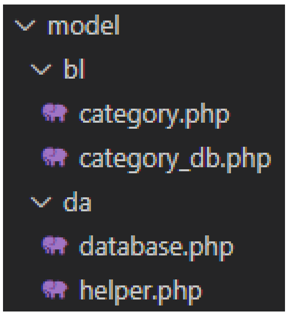
```
    font-weight: bold;
}
a:hover {
    color: rgb(208, 133, 4);
}
br {
    clear: left;
}
```



```
}

/* Additional styles for action form */
#action_form {
    margin: .5em 0;
}
#action_form label {
    width: 6em;
    padding-right: 1em;
    padding-top: .25em;
    padding-bottom: .5em;
    float: left;
}
#action_form span {
    padding-left: .5em;
    padding-top: .25em;
    padding-bottom: .5em;
    float: left;
}
#action_form input {
    float: left;
}
#action_form input[text] {
    width: 15em;
}
```

- **Bước 4**. Xây dựng các file cho thành phần **Model**., thành phần xử lý logic dữ liệu, tương tác trực tiếp với CSDL của ứng dụng.



```
⌄ model
  ⌄ bl
      category.php
      category_db.php
  ⌄ da
      database.php
      helper.php
```

+ file **da/database.php**

```
<?php
  class Database
  {
    private static $dsn = 'mysql:host=localhost;dbname=lab05';
    private static $username = 'root';
    private static $password = '123456';
    private static $con = null;
    public function __construct()
    {
      self::db_connect();
    }
    public static function db_connect()
    {
      try{
        if(is_null(self::$con))
        {
          self::$con = new PDO(self::$dsn,self::$username,self::$password);
          self::$con->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING);
        }
      }catch(PDOException $e)
      {
        $error_message = $e->getMessage();
        include_once('../errors/database_error.php');
      }
    }
    public static function db_disconnect()
    {
      if(!is_null(self::$con))
        self::$con = null;
    }
    public function __destruct()
    {
      self::db_disconnect();
```

```
}
public static function db_execute($sql = '',$params = [])
{
  if(!is_null(self::$con))
  {
    $result = self::$con->prepare($sql);
    $result->execute($params);
    if($result->rowCount() > 0)
    {
      $result->closeCursor();
      return true;
    }
  }
  return false;
}
public static function db_get_list($sql = '')
{
  if(!is_null(self::$con))
  {
    $result = self::$con->prepare($sql);
    $result->execute();
    if($result->rowCount() > 0)
    {
      $rows = $result->fetchAll();
      $result->closeCursor();
      return $rows;
    }
  }
  return false;
}
public static function db_get_list_condition($sql = '',$params = [])
{
  if(!is_null(self::$con))
  {
        $result = self::$con->prepare($sql);
        $result->execute($params);
        if($result->rowCount() > 0)
        {
          $rows = $result->fetchAll();
          $result->closeCursor();
          return $rows;
        }
      }
      return false;
    }
    public static function db_get_row($sql = '',$params = [])
    {
      if(!is_null(self::$con))
      {
        $result = self::$con->prepare($sql);
        $result->execute($params);
        if($result->rowCount() > 0)
        {
         $row = $result->fetch();
         $result->closeCursor();
         return $row;
        }
      }
      return false;
    }
  }
?>
```

**+ file da/helper.php**

```
<?php
class Helper
{
   //http://localhost:8080/project/
   public static function get_url($url = '')
```

```php
    {
        $uri = filter_input(INPUT_SERVER, 'REQUEST_URI', FILTER_SANITIZE_STRING);
        $app_path = explode('/', $uri);
        return 'http://' . $_SERVER['HTTP_HOST'] . '/' . $app_path[1] . '/' . $url;
    }
    public static function redirect($url)
    {
        header("Location:{$url}");
        exit();
    }
    public static function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    public static function input_value($inputname,$filter=FILTER_DEFAULT)
    {
        $value = filter_input(INPUT_POST, self::test_input($inputname),$filter);
        if ($value == NULL)
            $value = filter_input(INPUT_GET,self::test_input($inputname),$filter);
        return $value;
    }
    public static function is_submit($hidden)
    {
        return (!empty(self::input_value('action')) && self::input_value('action') == $hidden);
    }
}
```

**+ file bl/category.php**

```php
<?php
class Category {
    private $id,$name;
    public function getId() {
        return $this->id;
    }
    public function setId($value) {
        $this->id = $value;
    }
    public function getName() {
        return $this->name;
    }
    public function setName($value) {
        $this->name = $value;
    }
}
?>
```

