

Last indexed: 11 August 2025 (172577)

Overview

Repository Structure

Getting Started

Project 1: Product Management (Proy1)

Database & Configuration

Product Management System

Development Environment

Project 2: Course Platform (Proy2_Cursos)

Data Models & Database

Authentication & Middleware

Development & Testing

Project 3: Food Delivery Platform (Proy3_Pedidos)

Architecture & Dependencies

Database Models & Schema

Environment & Configuration

Development Environment

VS Code & Debugging

Dependency Management

Database & Configuration

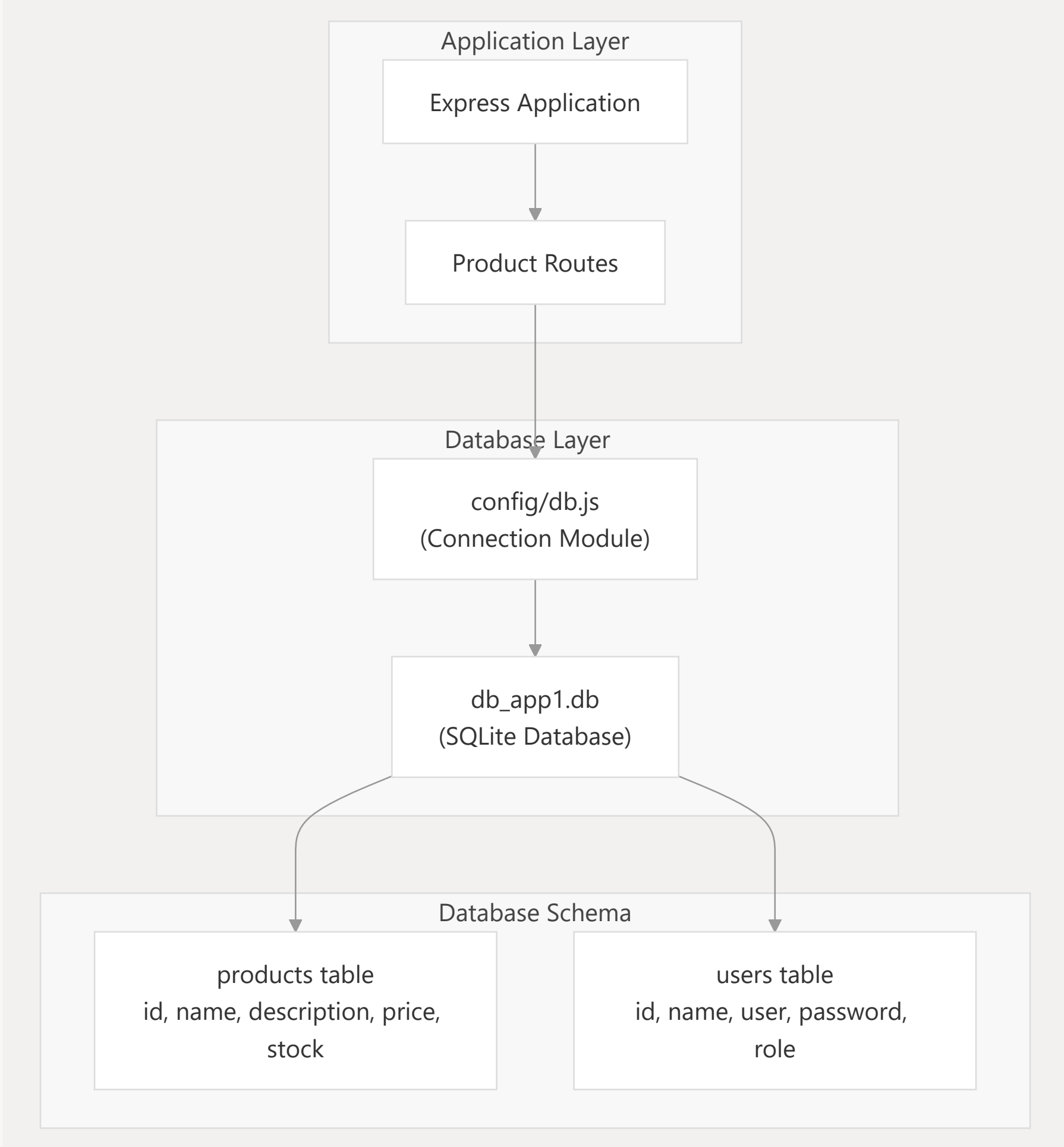
> Relevant source files

This document covers the database setup, configuration files, and environment management for Project 1 (Proy1) - the product management system. This includes the SQLite database schema, Express.js application configuration, and environment variable management.

