Astoria Yacht Club Website Modernization Proposal

Executive Summary

This document presents a proof-of-concept for a custom-built website to potentially replace the existing Wix-powered Astoria Yacht Club site. This proposal is designed to explore possibilities and spark conversation—not to criticize the excellent work done by our current website maintainers, who have served our club well with the tools available to them.

Important Disclaimer: This proof-of-concept demonstrates potential capabilities and design directions. Every aspect—from colors and fonts to features and content—is fully customizable and meant only as a starting point for discussion. Much of the current content is placeholder text or draft copy that would need review and revision. We can adopt all, some, or none of these features based on what best serves our club's needs.

Technical Note: The current proof-of-concept is a frontend-only demonstration. The production implementation would be built on Amazon Web Services (AWS), utilizing enterprise-grade cloud infrastructure for hosting, databases, serverless functions, content delivery, and other backend services. This provides unlimited scalability, reliability, and security while maintaining cost-effectiveness.

Accessing the Proof-of-Concept

Live Demo URL: AYC Proof Of Concept Website

Quick Start Guide

1. Open the URL in any modern web browser (Chrome, Safari, Firefox, Edge)
2. The site works on all devices—try it on your phone, tablet, and computer
3. No login required to explore

Complete Feature Walkthrough

Main Navigation Pages **Home Page** (`/`) - Quick action buttons for common tasks - Featured announcements section - "Why Join AYC" benefits overview - Newsletter signup form

About (`/about`) - Club history and mission - Leadership team profiles - Facilities overview - Membership benefits

Membership (`/membership`) - Membership tiers and pricing - Application process - Member benefits detailed - Join now call-to-action

Racing (`/racing`) - Race schedule and results - Fleet information - Racing rules and resources - Competitor standings

Events (`/events`) - Interactive calendar view - Upcoming events list - Event filtering by category - Event detail pages with registration

Gallery (`/gallery`) - Photo albums organized by event/date - Lightbox image viewer - Image descriptions and metadata

Contact (`/contact`) - Contact form - Location map - Office hours - Staff directory

Interactive Demo Features **Demo Section** (`/demo`)

This special section showcases advanced features that could be integrated:

1. **Interactive Forms**
   - Real-time validation
   - File upload capability
   - Multi-step wizards
   - Form submission confirmation
2. **Member Portal Preview**
   - Dashboard concept
   - Personalized content
   - Account management interface
3. **Content Management Preview**
   - How administrators would update content
   - Visual editing capabilities
   - Media library management
4. **Integration Examples**
   - Weather widget
   - Tide tables
   - Social media feeds
   - Email newsletter signup

Why Consider a Custom Solution?

The Current Situation

Our existing Wix website has served us well and represents significant effort by dedicated volunteers. However, as technology evolves and our club's needs grow, it's worth exploring whether a custom solution might better serve our long-term goals.

Comparison Overview

Feature	Wix Platform	Custom AWS Solution
Monthly Cost	$27-$45/month	~$5-15/month (estimated)
Annual Cost	$324-$540/year	~$60-180/year + domain
5-Year Cost	$1,620-$2,700	~$300-900 + domain ($60-75)
Custom Features	Limited to Wix apps	Unlimited possibilities
Performance	Good	Excellent (globally distributed)
Mobile Experience	Good	Excellent (custom-tailored)
SEO Control	Limited	Complete control
Data Ownership	Platform-dependent	100% owned by club
Integration Options	Wix marketplace only	Any service/API

Feature	Wix Platform	Custom AWS Solution
Design Flexibility	Template-based	Completely custom
Load Time	3-5 seconds typical	0.5-1.5 seconds (CDN-accelerated)
Vendor Lock-in	Yes (hard to migrate)	Minimal (standard cloud services)
Scalability	Pay more for growth	Auto-scales with usage
Database	Limited/platform-dependent	Full-featured, club-owned
Backend Functions	Limited/expensive	Serverless, pay-per-use
Security	Platform-managed	Enterprise-grade, customizable

Cost Analysis

Wix Platform Costs

- **Business Basic**: $27/month = $324/year
- **Business Unlimited**: $32/month = $384/year
- **Business VIP**: $45/month = $540/year
- **5-Year Total**: $1,620 - $2,700

