**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

🙢 🕮 🙠



**BÁO CÁO ĐỒ ÁN**

**CÔNG NGHỆ THÔNG TIN**

Sinh viên thực hiện

Đỗ Vũ Hưng – 20110499

Định Quân – 20110043

Giáo viên hướng dẫn: Thầy Lê Vĩnh Thịnh

Mục Lục

[I. Đặc tả: 3](#_Toc121076057)

[a) Ngữ cảnh sử dụng phần mềm 3](#_Toc121076058)

[b) Dữ liệu, thông tin đầu vào 3](#_Toc121076059)

[c) Mục đích và chức năng 3](#_Toc121076060)

[d) Giao diện dự kiến 4](#_Toc121076061)

[II. Phân công công việc: 6](#_Toc121076062)

[III. Thiết kế: 6](#_Toc121076063)

[1. Cơ sở lý thuyết 6](#_Toc121076064)

[2. Thiết Kế Lớp 10](#_Toc121076065)

[3. Thiết kế cơ sở dữ liệu: 28](#_Toc121076066)

[4. Mô tả các thuộc tính 29](#_Toc121076067)

[5. Thiết Kế Giao Diện 32](#_Toc121076068)

[IV. Cài Đặt Và Kiểm Thử 35](#_Toc121076069)

[V. Kết luận: 36](#_Toc121076070)

[Tài liệu tham khảo 37](#_Toc121076071)

PHỤ LỤC HÌNH ẢNH

[1. Trang đăng Nhập 32](#_Toc121076372)

[2.Trang đăng ký 32](#_Toc121076373)

[3.Trang home của website 33](#_Toc121076374)

[4.Trang cửa hàng chung 33](#_Toc121076375)

[5.Trang thông tin chi tiết sản phẩm 33](#_Toc121076376)

[6.Trang giỏ hàng 34](#_Toc121076377)

[7.Trang checkout 34](#_Toc121076378)

[8.Trang thanh toán 34](#_Toc121076379)

NỘI DUNG

# Đặc tả:

## Ngữ cảnh sử dụng phần mềm

Do cuộc sống ngày nay quá bộn bề và đây lo toan và mệt mỏi nên mỗi người đều tìm một sở thích cho bản thân mình, người thì thích chơi thể thao, người thì thích nấu ăn, người thích sưu tầm giầy, đồng xu, đồ cổ. Ngoài ra mấy năm gần đây ở trong Việt Nam có xuât hiện một loại sở thích mới đó là sưu tầm hoặc custom lại bàn phím cơ nhưng do đây là bộ môn còn khá mới nên những linh kiện để custom hoặc nhưng chiếc bàn phím custom thường được mua lại thông qua facebook hoặc qua giới thiệu của những người chung đam mê với nhau.

Để đáp ưng nhu cầu nêu trên chúng em sẽ thiết kế một website bán bàn phím cơ custom và những sản phẩm khác liên quan tới việc custom bàn phím.

## Dữ liệu, thông tin đầu vào

Đối với dữ liệu và thông tin đầu vào của website sẽ được chia là hai phần:

* Phần user (khách hàng): thông tin về khách hàng như họ và tên, ngày tháng năm sinh, tên đăng nhập, mật khẩu, số điện thoại email, số tài khoản (khi thanh toán), …; ý kiến nhận xét, đánh giá, bình luận; tìm kiếm sản phẩm.
* Phần admin (người quản trị cửa hàng): thông tin về sản phẩm (thêm, xóa, cập nhật), bộ sưu tập (thêm, xóa, cập nhật), thông tin về admin, …

## Mục đích và chức năng

* Mục đích: website của chúng em là đem lại trải nghiệm mua hàng thuận tiện, hiệu quả nhất cho khách hàng. Mọi người có thể xem thông tin sản phẩm, tìm kiếm các sản phẩm cần mua, đặt hàng và thanh toán nhanh chóng. Và khi việc mua hàng trở nên nhanh chóng và dễ tiếp cận thì sẽ giúp nhiều người mới dễ tiếp cận với bộ môn custom phím cơ này hơn
* Các tính năng chính của website:
* Đăng nhập, đăng ký
* Đối với user:
* Xem chi tiết sản phẩm
* Tìm kiếm sản phẩm theo, giá, theo yêu thích, theo đơn hàng
* Cho sản phẩm vào giỏ hàng
* Thanh toán

