**Name : Nayan Mandliya**

**Roll No. : 1911027**

**Batch : MERN 1**

**Experiment No. : 4**

**Title:** Implementation of CRUD operations with Mongoose.

**Problem statement:** A teacher wants to create a mongodb database to store the details of students. Details include roll no., name, division, year\_of\_study, branch, cgpa. Teacher wants a mongodb database to be made with the fields which are mentioned in the details as schema for the database. And also perform different operations specified below.

**Mongoose CRUD Operations:** Demonstrate the behavior of Create, Read, update and delete with the help of MongoDB

**1) Create and insert:**

**CODE:**

**Create.js:**

const express=require('express')

const app=express()

const mongoose=require('mongoose');

app.use(express.urlencoded({extended:false}));

app.use(express.json())

mongoose.connect('mongodb://localhost:27017/EXP4');

const db=mongoose.connection;

var StudentSchema=mongoose.Schema({

    roll\_no: Number,

    name: String,

    division: String,

    year\_of\_study: Number,

    branch: String,

    cgpa: Number

})

const path=require('path');

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

    res.sendFile(path.join(\_\_dirname, './public/create.html'));

})

app.post('/',(req,res)=>{

    var Student=mongoose.model('EXP4',StudentSchema,'Students')

    var s=new Student({

        roll\_no: req.body.roll\_no,

        name: req.body.name,

        division: req.body.division,

        year\_of\_study: req.body.year\_of\_study,

        branch: req.body.branch,

        cgpa: req.body.cgpa

    })

    s.save((err,stu)=>{

        if(err)

        {

            console.log(err);

        }

        else

        {

            console.log("Insertion successfull!!!");

            res.send("Insertion successfull!!!");

            res.end()

        }

    })

})

app.listen(5000,()=>{

    console.log('Server is listening on port 5000!!!!');

})

module.exports=mongoose.model('EXP4',StudentSchema,'Students');

**(model is exported from this file so that it can be used in another files as well)**

**Create.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form name="addstudent" method="post">

        <div class="input">

            <span class="label1">Roll No.:</span>

            <input type="number" name="roll\_no"/>

        </div>

        <div class="input">

            <span class="label1">Name:</span>

            <input type="text" name="name"/>

        </div>

        <div class="input">

            <span class="label1">Division:</span>

            <input type="text" name="division"/>

        </div>

        <div class="input">

            <span class="label1">Year of study:</span>

            <input type="number" name="year\_of\_study"/>

        </div>

        <div class="input">

            <span class="label1">Branch:</span>

            <input type="text" name="branch"/>

        </div>

        <div class="input">

            <span class="label1">CGPA:</span>

            <input type="number" name="cgpa"/>

        </div>

        <div class="actions">

          <input class="buttoner" type="submit" value="Store data"/>

        </div>

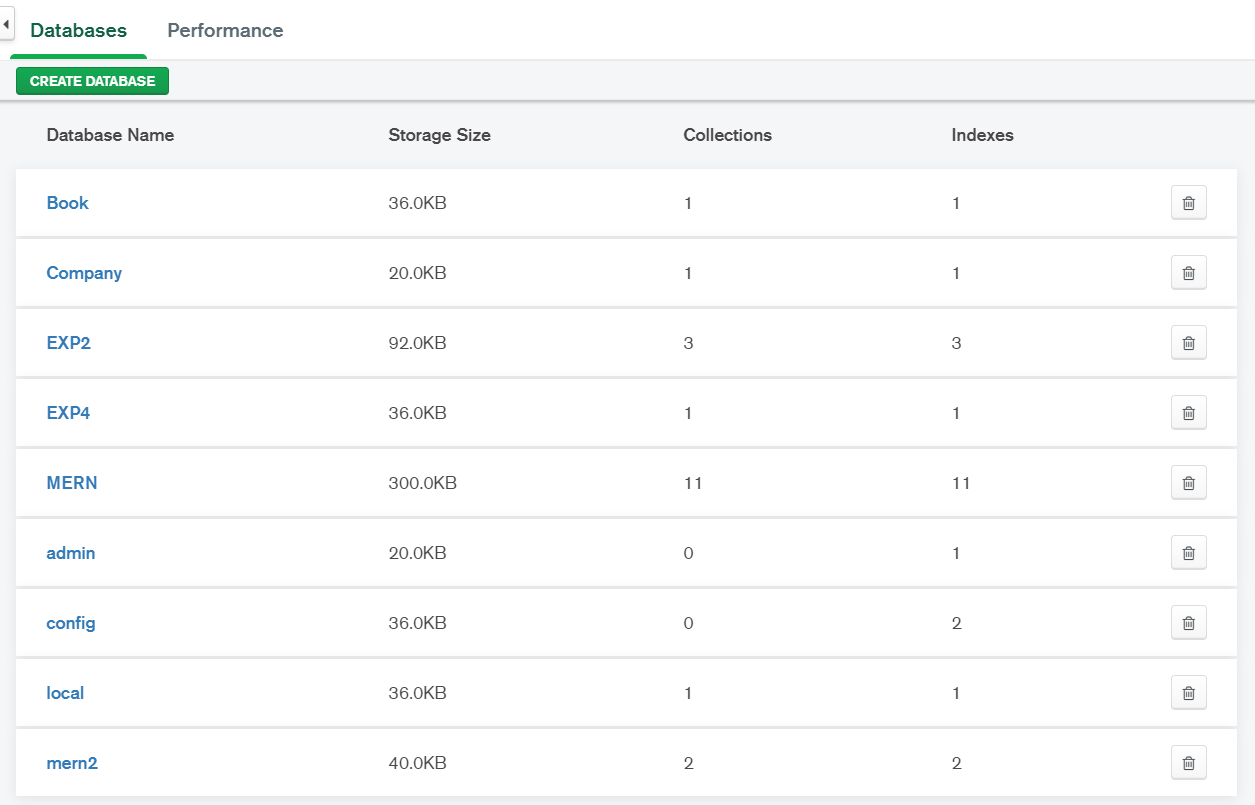
      </form>

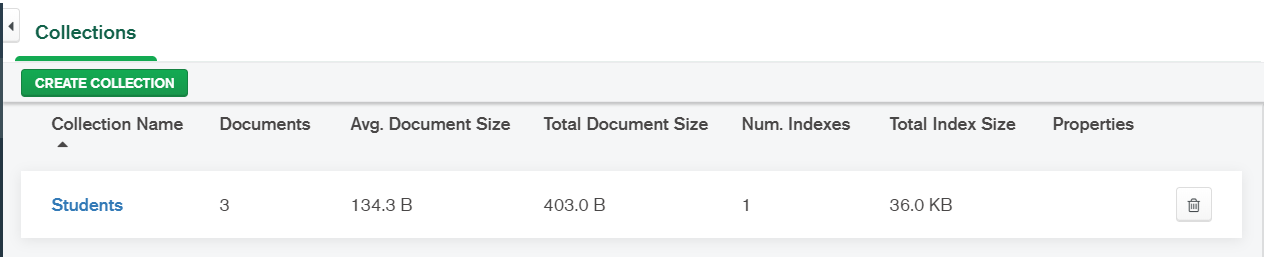
</body>

</html>

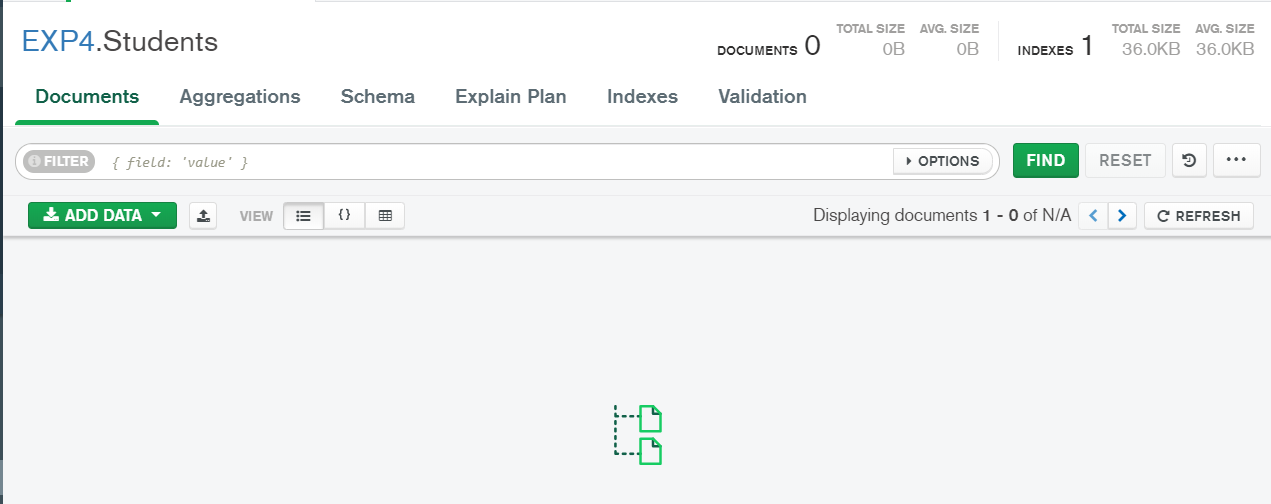
**OUTPUT:**

**Database (EXP4) and collection (Students) creation:**

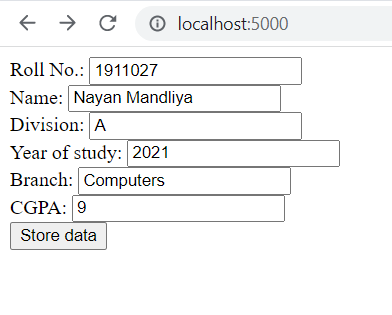
****

****

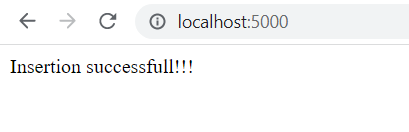
**Mongodb database: (Before Insertion)**

****

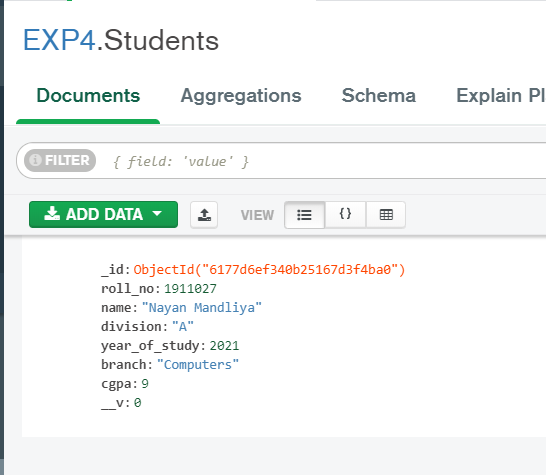
**Browser: (Before button clicked)**

****

**Browser: (After button clicked)**

****

**Mongodb database: (After insertion)**

****

**2) Read:**

**CODE:**

const express=require('express')

const app=express()

const mongoose=require('mongoose');

app.use(express.urlencoded({extended:false}));

app.use(express.json())

mongoose.connect('mongodb://localhost:27017/EXP4');

const db=mongoose.connection;

var Student=require('./1create');

