Html code:

@{

ViewBag.Title = "CheckTest";

}

<h2>Practice</h2>

<div>

<div>

<label for="userId">Id:</label>

<input type="number" id="userId" />

</div>

<div>

<label for="firstname">Firstname:</label>

<input type="text" id="firstname" />

</div>

<div>

<label for="lastname">Lastname:</label>

<input type="text" id="lastname" />

</div>

<div>

<label for="email">Email:</label>

<input type="text" id="email" />

</div>

<div>

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

<input type="text" id="username" />

</div>

<div>

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

<input type="password" id="password" />

</div>

<div>

<button onclick="Create()">Create</button>

<button onclick="Put()">Uodate</button>

<button onclick="Delete()">Delete</button>

</div>

</div>

<div>

<table class="table">

<thead>

<tr>

<th>ID</th>

<th>FirstName</th>

<th>LastName</th>

<th>Email</th>

<th>Username</th>

<th>Password</th>

</tr>

</thead>

<tbody id="tableBody"></tbody>

</table>

</div>

<script>

async function getUser() {

try {

const response = await fetch("http://localhost:62348/api/User")

if (response.ok) {

const data = await response.json()

const tableBody = document.getElementById("tableBody")

tableBody.innerHTML = ""

data.forEach((u) => {

tableBody.innerHTML += `<tr>

<td>${u.Id}</td>

<td>${u.Firstname}</td>

<td>${u.Lastname}</td>

<td>${u.Email}</td>

<td>${u.Username}</td>

<td>${u.Password}</td>

</tr>`

})

}

} catch (error) {

console.log(error)

}

}

document.addEventListener("DOMContentLoaded", getUser())

async function Create() {

try {

const user = {

FirstName: document.getElementById("firstname").value,

LastName: document.getElementById("lastname").value,

Email: document.getElementById("email").value,

Username: document.getElementById("username").value,

Password: document.getElementById("password").value,

}

const response = await fetch("http://localhost:62348/api/User", {

method:"POST",

headers: {

"Content-Type":"application/json"

},

body: JSON.stringify(user)

})

if (response.ok) {

getUser()

}

} catch (err) {

console.log(err)

}

}

async function Put() {

try {

const uid = document.getElementById("userId").value

const user = {

FirstName: document.getElementById("firstname").value,

LastName: document.getElementById("lastname").value,

Email: document.getElementById("email").value,

Username: document.getElementById("username").value,

Password: document.getElementById("password").value,

}

const response = await fetch(`http://localhost:62348/api/User?id=${uid}`, {

method: "PUT",

headers: {

"Content-Type":"application/json"

},

body: JSON.stringify(user)

})

if (response.ok) {

getUser()

}

} catch (err) {

console.log(err)

}

}

async function Delete() {

try {

const uid = document.getElementById("userId").value

const user = {

Id: document.getElementById("userId").value

}

const response = await fetch(`http://localhost:62348/api/User?id=${uid}`, {

method: "DELETE",

headers: {

"Content-Type":"application/json"

},

body: JSON.stringify(user)

})

if (response.ok) {

getUser()

}

} catch (err) {

console.log(err)

}

}

</script>

Apis Code:

namespace OEL\_Test\_1.Controllers

{

[Route("api/User")]

public class UserApiController : ApiController

{

private string connectionString = "Data Source=DESKTOP-27TFO05;Initial Catalog=UserData;Integrated Security=True";

[HttpGet]

public IHttpActionResult Get()

{

List<User> users = new List<User>();

using(SqlConnection con=new SqlConnection(connectionString))

{

con.Open();

string query = "select \* from users";

SqlCommand cmd = new SqlCommand(query, con);

SqlDataReader reader = cmd.ExecuteReader();

//reader.open

while (reader.Read())

{

User user = new User();

user.Id = (int)reader["ID"];

user.Firstname = reader["FirstName"].ToString();

user.Lastname = reader["LastName"].ToString();

user.Email = reader["Email"].ToString();

user.Username = reader["Username"].ToString();

user.Password = reader["Password"].ToString();

users.Add(user);

}

con.Close();

}

return Ok(users);

}

[HttpGet]

public IHttpActionResult GetById(int id)

{

using(SqlConnection con=new SqlConnection(connectionString))

{

con.Open();

string query="select \* from users where ID="+id;

SqlCommand cmd = new SqlCommand(query, con);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

User user = new User();

user.Id = (int)reader["ID"];

user.Firstname = reader["FirstName"].ToString();

user.Lastname = reader["LastName"].ToString();

user.Email = reader["Email"].ToString();

user.Username = reader["Username"].ToString();

user.Password = reader["Password"].ToString();

return Ok(user);

}

con.Close();

}

return Ok("Not Found");

}

[HttpPost]

public IHttpActionResult Create(User model)

{

int n = 0;

using (SqlConnection con=new SqlConnection(connectionString))

{

con.Open();

string query = String.Format("Insert into users(FirstName,LastName,Email,Username,Password)values('{0}','{1}','{2}','{3}','{4}')",model.Firstname,model.Lastname,model.Email,model.Username,model.Password);

SqlCommand cmd = new SqlCommand(query, con);

n=cmd.ExecuteNonQuery();

con.Close();

}

if (n == 1)

{

return Ok("Created");

}

else

{

return Ok("Not Created");

}

}

[HttpPut]

public IHttpActionResult Put(User model, int id)

{

using (SqlConnection con = new SqlConnection(connectionString))

{

con.Open();

string query = "SELECT \* FROM users WHERE id=" + id;

SqlCommand cmd = new SqlCommand(query, con);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

reader.Close();

string query2 = string.Format("UPDATE users SET FirstName='{0}', LastName='{1}', Email='{2}', Username='{3}', Password='{4}' WHERE id={5}",

model.Firstname, model.Lastname, model.Email, model.Username, model.Password, id);

SqlCommand cmd2 = new SqlCommand(query2, con);

int n = cmd2.ExecuteNonQuery();

if (n == 1 || n > 0)

{

return Ok("Updated");

}

else

{

return Ok("Not Updated");

}

}

con.Close();

}

return Ok("Not Found");

}

[HttpDelete]

public IHttpActionResult Delete(int id)

{

using (SqlConnection con = new SqlConnection(connectionString))

{

con.Open();

string query = "SELECT \* FROM users WHERE id=" + id;

SqlCommand cmd = new SqlCommand(query, con);

SqlDataReader reader = cmd.ExecuteReader();

if (reader.Read())

{

reader.Close();

string query2 = string.Format("DELETE FROM users WHERE ID={0}", id);

SqlCommand cmd2 = new SqlCommand(query2, con);

int n = cmd2.ExecuteNonQuery();

if (n == 1 || n > 0)

{

return Ok("Deleted");

}

else

{

return Ok("Not Deleted");

}

}

con.Close();

}

return Ok("Not Found");

}

}

}