

**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>
  </form>

```

```

</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:

Databases

Performance

CREATE DATABASE

Database Name	Storage Size	Collections	Indexes	
Book	36.0KB	1	1	
Company	20.0KB	1	1	
EXP2	92.0KB	3	3	
EXP4	36.0KB	1	1	
MERN	300.0KB	11	11	
admin	20.0KB	0	1	
config	36.0KB	0	2	
local	36.0KB	1	1	
mern2	40.0KB	2	2	

Collections						
CREATE COLLECTION						
Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
Students	3	134.3 B	403.0 B	1	36.0 KB	


## Mongodb database: (Before Insertion)

EXP4.Students

DOCUMENTS 0
TOTAL SIZE 0B
AVG. SIZE 0B


INDEXES 1
TOTAL SIZE 36.0KB
AVG. SIZE 36.0KB

Documents
Aggregations
Schema
Explain Plan
Indexes
Validation


FILTER { field: 'value' }
OPTIONS
FIND
RESET

...

ADD DATA
VIEW
{}

Displaying documents 1 - 0 of N/A
REFRESH



## Browser: (Before button clicked)

←
→

localhost:5000

Roll No.: 1911027

Name: Nayan Mandliya

Division: A

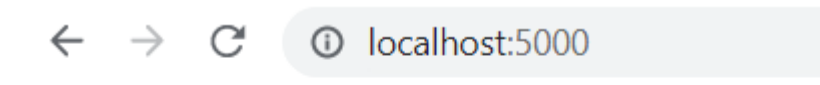
Year of study: 2021

Branch: Computers

CGPA: 9

Store data

**Browser: (After button clicked)**



Insertion successful!!!

**Mongodb database: (After insertion)**

A screenshot of the MongoDB Compass web interface. The title 'EXP4.Students' is at the top. Below it are tabs for 'Documents', 'Aggregations', 'Schema', and 'Explain Plan'. The 'Documents' tab is active. A filter bar shows a placeholder filter: '{ field: 'value' }'. Below the filter bar is a toolbar with an 'ADD DATA' button, an upload icon, a 'VIEW' button, and icons for list, JSON, and grid views. The main area displays a single document in JSON format:

```
{
  "_id": ObjectId("6177d6ef340b25167d3f4ba0"),
  "roll_no": 1911027,
  "name": "Nayan Mandliya",
  "division": "A",
  "year_of_study": 2021,
  "branch": "Computers",
  "cgpa": 9,
  "__v": 0
}
```

## 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; &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:

# EXP4.Students

Documents

Aggregations

Schema

Explain P

FILTER { field: 'value' }

ADD DATA

VIEW

{}

```
_id: ObjectId("6177d6ef340b25167d3f4ba0")  
roll_no: 1911027  
name: "Nayan Mandliya"  
division: "A"  
year_of_study: 2021  
branch: "Computers"  
cgpa: 9  
__v: 0
```

```
_id: ObjectId("6177d7d3c2d01522b231d636")  
roll_no: 1911031  
name: "Sam Mckain"  
division: "A"  
year_of_study: 2021  
branch: "Computers"  
cgpa: 10  
__v: 0
```

```
_id: ObjectId("6177d7e5c2d01522b231d638")  
roll_no: 1911055  
name: "Triven Sharma"  
division: "A"  
year_of_study: 2021  
branch: "Computers"  
cgpa: 8  
__v: 0
```

## OUTPUT:

```
← → ↻ ⓘ localhost:7000

{
  roll_no: 1911027,
  name: Nayan Mandliya,
  division: A,
  year_of_study: 2021,
  branch: Computers,
  cgpa: 9
},
{
  roll_no: 1911031,
  name: Sam Mckain,
  division: A,
  year_of_study: 2021,
  branch: Computers,
  cgpa: 10
},
{
  roll_no: 1911055,
  name: Triven Sharma,
  division: A,
  year_of_study: 2021,
  branch: Computers,
  cgpa: 8
},
}
```

After entering any roll number in url:

If roll number found:

```
← → ↻ ⓘ localhost:7000/1911031

{
  roll_no: 1911031,
  name: Sam Mckain,
  division: A,
  year_of_study: 2021,
  branch: Computers,
  cgpa: 10
}
```

If roll number not found:

```
← → ↻ ⓘ localhost:7000/1
```

Invalid roll number!!!



