# District Tech Lead - Use Case Index Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | **Use Case Title** | **Primary Screen(s)** | **Brief Description** |
| UC01 | Scholar Path Admin Sends Invite to a District Tech Lead | ScholarPath admin 🡪 Manage District Tech Leads | This describes how a ScholarPath Admin creates a new District Tech Lead and sends an invitation email that allows the tech lead to activate their account and access district-level tools. |
| UC02 | Complete Onboarding and District Activation | Onboarding Wizard (5 Steps) | Minimal onboarding flow covering District Profile, MOU upload, Security Setup, Optional Tech Team invites, and Final Activation. |
| UC03 | Monitor District Overview and Integration Health | Dashboard | Allows the District Tech Lead to view Data Sync Status, Security Snapshot, Compliance Alerts, and Recent Notifications with drill-down modals. |
| UC04 | Manage Compliance Documentation and Lifecycle | Compliance Documents → District tab | Upload, re-upload, delete, and track lifecycle of district-level compliance documents (MOU, Privacy Policy, Security Standards). Includes versioning and expiry workflows. |
| UC05 | Configure and Maintain Data Integrations (Manual CSV + Nightly Batch Upload) | Data Integrations | Configure Manual CSV Uploads and Nightly Batch FTP/SFTP sync. Includes upload controls, batch directory setup, sync schedule display, retry failed file syncs, and viewing sync status. |
| UC06 | Review and Analyze Data Sync Logs and Errors | Dashboard → Data Sync Status; Data Integrations → Recent Sync Errors / Data Sync Logs | View file-level errors, partial syncs, retry individual file syncs, view detailed error logs, and export logs for audit. |
| UC07 | Configure and Validate Single Sign-On (SSO) | Security Center → SSO Configuration tab | Manage Google, SAML, ClassLink SSO (tabs remain but integration list restricted). Includes attribute/claims mapping and Test Connection. |
| UC08 | Configure and Enforce Security Policies | Security Center → Security Policies | Manage password rules, MFA enforcement, session timeout, and geo-restrictions. Includes policy modal and full audit logging. |
| UC09 | Monitor Authentication Health and Access Logs | Dashboard → Security Snapshot; Security Center → Audit Logs | Monitor Failed Logins, Suspicious Logins, MFA percentages, and view detailed audit logs with export and filtering. |
| UC10 | Manage & Review All Notifications | Notifications & Settings → All Notifications tab | Unified view of Sync, Security, Compliance, and System alerts with filtering, mark as read, resolve, and export features. |
| UC11 | Manage District Tech Team Roles and Access | Notifications & Settings → Team tab | Manage District Tech Lead & Assistant Tech Leads. Assign granular permissions to assistants, deactivate/reactivate, remove, and audit all actions. |
| UC12 | Create New Resume from Scratch | Resume Builder → Create New Resume | Students manually create resumes using ATS-compliant templates with autosave, preview, and validation. Parents have view-only access. |
| UC13 | Generate AI-Assisted Resume | Resume Builder → Generate with AI | AI creates an editable resume draft from student input. Students can modify, save, or set as primary. Parents see view-only. |
| UC14 | Upload and Parse an Existing Resume | Resume Builder → Upload Resume | System parses uploaded DOCX/PDF resumes into structured sections for student editing before saving. |
| UC15 | Manage Resumes | Resume Builder → Your Resumes | Students manage up to 5 resumes: edit, duplicate, delete, download, share, rename, and set primary. |
| UC16 | Share Resume with Providers | Resume Builder → Share Resume Modal | Allows students (with parent messaging consent) to securely share resumes with employers, recruiters, or colleges. |

# Use Case 1: Scholar Path Admin Sends Invite to a District Tech Lead

**Description**

This use case describes how a ScholarPath Admin creates a new District Tech Lead and sends an invitation email that allows the tech lead to activate their account and access district-level tools.

**Actors**

**Primary Actor:** ScholarPath Admin

**Secondary Actors:** District Tech Lead (recipient)

**Goal**

To allow ScholarPath Admin to create a District Tech Lead and send an email invitation for account activation.

**Trigger**

* Admin clicks "Manage District Tech Leads" from the profile dropdown
* Admin clicks "Create District Tech Lead" button

**Business Rules**

|  |  |
| --- | --- |
| # | **Rule Description** |
| BR-01 | District Name, District Tech Lead Name, and Email are mandatory. |
| BR-02 | Send Invite button stays disabled until all mandatory fields are completed. |
| BR-03 | Email must be unique; duplicates are not allowed. |
| BR-04 | Email format validation must be applied. |
| BR-05 | New District Tech Lead is created with Pending status until they activate their account. |
| BR-06 | Upon creation, “Resend Invitation” button must appear instead of “Deactivate/Reactivate.” |
| BR-07 | Admin can resend the invitation any time before activation. |
| BR-08 | Deactivate button must not show for pending accounts. |
| BR-09 | Deactivate is disabled when district has only one active district tech lead. |
| BR-10 | Only ScholarPath Admin can create, deactivate, reactivate, or resend invitations. |

**Pre-Conditions**

* ScholarPath Admin is logged in.
* Admin has appropriate permissions.
* District name is not blank.
* The Manage District Tech Leads module is available and enabled.
* Email server/API is active and available.

**Steps**

1. **Navigation**

1. Admin logs into ScholarPath.
2. Admin clicks the Profile Icon.
3. Overlay menu appears.
4. Admin clicks Manage District Tech Leads option.
5. System displays the Manage District Tech Leads screen with the grid.
   1. District Name
   2. District Tech Lead Name
   3. Email Address
   4. No. of Schools
   5. Status (Active/Pending)
   6. Actions
   7. Expand icon for schools list
6. **Creating a District Tech Lead**
   1. Admin clicks Create District Tech Lead button.
   2. System displays the Create District Tech Lead popup.
7. **Entering Information**
   1. Admin enters District Name (mandatory).
   2. Admin enters District Tech Lead Name (mandatory).
   3. Admin enters Email Address (mandatory + email validation).
   4. System confirms fields.
   5. If all fields are valid → Send Invite button becomes enabled.
8. **Sending Invite**
9. Admin clicks Send Invite.
10. **System:**

* Saves the new District Tech Lead record.
* Assigns Pending status.
* Send invitation email to the entered email address.
* Closes the popup.
* System adds a new record in the grid with:
* Status = Pending
* Action = Resend Invitation

1. **Sent Invite Email Template**

**Send Invitation (First-Time Invitation):**

**Subject:**Your District Tech Lead Account Invitation

**Email Body:**Hello [District Tech Lead Name],

You have been added as the District Tech Lead for [District Name].  
To activate your account, please set your password using the link below:

Set Password:  
[Set Password Link]

This link will allow you to create your password and log in to your account for the first time.

If you did not expect this invitation, please contact your ScholarPath administrator.

Thank you,

ScholarPath Team.

1. **Resending Invitation**

* Admin clicks Resend Invitation.
* System displays a confirmation popup: “A new invitation email has been sent successfully to <Tech Lead Name>.”
* Admin clicks OK.
* System sends invitation email again.

**Resend Invitation email Template.**

**Resend Invitation (Second or Additional Invitation):**

**Subject:**Your District Tech Lead Invitation – Action Required

**Email Body:**Hello [District Tech Lead Name],

This is a reminder to activate your District Tech Lead account for [District Name].  
A new password setup link has been generated for your convenience:

Set Password:  
[Set Password Link]

Please use this link to create your password and complete your account activation.

If you have already activated your account, you may ignore this email.

Thank you,  
ScholarPath Team

1. Active users show **Deactivate** button.
2. Deactivate button disabled when district has only 1 active tech lead.
3. Deactivate popup must match text given.
   1. Deactivate confirmation pop up:
      1. Pop up title – **Deactivate District Tech Lead**
      2. Content – Are you sure you want to deactivate **John Smith**?

This action will at once remove their access to all district-level tools and permissions. This change cannot be undone without reactivating the user

* + 1. Buttons – Cancel, Yes,Deactivate

1. After deactivation → Reactivate button appears.
2. Reactivate popup must match given text.
   * 1. Pop up name - Reactivate District Tech Lead
     2. Content - Are you sure you want to reactivate John Smith?

Reactivating will restore their access to all district-level tools and permissions.

Buttons – Cancel, Yes, Reactivate

1. Expand icon displays associated schools list.
2. Pending users must not show deactivate option.

**Negative Flow**

|  |  |
| --- | --- |
| **Scenario** | **Description** |
| NF-01 | Missing required fields → Send Invite button stays disabled. |
| NF-02 | Invalid email format → System shows “Invalid email address.” |
| NF-03 | Duplicate email → System shows “This email address is already associated with another District Tech Lead.” |
| NF-04 | Network issue sending invite → System displays retry message. |

**Post-Condition**

* New District Tech Lead record appears in grid with Pending status.
* Invitation email is delivered to the tech lead.
* Admin can resend invitation any time before activation.

**Special Requirements**

* Email must include an activation link.
* Activation link must expire after configured duration.
* System must log email events in audit logs.
* High accessibility standards (WCAG-compliant form fields).

**Constraints**

* User cannot be deactivated if they are the only tech lead for the district.
* Only Pending users get the “Resend Invitation” button.
* Only Active users can be deactivated.
* System must block multiple submissions (browser double-click protection).

**Screen Element Matrix**

|  |  |  |
| --- | --- | --- |
| **Screen Element** | **Type** | **Description** |
| Manage District Tech Leads Menu | Navigation | Opens Manage Tech Lead screen |
| Create District Tech Lead Button | CTA | Opens popup |
| District Name | Input | Mandatory |
| District Tech Lead Name | Input | Mandatory |
| Email Address | Input | Mandatory, validated |
| Cancel Button | CTA | Closes popup |
| Send Invite Button | CTA | Enabled only when mandatory fields are valid |
| Grid Row | Data Row | Displays district tech lead details |
| Resend Invitation Button | CTA | Sends another invite email |
| Status Badge | Label | Shows Pending/Active |
| Popup: Resend Invitation | Confirmation | Shows success message |

**Screenshot**

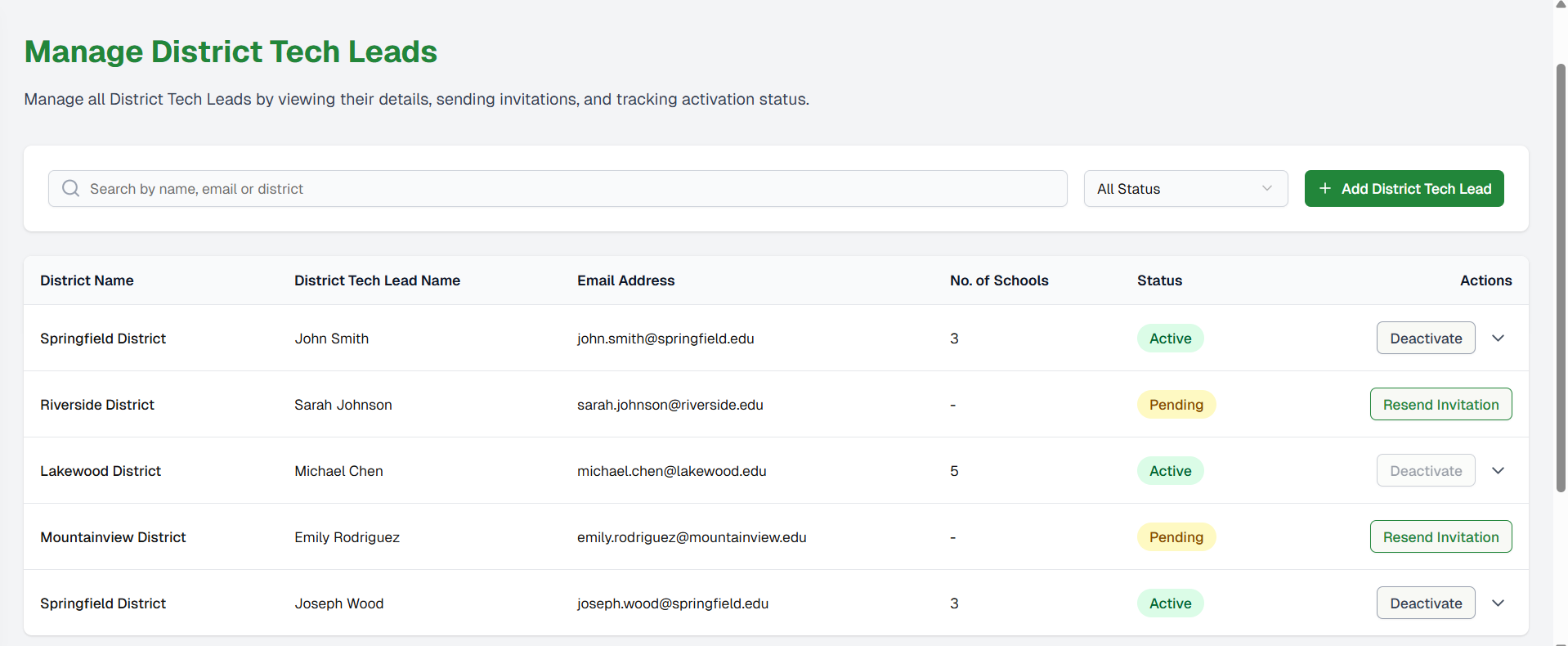


Figure 1 Manage District Tech leads grid

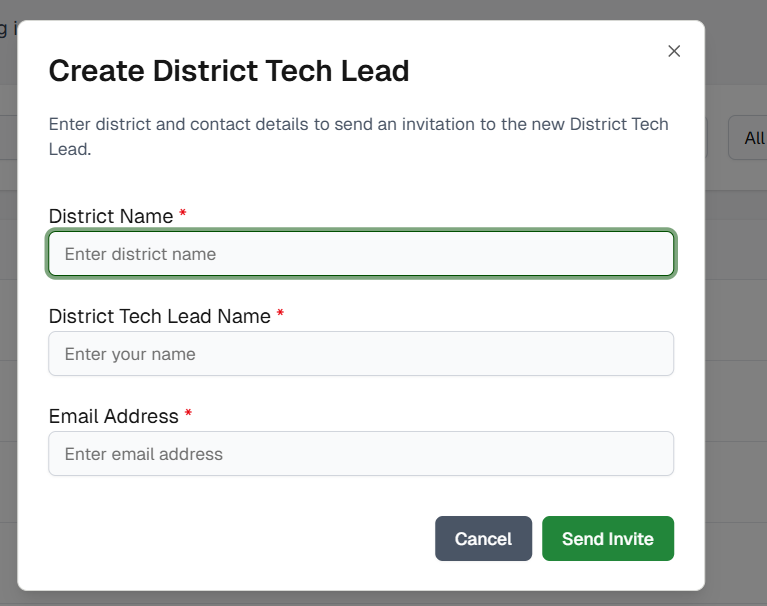


Figure 2: Create District Tech Lead pop up.

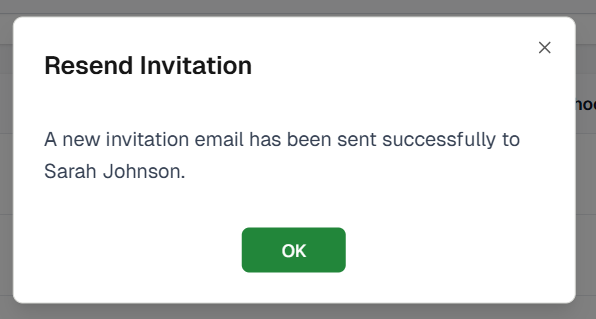


Figure 3: Resend Invitation confirmation pop up.

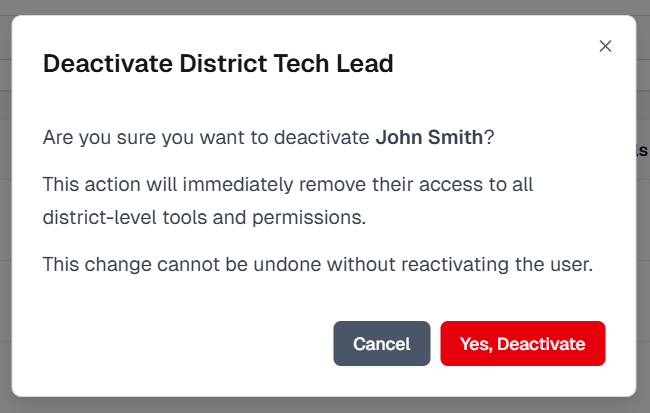


Figure 4: Deactivate District Tech Lead confirmation pops up

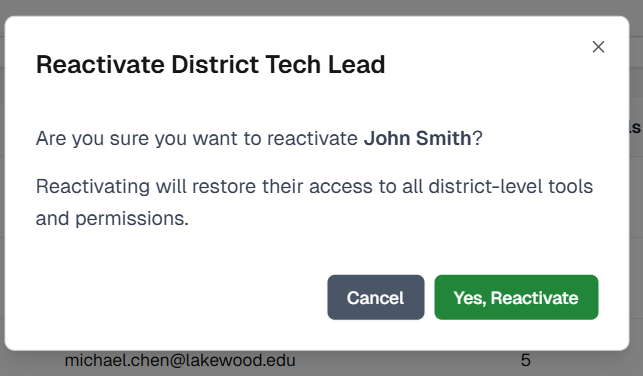


Figure 5: Reactivate District Tech Lead confirmation pop up

**Involved APIs**

|  |  |
| --- | --- |
| **Action** | **API** |
| Create District Tech Lead | POST /api/admin/district-tech-leads |
| Send Invitation | POST /api/admin/send-invite |
| Resend Invitation | POST /api/admin/resend-invite/{id} |
| Get District Tech Leads List | GET /api/admin/district-tech-leads |
| Validate Email | GET /api/admin/validate-email |

# Use Case 2: District Tech Lead Onboarding & District Activation (Updated to Minimal Flow)

**Description**

This use case describes how a District Tech Lead completes the minimal ScholarPath onboarding wizard, consisting of 5 lightweight steps:

1. District Profile Setup
2. Upload MOU
3. Basic Security Setup
4. Invite Additional Tech Team (Optional)
5. Completion Screen

The onboarding is intentionally short, focusing only on required data to activate the district account. After completion, the District Tech Lead gains access to all platform features, including Data Integrations, SSO, Notifications, and Security settings.

**Actors**

* **Primary:** District Tech Lead
* **Secondary:** ScholarPath Admin (reviews MOU after onboarding)

**Goal**

To activate the district environment so the District Tech Lead can access the full ScholarPath platform.

**Trigger**

District Tech Lead logs in for the first time, or onboarding is still incomplete.

**Business Rules**

1. The District Tech Lead must complete all onboarding steps before accessing the main platform.
2. Only one onboarding flow exists — no skipping except where explicitly allowed (Step 4).
3. Required fields must be filled out to proceed.
4. MOU upload is mandatory; however, Admin approval is not required to complete onboarding.
5. Onboarding is shown only once; after completion, it never appears again.
6. Onboarding auto-saves progress after each step.
7. If the user refreshes or logs out, they resume at the last incomplete step.
8. District is considered “Active” once onboarding completes.

**Pre-Conditions**

* District Tech Lead account exists and is verified.
* District is marked as Onboarding Required.
* User has not completed onboarding previously.

**Steps**

**STEP 1 — District Profile Setup**

1. System auto-opens Step 1: District Profile.
2. Screen displays:
   * District Name (read-only)
   * District Tech Lead Name
   * Email (read-only, verified)
   * Phone Number (optional)
   * Time Zone (required)
3. User completes required fields.
4. User clicks Save & Continue.
5. System validates and saves profile data.
6. Move to Step

**STEP 2 — Upload MOU**

1. Screen displays MOU upload card:
   * Drag/drop or click to upload PDF.
2. User uploads file.
3. System:
   * Validates file type (.pdf)
   * Stores file
   * Marks status as “Pending Review”
4. User clicks Upload & Continue.
5. Move to Step 3.

**STEP 3 — Basic Security Setup**

1. Screen shows minimal settings:
   * Require MFA (toggle ON by default)
   * Session Timeout (dropdown: 15m, 30m, 60m)
2. User edits values if desired.
3. User clicks Save & Continue.
4. System stores security settings.
5. Move to Step 4.

**STEP 4 — Invite District Tech Team (Optional)**

1. Screen displays:
   * Description: Optional step to invite members.
   * Add row: Name, Email.
2. User either:
   * Clicks Skip to continue
   * or Clicks Send Invites & Continue after entering valid rows
3. System sends invitations (only if user clicked Send).
4. Move to Step 5.

**STEP 5 — Onboarding Complete**

1. System displays a success screen:
   * “Your District Is Ready”
   * Four quick links:
     + Configure SIS Sync → Data Integrations
     + Set Up SSO → SSO & Authentication
     + Upload Additional Compliance Documents → Compliance Documents
     + Review Security Policies → Security & Controls
2. User clicks Go to Dashboard.
3. Onboarding is marked complete, district becomes Active.
4. User now enters the full platform.

**Negative Flow**

**A. Missing Required Fields (Step 1)**

* If required fields are blank → show validation error.
* Continue button is disabled.

**B. Invalid File Format (Step 2)**

* If file is not PDF → show inline error; block advancement.

**C. Security Settings Not Saved (Step 3)**

* Network errors → show top banner retry option.

**D. Invalid Email for Invitees (Step 4)**

* If “Send Invites” clicked but email format invalid → inline red error.

**E. User Refreshes Browser**

* System reloads onboarding and jumps automatically to the last incomplete step.

**Post-Conditions**

* District status becomes Active.
* Onboarding wizard is permanently disabled for this district.
* MOU is stored and available to ScholarPath Admin for later review.
* User can access the Dashboard and all navigation tabs.
* Optional Tech Team members receive their invite emails.

**Special Requirements**

* UI must follow ScholarPath style:
  + Center card layout
  + ScholarPath green (#0A8A45)
  + Stepper navigation
  + Clean, distraction-free design
* Auto-save after each step
* No ability to exit onboarding early

**Constraints**

* Only District Tech Lead can complete onboarding.
* No advanced security settings (geo-restrictions, password rules).
* No data integrations required during onboarding.
* MOU must be uploaded but not necessarily approved to finish onboarding.

**Screen Element Matrix (SEM)**

For all onboarding screens combined

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Type** | **Description** | **Validations** | **Dependencies** |
| Stepper | Indicator | Shows Step 1–5 | — | Progress tracking |
| District Name | Read-only text | Prefilled from backend | — | District record |
| Name Field | Input | District Tech Lead name | Required | None |
| Email | Read-only | Verified email | — | User account |
| Phone | Input | Optional phone | Numeric validation | None |
| Time Zone | Dropdown | User selects | Required | None |
| MOU Upload | Dropzone | Only PDF | Required | File validation |
| MFA Toggle | Switch | Default ON | — | None |
| Session Timeout | Dropdown | 15m/30m/60m | Required | None |
| Invite Row | Table Inputs | Name, email | Validation only if sending invites | None |
| Send Invites | Button | Sends emails | Only if valid | Email service |
| Skip | Link | Moves to next | — | Optional step |
| Final CTA | Button | Go to Dashboard | Always enabled | Onboarding completed |

