|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Tên trang** | **Chức năng** | **Hạn chốt** |
| 1 | Trang đăng nhập, đăng kí | ĐĂNG KÍ   * Phía FrontEnd sẽ kiểm tra tính hợp lệ (validation) của thông tin nhập vào trước khi gửi đến phía BackEnd. Sau đó sẽ gửi yêu cầu đến phía BackEnd thông qua các API. Xử lí phản hồi từ BackEnd và hiển thị thông báo cho người dùng. * Phía BackEnd sau khi tiếp nhận yêu cầu từ phía FrontEnd sẽ kiểm tra tính hợp lệ của thông tin đăng ký, đảm bảo rằng email chưa được đăng kí trước đó và role mặc định là Khách hàng.   SELECT \* FROM User  WHERE email = ‘email đăng kí’;  nếu hợp lệ sẽ mã hóa mật khẩu và lưu thông tin người dùng vào bảng User bên cơ sở dữ liệu.  INSERT INTO User (UserName, Email, Password, Phone, Age, Role)  VALUES ('value1', 'value2', 'value3', 'value4', 'value5, 'value6', 'Khách hàng');  tạo mã thông báo xác nhận gửi đến email để xác nhận đăng kí. Gửi phản hồi về trạng thái đăng kí. |  |
| 2 | ĐĂNG NHẬP   * Phía FrontEnd sẽ gửi yêu cầu đăng nhập đến BackEnd, phía BackEnd sẽ xác thực thông tin đăng nhập (email, mật khẩu). Nếu đúng sẽ tạo ra phiên làm việc cho người dùng để duy trì trạng thái đăng nhập, chuyển hướng về trang chủ. * Phía BackEnd sau khi tiếp nhận yêu cầu từ phía FrontEnd sẽ giải mã mật khẩu và so sánh thông tin đăng nhập với dữ liệu trong bảng User bên cơ sở dữ liệu.   SELECT \* FROM User  WHERE email = ‘email đăng nhập’;  Nếu thông tin đúng, tạo một token truy cập và gửi về cho Frontend. Lưu thông tin về trạng thái đăng nhập và gửi phản hồi về trạng thái đăng nhập thành công và token truy cập. |  |
| 3 | Trang chủ | Lấy danh sách sản phẩm.  SELECT \* FROM Product; |  |
| 4 | Tìm kiếm sản phẩm theo tên.   * Nhập tên sản phẩm cần tìm vào ô tìm kiếm, bấm Tìm kiếm sẽ gửi yêu cầu về BackEnd. * BackEnd sẽ sử dụng câu truy vấn để tìm kiếm sản phẩm dựa trên ProductName trong bảng Product.   SELECT \* FROM Product  WHERE ProductName = ‘từ khóa ô tìm kiếm’;  Phản hồi kết quả về phía FrontEnd |  |
| 5 | Tang chi tiết | Lấy thông tin chi tiết sản phẩm dựa trên ProductID  SELECT \* FROM Product  WHERE ProductID = ‘id sản phẩm’; |  |
| 6 | Thêm sản phẩm vào giỏ hàng khi bấm “Mua ngay”   * Gán sự kiện onclick gọi hàm themGH(${product.ProductID}), hàm truyền vào ID cuả sản phẩm, sau đó sử dụng chức năng lấy thông tin chi tiết sản phẩm để render vào giỏ hàng |  |
| 7 | Lấy danh sách bình luận  SELECT \* FROM Comment |  |
| 8 | Thêm bình luận.   * FrontEnd có giao diện để người dùng nhập nội dung bình luận và nhấn nút để gửi. Khi người dùng nhấn nút gửi, FrontEnd sẽ gửi yêu cầu đến BackEnd. * BackEnd sẽ giải mã chuỗi token khi đăng nhập để lấy UserID, ProductID lấy từ params, sau đó sẽ kiểm tra tính hợp lệ của nội dung bình luận.   INSERT INTO Comment (UserID, ProductID, ngay\_binh\_luan, noi\_dung)  VALUES (UserID\_value, ProductID\_value, ngay\_binh\_luan\_value, noi\_dung\_value);  Nếu hợp lệ sẽ lưu thông tin vào bảng Comment trong cơ sở dữ liệu và phản hồi kết quả về phía FrontEnd để render. |  |
| 9 | Xóa bình luận khi bấm vào nút X.   * Gán sự kiện onclick gọi hàm xoaBL(${comment.CommentID}), hàm truyền vào ID cuả bình luận được chọn, sau đó xóa bình luận theo Id của bình luận đó DELETE FROM Comment WHERE CommentID = comment.CommentID; |  |
