

4.

Node JS, Express JS and MongoDB

- i) Create a web application with username in HTML and node.js using express framework to handle different types of HTTP request namely get, post, put, options and delete.
- ii) Provide different route paths for each of the request.
- iii) Display the different request types along with the username in the browser when the application is executed.

Code:

node js code :

```
const express=require('express');
let ejs=require('ejs');
const app=express();

app.get('/',(req,res)=>{
    res.send('welcome to home page');
});

app.put('/put, function (req, res) {
    res.send('Got a PUT request');
});

app.delete('/delete', function (req, res) {
    res.send('Got a DELETE request ');
})

app.get('/user',(req,res)=>{
    res.render('form.ejs');
});

app.post('/submitUsrDetails',(req,res)=>{
    console.log(res.body);
    res.send('logged in');
});

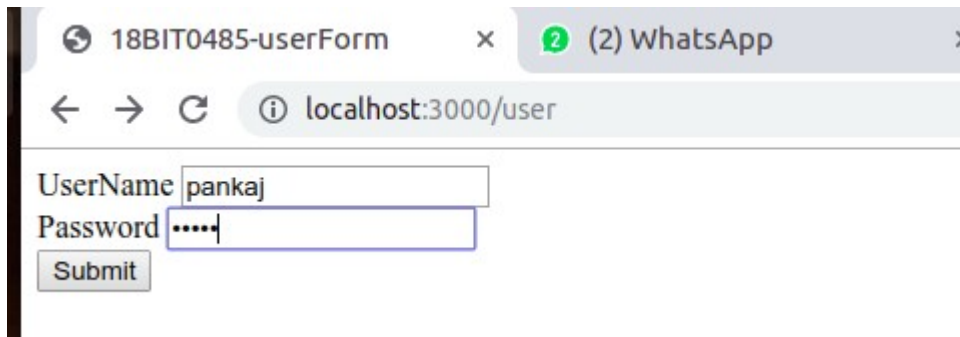
app.listen(3000,(err)=>{
    if (err) console.log(err);
    else{
        console.log("listening on port 3000");
    }
});
```

html code:

```
<html>
  <head><title>18BIT0485-userForm</title></head>
  <body>
    <form action="/submitUsrDetails" method="POST">
      <label name="userId">UserName</label>
      <input type="text" name="userId" value="">
      <br>
      <label name="password">Password</label>
      <input type="password" name="password" ><br>
      <button type="submit" name="submit">Submit</button>
    </form>
  </body>
</html>
```

Output:

form entry:

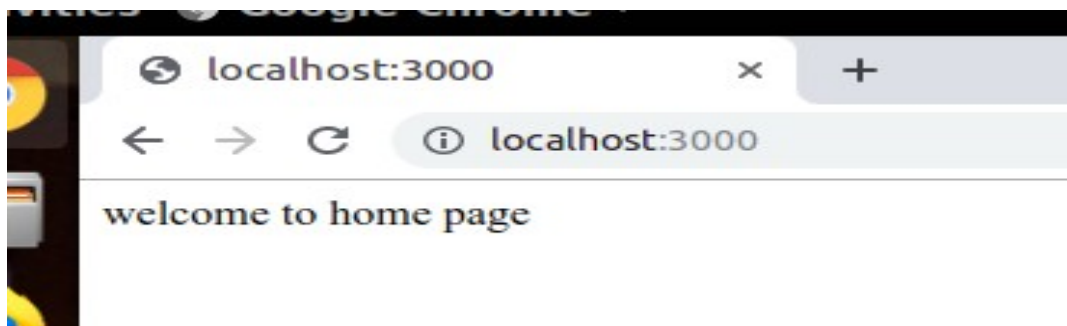
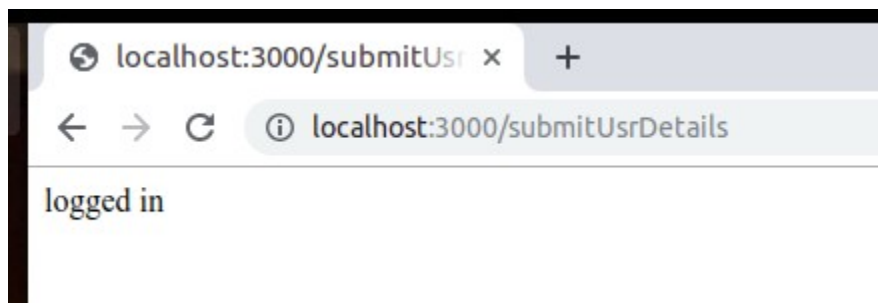


18BIT0485-userForm x (2) WhatsApp x

localhost:3000/user

UserName

Password



- 5.i) Design an application using node.js and mongodb to perform the following operations in a student collection with name, age, DOB and year of admission.
- ii) Insert multiple records into the collection.
 - iii) Sort all documents in the student collection
 - iv) Update the document of student with name='Kevin' to age=25
 - v) Limit the result to 4 documents
 - vi) Query the document on the basis of age>25

soln

code :

i)

node js base code:

```
const express=require('express');
let ejs=require('ejs');
const app=express();
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/MongoDatabase";
app.get('/createDb',(req,res)=>{

    // create database and collection on mongo db using node js
    MongoClient.connect(url, function(err, db) {
        if (err) throw err;
        var dbo = db.db("mydb");
        dbo.createCollection("studentRecords", function(err, res) {
            if (err) throw err;
            console.log("Collection is created!");
            res.send("collection is created");
            db.close();
        });
    });
});
```

Ouput:

```
leo@leo-Inspiron-3558:~/fat study/webStudy$ node mongo.js
listening on 3000
(node:25561) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
Collection is created!
```

ii)insert one and multiple

```
app.get('/insertOne',(req,res)=>{
```

```
    // insert one record into database
```

```

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var myobj = { name: "Pankaj Shukla", age: "18", dob: "19-JUL-2000", yoa: "2018" };
  dbo.collection("studentRecords").insertOne(myobj, function(err, res) {
    if (err) throw err;
    res.send("1 record inserted");
    console.log("1 record inserted");
    dbo.close();
  });
});
});

```

Insert many

```

app.get('/insertMany',(req,res)=>{
  // insert many records at once
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    var myobj = [{ name: "Pankaj Shukla", age: "18", dob: "19-jul-2000", yoa: "2018" },
      { name: "Somya Sethi", age: "25", dob: "18-aug-2000", yoa: "2018"},
      { name: "Kevin", age: "20", dob: "18-jan-2000", yoa: "2018"},
      { name: "david", age: "25", dob: "18-feb-2000", yoa: "2018"},
      { name: "Gunjan ", age: "20", dob: "18-sept-2000", yoa: "2017"},
      { name: "Senthil ", age: "20", dob: "17-oct-2000", yoa: "2018"},
    ];
    dbo.collection("studentRecords").insertMany(myobj, function(err, res) {
      if (err) throw err;

      console.log("multiple record inserted");
      db.close();
    });
    res.send("multiple record inserted");
  });
});
Output:

```

```

leo@leo-Inspiron-3558:~/fat study/webStudy$ node mongo.js
listening on 3000
(node:26829) DeprecationWarning: current Server Discovery and Monitoring
nd will be removed in a future version. To use the new Server Discover
ss option { useUnifiedTopology: true } to the MongoClient constructor.
1 record inserted
^C

```

```
leo@leo-Inspiron-3558:~/fat study/webStudy$ node mongo.js
listening on 3000
(node:27168) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
multiple record inserted
multiple record inserted
```

iii) Sort all documents in the student collection

```
app.get('/sort',(req,res)=>{
  // sort and then show in console
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    var mysort = { name: 1 };
    dbo.collection("studentRecords").find().sort(mysort).toArray(function(err, result) {
      if (err) throw err;
      console.log(result);
      db.close();

    });
    res.send('sorted');
  });
});
```

output:

```
listing on 3000
(node:29601) DeprecationWarning: current Server
nd will be removed in a future version. To use
ss option { useUnifiedTopology: true } to the
[ { _id: 5dbc424c71914770db5482ab,
  name: 'Gunjan ',
  age: '20',
  dob: '18-sept-2000',
  yoa: '2017' },
  { _id: 5dbc424c71914770db5482a9,
  name: 'Kevin',
  age: '25',
  dob: '18-jan-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482a7,
  name: 'Pankaj Shukla',
  age: '18',
  dob: '19-JUL-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482ac,
  name: 'Senthil ',
  age: '20',
  dob: '17-oct-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482a8,
  name: 'Somya Sethi',
  age: '25',
  dob: '18-aug-2000',
  yoa: '2018' },
```

iv) Update the document of student with name='Kevin' to age=25
code :

```
app.get('/updateOne',(req,res)=>{
  // updating the record
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    var myquery = { name: "Kevin" };
    var newvalues = { $set: {age:"25"} };
    dbo.collection("studentRecords").updateOne(myquery, newvalues, function(err, res) {
      if (err) throw err;
      console.log("1 document updated");
      db.close();
    });
  });
});
```

```
name: 'Somya Sethi',
age: '25',
dob: '18-aug-2000',
yoa: '2018' },
{ _id: 5dbc424c71914770db5482aa,
  name: 'david',
  age: '25',
  dob: '18-feb-2000',
  yoa: '2018' } ]
1 document updated
```

});
Output:

v) Limit the result to 4 documents

code:

```
app.get('/limit',(req,res)=>{
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    dbo.collection("studentRecords").find().limit(5).toArray(function(err, result) {
      if (err) throw err;
      console.log(result);
      db.close();
    });
  });
});
```

});
output:

```
[ { _id: 5dbc424c71914770db5482a7,
  name: 'Pankaj Shukla',
  age: '18',
  dob: '19-JUL-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482a8,
  name: 'Somya Sethi',
  age: '25',
  dob: '18-aug-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482a9,
  name: 'Kevin',
  age: '25',
  dob: '18-jan-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482aa,
  name: 'david',
  age: '25',
  dob: '18-feb-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482ab,
  name: 'Gunjan ',
  age: '20',
  dob: '18-sept-2000',
  yoa: '2017' } ]
```

vi) Query the document on the basis of age>25

code :

```
app.get('/query',(req,res)=>{
```

```
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("mydb");
    var query = { age: "25" };
    dbo.collection("studentRecords").find(query).toArray(function(err, result) {
      if (err) throw err;
      console.log(result);
      db.close();
    });
  });
```

```
});
```

output:

```
leo@leo-Inspiron-3558:~/fat study/webStudy$ node mongo.js
listening on 3000
(node:29456) DeprecationWarning: current Server Discovery and Monitoring engine
and will be removed in a future version. To use the new Server Discovery and
ss option { useUnifiedTopology: true } to the MongoClient constructor.
[ { _id: 5dbc424c71914770db5482a8,
  name: 'Somya Sethi',
  age: '25',
  dob: '18-aug-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482a9,
  name: 'Kevin',
  age: '25',
  dob: '18-jan-2000',
  yoa: '2018' },
  { _id: 5dbc424c71914770db5482aa,
  name: 'david',
  age: '25',
  dob: '18-feb-2000',
  yoa: '2018' } ]
```