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

    Student.find((err, data)=>{

        if(err)

        {

            console.log(err);

        }

        else

        {

            var str=" ";

            for(var i=0;i<data.length;i++)

            {

                str=str+"<br>{<br>&nbsp; &nbsp;&nbsp; &nbsp;roll\_no: "+data[i].roll\_no+",<br>&nbsp; &nbsp;&nbsp; &nbsp;name: "+data[i].name+",<br>&nbsp; &nbsp;&nbsp; &nbsp;division: "+data[i].division+",<br>&nbsp; &nbsp;&nbsp; &nbsp;year\_of\_study: "+data[i].year\_of\_study+",<br>&nbsp; &nbsp;&nbsp; &nbsp;branch: "+data[i].branch+",<br>&nbsp; &nbsp;&nbsp; &nbsp;cgpa: "+data[i].cgpa+"<br>},";

            }

            res.send(str);

        }

    });

})

app.get('/:id',(req,res)=>{

    Student.find({roll\_no: req.params.id},(err, data)=>{

        if(err)

        {

            console.log(err);

        }

        else if(data.length==0)

        {

            res.send("Invalid roll number!!!");

        }

        else

        {

            var str;

            str="<br>{<br>&nbsp; &nbsp;&nbsp; &nbsp;roll\_no: "+data[0].roll\_no+",<br>&nbsp; &nbsp;&nbsp; &nbsp;name: "+data[0].name+",<br>&nbsp; &nbsp;&nbsp; &nbsp;division: "+data[0].division+",<br>&nbsp; &nbsp;&nbsp; &nbsp;year\_of\_study: "+data[0].year\_of\_study+",<br>&nbsp; &nbsp;&nbsp; &nbsp;branch: "+data[0].branch+",<br>&nbsp; &nbsp;&nbsp; &nbsp;cgpa: "+data[0].cgpa+"<br>}";

            res.send(str);

        }

    });

})

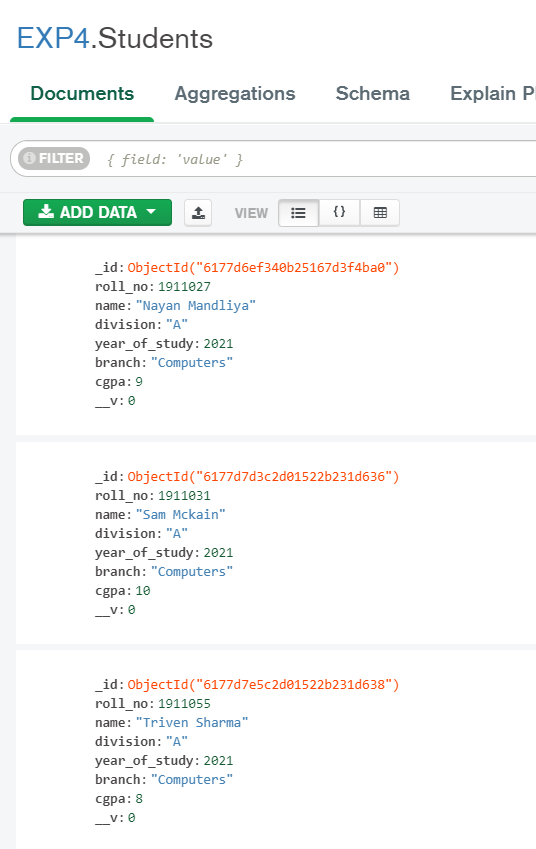
app.listen(7000,(req,res)=>{

    console.log("Server listening on port 7000!!!");

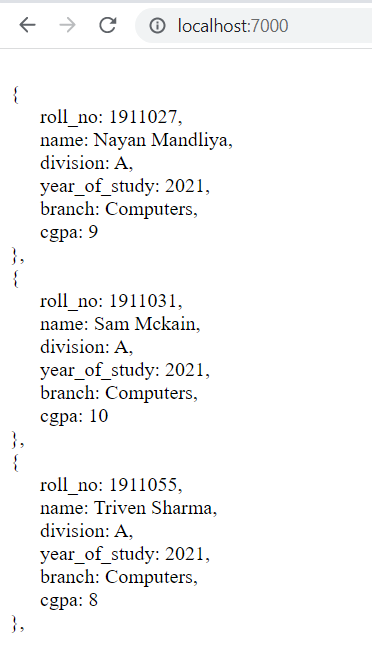
})

})