For information about the product management business logic and routing, see [2.2](#). For development environment setup, see [2.3](#).

Database Architecture

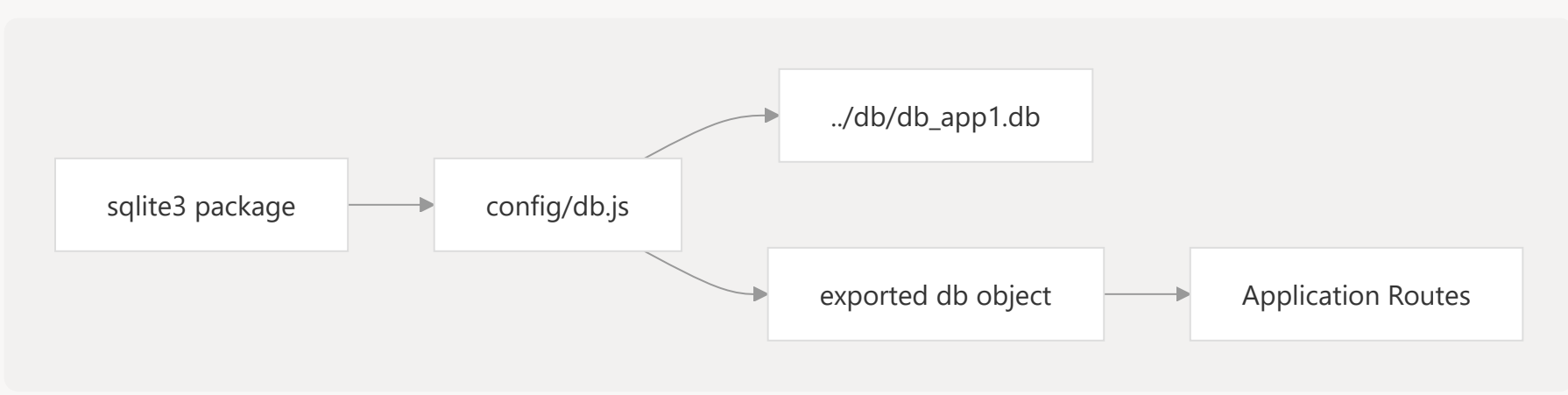
Proy1 uses SQLite as its database engine, configured through a dedicated connection module. The database stores product inventory and user authentication data in a simple, file-based structure.



Sources: [Proy1/config/db.js](#) 1-7 [Proy1/db/db_app1.db](#) 11-25

Database Configuration Module

The database connection is managed through a dedicated configuration module that establishes the SQLite connection and exports it for use throughout the application.



The `db.js` module creates a new SQLite database instance pointing to the database file and exports the connection object for use by other modules.

Sources: [Proy1/config/db.js](#) 1-7

Database Schema

The SQLite database contains two primary tables supporting the product management and user authentication functionality.

Table	Primary Key	Fields	Purpose
products	id (AUTOINCREMENT)	description, name, price, stock	Product inventory management
users	id (AUTOINCREMENT)	name, user, password, role	User authentication and authorization

Products Table Structure

products			
INTEGER	id	PK	AUTOINCREMENT
TEXT	description		NOT NULL UNIQUE
TEXT	name		NOT NULL
INTEGER	price		NOT NULL
INTEGER	stock		NOT NULL DEFAULT 0

