**EXPT 4 and EXPT 5:**

**courseController.js**

//Import the dependencies

const express = require('express');

const mongoose = require('mongoose');

//Creating a Router

var router = express.Router();

//Link

const Course = mongoose.model('Course');

//Router Controller for READ request

router.get('/', (req, res) => {

res.render("course/courseAddEdit", {

viewTitle: "Insert a New Course for FrCRCE"

});

});

//Router Controller for UPDATE request

router.post('/', (req, res) => {

if (req.body.\_id == '')

insertIntoMongoDB(req, res);

else

updateIntoMongoDB(req, res);

});

//Creating function to insert data into MongoDB

function insertIntoMongoDB(req, res) {

var course = new Course();

course.courseName = req.body.courseName;

course.courseId = req.body.courseId;

course.courseDuration = req.body.courseDuration;

course.courseFee = req.body.courseFee;

course.save((err, doc) => {

if (!err)

res.redirect('course/list');

else

console.log('Error during record insertion : ' + err);

});

}

//Creating a function to update data in MongoDB

function updateIntoMongoDB(req, res) {

Course.findOneAndUpdate({ \_id: req.body.\_id }, req.body, { new: true }, (err, doc) => {

if (!err) { res.redirect('course/list'); } else {

if (err.name == 'ValidationError') {

handleValidationError(err, req.body);

res.render("course/courseAddEdit", {

//Retaining value to be displayed in the child view

viewTitle: 'Update Course Details',

employee: req.body

});

} else

console.log('Error during updating the record: ' + err);

}

});

}

//Router to retrieve the complete list of available courses

router.get('/list', (req, res) => {

Course.find((err, docs) => {

if (!err) {

res.render("course/list", {

list: docs

});

} else {

console.log('Failed to retrieve the Course List: ' + err);

}

});

});

//Creating a function to implement input validations

function handleValidationError(err, body) {

for (field in err.errors) {

switch (err.errors[field].path) {

case 'courseName':

body['courseNameError'] = err.errors[field].message;

break;

default:

break;

}

}

}

//Router to update a course using it's ID

router.get('/:id', (req, res) => {

Course.findById(req.params.id, (err, doc) => {

if (!err) {

res.render("course/courseAddEdit", {

viewTitle: "Update Course Details",

course: doc

});

}

});

});

//Router Controller for DELETE request

router.get('/delete/:id', (req, res) => {

Course.findByIdAndRemove(req.params.id, (err, doc) => {

if (!err) {

res.redirect('/course/list');

} else { console.log('Failed to Delete Course Details: ' + err); }

});

});

module.exports = router;

**course.model.js**

const mongoose = require('mongoose');

//Attributes of the Course object

var courseSchema = new mongoose.Schema({

courseName: {

type: String,

required: 'This field is required!'

},

courseId: {

type: String

},

courseDuration: {

type: String

},

courseFee: {

type: String

}

});

mongoose.model('Course', courseSchema);

**mongodb.js**

const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/FRCRCE', { useNewUrlParser: true }, (err) => {

if (!err) {

console.log('Successfully Established Connection with MongoDB')

} else {

console.log('Failed to Establish Connection with MongoDB with Error: ' + err)

}

});

//Connecting Node and MongoDB

require('./course.model');

**courseAddEdit.hbs**

<!-- Obtaining value from the course controller -->

<h3>{{viewTitle}}</h3>

<form action="/course" method="POST">

<input type="hidden" name="\_id" value="{{course.\_id}}">

<div class = "form-group">

<label>Course Name</label>

<input type="text" maxlength="100" class="form-control" name="courseName" placeholder="Course Name" value="{{course.courseName}}">

<div class="text-danger">

{{course.courseNameError}}</div>

</div>

<div class = "form-group">

<label>Course ID</label>

<input type="number" min='10000' max='99999' class="form-control" name="courseId" placeholder="Course Id" value="{{course.courseId}} " required>

