Compatibility with EJB + React stack and Docker deployment

1. EJB Server & Version

- Which EJB container are you using?
    - ✅ WildFly (formerly JBoss)? Yes
    - ✅ GlassFish? No
    - ✅ Payara? Yes
    - ✅ TomEE? No
    - ✅ WebLogic? No
-
- Which version? (e.g., WildFly 27, GlassFish 6, etc.) WildFly and Payara
- Are you using EJB 3.x with JPA 2.x or newer? EJB 3.x with JPA 2.x

2. JPA Provider

- Which JPA implementation?
    - ✅ Hibernate (most common with WildFly)? Ys
    - ✅ EclipseLink (default with GlassFish)? No
    - ✅ OpenJPA? Yes
-

3. Database Connection

- Should I create:
    - ✅ DataSource configuration for EJB container? Yes
    - ✅ persistence.xml for JPA? Yes
    - ✅ Connection pool settings? Yes at the requirement
- Transaction type: JTA (container-managed) or RESOURCE_LOCAL? JTA (Container-Managed)

4. Docker Setup

- Do you want:
    - ✅ Separate containers (MySQL + EJB Server + React)? Yes
    - ✅ Docker Compose file for orchestration? Yes

    - ✅ Docker volumes for MySQL data persistence? You decide
    - ✅ Network configuration between containers? You decide
- MySQL Docker image preference: mysql:8.0 or mysql:latest? MySql 8.0

5. Entity Generation

- Should I generate:
    - ✅ EJB 3.x Entity Beans (JPA entities)? yes
    - ✅ Session Beans (Stateless/Stateful)?Yes
    - ✅ DAO/Repository pattern? You decide
    - ✅ DTOs for data transfer? You decide
-

6. React Integration

- Should I create:
    - ✅ RESTful API (JAX-RS) for React frontend? Yes
    - ✅ CORS configuration for cross-origin requests? You decide
    - ✅ JSON serialization setup? You decide
    - ✅ API documentation (Swagger/OpenAPI)? You decide
-

7. Existing Data Migration

- You have v2.0.0 installed with data. Should I:
    - ✅ Create migration script (ALTER TABLE statements)? No need
    - ✅ Preserve existing data? No need
    - ✅ Create backup script before migration? No need
    - ✅ Rollback script in case of issues? No need
-

8. Payment Structure Specifics

- Invoice Reference Format: Should it auto-generate (e.g., IN + timestamp) or manual? yes
- Cashier Management: Do you need a separate `cashier` table or just store name as string? Separate Cashier Table

- Payment Gateway Integration: Will you integrate online payment gateways later? Yes
- Receipt Generation: Should system auto-generate PDF receipts? System should auto generate

9. Business Rules

- Late Penalty Calculation: Should it run automatically (scheduled job) or manual? yes
- Scholarship Expiry: Auto-expire or manual process? Auto expire according to the finance structure
- Email Notifications: Should database support email queue for payment reminders?Yes
- SMS Notifications: Support for SMS alerts? yes

10. Security & Audit

- User Authentication:
    - Integrate with existing user management? no
    - Create new user/role tables? yes
-
- Audit Logging:
    - Should all changes be logged? yes
    - Separate audit table? Yes
- Data Encryption:
    - Encrypt sensitive fields (NIC, bank details)? yes
-

11. Performance & Scale

- Expected Load:
    - How many students? 20,000
    - How many concurrent users?1000
-
- Partitioning:
    - Should I partition large tables by date/year? Yes
- Caching:
    - Redis/Hazelcast integration needed? yes
-

12. Reporting

- Should I create:
    - ✅ Database views for common reports? Ok
    - ✅ Stored procedures for complex calculations? Yes
    - ✅ Materialized views for performance? yes