QA Task Report: Prisma SQLite Implementation & Web Books Page Visibility Test

1. Overview

This report summarizes the implementation of the **Prisma SQLite** task and the results of the **Web Books Login Page Visibility Test**. The goal was to set up a database using Prisma ORM with SQLite and verify database functionality, as well as to perform a visibility test on the Web Books login page to ensure that all essential elements are rendered correctly.

2. Prisma SQLite Implementation

✅ Task Requirements & Implementation

1. Database Initialization

- Implemented an init.sql file defining 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.
- Proper data types (TEXT, BOOLEAN, BIGINT, DATETIME) are used.

2. Prisma & SQLite Connection

- Prisma is correctly configured with SQLite.
- prisma/schema.prisma accurately represents the database schema.
- .env file is used for database configuration (DATABASE_URL=file:./dev.db).
- Prisma migration (npx prisma migrate dev --name init) successfully initializes the database.

3. Testing Prisma Functionality

- Created testPrisma.ts to validate database operations, ensuring:
    - Creation of a new user (**AppUser**) with a hashed password using bcrypt.

    - Roles (**Admin, User**) are properly stored in the **Role** table.
    - User-role assignments exist in **AppUserRole**.
    - Phone numbers are stored correctly in **UserPhone**.
- Ensured data integrity by verifying unique constraints and foreign key relationships.

◆ **Additional Enhancements**

- **Secure Password Handling**: Implemented bcrypt for password hashing.
- **Unique Test Data**: Used Faker.js to generate unique names, emails, and phone numbers.
- **Data Integrity**: Used upsert to prevent duplicate role entries and maintain consistency.

📌 **Conclusion**

The implementation fully meets the task requirements while incorporating best practices for security, uniqueness, and data integrity.

3. Web Books Page Visibility Test

🎯 **Test Objective**

The purpose of this test was to verify the visibility and availability of key elements on the **Web Books Login Page**. The test focused only on UI element rendering and interactivity, not functionality.

◆ **Test Scope**

- Verify the page title.
- Check the presence and visibility of essential login page elements:
    - **Email input field**
    - **Password input field**
    - **Login button**
    - **Register button**
    - **Forgot Password link**
- Ensure interactive elements (inputs and buttons) are enabled.

🛠 Test Execution

The test was conducted across multiple browsers and a mobile device using **Playwright**. The following validations were successfully performed:

- **Page Title**: Confirmed as **"Web Books"**
- **Email Input**: Visible and enabled
- **Password Input**: Visible and enabled
- **Login Button**: Visible and enabled
- **Register Button**: Visible and enabled
- **Forgot Password Link**: Visible
- A full-page screenshot was captured for documentation.

📊 Test Results

All required elements were present and visible across different environments. However, this test **did not** validate functionality (e.g., login authentication, form submission).

📌 Conclusion

The Web Books login page elements are properly rendered and interactable. Future tests should incorporate **functional validations**, including authentication, error handling, and navigation flow.

4. Final Remarks

This merged report documents the successful **Prisma SQLite implementation** and **Web Books page visibility test**. Future improvements should include additional **functional tests** for authentication and database transactions.