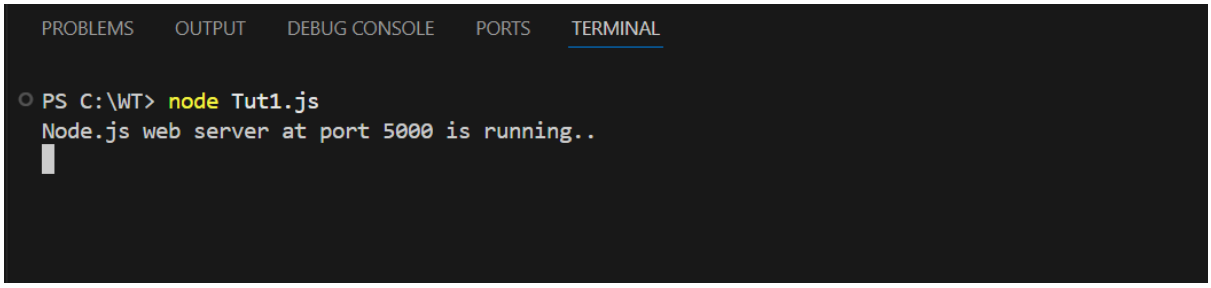
Unit 3

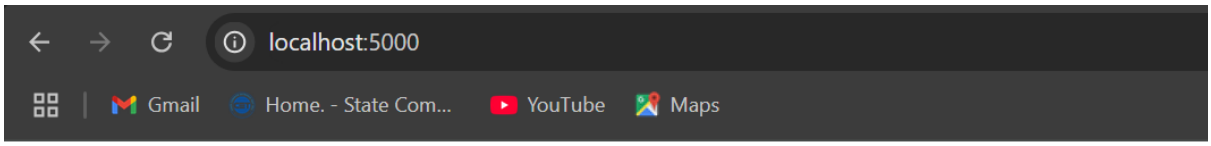
Q.1) Create an HTTP server and perform operations on it.

a) understand http request module :

```
var http = require('http'); // 1 - Import Node.js core module
var server = http.createServer(function (req, res) {   // 2 - creating server
   //handle incomming requests here..
   res.write("Hello Server");
   res.end();
});
server.listen(5000); //3 - listen for any incoming requests
console.log('Node.js web server at port 5000 is running..')
```

Output :





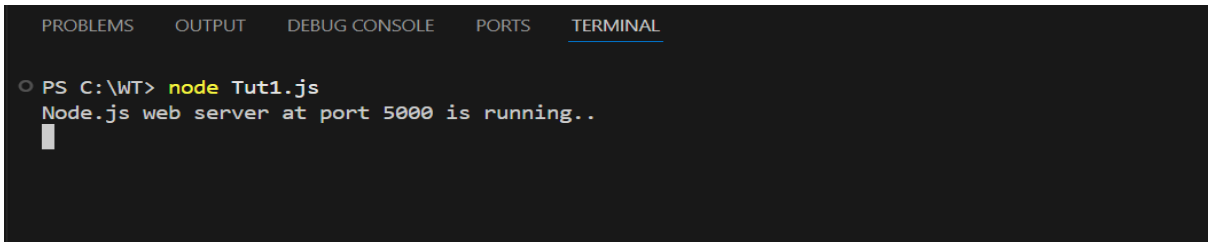
Hello Server

b) routing :

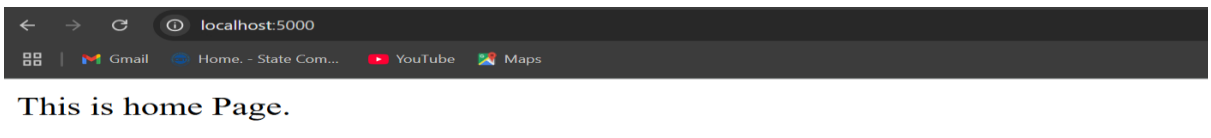
```
//understand routing in http module
var http = require('http'); // Import Node.js core module
var server = http.createServer(function (req, res) {   //create web server
   if (req.url == '/') { //check the URL of the current request
      // set response header
      res.writeHead(200, { 'Content-Type': '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(5000); //6 - listen for any incoming requests
console.log('Node.js web server at port 5000 is running..')
```