**Mongodb database:**

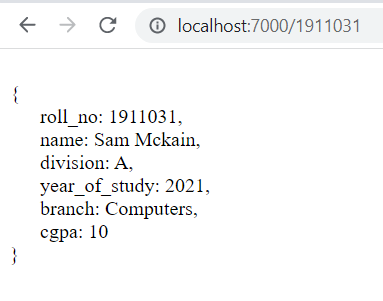
****

**OUTPUT:**

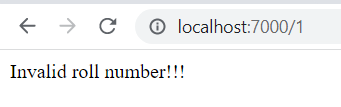
****

**After entering any roll number in url:**

**If roll number found:**

****

**If roll number not found:**

****

**3) Update:**

**Updateone:**

**CODE:**

**Update.js:**

const express=require('express');

const app=express();

app.engine('html', require('ejs').renderFile);

app.set('view engine', 'html');

const mongoose=require('mongoose');

app.use(express.urlencoded({extended:false}));

app.use(express.json())

mongoose.connect('mongodb://localhost:27017/EXP4');

const db=mongoose.connection;

var Student=require('./1create');

const path=require('path');

app.get('/:id',(req,res)=>{

    Student.find({name: req.params.id},(err, data)=>{

        if(err)

        {

            console.log(err);

        }

        else if(data.length==0)

        {

            res.send("Invalid name!!!");

        }

        else

        {

            var str;

            str=[data[0].roll\_no,data[0].name,data[0].division,data[0].year\_of\_study,data[0].branch,data[0].cgpa]

            res.render(path.join(\_\_dirname, '/public', 'update.html'), {stri: str});

        }

    });

})

app.post('/:id',(req,res)=>{

    Student.updateOne({name:req.params.id},{cgpa:Number(req.body.cgpa)},(err,data)=>{

        if(err)

        {

            console.log(err);

        }

        else

        {

            res.send("CGPA updated successfully!!!!");

        }

    });

})

app.listen(7000,(req,res)=>{

    console.log("Server listening on port 7000!!!");

})

