

# Payer Mapping Tool

## Admin Guide

Version 5.0 — March 2026

## **Table of Contents**

1. Overview
2. Logging In
3. Managing Clients
4. Viewing & Exporting Data
5. Editing Client Mappings
6. Backend Management
7. Firebase Console
8. PHI Stripper Tool
9. Versioning & Deployment

## 1. Overview

The Admin Dashboard provides centralized management of all client accounts, their payer mappings, and data exports. As an admin, you can:

- Create and delete client accounts
- Set and reset client passphrases
- View all client mappings in a read-only table
- Export client data as Excel or UiPath JSON
- Open the mapping tool pre-loaded with a client's data for editing

The tool suite consists of three pages:

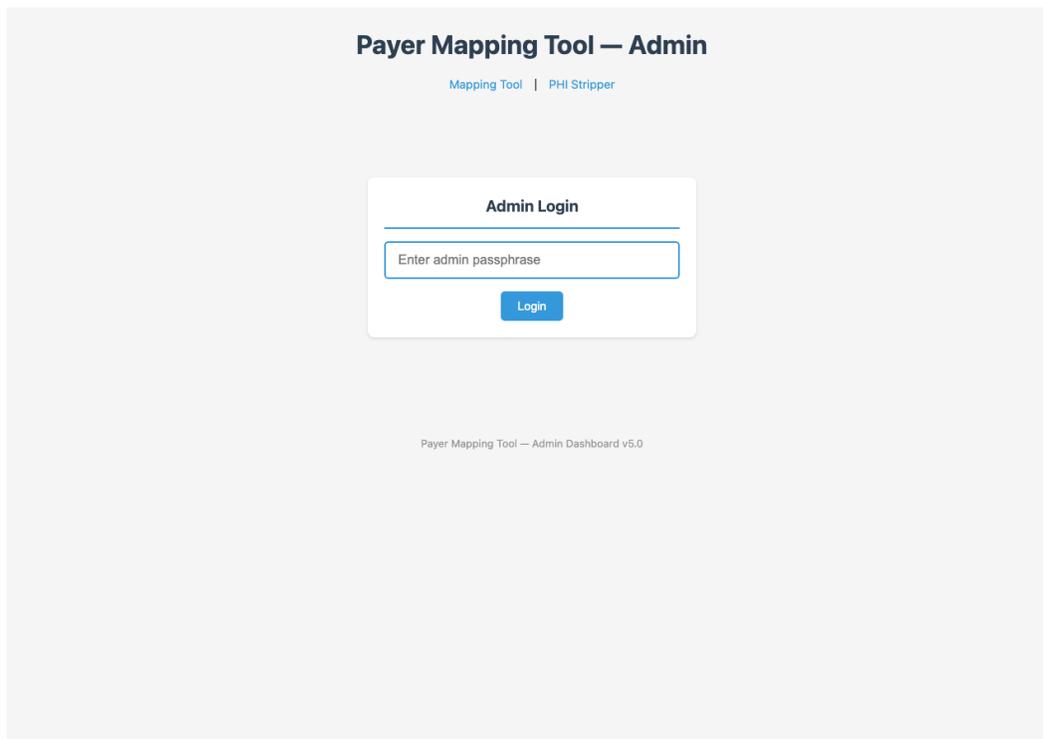
Page	Purpose
<b>Payer Mapping Tool (index.html)</b>	Main mapping interface for operations staff
<b>Admin Dashboard (admin.html)</b>	Client management and data oversight
<b>PHI Stripper (phi_stripper.html)</b>	Removes PHI from raw placement files

## 2. Logging In

Access the Admin Dashboard at:

[https://kellmat74.github.io/data\\_mapping/admin.html](https://kellmat74.github.io/data_mapping/admin.html)