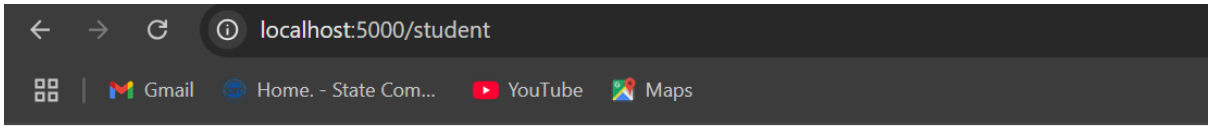
Output :



Home page :

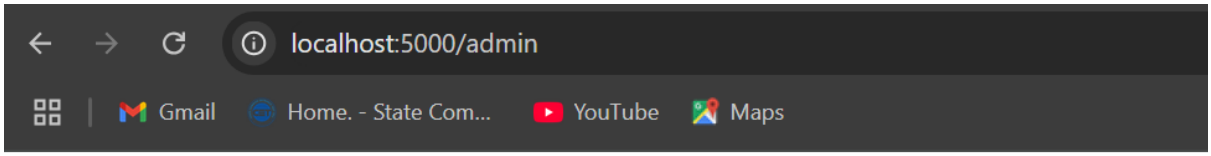


Student page :



This is student Page.

Admin page :



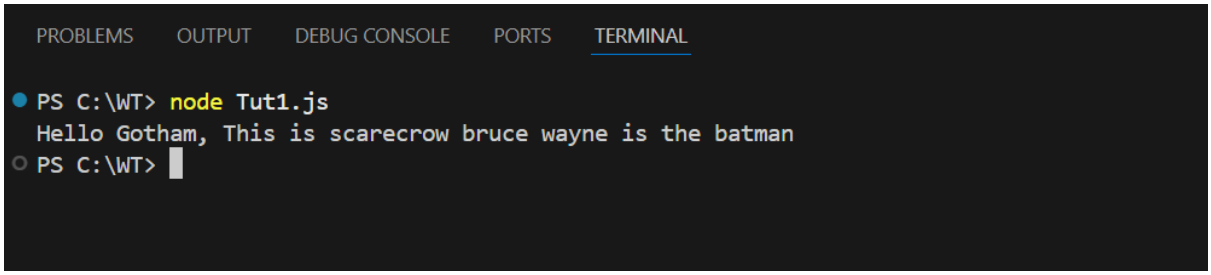
This is admin Page.

Q.2) Using File Handling operations demonstrate all basic operations(Create, Write, Read, Delete).

a) Reading File

```
var fs = require('fs');
fs.readFile('input.txt', function (err, data) {
        if (err) throw err;
   console.log(data.toString());
});
```

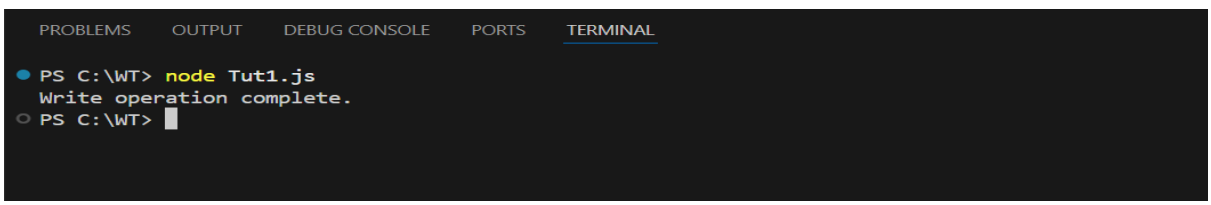
Ouput :



b) Writing to a file

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

fs.writeFile('test.txt', 'Hello Gotham ,Behold the almighty Ras-al-ghul
! ', function (err) {
    if (err)
       console.log(err);
    else
    console.log('Write operation complete.');
});
```

Ouput :



