

Q11) Create index.html file containing two forms for SignIn and SignUp. Submitting SignIn form should search for credentials in mysql database using server created in previous practical. On successful signin, a welcome page should be displayed. Submitting SignUp form should insert new entry for credentials in mysql database using server created in previous practical. On successful signup, user should be returned back to index.html.

=> index.html file :-

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

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

  <title>Sign In / Sign Up</title>

</head>

<body>

  <h2>Sign In</h2>

  <form action="/signin" method="post">

    <label for="signin-username">Username:</label>

    <input type="text" id="signin-username" name="username" required>

    <br>

    <label for="signin-password">Password:</label>

    <input type="password" id="signin-password" name="password" required>

    <br>

    <button type="submit">Sign In</button>

  </form>

  <h2>Sign Up</h2>

  <form action="/signup" method="post">

    <label for="signup-username">Username:</label>

    <input type="text" id="signup-username" name="username" required>

    <br>

    <label for="signup-password">Password:</label>

    <input type="password" id="signup-password" name="password" required>
```

```
<br>

<button type="submit">Sign Up</button>

</form>

</body>

</html>
```

==> JS file :-

```
const http = require('http');
const fs = require('fs');
const url = require('url');
const mysql = require('mysql');
const path = require('path');
// Create a MySQL connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'K1@2r3t4i5k6e7y8',
  database: 'kartikey_18',
});
// Connect to the database
db.connect((err) => {
  if (err) {
    console.error('Database connection error:', err);
  } else {
    console.log('Connected to the database');
  }
});
const server = http.createServer((req, res) => {
  const parsedUrl = url.parse(req.url, true);
  const pathname = parsedUrl.pathname;
  const filePath = path.join(__dirname, pathname);
```

```

if (req.method === 'GET') {
  if (pathname === '/') {
    // Serve index.html for the root path
    fs.readFile(path.join(__dirname, 'index11.html'), (err, data) => {
      if (err) {
        res.writeHead(500, { 'Content-Type': 'text/plain' });
        res.end('Internal Server Error');
      } else {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.end(data);
      }
    });
  } else if (pathname === '/welcome') {
    // Serve a welcome page for successful signin or signup
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.end('<h1>Welcome!</h1>');
  } else {
    // Serve static files
    fs.readFile(filePath, (err, data) => {
      if (err) {
        res.writeHead(404, { 'Content-Type': 'text/plain' });
        res.end('Not Found');
      } else {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.end(data);
      }
    });
  }
} else if (req.method === 'POST') {
  if (pathname === '/signin') {
    // Handle Sign In form submission

```

```

let body = '';

req.on('data', (chunk) => {

  body += chunk;

});

req.on('end', () => {

  const formData = new URLSearchParams(body);

  const username = formData.get('username');

  const password = formData.get('password');

  // Query the database to check credentials

  db.query(

    'SELECT * FROM users WHERE username = ? AND password = ?',

    [username, password],

    (err, results) => {

      if (err) {

        console.error('Database query error:', err);
        res.writeHead(500, { 'Content-Type': 'text/plain' });

        res.end('Internal Server Error');

      } else if (results.length > 0) {

        // Successful sign-in

        res.writeHead(302, { Location: '/welcome' });

        res.end();

      } else {

        // Invalid credentials

        res.writeHead(401, { 'Content-Type': 'text/plain' });

        res.end('Unauthorized');

      }

    }

  );

});

} else if (pathname === '/signup') {

  // Handle Sign Up form submission

```

```

let body = '';
req.on('data', (chunk) => {
  body += chunk;
});
req.on('end', () => {
  const formData = new URLSearchParams(body);
  const username = formData.get('username');
  const password = formData.get('password');
  // Insert new entry into the database
  db.query('INSERT INTO users (username, password) VALUES (?, ?)', [username, password], (err) => {
    if (err) {
      console.error('Database insertion error:', err);
      res.writeHead(500, { 'Content-Type': 'text/plain' });
      res.end('Internal Server Error');
    } else {
      // Successful signup
      res.writeHead(302, { Location: '/' });
      res.end();
    }
  });
});
});
const PORT = 10000;
server.listen(PORT, () => {
  console.log(`Server is running at http://localhost:${PORT}/`);
});

```

← → ↻ ⓘ localhost:10000

Sign In

Username:

Password:

Sign Up

Username:

Password: ⓘ

← → ↻ ⓘ localhost:10000

Sign In

Username:

Password: ⓘ

Sign Up

Username:

Password:

← ↻ ⓘ localhost:10000/welcome

Welcome!

```
mysql> select * from users;  
+-----+-----+  
| username      | password |  
+-----+-----+  
| kartikey_701  | kartikey |  
+-----+-----+  
1 row in set (0.00 sec)
```