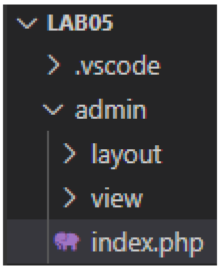
**+ file bl/category_db.php**

```
<?php
class CategoryDB extends Database
{
   public static function getCategories()
   {
        $sql = "select * from categories";
        if(!empty(self::db_get_list($sql)))
        {
            foreach(self::db_get_list($sql) as $row){
                $category = new Category();
                $category->setId($row['categoryID']);
                $category->setName($row['categoryName']);
                $categorys[] = $category;
            }
            return $categorys;
        }
        return false;
   }
   public static function getCategoryById($categoryid)
   {
        $sql = "select * from categories where categoryID=:categoryID";
        $params = ['categoryID' => $categoryid];
        $row = self::db_get_row($sql, $params);
        if(!empty($row))
        {
            $category = new Category();
            $category->setId($row['categoryID']);
            $category->setName($row['categoryName']);
            return $category;
        }
        return false;
    }

    public static function addCategeory($category)
    {
        $sql = "insert into categories(categoryName) values(:categoryName)";
        $params = [
            "categoryName" => $category->getName()
        ];
        if (self::db_execute($sql, $params))
            return true;
        else
            return false;
    }
    public static function updateCategeory($category)
    {
        $sql = "update categories set categoryName=:categoryName where categoryID=:categoryID";
        $params = [
            "categoryID" => $category->getId(),
            "categoryName" => $category->getName()
        ];
        if (self::db_execute($sql, $params))
            return true;
        else
```

```
            return false;
    }
    public static function deleteCategeory($category)
    {
        $sql = "delete from categories where categoryID=:categoryID";
        $params = [
            "categoryID" => $category->getId()
        ];
        if (self::db_execute($sql, $params))
            return true;
        else
            return false;
    }
}
?>
```

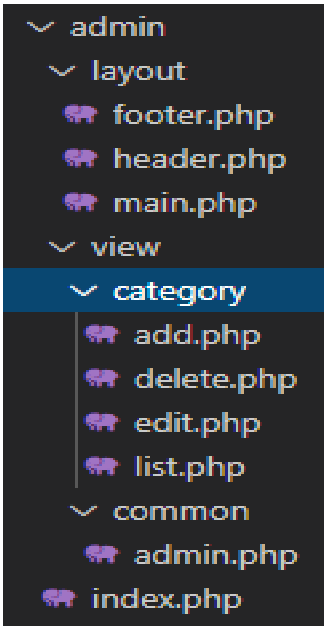
+ **Bước 5.** Xây dựng file **index.php** là thành phần controller, điều hướng và xử lý các yêu cầu của ứng dụng.



```
<?php
    include_once('../model/da/database.php');
    include_once('../model/da/helper.php');
    include_once('../model/bl/category.php');
    include_once('../model/bl/category_db.php');
    $db = new Database();
?>
<base href="<?php echo Helper::get_url('admin/'); ?>">
<link rel="stylesheet" type="text/css" href="../public/css/main.css">
<?php
  $view = filter_input(INPUT_GET, 'v');
  $action = filter_input(INPUT_GET, 'a');
```

```
 if(empty($view) || empty($action))
 {
   $view = 'common';
   $action = 'admin';
 }
 $path = 'view/'.$view.'/'.$action.'.php';
 if(file_exists($path))
 {
  include_once($path);
 }
 else
    {
     header('Location:../404.php');
    }
?>
```

+ **Bước 6**. Định nghĩa các file thành phần **layout** để tạo bố cục cho trang admin.php.



```
admin
  layout
    footer.php
    header.php
    main.php
  view
    category
      add.php
      delete.php
      edit.php
      list.php
    common
      admin.php
  index.php
```

+ **file view/footer.php**

```
<footer>
  <p class="copyright">
    &copy; <?php echo date("Y"); ?> My Laptop Shop.
  </p>
</footer>
```

```
</body>
</html>
```

## + file view/header.php

```
<!DOCTYPE html>
<html>
<!-- the head section -->
<head>
   <title>My Laptop Shop</title>
   </head>
<!-- the body section -->
<body>
<header><h1>My Laptop Shop</h1></header>
```

## + file view/main.php

```
<?php
  $content = Helper::input_value('c');
  if(!empty($content))
  {
     switch($content)
     {
        case "addcat":
          include_once("view/category/add.php");
          break;
        case "editcat":
          include_once("view/category/edit.php");
          break;
        case "deletecat":
          include_once("view/category/delete.php");
          break;
     }
  }
  else
     include_once("view/category/list.php");
?>
```

