



PayPulse Analytics
uNext payment intelligence

PRODUCTION FEATURE & TECHNICAL SPECIFICATION

Complete Application Features, Rules, Workflows and Operational Specification

A functional and technical reference for the SMU and MUJ payment analytics platform.

Document date	10 September 2026
Production URL	https://paypulse-unext.replit.app
Deployment	Healthy Replit Autoscale deployment; password-protected visibility
Technology	React + Vite frontend; Node.js + Express API; TypeScript; PostgreSQL; private object storage
Scope	Current application implementation and production deployment configuration

Document status and verification boundary

Production status verified. Replit reports an active Autoscale deployment with a successful current build at <https://paypulse-unext.replit.app>. The deployment is protected by a Replit password gate.

Because unauthenticated requests are redirected to the private-deployment login screen, this document does not claim to have bypassed production access controls. Detailed behavior is specified from the current deployed project implementation and verified development workflows. Items are labeled as follows:

- Production verified** deployment existence, URL, type, successful build and password protection.
- Implemented** behavior present in the current application source and API.
- Limitation** an intentional hold, operational dependency, or known boundary.

This is an application specification, not a data audit. Record counts, financial totals and active dataset contents can change when authorized users activate new source files or edit source records.

Contents

1 Product overview	11 Business calculations
2 Architecture and deployment	12 USD/INR conversion
3 Access and security	13 Exports
4 Navigation and organizations	14 SLCM automation
5 Dashboard and KPIs	15 API specification
6 Filters	16 Performance and caching
7 Charts and program analysis	17 Validation and quality
8 Learner records and detail	18 Known limitations
9 Source Data management	19 Glossary
10 Replace and Append workflows	

1. Product overview

PayPulse Analytics is a payment and re-registration analytics application for two university organizations: Sikkim Manipal University (SMU) and Manipal University Jaipur (MUJ). It combines operational payment-source management with public-facing analytics inside one web application.

Primary outcomes - Monitor collected, gross and outstanding fee values. - Track learners and re-registration activity. - Analyze performance by program, batch, semester, status and benefit category. - Normalize USD payments to INR with auditable historical-rate rules. - Export filtered records and program summaries.	Data governance model - Uploaded files are archived before activation. - Archived files do not affect analytics until activated. - Active datasets are separate for SMU and MUJ. - Replace and Append create a new dataset version. - Source edits and soft deletions are scoped to that dataset version.
--	--

2. Architecture and deployment

Layer	Specification
Frontend	React single-page application built with Vite and TypeScript. React Query manages server state. Responsive layouts adapt tables and cards for smaller screens.
Backend	Node.js runtime with an Express API written in TypeScript. The API is routed under <code>/api</code> .
Database	PostgreSQL stores users/sessions, upload metadata, per-organization active dataset state, dataset-scoped source changes, report automation history and persisted FX/calendar snapshots.
File storage	Private object storage holds archived CSV files, generated files and diagnostic screenshots. Signed upload/download URLs are time limited.
Analytics engine	CSV-derived records are normalized and held in process-local caches. Active multi-file datasets are materialized to temporary CSV files for efficient reads.
Production	Production verified Replit Autoscale, successful build, HTTPS URL, password-protected deployment.

Autoscale means the service can scale down while idle and start again on demand. Persistent state therefore resides in PostgreSQL and object storage, while in-memory and temporary-file caches are rebuilt after a cold start.

3. Access and security

3.1 Deployment access

The complete published site is protected by Replit’s password gate. An unauthenticated browser sees “Access Private Deployment” before it can reach the dashboard or API.

3.2 Application authentication

- Application sign-in uses Replit OpenID Connect with PKCE, state and nonce protections.
- Authentication requests standard identity scopes: OpenID, email and profile, with offline access support.
- Sessions are stored in PostgreSQL and represented by secure, HttpOnly, SameSite=Lax cookies.
- Expired sessions attempt token refresh where available.

3.3 Feature access

- Analytics pages are designed for broad viewing once deployment access is passed.
- Source Data upload, activation, editing, deletion, restoration and automation actions require an authenticated application user.
- Organization inputs are restricted to the two configured organization identifiers.