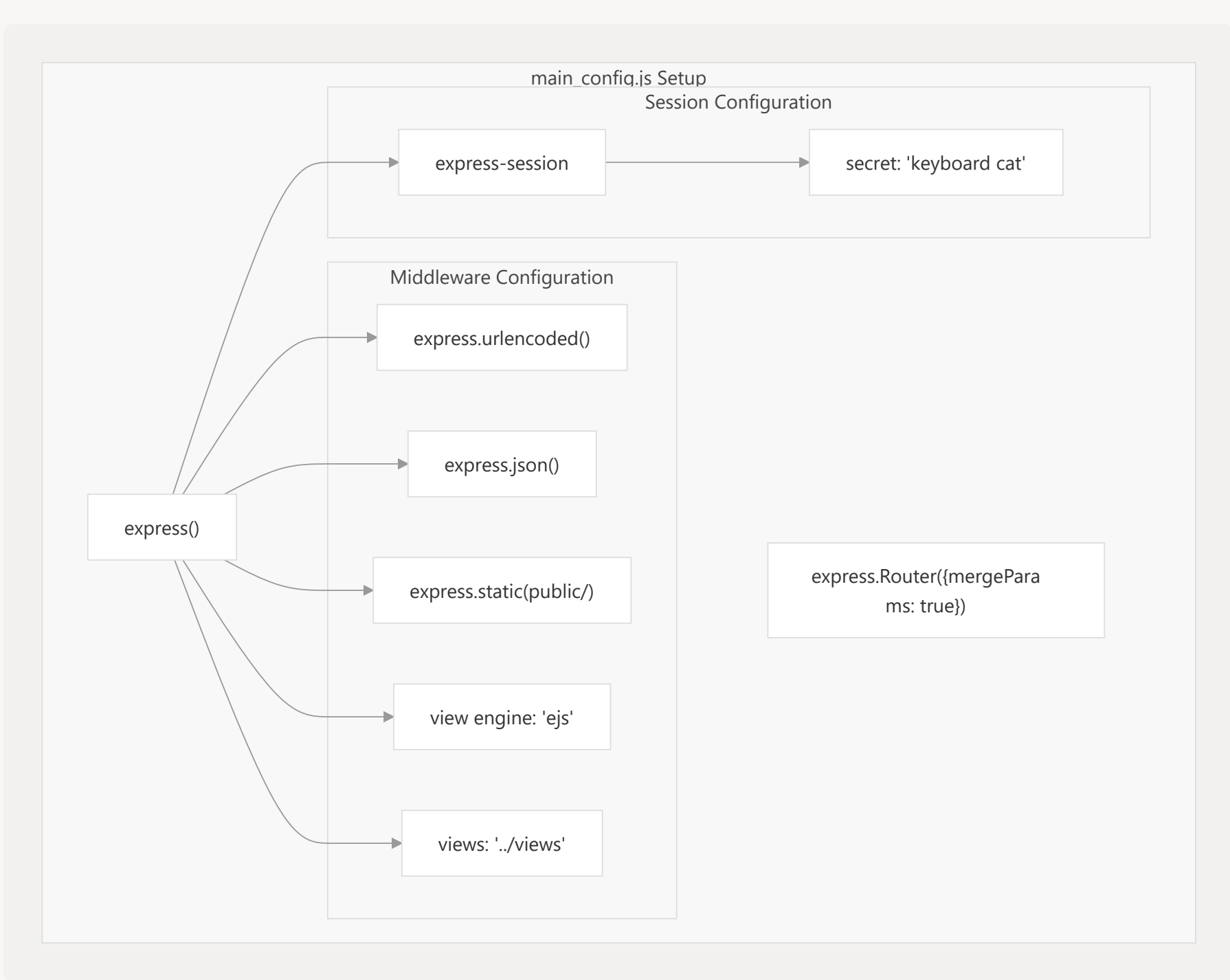
Users Table Structure

users			
INTEGER	id	PK	AUTOINCREMENT
TEXT	name		NOT NULL
TEXT	user		NOT NULL UNIQUE
TEXT	password		NOT NULL
TEXT	role		NOT NULL DEFAULT 'user'

Sources: [Proy1/db/db_app1.db](#) 11-25

Express Application Configuration

The main application configuration is handled by the `main_config.js` module, which sets up Express middleware, view engine, and session management.



Middleware Stack

The application configures the following middleware in sequence:

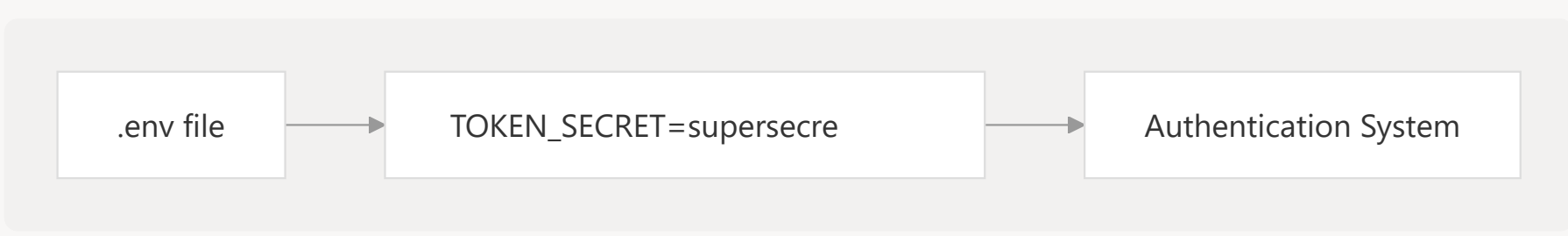
- Body Parsing:** `express.urlencoded()` and `express.json()` for form and JSON data
- View Engine:** EJS templating with views directory at `../views`
- Static Files:** Served from `../public` directory
- Session Management:** Express-session with hardcoded secret

Sources: [Proy1/config/main_config.js](#) 10-24

Environment Configuration

Environment variables are managed through a `.env` file containing application secrets and configuration values.

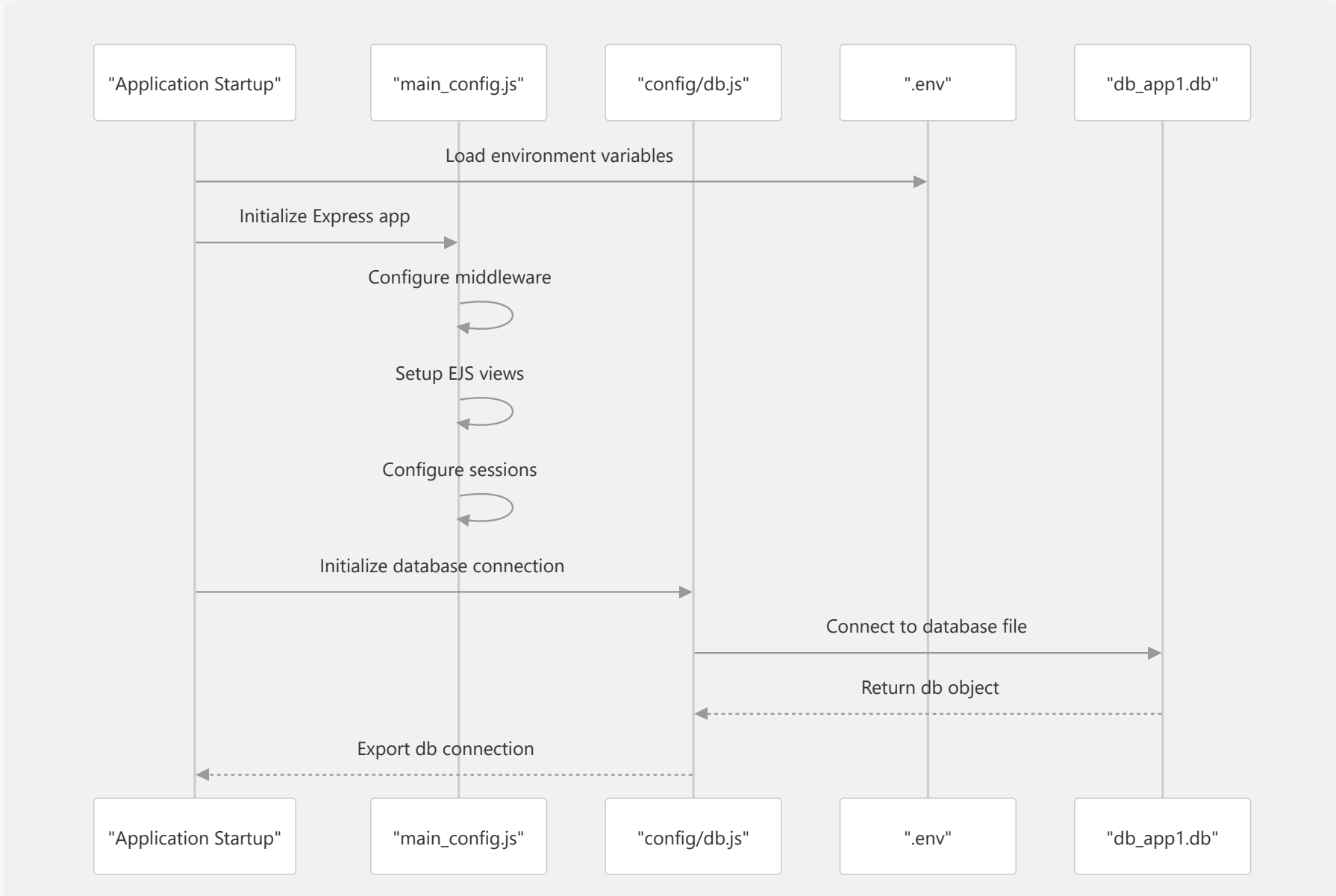
Variable	Value	Purpose
TOKEN_SECRET	supersecrettoken	JWT/Token authentication secret



Sources: [Proy1/.env](#) 1

Configuration Integration Flow

The complete configuration setup follows this initialization pattern:



Sources: [Proy1/config/main_config.js](#) 1-24 [Proy1/config/db.js](#) 1-7 [Proy1/.env](#) 1

Module Exports

The configuration modules export the following objects for application use:

- main_config.js:** `{app, router, session}` - Express application, router instance, and session middleware
- db.js:** Database connection object for SQLite operations

These exports provide the foundation for the product management system's routing and data persistence layers.

Sources: [Proy1/config/main_config.js](#) 24 [Proy1/config/db.js](#) 6