

## **SUMMARY:**

Enterprise level Application Development professional with working experience in Financial/Investment Management Software Industry. Proficient in application and database development with solid understanding of software architecture and design principles. Excellent problem-solving and debugging, communication and collaboration as well as customer support skills.

## **TECHNICAL SKILLS:**

- *Programming languages:* C#, SQL, PL/SQL, VB.NET, VB6, VB Script, JavaScript, CSS, HTML
- *Technologies:* .NET Core 9, WPF, gRPC, Web API, WinForms, WebForms, WCF/Windows Services, ADO.NET, LINQ, Blazor, ASP.NET Web Services, ASP, IBM and MS MQ, ASP.NET Core, REST API, React, Infragistics Controls, Crystal Reports, SSRS Reports, SOAP, JSON, XML, MSTest, DataDog
- *Databases:* SQL Server, Oracle, Sybase, InterSystems Cache
- *Development Tools:* MS Visual Studio, SQL Server Management Studio, Rapid SQL, Toad for Oracle, MS Visio, IIS, SMTP, VS Code
- *CI/CD Software:* Azure DevOps, TeamCity, Git, BitBucket, Jira, Confluence, Jenkins, Nexus, UrbanCode Deploy, SonarQube, TFS
- *Methodology/Architecture:* Agile, Scrum, Waterfall, Microservices, Plugins, MVVM, MVC, API, Dependency Injection
- *APIs/Libraries:* Charles River API/Plugins, Fincad API, MS Entity Framework
- *Trading Systems:* Charles River IMS, HelioGraph, Omgeo Oasys/CTM

## **EXPERIENCE:**

### **Senior Software Developer | Dynamo Software, Watertown, MA | 2024 – 2025**

- Communicated with project managers, product owners and development team to facilitate understanding of deliverables, estimates, and prioritization for ASP.NET Core FinTech **Research and Portfolio Management SaaS platform** solving challenges across the alternative investment landscape.
- Delivered **Composite benchmarks** feature to automatically populate synthetic benchmark values based on the weighted combination of constituent index values (C#, SQL Server).
- Developed **Dynamic benchmarks** for LP clients to compare the performance of their portfolio to the return of a benchmark which automatically took into consideration the changing weight of investment asset classes during the holding period. Created **composite investment** for each asset class in the account so users could assign a primary benchmark to it and use its portfolio NAV % to reflect exposure category weight for a specific period.
- Implemented **What-if scenario module** which calculated projected portfolio exposures based on selected cash flow and statistics models. The feature took into account not only existing investment positions but also the impact of hypothetical future transactions (ASP.NET Core, REST API, JavaScript).
- **Created** and executed **unit tests** to ensure the robustness and reliability of the application. Worked closely with QA team to identify issues and devise solutions to them (MSTest).
- Provided **resolution for tenant issues** reported by client support and deployment teams from production and staging application slots.

### **Software Engineering Architect | MFS Investment Management, Boston, MA | 2007 – 2023**

- Collaborated with other developers, business analysts, and stakeholders to gather requirements and ensure successful migration of **Portfolio Modelling Tool** into .Net Core trading application responsible for sending 80% of company stock orders to Charles River Investment Management System running in Azure.
- Took the lead in designing, architecting, and developing authorization, health check, emailing, client logging, performance recording and data retrieval of Grpc/Web **API microservices** for the new *Modelling Tool*.
- Following best practices and coding standards implemented generic service API libraries, logging and rule engine components to be utilized in .Net Core services and new *Modelling client*.
- Contributed to the design and architecture of new *Modelling Tool* screens and workflows, implemented user input validation and business logic functions.

- Worked on integration of model approval process with MS Teams located in Azure so portfolio managers can approve or deny portfolio rebalance scenarios.
- Led the development of the **Mfs Order Management system (MOM)** designed for investment portfolio managers and specialists to check pre-trade compliance, validate brokers for IPO orders and send tickets to the CRD trading system (C#).
- Developed functionality allowing creation and execution of Total Return and Interest Rate Swaps in *MOM* for portfolio managers and traders. Incorporated IPO Deals functionality into *MOM* in order to standardize the Limited Offering order creation process. Enabled options trading in *MOM* to make the orders for this asset class a subject to pre-trade compliance.
- Built robust, performant, and scalable ASP.NET **Web services** for trading client applications to report the status of nightly jobs, execute pre-trade compliance checks, provide user authentication.
- Worked on the company wide initiative of **integrating** in-house order management **client applications into** newly adopted **CRD Trading System**. That included routing equity orders to CRD, performing compliance rule checks in the new trading system and overriding those rules.
- Created and improved **WCF ETL services** used to enable the flow of issuers, orders and unavailable shares into CRD Trading system. Worked on identifying split/merged CRD orders. (IBM MQ, C# Windows service, CRD API, Oracle).
- Designed and implemented *Portfolio Manager Approval* and *Plugin services* along with several **UI add-ins** to customize **CRD Workbench** order modelling. In the process created CRD result sets and workflow rules for trading data retrieval and updates, configured portfolio manager/specialist trading blotter layouts.
- Using clean, efficient, and maintainable coding techniques developed a module which provided notifications and approval process for orders created as a result of the **Asset Mix** re-balance exercise.
- Developed **SSRS** reports with complex compliance and audit related **SQL**. Created and improved financial data retrieval and persistence stored procedures for in-house orders management systems.
- Trying to stay up-to-date with industry trends and technologies designed and put into action **Media Gallery** single page web application for effective browsing and search of photo and video content (React, .NET Core Web API).
- Assisted in troubleshooting/**resolution of production issues** and ad-hoc data requests coming from the equity trading desk. Provided support for regular disaster recovery exercises, nightly and holiday cycles.
- Participated in trading **systems upgrades**. Ensured compatibility of order management systems with the latest operating systems and database drivers. Maintained high level of **security protection** by regular password changes for generic database accounts.
- Applied fundamentals and principles of **agile development** by participating in scrum meetings, helping to slice work into Jira stories, filling the backlog with new issues and updating their status on the board as the features are progressing, preparing completed work presentations at the end of each sprint and participating in lessons learned discussions.
- Created setup projects in Visual Studio, built installation files using Jenkins, helped release engineering **deploy application releases** into test and production environments using UrbanCode Deploy.
- Focused on **product quality and risk mitigation** by addressing issues discovered by SonarQube and IBM AppScan Jenkins plugins.

#### **Internet Developer | PG Calc Incorporated, Cambridge, MA | 2000 – 2007**

- Ported ASP commercial charitable deduction calculator (**GiftCalcs**) to ASP.NET object-oriented application. Completely redesigned UI and replaced the database provider (ASP.NET, C#, JavaScript, HTML, SQL Server).
- Integrated gift presentation slideshow (**GiftStory**) into GiftCalcs (ASP.NET, Flash). Extended *GiftCalcs* admin features by adding templated **email broadcaster** (C#, Windows service).
- Implemented Windows service configurable from the OS system tray for handling bulk email files (VB.NET).
- Created *GiftCalcs* **database diagrams**, documented solution and SQL objects. Developed *GiftCalcs* InstallShield project and data migration application for rapid deployment to production server.
- Designed and implemented charitable deductions **Batch Calculator** capable of importing a file and producing a spreadsheet of computed results.
- Worked in the team of several programmers to migrate a gift planning administration product (**GiftWrap**) from the procedural into object-oriented application (VB6, SQL Server).
- Developed a component to merge *GiftWrap* database records with the templated PDF tax forms (FDF Toolkit, Adobe Acrobat).
- Created **Database Manager tool** for the customers using MSDE database engine to create users, backup and restore databases (VB6, SQLDMO, OSQL).
- Maintained *GiftWrap InstallShield project* that included several third party modules.

- Programmed desktop and web **tools for batch generation of GiftWrap authorization codes**.
- **Handled GiftWrap and MSDE installation support issues**, resolved application conflict situations and SQL Server connectivity problems, troubleshooted product setup and performance in various system/network configurations.
- Developed commercial **application to merge GiftWrap databases** (C#, SQL Server, HTML).
- Implemented **company web site** resources section, schedule of conferences, planned giving training service calendar and session registration section. Maintained employee *sign-in scheduler* (ASP).
- Extended company *site* shopping capabilities to promote new products and introduce discounted prices for the existing ones.

#### **EDUCATION:**

**B.A. in English Language and Literature** | Zaporizhzhia State University, Ukraine

**Contemporary Applications Development Certification** | Computer Learning Center, MA

#### **CERTIFICATIONS:**

##### ***Oracle University***

- Analytic SQL for Data Warehousing

##### ***Pluralsight***

- Build Rich Web Applications with C# using Blazor
- Building a RESTful API with ASP.NET Core
- Building an Async API with ASP.NET Core
- Dependency Injection in ASP.NET Core
- Front-End Web Development With HTML5, CSS, and JavaScript
- Implementing Advanced RESTful Concerns with ASP.NET Core
- JavaScript Objects and Prototypes
- React - The Big Picture
- Using gRPC in ASP.NET Core
- WPF MVVM in Depth