Authorization limitation: the current application checks authenticated-user presence for protected Source Data operations but does not implement granular roles, organization membership checks, or administrator-only permissions.

4. Navigation and organization behavior

Route	Purpose
<code>/</code>	Overview dashboard, KPIs, program analysis, insights charts and learner records.

Route	Purpose
/source-data	Source file status, archived uploads, activation controls, SLCM automation and editable source records.
Unknown route	Visible 404 page.

The sticky header shows PayPulse Analytics branding, a Source Data link and an SMU/MUJ switcher. SMU is the default organization. Filter state is maintained separately per organization, so switching organizations restores that organization’s staged and applied filters.

5. Dashboard and KPI features

The overview page presents live payment performance for the currently applied filter set. KPI tiles are configurable and grouped into financial, count and rate categories. User choices persist locally in the browser.

KPI	Meaning
Net Semester Fee	Total collected amount after USD records are converted to INR.
Gross Settlement Fee	Total semester fee value in INR for the current result set.
Outstanding	Uncollected balance represented by amount due.
Total Students	Distinct learner count for the filtered result.
Total Re-reg Done	Sum of paid and partially-paid payment-status records under the application’s re-registration rule.
Collection Rate	Collected value represented as a percentage of the corresponding gross value.
Total Discounts	Converted discount amount over the filtered records.
SMU/MUJ comparison	Collected amount and learner counts by organization when an all-organization context is used.
Status count cards	Dynamic count cards generated from payment statuses present in the data.

Alerts identify records using a user-provided fallback, the static fallback, own-term semester dates, or prior-semester reference dates. This makes conversion exceptions visible rather than silently blending them into totals.

6. Filtering specification

Filters are staged first and applied together using **Apply Filters**. Pending selections do not change the dashboard until applied. Reset clears the active filter set. Categorical filters support multi-selection, option search, and a special **Not defined** value.

6.1 Pinned filters

- Currency
- Purpose
- Form Type
- Batch
- Semester Paid For
- Program
- Payment Status
- Payment Date Upto
- USD/INR conversion control and snapshot status

6.2 Additional categorical filters

- Current Batch
- Current Semester
- Enrollment Status
- Fee Type
- Benefit Category

6.3 Text filters

Application Number, Student Name, Roll Number, Email, Term Name and Mentor Name support contains, starts with, ends with and does not contain operators.

6.4 Numeric ranges

Minimum/maximum filters are available for Amount Paid in INR, source Amount Due, source Semester Fee, source Discount Amount and Discount Percentage.

6.5 Payment Date Upto

- The boundary is inclusive.
- Date comparison uses the IST calendar day.
- The record payment date is its last populated installment payment date.
- Zero-value enrollment records can use term-created date under their enrollment rule.
- “Not defined” returns records with no applicable payment date.

6.6 Visibility rules

- Analytics excludes rows with no enrolled term.
- SMU analytics is limited to its approved program allowlist.
- MUJ data is not restricted by the SMU program allowlist.

7. Program analysis and insights charts

7.1 Program-wise breakdown

The sortable program table includes Program Group, Program Name, Students, Total Re-reg Done, Paid, Partially Paid, Gross Fee (INR) and Net Fee. A totals footer summarizes the visible result. Selecting a program name applies that program as a filter. On smaller screens the table becomes program cards.

The three approved Master of Arts programs are grouped together and retained in a requested fixed order before the remaining programs.

7.2 Configurable charts

Chart	Visualization	Metric modes
Payment Trend	Time-series area chart	Amount or count
Payments by Batch	Bar chart	Amount or count
Payments by Semester / Paid For	Bar chart	Amount or count
Top 10 Programs	Horizontal bar chart	Amount or count
Payment Status	Donut chart	Amount or count
Benefit Categories	Donut chart	Amount or count

Chart visibility preferences persist in the browser. When both organizations are represented, charts can split SMU and MUJ series; a selected organization shows a total series.

8. Learner records and learner detail

8.1 Learner records

- Server-side pagination, normally 50 records per page.
- Debounced search over name, roll number and email-related identifiers.
- Sortable columns and sticky table headers.
- Column visibility and order controls with drag, move arrows and reset.
- Column preferences persist in local browser storage.
- Loading shimmer/progress behavior preserves the previous result while a new page or filter loads.

