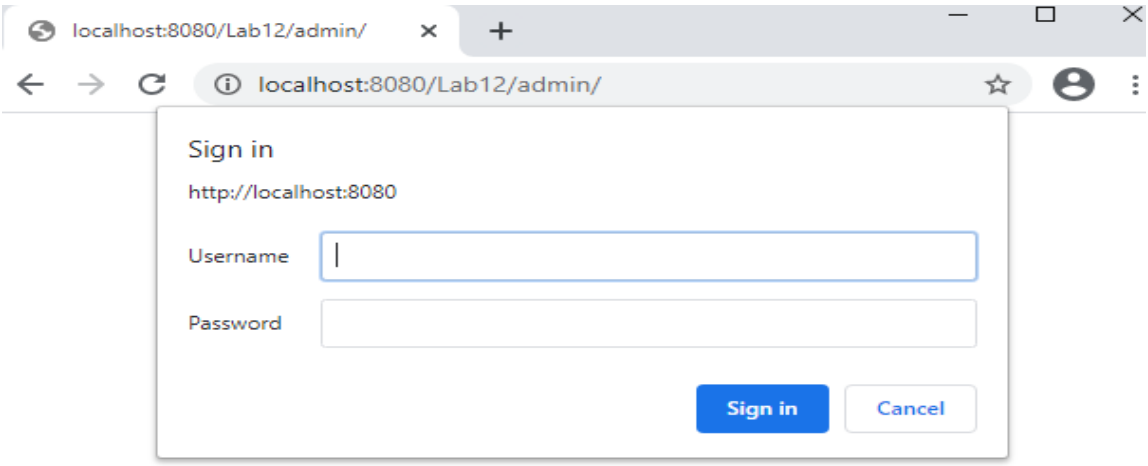
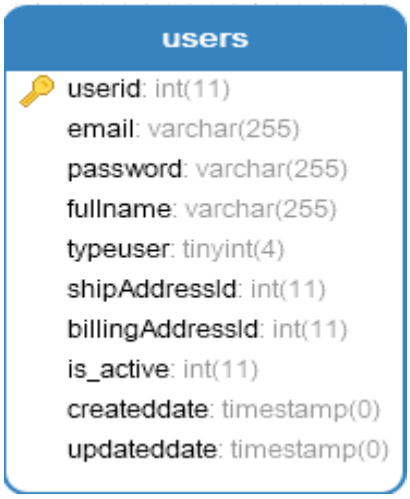
Lab 12.2. Xây dựng chức năng Authentication cho ứng dụng

Bài 01. Xây dựng chức năng **authentication basic** bảo vệ thư mục **Admin** cho ứng dụng web, với giao diện ứng dụng như sau:



Gợi ý:

+ **B1.** Tạo bảng **Users** lưu thông tin người dùng sử dụng hệ thống ứng dụng web.



create table users(

userid int **not null auto_increment primary key**,

email varchar(255) **not null**,

password varchar(255) **not null**,

fullname varchar(255) **null**,

typeuser tinyint **default 2** COMMENT '**1: admin 2: user**',

shipAddressId int **null**,

billingAddressId int **null**,

is_active int **default 1**,

createddate timestamp **null default current_timestamp**,

updateddate timestamp null **default null on update current_timestamp**

);