## Giao diện dự kiến

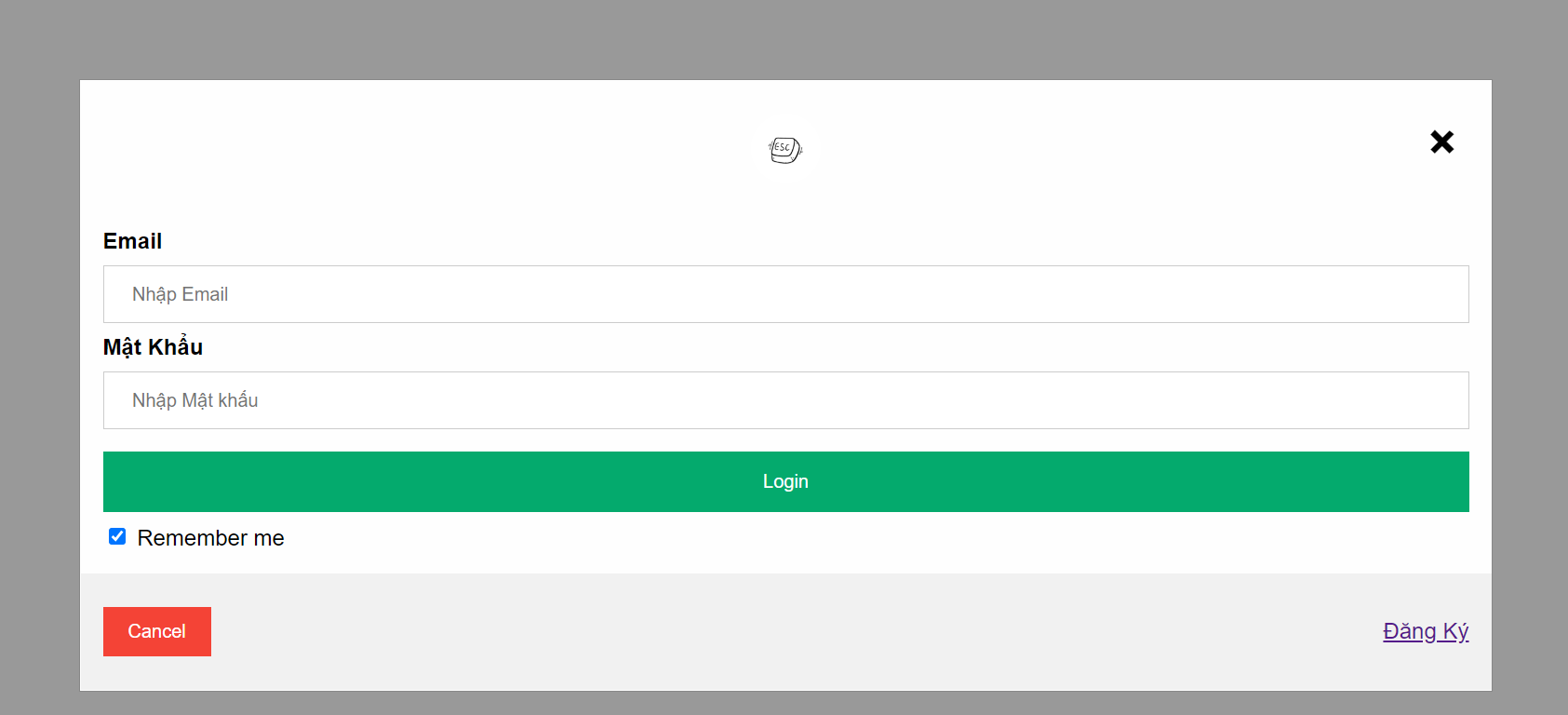


Figure 1: Trang đăng nhập

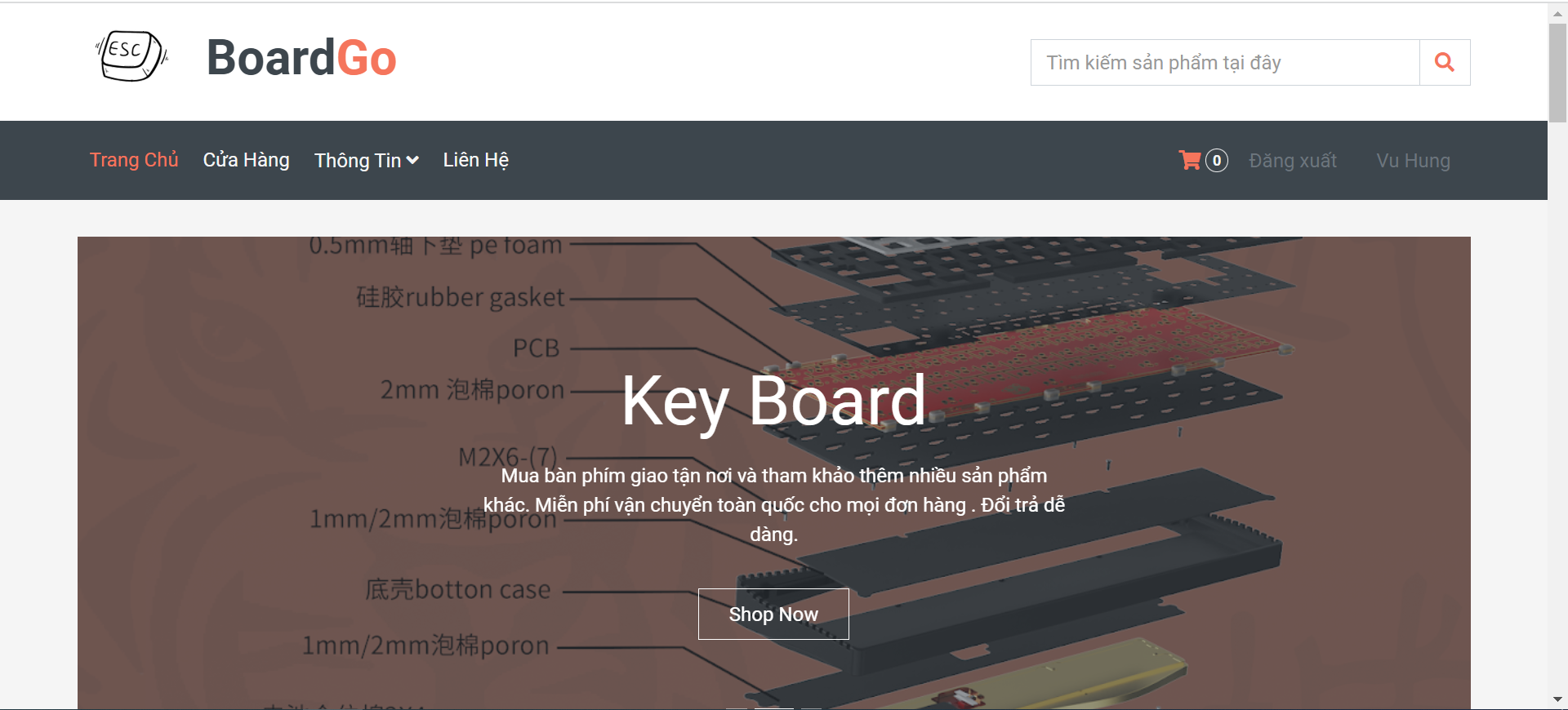


Figure 2: Trang chủ

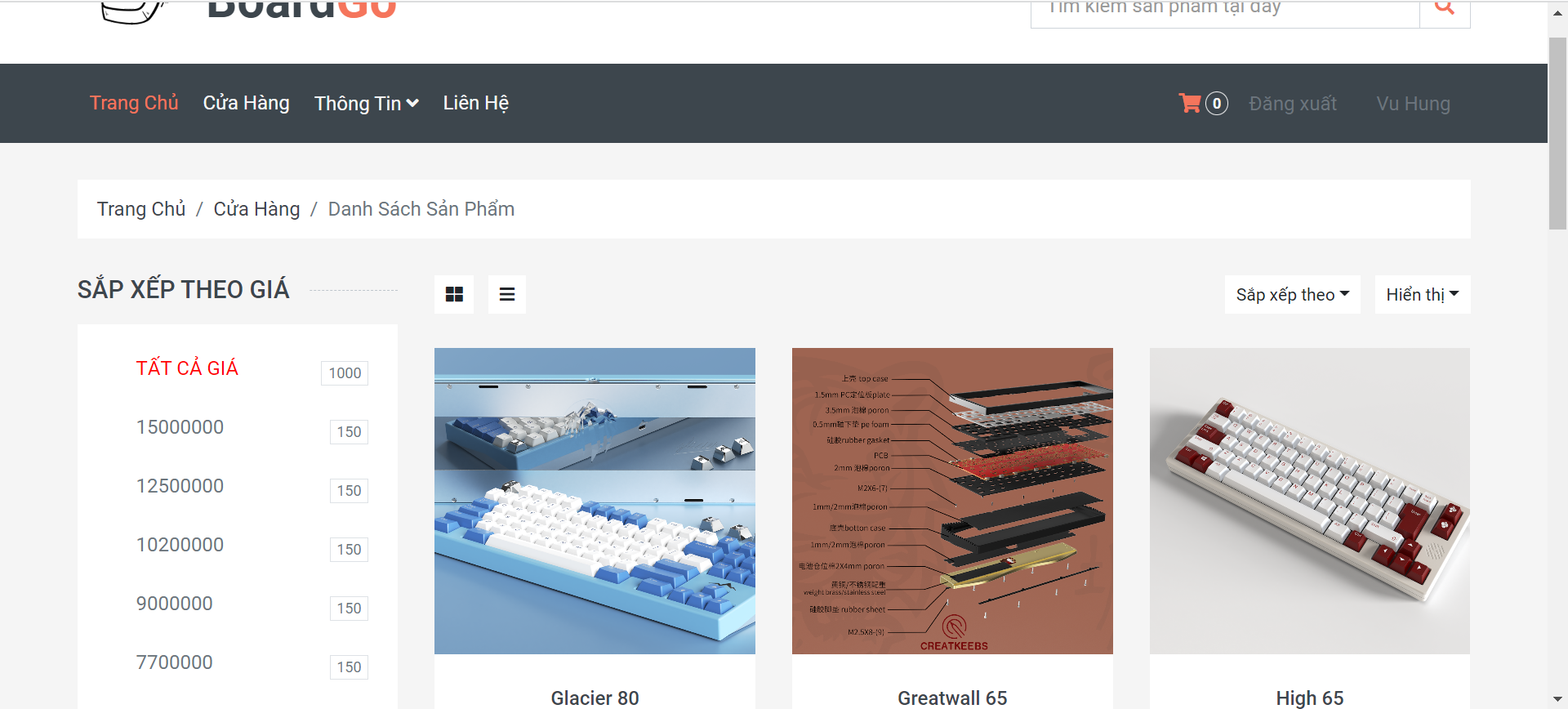


Figure 3: Cửa hàng chung

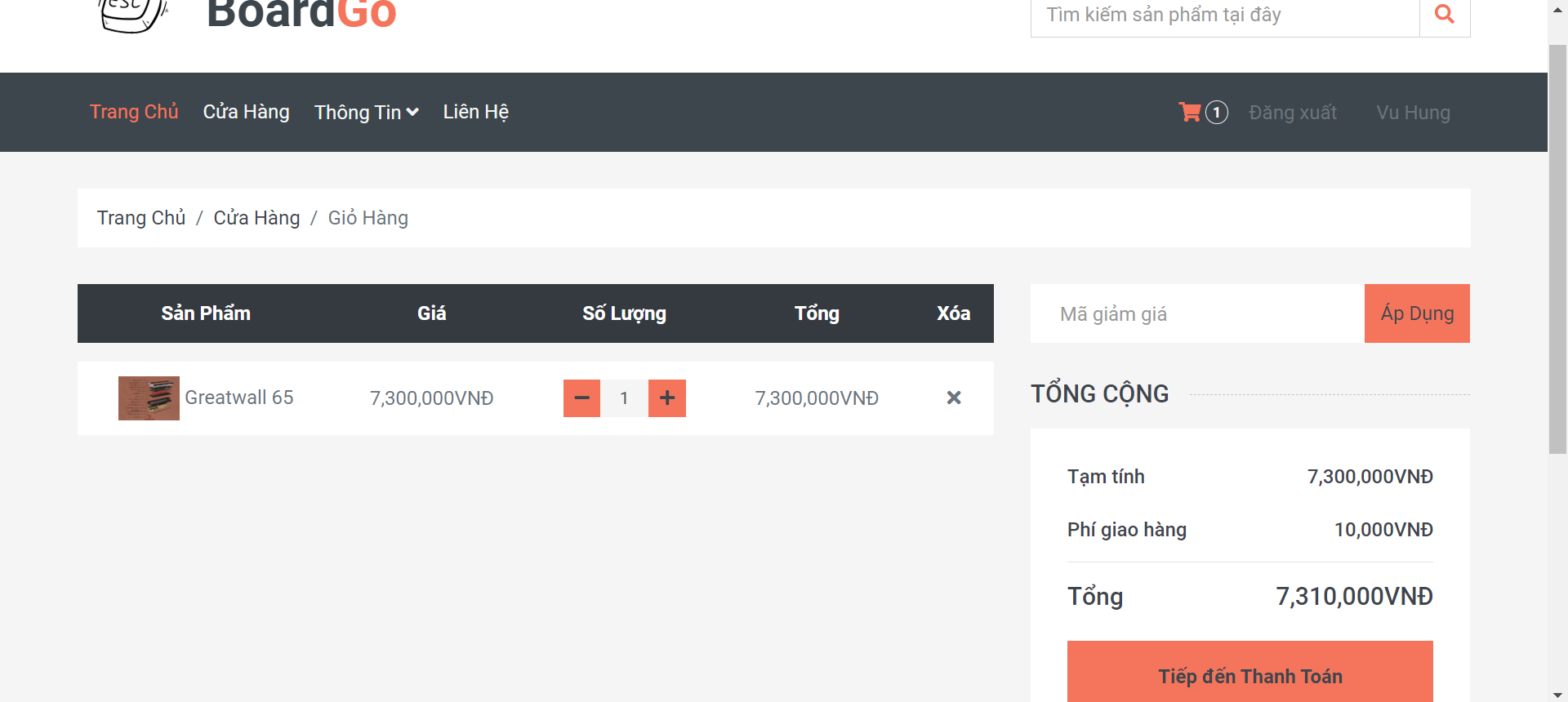


Figure 4: Trang Checkout

# Phân công công việc:

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Họ và tên | Công việc | Đánh giá |
| 1 | Đỗ Vũ Hưng | Đổ dữ liệu, phân trang, đăng nhập | 100% |
| 2 | Định Quân | Giỏ hàng, thanh toán, đăng ký | 100% |

# Thiết kế:

## Cơ sở lý thuyết

1. Node js

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Đây là một nền tảng (platform) phát triển độc lập dựa trên V8 JavaScript engine. Đây là một trình thông dịch thực thi mã JavaScript cho phép tạo các ứng dụng web như video clip và diễn đàn, đặc biệt có thể mở rộng nhanh chóng và dễ dàng cũng như thu hẹp phạm vi hoạt động của các trang mạng mạng xã hội.