8.2 Learner detail panel

Selecting a learner opens a responsive right-side detail sheet. It shows identity, organization, program, roll number, email and payment history by semester. Payment rows include status, fee type, currency, paid amount, due amount, discounts and up to three installment amounts, dates and statuses.

8.3 USD audit trail

USD rows expose the source amount, INR amount, conversion factor, rate source/date, reference date/source, settlement date, RBI calendar city, skipped non-working dates and calendar provenance. Native INR records are identified separately.

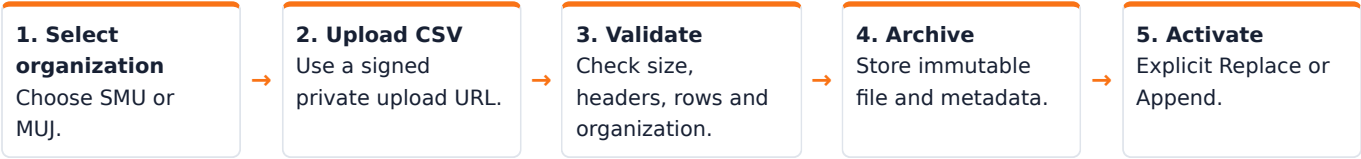
9. Source Data management

The Source Data page combines file lifecycle controls and row-level maintenance.

9.1 Status cards

- Current source file or composition type
- Active rows versus total rows
- Soft-deleted row count
- Last imported timestamp

9.2 Upload and archive



Archived uploads show filename, organization, size, validated row count, upload date and Active/Archived state. Users can download the original archived file.

9.3 Source records

- Search, pagination and “show deleted” control.
- Provenance, learner, program/term, batch/semester, financial and status columns.
- Edit selected source fields using a modal.
- Soft delete with confirmation.
- Restore a deleted row.
- Changes affect only the active dataset version and do not rewrite the archived CSV object.

10. Replace and Append activation

Mode	Behavior	Protections
Replace	The selected archived upload becomes the sole active source file for that organization. A new active dataset identifier is created.	Optimistic dataset check prevents a stale request from replacing a dataset that changed concurrently.
Append	The selected archived upload is added to the organization’s current active source composition. A new dataset identifier is created.	Rejects an already-active file, missing/ambiguous payment identifiers, duplicates inside the incoming file and duplicates against active data.

Both modes clear derived analytics caches, rebuild active source materialization and capture a new conversion snapshot. Archived uploads remain immutable and inactive uploads never alter analytics.

Payment identity requirement for Append: each incoming row must have one unambiguous `_id` or `recordId` . Row-position fallback IDs are not accepted as reliable append identities.

11. Business calculation rules

Field / metric	Rule
Last installment payment date	The last non-empty field in source order across <code>installments[0].paymentDate</code> through <code>installments[3].paymentDate</code> . This is field-order selection, not an independent maximum-date sort.
Payment / Enrollment Date	For zero-paid, zero-due credit enrollment records: term-created date. Otherwise: last installment payment date, falling back to term-created date.
Discount percentage	Sum of positive discount percentage values across discount slots 0-3.
Discount amount	Combined discount percentage multiplied by source semester fee.
Credits Applied	For eligible zero-value enrollment or paid-balance-mismatch records: source semester fee minus discount minus source paid amount, then converted to INR when required.
Total Re-reg Done	Sum of records whose normalized payment status is Paid or Partially Paid.
Program student count	Distinct learner identity based on roll number, application number, then record ID fallback.
Program learner payment state	Each learner contributes its clearest state, ranked Paid above Partially Paid above None.

12. USD-to-INR conversion specification

12.1 Reference date priority

1. **Last populated installment payment date** from installment slots 0–3.
2. If missing and Paid For semester is S1: the record's own `termCreatedOn`.
3. If missing and Paid For semester is S2 or later: the uniquely matched learner's immediately previous paid-for semester `termCreatedOn`.
4. If unresolved: record `createdAt`.
5. If still undated: no historical date; use the explicit fallback-rate path.

Dates are normalized to an IST calendar date. Prior-semester matching is organization-scoped. All populated learner identifiers must agree; ambiguous or conflicting matches fail closed instead of choosing an uncertain date.

12.2 T+1 settlement date

Installment, own-term and prior-semester references use a banking settlement date beginning at reference date + 1 calendar day. The app advances past:

- Sundays
- Second and fourth Saturdays
- RBI holidays for the applicable city: **Gangtok for SMU** and **Jaipur for MUJ**

The resulting settlement date is used for rate lookup. A `createdAt` fallback is outside the T+1 banking-calendar rule and uses its own IST date directly.

12.3 Rate lookup priority

1. User-entered global override, if valid.
2. Exact Google Finance historical USD/INR rate for the settlement/reference date.
3. Nearest prior available Google Finance market date.
4. User-entered unresolved-record fallback rate.
5. Static fallback of ₹84 per USD.

12.4 Dataset-locked snapshot policy

Stable conversion rule: Google Finance and RBI are polled only when an archived file is activated using Replace or Append. The resulting history and required calendars are persisted with the active dataset and reused unchanged until the next activation.

- Archiving an upload does not poll.
- Opening the dashboard does not poll.
- Applying filters, viewing learners and exporting do not poll.
- Server restart reloads the PostgreSQL snapshot and does not poll.
- Manual refresh is removed and the API rejects manual refresh attempts.
- The rate snapshot has no time-based expiry.

12.5 Audit fields

Conversion outputs include rate, rate date, rate source, reference date/source, settlement date, calendar city, source URL, calendar provenance and skipped-date reasons.

13. Export capabilities

13.1 Learner CSV

- Export the current filtered selection.
- Choose export fields and persist field preferences.
- Complete export can include original source fields plus conversion/audit columns.
- UTF-8 BOM supports Excel compatibility.
- Date-like values are formatted in IST.
- Formula-like text beginning with =, +, -, or @ is neutralized to reduce CSV formula injection risk.
- Large complete exports are streamed with backpressure rather than retained fully in application memory.

13.2 Program summary XLSX

- Program group/name, learners, re-registration, Paid/Partially Paid, gross and net values.
- Totals row and Excel autofilter.
- Conversion notes identify historical, override and fallback usage.
- Reference-date and RBI settlement explanations are embedded.
- Fallback exposure is visibly highlighted.

14. SLCM report automation

The backend contains Playwright-based automation that can authenticate to configured SMU/MUJ SLCM portals, generate the Payment Semester Level Report, poll until available, download it, validate it, archive it and replace the corresponding organization’s active dataset.

Capability	Current status
Run history	Persisted as queued, running, succeeded or failed, with step, actor, upload/dataset identifiers, errors and timestamps.
Concurrency	Only one active run per organization is allowed.
Diagnostics	Failure screenshots can be stored privately and downloaded.
Schedule text	The UI states a daily 6:00 AM IST schedule.
SMU manual action	“Run SMU” is available when authenticated and no active run blocks it.
MUJ action	On hold MUJ automation is disabled in the UI.
Both organizations	On hold until MUJ automation resumes.

Operational limitation: unattended OTP or CAPTCHA is not supported. Portal selector changes, report-generation delays, browser availability and source report-column compatibility can interrupt automation. The earlier operational decision to pause unreliable amount mapping should be respected before enabling MUJ or unattended combined runs.

15. API specification summary

Endpoint group	Main operations
/api/auth	Current user, login/callback and logout session lifecycle.
Analytics summary	GET /api/payments/summary
Dimensions and fields	GET /api/payments/dimensions
Learner records	GET /api/payments/records with paging and sorting; GET /api/payments/student/:rollNumber .
Grouped analytics	/by-batch , /by-semester , /by-program , /by-status , /by-benefit-category , /trend , /program-summary .
Conversion status	GET /api/payments/exchange-rates/status . Manual refresh POST is intentionally rejected.
Exports	GET /api/payments/download CSV and GET /api/payments/report.xlsx .
Uploads	Request signed URL, archive validated CSV, list history, download archive and activate Replace/Append.
Source records	List/search, PATCH edit, DELETE soft-delete and POST restore.
Automation	List/enqueue report runs and download diagnostics.

