

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

❑ FULL CRUD APP: Users Table

❑ Database Setup

In MySQL:

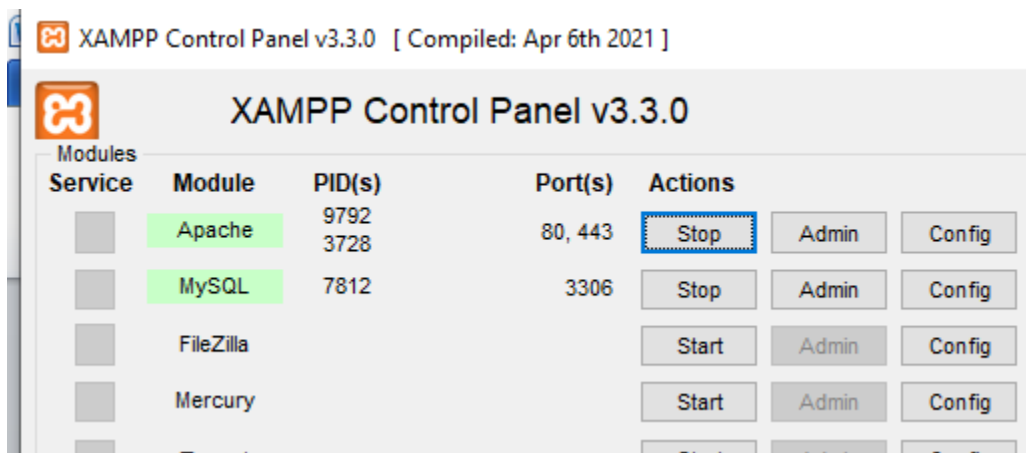
```
CREATE DATABASE omsir;

USE omsir;

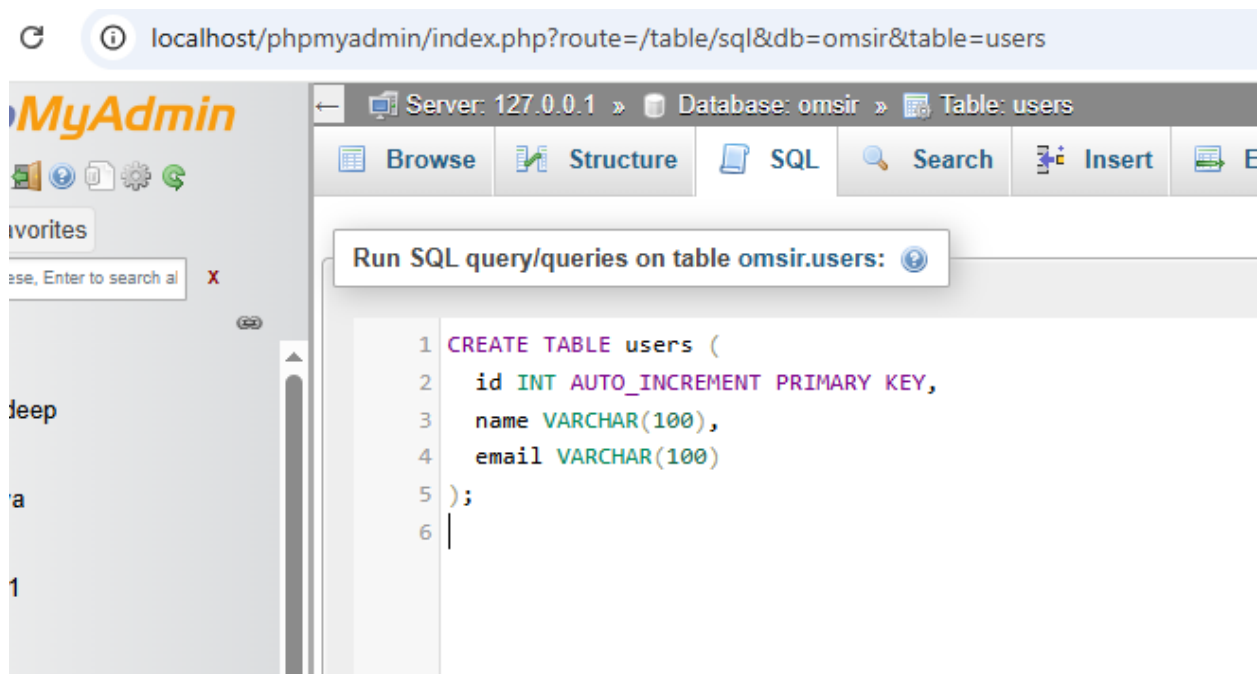
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  email VARCHAR(100)
);
```

As shown below :-

Run xampp software and start as shown below



And open your localhost/phpyadmin



☐ 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
```

❑ 1. app.js – Full Express Logic:-

```
const express = require('express');
const mysql = require('mysql');
const bodyParser = require('body-parser');
const path = require('path');

const app = express();
const PORT = 3000;

// MySQL connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '', // your MySQL password
  database: 'testdb'
});

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

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

// ❑ READ: Display form and user list
app.get('/', (req, res) => {
  const sql = 'SELECT * FROM users';
  db.query(sql, (err, results) => {
    if (err) throw err;
    res.render('form', { users: results });
  });
});

// ❑ CREATE: Insert new user
app.post('/add-user', (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('/');
  });
});

// ❑ UPDATE: Show edit form
app.get('/edit/:id', (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: Save edited user
app.post('/update-user/:id', (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', (req, res) => {
  const sql = 'DELETE FROM users WHERE id = ?';
  db.query(sql, [req.params.id], err => {
    if (err) throw err;
    res.redirect('/');
  });
});

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

□ 2. views/form.ejs – Create + Read + Delete + Edit Links

```
<!DOCTYPE html>
<html>
<head>
  <title>User Management</title>
  <style>
    table { border-collapse: collapse; width: 60%; margin-top: 20px; }
    th, td { border: 1px solid #ccc; padding: 10px; text-align: left; }
    form { margin-bottom: 20px; }
  </style>
</head>
<body>
  <h2>Add User</h2>
  <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>

  <h2>User List</h2>
  <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 – Update User Form


```

<!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>
    <a href="/">Cancel</a>
</body>
</html>

```

☐ **Done!**

☐ **Features:**

- ☐ Add users
- ☐ List users
-  ☐ Edit users
- ☐ Delete users

localhost:3000

Add User

Name:

Email:

Submit

User List

ID	Name	Email	Actions
1	om sir	omnetworkom@gmail.com	Edit Delete
3	NARGIS digital	narigs@gmail.com	Edit Delete