Đây được xem là một lợi thế khi NodeJS có thể hoạt động trên nhiều nền tảng hệ điều hành khác nhau, từ Windows, Linux đến OS X. NodeJS cung cấp một thư viện phong phú dưới dạng các mô-đun Javascript khác nhau giúp đơn giản hóa việc lập trình và giảm thiểu thời gian cần thiết.

Ý tưởng chính của Node js là sử dụng non – blocking, nhanh chóng định tuyến đầu vào/ đầu ra dữ liệu thông qua các tác vụ thời gian thực. Bởi vì Node js có khả năng mở rộng nhanh chóng và có thể xử lý một số lượng lớn các kết nối đồng thời với thông lượng cao.

Hầu hết các ứng dụng web truyền thống sử dụng tài nguyên hệ thống không hiệu quả khi các yêu cầu tạo ra các luồng xử lý yêu cầu mới và tiêu tốn khá nhiều bộ nhớ hệ thống. Vì vậy giải pháp mà NodeJS đưa ra chính là sử dụng luồng đơn có tên tiếng anh là Single – Threaded, đây là sự kết hợp non – blocking I/ O để thực hiện các yêu cầu, điều này cho phép nó hỗ trợ hàng chục hàng nghìn kết nối đồng thời.

* Ưu điểm
* IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
* Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
* Chia sẻ cùng code ở cả phía client và server.
* NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
* Cộng đồng hỗ trợ tích cực.
* Cho phép stream các file có kích thước lớn
* Nhược điểm
* Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
* Khó thao tác với cơ sử dữ liệu quan hệ.
* Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
* Cần có kiến thức tốt về JavaScript.
* Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

1. ExpessJS

ExpressJS là một framework ứng dụng web có mã nguồn mở và miễn phí được xây dựng trên nền tảng Node.js. ExpressJS được sử dụng để thiết kế và phát triển các ứng dụng web một cách nhanh chóng. Để hiểu ExpressJS, người dùng chỉ cần phải biết JavaScript, do đó nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn đối với các lập trình viên và nhà phát triển đã thành thạo JavaScript trước đó.

Vì ExpressJS là một framework của Node.js nên hầu hết các mã đã được viết sẵn cho các lập trình viên làm việc. Bạn có thể tạo các ứng dụng web cho một trang, nhiều trang hoặc kết hợp lại bằng cách sử dụng ExpressJS. framework này khá nhẹ, giúp tổ chức các ứng dụng web ở phía máy chủ thành một kiến trúc MVC hoàn hảo hơn.

ExpressJS hỗ trợ nâng cao các chức năng của NodeJS. Nếu bạn không sử dụng ExpressJS, bạn phải thực hiện rất nhiều lập trình phức tạp để xây dựng một API hiệu quả. ExpressJS đã giúp cho việc lập trình trong NodeJS trở nên dễ dàng hơn rất nhiều.

1. ReactJS

ReactJS được hiểu nôm na là một thư viện mã nguồn mở chứa nhiều JavaScript và người tạo ra ReactJS chính là ông trùm với cái tên quen thuộc Facebook, được ra đời vào năm 2013. Mục đích của việc tạo ReactJS là tạo ra các ứng dụng web nhanh, hiệu quả và hấp dẫn với nỗ lực viết mã tối thiểu. Mục tiêu chính của ReactJS là bất kỳ trang web nào sử dụng ReactJS phải mượt mà, nhanh chóng, khả năng mở rộng cao và dễ thực hiện.

Nhìn chung, các tính năng và điểm mạnh của ReactJS thường đến từ việc tập trung vào các phần riêng lẻ. Do đó, khi làm việc với web thay vì toàn bộ ứng dụng của một trang web bằng ReactJS, các nhà phát triển tính năng có thể tách rời và chuyển đổi giao diện người dùng từ những cách phức tạp và biến nó thành những phần đơn giản hơn. Điều này có nghĩa là kết xuất dữ liệu không chỉ ở phía máy chủ. Thực hiện với các vị trí, nhưng bạn cũng có thể thực hiện tại vị trí khách hàng khi sử dụng ReactJS.

1. MySQL

MySQL là một hệ quản trị cơ sở dữ liệu phổ biến nhất thế giới, được các nhà phát triển ưa chuộng.

MySQL có tốc độ cao, ổn định, dễ sử dụng, hoạt động trên nhiều hệ điều hành khác nhau.

MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên internet.

MySQL hỗ trợ nhiều ngôn ngữ lập trình như PHP, Perl,…. trong việc phát triển website.