```
test.txt
1   Hello Gotham ,Behold the almighty Ras-al-ghul
```

c) Delete the file

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

fs.unlink('test.txt', function () {


   console.log('delete operation complete.');


});
```

Ouput :

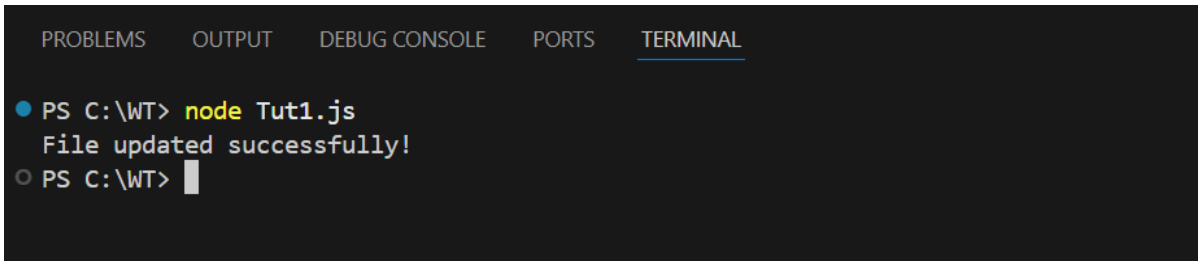
```
PROBLEMS   OUTPUT   DEBUG CONSOLE   PORTS   TERMINAL

PS C:\WT> node Tut1.js
delete operation complete.
PS C:\WT>
```

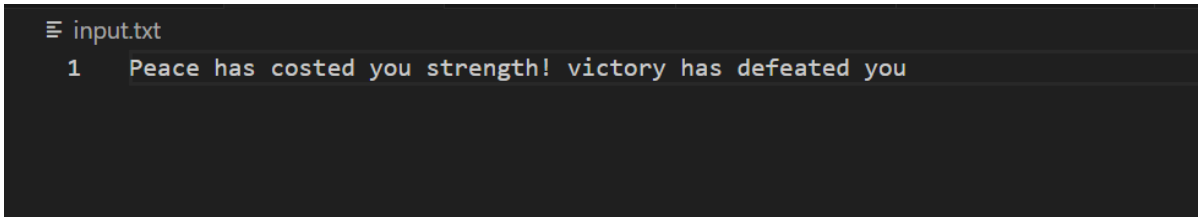
d) Update file

```
const fs = require('fs');

function updateFile(filePath, newContent) {
  fs.writeFile(filePath, newContent, 'utf8', (err) => {
    if (err) {
      console.error(`Error updating the file: ${err.message}`);
    } else {
      console.log('File updated successfully!');
    }
  });
}
```

updateFile('input.txt', 'Peace has costed you strength! victory has defeated you');

Output :



