

# Breslov Academy LMS

Coding Book v9.1 — Source of Truth for Implementation

Generated: 2026-01-01 18:41 (local)

This PDF is a print-friendly rendering of the Coding Book ZIP (v9.1). It is intended to be used as an implementation reference in future coding chats.

# Contents

<b>Placeholder for table of contents</b>	<b>0</b>
------------------------------------------	----------

## Included Files

- Breslov\_Academy\_Coding\_Book\_v9\_1/CODING\_BOOK\_V9\_1.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/00\_START\_HERE.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/09\_FUTURE\_BUILDS\_CONTINUOUS\_IMPROVEMENT.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/KNOWN\_ISSUES\_TECH\_DEBT.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/PROMPT\_FOR\_NEXT\_CHAT.txt
- Breslov\_Academy\_Coding\_Book\_v9\_1/REFERENCES.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SECURITY\_NOTES.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/PR\_SERIES\_V9\_1/PR-001\_PUBLIC\_SITE.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/PR\_SERIES\_V9\_1/PR-002\_AUTH\_ENTRY\_ROUTES.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/PR\_SERIES\_V9\_1/PR-003\_PORTAL\_SHELLS\_AND\_RESOLVER.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A1\_REPO\_TREE\_src\_depth3.txt
- Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A2\_FEATURE\_CATALOG.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A3\_ROUTE\_MAP.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A4\_ENTITIES\_TENANCY\_SCOPING.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A5\_KEY\_FILES.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/00\_INDEX.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/ARCHITECTURE\_NORTH\_STAR.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/BACKLOG\_STORIES.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/BRANDING\_TERMINOLOGY\_POLICY.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/CI\_CD\_PIPELINE.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/CMD\_PALETTE\_SPEC.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/DESIGN\_SYSTEM\_UI\_UX.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/FEATURE\_REGISTRY\_AND\_VAULT.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/FUTURE\_BUILDS\_PLAYBOOK.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/GITHUB\_SETUP\_WORKFLOW.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/IA\_ROUTES\_PORTALS.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/KNOWN\_ISSUES\_TECH\_DEBT.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/MIGRATIONS\_DATA\_MODEL.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/NEXT\_CHAT\_CONTEXT.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/OPS\_INTEGRITY\_QA.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/PORTALIZATION\_BUILD\_SCRIPT\_V9\_1.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/PORTAL\_FEATURE\_MATRIX.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/PR\_CHECKLIST.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/QA\_RELEASE\_CHECKLISTS.md
- Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/RBAC\_TENANCY\_MODEL.md

- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/README.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/RELEASE\\_TRAINS\\_FUTURE\\_BUILDS.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/RELEASE\\_TRAIN\\_CONTINUOUS\\_BUILDS.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/REPO\\_REALITY\\_AUDIT.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/ROADMAP\\_V9\\_DETAILED.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/SECURITY\\_THREAT\\_MODEL.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/STORE\\_MONEY\\_SUBSCRIPTIONS.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_AUTH\\_ONBOARDING\\_REALITY.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_CODEBASE\\_REALITY\\_MAP.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_CONTENT\\_PROTECTION\\_REALITY.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_FEATURE\\_REGISTRY\\_REALITY.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_PUBLIC\\_SITE\\_AND\\_PORTALS\\_PLAN.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_QA\\_RELEASES.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9 QUIZZES\\_REALITY.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_ROADMAP\\_EXTREME\\_DETAIL.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_ROUTES\\_REALITY\\_MAP.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_TENANCY\\_REALITY.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/SPECPACK\\_V5/V9\\_UI\\_UX\\_REVAMP\\_PLAN.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/STABLE\\_ROOT\\_DOCS/HEALTHCHECKS.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/STABLE\\_ROOT\\_DOCS/README.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/STABLE\\_ROOT\\_DOCS/ROADMAP\\_CODEX.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/STABLE\\_ROOT\\_DOCS/ROADMAP\\_STATUS.md](#)
- [Breslov\\_Academy\\_Coding\\_Book\\_v9\\_1/STABLE\\_ROOT\\_DOCS/SECURITY\\_INVARIANTS.md](#)

## Part I — Coding Book Core

### CODING\_BOOK\_V9\_1.md

Breslov\_Academy\_Coding\_Book\_v9\_1/CODING\_BOOK\_V9\_1.md

## Breslov Academy LMS — Coding Book (v9.1)

Generated: 2026-01-01 Inputs: stable repo ZIP + Spec Pack v5 text docs. Purpose: act as a \*single, extremely detailed\* build manual for subsequent coding chats.

---

### 1. Absolute Rules (Do Not Break)

#### 1 No features removed.

Any re-architecture must preserve all existing pages, routes, and capabilities, even if relocated behind portal shells.

#### 2 Tenant scoping everywhere

Any entity listed as school-scoped must always be queried/created/updated with `school_id` scoping. See:

- APPENDICES\_GENERATED/A4\_ENTITIES\_TENANCY\_SCOPING.md
- `src/components/api/scopedEntities.js`
- `src/components/api/tenancyEnforcer.js`

#### 3 Content protection is a hard security invariant

If a lesson is `LOCKED` or `DRIP_LOCKED`, the app must not fetch protected materials (Text/Transcript/Download/etc). Enforcement path:  
`src/components/hooks/useLessonAccess.jsx` + policy entities + UI gating.

#### 4 Server-side enforcement beats UI

UI gating is required for UX, but never assume UI gating equals security. Use Base44 permissions/policies/functions for “real” enforcement. Client adds defense-in-depth.

#### 5 Iterate PR-by-PR

Every PR must include: what changed, file paths, test commands, next steps. Update known-issues docs continuously.

## 2. Repo Reality (v9.0 Stable) — What You Actually Have

### 2.1 Tech stack snapshot (from `package.json`)

- Runtime: React + Vite
- Router: `react-router-dom`
- UI: TailwindCSS + Radix components
- SDK: `@base44/sdk` via `src/api/base44Client.js`
- Forms/Validation: `react-hook-form`, `zod`
- Themes: `next-themes`
- Charts: `recharts`

Scripts:

- `npm run dev`
- `npm run build`
- `npm run preview`
- `npm run lint`
- `npm run typecheck`

### 2.2 Entrypoints / Routing / Layout

- App bootstrap: `src/main.jsx`
- Route table: `src/App.jsx`
- Page registry: `src/pages.config.js`
- Current global shell: `src/Layout.jsx`

Routing model today

- 1 Explicit routes in `src/App.jsx`:
  - Quiz routes (`/my-quizzes`, `/quiz/:quizId`, `/teach/quizzes*`)
  - Storefront routes (`/s/:schoolSlug/*`)
- 2 Generated exact-case routes for every page in Pages:
  - `/${PageName}` (e.g. `/Dashboard`)
- 3 A catch-all resolver route:

- `/:pageName` — case-insensitive match → renders configured page

4 404 route \*

See full map: `APPENDICES_GENERATED/A3_ROUTE_MAP.md`.

## 2.3 Session + Tenancy + Scoping (today)

- Session “single source of truth”: `src/components/hooks/useSession.jsx`
- Auth state machine: `src/lib/AuthContext.jsx`
- School context: `src/components/school/SchoolContext.jsx`
- Scoping helpers: `src/components/api/scoped.jsx`
- Scoping registry: `src/components/api/scopedEntities.js`
- Runtime guard: `src/components/api/tenancyEnforcer.js` + `tenancyRuntime.js`

## 2.4 Feature Registry + Vault

- Canonical registry: `src/components/config/features.jsx`
- Adapter: `src/components/utils/featureRegistry.jsx`
- Command palette: `src/components/navigation/CommandPalette.jsx`

Generated feature catalog: `APPENDICES_GENERATED/A2_FEATURE_CATALOG.md`.

## 2.5 Project tree (high level)

See: `APPENDICES_GENERATED/A1_REPO_TREE_src_depth3.txt`.

---

# 3. Product Vision & Requirements (Master Plan)

This section stitches the spec pack’s most important docs into one continuous narrative.  
Do not change requirements unless explicitly approved.

## Architecture North Star (v9.0)

\_Source: `SPECPACK_V5/ARCHITECTURE_NORTH_STAR.md`\_

Prepared: January 01, 2026 (Asia/Jerusalem)

This document defines the target architecture that keeps all existing features while adding:

- Public marketing website
  - Separate Student/Teacher portals + separate login entry points
  - School/Entity signup + onboarding (multi-tenant)
  - Full premium UI/UX revamp (WOW, calm, academic)
- 

## 1) App Shell (React/Vite/Base44-style assumptions)

The existing project references a Vite alias of @ -> ./src and a Base44-style router/pages structure. Do not assume exact paths; Phase 0 audit must confirm real file locations.

### 1.1 Single session source-of-truth

Create or harden a session hook/provider that exposes:

- user (identity)
- memberships[] (tenant memberships + roles)
- activeSchoolId + activeSchoolSlug
- audience (student | teacher | admin | superadmin)
- isLoading, isAuthenticated

Rules:

- Audience is derived from role claims and/or portal entry point.
- Active school is chosen by:
  - 1) explicit selector (for multi-tenant users)
  - 2) last-used in localStorage (scoped)
  - 3) fallback to first membership
- Store only non-sensitive session hints client-side.

### 1.2 Multi-surface routing

We split surfaces by route groups:

- Public marketing: /, /about, /faq, etc.
- Student portal: /student/\*
- Teacher portal: /teacher/\*
- School admin: /admin/\*
- Platform admin: /superadmin/\* (global-only)



- Public storefront: /s/:schoolSlug/\*

Legacy routes must remain working via:

- alias routes in registry
- redirect adapters that preserve query strings

### **1.3 Guard layers**

Every protected surface requires *\*both\**:

- Route guard: blocks UI rendering if role/tenant missing
  - Data guard: all queries/mutations are server/app-layer protected by tenant + entitlement
- 

## **2) Tenant & RBAC Model**

### **2.1 Core concepts**

- Tenant (School/Entity) — owns courses, lessons, offers, settings, staff, branding.
- Membership — ties user to tenant + role(s).
- Role — student/teacher/admin/owner/superadmin

### **2.2 Tenant scoping enforcement**

- Never trust client-provided `tenant_id`.
  - Derive tenant scope from:
    - `activeSchoolId` for authenticated portal requests
    - `schoolSlug` -&gt; `school_id` for public storefront
- 

## **3) Content Protection & Materials Engine**

### **3.1 Access states (MUST)**

- LOCKED — no entitlement
- DRIP\_LOCKED — has entitlement, but not released yet
- UNLOCKED — entitled + released

Hard rule: In LOCKED or DRIP\_LOCKED, lesson materials must not be fetched, rendered, mounted, nor passed into components.

### **3.2 Materials engine contract**

The engine returns:

- `accessState`
  - `reasons[]` (why locked)
  - `unlockAt` (for drip)
  - `preview` (if allowed by policy)
  - `materials` only when UNLOCKED
- 

## **4) Feature Registry + Vault as the anti-loss safety net**

One typed registry drives:

- Main navigation (streamlined)
- Vault (full directory)
- Optional command palette
- Integrity checks

Registry must be a central module (`src/config/features.*` or `src/registry/featureRegistry.*`). Never inline registry arrays in UI pages.

---

## **5) Observability & Integrity Contract**

An admin-only `/integrity` diagnostic must:

- validate registry vs router existence (best-effort)
  - validate scoping heuristics
  - validate leakage heuristics (materials not fetched while locked)
  - export JSON report
- 

## **6) Delivery constraints**

- Extend-not-replace: prefer additive changes and adapters.
- No feature deletions. If nav is simplified, keep feature in Vault.
- “Build must pass” is a requirement — fix root causes (aliases/imports/guards), not symptoms.

## **IA + Route Map (Public + Portals + Storefront) — v9.0**

`_Source: SPECPACK_V5/IA_ROUTES_PORTALS.md_`

Prepared: January 01, 2026 (Asia/Jerusalem)

This is the canonical route map. It must be implemented without breaking existing routes.

---

### **1) Public Marketing Website (Trust + Conversion)**

Routes (required):

- / — landing (premium hero, value props, CTA)
- /about
- /how-it-works
- /faq
- /contact
- /privacy
- /terms

Top nav (required):

- Student Login -> /login/student
- Teacher Login -> /login/teacher
- Sign Up -> /signup (if enabled)
- School/Entity Sign Up -> /signup/school (or Request Demo)

Implementation:

- Public site uses marketing layout shell.
  - Must not require authentication.
-

## 2) Auth Entry Points (Separate Login Pages)

- /login/student
- /login/teacher
- /login/admin (optional)
- /login (legacy) -> redirect to chooser

Rules:

- All login pages use same underlying auth mechanism.
  - Each login page sets desired audience, then routes to correct portal post-login.
- 

## 3) Signup & Onboarding

### 3.1 Individual signup

- /signup (chooser)
- /signup/student
- /signup/teacher

Controls:

- Student: invite code OR domain match OR pending approval.
- Teacher: invite-only OR pending approval.

### 3.2 School/Entity signup (Tenant creation)

- /signup/school

Flow: 1) Submit org info -> create TenantApplication or Tenant in pending 2) Create Owner/Admin account or tie to existing user 3) Tenant onboarding wizard in /admin/onboarding

---

## 4) Portals (Separate Layouts)

### 4.1 Student portal: /student/\*

Minimum pages:

- /student/dashboard

- /student/courses
- /student/course/:courseId
- /student/lesson/:lessonId (lesson viewer)
- /student/assignments
- /student/progress
- /student/messages
- /student/account (or /account alias)

#### **4.2 Teacher portal:** /teacher/\*

Minimum pages:

- /teacher/dashboard
- /teacher/classes
- /teacher/course-builder (or existing course builder route)
- /teacher/grading
- /teacher/analytics
- /teacher/messages

#### **4.3 School admin portal:** /admin/\*

Minimum pages:

- /admin/onboarding
- /admin/users
- /admin/invites
- /admin/classes
- /admin/settings (branding, terminology, protection policy)
- /admin/approvals
- /admin/monetization (if already exists)
- /admin/audit-logs
- /admin/migration-tools
- /admin/integrity (or /integrity vault-only)

#### **4.4 Platform admin:** /superadmin/\*

Minimum pages:

- /superadmin/tenants

- /superadmin/users
- /superadmin/risk-audit
- /superadmin/impersonation
- /superadmin/system

Lockdown:

- Global role only
  - Must be audited
- 

## 5) Public Storefront (Per school slug)

Canonical:

- /s/:schoolSlug (school landing)
- /s/:schoolSlug/catalog
- /s/:schoolSlug/course/:courseId (sales page)
- /s/:schoolSlug/pricing (or offers)
- /s/:schoolSlug/checkout
- /s/:schoolSlug/thank-you

Rules:

- Guests may browse marketing + storefront.
  - Lesson content never shown here beyond preview policy.
- 

## 6) Legacy routes & compatibility

During Phase 0 audit, enumerate all existing routes and:

- set each to a canonical route in the registry
- add aliases[] for old routes
- add redirect adapters preserving query strings

Deliverable:

- Append a “Legacy Route Map” section after audit listing each old route -> new route.

## **Public Website + Separate Portals (Design + Implementation Plan)**

\_Source: SPECPACK\_V5/V9\_PUBLIC\_SITE\_AND\_PORTALS\_PLAN.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

This document translates the “unified master plan” into this exact v9.0 codebase.

### **Goal**

Add: 1) Public marketing site (trust + conversion) 2) Separate portal experiences:

- Student (/student/\*)
- Teacher (/teacher/\*)
- School Admin (/admin/\*)
- Super Admin (/superadmin/\*)

3) Separate login URLs:

- /login/student
- /login/teacher
- optional /login/admin

WITHOUT breaking:

- Existing Base44 legacy routes (/ : pageName)
  - Vault discoverability
  - Tenancy invariants
  - Access/entitlement gating
- 

### **A) Public marketing site**

#### **New routes**

- / (public landing if not signed in; otherwise route to portal)
- /about
- /how-it-works
- /faq
- /contact
- /privacy

- /terms

### **Implementation approach (Vite + React Router)**

1) Add explicit routes in `src/App.jsx` above the generic pages map. 2) Create new lightweight page components under:

- `src/pages/public/` (recommended) OR `src/pages/` (keep simple)

3) Add feature registry entries (audiences: `public`, `hidden: true` for legal pages if desired) 4) Reuse existing design system:

- `src/components/theme/tokens`
- `GlassCard`, `PageShell`, `SectionHeader`

### **Root routing logic**

/ should:

- show Landing if user is not authenticated
  - show “select school” if user has multiple memberships and no active school
  - show portal home if active school is set
- 

## **B) Portals**

### **New route groups**

- `/student/*`
- `/teacher/*`
- `/admin/*`
- `/superadmin/*`

### **The “smart” way (uses existing session audience)**

Create a tiny router bridge:

- `src/components/routing/PortalGate.jsx`

Responsibilities: 1) Read current session: `useSession()` 2) Enforce allowed roles:

- student portal: any membership role
- teacher portal: `instructor/TA/admin/owner`
- admin portal: `admin/owner`



- superadmin: platform global admin (see roles helper)

3) Set audience in session (or localStorage intent) so nav renders correct items. 4) Render the existing Layout with portal-specific default route.

