**NODE JS 09: MASTER PAGE FRONTEND - SẢN PHẨM NỔI BẬT - SẢN PHẨM MỚI**

A – LÝ THUYẾT

I – ASYNC & AWAIT TRONG JAVASCRIPT

**1. Vì sao phải sử dụng Async & Await**

Xử lý bất đồng bộ là một đặc sản của Javascript thông qua các Callback Function, và nó luôn là nỗi ám ảnh của chúng ta khi xử lý chúng trong ES5. Tuy nhiên với ES6, Promise đã giải quyết rất tốt với cấu trúc then… catch giúp code dễ đọc và bắt lỗi tốt hơn

|  |
| --- |
| let p = new Promise(function(resolve, reject) {  setTimeout(() => resolve(4), 2000);  });  // handler can't change promise, just value  p.then((res) => {  console.log(res);  }); |

Tuy nhiên nếu một bài toán nào đó đòi hỏi nhiều xử lý bất đồng bộ cùng lúc và phụ thuộc vào nhau (lồng nhau) thì phải giải quyết như thế nào ?

|  |
| --- |
| getA().then(  a => {  getB(a).then(  b => {  getC(b).then(  c => a + b + c,  err => console.log(err)  )  },  err => console.log(err)  )  },  err => console.log(err)  ) |

Rất may mắn là ES7 xuất hiện và giải quyết tình huống còn lại của Promise cực kỳ hiệu quả và khoa học. Đoạn code trên được viết lại một cách ngắn gọn và dễ hiểu như sau:

|  |
| --- |
| let getTotal = async () => {  try {  let a = await getA();  let b = await getB(a);  let c = await getC(b);  return Promise.resolve(a + b + c);  } catch (e) {  return Promise.reject(e);  }  }  getTotal()  .then(data => console.log(data))  .catch(err => console.log(err + '')); |

**2. Cách sử dụng**

Chúng ta cùng nhìn qua một ví dụ thực tế về cach thức một phương thức của Controller lấy 2 trị phụ thuộc nhau từ 2 Promise bên Model sẽ như thế nào khi sử dụng Async & Await

* Model:

|  |
| --- |
| function productFeatured(){  var promise = new Promise(function(rs, rj){    connection.query('SELECT \* FROM product WHERE prd\_featured = 1 LIMIT 6', function (error, results, fields) {  rs(results);  });  });  return promise;  }  function productLatest(){  var promise = new Promise(function(rs, rj){    connection.query('SELECT \* FROM product ORDER BY prd\_id DESC LIMIT 6', function (error, results, fields) {  rs(results);  });  });  return promise;  }  module.exports = {  productFeatured:productFeatured,  productLatest:productLatest,  } |

* Controller:

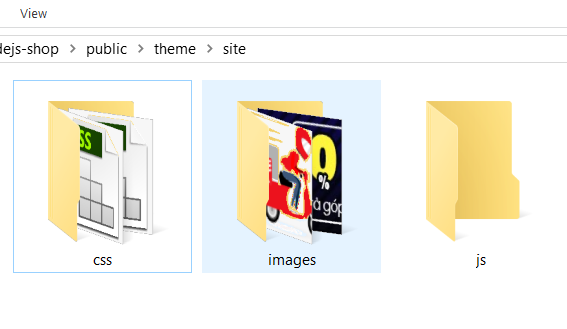
|  |
| --- |
| var siteModel = require('../../models/site');  async function home(req, res){    var productFeatured = await siteModel.productFeatured();  var productLatest = await siteModel.productLatest();    } |

B – TRAINING DỰ ÁN

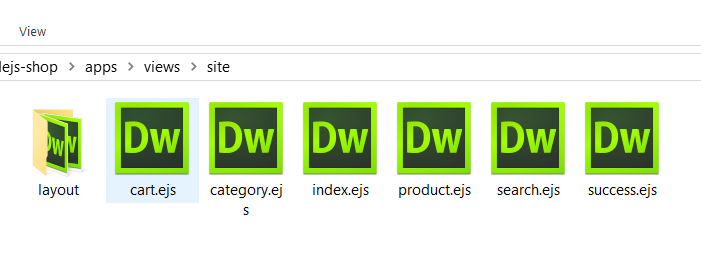
I – CÔNG VIỆC TRÊN LỚP

**1. Bóc tách giao diện Frontend (tương tự như làm với Backend)**

* Copy cấu trúc thư mục của Frontend HTML theo cấu trúc sau



* Copy các file giao diện của Frontend HTML theo cấu trúc sau



* Thư mục layout sẽ chưa các cấu trúc Header và Footer (giống phần Backend)
* Chuyển toàn bộ file HTML sang EJS
* Xóa tất cả Header & Footer của tất cả các file giao diện khi các code này đã được chuyển vào thư mục layout
* Chỉnh lại toàn bộ link frontend Images, CSS, Javascript trên giao diện theo Static URL

**2. Xây dựng tính năng hiển thị sản phẩm nổi bật & sản phẩm mới**

* Xây dựng logic model để lọc ra sản phẩm nổi bật
* Xây dựng logic model để lọc ra sản phẩm mới
* Sử dụng Async & Await bên Controller để lấy đồng thời các kết quả của sản phẩm nổi bật và sản phẩm mới bên Model
* Đẩy các kết quả lấy được về sản phẩm nổi bật và sản phẩm mới sang View
* Tại view sử dụng vòng lặp For… In và các thuật toán trình bầy dữ liệu đơn giản để hiển thị kết quả

II – CÔNG VIỆC VỀ NHÀ