

Unit 3

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

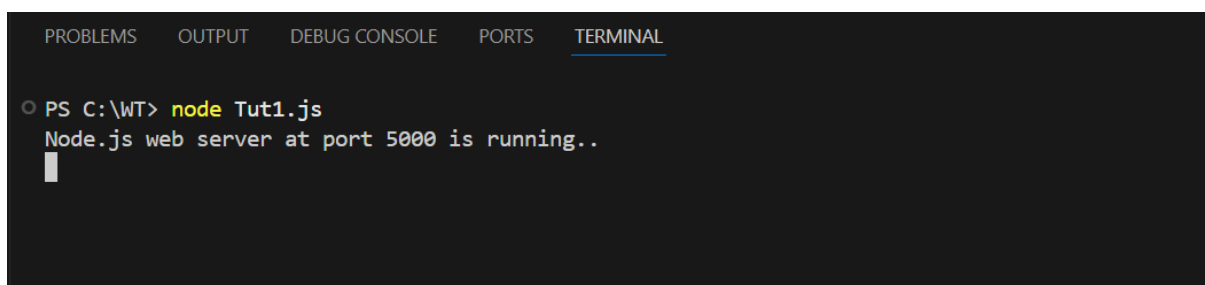
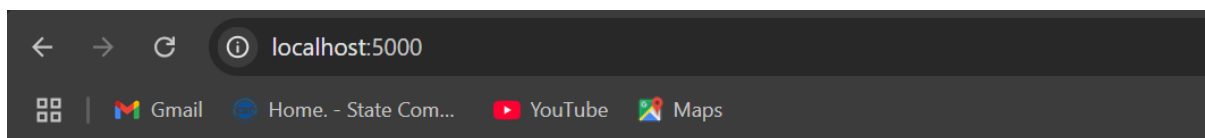
a) understand http request module :

```
var http = require('http'); // 1 - Import Node.js core module

var server = http.createServer(function (req, res) { // 2 - creating server
    //handle incoming 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 :

A screenshot of a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'PORTS', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal shows a command prompt 'PS C:\WT>' followed by the command 'node Tut1.js'. Below the command, the output 'Node.js web server at port 5000 is running..' is displayed, followed by a cursor line.

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();

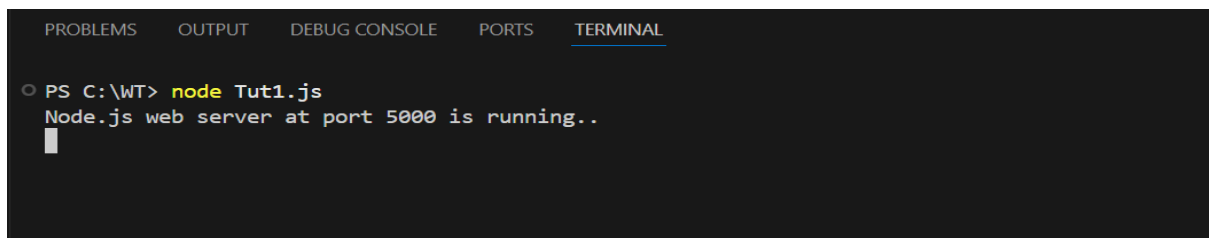
}
else if (req.url == "/admin") {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<html><body><p>This is admin Page.</p></body></html>');
    res.end();
}
else
    res.end('Invalid Request!');
});

```

server.listen(5000); //6 - listen for any incoming requests

console.log('Node.js web server at port 5000 is running..')

Output :



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  TERMINAL
○ PS C:\WT> node Tut1.js
Node.js web server at port 5000 is running..

