

CLIENT PROJECT OVERVIEW

Full Stack Web Application Development with AI Integration

React (TypeScript) · C# / .NET Core · Azure AI · OpenAI APIs

Prepared for Client Review · Confidential · May 2026

React + .NET Core

5 Pillars

Azure + OpenAI

CI/CD Ready

CONFIDENTIAL

Executive Summary

This document outlines the project overview for a modern, enterprise-grade Full Stack Web Application engagement. The solution combines a React / TypeScript frontend with a robust C# / .NET Core backend, enriched by cutting-edge AI/ML services — including Azure Cognitive Services and OpenAI APIs — to deliver intelligent, data-driven user experiences. The engagement spans design, development, API architecture, database management, DevOps, and ongoing code quality governance.

Project Scope

Scope Pillar	Description	Key Activities
Full Stack Development	Design, build & maintain modern web applications	React + TypeScript UI · .NET Core APIs · Middleware · State Management
AI Integration	Embed intelligent features across the application	Recommendation Engines · NLP · Predictive Analytics · OpenAI / Azure AI
API Development	RESTful & GraphQL APIs serving the React frontend	.NET API design · DTO patterns · Versioning · Error handling
AI-Assisted Coding	Leverage AI tools to accelerate delivery quality	GitHub Copilot · ChatGPT · Unit test gen · Refactoring · Debugging
Architecture & Design	Translate requirements into scalable technical specs	System design · Scalability planning · Tech selection · Review cadence
Database Management	Design & query enterprise-grade databases	SQL Server / PostgreSQL / Cosmos DB · EF Core · Dapper · Migrations
Code Quality	Champion testing, reviews & security best practices	Jest · xUnit · PR reviews · AI-assisted bug & vuln detection · Docs

Solution Architecture Overview

Layer / Tier	Technology	Notes
Frontend – UI	React 18+, TypeScript, Hooks	Component-driven, type-safe UI
State Management	Redux / Zustand / Recoil	Global & local state strategies
Styling	Tailwind CSS, Styled Components	Utility-first + component styles
Backend – API	C# / .NET 8, ASP.NET Core	RESTful + GraphQL endpoints
Middleware & DI	.NET Middleware, HILT-style DI	Pipeline patterns, IoC container
AI/ML Services	Azure Cognitive Services, OpenAI	NLP, vision, recommendations, chat

Layer / Tier	Technology	Notes
AI Coding Assistants	GitHub Copilot, ChatGPT, Codeium	Productivity, test gen, refactor
ORM / Data Access	Entity Framework Core, Dapper	Code-first migrations, raw SQL perf
Databases	SQL Server, PostgreSQL, Cosmos DB	Relational + NoSQL options
Authentication	Azure AD / OAuth 2.0 / JWT	Secure token-based auth
DevOps / CI-CD	GitHub Actions, Azure DevOps	Automated build, test, deploy
Cloud Platform	Microsoft Azure (preferred)	App Service, Functions, AKS ready

AI Integration Blueprint

AI Feature	Service / Tool	Business Value
Recommendation Engine	Azure Personalizer / OpenAI	Personalised content & product suggestions
Natural Language Processing	Azure Language / OpenAI GPT	Smart search, sentiment, chatbot support
Predictive Analytics	Azure ML / Custom Models	Forecast trends, churn, demand signals
Intelligent Code Assistance	GitHub Copilot / Codeium	Faster delivery, fewer bugs, auto-tests
Computer Vision	Azure Computer Vision	Image analysis, OCR, content moderation
Conversational AI	OpenAI ChatGPT API	In-app assistant, FAQ automation
Anomaly Detection	Azure Anomaly Detector	Real-time alerting on data irregularities

Key Deliverables

Scalable React / TypeScript Frontend	Fully typed, component-based UI with state management, routing, and modern CSS. Optimised for performance and accessibility.
.NET Core REST & GraphQL APIs	Clean, versioned APIs with middleware pipeline, dependency injection, error handling, and OpenAPI documentation.
AI/ML Feature Modules	Production-integrated AI features: recommendation engine, NLP search, predictive dashboard widgets, and conversational assistant.
Shared Component Library	Reusable React component library with Storybook documentation, Tailwind design tokens, and accessibility compliance.
Database Schema & Data Layer	Entity Framework Core migrations, repository pattern, Dapper for performance-critical queries, and seeding scripts.
CI/CD Pipeline	Fully automated build, test, and deploy pipelines via GitHub Actions / Azure DevOps targeting Azure App Service.
Testing Suite	Jest (frontend) + xUnit (backend) test suites with AI-generated test coverage, integration tests, and security scans.
Technical Documentation	Architecture decision records, API docs, onboarding guide, and runbooks — maintained with AI-assisted generation.

Required Skills Matrix

Skill Area	Technologies / Tools	Level
Frontend	React.js 18+, TypeScript, Hooks, Context API	Expert
State Management	Redux, Zustand, Recoil	Expert
Styling	Tailwind CSS, Styled Components	Proficient
Backend	C#, .NET Core / .NET 8+, ASP.NET Core	Expert
API Design	REST, GraphQL, OpenAPI / Swagger	Expert
Async Patterns	async/await, Task Parallel Library, Channels	Strong
AI / ML APIs	OpenAI, Azure Cognitive, Hugging Face	Proven
AI Coding Tools	GitHub Copilot, ChatGPT, Codeium	Active User
Databases	SQL Server, PostgreSQL, Cosmos DB	Proficient
ORM / Data	Entity Framework Core, Dapper	Proficient
DevOps	Git, GitHub Actions, Azure DevOps, Azure	Proficient
Testing	Jest, React Testing Library, xUnit, Playwright	Proficient

Non-Functional Requirements

NFR Category	Target / Requirement
Performance	Page load < 2s (LCP); API P95 response < 300ms; Core Web Vitals green
Scalability	Stateless API design; horizontal scaling on Azure; CDN-delivered static assets
Security	OWASP Top 10 compliance; JWT auth; HTTPS only; input validation + parameterised queries
AI Reliability	Graceful fallback if AI service unavailable; response caching; rate-limit handling
Maintainability	80%+ unit test coverage; TypeScript strict mode; automated code-quality gates
Accessibility	WCAG 2.1 AA compliance; keyboard navigation; screen-reader compatible components
Observability	Azure Monitor / Application Insights; structured logging; error alerting; dashboards
DevOps	Deployment frequency >= daily; MTTR < 1 hr; feature flags for safe rollouts

High-Level Engagement Timeline

Phase	Key Activities	Duration
Phase 1 Discovery	Requirements, existing codebase audit, tech selection	2 weeks
Phase 2 Architecture	System design, API contracts, DB schema, AI service plan	2 weeks
Phase 3 Foundation	Project scaffolding, CI/CD, auth, shared component lib	3 weeks
Phase 4 Core Development	Feature sprints: frontend, APIs, DB, AI integrations	8–12 weeks
Phase 5 AI Feature Sprints	Recommendation engine, NLP, predictive analytics go-live	4 weeks
Phase 6 QA & Hardening	Performance tests, security scans, accessibility audit	2 weeks
Phase 7 Handover	Documentation, knowledge transfer, runbooks	1 week

Value Proposition

AI-Accelerated Delivery GitHub Copilot and ChatGPT integration cuts boilerplate time and surfaces bugs earlier — reducing time-to-market.	Type-Safe, Maintainable Code TypeScript on the frontend and C# on the backend create a fully typed, refactoring-friendly codebase.
Intelligent User Experiences OpenAI and Azure AI integrations turn raw data into personalised, predictive, and conversational features.	Cloud-Native Scalability Azure-first architecture with CI/CD and horizontal scaling handles growing user loads without re-architecture.
Security by Design OWASP compliance, JWT auth, AI-assisted vulnerability scanning, and encrypted data pipelines from day one.	Continuous Quality Automated test suites (Jest + xUnit) and AI-powered code review create a high-confidence release cadence.