Backend Technical Documentation
DevSecOps PFE Implementation

Omar Trabelsi

January 10, 2026

Contents

1 Introduction

This document details the backend implementation for the DevSecOps PFE. The system is designed as a RESTful API built with Node.js and Express, focused on secure data handling, automated testing, and containerized deployment.

2 System Architecture

The backend follows a **Layered Architecture** pattern to separate concerns and improve maintainability.

- **Routes:** Defines API endpoints and maps them to controllers.
- **Controllers:** Handles the request/response logic and orchestration.
- **Models:** Defines the schema and interacts with the PostgreSQL database.
- **Middleware:** Intercepts requests for authentication (JWT) and security (Helmet).

3 Folder Structure

```
backend/
          src/
                controllers/    # Business logic
                routes/         # API Route definitions
                models/         # Database schemas (Sequelize/PG)
                middleware/     # Auth & Security checks
                app.js          # Entry point
          config/             # DB & Environment config
          package.json        # Dependencies & Scripts
          Dockerfile          # Container configuration
          .env                # Secret environment variables
```

Listing 1: Project Directory Tree

4 Key API Endpoints

The following table summarizes the core authentication and data endpoints:

Method	Endpoint	Description
POST	`/api/auth/register`	User registration with Bcrypt hashing
POST	`/api/auth/login`	Returns JWT Access Token
GET	`/api/users/profile`	Protected user data (JWT required)
GET	`/api/health`	Health check for monitoring

Table 1: API Endpoint Summary

5 DevSecOps Security Measures

As a DevSecOps project, security is integrated into the code level:

- **Data Encryption:** Passwords are never stored in plain text; `bcrypt` is used with a salt factor of 10.
- **JWT Security:** Tokens are signed with a high-entropy secret and include an expiration time.
- **Rate Limiting:** `express-rate-limit` prevents DDoS and brute-force attacks by limiting IPs to 100 requests per 15 minutes.
- **Header Protection:** `Helmet.js` removes the `X-Powered-By` header and sets XSS protection.

6 Deployment & Automation

6.1 Local Development

```
npm install
npm run dev
```

6.2 Dockerization

The application is containerized to ensure environment parity between development and production.

```
# Build the production-ready image
docker build -t devsecops-backend .

# Run with environment variables
docker run -p 5000:5000 --env-file .env devsecops-backend
```

Listing 2: Docker Deployment

7 Conclusion

The backend provides a secure, scalable foundation for the DevSecOps pipeline. By utilizing industry-standard security middlewares and a containerized approach, the system is ready for automated CI/CD integration.