

Professional Summary:

Full-stack software engineer passionate about building scalable systems, delivering reliable applications, and constantly improving through hands-on development. I am eager to take on challenging projects and expand my skills across modern technologies.

Projects

Frontend Payment Method Landing Page

HTML5 | Tailwind CSS | Javascript | [React.js](#)

Developed a sleek and functional landing page that effectively communicated the value proposition of the payment method while maintaining a high standard of design and usability.

Full Stack Learning System

Node.js | Express.js | MySQL2(Sequelizer) | Swagger | HTML5 | Tailwind CSS | Javascript | React.js

Developed a secure backend featuring email verification via SMTP, JWT-based role authentication, and cookie management, all implemented using a clean code structure and Triple-D architecture with strict separation of concerns. Built a responsive and modern frontend interface to ensure intuitive navigation and an improved user experience.

Full Stack Library Renting Books Website

HTML5 | Tailwind CSS | Javascript | [React.js](#) | C# | [Asp.net](#) | MySQL

Delivered a fully functional platform that streamlines book rentals using an MVC architecture, implementing both user and admin roles for secure and structured access control. Designed a clean and responsive interface to ensure a seamless and intuitive user experience.

Chat Application:

C# | ASP.NET Core 8 | GraphQL | SignalR (WebSockets) | PostgreSQL | Entity Framework Core | FluentValidation | AutoMapper | BCrypt | JWT | Google OAuth 2.0 | SendGrid API | HTML5 | TailwindCSS | Typescript | [React.js](#) (Working on it)

Developing a real-time chat application (Discord clone) with a secure, scalable backend built on ASP.NET Core 8 and GraphQL. The application features dual authentication methods (email+password with BCrypt hashing and Google OAuth 2.0), JWT-based token management with secure refresh token rotation stored in httpOnly cookies, and email verification via SendGrid API. The backend implements clean architecture principles with strict separation of concerns across GraphQL, service, and data access layers, utilizing Entity Framework Core for PostgreSQL database operations. Input validation is handled through FluentValidation, with automatic DTO mapping via AutoMapper ensuring seamless data transformation between layers. The frontend, built with React.js, TypeScript, and TailwindCSS, will provide a responsive, intuitive interface for real-time messaging with WebSocket support via SignalR, enabling live chat functionality and user presence features.

Technologies and Languages

Languages: C++, C#, Java, HTML5, CSS, Bootstrap, Tailwind CSS, JavaScript, React.js, Node.js, TypeScript, Flutter

Frameworks & Tools: Spring Boot, ASP.NET Core, Git, Swagger, Postman, [Express.js](#), Framer Motion

Databases: MySQL, MongoDB, Sequelizer(MySQL2), PostgreSQL

Other: Data Structures & Algorithms, Multithreading, OpenMPI, Mutex, Exception Handling, Fast Learner, Adaptable to New Technologies

Education

Antonine University

B.S. in Computer and Communications Engineering – Software Track - Undergraduate 3rd year