

❑ FULL CRUD APP with login and logout:-

Here's a **full CRUD (Create, Read, Update, Delete)** example using **Express.js, MySQL, and EJS** — all in a clean, beginner-friendly format.

Here is your **full updated app.js code** with:

- CRUD functionality
- Login & logout
- Route protection using session
- User display with form

❑ FULL CRUD APP with login and logout : Users Table:-

❑ Database Setup

In MySQL:

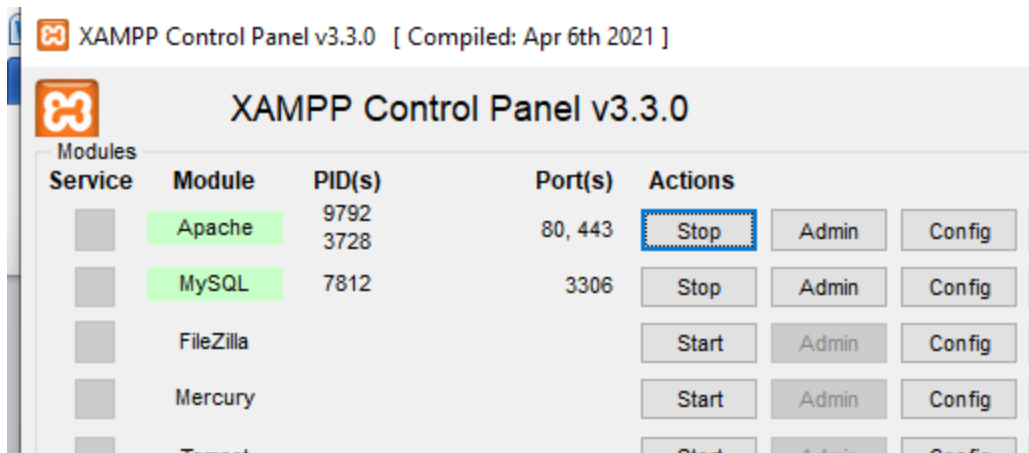
```
CREATE DATABASE omsir;

USE omsir;

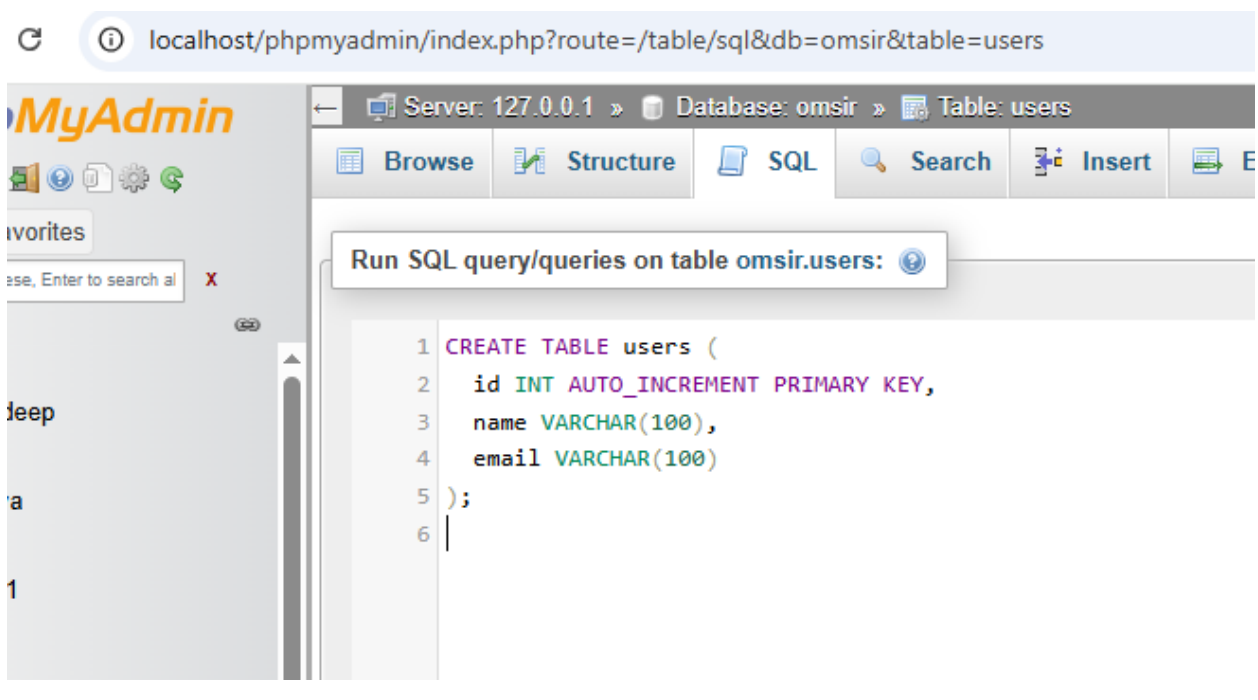
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100)
  password VARCHAR(100)
);
```

