Gowtham K
Full-Stack .NET Developer

+91-9344447120

GitHub   : github.com/ Gowthamkumar91

LinkedIn : linkedin.com/in/gowtham-kumar-a20614385

Email   : gowthamcoder91@gmail.com

Professional Summary

Full-stack .net developer with 1.6 years of hands-on experience in designing and developing scalable web applications using ASP.NET Core MVC, Web API, and React. Proficient in ADO.NET, Entity Framework Core, Razor, Bootstrap, SQL Server, and React with Redux Adept in implementing layered architecture, role-based security, payment integrations, and deploying staging builds. Seeking a challenging role to contribute technical expertise and deliver enterprise-grade solutions.

Technical Skills

Frontend	:	HTML5, CSS3, JavaScript, Bootstrap, Razor, React.js, Tailwind CSS, MUI
Backend	:	ASP.NET Core MVC, Web API, ADO.NET, Entity Framework Core (Code First)
Database	:	SQL Server, Stored Procedures, LINQ
Tools	:	Visual Studio 2022, VS Code, Postman, Git (Azure Repos),GitHub
Auth & Security	:	JWT, ASP.NET Identity, [Authorize], Refresh Token, Anti-Forgery Token
Architecture & Patterns	:	Repository & Service Layer, DTOs, Seed Data
Email Services	:	SMTP (for transactional emails)

Work Experience

Full Stack .NET Developer *(Trainee - Contract Role)*
Cloud Service For All, Chennai *April 2024-Present*

Project handled under this role:

Project 1	:	**SKP Handloom Sarees (E-Commerce ERP)**
Domain	:	E-Commerce
Tech Stack	:	React, Tailwind CSS, MUI, Redux Toolkit, ASP.NET Core Web API, Entity Framework Core (Code First), ADO.NET (Stored Procedures), Fluent Validation, JWT + Refresh Token, DTOs, SMTP

Description:

Developed a scalable e-commerce ERP platform to manage saree listings, customer orders, payments, and refunds. Implemented role-based access (Admin, ProductManager, SalesManager) with secure JWT authentication and centralized cart/order handling via Redux. Used DTOs and FluentValidation to streamline data flow and ensure input security. Backend logic and reporting were powered by ADO.NET with optimized stored procedures.

Roles & Responsibility:

- Developed modular and responsive UI using React, Tailwind CSS, MUI, and reusable functional components
- Applied React Hooks for UI lifecycle, interactivity, and carousel logic
- Built category-based filtering, banner sliders, and wishlist/cart modules

- Integrated Redux Toolkit for centralized state management (auth, cart, orders, admin modules)
- Created place order, order summary, and dashboards with discount/refund logic
- Secured Web API using JWT authentication and implemented Refresh Token for session management
- Enforced FluentValidation for secure admin/user input validation
- Used DTOs to decouple models from client contracts in the API layer
- Developed APIs using ASP.NET Core Web API with EF Core (Code First)
- Used ADO.NET with SPs for refund logic, cart handling, and sales reports
- Utilized ADO.NET with optimized stored procedures (JOINs, GROUP BY, CASE) to handle refunds, cart ops, and sales reports Configured SMTP to send transactional emails (orders/refunds)
- Seeded Admin, ProductManager, SalesManager roles using SeedData
- Applied [Authorize(Roles=...)] to restrict endpoint access based on roles
- Followed Repository & Service Layer architecture for business logic separation
- Participated in staging-level testing and supported internal deployment
- Application is actively used internally for product/order/payment management
- Delivered full-stack modules (UI, API, DB) as part of a 2-member team under a contract trainee role

Project 2   :   **Rock Music Band**

Domain   :   Entertainment Tech (Concert Booking ERP System)

Tech Stack   :   ASP.NET Core MVC, Razor, HTML5, CSS3, Bootstrap, SQL Server, Entity Framework Core (Code First), ADO.NET (for View Model-specific SP), Repository & Service Layer, ASP.NET Identity, SMTP, View Models, Enums

Description:
Developed a web-based ERP system using ASP.NET Core MVC to manage concerts, performances, ticket bookings, and refunds. Enabled secure role-based modules (Admin/User) with UPI/COD payment simulation. Designed Razor-based UI with integrated ViewModels for validation. Configured SMTP for automated emails, on bookings, cancellations, and refund status.

Roles & Responsibility:
- Designed responsive UI using Razor, Bootstrap, and layout views for modules like concert booking, ticket sales, admin/user dashboards
- Developed Razor views for Concerts, Performances, Payments, and Reports using ViewModels for UI and DB seperation
- Applied role-based access using [Authorize(Roles=...)] for admin/user previleges
- Configured ASP.NET Identity for signup, login, and role workflows
- Managed DB using EF Core Code-First and Migrations
- Created Repository & Service layers with DI configuration in Program.cs
- Implemented refund logic using stored procedures and SQL filters
- Streamlined UI logic using enums (PaymentMethod, RefundStatus, PriceType) and configured SMTP for booking/refund email alerts
- Seeded default admin credentials and roles via SeedData .cs
- Participated in staging testing and contributed to client-side UAT
- Application is used internally for managing concert events and refund tracking
- Collaborated in a 2-member team to deliver full-stack features across UI, APIs, and DB layers

- Delivered modules independently under a trainee contract, owning end-to-end frontend and backend tasks

Project 3 : Dairy Queen Farm
Domain : Agri Tech (Dairy Tech ERP System)
Tech Stack : ASP.NET MVC, Razor, HTML5, CSS3, JavaScript, Bootstrap, SQL Server, ADO.NET, Stored Procedures

Description:
Built a web-based ERP system using ASP.NET MVC to manage dairy operations Including livestock, milk tracking, and customer orders Implemented role-based access, responsive Razor UI with Bootstrap, and layered MVC architecture Performed data access and reporting via ADO.NET and stored procedures on SQL Server.

Roles & Responsibility:
- Built Razor UI screens for admin/user modules (livestock, milk entry, reports)
- Used ADO.NET with SPs for CRUD operations and reporting
- Designed SQL schema and wrote optimized SPs with joins and filters
- Handled form validations using DataAnnotations and ModelState
- Simulated UPI/COD payment and stored transaction status
- Aligned payment flow with DB structure and admin dashboards
- Configured SMTP for sending booking/payment emails
- Participated in client-side UAT and staging-level validation
- Final validated build deployed internally and used by staff
- Worked in a 2-member team delivering all modules end-to-end
- Contributed as a trainee developer in a 2-member team, building complete UI, API, and database flow

Additional Project:

Completed 4+ self-developed projects using ASP.NET MVC and SQL Server to strengthen backend logic, form validation, and ADO.NET integration
Source code available on Github: github.com/Gowthamkumar91

Education:

B.E. in Mechanical Engineering - Shivani College of Engineering and Technology, Tamil Nadu- 2014.

Certification:

React.js MasterClass – NoviTech R&D Pvt. Ltd. (Dec 2024 – Feb 2025)

Declaration:

I hereby declare that the information given above is true to the best of my knowledge.

Place: Chennai
Date :   (Gowtham K)