+ **B2.** Tạo file **model\bl\user.php** định nghĩa các thuộc tính và phương thức mô tả bảng dữ liệu users trong cơ sở dữ liệu **Lab12**.

```
<?php
class User
{
   private $userid,$email,$password,
        $fullname,$typeuser,$is_active,$shipaddressid,$billingaddressid;
   public function setUserId($value)
   {
      $this->userid = $value;
   }
   public function getUserId()
   {
      return $this->userid;
   }
   public function setEmail($value)
   {
      $this->email = $value;
   }
   public function getEmail()
   {
      return $this->email;
   }
   public function setPassword($value)
   {
      $this->password = $value;
   }
   public function getPassword()
   {
      return $this->password;
   }
```

```
public function setFullname($value)
{
    $this->fullname = $value;
}
public function getFullname()
{
    return $this->fullname;
}
public function setTypeUser($value)
{
    $this->typeuser = $value;
}
public function getTypeUser()
{
    return $this->typeuser;
}
public function setShipAddressId($value)
{
    $this->shipaddressid = $value;
}
public function getShipAddressId()
{
    return $this->shipaddressid;
}
public function setBillingAddressId($value)
{
    $this->billingaddressid = $value;
}
public function getBillingAddressId()
{
    return $this->billingaddressid;
}
public function setIsActive($value)
{
```

```
        $this->is_active = $value;
    }
    public function getIsActive()
    {
        return $this->is_active;
    }
}
?>
```

+ B3. Tạo file **model\bl\user_db.php** định nghĩa các phương thức xử lý nghiệp vụ như thêm, xóa, sửa và lấy thông tin người dùng … Trong đó, với chức năng demo ứng dụng bài tập này, quan trọng nhất sử dụng hai phương thức **is_valid_admin_login()** và **addUser()**.

```
<?php
class UserDB extends Database
{
  public static function getUsers()
  {
      $sql = "select * from users";
      if(!empty(self::db_get_list($sql)))
      {
          foreach(self::db_get_list($sql) as $row){
              $user = new User();
              $user->setUserId($row['userid']);
              $user->setEmail($row['email']);
              $user->setPassword($row['password']);
              $user->setFullname($row['fullname']);
              $user->setShipAddressId($row['ShipAddressId']);
              $user->setBillingAddressId($row['BillingAddressId']);
              $user->setTypeUser($row['typeuser']);
              $user->setIsActive($row['is_active']);
              $users[] = $user;
          }
          return $users;
      }
```

```
        return false;
    }

    public static function getUsersPaging(&$paging_html)
    {
        $link = Helper::get_url('admin/?c=listu&page={page}');
        $sql = "select * from users";
        $total_records = self::db_num_rows($sql);
        $current_page = Helper::input_value('page');
        $limit = 5;
        $paging = Helper::paging($link,$total_records,$current_page,$limit);
        $paging_html = $paging['html'];
        $sql = "select * from users limit {$paging['start']},{$paging['limit']}";
        if(!empty(self::db_get_list($sql)))
        {
            foreach(self::db_get_list($sql) as $row){
                $user = new User();
                $user->setUserId($row['userid']);
                $user->setEmail($row['email']);
                $user->setPassword($row['password']);
                $user->setFullname($row['fullname']);
                $user->setShipAddressId($row['ShipAddressId']);
                $user->setBillingAddressId($row['BillingAddressId']);
                $user->setTypeUser($row['typeuser']);
                $user->setIsActive($row['is_active']);
                $users[] = $user;
            }
            return $users;
        }
        return false;
    }
```

```
public static function getUserById($userid)
{
  $sql = "select * from users where userid=:userid";
  $params = ['userid'=>$userid];
  $row = self::db_get_row($sql, $params);
  if(!empty($row))
  {
    $user = new User();
    $user->setUserId($row['userid']);
    $user->setEmail($row['email']);
    $user->setPassword($row['password']);
    $user->setFullname($row['fullname']);
    $user->setShipAddressId($row['ShipAddressId']);
    $user->setBillingAddressId($row['BillingAddressId']);
    $user->setTypeUser($row['typeuser']);
    $user->setIsActive($row['is_active']);
    return $user;
  }
  return false;
}

public static function getUserByEmail($email)
{
  $sql = "select * from users where email=:email";
  $params = ['email'=>$email];
  $row = self::db_get_row($sql, $params);
  if(!empty($row))
  {
    $user = new User();
    $user->setUserId($row['userid']);
    $user->setEmail($row['email']);
    $user->setPassword($row['password']);
    $user->setFullname($row['fullname']);
    $user->setShipAddressId($row['ShipAddressId']);
```

```
        $user->setBillingAddressId($row['BillingAddressId']);
        $user->setTypeUser($row['typeuser']);
        $user->setIsActive($row['is_active']);
        return $user;
      }
     return false;
  }

  public static function register_user($user) {
      $sql ="insert into users(email,password,fullname) values(:email,:password,:fullname)";
      $params = [
          "email" => $user->getEmail(),
          "password" => password_hash($user->getPassword(),PASSWORD_DEFAULT),
          "fullname" => $user->getFullname()
      ];
      if (self::db_execute($sql, $params))
        return true;
      else
        return false;
  }

  public static function is_valid_admin_login($email, $password) {
     $sql = "select password, typeuser,is_active from users where email = :email";
     $params = ['email'=>$email];
     $row = self::db_get_row($sql, $params);
     if(!empty($row))
     {
        $hash = $row['password'];
        $typeuser = $row['typeuser'];
        $is_active = $row['is_active'];
    return (password_verify($password, $hash) && ($typeuser == 1) && ($is_active == 1));
     }
     return false;
  }
```

```
  public static function addUser($user)
  {
    $sql ="insert into users(email,password,fullname,typeuser,is_active) values(:email,
:password,:fullname,:typeuser,:is_active)";
     $params = [
        "email" => $user->getEmail(),
        "password" => password_hash($user->getPassword(),PASSWORD_DEFAULT),
        "fullname" => $user->getFullname(),
        "typeuser" => $user->getTypeUser(),
        "is_active" => $user->getIsActive()
     ];
     if (self::db_execute($sql, $params))
        return true;
     else
        return false;
  }
  public static function updateUser($user)
  {
    $sql = "update users set email=:email, password=:password, fullname=:fullname, typeuse
r=:typeruser, is_active=:is_active where userid=:userid";
    $params = [
      "userid" => $user->getUserId(),
      "email" => $user->getEmail(),
      "password" => $user->getPassword(),
      "fullname" => $user->getFullname(),
      "typeuser" => $user->getTypeUser(),
      "is_active" => $user->getIsActive()
    ];
    if (self::db_execute($sql, $params))
       return true;
    else
       return false;
  }
```

