Develop Simple Web Application with Node.js and MySQL

Backend

Express, Node.js, MySql

Frontend

HTML, Javascript, Angular

Develop a Simple Web Application with Node.js and MySql

Purpose: Understand how to develop web application using node.js, mysql, express, html etc.

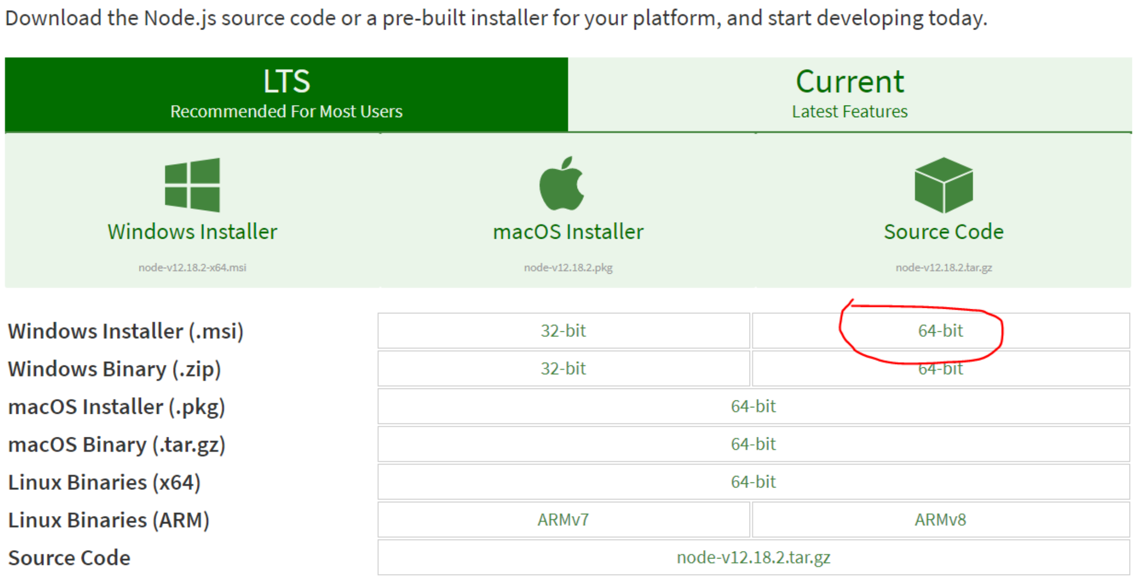
Backend: Express, Node.js, MySql

Frontend: HTML, Javascript, Angular

Build a development environment

1. Install Node.js

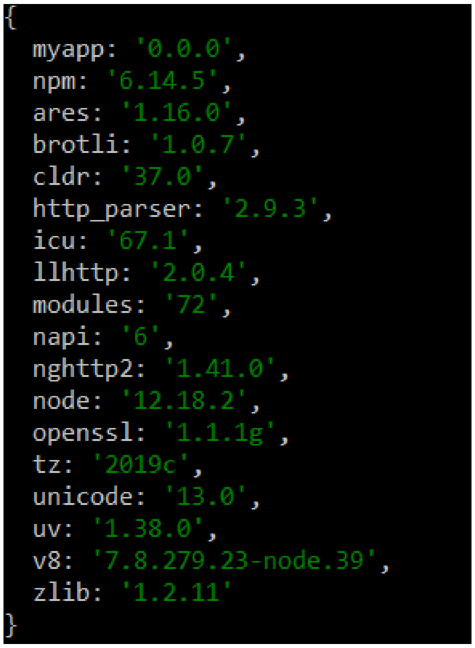
https://nodejs.org/en/download/



Download the Node.js source code or a pre-built installer for your platform, and start developing today.

Windows Installer (.msi)	32-bit	[✓ 64-bit]
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.18.2.tar.gz	

After install, you can try “npm version”



```
{
  myapp: '0.0.0',
  npm: '6.14.5',
  ares: '1.16.0',
  brotli: '1.0.7',
  cldr: '37.0',
  http_parser: '2.9.3',
  icu: '67.1',
  llhttp: '2.0.4',
  modules: '72',
  napi: '6',
  nghttp2: '1.41.0',
  node: '12.18.2',
  openssl: '1.1.1g',
  tz: '2019c',
  unicode: '13.0',
  uv: '1.38.0',
  v8: '7.8.279.23-node.39',
  zlib: '1.2.11'
}
```

2. Install MySql

   It recommends to install MySql 5.7.x community Edition.

   https://downloads.mysql.com/archives/get/p/25/file/mysql-installer-community-5.7.30.0.msi

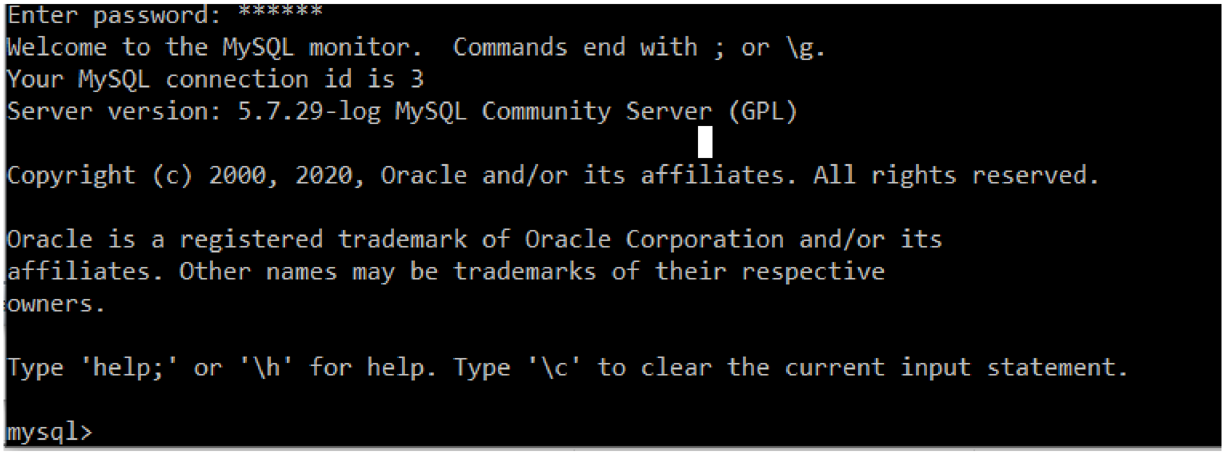
   When installing it, make sure the password for root is default as “123456”.

   You can install MySql Workbench to view data from your MySQL

   https://downloads.mysql.com/archives/get/p/8/file/mysql-workbench-community-6.3.10-winx64.msi

   You can use “Command Prompt” cmd to test your installation:

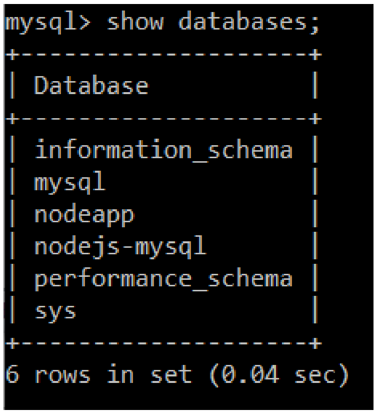
   >mysql –u root –p



```
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.29-log MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

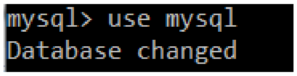
Enter password “123456”, then you will enter the system.

mysql> show database;



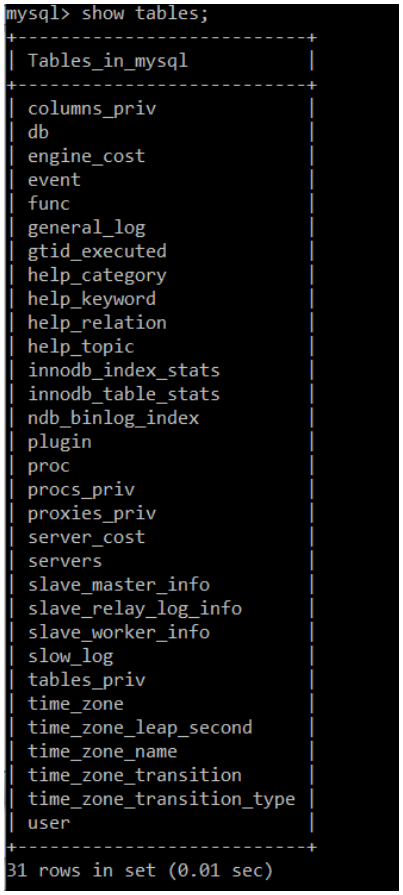
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| nodeapp            |
| nodejs-mysql       |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.04 sec)
```

If you want to see inside of which database, just use command “use”.



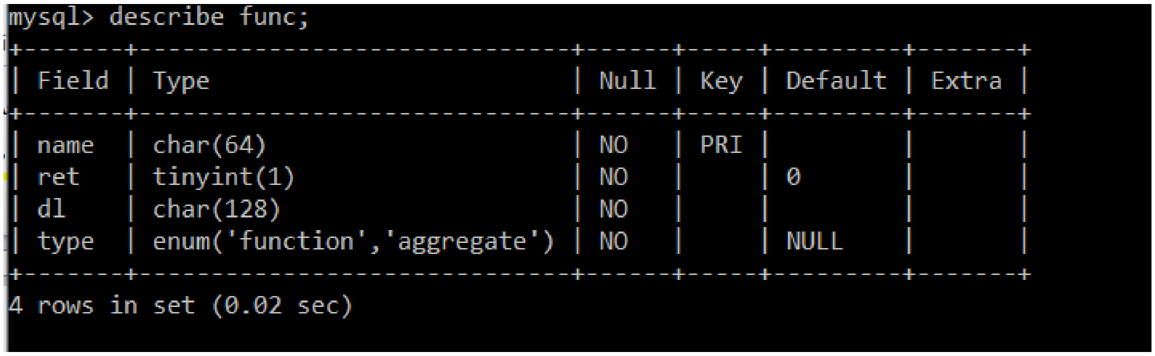
```
mysql> use mysql
Database changed
```

How many tables in the database “mysql”:



```
mysql> show tables;
+---------------------------+
| Tables_in_mysql           |
+---------------------------+
| columns_priv              |
| db                        |
| engine_cost               |
| event                     |
| func                      |
| general_log               |
| gtid_executed             |
| help_category             |
| help_keyword              |
| help_relation             |
| help_topic                |
| innodb_index_stats        |
| innodb_table_stats        |
| ndb_binlog_index          |
| plugin                    |
| proc                      |
| procs_priv                |
| proxies_priv              |
| server_cost               |
| servers                   |
| slave_master_info         |
| slave_relay_log_info      |
| slave_worker_info         |
| slow_log                  |
| tables_priv               |
| time_zone                 |
| time_zone_leap_second     |
| time_zone_name            |
| time_zone_transition      |
| time_zone_transition_type |
| user                      |
+---------------------------+
31 rows in set (0.01 sec)
```

