Backend Report for Project-SE

# 1. Introduction

This document provides a technical overview of the backend architecture, API endpoints, and database design for the Project-SE system. The backend is developed using Node.js and Express, with structured controller modules for admin and customer roles.

# 2. Backend Structure

The backend is organized under the 'nodejs' directory. It includes the following main components:  
- connectDB.js: Establishes database connection  
- index.js: Main entry point of the server  
- controller/: Contains business logic split by user roles  
 - admin/: Admin-specific operations (user management, statistics, login, etc.)  
 - customer/: Customer-specific operations  
- package.json: Project dependencies

# 3. API Endpoints Overview

The application exposes several RESTful API endpoints. Each controller handles specific functionality, for example:  
- /admin/login: Handles admin login  
- /admin/user: CRUD operations for users  
- /admin/statistics: Retrieves statistical data  
- /admin/home: Dashboard-related information  
Each route is handled through Express.js routing mechanism.

# 4. Database Design

The database schema is defined in the file 'data\_create.sql'. It contains tables for managing users, products, orders, and related business entities. The design follows normalization principles for relational databases.

# 5. System Architecture Overview

The system architecture consists of three layers:  
1. Controller Layer: Handles HTTP requests and routes them to appropriate services.  
2. Service Layer (if applied): Encapsulates business logic (partially integrated into controllers).  
3. Data Access Layer: Interacts with MySQL database via SQL queries.  
  
Example flow:  
Client Request → Express Router → Controller → SQL Query → Database → Response

# 6. Summary

This backend implementation supports a modular and scalable structure. Future improvements may include adding middleware for validation, authentication (JWT), and separating service logic into distinct files for better maintainability.