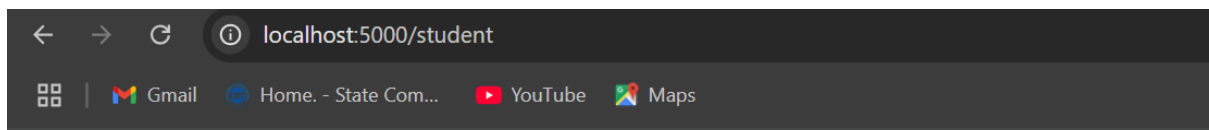
```

Home page :



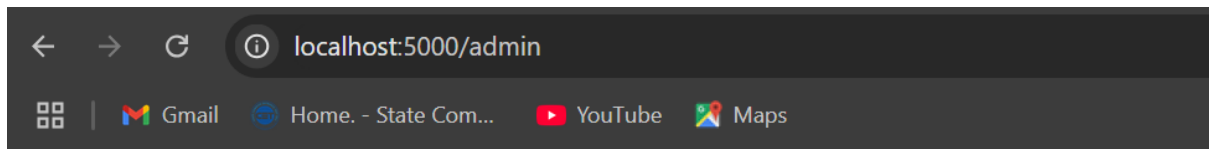
This is home Page.

Student page :



This is student Page.

Admin page :



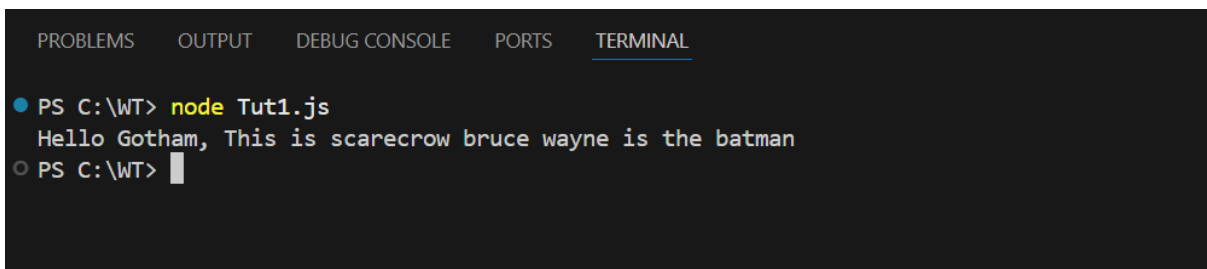
This is admin Page.

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 :

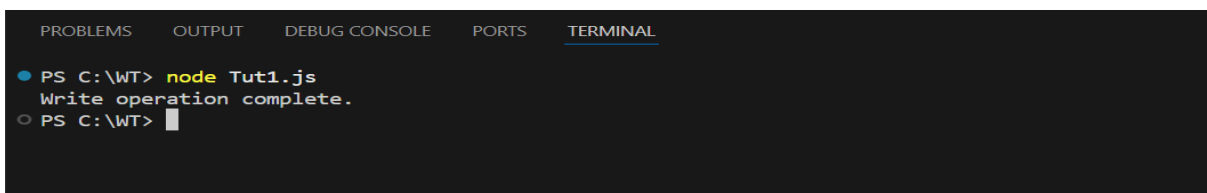


```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL  
● PS C:\WT> node Tut1.js  
Hello Gotham, This is scarecrow bruce wayne is the batman  
○ PS C:\WT> █
```

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 :



```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL  
● PS C:\WT> node Tut1.js  
Write operation complete.  
○ PS C:\WT> █
```

```
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 :

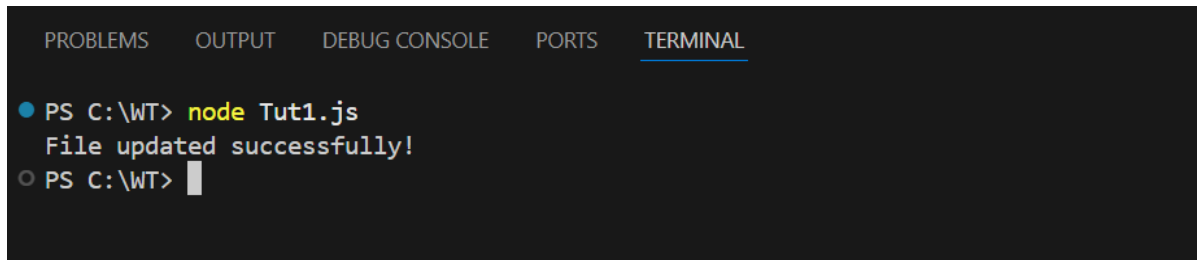
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  TERMINAL
● PS C:\WT> node Tut1.js
  delete operation complete.
○ PS C:\WT>
```

d) Update file

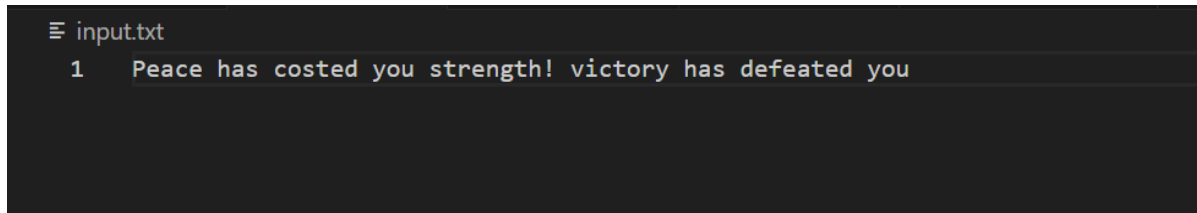
```
const fs = require('fs');
function updateFile(filePath, newContent) {
    fs.writeFile(filePath, newContent, 'utf8', (err) => {
        if (err) {
            console.error(`Error updating the file: ${err.message}`);
        } else {
            console.log('File updated successfully!');
       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