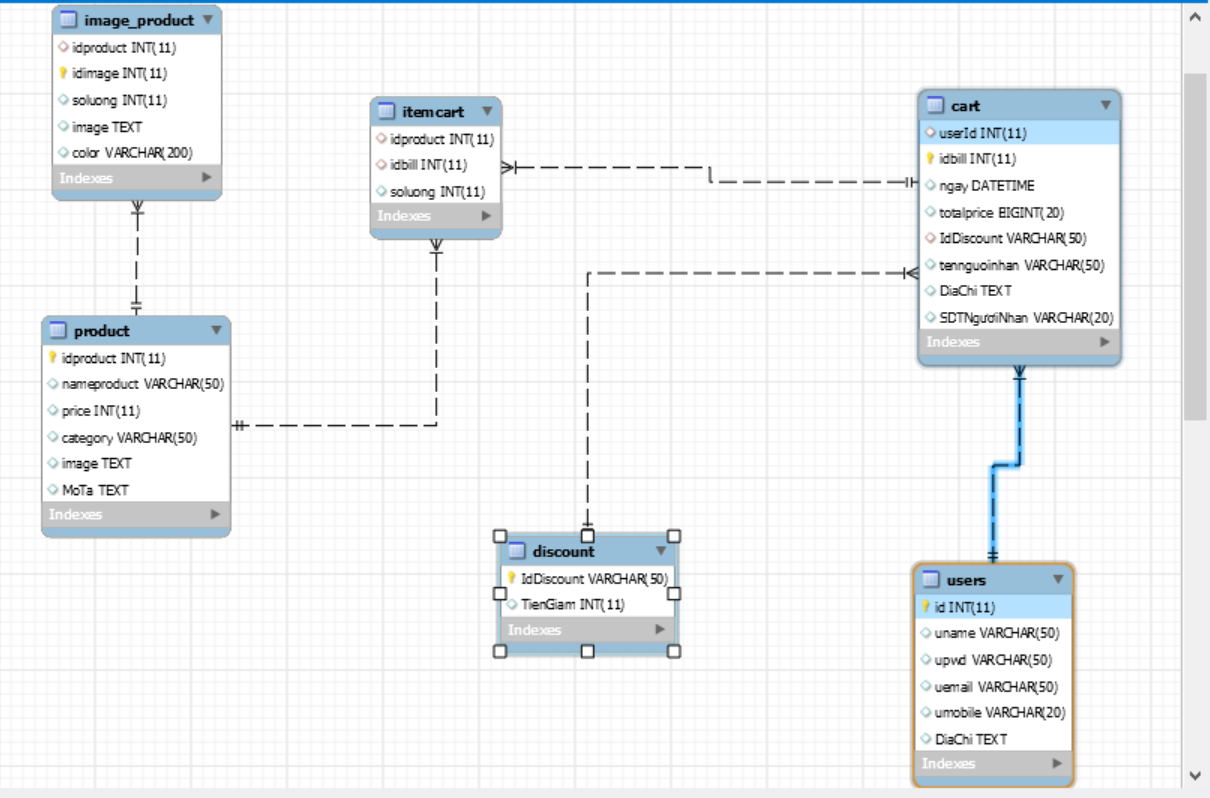
* Ưu điểm
* **Nhanh chóng**: Nhờ vào việc đưa ra một số những tiêu chuẩn và cho phép MySQL làm việc hiệu quả cũng như tiết kiệm chi phí, giúp gia tăng tốc độ thực thi.
* **Mạnh mẽ và khả năng mở rộng**: MySQL hoàn toàn có thể xử lý số lượng lớn dữ liệu và đặc biệt hơn thế nữa thì nó còn có thể mở rộng nếu như cần thiết.
* **Đa tính năng**: Ưu điểm MySQL là gì? MySQL hiện đang hỗ trợ nhiều những chức năng SQL rất được mong chờ từ 1 hệ quản trị CSDL quan hệ cả gián tiếp cũng như trực tiếp.
* **Độ bảo mật cao**: MySQL là gì? Hiện tại nó đang rất thích hợp cho những ứng dụng truy cập CSDL thông qua internet khi sở hữu rất nhiều những tính năng về bảo mật và thậm chí là đang ở cấp cao.
* Nhược điểm
* **Dung lượng hạn chế**: Trong trường hợp nếu như số lượng bản ghi của bạn đang lớn dần lên thì khi đó quá trình truy xuất dữ liệu sẽ diễn ra vô cùng khó khăn. Như vậy cần phải áp dụng rất nhiều những biện pháp khác nhau để có thể gia tăng được tốc độ truy xuất những dữ liệu ví dụ như tạo cache MySQL hoặc chia tải database ra nhiều server.
* **Độ tin cậy**: Nhược điểm MySQL là gì? Theo đó cách thức nhận chức năng cụ thể đang được xử lý cùng với MySQL (ví dụ như kiểm toán, những giao dịch, tài liệu tham khảo,…) khiến cho nó trở nên kém tin cậy hơn một số những hệ quản trị về cơ sở dữ liệu có quan hệ khác.
* **Giới hạn**: MySQL là gì? Theo thiết kế thì MySQL không có ý định thực hiện toàn bộ và nó đang đi kèm cùng với những hạn chế liên quan tới chức năng mà một số ứng dụng có thể cần tới. Vậy hiện tại điểm khác biệt giữa MySQL Server và MySQL là gì?

## Thiết Kế Lớp

CÁC PHƯƠNG THỨC

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên hàm | Mục đích | Người phụ trách |
| 1 | app.get('/', function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query("SELECT \* FROM heroku\_9921352427430fd.product LIMIT 8",(err,result)=>{  res.render('pages/index', {result:result});  })    }) | Khi nhận được đường dẫn là ‘/’ thì nó sẽ thực hiện hàm  function(req,res). Hàm này sẽ kết nối với database và lấy dữ liệu của các sản phẩm và đổ lên trang index.ejs thông qua tham số là result | Đỗ Vũ Hưng |
| 2 | app.get('/kit', function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })    con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query('SELECT \* FROM heroku\_9921352427430fd.product where category = "Kit" LIMIT 9',function(err,result){  res.render('pages/shop', {result:result});  })  }) | Khi nhận được đường dẫn là ‘/kit’ thì nó sẽ thực hiện hàm  function(req,res). Hàm này sẽ kết nối với database và lấy dữ liệu của các sản phẩm thuộc có loại là KIT và đổ lên trang shop.ejs thông qua tham số là result | Đỗ Vũ Hưng |
| 3 | app.get('/keycap', function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query('SELECT \* FROM heroku\_9921352427430fd.product where category = "Keycap" LIMIT 9',function(err,result){  res.render('pages/shop', {result:result});  })  }) | Khi nhận được đường dẫn là ‘/keycap’ thì nó sẽ thực hiện hàm  function(req,res). Hàm này sẽ kết nối với database và lấy dữ liệu của các sản phẩm thuộc có loại là KEYCAP và đổ lên trang shop.ejs thông qua tham số là result | Đỗ Vũ Hưng |
| 4 | app.get('/switch', function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query('SELECT \* FROM heroku\_9921352427430fd.product where category = "Switch" LIMIT 9',function(err,result){  res.render('pages/shop', {result:result});  })  }) | Khi nhận được đường dẫn là ‘/switch’ thì nó sẽ thực hiện hàm  function(req,res). Hàm này sẽ kết nối với database và lấy dữ liệu của các sản phẩm thuộc có loại là Switch và đổ lên trang shop.ejs thông qua tham số là result | Đỗ Vũ Hưng |
| 5 | app.get('/stab', function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query('SELECT \* FROM heroku\_9921352427430fd.product where category = "Stab" LIMIT 9',function(err,result){  res.render('pages/shop', {result:result});  })  }) | Khi nhận được đường dẫn là ‘/stab’ thì nó sẽ thực hiện hàm  function(req,res). Hàm này sẽ kết nối với database và lấy dữ liệu của các sản phẩm thuộc có loại là STAB và đổ lên trang shop.ejs thông qua tham số là result |  |
| 6 | app.get('/shop', function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })    con.query("SELECT \* FROM heroku\_9921352427430fd.product LIMIT 9",(err,result)=>{  res.render('pages/shop', {result:result});  })  }) | Khi nhận được đường dẫn là ‘/shop’ thì nó sẽ thực hiện hàm  function(req,res). Hàm này sẽ kết nối với database và lấy dữ liệu của các sản phẩm dưới database và đổ lên trang shop.ejs thông qua tham số là result |  |
| 7 | app.post('/search', function(req,res){    req.session.nameproduct = req.body.nameproduct  console.log(req.session.nameproduct)  res.redirect('/result')  }) | Khi lấy được đường link là ‘/search’ thì nó sẽ thực hiện hàm function(req,res) để lưu nameproduct(là tên của sản phẩm mà chúng ta muốn tìm) vào session.nameproduct và sau đó chuyển tới đường link ‘/result’. | Đỗ Vũ Hưng |
| 8 | app.get('/result',function(req,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query(`SELECT \* FROM heroku\_9921352427430fd.product where nameproduct LIKE '%${req.session.nameproduct}%'`,(err,result)=>{  console.log(result)  res.render('pages/shop', {result:result});  })  }) | Khi nhận được link ‘/result’ thì nó sẽ thực hiện hàm function(req,res), đổ dữ liệu lên trang shop qua biến result để hiện kết quả mà chúng ta tìm kiếm. | Đỗ Vũ Hưng |
| 9 | app.get('/:idproduct', function(req,res){  detailProduct(req.params.idproduct,res)  })  function detailProduct(id,res){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query(`SELECT \* FROM heroku\_9921352427430fd.product where idproduct = ${id}`,function(err,result){  console.log(result)  res.render('pages/detail', {result:result});  })  } | Khi nhận được link ‘/:idproduct’ chính là id của sản phẩm thì nó sẽ lấy id đó và chuyền vào hàm detailProduct. Sau đó kết nối và đổ dữ liệu lên trang detail qua biến result qua đó chúng ta có thể xem thông tin chi tiết của sản phẩm. | Đỗ Vũ Hưng |
| 10 | app.get("/login", function(req,res){  res.render('pages/login');  }) | Khi nhận được link ‘/login’ thì website sẽ đưa người dùng đến trang đăng nhập. | Đỗ Vũ Hưng |
| 11 | app.post("/login", function(req,res){  let email = req.body.uemail;  let password = req.body.psw;  console.log(email);  console.log(password);  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"    })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  con.query(`SELECT \* FROM heroku\_9921352427430fd.users where uemail = '${email}' and upwd='${password}'`,function(err,result){  console.log(result);  if(result!=null){  for(let i=0;i<result.length;i++){  req.session.user\_id = result[i].id;  req.session.user\_name = result[i].uname;  }  res.redirect("/");// res.render('pages/index', {result:result});  }  console.log(req.session.user\_id);  console.log(req.session.uesr\_name);  })    }) | Khi nhận được đường dẫn ‘/login’ nhưng mà theo kiểu post thì nó sẽ lấy thông tin tài khoản đăng nhập của người dùng để kiểm tra ở phía dưới database, nếu người dùng này đã đăng ký nick rồi thì nó sẽ đưa người dùng vào trang chủ của website. | Đỗ Vũ Hưng |
| 12 | app.get("/register", function(req,res){  res.render('pages/registration');  }) | Khi nhận được link ‘/resgister’ thì website sẽ đưa người dùng đến trang đăng ký. | Định Quân |
| 13 | app.post("/register", function(req,res){  var email = req.body.email  var password = req.body.psw;  var re\_password = req.body.psw\_repeat;  var name = req.body.name;  if (password ==re\_password){  var con = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"    })  con.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })      con.query(`INSERT INTO heroku\_9921352427430fd.users ( uname, upwd, uemail) VALUES ( '${name}', '${password}', '${email}');`,function(err,result){    res.redirect("/login");// res.render('pages/index', {result:result});  })  }  }) | Khi nhận được đường dẫn ‘/register nhưng mà theo kiểu post thì nó sẽ lấy thông tin người dung nhập vào như tên gọi, email. Mật khẩu để đẩy xuống database để tạo tài khoản đăng nhập cho người dùng | Định Quân |
| 14 | app.post('/addToCart', function(req,res){  var id = req.body.id;  var name = req.body.name;  var price = req.body.price;  var quantity = req.body.quantity;  var image = req.body.image;  var product = {id:id, name:name, price:price, image:image, quantity:quantity}  if(req.session.cart){  var cart = req.session.cart  if(!isProductInCart(cart,id)){  cart.push(product);  console.log("Connect successfully1");  }  }  else{  req.session.cart = [product];  var cart = req.session.cart;  console.log("Connect successfully3");  }  //calculateTotal  calculateTotal(cart,req)  res.redirect('/cart')  }); | Sau khi người dụng chọn món hàng mà họ muốn mua nếu chưa có món hàng nào trong giỏ hàng thì sẽ tạo giỏ hàng mới đồng thời dẫn đến phường thức cart | Định Quân |
| 16 | app.get('/cart', function(req,res){  var name = req.session.user\_name;    if(req.session.user\_id){  var cart = req.session.cart  var total = req.session.total  //return cart page  res.render('pages/cart',{cart:cart, total:total,session:req.session,name:name});  }  else  {  res.render('pages/login');  } | Sẽ dẫn người dung đến trang cart đồng thời hiện các sp họ đã chọn và tổng số tiền cần thanh toán nhưng nếu người dùng chưa đăng nhập thì sẽ dẫn đến trang đăng ký thay vì trang cart | Định Quân |
| 17 | app.post('/removeProduct', function(req,res){  var id = req.body.id;  var cart = req.session.cart;  console.log(cart);  for(let i=0;i<cart.length;i++){  cart.splice(cart.indexOf(i),1);  }  calculateTotal(cart,req);  res.redirect('/cart');  }); | Loại bỏ món hàng mà khách hàng không muốn mua nữa ra khỏi giỏ hàng | Định Quân |
| 18 | app.post('/edit\_product\_quantity', function(req,res){  var id =req.body.id;  var quantity=req.body.quantity;  var increase\_btn=req.body.increase\_product\_quantity;  var decrease\_btn=req.body.decrease\_product\_quantity;  console.log(decrease\_btn);  var cart = req.session.cart;  if (increase\_btn){  for(let i=0;i<cart.length; i++)  {  if (cart[i].id==id)  {  if(cart[i].quantity>0)  {  cart[i].quantity=parseInt(cart[i].quantity)+1;  }  }  }  }  if (decrease\_btn){  for(let i=0;i<cart.length; i++)  {  if (cart[i].id==id)  {  if(cart[i].quantity>1)  {  cart[i].quantity=parseInt(cart[i].quantity)-1;  }  }  }  }  calculateTotal(cart,req);  res.redirect('/cart');  }); | Cập nhật lại số lượng món hàng của những món hàng khi khách hàng muốn mua thêm hay giảm bớt đi sau đó cập nhật lại trên trang cart | Định Quân |
| 19 | app.get('/checkout',function(req,res){  var name = req.session.user\_name;  var cart = req.session.cart;  var total = req.session.total;  var connect = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  connect.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  connect.query(`SELECT \* FROM heroku\_9921352427430fd.users where id=${req.session.user\_id}`,(err,result)=>{  res.render('pages/checkout',{result:result,cart:cart,total:total, session:req.session,name:name});  console.log(result);  })    }); | Hiện các thông tin như danh sách các món hàng khách hàng đã chọn, số tiền từng loại, số lượng từng loại, tổng giá tiền tạm tính và tổng số tiền phải trả, đồng thời hiện tên, email, mà người dùng đã cung cấp khi tạo tài khảon lên trên trang checkout | Định Quân |
| 20 | app.post('/place\_orders',function(req,res){  var name = req.body.name;  var email = req.body.email;  var mobile\_no=req.body.mobile\_no;  var DiaChi=req.body.DiaChi;  var cost =req.session.total;  var MyDate = new Date();  idbill=Math.floor(Math.random() \* 93876);  var cart= req.session.cart;  console.log( cart);  var connect = mysql.createConnection({  host:"us-cdbr-east-06.cleardb.net",  user:"b060c3fa72ae89",  password:"0e6af735",  database:"heroku\_9921352427430fd"  })  connect.connect(function(error){  if(error){  throw error;  }  else{  console.log("Connect successfully")  }  })  connect.query(`INSERT INTO heroku\_9921352427430fd.cart (userId, idbill, ngay, totalprice, tennguoinhan , DiaChi ,SDTNgươiNhan) VALUES ('${req.session.user\_id}', '${idbill}', '${MyDate.toMysqlFormat()}', '${cost}', '${name}', '${DiaChi}', '${mobile\_no}')`,(err,result)=>{  for(let i =0; i<cart.length; i++){  connect.query(`INSERT INTO heroku\_9921352427430fd.itemcart (idbill, idproduct ,soluong ) VALUES ('${idbill}','${cart[i].id}','${cart[i].quantity}')`,(err,result)=> {console.log(result)});  }  res.redirect('/payment');  })  }); | Lấy các thông tin như địa chỉ giao hàng, số điện thoải để nhận hàng, email, tên người nhận, danh sách sản phẩm mua hàng, tổng số tiền và lưu xuống database đồng thời dẫn tới trang payment | Định Quân |
| 21 | app.get('/payment',function(req,res){  var name = req.session.user\_name;  cost=req.session.total;  cost=cost/24000;  console.log(cost);  res.render('pages/payment',{cost:cost, session:req.session,name:name});  }); | Khi nhận được link ‘/payment’ thì website sẽ đưa người dùng đến trang payment. Ở đây người dùng lựa chọn phương thức thanh toán | Định Quân |
| 22 | <script src="https://www.paypal.com/sdk/js?client-id=test&currency=USD"></script>  <!-- Set up a container element for the button -->  <div id="paypal-button-container"></div>  <script>  paypal.Buttons({  // Sets up the transaction when a payment button is clicked  createOrder: (data, actions) => {  return actions.order.create({  purchase\_units: [{  amount: {  value: '<%= cost %>' // Can also reference a variable or function  }  }]  });  },  // Finalize the transaction after payer approval  onApprove: (data, actions) => {  return actions.order.capture().then(function(orderData) {  // Successful capture! For dev/demo purposes:  console.log('Capture result', orderData, JSON.stringify(orderData, null, 2));  const transaction = orderData.purchase\_units[0].payments.captures[0];  alert(`Transaction ${transaction.status}: ${transaction.id}\n\nSee console for all available details`);  });  }  }).render('#paypal-button-container');  </script> | Khi người dùng chọn phương thức thanh toán xong thì sẽ gọi PayPal Orders API để bắt đầu thực hiện giao dịch | Định Quân |