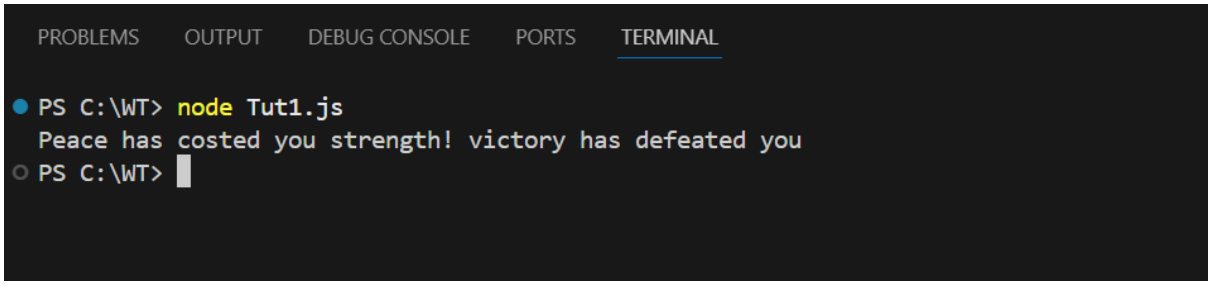


Open a file :

```
const fs = require('fs');
fs.open('input.txt', 'r', (err, fd) => {
 if (err) {
  console.error('Error opening file:', err.message);
  return;
 }
 // Allocate buffer with modern API
 const buffer = Buffer.alloc(10240);
 fs.read(fd, buffer, 0, buffer.length, 0, (err, bytesRead) => {
  if (err) {
   console.error('Error reading file:', err.message);
   fs.close(fd, () => {}); // Ensure file is closed
   return;
  }
  if (bytesRead > 0) {
   console.log(buffer.slice(0, bytesRead).toString());
  }
```

```
  fs.close(fd, (err) => {
    if (err) {
      console.error('Error closing file:', err.message);
    }
  });
 });
});
```

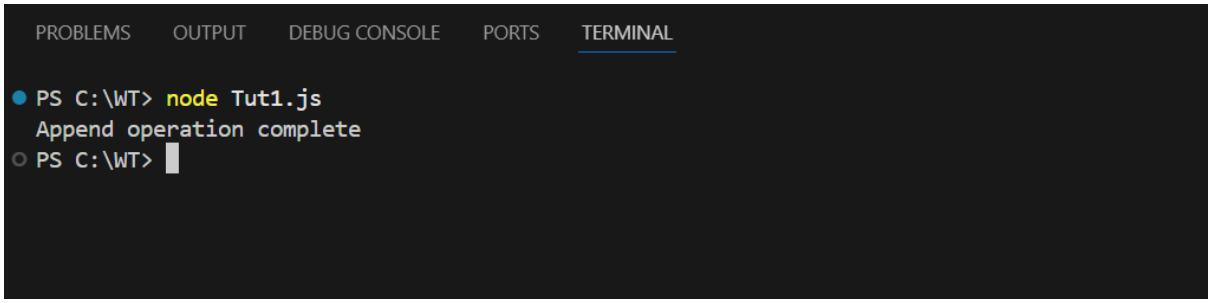
Output :



e) Append to a file.

```
var fs = require('fs');
fs.appendFile('input.txt',', This is Bane Speaking',function(err)
{
if(err)
 console.log(err);
 else
 console.log('Append operation complete');
});
```

Output :



input.txt

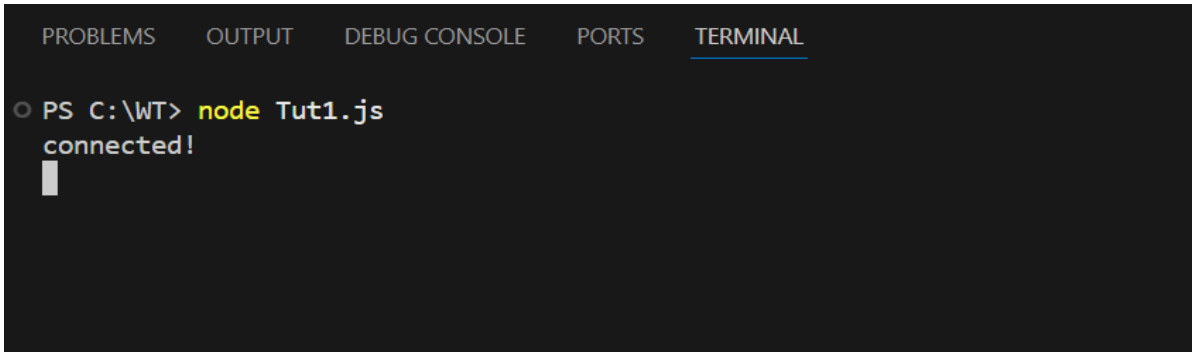
```
1   Peace has costed you strength! victory has defeated you, This is Bane Speaking
```

