



GYMZONE

Dublin Business School

December 20, 2025

Student Name:Prajwol Shrestha

INTRODUCTION TO WEB

Table of content

1. Summary.....	3
2. What is a web application?.....	3
3. Technologies used and purpose.....	3
4. Alternatives considered.....	4
5. Site architecture and pages.....	4
6. Database design & data flow.....	4
7. Client-side interactivity & accessibility.....	5
9. Testing & debugging.....	5
11. Limitations & future work.....	5
12. Conclusion.....	5

1. Summary

This project **GymZone**, a multi-page ASP.NET Web Forms application is built on visual studio 2022 with the requirements mentioned on the project file. The site demonstrates server-side programming (ASP.NET/C#), client-side interactivity (JavaScript and jQuery), and persistent storage using **Azure SQL Database** during development as an enhancement.. The site implements a MasterPage, a membership form that stores data in SQL, a members display page, a jQuery-powered FAQ, and interactive calculators (BMI and price estimator)

2. What is a web application?

A web application is software delivered over HTTP that combines server-side logic, persistent storage and browser based presentation to provide interactive functionality. The server executes business logic and interacts with databases, while the client (browser) handles presentation and lightweight interactions. Gymzone uses ASP.NET Web Forms for server-side rendering and business logic, HTML/CSS for layout, and JavaScript/jQuery for client interactivity.

3. Technologies used and purpose

ASP.NET Web Forms (C#): Chosen because it meets the module learning outcomes and because it has a fast development model with code-behind separation and server controls that make data binding easier.

Azure SQL Database: Used during development and testing as an enhancement to demonstrate cloud database integration. The Azure DB hosts production-like data while development and final submission use LocalDB for portability.

HTML5 & CSS3: All styles are custom (Flexbox, responsive media queries) to meet the brief prohibition on frameworks such as Bootstrap.

JavaScript & jQuery: Used for small UI improvements like the FAQ toggle, the testimonial rotator, the BMI and price calculators, and more. jQuery is only used as a lightweight DOM helper, not as a UI framework.

ADO.NET (SqlConnection/SqlCommand): Used for secure parameterized database access, preventing SQL injection and enabling maintainable server-side code.

4. Alternatives considered

Alternatives included ASP.NET MVC, full SPA frameworks (React/Angular), and server-side stacks such as Node/Express or PHP. MVC provides better separation of concerns, but Web Forms was selected because it aligns directly with the module brief and the lecturer's expectations. Full front-end frameworks were not used due to brief restrictions.

5. Site architecture and pages

A single MasterPage (`Site.master`) provides consistent header, navigation and footer across pages. The site contains the following pages:

- **Default.aspx (Home):** Hero banner, feature cards, weekly classes, testimonial rotator and newsletter demo.
- **About.aspx:** Mission, facilities, quick facts and testimonials.
- **Memberships.aspx:** Pricing cards for Basic/Standard/Premium plans.
- **Team.aspx:** Staff profiles (minimum three trainers).
- **FAQ.aspx:** jQuery accordion-style FAQ.
- **Join.aspx:** Membership form that inserts into the database (server-side ADO.NET parameterized INSERT).
- **Members.aspx:** Displays members in an ASP.NET GridView, bound in code-behind with `ORDER BY MemberID ASC` to ensure ascending (oldest→newest) display.
- **Calculator.aspx:** BMI and membership-price calculators implemented in jQuery.

6. Database design & data flow

The schema centers on a `Members` table with these fields: `MemberID` (INT IDENTITY, PK), `FullName`, `Email`, `MembershipType`, `StartDate`, and `CreatedAt` (default GETDATE()). The Join page captures user input and uses ADO.NET parameterized SQL to insert a new member. Members.aspx uses `SqlDataAdapter` to fill a `DataTable` which is then bound to a GridView; the `ORDER BY MemberID ASC` clause guarantees stable ordering. During development Azure SQL hosted the database to validate cloud connectivity and multi-machine behavior; for submission LocalDB is used to avoid exposing credentials and to ensure the marker can run the project locally.

7. Client-side interactivity & accessibility

jQuery features include an FAQ collapse/expand, a BMI calculator, a membership price estimator, a testimonial rotator, and small UI animations. All interactive elements degrade gracefully if JavaScript is unavailable. Accessibility considerations include semantic markup, associated `<label>` elements for form inputs, readable colour contrast, and responsive layout via media queries to support mobile devices.

9. Testing & debugging

Testing covered functional flows (form submit → DB insert → retrieval), client-side scripts, and cross-browser checks (Chrome, Edge, Firefox). Common development issues were handled: jQuery script order (ensured CDN jQuery loads before main.js), browser caching problems (hard refresh or DevTools disable cache), and IIS Express compression issues ([ERR_CONTENT_DECODING_FAILED](#)) resolved by clearing IIS Express cache or using a local jQuery copy.

11. Limitations & future work

This submission omits user authentication and role management to remain within scope and avoid introducing fragile dependencies. Extra improvements could include optional secure authentication (ASP.NET Identity), Azure App Service deployment (hosted web app), class booking functionality, or analytics.

12. Conclusion

GymZone fully meets the assessment brief by delivering a functional ASP.NET Web Forms application that incorporates persistent data storage, client-side interactivity, a MasterPage for consistent layout, and a clean, responsive user interface. All features were implemented using custom HTML, CSS, C#, and jQuery, without the use of prohibited frameworks such as Bootstrap. Azure SQL Database was utilised during development to demonstrate cloud-based data integration. The final solution is well-structured, maintainable, and suitable for academic demonstration and assessment.