

|<----->[BEGIN-CRFCF]<----->|

NOTICE: UNAUTHORIZED MODIFICATION OF THIS DOCUMENT IS EXPRESSLY PROHIBITED. EDITING PRIVILEGES ARE EXCLUSIVELY RESERVED FOR THE AUTHOR, DESIGNATED CONTRIBUTORS, AND AUTHORIZED ARTIFICIAL INTELLIGENCE AGENTS. DOCUMENT INTEGRITY IS SECURED BY A CRYPTOGRAPHIC CHECKSUM; AUTOMATED VERIFICATION SCRIPTS ARE EMPLOYED TO DETECT AND PREVENT UNAUTHORIZED ALTERATIONS.

0. METADATA:

- RFC: RESUME-RFC-000000
- GROUP: PROFESSIONAL RESUME
- TITLE: RICHEVE BEBEDOR BACKEND ENGINEER
- ACRONYM: RBBNG
- AUTHOR: RICHEVE BEBEDOR <richeve.bebedor+vs-work@gmail.com>
- CATEGORY: INFORMATIONAL
- STATUS: DRAFT
- VERSION: 000000 (0.0.0)
- CREATED: 2026-01-28
- MODIFIED: 2026-02-04
- REPOSITORY: <https://github.com/vs-scripts/get-richeve-bebedor-resume>
- CHECKSUM: <1878322766FE759D7D9529593662ADAA9403C1CD4125D996A514BB6D8FB7F641>

1. Summary:

This document specifies the professional profile of Richeve S. Bebedor, a senior-level Node.js backend engineer with over a decade of experience designing APIs, distributed systems, and automation platforms. The author specializes in TypeScript-based backend services, background processing systems, cloud-integrated platforms, and reusable infrastructure tooling.

Primary focus areas include scalable backend architecture, system modularity, automation-first engineering, and performance-aware data and media processing pipelines.

2. Permission:

NOTICE: PERMISSION TO USE, MODIFY, COPY, AND/OR DISTRIBUTE THIS DOCUMENT IS GRANTED FOR EVALUATION, INTERVIEWS, AND PROFESSIONAL REVIEW PURPOSES.

3. Contents:

- METADATA: 009
- Summary: 023
- Permission: 035
- Contents: 039
- Career Status: 055
- Engineering Principles: 070
- Technical Scope Definitions: 080
- Core Competencies: 108
- Professional Experience: 170
- Selected System Implementations: 256
- Open Source and Tooling Contributions: 281
- System Architecture Philosophy: 292
- Education: 298

4. Career Status:

This document represents a stable description of production-level engineering capabilities demonstrated across startup, enterprise, and contract environments. The intended audience includes engineering teams, system architects, and technical recruiters evaluating backend and platform engineering roles.

- Focus areas:
 - Distributed systems
 - APIs
 - Automation
 - Cloud platforms
 - Utilities

5. Engineering Principles:

The author follows these engineering principles when designing systems:

- Systems MUST be modular, observable, and testable
- APIs SHOULD be stateless and versioned
- Background processing MUST be fault-tolerant and concurrency-aware
- Infrastructure SHOULD be automated and reproducible
- Integrations MUST include resilience patterns such as retries, logging, and rate handling

6. Technical Scope Definitions:

This section defines key technical terms used throughout this document.

- Backend System:

Server-side services responsible for APIs, business logic, and data processing.

- Distributed Processing:

Workloads executed across multiple workers, threads, or services to improve throughput and reliability.

- Worker Pipeline:

Background job architecture used for CPU or I/O intensive processing such as media transformation.

- Integration Layer:

Abstraction layer responsible for communication with third-party APIs and external services.

- Automation Tooling:

CLI or programmatic utilities used to streamline operational or developer workflows.

7. Core Competencies:

This section describes the technical competencies organized as system specification subsections.

