**Documentation**

**Authentication Flow**

1. **Login Process**:
   * User enters username and password.
   * AuthService sends a POST request to the backend.
   * Backend validates credentials and returns a JWT token.
   * AuthService stores the token and redirects the user to the home page.
2. **Protected Routes**:
   * AuthGuard checks if the user is logged in.
   * If not logged in, user is redirected to the login page.
3. **Logout Process**:
   * AuthService clears the token and redirects to the login page.

**Backend Details**

* **AuthController**:
  + Handles the /login endpoint.
  + Validates user credentials and returns a JWT token.
* **AuthService**:
  + Loads user credentials from users.txt.
  + Validates user credentials and generates a JWT token.
* **SecurityConfig**:
  + Configures security settings and disables CSRF for simplicity.
  + Allows unauthenticated access to the login endpoint.

**Frontend Details**

* **AuthService**:
  + Manages login, logout, and authentication status.
  + Stores the JWT token in localStorage.
* **LoginComponent**:
  + Handles user login form and validation.
  + Displays success and error messages based on login status.
* **AuthGuard**:
  + Protects routes from unauthenticated access.

**Build and Deployment**

* **Backend**:
  + Use Gradle for building the project.
* **Frontend**:
  + Use npm for managing dependencies.
  + ng serve to run the frontend server.