

1. Create the Visit Tracking Middleware

This middleware will handle incrementing the visit count for each page request.

```
using MySql.Data.MySqlClient;
using System;
using System.Threading.Tasks;

public class VisitTrackingMiddleware
{
    private readonly RequestDelegate _next;
    private readonly string _connectionString;

    public VisitTrackingMiddleware(RequestDelegate next, string connectionString)
    {
        _next = next;
        _connectionString = connectionString;
    }

    public async Task InvokeAsync(HttpContext context)
    {
        string pageName = context.Request.Path.Value;

        // Get or update the visit count in the MySQL database
        await TrackVisitAsync(pageName);

        await _next(context);
    }

    private async Task TrackVisitAsync(string pageName)
    {
        using (var connection = new MySqlConnection(_connectionString))
        {
            await connection.OpenAsync();

            // Check if the page exists in the database
            string selectQuery = "SELECT Id, VisitCount FROM PageVisits WHERE PageName = @PageName";
            var command = new MySqlCommand(selectQuery, connection);
            command.Parameters.AddWithValue("@PageName", pageName);
```

```

        using (var reader = await command.ExecuteReaderAsync())
        {
            if (reader.Read())
            {
                // If page exists, update the visit count
                int visitCount = reader.GetInt32("VisitCount");
                int id = reader.GetInt32("Id");
                await UpdateVisitCountAsync(connection, id, visitCount + 1);
            }
            else
            {
                // If page doesn't exist, create a new record
                await CreateNewPageVisitAsync(connection, pageName);
            }
        }
    }
}

private async Task UpdateVisitCountAsync(MySqlConnection connection, int id, int newVisitCount)
{
    string updateQuery = "UPDATE PageVisits SET VisitCount = @VisitCount, LastUpdated = @LastUpdated WHERE Id = @Id";

    var command = new MySqlCommand(updateQuery, connection);
    command.Parameters.AddWithValue("@VisitCount", newVisitCount);
    command.Parameters.AddWithValue("@LastUpdated", DateTime.Now);
    command.Parameters.AddWithValue("@Id", id);

    await command.ExecuteNonQueryAsync();
}

private async Task CreateNewPageVisitAsync(MySqlConnection connection, string pageName)
{
    string insertQuery = "INSERT INTO PageVisits (PageName, VisitCount, LastUpdated) VALUES (@PageName, @VisitCount, @LastUpdated)";

    var command = new MySqlCommand(insertQuery, connection);
    command.Parameters.AddWithValue("@PageName", pageName);
    command.Parameters.AddWithValue("@VisitCount", 1);
    command.Parameters.AddWithValue("@LastUpdated", DateTime.Now);

    await command.ExecuteNonQueryAsync();
}
}

```

4. Configure MySQL Connection in Startup.cs

In the ConfigureServices method of Startup.cs, add a configuration for the connection string.

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddSingleton<VisitTrackingMiddleware>(new VisitTrackingMiddleware(
        next: null,
        connectionString: Configuration.GetConnectionString("DefaultConnection")));
}
```

Make sure to add the connection string in your appsettings.json file.

```
{
  "ConnectionStrings": {
    "DefaultConnection":
    "Server=localhost;Database=your_db_name;User=root;Password=your_password;"
  }
}
```

5. Add the Middleware to the Request Pipeline

In the Configure method of Startup.cs, add your middleware to the pipeline.

```
public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        app.UseHsts();
    }

    // Add the visit tracking middleware
```

```

app.UseMiddleware<VisitTrackingMiddleware>();

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthorization();

app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
});
}

```

6. Retrieve the Visit Count (Optional)

If you want to retrieve and display the visit count on a page, you can do so by querying the PageVisits table directly using ADO.NET:

```

public class HomeController : Controller
{
    private readonly string _connectionString;

    public HomeController(IConfiguration configuration)
    {
        _connectionString = configuration.GetConnectionString("DefaultConnection");
    }

    public async Task<IActionResult> Index()
    {
        string pageName = "/home/index";
        int visitCount = await GetVisitCountAsync(pageName);
        ViewData["VisitCount"] = visitCount;
        return View();
    }

    private async Task<int> GetVisitCountAsync(string pageName)
    {

```

```
using (var connection = new MySqlConnection(_connectionString))
{
    await connection.OpenAsync();

    string selectQuery = "SELECT VisitCount FROM PageVisits WHERE PageName = @PageName";
    var command = new MySqlCommand(selectQuery, connection);
    command.Parameters.AddWithValue("@PageName", pageName);

    var result = await command.ExecuteScalarAsync();
    return result != null ? Convert.ToInt32(result) : 0;
}
}
```