

Vishvam Amin

+1 514 449 5789

<https://github.com/aminvishvam>

vishvamamin55@gmail.com

<https://www.linkedin.com/in/vishvam-amin/>

Toronto, ON, Canada

Professional Summary

- Software Engineer with 6+ years of experience specializing in full-stack development (MERN/MEAN), Microsoft 365 solutions, and cloud-native applications. Proven ability to design, develop, and deploy scalable web applications, with recent experience in integrating AI/LLM technologies to enhance productivity within legal workflows.
- Strong full-stack expertise in MERN/MEAN stacks, with solid hands-on experience in ReactJS (for building user interfaces in Lexis Create and the Microsoft Teams add-in), Next.js, Node.js, FastAPI, ASP.net core web applications.
- With the open-source solution like webpack making sure to develop seamless applications with more than 18 Micro-frontend applications
- Managing micro frontend deployment using the jenkins and deploying to Amazon S3 bucket for frontend deployment.
- Hands on experience with multiple microservices and managing multiple middleware to manage seamless authentication between the MFE's and backend services
- Proficient in building user-centric, responsive front ends using React, TypeScript, Fluent UI, Tailwind CSS, and Office.js
- Deep experience working with Microsoft 365 ecosystem including MS Teams, MS Outlook (classic and new), MS Word, MS Graph Explorer, and MS CoPilot
- Skilled in developing and integrating Microsoft Teams Chat Bots, Microsoft 365 add-ins, and custom solutions using Microsoft Graph APIs
- Hands-on knowledge of Azure Active Directory, managing Application & Delegated permissions, WAM OIDC flows, and secure app authentication
- Well understanding of the Microsoft Sharepoint permissions as the majority of the MS applications were using the Sharepoint for storing law related documents.
- Experience with Supabase (database + auth), Redis for caching, and API security via JWT and OAuth2
- Adept at cloud-native development and deployment using AWS (EC2, Lambda, S3, DynamoDB) and Azure (App Services, Functions, SQL, AI Bot Service), with the use of this making sure to have scalability, high availability, and cost-efficiency. Proficient in configuring cloud services, managing environments, and optimizing application deployments for enhanced performance.
- Strong CI/CD and DevOps knowledge using Docker, Podman, Jenkins, GitHub Actions, and Azure DevOps
- Experienced in implementing microservices, containerization, and infrastructure automation with Docker, Kubernetes, and Terraform
- Familiar with SOC 2 compliance documentation and maintaining secure, audit-ready systems
- Strong grasp of Agile methodologies, Jira, Confluence, and cross-functional team collaboration
- Passionate about modern software architecture, scalable design patterns, and delivering high-quality, maintainable code
- In addition, as a side project in the company I made an add-in for Azure devops which can correct/write user story wordings. Also, that add-in can review project guidelines and make comments on the current Pull Request automatically using the knowledge from LLM models.

Technical Skills

- **Frontend Development:** ReactJS, Next.js, TypeScript, HTML5, CSS3, Fluent UI, Bootstrap, Webpack, Redux, Storybook, Office.js, Micro-frontend development, Webpack.
- **Backend Development & APIs:** Node.js, Express.js, Python (FastAPI), REST APIs, GraphQL, JWT, WAM OIDC, Docker, Podman, LangChain, ASPOSE, LangGraph.
- **Testing & QA:** Jest, React Testing Library
- **Database Management:** MongoDB, MySQL, PostgreSQL.
- **AWS:** AWS: EC2 (for deploying backend services), S3 (for storing static assets and MFE deployments), Amplify, CloudFront, Lambda, API Gateway, IAM
- **Azure:** App Services, App Registrations, Virtual Machines, Azure Functions, Azure SQL Database, Azure AI Bot Service, Azure Active Directory (AAD), Application & Delegated Permissions management
- **Authentication & Security:** JWT, OAuth2, Role-Based Access Control (RBAC), Supabase Auth, WAM OIDC integration, MS Graph Explorer, Azure AD application authentication flows
- **DevOps & Infrastructure:** Git, Docker, Podman, Jenkins, Azure DevOps, Terraform, CI/CD pipelines, Redis for caching and session storage
- **Documentation & Compliance:** Familiar with SOC 2 documentation requirements and practices
- **Tools & Collaboration:** Jira, Confluence, Ollama (local LLM workflows), MS Graph Explorer, Postman
- **Accessibility Tools:** NVDA, AsLint, AXE developer