Q.3) Create an application to establish a connection with MySQL database and perform database operations on it.

a) Database connectivity

```
var mysql = require("mysql");
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
});
con.connect(function (err) {
  if (err) throw err;
  console.log("connected!");
});
```

Output :

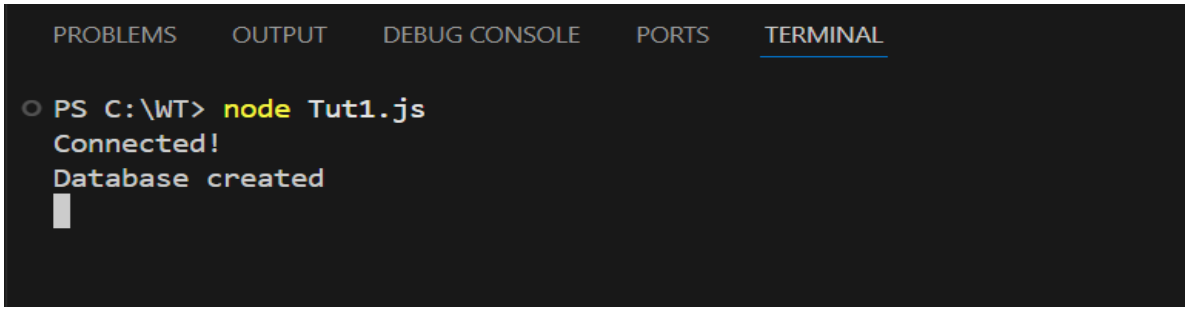


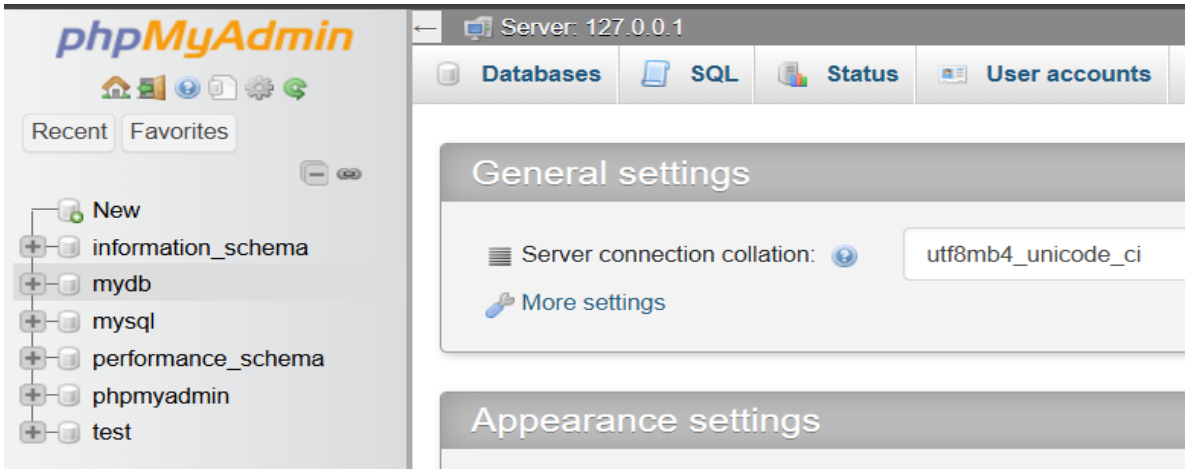
b) Create database.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: ""
});