| 10 | Lấy danh sách bình luận theo ProductID  SELECT \* FROM Comment  WHERE ProductID = 'id lấy từ params'; |  |
| 11 | Trang quản trị | Lấy danh sách sản phẩm như trang chủ, sau đó render ra theo dạng table. |  |
| 12 | Tìm kiếm sản phẩm theo tên.   * DOM tới ô input và gán event keyup và hàm handleSearch * Gọi hàm searchUser(keyword, data) với keyword là value của ô input tìm kiếm và data là danh sách sản phẩm. * Gọi 1 mảng rỗng có tên là searchResult * Tạo 1 vòng lặp danh sách sản phẩm(data) và kiểm tra xem tên của từng sản phẩm có bao gồm keyword không, nếu là true sẽ push vào mảng searchResult.   let product = data[i];  if(product.ProductName.toLowerCase().include(keyword.toLowerCase)){ searchResult.push(product);  }   * Gọi hàm handleSearch để render kêt quả. Nếu có kết quả tìm kiếm, tiến hành lọc dữ liệu và hiển thị kết quả tìm kiếm. Nếu không có từ khóa tìm kiếm, hiển thị toàn bộ dữ liệu |  |
| 13 | Thêm 1 sản phẩm.   * Khi bấm vào nút thêm sản phẩm bên FrontEnd sẽ hiện ra 1 modal để nhập thông tin về sản phẩm cần thêm, sau khi nhập xong FrontEnd sẽ kiểm tra tính hợp lệ (validation) của thông tin nhập vào. Nếu hợp lệ sẽ gửi yêu cầu đến phía BackEnd để xử lí. * BackEnd sẽ kiểm tra tính hợp lệ của thông tin (có đầy đủ thông tin các thuộc tính không, có trùng tên không), nếu thành công sẽ lưu thông tin vào bảng Product trong cơ sở dữ liệu.   INSERT INTO Product (ProductName, Price, Description, StockQuantity, CategoryID)  VALUES (‘thông tin nhập trong input bên FrontEnd’);  Sau đó sẽ phản hồi về phía FrontEnd kết quả, phía FrontEnd sẽ render ra sản phẩm mới được thêm vào. |  |
| 14 | Lấy danh sách loại sản phẩm  SELECT \* FROM Categories |  |
| 15 | Cập nhật sản phẩm.   * FrontEnd có giao diện giống với chức năng thêm sản phẩm nhưng sẽ có sẵn thông tin của sản phẩm cần cập nhật (sử dụng chức năng lấy thông tin sản phẩm và gán thông tin vào những ô input). Sau đó DOM tới các thẻ input lấy value và tiến hành cập nhật thông tin sản phẩm. Nhấn nút cập nhật sẽ gửi yêu cầu qua BackEnd. * BackEnd sẽ lấy ProductID để tìm ra sản phẩm và cập nhật thông tin mới vào sản phẩm có id là ProductID. Sau đó sẽ gửi phản hồi kết quả về phía FrontEnd. FrontEnd sẽ tiến hành render lại bảng để cập nhật thông tin mới của sản phẩm.   Cập nhật thông tin cho sản phẩm đó  UPDATE Product  SET ProductName = ‘value1’, Price = ‘value2’, Description = ‘value3’, StockQuantity = ‘value4’, CategoryID = ‘value5’  WHERE ProductID = ‘id của sản phẩm’; |  |
| 16 | Xóa sản phẩm   * Gán sự kiện onclick gọi hàm xoaSP(${product.ProductID}), hàm truyền vào ID cuả sản phẩm được chọn, sau đó xóa sản phẩm theo Id của sản phẩm đó DELETE FROM Product WHERE ProductID= product.ProductID; |  |
| 17 | Thêm 1 người dùng   * Khi bấm vào thêm người dùng bên FrontEnd sẽ hiện ra 1 modal tương tự với chức năng đăng kí để nhập thông tin về sản phẩm cần thêm, khác với đăng kí sẽ có thêm thẻ select roles để chọn loại người dùng (chỉ có admin mới có thể thêm 1 tài khoản cho admin sử dụng). Kiểm tra validation của thông tin nhập vào và gửi yêu cầu đến phía BackEnd. * Phía BackEnd sau khi tiếp nhận yêu cầu từ phía FrontEnd sẽ kiểm tra tính hợp lệ của thông tin đăng ký, đảm bảo rằng email chưa được đăng kí trước đó.   SELECT \* FROM User  WHERE email = ‘email đăng kí’;  nếu hợp lệ sẽ mã hóa mật khẩu và lưu thông tin người dùng vào bảng User bên cơ sở dữ liệu.  INSERT INTO User (UserName, Email, Password, Phone, Age, Role)  VALUES ('value1', 'value2', 'value3', 'value4', 'value5, 'value6', ‘value7’);  Gửi phản hồi kết quả về FrontEnd. FrontEnd sẽ render ra 1 user mới. |  |