Education

- **AEC-Information Technology Programmer Analyst**

LaSalle College, Montreal

Subjects: Object Oriented Programming, Scripting Languages, Application & Web Development.

2020 - 2021

- **Bachelor of Engineering, Information Technology**

Government Engineering College, Modasa, Gujarat, India

Subjects: Data Structure and Algorithms, Artificial Intelligence, Database System, Computer Science.

2015 - 2019

Experience

Software Engineer II, LexisNexis Inc, Toronto, Canada.

Jan 2022 - Present

Company website: [LexisNexis](#)

Lexis Create: Developing and maintaining **multiple micro-frontend (MFE)** applications using **React, TypeScript, Redux, and ASP.NET Core Web API**, working on both frontend and backend development. Collaborated closely with backend developers and cross-functional teams to ensure seamless integration, optimal performance, and alignment with project objectives.

- Working on an Artificial Intelligence (AI) project for the company *Lexis Plus AI (Protege)*. The project involves leveraging the main product functionality within Word and Outlook platforms to interact with Word documents and Outlook emails (both read and compose functionalities).
- Building an **MCP server** in **node js** to read storybook stories and ramping up the development for the Microfrontend development to support the faster update in the code with Github Copilot.
- Collaborated with a team of data scientists to integrate a RAG (Retrieval-Augmented Generation) pipeline using the LangChain library in Python.
- Designed and maintained the Python codebase using the **FastAPI** framework.
- Developed a robust ecosystem to support multi-document upload functionality with session-based tracking and integrated vector database retrieval to efficiently query and extract responses from LLM models.
- A key feature involves extracting Outlook email attachments and processing them through a RAG (Retrieval-Augmented Generation) pipeline integrated with the company's proprietary LLM models. Based on the contents of the email attachments, users can ask questions to generate summaries and identify legal obligations present in the attached documents.
- Detailed Proof of Concepts (POCs) were constructed prior to feature development, with all necessary permissions obtained from Azure Active Directory.
- Implemented **Husky** in frontend repositories to enforce code quality standards and streamline development workflows.
- Serving as the lead frontend architect for AI-related Micro-Frontend (MFE) development, focusing on code reusability across three platforms: Outlook, Word, and MS Teams. Developed shared library components for 3 MFEs, reducing code duplication by 30% and development time by 20%.
- Styled applications using Fluent UI to ensure visual consistency and enhance user experience, working closely with designers.
- Led a project to update Fluent UI to align with Microsoft Store standards, resulting in successful deployment to the Microsoft Store and improved UI consistency across 18 micro-frontends.
- Developed and maintained **Storybook** component libraries to support over 18 micro-frontends, enhancing code reusability and developer efficiency.
- Managed MFE deployments on AWS S3 and utilized ECS for backend hosting, ensuring robust version control and deployment processes.
- Implemented blue-green deployment strategies for seamless, zero-downtime releases.
- Deployed serverless functions using AWS Lambda with API Gateway, improving the scalability and integration of microservices.
- Optimized CI/CD pipelines to enable lean deployments during critical periods, with fast rollback mechanisms and minimal downtime.
- Automated canary deployments to test new features with a subset of users before full rollout, reducing the risk of production failures.
- Designed failover strategies to ensure high availability during unexpected infrastructure failures.
- Provided platform-side support for Lexis Plus AI, extending Generative AI capabilities to enhance overall functionality.
- Mentored intern developers, conducted code reviews, and led knowledge-sharing sessions to uphold high development standards.
- Played a key role in architecting and delivering large-scale enterprise web applications with a focus on long-term maintainability and performance.