### 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,d
    ata)=>{
        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] %></h4>
    <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:

## EXP4.Students

Documents

Aggregations

Schema

Explain P

 FILTER { field: 'value' }

 ADD DATA ▾



VIEW



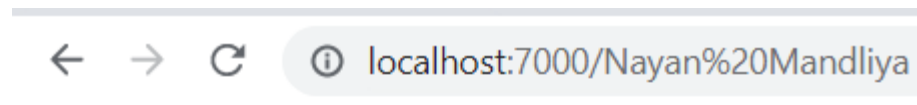
```
_id: ObjectId("6177d6ef340b25167d3f4ba0")
roll_no: 1911027
name: "Nayan Mandliya"
division: "A"
year_of_study: 2021
branch: "Computers"
cgpa: 9
__v: 0
```

```
_id: ObjectId("6177d7d3c2d01522b231d636")
roll_no: 1911031
name: "Sam Mckain"
division: "A"
year_of_study: 2021
branch: "Computers"
cgpa: 10
__v: 0
```

```
_id: ObjectId("6177d7e5c2d01522b231d638")
roll_no: 1911055
name: "Triven Sharma"
division: "A"
year_of_study: 2021
branch: "Computers"
cgpa: 8
__v: 0
```

## OUTPUT:

If valid name entered in url:



**Roll No. : 1911027**

**Name : Nayan Mandliya**

**Division : A**

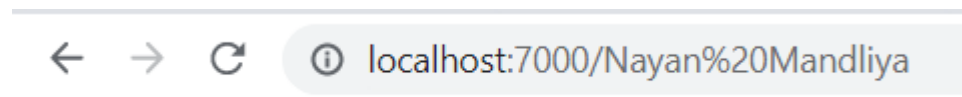
**Year of study : 2021**

**Branch : Computers**

**CGPA : 9**

CGPA:

After update button is clicked:



**CGPA updated successfully!!!!**

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

**EXP4.Students**

**Documents** Aggregations Schema Explain

**FILTER** { field: 'value' }

**ADD DATA** VIEW [List View] [JSON View] [Grid View]

```
{
  "_id": ObjectId("6177d6ef340b25167d3f4ba0"),
  "roll_no": 1911027,
  "name": "Nayan Mandliya",
  "division": "A",
  "year_of_study": 2021,
  "branch": "Computers",
  "cgpa": 5,
  "__v": 0
}
```

```
{
  "_id": ObjectId("6177d7d3c2d01522b231d636"),
  "roll_no": 1911031,
  "name": "Sam Mckain",
  "division": "A",
  "year_of_study": 2021,
  "branch": "Computers",
  "cgpa": 10,
  "__v": 0
}
```

```
{
  "_id": ObjectId("6177d7e5c2d01522b231d638"),
  "roll_no": 1911055,
  "name": "Triven Sharma",
  "division": "A",
  "year_of_study": 2021,
  "branch": "Computers",
  "cgpa": 8,
  "__v": 0
}
```

**If invalid name entered in url:**

← → ↻ ⓘ localhost:7000/danny

Invalid name!!!

## 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({division: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] %></h4>
            <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:

    localhost:7000/A

**Roll No. : 1911027**

**Name : Nayan Mandliya**

**Division : A**

**Year of study : 2021**

**Branch : Computers**

**CGPA : 5**

**Roll No. : 1911031**

**Name : Sam Mckain**

**Division : A**

**Year of study : 2021**

**Branch : Computers**

**CGPA : 10**

**Roll No. : 1911055**

**Name : Triven Sharma**

**Division : A**

**Year of study : 2021**

**Branch : Computers**

**CGPA : 8**

Year of study:



Before update database:

# EXP4.Students

[Documents](#)[Aggregations](#)[Schema](#)[Explain](#)

FILTER

{ field: 'value' }

ADD DATA

VIEW

```
_id: ObjectId("6177d6ef340b25167d3f4ba0")
roll_no: 1911027
name: "Nayan Mandliya"
division: "A"
year_of_study: 2021
branch: "Computers"
cgpa: 5
__v: 0
```

```
_id: ObjectId("6177d7d3c2d01522b231d636")
roll_no: 1911031
name: "Sam Mckain"
division: "A"
year_of_study: 2021
branch: "Computers"
cgpa: 10
__v: 0
```

>

```
_id: ObjectId("6177d7e5c2d01522b231d638")
roll_no: 1911055
name: "Triven Sharma"
division: "A"
year_of_study: 2021
branch: "Computers"
cgpa: 8
__v: 0
```

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

## EXP4.Students

DocumentsAggregationsSchemaExplain Pla

FILTER

{ field: 'value' }

ADD DATA

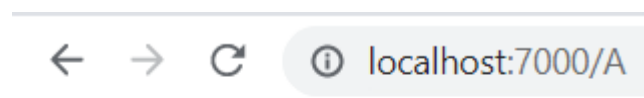
VIEW

```
_id: ObjectId("6177d6ef340b25167d3f4ba0")
roll_no: 1911027
name: "Nayan Mandliya"
division: "A"
year_of_study: 2022
branch: "Computers"
cgpa: 5
__v: 0
```

```
_id: ObjectId("6177d7d3c2d01522b231d636")
roll_no: 1911031
name: "Sam Mckain"
division: "A"
year_of_study: 2022
branch: "Computers"
cgpa: 10
__v: 0
```

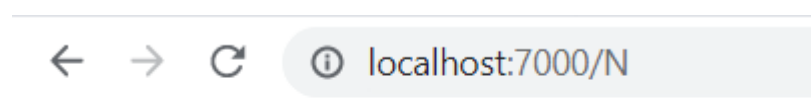
```
_id: ObjectId("6177d7e5c2d01522b231d638")
roll_no: 1911055
name: "Triven Sharma"
division: "A"
year_of_study: 2022
branch: "Computers"
cgpa: 8
__v: 0
```

After update browser:



Updates successful!!!!

If invalid division entered in url:



Invalid division!!!

#### 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] %></h4>
    <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:

←
→
↻
🔒 localhost:7000/Nayan%20Mandliya

---

**Roll No. : 1911027**

**Name : Nayan Mandliya**

**Division : A**

**Year of study : 2022**

**Branch : Computers**

**CGPA : 5**

Before delete database:

# EXP4.Students

[Documents](#)[Aggregations](#)[Schema](#)[Explain Plan](#)[Index](#)

FILTER

{ field: 'value' }

ADD DATA

VIEW

\_id: ObjectId("6177d6ef340b25167d3f4ba0")

roll\_no: 1911027

name: "Nayan Mandliya"

division: "A"

year\_of\_study: 2022

branch: "Computers"

cgpa: 5

\_\_v: 0

\_id: ObjectId("6177d7d3c2d01522b231d636")

roll\_no: 1911031

name: "Sam Mckain"

division: "A"

year\_of\_study: 2022

branch: "Computers"

cgpa: 10

\_\_v: 0

\_id: ObjectId("6177d7e5c2d01522b231d638")

roll\_no: 1911055

name: "Triven Sharma"

division: "A"