What's structure of table "func":



```
mysql> describe func;
+-------+------------------------------+------+-----+---------+-------+
| Field | Type                         | Null | Key | Default | Extra |
+-------+------------------------------+------+-----+---------+-------+
| name  | char(64)                     | NO   | PRI |         |       |
| ret   | tinyint(1)                   | NO   |     | 0       |       |
| dl    | char(128)                    | NO   |     |         |       |
| type  | enum('function','aggregate') | NO   |     | NULL    |       |
+-------+------------------------------+------+-----+---------+-------+
4 rows in set (0.02 sec)
```

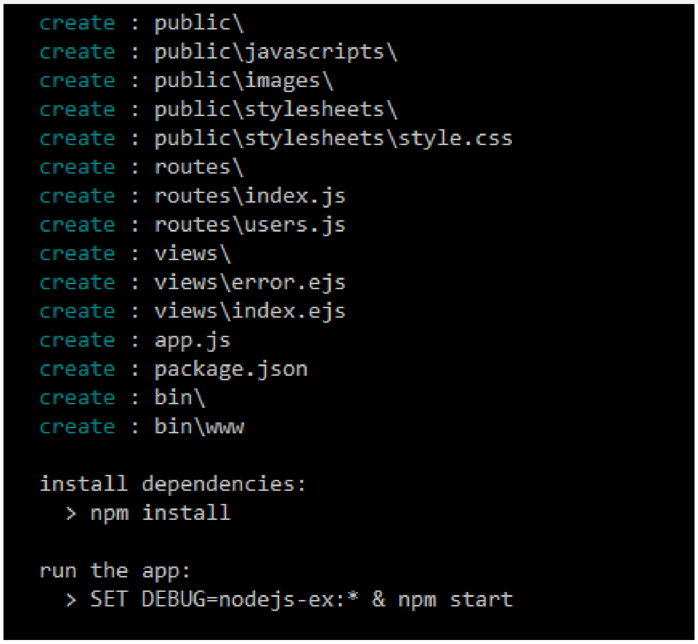
3. Install Express

https://codingstatus.com/how-to-install-express-application-using-express-generator-tool/

Create a project folder, for example "myapp"

Enter this folder

```
npm install -g express-generator
npx express --view=ejs
```

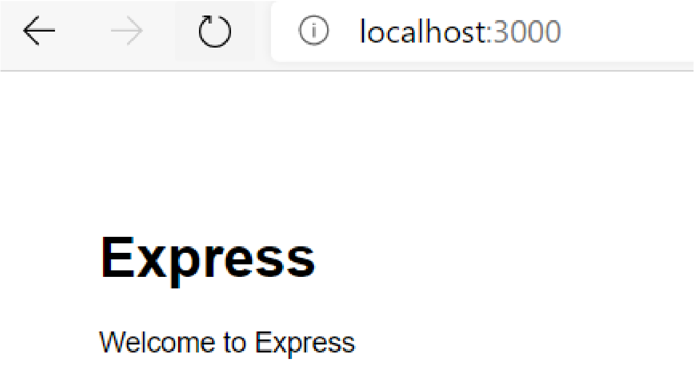


```
  create : public\
  create : public\javascripts\
  create : public\images\
  create : public\stylesheets\
  create : public\stylesheets\style.css
  create : routes\
  create : routes\index.js
  create : routes\users.js
  create : views\
  create : views\error.ejs
  create : views\index.ejs
  create : app.js
  create : package.json
  create : bin\
  create : bin\www

  install dependencies:
    > npm install

  run the app:
    > SET DEBUG=nodejs-ex:* & npm start
```

You can test your installation:

Run “npm start”

Then in the browser http://localhost:3000. You will see



Express

Welcome to Express

Practice 1

How to Insert Form Data Into the Table Using Node.js and MySQL

https://codingstatus.com/how-to-insert-form-data-into-the-table-using-node-js-and-mysql/

1. Create a database – “NodeApp” with a table of “Users”

(a)Create a txt file – createDBTable.txt

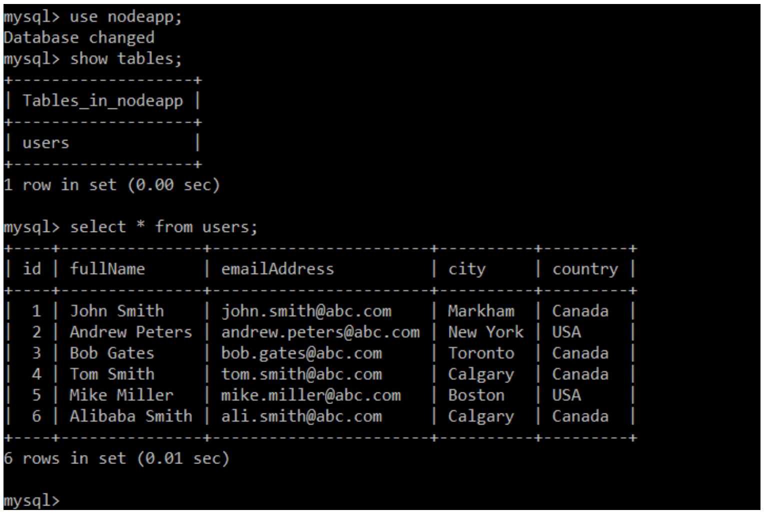
```
 drop database if exists NodeApp;
 create database NodeApp CHARACTER SET utf8 COLLATE utf8_general_ci;
 use NodeApp;
CREATE TABLE IF NOT EXISTS USERS (id int(10) UNSIGNED PRIMARY KEY NOT NULL
 AUTO_INCREMENT,
            fullName varchar(255) DEFAULT NULL,
            emailAddress varchar(255) DEFAULT NULL,
            city varchar(255) DEFAULT NULL,
           country varchar(50) DEFAULT NULL

 ) DEFAULT CHARSET=utf8;
```

(b) Login to mysql

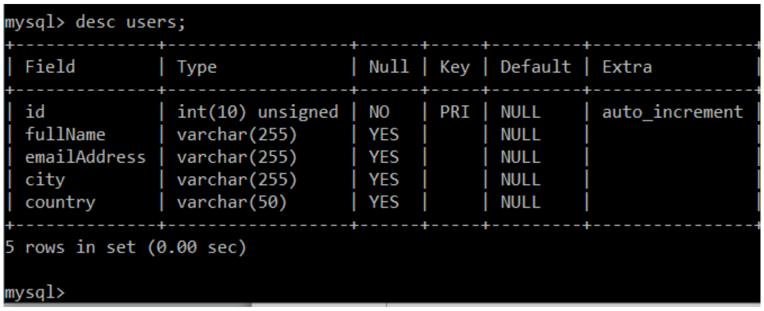
```
>>  mysql –u root –p < createDBTable.txt
```

Enter password: 123456

You can view database and table. This time you should not see any data inside table.



```
mysql> use nodeapp;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_nodeapp |
+-------------------+
| users             |
+-------------------+
1 row in set (0.00 sec)

mysql> select * from users;
+----+---------------+-----------------------+----------+---------+
| id | fullName      | emailAddress          | city     | country |
+----+---------------+-----------------------+----------+---------+
|  1 | John Smith    | john.smith@abc.com    | Markham  | Canada  |
|  2 | Andrew Peters | andrew.peters@abc.com | New York | USA     |
|  3 | Bob Gates     | bob.gates@abc.com     | Toronto  | Canada  |
|  4 | Tom Smith     | tom.smith@abc.com     | Calgary  | Canada  |
|  5 | Mike Miller   | mike.miller@abc.com   | Boston   | USA     |
|  6 | Alibaba Smith | ali.smith@abc.com     | Calgary  | Canada  |
+----+---------------+-----------------------+----------+---------+
6 rows in set (0.01 sec)

mysql>
```



```
mysql> desc users;
+--------------+------------------+------+-----+---------+----------------+
| Field        | Type             | Null | Key | Default | Extra          |
+--------------+------------------+------+-----+---------+----------------+
| id           | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| fullName     | varchar(255)     | YES  |     | NULL    |                |
| emailAddress | varchar(255)     | YES  |     | NULL    |                |
| city         | varchar(255)     | YES  |     | NULL    |                |
| country      | varchar(50)      | YES  |     | NULL    |                |
+--------------+------------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql>
```

2. Create database.js to connect databse. Database.js will be located in the folder of “myapp”.

```
var mysql = require('mysql');
var conn = mysql.createConnection({
 host: 'localhost', // Replace with your host name
 user: 'root',      // Replace with your database username
 password: '123456',      // Replace with your database password
 database: 'nodeapp' // // Replace with your database Name
 });
 conn.connect(function(err) {
 if (err) throw err;
   console.log('Database is connected successfully !');
 });
 module.exports = conn;
```

3. Create an HTML Form in Node.js App

Now, Use the following HTML code in the `views/user.ejs` to insert data using Node.js & MySQL.

File Name: user.ejs

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
* {
  box-sizing: border-box;}
.user-detail {
    height: 100vh;
    border: 2px solid #f1f1f1;
    padding: 16px;
    background-color: white;
    width: 30%;}
input{
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;}
input[type=text]:focus, input[type=password]:focus {
    background-color: #ddd;
    outline: none;}
button[type=submit] {
    background-color: #434140;
    color: #ffffff;
    padding: 10px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%;
    opacity: 0.9;
    font-size: 20px;}
label{
  font-size: 18px;;}
button[type=submit]:hover {
  background-color:#3d3c3c;}
</style>
</head>
<body>
<!--====form section start====-->
<div class="user-detail">
    <h2>Create User Data</h2>
    <form action="/user/create" method="POST">
          <label>Full Name</label>
          <input type="text" placeholder="Enter Full Name"
name="fullName" required>
          <label>Email Address</label>
          <input type="email" placeholder="Enter Email Address"
name="emailAddress" required>
          <label>City</label>
          <input type="city" placeholder="Enter Full City" name="city"
required>
          <label>Country</label>
          <input type="text" placeholder="Enter Full Country"
name="country" required>
          <button type="submit">Submit</button>
        </div>
</div>
<!--====form section start====-->
</body>
</html>
```

This form page will open through `http://localhost:3000/user` URL.

Use the following script in the `routes/user.js` file and lo to load the HTML form.