## Portal home pages

Define these defaults:

- Student: /student/dashboard → renders existing Dashboard module (or new StudentDashboard)
- Teacher: /teacher/dashboard → renders existing Teach / TeachAnalytics
- Admin: /admin/dashboard → renders existing SchoolAdmin / SchoolAnalytics
- Superadmin: /superadmin/network → renders NetworkAdmin + audit log

## Keep legacy routes

All existing routes remain working:

- /dashboard, /Dashboard
- /teach, /Teach

etc.

Over time we add gentle redirects:

- /teach → /teacher/teach (optional)

but only after stability is proven.

---

## C) Separate login pages

Add explicit routes:

- /login/student
- /login/teacher
- /login/admin (optional)

These pages:

- set intended audience
- explain what happens
- trigger `base44.auth.redirectToLogin()`

## D) School/entity signup

We already have SchoolNew and SchoolJoin. We will evolve them into:

- /signup/school (application wizard)
  - admin approvals inside /admin/\*
- 

## E) Navigation

The existing layout uses:

- getNavGroupsForAudience(audience) from the registry

So portals become “real” immediately once audience is set.

We will add:

- Portal switch links (if user has multiple roles)
- Distinct portal topbars (colors/icons) without duplicating logic

## Tenancy & Data Scoping (Reality)

\_Source: SPECPACK\_V5/V9\_TENANCY\_REALITY.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

## Core principle

If an entity is “school-scoped”, every query + write must be scoped to school\_id.

### Source of truth for scoping

- src/components/api/scopedEntities.js
- SCHOOL\_SCOPED\_ENTITIES (must include school\_id)
- GLOBAL\_ENTITIES (no school\_id requirement)

### Safe query/write helpers

- src/components/api/scoped.jsx

- `scopedList(entity, schoolId, sort, limit)`
- `scopedFilter(entity, schoolId, filter)`
- `scopedCreate(entity, payload, schoolId)`
- `scopedUpdate(entity, id, payload, schoolId, strict?)`
- `scopedDelete(entity, id, schoolId, strict?)`

### **Runtime enforcement (automatic guard)**

- Installed in `src/api/base44Client.js` via `installTenancyEnforcer(...)`
- Runtime context is kept in sync by:
- `src/components/api/TenancyBridge.jsx`
- Tenancy context lives in:
- `src/components/api/tenancyRuntime.js` (module state)

### **Session source of truth**

- `src/components/hooks/useSession.jsx`
- `activeSchoolId`
- `activeSchool`
- `memberships`
- `role` (per active school)
- `audience` (student/teacher/admin) derived from role

### **Practical coding rules**

- 1) When you touch any page that reads/writes scoped entities:
  - replace direct `base44.entities.X.*` calls with `scoped*`
- 2) For global admins:
  - use explicit “global escape hatch” functions (as already provided by `tenancyEnforcer`)
- 3) Never rely on UI-only tenancy enforcement — the API layer must enforce too.

### **Smoke checks to run after tenant changes**

(See `HEALTHCHECKS.md` in repo root.)

## **V9.0 Stable — Feature Registry (Reality Dump)**

\_Source: SPECPACK\_V5/V9\_FEATURE\_REGISTRY\_REALITY.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

Source of truth:

- `src/components/config/features.jsx`

This registry powers:

- Left navigation groups (`src/Layout.jsx` via `getNavGroupsForAudience`)
- Vault discoverability (`src/pages/Vault.jsx`)
- Canonical route selection (`src/utils/index.ts`)

## Key policies

- Never delete a feature from the registry.
- If a page exists but is missing from the registry, add it as hidden (and/or `vaultOnly`) rather than omitting it.

## Known mismatch in v9.0 ZIP

The following page components are present in `pages.config` but are not in the feature registry:

- `TeachQuizEditor`
- `QuizTake`

Fix plan:

- Add registry entries for both (hidden), and reference their canonical routes:
- `TeachQuizEditor` → `/teach/quizzes/new` (and `/teach/quizzes/:quizId`)
- `QuizTake` → `/quiz/:quizId`

---

## Feature list (76)

Feature key	Label	Canonical route	Area	Audiences	Nav order	Hidden	Vault only	
--- --- --- --- --- ---	SchoolAdmin	School Admin	/schooladmin	admin	admin	1		
SchoolAnalytics	School Analytics	/schoolanalytics	admin	admin	2			
SchoolMonetization	Monetization	/schoolmonetization	admin	admin	3			
AuditLogViewer	Audit Log	/auditlogviewer	admin	admin		SchoolNew		
Create School	/schoolnew	admin	admin		LegacyMigration	Legacy Migration		

```

/legacymigration | admin | admin | | | | NetworkAdmin | Network Admin |
/networkadmin | admin | admin | | | | AdminHardening | Security Hardening |
/adminhardening | admin | admin | | | | SchoolStaff | Staff Management |
/schoolstaff | admin | admin | | | | Dashboard | Dashboard | /dashboard | core |
student,teacher,admin | 1 | | | Courses | Courses | /courses | core |
student,teacher,admin | 2 | | | Reader | Smart Reader | /reader | core |
student,teacher,admin | 3 | | | Feed | Community | /feed | core |
student,teacher,admin | 4 | | | SchoolSearch | Search | /schoolsearch | core |
student,teacher,admin | 5 | | | MyQuizzes | My Quizzes | /my-quizzes | core |
student,teacher,admin | 6 | | | Account | My Account | /account | core |
student,teacher,admin | 10 | | | Community | Community | /community | core |
student,teacher,admin | | | | CourseDetail | Course Detail | /coursedetail | core |
student,teacher,admin | | | | Downloads | Downloads | /downloads | core |
student,teacher,admin | | | | Forums | Forums | /forums | core |
student,teacher,admin | | | | LearningPaths | Learning Paths | /learningpaths |
core | student,teacher,admin | | | | LessonViewer | Lesson Viewer | /lessonviewer |
core | student,teacher,admin | | | | LessonViewerPremium | Lesson Viewer Premium |
/lessonviewerpremium | core | student,teacher,admin | | | | Messages | Messages |
/messages | core | student,teacher,admin | | | | MyProgress | My Progress |
/myprogress | core | student | | | | Achievements | Achievements | /achievements |
labs | student,teacher,admin | | | | AdaptiveLearning | Adaptive Learning |
/adaptivelearning | labs | student,teacher,admin | | | | Affiliate | Affiliate
Program | /affiliate | labs | student,teacher,admin | | | | AlumniNetwork | Alumni
Network | /alumninetwork | labs | student,teacher,admin | | | | Analytics |
Analytics | /analytics | labs | student,teacher,admin | | | | CareerPaths | Career
Paths | /careerpaths | labs | student,teacher,admin | | | | Challenges | Challenges |
/challenges | labs | student,teacher,admin | | | | CohortDetail | Cohort Detail |
/cohortdetail | labs | student,teacher,admin | | | | Cohorts | Cohorts | /cohorts |
labs | student,teacher,admin | | | | Events | Events | /events | labs |
student,teacher,admin | | | | HabitTracker | Habit Tracker | /habittracker | labs |
student | | | | LanguageLearning | Languages | /languagelearning | labs |
student,teacher,admin | | | | Languages | Languages (Legacy) | /languages | labs |
student,teacher,admin | | | | Leaderboard | Leaderboard | /leaderboard | labs |
student,teacher,admin | | | | LiveStreams | Live Streams | /livestreams | labs |
student,teacher,admin | | | | Marketplace | Marketplace | /marketplace | labs |
student,teacher,admin | | | | Mentorship | Mentorship | /mentorship | labs |
student,teacher,admin | | | | Microlearning | Microlearning | /microlearning | labs
| student | | | | StudySetNew | New Study Set | /studyssetnew | labs |
student,teacher,admin | | | | Offline | Offline | /offline | labs |
student,teacher,admin | | | | StudySetPractice | Practice | /studyssetpractice | labs
| student,teacher,admin | | | | RewardsShop | Rewards Shop | /rewardsshop | labs |
student | | | | Scholarships | Scholarships | /scholarships | labs | student | | | |
Skills | Skills | /skills | labs | student,teacher,admin | | | | StudyBuddies | Study
Buddies | /studybuddies | labs | student | | | | StudySet | Study Set | /studysset |
labs | student,teacher,admin | | | | StudySets | Study Sets | /studyssets | labs |
student,teacher,admin | | | | Tournaments | Tournaments | /tournaments | labs |
student | | | | AffiliateProgram | Affiliate Program Info | /affiliateprogram |
marketing | public,student,teacher,admin | | | | CertificateVerify | Certificate

```

Verification | /certificateverify | marketing | public | | | | SchoolCheckout |  
Checkout | /schoolcheckout | marketing | public, student, teacher, admin | | | |  
SchoolCourses | Course Catalog | /schoolcourses | marketing |  
public, student, teacher, admin | | | | SchoolLanding | Landing Page | /schoollanding  
| marketing | public, student, teacher, admin | | | | SchoolPricing | Pricing |  
/schoolpricing | marketing | public, student, teacher, admin | | | | CourseSales |  
Sales Page | /coursesales | marketing | public, student, teacher, admin | | | |  
SchoolThankYou | Thank You | /schoolthankyou | marketing |  
public, student, teacher, admin | | | | InviteAccept | Accept Invite | /inviteaccept |  
system | public | | | | Integrations | Integrations | /integrations | system |  
student, teacher, admin | | | | Integrity | Integrity Check | /integrity | system |  
admin | | | | SchoolJoin | Join School | /schooljoin | system |  
student, teacher, admin | | | | oauth2callback | OAuth Callback | /oauth2callback |  
system | public | | | | Portfolio | Profile | /portfolio | system |  
student, teacher, admin | | | | SchoolSelect | School Select | /schoolselect | system |  
student, teacher, admin | | | | Subscription | Subscription | /subscription | system |  
student, teacher, admin | | | | Vault | Vault | /vault | system |  
student, teacher, admin | | | | Teach | Teach | /teach | teach | teacher, admin | 1 | | | |  
TeachAnalytics | Teaching Analytics | /teachanalytics | teach | teacher, admin | 2 | | | |  
| TeachQuizzes | Quizzes | /teach/quizzes | teach | teacher, admin | 3 | | | |  
TeachCourse | Course Builder | /teachcourse | teach | teacher, admin | | | |  
TeachLesson | Lesson Editor | /teachlesson | teach | teacher, admin | | | |  
TeachCourseNew | New Course | /teachcoursenew | teach | teacher, admin | | | |

## Content Protection + Materials Engine Spec — v9.0

\_Source: SPECPACK\_V5/CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

This is the heart of the platform. It must never leak protected material.

---

### 1) Threat model (what must never happen)

- Guest sees full lesson text/transcripts.
  - Locked user receives direct file URLs.
  - DRIP\_LOCKED user fetches materials early “in the background.”
  - Search indexes full text and exposes it to unauthorized users.
  - A component mounts material payload even briefly before gate renders.
-

## 2) Access state machine

Inputs:

- user session + membership
- entitlement(s) (active, unexpired)
- drip schedule (release logic)
- admin policy (preview limits, watermark, copy/download licensing)
- lesson/courses flags

Outputs:

- LOCKED (no entitlement)
  - DRIP\_LOCKED (entitled but unreleased)
  - UNLOCKED (entitled + released)
- 

## 3) Hard enforcement rules

- In LOCKED or DRIP\_LOCKED, the UI must render AccessGate only.
  - Do not call materials fetch functions.
  - Do not render child viewer components that expect material.
  - Do not pass placeholder material objects downstream.
- 

## 4) Materials Engine API

Suggested functions:

- `getLessonAccess({lessonId, userId, tenantId}) -> { state, unlockAt, reasons, previewAllowed }`
- `getLessonMaterials({lessonId, userId, tenantId}) -> materials (must assert UNLOCKED)`
- `getDownloadUrl({fileId, userId, tenantId}) -> signedOrEphemeralUrl (must log)`

Implementation patterns:

- One hook: `useLessonAccess(lessonId)` returns state and UI gating.

- Viewer mounts only when state === 'UNLOCKED'.
- 

## 5) Preview policy

Admin-configurable:

- preview minutes for video
  - preview paragraphs for text
  - watermark configuration
  - copy/print/right-click blocking toggles (UI layer + logging)
  - “copy/download license” add-ons
- 

## 6) Search leakage prevention

Search results:

- metadata only: title, tags, snippet from allowed preview
- never query full transcripts or full text for unauthorized users
- log suspicious repeated search requests

## Security Threat Model (Practical) — v9.0

\_Source: SPECPACK\_V5/SECURITY\_THREAT\_MODEL.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

This is a pragmatic checklist for the app-layer security model.

---

### 1) Primary threats

- Unauthorized lesson material viewing
- Download URL leakage
- Cross-tenant data leakage
- Privilege escalation (role spoofing)



- Replay/duplication in checkout flows
  - Invite token reuse
  - Bot abuse (signup/login/checkout)
- 

## **2) Controls**

### **2.1 Tenant scoping**

- Every query includes `school_id` filter.
- Storefront resolves slug -> `school_id` at root and reuses it.

### **2.2 RBAC**

- Route guards + data guards.
- Superadmin tools are locked down and audited.

### **2.3 Content gating**

- `LOCKED/DRIP_LOCKED` never fetch materials.
- Preview rendering uses separate “safe preview” payload.

### **2.4 Download reveal**

- A “reveal” action fetches URL only after entitlement verification.
- Log reveal event.
- Prefer ephemeral URLs if supported; otherwise best-effort patterns.

### **2.5 Idempotency**

- Payment intent id unique
- Entitlement grants dedupe
- Commission + payout dedupe

### **2.6 Abuse protections**

- Rate limit login/signup/reveal/download/checkout calls.
- Basic bot mitigation on public forms (where feasible).

