**ASSIGNMENT 12 – W3D4 – LESSON13 –MongoDB App Programming:**

**In this assignment, you will practice setting up the tools and programming database application for MongoDB, based on the following requirement:**

**Task 1: Implement an application to read existing data from a MongoDB database:**

**Create a new project for a node.js CLI application, named, my\_library\_books\_app, using npm.**

**Implement code for the application, such that when executed, it connects to your MongoDB server and queries/fetches data (i.e. Books documents) from the booksCollection found in the database named, MyLibraryBooks-DB (which you created in Assignment 9) and displays the data to the console.**

const MongoClient = require('mongodb').MongoClient;

let conn = null;

async function main(){

conn = await MongoClient.connect('mongodb://localhost:27017', {

    useNewUrlParser:true,

    useUnifiedTopology: true

})

        console.log('Connected......');

        const db = conn.db('library-db');

        const collection = db.collection('books');

const books = await getBooks(collection);

books.forEach(book => console.log(book));

return "Operation Completed";

};

const getBooks = async function(collection){

return await collection.find({}).toArray();

};

main()

.then(console.log)

.catch(console.error)

.finally(()=>{if(!conn) conn.close();})

PS C:\Users\Admissions\Desktop\MSDClasses\CS415\workplaceCS415\cs415\my\_library\_books\_app> node .\app.js

Connected......

{

\_id: new ObjectId("61094c49504968a0e89c0f5a"),

ISBN: '978-0135166307',

Title: 'Core Java Fundamentals',

OverdueFee: '$1.75',

Publisher: 'Pearson',

'Date Published': '2018-08-27'

}

{

\_id: new ObjectId("61094d8b504968a0e89c0f5b"),

ISBN: '978-0321714114',

Title: 'C++ Primer',

OverdueFee: '$9.99',

Publisher: 'Stanley B. Lipmann',

'Date Published': '2019-08-08'

}

{

\_id: new ObjectId("61094daf504968a0e89c0f5c"),

ISBN: '1234534',

Title: 'Dagmay Fetene',

OverdueFee: '$50.00',

Publisher: 'Dr Kibreab Kibrewosen',

'Date Published': '1999-02-04'

}

{

\_id: new ObjectId("61094db8504968a0e89c0f5d"),

ISBN: '978-0073523323',

Title: 'Database System Concept 6th Edition',

OverdueFee: '$1.47',

Publisher: 'MC-Graw Hill',

'Date Published': '2011-05-19'

}

Operation Completed

**Task 2: Implement an application to write data to a new MongoDB database:**

**Create a new project for a node.js CLI application, named, my\_products\_app, using npm.**

**Implement code for the application, such that when executed, it connects to your MongoDB server and creates a new database named, MyProducts-DB, creates a collection named, productsCollection and it inserts the following Product data:**

Product1:{“product\_id”: 101, “name”: “Apple”, “unit-price”: 1.09, “quantity”: 125}

Product2:{“product\_id”: 102, “name”: “Banana”, “unit-price”: 0.89, “quantity”: 94}

Product3: {“product\_id”: 103, “name”: “Carrot”, “unit-price”: 1.15, “quantity”: 341}

const MongoClient = require('mongodb').MongoClient;

let conn = null;

const ManyProducts = [{"product\_id": 101, "name": "Apple", "unit-price": 1.09, "quantity": 125},

    {"product\_id": 102, "name": "Banana", "unit-price": 0.89, "quantity": 94},

    {"product\_id": 103, "name": "Carrot", "unit-price": 1.15, "quantity": 341}]

async function main() {

    conn = await MongoClient.connect('mongodb://localhost:27017', {

        useNewUrlParser: true,

        useUnifiedTopology: true

    })

    console.log('Connected......');

    const db = conn.db('product-db');

    const collection = db.collection('products');

    addManyProducts(collection,ManyProducts);

    const products = await getProducts(collection);

    products.forEach(product => console.log(product));

    return "Operation Completed";

};

const getProducts = async function (collection) {

    return await collection.find({}).toArray();

};

const addNewProduct = async function (collection, newProduct) {

    return await collection.insertOne(newProduct);

};

const addManyProducts = async function (collection, ManyProducts) {

    return await collection.insertMany(ManyProducts);

};

main()

    .then(console.log)

    .catch(console.error)

    .finally(() => { if (!conn) conn.close(); })

PS C:\Users\Admissions\Desktop\MSDClasses\CS415\workplaceCS415\cs415\my\_products\_app> node .\app.js

Connected......

{

\_id: new ObjectId("611138d3b955f21b31eb0916"),

product\_id: 101,

name: 'Apple',

'unit-price': 1.09,

quantity: 125

}

{

\_id: new ObjectId("611138d3b955f21b31eb0917"),

product\_id: 102,

name: 'Banana',

'unit-price': 0.89,

quantity: 94

}

{

\_id: new ObjectId("611138d3b955f21b31eb0918"),

product\_id: 103,

name: 'Carrot',

'unit-price': 1.15,

quantity: 341

}

Operation Completed

**Note: While performing the above tasks, remember to take screenshot(s) of your work and save into a document, which you will also submit to the Assignment 12 item on Sakai.**

**//-- The End --//**