**Worksheet 5**

|  |  |
| --- | --- |
| **Student Name: Aibansuk Snaitang** | **UID: 23MCA20072** |
| **Branch: MCA** | **Section/Group: 3/A** |
| **Semester: 3rd** | **Date of Performance: 20/09/24** |
| **Subject Name:Back End Technologies** | **Subject Code: 23CAH-705** |

**Aim/Overview of the practical:**

**Use mongoose to connect database for performing CURD operations.**

**Code :- Index.ejs:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title> CRUD </title>

</head>

<body>

<h1>Student CRUD Operations</h1>

<h2>Add Student</h2>

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

<label for="name">Name:</label>

<input type="text" name="name" required><br><br>

<label for="uid">UID:</label>

<input type="text" name="uid" required><br><br>

<label for="section">Section:</label>

<input type="text" name="section" required><br><br>

<input type="submit" value="Add Student">

</form>

<h2>Update Student Name</h2>

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

<label for="updateUid">UID:</label>

<input type="text" name="updateUid" required><br><br>

<label for="updateName">New Name:</label>

<input type="text" name="updateName" required><br><br>

<input type="submit" value="Update Name">

</form>

<h2>Delete Student</h2>

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

<label for="deleteUid">UID:</label>

<input type="text" name="deleteUid" required><br><br>

<input type="submit" value="Delete Student">

</form>

<h2>Student List</h2>

<table display="block" border="1px">

<tr>

<th>Student Name</th>

<th>Student UID</th>

<th>Student Section</th>

</tr>

<% students.forEach(s =>{ %>

<tr>

<td><%= s.name %></td>

<td><%= s.uid %></td>

<td><%= s.section %></td>

</tr>

<% }); %>

</table>

</body>

</html>

**App.js:**

const express = require('express'); const mongoose = require('mongoose'); const ejs = require('ejs'); const app = express(); app.use(express.urlencoded({ extended: true })); app.set('view engine', 'ejs'); const connectDB = async () => {

try {

await mongoose.connect("mongodb://127.0.0.1:27017/demoExp"); console.log("MongoDB connected successfully");

} catch (err) { console.error("Failed to connect to MongoDB", err.message); process.exit(1);

}

};

connectDB(); const studentSchema = { name: { type: String,

required: true,

}, uid: { type: String, required: true,

}, section: { type: String, required: true,

},

};

const studentData = mongoose.model('students', studentSchema); app.post('/addStudent', async (req, res) => { try { const { name, uid, section } = req.body; const newStudent = new studentData({ name, uid, section }); await newStudent.save();

res.redirect('/'); } catch (error) { console.error("Error adding student:", error);

} }); app.post('/updateStudent', async (req, res) => { try { const { updateUid, updateName } = req.body; const result = await studentData.updateOne({ uid: updateUid }, { $set: { name: updateName } }); res.redirect('/'); } catch (error) { console.error("Error updating student:", error);

} }); app.post('/deleteStudent', async (req, res) => { try { const { deleteUid } = req.body; const result = await studentData.deleteOne({ uid: deleteUid });

res.redirect('/'); } catch (error) { console.error("Error deleting student:", error);

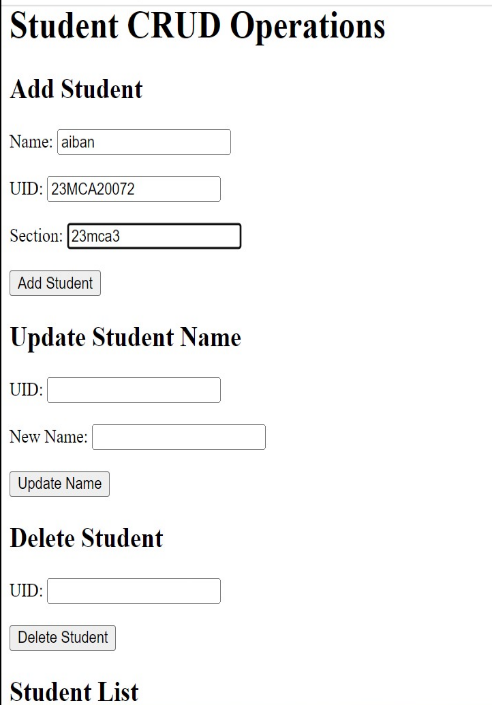
} }); app.get('/', async (req, res) => { try { const students = await studentData.find(); res.render('index', { students });

} catch (error) { console.error("Error fetching students:", error);

} }); app.listen(3000, () => { console.log('Server listening on port 3000');

});

**Output :-**





**Task to be done:**

In this experiment i performed CURD operation using mongooes and express js, node js and mongodb. **Learning outcomes (What I have learnt)**

About mongodb.

Learn some basic command to create, update, delete, retrieve data from mongodb.

Implementation of express js, node js code to perform CURD operation.

Implementation of various routes to perform CURD operation in node js.