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

Output :



The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The command prompt shows 'PS C:\WT> node Tut1.js' followed by the output 'File updated successfully!'. The prompt then shows 'PS C:\WT>' with a cursor.



The screenshot shows a VS Code editor window with the file 'input.txt' open. The file contains a single line of text: '1 Peace has costed you strength! victory has defeated you'.

Open a file :

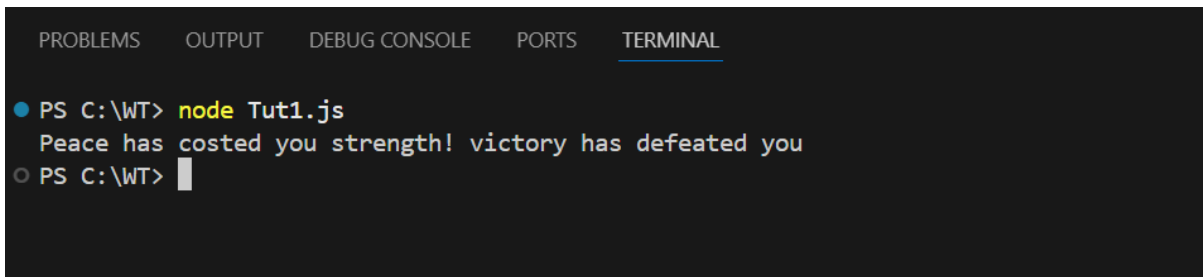
```
const fs = require('fs');
fs.open('input.txt', 'r', (err, fd) => {
  if (err) {
    console.error('Error opening file:', err.message);
    return;
  }
  // Allocate buffer with modern API
  const buffer = Buffer.alloc(10240);
  fs.read(fd, buffer, 0, buffer.length, 0, (err, bytesRead) => {
    if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



The screenshot shows a VS Code interface with the 'TERMINAL' tab selected. The terminal output is as follows:

```

PS C:\WT> node Tut1.js
Peace has costed you strength! victory has defeated you
PS C:\WT>

```

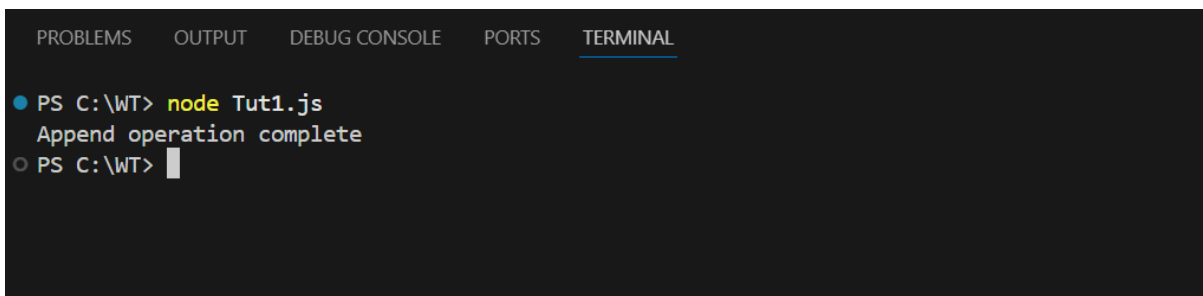
e) Append to a file.

```

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 :



The screenshot shows a VS Code interface with the 'TERMINAL' tab selected. The terminal output is as follows:

```

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

```

≡ input.txt

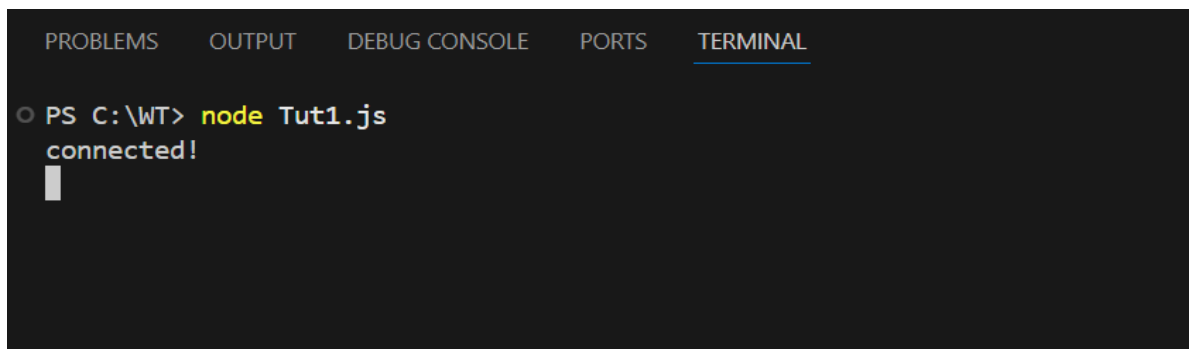
1 Peace has costed you strength! victory has defeated you, This is Bane Speaking

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 :

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'PORTS', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal shows a command prompt 'PS C:\WT>' followed by the command 'node Tut1.js'. Below the command, the output 'connected!' is displayed, followed by a cursor line.

```
PS C:\WT> node Tut1.js  
connected!  
|
```

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;

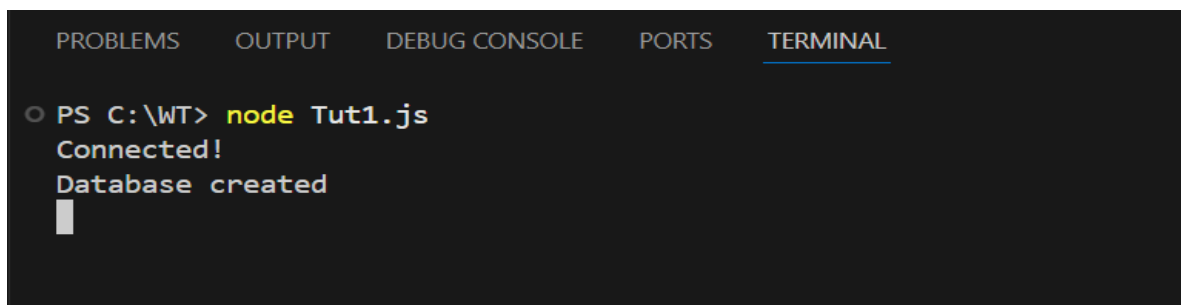
  console.log("Database created");

});

});

```

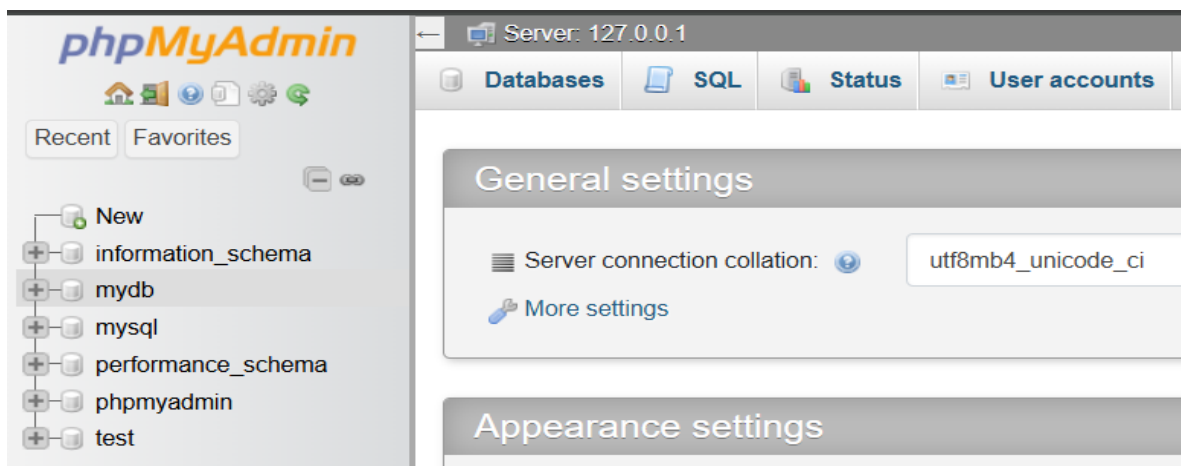
Ouptut :



```

PS C:\WT> node Tut1.js
Connected!
Database created

```



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;

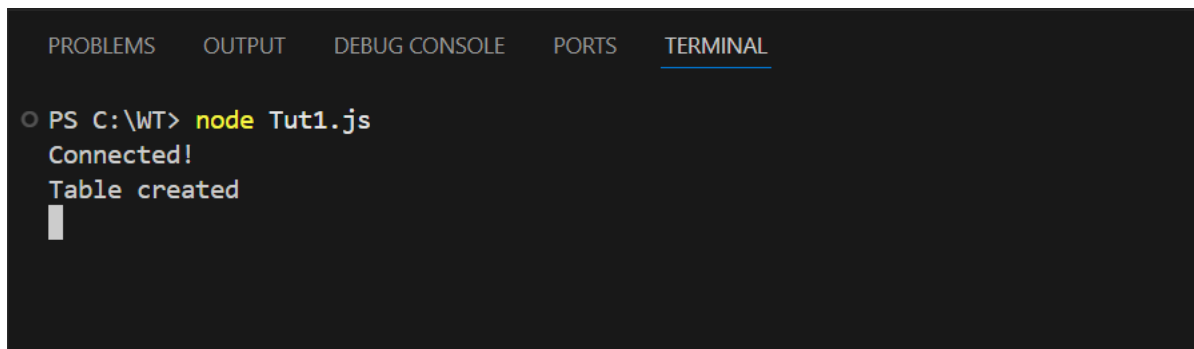
    console.log("Table created");

  });

});

```

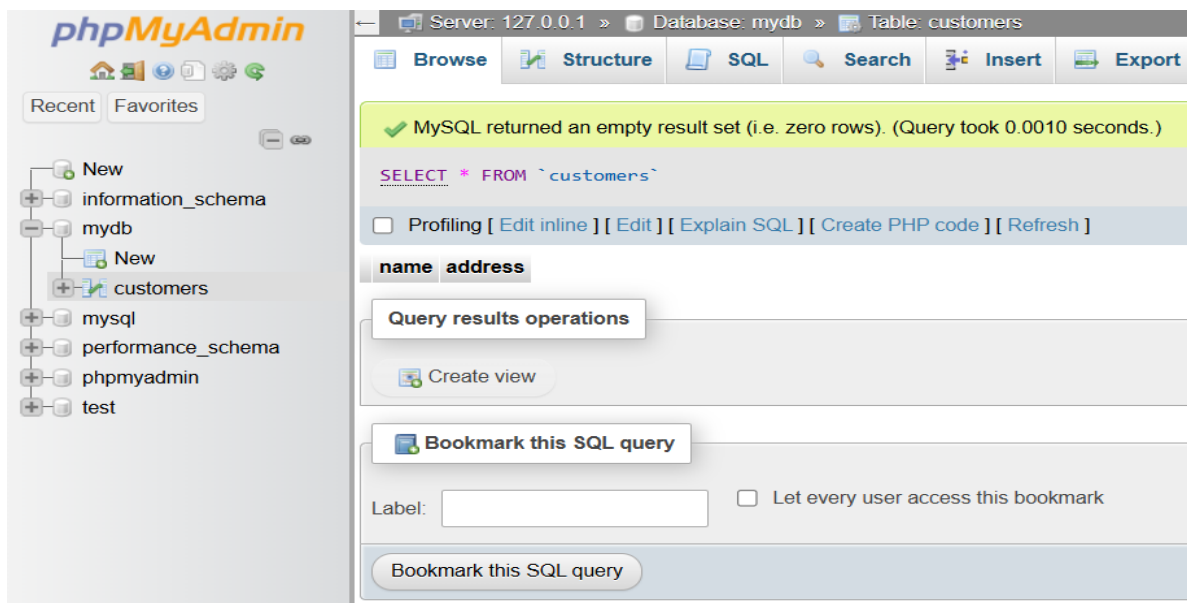
Output :



```

PS C:\WT> node Tut1.js
Connected!
Table created

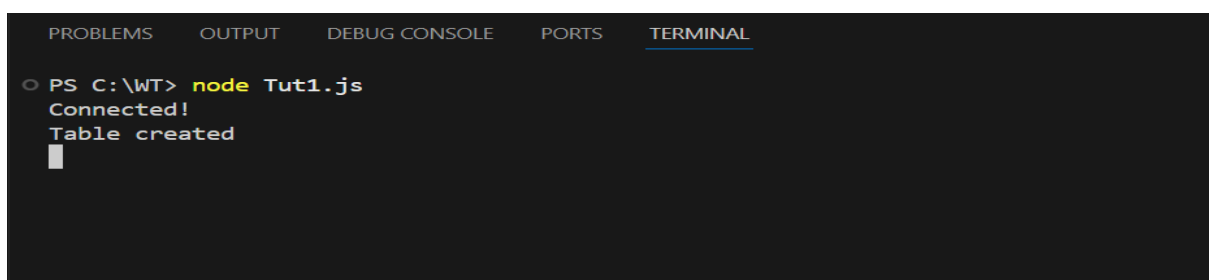
```