**Update.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form name="updatestudent" method="post">

        <h4>Roll No. : <%= stri[0] %></h4>

        <h4>Name : <%= stri[1] %></h4>

        <h4>Division : <%= stri[2] %></h4>

        <h4>Year of study : <%= stri[3] %></h>

        <h4>Branch : <%= stri[4] %></h4>

        <h4>CGPA : <%= stri[5] %></h4>

        <div class="input">

            <span class="label1">CGPA:</span>

            <input type="number" name="cgpa"/>

        </div>

        <br>

        <div class="actions">

          <input class="buttoner" type="submit" value="Update data"/>

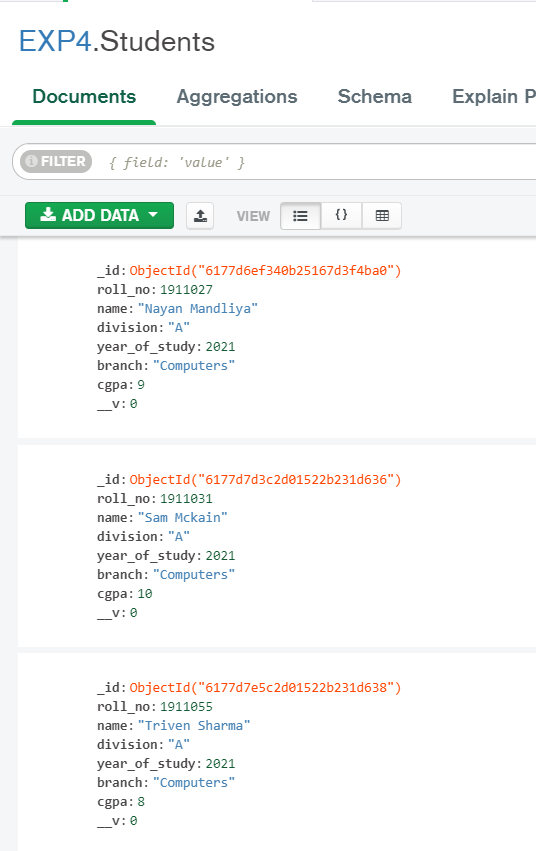
        </div>

      </form>

</body>

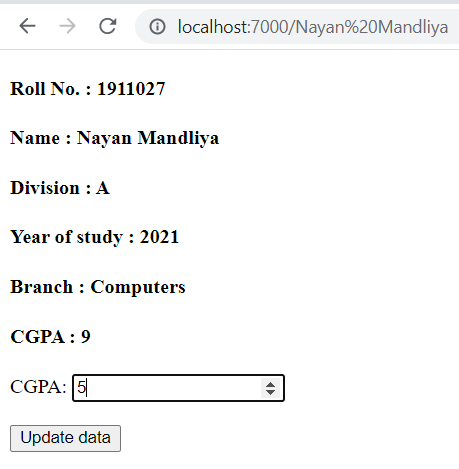
</html>

**MongoDB database:**

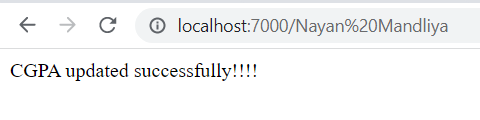
****

**OUTPUT:**

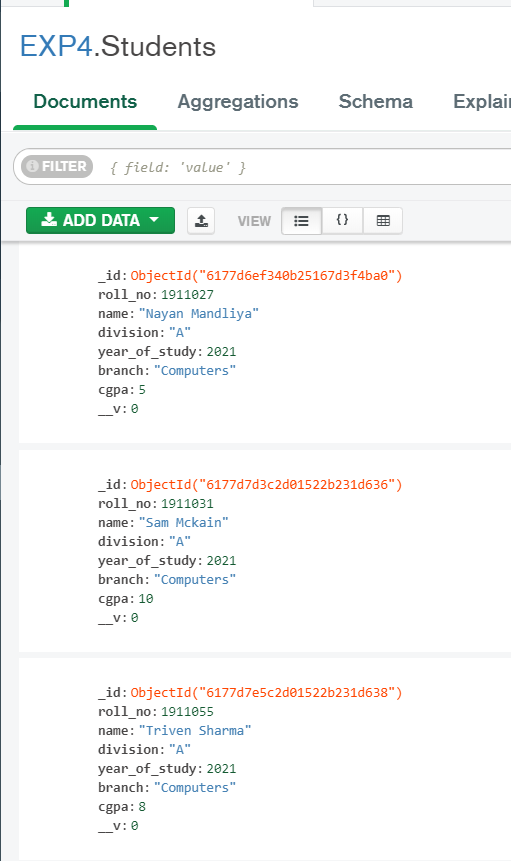
**If valid name entered in url:**

****

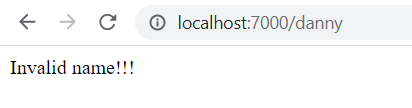
**After update button is clicked:**

****

**MongoDB database: As you can see in the initial DB cgpa for “Nayan Mandliya” was 9 and after updation cgpa is 5.**

****

**If invalid name entered in url:**

****

**Updatemany:**

**CODE:**

**Update.js:**

const express=require('express');

const app=express();

app.engine('html', require('ejs').renderFile);

app.set('view engine', 'html');

const mongoose=require('mongoose');

app.use(express.urlencoded({extended:false}));

app.use(express.json())

mongoose.connect('mongodb://localhost:27017/EXP4');

const db=mongoose.connection;

var Student=require('./1create');

const path=require('path');

app.get('/:id',(req,res)=>{

    Student.find({division: req.params.id},(err, data)=>{

        if(err)

        {

            console.log(err);

        }

        else if(data.length==0)

        {

            res.send("Invalid division!!!");

        }

        else

        {

            var str=[];

            var temp;

            for(var i=0;i<data.length;i++)

            {

                temp=[data[i].roll\_no,data[i].name,data[i].division,data[i].year\_of\_study,data[i].branch,data[i].cgpa]

                str.push(temp);

            }

            res.render(path.join(\_\_dirname, '/public', 'update.html'), {stri: str});

        }

    });

})

app.post('/:id',(req,res)=>{

    Student.updateMany({divsion:req.params.id},{year\_of\_study:Number(req.body.year\_of\_study)},(err,data)=>{

        if(err)

        {

            console.log(err);

        }

        else

        {

            res.send("Updations successful!!!!");

        }

    });

})

app.listen(7000,(req,res)=>{

    console.log("Server listening on port 7000!!!");

})

