React JS, Node JS, Express JS, MYSQL

1. Set Up the Project Structure

bash

Copy code

```
project-folder/
│
├── backend/
│ ├── server.js
│ ├── package.json
│ ├── config/
│ │ └── db.js
│ ├── routes/
│ │ └── users.js
│ └── models/
│ └── userModel.js
│
└── frontend/
├── src/
│ ├── App.js
│ ├── index.js
│ ├── components/
│ │ ├── UserForm.js
│ │ └── UserList.js
└── package.json
```

2. Backend Setup (Node.js + Express.js + MySQL)

Install Dependencies

Navigate to the backend folder and initialize your Node.js project:

bash

Copy code

```
cd backend
npm init -y
npm install express mysql2 cors
```

Database Configuration (config/db.js)

javascript

Copy code

```
const mysql = require('mysql2');
const connection = mysql.createConnection({
host: 'localhost',
user: 'root',
password: 'your_password',
database: 'your_database'
});
connection.connect((err) => {
if (err) throw err;
console.log('Connected to MySQL');
});
module.exports = connection;
```

User Model (models/userModel.js)

javascript

Copy code

```
const db = require('../config/db');
const UserModel = {
getAllUsers: (callback) => {
const sql = 'SELECT * FROM users';
db.query(sql, callback);
},
createUser: (user, callback) => {
const sql = 'INSERT INTO users SET ?';
db.query(sql, user, callback);
},
updateUser: (id, user, callback) => {
const sql = 'UPDATE users SET ? WHERE id = ?';
db.query(sql, [user, id], callback);
},
deleteUser: (id, callback) => {
const sql = 'DELETE FROM users WHERE id = ?';
db.query(sql, id, callback);
}
};
module.exports = UserModel;
```

User Routes (routes/users.js)

javascript

Copy code

```
const express = require('express');
const router = express.Router();
const UserModel = require('../models/userModel');
router.get('/', (req, res) => {
UserModel.getAllUsers((err, users) => {
if (err) res.status(500).send(err);
res.json(users);
});
});
router.post('/', (req, res) => {
const newUser = req.body;
UserModel.createUser(newUser, (err, result) => {
if (err) res.status(500).send(err);
res.json({ id: result.insertId, ...newUser });
});
});
router.put('/:id', (req, res) => {
const id = req.params.id;
const updatedUser = req.body;
UserModel.updateUser(id, updatedUser, (err, result) => {
```

```
if (err) res.status(500).send(err);
res.json({ id, ...updatedUser });
});
});
router.delete('/:id', (req, res) => {
const id = req.params.id;
UserModel.deleteUser(id, (err, result) => {
if (err) res.status(500).send(err);
res.json({ message: 'User deleted' });
});
});
module.exports = router;
```

Server Setup (server.js)

javascript

Copy code

```
const express = require('express');
const cors = require('cors');
const bodyParser = require('body-parser');
const userRoutes = require('./routes/users');
const app = express();
app.use(cors());
app.use(bodyParser.json());
app.use('/api/users', userRoutes);
const PORT = 5000;
app.listen(PORT, () => {
console.log(`Server running on port ${PORT}`);
});
```

3. Frontend Setup (React.js)

Initialize React Project

Navigate to the frontend folder and create a React app:

bash

Copy code

```
cd frontend
npx create-react-app .
npm install axios
```

App Component (src/App.js)

javascript

Copy code

```
import React from 'react';
import UserForm from './components/UserForm';
import UserList from './components/UserList';
function App() {
return (
<div className="App">
<h1>User Management</h1>
<UserForm />
<UserList />
</div>
);
}
export default App;
```

UserForm Component (src/components/UserForm.js)

javascript

Copy code

```
import React, { useState } from 'react';
import axios from 'axios';
function UserForm() {
const [name, setName] = useState('');
const [email, setEmail] = useState('');
const handleSubmit = async (e) => {
e.preventDefault();
const newUser = { name, email };
await axios.post('http://localhost:5000/api/users', newUser);
setName('');
setEmail('');
};
return (
<form onSubmit={handleSubmit}>
<input
type="text"
placeholder="Name"
value={name}
onChange={(e) => setName(e.target.value)}
/>
<input
type="email"
placeholder="Email"
value={email}
onChange={(e) => setEmail(e.target.value)}
/>
<button type="submit">Add User</button>
</form>
);
}
export default UserForm;
```

UserList Component (src/components/UserList.js)

javascript

Copy code

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
function UserList() {
const [users, setUsers] = useState([]);
useEffect(() => {
const fetchUsers = async () => {
```

```
const response = await axios.get('http://localhost:5000/api/users');
setUsers(response.data);
};
fetchUsers();
}, []);
return (
<div>
<h2>User List</h2>
<ul>
{users.map((user) => (
<li key={user.id}>{user.name} - {user.email}</li>
))}
</ul>
</div>
);
}
export default UserList;
```

4. Database Setup (MySQL)

Create a MySQL database and a users table:

sql

Copy code

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

5. Running the Application

Start the backend server by navigating to the backend directory and running:

bash

Copy code

```
node server.js
```

Start the React frontend by navigating to the frontend directory and running:

bash

Copy code

```
npm start
```

6. Testing

You should now be able to open the React application in your browser, add new users via the form, and see the list of users updated in real-time.