### 3) Security regression tests

- Manually test locked/drip locked states
- Verify search never shows full content
- Try cross-tenant switching and ensure no data bleed

## Premium UI/UX Revamp Plan (University-grade, Calm, WOW)

\_Source: SPECPACK\_V5/V9\_UI\_UX\_REVAMP\_PLAN.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

This codebase already has:

- Tailwind
- shadcn/ui + Radix
- Theme tokens: `src/components/theme/tokens`
- Theme toggle: `src/components/theme/ThemeToggle`

### Strategy: “Shell first, then modules”

1) Upgrade tokens + themes (light/dark/night) 2) Make Layout + PageShell consistent 3) Standardize reusable components (cards, tables, forms, empty states) 4) Convert pages module-by-module without feature loss 5) Final polish: micro-interactions, accessibility audit, perf budgets

### Hard requirements

- No harsh whites (use surfaces)
- Large readable typography
- Reduced motion support
- Responsive layouts, especially dashboard + course pages

### Deliverables (concrete)

- Token updates:
- `src/components/theme/tokens.*`
- New component primitives:

- GlassCard (already exists)
- DataTable, StatCard, KPIGrid
- WizardShell (for onboarding)
- Landing hero:
- subtle gradients, strong type, CTAs

## **Ops Hardening + Integrity + QA (v9.0)**

\_Source: SPECPACK\_V5/OPS\_INTEGRITY\_QA.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

---

### **1) Ops hardening goals**

- No white screens; degrade gracefully with actionable errors.
  - No data leaks; tenant scoping must be consistent.
  - Idempotent money flows.
  - Auditability: admin actions are traceable.
- 

### **2) Integrity page contract**

Admin-only, vaultOnly. Must include:

- Registry checks:
- feature count
- route existence best-effort
- Security/leakage heuristics:
- “materials fetched while locked” detection (best-effort)
- downloads url exposure checks
- search metadata-only checks
- Tenancy heuristics:
- scan queries for missing school\_id filters (best-effort patterns)
- RBAC checks:

- Network admin requires global role
- Staff/invites/tools require admin

Export:

- JSON report with timestamp, school\_id, checks, warnings, environment metadata.
- 

### **3) Regression net (smoke tests)**

Manual smoke checklist (must pass before release):

- Guest:
    - browse /s/:slug catalog and sales pages
    - confirm no full lesson materials visible
  - Student locked:
    - open lesson viewer -> AccessGate only
    - confirm no materials fetch calls
  - Student drip locked:
    - countdown visible; no materials fetch
  - Student unlocked:
    - lesson renders; watermark/policy applied
  - Downloads:
    - unauthorized -> no URL; license CTA
    - authorized -> click "Get download"; url revealed; event logged
  - Staff:
    - invite staff; accept token; cannot reuse token; audit logs present
    - revoke staff; access removed
  - Money:
    - approve same transaction twice -> no duplicates
    - create payout batch twice -> no duplicates
- 

### **4) Performance budgets**

- list queries: limit + stable sort

- cache keys include activeSchoolId
- heavy tables become cards on mobile

## **Future Builds — Detailed Playbook (Always Improving)**

\_Source: SPECPACK\_V5/FUTURE\_BUILDS\_PLAYBOOK.md\_

Prepared: January 01, 2026 (Asia/Jerusalem)

### **The loop (repeat every build)**

1) Measure (activation, learning, money, reliability) 2) Pick 3-5 high-impact items + 5-10 quality items 3) Ship safely (small PRs, flags if risky) 4) QA + integrity export 5) Review + document + update backlog

### **Parallel tracks (always-on)**

- UX polish
- Security/trust
- Performance/stability
- Conversion/onboarding

### **Never-regress guard rails**

- Feature registry diff check each build
  - Locked-content network-call monitor (best-effort)
  - Tenant-scope tracer (best-effort)
  - Money idempotency tests
- 

## **4. v9.1 — Portalization + Public Site (Implementation Blueprint)**

### **4.1 Core idea**

Keep the existing app intact, but introduce:

- Public site routes (marketing/trust) that do not depend on app shell.
- Portal shells that present different nav + UX for:

- Student
- Teacher
- School Admin
- Superadmin (locked down)

Important Base44 constraint (design implication): Base44 handles the actual login UI; we can implement *\*entry pages\** that redirect to Base44's auth and return users to the correct portal.

## 4.2 Compatibility requirements (do not break)

- Existing deep links must keep working:
- `/dashboard`, `/courses`, `/:pageName`, `/s/:schoolSlug/*`, quiz routes, etc.
- Existing feature registry routes remain canonical. Portal routes can be aliases.
- Tenant scoping and content protection must remain intact after refactors.

## 4.3 Portal route scheme (v9.1)

Add these top-level route groups:

- Public site:
- `/`, `/about`, `/how-it-works`, `/faq`, `/contact`, `/privacy`, `/terms`
- Auth entry points:
- `/login/student`, `/login/teacher`, `/login/admin` (optional)
- Portals:
- `/student/*`
- `/teacher/*`
- `/admin/*`
- `/superadmin/*` (future hardening)

How to implement without moving all pages:

- Keep legacy routes as-is.
- Add new portal routes that render the same page components but inside a portal layout.
- Use a small “page resolver” utility that maps page slug → Pages component (reusing the existing case-insensitive logic in `App.jsx`).

## 5. PR Series — Extremely Detailed Build Steps

### PR-000: Baseline hygiene (no behavior change)

Goal: prepare repo for portalization PRs without breaking anything.

- 1 Create /docs/ folder and copy in spec docs (or reference existing).
- 2 Add docs/REALITY\_MAP.md that links to the generated appendices.
- 3 Add docs/PR\_GUIDE.md describing the “PR-by-PR” output format.

#### Files

- Add: docs/README.md, docs/REALITY\_MAP.md, docs/PR\_GUIDE.md
- Update: README.md (link to docs)

#### Test

- npm ci
  - npm run dev
  - Navigate to: /dashboard, /courses, /s/<schoolSlug>, /my-quizzes
- 

### PR-001: Introduce Public Site routes (no auth changes yet)

Goal: create a marketing-quality public site without impacting app pages.

#### Strategy

- Add public pages under a new folder src/public/.
- Add a PublicLayout (top nav with CTA buttons).
- Add routes in src/App.jsx *before* legacy routes so they match first.

#### Files to add

- src/public/PublicLayout.jsx
- src/public/pages/Landing.jsx
- src/public/pages/About.jsx
- src/public/pages/HowItWorks.jsx
- src/public/pages/FAQ.jsx
- src/public/pages/Contact.jsx
- src/public/pages/Privacy.jsx
- src/public/pages/Terms.jsx

- `src/public/components/*` (hero, CTA, testimonials placeholders)

#### Files to modify

- `src/App.jsx`:
- Change `/` from “main app page” to `<Landing />`
- Add the other public routes
- Add a safe redirect route so `/app` (or `/dashboard`) still enters the app.

#### Compatibility

- The old “main page” (Dashboard) must remain accessible at `/dashboard` and `/Dashboard`.
- If marketing pages need school context, they must remain guest-safe.

#### Test

- `npm run dev`
  - Validate routes:
  - `/` shows Landing
  - `/about`, `/faq` etc show content
  - `/dashboard` still works
  - Quizzes and storefront still work
- 

### **PR-002: Auth entry routes (student/teacher/admin) — redirects only**

Goal: create separate “login” URLs that *only* redirect into Base44 auth and then return to the right portal.

#### Files to add

- `src/auth/LoginEntry.jsx` (generic component)
- `src/auth/loginPages/LoginStudent.jsx`
- `src/auth/loginPages/LoginTeacher.jsx`
- `src/auth/loginPages/LoginAdmin.jsx` (optional)

#### Behavior

- Each login page:

1) stores `portal_intent` in `localStorage` (student | teacher | admin) 2) calls `base44.auth.redirectToLogin(nextUrl)` where `nextUrl` is an app URL such as:



```
- `/student/dashboard` or `/teacher/dashboard`
```

Files to modify

- `src/App.jsx` add routes:
- `/login/student`, `/login/teacher`, `/login/admin`

Test

- Visit `/login/student` → should redirect to Base44 login
  - After auth callback, user should land in `/student/dashboard` (or fallback if not allowed)
- 

### **PR-003: Portal shells + portal resolver routing**

Goal: introduce `/student/*` `/teacher/*` `/admin/*` without moving pages.

New folders

- `src/portals/student/`
- `src/portals/teacher/`
- `src/portals/admin/`
- `src/portals/shared/`

Files to add

- `src/portals/shared/PortalPageResolver.jsx`
- Takes `portalRole` and `pageSlug` (dashboard, courses, etc)
- Resolves to correct Page component using the same mapping used by `/:pageName`.
- `src/portals/student/StudentLayout.jsx`
- `src/portals/teacher/TeacherLayout.jsx`
- `src/portals/admin/AdminLayout.jsx`
- `src/portals/shared/PortalGuard.jsx`
- Uses session + role checks
- Redirects to the appropriate login entry or “not authorized”.

Files to modify

- `src/App.jsx`
- Add layout routes (React Router nested routes) for:
- `/student/*`
- `/teacher/*`

- `/admin/*`
- Inside each, add:
- index redirect to `/student/dashboard`
- `:pageName` rendering through `PortalPageResolver`

#### RBAC gating

- Student portal only allows features tagged for students.
- Teacher portal only allows features tagged for teachers.
- Admin portal only allows admin features.
- Use canonical feature registry audiences (`FEATURES[*].audiences`).

#### Test

- `/student/dashboard` loads dashboard inside Student layout
  - `/teacher/teachcourse` loads teacher pages
  - Unauthorized user hitting `/admin/schooladmin` is blocked
  - Legacy routes still work.
- 

### **PR-004: Navigation split + “feature registry driven nav”**

Goal: stop showing “everything to everyone”; nav becomes portal-specific.

#### Strategy

- Keep the canonical feature registry.
- Create nav builders:
- `buildStudentNav(FEATURES)`
- `buildTeacherNav(FEATURES)`
- `buildAdminNav(FEATURES)`
- Each returns groups by `FEATURE_AREAS` + order.

#### Files to add

- `src/portals/shared/navBuilders.js`
- `src/portals/shared/PortalSidebar.jsx`

#### Files to modify

- Layouts from PR-003 to use `PortalSidebar`.

### Test

- Student sidebar shows student features only.
  - Teacher sidebar shows teacher tools.
  - Admin sidebar shows admin tools + audit log.
- 

## **PR-005: Tenant selection + “active school” UX inside portals**

Goal: ensure tenant context is explicit and safe.

### Strategy

- Portals must:
- show active school in top bar
- provide “switch school” if multi-membership
- Use existing `SchoolContext` and `useSession` logic.
- Ensure `tenancyRuntime` is updated when active school changes.

### Files likely touched

- `src/components/school/SchoolContext.jsx`
- `src/components/hooks/useSession.jsx`
- `src/portals/shared/*` topbar components

### Test

- When switching schools, data updates and scoped queries reflect correct `school_id`.
- 

## **PR-006: Public site “WOW pass” (design system starter)**

Goal: deliver the first layer of the premium calm design system on the public site, then later apply to portals.

- Implement typography scale, surface colors, spacing, accessible contrast.
- Add dark mode + “night reading” mode scaffolding.

### Files

- `src/styles/tokens.css` (or extend `globals.css`)
- Update tailwind config if necessary.

- Public site components updated to use tokens.

Test

- Verify dark mode toggle.
  - Verify prefers-reduced-motion respects user setting.
- 

## 6. QA & Security Checklist (Apply Every PR)

- Run: `npm run lint` and `npm run build`
- Smoke routes:
- `/` and marketing pages
- `/dashboard` (legacy)
- `/student/dashboard`, `/teacher/teachcourse`, `/admin/schooladmin`
- storefront `/s/:schoolSlug/*`
- quizzes `/my-quizzes`, `/quiz/:quizId`
- Validate scoping:
- any list/create on a school entity must show only that school's records
- Validate content protection:
- LOCKED lessons do not load Text/Transcript/Downloads
- Update:
- `KNOWN_ISSUES_TECH_DEBT.md` (what you saw, what you deferred)

For more: see `SPECPACK_V5/OPS_INTEGRITY_QA.md` and `SPECPACK_V5/V9_QA_RELEASES.md`.

---

## 7. Future Builds (Continuous Improvement Train)

Use `SPECPACK_V5/FUTURE_BUILDS_PLAYBOOK.md` as the baseline; expand each future build into PR sequences similar to v9.1.

Recommended release train:

- v9.1: portalization + public site skeleton + RBAC tightening
- v9.2: design system + full shell replacement

- v9.3: onboarding wizard + school/entity signup flows
  - v9.4: content authoring upgrades + analytics dashboards
  - v9.5: performance + a11y audits + e2e test suite
  - v10+: integrations, mobile-first offline, advanced personalization
- 

## **8. Appendices (Generated)**

- APPENDICES\_GENERATED/A1\_REPO\_TREE\_src\_depth3.txt
- APPENDICES\_GENERATED/A2\_FEATURE\_CATALOG.md
- APPENDICES\_GENERATED/A3\_ROUTE\_MAP.md
- APPENDICES\_GENERATED/A4\_ENTITIES\_TENANCY\_SCOPING.md
- APPENDICES\_GENERATED/A5\_KEY\_FILES.md

## 00\_START\_HERE.md

Breslov\_Academy\_Coding\_Book\_v9\_1/00\_START\_HERE.md

# Breslov Academy LMS — Coding Book (v9.1)

Generated: 2026-01-01 Inputs used:

- Stable code snapshot ZIP: breslov-academy\_v9.0-stable.zip
- Spec pack ZIP: Breslov\_Academy\_V9\_SpecPack\_TextDocs\_v5\_fromStable.zip

## What this is

This folder is the single coding reference for building the next versions of Breslov Academy LMS. It is designed so another ChatGPT thread can follow it like a book and produce PR-ready code changes.

## How to use (in the next chat)

- 1 Paste the contents of PROMPT\_FOR\_NEXT\_CHAT.txt as the first message.
- 2 The assistant must:
  - Read the repo's spec/plan docs (mirrored here under SPECPACK\_V5/).
  - Read the \*generated\* "reality maps" under APPENDICES\_GENERATED/.
  - Then propose a PR-by-PR implementation, with exact file paths + runnable Win11 commands.

## Non-negotiable invariants (summary)

- No features removed. Preserve everything; only reorganize, upgrade, and harden.
- Tenancy scoping is mandatory for school-owned data.
- Content protection: if a lesson is LOCKED or DRIP\_LOCKED, do not fetch protected materials.
- Server-side enforcement: UI hiding is not security.

Canonical source files for these invariants:

- STABLE\_ROOT\_DOCS/SECURITY\_INVARIANTS.md
- SPECPACK\_V5/SECURITY\_THREAT\_MODEL.md
- SPECPACK\_V5/CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md
- SPECPACK\_V5/V9\_TENANCY\_REALITY.md

## **Table of Contents (recommended reading order)**

- 1 CODING\_BOOK\_V9\_1.md (this is the “full book”)
- 2 APPENDICES\_GENERATED/A5\_KEY\_FILES.md
- 3 APPENDICES\_GENERATED/A3\_ROUTE\_MAP.md
- 4 APPENDICES\_GENERATED/A4\_ENTITIES\_TENANCY\_SCOPING.md
- 5 SPECPACK\_V5/V9\_ROADMAP\_EXTREME\_DETAIL.md
- 6 SPECPACK\_V5/PORTALIZATION\_BUILD\_SCRIPT\_V9\_1.md
- 7 SPECPACK\_V5/FUTURE\_BUILDS\_PLAYBOOK.md

## **GitHub repo details**

- HTTPS: <https://github.com/yeshivabachur/breslov-academy.git>
- SSH: <git@github.com:yeshivabachur/breslov-academy.git>
- GitHub CLI: `gh repo clone yeshivabachur/breslov-academy`

## **09\_FUTURE\_BUILDS\_CONTINUOUS\_IMPROVEMENT.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/09\_FUTURE\_BUILDS\_CONTINUOUS\_IMPROVEMENT.md

# **Continuous Improvement Roadmap (Beyond v9.1)**

This is the “always improving” plan — each release is a small batch of PRs, shipped frequently.

## **Release Train Philosophy**

- Keep main deployable at all times.
  - Feature flags for risky changes.
  - Small PRs with strong smoke tests.
  - Every release has:
    - changelog entry
    - migration notes (if any)
    - QA checklist run
- 

## **v9.2 — Design System + Full Shell Revamp**

### **Goals**

- Replace legacy `src/Layout.jsx` with a modern shell:
- calm surfaces, better typography, accessible focus states
- consistent page templates
- Introduce:
  - `src/design/` tokens
  - “night reading mode”
- skeleton loaders, micro-interactions with reduced-motion support

### **PR batches**

- 1 Tokens + theme provider + typography scale
- 2 Component library consolidation (buttons/inputs/cards/tables)
- 3 Portal shells migrate to new components



## 4 Legacy pages gradually refit (module-by-module)

---

### **v9.3 — School/Entity Onboarding + Invites + Approvals**

#### **Goals**

- /signup/school tenant application flow
- Invite teachers + import students
- Pending approvals + audit logs

#### **PR batches**

- 1 Org application entity + admin approval UI
  - 2 Tenant onboarding wizard
  - 3 Invite flows + email templates (Base44-supported)
  - 4 Approval gating + admin dashboards
- 

### **v9.4 — Content Authoring + Learning Enhancements**

#### **Goals**

- Better course builder UX
  - Draft/publish workflow
  - Quizzes + assignments deepening
  - Certificates & verification polish
- 

### **v9.5 — Quality: Performance + A11y + E2E Tests**

#### **Goals**

- Lighthouse targets for public site
- Core flows covered by e2e suite
- Accessibility audits + fixes
- Error monitoring baseline

## **v10+ — Integrations + Mobile-first + Personalization**

- Integrations: calendars, CRM, payment improvements
- Offline-first reading and downloads
- Personalized “study plan” + spaced repetition analytics

## **KNOWN\_ISSUES\_TECH\_DEBT.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/KNOWN\_ISSUES\_TECH\_DEBT.md

## **KNOWN ISSUES / TECH DEBT (Template)**

Update this file as you work. Do NOT let issues get lost in chat history.