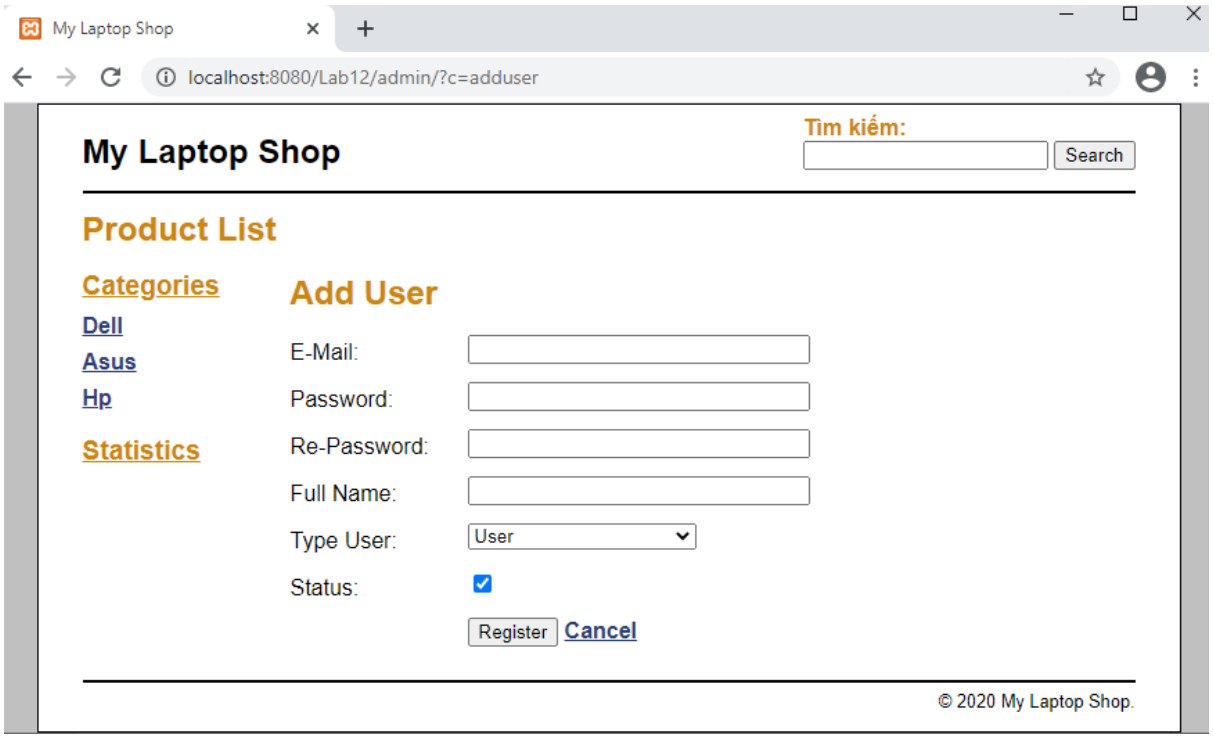
```
  public static function deleteUser($user)
  {
    $sql = "delete from users where userid=:userid";
    $params = [
      "userid" => $user->getUserId()
    ];
    if (self::db_execute($sql, $params))
      return true;
    else
      return false;
  }
}
?>
```

