QA Task Report: Prisma SQLite Implementation

1. Overview

This report summarizes the implementation of the SQL-based task using Prisma ORM with SQLite. The objective was to create a database schema, connect Prisma to SQLite, and write a test to validate database functionality.

2. Task Requirements & Implementation

✅ 1. Create an Init Method for Database with Required Tables

- Implemented a init.sql file that defines the following tables:
    - **AppUser**: Stores user information.
    - **Role**: Defines roles for users.
    - **AppUserRole**: Establishes role-based access control.
    - **UserPhone**: Stores user phone numbers.
- **Foreign key constraints** are correctly set.
- **Primary keys and default values** match the requirements.
- **Data types (TEXT****, BOOLEAN, BIGINT, DATETIME)** are properly defined.

✅ 2. Connect Prisma to SQLite

- **Prisma is correctly configured** with SQLite.
- **prisma/schema.prisma** accurately represents the database schema.
- **.env**** file** is used to configure the database connection (DATABASE_URL=file:./dev.db).
- Prisma migration (npx prisma migrate dev --name init) successfully creates and updates the database.

✅ 3. Write a Test to Show That Prisma Works

- **Created ****testPrisma.ts** to verify database operations.
- **Validates** the creation of:
    - A new user (AppUser) with a **hashed password** using bcrypt.
    - **Roles (Admin****, User)** in the Role table.

    - **User-role assignment** in AppUserRole.
    - **Phone number storage** in UserPhone.
- **Ensures data integrity** by checking for unique constraints and foreign key relationships.

3. Additional Enhancements

- **Secure Password Handling**: Implemented **bcrypt** to hash passwords before storage.
- **Unique Test Data**: Used **Faker.js** to generate unique first names, last names, emails, and phone numbers.
- **Data Integrity**: Used **upsert** to prevent duplicate role entries and maintain database consistency.

4. Conclusion

The implementation **fully meets** the task requirements while incorporating best practices for **security, uniqueness, and data integrity**. The solution is **fully functional, tested, and ready for submission**.