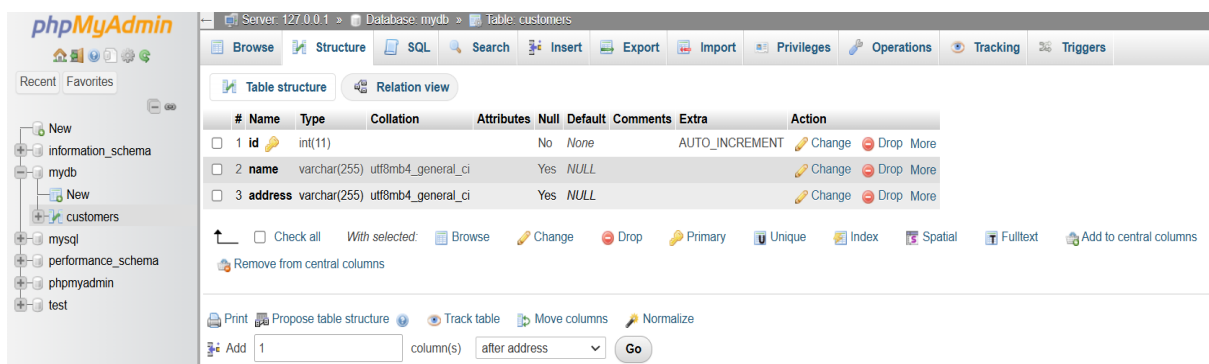
d) Create table with primary key.

```
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 :



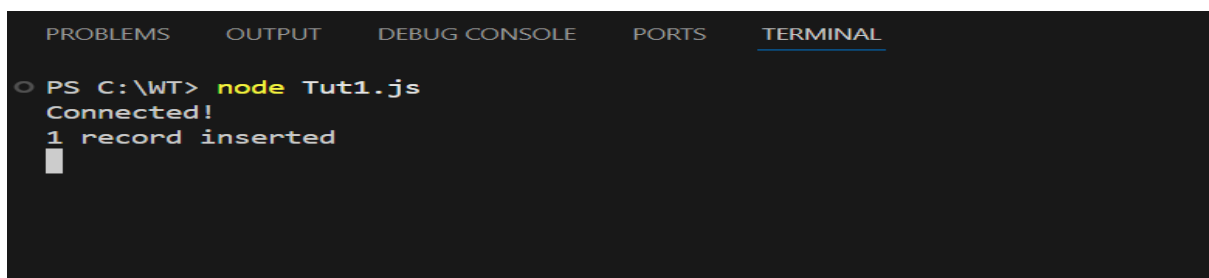
A terminal window with a dark background. The title bar shows 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'PORTS', and 'TERMINAL'. The terminal text is:
PS C:\WT> node Tut1.js
Connected!
Table created



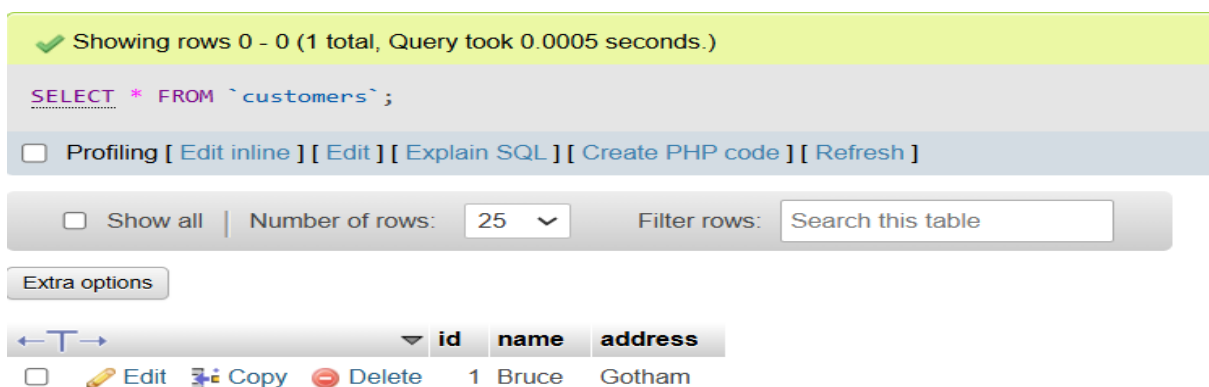
e) Insert a record into table.