### **Active issues**

- (date) (area) — description — impact — planned fix PR

### **Deferred items**

- (date) (area) — why deferred — revisit in version

### **Notes**

- Always include reproduction steps if a bug.

## **PROMPT\_FOR\_NEXT\_CHAT.txt**

Breslov\_Academy\_Coding\_Book\_v9\_1/PROMPT\_FOR\_NEXT\_CHAT.txt

### Repo + Baseline

- GitHub repo: yeshivabachur/breslov-academy
- Base branch: main
- Baseline commit: (paste the current SHA from `git rev-parse origin/main`)
- You MUST treat the repo files as the source of truth.

### Mission

- Read the “coding book” docs committed to the repo (or use the mirror in this ZIP).
- Produce PR-ready implementation steps and then generate code changes in follow-up requests.

### Non-negotiables

- No features removed.
- Tenancy + RBAC enforced server-side (via Base44 policies/functions where possible) AND client-side guardrails.
- Content protection: never fetch materials when LOCKED/DRIP\_LOCKED.

Process & Output Format For every PR / step: 1) What changed (human summary) 2) Exact file paths 3) Exact commands to run/test on Win11 (PowerShell) 4) What's next  
Always update:

- KNOWN\_ISSUES\_TECH\_DEBT.md
- SECURITY\_NOTES.md (if touched)

### Git operations (Win11 / PowerShell)

- Clone: `gh repo clone yeshivabachur/breslov-academy`
- Or: `git clone git@github.com:yeshivabachur/breslov-academy.git`
- Standard flow: create branch → commit → push → PR

## **REFERENCES.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/REFERENCES.md

## **References (External)**

These are non-authoritative references used for best practices. The repo docs remain the source of truth.

### **Base44 SDK + Auth**

- Base44 SDK overview (modules, entities, auth)
- `base44.auth.redirectToLogin(nextUrl)` and `logout()` docs

### **Routing (React Router v6)**

- Layout routes and nested routes (`<Outlet />`)

### **Security**

- OWASP Top 10 (Broken Access Control guidance)
- OWASP ASVS / authorization best practices

### **GitHub Workflow**

- Protected branches, required PR reviews, required status checks

## **SECURITY\_NOTES.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SECURITY\_NOTES.md

# **SECURITY NOTES (Template)**

This file is for \*implementation notes\* and decisions that affect security.

## **Access control decisions**

- (date) decision, rationale, references

## **Tenancy / scoping**

- (date) what changed, how verified

## **Content protection**

- (date) what changed, test cases

## **Auth / session**

- (date) changes to redirects, callback URLs, etc

## Part II — PR Series v9.1

### PR-001\_PUBLIC\_SITE.md

Breslov\_Academy\_Coding\_Book\_v9\_1/PR\_SERIES\_V9\_1/PR-001\_PUBLIC\_SITE.md

## PR-001 — Public Website Routes (Marketing + Trust)

### Objective

Add a premium public landing website while preserving the LMS app.

### Route targets

- / landing
- /about
- /how-it-works
- /faq
- /contact
- /privacy, /terms

### Constraints

- Public pages must remain guest-safe.
- If the app is configured to “require login globally” in Base44, public routes may still be blocked. If that happens, host marketing pages outside this app or create a separate Base44 app for marketing.

### Implementation steps

- 1 Create `src/public/PublicLayout.jsx`
- 2 Create pages under `src/public/pages/`
- 3 Add routes in `src/App.jsx` *\*before\** the legacy app routes.

### PublicLayout skeleton

Code (jsx):

```
import React from "react";
import { Link, Outlet } from "react-router-dom";

export default function PublicLayout() {
  return (
    <div className="min-h-screen">
      <header className="sticky top-0 z-20 border-b bg-background/80 backdrop-blur">
        <div className="mx-auto max-w-6xl px-4 py-3 flex items-center justify-between">
          <Link to="/" className="font-semibold">Breslov Academy</Link>
          <nav className="flex items-center gap-4 text-sm">
            <Link to="/about">About</Link>
            <Link to="/how-it-works">How it works</Link>
            <Link to="/faq">FAQ</Link>
            <Link to="/contact">Contact</Link>
            <Link to="/login/student" className="font-medium">Student Login</Link>
            <Link to="/login/teacher" className="font-medium">Teacher Login</Link>
          </nav>
        </div>
      </header>

      <main>
        <Outlet />
      </main>

      <footer className="border-t">
        <div className="mx-auto max-w-6xl px-4 py-8 text-sm text-muted-foreground">
          © {new Date().getFullYear()} Breslov Academy
        </div>
      </footer>
    </div>
  );
}
```

## App.jsx routing example

Code (jsx):

```
import PublicLayout from "@public/PublicLayout";
import Landing from "@public/pages/Landing";
import About from "@public/pages/About";
// ...

<Route element={<PublicLayout />}>
  <Route path="/" element={<Landing />} />
  <Route path="/about" element={<About />} />
  <Route path="/how-it-works" element={<HowItWorks />} />
  <Route path="/faq" element={<FAQ />} />
  <Route path="/contact" element={<Contact />} />
  <Route path="/privacy" element={<Privacy />} />
  <Route path="/terms" element={<Terms />} />
</Route>
```

## Test plan

- npm run dev
- Verify each public route loads without app sidebar.



- Verify legacy routes still work (/dashboard, /courses, etc).

## PR-002\_AUTH\_ENTRY\_ROUTES.md

Breslov\_Academy\_Coding\_Book\_v9\_1/PR\_SERIES\_V9\_1/PR-002\_AUTH\_ENTRY\_ROUTES.md

# PR-002 — Auth Entry Routes (student / teacher / admin)

## Objective

Add separate login URLs that set portal intent and then redirect to Base44's built-in login.

> Note: We do NOT implement a custom login form. Base44 owns the login UI.

## Files to add

src/auth/LoginEntry.jsx

Code (jsx):

```
import React, { useEffect } from "react";
import { base44 } from "@api/base44Client";

// Props:
// - portalIntent: "student" | "teacher" | "admin"
// - nextUrl: string (where to go after successful login)
export default function LoginEntry({ portalIntent, nextUrl }) {
  useEffect(() => {
    try {
      localStorage.setItem("portal_intent", portalIntent);
      // optional: store desired post-login URL
      localStorage.setItem("post_login_url", nextUrl);

      base44.auth.redirectToLogin(nextUrl);
    } catch (e) {
      // fallback: if localStorage not available, still redirect
      base44.auth.redirectToLogin(nextUrl);
    }
  }, [portalIntent, nextUrl]);

  return (
    <div className="min-h-screen flex items-center justify-center">
      <div className="text-center space-y-2">
        <div className="text-lg font-medium">Redirecting to login...</div>
        <div className="text-sm text-muted-foreground">
          If you are not redirected, please refresh the page.
        </div>
      </div>
    </div>
  );
}
```

**src/auth/loginPages/LoginStudent.jsx**

Code (jsx):

```
import React from "react";
import LoginEntry from "../LoginEntry";

export default function LoginStudent() {
  return <LoginEntry portalIntent="student" nextUrl="/student/dashboard" />;
}
```

**src/auth/loginPages/LoginTeacher.jsx**

Code (jsx):

```
import React from "react";
import LoginEntry from "../LoginEntry";

export default function LoginTeacher() {
  return <LoginEntry portalIntent="teacher" nextUrl="/teacher/dashboard" />;
}
```

**src/auth/loginPages/LoginAdmin.jsx (optional)**

Code (jsx):

```
import React from "react";
import LoginEntry from "../LoginEntry";

export default function LoginAdmin() {
  return <LoginEntry portalIntent="admin" nextUrl="/admin/schooladmin" />;
}
```

**Files to modify****src/App.jsx**

Add routes near the top of `<Routes>`; so they are always reachable.

Code (jsx):

```
import LoginStudent from "../auth/loginPages/LoginStudent";
import LoginTeacher from "../auth/loginPages/LoginTeacher";
import LoginAdmin from "../auth/loginPages/LoginAdmin";

// ...
<Route path="/login/student" element={<LoginStudent />} />
<Route path="/login/teacher" element={<LoginTeacher />} />
<Route path="/login/admin" element={<LoginAdmin />} />
```

**Edge cases / decisions**

- If the user lands in the wrong portal after login:

- Read `portal_intent` and redirect accordingly inside the app's post-auth bootstrap step.
- If the user is authenticated but hits `/login/*`:
- Immediately redirect to portal home.

## Test plan

- `npm run dev`
- Visit `/login/student` → Base44 login screen should appear.
- Complete login → should return to `/student/dashboard`.

## PR-003\_PORTAL\_SHELLS\_AND\_RESOLVER.md

Breslov\_Academy\_Coding\_Book\_v9\_1/PR\_SERIES\_V9\_1/PR-003\_PORTAL\_SHELLS\_AND\_RESOLVER.md

# PR-003 — Portal Shells + Portal Resolver Routing

## Objective

Add portal route groups that render existing Pages without moving them, but inside distinct layouts + guards.

## Key design

- PortalPageResolver maps a pageSlug to the right Page component.
- PortalGuard ensures the user is authenticated and allowed.
- Each portal has its own layout and nav.

## Files to add

src/portals/shared/portalPages.js

Exports the Pages registry so resolver code can share it.

Code (js):

```
import pagesConfig from "@pages.config";
export const Pages = pagesConfig.Pages;
```

src/portals/shared/resolvePageBySlug.js

Code (js):

```
import { Pages } from "../portalPages";

// pageSlug: "dashboard" or "Dashboard" etc
export function resolvePageBySlug(pageSlug) {
  if (!pageSlug) return null;

  // case-insensitive match against Pages keys
  const matchKey = Object.keys(Pages).find(
    (k) => k.toLowerCase() === pageSlug.toLowerCase()
  );
  return matchKey ? { key: matchKey, Component: Pages[matchKey] } : null;
}
```

**src/portals/shared/PortalGuard.jsx**

Code (jsx):

```
import React from "react";
import { Navigate, useLocation } from "react-router-dom";
import { useSession } from "@components/hooks/useSession";
import { hasRole } from "@components/utils/permissions";

// portalRole: "student" | "teacher" | "admin"
export default function PortalGuard({ portalRole, children }) {
  const { isAuthenticated, isLoading, role } = useSession();
  const location = useLocation();

  if (isLoading) return null;

  if (!isAuthenticated) {
    const loginPath =
      portalRole === "teacher" ? "/login/teacher" :
      portalRole === "admin" ? "/login/admin" :
      "/login/student";

    return <Navigate to={loginPath} replace state={{ from: location.pathname }} />;
  }

  // role check: you may have a richer role model; this is a placeholder.
  // Use existing permission helpers where possible.
  if (!hasRole(role, portalRole)) {
    return <Navigate to="/not-authorized" replace />;
  }

  return children;
}
```

**src/portals/shared/PortalPageResolver.jsx**

Code (jsx):

```
import React, { Suspense } from "react";
import { useParams, Navigate } from "react-router-dom";
import { resolvePageBySlug } from "../resolvePageBySlug";

export default function PortalPageResolver({ portalHome = "dashboard" }) {
  const { pageName } = useParams(); // from /student/:pageName
  const resolved = resolvePageBySlug(pageName);

  if (!pageName) return <Navigate to={portalHome} replace />;

  if (!resolved) return <Navigate to="/not-found" replace />;

  const { Component } = resolved;
  return (
    <Suspense fallback={<div className="p-6">Loading...</div>}>
      <Component />
    </Suspense>
  );
}
```

**Layouts**

Create portal layouts that wrap children with portal UI (sidebar/topbar, etc):

- `src/portals/student/StudentLayout.jsx`
- `src/portals/teacher/TeacherLayout.jsx`
- `src/portals/admin/AdminLayout.jsx`

Minimal skeleton:

Code (jsx):

```
import React from "react";
import { Outlet } from "react-router-dom";
import PortalSidebar from "@portals/shared/PortalSidebar";

export default function StudentLayout() {
  return (
    <div className="min-h-screen flex">
      <PortalSidebar portalRole="student" />
      <main className="flex-1">
        <Outlet />
      </main>
    </div>
  );
}
```

## Files to modify

`src/App.jsx`

Add nested layout routes:

Code (jsx):

```
import { Outlet, Navigate } from "react-router-dom";
import StudentLayout from "@portals/student/StudentLayout";
import TeacherLayout from "@portals/teacher/TeacherLayout";
import AdminLayout from "@portals/admin/AdminLayout";
import PortalGuard from "@portals/shared/PortalGuard";
import PortalPageResolver from "@portals/shared/PortalPageResolver";

// ...
<Route
  path="/student"
  element={
    <PortalGuard portalRole="student">
      <StudentLayout />
    </PortalGuard>
  }
>
  <Route index element={<Navigate to="dashboard" replace />} />
  <Route path=":pageName" element={<PortalPageResolver portalHome="dashboard" />} />
</Route>

<Route path="/teacher" element={
  <PortalGuard portalRole="teacher">
    <TeacherLayout />
  }
>
```

```
    </PortalGuard>
  }>
  <Route index element={<Navigate to="dashboard" replace />} />
  <Route path=":pageName" element={<PortalPageResolver portalHome="dashboard" />} />
</Route>

<Route path="/admin" element={
  <PortalGuard portalRole="admin">
    <AdminLayout />
  </PortalGuard>
}>
  <Route index element={<Navigate to="schooladmin" replace />} />
  <Route path=":pageName" element={<PortalPageResolver portalHome="schooladmin" />} />
</Route>
```

## Compatibility tests

- All legacy pages still work at /:pageName and /\${PageName}.
- Portal routes work for authorized users.
- Portal routes are blocked for unauthenticated users.