## Thiết kế cơ sở dữ liệu:



Sơ đồ quan hệ

MÔ TẢ CÁC BẢNG

|  |  |  |
| --- | --- | --- |
| TT | Tên bảng | Mục đích |
| 1 | users | Lưu thông tin cá nhân của người dùng. |
| 2 | image\_product | Lưu hình ảnh của các sản phẩm. |
| 3 | product | Lưu thông tin về sản phẩm. |
| 4 | cart | Lưu thông tin chi tiết về đơn hàng của người mua để có thông tin và vận chuyển cho khách hàng. |
| 5 | itemcart | Lưu những sản phẩm mà khách hàng đã mua trước đây |
| 6 | discount | Lưu những mã giảm giá |

## Mô tả các thuộc tính

CÁC THUỘC TÍNH TRONG TABLE USERS

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | id | INT(11) | Lưu id để phân biệt với các người dùng khác |
| 2 | uname | VARCHAR(50) | Lưu tên của người dùng |
| 3 | upwd | VARCHAR(50) | Lưu mật khẩu của người dùng |
| 4 | uemail | VARCHAR(50) | Lưu email của người dùng |
| 5 | umobile | VARCHAR(20) | Lưu số điện thoại của người dùng |
| 6 | DiaChi | TEXT | Lưu địa chỉ của người dùng |

CÁC THUỘC TÍNH TRONG TABLE PRODUCT

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | idproduct | INT(11) | Lưu id của sản phẩm để phân biệt với các sản phẩm khác |
| 2 | nameproduct | VARCHAR(50) | Lưu tên của sản phẩm |
| 3 | price | INT(11) | Lưu giá tiền của sản phẩm |
| 4 | category | VARCHAR(50) | Lưu loại của sản phẩm |
| 5 | image | TEXT | Lưu hình ảnh đại diện của sản phẩm |
| 6 | MoTa | TEXT | Lưu những thông tin cơ bản về nguồn gốc của sản phẩm |

CÁC THUỘC TÍNH TRONG TABLE IMAGE\_PRODUCT

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | idproduct | INT(11) | Lưu id của sản phẩm để phân biệt với các sản phẩm khác |
| 2 | idimage | INT(11) | Lưu id của các hình ảnh của sản phẩm để phân biệt với các hình ảnh khác |
| 3 | soluong | INT(11) | Lưu số lượng hiện đang có của sản phẩm trong kho |
| 4 | image | TEXT | Lưu đường link của các hình ảnh |
| 5 | color | VARCHAR(20) | Lưu màu của sản phẩm |