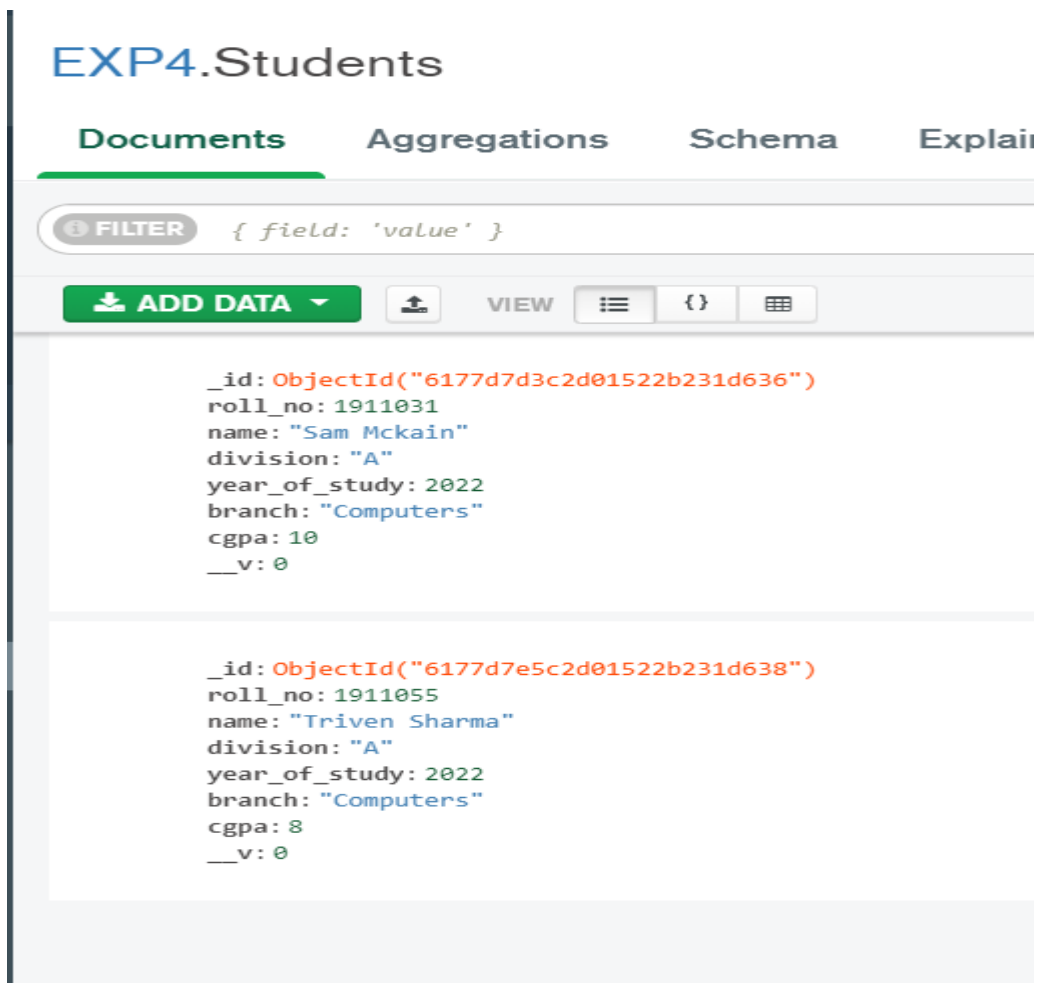
year\_of\_study: 2022

branch: "Computers"

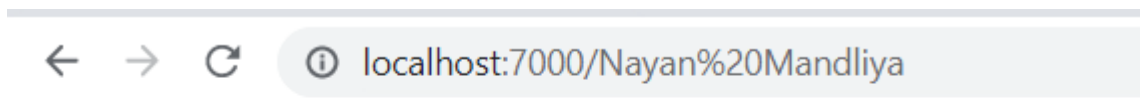
cgpa: 8

\_\_v: 0

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

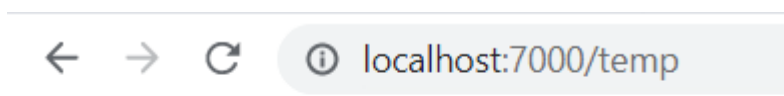


**After delete browser:**



Data deleted successfully!!!!

**If invalid name entered in url:**



Invalid name!!!