## Part III — Generated Appendices

### A1\_REPO\_TREE\_src\_depth3.txt

Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A1\_REPO\_TREE\_src\_depth3.txt

```
src/ |─ api/ |─ base44Client.js |─ components/ |─ academic/ |─ quizEngine.jsx |─
accessibility/ |─ AccessibilityMenu.jsx |─ admin/ |─ ContentProtectionSettings.jsx
|─ ai/ |─ AcademicAdvisorChat.jsx |─ AiTutor.jsx |─ AiTutorPanel.jsx |─
CourseRecommendations.jsx |─ analytics/ |─ attribution.jsx |─
ConversionFunnel.jsx |─ EngagementPredictor.jsx |─ LearningStyleAssessment.jsx
|─ RevenueChart.jsx |─ StudentDashboard.jsx |─ track.jsx |─
announcements/ |─ AnnouncementsPanel.jsx |─ api/ |─ contracts.js |─
scoped.jsx |─ scopedEntities.js |─ TenancyBridge.jsx |─ tenancyEnforcer.js |─
|─ tenancyRuntime.js |─ tenancyWarnings.js |─ auth/ |─ roles.jsx |─
automation/ |─ AutomationBuilder.jsx |─ career/ |─ ResumeBuilder.jsx |─
SalaryNegotiator.jsx |─ certificates/ |─ certificatesEngine.jsx |─
CertificateView.jsx |─ collaboration/ |─ Whiteboard.jsx |─ community/ |─
ReportButton.jsx |─ config/ |─ features.jsx |─ courses/ |─ CourseCard.jsx |─
creator/ |─ BulkOperations.jsx |─ dashboard/ |─ StatCard.jsx |─ drip/ |─
dripEngine.jsx |─ enterprise/ |─ TeamDashboard.jsx |─ focus/ |─
PomodoroTimer.jsx |─ gamification/ |─ AchievementBadge.jsx |─
PowerUpShop.jsx |─ hooks/ |─ useLessonAccess.jsx |─ useSession.jsx |─
useStorefrontContext.jsx |─ useTerminology.jsx |─ insights/ |─
LearningInsights.jsx |─ instructor/ |─ CourseBuilder.jsx |─
InstructorDashboard.jsx |─ TeachCourseCurriculum.jsx |─ TeachCoursePricing.jsx |─
|─ TeachCourseSettings.jsx |─ TeachCourseStudents.jsx |─ integrations/ |─
GoogleServiceConnector.jsx |─ language/ |─ FlashcardPractice.jsx |─
LanguageSelector.jsx |─ learning/ |─ BookmarkPanel.jsx |─ CodePlayground.jsx |─
|─ DiscussionThread.jsx |─ NotesPanel.jsx |─ PremiumVideoPlayer.jsx |─
QuizCard.jsx |─ TranscriptPanel.jsx |─ marketing/ |─ CourseLandingPage.jsx |─
materials/ |─ materialsEngine.jsx |─ mobile/ |─ OfflineMode.jsx |─ moderation/
|─ ContentReportForm.jsx |─ monetization/ |─ OfferCard.jsx |─ navigation/ |─
|─ CommandPalette.jsx |─ notifications/ |─ NotificationCenter.jsx |─ onboarding/
|─ OnboardingFlow.jsx |─ parent/ |─ StudentProgressReport.jsx |─ payouts/ |─
|─ PayoutBatchManager.jsx |─ proctoring/ |─ ProctorView.jsx |─ protection/ |─
ProtectedContent.jsx |─ scheduling/ |─ StudyCalendar.jsx |─ school/ |─
SchoolAnalytics.jsx |─ SchoolAnnouncements.jsx |─ SchoolAuditLog.jsx |─
SchoolContext.jsx |─ SchoolModeration.jsx |─ SchoolPayouts.jsx |─
SchoolProtectionSettings.jsx |─ SchoolSwitcher.jsx |─ TerminologySettings.jsx |─
search/ |─ AdvancedSearch.jsx |─ security/ |─ AccessGate.jsx |─ seo/ |─
MetaTags.jsx |─ study/ |─ MatchGame.jsx |─ WriteMode.jsx |─ subscription/ |─
|─ PricingCard.jsx |─ subscriptions/ |─ subscriptionEngine.jsx |─ system/ |─
codeScanner.js |─ Drawer.jsx |─ ErrorBoundary.jsx |─ Modal.jsx |─
usePageState.jsx |─ useToast.js |─ VirtualizedList.jsx |─ testimonials/ |─
TestimonialCard.jsx |─ theme/ |─ ThemeProvider.jsx |─ ThemeToggle.jsx |─
tokens.js |─ ui/ |─ accordion.jsx |─ alert-dialog.jsx |─ alert.jsx |─
```

```

aspect-ratio.jsx | | | avatar.jsx | | | badge.jsx | | | breadcrumb.jsx | | | button.jsx
| | | calendar.jsx | | | card.jsx | | | carousel.jsx | | | chart.jsx | | | checkbox.jsx |
| | | collapsible.jsx | | | command.jsx | | | context-menu.jsx | | | dialog.jsx | | |
drawer.jsx | | | dropdown-menu.jsx | | | EmptyState.jsx | | | form.jsx | | |
GlassCard.jsx | | | hover-card.jsx | | | IconButton.jsx | | | input-otp.jsx | | |
input.jsx | | | label.jsx | | | menubar.jsx | | | navigation-menu.jsx | | |
PageShell.jsx | | | pagination.jsx | | | popover.jsx | | | progress.jsx | | |
radio-group.jsx | | | resizable.jsx | | | scroll-area.jsx | | | SectionHeader.jsx | | |
select.jsx | | | separator.jsx | | | sheet.jsx | | | sidebar.jsx | | | skeleton.jsx | | |
slider.jsx | | | sonner.jsx | | | StatusBadge.jsx | | | Stepper.jsx | | | switch.jsx | |
| table.jsx | | | tabs.jsx | | | textarea.jsx | | | toast.jsx | | | toaster.jsx | | |
toggle-group.jsx | | | toggle.jsx | | | tooltip.jsx | | | use-toast.jsx | | | utils/ | | |
ARCHITECTURE.md | | | ARCHITECTURE.md.jsx | | | CHANGELOG.md | | |
CHANGELOG.md.jsx | | | docs.jsx | | | entitlements.jsx | | | featureFlags.jsx | | |
featureRegistry.jsx | | | permissions.jsx | | | queryUtils.js | | | RECOVERY.md | | |
RECOVERY.md.jsx | | | routes.jsx | | | scopedQueries.jsx | | |
SECURITY_CONTRACT.md | | | SECURITY_CONTRACT.md.jsx | | | video/ | | |
AdvancedVideoPlayer.jsx | | | TranscriptViewer.jsx | | | VideoAnnotations.jsx | | |
voice/ | | | SpeechPractice.jsx | | | vr/ | | | VRViewer.jsx | | | wellness/ | | |
WellnessTracker.jsx | | | UserNotRegisteredError.jsx | | | hooks/ | | | use-mobile.jsx | | | lib/
| | | app-params.js | | | AuthContext.jsx | | | NavigationTracker.jsx | | |
PageNotFound.jsx | | | query-client.js | | | utils.js | | | pages/ | | | Account.jsx | | |
Achievements.jsx | | | AdaptiveLearning.jsx | | | AdminHardening.jsx | | | Affiliate.jsx |
| | | AffiliateProgram.jsx | | | AlumniNetwork.jsx | | | Analytics.jsx | | | AuditLogViewer.jsx
| | | CareerPaths.jsx | | | CertificateVerify.jsx | | | Challenges.jsx | | | CohortDetail.jsx |
| | | Cohorts.jsx | | | Community.jsx | | | CourseDetail.jsx | | | Courses.jsx | | |
CourseSales.jsx | | | Dashboard.jsx | | | Downloads.jsx | | | Events.jsx | | | Feed.jsx | | |
Forums.jsx | | | HabitTracker.jsx | | | Integrations.jsx | | | Integrity.jsx | | |
InviteAccept.jsx | | | LanguageLearning.jsx | | | Languages.jsx | | | Leaderboard.jsx | | |
LearningPaths.jsx | | | LegacyMigration.jsx | | | LessonViewer.jsx | | |
LessonViewerPremium.jsx | | | LiveStreams.jsx | | | Marketplace.jsx | | | Mentorship.jsx |
| | | Messages.jsx | | | Microlearning.jsx | | | MyProgress.jsx | | | MyQuizzes.jsx | | |
NetworkAdmin.jsx | | | oauth2callback.jsx | | | Offline.jsx | | | Portfolio.jsx | | |
QuizTake.jsx | | | Reader.jsx | | | RewardsShop.jsx | | | Scholarships.jsx | | |
SchoolAdmin.jsx | | | SchoolAnalytics.jsx | | | SchoolCheckout.jsx | | | SchoolCourses.jsx
| | | SchoolJoin.jsx | | | SchoolLanding.jsx | | | SchoolMonetization.jsx | | | SchoolNew.jsx
| | | SchoolPricing.jsx | | | SchoolSearch.jsx | | | SchoolSelect.jsx | | | SchoolStaff.jsx | | |
SchoolThankYou.jsx | | | Skills.jsx | | | StudyBuddies.jsx | | | StudySet.jsx | | |
StudySetNew.jsx | | | StudySetPractice.jsx | | | StudySets.jsx | | | Subscription.jsx | | |
Teach.jsx | | | TeachAnalytics.jsx | | | TeachCourse.jsx | | | TeachCourseNew.jsx | | |
TeachLesson.jsx | | | TeachQuizEditor.jsx | | | TeachQuizzes.jsx | | | Tournaments.jsx |
| | | Vault.jsx | | | utils/ | | | index.ts | | | App.jsx | | | globals.css | | | index.css | | | Layout.jsx |
| | | main.jsx | | | pages.config.js

```

**A2\_FEATURE\_CATALOG.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A2\_FEATURE\_CATALOG.md

**Feature Catalog (Generated from  
src/components/config/features.jsx)**

- Generated: 2026-01-01
- Source file: src/components/config/features.jsx
- Total features: 76

feature_key	label	route	area	audiences	icon	order	
schoolAdmin	School Admin	/schooladmin	admin	admin	Settings	1	
schoolAnalytics	School Analytics	/schoolanalytics	admin	admin	TrendingUp	2	
schoolMonetization	Monetization	/schoolmonetization	admin	admin	DollarSign	3	
auditLogViewer	Audit Log	/auditlogviewer	admin	admin	Shield	nan	
schoolNew	Create School	/schoolnew	admin	admin	Plus	nan	
legacyMigration	Legacy Migration	/legacymigration	admin	admin	Database	nan	
networkAdmin	Network Admin	/networkadmin	admin	admin	Globe	nan	
adminHardening	Security Hardening	/adminhardening	admin	admin	Shield	nan	
schoolStaff	Staff Management	/schoolstaff	admin	admin	Users	nan	
dashboard	Dashboard	/dashboard	core	student,teacher,admin	Home	1	
courses	Courses	/courses	core	student,teacher,admin	BookOpen	2	
reader	Smart Reader	/reader	core	student,teacher,admin	BookMarked	3	
feed	Community	/feed	core	student,teacher,admin	Users	4	
schoolSearch	Search	/schoolsearch	core	student,teacher,admin	Search	5	
myQuizzes	My Quizzes	/my-quizzes	core	student,teacher,admin	ClipboardCheck	6	
account	My Account	/account	core	student,teacher,admin	User	10	
community	Community	/community	core	student,teacher,admin	Users	nan	
courseDetail	Course Detail	/coursedetail	core	student,teacher,admin	Book	nan	
downloads	Downloads	/downloads	core	student,teacher,admin	Download	nan	
forums	Forums	/forums	core	student,teacher,admin	MessageSquare	nan	
learningPaths	Learning Paths	/learningpaths	core	student,teacher,admin	Map	nan	
lessonViewer	Lesson Viewer	/lessonviewer	core	student,teacher,admin	Play	nan	
lessonViewerPremium	Lesson Viewer Premium	/lessonviewerpremium	core	student,teacher,admin	Play	nan	
messages	Messages	/messages	core	student,teacher,admin	Mail	nan	
myProgress	My Progress	/myprogress	core	student	TrendingUp	nan	
achievements	Achievements	/achievements	labs	student,teacher,admin	Award	nan	
adaptiveLearning	Adaptive Learning	/adaptivelearning	labs	student,teacher,admin	Brain	nan	
affiliate	Affiliate Program	/affiliate	labs	student,teacher,admin	Users	nan	
alumniNetwork	Alumni Network	/alumninetwork	labs	student,teacher,admin	Network	nan	
analytics	Analytics	/analytics	labs	student,teacher,admin	BarChart	nan	
careerPaths	Career Paths	/careerpaths	labs	student,teacher,admin	Briefcase	nan	
challenges	Challenges	/challenges	labs				

student,teacher,admin | Trophy | nan | | cohortDetail | Cohort Detail | /cohortdetail | labs |  
 student,teacher,admin | | nan | | cohorts | Cohorts | /cohorts | labs | student,teacher,admin  
 | Users | nan | | events | Events | /events | labs | student,teacher,admin | Calendar | nan | |  
 habitTracker | Habit Tracker | /habitracker | labs | student | CheckSquare | nan | |  
 languageLearning | Languages | /languagelearning | labs | student,teacher,admin | Globe |  
 nan | | languages | Languages (Legacy) | /languages | labs | student,teacher,admin | Globe  
 | nan | | leaderboard | Leaderboard | /leaderboard | labs | student,teacher,admin | Trophy |  
 nan | | liveStreams | Live Streams | /livestreams | labs | student,teacher,admin | Video |  
 nan | | marketplace | Marketplace | /marketplace | labs | student,teacher,admin | Store |  
 nan | | mentorship | Mentorship | /mentorship | labs | student,teacher,admin | Users | nan |  
 | microlearning | Microlearning | /microlearning | labs | student | Zap | nan | | studySetNew  
 | New Study Set | /studyssetnew | labs | student,teacher,admin | | nan | | offline | Offline |  
 /offline | labs | student,teacher,admin | Download | nan | | studySetPractice | Practice |  
 /studyssetpractice | labs | student,teacher,admin | | nan | | rewardsShop | Rewards Shop |  
 /rewardsshop | labs | student | Gift | nan | | scholarships | Scholarships | /scholarships |  
 labs | student | Award | nan | | skills | Skills | /skills | labs | student,teacher,admin | Target |  
 nan | | studyBuddies | Study Buddies | /studybuddies | labs | student | Users | nan | |  
 studySet | Study Set | /studysset | labs | student,teacher,admin | | nan | | studySets | Study  
 Sets | /studyssets | labs | student,teacher,admin | Layers | nan | | tournaments |  
 Tournaments | /tournaments | labs | student | Swords | nan | | affiliateProgram | Affiliate  
 Program Info | /affiliateprogram | marketing | public,student,teacher,admin | Share | nan |  
 | certificateVerify | Certificate Verification | /certificateverify | marketing | public | | nan | |  
 schoolCheckout | Checkout | /schoolcheckout | marketing | public,student,teacher,admin |  
 CreditCard | nan | | schoolCourses | Course Catalog | /schoolcourses | marketing |  
 public,student,teacher,admin | Library | nan | | schoolLanding | Landing Page |  
 /schoollanding | marketing | public,student,teacher,admin | Globe | nan | | schoolPricing |  
 Pricing | /schoolpricing | marketing | public,student,teacher,admin | DollarSign | nan | |  
 courseSales | Sales Page | /coursesales | marketing | public,student,teacher,admin |  
 ShoppingBag | nan | | schoolThankYou | Thank You | /schoolthankyou | marketing |  
 public,student,teacher,admin | | nan | | inviteAccept | Accept Invite | /inviteaccept | system  
 | public | | nan | | integrations | Integrations | /integrations | system |  
 student,teacher,admin | Plug | nan | | integrity | Integrity Check | /integrity | system |  
 admin | CheckCircle | nan | | schoolJoin | Join School | /schooljoin | system |  
 student,teacher,admin | | nan | | oauth2callback | OAuth Callback | /oauth2callback |  
 system | public | | nan | | portfolio | Profile | /portfolio | system | student,teacher,admin |  
 User | nan | | schoolSelect | School Select | /schoolselect | system | student,teacher,admin  
 | Building | nan | | subscription | Subscription | /subscription | system |  
 student,teacher,admin | | nan | | vault | Vault | /vault | system | student,teacher,admin |  
 Archive | nan | | teach | Teach | /teach | teach | teacher,admin | GraduationCap | 1 | |  
 teachAnalytics | Teaching Analytics | /teachanalytics | teach | teacher,admin | BarChart | 2  
 | | teachQuizzes | Quizzes | /teach/quizzes | teach | teacher,admin | ClipboardList | 3 | |  
 teachCourse | Course Builder | /teachcourse | teach | teacher,admin | Edit | nan | |  
 teachLesson | Lesson Editor | /teachlesson | teach | teacher,admin | FileEdit | nan | |  
 teachCourseNew | New Course | /teachcoursenew | teach | teacher,admin | Plus | nan |

## A3\_ROUTE\_MAP.md

Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A3\_ROUTE\_MAP.md

### Route Map (Generated)

- Generated: 2026-01-01
- Source: src/App.jsx, src/pages.config.js

### Explicit routes in src/App.jsx

#### Quiz routes

- /my-quizzes → <Pages.MyQuizzes />
- /quiz/:quizId → <Pages.QuizTake />
- /teach/quizzes → <Pages.TeachQuizzes />
- /teach/quizzes/new → <Pages.TeachQuizEditor />
- /teach/quizzes/:quizId → <Pages.TeachQuizEditor />

#### Storefront routes (guest-safe)

- /s/:schoolSlug → <LayoutWrapper  
currentPageName="SchoolLanding"><Pages.SchoolLanding  
/></LayoutWrapper>
- /s/:schoolSlug/courses → <LayoutWrapper  
currentPageName="SchoolCourses"><Pages.SchoolCourses  
/></LayoutWrapper>
- /s/:schoolSlug/course/:courseId → <LayoutWrapper  
currentPageName="CourseSales"><Pages.CourseSales  
/></LayoutWrapper>
- /s/:schoolSlug/pricing → <LayoutWrapper  
currentPageName="SchoolPricing"><Pages.SchoolPricing  
/></LayoutWrapper>
- /s/:schoolSlug/checkout → <LayoutWrapper  
currentPageName="SchoolCheckout"><Pages.SchoolCheckout  
/></LayoutWrapper>
- /s/:schoolSlug/thank-you → <LayoutWrapper  
currentPageName="SchoolThankYou"><Pages.SchoolThankYou  
/></LayoutWrapper>