CÁC THUỘC TÍNH TRONG TABLE CART

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | userId | INT(11) | Lưu id của người dùng để biết được đơn hàng này là của người dùng nào |
| 2 | idbill | INT(11) | Lưu id đơn hàng của khách hàng để phân biệt với các đơn hàng khác |
| 3 | ngay | DATETIME | Lưu ngày giờ đặt hàng của khách hàng |
| 4 | totalprice | BIGINT(20) | Lưu tổng giá tiền của đơn hàng |
| 5 | IdDiscount | VARCHAR(50) | Lưu mã giảm giá của đơn hàng để biết được là giảm bao nhiêu |
| 6 | tennguoinhan | VARCHAR(50) | Lưu tên của người nhận |
| 7 | DiaChi | TEXT | Lưu địa chỉ của người nhận để giao hàng |
| 8 | SDTNguoiNhan | VARCHAR(20) | Lưu số điện thoại của người nhận |

CÁC THUỘC TÍNH TRONG BẢNG ITEMCART

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | idproduct | INT(11) | Lưu id sản phẩm để phân biệt với các sản phẩm khác |
| 2 | idbill | INT(11) | Lưu id đơn hàng mà đã mua sản phẩm này |
| 3 | soluong | INT(11) | Lưu số lượng mua sản phẩm này của đơn hàng |

CÁC THUỘC TÍNH TRONG BẢNG DISCOUNT

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | idDiscount | VARCHAR(50) | Lưu mã giảm giá |
| 2 | TienGiam | INT(11) | Lưu số tiền giảm của mã giảm giá này |

## Thiết Kế Giao Diện

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Trang đăng nhập:    1. Trang đăng Nhập | Người dùng sẽ đăng nhập vào website qua trang này. | Đỗ Vũ Hưng  Thiết kế theo như vậy thì người dùng có thể dễ dàng đăng ký tài khoản khi mà chưa có tài khoản |
| 2 | Trang đăng ký:    2.Trang đăng ký | Người dùng sẽ đăng ký tài khoản ở trang này. | Định Quân  Khi người dùng đang ký thành công có thể nhấn vào chữ ‘ Đăng nhập’ nó sẽ chuyển qua trang login cho người dùng đăng nhập. |
| 3 | Trang home của website:    3.Trang home của website | Người dùng sẽ thấy được các sản phẩm mà cửa hàng đang bán ở đây | Đỗ Vũ Hưng  Trong trang này sẽ có nơi phân biệt các sản phẩm mà người dùng muốn mua nếu người dùng muốn mua loại nào thì sẽ chỉ hiện ra các loại đó cho người dùng |
| 4 | Trang cửa hàng chung của website:    4.Trang cửa hàng chung | Người dùng sẽ thấy được tất cả các sản phẩm mà cửa hàng đang bán tại đây | Đỗ Vũ Hưng:  Người dùng sẽ thấy được toàn bộ sản phẩm ở đây |
| 5 | Trang thông tin chi tiết của sản phẩm:    5.Trang thông tin chi tiết sản phẩm | Người dùng sẽ thấy được thông tin chi tiết của sản phẩm ở đây | Đỗ Vũ Hưng:  Người dùng có thể xem thông tin nguồn gốc của sản phẩm ở đây và nếu muốn có thể nhấn Thêm vào giỏ hàng |
| 6 | Trang giỏ hàng:    6.Trang giỏ hàng | Người dùng có thể theo dõi giỏ hàng của mình tại đây | Định Quân:  Người dùng có thể thay đổi số lượng bằng cách nhấn dấu + hoăc - |
| 7 | Trang checkout:    7.Trang checkout | Người dùng có thể xem lại số lượng, tổng tiền trước khi order | Định Quân:  Người dùng kiểm tra thông tin có thể nhấn Place Order để thanh toán. |
| 9 | Chọn phương thức thanh toán    8.Trang thanh toán | Người dùng chọn phương thức thanh toán | Định Quân:  Người dùng có thể chọn phương thức thanh toán giữa paypal hoặc thẻ tín dụng |

# Cài Đặt Và Kiểm Thử

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Tình huống 1:  Dữ liệu vào: các sản phẩm  Kết quả dự kiến: các sản phẩm sẽ xuất hiện đổ lên khi vô trang chủ của website | Kiểm tra dữ liệu có được đổ lên hay không | Xem thử có kết nối được với database và lấy được dữ liệu hay không |
| 2 | Tình huống 2:  Dữ liệu vào: các sản phẩm theo loại được chọn  Kết quả dự kiến: sẽ hiện ra các sản phẩm theo loại mà người dùng đã chọn | Kiểm tra dữ liệu đổ lên có đúng hay không | Xem thử có kết nối và đổ dữ liệu lên có đúng theo với yêu cầu của người dùng hay không |
| 3 | Tính huống 3:  Dữ liệu vào: các sản phẩm khách hàng muốn mua  Kết quả dự kiến: các sản phẩm khách hàng muốn mua sẽ hiện ở trong giỏ hàng | Kiểm tra xem sản phẩm có được lưu trữ trong giỏ hàng không | Kiểm tra thử có đúng sản phẩm khách hàng muốn mua có ở trong giỏ hàng không , nếu như khách hàng chọn một sản phẩm trùng với sản phẩm đã có trong giỏ hàng thì số lượng của sản phẩm đó sẽ tăng lên chứ không thêm vào một cái mới. Nếu khách hàng giảm sản phẩm đó xuống 0 thì nó sẽ bị xóa luôn |

# Kết luận:

* Mức độ hoàn thành: Chúng em đánh giá là đã hoàn thành yêu cầu cơ bản của thầy, tuy nhiên vẫn còn một vài thiếu sót.
* Khó khăn: Node js là một ngôn ngữ khá mới nên và lạ với chúng em, đồng thời vì cũng chỉ mới học lập trình web trong kỳ học này nên để sử dụng node js làm đồ án này thì chúng em đã gặp một vài khó khăn về cấu trúc tổ chức, phương thức,…
* Khắc phục: Tham khảo các tài liệu được cung cấp và trên Internet, đồng thời hỏi các bạn cùng khóa và các anh khóa trên.
* Ưu điểm: có khả năng thanh toán bằng Paypal, giao diện thân thiện với người dùng
* Nhược điểm: website chưa có trang admin để quản lý trực tiếp

# Tài liệu tham khảo

https://aptech.fpt.edu.vn/nodejs-la-gi.html

https://wiki.tino.org/expressjs-la-gi

https://aptech.fpt.edu.vn/reactjs-la-gi.htmh

https://fptcloud.com/mysql-la-gi/