**Update.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form name="updatestudent" method="post">

        <% for(var i=0; i < stri.length; i++) { %>

            <h4>Roll No. : <%= stri[i][0] %></h4>

            <h4>Name : <%= stri[i][1] %></h4>

            <h4>Division : <%= stri[i][2] %></h4>

            <h4>Year of study : <%= stri[i][3] %></h>

            <h4>Branch : <%= stri[i][4] %></h4>

            <h4>CGPA : <%= stri[i][5] %></h4>

         <% } %>

        <div class="input">

            <span class="label1">Year of study:</span>

            <input type="number" name="year\_of\_study"/>

        </div>

        <br>

        <div class="actions">

          <input class="buttoner" type="submit" value="Update data"/>

        </div>

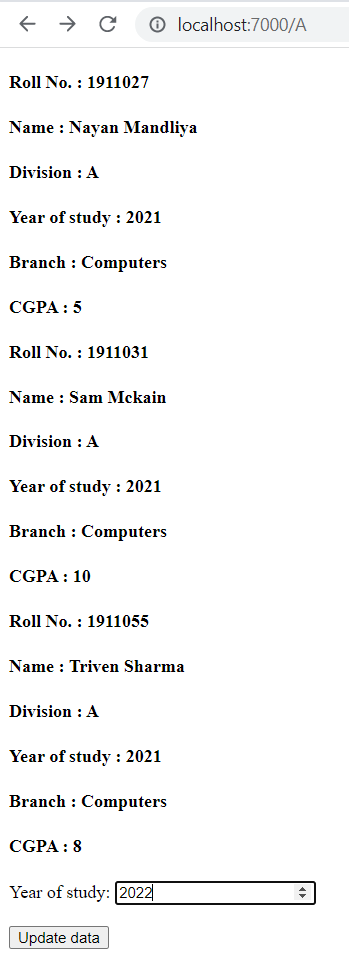
      </form>

</body>

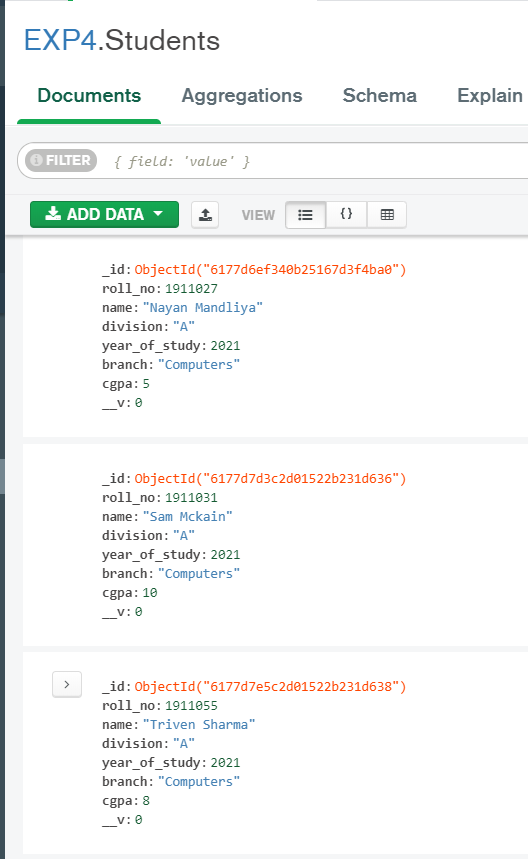
</html>

**OUTPUT:**

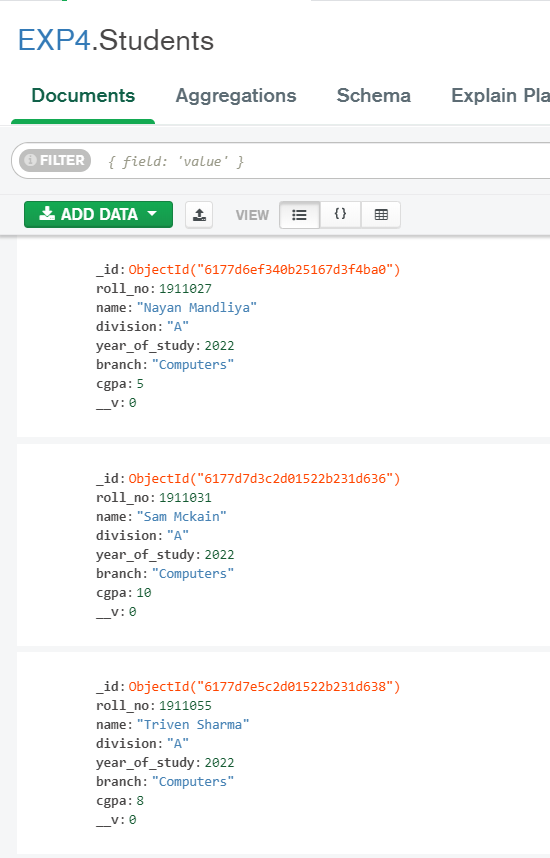
**If valid division entered in url:**

****

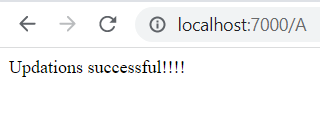
**Before update database:**

****

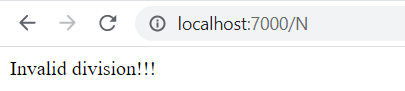
**After update database: As you can see year\_of\_study for all the A division students are updated**

****

**After update browser:**

****

**If invalid division entered in url:**

****

**4) Delete:**

**Deleteone:**

**CODE:**

**Delete.js:**

const express=require('express');

const app=express();

app.engine('html', require('ejs').renderFile);

app.set('view engine', 'html');

const mongoose=require('mongoose');

app.use(express.urlencoded({extended:false}));

app.use(express.json())

mongoose.connect('mongodb://localhost:27017/EXP4');

const db=mongoose.connection;

var Student=require('./1create');

const path=require('path');

app.get('/:id',(req,res)=>{

    Student.find({name: req.params.id},(err, data)=>{

        if(err)

        {

            console.log(err);

        }

        else if(data.length==0)

        {

            res.send("Invalid name!!!");

        }

        else

        {

            var str;

            str=[data[0].roll\_no,data[0].name,data[0].division,data[0].year\_of\_study,data[0].branch,data[0].cgpa]

            res.render(path.join(\_\_dirname, '/public', 'delete.html'), {stri: str});

        }

    });

})

app.post('/:id',(req,res)=>{

    Student.deleteOne({name:req.params.id},(err,data)=>{

        if(err)

        {

            console.log(err);

        }

        else

        {

            res.send("Data deleted successfully!!!!");

        }

    });

})

app.listen(7000,(req,res)=>{

    console.log("Server listening on port 7000!!!");

})

