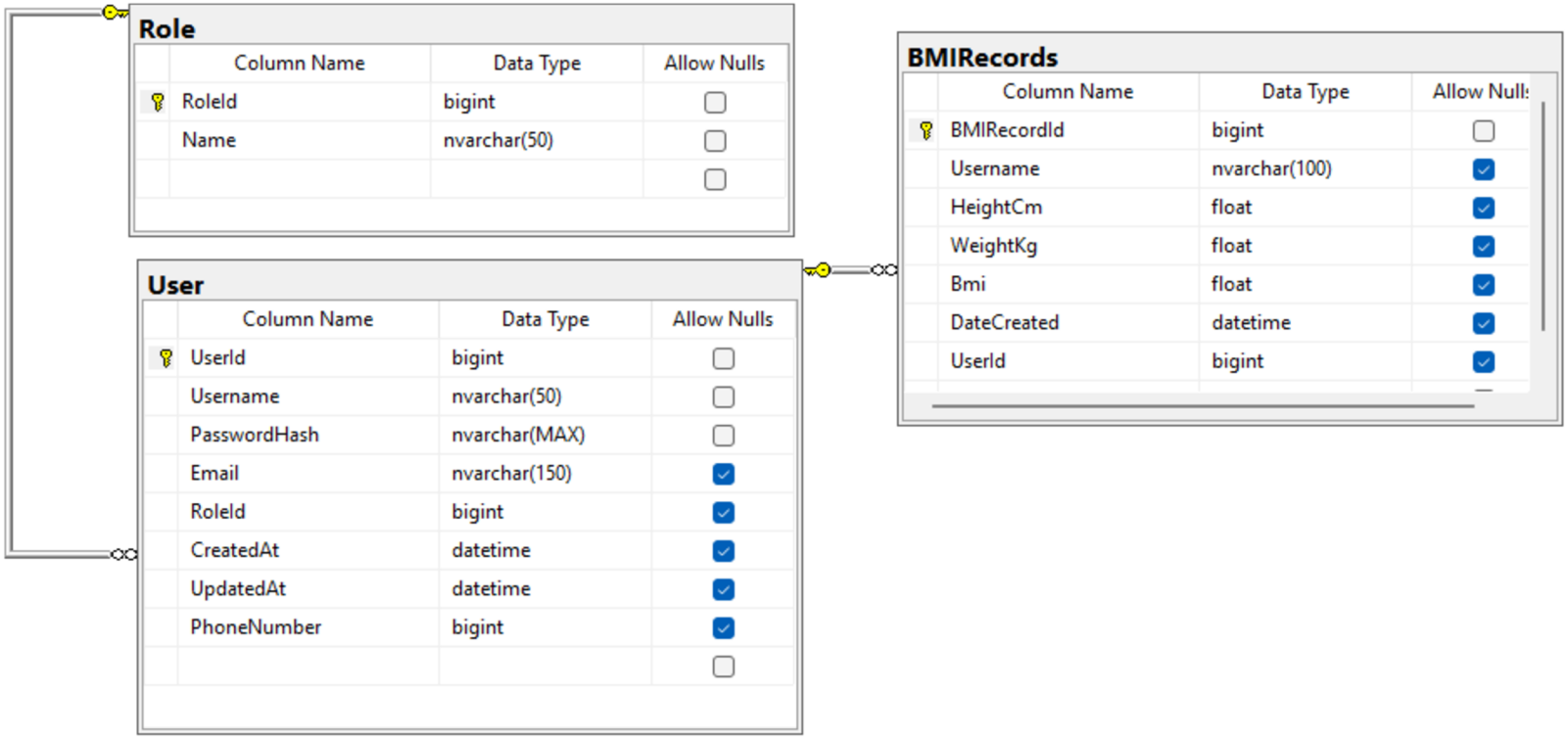
Integrating BMI Tracking with User Authentication in a Flutter & .NET Core Application

Project Overview

- Brief introduction to the BMI tracking application.
- Technologies used: Flutter (Frontend), .NET Core (Backend), SQL Server (Database).
- Objective: Demonstrate how BMI records are securely linked to authenticated users.

Overview

The whole idea is that when someone logs in and adds a BMI, it's saved to their account. No random data is floating around. Just personalized, trackable info.
This diagram shows how everything is connected — each BMI record has a UserId that links to the User table, and users have a RoleId too.



Registering a New User

When someone registers, we're not just storing a password directly — we hash it using BCrypt. That means even if someone broke into the database, they wouldn't get plain passwords. I also chose to return a JWT right away after registration so the user is instantly logged in and doesn't need to sign in again. That keeps the UX smooth.

```
Future<bool> register(String username, String password, String role) async {
  final response = await http.post(
    Uri.parse('$baseUrl/register'),
    headers: {'Content-Type': 'application/json'},
    body: jsonEncode({
      "username": username,
      "passwordHash": password,
      "role": role,
    }),
  );

  return response.statusCode == 200;
}
```

Logging In + Getting the Token

This login system uses the same logic — we check if the user exists, and then verify the password using BCrypt again. If it works, we build a JWT. The cool part is the token contains their UserId in the claims — that's how we later figure out whose data is whose.
Flutter stores this JWT in SharedPreferences, which is like a little local drawer where I stash user info for future requests

```
final response = await http.post(
  Uri.parse('$baseUrl/login'),
  headers: {'Content-Type': 'application/json'},
  body: jsonEncode({
    "username": username,
    "passwordHash": password,
  }),
);
```

```
final prefs = await SharedPreferences.getInstance();
print('userId in prefs: ${prefs.getInt('userId')}');

await prefs.setString('jwt', body['token']);

await prefs.setInt('userId', int.parse(body['userId'].toString()));

await prefs.setString('username', body['username']);
```

In Flutter, we make a POST request to /api/BMIRecords. I send that JSON payload from the Flutter frontend, but in the .NET backend, it gets mapped to the BMIRecord class.

This shows how the app sends the login request and stores the token + user info for later use.

```
var user = _context.User.FirstOrDefault(u => u.Username == loginData.Username);

if (user == null || loginData.PasswordHash == null || !BCrypt.Net.BCrypt.Verify(loginData.PasswordHash, user.PasswordHash))
    return Unauthorized("Invalid email or password.");

var key = Encoding.UTF8.GetBytes(_config["JwtKey"] ?? "this is a secsadfh;adfhlkasdhfluashdfl;iuhfasduasdfukret key for jwt");
```

The token proves "this is me" every time the user wants to save or view data. Without it, the backend wouldn't know who's asking.

Program.cs:

```
builder.Services.AddDbContext<BmiContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));
```

This connects your app to the SQL Server database using the connection string from appsettings.json.

```
"ConnectionStrings": {
  "DefaultConnection": "Data Source=CS-10\\SQLEXPRESS1;Initial Catalog=BMIDb;Integrated Security=True;Connect Timeout=30;Encrypt=True;Trust Server Certificate=True;Application
}
```

Inside bmi_submit_button.dart, the code grabs the values from the bmiController, builds a JSON object, and makes a POST request to the backend:

```dart
final url = Uri.parse('https://localhost:7144/api/BMIRecords/PostRecords');

try {
  final response = await http.post(
    url,
    headers: {
      'Content-Type': 'application/json',
      'Authorization': 'Bearer ${prefs.getString('jwt') ?? ''}',
    },
    body: jsonEncode({
      'userId': userId,
      'username': username,
      'heightCm': bmi.height,
      'weightKg': bmi.weight,
    }),
  );
```

Thank you!