

I am building a website for my local wrestling club. Here are some features that I have personally come up with/I want to integrate into the website. It is called Kelowna Wrestling Club. Or KWC for short.

For now I am thinking of just keeping two user groups. A normal person that views the website and an admin account that lets them change around certain stuff. For example, the merch page.

Form submission: Currently the coaches make athlete feedback forms using Google Forms. I have submitted a PDF of what that looks like. But I was thinking, could there be a feedback form anyone can type into anonymously, and the coaches would directly get the feedback sent to them.

Coaches section: have a photo of the coaches and little about them, kind of a section, explaining who they are and why they are there

Liability Waiver: If possible, I am not sure how this would work with legality and how difficult, for example, it is to get electronic signatures. However, ideally we could have the waiver be signed online, or they could print the form and bring it in person.

Merch Page: Every now and then one of the coaches orders custom merch for us. But this page would have to be extremely dynamic. How it would have to work is that you would sign into an admin page, and then on this page you would upload photos of the custom merch, give a short description of what it is (for example, a sweatshirt or a hoodie), and set a price. And then it would update the website dynamically for normal users. In terms of payment, I am not sure how difficult this would be. I am not sure if it is easy to set up where the money directly goes to someone's bank account.

Booking: There needs to be a place where you can book the first session. The way it works in our club is that the first session is free, and then after that you have to pay a monthly fee. This feature should be setup in a way that you pick the time and date and then it sends you a confirmation email for your first session. It should also send an email to the coaching letting them know a new student is coming so they can prepare in advance.

Sign up for subscription: After your first time going, if you feel like you want to come regularly, this is where you would sign up for membership. Now again, I'm not sure how this would work/how complex it would be to set this up. But the full amount should be going directly into one person's bank account

Announcement/News Feed Page: here we can show information like practice cancellations, tournament dates, fee changes, new merch drops etc. Again on this page information should only be able to be updated by admins.

Landing Page: This will be the homepage for the website; it will have the KWC logo, where KWC is located, the club schedule and little about what freestyle wrestling is since some people mix it up with greco wrestling or other martial arts like jujitsu.

Group playlist: I have made a group spotify playlist for the club, so put a link to the playlist.

Tech Stack Overview

****Next.js (React + TypeScript)****

The core framework powering the website. It handles routing, server-side rendering, SEO optimization, and lets you build both the public site and future dynamic/admin features in one codebase. TypeScript ensures type safety and more reliable development.

****Tailwind CSS****

A utility-first CSS framework used to style all pages and components quickly while keeping the codebase consistent, clean, and easy to maintain. Perfect for rapidly building modern UI that matches the club's branding.

****Vercel Hosting****

The website will be deployed on Vercel, which provides seamless hosting for Next.js apps. It handles builds, SSL certificates, and global CDN distribution. Connecting the GoDaddy custom domain is straightforward through Vercel's dashboard.

****Public Assets (Images & Static Files)****

All logos, hero images, and coach pictures will live in the `/public` directory. Next.js optimizes them automatically through `next/image`.

****Expandable Architecture****

Although we are starting with only a landing page, the structure is designed so that future developers can easily add:

- Database integration with Postgres
- Backend API routes (Next.js server actions)
- Admin dashboards
- Authentication
- Payments (Stripe)
- Form handling and email notifications

This stack is modern, scalable, easy for developers to understand, and strong on a résumé since it reflects current industry best practices.