```
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 :



```
PS C:\WT> node Tut1.js  
Connected!  
1 record inserted
```



Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT * FROM `customers`;
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

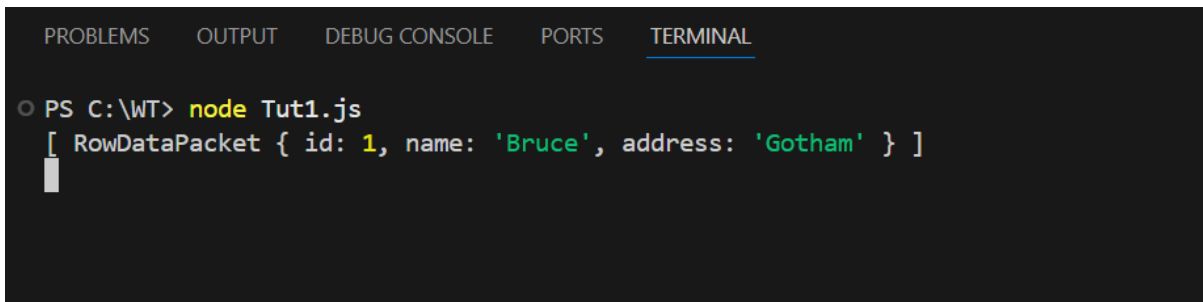
Extra options

	id	name	address
<input type="checkbox"/> Edit Copy Delete	1	Bruce	Gotham

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 :



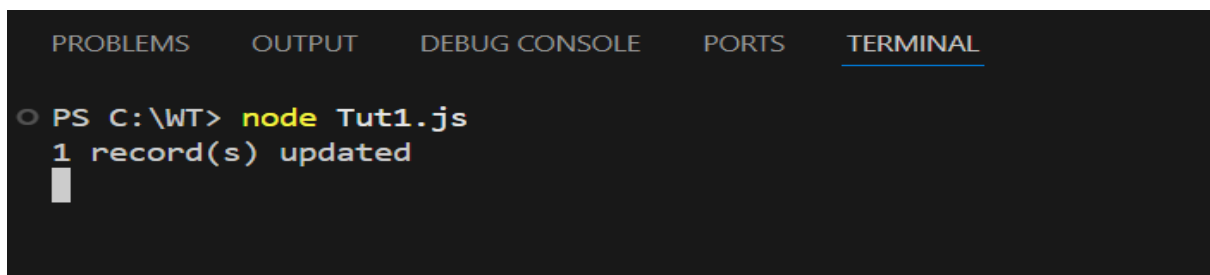
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  TERMINAL
PS C:\WT> node Tut1.js
[ RowDataPacket { id: 1, name: 'Bruce', address: 'Gotham' } ]
```

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 :



```
PS C:\WT> node Tut1.js
1 record(s) updated
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

`SELECT * FROM `customers`;`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	id	name	address
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Bruce	Wayne Tower

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 :



The image shows two screenshots. The top screenshot is a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, PORTS, and TERMINAL. The terminal shows the command `PS C:\WT> node Tut1.js` and the output `Number of records deleted: 1`. The bottom screenshot is a MySQL query result interface. It shows a green checkmark and the message "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)". Below this, the query `SELECT * FROM `customers`;` is displayed. There are links for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. At the bottom, a table header is shown with columns `id`, `name`, and `address`.