+ **B4.** Tạo file **model\da\auth_basic.php**, định nghĩa lớp **Auth_Basic** xử lý nghiệp vụ authentication basic để bảo vệ thư mục **Admin** cho ứng dụng web.

```
<?php
  class Auth_Basic
  {
    private static $email = '';
    private static $password = '';
    public function __construct()
    {
 if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER['PHP_AUTH_PW'])){
        Auth_Basic::$email = $_SERVER['PHP_AUTH_USER'];
        Auth_Basic::$password = $_SERVER['PHP_AUTH_PW'];
      }
 if(!UserDB::is_valid_admin_login(Auth_Basic::$email,Auth_Basic::$password))
      {
        header('WWW-Authenticate: Basic realm="Admin"');
        header('HTTP/1.0 401 Unauthorized');
        Helper::redirect_js(".");
        exit();
      }
    }
```

```
  }
?>
```

+ **B5.** Xây dựng giao diện trang **admin\view\user\add.php,** tạo người dùng cho **Admin**, với giao diện chức năng như sau:




```
<?php
  $action = filter_input(INPUT_POST,'action');
  if(Helper::is_submit('adduser'))
  {
    $user = new User();
    $user->setEmail(Helper::input_value('email'));
    $user->setPassword(Helper::input_value('password'));
    $user->setFullname(Helper::input_value('fullname'));
    $user->setTypeUser(Helper::input_value('typeuser',FILTER_VALIDATE_INT));
    $user->setIsActive((Helper::input_value('is_active',FILTER_VALIDATE_INT)==1)?1:0);
    if(UserDB::addUser($user))
    {
      Helper::redirect('?c=listuser');
      //echo "<h4>Đã thêm người dùng !</h4>";
    }
  }
```

```
?>
<main>
  <h1>Add User</h1>
  <form action="" method="post" id="login_form">
    <input type="hidden" name="action" value="adduser">
    <label>E-Mail:</label>
    <input type="text" name="email"
        value="<?php echo Helper::input_value('email');  ?>" size="30">
    <br>
    <label>Password:</label>
    <input type="password" name="password" id="password" size="30">
    <br>
    <label>Re-Password:</label>
    <input type="password" name="repassword" size="30">
    <br>
    <label>Full Name:</label>
    <input type="text" name="fullname"
        value="<?php echo Helper::input_value('fullname');  ?>" size="30">
    <br>
    <label>Type User:</label>
    <select name="typeuser">
      <option value="">-- Choose a user type --</option>
      <option value=1 <?php echo (Helper::input_value('typeuser'))==1?'selected':'';  ?>>Admin
</option>
      <option value=2 selected <?php echo (Helper::input_value('typeuser'))==2?'selected':'';  ?>>
User</option>
    </select>
    <br>
    <label>Status:</label>
    <input type="checkbox" name="is_active" value=1 checked <?php echo (Helper::input_
value('is_active'))==1?'checked':'';  ?>>
    <br>
    <label> </label>
    <input type="submit" value="Register">
```

```
      <a href="#">Cancel</a>
   </form>
</main>
<script>
   $(document).ready(function(){
    //validation
    $("#login_form").validate({
        rules:{
         email:{
            required:true,
            email: true
         },
         password:{
            required:true
         },
         repassword:{
            equalTo:"#password"
         },
         fullname:{
            required:true
         }
        },
        messages:{
         email:{
            required:"<div>Input email !</div>",
            email:"<div>Incorrect email format !</div>"
         },
         password:{
            required:"<div>Input pasword !</div>"
         },
         repassword:{
            equalTo:"<div>Incorrect !</div>"
         },
         fullname:{
```

```
            required:"<div>Input fullname !</div>"
          }
        }
    });
    $.validator.methods.email = function(value, element) {
     return this.optional(element) || /^[\w\.]+\@\w+\.[a-zA-Z]{2,3}(\.[a-zA-Z]{2,3})?$/.test(value);
    }
  });
</script>
```

