```
Microsoft Windows [Version 10.0.19044.2846]
(c) Microsoft Corporation. All rights reserved.

/* Going into Project Directory */

C:\Users\HP>cd C:\Users\HP\Documents\WADL\Project


/* Initializing a node project */

C:\Users\HP\Documents\WADL\Project>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (project)
version: (1.0.0)
description:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\HP\Documents\WADL\Project\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": {
  },
  "description": ""
}

Is this OK? (yes) yes
```

```
/* Installing all Dependencies*/

C:\Users\HP\Documents\Node.js-Expess-MongoDB-CRUD-master\project>npm install -g body-parser

added 30 packages in 2s

6 packages are looking for funding
  run `npm fund` for details

C:\Users\HP\Documents\Node.js-Expess-MongoDB-CRUD-master\project>npm install -g express

added 57 packages in 1s

7 packages are looking for funding
  run `npm fund` for details

C:\Users\HP\Documents\Node.js-Expess-MongoDB-CRUD-master\project>npm install -g
express-handlebars

added 48 packages in 4s

14 packages are looking for funding
  run `npm fund` for details


C:\Users\HP\Documents\WADL\Project>npm i mongoose@5.3.4
npm WARN old lockfile
npm WARN old lockfile The package-lock.json file was created with an old version of npm,
npm WARN old lockfile so supplemental metadata must be fetched from the registry.
npm WARN old lockfile
npm WARN old lockfile This is a one-time fix-up, please be patient...
npm WARN old lockfile
npm WARN deprecated bson@1.0.9: Fixed a critical issue with BSON serialization documented in
CVE-2019-2391, see https://bit.ly/2KcpXdo for more details
npm WARN deprecated bson@1.1.0: Fixed a critical issue with BSON serialization documented in
CVE-2019-2391, see https://bit.ly/2KcpXdo for more details

added 98 packages, and audited 99 packages in 4s

15 vulnerabilities (2 moderate, 7 high, 6 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force
```

```
Run `npm audit` for details.
```

```
/* Running server.js file */

C:\Users\HP\Documents\Node.js-Expess-MongoDB-CRUD-master\project>node server.js
(node:12864) Warning: Accessing non-existent property 'count' of module exports inside circular
dependency
(Use `node --trace-warnings ...` to show where the warning was created)
(node:12864) Warning: Accessing non-existent property 'findOne' of module exports inside circular
dependency
(node:12864) Warning: Accessing non-existent property 'remove' of module exports inside circular
dependency
(node:12864) Warning: Accessing non-existent property 'updateOne' of module exports inside circular
dependency
Express server started at port : 3000
MongoDB Connection Succeeded.
```

```
/* 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": ""
}

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