As shown below :-

Run xampp software and start as shown below



And open your localhost/phpmyadmin



Install Needed Packages

In your project folder:

```
npm init -y
```

```
npm install express mysql body-parser ejs
```

Project Structure

```
express-js/  
├── views/  
│   ├── form.ejs  
│   └── edit.ejs  
└── app.js
```

Here is your **full updated app.js code** with:

- CRUD functionality
 - Login & logout
 - Route protection using session
 - User display with form
-

app.js (Full Updated Code):-

```
const express = require('express');  
const mysql = require('mysql');  
const bodyParser = require('body-parser');  
const path = require('path');  
const session = require('express-session');  
  
const app = express();  
const PORT = 3000;  
  
//  MySQL Database Connection  
const db = mysql.createConnection({  
  host: 'localhost',  
  user: 'root',  
  password: '', // Update if you have a MySQL password  
  database: 'testdb'
```

```

});

db.connect(err => {
  if (err) throw err;
  console.log('☐ Connected to MySQL');
});

// ☐ Middleware
app.use(bodyParser.urlencoded({ extended: false }));
app.use(express.static(path.join(__dirname, 'public')));
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

app.use(session({
  secret: 'your-secret-key',
  resave: false,
  saveUninitialized: false
}));

// ☐ Authentication Middleware
function isLoggedIn(req, res, next) {
  if (req.session.loggedIn) {
    next();
  } else {
    res.redirect('/login');
  }
}

// ===== ROUTES ===== //

// ☐ Login page
app.get('/login', (req, res) => {
  res.render('login', { error: null });
});

// ☐ Login form handling
app.post('/login', (req, res) => {
  const { email, password } = req.body;
  const sql = 'SELECT * FROM users WHERE email = ? AND password = ?';
  db.query(sql, [email, password], (err, results) => {
    if (err) throw err;
    if (results.length > 0) {
      req.session.loggedIn = true;
      req.session.user = results[0];
      res.redirect('/');
    } else {
      res.render('login', { error: 'Invalid credentials' });
    }
  });
});

// ☐ Logout
app.get('/logout', (req, res) => {
  req.session.destroy(err => {
    res.redirect('/login');
  });
});

```

```

// □ READ: Home page with form + user list
app.get('/', isLoggedIn, (req, res) => {
  const sql = 'SELECT * FROM users';
  db.query(sql, (err, results) => {
    if (err) throw err;
    res.render('form', { users: results, user: req.session.user });
  });
});

// □ CREATE: Add user
app.post('/add-user', isLoggedIn, (req, res) => {
  const { name, email } = req.body;
  const sql = 'INSERT INTO users (name, email) VALUES (?, ?)';
  db.query(sql, [name, email], err => {
    if (err) throw err;
    res.redirect('/');
  });
});

// □ EDIT: Show edit form
app.get('/edit/:id', isLoggedIn, (req, res) => {
  const sql = 'SELECT * FROM users WHERE id = ?';
  db.query(sql, [req.params.id], (err, results) => {
    if (err) throw err;
    res.render('edit', { user: results[0] });
  });
});

// □ UPDATE: Submit edited data
app.post('/update-user/:id', isLoggedIn, (req, res) => {
  const { name, email } = req.body;
  const sql = 'UPDATE users SET name = ?, email = ? WHERE id = ?';
  db.query(sql, [name, email, req.params.id], err => {
    if (err) throw err;
    res.redirect('/');
  });
});

// □ DELETE: Remove user
app.get('/delete/:id', isLoggedIn, (req, res) => {
  const sql = 'DELETE FROM users WHERE id = ?';
  db.query(sql, [req.params.id], err => {
    if (err) throw err;
    res.redirect('/');
  });
});

// □ Start the Server
app.listen(PORT, () => {
  console.log(`□ Server running at http://localhost:${PORT}`);
});

```

□ **You Also Need These Files:-**

1. `views/login.ejs`:-

```
<!DOCTYPE html>
<html>
<head>
  <title>Login</title>
</head>
<body>
  <h2>Login</h2>
  <% if (error) { %>
    <p style="color:red;"><%= error %></p>
  <% } %>
  <form action="/login" method="POST">
    <label>Email:</label><br>
    <input type="email" name="email" required><br><br>
    <label>Password:</label><br>
    <input type="password" name="password" required><br><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
```