Additional potential Wix costs: - Premium apps from marketplace - Increased storage - Additional team member accounts - E-commerce features (if needed) - Database/backend features

Custom AWS Solution Costs Production AWS Implementation (Estimated)

Monthly AWS Services: - **S3 (Static hosting/storage)**: $1-3 - **CloudFront (CDN)**: $1-5 - **Lambda (Serverless functions)**: $0-2 - **DynamoDB or RDS (Database)**: $0-5 - **Route 53 (DNS)**: $0.50 - **API Gateway**: $0-2 - **SES (Email service)**: $0-1 - **CloudWatch (Monitoring)**: $0-2 - **Estimated Monthly Total**: $5-15

Annual Costs: - **AWS Services**: $60-180/year - **Domain Registration**: $12-15/year - **Annual Total**: $72-195/year

5-Year Total: $360-975

Cost Notes: - AWS uses pay-per-use pricing—you only pay for what you consume - Estimates assume modest traffic (typical for yacht club website) - AWS Free Tier covers many services for the first 12 months - Costs scale with actual usage, not fixed tier pricing - No surprise “upgrade required” fees for additional features - Database, authentication, file storage all included in estimates

Potential Savings: $1,260 - $2,340 over 5 years (vs. Wix)

Performance Comparison

Metric	Typical Wix Site	This Custom Site
Initial Page Load	3-5 seconds	0.5-1.5 seconds
Time to Interactive	4-6 seconds	1-2 seconds
Mobile Performance Score	60-75/100	90-100/100
SEO Score	70-80/100	95-100/100

Metric	Typical Wix Site	This Custom Site
Accessibility Score	75-85/100	95-100/100

Note: Scores based on industry standards using Lighthouse testing

Key Advantages of Custom Development

1. Performance

- Lightning-fast page loads improve user experience
- Better mobile performance (critical—60%+ of web traffic is mobile)
- Improved search engine rankings due to speed

2. Cost Savings

- Eliminate recurring monthly/annual platform fees
- No surprise price increases
- No feature paywalls

3. Flexibility

- Build exactly what we need, nothing more, nothing less
- Integrate with any service (weather, tides, race results, member database)
- Add features without checking if an app exists
- Full backend capabilities for member portals, event registration, payment processing, etc.

4. Ownership

- Complete control over our data
- Can move hosting anywhere at any time
- No platform can discontinue features we rely on
- Code is an asset that appreciates, not an expense

5. Modern User Experience

- Designed specifically for yacht club needs
- Optimized for both desktop and mobile
- Accessible to all users, including those with disabilities
- Future-proof technology stack

6. Maintainability

- Content updates can be as simple as editing a form
- Can integrate with a headless CMS for non-technical editors
- Version control means we can track all changes
- Multiple team members can collaborate safely

Addressing Common Concerns

"But our current site works fine"

Absolutely true! This proposal isn't about fixing something broken—it's about exploring whether we could do even better while saving money and gaining capabilities.

"Who will maintain it?"

Development & Setup: I (the author of this proposal) am offering to handle all development work pro-bono as a contribution to the club. This includes: - Building out the full AWS infrastructure - Implementing all features and functionality - Setting up the production environment - Initial content migration from the existing site

Day-to-Day Maintenance: For routine content updates and site management, I will create comprehensive, step-by-step documentation designed specifically for non-technical volunteers. This will include: - Illustrated guides for updating events, news, photos, and other content - Video tutorials for common tasks - Troubleshooting guides for typical issues - Clear instructions written for club members with no technical background

Knowledge Transfer: Should I leave the club or become unavailable: - Complete documentation will be in place for the next volunteer - All code and configurations will be thoroughly commented - The system will be designed to be as self-service as possible - Critical procedures will have multiple layers of documentation - Any club member comfortable with basic computer tasks can handle routine updates

"What if the developer disappears?"

This concern is addressed through comprehensive documentation and standard technology:

Documentation for Non-Technical Users: - **User Manuals**: Step-by-step guides with screenshots for every common task - **Video Walkthroughs**: Screen recordings showing exactly how to update content, add events, upload photos, etc. - **Quick Reference Guides**: One-page cheat sheets for frequent operations - **FAQ Document**: Answers to common questions and solutions to typical issues - **Emergency Contacts**: List of AWS support resources and community forums

For Future Technical Volunteers: - **Complete Codebase Documentation**: Every component and function explained - **Architecture Diagrams**: Visual guides to how the system works - **Setup Guides**: How to access and modify the AWS infrastructure - **Deployment Procedures**: How to push updates and changes - **Maintenance Checklists**: Regular tasks and their schedules

Built-In Safeguards: - All code uses standard, well-documented technology (Next.js, React, AWS) - Any developer familiar with these common tools can understand and modify the system - AWS has extensive official documentation and tutorials - Large community support available online - If needed, professional AWS contractors are widely available - No proprietary or custom frameworks—everything is industry-standard

"Isn't custom development risky?"

This proof-of-concept significantly reduces risk by: - Demonstrating feasibility before commitment - Using stable, mature technologies backed by major companies - Following industry best practices - Providing a working example to build from

"What about our current content and images?"

Everything can be migrated: - Content can be copied over - Images can be transferred - We can run both sites in parallel during transition - No rush to flip a switch—move at comfortable pace

---

Technical Foundation (For the Curious)

Current Proof-of-Concept

The frontend demonstration is built with modern, industry-standard technology:

- **Next.js 14**: Web framework used by Netflix, TikTok, Twitch
- **React**: UI library used by Facebook, Instagram, Airbnb
- **TypeScript**: Reliable, type-safe code
- **Tailwind CSS**: Efficient, consistent styling
- **Vercel**: Temporary hosting for demonstration purposes

Production AWS Architecture

The final implementation would leverage enterprise-grade AWS services:

Frontend & Content Delivery: - **S3**: Static asset hosting with 99.999999999% durability - **CloudFront**: Global CDN with edge locations worldwide for fast content delivery - **Route 53**: DNS management with health checks and failover

Backend & Application Logic: - **Lambda**: Serverless functions for backend logic (no servers to manage) - **API Gateway**: RESTful API management and authentication - **Cognito**: User authentication and member portal access

Data Storage: - **DynamoDB**: NoSQL database for flexible data storage (or RDS for relational needs) - **S3**: Image and document storage with automatic backups

Communication & Integration: - **SES**: Email service for contact forms, newsletters, event notifications - **SNS/SQS**: Event notifications and asynchronous processing - **EventBridge**: Integration with external services

Security & Monitoring: - **WAF**: Web application firewall for security - **CloudWatch**: Monitoring, logging, and alerts - **IAM**: Fine-grained access control - **Certificate Manager**: Free SSL/TLS certificates

Why AWS? - Powers Netflix, Airbnb, NASA, and millions of websites - 99.99% uptime SLA with automatic failover - Used by organizations that require enterprise-grade reliability - Scales automatically from 10 visitors to 10 million - Pay only for what you use—no wasted capacity

This isn't experimental technology—these are battle-tested services trusted by the world's most demanding applications.

Next Steps

This document is intended to open a conversation, not close one. Potential next steps could include:

1. **Feedback Round**: Gather input from board members and key stakeholders
2. **Feature Prioritization**: Decide which POC features matter most
3. **Content Review**: Audit current site content for migration
4. **Pilot Program**: Run the new site alongside existing for a trial period
5. **Training Plan**: Develop documentation for content maintainers
6. **Launch Timeline**: Create a comfortable transition schedule

---

Questions & Discussion

We welcome all questions, concerns, and suggestions. Some questions to consider:

- What features from the current site are absolutely essential?
- What frustrations exist with the current site?
- What features would members love to have?
- What's our comfort level with change?
- How do we want to handle content updates long-term?
- What's our appetite for upfront effort vs. ongoing costs?

Acknowledgments

This proposal builds on the foundation laid by our current website team. Their dedication has kept our online presence active and informative. Any transition would honor their work by preserving what works while exploring new possibilities.

Contact

For questions, feedback, or to discuss this proposal:

Adam Shaleen adamshaleen@gmail.com

Document Version 1.0 - 1.11.26