- `/s/:schoolSlug/certificate/:certificateId → <LayoutWrapper  
currentPageName="CertificateVerify"><Pages.CertificateVerify  
></LayoutWrapper>`

## Core routing

- `/` → main page from `src/pages.config.js` (currently Dashboard) wrapped in `LayoutWrapper`
- `/:pageName` → case-insensitive page resolver that renders any configured page inside `LayoutWrapper`
- `/${PageName}` (exact-case) → explicit routes auto-generated for every page in `Pages` (kept for backwards compatibility)
- `*` → 404 handler

## Pages available via `/:pageName`

- Achievements (canonical path: `/Achievements` and also `/achievements` via `/:pageName`)
- AdaptiveLearning (canonical path: `/AdaptiveLearning` and also `/adaptivelearning` via `/:pageName`)
- AdminHardening (canonical path: `/AdminHardening` and also `/adminhardening` via `/:pageName`)
- Affiliate (canonical path: `/Affiliate` and also `/affiliate` via `/:pageName`)
- AffiliateProgram (canonical path: `/AffiliateProgram` and also `/affiliateprogram` via `/:pageName`)
- AlumniNetwork (canonical path: `/AlumniNetwork` and also `/alumninetwork` via `/:pageName`)
- Analytics (canonical path: `/Analytics` and also `/analytics` via `/:pageName`)
- CareerPaths (canonical path: `/CareerPaths` and also `/careerpaths` via `/:pageName`)
- Challenges (canonical path: `/Challenges` and also `/challenges` via `/:pageName`)
- CohortDetail (canonical path: `/CohortDetail` and also `/cohortdetail` via `/:pageName`)
- Cohorts (canonical path: `/Cohorts` and also `/cohorts` via `/:pageName`)
- Community (canonical path: `/Community` and also `/community` via `/:pageName`)
- CourseDetail (canonical path: `/CourseDetail` and also `/coursedetail` via `/:pageName`)
- CourseSales (canonical path: `/CourseSales` and also `/coursesales` via `/:pageName`)
- Courses (canonical path: `/Courses` and also `/courses` via `/:pageName`)
- Dashboard (canonical path: `/Dashboard` and also `/dashboard` via `/:pageName`)

- Downloads (canonical path: /Downloads and also /downloads via /:pageName)
- Events (canonical path: /Events and also /events via /:pageName)
- Feed (canonical path: /Feed and also /feed via /:pageName)
- Forums (canonical path: /Forums and also /forums via /:pageName)
- HabitTracker (canonical path: /HabitTracker and also /habittracker via /:pageName)
- Integrations (canonical path: /Integrations and also /integrations via /:pageName)
- Integrity (canonical path: /Integrity and also /integrity via /:pageName)
- LanguageLearning (canonical path: /LanguageLearning and also /languagelearning via /:pageName)
- Languages (canonical path: /Languages and also /languages via /:pageName)
- Leaderboard (canonical path: /Leaderboard and also /leaderboard via /:pageName)
- LearningPaths (canonical path: /LearningPaths and also /learningpaths via /:pageName)
- LegacyMigration (canonical path: /LegacyMigration and also /legacymigration via /:pageName)
- LessonViewer (canonical path: /LessonViewer and also /lessonviewer via /:pageName)
- LessonViewerPremium (canonical path: /LessonViewerPremium and also /lessonviewerpremium via /:pageName)
- LiveStreams (canonical path: /LiveStreams and also /livestreams via /:pageName)
- Marketplace (canonical path: /Marketplace and also /marketplace via /:pageName)
- Mentorship (canonical path: /Mentorship and also /mentorship via /:pageName)
- Messages (canonical path: /Messages and also /messages via /:pageName)
- Microlearning (canonical path: /Microlearning and also /microlearning via /:pageName)
- MyProgress (canonical path: /MyProgress and also /myprogress via /:pageName)
- Offline (canonical path: /Offline and also /offline via /:pageName)
- Portfolio (canonical path: /Portfolio and also /portfolio via /:pageName)
- Reader (canonical path: /Reader and also /reader via /:pageName)
- RewardsShop (canonical path: /RewardsShop and also /rewardsshop via /:pageName)
- Scholarships (canonical path: /Scholarships and also /scholarships via /:pageName)
- SchoolAdmin (canonical path: /SchoolAdmin and also /schooladmin via /:pageName)
- SchoolAnalytics (canonical path: /SchoolAnalytics and also /schoolanalytics via /:pageName)

- SchoolCheckout (canonical path: /SchoolCheckout and also /schoolcheckout via /:pageName)
- SchoolCourses (canonical path: /SchoolCourses and also /schoolcourses via /:pageName)
- SchoolJoin (canonical path: /SchoolJoin and also /schooljoin via /:pageName)
- SchoolLanding (canonical path: /SchoolLanding and also /schoollanding via /:pageName)
- SchoolMonetization (canonical path: /SchoolMonetization and also /schoolmonetization via /:pageName)
- SchoolNew (canonical path: /SchoolNew and also /schoolnew via /:pageName)
- SchoolPricing (canonical path: /SchoolPricing and also /schoolpricing via /:pageName)
- SchoolSearch (canonical path: /SchoolSearch and also /schoolsearch via /:pageName)
- SchoolSelect (canonical path: /SchoolSelect and also /schoolselect via /:pageName)
- SchoolThankYou (canonical path: /SchoolThankYou and also /schoolthankyou via /:pageName)
- Skills (canonical path: /Skills and also /skills via /:pageName)
- StudyBuddies (canonical path: /StudyBuddies and also /studybuddies via /:pageName)
- StudySet (canonical path: /StudySet and also /studysset via /:pageName)
- StudySetNew (canonical path: /StudySetNew and also /studyssetnew via /:pageName)
- StudySetPractice (canonical path: /StudySetPractice and also /studyssetpractice via /:pageName)
- StudySets (canonical path: /StudySets and also /studyssets via /:pageName)
- Subscription (canonical path: /Subscription and also /subscription via /:pageName)
- Teach (canonical path: /Teach and also /teach via /:pageName)
- TeachAnalytics (canonical path: /TeachAnalytics and also /teachanalytics via /:pageName)
- TeachCourse (canonical path: /TeachCourse and also /teachcourse via /:pageName)
- TeachCourseNew (canonical path: /TeachCourseNew and also /teachcoursenew via /:pageName)
- TeachLesson (canonical path: /TeachLesson and also /teachlesson via /:pageName)
- TeachQuizzes (canonical path: /TeachQuizzes and also /teachquizzes via /:pageName)



- TeachQuizEditor (canonical path: /TeachQuizEditor and also /teachquizeditor via /:pageName)
- QuizTake (canonical path: /QuizTake and also /quiztake via /:pageName)
- MyQuizzes (canonical path: /MyQuizzes and also /myquizzes via /:pageName)
- Tournaments (canonical path: /Tournaments and also /tournaments via /:pageName)
- Vault (canonical path: /Vault and also /vault via /:pageName)
- oauth2callback (canonical path: /oauth2callback and also /oauth2callback via /:pageName)
- Account (canonical path: /Account and also /account via /:pageName)
- NetworkAdmin (canonical path: /NetworkAdmin and also /networkadmin via /:pageName)
- CertificateVerify (canonical path: /CertificateVerify and also /certificateverify via /:pageName)
- SchoolStaff (canonical path: /SchoolStaff and also /schoolstaff via /:pageName)
- InviteAccept (canonical path: /InviteAccept and also /inviteaccept via /:pageName)
- AuditLogViewer (canonical path: /AuditLogViewer and also /auditlogviewer via /:pageName)

## **A4\_ENTITIES\_TENANCY\_SCOPING.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A4\_ENTITIES\_TENANCY\_SCOPING.md

### **Data Entities & Tenancy Scoping (Generated)**

- Generated: 2026-01-01
- Source: src/components/api/scopedEntities.js

#### **Global entities (NOT school-scoped)**

- User
- School
- Google0AuthToken
- GoogleDriveToken
- Notification
- Integration
- UserSchoolPreference

#### **School-scoped entities (MUST include school\_id filter)**

Total: 56

- Course
- Lesson
- Post
- Comment
- UserProgress
- Offer
- Coupon
- Transaction
- Entitlement
- Purchase
- StudySet
- StudyCard

- LanguageVariant
- SpacedRepetitionItem
- Cohort
- CohortMember
- CohortScheduleItem
- SchoolMembership
- SchoolInvite
- StaffInvite
- ContentProtectionPolicy
- Testimonial
- InstructorPayout
- AuditLog
- EventLog
- Announcement
- UserAnnouncementRead
- Affiliate
- Referral
- AiTutorSession
- AiTutorPolicyLog
- RateLimitLog
- Bookmark
- LessonNote
- Highlight
- Transcript
- Text
- CourseReview
- Discussion
- ContentReport
- ModerationAction
- SchoolMetricDaily
- CourseMetricDaily
- Assignment
- Submission
- Quiz

- QuizAttempt
- QuizQuestion
- Download
- Bundle
- SubscriptionPlan
- PayoutBatch
- AnalyticsEvent
- SubscriptionInvoice
- Certificate
- StudySession

## A5\_KEY\_FILES.md

Breslov\_Academy\_Coding\_Book\_v9\_1/APPENDICES\_GENERATED/A5\_KEY\_FILES.md

# Key Files (Reality Map)

Generated: 2026-01-01

- `src/main.jsx` — App bootstrap; mounts React; BrowserRouter entry.
- `src/App.jsx` — Primary route table (quiz/storefront/pages/404) + auth gating wrapper.
- `src/pages.config.js` — Defines Pages map, mainPage, Layout binding.
- `src/Layout.jsx` — Current global app shell (nav/sidebar) + logout + feature gating.
- `src/lib/AuthContext.jsx` — Auth state machine: `me()`, `redirectToLogin()`, `logout()`; public settings check.
- `src/components/hooks/useSession.jsx` — Session + preferred school selection + membership resolution.
- `src/components/school/SchoolContext.jsx` — School context provider (active school, membership).
- `src/components/api/scoped.jsx` — Helpers for school-scoped entity queries.
- `src/components/api/scopedEntities.js` — Single source list of school-scoped entities.
- `src/components/api/tenancyEnforcer.js` — Runtime guard: prevents unscoped access.
- `src/components/hooks/useLessonAccess.jsx` — Content protection engine: LOCKED/DRIP\_LOCKED gating.
- `src/components/config/features.jsx` — Canonical feature registry.
- `src/components/utils/featureRegistry.jsx` — Adapter layer re-export; used across app.
- `src/components/navigation/CommandPalette.jsx` — Command palette / vault surfacing.

## Part IV — SpecPack v5 (Source Docs)

### 00\_INDEX.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/00\_INDEX.md

## Spec Pack Index — v9.0 Stable (Reality Integrated)

Prepared: January 01, 2026 (Asia/Jerusalem)

### Files

- 00\_INDEX.md
- ARCHITECTURE\_NORTH\_STAR.md
- BACKLOG\_STORIES.md
- BRANDING\_TERMINOLOGY\_POLICY.md
- CI\_CD\_PIPELINE.md
- COMMAND\_PALETTE\_SPEC.md
- CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md
- DESIGN\_SYSTEM\_UI\_UX.md
- FEATURE\_REGISTRY\_AND\_VAULT.md
- FUTURE\_BUILDS\_PLAYBOOK.md
- GITHUB\_SETUP\_WORKFLOW.md
- IA\_ROUTES\_PORTALS.md
- KNOWN\_ISSUES\_TECH\_DEBT.md
- MIGRATIONS\_DATA\_MODEL.md
- NEXT\_CHAT\_CONTEXT.md
- OPS\_INTEGRITY\_QA.md
- PORTALIZATION\_BUILD\_SCRIPT\_V9\_1.md
- PORTAL\_FEATURE\_MATRIX.md
- PR\_CHECKLIST.md
- QA\_RELEASE\_CHECKLISTS.md
- RBAC\_TENANCY\_MODEL.md
- README.md
- RELEASE\_TRAINS\_FUTURE\_BUILDS.md

- `RELEASE_TRAIN_CONTINUOUS_BUILDS.md`
- `REPO_REALITY_AUDIT.md`
- `ROADMAP_V9_DETAILED.md`
- `SECURITY_THREAT_MODEL.md`
- `STORE_MONEY_SUBSCRIPTIONS.md`
- `V9_AUTH_ONBOARDING_REALITY.md`
- `V9_CODEBASE_REALITY_MAP.md`
- `V9_CONTENT_PROTECTION_REALITY.md`
- `V9_FEATURE_REGISTRY_REALITY.md`
- `V9_PUBLIC_SITE_AND_PORTALS_PLAN.md`
- `V9_QA_RELEASES.md`
- `V9 QUIZZES_REALITY.md`
- `V9_ROADMAP_EXTREME_DETAIL.md`
- `V9_ROUTES_REALITY_MAP.md`
- `V9_TENANCY_REALITY.md`
- `V9_UI_UX_REVAMP_PLAN.md`

## How to use this spec pack

1) Commit these docs under docs/ in the repo. 2) In the next coding chat, follow `PORTALIZATION_BUILD_SCRIPT_V9_1.md` first (v9.1.x), then `V9_ROADMAP_EXTREME_DETAIL.md`. 3) Use `RELEASE_TRAIN_CONTINUOUS_BUILDS.md` to plan ongoing releases.

## **ARCHITECTURE\_NORTH\_STAR.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/ARCHITECTURE\_NORTH\_STAR.md

# **Architecture North Star (v9.0)**

Prepared: January 01, 2026 (Asia/Jerusalem)

This document defines the target architecture that keeps all existing features while adding:

- Public marketing website
  - Separate Student/Teacher portals + separate login entry points
  - School/Entity signup + onboarding (multi-tenant)
  - Full premium UI/UX revamp (WOW, calm, academic)
- 

## **1) App Shell (React/Vite/Base44-style assumptions)**

The existing project references a Vite alias of @ -> ./src and a Base44-style router/pages structure. Do not assume exact paths; Phase 0 audit must confirm real file locations.

### **1.1 Single session source-of-truth**

Create or harden a session hook/provider that exposes:

- user (identity)
- memberships[] (tenant memberships + roles)
- activeSchoolId + activeSchoolSlug
- audience (student | teacher | admin | superadmin)
- isLoading, isAuthenticated

Rules:

- Audience is derived from role claims and/or portal entry point.
- Active school is chosen by:
  - 1) explicit selector (for multi-tenant users)
  - 2) last-used in localStorage (scoped)
  - 3) fallback to first membership
- Store only non-sensitive session hints client-side.

### **1.2 Multi-surface routing**



We split surfaces by route groups:

- Public marketing: /, /about, /faq, etc.
- Student portal: /student/\*
- Teacher portal: /teacher/\*
- School admin: /admin/\*
- Platform admin: /superadmin/\* (global-only)
- Public storefront: /s/:schoolSlug/\*

Legacy routes must remain working via:

- alias routes in registry
- redirect adapters that preserve query strings

### 1.3 Guard layers

Every protected surface requires *\*both\**:

- Route guard: blocks UI rendering if role/tenant missing
  - Data guard: all queries/mutations are server/app-layer protected by tenant + entitlement
- 

## 2) Tenant & RBAC Model

### 2.1 Core concepts

- Tenant (School/Entity) — owns courses, lessons, offers, settings, staff, branding.
- Membership — ties user to tenant + role(s).
- Role — student/teacher/admin/owner/superadmin

### 2.2 Tenant scoping enforcement

- Never trust client-provided `tenant_id`.
  - Derive tenant scope from:
    - `activeSchoolId` for authenticated portal requests
    - `schoolSlug` -&gt; `school_id` for public storefront
-

## 3) Content Protection & Materials Engine

### 3.1 Access states (MUST)

- LOCKED — no entitlement
- DRIP\_LOCKED — has entitlement, but not released yet
- UNLOCKED — entitled + released

Hard rule: In LOCKED or DRIP\_LOCKED, lesson materials must not be fetched, rendered, mounted, nor passed into components.

### 3.2 Materials engine contract

The engine returns:

- `accessState`
  - `reasons[]` (why locked)
  - `unlockAt` (for drip)
  - `preview` (if allowed by policy)
  - `materials` only when UNLOCKED
- 

## 4) Feature Registry + Vault as the anti-loss safety net

One typed registry drives:

- Main navigation (streamlined)
- Vault (full directory)
- Optional command palette
- Integrity checks

Registry must be a central module (`src/config/features.*` or `src/registry/featureRegistry.*`). Never inline registry arrays in UI pages.