2. `views/form.ejs`:-

```
<!DOCTYPE html>
<html>
<head>
  <title>User Dashboard</title>
  <style>
    table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
      padding: 8px;
    }
    table {
      margin-top: 20px;
      width: 60%;
    }
  </style>
</head>
<body>
  <h2>Welcome, <%= user.name %> | <a href="/logout">Logout</a></h2>

  <h3>Add User</h3>
  <form action="/add-user" method="POST">
    <label>Name:</label><br>
    <input type="text" name="name" required><br><br>
    <label>Email:</label><br>
    <input type="email" name="email" required><br><br>
    <input type="submit" value="Submit">
  </form>

  <h3>User List</h3>
  <table>
    <tr>
```

```

    <th>ID</th><th>Name</th><th>Email</th><th>Actions</th>
  </tr>
  <% if (users.length > 0) { %>
    <% users.forEach(user => { %>
      <tr>
        <td><%= user.id %></td>
        <td><%= user.name %></td>
        <td><%= user.email %></td>
        <td>
          <a href="/edit/<%= user.id %>">Edit</a> |
          <a href="/delete/<%= user.id %>" onclick="return confirm('Delete
this user?')">Delete</a>
        </td>
      </tr>
    <% }) %>
  <% } else { %>
    <tr><td colspan="4">No users found.</td></tr>
  <% } %>
</table>
</body>
</html>

```

3. views/edit.ejs:-

```

<!DOCTYPE html>
<html>
<head>
  <title>Edit User</title>
</head>
<body>
  <h2>Edit User</h2>
  <form action="/update-user/<%= user.id %>" method="POST">
    <label>Name:</label><br>
    <input type="text" name="name" value="<%= user.name %>" required><br><br>
    <label>Email:</label><br>
    <input type="email" name="email" value="<%= user.email %>"
required><br><br>
    <input type="submit" value="Update">
  </form>
  <br>
  <a href="/">Cancel</a>
</body>
</html>

```

☐ Test Login

Insert a test login user manually in MySQL:

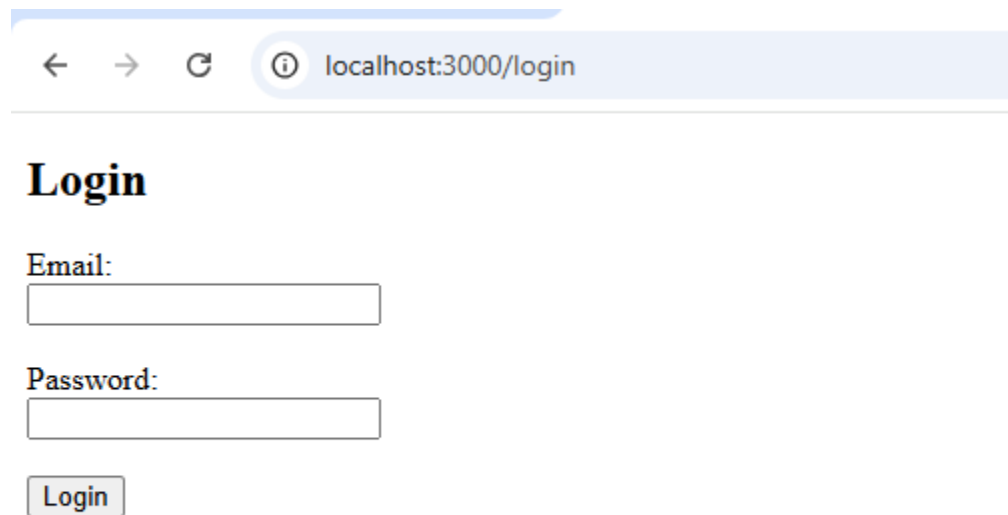
```
INSERT INTO users (name, email, password) VALUES ('Admin',  
'admin@example.com', 'admin123');
```

Then visit <http://localhost:3000/login>

Login using:

- Email: admin@example.com
- Password: admin123

Output:-



The screenshot shows a web browser window with the address bar displaying "localhost:3000/login". The page content includes a heading "Login", followed by an "Email:" label and an empty text input field. Below that is a "Password:" label and another empty text input field. At the bottom of the form is a "Login" button.

After login :-

Welcome, Admin | [Logout](#)

Add User

Name:

Email:

User List

ID	Name	Email	Actions
1	om sir	omnetworkom@gmail.com	Edit Delete
4	Admin	admin@example.com	Edit Delete