CSE3002-INTERNET AND WEB PROGRAMMING

NAME: YOGESWARI SAHU

REGISTRATION NUMBER: 18BCE0928

FACULTY: JAYAKUMAR K

SLOT: L13+L14

LAB ASSIGNMENT-12

QUESTION:

12. Write a Node JS program to perform database operations. Use MongoDB for data integration services.

i. Write a NodeJs program to perform debit operation for a bank account. The HTML form should get input for the account no and the amount to be debited. The entered amount has to be reduced from their balance. In the database, maintain account number and balance.

PROCEDURE:

The tags and attributes used:

<html>-

Root element of an HTML page(contains everything in the html)

<head> -

Tells information about the document and contains any links

<body>-

Contains the body, i.e. the content of the body

<h1>,<h2>,<h3>,<h4>,<h5>,<h6>-

The different sizes of headings we can use in which h1 is the largest and h6 is the smallest

<form> -

used to create an HTML form for user input.

<input> -

specifies an input field where the user can enter data.

Libraries:

1. **Express**: It is a Node.js web application framework that provides a robust set of features for web APIs.

2. **Mongoose**: Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js.

3. **Mongodb**: The Database used for the database operations.

4. **Body-parser**: Is used to send data from frontend form to server.

Procudure/Algorithm:

1) Make a form with the required inputs i.e., Account number and Amount to be debited
2) Import all the libraries needed i.e. Express,Moongoose, Body Parser
3) Make a Mongo DB Schema for the account number with its balance
4) To create a database in MongoDB, start by creating a MongoClient object, then specify a connection URL with the correct ip address and the name of the database you want to create.
5) Make two routes , one for the index page that will be called when the server is fired and second when we submit the form of the amount to be debited.

6) When the second route(/debited) is called, the amount gets deducted from the balance currently in the database

7) We print them out in the terminal as soon as the amount is debited.

CODE:

app.js

```
var express = require("express");
var bodyParser = require("body-parser");
const mongoose = require('mongoose');
mongoose.connect('mongodb://localhost:27017/bankDebit');
var db = mongoose.connection;
db.on('error', console.log.bind(console, "connection error"));
db.once('open', function (callback) {
 console.log("connection succeeded");
})
var app = express();
app.use(bodyParser.json());
app.use(express.static('public'));
app.use(bodyParser.urlencoded({extended: true}));
var Schema = mongoose.Schema;
var bankSchema = new Schema({
 account: String,
 amount: Number
 });
var Bank = mongoose.model('Bank', bankSchema);
var Bn = new Bank({
 account: "123",
 amount: 9000
 });

 Bn.save(function(error) {
 console.log("Your bn has been saved!");
 if (error) {
   console.error(error);
 }});
 app.get('/',function(req,res){
  res.render('index.html')
 })
app.post('/debited', function (req, res) {
   var account = req.body.account;
   var amount = req.body.amount;
   console.log(account);
   console.log(amount);
   Bank.findOne({account: account}, function (err, bankval) {
      if (err) {
    console.log("Error");
  }
  bankval.amount = bankval.amount - amount;
  bankval.save();
  console.log(bankval);
   });
 return res.redirect('debit.html');
})
app.get('/', function (req, res) {
 Bank.inserOne({account: account}, function (err, bankval) {
 if (err) {
 console.log("Error");
 }
 bankval.amount = baamount;
 console.log(bankval.amount);

 });
 res.set({
 'Access-control-Allow-Origin': '*'
 });
 return res.redirect('index.html');
}).listen(3000)
console.log("server listening at port 3000");
```

