

Yasin Amirany

Tehran, Iran

yasin@amirany.net

Objective

Aspiring Back-End Developer with hands-on experience in Node.js, Express.js, Nest.js and relational databases. Passionate about Web Technologies. Building scalable web applications and optimizing system performance. Seeking an opportunity to apply my technical skills in real-world problems while continuing to learn and grow.

Skills

Languages: JavaScript (Node.js), Python, TypeScript

Frameworks & Tools:

- **Back-End:** Express.js, NestJS, FastAPI, socket.io (WebSockets)
- **DevOps & Cloud:** Docker, nginx, GitHub Actions (CI/CD), AWS Lambda (familiar)
- **Messaging/Architecture:** RabbitMQ, event-driven systems, microservices
- **Databases:** PostgreSQL, MongoDB, Redis
- **Security & API Design:** RESTful APIs, GraphQL, OAuth/JWT, rate limiting
- **Testing & Documentation:** Unit/Integration Testing, Swagger/OpenAPI, TDD

Soft Skills

- Fast learning
- Adaptability to new technologies
- Team working
- Problem-solving
- Efficient communicating
- Leveraging AI tools

Projects

[Ecommerce-platform](#) - Fully Dockerized E-Commerce Platform with Real-Time Inventory Management. Redis, Postgresql, Express.

Features:

- **User Authentication:**

- Register and login with JWT-based authentication.
 - Role-based access control (admin and user roles).
 - **Product Management:**
 - Admins can add, update, and delete products.
 - Users can browse products with search and filtering.
 - **Shopping Cart:**
 - Add/remove products to/from the cart.
 - Cart data stored in Redis for fast access.
 - **Order Management:**
 - Place orders and deduct inventory in real-time.
 - View order history.
 - **Real-Time Inventory Updates:**
 - Real-time updates using Redis pub/sub.
 - Notify users of low stock or out-of-stock products.
-

IoT-DMS - IoT Data Management System using MongoDB, and nestJS with RabbitMQ Integration.

Features:

- **Real-time Data Processing:** Utilizes **RabbitMQ** to handle and process real-time x-ray signals from IoT devices
- **Scalable Data Storage:** Employs **MongoDB** with **Mongoose** for efficient and scalable data storage
- **RESTful API:** Offers a well-documented RESTful API for easy integration with external systems. **query parameters** for filtering also supported
- **Swagger Documentation:** Includes interactive API documentation for developers
- **IoT Simulator:** Comes with a built-in **IoT Simulator** that generates and sends mock x-ray data for testing and development purposes
- **Docker Support:** The system is fully Dockerized, making it easy to deploy and manage in any environment
- **Environment Configuration:** The **.env** file is used to manage environment-specific configurations, ensuring security and flexibility
- **Testing:** The system includes a comprehensive test suite to validate functionality and reliability

[Real-time Chat Server](#) - A scalable real-time chat application using Socket.IO with clustering support and message persistence.

Features:

- Multi-core scaling with Node.js cluster module
- Real-time messaging with Socket.IO
- Message persistence with SQLite
- Connection state recovery for clients
- Automatic load balancing across worker processes

Education

Chamran Technical University, Rasht - Bachelor of Computer Science

2021 - present

Languages

English - Proficient

Farsi - Native