**Navatar Project - Module Update Report**

**Module Name: Authentication System (Clerk Integration)**

**Overview of the Module**

The module I am working on is the **Authentication system** for the Navatar web application. It enables users (specifically doctors) to sign up, log in, and securely access protected parts of the application like video consultations and appointment booking.

We are using **Clerk** as the authentication service provider. Clerk provides a secure, plug-and-play solution for authentication, which includes:

* Prebuilt UI for sign-in and sign-up
* Email verification with 6-digit code (email code)
* Session management
* Account management UI
* React hooks for user data and session control

**Completed Features**

**1. Sign-In and Sign-Up Pages**

* Developed two dedicated components:

Login.jsx → For users to log in

Signup.jsx → For new users to register

* Clerk's prebuilt components (<SignIn /> and <SignUp />) are used for:

Email address collection

Sending verification codes

Creating user sessions after authentication

* Both components are wrapped in styled containers (.login-page, .login-container, .login-card) for consistent UI.

**Code Example: Login.jsx**

<SignIn

signUpUrl="/signup"

forceRedirectUrl="/landingpage"

/>

**Code Example: Signup.jsx**

<SignUp

signInUrl="/login"

forceRedirectUrl="/landingpage"

/>

**2. Account Management in Navbar**

* Integrated Clerk's UserButton component in the Navbar.jsx for signed-in users.
* When signed in, users will see:

A welcome message

A dropdown account menu with options to manage their account and log out

**Dynamic Display Based on Session:**

* SignedIn → Shows user info and UserButton
* SignedOut → Shows “Login” button

**3. Route Protection**

* Built a ProtectedRoute.jsx component using Clerk’s useAuth() hook to protect sensitive pages like:

/booking

/consultation

**Unauthenticated users are automatically redirected to the login page.**

**Code Snippet:**

if (!isSignedIn) {

return <Navigate to="/login" replace />;

}

**4. Routing Integration**

* The authentication routes (/login, /signup) and protected routes are properly added in App.jsx.
* After login or sign-up, users are redirected to /landingpage.

**5. Session and Verification Code Handling**

* Clerk handles session persistence and verification code (OTP) delivery.
* The sign-in and sign-up flows are fully functional.
* The UI responds correctly to state changes (SignedIn/SignedOut).

The **Authentication module** is integrated using Clerk, and operational with email code verification, secure login/signup, and user session handling.