

Internet and Web Programming
CSE3002
Fall Semester 2020-21

Lab Assignment 12

ISHAAN OHRI
18BCE0265

Question:

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.

Code:

index.js (Backend)

```
const express = require('express');
const mongoose = require('mongoose');
const path = require('path');

mongoose.connect(
  'mongodb+srv://Ishaan0hri:hello123@urls.7gw6g.gcp.mongodb.net/Bank?retryWrites=true&w=majority',
  { useNewUrlParser: true, useUnifiedTopology: true, useCreateIndex: true },
  (error) => {
    if (error) {
      console.log(`Database connection failed\n${error}`);
      return;
    }

    console.log('Database connection successful');
  }
);

const app = express();

app.use(express.json());
app.use(
  express.static(path.join(__dirname, './'), {
    index: 'index.html'
  })
);
```

```

    })
  );

const AccountSchema = new mongoose.Schema({
  account: {
    type: Number,
    required: true
  },
  balance: {
    type: Number,
    required: true
  }
});

const Account = mongoose.model('Account', AccountSchema);

app.post('/debit', async (req, res) => {
  const { account, amount } = req.body;

  const user = await Account.findOne({ account });

  if(user.balance < amount){
    res.send({
      success: false,
      message: 'Insufficient balance'
    });
    return;
  }

  user.balance -= amount;

  await user.save();

  res.send({
    success: true,
    message: `Balance deducted successfully. New balance is Rs ${user.balance}`
  });
});

app.post('/create', async (req, res) => {
  const { account, amount } = req.body;

  const user = new Account({
    account,
    balance: amount
  })

  await user.save();

  res.send({
    success: true
  });
});

```

```

    });
});

const PORT = 4000;

app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`);
});

```

index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ishaan Ohri Bank</title>
  <link rel="stylesheet" type="text/css" href="./style.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
<body>
  <div>
    <p align="center">DEBIT FORM</p>
    <p align="center">Enter Account Number:</p>
    <input id="account" type="number" required>
    <br>
    <p align="center">Enter Amount to debit:</p>
    <input id="amount" type="number" required>
    <br>
    <p align="center"><button id="submit">Submit</button></p>
  </div>
</body>
<script src="app.js"></script>
</html>

```

style.css

```

div{
  padding: 40px;
  background-color: white;
  margin: 0;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
input{

```

```
font-family: Georgia, 'Times New Roman', Times, serif;
font-size: 30px;
margin: 20px;
}
body{
background-color: coral;
font-family: Georgia, 'Times New Roman', Times, serif;
font-size: 30px;
}
```

app.js

```
$(document).ready(function() {
    $('#submit').click(function(){
        var account = $('#account').val();
        var amount = $('#amount').val();

        var settings = {
            "url": "http://localhost:4000/debit",
            "method": "POST",
            "timeout": 0,
            "headers": {
                "Content-Type": "application/json"
            },
            "data": JSON.stringify({"account":account,"amount":amount}),
        };

        $.ajax(settings).done(function (response) {
            alert(response.message)
        });

    })
})
```

Output:

ISHAAN OHRI 18BCE0265

DEBIT FORM

Enter Account Number:

Enter Amount to debit:

Submit

cloud.mongodb.com

ISHAAN OHRI Bank

Access Manager Support Billing

URL-Shortener

Clusters

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

URLs

Overview Real Time Metrics Collections Profiler Performance Advisor Online Archive BETA Command Line Tools

DATABASES: 2 COLLECTIONS: 3

+ Create Database

Q NAMESPACES

Bank

accounts

URL-Shortener

Bank.accounts

COLLECTION SIZE: 57B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER {"filter":"example"} Find Reset

QUERY RESULTS 1-1 OF 1

```
{
  "_id": ObjectId("5f86da17ea5d45bd254de2b"),
  "account": 987654321,
  "balance": 4800,
  "__v": 0
}
```

Feature Requests

System Status: All Good