Lexis Connect: Team Lead for the Connect Team, where we Engineered a Microsoft Teams add-in using React, TypeScript, JavaScript, and ASP.NET for the backend, ensuring seamless integration and functionality. Currently continuing development to enhance the application based on user feedback and evolving business requirements.

- Developed a legal matter management application within Microsoft Teams aimed at reducing the volume of daily emails exchanged between lawyers, thereby lowering operational costs.

- Built a product that supports multiple chat-related features, allowing users to create chats on assigned matters directly from the application, on the fly.
- Worked with Microsoft Teams and SharePoint permissions to securely manage confidential data.
- Managed role-based permissions via Azure Active Directory to streamline access control, particularly for the sales team.
- Maintained continuous collaboration with sales and business teams to better understand product requirements and make the application more robust, solving multiple challenges simultaneously.
- Designed and implemented feature flags using Azure App Services and Terraform to enable dark launches and controlled rollouts, accelerating production releases by 15% while minimizing risk.
- Led the development of a critical chat backup feature to address a specific client issue where chat data was auto-deleted due to retention policies.
- Adapted the MS Teams application for mobile, enabling its launch in the Microsoft Store. This involved a close understanding of Microsoft Store requirements and consistent coordination with the MS Teams development team.
- Managed multiple environments for pre-release beta testing and effective user management.
- Implemented support for Micro-Frontend (MFE) architecture, overcoming challenges due to the product's tight coupling with the Microsoft Teams Toolkit.
- After decoupling and enabling MFE hosting, integrated Lexis AI MFE onto the Connect platform, allowing users to leverage AI capabilities within the application.
- Refactored the project structure to support new features and ongoing maintenance while incorporating the latest updates from the Microsoft Teams Toolkit.
- Collaborated with ASPOS to generate analytics reports and PDF backups of chat histories for customer access.
- Integrated middleware authentication mechanisms using Web Account Manager (WAM) to improve login security and session management.
- Implemented pagination, reducing page load times by 50% for users accessing high-volume data.
- Introduced Redis caching to accelerate backend user search and enhance performance with Microsoft Graph Services. This lead to the development of the user search 70% faster than before.
- Created an **Admin Settings** tab in the Teams application to manage administrative controls (e.g., Teams notifications, email notifications, user roles, and automatic matter assignments).
- Designed and enforced Role-Based Access Control (RBAC), defining secure, role-specific access to features.
- Implemented a feature flag system using Azure App Services and Terraform to enable dark launches and controlled rollouts.
- Researched, evaluated, and synthesized technical requirements to design, develop, and test Microsoft Teams applications, including mobile versions.
- Assessed, troubleshooted, upgraded, and maintained MS Teams applications to ensure optimal performance.
- Created and maintained comprehensive documentation to simplify onboarding and facilitate knowledge transfer for new team members.
- Reviewed Microsoft support documents to follow best coding practices and uphold high standards for LexisNexis products.
- Prepared SOC 2 compliance documentation for launching the application on the Microsoft Teams Store.
- Created installation documentation to be shared with prospective customers.
- Conducted code reviews to ensure quality, performance, and maintainability across the development team.
- Styled the application using fluent-ai and fluentui/react-northstar. Later, migrated the entire UI to Fluent UI v9, which enabled the implementation of theme-based designs.
- Built dynamic form functionality using React libraries such as JSON Forms to handle form-based workflows efficiently.
- For state management we were using redux, redux-saga and for unit and integration testing we were using the react-testing library.

Front-end Developer, Techrangle Inc, Toronto Canada

May 2021 - Jan 2022

SCENE plus Project: bank project in which we have the card points rewards platform.