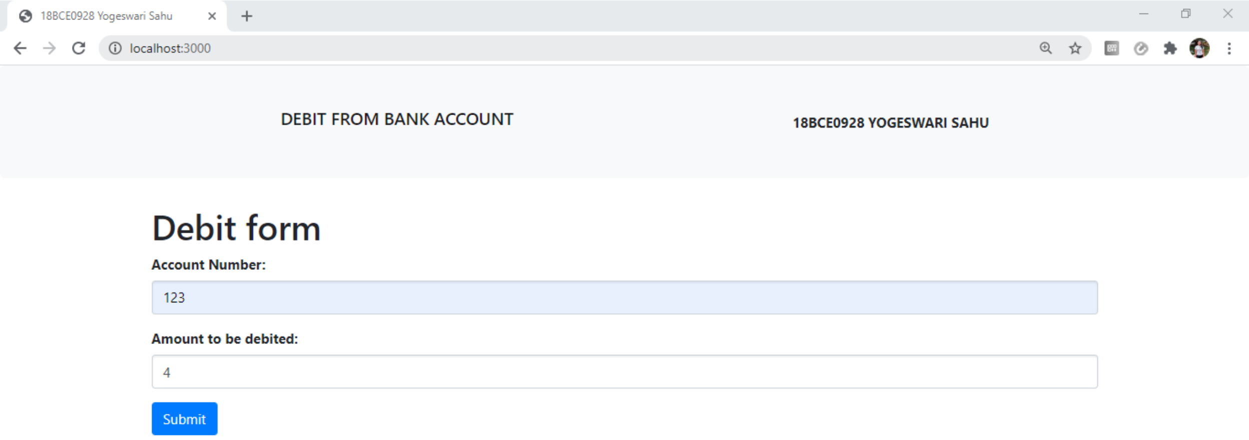
index.html

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
<title>18BCE0928 Yogeswari Sahu</title>
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
</head>
<body >

    <nav class="navbar navbar-light bg-light">
        <span class="navbar-brand mb-0 h1"> DEBIT FROM BANK ACCOUNT</span>
        <span class="form-inline my-2 my-lg-0"> <b> 18BCE0928 YOGESWARI SAHU</b> </span>
        </nav>
        <div class="container-sm">
    <form action="/debited" method="post">
        <h1> Debit form </h1>
        <div class="form-group">
          <label for="account">Account Number:</label>
          <input type="text" class="form-control"  name="account" id="account" aria-describedby="emailHelp" placeholder="Account Number" required>
        </div>
        <div class="form-group">
          <label for="amount">Amount to be debited:</label>
          <input type="number" class="form-control" name="amount" id="amount" placeholder="Amount" required>
        </div>
        <button type="submit" id="submitDetails" name="submitDetails" class="btn btn-primary" value="Submit">Submit</button>
      </form>
 <br>
 <br>
 <br>

</div>

</body>
</html>
```

debit.html

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Debit</title>
</head>
<body>
    <h1> 18BCE0928 Yogeswari Sahu</h1>

    <h1>Amount Debited</h1>
</body>
</html>
```

OUTPUT:



DEBIT FROM BANK ACCOUNT

18BCE0928 YOGESWARI SAHU

Debit form

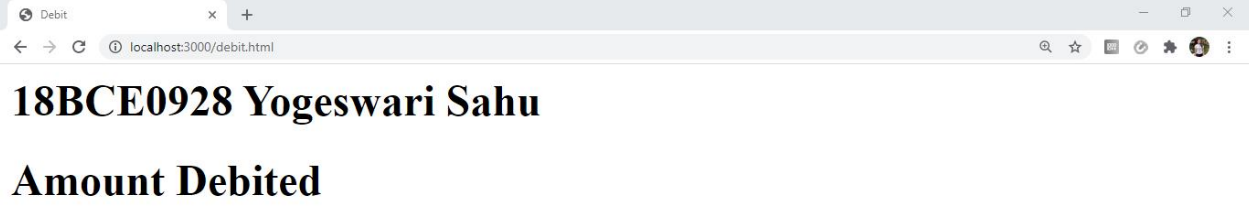
Account Number:

123

Amount to be debited:

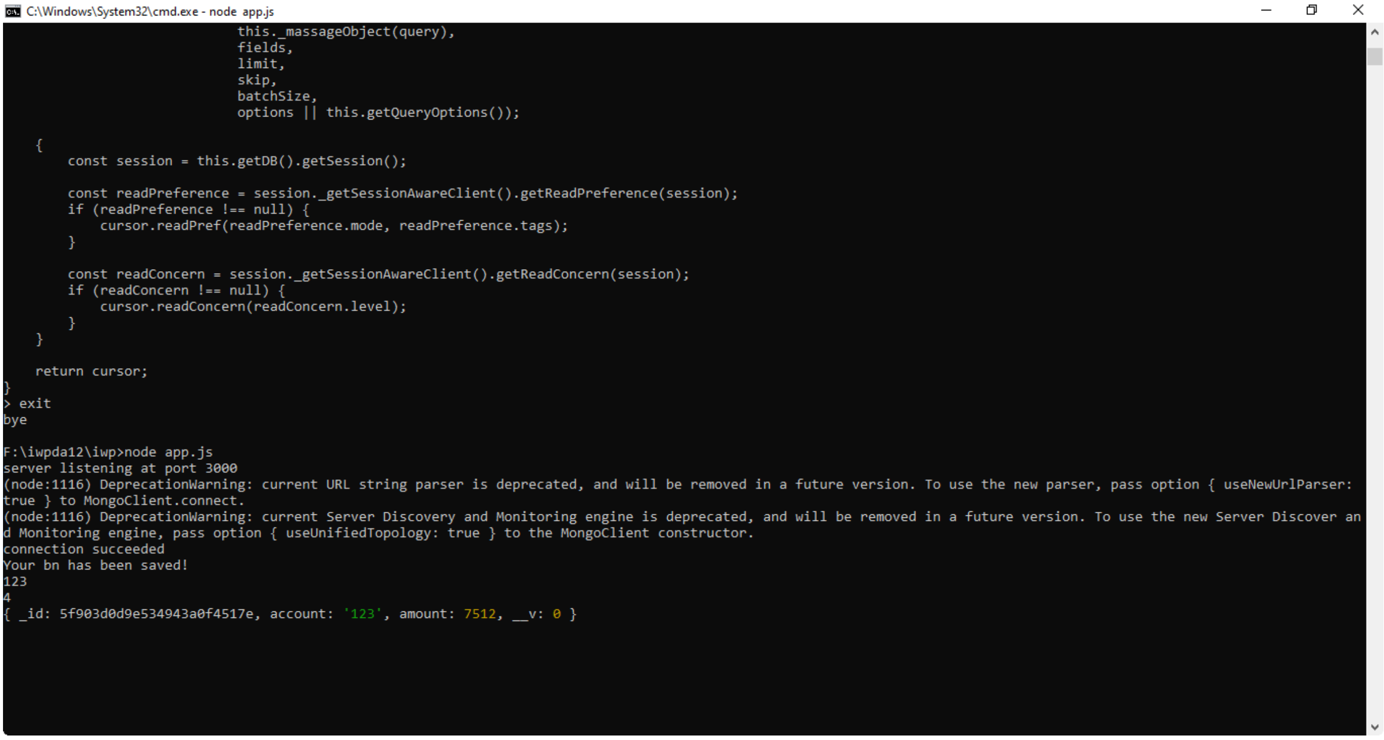
4

Submit



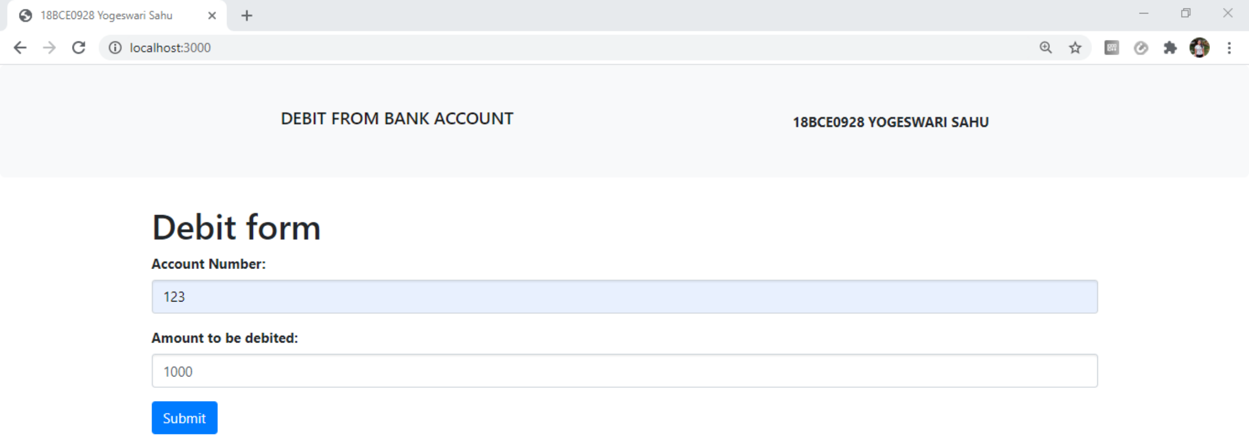
18BCE0928 Yogeswari Sahu

Amount Debited



C:\Windows\System32\cmd.exe - node app.js

```
                             this._massageObject(query),
                             fields,
                             limit,
                             skip,
                             batchSize,
                             options || this.getQueryOptions());

    {
        const session = this.getDB().getSession();

        const readPreference = session._getSessionAwareClient().getReadPreference(session);
        if (readPreference !== null) {
            cursor.readPref(readPreference.mode, readPreference.tags);
        }

