1. Express – validator : <https://www.npmjs.com/package/express-validator>

Install:

npm i express-validator

Config: Router

const { check, validationResult } = require('express-validator/check');

var regValid = [

check('email').isEmail().withMessage('Email không hợp lệ'),

check('password').isLength({min:5}).withMessage('Mật khẩu không được ngắn hơn 5 ký tự'),

check('password\_confirmation').custom((value, { req }) => {

if (value !== req.body.password) {

throw new Error('Mật khẩu và xác nhận mật khẩu không chính xác');

}else{

return true;

}

}),

check('fullname').not().isEmpty().withMessage('Fullname không được để trống'),

check('phone').not().isEmpty().withMessage('Phone không được để trống'),

check('fax').not().isEmpty().withMessage('Fax không được để trống')

];

**router.post('/register',regValid,function(req, res, next){**

var errors = validationResult(req);

if(!errors.isEmpty()){

res.render('register',{errors : errors.array()});

}else{

var dataInsert = {

email : req.body.email,

password : req.body.password,

fullname : req.body.fullname,

phone : req.body.phone,

fax: req.body.fax

};

var newUser = new userModel(dataInsert);

userModel.createUser(newUser);

res.redirect('/');

1. Express – session : <https://github.com/expressjs/session>
2. Connect – flash : <https://www.npmjs.com/package/connect-flash>

Install:

npm install connect-flash

Config: app.js

var flash **=** require('connect-flash');

app.use(flash());

app.use(function(req, res, next){

res.locals.success\_mess = req.flash(‘success\_mess’);

res.locals.error\_mess = req.flash(‘error\_mess’);

res.locals.error = req.flash(‘error’);

next();

});

1. Bcryptjs : <https://www.npmjs.com/package/bcryptjs>

Install

npm install bcryptjs

Config: ./model/user.js

var bcrypt **=** require('bcryptjs');

var mongoose = require('mongoose'), Schema = mongoose.Schema,bcrypt = require('bcryptjs');

var UserSchema = new Schema({

email : { type: String, required : true, trim : true},

password : String,

fullname : String,

phone : String,

fax: String,

created\_at : { type: Date, default: Date.now}

}, { collection : "users"});

**var User** = module.exports = mongoose.model('User', UserSchema);

**module.exports.createUser** = function(data){

bcrypt.genSalt(10, function(err, salt) {

bcrypt.hash(data.password, salt, function(err, hash) {

data.password = hash;

data.save();

**module.exports.getUserByEmail** = function(email, callback){

User.findOne({email : email}, callback);

**module.exports.comparePass** = function(password, hash, callback){

bcrypt.compare(password, hash, function(err, isMatch) {

if (err) throw err;

callback(null, isMatch);

**module.exports.getUserById** = function(id, callback){

User.findById(id, callback);

1. Mongoose : <https://mongoosejs.com/>
2. PassportJs : <http://www.passportjs.org/>

Install

npm install passport

npm install passport-local

Config

var passport = require('passport');

var LocalStrategy = require('passport-local').Strategy;

app.use(passport.initialize());

app.use(passport.session());

Login

**passport.use(new LocalStrategy({**

usernameField: 'email',

passwordField: 'password'}, function(email, password, done) {

userModel.getUserByEmail(email, function(err, user) {

if (err) { return done(err); }

if (!user) {

return done(null, false, { message: 'Incorrect email.' });

}

userModel.comparePass(password, user.password, function(err, isMatch){

if(err) throw err;

if(isMatch){

return done(null, user);

}else{

return done(null, false, { message: 'Incorrect password.' });

**passport.serializeUser(function(user, done) {**

done(null, user.id);

**passport.deserializeUser(function(id, done) {**

userModel.getUserById(id, function(err, user) {

done(err, user);

router.get('/login', function(req, res, next){

res.render('login', {errors:null});

router.post('/login',

passport.authenticate('local', { successRedirect: '/',

failureRedirect: '/login',

failureFlash: true })

KIỂM SOÁT ĐĂNG NHẬP – Logout

function checkAuth (req , res , next) {

if (req.isAuthenticated()) {

next();

} else {

res.redirect('/login');

router.get('/admin', checkAuth ,function (req , res , next) {

res.send("Hello " + req.user.fullname + "<br /><a href='/logout'>Logout</a>");

});

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

req.logout();

res.redirect('/login');

});