1) Write a Node JS program that accepts a port from the user and runs a node server at that port --“server.js”

```
var http = require('http');

var server = http.createServer(function (req, res) {
    if (req.url == '/') { //check the URL of the current request

        // set response header
        res.writeHead(200, { 'Content-Type': 'text/html' });
        // set response content
        res.write('<html><body><p>This is home Page.</p></body></html>');
        res.end();
    }
    else if (req.url == "/student") {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This is student Page.</p></body></html>');
        res.end();
    }
    else if (req.url == "/admin") {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This is admin Page.</p></body></html>');
        res.end();
    }
    else
        res.end('Invalid Request!');


});

server.listen(8000);
console.log('Node.js web server at port 8000 is running..')
```

EXPECTED OUTPUT

```
node server.js

Node.js web server at port 8000 is running..
```

2) Write a JavaScript program to remove duplicate items from an array – ‘three.js’

```
function removeDuplicates(num) {
        len=num.length;
        uniqueChars=[];

        num.forEach((c) => {
            if (!uniqueChars.includes(c)) {
                uniqueChars.push(c);
            }
        });
    return uniqueChars;
  }
  let Mynum = [1, 2, 2, 4, 5, 4, 7, 8, 7, 3, 6];
  result = removeDuplicates(Mynum);
  console.log("Original List:   "+Mynum);
  console.log("Unique List:    "+result);
```

3)   Create a student database in MongoDB with all the details of students of a class

```
1.show dbs;

2. use student;
switched to db student
insert into studentinfo collection
3.
db.studentinfo.insert({name:"john",id:"20bd1a05051",course:"b.tech",branch:"cse"})
WriteResult({ "nInserted" : 1 })

4.
db.studentinfo.insert({name:"reena",id:"20bd1a0502",course:"M.tech",branch:"it"})
WriteResult({ "nInserted" : 1 })

5. db.studentinfo.insert({name:"ram",id:"20bd1a0503",course:"b.tech",branch:"cse"})
WriteResult({ "nInserted" : 1 })

6. db.studentinfo.find({})
{ "_id" : ObjectId("62a99e693dbaba59a0af05cf"), "name" : "john", "id" :
"20bd1a05051", "course" : "b.tech", "branch" : "cse" }
{ "_id" : ObjectId("62a99ebd3dbaba59a0af05d0"), "name" : "reena", "id" :
"20bd1a0502", "course" : "M.tech", "branch" : "it" }
{ "_id" : ObjectId("62a99f123dbaba59a0af05d1"), "name" : "ram", "id" :
"20bd1a0503", "course" : "b.tech", "branch" : "cse" }
```

EXPECTED OUTPUT

view studentinfo collection: