**TÀI LIỆU CODE LOGIN EXAMPLE**

1. File **server.js** :

*var express = require('express');*

*var bodyParser = require('body-parser');*

*var session = require('express-session');*

*var app = express();*

Khai báo biến, sau đó gán module vào và khởi tạo app bằng câu lệnh app = express() , ở đây chúng ta dùng các module express , express-session và body-parser.

*app.use(bodyParser());*

*app.use(session({secret: 'ssshhhhh'}));*

*app.use(express.static(\_\_dirname + '/public'));*

App sử dụng bodyParser, mục ích để đưa các value của input lên server, sử dụng session để ghi nhớ phiên làm việc , không cần phải login lại , và sử dụng dirname public để include css, js hay image vào các trang view html sau này.

*require('./router/main')(app);*

Câu lệnh này để include file main.js ở đường dẫn route/main vào app để sử dụng như bình thường.

*app.set('views',\_\_dirname + '/views');*

*app.set('view engine', 'ejs');*

*app.engine('html', require('ejs').renderFile);*

Các câu lệnh này để cài đặt ejs vào trong view , thay thế html bằng ejs, ejs là một loại template có sẵn trong nodejs, có thể xuất data từ javascript ra, …

*var server = app.listen(3000,function(){*

*console.log("We have started our server on port 3000");*

*});*

Khởi tạo server có port là 3000 ( tức là 127.0.0.1:3000) và in ra command line dòng “We have started our server on port 3000”.

1. File **main.js :**

*app.get('/',function(req,res){*

*sess = req.session;*

*if(sess.username){*

*res.render('index');*

*}else{*

*res.redirect('/login');*

*}*

*});*

Dùng phương thức GET ở trang chủ ( tức là đường dẫn “/” ), khai báo biến sess = request của session . Kiểm tra nếu tồn tại session của username thì sẽ trả về trang chủ (render bằng view có tên là index) , nếu không thì sẽ tự động chuyển sang trang login , bắt yêu cầu đăng nhập.

*app.get('/login',function(req,res){*

*res.render('login', {message});*

*});*

Dùng phương thức GET ở trang login , render bằng view có tên là login, và một biến message.

*app.post('/login',function(req,res){*

*sess = req.session;*

*account = {*

*username : 'admin',*

*password : '1234'*

*}*

*var username = req.body.username;*

*var password = req.body.password;*

*if(username == account.username && password == account.password){*

*sess.username = req.body.username;*

*res.redirect('/');*

*}else{*

*res.render('login', {message : "Username or password is incorrect!"} );*

*}*

*});*

Dùng phương thức POST ở trang login , gửi giá trị của input username và password lên server để xử lý. Đầu tiên khai báo sess bằng request của session. Sau đó tạo một mảng account , có trường username = “admin” và trường password = “1234” . Tiếp tục lấy giá trị của input username và password gửi lên , so sánh với giá trị trong mảng account . Nếu bằng nhau thì gán sess.username chính bằng tên username, sau đó chuyển trang sang trang chủ. Nếu sai username hoặc password thì trả về trang login với message là Username or password is incorrect.

*app.get('/logout',function(req,res){*

*req.session.destroy(function(err){*

*if(err){*

*console.log(err);*

*}*

*else*

*{*

*res.redirect('/');*

*}*

*});*

*});*

Dùng phương thức GET ở đường dẫn /logout , mục đích là để đăng xuất tài khoản ra . Câu lệnh req.session.destroy mục đích là hủy đi, xóa đi session(phiên làm việc), để đăng xuất tài khoản , sau đó chuyển về trang chủ.

1. Các file index.ejs, login.ejs là các trang HTML, nằm trong thư mục view, được render theo đường dẫn(route) , là những thứ thấy được trên client.
2. Các file css, js hay image trong thư mục public là các resource để hỗ trợ cho các file HTML đã nói ở trên.