con.connect(function(err) {
```

```
  if (err) throw err;
  console.log("Connected!");
  con.query("CREATE DATABASE mydb", function (err, result) {
    if (err) throw err;
    console.log("Database created");
  });
});
```

Ouptut :



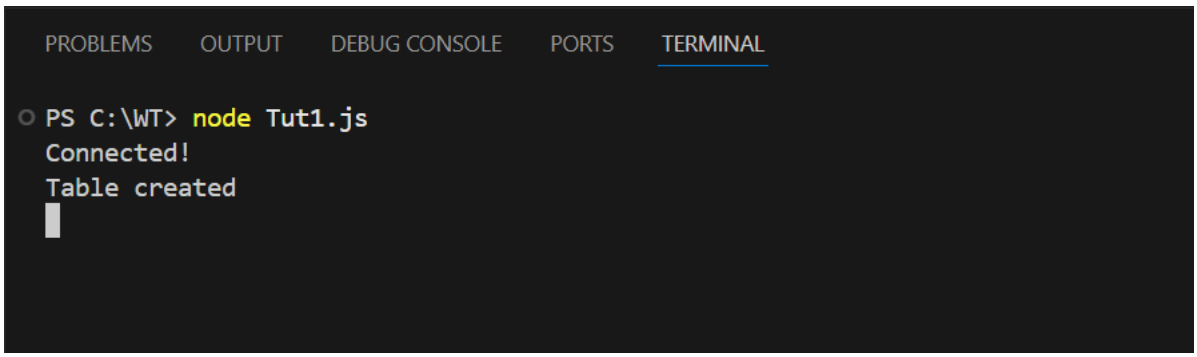


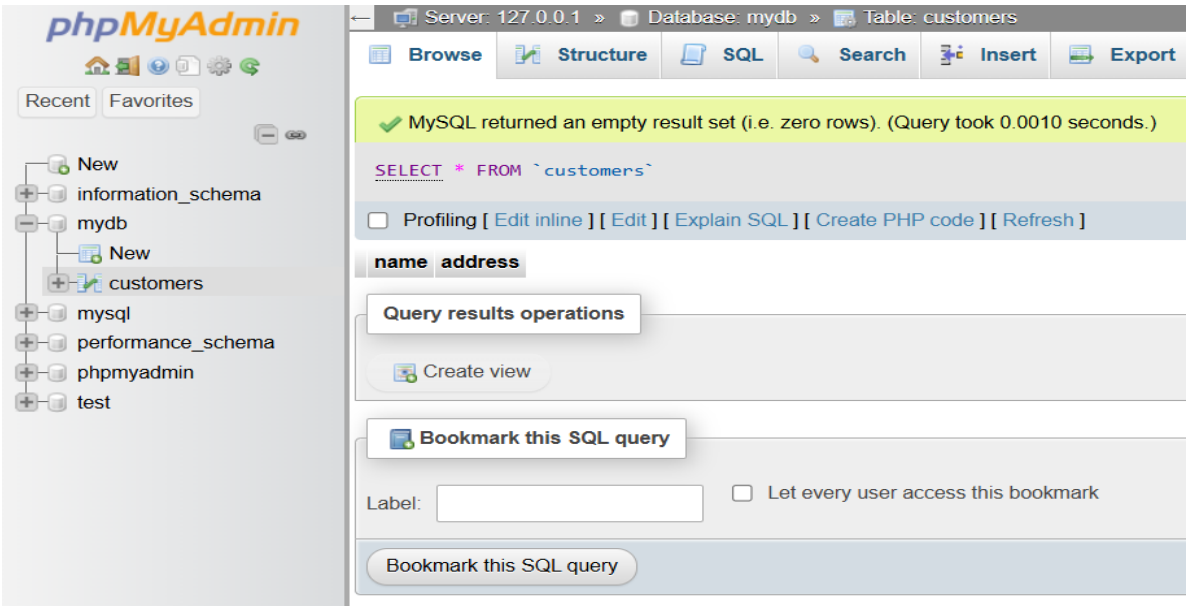
c) Create a table in database.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"
```

```
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE customers (name VARCHAR(255), address VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

Output :

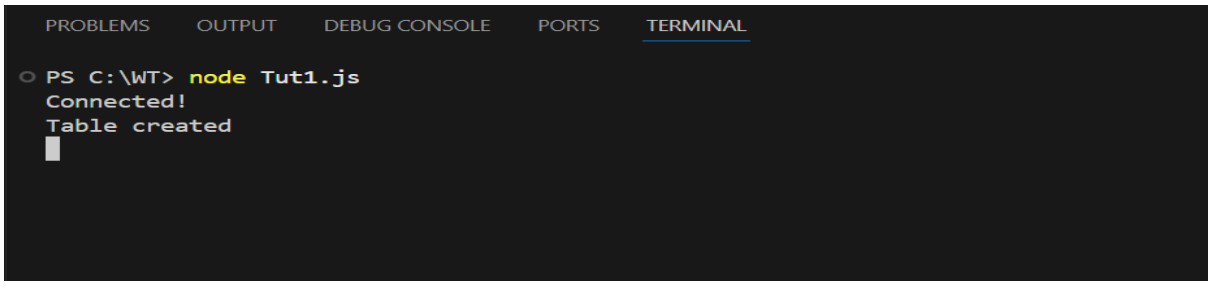




d) Create table with primary key.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "CREATE TABLE customers (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), address VARCHAR(255))";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Table created");
  });
});
```