- Recently, working on a major bank project with the team to develop a web application in the next JS. The design of the application is in tailwindCSS and styled components.
- Working with the CMS on which we were managing UI coming for the multiple language support for the frontend development.
- State management was handled with the react redux.
- Making a whole application AA-level accessible for disabled people to use a website. For that, we used tools like NVDA, AsLint, AXE developer tools, etc.
- Writing integration tests for the application in JEST and React testing library.
- Writing code to collect data for google analytics in application with the help of react-gtm-module. And, debugging code in Google tag manager.

Review my agent

- Designed a Full Stack web application in angular framework with an agile team. And made a use of REST API to work with the Data. Made the reusable components, which results in less code.

- With the use of angular material UI and SASS designed the website and made it scalable.

Techrangle Website

- Working on the website which is built with Gatsby frame-work. Also for the backend worked with the strapi and graphql.

Full-stack Developer, Innoventaa Technology, India

Jan 2018 - Dec 2019

- Completed training as a Full-stack Developer with additional focus on database management, collaborating with teams to identify the best software development processes and align with customer needs.
- Developed dynamic and responsive web applications by combining React JS for the front end and Fast API for the back end, leveraging RESTful APIs to ensure smooth communication between components.
- Actively participated in brainstorming sessions with development teams and product managers to develop innovative software solutions.
- Created Fast API views and templates using Jinja templating engine, ensuring user-friendly and accessible web interfaces for a seamless interaction experience.
- Gained hands-on experience in Amazon EC2, setting up instances, configuring Virtual Private Clouds (VPCs), and managing security groups to ensure secure deployment of applications.
- Applied MVC architecture using Fast API, ensuring a clean separation of concerns between logic, data, and presentation layers for maintainable code structure.
- Build Python scripts for database content updates and file manipulation, ensuring efficient data processing and seamless integration with the front-end.
- Designed and implemented complex SQL Views, Stored Procedures, Functions, and Triggers to optimize data manipulation and maintain data consistency across platforms.
- Reviewed and optimized Python code, running troubleshooting test cases to identify and resolve issues, ensuring a smooth user experience.
- Utilized Python packages like cx_Oracle, pyodbc, and MySQLdb to integrate with Oracle, SQL Server, and MySQL databases.
- Followed Test-Driven Development (TDD) practices, writing unit tests using Fast API testing tools to ensure application reliability and bug-free functionality.
- Designed and automated CI/CD pipelines using GitHub Actions, optimizing deployment workflows for serverless applications on AWS, ensuring continuous integration, high availability, and cost-effective scaling.
- Managed SQL and NoSQL databases such as MongoDB, ensuring data integrity, efficient access, and high availability.

Technical Achievement

Hackathon: 1st Place - Awarded 1st place at a LexisNexis hackathon for developing a Generative AI-driven Azure DevOps extension to streamline user story enhancement and PR commenting. Key technologies include:

- RAG pipeline implementation
- FastAPI (Python)
- LangChain
- Local OLLAMA models (Podman containers)
- LLM-powered analysis of project guidelines (PDF) for automated PR feedback

The solution was completed in 48 hours.

MCP Server for Storybook components: Building an MCP server with node js for the internal LexisNexis storybook component to be able to update the multiple frontend project in high pace with the help of github copilot.

- Implemented proper scan that can read all stories from storybook and index them to use it further for the replacement.
- Implemented component details to get all component information such as props and styling to use that further to compare current codebase with most comparison percentage.
- Implemented scan that can scan open components on the repo with the storybook component and the knowledge gathered before.

Design Agent: Automated Frontend Component Builder: We are creating an AI-powered agent that can handle the entire frontend development workflow and generate production-ready components inside a Storybook component library.

- Core technologies: **LangGraph Studio with python.**
- Workflow:** Requirement Understanding The agent converts a natural language user prompt into a structured component requirement. Knowledge Retrieval It searches the Fluent UI documentation to identify the most relevant component patterns, APIs, and styling guidelines. It inspects the existing component library to align with established coding and architectural standards. Code & Story Generation Using LLMs, the agent generates React components following Fluent UI conventions. It produces corresponding Storybook stories for documentation, testing, and reuse.