| 18 | Tìm kiếm người dùng theo tên.   * DOM tới ô input và gán event keyup và hàm handleSearch * Gọi hàm searchUser(keyword, data) với keyword là value của ô input tìm kiếm và data là danh sách người dùng. * Gọi 1 mảng rỗng có tên là searchResult * Tạo 1 vòng lặp danh sách người dùng(data) và kiểm tra xem tên của từng người dùng có bao gồm keyword không, nếu là true sẽ push vào mảng searchResult.   let user = data[i];  if(user.UserName.toLowerCase().include(keyword.toLowerCase)){ searchResult.push(user);  }   * Gọi hàm handleSearch để render kêt quả. Nếu có kết quả tìm kiếm, tiến hành lọc dữ liệu và hiển thị kết quả tìm kiếm. Nếu không có từ khóa tìm kiếm, hiển thị toàn bộ dữ liệu |  |
| 19 | Lấy thông tin người dùng bằng UserID  SELECT \* FROM User  WHERE ProductID = ‘id của người dùng’; |  |
| 20 | Cập nhật người dùng.   * FrontEnd có giao diện giống với chức năng thêm người dùng nhưng sẽ có sẵn thông tin của người dùng cần cập nhật (sử dụng chức năng lấy thông tin người dùng và gán thông tin vào những ô input). Sau đó DOM tới các thẻ input lấy value và tiến hành cập nhật thông tin sản phẩm. Nhấn nút cập nhật sẽ gửi yêu cầu qua BackEnd. * BackEnd sẽ lấy UserID để tìm ra sản phẩm và cập nhật thông tin mới vào người dùng có id là UserID. Sau đó sẽ gửi phản hồi kết quả về phía FrontEnd. FrontEnd sẽ tiến hành render lại bảng để cập nhật thông tin mới của người dùng được chọn.   Cập nhật thông tin cho sản phẩm đó  UPDATE User  SET UserName= ‘value1’, Email = ‘value2’, Password= ‘value3’, Phone= ‘value4’, Age= ‘value5’,Role=’value6’  WHERE UserID= ‘id của người dùng’; |  |
| 21 | Xóa người dùng   * Gán sự kiện onclick gọi hàm xoaUser(${user.UserID}), hàm truyền vào ID cuả người dùng được chọn, sau đó xóa sản phẩm theo Id của người dùng đó DELETE FROM User WHERE UserID= user.UserID; |  |
| 22 | Sắp xếp theo giá tiền của sản phẩm.   * Gọi hàm sapXepTheoGia(data, sort) * Tạo một bản sao của mảng data để không ảnh hưởng đến dữ liệu ban đầu.   Let sortedPrice = […data];   * Nếu sort là “Từ bé đến lớn”: Mảng sẽ được sắp xếp theo giá từ bé đến lớn * Nếu sort là “Từ lớn đến bé”: Mảng sẽ được sắp xếp theo giá từ lớn đến bé * DOM tới thẻ select và gán 1 event là change 1 hàm để render kết quả * Gọi danh sách sản phẩm gán vào biến res * Gọi biến sort là value của thẻ input * Ta goi hàm sapXepTheoGia (res,sort); * Tiến hành render lại theo sortedPrice. |  |
| 23 | Sắp xếp theo tên của sản phẩm, user   * Gọi hàm sapXepTheoTen(data, sort) * Tạo một bản sao của mảng data để không ảnh hưởng đến dữ liệu ban đầu.   Let sortedName = […data];   * Nếu sort là “Từ A đến Z”: Mảng sẽ được sắp xếp theo tên từ A đến Z * Nếu sort là “Từ Z đến A”: Mảng sẽ được sắp xếp theo tên từ Z đến A * DOM tới thẻ select và gán 1 event là change 1 hàm để render kết quả * Gọi danh sách sản phẩm(danh sách người dùng) gán vào biến res * Gọi biến sort là value của thẻ input * Ta goi hàm sapXepTheoTen (res,sort);   Tiến hành render lại theo sortedName. |  |