</div>

<div class = "form-row">

<div class = "form-group col-md-6">

<label>Course Duration</label>

<input type="number" min='10' max='99' class="form-control" name="courseDuration" placeholder="Course Duration (Hrs)" "{{course.courseDuration}} " required>

</div>

<div class = "form-group col-md-6">

<label>Course Fee</label>

<input type="number" min='100' max="100000" class="form-control" name="courseFee" placeholder="Course Fee (USD)" "{{course.courseFee}} " required>

</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="/course/list"><i class="fa fa-list-alt"></i> View All Courses</a>

</div>

</form>

**list.hbs**

<div>

<a class="btn btn-secondary" href="/course"><i class="fa fa-plus"></i> Create New</a>

<h3 align="center">FrCRCE's Course List</h3>

</div>

<table class="table table-striped">

<thead>

<tr>

<th>Course Name</th>

<th>Course Id</th>

<th>Course Duration(Hrs)</th>

<th>Course Fee(USD)</th>

<th></th>

</tr>

</thead>

<tbody>

{{#each list}}

<tr align="center">

<td>{{this.courseName}}</td>

<td>{{this.courseId}}</td>

<td>{{this.courseDuration}}</td>

<td>{{this.courseFee}}</td>

<td>

<a href="/course/{{this.\_id}}"> Edit </a>

<a href="/course/delete/{{this.\_id}}" onclick="return confirm('Are you sure to delete this record ?');"> Delete </a>

</td>

</tr>

{{/each}}

</tbody>

</table>

**mainlayout.hbs**

<!DOCTYPE html>

<html>

<head>

<title>Fr CRCE</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<img src="" data-wp-preserve="%3Cscript%20src%3D%22https%3A%2F%2Fajax.googleapis.com%2Fajax%2Flibs%2Fjquery%2F3.2.1%2Fjquery.min.js%22%3E%3C%2Fscript%3E" data-mce-resize="false" data-mce-placeholder="1" class="mce-object" width="20" height="20" alt="&lt;script&gt;" title="&lt;script&gt;" />

<img src="" data-wp-preserve="%3Cscript%20src%3D%22https%3A%2F%2Fmaxcdn.bootstrapcdn.com%2Fbootstrap%2F3.3.7%2Fjs%2Fbootstrap.min.js%22%3E%3C%2Fscript%3E" data-mce-resize="false" data-mce-placeholder="1" class="mce-object" width="20" height="20" alt="&lt;script&gt;" title="&lt;script&gt;" />

</head>

<body class = "bg-info">

<div class="row">

<div class="col-md-6 offset-md-3" style="background-color: #fff; margin-top: 40px; padding:20px;">

<!-- retrieving HTML String from the child Views -->

{{{body}}}

</div>

</div>

</body>

</html>

**script.js**

require('./models/mongodb');

//Import the necessary packages

const express = require('express');

var app = express();

const path = require('path');

const exphb = require('express-handlebars');

const bodyparser = require('body-parser');

const courseController = require('./controllers/courseController');

app.use(bodyparser.urlencoded({

extended: true

}));

//Create a welcome message and direct them to the main page

app.get('/', (req, res) => {

res.send('<h2 style = "font-family: Malgun Gothic; color: midnightblue "> Welcome to Fr Conceicao Rodrigues College of Enginerring!!</h2> Click Here to go to <b><a href = "/course">Course Page</a></b > ');

});

app.use(bodyparser.json());

//Configuring Express middleware for the handlebars

app.set('views', path.join(\_\_dirname, '/views/'));

app.engine('hbs', exphb({ extname: 'hbs', defaultLayout: 'mainLayout', layoutDir: \_\_dirname + 'views/layouts/' }));

app.set('view engine', 'hbs');

//Establish the server connection

//PORT ENVIRONMENT VARIABLE

const port = process.env.PORT || 8080;

app.listen(port, () => console.log(`Listening on port ${port}..`));

//Set the Controller path which will be responding the user actions

app.use('/course', courseController);