1. Enter the admin passphrase in the password field
2. Click "Login"
3. The dashboard will load showing all clients

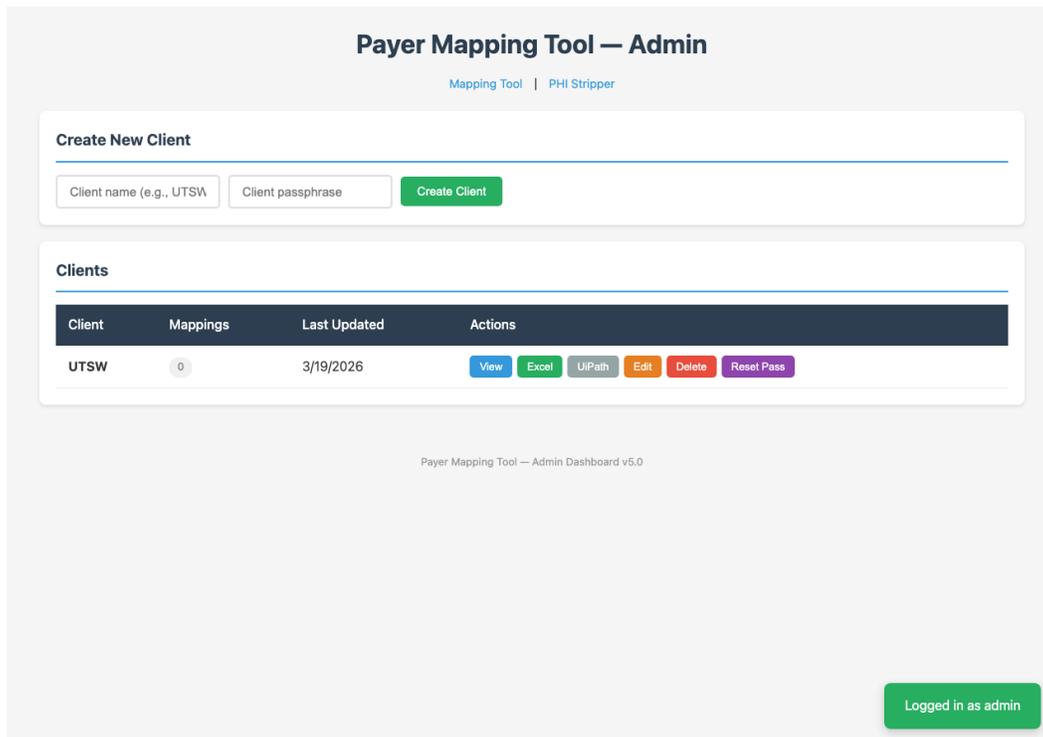


**Note:** The admin passphrase is validated against a SHA-256 hash stored in Firebase. It cannot be changed from the dashboard — it must be updated directly in the Firebase Console.

### 3. Managing Clients

#### Creating a New Client

1. In the "Create New Client" section, enter the client name (e.g., "UTSW")
2. Enter a passphrase for the client (this is what operations staff will use to connect)
3. Click "Create Client"
4. The client will appear in the Clients table below



#### Client Table Actions

Each client row has the following action buttons:

Button	Color	Action
View	Blue	Opens a modal showing all of the client's payer mappings
Excel	Green	Downloads the client's mappings as an Excel file
UiPath	Gray	Downloads the client's mappings as a UiPath JSON file
Edit	Orange	Opens the main mapping tool

		in a new tab, pre-loaded with this client
<b>Delete</b>	Red	Deletes the client and all their mappings (requires double confirmation)
<b>Reset Pass</b>	Purple	Prompts for a new passphrase and updates it in the database

## Deleting a Client

**Warning: Deleting a client permanently removes all their mapping data.**

The delete process requires two confirmations:

1. Click the red "Delete" button
2. Confirm in the first dialog
3. Type the client name in the second dialog to confirm
4. The client and all data are permanently removed

## 4. Viewing & Exporting Data

### View Mappings

Click "View" to open a modal showing all of a client's mappings in a table with columns: State, Plan Name, Payer ID, and Payer Name. This is read-only.

### Export as Excel

Click "Excel" to download the client's mappings as an Excel file. The file contains: State, Plan Name, Payer ID, and Payer Name columns. File name: payer\_mapping\_[ClientName].xlsx

### Export for UiPath

Click "UiPath" to download the client's mappings as a JSON file formatted for UiPath automation import. Plans mapped to non-Availity portals are automatically excluded. File name: [ClientName]\_uipath.json

## **5. Editing Client Mappings**

Click "Edit" on any client row to open the main Payer Mapping Tool in a new tab, pre-loaded with that client's data. This is the same interface operations staff use, allowing you to make mapping changes on behalf of a client.

