Web Application Programming

(CS472)

June 2021

Final Examination

SCI Question (5 points)

Write one main point (1 - 2 sentences) on any aspect of *Node.js*, along with a corresponding SCI point. Do not use any of the main points from the Main Point Charts in the slides. (Use some other SCI point than “Do Less and Accomplish More.”)

Instructions

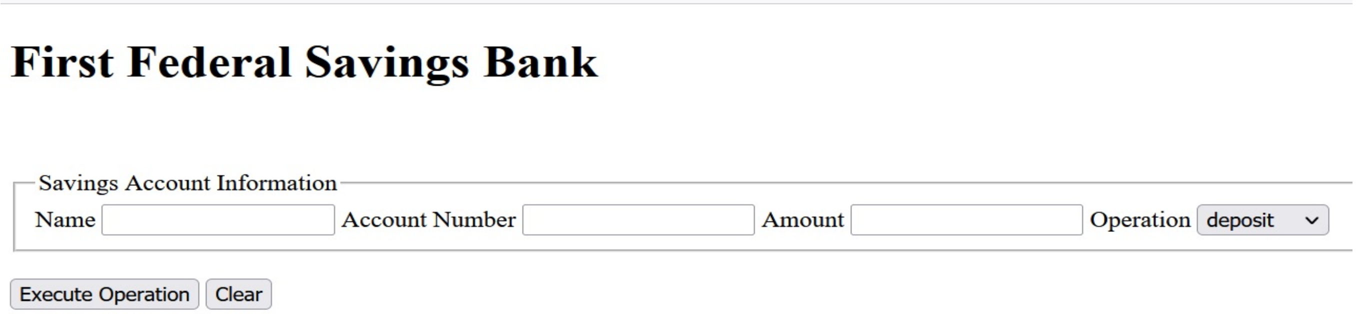
This is an Open Book exam. You may use any books or paper reference materials, or any electronic reference materials, including the coures slides.

However, you may not consult with any other person. You may not send or receive any email. You may not send or receive text messages. You must work on the Exam by yourself. You have exactly two hours to complete your work.

Consider a Web Page used by Bank Tellers to update *Savings Account* information for their customers. For each Savings Account, the following information is recorded in a database:

- owner's name
- account number
- current balance

After any transaction on one of the Savings Accounts, the Web Page is used to update the database. Following is screen shot of the Web Page:



First Federal Savings Bank

Savings Account Information

Name

Account Number

Amount

Operation deposit

There are four possible Operations:

- deposit money
- withdraw money
- create a new account
- get the current balance

Your job is to implement this web page using Nodejs Javascript, HTML, and MySQL. We recommend that you also use the *Express* frameework, but this is not a requirement. You should create a folder containing the following two files:

```
index.html
SavingsServer.js
```

Your `SavingsServer.js` file will be executed in a Command Window as follows:

```
> node SavingsServer.js
```

Then the following URL will be put into the Web Browser: http://localhost:5000/

This should bring up the above web page for First Federal Savings Bank. The User may then enter information about any Savings Account, select an operation from the drop-down list, and click the *Execute Operation* button. Your `SavingsServer.js` code should receive the http request from the Browser and carry out the selected operation on the Savings Account database stored in MySQL.

To help you to finish within two hours, we provide you with the `index.html` file. You may modify this file if you wish. If you are not using the *Express* framework, you probably will want to change the form method from POST to GET. Here is a listing of the `index.html` file:

```
<!DOCTYPE html>
  <html>
    <head><meta charset="utf-8"/>
      <title>My Savings Account</title>
    </head>
    <body>
      <h1>First Federal Savings Bank</h1>
      <br>
      <br>
      <form action = "http://localhost:5000/" method="POST">
        <div>
          <fieldset>
            <legend>Savings Account Information</legend>
            Name
            <input type = "text" name="user"/>
            Account Number
            <input type = "text" name="account"/>
            Amount
            <input type = "text" name="amount"/>
            Operation
            <select name="operation">
              <option>deposit</option>
              <option>withdraw</option>
              <option>balance</option>
              <option>new</option>
             </select>
           </fieldset>
           <br>
           <input type="submit" value="Execute Operation"/>
           <input type="reset" value= "Clear"
         </div>
       </form>
     </body>
   </html>
```

The *balance* operation displays a message to the User like the following:

> The Current Balance is $5000.67

The *deposit* operation adds the specified amount to the account balance. Similarly, the *withdraw* operation subtracts the specified amount from the account balance. The *new* operation adds a new row to the Accounts table using the information in the form provided by the User.

To simplify the coding, you may assume that the User always enters valid data in the form. Thus, data validation is not necessary. Also, assume the Account Number for each account is unique.

To create the Savings Account table in MySQL and populate it with sample data for testing, execue the following script in the MySQL Workbench query window:

```
DROP DATABASE IF EXISTS Bank;
CREATE DATABASE Bank;
USE Bank;
DROP TABLE IF EXISTS Accounts;

CREATE TABLE Accounts (
       name VARCHAR(50),
       number INT,
       balance DEC(8,2),
       PRIMARY KEY (number)
);

INSERT INTO Accounts VALUES ('George Smith', 3344, 1000.00);
INSERT INTO Accounts VALUES ('John Jones', 1234, 2000.00);
INSERT INTO Accounts VALUES ('Adam Miller', 1900, 3000.00);
INSERT INTO Accounts VALUES ('Sally Fields', 3666, 700.00);
INSERT INTO Accounts VALUES ('Oliver Hardy', 5555, 90000.00);
INSERT INTO Accounts VALUES ('Sam Holidays', 6666, 45600.00);
INSERT INTO Accounts VALUES ('Rachel Greene', 300550, 39000.00);
INSERT INTO Accounts VALUES ('Burt Barnow', 98333, 7008.00);


SELECT * FROM Accounts;
```

When you are finished, zip the folder containing your files, and write it into the Microsoft Teams assignment called Final Examinaion.. Also, include a file containing your answer to the SCI Question. Do not use GitHub to submit your files.