**Server.js**

require(**'./models/db'**);  
  
**const** express = require(**'express'**);  
**const** path = require(**'path'**);  
**const** *exphbs* = require(**'express-handlebars'**);  
**const** bodyparser = require(**'body-parser'**);  
  
**const** employeeController = require(**'./controllers/employeeController'**);  
  
**var** app = express();  
app.use(bodyparser.**urlencoded**({  
 **extended**: **true**}));  
app.use(bodyparser.**json**());  
app.set(**'views'**, path.join(\_\_dirname, **'/views/'**));  
app.engine(**'hbs'**, *exphbs*({ **extname**: **'hbs'**, **defaultLayout**: **'mainLayout'**, **layoutsDir**: \_\_dirname + **'/views/layouts/'** }));  
app.set(**'view engine'**, **'hbs'**);  
  
app.listen(3001, () => {  
 ***console***.log(**'Express server started at port : 3001'**);  
});  
  
app.use(**'/employee'**, employeeController);

**AddorEdit.hbs**

<**h3**>{{**viewTitle**}}</**h3**>  
  
<**form action="/employee" method="POST" autocomplete="off"**>  
 <**input type="hidden" name="\_id" value="**{{**employee**.**\_id**}}**"**>  
  
 <**div class="form-group"**>  
 <**label**>Customer ID</**label**>  
 <**input type="text" class="form-control" name="custid" placeholder="Custid" value="**{{**employee**.**custid**}}**"**>  
 <**div class="text-danger"**>  
 {{**employee**.**custidError**}}</**div**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>Customer Name</**label**>  
 <**input type="text" class="form-control" name="fullName" placeholder="Full Name" value="**{{**employee**.**fullName**}}**"**>  
 <**div class="text-danger"**>  
 {{**employee**.**fullNameError**}}</**div**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>Customer Email</**label**>  
 <**input type="text" class="form-control" name="email" placeholder="Email" value="**{{**employee**.**email**}}**"**>  
 <**div class="text-danger"**>  
 {{**employee**.**emailError**}}</**div**>  
 </**div**>  
 <**div class="form-group"**>  
 <**button type="submit" class="btn btn-info"**><**i class="fa fa-database"**></**i**> Submit</**button**>  
 <**a class="btn btn-secondary" href="/employee/list"**><**i class="fa fa-list-alt"**></**i**> View All</**a**>  
 </**div**>  
</**form**>

**List.hbs**

<**h3**><**a class="btn btn-secondary" href="/employee"**><**i class="fa fa-plus"**></**i**> Create New</**a**> Customer List</**h3**>  
<**table class="table table-striped"**>  
 <**thead**>  
 <**tr**>  
 <**th**>Customer ID</**th**>  
 <**th**>Customer Name</**th**>  
 <**th**>Customer Email</**th**>  
 <**th**></**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {{#**each list**}}  
 <**tr**>  
 <**td**>{{**this**.**custid**}}</**td**>  
 <**td**>{{**this**.**fullName**}}</**td**>  
 <**td**>{{**this**.**email**}}</**td**>  
 <**td**>  
 <**a href="/employee/**{{**this**.**\_id**}}**"**><**i class="fa fa-pencil fa-lg" aria-hidden="true"**></**i**></**a**>  
 <**a href="/employee/delete/**{{**this**.**\_id**}}**" onclick="return** *confirm*(**'Are you sure to delete this record ?'**);**"**><**i class="fa fa-trash fa-lg" aria-hidden="true"**></**i**></**a**>  
 </**td**>  
 </**tr**>  
 {{/**each**}}  
 </**tbody**>  
</**table**>

**db.js**

**const mongoose** = require(**'mongoose'**);  
  
**mongoose**.connect(**'mongodb://localhost:27017/EmployeeDB'**, { **useNewUrlParser**: **true** }, (err) => {  
 **if** (!err) { ***console***.log(**'MongoDB Connection Succeeded.'**) }  
 **else** { ***console***.log(**'Error in DB connection : '** + err) }  
});  
  
require(**'./employee.model'**);

