User Story : Unauthenticated Users

**Files and their functions:**

1.js:  
**What it does:**

* Verifies that the token is signed with the correct secret key.
* Confirms that the user's role is "admin" before allowing access to protected endpoints.
* If the check fails, it blocks the request with 401 or 403 errors.

**Used in:**  
Other route files (like file uploads/deletes) to **protect sensitive routes**.

2.js :

**What it does:**

* Provides a **dummy /api/auth/login endpoint**.
* Accepts username and password in a POST request.
* This token is used by the client to access admin-only features.

**Used for:**

* Simulating login to test your protected routes.

3. js:

**What it does:**

* Defines routes for:
  + Uploading files (POST /api/files/upload)
  + Deleting files (DELETE /api/files/delete/:id)
* If no valid admin token is provided, access is denied.

**Why this matters:**

* You don’t want regular users (or unauthenticated requests) to upload or delete your archive data.

4.js

**What it does:**

* Sets up the Express app.
* Loads the routes defined in 2.js and 3.js.
* Starts the server on port 3000.