=> **Lưu ý:** các em dựa vào các phương thức xử lý nghiệp vụ đã định nghĩa ở lớp **UserDB** (bl\user_db.php đã tạo ở **bước số 3**). Bổ sung thêm các chức năng xóa, sửa, hiển thị danh sách, tìm kiếm thông tin người dùng cho Admin.

+ **B6.** Cập nhật lại nội dung file **admin\layout\main.php**, cho phép ứng dụng web điều hướng đến các chức năng thêm, xóa, sửa, hiển thị thông tin người dùng.

```
<?php
 $content = Helper::input_value('c');
 if(!empty($content))
 {
   switch($content)
   {
      .....
      case "adduser":
        include_once("view/user/add.php");
        break;
      case "deleteuser":
        include_once("view/user/delete.php");
        break;
      case "edituser":
        include_once("view/user/edit.php");
        break;
      case "listuser":
        include_once("view/user/list.php");
        break;
   }
```

```
  }
  else
      include_once("view/product/list.php");
?>
```

+ **B7.** Mở file **admin\index.php** cập nhật lại nội dung đoạn mã sau để thêm các file thư viện đã định nghĩa và bảo vệ nội dung thư mục **Admin**.

```
<?php
    ……….
    include_once('../model/da/auth_basic.php');
    include_once('../model/bl/category.php');
    include_once('../model/bl/category_db.php');
    include_once('../model/bl/product.php');
    include_once('../model/bl/product_db.php');
    include_once('../model/bl/user.php');
    include_once('../model/bl/user_db.php');
    $db = new Database();
    //Create key for encryption
    $h = new Helper();
?>
……..
<script src="../public/js/jquery.validate.js"></script>
<?php
  $auth = new Auth_Basic();//Lệnh này mục đích bảo vệ thư mục Admin, nếu người dùng không xác thực thông tin username và password sẽ không truy cập được nội dung trong thư mục Admin.
  $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');
    }
?>
```

Bài 02. Tương tự các em, xây dựng chức năng **authentication form** để bảo vệ thư mục **Admin** cho ứng dụng web.

Bài 03. Xây dựng chức năng **authentication form** cho phân hệ người dùng. Nếu người dùng muốn đặt hàng hoặc xem chi tiết thông tin về đơn hàng … bắt buộc người dùng phải đăng ký tài khoản với ứng dụng web.

