User Login:

//Javascript logic for user

|  |
| --- |
| const bcrypt = require('bcryptjs');  const User = require('../../models/User');  const nodemailer = require('nodemailer');  var transporter = nodemailer.createTransport({  service: 'gmail',  auth: {  user: 'rajatmohantyrm768@gmail.com',  pass: 'huibjwgevwwcaozp'  }  });  exports.getLogin = (req, res, next) => {  res.render('user/login', {  pageTitle: 'Login',  });  };  exports.getRegister = (req, res, next) => {  res.render('user/register', {  pageTitle: 'Register',  });  };  exports.postRegister = (req, res, next) => {  const name = req.body.name;  const email = req.body.email;  const password = req.body.password;  const password2 = req.body.password2;  const time = new Date().toISOString().split('T')[0];  let errors = [];  if (!name || !email || !password || !password2) {  errors.push({ msg: "Please fill all the fields." });  }  if (password !== password2) {  errors.push({ msg: 'Passwords do not match' });  }  if (password.length < 5) {  errors.push({ msg: 'Passwords should be atleast 6 characters' });  }  if (errors.length > 0) {  res.render('user/register', {  pageTitle: 'Register',  errors,  name,  email,  password,  password2  });  } else {  User.findOne({ email: email })  .then(user => {  if (user) {  errors.push({ msg: 'User is already resisterd with this email.' });  res.render('user/register', {  pageTitle: 'Register',  errors,  name,  email,  password,  password2  });  } else {  const cart = {};  const newUser = new User({ //instance of the user model  name,  email,  password,  cart,  time  });  //hasing password and saving user in database  bcrypt.genSalt(10, (err, salt) => {  bcrypt.hash(newUser.password, salt, (err, hash) => {  if (err)  throw err;  newUser.password = hash;  newUser.save()  .then(user => {  var mailOptions = {  from: 'rajatmohantyrm768@gmail.com',  to: email,  subject: 'Thank you for Registration',  html: 'Your Registration is success'  };  transporter.sendMail(mailOptions, function(error, info) {  if (error) {  console.log(error);  } else {  console.log('Email sent: ' + info.response);  }  });  req.flash('success\_msg', 'You are registered Successfully!');  res.redirect('/login');  })  .catch((err) => { //console.log(err)  req.flash('error\_msg', 'Something wrong happend.');  res.redirect('/home');  });  })  });  }  })  .catch((err) => { //console.log(err)  req.flash('error\_msg', 'Something wrong happend.');  res.redirect('/home');  });  }  };  exports.postLogin = (req, res, next) => {  const email = req.body.email;  const password = req.body.password;  User.findOne({ email: email })  .then(user => {  if (!user) {  req.flash('error\_msg', 'This is not registerd email Id.');  return res.redirect('/login');  }  bcrypt  .compare(password, user.password)  .then(doMatch => {  if (doMatch) {  req.session.isUserLoggedIn = true;  req.session.user = user;  return req.session.save(err => {  res.redirect('/home');  });  }  req.flash('error\_msg', 'Wrong Password.');  res.redirect('/login');  })  .catch(err => {  //console.log(err);  req.flash('error\_msg', 'Something wrong happened , please retry again.');  res.redirect('/login');  });  })  .catch((err) => { //console.log(err)  req.flash('error\_msg', 'Something wrong happend.');  res.redirect('/home');  });  };  exports.postLogout = (req, res, next) => {  req.session.destroy(err => {  //console.log(err);  res.redirect('/');  });  }; |

//User Login

|  |
| --- |
| <%- include('../includes/head.ejs') %>  <body>  <%- include('../includes/navigation.ejs') %>  <% if(error\_msg !='undefined') {%>  <strong> <%= error\_msg %></strong>  <% } %>  <div class="card" style="width:auto;border: none;text-align: center;">  <div class="card-body">  <form class="text-center p-5" style="width:20rem;text-align: center;" action="/login" method="POST">  <p class="h4 mb-4">Sign in</p>  <input type="email" id="email" class="form-control mb-4" placeholder="E-mail" name="email" value="<%= typeof email !='undefined' ? email:'' %>" required>  <input type="hidden" name="\_csrf" value="<%= csrfToken %>">  <input type="password" id="password" class="form-control mb-4" placeholder="Password" name="password" required>  <button class="btn btn-info btn-block my-4" type="submit">Sign in</button>  <p>Not a member?  <a href="/register">Register</a>  </p>  </form>  </div>  </div>  <%- include('../includes/end.ejs') %> |

// user register

|  |
| --- |
| <%- include('../includes/head.ejs') %>  <body>  <%- include('../includes/navigation.ejs') %>  <% if(typeof errors !='undefined') {%>  <%for(error of errors) { %>  <div class="alert alert-dismissible alert-danger">  <button type="button" class="close" data-dismiss="alert">&times;</button>  <strong> <%= error.msg %></strong>  </div>  <% } %>  <% } %>  <div class="card" style="width:auto;border: none;text-align: center; background-image: url('gif\_image.gif');">  <div class="card-body">  <form class="text-center p-5" style="width:20rem;text-align: center;" action="/register" method="POST">  <p class="h4 mb-4">Sign up</p>  <input type="text" id="name" class="form-control mb-4" placeholder="Name" name="name" value="<%= typeof name !='undefined' ? name:'' %>" required>  <input type="email" id="email" class="form-control mb-4" placeholder="E-mail" name="email" value="<%= typeof email !='undefined' ? email:'' %>" required>  <input type="password" id="password" class="form-control mb-4" placeholder="Password" name="password" value="<%= typeof password !='undefined' ? password:'' %>" required>  <input type="hidden" name="\_csrf" value="<%= csrfToken %>">  <input type="password" id="password" class="form-control mb-4" placeholder="Confirm Password" name="password2" value="<%= typeof password2 !='undefined' ? password2:'' %>" required>  <button class="btn btn-info btn-block my-4" type="submit">Sign up</button>  <p>Already Registered?  <a href="/login">Login</a>  </p>  </form>  </div>  </div>  <%- include('../includes/end.ejs') %> |

//////////////////////

//Admin Login:

|  |
| --- |
| <%- include('../includes/head.ejs') %>  <body>  <%- include('../includes/navigation.ejs') %>  <% if(error\_msg !='undefined') {%>  <strong> <%= error\_msg %></strong>  <% } %>  <div class="card" style="width:auto;border: none;text-align: center;">  <div class="card-body">  <form class="text-center p-5" style="width:20rem;text-align: center;" action="/admin/login" method="POST">  <p class="h4 mb-4">Sign in</p>  <input type="text" id="email" class="form-control mb-4" placeholder="E-mail" name="username" value="<%= typeof email !='undefined' ? email:'' %>" required>  <input type="hidden" name="\_csrf" value="<%= csrfToken %>">  <input type="password" id="password" class="form-control mb-4" placeholder="Password" name="password" required>  <button class="btn btn-info btn-block my-4" type="submit">Sign in</button>  </form>  </div>  </div>  <%- include('../includes/end.ejs') %>  <%- include('../includes/head.ejs') %>  <body> |

//Admin Register:

|  |
| --- |
| <%- include('../includes/navigation.ejs') %>  <% if(typeof errors !='undefined') {%>  <%for(error of errors) { %>  <div class="alert alert-dismissible alert-danger">  <button type="button" class="close" data-dismiss="alert">&times;</button>  <strong> <%= error.msg %></strong>  </div>  <% } %>  <% } %>  <div class="card" style="width:auto;border: none;text-align: center;">  <div class="card-body">  <form class="text-center p-5" style="width:20rem;text-align: center;" action="/admin/register" method="POST">  <p class="h4 mb-4">Sign up</p>  <input type="email" id="email" class="form-control mb-4" placeholder="E-mail" name="email" value="<%= typeof email !='undefined' ? email:'' %>" required>  <input type="password" id="password" class="form-control mb-4" placeholder="Password" name="password" value="<%= typeof password !='undefined' ? password:'' %>" required>  <input type="hidden" name="\_csrf" value="<%= csrfToken %>">  <button class="btn btn-info btn-block my-4" type="submit">Register</button>  </form>  </div>  </div>  <%- include('../includes/end.ejs') %> |

//

|  |
| --- |
| const express = require('express');  const path = require('path');  const rootDir = require('./util/path.js');  const bodyParser = require('body-parser');  const errorControllers = require('./controllers/error');  const mongoose = require('mongoose');  const flash = require('connect-flash');  const session = require('express-session');  const MongoDBStore = require('connect-mongodb-session')(session);  const csrf = require('csurf');  const multer = require('multer');  const bcrypt = require('bcryptjs');  const indexRoutes = require('./routes/index');  const adminRoutes = require('./routes/admin/admin');  const adminAuthRoutes = require('./routes/admin/adminAuth');  const userAuthRoutes = require('./routes/user/userAuth');  const Admin = require('./models/Admin');  const User = require('./models/User.js');  const app = express();  const csrfProtection = csrf();  //Db connection  const MONGODB\_URI = require('./config/keys').MongoURI; //db keys  mongoose.connect(MONGODB\_URI, { useNewUrlParser: true, useUnifiedTopology: true })  .then(() => {  console.log("Database Connection established");  })  .catch(err => {  console.log(err);  });  const fileStorage = multer.diskStorage({  destination: function(req, file, cb) {  console.log('======', file);  cb(null, 'images/');  //cb(null, path.join(\_\_dirname, './models'))  },  filename: function(req, file, cb) {  // cb(null, new Date().toISOString() + '-' + file.originalname);  //let ext=path.extname(file.originalname)  cb(null, file.originalname);  }  });  //console.log('filestorage=====', fileStorage.destination);  const fileFilter = function(req, file, cb) {  if (  file.mimetype === 'image/png' ||  file.mimetype === 'image/jpg' ||  file.mimetype === 'image/jpeg'  ) {  cb(null, true);  } else {  cb(null, false);  }  };  //var upload = multer({ storage: fileStorage, fileFilter: fileFilter, limits: { fileSize: 1282810 } }).single("imageUp")  app.use(multer({ storage: fileStorage, fileFilter: fileFilter, limits: { fileSize: 1282810 } }).single("imageUp"));  app.use('/images', express.static(path.join(rootDir, 'images')));  app.use(bodyParser.json()); //we are dealing with json  app.use(bodyParser.urlencoded({ extended: false })); //parsing the post request datas  app.use(express.static(path.join(rootDir, 'public'))); //including public folder accesseible like css and other stuffs in public folder are now accessible  app.set('view engine', 'ejs'); //express ko batata hai hum by deafult kaun sa templating engine use kar rahe  app.set('views', 'views');  //express session middlewares  const store = new MongoDBStore({  uri: MONGODB\_URI,  collection: 'sessions',  });  app.use(  session({  secret: 'secret', //used to sign the hashed  resave: false,  saveUninitialized: false,  store: store,  // cookie: {  // maxAge: 1000\*60 // 1 min  // },  })  );  //to store user object, because in req.session.user we have only data not have full object that's why we can't use this with models methods like add to cart  app.use((req, res, next) => {  if (req.session.user) {  User.findById(req.session.user.\_id)  .then(user => {  req.user = user;  next();  }).catch(err => console.log(err));  } else {  next();  }  });  app.use(csrfProtection);  app.use(flash());  // Global variables -> saare views mai hum es variables ko use kar skte hai  app.use(function(req, res, next) {  res.locals.success\_msg = req.flash('success\_msg');  res.locals.error\_msg = req.flash('error\_msg');  res.locals.error = req.flash('error');  res.locals.isAuthenticated = req.session.isUserLoggedIn;  res.locals.isAdmin = req.session.isAdminLoggedIn;  res.locals.csrfToken = req.csrfToken();  next();  });  const port = process.env.PORT || 3000  app.use(adminAuthRoutes);  app.use(adminRoutes);  app.use(indexRoutes);  app.use(userAuthRoutes);  app.use(errorControllers.get404);  app.listen(port, () => {  console.log(`app is listening on port${port  }`);  }); |

//Product Design:

|  |
| --- |
| <%- include('../includes/head.ejs') %>  <body>  <%- include('../includes/navigation.ejs') %>  <% if(error\_msg !='undefined') {%>  <strong> <%= error\_msg %></strong>  <% } %>  <main class="centered">  <h1><%= product.title %></h1>  </main>  <hr>  <div class="centered">  <img class="centere" src="/<%= product.imageUrl %>" class="img-thumbnail" width="360" height="320" alt="<%= product.title %>">  </div>  <b><h3 class="centered">$ <%= product.price %></h2></b>  <p class="centered"><%= product.description %></p>  <div class="card\_\_actions">  <% if(isAuthenticated) { %>  <%- include('../includes/addToCart.ejs',{product:product.\_id}) %>  <% } %>  </div>  <%- include('../includes/end.ejs') %> |