### + file common/admin.php

```
<?php include_once('layout/header.php')?>
<main>
   <section>
      <!-- display table of categories -->
      <?php include_once('layout/main.php')?>
   </section>
</main>
<?php include_once('layout/footer.php') ?>
```

### + file category/add.php

```
<?php
  if(Helper::is_submit('add_category'))
  {
    $category = new Category();
    $category->setName(Helper::input_value('name'));
    if(!empty($category->getName()) && CategoryDB::addCategeory($category))
    {
       Helper::redirect('.');
    }
  }
?>
<h1>Add Category</h1>
<form action="" method="post" id="action_form">
   <input type="hidden" name="action" value="add_category">
   <label>Name:</label>
   <input type="input" name="name" value="<?php echo Helper::input_value('name');  ?>">
   <br>
   <label>&nbsp;</label>
   <input type="submit" value="Add Category">
   <br>
</form>
<p><a href="<?php echo Helper::get_url('admin/');?>">View Category List</a></p>
```

**+ file category/delete.php**

```
<?php
 if(!empty(Helper::input_value('id')))
 {
   $category = CategoryDB::getCategoryById(Helper::input_value('id'));
   if(Helper::is_submit('delete_category'))
   {
     if(CategoryDB::deleteCategeory($category))
     {
       Helper::redirect('.');
     }
   }
 }
?>
<h2>Confirm deletion information</h2>
<h3>Do you really want to delete this item ?</h3>
<p style="margin-left:50px"><?php echo $category->getName();?></p>
<div>
   <form action="" method="post">
     <input type="hidden" name="action" value="delete_category">
     <input type="submit" value="Yes">
     <a href="<?php Helper::get_url('.'); ?>">No</a>
   </form>
</div>
```

**+ file category/edit.php**

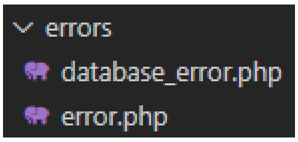
```
<?php
  if(!empty(Helper::input_value('id')))
  {
    $category = CategoryDB::getCategoryById(Helper::input_value('id'));
    if(Helper::is_submit('edit_category'))
    {
      $category->setName(Helper::input_value('name'));
      if(CategoryDB::updateCategeory($category))
         Helper::redirect('.');
      }
   }
?>
<h1>Edit Category</h1>
<form action="" method="post" id="action_form">
   <input type="hidden" name="action" value="edit_category">
   <label>Name:</label>
   <input type="input" name="name" value="<?php echo $category->getName();  ?>">
   <br>
   <label>&nbsp;</label>
   <input type="submit" value="Update">
   <br>
</form>
<p><a href="<?php echo Helper::get_url('admin/');?>">View Category List</a></p>
```

### + file category/list.php

```
<h1>Categories</h1>
      <table>
         <tr>
            <th>Id</th>
            <th>Name</th>
            <th>&nbsp;</th>
         </tr>
         <?php
            if(!empty(CategoryDB::getCategories()))
              foreach (CategoryDB::getCategories() as $category) : ?>
         <tr>
            <td><?php echo $category->getId(); ?></td>
            <td><?php echo $category->getName(); ?></td>
            <td>
            <a href="<?php echo Helper::get_url('admin/?c=editcat&id=' . $category->getId());?>">Edit</a>
            <a href="<?php echo Helper::get_url('admin/?c=deletecat&id=' . $category->getId());?>">Delete</a>
            </td>
         </tr>
         <?php endforeach; ?>
```

```
    </table>
<p><a href="<?php echo Helper::get_url('admin/?c=addcat');?>">Add Category</a></p>
```

- **Bước 7.** Định nghĩa các file xử lý lỗi ứng dụng



```
errors
  database_error.php
  error.php
```

```
404.php
```

**+ file errors/database_error.php**

```
<main>
  <h1>Database Error</h1>
  <p class="first_paragraph">There was an error connecting to the database.</p>
  <p class="last_paragraph">Error message: <?php echo $error_message; ?></p>
</main>
```

**+ file errors/error.php**

```
<main>
  <h1>Error</h1>
  <p class="first_paragraph"><?php echo $error; ?></p>
</main>
```

**+ file 404.php**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Not Found</title>
</head>
<body>
  <h3>Not Found</h3>
  <hr>
  HTTP Error 404. The requested resource is not found.
</body>
</html>
```