**employeemodel.js**

**const mongoose** = require(**'mongoose'**);  
  
**var** employeeSchema = **new mongoose**.**Schema**({  
 **custid**: {  
 **type**: ***String***,  
 **required**: **'This field is required.'** },  
 **fullName**: {  
 **type**: ***String***,  
 **required**: **'This field is required.'** },  
 **email**: {  
 **type**: ***String*** }  
});  
  
*// Custom validation for email*employeeSchema.path(**'email'**).validate((val) => {  
 **emailRegex** = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)\*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;  
 **return emailRegex**.test(val);  
}, **'Invalid e-mail.'**);  
  
**mongoose**.model(**'Employee'**, employeeSchema);

**employeecontroller.js**

**const** express = require(**'express'**);  
**var** router = express.Router();  
**const mongoose** = require(**'mongoose'**);  
**const** Employee = **mongoose**.model(**'Employee'**);  
  
router.get(**'/'**, (req, res) => {  
 res.render(**"employee/addOrEdit"**, {  
 **viewTitle**: **"Insert Customer"** });  
});  
  
router.post(**'/'**, (req, res) => {  
 **if** (req.**body**.**\_id** == **''**)  
 *insertRecord*(req, res);  
 **else** *updateRecord*(req, res);  
});  
  
  
**function** *insertRecord*(req, res) {  
 **var** employee = **new** Employee();  
 employee.**custid** = req.**body**.**custid**;  
 employee.**fullName** = req.**body**.**fullName**;  
 employee.**email** = req.**body**.**email**;  
 employee.save((err, doc) => {  
 **if** (!err)  
 res.**redirect**(**'employee/list'**);  
 **else** {  
 **if** (err.**name** == **'ValidationError'**) {  
 *handleValidationError*(err, req.**body**);  
 res.render(**"employee/addOrEdit"**, {  
 **viewTitle**: **"Insert Customer"**,  
 **employee**: req.**body** });  
 }  
 **else  
 *console***.log(**'Error during record insertion : '** + err);  
 }  
 });  
}  
  
**function** *updateRecord*(req, res) {  
 Employee.findOneAndUpdate({ **\_id**: req.**body**.**\_id** }, req.**body**, { **new**: **true** }, (err, doc) => {  
 **if** (!err) { res.**redirect**(**'employee/list'**); }  
 **else** {  
 **if** (err.**name** == **'ValidationError'**) {  
 *handleValidationError*(err, req.**body**);  
 res.render(**"employee/addOrEdit"**, {  
 **viewTitle**: **'Update Customer'**,  
 **employee**: req.**body** });  
 }  
 **else  
 *console***.log(**'Error during record update : '** + err);  
 }  
 });  
}  
  
  
router.get(**'/list'**, (req, res) => {  
 Employee.find((err, docs) => {  
 **if** (!err) {  
 res.render(**"employee/list"**, {  
 **list**: docs  
 });  
 }  
 **else** {  
 ***console***.log(**'Error in retrieving customer list :'** + err);  
 }  
 });  
});  
  
  
**function** *handleValidationError*(err, body) {  
 **for** (**field in** err.**errors**) {  
 **switch** (err.**errors**[**field**].**path**) {  
 **case 'custid'**:  
 body[**'custidError'**] = err.**errors**[**field**].**message**;  
 **break**;  
 **case 'fullName'**:  
 body[**'fullNameError'**] = err.**errors**[**field**].**message**;  
 **break**;  
 **case 'email'**:  
 body[**'emailError'**] = err.**errors**[**field**].**message**;  
 **break**;  
 **default**:  
 **break**;  
 }  
 }  
}  
  
router.get(**'/:id'**, (req, res) => {  
 Employee.findById(req.**params**.**id**, (err, doc) => {  
 **if** (!err) {  
 res.render(**"employee/addOrEdit"**, {  
 **viewTitle**: **"Update Customer"**,  
 **employee**: doc  
 });  
 }  
 });  
});  
  
router.get(**'/delete/:id'**, (req, res) => {  
 Employee.findByIdAndRemove(req.**params**.**id**, (err, doc) => {  
 **if** (!err) {  
 res.**redirect**(**'/employee/list'**);  
 }  
 **else** { ***console***.log(**'Error in customer delete :'** + err); }  
 });  
});  
  
module.**exports** = router;