# Backend Documentation

## 1. Project Overview

This project is a student academic management system built using .NET with Onion Architecture principles. It focuses on enabling students to register, manage their academic courses, and calculate their GPA. It emphasizes clean separation of concerns, modularity, and high maintainability. Security, performance, and scalability are core design priorities

## 2. Architecture Style (Onion Architecture)

The solution adopts **Onion Architecture**, which ensures that:

- The **Domain layer** is at the center and contains business logic and entities.
- The **Application layer** defines interfaces and uses cases without depending on infrastructure.
- The **Infrastructure layer** contains implementations (e.g., for data access, authentication, etc.) and depends on interfaces from the Application layer.
- The **Presentation layer** (or API layer) interacts with users and relies on services through dependency injection.

This structure enables high testability, loose coupling, and better separation of concerns

## 3. Solution Structure Explanation

- **Domain**: Contains core business entities and interfaces (e.g., Student, Course).
- **Application**: Defines services, DTOs, use cases, and application-level logic.
- **Infrastructure**: Implements repositories, database access, and external services (e.g., identity, email).
- **DataTransferObjects**: Contains models used to exchange data between layers, keeping domain models isolated.
- **finalProject**: The main web/API project containing controllers, configurations, and dependency injection.
- **Shared**: Includes utilities and common code shared across multiple layers

**4. Technologies Used**

- .NET (C#)
- Entity Framework Core
- JWT Authentication
- Bcrypt for password hashing
- MySQL
- Onion Architecture
- Clean Code and SOLID principles
- Dependency Injection

**5. Authentication**

- JWT (JSON Web Token) is used for secure authentication.
- Passwords are hashed with Bcrypt before being saved to the database.
- Users receive a JWT upon successful login for authorization of future requests.
- Role-based access is supported (e.g., Student).

**6. Security & Performance Enhancements**

- **JWT Protection**: Authenticated access only.
- **CORS**: Configured to allow secure cross-origin requests.
- **Bcrypt**: Encrypts passwords before storing them.
- **Database Indexes and Constraints**: Improve performance and maintain data consistency

**7. Student Role**

Students can:

- Register a new account and verify it using a confirmation code.
- Log in and receive a JWT token.
- Select, update, academic courses.

- View and edit their personal profile.
- Choose their academic department or track.
- Monitor GPA based on registered course grades.

**8. GPA Calculation**

- GPA is dynamically calculated based on registered courses.
- Supports multiple grading formats (e.g., percentage, letter grades).
- GPA updates in real-time as grades are entered.

**9. Notification System**

- The system can notify users via email or in-app alerts for:
    - Account registration confirmation
    - Account Reset Password

**10. Validation**

- Validation occurs both on the client and server sides.
- Server-side checks enforce business logic and rules.
- User-friendly error messages guide users in case of invalid input

**11. Security Measures**

- Tokens have expiration time and are validated.
- Passwords are hashed and never stored as plain text.
- Protection against SQL Injection through parameterized queries.

**12. Extensibility**

- Architecture allows for easy addition of new features or services.
- Interfaces decouple implementations, making it easier to test or replace parts.

- Follows the Open-Closed Principle for extensibility without modifying existing code.
- Suitable for unit testing and integration testing.

---

**13. Conclusion**

This system provides a robust and secure foundation for managing academic data using Onion Architecture. It is scalable, secure, and easy to maintain, making it an excellent fit for academic environments needing custom GPA tracking, user management, and course registration features