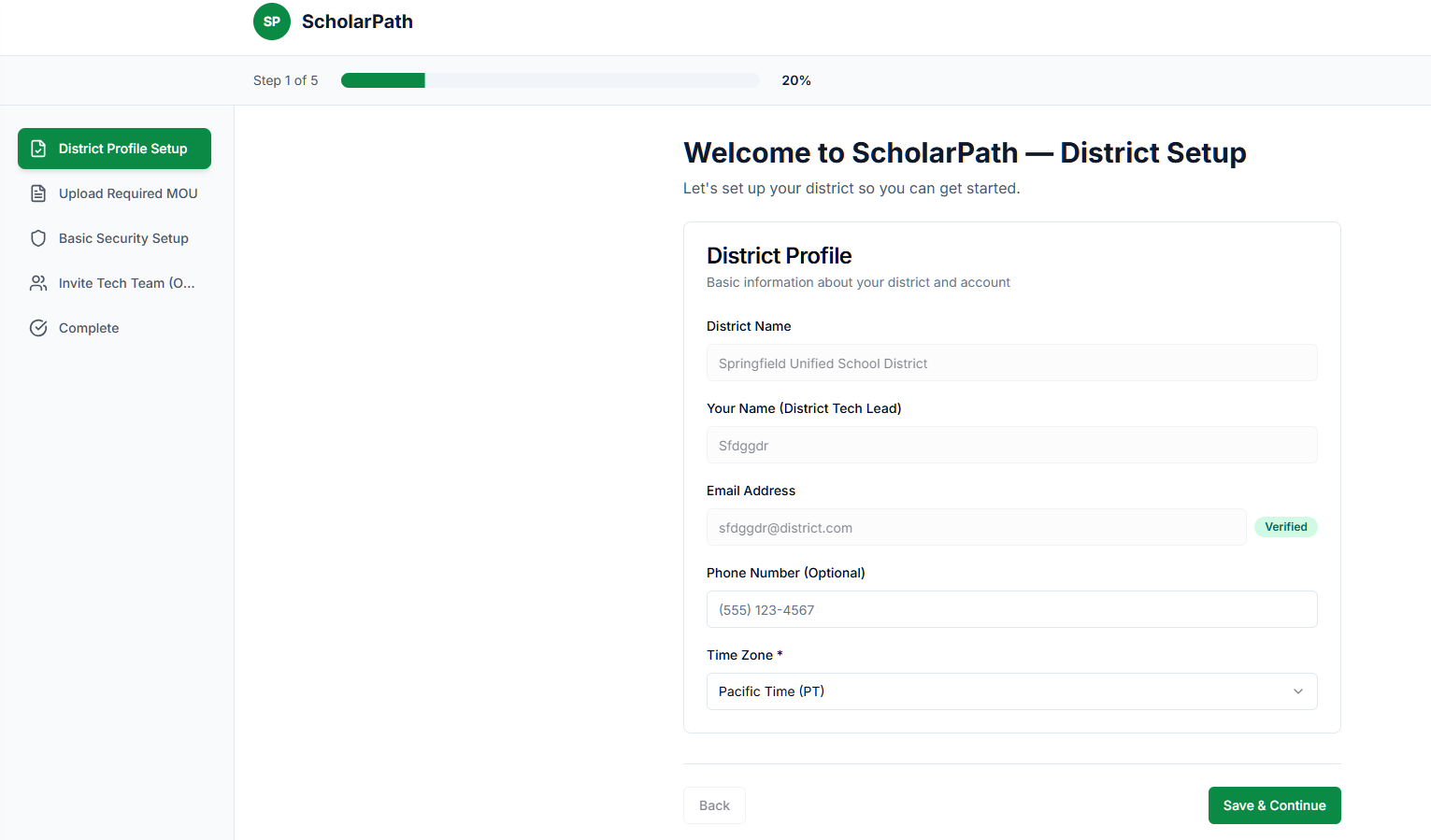
**Screenshots**

Figure : District Setup

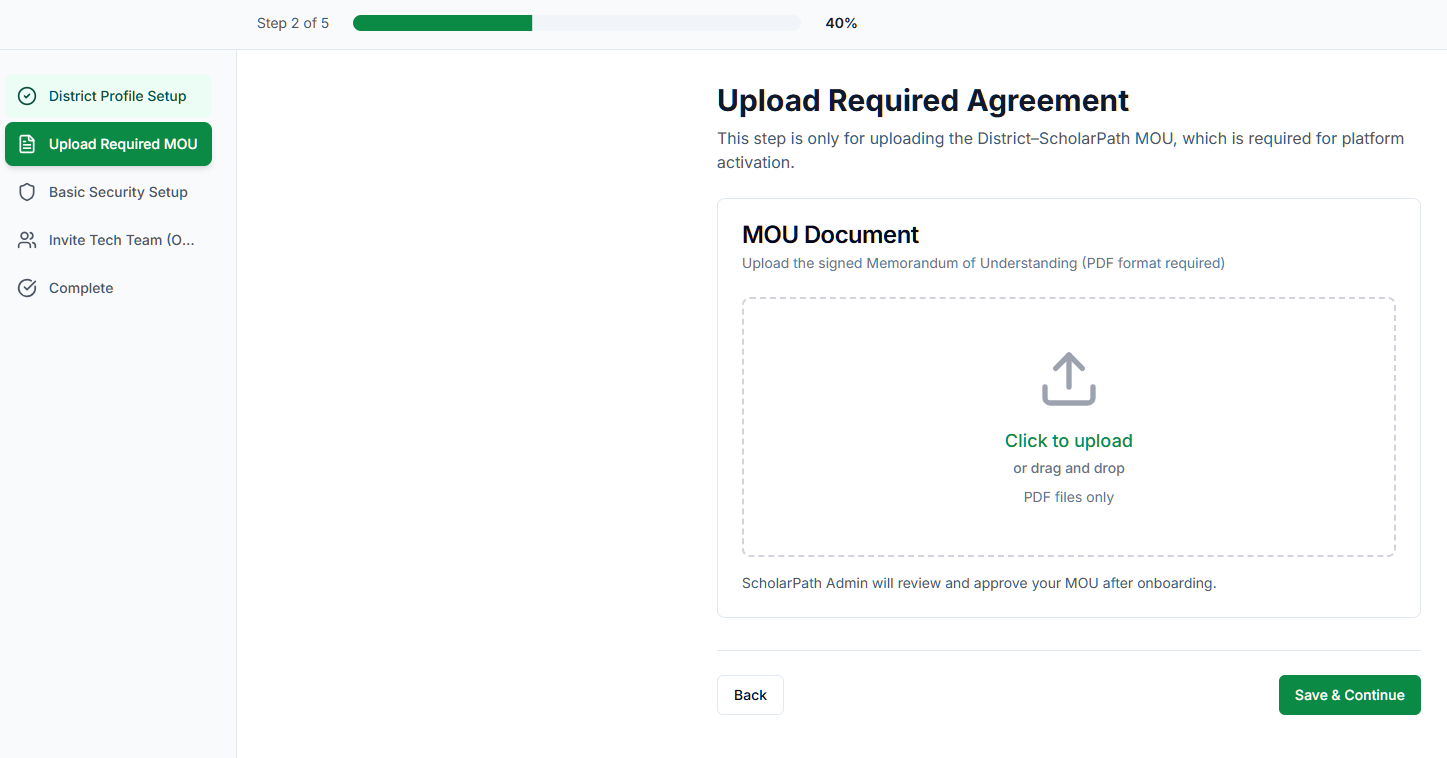


Figure : Upload MOU

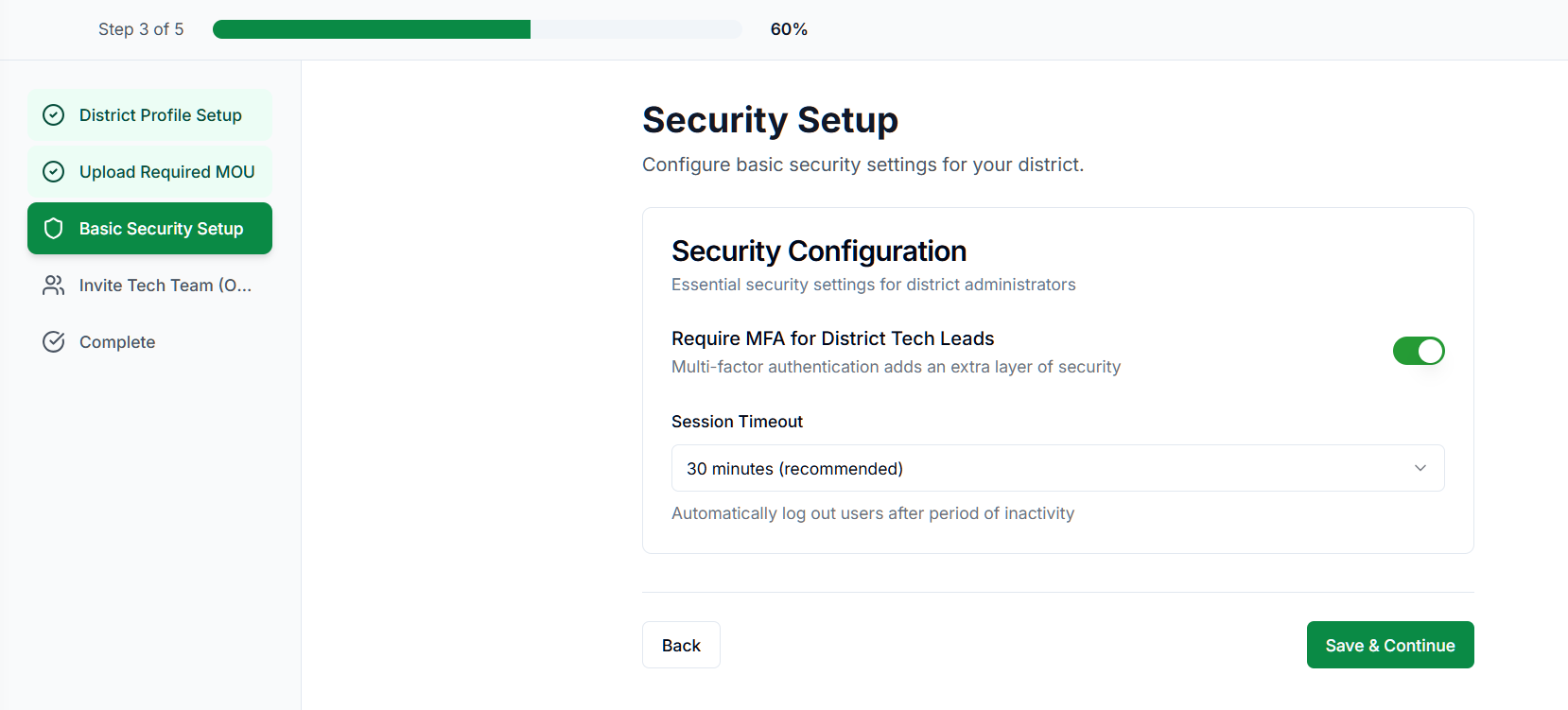


Figure : Security Setup

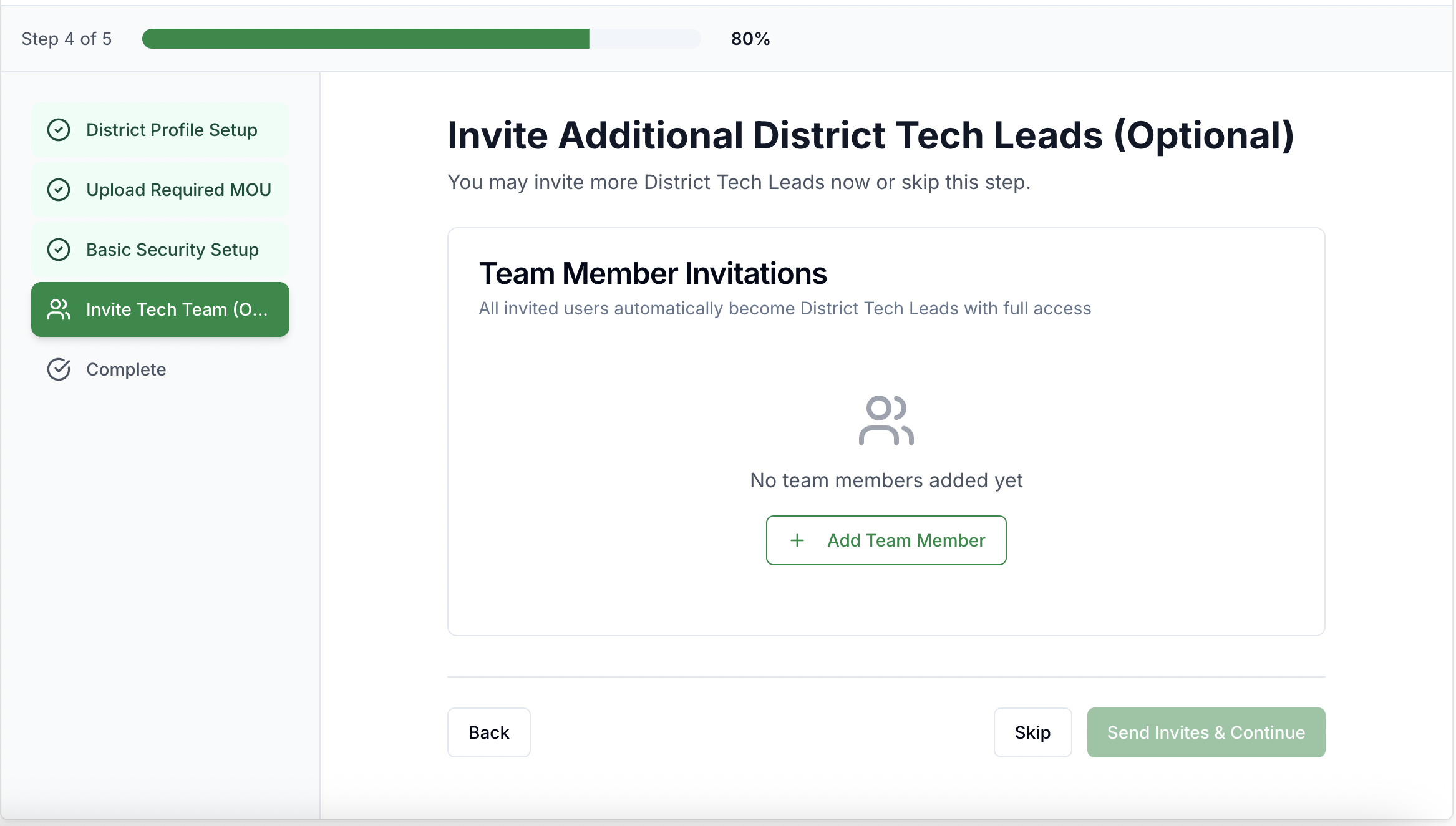


Figure : Invite District Tech Team Members

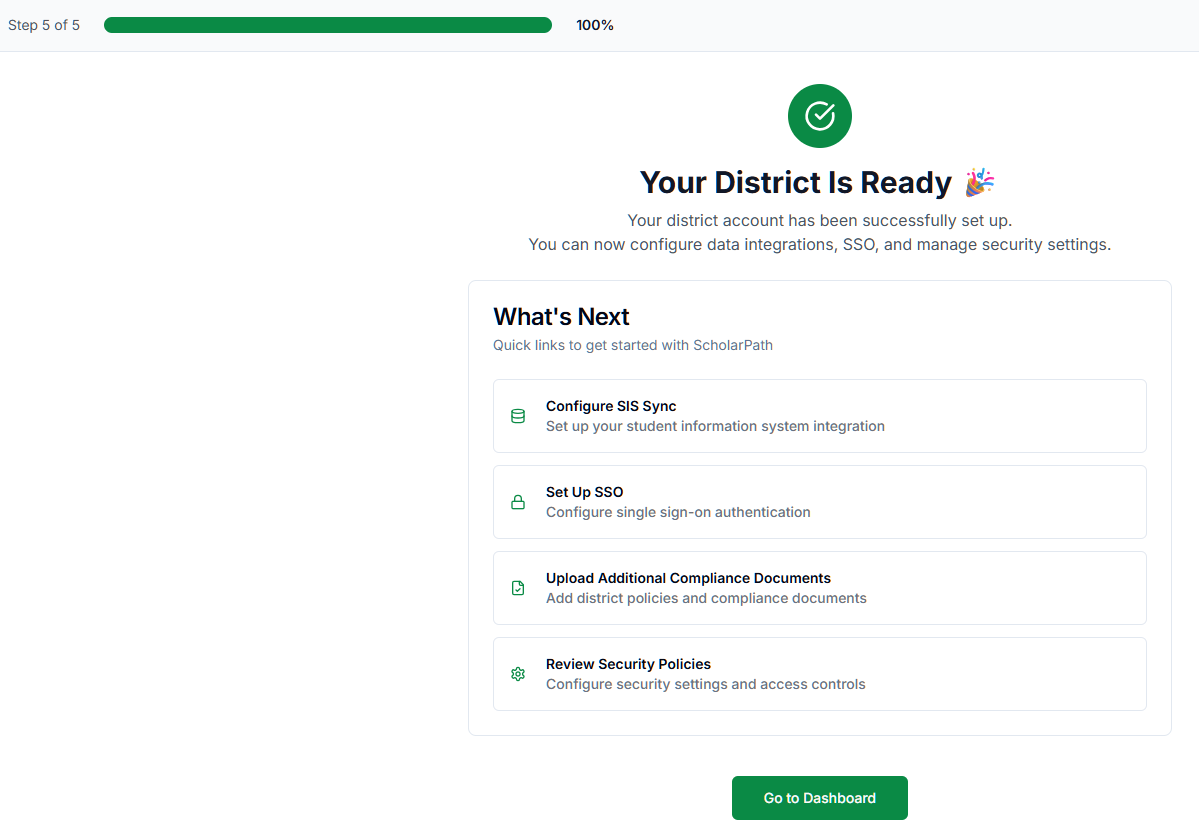


Figure : Complete Onboarding Setup

**APIs Involved (Updated)**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When Called** |
| Get Onboarding Status | GET /onboarding/status | Returns current onboarding step | On load |
| Save District Profile | POST /onboarding/profile | Saves Step 1 data | Save & Continue |
| Upload MOU | POST /onboarding/mou | Uploads PDF | Step 2 |
| Save Security Settings | POST /onboarding/security | Saves MFA + timeout | Step 3 |
| Invite Tech Members | POST /onboarding/invite | Sends invite emails | Step 4 (Send Invites) |
| Complete Onboarding | POST /onboarding/complete | Marks district active | Step 5 |
| Get MOU Review Status | GET /documents/mou | Shows Pending Review | Step 2/Completion |

# Use Case 3: Monitor District Overview, Data Sync Status & Security Health (Dashboard)

**Description**

This use case describes how the District Tech Lead uses the Dashboard screen to view:

* High-level district statistics (Students, Staff, Schools)
* Data Sync Status (manual sync + nightly FTP batch sync)
* Security Snapshot (failed logins, MFA adoption, suspicious logins)
* Documents & Compliance Alerts
* Recent Notifications
* File-level sync errors via View Error Logs modal

It includes all interactions from the dashboard including deep-linking, modal opening, and clickable alerts.

**Actors**

* **Primary Actor:** District Tech Lead
* **Secondary Actors:** District Tech Team (view-only), ScholarPath Admin

**Goal**

To quickly assess the overall health of district data, security, document compliance, and system activity using the dashboard.

**Trigger**

District Tech Lead logs in and lands on the Dashboard.

**Business Rules**

1. Dashboard loads data from the last 24 hours + scheduled sync information.
2. “View Error Logs” opens the Recent Sync Errors modal (file-level errors only).
3. Security Snapshot numbers must be clickable and open the relevant screen/modal.
4. Compliance alerts must be clickable and deep-link to the proper tab/folder in the Compliance Documents screen.
5. “View All Notifications” opens Notifications & Settings → All Notifications tab.
6. Sync status must display both:
   * Last Sync Time
   * Last Sync Source (Manual or FTP Batch Upload)
   * Next Sync Time
   * Next Sync Source
7. If there are no sync errors, the error box is not shown.

**Pre-Conditions**

* User has completed onboarding.
* District is in “Active” state.
* At least one sync has occurred (manual or batch).
* Security events and login data exist in the system.

**Steps**

**View District Overview Cards**

1. User sees 3 high-level insight tiles:

1. Total Students
2. Total Staff
3. Active Schools

2. Each tile may show a trend badge (e.g., +3.2%).

3. No interaction needed; these are display-only metrics.

**Review Data Sync Status Section**

1. User sees:

* Last Sync: “2 hours ago”
* Last Sync Source: Manual
* Next Sync: “In 22 hours”
* Next Sync Source: FTP Batch Upload
* Error banner, if applicable:

⚠️ 2 sync errors detected. View Error Logs

1. View Error Logs
2. Opens a modal, as shown in screenshot:

**Recent Sync Errors Modal**

Columns:

* File Name
* Status (Failed / Partial)
* Total Records
* Failed Records
* Error Summary
* Actions: Retry File, View Errors

Example:

* student\_data\_nov\_2025.csv → Failed
* staff\_data\_nov\_2025.csv → Partial

**User Actions in Modal**

**View Errors**

* Opens file-level detailed error modal
* Shows row-level errors
* User can export errors

**Retry File**

* Appears ONLY for FTP batch-uploaded errors
* Immediately triggers reprocessing of that single file
* Does *not* trigger a full batch sync
* Manual uploads do NOT show Retry File

**View Full Logs**

* Navigates user to Data Integrations → Sync Logs
* Pre-filters to the most recent batch

**Close**

* Closes modal

**Step 3 — Review Security Snapshot Section**

User sees:

* Failed Logins (24h): 3
* MFA Adoption: 78%
* Suspicious Logins: 1

All numbers are clickable:

**3.1 Clicking Failed Logins**

Navigates to:

* Security & Controls → Access Monitoring
* Pre-filtered to show failed login events

**3.2 Clicking MFA Adoption**

Navigates to:

* Security & Controls → Audit & Logging Controls
* Opens Security Events modal filtered by MFA events

**3.3 Clicking**

**Suspicious Logins**

The column name Risk Score should be changes as Severity in Authentication Metrics >> Suspicious login modal

The column name Risk Level should be changes as Severity in Dashboard >> Suspicious login modal

Navigates to

* Security & Controls → Access Monitoring
* Filters for “Suspicious” events

**Step 4 — Review Documents & Compliance Alerts**

User sees clickable alerts, e.g.:

Privacy policy for Polar Middle School is missing.

Privacy policy for Washington Middle School is missing.

**Clicking a compliance alert performs:**

1. Navigates to **Compliance Documents** screen
2. Auto-selects **Schools** tab
3. Auto-opens the school’s folder
4. Highlights the missing document type

If folder doesn’t exist → show “No documents uploaded” state.

**Step 5 — Review Recent Notifications**

User sees most recent notifications:

* Sync failed
* Document expiring
* Suspicious login
* System state alerts

**Clicking View All Notifications**

Navigates to:

* Notifications & Settings → All Notifications tab

**Negative Flow**

1. No Sync Errors Exist

Error banner is hidden “View Error Logs” link is not shown

B. Sync Errors Modal Cannot Load

Unable to load sync error details. Please try again.

C. Compliance Document Not Found

If clicking an alert and the corresponding folder/file is missing:

Show banner:

Document folder not found — please upload missing documents.

D. Security Snapshot Data Failure

Show fallback:

No security data available. Please refresh.

**Post-Conditions**

* User gains accurate understanding of data sync, security posture, and compliance issues.
* User may move to deeper areas (Data Integrations, Security, Compliance).
* Any modal opened can be closed and user returns to Dashboard seamlessly.

**Special Requirements**

* UI must match ScholarPath theme
* Errors use red (#D9534F), warnings orange (#F0AD4E), success green (#0A8A45)
* Sync modal must support horizontal scrolling
* Sync errors are refreshed every 5 minutes
* Security numbers refresh every 10 minutes

**Constraints**

* Only District Tech Lead sees the full dashboard
* Retry File is only enabled for FTP batch files
* Manual CSV sync errors cannot be retried

**Screen Element Matrix (SEM)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Element** | **Type** | **Description** | **Validations** | **Dependencies** |
| Total Students Tile | Card | Shows student count | — | Student dataset |
| Total Staff Tile | Card | Shows staff count | — | Staff dataset |
| Active Schools Tile | Card | Shows active schools | — | School dataset |
| Last Sync | Text | Shows “X hours ago” | — | Sync logs |
| Last Sync Source | Text | Manual / FTP Batch Upload | — | Sync config |
| Next Sync | Text | Time until next sync | — | Scheduler |
| Next Sync Source | Text | FTP Batch Upload | — | FTP config |
| Sync Errors Banner | Alert | Shows number of errors | Hide if 0 | Sync results |
| View Error Logs | Link | Opens Sync Errors modal | Only if errors > 0 | Modal component |
| Sync Errors Modal | Modal | Shows list of file errors | — | Latest batch errors |
| Retry File | Button | Retries file | Only for FTP batch | Sync engine |
| View Errors | Button | File error details | File must have errors | Error store |
| Security Metrics | Cards | Failed logins, MFA, Suspicious | — | Security logs |
| Compliance Alerts | List | Missing documents | — | Compliance engine |
| View All Notifications | Link | Opens All Notifications | — | Notifications screen |

**Screenshots**

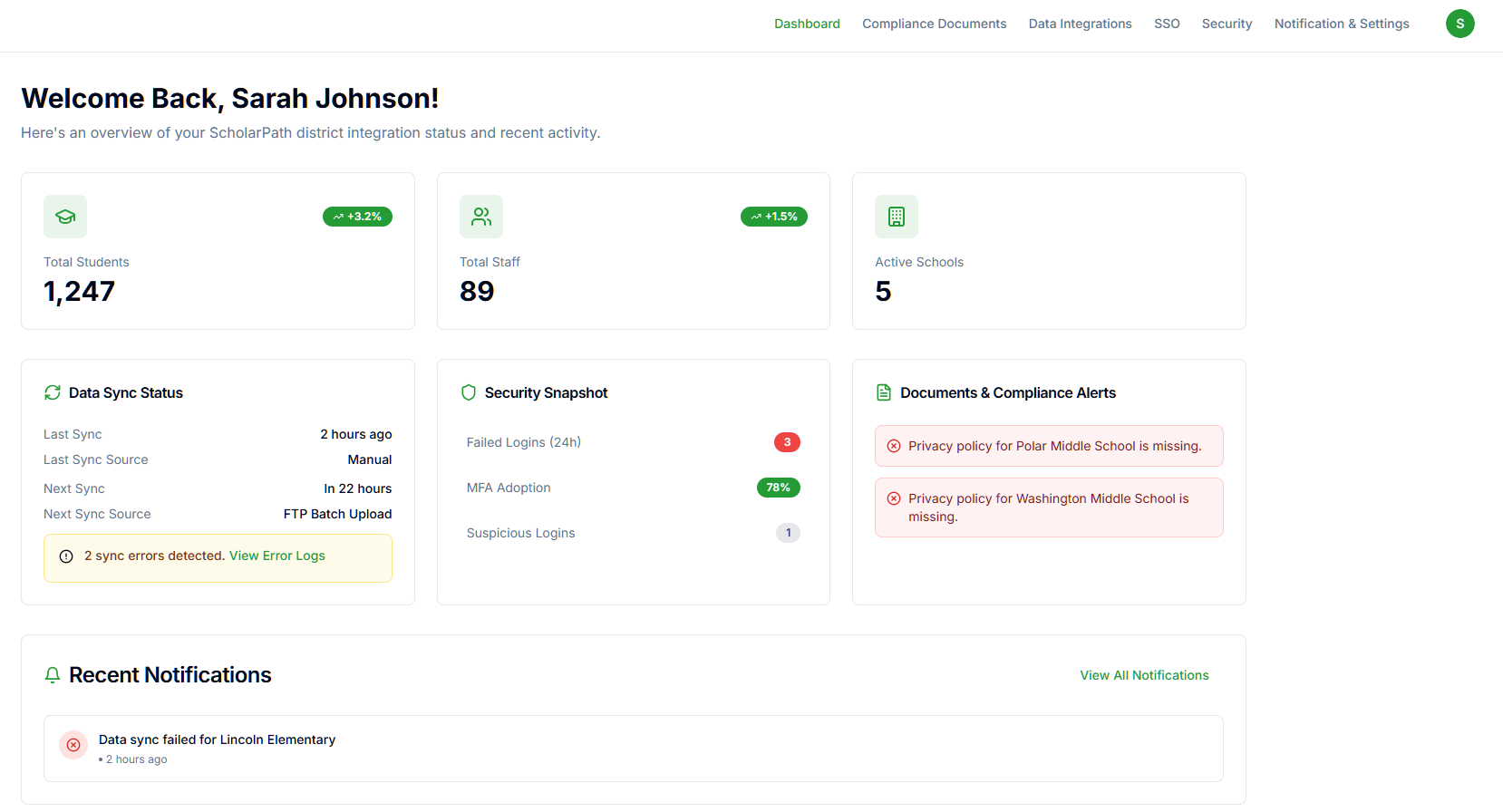


Figure : Dashboard Screen

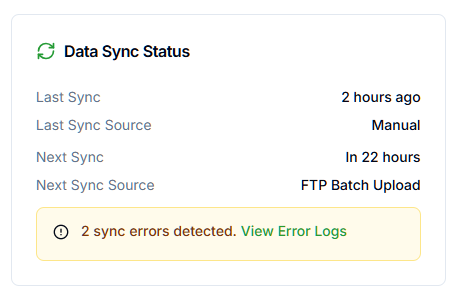


Figure : Data Sync Status

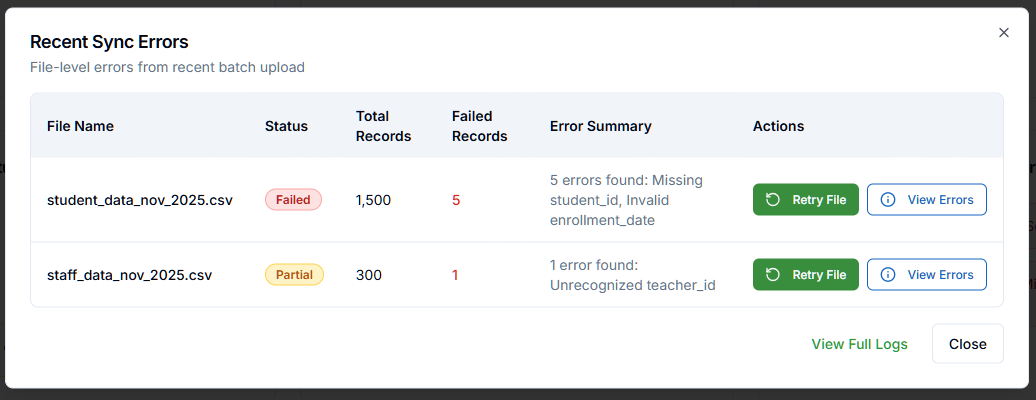


Figure : Recent Sync Errors

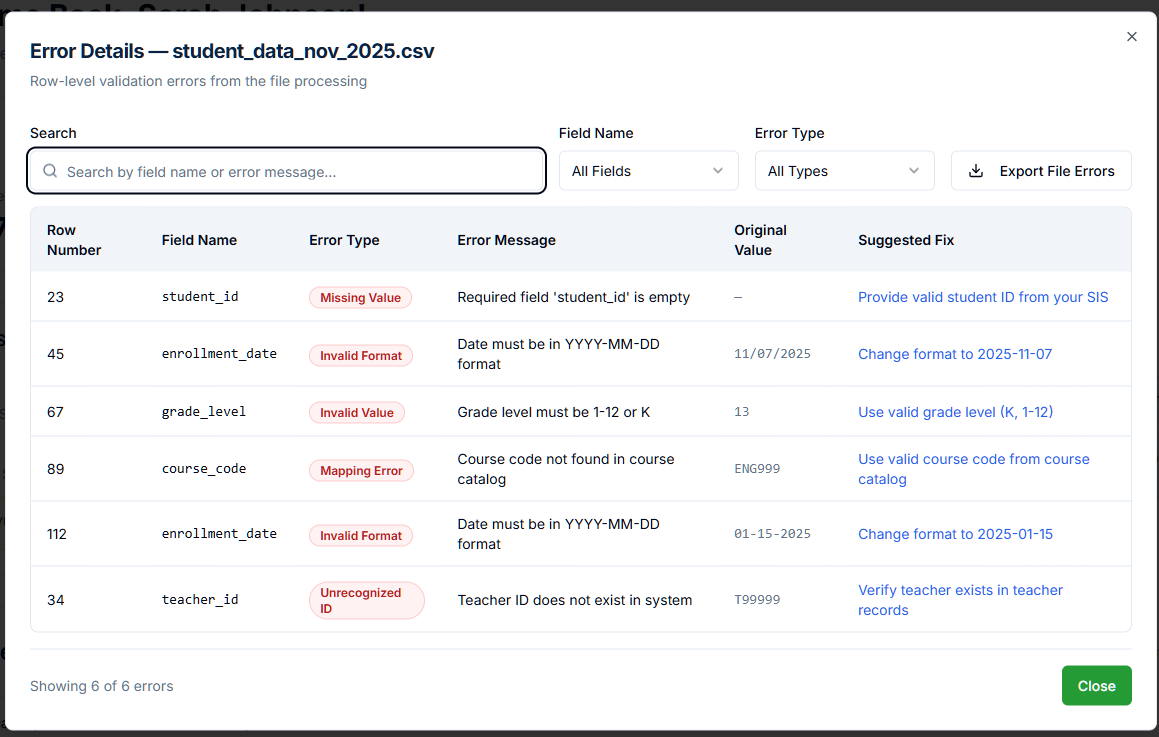


Figure : Error Details



Figure : Security Snapshot

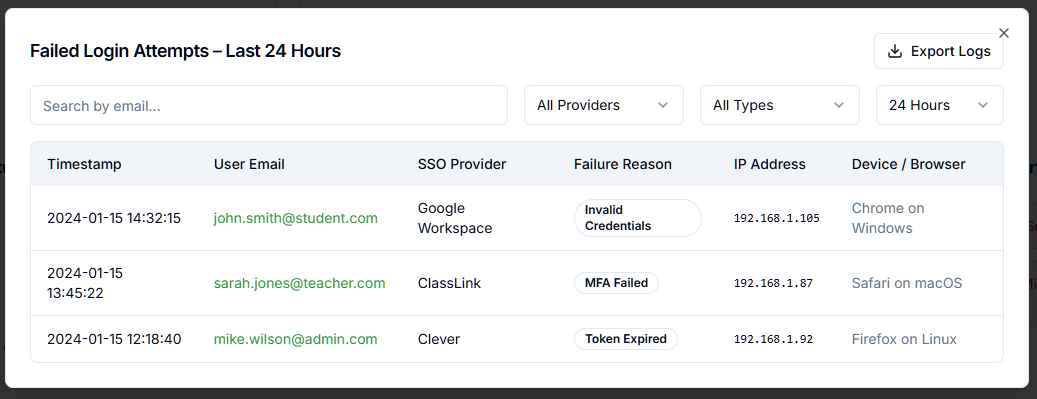


Figure : Failed Login Attempts

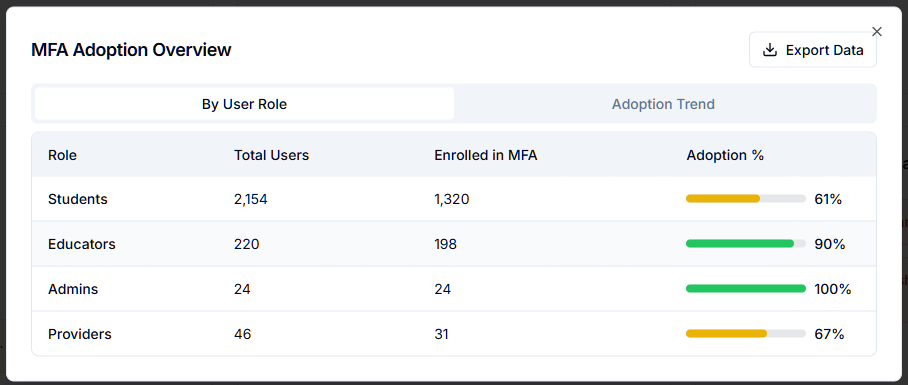


Figure : MFA Adoption Overview



Figure : MFA Adoption Trend

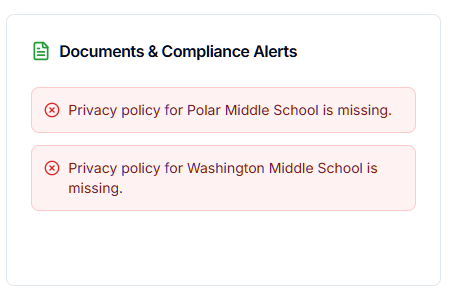


Figure : Documents & Compliance Alerts

* staff\_data\_nov\_2025.csv

These UIs have been fully incorporated into the flows.

**APIs Involved**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When Called** |
| Get Dashboard Metrics | GET /dashboard/overview | Loads counts, alerts, sync info | On dashboard load |
| Get Recent Sync Errors | GET /sync/errors/recent | Gets file-level errors | On clicking View Error Logs |
| Retry Single File | POST /sync/retry-file | Retries a failed batch file | Retry File button |
| Get File Error Details | GET /sync/errors/:fileId | Fetch detailed errors | View Errors |
| Get Security Snapshot | GET /security/snapshot | Loads failed login count, MFA%, suspicious | Dashboard load |
| Get Compliance Alerts | GET /compliance/alerts | Missing documents | Dashboard load |
| Get Notifications | GET /notifications/recent | Recent notifications | Dashboard load |

# Use Case 4: Manage Compliance Documentation and Lifecycle

**Description:**

This use case defines how the District Tech Lead manages all district-wide and school-level compliance documents within the Documentation & Compliance module.

It covers:

* Uploading, re-uploading, replacing, and deleting documents
* Managing document lifecycle (Pending Review → Active → Soon to Expire → Expired)
* Viewing ScholarPath-provided documents (read-only)
* Maintaining school-specific documents
* Handling dashboard deep-linking for missing school documents
* Monitoring document health across district & schools
* Viewing the Audit Trail for all document activity

This ensures district-wide FERPA and data protection compliance.

**Actors**

Primary Actor

* District Tech Lead

Supporting Actors

* ScholarPath Admin (reviews & verifies documents)
* System (for status computation, alerts, validation)

**Goal**

To enable the District Tech Lead to manage all compliance documents—district-level and school-level—ensuring they remain accurate, verified, and compliant with ScholarPath and FERPA requirements.

**Trigger**

User navigates to Compliance Documents screen

OR

User clicks a compliance alert on the Dashboard, which deep-links into a specific school folder.

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **#** | **Rule** | **Description** |
| BR-01 | MOU Scope | MOU is only between District ↔ ScholarPath (not schools). |
| BR-02 | Admin Verification | Documents must be reviewed & approved by ScholarPath Admin before marked Active. |
| BR-03 | Lifecycle Status | Pending Review → Active → Soon to Expire → Expired. |
| BR-04 | Re-Upload Control | Re-upload allowed only when document is Soon to Expire or Expired. |
| BR-05 | Deletion Restriction | Only expired or unverified (Pending Review) documents can be deleted. |
| BR-06 | Expiry Reminder | Alerts appear 30 days before expiry (Dashboard + Notifications). |
| BR-07 | File Type Control | PDF only, Max size 10MB. (DOCX removed for FERPA compliance.) |
| BR-08 | One Active Version | Only one Active document per type across district/school. |
| BR-09 | Versioning | Every re-upload creates a new version number, old versions auto-archived. |
| BR-10 | School Folder Structure | Each school has its own folder with required doc categories. |
| BR-11 | Deep Linking | Clicking dashboard alert must open Schools tab → specific school → required folder. |
| BR-12 | ScholarPath Documents | Read-only documents uploaded by ScholarPath Admin appear under the ScholarPath tab. |

**Pre-Conditions**

* District status = Active (onboarding complete)
* District Tech Lead has document permissions
* Compliance document structure (District, Schools, ScholarPath tabs) is created
* ScholarPath Admin has access to verify documents

**Steps**

1. User Opens Documentation & Compliance Screen

System displays four tabs:

1. ScholarPath
2. District
3. Schools
4. Audit Trail

**2. District Tab Flow**

**2.1 User views District-level document cards:**

* MOU
* Privacy Policy
* Security Standards

System displays:

* Status badge
* Expiry date
* Version number
* Verification date

**2.2 Upload or Replace Document**

|  |  |  |
| --- | --- | --- |
| **Step** | **Action** | **System Behavior** |
| 2.2.1 | User clicks **Upload** or **Replace** | File picker opens |
| 2.2.2 | User selects the Document Type | Dropdown displays with below values.   * MOU * Privacy Policy * Security Standards |
| 2.2.2 | User selects file | System validates (PDF, ≤ 10MB) |
| 2.2.3 | On success | Status → Pending Review |
| 2.2.4 | ScholarPath Admin reviews | Status → Active |
| 2.2.5 | Version increases | Previous version archived |

1. **School Documents Flow**

**3.1 User selects Schools tab**

System displays list of district schools:

* Polar Middle School
* Washington Middle School
* Lincoln Elementary
* …

Each school shows:

* Number of missing docs
* Last updated date
* Folder structure

**3.2 User opens a school**

System shows category folders, for example:

* Data Sharing Agreement
* Directory Information
* Annual Notifications
* Other

Each folder contains upload controls and document status indicators.

**3.3 Deep-Linking from Dashboard**

Example alert:

“Privacy policy for Polar Middle School is missing.”