7.1 Core Languages:

- JavaScript Scripting Language (2012-2026 | 14 years)
- TypeScript Scripting Language (2022-2026 | 4 years)
- Node.js Runtime Ecosystem (2012-2026 | 14 years)
- Bash Scripting Language (2012-2026 | 14 years)

7.2 Backend Architecture:

- NestJS
- Express
- Fastify
- ActionHero
- RESTful API design
- Microservices
- Service-oriented architecture

7.3 Data Systems:

- PostgreSQL
- Sequelize
- MongoDB
- Mongoose
- LevelDB ecosystem

7.4 Distributed and Background Processing:

- Node.js Worker Threads and clustering
- Queue-based job processing
- FFmpeg-based media pipelines
- Concurrency and parallel task execution

7.5 Cloud and Infrastructure:

- AWS
 - EC2
 - S3
 - Lambda
 - SNS
- Microsoft Azure
 - App Services
 - Cosmos DB
 - Networking
- Docker
- Nginx
- PM2
- Linux server environments

7.6 Integration and Automation:

- Payment, media, and SaaS API integrations
- Authentication and SSO integrations
- CLI tooling and automation scripts

8. Professional Experience:

This section describes professional work history structured as system implementation records.

8.1 Senior Backend Developer:

Staizen (September, 2024-April, 2025 | 8 months)

- System Type:

Enterprise Planning Platform (Gembaa)
- Responsibilities:
 - Implemented and maintained NestJS microservices for enterprise workflow processing
 - Resolved production issues and delivered new backend features
 - Contributed to API reliability and test coverage using Jest
- Technologies:
 - Node.js
 - TypeScript
 - NestJS
 - PostgreSQL
- Impact:

Improved system reliability and feature delivery velocity for enterprise planning workflows.

8.2 Full Stack Developer:

Narrasoft (March, 2022-July, 2023 | 1 year & 5 months)

- System Type:

Accounting and Billing Platform

- Responsibilities:

- Maintained and enhanced backend services supporting financial workflows
- Worked across API, database, and frontend integration layers

- Technologies:

- Node.js
- TypeScript
- Angular
- Express
- MongoDB
- PostgreSQL

- Impact:

Enabled continuous feature delivery for financial workflow automation.

8.3 Lead Web Developer and System Architect:

BIA Jiros Travel and Tours (August, 2015-October, 2023 | 8 years & 3 months)

- System Type:

Online Transport Booking Platform (Biyaheroes)

- Responsibilities:

- Served as system architect and lead engineer for full platform lifecycle
- Designed backend services, databases, and infrastructure deployment
- Managed production AWS EC2 environments and MongoDB systems

- Technologies:

- Node.js
- Express
- AngularJS
- React
- MongoDB
- AWS EC2

- Impact:

Architected and delivered a nationwide transport booking platform serving thousands of users.

9. Selected System Implementations:

This section highlights engineering showcases structured as system specifications.

9.1 Distributed Media Processing Pipelines:

Designed concurrent media processing services using Node.js worker threads and FFmpeg for automated transcoding and metadata extraction.

9.2 Modular API Integration Frameworks:

Developed reusable service patterns for integrating third-party platforms with standardized logging, retry logic, and error handling.

9.3 Background Worker Orchestration Systems:

Built queue-driven worker systems for handling asynchronous and high-volume processing workloads.

9.4 Transport Booking System Architecture:

Architected scalable backend services and infrastructure for a nationwide transport booking platform.

A. Open Source and Tooling Contributions:

This section describes ecosystem contributions and reusable infrastructure development.

- Author of some GitHub repositories focused on backend utilities and developer tooling
- Publisher of some NPM packages related to Node.js infrastructure, automation, and data processing
- Emphasis on reusable modules, CLI tools, and backend service helpers

B. System Architecture Philosophy:

The author favors modular service-oriented design, observable backend components, automation-first infrastructure, performance-aware data pipelines, and cloud-native scalability.

C. Education:

- Bachelor of Science in Computer Science

AMA University (2007 - 2011 | 4 years)

Thesis involved neural network concepts and handwriting analysis using object-oriented simulation models.

Professional Email: richeve.bebedor963+vs-work@proton.me
Alternative Email: richeve.bebedor+vs-work@gmail.com
Preferred Contact: +63 928 934 0951
Alternative Contact: +63 999 466 4671
Github Account: <https://github.com/volkovasystems>
LinkedIn Profile: <https://www.linkedin.com/in/richevebebedor>

|<----->[ENDED-CRFCF]<----->|