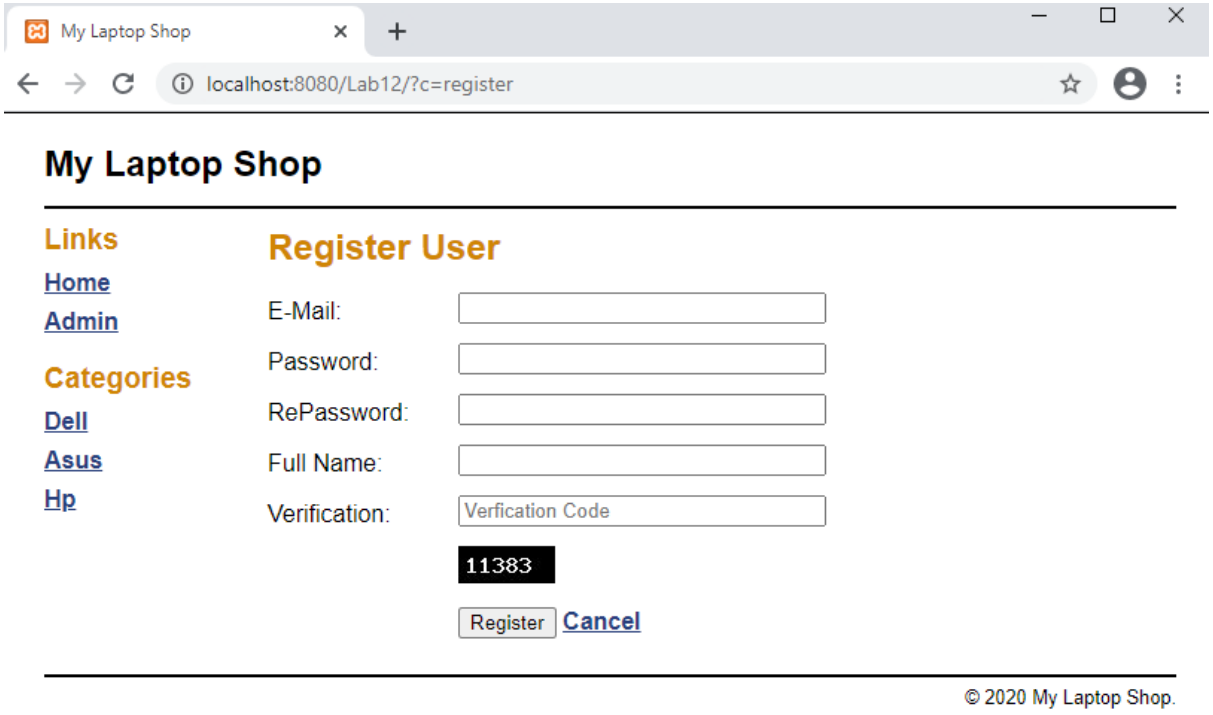
Gợi ý:

- B1. Tạo file **model\da\captcha.php**, xây dựng chức năng **Captcha** cho ứng dụng.

```
<?php
   session_start();
   class Captcha
   {
      public static $text;
      public function __construct()
      {
         self::$text = mt_rand(10000,99999);
         $_SESSION['vercode'] = self::$text;
         $height = 25;
         $width = 65;
         $image_p = imagecreate($width, $height);
         $black = imagecolorallocate($image_p, 0, 0, 0);
         $white = imagecolorallocate($image_p, 255, 255, 255);
         $font_size = 14;
         imagestring($image_p, $font_size, 5, 5,  self::$text, $white);
```

```
            imagejpeg($image_p, null, 80);
        }
    }
    $captcha = new Captcha();
?>
```