**Delete.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form name="deletestudent" method="post">

        <h4>Roll No. : <%= stri[0] %></h4>

        <h4>Name : <%= stri[1] %></h4>

        <h4>Division : <%= stri[2] %></h4>

        <h4>Year of study : <%= stri[3] %></h>

        <h4>Branch : <%= stri[4] %></h4>

        <h4>CGPA : <%= stri[5] %></h4>

        <br>

        <div class="actions">

          <input class="buttoner" type="submit" value="Delete data"/>

        </div>

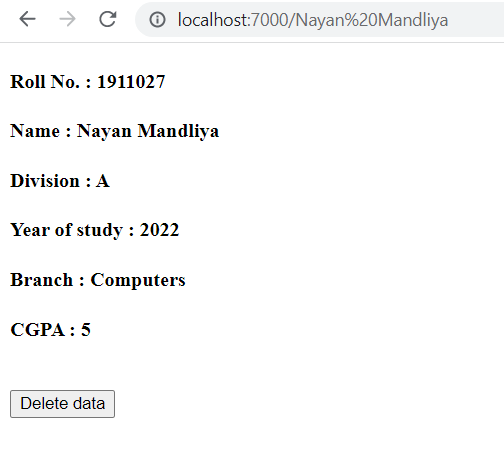
      </form>

</body>

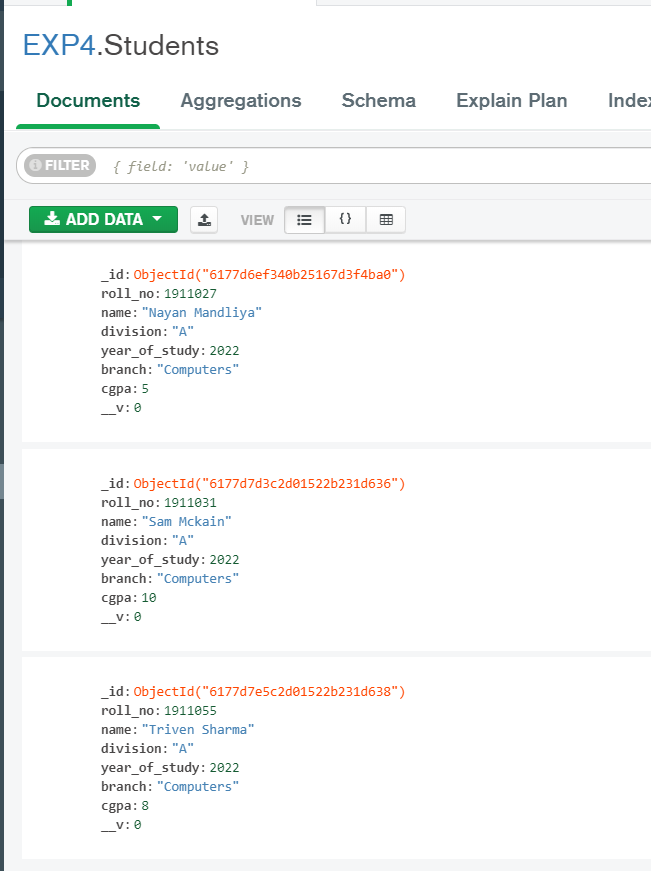
</html>

**OUTPUT:**

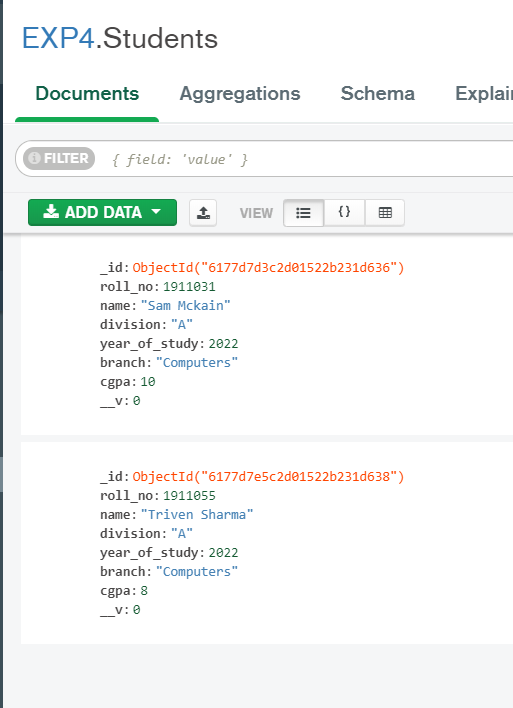
**If valid name entered in url:**

****

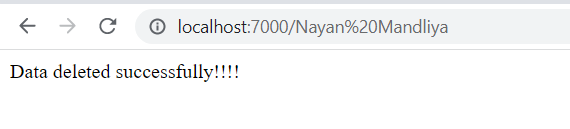
**Before delete database:**

****

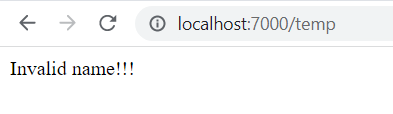
**After delete database: As you can see the student with name “Nayan Mandliya” is deleted.**

****

**After delete browser:**

****

**If invalid name entered in url:**

****

**Deletemany:**

**CODE:**

**Delete.js:**

const express=require('express');

const app=express();

app.engine('html', require('ejs').renderFile);

app.set('view engine', 'html');

const mongoose=require('mongoose');

app.use(express.urlencoded({extended:false}));

app.use(express.json())

mongoose.connect('mongodb://localhost:27017/EXP4');

const db=mongoose.connection;

var Student=require('./1create');

const path=require('path');

app.get('/:id',(req,res)=>{

    Student.find({division: req.params.id},(err, data)=>{

        if(err)

        {

            console.log(err);

        }

        else if(data.length==0)

        {

            res.send("Invalid division!!!");

        }

        else

        {

            var str=[];

            var temp;

            for(var i=0;i<data.length;i++)

            {

                temp=[data[i].roll\_no,data[i].name,data[i].division,data[i].year\_of\_study,data[i].branch,data[i].cgpa]

                str.push(temp);

            }

            res.render(path.join(\_\_dirname, '/public', 'delete.html'), {stri: str});

        }

    });

})

app.post('/:id',(req,res)=>{

    Student.deleteMany({division:req.params.id},(err,data)=>{

        if(err)

        {

            console.log(err);

        }

        else

        {

            res.send("Data deleted successfully!!!!");

        }

    });

})

app.listen(7000,(req,res)=>{

    console.log("Server listening on port 7000!!!");

})

**Delete.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form name="deletestudent" method="post">

        <% for(var i=0; i < stri.length; i++) { %>

            <h4>Roll No. : <%= stri[i][0] %></h4>

            <h4>Name : <%= stri[i][1] %></h4>

            <h4>Division : <%= stri[i][2] %></h4>

            <h4>Year of study : <%= stri[i][3] %></h>

            <h4>Branch : <%= stri[i][4] %></h4>

            <h4>CGPA : <%= stri[i][5] %></h4>

         <% } %>

        <br>

        <div class="actions">

          <input class="buttoner" type="submit" value="Delete data"/>

        </div>

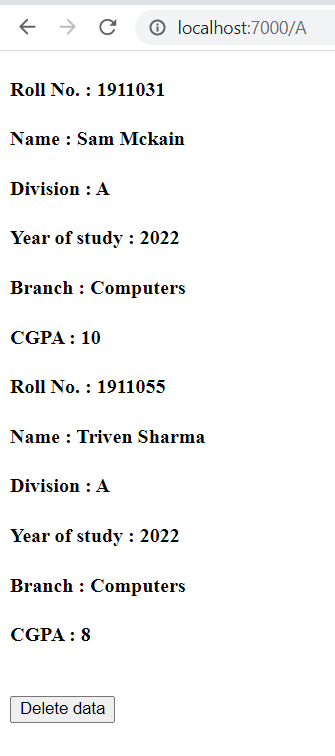
      </form>

</body>

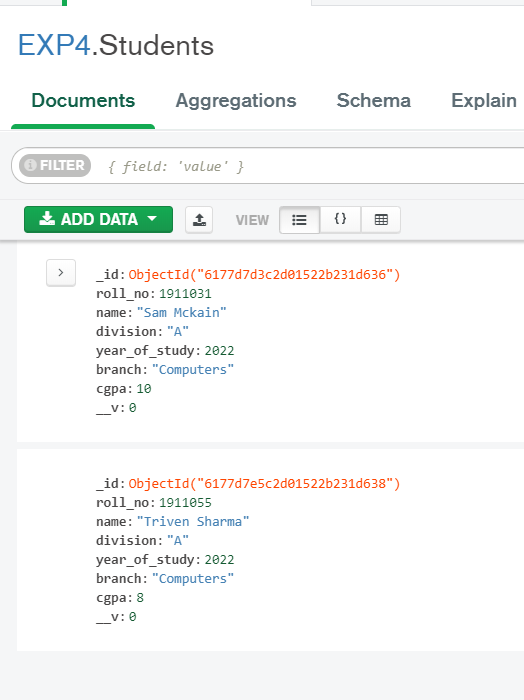
</html>

**OUTPUT:**

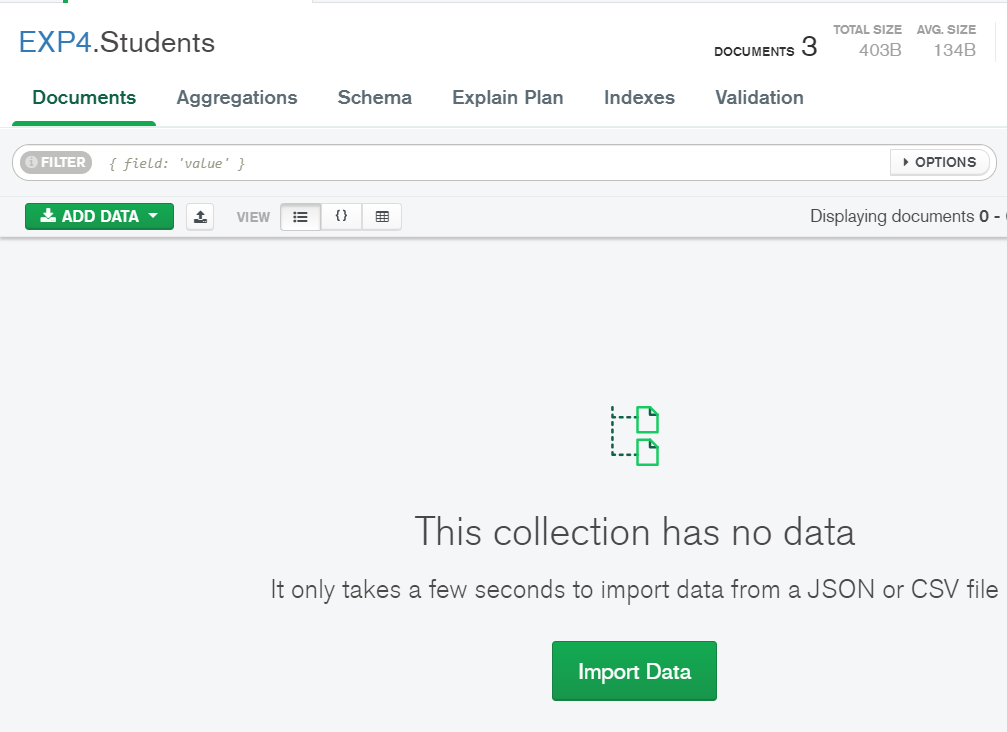
**If valid division entered:**

****

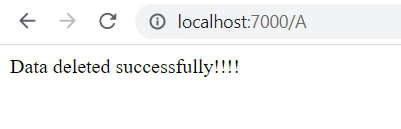
**Before delete database:**

****

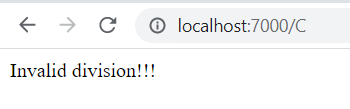
**After delete database:**

****

**After delete browser:**

****

**If invalid division entered:**

****