Spartan Community India Login Functionality Test Report

Summary

I tested the login functionality of the Spartan Community India application and found that while the authentication logic is implemented correctly, there appears to be an issue with the redirection after successful login. The application is using a mock authentication system with hardcoded users, and the authentication state is stored in localStorage.

Test Results

Regular User Login Test

- **Email**: user@example.com

- **Password**: password123

- **Result**: FAILED

- **Observations**:

- The login form submits correctly

- The authentication logic appears to be working (no visible errors)

- However, the user is not redirected to the dashboard after login

- The application remains on the login page

Admin User Login Test

- **Email**: admin@example.com

- **Password**: admin123

- **Result**: FAILED

- **Observations**:

- The login form submits correctly

- The authentication logic appears to be working (no visible errors)

- User data is stored in localStorage correctly

- However, the user is not redirected to the admin dashboard after login

- The application remains on the login page

Technical Analysis

1. **Authentication Logic**: The application uses a mock authentication system defined in `/lib/auth-context.tsx`. It has hardcoded users including:
   - Regular user: email: "riya@example.com", password: "password123"
   - Admin user: email: "admin@example.com", password: "admin123"

2. **Login Process**:
   - The login function in the auth context correctly validates credentials
   - On successful login, it stores the user data in localStorage with the key "spartanUser"
   - It attempts to redirect to the appropriate dashboard based on the user's role

3. **Issue Identified**:
   - The login function is working correctly and storing user data in localStorage
   - However, the redirection to the dashboard is not working as expected
   - This could be due to an issue with the Next.js router or a navigation guard that's preventing the redirect

4. **API Endpoint**: The application is missing the `/api/auth/login` endpoint, which returns a 404 error when accessed directly. This suggests that the API routes might not be properly configured.

Recommendations

1. **Fix API Routes**: Implement the missing `/api/auth/login` API endpoint to handle authentication requests properly.

2. **Debug Router Redirection**: Add additional logging to the router redirection logic to identify why the redirect is not working.

3. **Check Navigation Guards**: Ensure that there are no navigation guards or middleware preventing access to the dashboard routes.

4. **Update Mock Users**: Update the mock users to include the test credentials mentioned in the requirements:
   - Regular user: email: "user@example.com", password: "password123"
   - Admin user: email: "admin@example.com", password: "admin123"

5. **Implement Proper Error Handling**: Add better error handling and user feedback for login failures.

Conclusion

The login functionality of the Spartan Community India application is not working correctly due to issues with the redirection after successful login and possibly missing API endpoints. The authentication logic itself appears to be implemented correctly, but the application is not properly redirecting users to the appropriate dashboards after login.