Analytics endpoints share the same filter vocabulary. Record paging defaults to 50 with a server maximum of 200. Invalid organizations, malformed uploads, duplicate append identities, activation conflicts and missing learners return explicit HTTP errors.

16. Performance, caching and consistency

- Active-dataset state is cached briefly and concurrent loads are deduplicated.
- Normalized active records and enrolled analytics records are shared across endpoints.
- One applied filter result is reused by summary, records, program, grouped chart and trend requests.
- Converted analytics are cached per active record generation and conversion controls.
- Filter keys are canonicalized so paging, sort and chart presentation parameters do not force duplicate full-data preparation.
- Cache invalidation occurs after dataset activation and Source Data edit/delete/restore.
- External Google/RBI calls never occur during ordinary analytics requests under the snapshot policy.
- Complete exports stream from the materialized source representation to avoid retaining a duplicate full raw dataset in cache.

A cold Autoscale instance may still need to rebuild process-local CSV and derived caches. Warm filter requests are significantly faster because all dashboard panels reuse prepared results.

17. Validation, testing and reliability controls

17.1 CSV validation

- Maximum upload size: 200 MB.
- Required headers: orgId , amountPaid , currency , termName , semester , currentSemester .
- Rejects empty files, missing headers, malformed parsing, unknown organizations and selected-organization mismatches.

17.2 Concurrency and consistency

- Activation is serialized within the application process.
- Database updates include the expected current dataset ID to detect concurrent changes.
- Materialized active CSV replacement uses a temporary file and atomic rename.
- Exports recheck source-file identity to avoid mixing data generations.

17.3 Automated checks

The API test suite covers filtering, date boundaries, semester and settlement logic, Google rate precedence, RBI parsing, program summaries, exports, upload validation, append duplicates, dataset behavior and authentication fallback. Dashboard tests cover filter-state serialization and behavior. TypeScript checks run for both frontend and API.

18. Known limitations and operational dependencies

1. **Production inspection boundary:** the published app is password-protected; this document verifies the deployment metadata but does not bypass the password gate.
2. **Cold starts:** Autoscale restart can require CSV rematerialization and cache rebuild.
3. **Role model:** protected Source Data functions require authentication but do not yet have granular role-based permissions.
4. **Source-change history:** row changes are current JSON patches per dataset, not a full immutable edit audit trail with editor identity for every revision.
5. **Archive retention:** archived uploads do not currently expose a deletion or retention-policy workflow.
6. **Append identity:** correctness depends on stable `_id` or `recordId` values supplied by the source.
7. **SLCM automation:** MUJ and combined actions are on hold; OTP/CAPTCHA and portal UI changes require human intervention.
8. **External source dependency:** activation-time Google/RBI polling uses undocumented Google Finance chart behavior and the RBI public holiday matrix. Explicit fallbacks protect calculation availability.
9. **Error presentation:** some generic dashboard query failures may appear as loading/empty states rather than a highly specific user-facing error panel.
10. **Accessibility:** semantic labels, focus states and sortable-table metadata exist, but no formal WCAG audit is represented by this document.

19. Glossary

Term	Definition
Archived upload	A validated immutable CSV stored privately but not necessarily used for analytics.
Active dataset	The organization-specific source-file composition currently powering analytics.
Replace	Activate one archived file as the organization's complete source.
Append	Add one archived file to the existing active source composition after duplicate checks.
Reference date	The source-derived date selected for an exchange-rate transaction.
Settlement date	The T+1 working date after applying weekend and RBI-holiday rules.
Conversion snapshot	Persisted Google Finance history and RBI calendars fixed for an active dataset generation.
Fallback rate	User-provided or static ₹84 rate used when no dated historical rate can be resolved.

Term	Definition
Paid For semester	The semester represented by <code>currentSemester</code> for conversion and filtering logic; distinct from similarly named learner academic-semester fields.
Soft delete	Hide a source record from active analytics while retaining it for restoration.

Prepared for PayPal Analytics from the current project implementation and verified Replit deployment metadata. Document date: 10 September 2026. Production URL: <https://paypal-unext.replit.app>.