System behavior:

1. Opens Documentation & Compliance
2. Switches to Schools tab
3. Expands Polar Middle School
4. Scrolls to the missing category
5. Highlights area in red or shows empty upload dropzone

**3.4 Uploading a School Document**

Works the same as District-level upload:

* Upload → Pending Review
* Admin review → Active
* Expiry calculations identical
* Each re-upload increments version

**4. ScholarPath Tab**

Displays documents uploaded by ScholarPath Admin:

* State Data Sharing Agreement
* Master Templates
* FERPA Policy Framework

User can:

* View
* Download

Cannot:

* Replace
* Upload
* Delete

**. Audit Trail Tab**

Shows chronological log of:

* Uploads
* Replacements
* Deletions
* Admin approvals
* Rejections
* Expiry events

Columns:

* Document Name
* Document Type
* School / District
* Action
* User
* Date
* Version
* Status

**Negative Flow**

Scenario 1 — Invalid File Type

Error:

“Unsupported format. Only PDF files are allowed.”

Scenario 2 — Size Exceeds Limit

Error:

“File size exceeds 10MB.”

Scenario 3 — Delete Active Document

Error:

“Active documents cannot be deleted.”

Scenario 4 — Re-upload Not Allowed

Error:

“You can only replace documents that are expiring or expired.”

Scenario 5 — Admin Rejects File

Status changes to Rejected

Tooltip shows rejection reason.

**Post-Conditions**

* District & school documents are updated and stored securely.
* Verification status updated after admin approval.
* Expiry notifications triggered as needed.
* Dashboard alerts clear after approval.
* Audit trail entries created for every action.

**Special Requirements**

* All versions and actions stored permanently for FERPA compliance.
* Version history shown per document (expandable panel).
* Deep-linking must work from Dashboard → Compliance screen.
* All uploads scanned for malware.

**Constraints**

* Only District Tech Lead can upload/delete documents.
* ScholarPath Admin must approve before document becomes Active.
* File size & format strictly enforced.
* Network drop triggers “Retry upload” modal.

**Screen Element Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Function / Use** | **API / Dependency** |
| Tabs | Navigation | Switch between District / Schools / ScholarPath / Audit Trail | UI only |
| Document Card | Section | Shows document name, status, expiry | GET /compliance/documents |
| Upload Button | CTA | Upload or replace document | POST /upload |
| Delete Button | CTA | Delete allowed document | DELETE /documents/{id} |
| School Folder | Expandable section | Shows school categories & docs | GET /schools/{id}/documents |
| Status Badge | Label | Active / Pending / Expired etc | System computed |
| Audit Trail Table | Data table | Shows events | GET /compliance/audit |

**Screenshots**



Figure : Compliance Documents - ScholarPath

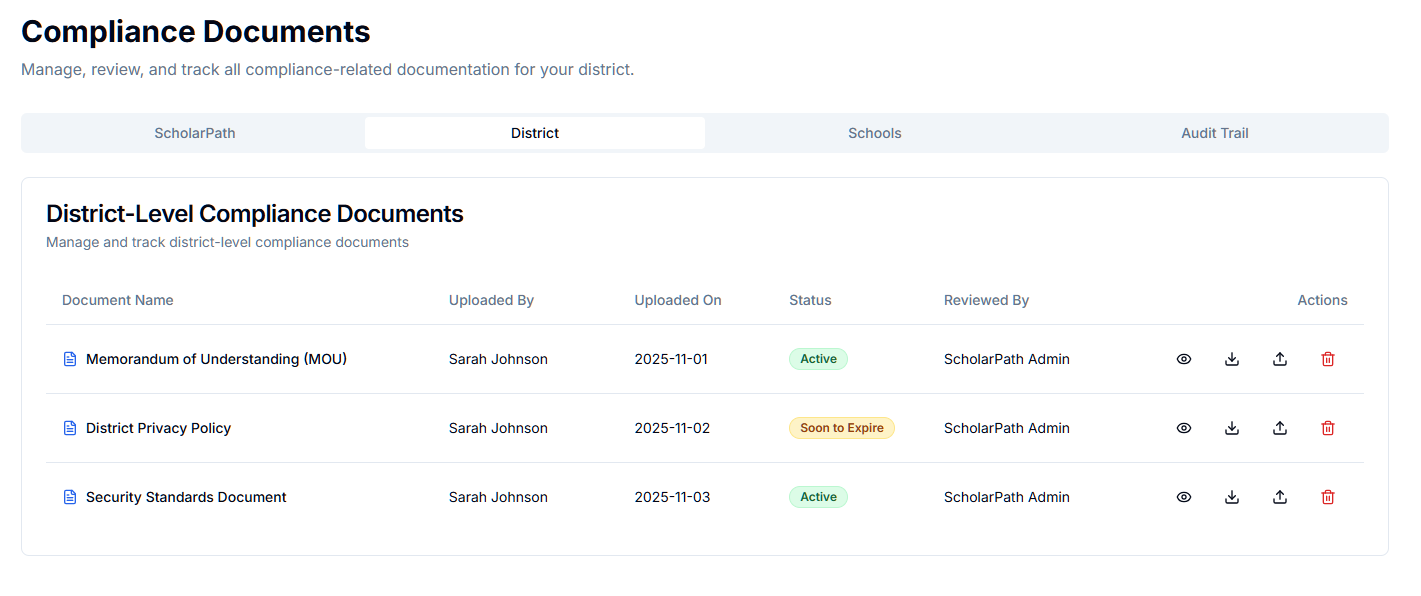


Figure : Compliance Documents – District

A screenshot of a document

AI-generated content may be incorrect.

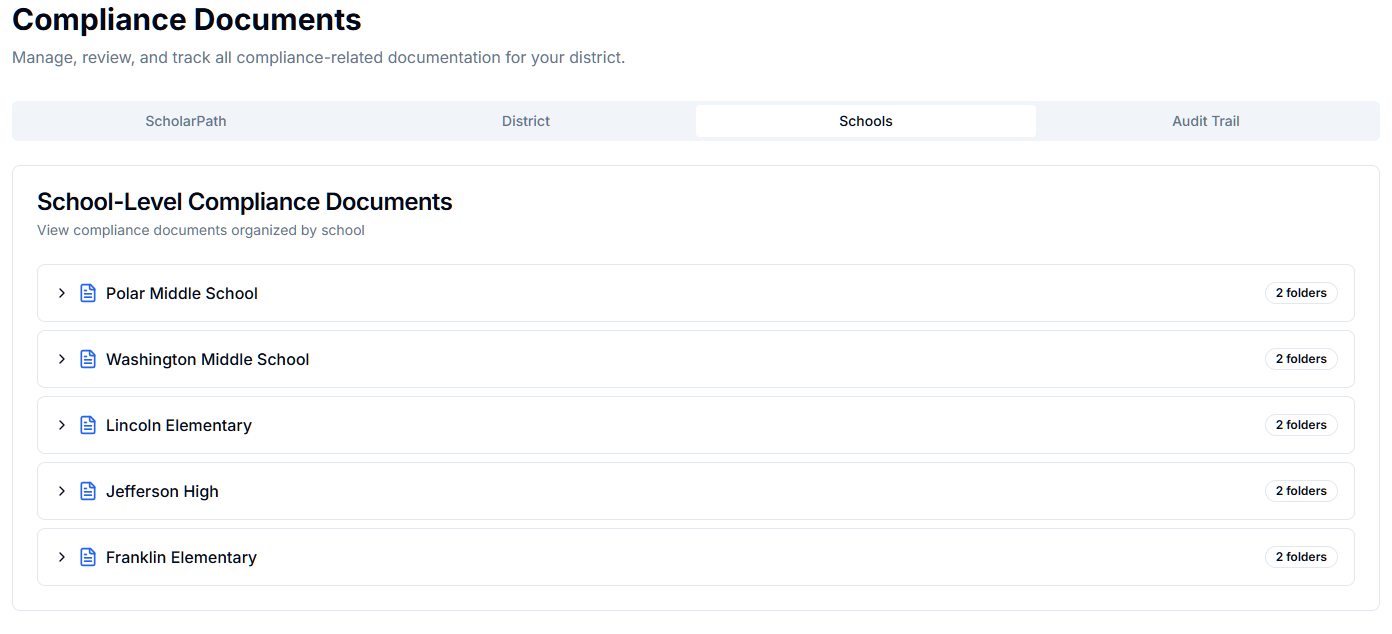


Figure : Compliance Documents - Schools

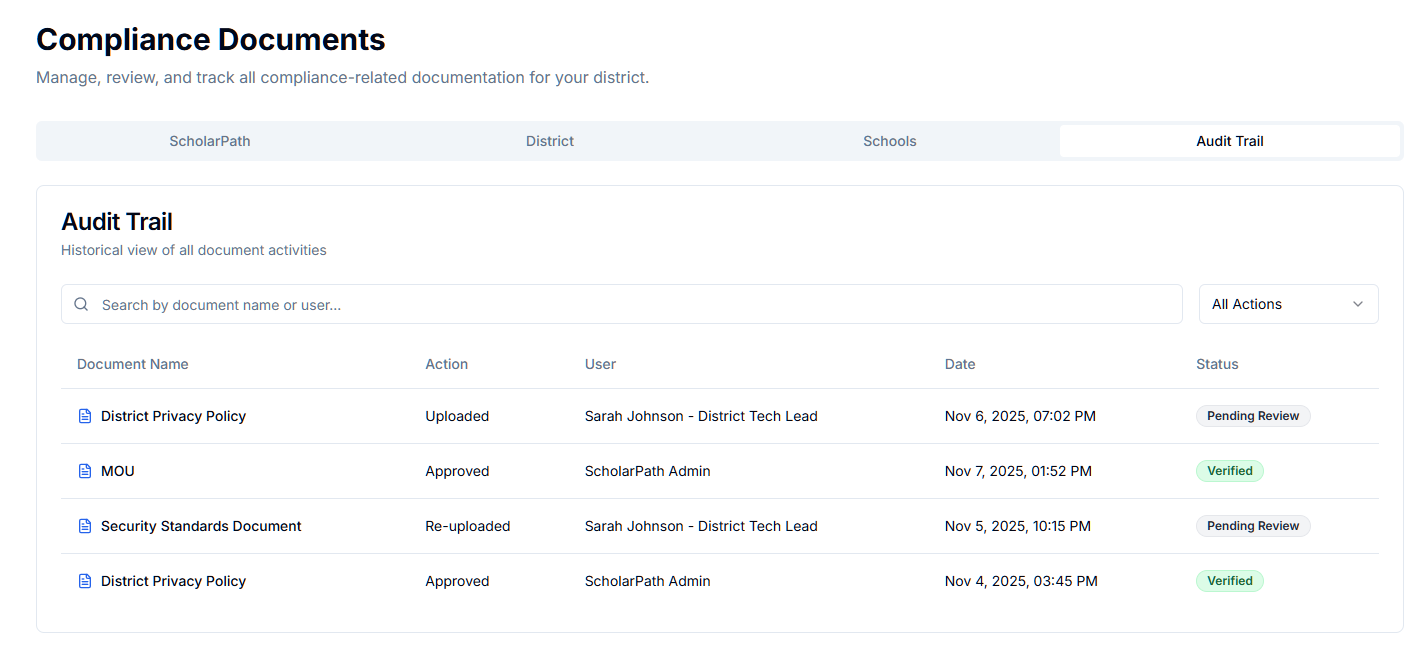


Figure : Compliance Documents - Audit Trail

**Involved APIs (Updated)**

District Document APIs

|  |  |  |
| --- | --- | --- |
| **API Name** | **Endpoint** | **Description** |
| Get District Documents | GET /api/districts/{id}/documents | Retrieves all district-level docs |
| Upload Document | POST /api/districts/{id}/documents | Upload/replace document |
| Delete Document | DELETE /api/districts/{id}/documents/{docId} | Delete allowed doc |
| Approve Document | POST /api/admin/documents/{docId}/approve | Admin verification |
| Reject Document | POST /api/admin/documents/{docId}/reject | Mark as rejected |

School Document APIs

|  |  |  |
| --- | --- | --- |
| **API** | **Endpoint** | **Description** |
| Get School Documents | GET /api/schools/{schoolId}/documents | Get all school-level docs |
| Upload School Doc | POST /api/schools/{schoolId}/documents | Upload/replace |
| Delete School Doc | DELETE /api/schools/{schoolId}/documents/{docId} | Delete |
| Missing Docs Report | GET /api/districts/{id}/compliance/missing | Drives dashboard alerts |

Audit Trail

|  |  |
| --- | --- |
| **API** | **Endpoint** |
| GET Audit Log | GET /api/districts/{id}/compliance/audit |

# Use Case 5: Configure and Maintain Data Integrations

**Description**

This use case defines how the District Tech Lead configures and maintains ScholarPath’s SIS data ingestion using either:

1. Manual CSV Upload, or
2. Nightly Batch Upload via FTP/SFTP

The workflows include configuring the data source, testing FTP/SFTP connectivity, selecting the integration method, viewing configuration status, and managing sync settings. It also describes conditional availability of Run Sync Now, and how the system handles updates to the integration type.

**Actors**

* **Primary Actor:** District Tech Lead
* **Supporting Actors:** ScholarPath Admin (backend monitoring, support)
* **System:** ScholarPath Integration Engine

**Goal**

To configure and maintain SIS data integration so ScholarPath can import student, staff, school, and academic datasets using the selected method.

**Trigger**

District Tech Lead navigates to:

Data Integrations → Configure Data Source

**Business Rules**

|  |  |
| --- | --- |
| **Rule ID** | **Description** |
| BR-01 | Only one active integration source is allowed at a time (Manual CSV or FTP/SFTP). |
| BR-02 | Manual CSV Upload does *not* support Run Sync Now. |
| BR-03 | FTP/SFTP supports Run Sync Now only when a *new, fully uploaded dataset* is detected. |
| BR-04 | District must upload all 8 required files under Manual CSV Upload to execute sync. |
| BR-05 | FTP/SFTP credentials must successfully pass a Test Connection before saving. |
| BR-06 | Integration type switches require confirmation; switching resets incomplete or pending sync jobs. |
| BR-07 | For FTP/SFTP, missing files in nightly batch must trigger alerts and appear in Recent Sync Errors. |
| BR-08 | Integration configuration persists until manually updated. |
| BR-09 | District Tech Lead can retry only batch-uploaded failed files (not manual uploads). |
| BR-10 | Running a sync does not pause the system unless the user explicitly chooses Pause System elsewhere. |

**Pre-Conditions**

* District onboarding completed
* User has integration permissions
* District has not disabled integrations
* If FTP/SFTP is selected, credentials must be valid and verified using Test Connection

**Steps**

Step 1 — Open Data Integrations Screen

1. User navigates to Data Integrations.
2. System loads:
   * Integration Type Selector
   * Configure Data Source panel
   * Data Sync Settings
   * Recent Sync Errors
   * Data Sync Logs

Step 2 — Select Integration Method

The Integration Type card shows two radio options:

Option A — Manual CSV Upload

Option B — Nightly Batch Upload (FTP/SFTP)

User selects one. System updates UI panels accordingly.

Option A — Manual CSV Upload

Step 3A — Upload Required Files

System displays 8 required files:

1. courses
2. district\_schools
3. parent\_information
4. student\_historic
5. student\_information
6. student\_schedule
7. teacher\_information
8. teacher\_schedule

For each:

* Download Template
* Upload
* Status badge: Uploaded / Missing

User uploads all required files.

System validates:

* Correct file format (CSV)
* Required headers exist
* File not empty

If all 8 files uploaded → Enable Process Sync button.

Step 4A — Process Manual Sync

1. User clicks Process Sync.
2. System:
   * Reads & validates all files
   * Performs row-level validation
   * Stores results
   * Generates sync batch
3. Errors (if any) appear in Recent Sync Errors.

Note: Manual CSV does *not* support Run Sync Now.

Option B — Nightly Batch Upload via FTP/SFTP

Step 3B — Configure FTP/SFTP Credentials

Fields shown:

* Protocol (FTP or SFTP)
* Host
* Port
* Username
* Password or SSH Key
* Remote Directory Path
* (Optional) File Naming Pattern

User enters credentials.

Step 4B — Test Connection

User clicks Test Connection.

System verifies:

* Authentication
* Directory exists
* Permissions
* Connectivity
* Response time

If successful → show green success banner: “Connection Successful.”

If failure → show error and block continuation.

Step 5B — Configure Sync Settings

Fields:

* Sync Time (default 2:00 AM)
* Time Zone
* File Missing Behavior:
  + Retry 3 times
  + Mark Failed
  + Skip
* Auto Process Files toggle
* Archive Files After Processing toggle

System saves settings.

Step 6 — Conditional “Run Sync Now”

This button follows Option 2 logic:

|  |  |
| --- | --- |
| Condition | Run Sync Now |
| New file detected on FTP/SFTP | Enabled |
| No new dataset available | Disabled (tooltip shown) |
| Manual CSV selected | Hidden entirely |

On Click → Confirmation Modal “A new dataset was detected. Run sync now instead of waiting for nightly job?”

Actions:

* Run Now
* Cancel

System processes the file immediately.

Step 7 — View & Use Recent Sync Errors

System displays file-level results:

Columns:

* File Name
* Status (Success / Partial / Failed / Missing)
* Total Records
* Failed Records
* Error Summary
* Actions (Retry File Sync, View Errors)

Actions:

View Errors

* Opens detailed modal
* Shows row-level errors
* Export allowed

Retry File Sync

Available only if:

* Integration = FTP Batch
* Status = Failed or Partial

System reprocesses only the failed file.

Step 8 — Data Sync Logs

User may click: View Details, opens modal showing:

* Batch summary
* Status of all 8 files
* Errors
* Processing duration
* Retry options

View All

Shows paginated history of all past syncs.

Export Logs

Downloads entire batch details.

**Negative Flow**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| Invalid CSV headers | Error modal: “Required header missing.” |
| FTP Authentication Failed | “Unable to authenticate. Check username/password.” |
| Directory Missing | “Folder not found on remote server.” |
| Run Sync Now clicked without new file | Button disabled + tooltip |
| Retry File clicked for manual upload | Not allowed; button hidden |
| Upload missing file in manual mode | Sync blocked until all 8 files provided |
| File naming pattern mismatch | Appears as Missing in Sync Errors |

**Post-Conditions**

* Integration method is saved.
* Sync logs updated.
* Recent Sync Errors displayed.
* New nightly sync schedule activated.
* Dashboard displays updated sync status.

**Special Requirements**

* All credentials must be encrypted.
* File headers must match ScholarPath templates exactly.
* Run Sync Now follows “Option 2” logic strictly.
* Error logs must maintain CSV row references.
* School admins cannot access this screen.

**Constraints**

* Only District Tech Lead can modify integration settings.
* Only FTP/SFTP supports conditional Run Sync Now.
* Only FTP/SFTP files can be retried.
* Maximum file size for manual CSV = 20MB.
* Retry fails if previous dataset archived.

**Screen Element Matrix (SEM)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Function** | **Dependency** |
| Integration Type Selector | Radio | Choose Manual vs FTP/SFTP | None |
| FTP Fields | Inputs | Host, Port, Directory | Shown only if FTP selected |
| Test Connection Button | Button | Validates FTP credentials | FTP server |
| CSV Upload Rows | File Input | Upload 8 files | Manual mode only |
| Process Sync | Button | Processes manual files | All files uploaded |
| Run Sync Now | Button | Trigger immediate sync | New file exists on FTP |
| Sync Settings | Form | Save schedule & rules | FTP selected |
| Recent Sync Errors | Table | Show file-level errors | Last batch |
| Retry File | Button | Retry single file | FTP + failed status |
| Data Sync Logs | Table | Show past batch logs | Integration engine |

**Screenshots**

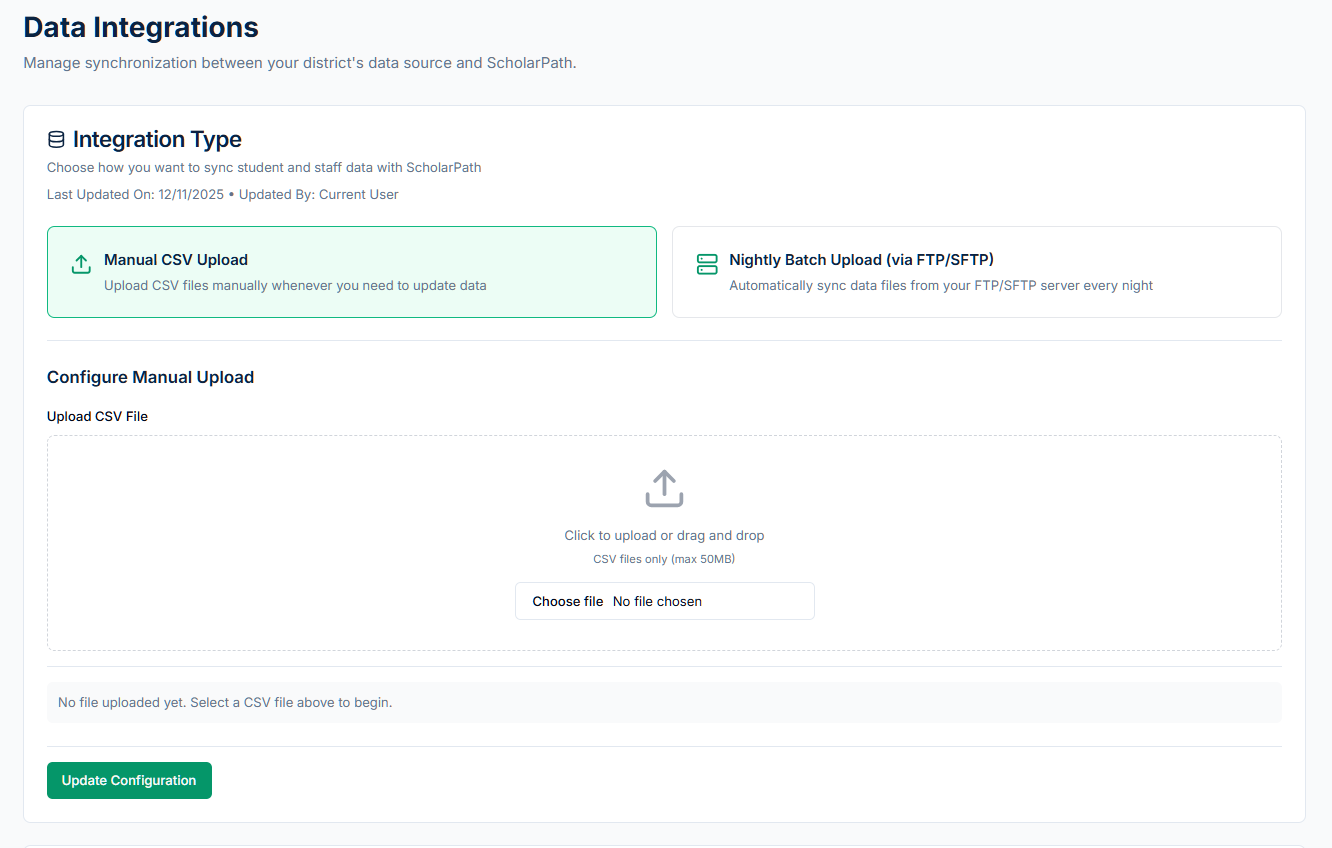


Figure : Data Integrations

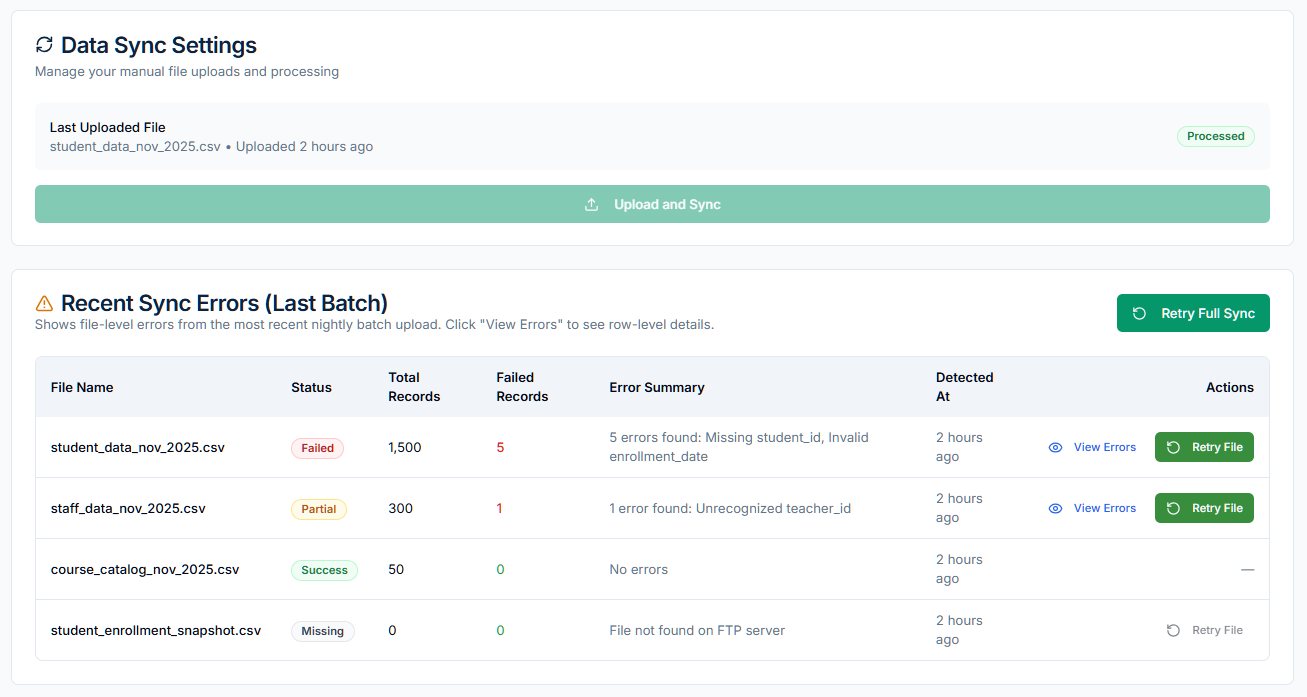


Figure : Data Sync Settings

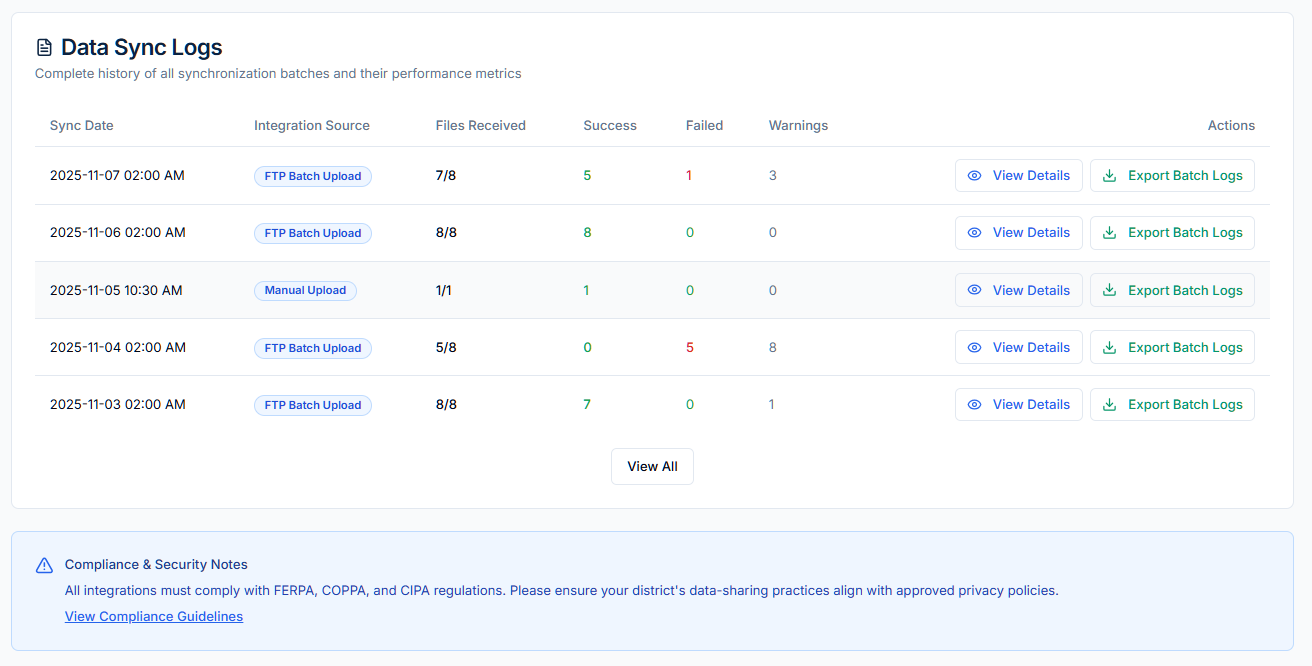


Figure : Data Sync Logs

**APIs Involved**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When Called** |
| Get Integration Config | GET /integration/config | Load saved method + settings | On screen load |
| Save Integration Config | POST /integration/config | Save selected integration type | On Save |
| Test FTP Connection | POST /integration/ftp/test | Validates credentials | Test Connection |
| Upload Manual File | POST /integration/manual/upload | Upload CSV file | On file upload |
| Process Manual Sync | POST /integration/manual/process | Executes manual sync | Process Sync |
| Detect FTP Dataset | GET /integration/ftp/latest-file | Checks for new dataset | Before showing Run Sync Now |
| Run Sync Now | POST /integration/ftp/run-now | Starts immediate sync | Run Sync Now button |
| Retry File Sync | POST /integration/ftp/retry-file | Retry failed file | Retry File |
| Get Recent Sync Errors | GET /integration/sync/errors/recent | Load file-level errors | Sync Errors modal |
| Get Sync Logs | GET /integration/sync/logs | Load batch logs | Data Sync Logs |
| Export Logs | GET /integration/sync/export | Export batch logs | Export button |

# Use Case 6: Review and Analyze Data Sync Logs and Errors

**Description**

This use case explains how the District Tech Lead reviews, investigates, and resolves issues related to SIS data ingestion.

It covers:

* Viewing the Recent Sync Errors panel (latest batch)
* Opening file-level error details
* Opening batch-level sync logs
* Using Retry File Sync for failed FTP files
* Navigating via deep links from:
  + Dashboard (“View Error Logs”)
  + Notifications (“Open Source”)
* Understanding missing file alerts
* Exporting error logs

This use case ensures reliable SIS data ingestion using either Manual CSV Upload or Nightly Batch FTP/SFTP.

**Actors**

* **Primary Actor:** District Tech Lead
* **Supporting Actors:** District Tech Team, ScholarPath Admin
* **System:** ScholarPath Integration Engine

**Goal**

To allow the District Tech Lead to identify, analyze, and resolve SIS sync errors at both file and batch levels.

**Trigger**

Triggered when user:

* Clicks View Error Logs on the Dashboard
* Opens Data Integrations → Recent Sync Errors
* Clicks View Details inside Data Sync Logs
* Clicks Open Source from Notifications

**Business Rules**

|  |  |
| --- | --- |
| **Rule ID** | **Description** |
| BR-01 | Each nightly batch consists of 8 required SIS files. |
| BR-02 | Errors must be shown at file-level, not just batch-level. |
| BR-03 | Only FTP/SFTP files support Retry File Sync. |
| BR-04 | Manual CSV uploads do NOT support Retry File Sync. |
| BR-05 | Missing files appear as Status = Missing in Recent Sync Errors. |
| BR-06 | File-level error modal must support filtering, search, and CSV export. |
| BR-07 | Batch logs must show all 8 files regardless of missing or failed. |
| BR-08 | Deep-linking should always open the correct file & batch. |
| BR-09 | Sync errors persist until resolved or next successful sync. |
| BR-10 | Export format must match district Excel template. |

**Pre-Conditions**

* District Tech Lead has Integration permissions.
* At least one sync has occurred.
* Recent Sync Errors exist OR batch history exists.
* FTP path is processed or manual CSV uploaded.

**Steps**

Step 1 — Open Sync Error Logs

User can enter via:

A. Dashboard → “View Error Logs”

Shows Latest Sync Errors modal directly.

B. Data Integrations → Recent Sync Errors section

Opens default errors list.

C. Notifications → “Open Source”

Automatically deep-links to the exact file error modal.

D. Data Integrations → Data Sync Logs → “View Details”

Opens batch-level detail modal.

Step 2 — Recent Sync Errors Modal

The modal displays the latest batch’s 8 files:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **File Name** | **Status** | **Total Records** | **Failed Records** | **Error Summary** | **Actions** |
| student\_information.csv | Partial | 1254 | 42 | Missing school\_id | View Errors / Retry (FTP only) |
| teacher\_schedule.csv | Missing | 0 | 0 | File not found | — |
| courses.csv | Success | 423 | 0 | — | — |

Statuses allowed:

* Success
* Partial
* Failed
* Missing

User Actions:

A. View Errors

Opens file-level detailed modal.

1. Retry File Sync (FTP only)

Enabled only for failed/partial FTP files.

Not shown for:

* manual uploads
* successful files
* missing files

C. View Full Logs

Navigates to Data Sync Logs → opens batch log.

Step 3 — File-Level Error Detail Modal

Modal content includes:

Table Columns:

* Row Number
* Field Name
* Error Type
* Error Message
* Original Value
* Suggested Fix
* (Optional) Error Code

Toolbar:

* Filter by Field
* Filter by Error Type
* Search
* Pagination
* Export File Errors (CSV)

User Interactions:

* Fix file offline
* Retry File (FTP)
* View next/previous file in same batch

Step 4 — Data Sync Logs (Batch History)

Panel lists all previous sync batches:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date/Time** | **Source** | **Files Received** | **Success** | **Failed** | **Warnings** | **Actions** |
| Feb 20, 2025 - 2:14 AM | FTP Batch | 8/8 | 6 | 2 | 1 | View Details / Export |

User clicks:

A. View Details

* Opens batch detail modal:
* Summary cards
* Status breakdown for all 8 files
* Error summaries
* Retry File buttons (FTP only)

B. Export Batch Logs

Exports full dataset across all 8 files.

Step 5 — Retry File Sync (FTP Only)

Triggered from:

Recent Sync Errors

Batch Details modal

Flow:

User clicks Retry File Sync.

System confirms:

“Retrying will reprocess the previously downloaded file. Continue?”

System reprocesses only that file.

Updates results in:

* Recent Sync Errors
* Batch Logs
* Notifications
* Dashboard Alerts

Manual CSV files cannot be retried.

Step 6 — Deep Linking (Notifications → Data Integrations)

Example:

Notification:

“student\_information.csv failed due to missing school\_id.”

When clicking Open Source, system performs:

Navigate to Data Integrations

Auto-open the correct batch

Highlight the file

Auto-open file error modal

This eliminates manual searching in batch logs.

**Negative Flows**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| File not found in FTP | Status = Missing. Error explains missing file. |
| Retry File clicked for manual upload | Button not shown; no action. |
| Large error logs > 10k rows | System paginates automatically. |
| Batch logs older than retention period | “This log has been archived.” |
| User without permission opens logs | 403 Access Denied page. |

**Post-Conditions**

* All error details are visible and exportable.
* User understands exactly which files failed and why.
* File-level retry actions may resolve errors.
* Alerts clear after successful reprocessing.
* Logs are updated across the platform.

**Special Requirements**

* Must support 8-file architecture.
* Error logs must match district-required Excel template.
* Modal must support up to 50k error rows with pagination.
* Deep linking is mandatory from Dashboard + Notifications.
* Highlighting and auto-scrolling required.

**Constraints**

* Retry File only applies to FTP/SFTP uploads.
* Files must remain in storage archive to retry.
* Sync cannot be retried if previous dataset auto-deleted.
* Batch logs older than X days auto-archived (configurable).

**Screen Element Matrix (SEM)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Description** | **Dependency** |
| Recent Sync Errors | Modal | Shows last batch errors | Sync engine |
| View Errors | Button | Opens file-level detail | Error dataset |
| Retry File | Button | FTP-only retry | FTP archive |
| View Full Logs | Link | Opens batch logs | Sync logs |
| Batch Logs Table | Table | Shows all past batches | Logging system |
| Batch Details Modal | Modal | Shows file-level breakdown | Sync logs |
| Export Logs | Button | Exports full batch | Export service |
| Error Table | Table | Row-level errors | Error store |

**Screenshots**

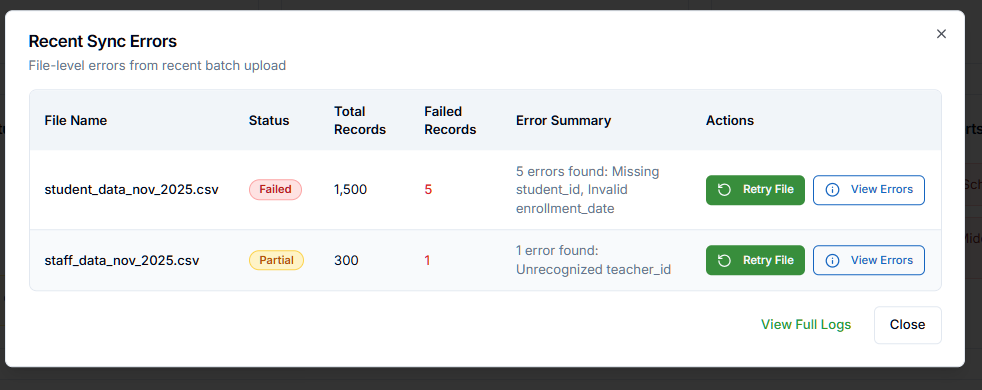


Figure : Dashboard - Recent Sync Errors

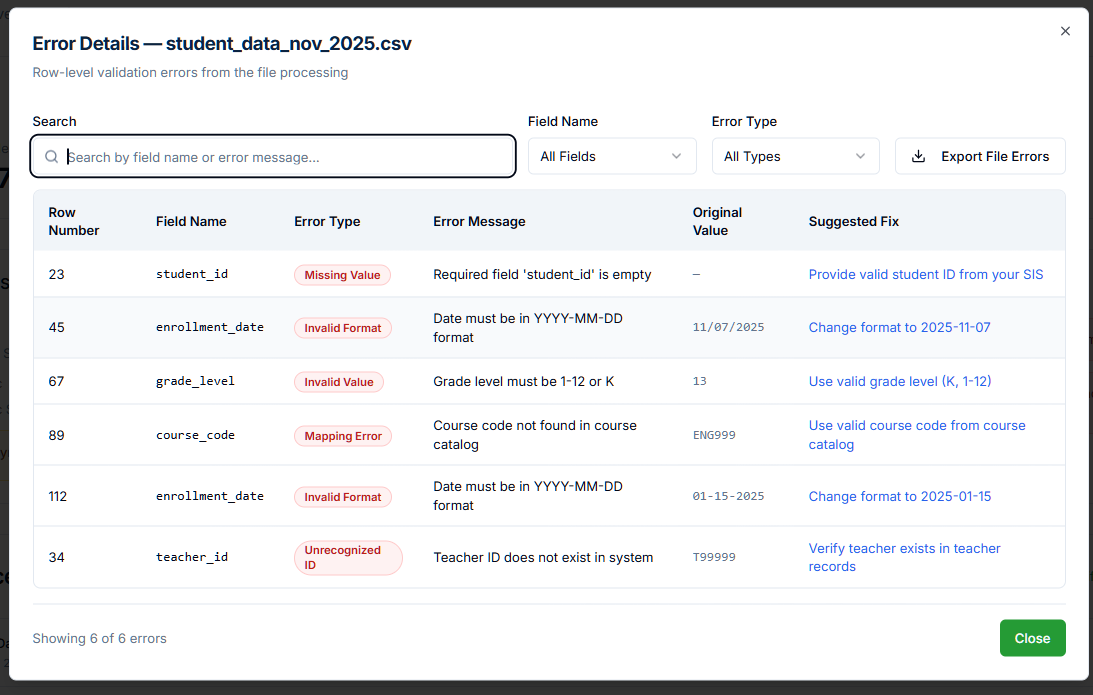


Figure : Error Details

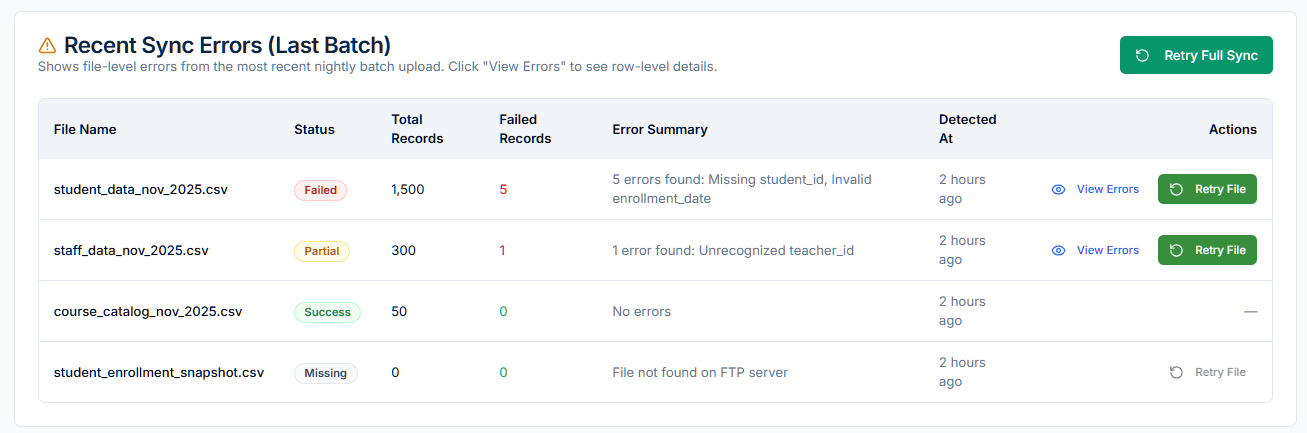


Figure : Data Integration - Recent Sync Errors

**Involved APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **Trigger** |
| Get Recent Errors | GET /integration/sync/errors/recent | Loads latest errors | Recent Sync Errors |
| Get File Error Details | GET /integration/sync/file-errors/{fileId} | Row-level errors | View Errors |
| Retry File Sync | POST /integration/ftp/retry-file | Retry only that file | Retry File |
| Get Batch Logs | GET /integration/sync/logs | Load batch history | Sync Logs |
| Get Batch Detail | GET /integration/sync/logs/{batchId} | Batch breakdown | View Details |
| Export File Errors | GET /integration/sync/file-errors/{fileId}/export | Export CSV | File error modal |
| Export Batch Logs | GET /integration/sync/logs/{batchId}/export | Export CSV | Export Batch Logs |
| Detect FTP Dataset | GET /integration/ftp/latest-file | Used for Run Sync Now | Pre-check |

# Use Case 7: Configure and Validate Single Sign-On (SSO)

**Description:**

This use case defines how the District Tech Lead sets up, tests, and maintains the district’s SSO configuration for secure user authentication into the ScholarPath platform.

It includes per-provider configuration (Google Workspace, Clever, ClassLink, Edlink, and Generic SAML 2.0), attribute and claim mapping, and connection testing.

The Goal: is to ensure that district users can authenticate seamlessly via their organization’s identity provider while maintaining compliance and security.

**Actors:**

* **Primary Actor:** District Tech Lead
* **Supporting Actor:** ScholarPath Admin (for validation and technical assistance)
* **System:** ScholarPath Platform

**Goal:**

To configure and validate Single Sign-On integration between ScholarPath and the district’s chosen authentication provider(s).

**Trigger:**

User navigates to the Security Center > SSO screen.

**Business Rules:**

|  |  |  |
| --- | --- | --- |
| **#** | **Rule** | **Description** |
| BR-01 | Provider Options | Supported SSO providers: Google Workspace, Clever, ClassLink, Edlink, Generic SAML 2.0. |
| BR-02 | Tab Separation | Each SSO provider appears under a separate configuration tab. |
| BR-03 | Field Validation | All required fields (Client ID, Secret, or Metadata URL) must be filled before saving. |
| BR-04 | Attribute Mapping | Mapping fields (email, first name, last name, role) must be defined for each provider. |
| BR-05 | Test Requirement | SSO configuration must be tested successfully before activation. |
| BR-06 | Credential Encryption | API credentials must be encrypted at rest. |
| BR-07 | One Provider per Type | Only one configuration per provider type is allowed. |
| BR-08 | Login Branding Disabled | Login branding customization is not available (ScholarPath uses a universal login page). |

**Pre-Conditions:**

* District account must be Active.
* SSO provider access credentials (Client ID, Secret, or Metadata URL) must be available.
* ScholarPath Admin must have provided redirect URIs.

**Steps:**

|  |  |  |
| --- | --- | --- |
| **Step No.** | **Action** | **System Behavior** |
| 1 | User navigates to SSO & Authentication. | System loads provider tabs: Google Workspace, Clever, ClassLink, Edlink, SAML 2.0. |
| 2 | User clicks on a provider tab (e.g., Google Workspace). | The configuration form for that provider appears. |
| 3 | User fills in required fields (e.g., Client ID, Client Secret, Domain). | System validates mandatory fields and field format. |
| 4 | User clicks Test Connection. | System runs a live connection test and displays the result (Success / Failed). |
| 5 | If test succeeds, user clicks Save Configuration. | System saves configuration securely and updates provider status to “Active.” |
| 6 | User scrolls to Attributes & Claims Mapping section within the same tab. | System loads default mapping fields for email, first name, last name, and role. |
| 7 | User customizes mappings as per district’s identity schema. | System validates field presence and format. |
| 8 | User clicks Test Mapping. | System validates mapping using a mock SSO payload and displays sample results. |
| 9 | User clicks Save Mapping. | Mapping settings are stored for that provider. |
| 10 | User clicks Reset Defaults (optional). | Mapping reverts to ScholarPath’s default claim structure. |
| 11 | User repeats setup for any other provider (optional). | System enforces one active configuration per provider type. |
| 12 | User reviews Monitoring & Alerts metrics. | Displays SSO health, connection uptime, and authentication trends. |

**Negative Flow:**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| Invalid credentials entered. | “⚠️ Invalid Client ID or Secret. Please recheck your credentials.” |
| Metadata URL not reachable (SAML). | “⚠️ Could not fetch metadata. Please verify the URL.” |
| Missing required fields. | “⚠️ All required fields must be completed.” |
| Test connection fails. | “❌ SSO Test Failed. Check redirect URI or credentials.” |
| Mapping contains missing claim. | “⚠️ Claim ‘email’ not found in provider payload.” |
| Save fails due to permission issue. | “❌ Unable to save configuration. Contact ScholarPath Admin.” |

**Post-Condition:**

* SSO configuration is successfully established and saved for selected providers.
* Default and custom mappings are validated and stored.
* Connection status is “Active” in the provider tab.
* Users can authenticate using the configured provider.

**Special Requirements:**

* Each tab retains unsaved changes warning on switch.
* Success feedback includes green banner: “✅ Connection established successfully.”
* Failed connection results include red banner with provider-specific error.
* Each SSO test displays latency and response time.
* All configurations are versioned and stored securely.
* SSO Monitoring widget shows metrics for active connections and last login.

**Constraints:**

* Only one active SSO per provider type (Google, Clever, etc.).
* All provider secrets are encrypted using AES-256.
* Redirect URIs must match domain whitelisting rules.
* Test Connection and Save must occur sequentially (cannot save untested config).
* Mapping must be validated before users can log in using the provider.

**Screen Element Matrix:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Function / Use** | **Source / Dependency** |
| Provider Tabs | Tab Navigation | Separates configuration forms per provider. | System UI |
| Client ID / Secret Fields | Input Fields | Capture authentication credentials. | Provider API |
| Domain Field | Input Field | Defines district domain for SSO. | User Input |
| Redirect URI | Read-Only Text | Displays pre-defined ScholarPath callback URL. | System Preconfigured |
| Test Connection Button | Action | Tests provider connection. | /api/sso/{provider}/test |
| Save Configuration Button | Action | Saves validated credentials. | /api/sso/{provider}/config |
| Attributes & Claims Mapping Table | Table | Allows editing of claim mappings. | /api/sso/{provider}/claims/mapping |
| Save Mapping Button | Action | Saves claim configuration. | /api/sso/{provider}/claims/map |
| Test Mapping Button | Action | Tests mappings with sample payload. | /api/sso/{provider}/claims/test |
| Reset Defaults Button | Action | Resets mapping to default. | /api/sso/{provider}/claims/reset |
| Status Badge | Label | Shows provider connection status (Active, Inactive, Failed). | System Computed |
| Monitoring & Alerts Card | Card | Displays authentication health metrics. | /api/sso/monitoring |

**Screenshot:**

*A screenshot of a computer

AI-generated content may be incorrect.*

Figure : SSO & Authentication section

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Google SSO Provider

A screenshot of a computer

AI-generated content may be incorrect.

Figure : SSO Provider – Clever

A screenshot of a computer

AI-generated content may be incorrect.

Figure : SSO Provider – ClassLink

A screenshot of a computer

AI-generated content may be incorrect.

Figure : SSO Provider - Edlink

**Involved APIs:**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When is it Called?** |
| Get SSO Configuration | /api/districts/{districtId}/sso/config | Fetches saved configurations for all providers. | On screen load. |
| Test SSO Connection | /api/sso/{provider}/test | Tests authentication for selected provider. | On “Test Connection” click. |
| Save SSO Configuration | /api/sso/{provider}/config | Saves configuration details for selected provider. | On “Save Configuration.” |
| Fetch Claims Mapping | /api/sso/{provider}/claims/mapping | Retrieves current claim mapping. | On tab load. |
| Save Claims Mapping | /api/sso/{provider}/claims/map | Saves mapping fields for provider. | On “Save Mapping.” |
| Test Mapping | /api/sso/{provider}/claims/test | Tests mapping with sample payload. | On “Test Mapping.” |
| Reset Mapping | /api/sso/{provider}/claims/reset | Resets mapping to ScholarPath default. | On “Reset Defaults.” |
| Fetch Monitoring Stats | /api/districts/{districtId}/sso/monitoring | Fetches uptime and authentication trend metrics. | On Monitoring section load. |

# Use Case 8: Configure and Enforce Security Policies

**Description**

This use case defines how the District Tech Lead configures district-level security, system governance, and automated operational controls.

It includes:

* Authentication & security policy configuration
* District system operational controls (pause/resume)
* Data retention & automatic log archival
* Automated email notifications (Daily Sync Summary, Weekly Audit Exports)
* FTP health monitoring

All actions are logged for audit compliance.

**Actors**

**Primary Actor:** District Tech Lead

**Supporting Actor:** ScholarPath Admin (platform-level oversight)

**Goal**

To configure and enforce all district-level authentication, security, retention, and automated system settings.

**Triggers**

* User navigates to:

Security & System Control → Security Policies / System Preferences

* System sends automated emails daily or weekly (based on settings)

**Business Rules**

|  |  |  |
| --- | --- | --- |
| # | **Rule** | **Description** |
| BR-01 | Policy Hierarchy | ScholarPath global security policies override district policies where needed. |
| BR-02 | Audit Logging Required | All changes to security and system configuration must be logged. |
| BR-03 | Pause Constraints | Pausing the system prevents new logins but keeps active sessions intact. |
| BR-04 | Retention Enforcement | System deletes or archives logs based on district’s retention selection. |
| BR-05 | Automation Reliability | Daily & weekly automation emails must only trigger when setting is ON. |
| BR-06 | FTP Monitoring Scope | Only applies when Nightly Batch Sync is enabled. |
| BR-07 | Permission Restriction | Only District Tech Leads may modify system-level controls. |
| BR-08 | Email Delivery Logic | If a report is empty, “No issues found” must still be emailed. |

**Pre-Conditions**

* District account is active.
* Tech Lead has system/security management permissions.
* Initial onboarding is completed.
* Nightly FTP Sync (if applicable) is configured.

**Post-Conditions**

* All security & system settings updated and stored.
* Logs are archived or retained based on preferences.
* Automated emails begin/stop based on toggles.
* Any modifications appear in audit log.

**Steps**

1. Security Policies

|  |  |  |
| --- | --- | --- |
| **Step** | **User Action** | **System Response** |
| 1 | User opens Security Policies | System loads: Password Rules, MFA, Timeout, Geo Restrictions |
| 2 | User toggles a policy (e.g., “MFA Required”) | UI updates, system validates |
| 3 | User clicks Save Changes | System updates configuration + logs event |
| 4 | Settings applied immediately | All future logins follow new rules |

1. System State Management

|  |  |  |
| --- | --- | --- |
| **Step** | **User Action** | **System Response** |
| 1 | User opens System State Management | Shows current status (Active / Paused) |
| 2 | User clicks Pause System | Modal asks for maintenance message |
| 3 | User confirms | New logins blocked; active sessions remain |
| 4 | User resumes system later | Normal activity restored |

1. Data Retention Preferences

|  |  |  |
| --- | --- | --- |
| **Step** | **User Action** | **System Behavior** |
| 1 | User selects “Retention Period” (1–7 years) | System displays confirmation |
| 2 | User saves | System updates retention policy |
| 3 | System enforces deletion or archival automatically | Based on the Auto-Archive toggle |

1. Automation Settings

Includes:

* Daily Sync Summary Email
* Auto-Export Weekly Audit Logs
* FTP Health Monitoring

|  |  |  |
| --- | --- | --- |
| **Step** | **User Action** | **System Response** |
| 1 | User toggles an automation feature | UI visually updates |
| 2 | System validates | Starts/stops automation scheduler |
| 3 | Daily or weekly job runs | Reports delivered via email |
| 4 | Issues during automation | Logged in audit + retry scheduled |

**Negative Flow**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| FTP unreachable | Email sent: “FTP Server Unreachable” |
| Daily Summary disabled | No daily sync email sent |
| Weekly Audit Export fails | System retries + sends failure notification |
| Invalid retention selection | Error: “Invalid retention configuration” |
| User without permission attempts update | “Access denied” |

**Special Requirements**

* All emails use ScholarPath branding.
* Automated email dispatch uses retry logic.
* UTC timestamps must be displayed in audit logs.
* Sensitive logs must be encrypted at rest.
* Data retention timeline must be visible in tooltips.

**Screen Element Matrix (SEM)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Function** | **API** |
| Password Rules | Toggle/Link | Opens modal | /api/security/policies |
| MFA Enforcement | Toggle | Require MFA | /api/security/policies |
| Session Timeout | Input | Sets inactivity timeout | /api/security/policies |
| Pause System | Button | Pauses district logins | /api/system/state/pause |
| Retention Dropdown | Dropdown | Set retention period | /api/system/preferences |
| Auto-Archive Logs | Toggle | Archive logs after period | /api/system/preferences |
| Daily Sync Summary | Toggle | Enables daily summary | /api/system/automation/syncsummary |
| Weekly Audit Export | Toggle | Export weekly audit logs | /api/system/automation/auditexport |
| FTP Monitoring | Toggle | Monitor FTP health | /api/system/automation/ftpmonitor |
| Save Preferences | Button | Saves all system preferences | /api/system/preferences/save |

**Email Templates**

A. Daily Sync Summary Email Template

**Subject:** Daily SIS Sync Summary — {{DistrictName}}

**Body:**

Hello {{UserName}},

Here is your daily SIS sync summary for {{Date}}.

Sync Overview

* Total files processed: {{TotalFiles}}
* Successful files: {{SuccessCount}}
* Files with errors: {{ErrorCount}}
* Next scheduled sync: {{NextSyncTime}}

File-Level Status

{{#Each Files}}

* {{FileName}} — {{Status}}

{{#If Errors}}

Errors:

{{#Each Errors}}

* + {{this}}

{{/Each}}

{{/If}}

{{/Each}}

If all files succeeded:

No issues detected today. Great job!

If errors occurred:

⚠ Some files require attention.

View error logs: {{ErrorLogsURL}}

Thanks,

ScholarPath Automated Sync Service

B. Auto-Export Weekly Audit Logs Email Template

**Subject:** Weekly Audit Log Export — {{DistrictName}}

**Body:**

Hello {{UserName}},

Your audit logs for {{WeekStart}} – {{WeekEnd}} have been exported.

Export Summary

* Total audit events: {{EventCount}}
* File size: {{FileSize}}
* Format: CSV
* Retention Policy: {{RetentionYears}} years

You can download your audit logs here:

👉 {{DownloadURL}}

If your district has auto-archive enabled, older logs have been archived automatically.

Thanks,

ScholarPath Security & Compliance Service

**Screenshots:**



Figure 35: Notification & Settings > System section

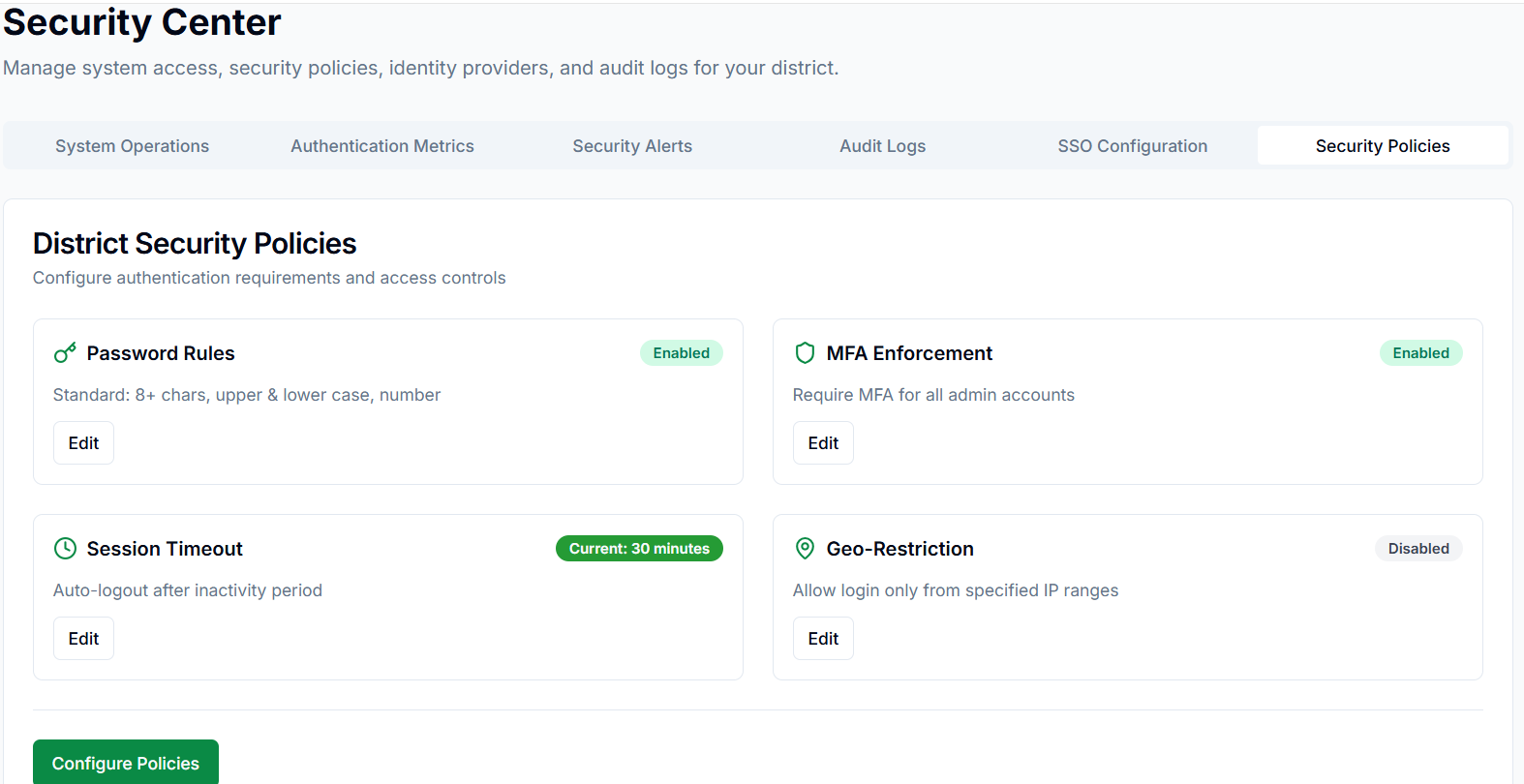


Figure 36: Security Center > Security Policies section

**Involved APIs**

(Security Policies + System State + System Preferences + Automation Settings)

1. Security Policies APIs

|  |  |  |  |
| --- | --- | --- | --- |
| **Area** | **API Endpoint** | **Method** | **Description** |
| Password Rules | /api/districts/{districtId}/security/password-rules | GET / PUT | Fetch/update password requirements |
| MFA Enforcement | /api/districts/{districtId}/security/mfa | GET / PUT | Enable/disable MFA enforcement |
| Session Timeout | /api/districts/{districtId}/security/session-timeout | GET / PUT | Configure inactivity timeout |
| Geo Restriction | /api/districts/{districtId}/security/geo-restriction | GET / PUT | Allow/block IP ranges |
| Fetch All Policies | /api/districts/{districtId}/security/policies | GET | Get all security settings in one payload |

2. System State Management APIs

|  |  |  |
| --- | --- | --- |
| **API Endpoint** | **Method** | **Description** |
| /api/districts/{districtId}/system/state | GET | Returns current system state: Active / Paused |
| /api/districts/{districtId}/system/pause | POST | Pauses new logins; requires maintenance message |
| /api/districts/{districtId}/system/resume | POST | Restores normal login operations |
| /api/districts/{districtId}/system/state/log | GET | Fetches pause/resume event history |

3. System Preferences (Retention & Archiving)

|  |  |  |  |
| --- | --- | --- | --- |
| **Area** | **API Endpoint** | **Method** | **Description** |
| Data Retention Policy | /api/districts/{districtId}/system/retention | GET / PUT | Set retention period (1–7 years) |
| Auto-Archive Logs | /api/districts/{districtId}/system/auto-archive | GET / PUT | Enable auto-archiving after retention |
| Fetch All Preferences | /api/districts/{districtId}/system/preferences | GET | Get retention + automation settings |

4. Automation Settings APIs

A. Daily Sync Summary Email

|  |  |  |
| --- | --- | --- |
| **API Endpoint** | **Method** | **Description** |
| /api/districts/{districtId}/automation/daily-sync-summary | GET / PUT | Toggle ON/OFF for daily summary email |
| /api/automation/daily-sync/run | POST | Internal — executes daily summary job |
| /api/automation/daily-sync/email-preview | GET | Returns preview of next summary email |

B. Auto-Export Weekly Audit Logs

|  |  |  |
| --- | --- | --- |
| **API Endpoint** | **Method** | **Description** |
| /api/districts/{districtId}/automation/audit-export | GET / PUT | Enable/disable weekly audit exports |
| /api/automation/audit-export/run | POST | Internal — runs weekly export |
| /api/automation/audit-export/files | GET | Lists past exports |
| /api/automation/audit-export/{fileId}/download | GET | Downloads the exported file |

C. FTP Health Monitoring

|  |  |  |
| --- | --- | --- |
| **API Endpoint** | **Method** | **Description** |
| /api/districts/{districtId}/automation/ftp-monitoring | GET / PUT | Enable/disable FTP monitoring |
| /api/automation/ftp-monitor/heartbeat | GET | Returns latest FTP health check result |
| /api/automation/ftp-monitor/run | POST | Internal — executes heartbeat test |
| /api/automation/ftp-monitor/logs | GET | Returns FTP downtime/error logs |

5. Audit Logging APIs

*(Every settings change must trigger audit entries)*

|  |  |  |
| --- | --- | --- |
| **API Endpoint** | **Method** | **Description** |
| /api/districts/{districtId}/audit/system | GET | Fetch audit logs related to system & security changes |
| /api/districts/{districtId}/audit/system/export | GET | Export audit history (CSV) |
| /api/districts/{districtId}/audit/system/{eventId} | GET | View details of a specific configuration event |

# Use Case 9: Monitor Authentication Health and Access Logs

**Description**

This use case defines how the District Tech Lead monitors authentication health, suspicious login activity, MFA adoption, failed login attempts, and historical access events for all district-level user accounts.

The user views high-level login metrics on the Dashboard and Security Center, then can drill down into detailed event logs via modals or the Access Monitoring table inside the Audit Logs tab.

This supports security posture monitoring, forensic analysis, and compliance reporting.

**Goal**

To enable the District Tech Lead to view and analyze authentication activity, suspicious events, failed logins, MFA metrics, and detailed access logs for all users in the district.

**Actors**

* **Primary Actor:** District Tech Lead
* **System:** ScholarPath Authentication Engine & Security Center

**Trigger**

User navigates to:

* Dashboard → Security Snapshot, OR
* Security Center → Authentication Metrics tab, OR
* Security Center → Audit Logs tab

Clicking any dashboard metric opens the corresponding modal or navigates directly to filtered audit logs.

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **#** | **Rule** | **Description** |
| **BR-01** | Real-Time Metrics | Security Snapshot metrics auto-refresh every 15 minutes. |
| **BR-02** | Filtered Drilling | Clicking any metric shows filtered details (e.g., failed logins in last 7 days). |
| **BR-03** | Access Monitoring | Real-time events appear in Access Monitoring table with 30-second auto-refresh. |
| **BR-04** | Role Scope | District Tech Lead sees all district users. |
| **BR-05** | Suspicious Login Detection | Based on IP mismatch, device fingerprint changes, or login velocity. |
| **BR-06** | MFA Exclusion | Students under age 13 excluded from MFA metrics. |
| **BR-07** | Audit Logging | Viewing or exporting logs does not modify log data. |
| **BR-08** | Export History | Exported logs must follow a fixed CSV audit format. |

**Pre-Conditions**

* Authentication and SSO systems must be active.
* District Tech Lead must have permission to access Security Center.
* System must have generated at least one login event since activation.

**Steps**

1. View Summary Metrics (Dashboard → Security Snapshot)

1. User sees metrics:
   * Failed Logins (24h)
   * Suspicious Activity (7d)
   * MFA Adoption (%)
   * Total Logins (7d)
2. User clicks a metric.
3. System opens:
   * Failed Logins → *Failed Login Attempts Modal*
   * Suspicious Logins → *Suspicious Login Activity Modal*
   * MFA Adoption → *MFA Breakdown Modal*
   * Total Logins → *Authentication Activity Modal*

2. Authentication Metrics (Security Center → Authentication Metrics Tab)

1. **User navigates to Authentication Metrics tab.**
2. System displays:
   * Total Logins
   * Failed Logins
   * MFA Adoption
   * Suspicious Activity
3. All metrics are clickable.
4. **Clicking a metric opens the same modals as Dashboard.**

3. Access Monitoring (Security Center → Audit Logs Tab)

1. User navigates to Audit Logs tab.
2. System displays a real-time “Recent Access Events” table:
   * Event Type
   * User
   * IP Address
   * Time
   * Status
3. Table auto-refreshes every 30 seconds.
4. User can:
   * Filter table by User, Event Type, IP Address, Status
   * Search by keyword
   * Export filtered logs
   * Click row → “Event Detail Modal”

4. Full Audit Log Review

1. Below Access Monitoring, the user sees a full audit log table.
2. User applies filters:
   * Days
   * Event type (login, MFA failure, suspicious attempt, Login success, login failed, configuration change)
   * Search
3. User clicks View Details on an event.
   * Modal shows:
   * Event ID
   * Timestamp
   * Event Type
   * Severity
   * Status (Open/Resolved/Flagged)
   * User Email (or “System”)
   * IP Address + Country/Region (if available)
   * Provider (Google / SAML / EntraID, if applicable)
   * Anomaly Type (Geo shift / multiple failed logins / MFA fail, etc.)
   * Raw metadata (collapsed JSON section)
   * Actions:
   * User exports results (CSV).

**Negative Flow**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| No login data available | Shows “No authentication events available for this period.” |
| Suspicious login detection service down | “Unable to verify risk indicators for login events.” |
| Export failure | “Export failed — please try again.” |
| Invalid filters | System resets filter and shows info message. |

**Post-Conditions**

* Tech Lead has full visibility into all district-level authentication activity.
* Suspicious login attempts are easily accessible and actionable.
* All log exports and drill-down events are preserved for compliance.
* System auto-refreshes keep the view up to date.

**Special Requirements**

* Modals must support paging, filters, and CSV export.
* MFA modal displays breakdown by user type.
* Suspicious login modal includes map view (optional).
* Highlight repeated failed logins (5+ attempts in 10 minutes).
* Timezone formatting based on district timezone.

**Screen Element Matrix (SEM)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Use** | **API** |
| Security Snapshot | Card | Shows summary metrics | /api/security/metrics/snapshot |
| Failed Logins Modal | Modal | Shows failed login attempts | /api/security/logs?event=failed |
| Suspicious Login Modal | Modal | Shows flagged risky events | /api/security/logs?event=suspicious |
| MFA Breakdown Modal | Modal | MFA stats | /api/security/mfa/stats |
| Access Monitoring Table | Table | Real-time login events | /api/security/logs/live |
| Audit Logs Table | Table | Historical system events | /api/security/audit |
| Export Logs Button | Action | CSV export | /api/security/audit/export |
| Filters (date, event, user) | UI elements | Log filtering | Client + API |

**Screenshots**

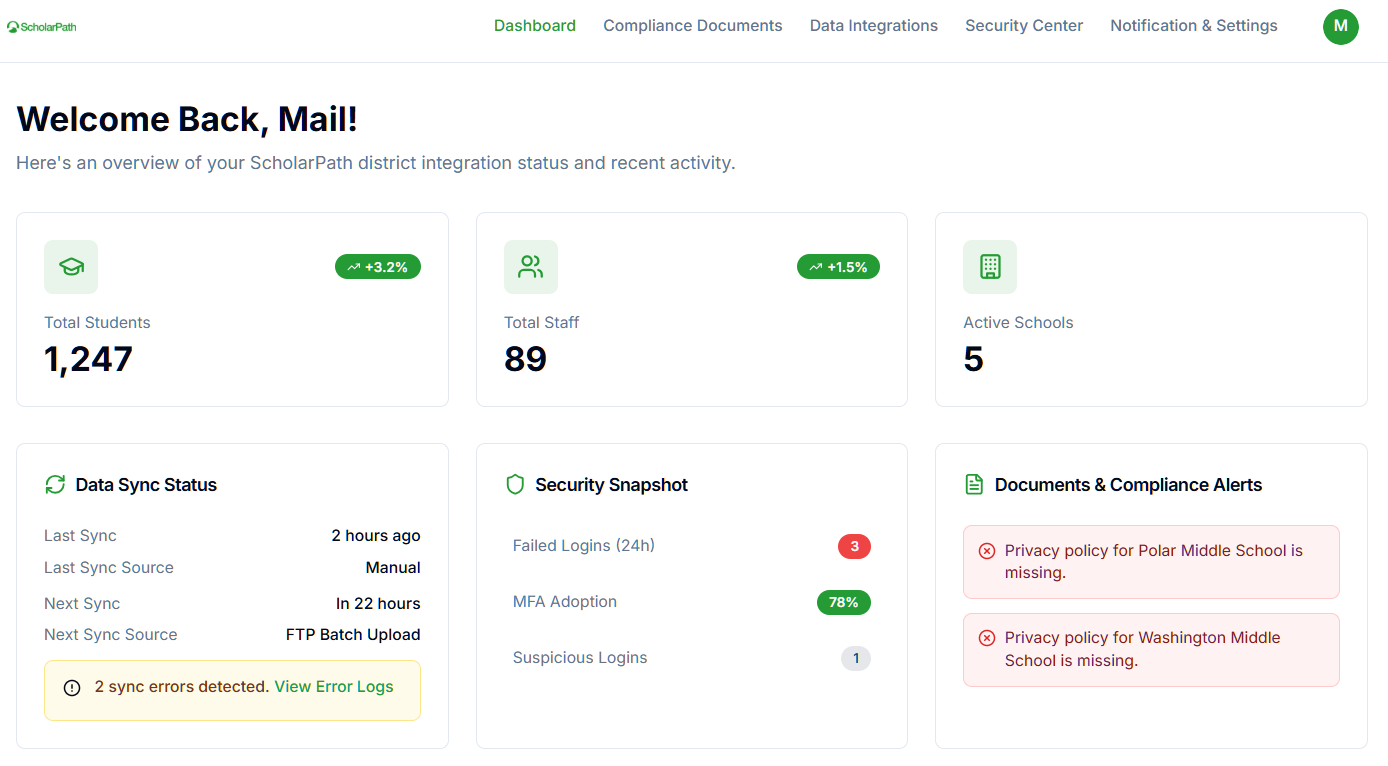


Figure : Dashboard - Security Snapshot

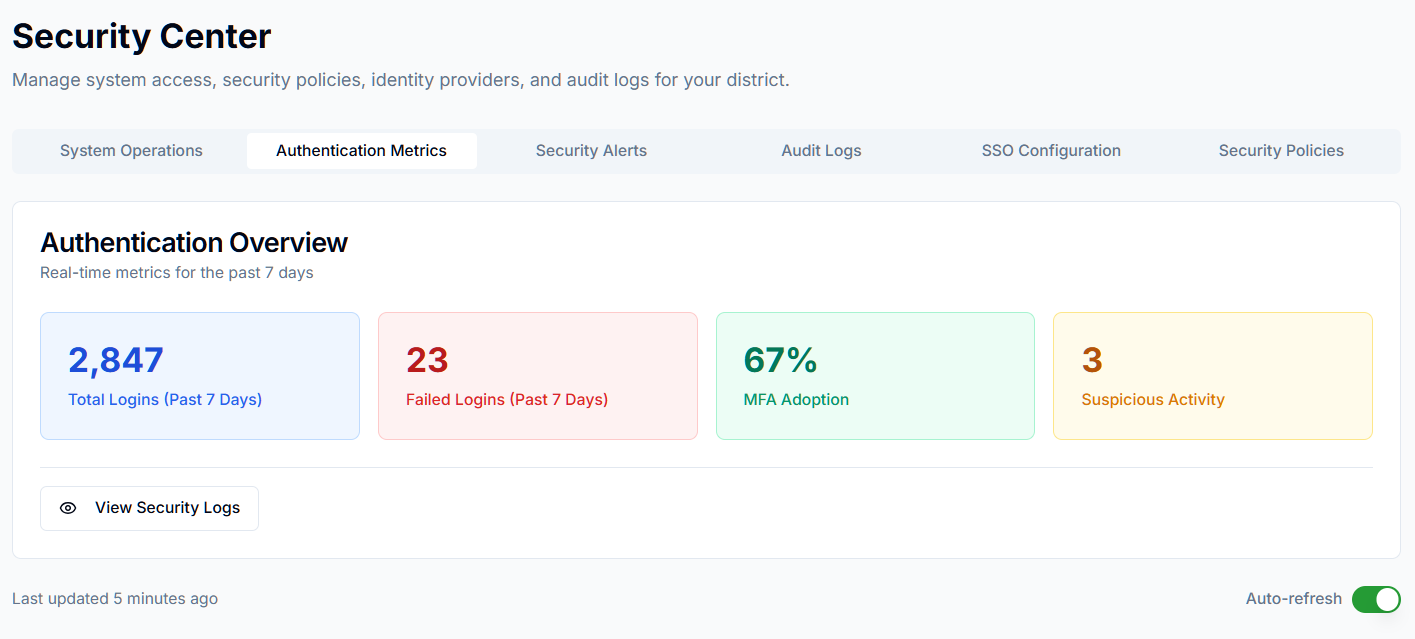


Figure : Authentication Metrics

**Involved APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When Called** |
| Security Snapshot | /api/security/metrics/snapshot | Returns dashboard authentication metrics | On dashboard load |
| Failed Login Events | /api/security/logs?event=failed | Fetch failed login attempts | On metric click |
| Suspicious Login Events | /api/security/logs?event=suspicious | Fetch suspicious/risky attempts | On metric click |
| MFA Stats | /api/security/mfa/stats | Computes MFA adoption breakdown | On MFA click |
| Live Access Logs | /api/security/logs/live | Returns latest login events | Audit Logs tab |
| Audit Logs | /api/security/audit | Shows system-wide event history | Audit Logs tab |
| Export Audit Logs | /api/security/audit/export | CSV export | On Export click |

# Use Case 10: Configure Alerts, Notifications, and Delivery Preferences

**Description**

This use case defines how the District Tech Lead configures how they receive system alerts, compliance warnings, data sync notifications, and security-related messages.

It includes:

* Managing notification delivery preferences
* Enabling/disabling email alerts
* Configuring categories of system notifications
* Viewing category-level notification rules
* Managing unread notifications
* Using “Open Source” deep-linking
* Marking and resolving notification items
* Understanding alert severity (Info, Warning, Critical)

All notifications are scoped only to the district and cannot be modified globally.

**Actors**

* **Primary Actor:** District Tech Lead
* **Supporting Actor:** ScholarPath Admin (creates system-wide rules)

**Goal**

To allow the District Tech Lead to configure, personalize, and manage how system alerts and notifications are delivered and displayed.

**Trigger**

User navigates to: Notification & Settings → All Notifications

OR

Clicks View All Notifications on the Dashboard.

**Business Rules**

|  |  |
| --- | --- |
| **Rule ID** | **Description** |
| BR-01 | Email Delivery is optional but defaults ON for Critical Alerts. |
| BR-02 | District Tech Lead can enable/disable notification categories locally. |
| BR-03 | Critical Alerts (Security, Data Sync Failures) cannot be fully disabled. Email is always ON. |
| BR-04 | “Resolve” actions only mark the notification as resolved; they do not delete it. |
| BR-05 | “Open Source” deep-links to the relevant module (Compliance, Data Integrations, Security). |
| BR-06 | Only District-scoped notifications are visible (no cross-district alerts). |
| BR-07 | Each notification is timestamped and stored for at least 180 days. |
| BR-08 | Changing delivery preferences does not affect past notifications. |
| BR-09 | Marking a notification as Read/Resolved updates the unread count across all screens. |
| BR-10 | System-generated notifications follow immutable templates; users cannot edit text. |

**Pre-Conditions**

* User has completed onboarding.
* Notifications engine is active.
* At least one notification category is enabled.
* District Tech Lead hsas permission to change personal delivery settings.

**Steps**

Step 1 — Open Notifications & Settings

User navigates to the Notifications screen.

Tabs shown:

1. All Notifications
2. System Alerts
3. Compliance Alerts
4. Data Sync Alerts
5. Security Alerts
6. Email Delivery Settings

Each tab displays a notification list filtered by category.

Step 2 — Configure Delivery Preferences

Inside Email Delivery Settings, the user sees toggles for:

* Data Sync Failures
* Compliance Document Expiry
* Security Alerts (Critical Only – enforced ON)
* System Alerts
* All Other Notifications

Actions:

1. User toggles any category ON/OFF.
2. System validates:
   * Critical alerts email toggle cannot be disabled.
3. User clicks Save Preferences.
4. Success toast appears.

Step 3 — View All Notifications (Unified List)

User clicks All Notifications tab.

System displays a combined timeline with:

* Title
* Category tag
* Timestamp
* Severity badge (Info / Warning / Critical)
* Actions:
  + Open Source
  + Resolve
  + Mark as Read

Items are sorted newest → oldest.

Step 4 — Open Source Deep-Link

User clicks **Open Source** on a notification.

Examples:

* Data Sync Failure → Data Integrations → File Error modal
* Missing School Document → Compliance → Schools tab → specific folder
* Security Event → Security & Controls → Access Logs filtered by event

System automatically:

* Navigates to correct screen
* Pre-filters data
* Auto-opens modal (if applicable)

Step 5 — Mark as Read

User clicks Mark as Read:

* Item is flagged as read
* Unread counter on Dashboard & top bar decreases
* Item remains in list (state = Read)

Step 6 — Resolve Notification

User clicks **Resolve**:

* System marks notification state = Resolved
* No deletion occurs
* Dashboard alert (if any) clears if all related items resolved
* Audit Trail logs the action:

“Notification Resolved —  —  — ”

Step 7 — Search & Filter Notifications

Filters available:

* Date Range
* Category
* Severity
* Status (Unread, Resolved)
* Keyword Search

Results update dynamically.

Step 8 — Export Notifications (Optional)

User clicks **Export** button.

System downloads:

District\_Notifications\_<timestamp>.csv

Including:

* Title
* Category
* Severity
* Source Module
* Timestamp
* State (Read/Resolved)

**Step 9 – Notifications categories and their preferences (System Alerts – Tab Name)**

**Category Names**

1. SIS Data Sync Alerts
2. FTP/SFTP Health Alerts
3. Security Alerts
4. System
5. Compliance Alerts
6. Delivery Channels

**Notification Message Formats for Every Notification Type**

These are the final message templates for ALL notification types shown in your “System Alerts & Notification Preferences” screen — including SIS Sync, File Errors, FTP Health, System, Compliance, and Security.

**Each template follows the structure:**

* Title
* Message Body
* Metadata included
* Example Notification (as shown in All Notifications tab)

**1. SIS Data Sync Alert**

**1.1 Nightly Sync Success**

Title: Nightly Sync Completed Successfully

Message: All SIS files were processed successfully during the nightly sync.

Metadata: timestamp, sync\_id

Example:

“Nightly sync completed successfully at 02:15 AM.”

**1.2 Nightly Sync Failed**

Title: Nightly Sync Failed

Message: The nightly sync failed. No files were processed. Please review error logs.

Metadata: timestamp, sync\_id, failure\_reason

Example:

“Nightly sync failed at 02:10 AM — no files processed.”

**1.3 Nightly Sync Completed With Errors**

Title: Nightly Sync Completed With Errors

Message: The nightly sync completed, but some files contain errors. Review file-level logs.

Metadata: timestamp, sync\_id, list\_of\_failed\_files

Example:

“Nightly sync completed with errors — 3 files require attention.”

**2. File-Level Alerts (CSV Batch Files)**

 These alerts appear when a specific file has errors.

**2.1 courses — File Errors**

Title: Courses File Errors

Message: The courses.csv file contains errors. Review detailed logs for failed records.

Example:

“courses.csv — 12 records failed validation.”

**2.2 district\_schools — File Errors**

Title: School File Errors

Message: The district\_schools.csv file contains errors.

Example:

“district\_schools.csv — Missing School ID in 5 rows.”

**2.3 parent\_information — File Errors**

 Title: Parent Information File Errors

Message: The parent\_information.csv file contains validation issues.

Example:

“parent\_information.csv — Invalid email format in 3 rows.”

**2.4 student\_historic — File Errors**

 Title: Student Historic File Errors

Message: The student\_historic.csv file has rows with missing or invalid data.

Example:

“student\_historic.csv — 2 failed records (missing student\_id).”

**2.5 student\_information — File Errors**

 Title: Student Information File Errors

Message: Errors detected in student\_information.csv.

Example:

“student\_information.csv — Invalid grade\_level in 4 rows.”

**2.6 student\_schedule — File Errors**

 Title: Student Schedule File Errors

Message: The student\_schedule.csv file contains errors.

Example:

“student\_schedule.csv — Duplicate class assignments found.”

**2.7 teacher\_information — File Errors**

Title: Teacher Information File Errors

Message: Errors found in teacher\_information.csv.

Example:

“teacher\_information.csv — Unrecognized school\_id in 6 rows.”

**2.8 teacher\_schedule — File Errors**

 Title: Teacher Schedule File Errors

Message: The teacher\_schedule.csv file has issues.

Example:

“teacher\_schedule.csv — Invalid teacher\_id in 3 rows.”

**3. FTP / SFTP Health Alerts**

**3.1 FTP Health — Connection Healthy**

Title: FTP Connection Healthy

Message: FTP/SFTP connection to your SIS source is functioning normally.

Example:

“FTP connection verified successfully.”

**3.2 FTP Health — Connection Failed**

 Title: FTP Connection Failed

Message: Unable to connect to FTP/SFTP server. Nightly sync may be interrupted.

Example:

“FTP connection failed — cannot reach server at 10.21.3.6.”

**3.3 FTP Health — Connection Unstable**

 Title: FTP Connection Unstable

Message: FTP/SFTP connection is intermittent. Some uploads may fail.

Example:

“FTP connection unstable — high latency detected.”

**4. System Alerts (NEW category)**

**4.1 System Paused**

 Title: System Paused

Message: ScholarPath has been temporarily paused for maintenance.

Example:

“System paused — user logins temporarily disabled.”

**4.2 System Resumed**

 Title: System Resumed

Message: ScholarPath is active again. All services restored.

Example:

“System resumed — normal operation restored.”

**4.3 System Outage / Degradation**

 Title: System Outage / Degradation

Message: ScholarPath is experiencing degraded performance or downtime.

Example:

“System outage — some features may be unavailable.”

**4.4 System Recovery**

 Title: System Recovery

Message: ScholarPath has recovered from an outage or performance degradation.

Example:

“System recovery — services back to normal.”

**4.5 Auto-Archive Logs Completed**

 Title: Logs Auto-Archived

Message: Logs older than the retention period were archived successfully.

Example:

“Log auto-archive completed — 120 logs archived.”

**4.6 Auto-Archive Logs Failed**

 Title: Log Archiving Failed

Message: System was unable to archive logs.

Example:

“Auto-archive failed — storage permission error.”

**4.7 Logs Purged After Retention Policy**

 Title: Logs Purged

Message: Logs older than the retention period were purged.

Example:

“Log purge complete — 34 expired logs removed.”

**4.8 Team Member Invited**

 Title: New Team Member Invited

Message: A new District Tech Lead has been invited to your district account.

Example:

“Invitation sent to sarah.johnson@district.com.”

**4.9 Team Member Deactivated**

 Title: Team Member Deactivated

Message: A District Tech Lead’s access has been revoked.

Example:

“Account deactivated: mike.davis@district.com.”

**4.10 Team Member Reactivated**

 Title: Team Member Reactivated

Message: A District Tech Lead’s access has been restored.

Example:

“Account reactivated: emily.chen@district.com.”

**4.11 User Role Updated**

 Title: User Role Updated

Message: A team member’s role has been updated.

Example:

“Role updated for john.smith@district.com.”

**5. Compliance Alerts**

**5.1 Missing Required Document**

 Title: Missing Required Compliance Document

Message: A required compliance document has not been uploaded.

Example:

“Missing compliance document — Privacy Policy for Washington Middle School.”

**5.2 Document Expiring Soon**

 Title: Document Expiring Soon

Message: A compliance document will expire in less than 30 days.

Example:

“MOU expiring soon — 28 days remaining.”

**5.3 Document Expired**

Title: Document Expired

Message: A compliance document has expired and must be renewed.

Example:

“Privacy Policy expired — immediate action required.”

**5.4 Document Approved**

 Title: Document Approved

Message: Your uploaded compliance document has been reviewed and approved.

Example:

“MOU approved by ScholarPath Admin.”

**5.5 Document Rejected**

 Title: Document Rejected

Message: A compliance document was rejected during review.

Example:

“MOU rejected — incorrect signature format.”

**6. Security Alerts**

**6.1 Suspicious Login Detected**

Title: Suspicious Login Detected

Message: A login attempt matched suspicious or abnormal patterns.

Example:

“Suspicious login detected — jane.doe@student.com from China (Geo Shift).”

**6.2 Multiple Failed Logins**

 Title: Multiple Failed Login Attempts

Message: Multiple failed login attempts were detected for this user.

Example:

“Multiple failed logins — admin@district.com (5 attempts).”

**6.3 Failed MFA Attempt**

 Title: Failed MFA Attempt

Message: A user failed a required MFA challenge.

Example:

“Failed MFA — teacher@district.com from IP 192.168.1.44.”

**6.4 Geo-Bypass Attempt**

 Title: Geo-Bypass Attempt

Message: A login attempt originated outside the allowed region.

Example:

“Geo-Bypass Attempt — tech.lead@district.com from 89.45.23.67.”

**6.5 Suspicious IP Activity**

 Title: Suspicious IP Activity

Message: Login detected from a suspicious IP address.

Example:

“Suspicious IP — principal@district.com from 192.168.1.99.”

**Note:** Implement a scrollbar within the preferences section so users can scroll and enable or disable the required notification types as needed.

**Negative Flow**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| User disables Critical Alert email toggle | Error: “Critical alerts cannot be disabled.” |
| Email delivery service unreachable | Warning banner: “Email notifications may be delayed.” |
| User clicks Open Source but record no longer exists | Error modal: “Source data not available.” |
| Unauthorized user attempts to edit settings | “Access Denied — insufficient privileges.” |
| Export fails | “Export could not be generated. Please retry.” |

**Post-Conditions**

* Notification preferences saved.
* Dashboard unread count synced.
* Event logs updated in Audit Trail.
* Newly generated notifications follow updated settings.
* Deep links remain accurate.

**Special Requirements**

* ScholarPath UI theme:
  + Primary green (#0A8A45) for buttons
  + Orange (#F0AD4E) for warnings
  + Red (#D9534F) for critical
* All timestamps displayed relative (“2 hours ago”) + full datetime in tooltip.
* Pagination for > 50 notifications.
* “Resolve” should not hide or delete items.
* Multiselect filters must be preserved on navigation.

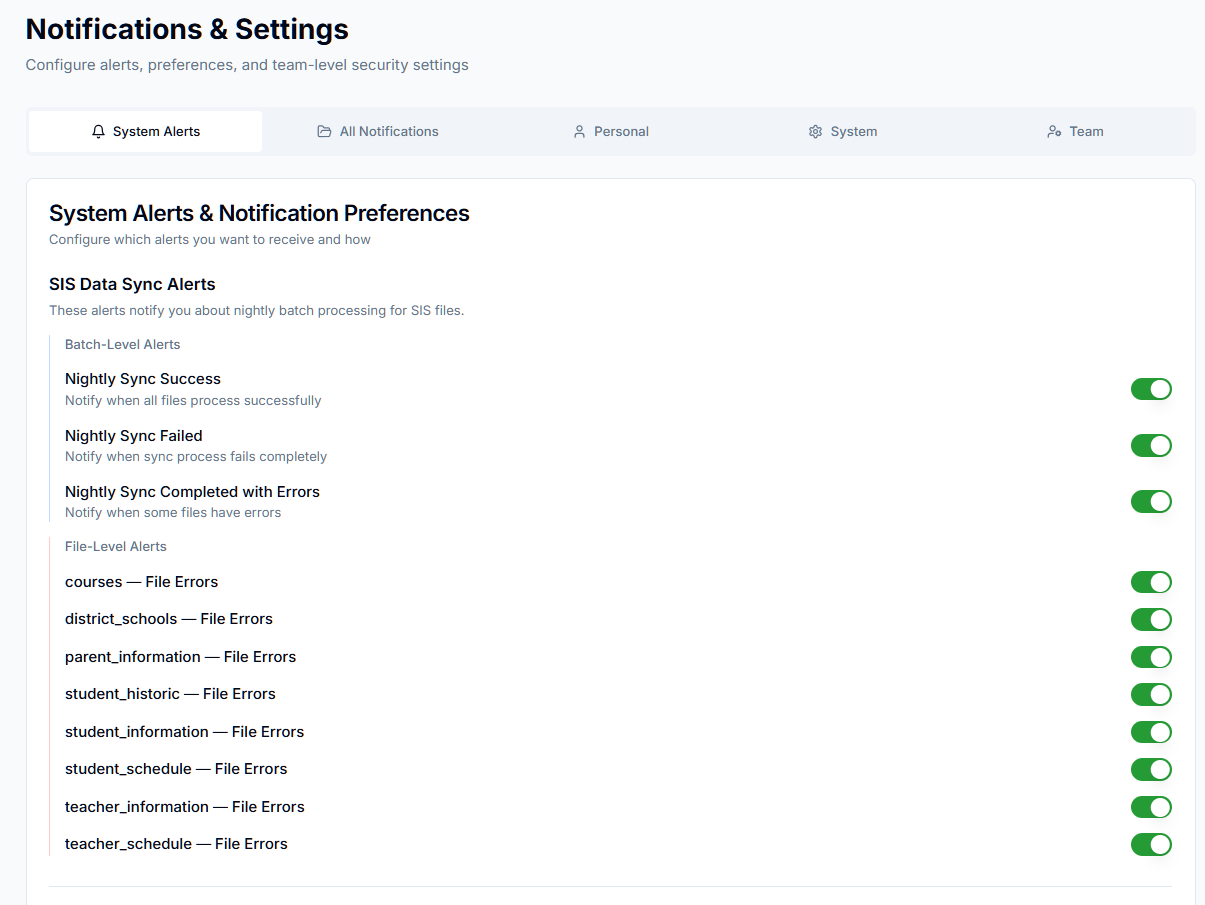
**Constraints**

* District Tech Lead sees only district-level alerts.
* Push notifications (mobile/app) are not supported.
* Logs retained for 180 days only.
* Email batching not supported (1 email per alert).
* Notification templates cannot be edited.

**Screen Element Matrix (SEM)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Purpose** | **Dependency** |
| All Notifications Tab | List | Unified timeline | /notifications/all |
| Resolve Button | Action | Mark notification as resolved | /notifications/{id}/resolve |
| Mark as Read | Action | Marks as read | /notifications/{id}/read |
| Open Source | Deep Link | Opens source module & record | Client-side routing |
| Email Toggles | Switch | Turn email alerts on/off | /notifications/settings/update |
| Category Filters | Dropdown | Filters list | Client side |
| Export Button | CTA | Export records | /notifications/export |
| Severity Badge | Label | Critical/Warning/Info | System metadata |

**Screenshot**



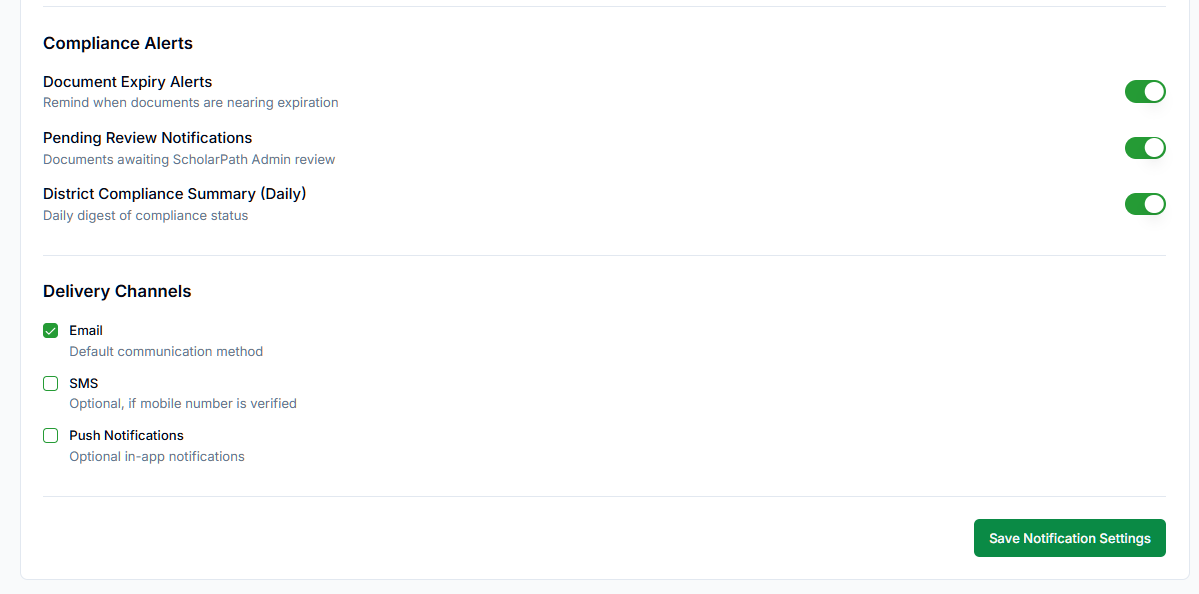
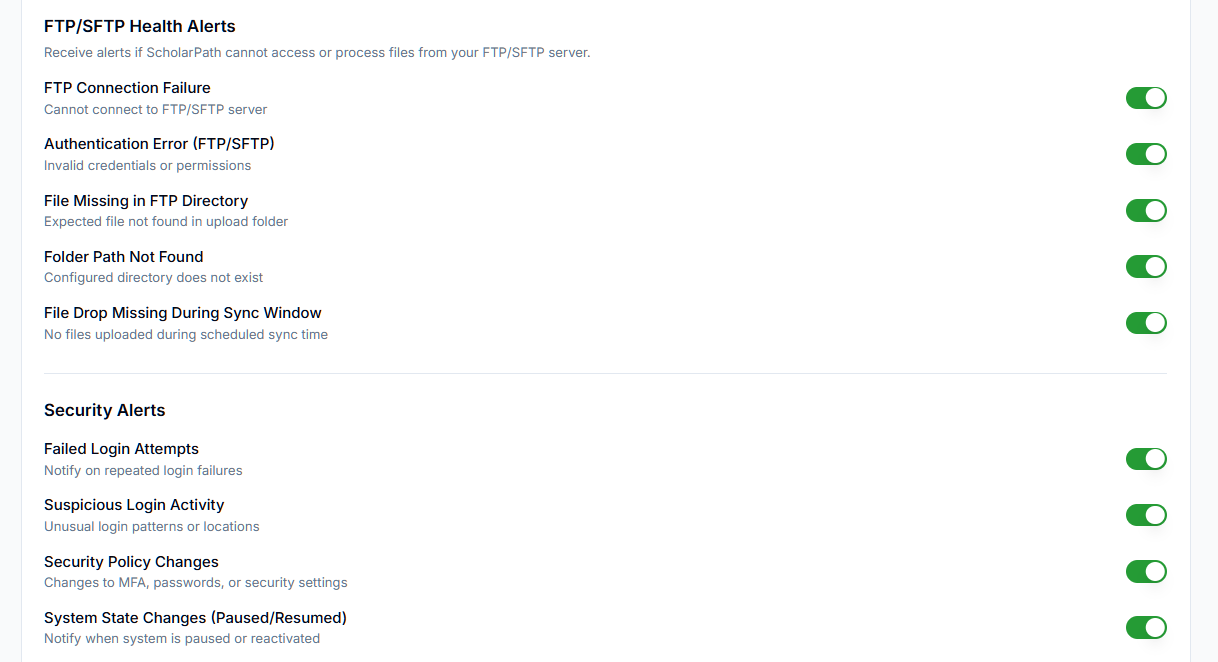


Figure :System Alerts

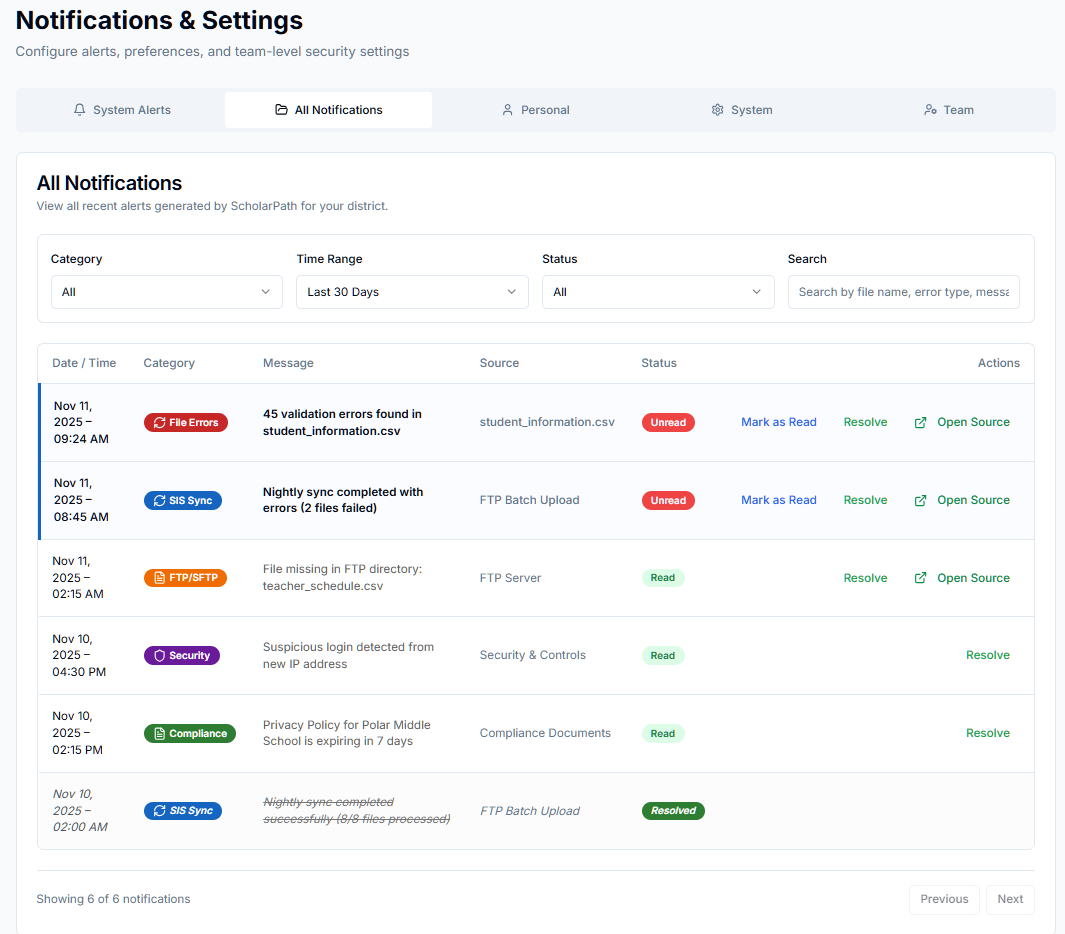


Figure : All Notifications

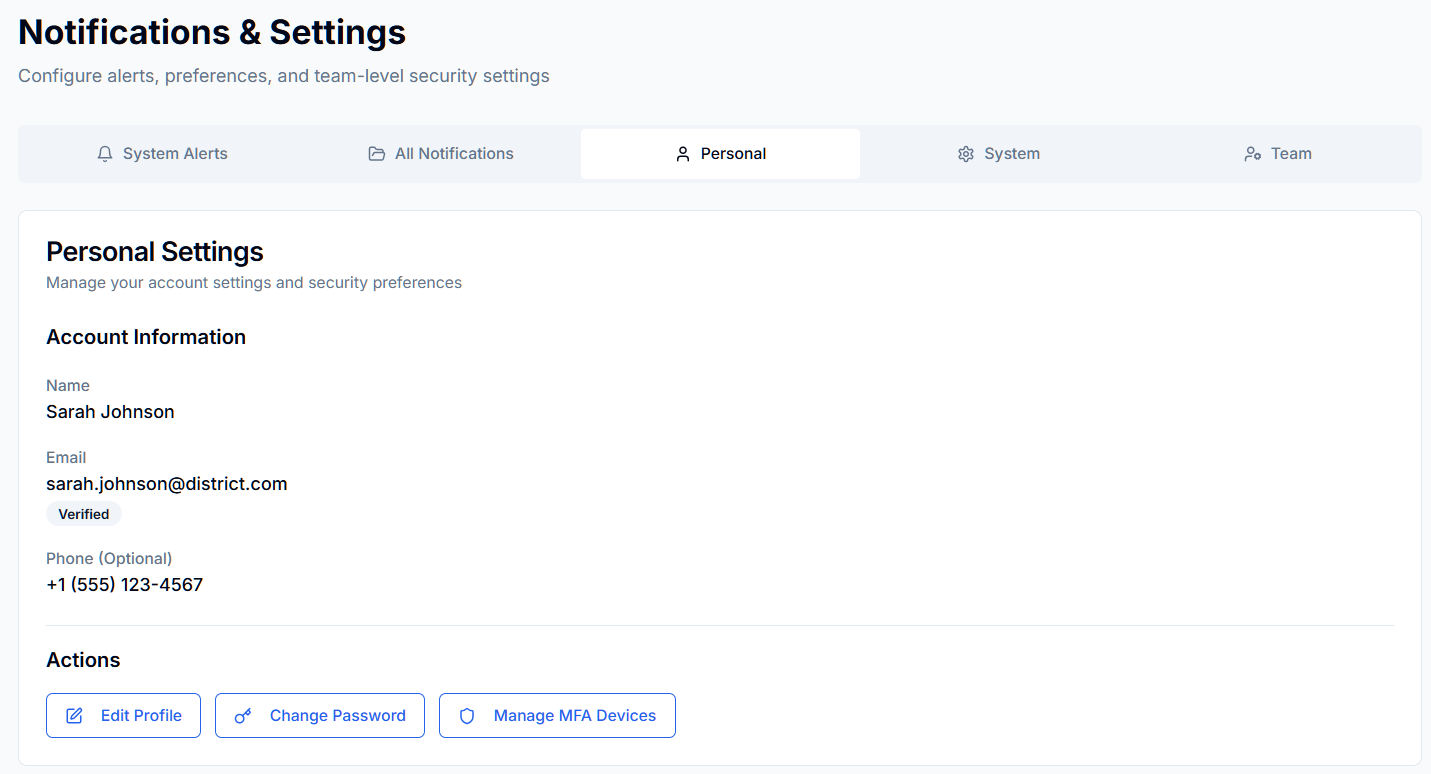


Figure : Personal Settings

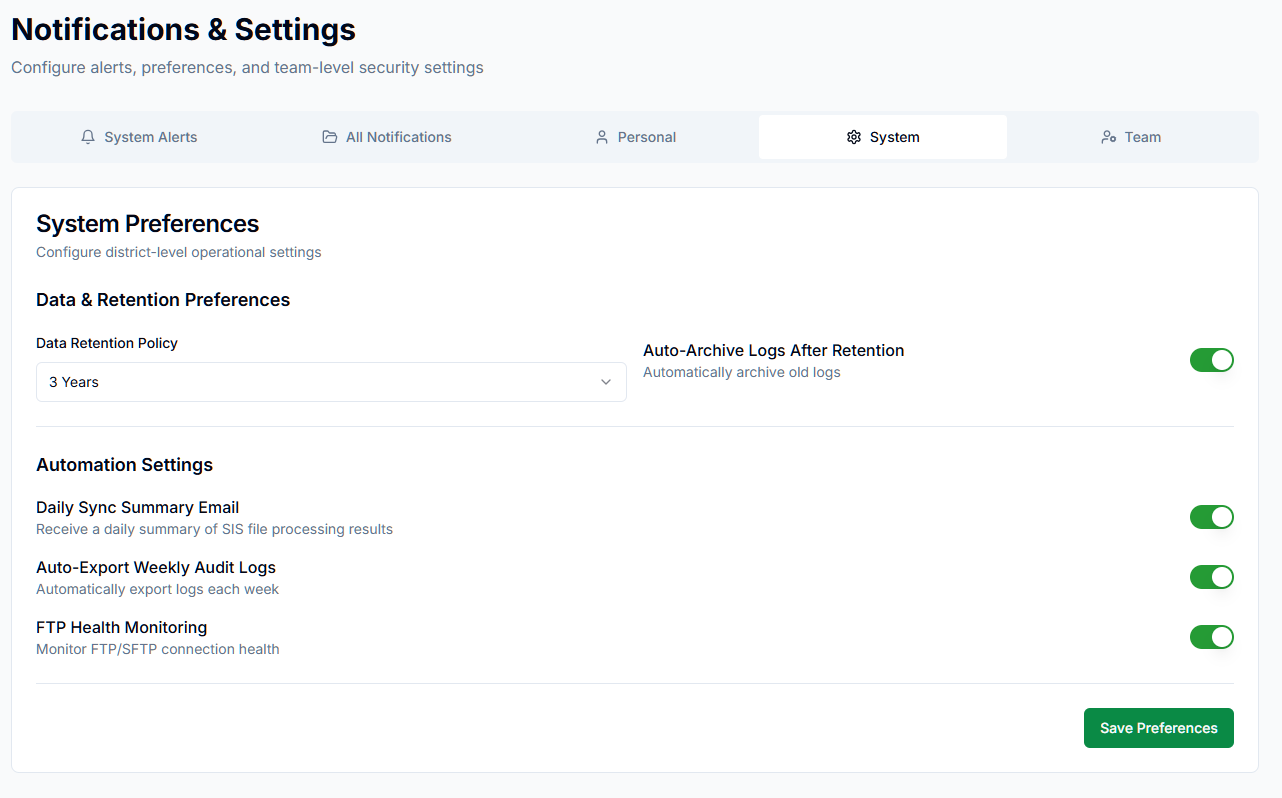


Figure : System Settings

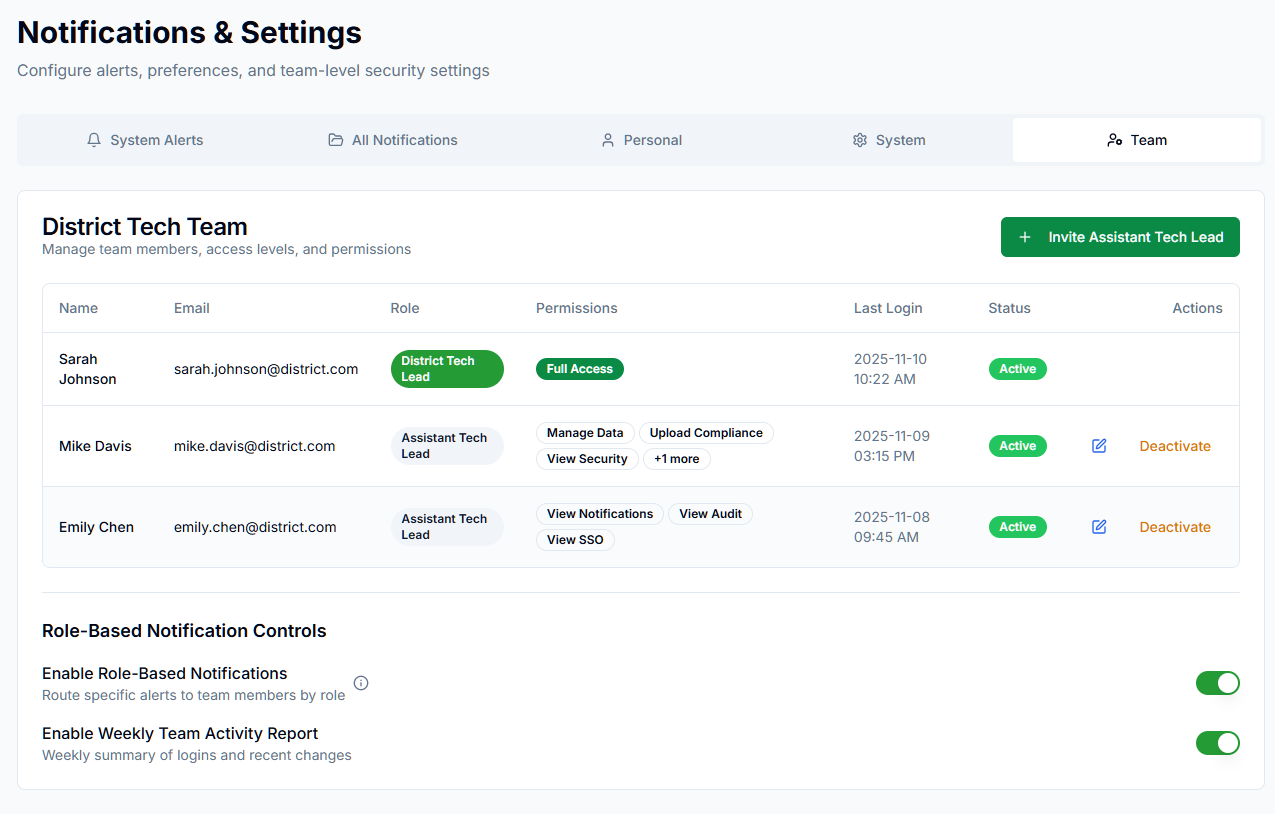


Figure : Team Settings

**Involved APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When Called** |
| Get Notifications | GET /notifications/all | Loads unified list | On tab load |
| Update Delivery Settings | POST /notifications/settings | Saves user preferences | Save Preferences |
| Mark as Read | POST /notifications/{id}/read | Marks item read | On click |
| Resolve Notification | POST /notifications/{id}/resolve | Marks item resolved | On click |
| Export Notifications | GET /notifications/export | CSV download | Export |
| Get Category Alerts | GET /notifications?category=<type> | Loads filtered lists | Category tabs |
| Audit Logging | POST /audit | Records notification actions | Every action |

# Use Case 11: Manage District Tech Team Roles and Access

**Description**

This use case describes how District Tech Leads manage the district’s technical team by inviting additional District Tech Leads, viewing team member details, and monitoring access status.

District Tech Leads cannot deactivate or remove other Tech Leads; these actions are restricted to ScholarPath Admin for security.

The Team tab provides full visibility into active, invited, and deactivated Tech Lead accounts and ensures audit logging for all team-related actions.

**Goal**

To allow District Tech Leads to invite new team members as District Tech Leads, view team details, resend invitations, and monitor login activity—without the ability to remove or deactivate any Tech Lead.

**Actors**

* **Primary Actor:** District Tech Lead
* **Secondary Actor:** ScholarPath Admin (for account deactivation/removal)

**Trigger**

User navigates to Notifications & Settings → Team.

**Business Rules**

|  |  |  |
| --- | --- | --- |
| **#** | **Rule** | **Description** |
| BR-01 | Single Role Model | Every member is a District Tech Lead. No Assistant or secondary roles. |
| BR-02 | Invitation Power | District Tech Leads may invite additional District Tech Leads. |
| BR-03 | No Peer Removal | District Tech Leads cannot deactivate or remove other Tech Leads. Only ScholarPath Admin can perform these actions. |
| BR-04 | Minimum Team Rule | At least one active District Tech Lead must always exist. |
| BR-05 | Invitation Restrictions | Invited email must not exist in another district. |
| BR-06 | Status Types | Member statuses: Invited, Active, Deactivated. |
| BR-07 | Login Visibility | District Tech Leads can view last login timestamp for each tech member. |
| BR-08 | Audit Logging | Invite, resend invitation, activation, and admin removals must be logged. |
| BR-09 | Activation Process | Invited Tech Leads must set a password through a secure 24-hour activation link. |

**Pre-Conditions**

* District account is active.
* User is an authenticated District Tech Lead.
* User has completed onboarding.

**Steps**

1. View District Tech Team
2. User navigates to Notifications & Settings → Team.
3. System displays team members in a table with columns: Name, Email, Status, Last Login, Actions.

2. Invite a New District Tech Lead

* User clicks Invite District Tech Lead.
* System opens the *Invite Tech Lead* modal with fields:

Full Name

Email Address

User enters details and clicks Send Invite.

System validates uniqueness of email.

System sends invitation email with 24-hour activation link.

System logs audit entry:

“District Tech Lead Invited — {{email}} by {{inviter\_name}}”

Table updates with member status = Invited.

ScholarPath — District Tech Lead Invitation Email Template

Subject: You’ve Been Invited to Join ScholarPath as a District Tech Lead

Hi {{recipient\_name}},

You’ve been invited to join the {{district\_name}} District on ScholarPath as a District Tech Lead.

As a District Tech Lead, you’ll help manage:

Data integrations & nightly syncs

SSO configuration (Google, SAML, ClassLink)

Security policies & authentication health

Compliance documentation

Notifications & district-level settings

To get started, please activate your account:

Activate Your Account

Click the link below to set your password and complete your profile:

{{activation\_link}}

This link will expire in 24 hours for security purposes.

What Happens Next?

Once you activate your account, you’ll complete a quick onboarding setup:

Confirm your district profile

Upload required MOU (if not completed already)

Review basic security settings

Start accessing your district Dashboard and Security Center

Need Help?

Our support team is available if you need assistance.

Email: support@scholarpath.com

Website: www.scholarpath.com

Thank you,

The ScholarPath Team

Security Notes (Shown at Bottom of Email)

*For your security, never share your activation link with anyone.*

*If you did not expect this invitation, please contact ScholarPath Support immediately.*

3. Resend Invitation

User clicks Resend Invite for an Invited user.

System sends a fresh activation email.

System logs audit event:

“Invitation Resent — {{email}} by {{sender\_name}}”

Success toast shown.

4. View Status and Last Login

User views each member’s:

Status (Active / Invited / Deactivated)

Last Login date/time

No edit or deactivate options are shown.

5. ScholarPath Admin Actions (Outside This Screen)

ScholarPath Admin may:

Deactivate

Reactivate

When this occurs, District Tech Leads will see updated status in the Team table.

System logs each admin action in audit history.

**Negative Flows**

|  |  |
| --- | --- |
| **Scenario** | **System Response** |
| Email already exists under another district | “⚠ This email is already associated with another district.” |
| Invalid email format | “⚠ Please enter a valid email address.” |
| Email sending failure | “⚠ Unable to send invitation. Please try again.” |
| Attempt to deactivate/remove a user as Tech Lead | No button shown; action impossible. |
| All Tech Leads accidentally deactivated by Admin | Blocked: “❌ At least one active District Tech Lead is required.” |

**Post-Conditions**

1. New District Tech Leads are invited and visible in the team table.
2. Invitation events are logged in audit logs.
3. Status and last login timestamps are always up-to-date.
4. Removal and deactivation actions are restricted to ScholarPath Admin.

**Special Requirements**

1. Email invitation must use ScholarPath-branded template.
2. Activation link must expire in 24 hours.
3. Table supports search by name and email.
4. Status chips:
   * 🟢 Active
   * 🟡 Invited
   * 🔴 Deactivated
5. No role dropdowns or permissions panel anywhere.
6. Tech Lead cannot modify or remove any other Tech Lead.

**Screen Element Matrix (SEM)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Type** | **Purpose / Use** | **Dependency / API** |
| Team Table | Table | Shows all District Tech Leads | /api/districts/{districtId}/team |
| Invite Button | Button | Opens invitation modal | Client-side |
| Invite Modal | Modal | Input for Name + Email | /api/districts/{id}/team/invite |
| Resend Invite | Action | Resend activation link | /api/districts/{id}/team/invite/resend |
| Status Chip | Label | Shows Invited/Active/Deactivated | System computed |
| Last Login | Text | Shows last login timestamp | Auth service |
| Audit Log Link | Text/CTA | Opens Security Center → Audit Logs | /api/security/audit |

**Screenshots**

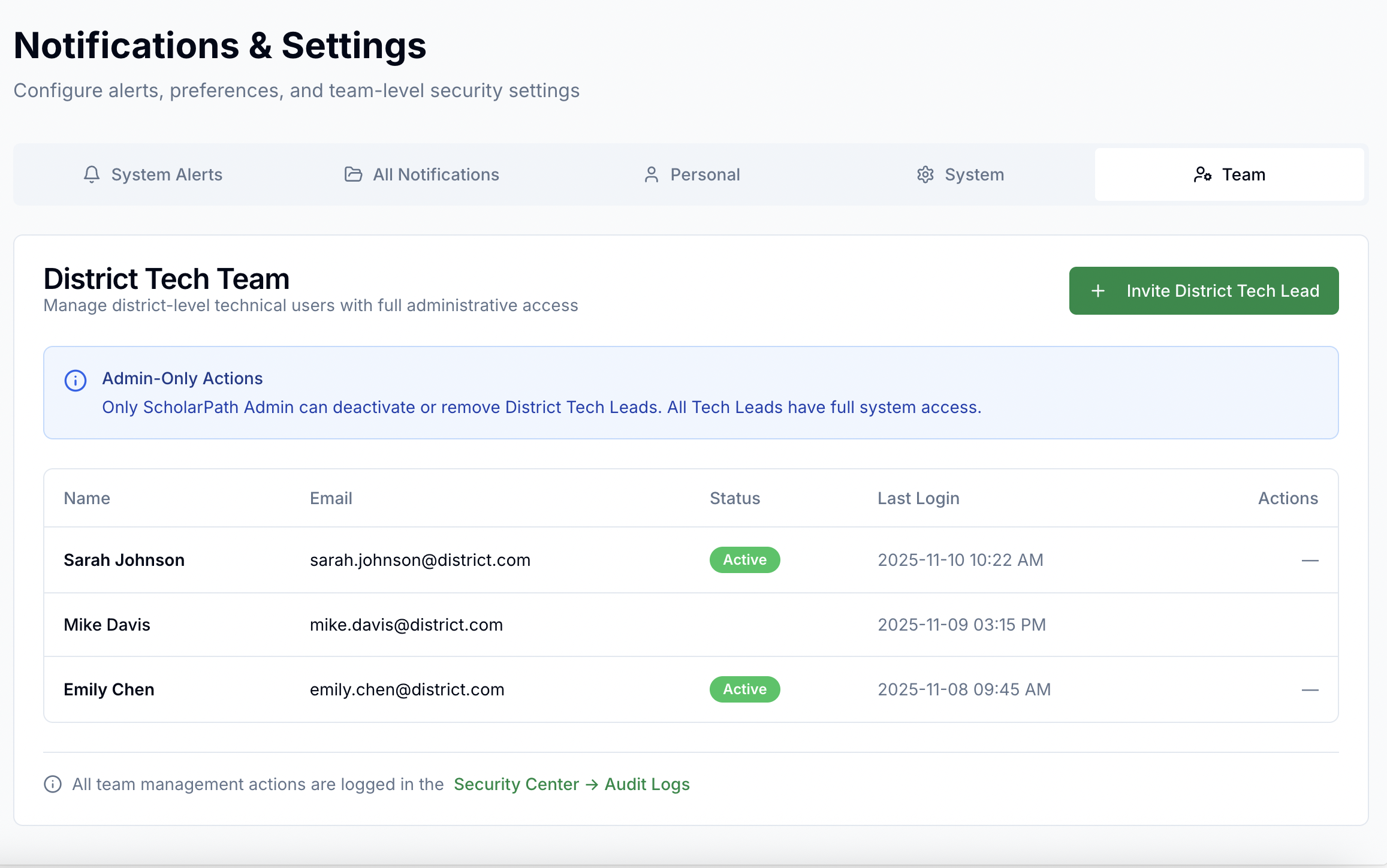


Figure : Notifications & Settings >> Teams tab

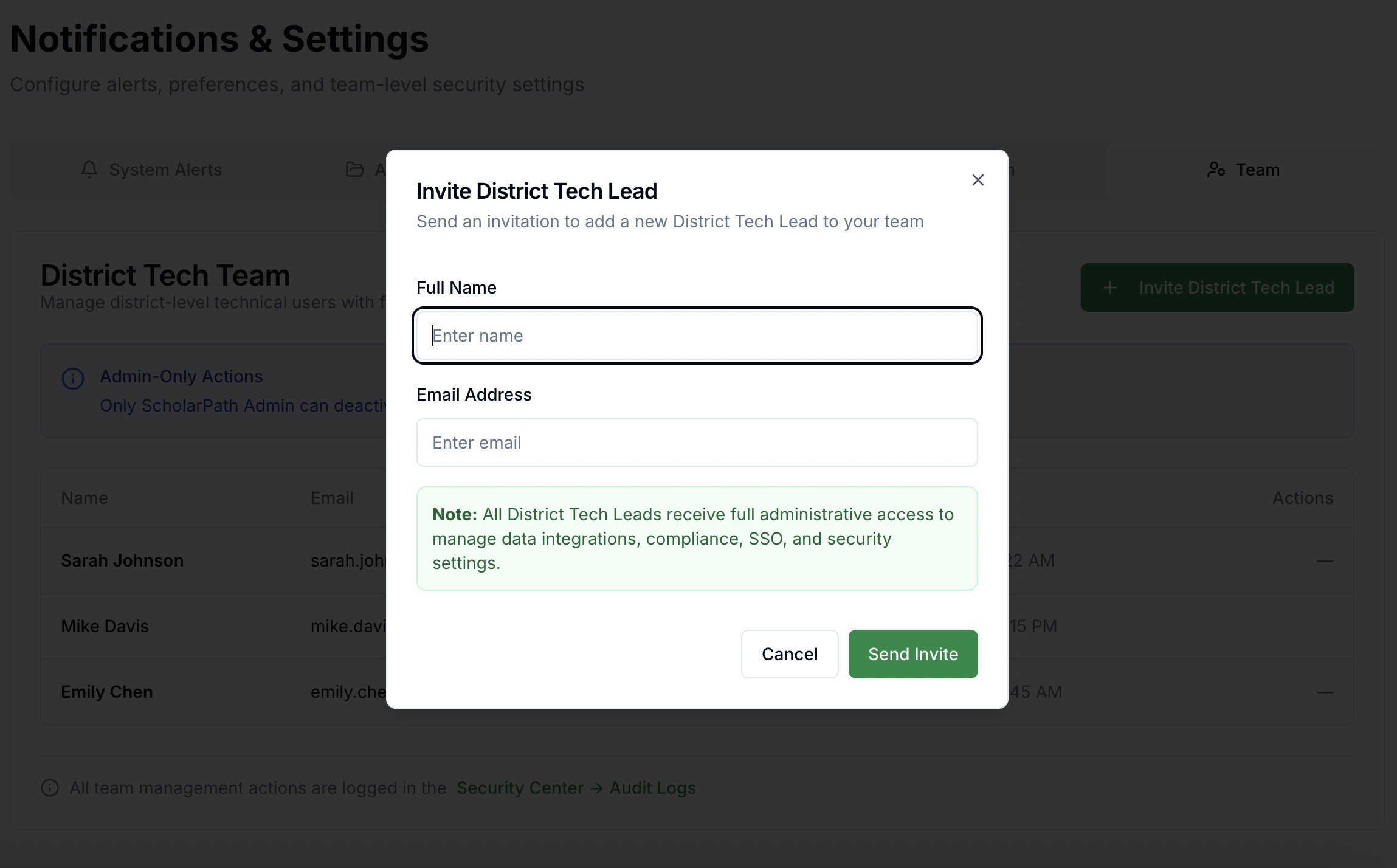


Figure : Invite new team member pop up

**Involved APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Description** | **When Called** |
| Get Tech Team | /api/districts/{districtId}/team | Fetch all Tech Leads | On Team tab load |
| Invite Tech Lead | /api/districts/{districtId}/team/invite | Sends invitation | On Send Invite |
| Resend Invitation | /api/districts/{districtId}/team/invite/resend | Resends activation link | On Resend Invite |
| Account Activation | /api/auth/activate/{token} | Activates account after invite | When user clicks activation link |
| Audit Logging | /api/security/audit | Logs invite/resend/admin actions | On every action |

# Use Case 12: Create New Resume from Scratch

### Description:

A student creates a new resume from the Resume Builder using a blank/templated layout, manually entering profile, education, experience, skills, and optional sections. The flow supports template selection (2 ATS-compliant templates), inline validation, autosave, preview, and saving as draft or primary.

### Actors:

* Primary: Student
* Supporting (read-only/controls): Parent (view/permission context)

### Goal:

Allow a student to quickly produce an ATS-compliant resume from scratch and save it (as a draft or primary) for later editing, sharing, or export.

### Trigger:

Student selects Plan & Build → Resume Builder → Create New Resume.

### Business Rules:

1. Students may hold a maximum of 5 resumes.
2. One resume must always be marked as Primary.
3. Students must select one of two available ATS-compliant templates to start a resume.
4. Templates enforce ATS compliance (simple headings, standard fonts, no tables/images, MMM YYYY date format).
5. Section requirements:

* Required: Name, at least one Education or Experience entry.
* Optional: Summary, Skills, Projects, Certifications, Activities, References.

1. Resume name must be unique; default naming = *Untitled Resume {n}*.
2. Warning message if navigating away with unsaved changes:

⚠️ You Have Unsaved Changes   
“It looks like you’ve made changes that haven’t been saved yet. If you leave this page now, your updates will be lost.”

1. All actions logged for audit ( resumeId, version, timestamp).

### Pre-Conditions:

* Student is authenticated.
* Student has < 5 existing resumes.

### Steps:

1. Student opens Resume Builder and clicks Create New Resume.
2. System initializes a new draft with a default ATS-compliant template and default name (e.g., “Untitled Resume 1”).
3. Student selects a Template (can switch at any time; content persists).
4. Student updates Resume Name.
5. Student completes Personal Info (name, email, phone, city/state, links).
6. Student adds Summary (optional, word/char counter shown).
7. Student adds Education (school, credential, location, dates, highlights).
8. Student adds Experience entries with bullet points (responsibility/action + impact).
9. Student adds Skills (tag list) and any optional sections (Projects, Certifications, Activities, References).
10. Live Preview renders updates; errors/omissions flagged inline.
11. Student clicks Save (or it autosaves).
12. (Optional) Student toggles Set as Primary.
13. Confirmation toast appears; resume persists as Draft or Primary and is visible in Your Resumes table.

### Negative Flow:

* Limit reached → Show: “You can only have up to 5 resumes. Delete one to create a new resume.”
* Validation errors (e.g., end date before start date, invalid email/URL) → Block save; show inline messages.
* Network/API failure on save → Toast: “Couldn’t save changes. Retry.” Autosave retries in the background.
* Template switch failure → Keep current template; message: “Template couldn’t load. Try again.”
* Duplicate name → Prompt to enter a unique name.
* Session timeout → Prompt re-authentication; preserve un-synced edits locally, then resume.

### Post-Condition:

* Resume is saved with the selected template and student-entered data.
* If “Set as Primary” is chosen, it becomes the Primary Resume.
* Draft appears in the Your Resumes table with an updated timestamp.

### Special Requirements:

* Accessibility: WCAG 2.1 AA; full keyboard navigation, ARIA labels, error summaries.
* Performance: Initial load < 2s; save/autosave < 500 ms p95.
* Localization: All labels/help text i18nready; date formats locale-aware (stored in ISO).
* Observability: Emit resume\_create\_started/succeeded/failed, template\_id, error\_code (no PII).
* Editing UX: Undo/redo, rich bullets, dragreorder sections; date pickers with month/year granularity.

### Constraints:

* Only two ATS-compliant templates available at launch.
* No external images/logos embedded in ATS templates.

### Screen Element Matrix:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen Name** | **Section Name** | **Element** | **Description:** | **Type** | **Options** | **Accepted Values** | **Accepted Size** | **Mandatory (Yes/No)** | **Validations** | **Comments** |
| Resume Builder | Page Header | Create New Resume | Starts manual resume creation flow | Button | — | — | — | No | Max 5 resumes can be created |  |
| Resume Builder | Page Header | Generate with AI | Starts AI assisted resume creation | Button | — | — | — | No | Max 5 resumes can be created |  |
| Resume Builder | Page Header | Upload Resume | Upload and parse an existing file | Button | — | File: PDF/DOCX | ≤ 5 MB | No | File type/size validation |  |
| Resume Builder | Your Resumes (Table) | Resume Name (column) | Displays resume title. | Text link | — | 1–100 chars, unique per student | 100 chars | Yes | Duplicate Names is restricted. |  |
| Resume Builder | Your Resumes (Table) | Last Updated (column) | Shows last modified date | Text | — | Date (MM/DD/YYYY) | — | Yes | Valid date |  |
| Resume Builder | Your Resumes (Table) | Status (column) | Denotes Primary/Secondary | Badge | Primary, Secondary | Fixed values | — | Yes | One Primary in list |  |
| Resume Builder | Your Resumes (Table) | Actions (column) | Row action menu Trigger: | Icon Button (⋮) | Edit  Rename  Make a Copy  Share  Download  Set as Primary  Delete | — | — | Yes | — |  |
| Resume Builder | Row Actions (⋮ menu) | Edit | Opens resume in editor | Menu Item | — | — | — | No | — |  |
| Resume Builder | Row Actions (⋮ menu) | Rename | Rename selected resume | Menu Item | — | 1–100 chars, unique per student | 100 chars | Yes | Required; uniqueness; disallow only whitespace |  |
| Resume Builder | Row Actions (⋮ menu) | Make a Copy | Duplicates resume | Menu Item | — | New name auto: “{Old Name} – Copy {n}” | 100 chars | No | Block if limit (=5) reached |  |
| Resume Builder | Row Actions (⋮ menu) | Share | Share resume with provider(s) | Menu Item | — | Provider with existing message thread | — | No | Enforce parental messaging permission; Grades 9–12; existing conversation |  |
| Resume Builder | Row Actions (⋮ menu) | Download | Export resume | Menu Item (submenu) | PDF, DOCX, TXT | File per selection | ≤ 5 MB | No | Export must be ATScompliant |  |
| Resume Builder | Row Actions (⋮ menu) | Set as Primary | Makes this resume Primary | Menu Item | — | — | — | No | Visible only for Secondary; switches Primary flag |  |
| Resume Builder | Row Actions (⋮ menu) | Delete | Deletes resume | Menu Item | — | — | — | No | Disabled/hidden for Primary; confirm required |  |
| Resume Builder | Rename (Modal) | Resume Name | Input for new title | Text Input | — | 1–100 chars | 100 chars | Yes | Required; uniqueness; trim leading/trailing spaces |  |
| Resume Builder | Rename (Modal) | Save | Confirms rename | Button | — | — | — | Yes | Enabled only when valid input |  |
| Resume Builder | Rename (Modal) | Cancel | Closes modal | Button | — | — | — | No | — |  |
| Resume Builder | Delete (Confirm) | Confirmation Text | Warns about deletion | Text | — | — | — | — | — |  |
| Resume Builder | Delete (Confirm) | Delete | Confirms deletion | Button | — | — | — | No | Enabled only for Secondary |  |
| Resume Builder | Delete (Confirm) | Cancel | Aborts deletion | Button | — | — | — | No | — |  |
| Resume Builder | Unsaved Changes (global) | Warning Popup | Shown on navigation with unsaved edits | Modal | — | Message: “⚠️ You Have Unsaved Changes…your updates will be lost.” | — | — | Blocks navigation until a choice. |  |
| Resume Builder | Template Cards | Professional Classic Creative Modern | Template card with preview thumbnail | Card | Professional Classic, Creative Modern | Fixed | — | Yes | One must be chosen before continuing. |  |
| Resume Builder | Template Card – Actions | Preview | Opens full preview of the Modern template | Button | — | — | — | No | Opens preview modal/pane |  |
| Resume Builder | Template Card – Actions | Use This Template | Selects Modern and closes modal | Button | — | — | — | Yes (to proceed) | Commits selection |  |
| Resume Builder | Template Modal | Close (X) | Dismisses the modal | Icon Button | — | — | — | No | Esc/Click closes |  |
| Resume Builder | Name Modal | Close (X) | Dismisses the modal | Icon Button | — | — | — | No | Esc/Click closes |  |
| Resume Builder | Name Modal | Resume Name | Input field to name the resume | Text Input | — | 1–100 characters; printable text | 100 chars | Yes | Required; not only whitespace; trim leading/trailing spaces; uniqueness per student |  |
| Resume Builder | Name Modal | Back | Returns to previous step (template selection) | Button | — | — | — | No | — |  |
| Resume Builder | Name Modal | Continue | Proceeds to editor and creates draft | Button | — | Valid name only | — | Yes | Enabled only when input passes validation |  |
| Resume Builder – Editor | Personal Info | Name | Student full name | Text Input | — | Text | 1–120 chars | Yes | Required; no numerics-only; shows in preview |  |
| Resume Builder – Editor | Personal Info | Email | Contact email | Text Input | — | Valid email | 5–254 chars | No | Email format validation |  |
| Resume Builder – Editor | Personal Info | Phone | Contact phone | Text Input | — | E.164 / local format (XXX)XXX-XXXX | 7–20 chars | No | Phone format validation |  |
| Resume Builder – Editor | Personal Info | City | City | Text Input | — | Text | 1–80 chars | No | — |  |
| Resume Builder – Editor | Personal Info | State | State/Province | Text Input / Dropdown | (State list) | 2–50 chars | 2–50 chars | No | — |  |
| Resume Builder – Editor | Personal Info | Links | Portfolio / LinkedIn / GitHub / Website | Text Inputs | — | Valid URL(s) | 1–2048 chars each | No | URL format validation |  |
| Resume Builder – Editor | Summary | Summary | Optional professional summary with counter | Text Area | — | Text | up to 1,000 chars (recommended) | No | Counter shown; spellcheck; hint for recommended length |  |
| Resume Builder – Editor | Education | School | Institution name | Text Input | — | Text | 1–150 chars | Conditional\* | Required if no Experience exists (Education OR Experience ≥1) |  |
| Resume Builder – Editor | Education | Credential | Degree/qualification | Text Input | — | Text | 1–120 chars | Conditional\* | — |  |
| Resume Builder – Editor | Education | Location | City, State | Text Input | — | Text | 1–120 chars | No | — |  |
| Resume Builder – Editor | Education | Start Date | Education start | Month/Year Picker | — | MMM YYYY | — | No | Date format MMM YYYY |  |
| Resume Builder – Editor | Education | End Date | Education end or “Present” | Month/Year Picker / Toggle | — | MMM YYYY / Present | — | No | End ≥ Start when provided |  |
| Resume Builder – Editor | Education | Highlights | Bulleted achievements | Repeating Text Inputs | — | Text | ≤ 200 chars per bullet | No | Encourage action + impact phrasing |  |
| Resume Builder – Editor | Experience | Employer | Company/organization | Text Input | — | Text | 1–150 chars | Conditional\* | Required if no Education exists (Education OR Experience ≥1) |  |
| Resume Builder – Editor | Experience | Role/Title | Position title | Text Input | — | Text | 1–120 chars | Conditional\* | — |  |
| Resume Builder – Editor | Experience | Location | City, State | Text Input | — | Text | 1–120 chars | No | — |  |
| Resume Builder – Editor | Experience | Start Date | Employment start | Month/Year Picker | — | MMM YYYY | — | Conditional\* | Chronology validation |  |
| Resume Builder – Editor | Experience | End Date | Employment end or “Present” | Month/Year Picker / Toggle | — | MMM YYYY / Present | — | No | End ≥ Start when provided |  |
| Resume Builder – Editor | Skills | Skill Tags | Technical/soft skills | Tag Input | — | Text | ≤ 50 chars per tag | No | Max 10 skills; block duplicates; normalize casing |  |
| Resume Builder – Editor | Right Pane | Live Preview | ATS-styled live render | Render Panel | Chosen template | — | — | — | Reflects content/template; ATS fonts only; no tables/images |  |
| Resume Builder – Editor | Footer | Save | Persist resume | Button | — | — | — | Yes | Server-side validate on save; errors show inline |  |
| Resume Builder – Editor | Footer | Set as Primary | Mark this resume Primary | Toggle/Button | — | — | — | No | Enforces “exactly one Primary”; unsets existing Primary |  |
| Resume Builder – Editor | Footer | Cancel | Exit editor | Button | — | — | — | — | Shows unsaved-changes warning dialog |  |

Manual Resume  

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Section Name** | **Element** | **Description:** | **Type** | **Options** | **Accepted Values** | **Accepted Size** | **Mandatory (Yes/No)** | **Validations** | **Comments** |
| Header | Page Title | Displays main page heading “Create New Resume” | Text Label | — | — | — | — | Static text only | Always visible |
| Header | Instructional Text | “Build your resume from scratch…” helper text | Text Label | — | — | — | — | — | Guiding text |
| Personal Information | Section Header | “Personal Information” | Text Label | — | — | — | — | — | Section heading |
| Personal Information | Full Name | Student’s legal full name | Text Input | — | Alphabets, hyphens, spaces | 100 chars | Yes | Required; cannot contain only digits | Displayed first |
| Personal Information | Email | Student’s email address | Text Input | — | Valid email format | 100 chars | Yes | Must contain “@” and valid domain | Inline red message if invalid |
| Personal Information | Phone Number | Student’s phone number | Text Input | — | Digits and “+” | 15 digits | No | Must contain 10–15 digits if entered | Optional |
| Personal Information | Address (City, State, Zip) | Enter address information | Text Input | — | Alphanumeric + commas | 150 chars | No | Optional; no special validation | Displayed beside phone number |
| Personal Information | City | Select or enter city name | Dropdown | City List | City Name | - | No | Must not contain digits | Linked to State dropdown |
| Personal Information | State | Select state from dropdown | Dropdown | Full state list | State name | — | No | Required if City entered | Drop-down with fixed state list |
| Personal Information | Pincode | Enter postal or ZIP code | Text Input | — | 5–6 digit numeric | 6 digits | No | Must be valid numeric PIN/ZIP | Required |
| Personal Information | LinkedIn Profile | Provide LinkedIn profile link | Text Input | — | Must start with “[https://linkedin.com/”](https://linkedin.com/%E2%80%9D) | 200 chars | No | Must be valid URL | Optional |
| Personal Information | Portfolio/Website | Provide personal portfolio or website | Text Input | — | Must start with “http://” or “https://” | 200 chars | No | Optional URL | Optional |
| Career Objective / Summary | Section Header | “Career Objective / Summary” | Text Label | — | — | — | — | — | — |
| Career Objective / Summary | Professional Summary | Enter short professional summary | Text Area | — | Alphanumeric text | 2000 chars | No | Inline counter shows word/char limit | Rich text box |
| Career Objective / Summary | AI Assist Button | Trigger: AI suggestion for summary | Button | — | — | — | No | Requires non-empty summary box | Tooltip: “Get AI Suggestions” |
| Education | Section Header | “Education” | Text Label | — | — | — | — | — | — |
| Education | School Name | Enter name of educational institution | Text Input | — | Text | 150 chars | Conditional | Required if Education section used | — |
| Education | Location (City, State) | Enter school location | Text Input | — | Text | 100 chars | No | Optional | — |
| Education | Start Date | Select start date | Date Picker | — | mm/dd/yyyy | — | Conditional | Must be valid date; not future | Month/Year picker |
| Education | Expected Graduation | Select expected graduation year | Dropdown | 2024–2035 | Year | — | No | Required if Start Date entered | — |
| Education | GPA (optional) | Enter GPA value | Text Input | — | Decimal (0–4) | 4 chars | No | Must be numeric 0.00–4.00 | Optional |
| Education | + Add another education | Add new Education block | Button | — | — | — | No | Adds new entry | Repeatable field group |
| Experience | Section Header | “Experience” | Text Label | — | — | — | — | — | — |
| Experience | Role/Title | Job title or role | Text Input | — | Text | 100 chars | Conditional | Required with Company name | — |
| Experience | Company/Organization | Name of employer or organization | Text Input | — | Text | 150 chars | Conditional | Required if Role entered | — |
| Experience | Location | Enter workplace location | Text Input | — | Text | 100 chars | No | Optional | — |
| Experience | Start Date | Enter work start date | Date Picker | — | mm/dd/yyyy | — | Conditional | Must be before End Date | — |
| Experience | End Date | Enter work end date | Date Picker | — | mm/dd/yyyy | — | Conditional | Must be after Start Date | Disabled if “I currently work here” checked |
| Experience | I currently work here | Checkbox | Marks job as ongoing | Checkbox | Checked/Unchecked | — | No | Disables End Date when checked | — |
| Experience | Description: / Responsibilities | Describe key duties, outcomes, achievements | Text Area | — | Alphanumeric, bullet list supported | 2000 chars | No | Optional | Allows bullets and formatting |
| Experience | + Add another experience | Add new Experience entry | Button | — | — | — | No | Adds repeatable Experience fields | — |
| Skills | Section Header | “Skills” | Text Label | — | — | — | — | — | — |
| Skills | Technical Skills | Add one or more technical skills | Tag Input | — | Text | 50 chars per skill | No | Cannot include numbers/symbols | Tag entry control |
| Skills | + Add another technical skill | Add additional skill entry | Button | — | — | — | No | Adds more tag fields | Unlimited entries |
| Skills | Soft Skills | Add soft/interpersonal skills | Tag Input | — | Text | 50 chars per skill | No | Cannot include digits | Tag entry control |
| Skills | + Add another soft skill | Add additional soft skill | Button | — | — | — | No | Adds more soft skill entries | Unlimited |
| Projects (optional) | Section Header | “Projects (optional)” | Text Label | — | — | — | — | — | — |
| Projects (optional) | Project Title | Enter project name | Text Input | — | Text | 150 chars | No | Required if Description: filled | — |
| Projects (optional) | Role | Enter role in project | Text Input | — | Text | 100 chars | No | Optional | — |
| Projects (optional) | Description: | Describe project, outcomes, and tech used | Text Area | — | Text | 2000 chars | No | Optional | — |
| Projects (optional) | Links (GitHub, demo, etc.) | Add project-related links | Text Input | — | Must start with “https://” | 200 chars | No | Must be valid URL | Optional |
| Projects (optional) | + Add another project | Add more projects | Button | — | — | — | No | Adds additional project group | Repeatable |
| Certifications (optional) | Section Header | “Certifications (optional)” | Text Label | — | — | — | — | — | — |
| Certifications (optional) | Certification Name | Enter name of certification | Text Input | — | Text | 150 chars | No | Required if Date Achieved entered | — |
| Certifications (optional) | Issuing Authority | Enter certifying organization | Text Input | — | Text | 150 chars | No | Optional | — |
| Certifications (optional) | Date Achieved | Select date achieved | Date Picker | — | mm/dd/yyyy | — | No | Must be valid past date | — |
| Certifications (optional) | Expiry Date | Select expiry date | Date Picker | — | mm/dd/yyyy | — | No | Must be after Date Achieved | Optional |
| Certifications (optional) | + Add another certification | Add another certification block | Button | — | — | — | No | Adds new certification fields | Repeatable |
| Extracurriculars / Leadership (optional) | Section Header | “Extracurriculars / Leadership (optional)” | Text Label | — | — | — | — | — | — |
| Extracurriculars / Leadership (optional) | Organization/Club | Enter name of organization or club | Text Input | — | Text | 150 chars | No | Optional | — |
| Extracurriculars / Leadership (optional) | Role/Title | Enter leadership title | Text Input | — | Text | 100 chars | No | Optional | — |
| Extracurriculars / Leadership (optional) | Achievements & Impact | Describe activities and outcomes | Text Area | — | Text | 1000 chars | No | Optional | — |
| Extracurriculars / Leadership (optional) | + Add another activity | Add new extracurricular entry | Button | — | — | — | No | Adds another activity block | Repeatable |
| References (optional) | Section Header | “References (optional)” | Text Label | — | — | — | — | — | — |
| References (optional) | Name | Reference name | Text Input | — | Text | 100 chars | No | Required if Relationship entered | — |
| References (optional) | Relationship | Relationship with reference | Text Input | — | Text | 100 chars | No | Optional | Examples: Teacher, Coach |
| References (optional) | Contact Information | Enter reference email or phone | Text Input | — | Valid email or phone | 100 chars | No | Must be valid if filled | Optional |
| References (optional) | + Add another reference | Add another reference entry | Button | — | — | — | No | Adds repeatable field set | — |
| Footer | Save as Draft | Saves resume as draft | Button | — | — | — | No | Saves resume with draft status | Timestamp added |
| Footer | Preview | Preview resume | Button | — | — | — | No | Opens live preview modal | Available any time |
| Footer | Download | Download resume file | Button (disabled until save) | — | PDF | — | No | Enabled after save | Downloads ATS file |
| Footer | Save & Continue | Save and return to Resume Builder | Button | — | — | — | Yes | Required to finalize changes | Confirmation toast appears |
| Footer | Cancel | Cancel resume creation | Button | — | — | — | No | Displays unsaved changes warning | Modal: “⚠ You Have Unsaved Changes” |
| Inline Controls | Autosave Indicator | Displays “Saving…” or “All changes saved” | Label/Status | — | — | — | No | Refreshes after every 15 sec or blur | Bottom-right indicator |
| Inline Controls | Toast Confirmation | “Resume saved successfully” or “Retry failed” | Popup | Success / Error | — | — | No | Shows after save or error | Persistent for 3s |
| Inline Controls | Unsaved Changes Modal | “You have unsaved changes…” warning popup | Modal Dialog | Leave / Stay buttons | — | — | Yes (conditional) | Triggered if navigation before save | Modal prevents accidental loss |

### Screenshot:

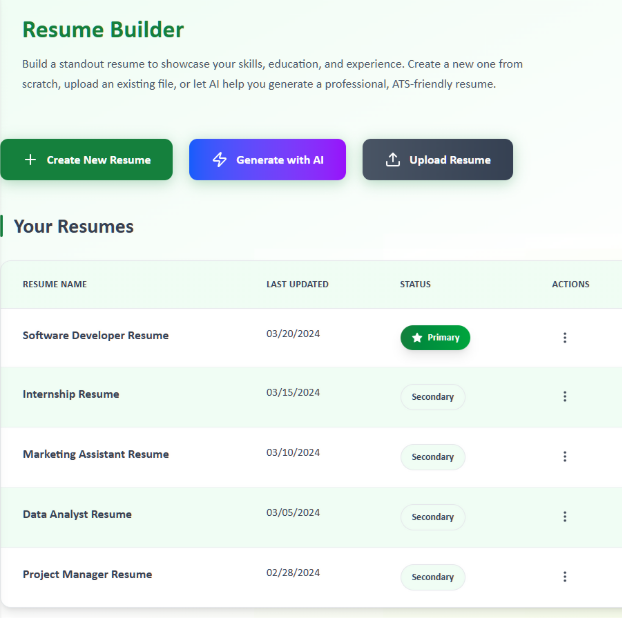


Figure : Resume Builder

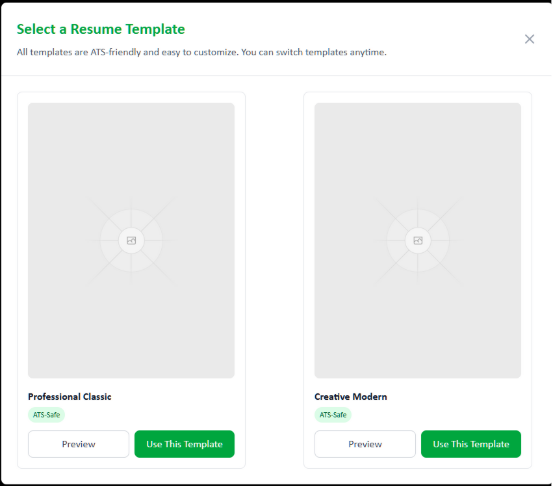
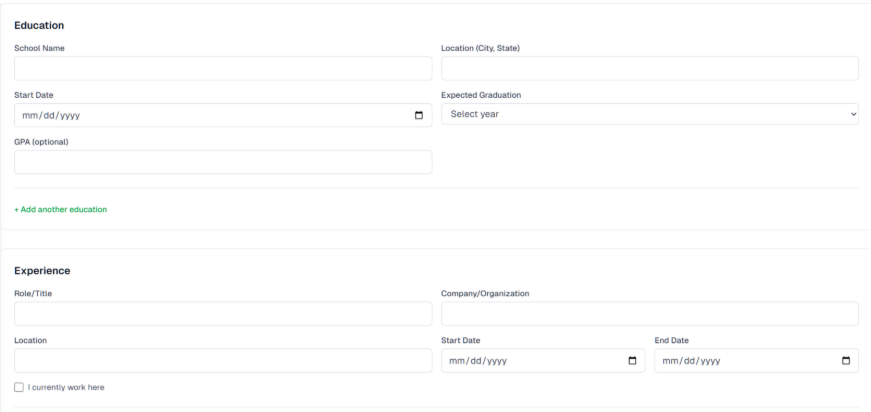
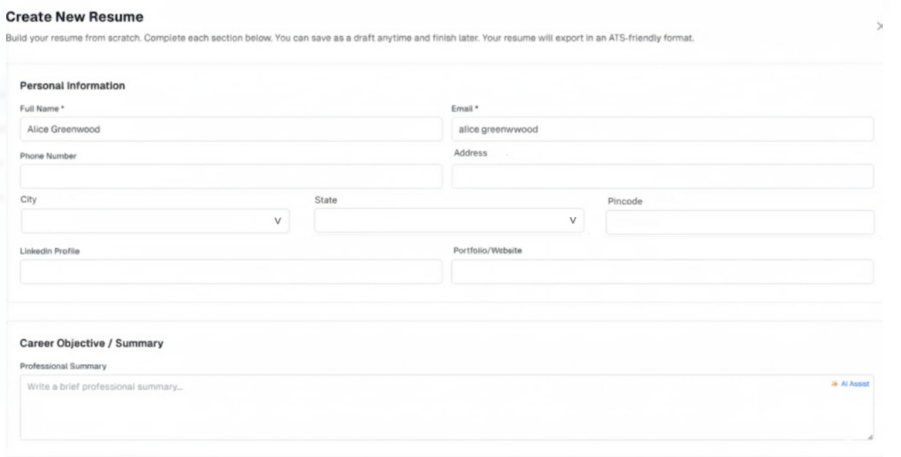
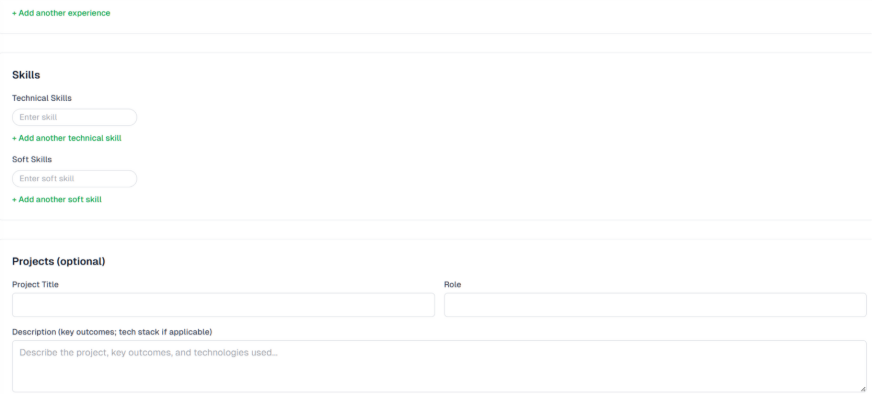
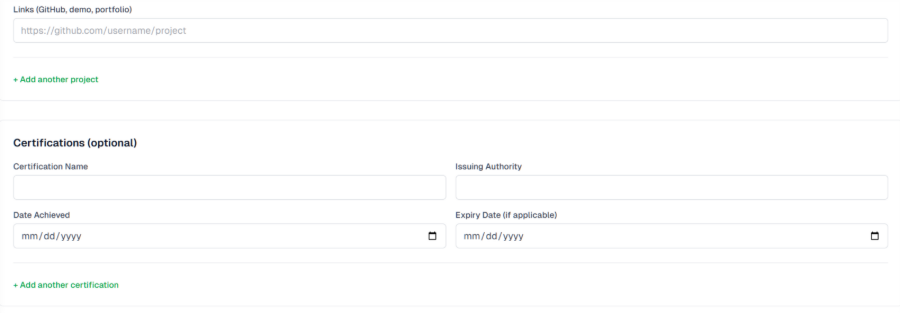


Figure : Select Resume Template







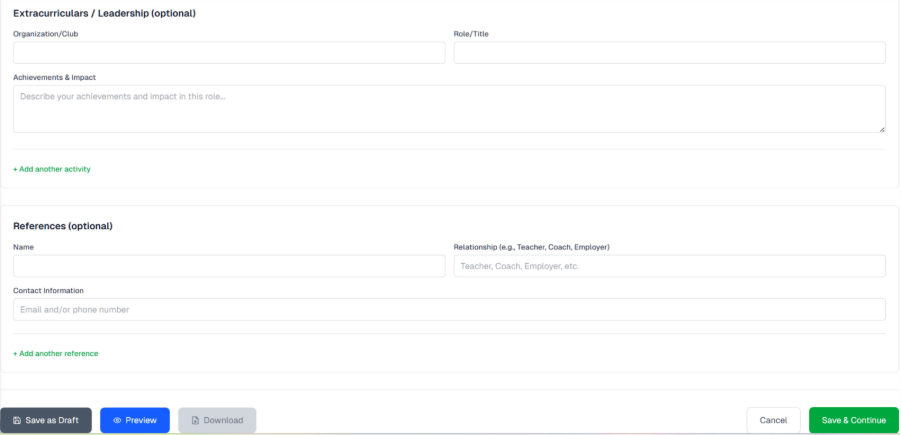


Figure : Create a Resume Manually

### Involved APIs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Method** | **Description:** | **When is Called?** |
| InitializeResumeDraft | /api/resumes | POST | Creates a new resume draft with default template and name | On “Create New Resume” click |
| UpdateResumeMeta | /api/resumes/{resumeId} | PUT | Updates resume name/template/ordering metadata | On rename or template switch |
| UpsertResumeSection | /api/resumes/{resumeId}/sections | PUT | Creates/updates section payloads (education, experience, skills, etc.) | On field blur/autosave/save |
| SetPrimaryResume | /api/resumes/{resumeId}/set-primary | PUT | Sets the resume as primary (and unsets existing primary) | On “Set as Primary” |
| ValidateResume | /api/resumes/{resumeId}/validate | POST | Runs serverside validation (dates, formats, required sections) | On save and preexport checks |
| GetTemplates | /api/resumes/templates | GET | Retrieves available ATS templates and capabilities | On screen load/template panel open |
| LogAuditEvent | /api/audit/resume | POST | Logs create/update/primaryset events | After each successful action |

# Use Case 13: Generate AI-Assisted Resume

### Description:

This use case describes how a student can generate a resume using AI assistance in ScholarPath. The system leverages AI to create a draft resume based on minimal input (such as education, experience, and skills). The student can then review, preview in a chosen template, and edit the AI-generated resume in the Resume Editor before saving it. Parents have read-only visibility of the generated resume, while educators and school admins have no access.

### Actors:

* **Primary Actor**: Student (Grades 6–12)
* **Supporting Actor**: Parent (read-only view of generated resumes)

### Goal:

Enable students to quickly create a professional, ATS-friendly resume draft using AI with minimal manual input.

### Trigger:

Student clicks “Generate with AI” button on the Resume Builder landing screen.

Business Rules:

* AI generation is only available to students (not parents, educators, or admins).
* A maximum of 5 resumes per student is enforced (AI resumes included).
* One resume must always be marked as Primary; Primary cannot be deleted.
* AI-generated resumes must follow ATS-friendly formatting (no tables, graphics; standard fonts).
* Students can change templates after AI generation.
* Parents can view but not edit AI-generated resumes.

### Pre-Conditions:

* Students are logged into ScholarPath.
* Resume Builder module is accessible from Portfolio > Plan & Build.
* AI service integration is available and active.
* Students have fewer than 5 resumes stored.

### Steps:

1. Login as Student.
2. Navigate to Portfolio > Plan & Build > Resume Builder.
3. Click Generate with AI button.
4. System opens an input form for minimal details (Education, Work Experience, Skills).
5. Student fills required fields and submits.
6. AI Resume Service generates a draft resume.
7. System displays generated resume in default template with preview mode.
8. Students choose a different template (optional) and system re-renders resume.
9. Student clicks Save Resume; resume stored in Your Resumes table with status (Primary/Secondary).

### Negative Flow:

* Student already has 5 resumes → system shows error: “Maximum resume limit reached. Delete an existing resume to create a new one.”
* NF2: AI service fails/unavailable → system shows error: “We’re unable to generate your resume right now. Please try again later.”
* NF3: Mandatory fields in input form left empty → inline error” This field is required” validation prevents submission.

### Post-Condition:

* A new AI-generated resume is saved in the student’s profile and visible in the Resume Builder landing table.
* Parents can view the AI-generated resume in read-only mode.
* Resume can be downloaded, shared, or edited further.

### Special Requirements:

* WCAG 2.1 AA accessibility compliance for forms, preview, and template switching.
* Audit logs must capture AI resume generation attempts (studentId, timestamp, status).
* Resume export formats: PDF, DOCX, TXT (ATS-compliant).

### Constraints:

* ScholarPath depends on the external AI resume service; downtime or failures affect availability.
* AI-generated resumes cannot exceed 5 stored resumes per student.
* Formatting must remain ATS-compliant and cannot use images or advanced styling.

### Screen Element Matrix:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen Name** | **Section Name** | **Element** | **Description:** | **Type** | **Options** | **Accepted Values** | **Accepted Size** | **Mandatory (Yes/No)** | **Validations** | **Comments** |
| Resume Builder – Landing | Actions | Generate with AI | Primary action to open the AI Resume Generator modal. Visible only to Student role. | Button | Click | N/A | N/A | No | Enabled only for Students with < 5 stored resumes; on click open modal. If resume count ≥ 5, show error “Maximum resume limit reached. Delete an existing resume to create a new one.” |  |
| AI Resume Generator – Modal | Header | Close | Dismisses the modal. | Icon Button (X) | Click | N/A | N/A | No | Closes modal without saving. If unsaved edits are present, confirm discard (system default). |  |
| AI Resume Generator – Modal | Profile Information (Auto-filled) | Full Name | Student’s full name, auto-filled from profile; editable in the form. | Text Input | N/A | Letters, spaces, hyphen, apostrophe, period | 2–100 chars | Yes | Required; trim whitespace; must contain alpha characters. |  |
| AI Resume Generator – Modal | Profile Information (Auto-filled) | Email | Student’s email, auto-filled from profile; editable. | Email Input | N/A | Valid email format (RFC 5322) | 5–254 chars | Yes | Required; must be valid email pattern; no leading/trailing spaces. |  |
| AI Resume Generator – Modal | Profile Information (Auto-filled) | Phone | Student’s phone number, auto-filled from profile; editable. | Tel Input | N/A | Digits, spaces, “+”, “( )”, “-”  (XXX)XXX-XXXX | 7–20 digits (excluding formatting) | Yes | Required; normalize to E.164 on submit when possible; reject letters; show inline error for invalid format. |  |
| AI Resume Generator – Modal | Education | School | School or institution name. | Text Input | N/A | Alphanumeric & punctuation | 2–120 chars | Yes | Required; trim; no all-numeric strings. |  |
| AI Resume Generator – Modal | Education | Location | City and state (or city & country) of the school. | Text Input | N/A | Letters, spaces, comma, hyphen | 2–100 chars | No | Optional; if provided, must include at least one letter; trim. |  |
| AI Resume Generator – Modal | Education | Graduation Year | Expected or actual graduation year. | Number/Input | N/A | 4-digit year (1900–2100) | 4 chars | Yes | Required; must be 4 digits within range; reject non-numeric; inline error on invalid year. |  |
| AI Resume Generator – Modal | Education | GPA (Optional) | Grade point average (optional). | Number/Input | N/A | 0–4 scale; allow 0.0–4.0 | 1–4 chars including decimal | No | Optional; if provided, numeric; 0.0 ≤ GPA ≤ 4.0; up to one decimal place; reject >4.0. |  |
| AI Resume Generator – Modal | Required Information | Desired Job Title | Target role for resume generation (e.g., “Software Engineering Intern”). | Text Input | N/A | Alphanumeric & punctuation | 2–80 chars | Yes | Required; trim; cannot be only whitespace; disallow emojis; show inline “This field is required” if empty. |  |
| AI Resume Generator – Modal | Required Information | Skills | Multi-value input to add skills; at least 3 required. | Chips/Tag Input | Add by Enter/Comma; remove via “x” on chip; keyboard accessible | Each skill: letters/numbers/symbols/spaces | 1–40 chars per skill; up to 50 skills; no duplicates | Yes (≥3) | Must have ≥3 unique skills; case-insensitive de-dup; trim; prevent empty chips; inline counter & error if <3. |  |
| AI Resume Generator – Modal | Experience Snapshots (Optional, 0–3 entries) | Add Experience | Adds an optional quick experience entry block (0–3 total). | Button | Click to add entry | N/A | N/A | No | Cap at 3 entries; disable “Add Experience” at 3; if an experience sub-form appears, all of its own required fields must pass before submit (system default). |  |
| AI Resume Generator – Modal | Footer | Cancel | Closes the modal without generating. | Button | Click | N/A | N/A | No | Closes modal; discard inputs; confirm if changes exist (system default). |  |
| AI Resume Generator – Modal | Footer | Generate Resume | Submits the form to the AI service and proceeds to preview. | Button | Click | N/A | N/A | Yes (gated by form) | Enabled only when all required fields valid and skills ≥3; on submit: re-validate, enforce max 5 resumes, call AI service; if service fails, show “We’re unable to generate your resume right now. Please try again later.” |  |

### Screenshot:

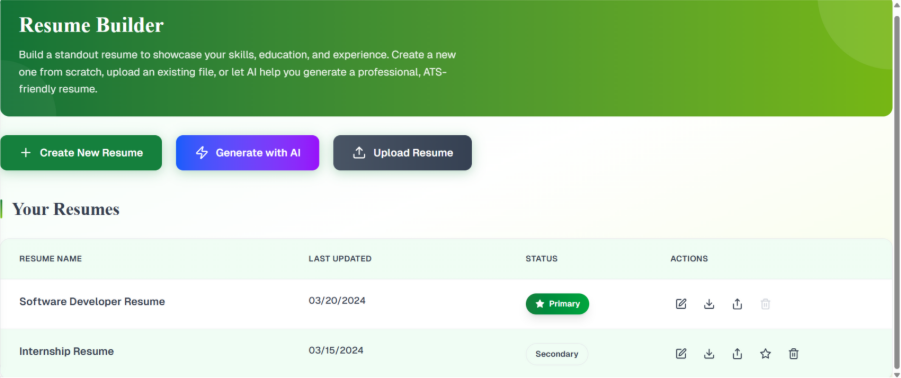


Figure : Generate With AI button

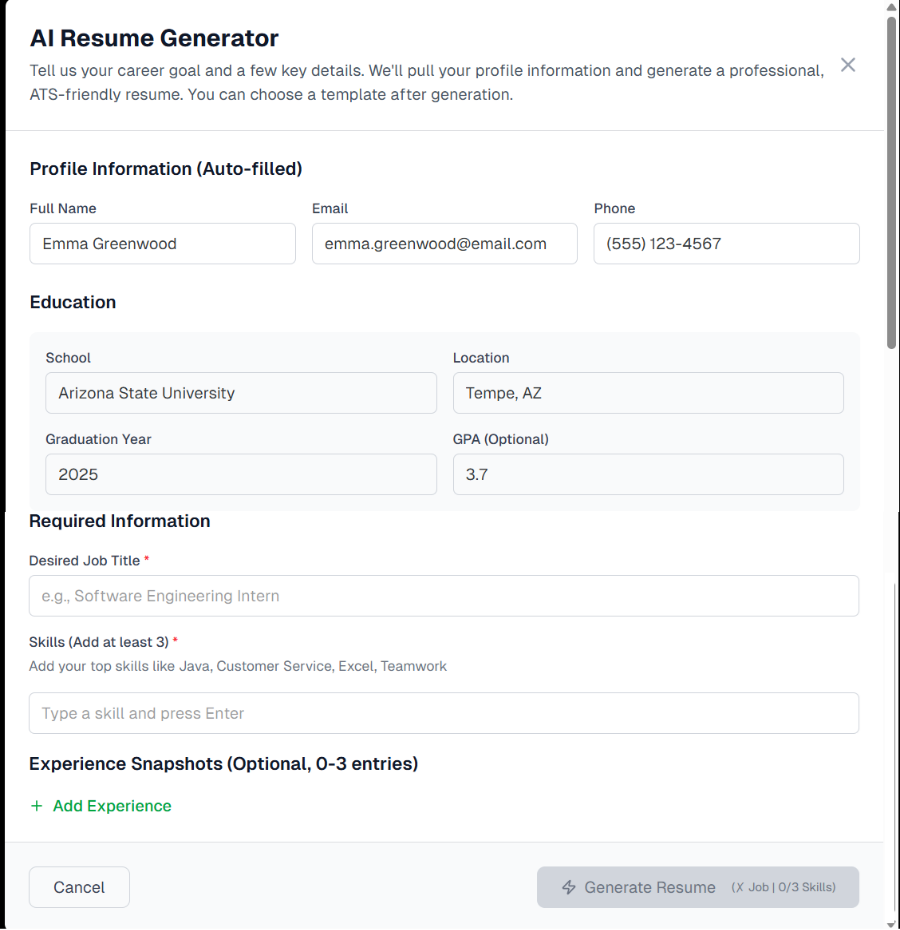


Figure : AI Resume generator pop-up

### Involved APIs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API ID** | **Endpoint** | **Method** | **Description:** | **When is it called?** |
| API-01 | /resumes/aiGenerate | POST | Submits minimal input (education, experience, skills) to generate AI-assisted resume draft. | When student clicks Submit in AI Resume Form. |
| API-02 | /resumes/templates | GET | Fetches available ATS-friendly templates for preview. | When student opens AI Resume Preview or changes template. |
| API-03 | /resumes/{resumeId}/save | POST | Saves the AI-generated resume to student’s profile. | When student clicks Save Resume in preview screen. |
| API-04 | /resumes/{resumeId}/setPrimary | PUT | Marks one resume as Primary. | When student selects “Set as Primary” from Actions column. |
| API-05 | `/resumes/{resumeId}/export?format=pdf | docx | txt` | GET |

# Use Case 14: Upload and Parse an Existing Resume

### Description:

This use case explains how a student can upload their existing resume (PDF or Word) into ScholarPath. After the upload, the system tries to break the resume into different sections like Personal Information, Education, Work Experience, and Skills. The student can then check, edit, and save the resume inside ScholarPath.

### Actors:

* **Primary Actor**: Student (Grades 6–12).
* **Supporting Actor**: Parent/Guardian (view-only access).
* **External Actor**: Document Parsing Service.

### Goal:

Allow students to Upload their existing resumes into ScholarPath so they can edit, save, and use them in a structured format.

### Trigger:

Student clicks Upload Resume from the Resume Builder screen.

### Business Rules:

* Supported file formats: DOCX and PDF (up to 5MB).
* A student can have a maximum of 5 resumes in total (uploaded, created manually, or created with AI).
* Resume names must be unique for each student.
* If the system cannot read everything, the extra information is placed into an “Additional Information” section.
* Uploaded resumes are marked as Secondary unless the student sets one as Primary.
* Parents can view resumes but cannot upload or edit them.

### Pre-Conditions:

* Students are logged into ScholarPath.
* Resume Builder is available.
* Students have fewer than 5 resumes saved.

### Steps:

1. Login as Student.
2. Go to Portfolio → Plan & Build → Resume Builder.
3. Click on Upload Resume.
4. Select a file (PDF or Word) from the computer.
5. The system checks file size, format, and resume limit.
6. File is uploaded.
7. The system reads the file and fills resume details into sections like Personal Information, Education, Work Experience, and Skills.
8. The student reviews and edits the details.
9. The student clicks Save Resume.
10. The resume appears in the Your Resumes list.

### Negative Flow:

* File is larger than 5MB → Show error: “File size must be 5MB or less.”
* Wrong file type → Show error: “Only PDF and Word files are supported.”
* Student already has 5 resumes → Show message: “You have reached the maximum resume limit. Delete one to upload a new one.”

### Post-Condition:

* Resume is saved in Scholar Path.
* Resume shows up in Your Resumes list (Secondary by default).
* Parents can view but not edit resumes.

### Special Requirements:

* System must be accessible (WCAG 2.1 AA).
* Errors must be logged for tracking.
* Audit log must record all upload attempts with student ID, file name, time, and result.

### Constraints:

* The quality of reading depends on how the resume is formatted.
* Only one file can be uploaded at a time.
* The system depends on an external service to read files.

### Screen Element Matrix: (SEM)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen Name** | **Section Name** | **Element** | **Description:** | **Type** | **Options** | **Accepted Values** | **Accepted Size** | **Mandatory (Yes/No)** | **Validations** |
| Resume Builder – Landing | Actions | Upload Resume | Starts the upload flow; opens the Upload Your Resume dialog. Visible to Student role only. | Button | Click | N/A | N/A | Yes | Only Students can initiate. Block if student already has 5 resumes → show “You have reached the maximum resume limit. Delete one to upload a new one.” |
| Upload Your Resume – Dialog | Header | Close | Dismisses the dialog without uploading. | Icon Button (X) | Click | N/A | N/A | No | If a file is mid-upload, confirm cancellation; otherwise close immediately. |
| Upload Your Resume – Dialog | Drop Zone | Drag-and-Drop Area | Bordered area where users can drag & drop a file; also focuses keyboard drop. | Drop Zone | Drag file, Drop file | Single file, PDF/DOCX | ≤ 5MB | Yes | Enforce single-file; reject folders/multiple files. |
| Upload Your Resume – Dialog | Drop Zone | Upload Icon | Decorative upload arrow shown inside the drop zone. | Icon (decorative) | N/A | N/A | N/A | No | Hidden from AT or marked aria-hidden; accessible name provided by instructional text. |
| Upload Your Resume – Dialog | Drop Zone | Supported Formats Helper | Helper subtext: “Supported formats: PDF, DOCX (Max 5MB)”. | Text (helper) | N/A | “PDF”, “DOCX”; max 5MB | N/A | No | Must stay visible for accessibility; communicates Constraints: but does not replace enforcement. |
| Upload Your Resume – Dialog | Progress | Upload Status | Shows upload progress after file selection. | Progress Bar | Auto-updates | 0–100% | N/A | Yes (shown after selection) | Must reflect actual bytes uploaded; displays error state on failure. |
| Upload Your Resume – Dialog | Feedback | Error: Invalid Type | Inline/Toast error when file is not PDF/DOCX. | Alert/Toast | Dismiss | N/A | N/A | No | Message: “Only PDF and Word files are supported.” Prevent upload. |
| Upload Your Resume – Dialog | Feedback | Error: File size exceed | Inline/Toast error when file > 5MB. | Alert/Toast | Dismiss | N/A | N/A | No | Message: “File size must be 5MB or less.” Prevent upload. |
| Upload Your Resume – Dialog | Feedback | Error: Resume Limit Reached | Error shown if student already has 5 resumes. | Alert/Toast | Dismiss | N/A | N/A | No | Message: “You have reached the maximum resume limit. Delete one to upload a new one.” Block flow. |
| Resume Editor – Parsed View | Personal Information | Name | Parsed full name; editable. | Text Input | N/A | Letters, spaces, punctuation | 2–100 chars | Yes | Required; trim spaces; must contain letters. |
| Resume Editor – Parsed View | Personal Information | Email | Parsed email; editable. | Email Input | N/A | Valid email address | 5–254 chars | Yes | Must match email pattern; no leading/trailing spaces. |
| Resume Editor – Parsed View | Personal Information | Phone | Parsed phone; editable. | Tel Input | N/A | Digits, +, (), -, spaces | 7–20 digits (normalized) | Yes | Numeric after normalization; show inline error on invalid phone. |
| Resume Editor – Parsed View | Education | School | Parsed institution name; editable. | Text Input | N/A | Alphanumeric & punctuation | 2–120 chars | Yes | Required; not all-numeric; trim. |
| Resume Editor – Parsed View | Education | Location | Parsed school location; editable. | Text Input | N/A | City, State/Country | 2–100 chars | No | If provided, must include letters; trim. |
| Resume Editor – Parsed View | Education | Graduation Year | Parsed graduation year; editable. | Number/Input | N/A | 4-digit year | 4 chars | No | If provided, numeric and within 1900–2100. |
| Resume Editor – Parsed View | Education | GPA | Parsed GPA; editable, optional. | Number/Input | N/A | 0.0–4.0 | up to 4 chars | No | If provided, numeric; 0.0 ≤ GPA ≤ 4.0; up to one decimal. |
| Resume Editor – Parsed View | Work Experience (repeatable) | Experience Entry | Employer, Role/Title, Location, Start Date, End Date/Present, Description: (rich text). | Field Group | Add/Remove entry | Text/date values | N/A | No (fields within may be required per design) | Validate dates (Start ≤ End); Description: length ≤ 2000 chars; safe formatting only (no images/tables). |
| Resume Editor – Parsed View | Skills | Skills List | Parsed skills as tags; editable (add/remove). | Chips/Tags Input | Add, Remove | Each skill 1–40 chars | Up to 50 skills | No | De-duplicate case-insensitive; trim each tag; disallow empty tags. |
| Resume Editor – Parsed View | Additional Information | Additional Info | Bucket for content the parser couldn’t map to standard fields; editable notes. | Text Area | N/A | Free text | Up to 5000 chars | No | Optional; sanitize input. |
| Resume Editor – Actions | Save Resume | Saves parsed/edited resume to profile (defaults to Secondary). | Primary Button | Click | Resume object with unique name | N/A | Yes | Enforce unique resume name per student; if name conflicts, prompt to rename. If total resumes would exceed 5, block with limit message. |  |

### Screenshot:

A screenshot of a computer

AI-generated content may be incorrect.

Figure : Upload your Resume section

### Involved APIs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API ID** | **Endpoint** | **Method** | **Description:** | **When is it called?** |
| API-UP-01 | /resumes/upload | POST | Uploads the file to the server. | When student selects and uploads a resume. |
| API-UP-02 | /resumes/read | POST | Reads the resume file and fills details. | Right after upload succeeds. |
| API-UP-03 | /resumes/{resumeId}/save | POST | Saves the uploaded resume into the student’s profile. | When student clicks Save Resume. |
| API-UP-04 | /resumes/{resumeId}/setPrimary | PUT | Sets uploaded resume as Primary. | When student chooses “Set as Primary.” |
| API-UP-05 | `/resumes/{resumeId}/export?format=pdf | docx | txt` | GET |

# Use Case 15: Manage Resumes

### Description:

This use case describes how a student manages their resumes within the Resume Builder module. It covers actions such as updating, renaming, setting a resume as primary, deleting secondary resumes, making a copy, and downloading or exporting resumes in ATS-compliant formats (PDF, DOCX, TXT).

### Actors:

* Primary Actor: Student
* Supporting Actors: Parent (view-only)

### Goal:

Enable students to manage multiple resumes effectively, ensuring one is always primary, and allow exporting resumes in standardized formats that are ATS-friendly.

### Trigger:

Student navigates to Resume Builder → Your Resumes Table and clicks the 3-dot menu at the end of a resume entry in the “Your Resumes” table.

### Business Rules:

1. A maximum of 5 resumes can be created per student.
2. Primary Resume: Exactly 1 resume must always be marked as Primary.

* The Primary resume is the default/shared/exported resume unless another is explicitly chosen.
* It cannot be deleted, only edited, renamed, downloaded, copied, or shared.
* Remaining 4 Resumes:
* These are classified as Secondary Resumes.
* A student may have 0 to 4 secondary resumes in addition to the Primary.
* Secondary resumes can be renamed, edited, downloaded/exported, shared, copied, & deleted
* Any Secondary resume can be promoted to Primary via the “Set as Primary” action (making the old Primary a Secondary).

1. Resumes can be exported in PDF, DOCX, or TXT formats.
2. Copying a resume creates a new draft with “Copy” appended to the title. Set as Primary option is only available for non-primary resumes.
3. Sharing a resume requires selecting at least one provider or authorized recipient.
4. Exported resumes must comply with ATS standards:

* Supported formats: PDF, DOCX, TXT.
* No images, tables, or graphics.
* Standard fonts (Arial, Calibri, Times).
* Consistent date formatting (MMM YYYY).
* Bulleted responsibilities only.

1. Download actions should complete within ≤5 seconds for text formats.
2. All actions must be audit-logged with studentId, resumeId, and timestamp.

### Pre-Conditions:

* Student is logged in and authenticated.
* At least one resume exists.
* For delete action: selected resume is not Primary.

### Steps:

1. Student navigates to Resume Builder > Your Resumes.
2. System displays a table listing all resumes with their details (Name, Status, Last Modified, Actions).
3. Student clicks the 3-dot menu beside a resume.
4. System displays the action options:

* Edit → Opens resume editor.
* Rename → Opens inline rename modal.
* Download → Provides export format options.
* Make a Copy → Duplicates resume.
* Set as Primary → Sets resume as Primary (only for non-primary resumes).
* Share → Opens provider selection modal.
* Delete → Deletes resume (if not primary).

1. System validates rules (e.g., cannot delete Primary, resume count ≤5).
2. System completes the requested action and updates the UI.
3. Audit event recorded in the backend.

### Negative Flow:

* If the student tries to delete Primary Resume, the Delete option is disabled for Primary Resume.
* If the student has 5 resumes and attempts to copy/create → System displays: *“Resume limit reached. Please delete one to add another.”*
* If download/export fails → Show error: *“Unable to export resume. Please try again.”*
* If provider sharing fails → Show error: *“Sharing failed. Please check permissions or try again.”*

### Post-Condition:

* Resume actions (edit, rename, copy, share, download, delete) are completed and reflected in the “Your Resumes” table.

### Special Requirements:

* 3-dot menu must be accessible with keyboard navigation (WCAG 2.1 AA).
* Each action should display success/failure toast notifications.
* Resume actions (edit, rename, copy, share, download, delete) are completed and reflected in the “Your Resumes” table.

### Constraints:

* Only ATS-compliant export templates are supported.
* No offline editing; all resume actions occur inside the platform.
* File size limit per resume export: 5 MB.
* “Set as Primary” option not available for the current Primary resume.

### Screen Element Matrix:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen Name** | **Section Name** | **Element** | **Description:** | **Type** | **Options** | **Accepted Values** | **Accepted Size** | **Mandatory (Yes/No)** | **Validations** |
| Resume Editor – Actions | Save Resume | Saves parsed/edited resume to profile (defaults to Secondary). | Primary Button | Click | Resume object with unique name | N/A | Yes | Enforce unique resume name per student; if name conflicts, prompt to rename. If total resumes would exceed 5, block with limit message. |  |
| Resume Editor – Actions | Cancel | Discards changes and returns to previous screen. | Button | Click | N/A | N/A | No | Confirm if there are unsaved edits. |  |
| Your Resumes – Table | Row | Resume Entry | Shows the uploaded/parsed resume with status, created date, and actions. | Table Row | Edit, Set Primary, Delete, Download | Valid resume metadata | N/A | Yes | Exactly one Primary must exist; newly uploaded is Secondary by default. Prevent delete if it is the only Primary. |
| Your Resumes – Table | Row Actions | Set Primary | Marks the resume as Primary. | Button/Link | Click | N/A | N/A | No | On success, previous Primary becomes Secondary. |
| Your Resumes – Table | Row Actions | Delete | Removes the resume. | Button/Link (destructive) | Click → confirm | N/A | N/A | No | Block delete if it’s Primary and the only resume; require confirmation. |
| Your Resumes – Table | Row Actions | Download | Downloads the stored resume (exported). | Button/Link | PDF, DOCX, TXT | Valid export | N/A | No | Must export successfully; log failures. |
| Resume Builder – Landing | Actions Bar | Create New Resume | Creates a blank resume (outside this use case but present on the screen). | Button | Click | N/A | N/A | No | Available to Students only; disabled if total resumes = 5 (show limit message). |
| Resume Builder – Landing | Actions Bar | Generate with AI | Opens AI generator flow (present on screen). | Button | Click | N/A | N/A | No | Available to Students only; disabled if total resumes = 5. |
| Resume Builder – Landing | Actions Bar | Upload Resume | Opens upload dialog (present on screen). | Button | Click | N/A | N/A | No | Available to Students only; disabled if total resumes = 5. |
| Share Resume – Modal | Body | Recipient(s) | Select one or more providers/authorized recipients. | Multi-select / Chips | Search, Add, Remove | Valid recipients from directory | Up to 50 recipients | Yes | Must select ≥1; read-only share (no edit). |
| Share Resume – Modal | Footer | Share | Sends the resume/link to selected recipient(s). | Primary Button | Click | N/A | N/A | Yes | Disabled until ≥1 recipient selected; show error “Sharing failed…” on backend failure. |
| Share Resume – Modal | Footer | Cancel | Closes share modal. | Button | Click | N/A | N/A | No | No changes saved. |
| Download – Options | Body | Format | File type to export. | Radio Group / Menu | PDF, DOCX, TXT | PDF/DOCX/TXT | N/A | Yes | One option must be chosen; export must be ATS-compliant; file ≤5MB; complete within ≤5 seconds for TXT/DOCX. |
| Download – Options | Footer | Download | Generates and downloads the file. | Primary Button | Click | N/A | N/A | Yes | Show error “Unable to export resume. Please try again.” on failure. |
| Delete Resume – Confirm | Body | Confirmation Text | Confirmation message to delete a secondary resume. | Text (static) | N/A | N/A | N/A | Yes | Warn that action is irreversible; identify resume by name. |
| Delete Resume – Confirm | Footer | Delete | Confirms deletion. | Destructive Button | Click | N/A | N/A | Yes | Only allowed if the resume is Secondary; after delete update table; audit log. |
| Delete Resume – Confirm | Footer | Cancel | Aborts deletion. | Button | Click | N/A | N/A | No | Close dialog; no changes. |
| Toast Notification | Feedback | Success/Error Toast | Displays success or failure message after resume sharing | Banner | Success, Error | Varies | N/A | Yes (if triggered) | Shows "Resume shared successfully with [Provider]" or appropriate error message |

### Screenshot:

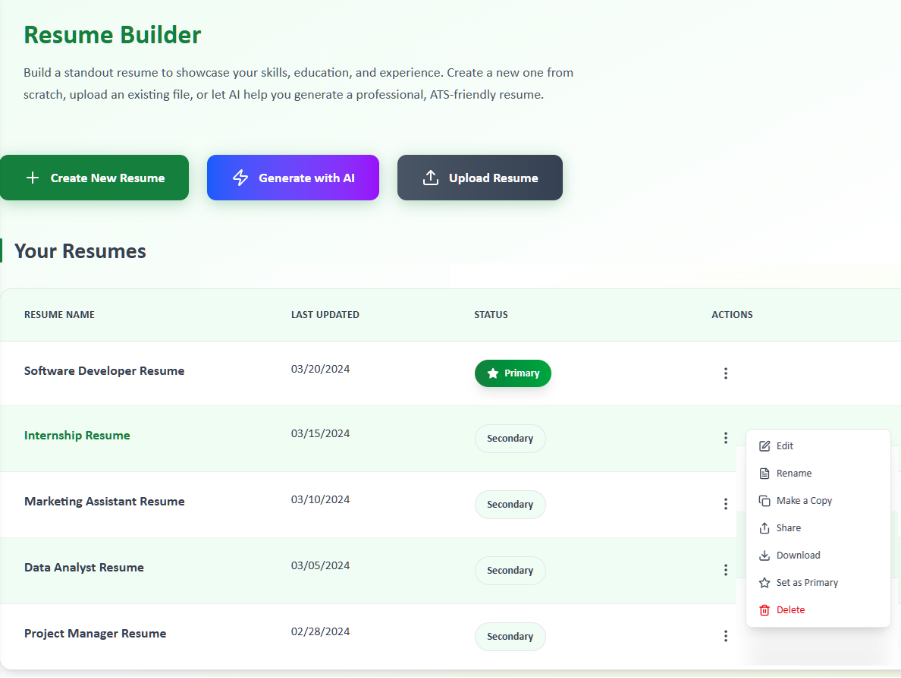


Figure : Manage Resume

### Involved APIs:

|  |  |  |
| --- | --- | --- |
| **API Name** | **Endpoint** | **Description:** |
| GetResumes | GET /api/resumes | Fetches all resumes for student |
| UpdateResume | PUT /api/resumes/{id} | Updates resume details (edit/rename) |
| DownloadResume | GET /api/resumes/{id}/download | Downloads resume in selected format |
| CopyResume | POST /api/resumes/{id}/copy | Creates a duplicate of the resume |
| ShareResume | POST /api/resumes/{id}/share | Shares resume with providers |
| DeleteResume | DELETE /api/resumes/{id} | Deletes resume (if not primary) |

# Use Case 16: Share Resume with Providers

### Description:

This use case allows students to securely share their resumes with providers (Colleges, Businesses, Armed Services) through ScholarPath’s messaging system. Resumes can only be shared under specific eligibility conditions to ensure compliance and relevance.

### Actors:

* Primary: Student
* Supporting: Parent/Guardian (consent/permissions), Provider (recipient)

### Goal:

To enable students in Grades 9–12 to share their resumes with providers they have previously interacted with.

### Trigger:

The student clicks the Share option from the 3-dot menu on a resume in the Your Resumes table.

### Business Rules:

1. Parental Permission Check:

* Resume sharing is only available if parental permission for messaging is enabled.

1. Grade Level Restriction:

* Only students in Grades 9–12 are eligible to share resumes.

1. Provider Interaction Requirement:

* The student must have previously exchanged messages with the provider.
* Only providers with existing conversation threads are displayed in the list.

1. Format:

* Resumes are shared as ATS-compliant PDFs (generated/exported on the fly).

1. Audit Logging:

* Each share event must be logged with studentId, resumeId, providerId, timestamp, and consentFlag.

### Pre-Conditions:

* Student is logged in and authenticated.
* At least one resume exists in the system.
* Student is in Grades 9–12.
* Parental permissions for messaging are enabled.
* At least one provider exists with whom the student has an active message history.

### Steps:

1. Student navigates to Resume Builder → Your Resumes.
2. Student clicks the 3-dot menu for a resume and selects Share.
3. System opens the Share Resume Modal.
4. System validates eligibility:

* Confirms student is in Grades 9–12.
* Confirms parental permission for messaging is enabled.
* Filters the provider list to only those with existing conversations.

1. Student selects a provider from the list.
2. Student optionally enters a short note.
3. System generates an ATS-compliant PDF of the selected resume.
4. System attaches the PDF and sends it through the existing message thread.
5. Audit event is logged.
6. Student sees success toast: *“Resume shared successfully with [Provider Name].”*

### Negative Flow:

* Missing parental permission → Block sharing. Show: *“Messaging not enabled by your* parent/guardian.”
* Student not in Grades 9–12 → Block sharing. Show: *“Resume sharing is available only for Grades 9–12.”*
* No provider interaction → Provider list empty. Show: *“You can only share resumes with providers you’ve messaged before.”*
* Export/attachment error → Show: *“Couldn’t generate resume PDF. Try again.”*
* Network/API error → Retry prompt.
* Export failure (PDF generation) → Show “Couldn’t generate PDF. Try again.”; log error; allow retry.
* Ratelimit exceeded → Show “You’ve reached the share limit. Try again later.”
* Auth/session expired → Prompt reauthentication; return to modal after login.

### Post Conditions

* Resume is shared with the selected provider via the existing message thread.
* Event captured in audit log; student sees confirmation and can open the conversation.

### Special Requirements:

* Accessibility: Modal, search, and preview are keyboardnavigable with ARIA labels.
* Localization: All labels/messages support i18n.
* Performance: Provider search results < 300 ms (99p); export uses caching where safe.
* Observability: Emit telemetry (share\_started, share\_succeeded, share\_failed) with error codes.

### Constraints:

* Student must be in Grades 9–12.
* At least one existing provider conversation is required.
* Parental messaging permissions must remain enabled.
* Resume sharing is strictly in PDF format.

### Screen Element Matrix:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen Name** | **Section Name** | **Element** | **Description:** | **Type** | **Options** | **Accepted Values** | **Accepted Size** | **Mandatory (Yes/No)** | **Validations** | **Comments** |
| Your Resumes (Table) | Row actions | 3-dot Menu (⋮) → Share | Opens the Share Resume modal for the selected resume. | Button / Icon | Edit, Rename, Download, Share, Delete | — | — | — | Share action available for any resume (Primary or Secondary); Delete disabled for Primary. | Entry point to this flow. |
| Share Your Resume (Modal) | Header | Modal Title | “Share Your Resume”. | Text | — | Fixed string | — | Yes | — | Title per mock. |
| Share Your Resume (Modal) | Header | Selected Resume | Shows the selected resume name and file type/primary badge. | Label / Card | — | Resume display name; status (Primary/Secondary) | — | Yes | Value pulled from Your Resumes table. | Matches the green resume card in UI. |
| Share Your Resume (Modal) | Provider Selection | Select a Provider | Dropdown to choose a provider with whom the student already has an active conversation. | Dropdown | List of eligible providers | ProviderId from existing message threads | — | Yes | List is pre-filtered to providers with existing conversations; disabled/empty if none. | Also checks eligibility: Grades 9–12 and parental permission for messaging. |
| Share Your Resume (Modal) | Optional Note | Message Box | Optional short note to accompany the shared resume. | Textarea | — | Free text | ≤ 500 chars | No | Character limit enforced; block over-limit input. | Include basic profanity/PII filters if required by policy (non-blocking). |
| Share Your Resume (Modal) | Footer | Send Resume | Sends the selected resume as ATS-compliant PDF to the chosen provider via existing message thread. | Primary Button | — | — | — | Yes (conditional) | Enabled only when: (1) Grades 9–12; (2) parental messaging permission enabled; (3) provider selected. On click: generate ATS PDF and attach; log audit event (studentId, resumeId, providerId, timestamp, consentFlag). | Sends within the existing conversation; PDF generated/exported on the fly. |
| Share Your Resume (Modal) | Footer | Cancel | Closes the modal without sending. | Secondary Button | — | — | — | No | Simply dismisses modal; no state change. | — |
| Share Your Resume (Modal) | System | ATS PDF Export | Server-side generation of an ATS-compliant PDF of the selected resume. | System action | — | PDF only | ≤ 5 MB (export cap) | — | Must succeed to send; on failure show: “Couldn’t generate resume PDF. Try again.” and allow retry. | Export uses caching where safe; ATS rules enforced. |
| Share Your Resume (Modal) | System | Eligibility Gate | Checks grade & permission before enabling sharing. | Rule | — | Grade: 9–12; consentFlag: true | — | — | If not eligible: show messages: (a) “Messaging not enabled by your parent/guardian.” (b) “Resume sharing is available only for Grades 9–12.” | Runs when modal opens and on Send click. |
| Share Your Resume (Modal) | System | Provider Filter | Populates dropdown with only providers that have active message threads with the student. | Rule | — | ProviderId list | — | — | If none: dropdown shows empty state and message: “You can only share resumes with providers you’ve messaged before.” | List fetch via /api/providers/list. |
| Share Your Resume (Modal) | Notifications | Toasts | Success and failure banners. | Banner | Success / Error | Text | — | — | Success: “Resume shared successfully with [Provider Name].” Errors: export/network/rate-limit/auth messages per BR/NF. | Rate-limit: “You’ve reached the share limit. Try again later.” Auth expired: re-authenticate then return to modal. |

### Screenshot:

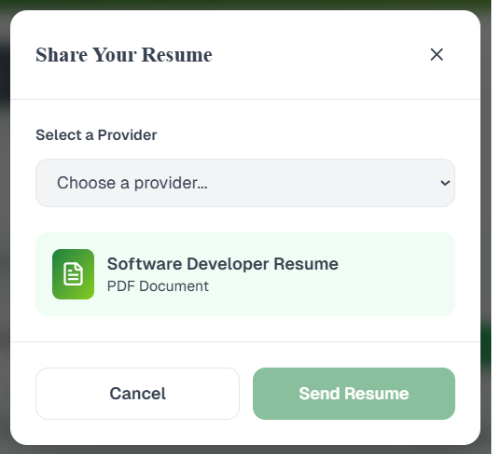


Figure : Share Resume

### Involved APIs:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Name** | **Endpoint** | **Method** | **Description:** | **When is it Called?** |
| GetProvidersList | /api/providers/list | GET | Fetches a list of active and authorized providers (College, Business, Armed Services). | When student opens “Share Resume” modal. |
| GetResumeList | /api/resumes | GET | Retrieves all resumes uploaded, created, or generated by the student. | On modal load to display available resumes. |
| ShareResume | /api/resumes/share | POST | Shares a selected resume with chosen provider(s); stores audit log. | On clicking **Share** button. |
| RevokeResumeShare | /api/resumes/revoke | POST | Revokes access previously granted to a provider. | When student withdraws a shared resume. |
| GetSharedResumes | /api/resumes/shared/:providerId | GET | Retrieves the list of resumes already shared with a specific provider. | On provider profile view. |
| LogShareActivity | /api/audit/resume-share | POST | Records the transaction in compliance logs (studentId, providerId, timestamp). | After resume is successfully shared. |