Protecting an Admin-Only Data Import Page

A role-based system must enforce permissions on the server. Hiding a link is helpful navigation, but it does not protect the page.

Store trusted session values at login

After verifying the password and retrieving the database user:

```
session_regenerate_id(true);
$_SESSION["user_id"] = (int) $user["user_id"];
$_SESSION["role"] = $user["role"];
```

The role must come from the database, not from a form field supplied by the user.

Protect the import page

Place this code before any HTML output:

```
<?php
session_start();

if (!isset($_SESSION["user_id"])) {
    header("Location: login.php");
    exit;
}

if (($_SESSION["role"] ?? "") !== "admin") {
```

```
    http_response_code(403);
    exit("You do not have permission to access this page.");
}
```

Protect processing as well as display

The role check must run on the script that processes the uploaded CSV. A user can send a request directly even when the navigation link is hidden.

You may keep the form and processing in one protected file or require the same protection file from both scripts.

Show navigation by role

```
<?php if (($_SESSION["role"] ?? "") === "admin"): ?>
  <a href="import.php">Import dataset</a>
<?php endif; ?>
```

This improves usability but is not the security control.

Test the access rules

Test	Expected result
Logged out user opens import URL	Redirected to login
Standard user opens import URL	403 response
Administrator opens import URL	Upload form appears
Standard user submits directly	Request rejected
Changed browser role field	No effect because role comes from session

Check

- [ ] Sessions start before output.
- [ ] Login and role are checked server-side.
- [ ] Processing route repeats the protection.
- [ ] Roles come from stored user records.
- [ ] Tests include direct URL access.
- [ ] Test accounts and data are fictional.

Revision #1
Created 2026-08-18 23:33:52 UTC by Mr Napper
Updated 2026-08-18 23:33:56 UTC by Mr Napper