- B2. Xây dựng giao diện cho trang **view\common\register.php**, cho phép người dùng có thể tự tạo tài khoản trên hệ thống ứng dụng web.




```
<?php
$action = filter_input(INPUT_POST,'action');
$vercode = filter_input(INPUT_POST,'vercode');
if(Helper::is_submit('register'))
{
  $user = new User();
  $user->setEmail(Helper::input_value('email'));
  $user->setPassword(Helper::input_value('password'));
  $user->setFullname(Helper::input_value('fullname'));
  if(isset($vercode) && isset($_SESSION['vercode']) && ($vercode == $_SESSION['vercode']) && UserDB::register_user($user))
  {
     echo "<h4>Đã thêm người dùng !</h4>";
```

```
    }
}
?>
<main>
    <h1>Register User</h1>
    <form action="" method="post" id="login_form">
        <input type="hidden" name="action" value="register">
        <label>E-Mail:</label>
        <input type="text" name="email"
            value="<?php echo Helper::input_value('email');  ?>" size="30">
        <br>
        <label>Password:</label>
        <input type="password" name="password" size="30" id="password">
        <br>
        <label>RePassword:</label>
        <input type="password" name="repassword" size="30">
        <br>
        <label>Full Name:</label>
        <input type="text" name="fullname"
            value="<?php echo Helper::input_value('fullname');  ?>" size="30">
        <br>
        <label>Verification:</label>
        <input type="text" name="vercode" placeholder="Verfication Code" size="30"> <br>
        <img style="margin-left:8em;margin-
bottom:1em" src="model/da/captcha.php" alt="captcha">
        <br>
        <label> </label>
        <input type="submit" value="Register">
        <a href="#">Cancel</a>
    </form>
</main>
<script>
    $(document).ready(function(){
      //validation
```

```
$("#login_form").validate({
    rules:{
      email:{
          required:true,
          email: true
      },
      password:{
          required:true
      },
      repassword:{
          equalTo:"#password"
      },
      fullname:{
          required:true
      }
    },
    messages:{
      email:{
          required:"<div>Input email !</div>",
          email:"<div>Incorrect format !</div>"
      },
      password:{
          required:"<div>Input pasword !</div>"
      },
      repassword:{
          equalTo:"<div>Incorrect !</div>"
      },
      fullname:{
          required:"<div>Input fullname !</div>"
      }
    }
});
$.validator.methods.email = function(value, element) {
  return this.optional(element) || /^[\w\.]+\@\w+\.[a-zA-Z]{2,3}(\.[a-zA-Z]{2,3})?$/.test(value);
```

```
    }
  });
</script>
```

- B3. Mở file **layout\main.php** cập nhật thêm đoạn mã sau để điều hướng đến các trang **common\login.php** và **common\register.php**

```
<?php
 $content = Helper::input_value('c');
 if(!empty($content))
 {
   switch($content)
   {
      case "listcatid":
        include_once("view/product/listByCatId.php");
        break;
      case "productid":
        include_once("view/product/productById.php");
        break;
      case "register":
        include_once("view/common/register.php");
        break;
      case "login":
        include_once("view/common/login.php");
        break;
   }
 }
 else
     include_once("view/product/featuredProducts.php");
?>
```

- B4. Mở file controller: **index.php** cập nhật lại đoạn mã code sau:

```
<?php
   include_once('model/da/database.php');
   include_once('model/da/helper.php');
   include_once('model/da/session.php');
   include_once('model/da/role.php');
```

```
    include_once('model/bl/category.php');
    include_once('model/bl/category_db.php');
    include_once('model/bl/product.php');
    include_once('model/bl/product_db.php');
    include_once('model/bl/user.php');
    include_once('model/bl/user_db.php');
    $db = new Database();
?>
<base href="<?php echo Helper::get_url(''); ?>">
<link rel="stylesheet" type="text/css" href="public/css/main.css">
<script src="public/Js/jquery-3.4.1.min.js"></script>
<script src="public/js/jquery.validate.js"></script>

<?php
  $view = filter_input(INPUT_GET, 'v');
  $action = filter_input(INPUT_GET, 'a');

  if(empty($view) || empty($action))
  {
     $view = 'common';
     $action = 'home';
  }

  $path = 'view/'.$view.'/'.$action.'.php';

  if(file_exists($path))
  {
    include_once($path);
  }
  else
      {
       header('Location:404.php');
      }
?>
```

- B5. Tương tự sinh viên tự xây dựng chức năng **login** cho **phân hệ người dùng**, với giao diện ứng dụng như sau: