# MERN stack - Shopping Online
# Lab 02

## Contents

# 1 MERN stack

### 1.0.1 Models

- Create 'server/models/Models.js' file

```
//CLI: npm install mongoose --save
const mongoose = require('mongoose');
// schemas
const AdminSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  username: String,
  password: String
}, { versionKey: false });
const CategorySchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  name: String
}, { versionKey: false });
const CustomerSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  username: String,
  password: String,
  name: String,
  phone: String,
  email: String,
  active: Number,
  token: String,
}, { versionKey: false });
const ProductSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  name: String,
  price: Number,
  image: String,
  cdate: Number,
  category: CategorySchema
}, { versionKey: false });
const ItemSchema = mongoose.Schema({
  product: ProductSchema,
  quantity: Number
}, { versionKey: false, _id: false });
const OrderSchema = mongoose.Schema({
  _id: mongoose.Schema.Types.ObjectId,
  cdate: Number,
  total: Number,
  status: String,
  customer: CustomerSchema,
  items: [ItemSchema]
}, { versionKey: false });
// models
const Admin = mongoose.model('Admin', AdminSchema);
const Category = mongoose.model('Category', CategorySchema);
const Customer = mongoose.model('Customer', CustomerSchema);
const Product = mongoose.model('Product', ProductSchema);
const Order = mongoose.model('Order', OrderSchema);
module.exports = { Admin, Category, Customer, Product, Order };
```

### 1.0.2 Utils

- Create 'server/utils/MyConstants.js' file

```
const MyConstants = {
  DB_SERVER: '<cluster>.mongodb.net',
  DB_USER: '<db_user>',
  DB_PASS: '<db_pass>',
  DB_DATABASE: '<db_name>',
  EMAIL_USER: '<email_user>', // Microsoft mail service
  EMAIL_PASS: '<email_pass>',
  JWT_SECRET: '<jwt_secret>',
  JWT_EXPIRES: '<jwt_expires>', // in milliseconds
};
module.exports = MyConstants;
```

- Create 'server/utils/MongooseUtil.js' file

```
//CLI: npm install mongoose --save
const mongoose = require('mongoose');
const MyConstants = require('./MyConstants');
const uri = 'mongodb+srv://' + MyConstants.DB_USER + ':' + MyConstants.DB_PASS + '@' + MyConstants.DB_SERVER + '/' + MyConstants.DB_DATABASE;
mongoose.connect(uri, { useNewUrlParser: true })
  .then(() => { console.log('Connected to ' + MyConstants.DB_SERVER + '/' + MyConstants.DB_DATABASE); })
  .catch((err) => { console.error(err); });
```

- Create 'server/utils/CryptoUtil.js' file

```
//CLI: npm install crypto --save
const CryptoUtil = {
  md5(input) {
    const crypto = require('crypto');
    const hash = crypto.createHash('md5').update(input).digest('hex');
    return hash;
  }
};
module.exports = CryptoUtil;
```

- Create 'server/utils/EmailUtil.js' file

```
//CLI: npm install nodemailer --save
const nodemailer = require('nodemailer');
const MyConstants = require('./MyConstants');
const transporter = nodemailer.createTransport({
  service: 'hotmail',
  auth: {
    user: MyConstants.EMAIL_USER,
    pass: MyConstants.EMAIL_PASS
  }
});
const EmailUtil = {
  send(email, id, token) {
    const text = 'Thanks for signing up, please input these informations to activate your account:\n\t .id: ' + id + '\n\t .token: ' + token;
    return new Promise(function (resolve, reject) {
      const mailOptions = {
        from: MyConstants.EMAIL_USER,
        to: email,
        subject: 'Signup | Verification',
        text: text
      };
      transporter.sendMail(mailOptions, function (err, result) {
        if (err) reject(err);
        resolve(true);
      });
    });
  }
};
module.exports = EmailUtil;
```

- Create 'server/utils/JwtUtil.js' file

```
//CLI: npm install jsonwebtoken --save
const jwt = require('jsonwebtoken');
const MyConstants = require('./MyConstants');
const JwtUtil = {
  genToken(username, password) {
    const token = jwt.sign(
      { username: username, password: password },
      MyConstants.JWT_SECRET,
      { expiresIn: MyConstants.JWT_EXPIRES }
    );
    return token;
  },
  checkToken(req, res, next) {
    const token = req.headers['x-access-token'] || req.headers['authorization'];
    if (token) {
      jwt.verify(token, MyConstants.JWT_SECRET, (err, decoded) => {
        if (err) {
          return res.json({
            success: false,
            message: 'Token is not valid'
          });
        } else {
          req.decoded = decoded;
          next();
        }
      });
    } else {
      return res.json({
        success: false,
        message: 'Auth token is not supplied'
      });
    }
  }
};
module.exports = JwtUtil;
```

### 1.0.3 Stylesheets

- Update 'client-admin/src/App.css' and 'client-customer/src/App.css' files

```
/* styles.css */
.body-customer {
  width: 1150px;
  height: fit-content;
  margin: 0px auto;
}
.body-admin {
  width: fit-content;
  height: fit-content;
  margin: 0px auto;
}
div.inline {
  display: inline-block;
}
figure.caption-right {
  display: flex;
  align-items: center;
}
.align-center {
  width: fit-content;
  margin: 0px auto;
}
.align-valign-center {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  padding: 0px 20px 20px 20px;
  border: 1px solid black;
}
.text-center {
  text-align: center;
}
span.link {
  cursor: pointer;
  color: blue;
  text-decoration: underline;
}
/* menu.css */
div.border-bottom {
  border-bottom: 1px solid black;
  padding: 10px 0px 10px 0px;
}
div.float-left {
  float: left;
}
div.float-right {
  float: right;
}
div.float-clear {
  clear: both;
}
ul.menu {
  margin: 0px;
  padding: 0px;
}
li.menu {
  display: inline;
  margin-right: 20px;
  text-transform: uppercase;
}
form.search {
  margin: 0px;
}
input.keyword {
  width: 110px;
}
/* datatable.css */
.datatable {
  text-align: center;
  border-collapse: collapse;
}
/* header */
tr:first-child.datatable {
  background-color: #ffcc00;
}
/* even rows, except header */
tr:not(:first-child):nth-child(even).datatable {
  background-color: #f5f5f5;
}
/* odd rows, except header */
tr:not(:first-child):nth-child(odd).datatable {
  background-color: #ffeeaa;
}
/* except header */
tr:not(:first-child):hover.datatable {
  background-color: #dcfcfc;
  cursor: pointer;
}
```

## 1.1 Functionals

### 1.1.1 Admin - login & logout

#### 1.1.1.1 Server

- Create 'server/models/AdminDAO.js' file

```
require('../utils/MongooseUtil');
const Models = require('./Models');

const AdminDAO = {
  async selectByUsernameAndPassword(username, password) {
    const query = { username: username, password: password };
    const admin = await Models.Admin.findOne(query);
    return admin;
  }
};
module.exports = AdminDAO;
```

- Create ’server/api/admin.js’ file

```
const express = require('express');
const router = express.Router();
// utils
const JwtUtil = require('../utils/JwtUtil');
// daos
const AdminDAO = require('../models/AdminDAO');
// login
router.post('/login', async function (req, res) {
  const username = req.body.username;
  const password = req.body.password;
  if (username && password) {
    const admin = await AdminDAO.selectByUsernameAndPassword(username,password);
    if (admin) {
      const token = JwtUtil.genToken();
      res.json({ success: true, message: 'Authentication successful', token: token });
    } else {
      res.json({ success: false, message: 'Incorrect username or password' });
    }
  } else {
    res.json({ success: false, message: 'Please input username and password' });
  }
});
router.get('/token', JwtUtil.checkToken, function (req, res) {
  const token = req.headers['x-access-token'] || req.headers['authorization'];
  res.json({ success: true, message: 'Token is valid', token: token });
});
module.exports = router;
```

- Update ’server/index.js’ file

```
...
// apis
app.use('/api/admin', require('./api/admin.js'));
```

- Postman:

- (POST) http://localhost:3000/api/admin/login
  - Body (raw+JSON): { ”username”: ”admin”, ”password”: ”123” }
- (GET) http://localhost:3000/api/admin/token
  - Headers: ”x-access-token”: <token>

#### 1.1.1.2 Client-admin

- Create ’client-admin/src/contexts/MyContext.js’ file

```
import React from 'react';
const MyContext = React.createContext();
export default MyContext;
```

- Create ’client-admin/src/contexts/MyProvider.js’ file

```
import React, { Component } from 'react';
import MyContext from './MyContext';

class MyProvider extends Component {
  constructor(props) {
    super(props);
    this.state = { // global state
      // variables
      token: '',
      username: '',
      // functions
      setToken: this.setToken,
      setUsername: this.setUsername
    };
  }
  setToken = (value) => {
    this.setState({ token: value });
  }
  setUsername = (value) => {
    this.setState({ username: value });
  }
  render() {
    return (
      <MyContext.Provider value={this.state}>
        {this.props.children}
      </MyContext.Provider>
    );
  }
}
export default MyProvider;
```

- Update 'client-admin/src/App.js' file

```
import './App.css';
import React, { Component } from 'react';
import MyProvider from './contexts/MyProvider';
import Login from './components/LoginComponent';
import Main from './components/MainComponent';

class App extends Component {
  render() {
    return (
      <MyProvider>
        <Login />
        <Main />
      </MyProvider>
    );
  }
}
export default App;
```

- Create 'client-admin/src/components/LoginComponent.js' file

```
import axios from 'axios';
import React, { Component } from 'react';
import MyContext from '../contexts/MyContext';

class Login extends Component {
  static contextType = MyContext; // using this.context to access global state
  constructor(props) {
    super(props);
    this.state = {
      txtUsername: '',
      txtPassword: ''
    };
  }
  render() {
    if (this.context.token === '') {
      return (
        <div className="align-valign-center">
          <h2 className="text-center">ADMIN LOGIN</h2>
          <form>
            <table className="align-center">
              <tbody>
                <tr>
                  <td>Username</td>
                  <td><input type="text" value={this.state.txtUsername} onChange={(e) => { this.setState({ txtUsername: e.target.value }) }} /></td>
                </tr>
                <tr>
                  <td>Password</td>
                  <td><input type="password" value={this.state.txtPassword} onChange={(e) => { this.setState({ txtPassword: e.target.value }) }} /></td>
                </tr>
                <tr>
                  <td></td>
                  <td><input type="submit" value="LOGIN" onClick={(e) => this.btnLoginClick(e)} /></td>
                </tr>
              </tbody>
            </table>
          </form>
        </div>
      );
    }
    return (<div />);
  }
  // event-handlers
  btnLoginClick(e) {
    e.preventDefault();
    const username = this.state.txtUsername;
    const password = this.state.txtPassword;
    if (username && password) {
      const account = { username: username, password: password };
      this.apiLogin(account);
    } else {
      alert('Please input username and password');
    }
  }
  // apis
  apiLogin(account) {
    axios.post('/api/admin/login', account).then((res) => {
      const result = res.data;
      if (result.success === true) {
        this.context.setToken(result.token);
        this.context.setUsername(account.username);
      } else {
        alert(result.message);
      }
    });
  }
}
export default Login;
```

- Create 'client-admin/src/components/MenuComponent.js' file

```
import React, { Component } from 'react';
import MyContext from '../contexts/MyContext';

class Menu extends Component {
  static contextType = MyContext; // using this.context to access global state
  render() {
    return (
      <div className="border-bottom">
        <div className="float-left">
          <ul className="menu">
            <li className="menu"><a href="">Home</a></li>
            <li className="menu"><a href="">Category</a></li>
            <li className="menu"><a href="">Product</a></li>
            <li className="menu"><a href="">Order</a></li>
            <li className="menu"><a href="">Customer</a></li>
          </ul>
        </div>
        <div className="float-right">
          Hello <b>{this.context.username}</b> | <a href="" onClick={() => this.lnkLogoutClick
```

```
              ()}>Logout</a>
        </div>
        <div className="float-clear" />
      </div>
    );
  }
  // event-handlers
  lnkLogoutClick() {
    this.context.setToken('');
    this.context.setUsername('');
  }
}
export default Menu;
```

- Create 'client-admin/src/components/HomeComponent.js' file

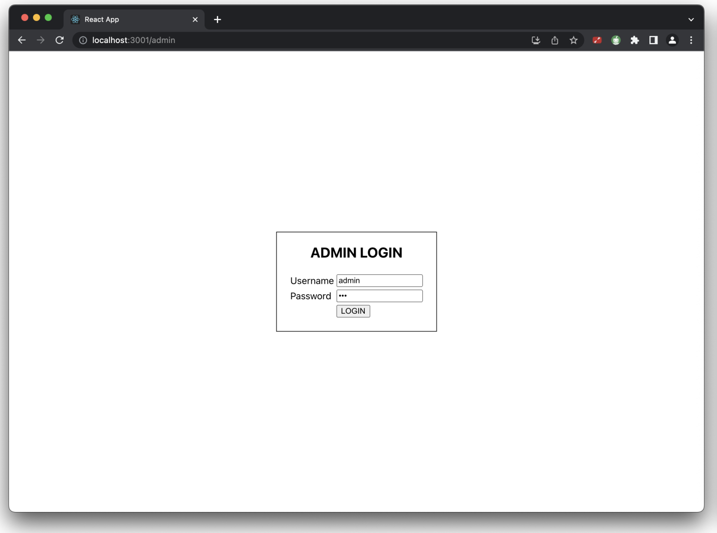
```
import React, { Component } from 'react';

class Home extends Component {
  render() {
    return (
      <div className="align-center">
        <h2 className="text-center">ADMIN HOME</h2>
        <img src="http://cliparting.com/wp-content/uploads/2018/03/animated-emoticons-2018-13.gif" width="800px" height="600px" alt="" />
      </div>
    );
  }
}
export default Home;
```

- Create 'client-admin/src/components/MainComponent.js' file

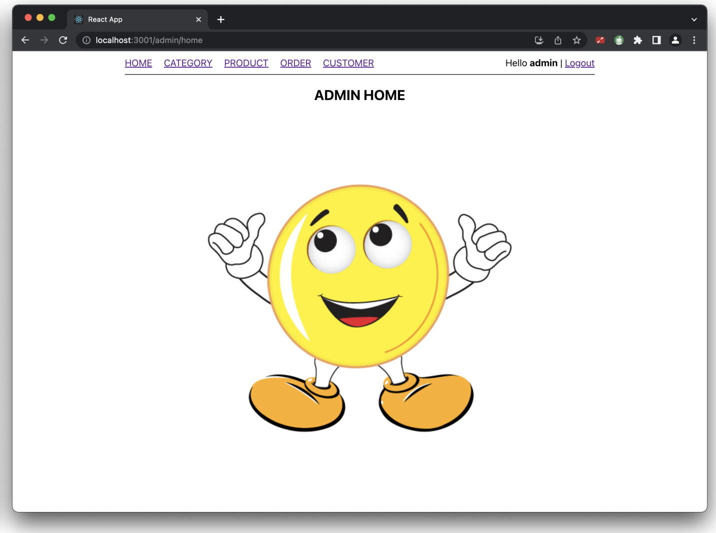
```
import React, { Component } from 'react';
import MyContext from '../contexts/MyContext';
import Menu from './MenuComponent';
import Home from './HomeComponent';

class Main extends Component {
  static contextType = MyContext; // using this.context to access global state
  render() {
    if (this.context.token !== '') {
      return (
        <div className="body-admin">
          <Menu />
          <Home />
        </div>
      );
    }
    return (<div />);
  }
}
export default Main;
```

- Browser: http://localhost:3001/admin

#### 1.1.1.3 Screenshots



Username admin
Password •••



#### 1.1.1.4 React Router

- Install react-router-dom library.

```
$ npm install react-router-dom --save
```

- Update 'client-admin/src/App.js' file

```
...
import { BrowserRouter } from 'react-router-dom';

class App extends Component {
  render() {
    return (
      ...
      <BrowserRouter >
        <Main />
      </BrowserRouter>
      ...
    );
  }
}
...
```

- Update 'client-admin/src/components/MainComponent.js' file

```
...
import { Routes, Route, Navigate } from 'react-router-dom';

class Main extends Component {
  ...
  render() {
    ...
    return (
      <div className="body-admin">
        <Menu />
        <Routes>
          <Route path='/admin' element={<Navigate replace to='/admin/home' />} />
          <Route path='/admin/home' element={<Home />} />
        </Routes>
      </div>
    );
    ...
  }
}
...
```

- Update 'client-admin/src/components/MenuComponent.js' file

```
...
import { Link } from 'react-router-dom';

class Menu extends Component {
  ...
  render() {
    return (
      ...
      <div className="float-left">
        <ul className="menu">
          <li className="menu"><Link to='/admin/home'>Home</Link></li>
          <li className="menu"><Link to=''>Category</Link></li>
          <li className="menu"><Link to=''>Product</Link></li>
          <li className="menu"><Link to=''>Order</Link></li>
          <li className="menu"><Link to=''>Customer</Link></li>
        </ul>
      </div>
      <div className="float-right">
        ... <Link to='/admin/home' onClick={() => this.lnkLogoutClick()}>Logout</Link>
      </div>
      ...
    );
  }
  ...
}
...
```

- Browser: http://localhost:3001/admin