```
router.get('/', function(req, res, next) {
res.render('user');
});
```

You must declare the following points:

- Form method must be `POST` like `method="POST"`
- Form action contains `user/create/` (It is created in `routes/user.js` to post form data) like `action="user/create"`.
- You should declare field name the same as the column name of the `users` table.

Form Field	**Field Name**
Full Name	name=”fullName”
Email Address	name=”emailAddress”
City	name=”city”
Country	name=”country”

4. Write MySQL Query in Node.js to Insert Data

First of all, Include the database connection file in the `routes/user.js` file

```
var db=require('../database');
```

Complete Script: **user.js**

```
var express = require('express');
var router = express.Router();
var db=require('../database');

router.get('/', function(req, res, next) {
res.render('user');

});

router.post('/create', function(req, res, next) {

  // store all the user input data
  const userDetails=req.body;

  // insert user data into users table
  var sql = 'INSERT INTO users SET ?';
  db.query(sql, userDetails,function (err, data) {
      if (err) throw err;
         console.log("User dat is inserted successfully ");
  });
 res.redirect('/user');  // redirect to user form page after inserting the data

});
module.exports = router;
```

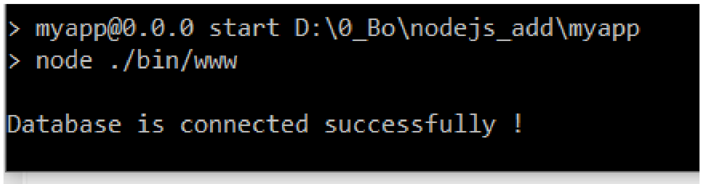
Include the user.js file in app.js root file

You have to include `user.js` route file in `app.js` the root file as.

```
var userRouter = require('./routes/user');

app.use('/user',userRouter);
```

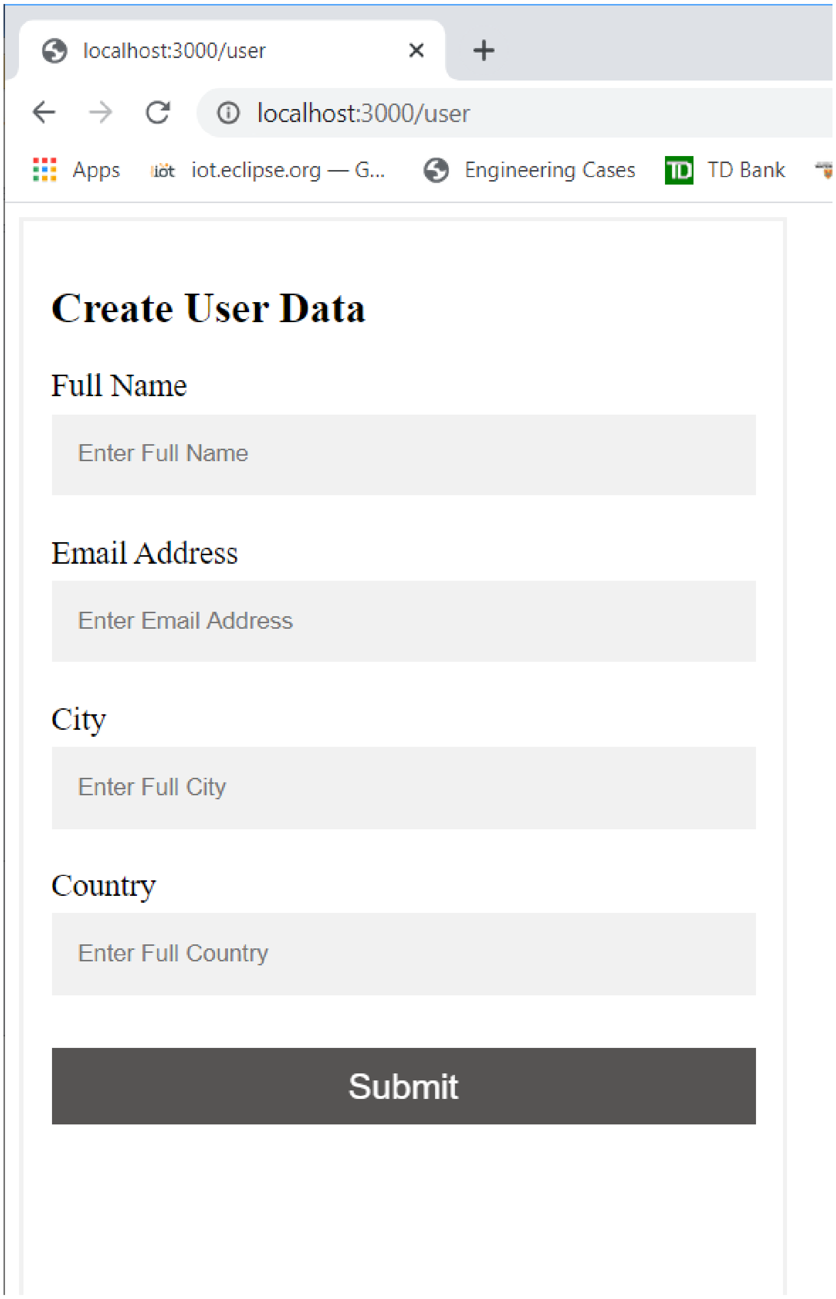
5. Start nodejs server

>>npm start



```
> myapp@0.0.0 start D:\0_Bo\nodejs_add\myapp
> node ./bin/www

Database is connected successfully !
```

Apply Web Browser



Create User Data

Full Name

Enter Full Name

Email Address

Enter Email Address

City

Enter Full City

Country

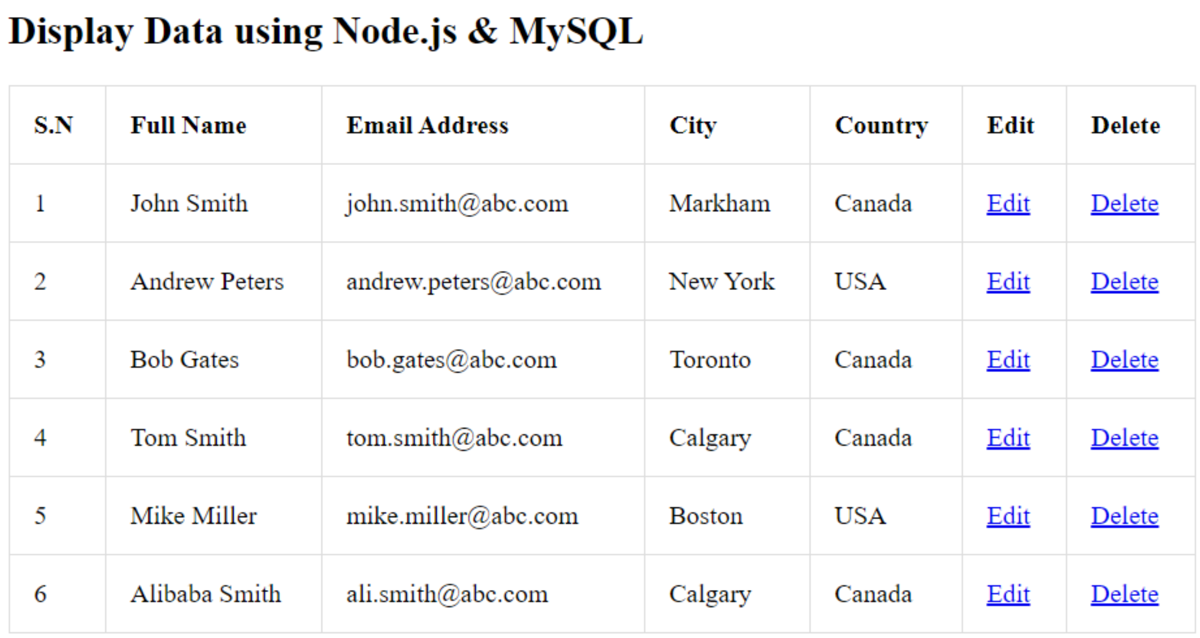
Enter Full Country

Submit

Practice 2

How to display Data from MySQL database table in Node.js

This practice will display all data added into the database in Practice 1.



Display Data using Node.js & MySQL

S.N	Full Name	Email Address	City	Country	Edit	Delete
1	John Smith	john.smith@abc.com	Markham	Canada	Edit	Delete
2	Andrew Peters	andrew.peters@abc.com	New York	USA	Edit	Delete
3	Bob Gates	bob.gates@abc.com	Toronto	Canada	Edit	Delete
4	Tom Smith	tom.smith@abc.com	Calgary	Canada	Edit	Delete
5	Mike Miller	mike.miller@abc.com	Boston	USA	Edit	Delete
6	Alibaba Smith	ali.smith@abc.com	Calgary	Canada	Edit	Delete

1. **Create the folder of nodejs_display**
2. **cd nodejs_display**
3. **Install Express as described in Step 3 of “build development environment”**
4. **Create database.js to connect databse. Database.js will be located in the folder of “nodejs_display”.**

```
var mysql = require(‘mysql');

     var conn = mysql.createConnection({
      host: 'localhost', // Replace with your host name
      user: 'root',      // Replace with your database username
     password: '123456',      // Replace with your database password
     database: 'nodeapp' // // Replace with your database Name
     });
     conn.connect(function(err) {
     if (err) throw err;
       console.log('Database is connected successfully !');
     });
module.exports = conn;
```

5. Write MySQL Query in Node.js to Fetch Data

Before Displaying MySQL data, You have to do the following things

- Include the database connection file `database.js` in the `routes/user.js` file.

```
var db=require('../database');
```

- Write the following script in the `routes/user.js` file to fetch data from the MySQL `users` table.

```
router.get('/user-list', function(req, res, next) {
var sql='SELECT * FROM users';
db.query(sql, function (err, data, fields) {
if (err) throw err;
res.render('user-list', { title: 'User List', userData: data});
});
});
```

- Include `routes/user.js` file in the root file `app.js`

```
var userRouter = require('./routes/user');
app.use('/user',userRouter);
```

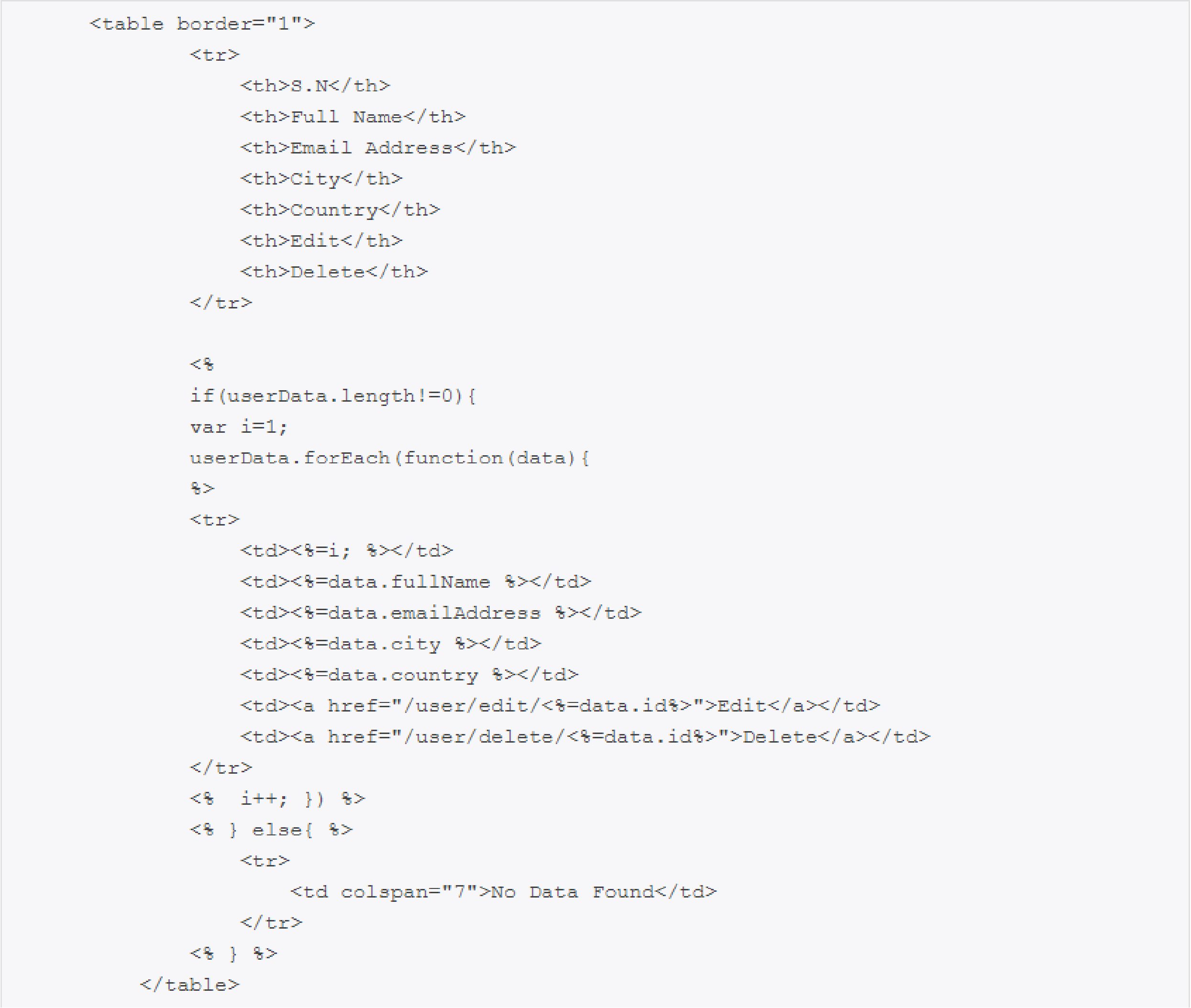
Complete Script: routes/user.js

```
var express = require('express');
var router = express.Router();
var db=require('../database');
// another routes also appear here
// this script to fetch data from MySQL databse table
router.get('/user-list', function(req, res, next) {
var sql='SELECT * FROM users';
db.query(sql, function (err, data, fields) {
if (err) throw err;
res.render('user-list', { title: 'User List', userData: data});
});
});
module.exports = router;
```

6. Create HTML Table & Display data using Node.js

Create a file `user-list.ejs` in the `views` folder.

Now, Use the following `HTML` code and `ejs` Script to display data in the HTML table.



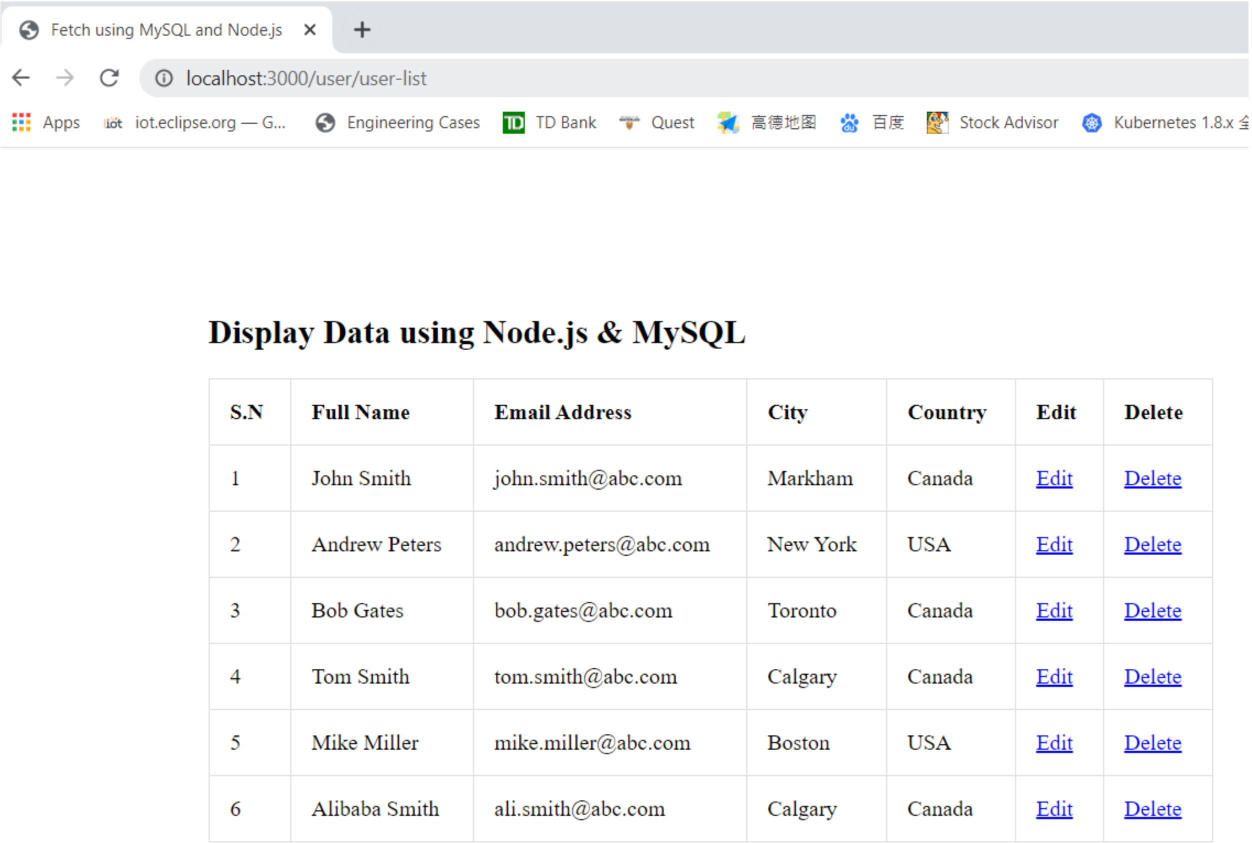
```
<table border="1">
        <tr>
            <th>S.N</th>
            <th>Full Name</th>
            <th>Email Address</th>
            <th>City</th>
            <th>Country</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>

        <%
        if(userData.length!=0){
        var i=1;
        userData.forEach(function(data){
        %>
        <tr>
            <td><%=i; %></td>
            <td><%=data.fullName %></td>
            <td><%=data.emailAddress %></td>
            <td><%=data.city %></td>
            <td><%=data.country %></td>
            <td><a href="/user/edit/<%=data.id%>">Edit</a></td>
            <td><a href="/user/delete/<%=data.id%>">Delete</a></td>
        </tr>
        <%  i++; }) %>
        <% } else{ %>
            <tr>
                <td colspan="7">No Data Found</td>
            </tr>
        <% } %>
    </table>
```

Complete HTML Table Code: views/user-list.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Fetch using MySQL and Node.js</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <style>
    table, td, th {
      border: 1px solid #ddd;
      text-align: left;
    }
    table {
      border-collapse: collapse;
      width: 50%;


    }
    .table-data{
        position: relative;
      left:150px;
      top:100px;
    }
    th, td {
      padding: 15px;
    }

    </style>
</head>
<body>
    <div class="table-data">
<h2>Display Data using Node.js & MySQL</h2>
    <table border="1">
        <tr>
            <th>S.N</th>
            <th>Full Name</th>
            <th>Email Address</th>
            <th>City</th>
            <th>Country</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>

        <%
        if(userData.length!=0){
        var i=1;
        userData.forEach(function(data){
        %>
        <tr>
            <td><%=i; %></td>
            <td><%=data.fullName %></td>
            <td><%=data.emailAddress %></td>
            <td><%=data.city %></td>
            <td><%=data.country %></td>
            <td><a href="/user/edit/<%=data.id%>">Edit</a></td>
            <td><a
href="/user/delete/<%=data.id%>">Delete</a></td>
        </tr>
        <%  i++; }) %>
        <% } else{ %>
            <tr>
                <td colspan="7">No Data Found</td>
            </tr>
        <% } %>
    </table>
    </div>
</body>
</html>
```

You can see the above table with displaying data from `users` table by entering the following URL in the browser

```
http://localhost:3000/user/user-list
```



Display Data using Node.js & MySQL

S.N	Full Name	Email Address	City	Country	Edit	Delete
1	John Smith	john.smith@abc.com	Markham	Canada	Edit	Delete
2	Andrew Peters	andrew.peters@abc.com	New York	USA	Edit	Delete
3	Bob Gates	bob.gates@abc.com	Toronto	Canada	Edit	Delete
4	Tom Smith	tom.smith@abc.com	Calgary	Canada	Edit	Delete
5	Mike Miller	mike.miller@abc.com	Boston	USA	Edit	Delete
6	Alibaba Smith	ali.smith@abc.com	Calgary	Canada	Edit	Delete

Reference

1) https://codingstatus.com/how-to-insert-form-data-into-the-table-using-node-js-and-mysql/
2) https://codingstatus.com/how-to-display-data-from-mysql-database-table-in-node-js/
3) https://github.com/codethereforam/express-mysql-demo
4) https://codingstatus.com/how-to-insert-data-into-mongodb-using-mongoose-and-node-js/
5) https://codingstatus.com/delete-mongodb-data-using-mongoose/
6) https://codingstatus.com/update-mongodb-data-using-mongoose/
7) https://www.youtube.com/watch?v=YYEC7ydDj4k
8) https://www.simplilearn.com/nodejs-tutorial-article