/* Assignment No. 3B */

****

/* File name : 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(3000, () => {
   console.log('Express server started at port : 3000');
});

app.use('/employee', employeeController);
```

****

/* Folder name : controllers */
/* File name : 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 Employee"
   });
});
```

```
router.post('/', (req, res) => {
    if (req.body._id == '')
        insertRecord(req, res);
        else
        updateRecord(req, res);
});


function insertRecord(req, res) {
    var employee = new Employee();
    employee.fullName = req.body.fullName;
    employee.email = req.body.email;
    employee.mobile = req.body.mobile;
    employee.city = req.body.city;
    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 Employee",
                    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 Employee',
                    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 employee list :' + err);
        }
    });
});


function handleValidationError(err, body) {
    for (field in err.errors) {
        switch (err.errors[field].path) {
            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 Employee",
                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 employee delete :' + err); }
    });
});

module.exports = router;
```

****

```
/* Folder name : models */
/* File name : 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');
```

****

```
/* Folder name : models */
/* File name : employee.model.js */

const mongoose = require('mongoose');

var employeeSchema = new mongoose.Schema({
    fullName: {
        type: String,
        required: 'This field is required.'
    },
    email: {
        type: String
    },
    mobile: {
        type: String
    },
    city: {
        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);

********************************************************************************

/* Folder name : views/employee */
/* File name : 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>Full 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>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-row">
      <div class="form-group col-md-6">
         <label>Mobile</label>
         <input type="text" class="form-control" name="mobile" placeholder="Mobile"
value="{{employee.mobile}}">
      </div>
      <div class="form-group col-md-6">
         <label>City</label>
         <input type="text" class="form-control" name="city" placeholder="City"
value="{{employee.city}}">
      </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>
```

```
**********************************************************************************

/* Folder name : views/employee */
/* File name : list.hbs */

<h3><a class="btn btn-secondary" href="/employee"><i class="fa fa-plus"></i> Create New</a>
Employee List</h3>
<table class="table table-striped">
   <thead>
      <tr>
         <th>Full Name</th>
         <th>Email</th>
         <th>Mobile</th>
         <th>City</th>
         <th></th>
      </tr>
   </thead>
   <tbody>
      {{#each list}}
      <tr>
         <td>{{this.fullName}}</td>
         <td>{{this.email}}</td>
         <td>{{this.mobile}}</td>
         <td>{{this.city}}</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>


**********************************************************************************

/* Folder name : views/layouts */
/* File name : mainLayout.hbs */

<!DOCTYPE html>

<html>

<head>
   <title>Node.js express mongDB CRUD</title>
```

```
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdkn
LPMO"
      crossorigin="anonymous">
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
</head>

<body class="bg-info">
   <div class="row">
      <div class="col-md-6 offset-md-3" style="background-color: #fff;margin-top:
25px;padding:20px;">
         {{{body}}}
      </div>
   </div>

</body>

</html>

**********************************************************************************

/* File name : package.json */

{
  "name": "project",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node server.js"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.18.3",
    "express": "^4.16.4",
    "express-handlebars": "^3.0.0",
    "mongoose": "^5.13.17"
  },
  "description": ""
}

**********************************************************************************
```