Changes are saved to the cloud automatically, just like regular user sessions.

## 6. Backend Management

### Updating the Availity Payer List

The Availity payer list is stored as an Excel file in the repository. When Availity sends an updated payer list:

1. Replace the file `availity_payers.xlsx` in the repository root with the new file
2. Keep the filename exactly as "availity\_payers.xlsx"
3. Commit and push to GitHub
4. The app will automatically use the new payer list on next page load

**Important:** The file must contain "State", "Payer Name", and "Payer ID" columns. The format from Availity should work without modification.

### Deploying Updates

The tool is hosted on GitHub Pages. To deploy updates:

1. Make your changes to the HTML files
2. Commit and push to the main branch
3. GitHub Pages automatically deploys within a few minutes
4. Users will see the updated version on their next page refresh

## 7. Firebase Console

Access the Firebase Console at:

<https://console.firebase.google.com>

Project: payer-mapping-tool (Free Spark plan)

### Firestore Data Structure

Collection / Document	Contents
<code>/clients/{clientName}</code>	clientName, passphraseHash, mappings, lastUpdated, createdAt
<code>/config/admin</code>	passphraseHash (admin passphrase hash)

### Security Rules

Current Firestore security rules:

- `/clients/{clientId}` — read/write: open (passphrase validation is client-side)
- `/config/{doc}` — read: open, write: blocked (admin config can only be changed in console)

### Changing the Admin Passphrase

To change the admin passphrase:

1. Generate a SHA-256 hash of your new passphrase
2. Go to Firebase Console → Firestore Database → Data tab
3. Navigate to config → admin
4. Update the passphraseHash field with the new hash
5. Click Save

## 8. PHI Stripper Tool

The PHI Stripper ([phi\\_stripper.html](#)) is a standalone tool that operations staff use to remove PHI from raw placement files before uploading to the mapping tool.

Key points:

- Runs 100% locally in the browser — no data is transmitted anywhere
- Keeps only State and Payer columns; discards all other data
- Aggregates rows by unique (State, Plan) combinations with a volume count
- No admin configuration or backend needed
- The main mapping tool has a PHI column guard that rejects files containing known PHI columns

The screenshot shows the user interface for the 'Placement File PHI Stripper' tool. At the top, the title 'Placement File PHI Stripper' is centered, followed by the instruction 'Prepare your placement file for the Payer Mapping Tool by removing all PHI'. Below this is a green-bordered box with the text '100% Local Processing — No Data Leaves Your Computer' and a sub-note: 'This tool runs entirely in your browser. Your placement file is never uploaded or transmitted anywhere.' The main section is titled 'Upload Placement File' and contains a dashed-line box for file upload. Inside this box, it says 'Upload your standard placement file (.xlsx or .csv)' and 'Only the State and Payer 1 columns will be kept. All other data is discarded.' A blue 'Choose File' button is positioned at the bottom of the dashed box. Below the upload section is a 'How It Works' section with a numbered list: 1. Upload your standard placement file; 2. The tool finds the State and Payer 1 columns and discards everything else; 3. Rows are aggregated into unique (State, Plan) combinations with a volume count; 4. Download the clean file and upload it to the [Payer Mapping Tool](#). At the very bottom, the version number 'Placement File PHI Stripper v5.0' is displayed.

## 9. Versioning & Deployment

All three pages (index.html, admin.html, phi\_stripper.html) share the same version number, displayed in each page's footer.

Version numbering:

- Minor version increments automatically with each push (e.g., 5.0 → 5.1)
- Major version increments only when explicitly requested (e.g., 5.x → 6.0)
- The version must be updated in all three HTML files on each push

Deployment URL: [https://kellmat74.github.io/data\\_mapping/](https://kellmat74.github.io/data_mapping/)

Repository: [https://github.com/kellmat74/data\\_mapping](https://github.com/kellmat74/data_mapping)