        const readConcern = session._getSessionAwareClient().getReadConcern(session);
        if (readConcern !== null) {
            cursor.readConcern(readConcern.level);
        }
    }

    return cursor;
}
> exit
bye

F:\iwpda12\iwp>node app.js
server listening at port 3000
(node:1116) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser:
true } to MongoClient.connect.
(node:1116) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover an
d Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
connection succeeded
Your bn has been saved!
123
4
{ _id: 5f903d0d9e534943a0f4517e, account: '123', amount: 7512, __v: 0 }
```



DEBIT FROM BANK ACCOUNT

18BCE0928 YOGESWARI SAHU

Debit form

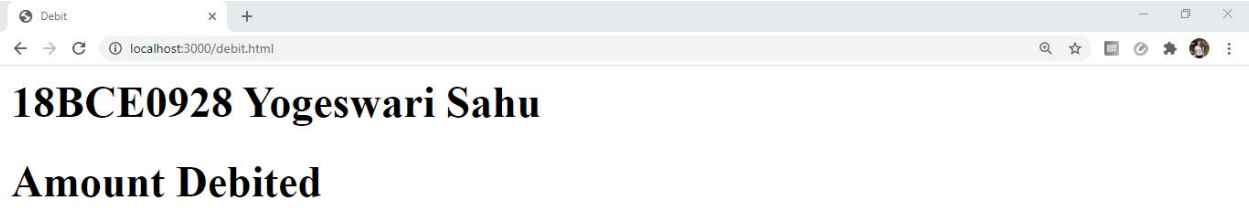
Account Number:

123

Amount to be debited:

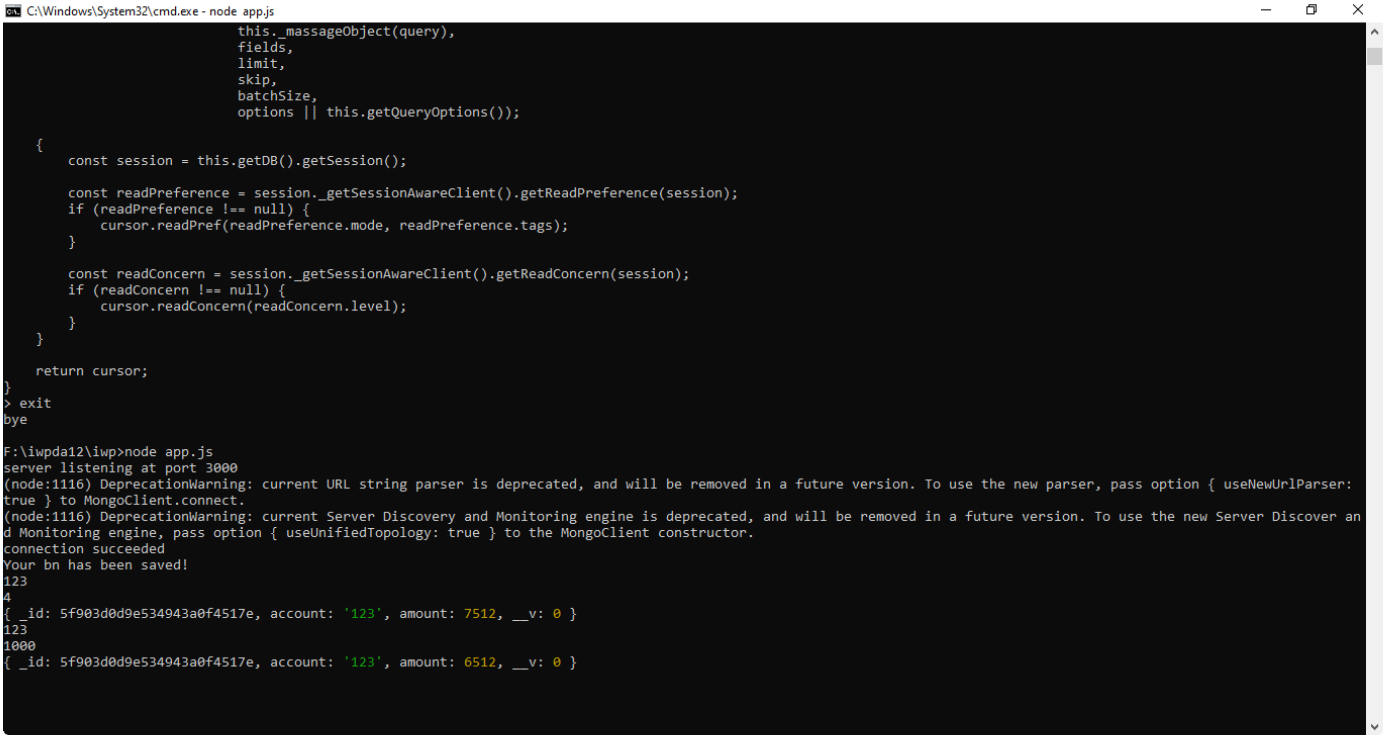
1000

Submit



18BCE0928 Yogeswari Sahu

Amount Debited



```
                             this._massageObject(query),
                             fields,
                             limit,
                             skip,
                             batchSize,
                             options || this.getQueryOptions());

    {
        const session = this.getDB().getSession();

        const readPreference = session._getSessionAwareClient().getReadPreference(session);
        if (readPreference !== null) {
            cursor.readPref(readPreference.mode, readPreference.tags);
        }

        const readConcern = session._getSessionAwareClient().getReadConcern(session);
        if (readConcern !== null) {
            cursor.readConcern(readConcern.level);
        }
    }

    return cursor;
}
> exit
bye

F:\iwpda12\iwp>node app.js
server listening at port 3000
(node:1116) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser:
true } to MongoClient.connect.
(node:1116) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover an
d Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
connection succeeded
Your bn has been saved!
123
4
{ _id: 5f903d0d9e534943a0f4517e, account: '123', amount: 7512, __v: 0 }
123
1000
{ _id: 5f903d0d9e534943a0f4517e, account: '123', amount: 6512, __v: 0 }
```