## 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:

```

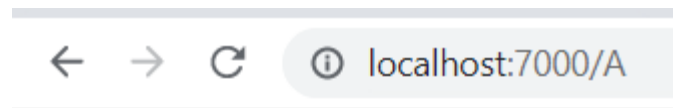
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form name="deletestudent" method="post">
        <% for(var 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] %></h4>
            <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:**



**Roll No. : 1911031**

**Name : Sam Mckain**

**Division : A**

**Year of study : 2022**

**Branch : Computers**

**CGPA : 10**

**Roll No. : 1911055**

**Name : Triven Sharma**

**Division : A**

**Year of study : 2022**

**Branch : Computers**

**CGPA : 8**

Delete data

Before delete database:

## EXP4.Students

Documents

Aggregations

Schema

Explain

 FILTER { field: 'value' }

 ADD DATA ▾



VIEW

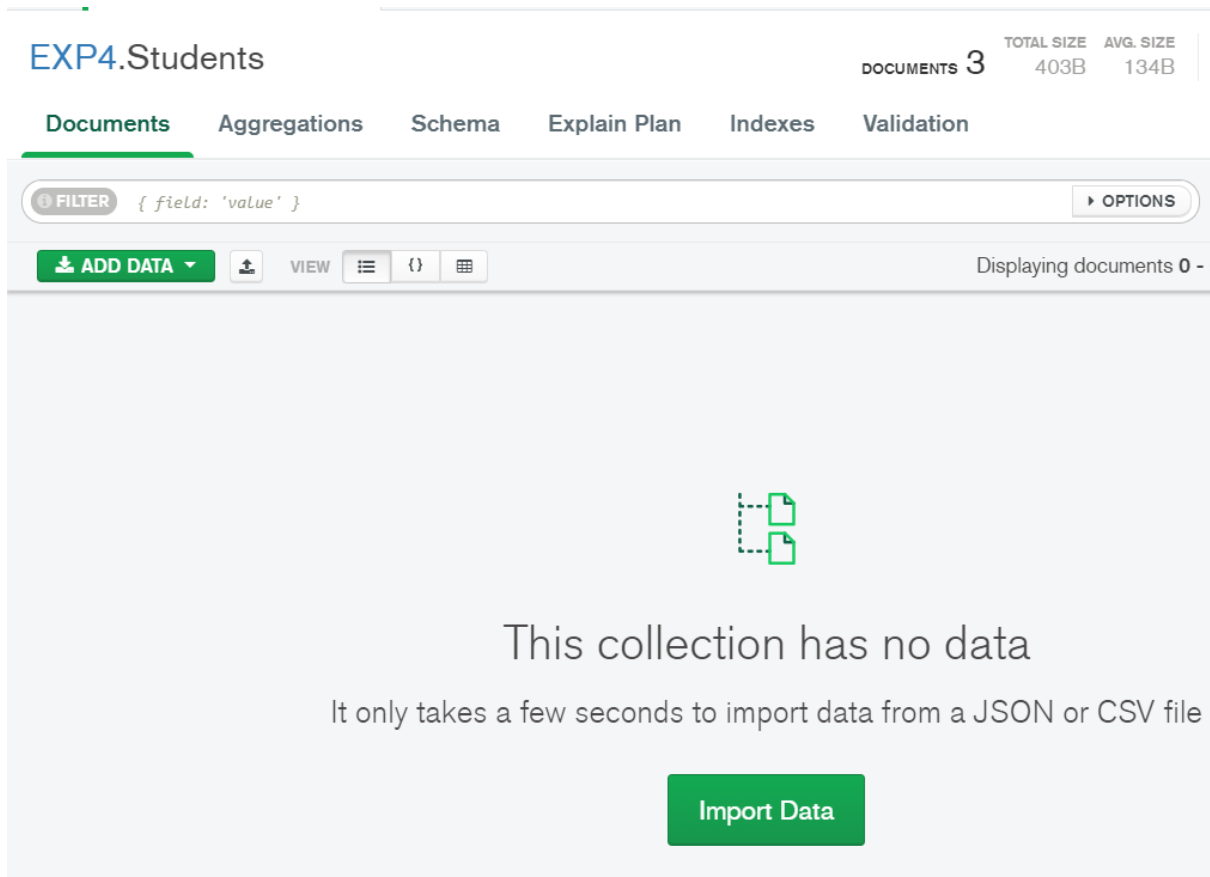


>

```
_id: ObjectId("6177d7d3c2d01522b231d636")
roll_no: 1911031
name: "Sam Mckain"
division: "A"
year_of_study: 2022
branch: "Computers"
cgpa: 10
__v: 0
```

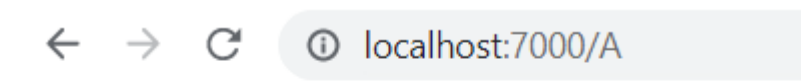
```
_id: ObjectId("6177d7e5c2d01522b231d638")
roll_no: 1911055
name: "Triven Sharma"
division: "A"
year_of_study: 2022
branch: "Computers"
cgpa: 8
__v: 0
```

### After delete database:



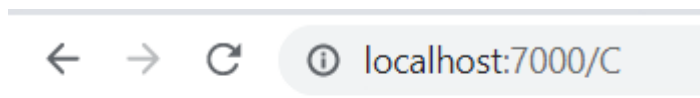
The screenshot shows the MongoDB Compass interface for a database named 'EXP4.Students'. The 'Documents' tab is selected. At the top right, it displays 'DOCUMENTS 3', 'TOTAL SIZE 403B', and 'AVG. SIZE 134B'. Below the tabs, there is a filter bar with a filter icon, the text '{ field: 'value' }', and an 'OPTIONS' button. Below the filter bar, there is a toolbar with an 'ADD DATA' button, an upload icon, a 'VIEW' button, and three view icons (list, JSON, grid). To the right of the toolbar, it says 'Displaying documents 0 - 0'. The main area is empty and contains a message: 'This collection has no data' with a subtext 'It only takes a few seconds to import data from a JSON or CSV file' and an 'Import Data' button.

### After delete browser:



Data deleted successfully!!!!

### If invalid division entered:



Invalid division!!!