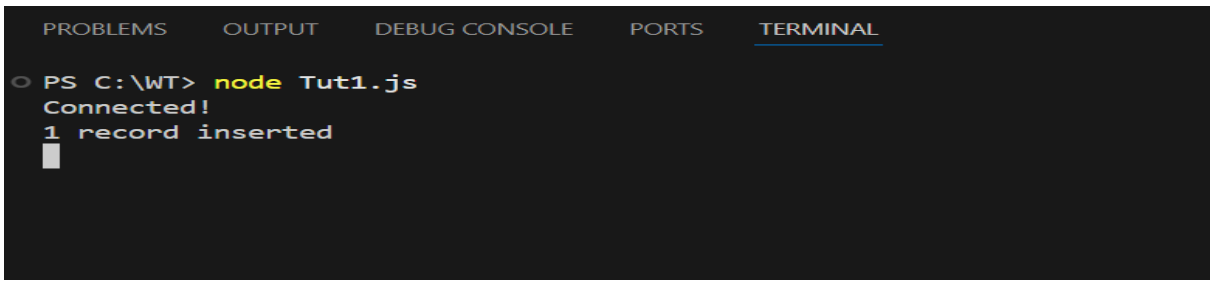
Output :

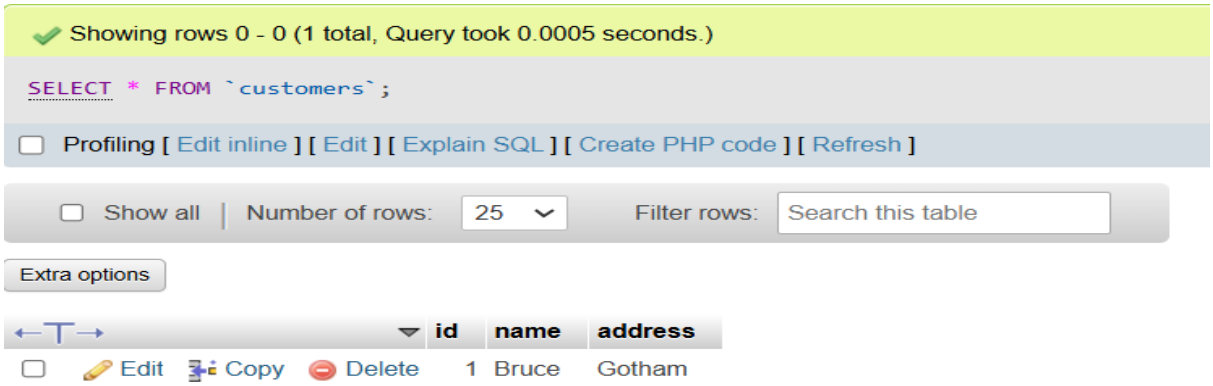


e) Insert a record into table.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"
});
con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
  var sql = "INSERT INTO customers (id, name, address) VALUES (1, 'Bruce', 'Gotham')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
  });
});
```

Output :

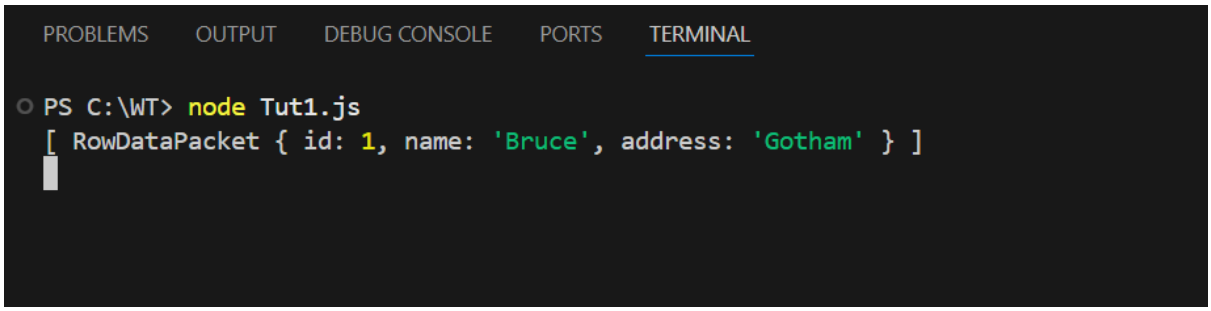




f) Reading from a table.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"
});
con.connect(function(err) {
  if (err) throw err;
  con.query("SELECT * FROM customers", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });
});
```

Output :

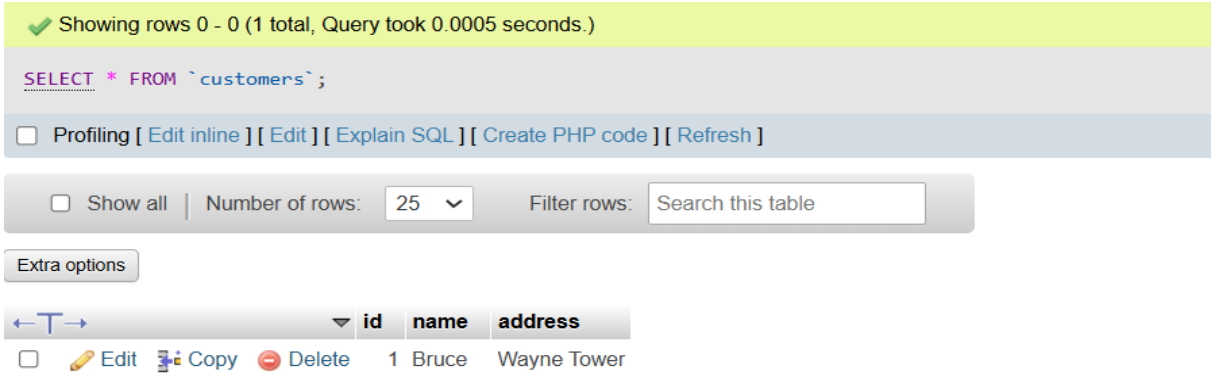


g) Update a record in a table.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"
```

```
});
con.connect(function(err) {
  if (err) throw err;
  var sql = "UPDATE customers SET address = 'Wayne Tower' WHERE address = 'Gotham'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log(result.affectedRows + " record(s) updated");
  });
});
```

Output :



h) Delete a record from table.

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "mydb"
```

```
});
con.connect(function(err) {
  if (err) throw err;
  var sql = "DELETE FROM customers WHERE address = 'Wayne Tower'";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("Number of records deleted: " + result.affectedRows);
  });
});
```

Output :