---

## 5) Observability & Integrity Contract

An admin-only `/integrity` diagnostic must:

- validate registry vs router existence (best-effort)
- validate scoping heuristics

- validate leakage heuristics (materials not fetched while locked)
  - export JSON report
- 

## **6) Delivery constraints**

- Extend-not-replace: prefer additive changes and adapters.
- No feature deletions. If nav is simplified, keep feature in Vault.
- “Build must pass” is a requirement — fix root causes (aliases/imports/guards), not symptoms.

## **BACKLOG\_STORIES.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/BACKLOG\_STORIES.md

# **Backlog (Stories + Acceptance Criteria) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This backlog is intentionally verbose so the next coding chat can execute it like a checklist. As soon as the v9.0 zip is uploaded, Phase 0 Audit must replace any “TBD” placeholders with real file paths/routes/entities.

---

## **Epic A — Surface Split (Public + Student + Teacher + Admin)**

### **A1 — Public Marketing Shell**

Story: As a guest, I can browse a premium marketing site and reach the correct login/signup.

- Tasks:
- Implement marketing layout shell (header/footer, responsive).
- Add routes: /, /about, /how-it-works, /faq, /contact, /privacy, /terms.
- Ensure nav CTAs exist and are correct.
- Acceptance:
- All pages render unauthenticated.
- Lighthouse basics: no blocking errors; text readable on mobile.

### **A2 — Separate Login Entry Points**

Story: As a user, I can choose Student vs Teacher login and land in the right portal.

- Tasks:
- Implement /login/student, /login/teacher (+ optional /login/admin).
- Preserve legacy /login by redirecting to a chooser.
- Persist chosen audience during auth redirect/callback.
- Acceptance:
- Student login routes to /student/dashboard.
- Teacher login routes to /teacher/dashboard (or closest existing page).
- If user lacks the chosen role, show a helpful “wrong portal” screen with a switch button.

### **A3 — Portal Layout Shells + Guards**

Story: As a logged-in user, I see the correct portal navigation and cannot access other portals.

- Tasks:
  - Create layouts: StudentShell, TeacherShell, AdminShell, SuperAdminShell.
  - Add guard wrappers:
  - Auth required
  - Role required
  - Active tenant required
  - Add “Choose school” modal for multi-tenant users.
  - Acceptance:
  - Unauthorized portal routes redirect to appropriate login.
  - Cross-tenant access is blocked.
  - Active school selector persists safely.
- 

## **Epic B — Registry + Vault + Integrity**

### **B1 — Central Feature Registry (single source of truth)**

Story: As a developer/admin, all features are registered and discoverable.

- Tasks:
- Create/restore registry module.
- Populate from router map (do not guess).
- Add legacy aliases for deep links.
- Acceptance:
- Registry lists every page/route.
- No inline registries exist inside Vault/nav.

### **B2 — Vault Directory**

Story: As a user, I can find any feature (even “hidden” ones) via Vault.

- Tasks:
- Vault renders from registry.
- Add filters (area, audience) + search.
- Add “open” buttons.

- Acceptance:
- Vault never renders links user cannot access.
- Vault exposes all accessible features.

### **B3 — Integrity Diagnostics (admin-only)**

Story: As an admin, I can run integrity checks and export a report.

- Tasks:
  - Implement /integrity page.
  - Checks: registry vs router, tenancy heuristics, leakage heuristics.
  - Add “Export report” JSON.
  - Acceptance:
  - Page never crashes; shows warnings instead.
  - Export works and includes timestamp + tenant id.
- 

## **Epic C — Signup & Onboarding**

### **C1 — Student signup controls**

- Invite code flow
- Domain match flow
- Approval queue flow

Acceptance:

- A student cannot join a tenant without meeting one of the configured rules.

### **C2 — Teacher signup controls**

- Invite-only preferred
- Approval fallback

Acceptance:

- Teacher role is never self-assigned without admin approval.

### **C3 — School/Entity signup**

- Create pending tenant application
- Create owner/admin account

- Admin onboarding wizard

Acceptance:

- Tenant exists in pending until approved.
- After approval, admin can complete onboarding.

## **C4 — Invitations lifecycle**

- Create invitations (role + tenant + expiry)
- Accept once; invalidate token
- Revoke membership; audit log

Acceptance:

- Tokens cannot be reused.
  - Revoke removes access immediately.
- 

## **Epic D — Design System + Revamp**

### **D1 — Theme tokens**

- Light/dark/night
- Typography scale

Acceptance:

- No pure-white backgrounds; reading surfaces comfortable.

### **D2 — Shared components**

Acceptance:

- All portals use the shared button/input/table/card.
- Tables collapse to cards on mobile.

### **D3 — Module conversions (safe order)**

- Dashboard -> Courses -> Lesson chrome -> Admin settings -> Monetization -> Tools

Acceptance:

- Each conversion preserves behavior; adapters used where needed.
-

## **Epic E — Security + Content Protection Stability**

### **E1 — Lesson access gating correctness**

Acceptance:

- LOCKED/DRIP\_LOCKED: no material fetch/render/mount.
- UNLOCKED: materials engine can fetch/render.

### **E2 — Downloads protection**

Acceptance:

- No URL visible unless authorized; “Get download” logs event.

### **E3 — Search leakage prevention**

Acceptance:

- Search is metadata-only; never queries full transcripts for unauthorized.

### **E4 — Rate limiting / abuse protections**

Acceptance:

- Auth endpoints and download reveal are rate-limited (best-effort).
- 

## **Epic F — Storefront + Monetization polish**

- Branding surfaces
- Checkout idempotency
- Coupons
- Affiliate dashboards + payouts

Acceptance:

- Approving same transaction twice creates no duplicate entitlements.
  - Creating same payout batch twice creates no duplicates.
-



## **Epic G — Release & Recovery**

- Update CHANGELOG
- Update RECOVERY “how to recover” steps
- Export integrity report sample

Acceptance:

- Someone can resume work after interruption using RECOVERY.

## **BRANDING\_TERMINOLOGY\_POLICY.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/BRANDING\_TERMINOLOGY\_POLICY.md

# **Branding, Terminology, and Policy Settings — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

These settings must be tenant-scoped and admin-editable.

---

## **1) Branding settings (per tenant)**

- Name + short name
- Logo and favicon
- Accent color (mapped into design tokens)
- Optional domain/custom URL (if supported)
- Storefront hero copy + tagline

Where it should live:

- Admin portal: /admin/settings -> Branding tab
- 

## **2) Terminology settings (per tenant)**

Goal: UI labels should adapt to each school. Examples:

- Teacher: Rabbi / Rav / Instructor
- Student: talmid / bachur / learner
- Course: shiur / course
- Lesson: lesson / shiur

Where it should live:

- Admin portal: /admin/settings -> Terms tab

Rule:

- If terminology system already exists, use it (do not replace).
- If missing, implement minimally with safe fallbacks.

### **3) Content protection policy (per tenant)**

Configurable:

- Preview limits (video minutes, text paragraphs)
- Watermark
- Copy/print blocking (UI layer)
- Download/copy license add-ons

Where it should live:

- Admin portal: /admin/settings -> Protection tab
- 

### **4) Monetization settings (per tenant)**

- Offers, coupons, bundles, subscriptions
- Payout settings
- Affiliate settings

Where it should live:

- Admin portal: /admin/monetization

## **CI\_CD\_PIPELINE.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/CI\_CD\_PIPELINE.md

# **CI/CD Plan (GitHub Actions) — v9+**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **CI on pull requests**

Recommended jobs: 1) Install (cache deps) 2) Lint 3) Typecheck (if TS) 4) Unit tests (gating + access state + idempotency) 5) Build (production) 6) Dependency audit (best-effort) 7) Upload artifact (optional)

## **CD on tags**

- Tag stable releases: v9.0.0, v9.0.1, v9.1.0
- Generate release notes from CHANGELOG
- Optional: attach build artifact to GitHub Release

## **Merge gates**

- CI must be green
- QA ship gate must be green for stable tags
- Integrity report sample stored under docs/integrity-samples/ (for stable)

## **COMMAND\_PALETTE\_SPEC.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/COMMAND\_PALETTE\_SPEC.md

# **Command Palette Spec (Optional, Premium UX) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

If the project already includes a command palette dependency (or it's easy to add safely), implement a Cmd/Ctrl+K palette that searches Feature Registry.

---

## **Requirements**

- Opens with Cmd/Ctrl+K
- Lists features filtered by user permissions
- Can optionally show vaultOnly items (toggle or separate section)
- Selecting navigates to the canonical route

Fallback:

- If dependency missing and adding is risky, skip without breaking build.

## **CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md

# **Content Protection + Materials Engine Spec — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This is the heart of the platform. It must never leak protected material.

---

## **1) Threat model (what must never happen)**

- Guest sees full lesson text/transcripts.
  - Locked user receives direct file URLs.
  - DRIP\_LOCKED user fetches materials early “in the background.”
  - Search indexes full text and exposes it to unauthorized users.
  - A component mounts material payload even briefly before gate renders.
- 

## **2) Access state machine**

Inputs:

- user session + membership
- entitlement(s) (active, unexpired)
- drip schedule (release logic)
- admin policy (preview limits, watermark, copy/download licensing)
- lesson/courses flags

Outputs:

- LOCKED (no entitlement)
  - DRIP\_LOCKED (entitled but unreleased)
  - UNLOCKED (entitled + released)
-

### 3) Hard enforcement rules

- In LOCKED or DRIP\_LOCKED, the UI must render AccessGate only.
  - Do not call materials fetch functions.
  - Do not render child viewer components that expect material.
  - Do not pass placeholder material objects downstream.
- 

### 4) Materials Engine API

Suggested functions:

- `getLessonAccess({lessonId, userId, tenantId})` -> { state, unlockAt, reasons, previewAllowed }
- `getLessonMaterials({lessonId, userId, tenantId})` -> materials (must assert UNLOCKED)
- `getDownloadUrl({fileId, userId, tenantId})` -> signedOrEphemeralUrl (must log)

Implementation patterns:

- One hook: `useLessonAccess(lessonId)` returns state and UI gating.
  - Viewer mounts only when state === 'UNLOCKED'.
- 

### 5) Preview policy

Admin-configurable:

- preview minutes for video
  - preview paragraphs for text
  - watermark configuration
  - copy/print/right-click blocking toggles (UI layer + logging)
  - “copy/download license” add-ons
- 

### 6) Search leakage prevention

Search results:

- metadata only: title, tags, snippet from allowed preview
- never query full transcripts or full text for unauthorized users
- log suspicious repeated search requests



## **DESIGN\_SYSTEM\_UI\_UX.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/DESIGN\_SYSTEM\_UI\_UX.md

# **Design System + UI/UX Spec (WOW + calm + academic) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This document defines the design system that must be reused across all surfaces.

---

## **1) Design goals**

- University-grade: structured, confident, not flashy.
  - Eye-strain friendly: no harsh whites; comfortable reading surfaces.
  - Clear hierarchy: predictable layouts, strong typography, generous spacing.
  - Accessibility-first: keyboard navigation, focus rings, contrast, reduced motion.
  - Delight without distraction: subtle motion, premium micro-interactions.
- 

## **2) Theming + tokens**

### **2.1 Color system**

Define tokens (example naming; adapt to existing system):

- `bg.canvas`, `bg.surface`, `bg.elevated`, `bg.overlay`
- `text.primary`, `text.secondary`, `text.muted`
- `border.default`, `border.subtle`
- `brand.primary`, `brand.accent`
- semantic: `success`, `warning`, `error`, `info`

Modes:

- Light (soft ivory/gray, not pure white)
- Dark
- Night Reading (extra low contrast + warm)

## 2.2 Typography

- Set a readable font stack and a consistent scale:
- Display / H1 / H2 / H3
- Body
- Small
- Caption
- Increase line-height for lesson text.
- Use max-widths for reading (e.g., 60–75ch).

## 2.3 Spacing & layout

- 8pt grid.
  - Consistent breakpoints.
  - Layout patterns:
  - Dashboard: header + cards
  - Data pages: filters + table/cards
  - Reader: centered column + sticky TOC
- 

## 3) Component library (must be shared)

### 3.1 Base components

- Button (primary/secondary/ghost/destructive), IconButton
- Input, Textarea, Select, Combobox, DatePicker
- Tabs, Segmented control
- Modal/Dialog, Drawer
- Tooltip, Popover, DropdownMenu
- Toasts/notifications
- Badge/StatusPill
- Table (responsive: collapses to cards on mobile)
- Card
- Skeleton loaders
- EmptyState blocks

### **3.2 LMS components**

- CourseCard (cover + meta + CTA)
- LessonViewerChrome (title, progress, actions)
- LessonAccessGate (locked/drip/unlocked UI)
- ProgressWidget (course + lesson)
- GradebookTable + rubric UI
- AssignmentComposer + SubmissionViewer
- MessageComposer + ThreadList
- Timeline/Schedule for drip

### **3.3 Storefront components**

- SchoolHero (branding)
  - OfferCard, PricingTable, BundleTile
  - CheckoutSummary, CouponInput, PaymentCTA
  - TrustBadges, Testimonials, FAQ accordion
- 

## **4) UX patterns**

- Loading: skeletons (avoid spinners for whole pages).
  - Empty states: explain “why” + one primary action.
  - Inline validation: friendly, exact error messages.
  - Reduced motion: respect prefers-reduced-motion.
- 

## **5) Revamp strategy (no breakage)**

1) Replace global shell first (layout, nav, tokens). 2) Convert modules behind stable routes. 3) Preserve backend behavior; use adapters, not rewrites. 4) Polish pass last: animation, accessibility audit, perf budgets.

## FEATURE\_REGISTRY\_AND\_VAULT.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/FEATURE\_REGISTRY\_AND\_VAULT.md

# Feature Registry + Vault Spec — v9.0

Prepared: January 01, 2026 (Asia/Jerusalem)

Goal: A single registry drives navigation, Vault, command palette, and integrity diagnostics.

---

## 1) Registry module

Location (choose one; confirm in Phase 0):

- `src/config/features.ts` (or `.js`)
- OR `src/registry/featureRegistry.ts`

Registry entry schema:

- `key` (string, stable id)
- `label`
- `description` (short)
- `route` (canonical)
- `aliases[]` (legacy)
- `area` (core | teach | admin | marketing | labs | system | vault)
- `audiences[]` (student | teacher | admin | superadmin)
- `vaultOnly` (boolean)
- `showInMainNav` (boolean)
- `order` (number)
- `requiredRoles[]` (optional)

Helpers:

- `getFeatureByKey(key)`
- `getAllFeatures()`
- `getVaultFeatures(options)` where `options` includes audience, roles, tenant
- `getMainNavFeatures(options)` where `options` includes audience, roles, tenant
- `getAllRoutes()`

- `normalizeAudienceFromRole(roleString)`

Rules:

- Do not inline feature arrays in any UI page (Vault/nav).
  - Vault must render full accessible directory (filtered only by permissions).
  - Main nav is streamlined; everything else stays in Vault.
- 

## 2) Vault requirements

- Search/filter by label/area/audience.
  - Badges: area + vaultOnly + audiences.
  - “Open” button navigates to route.
  - If feature is not accessible, don’t render it.
- 

## 3) Legacy route compatibility

- All old routes must be in `aliases[]`.
  - Add redirect adapters for common deep links.
  - Preserve query string and hash fragments.
- 

## 4) Integrity checks

Admin-only diagnostic (page can be vaultOnly):

- Feature count and list
- Route list
- Best-effort route existence check vs router map
- Warnings for features missing pages
- Link to Vault + key admin pages

Export report JSON:

- timestamp

- tenant/school id
- checks [{

```
"key": "string",  
"status": "pass|warn|fail",  
"message": "string"
```

```
}]
```

- warnings []
- environment { "appVersion": "string?", "buildTime": "string?" }

## **FUTURE\_BUILDS\_PLAYBOOK.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/FUTURE\_BUILDS\_PLAYBOOK.md

# **Future Builds — Detailed Playbook (Always Improving)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **The loop (repeat every build)**

1) Measure (activation, learning, money, reliability) 2) Pick 3-5 high-impact items + 5-10 quality items 3) Ship safely (small PRs, flags if risky) 4) QA + integrity export 5) Review + document + update backlog

## **Parallel tracks (always-on)**

- UX polish
- Security/trust
- Performance/stability
- Conversion/onboarding

## **Never-regress guard rails**

- Feature registry diff check each build
- Locked-content network-call monitor (best-effort)
- Tenant-scope tracer (best-effort)
- Money idempotency tests

## GITHUB\_SETUP\_WORKFLOW.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/GITHUB\_SETUP\_WORKFLOW.md

# GitHub Setup + Day-to-Day Workflow — Breslov Academy

Prepared: January 01, 2026 (Asia/Jerusalem)

Repo:

- HTTPS: `https://github.com/yeshivabachur/breslov-academy.git`
  - SSH: `git@github.com:yeshivabachur/breslov-academy.git`
  - GitHub CLI: `gh repo clone yeshivabachur/breslov-academy`
- 

## 1) Clone

### HTTPS

Code (bash):

```
git clone https://github.com/yeshivabachur/breslov-academy.git
cd breslov-academy
```

### SSH (recommended if you use SSH keys)

Code (bash):

```
git clone git@github.com:yeshivabachur/breslov-academy.git
cd breslov-academy
```

### GitHub CLI

Code (bash):

```
gh repo clone yeshivabachur/breslov-academy
cd breslov-academy
```

---

## 2) Branching model (simple + safe)

- `main` = always deployable (PR-only)
- optional: `release/v9.x` = stabilization branch if you want to lock releases



- feature branches:
- feat/<topic>;
- fix/<topic>;
- chore/<topic>;

Rules:

- No direct commits to main.
  - PRs must pass CI + QA gates before merge.
  - For risky changes, use feature flags.
- 

### **3) Commit convention (keeps changelog clean)**

Examples:

- feat(student): add portal shell + route guards
  - fix(gating): prevent materials fetch when locked
  - chore(ci): add lint + typecheck actions
  - docs(spec): update roadmap and QA gates
- 

### **4) PR discipline**

- Keep PRs small and scoped.
- Every PR must preserve:
- Vault discoverability
- tenant scoping
- content gating (no material fetch while locked)
- idempotent purchases

## **IA\_ROUTES\_PORTALS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/IA\_ROUTES\_PORTALS.md

# **IA + Route Map (Public + Portals + Storefront) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This is the canonical route map. It must be implemented without breaking existing routes.

---

## **1) Public Marketing Website (Trust + Conversion)**

Routes (required):

- / — landing (premium hero, value props, CTA)
- /about
- /how-it-works
- /faq
- /contact
- /privacy
- /terms

Top nav (required):

- Student Login -> /login/student
- Teacher Login -> /login/teacher
- Sign Up -> /signup (if enabled)
- School/Entity Sign Up -> /signup/school (or Request Demo)

Implementation:

- Public site uses marketing layout shell.
  - Must not require authentication.
- 

## **2) Auth Entry Points (Separate Login Pages)**

- /login/student

- /login/teacher
- /login/admin (optional)
- /login (legacy) -> redirect to chooser

Rules:

- All login pages use same underlying auth mechanism.
  - Each login page sets desired audience, then routes to correct portal post-login.
- 

### **3) Signup & Onboarding**

#### **3.1 Individual signup**

- /signup (chooser)
- /signup/student
- /signup/teacher

Controls:

- Student: invite code OR domain match OR pending approval.
- Teacher: invite-only OR pending approval.

#### **3.2 School/Entity signup (Tenant creation)**

- /signup/school

Flow: 1) Submit org info -> create TenantApplication or Tenant in pending 2) Create Owner/Admin account or tie to existing user 3) Tenant onboarding wizard in /admin/onboarding

---

### **4) Portals (Separate Layouts)**

#### **4.1 Student portal: /student/\***

Minimum pages:

- /student/dashboard
- /student/courses
- /student/course/:courseId

- /student/lesson/:lessonId (lesson viewer)
- /student/assignments
- /student/progress
- /student/messages
- /student/account (or /account alias)

#### **4.2 Teacher portal:** /teacher/\*

Minimum pages:

- /teacher/dashboard
- /teacher/classes
- /teacher/course-builder (or existing course builder route)
- /teacher/grading
- /teacher/analytics
- /teacher/messages

#### **4.3 School admin portal:** /admin/\*

Minimum pages:

- /admin/onboarding
- /admin/users
- /admin/invites
- /admin/classes
- /admin/settings (branding, terminology, protection policy)
- /admin/approvals
- /admin/monetization (if already exists)
- /admin/audit-logs
- /admin/migration-tools
- /admin/integrity (or /integrity vault-only)

#### **4.4 Platform admin:** /superadmin/\*

Minimum pages:

- /superadmin/tenants
- /superadmin/users
- /superadmin/risk-audit

- /superadmin/impersonation
- /superadmin/system

Lockdown:

- Global role only
  - Must be audited
- 

## 5) Public Storefront (Per school slug)

Canonical:

- /s/:schoolSlug (school landing)
- /s/:schoolSlug/catalog
- /s/:schoolSlug/course/:courseId (sales page)
- /s/:schoolSlug/pricing (or offers)
- /s/:schoolSlug/checkout
- /s/:schoolSlug/thank-you

Rules:

- Guests may browse marketing + storefront.
  - Lesson content never shown here beyond preview policy.
- 

## 6) Legacy routes & compatibility

During Phase 0 audit, enumerate all existing routes and:

- set each to a canonical route in the registry
- add `aliases[]` for old routes
- add redirect adapters preserving query strings

Deliverable:

- Append a “Legacy Route Map” section after audit listing each old route -> new route.

## **KNOWN\_ISSUES\_TECH\_DEBT.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/KNOWN\_ISSUES\_TECH\_DEBT.md

### **Known Issues / Tech Debt (Seed List)**

Prepared: January 01, 2026 (Asia/Jerusalem)

This list starts from the v9.0 stable ZIP inspection.

#### **Registry gaps**

- [ ] TeachQuizEditor page exists but is missing from FEATURES registry.
- [ ] QuizTake page exists but is missing from FEATURES registry.

#### **Portal split not implemented yet**

- [ ] No /student/\*, /teacher/\*, /admin/\* route groups yet.
- [ ] No /login/student and /login/teacher pages yet.

#### **Public marketing site not implemented yet**

- [ ] No /about, /how-it-works, /faq, /contact, /privacy, /terms pages.

#### **Security follow-ups**

- [ ] Audit for any direct calls to  
base44.entities.<scoped>.create/update/delete
- [ ] Audit lesson viewer + quizzes for any fetch while LOCKED/DRIP\_LOCKED

#### **Documentation**

- [ ] Ensure docs live under /docs in the real repo and are kept current

## **MIGRATIONS\_DATA\_MODEL.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/MIGRATIONS\_DATA\_MODEL.md

# **Data Model + Migrations (Extend-not-replace) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This is conceptual; align to existing Base44 entity JSONs and safely migrate.

---

## **1) Entities (minimum)**

Tenant + auth:

- School / Tenant
- User
- Membership
- Invitation
- OrgApplication / TenantApplication

Learning:

- Course, Lesson
- Enrollment
- Progress
- Assignment, Submission, Grade
- MessageThread, Message

Content protection:

- LessonMaterial (if present) or equivalent material store
- ProtectionPolicy (per tenant)
- DownloadAttempt / BlockedAttempt logs

Storefront:

- Offer, Bundle
- Purchase, Transaction
- Entitlement

- `Coupon`, `CouponRedemption`
- `SubscriptionPlan`, `Subscription`, `SubscriptionInvoice`
- `Affiliate`, `ReferralClick`, `Commission`
- `PayoutBatch`, `PayoutRecord`

Ops:

- `AuditLog`
  - `AnalyticsEvent`
- 

## 2) Migration rules

- Never delete fields in a “stable” pass.
  - Prefer additive changes: new fields with defaults.
  - Write backfill tools:
  - `MigrationTools` page: runs safe backfills, previews changes, emits audit logs.
- 

## 3) Tenant scoping additions

For any entity missing `school_id`, add it (non-destructively), then backfill from related entities.

---

## 4) Idempotency keys

Add idempotency fields where needed:

- `Purchase.payment_intent_id` (unique)
- `Entitlement.source_purchase_id` + composite uniqueness
- `Commission.purchase_id` unique
- `PayoutRecord.commission_id` unique



## **NEXT\_CHAT\_CONTEXT.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/NEXT\_CHAT\_CONTEXT.md

You are my Full-Stack Developer for the Breslov Academy LMS project.

NON-NEGOTIABLES 1) Do NOT remove features. No deletions/renames of existing pages/routes/entities/components. 2) Prefer production-grade solutions (security, scalability, maintainability, UX). 3) Work iteratively; each major step ends with: what changed • where (file paths) • how to run/test • what's next. 4) Maintain a running KNOWN\_ISSUES\_TECH\_DEBT.md and resolve after forward progress. 5) Vault always: every feature remains discoverable via Vault + the feature registry. 6) Multi-tenant isolation: tenant scoping is mandatory (activeSchoolId for private portals; schoolSlug→school\_id for public storefront). 7) Public browsing OK; materials protected: do not fetch/render lesson materials unless access state is UNLOCKED.

### **TARGET SURFACES (v9.0)**

- Public marketing site: /, /about, /how-it-works, /faq, /contact, /privacy, /terms
- Separate login pages: /login/student, /login/teacher, (optional) /login/admin
- Separate portals: /student/\*, /teacher/\*, /admin/\*, /superadmin/\*
- Public storefront per school slug: /s/:schoolSlug/\* (catalog/sales/checkout/thank-you)

**CRITICAL SECURITY RULE** In LOCKED or DRIP\_LOCKED states, lesson materials must NOT be fetched, rendered, or passed to downstream components.

**YOU WILL BE GIVEN:** a v9.0 zip. First do Phase 0 audit from ROADMAP\_V9\_DETAILED.md, then execute the roadmap in order.

## **OPS\_INTEGRITY\_QA.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/OPS\_INTEGRITY\_QA.md

# **Ops Hardening + Integrity + QA (v9.0)**

Prepared: January 01, 2026 (Asia/Jerusalem)

---

## **1) Ops hardening goals**

- No white screens; degrade gracefully with actionable errors.
  - No data leaks; tenant scoping must be consistent.
  - Idempotent money flows.
  - Auditability: admin actions are traceable.
- 

## **2) Integrity page contract**

Admin-only, vaultOnly. Must include:

- Registry checks:
- feature count
- route existence best-effort
- Security/leakage heuristics:
- “materials fetched while locked” detection (best-effort)
- downloads url exposure checks
- search metadata-only checks
- Tenancy heuristics:
- scan queries for missing school\_id filters (best-effort patterns)
- RBAC checks:
- Network admin requires global role
- Staff/invites/tools require admin

Export:

- JSON report with timestamp, school\_id, checks, warnings, environment metadata.

### 3) Regression net (smoke tests)

Manual smoke checklist (must pass before release):

- Guest:
  - browse /s/:slug catalog and sales pages
  - confirm no full lesson materials visible
  - Student locked:
  - open lesson viewer -> AccessGate only
  - confirm no materials fetch calls
  - Student drip locked:
  - countdown visible; no materials fetch
  - Student unlocked:
  - lesson renders; watermark/policy applied
  - Downloads:
  - unauthorized -> no URL; license CTA
  - authorized -> click "Get download"; url revealed; event logged
  - Staff:
  - invite staff; accept token; cannot reuse token; audit logs present
  - revoke staff; access removed
  - Money:
  - approve same transaction twice -> no duplicates
  - create payout batch twice -> no duplicates
- 

### 4) Performance budgets

- list queries: limit + stable sort
- cache keys include activeSchoolId
- heavy tables become cards on mobile

## **PORTALIZATION\_BUILD\_SCRIPT\_V9\_1.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/PORTALIZATION\_BUILD\_SCRIPT\_V9\_1.md

# **Portalization & Public Site — Step-by-Step Build Script (v9.1.x)**

Prepared: January 01, 2026 (Asia/Jerusalem)

This is the “do this, then this” plan that we will follow in the next coding chat.

---

## **PR v9.1.0 — Public Landing + Separate Logins**

### **Goal**

- Guest-friendly public site at /
- Dedicated login routes:
- /login/student
- /login/teacher
- No breakage to existing /dashboard etc.

### **Code tasks (ordered)**

1) Add new pages

- Create folder: src/pages/public/
- Files:
- Landing.jsx
- About.jsx
- HowItWorks.jsx
- FAQ.jsx
- Contact.jsx
- Privacy.jsx
- Terms.jsx

2) Add login pages

- Create folder: src/pages/auth/
- Files:

- `LoginStudent.jsx`
- `LoginTeacher.jsx`
- Each page:
- uses `base44.auth.redirectToLogin()` on CTA click
- writes `ba_intended_audience` in `localStorage`

### 3) Wire routes

- Edit: `src/App.jsx`
- Add explicit routes above the quizzes/storefront routes:
- `/about`, `/how-it-works`, etc.
- `/login/student`, `/login/teacher`
- Update root (`/`) behavior:
- If authenticated + active school → redirect to appropriate portal home
- Else show Landing

### 4) Integrate with feature registry (optional but recommended)

- Edit: `src/components/config/features.jsx`
- Add features for public pages (audience: public, hidden: true for legal pages)

### 5) Nav split

- Add a “public top nav” inside the public pages (not the main Layout)
- Keep existing Layout untouched for authenticated portals.

## Acceptance checks

- Guest can load / without errors
  - Guest can navigate to `/about`, `/faq`, etc.
  - Clicking “Student Login” redirects to Base44 auth
  - After login return:
  - If no school selected: go to `SchoolSelect`
  - Else: go to `/student/dashboard` (later PR) or existing `/dashboard`
- 

## PR v9.1.1 — Portal Route Groups (Student/Teacher/Admin)

### Goal

- URL namespaces for clarity and future expansion:
- `/student/*`
- `/teacher/*`
- `/admin/*`
- Role gates enforced for teacher/admin

## Code tasks (ordered)

1) Add portal router helpers:

- `src/components/routing/PortalGate.jsx`
- `src/components/routing/RequireRole.jsx` (small RBAC helper)

2) Extend router:

- Edit: `src/App.jsx`
- Add route groups:
- `/student/*` → `<PortalGate audience="student" />`
- `/teacher/*` → `<PortalGate audience="teacher" />`
- `/admin/*` → `<PortalGate audience="admin" />`

3) Inside `PortalGate`:

- Read session via `useSession()`
- Validate membership + role
- Set audience intent (`localStorage` + optional context setter)
- Render the existing Layout or redirect to canonical pages

4) Add initial portal home redirects:

- `/student` → `/student/dashboard`
- `/teacher` → `/teacher/teach` (or `/teacher/dashboard`)
- `/admin` → `/admin/schooladmin`

## Acceptance checks

- Student cannot load `/admin/*`
  - Teacher can load `/teacher/*` and still access `/teach/*`
  - Legacy routes keep working
-

## **PR v9.1.2 — Portal Switching + Audience Persistence**

### **Goal**

If a user is:

- teacher in one school, student in another, etc...

they can switch portals cleanly.

Tasks:

- Add portal switcher in `src/Layout.jsx` topbar when multiple roles
  - Persist last chosen portal per school:
  - `ba_portal_pref:{schoolId} = student|teacher|admin`
  - On login return, choose portal:
  - use stored preference, else infer from role, else default student
- 

## **PR v9.1.3 — Onboarding Routing Cleanup**

- Make “no memberships” user experience clean:
  - show wizard to create/join school
  - never dump them into empty dashboard
  - Ensure SchoolSelect is always reachable and stable.
- 

## **PR v9.1.4 — Polish & Accessibility**

- Keyboard nav for public site
- ARIA for nav + dialogs
- “reduced motion” check for animations
- Perf pass: lazy load heavy pages, defer charts

**PORTAL\_FEATURE\_MATRIX.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/PORTAL\_FEATURE\_MATRIX.md

**Portal Feature Matrix (Derived from Feature Registry)**

Prepared: January 01, 2026 (Asia/Jerusalem)

Source: src/components/config/features.jsx

Portal classification	Feature key	Label	Route	Area	Audiences	Hidden	VaultOnly
admin	SchoolAdmin	School Admin	/schooladmin	admin			
admin	SchoolAnalytics	School Analytics	/schoolanalytics	admin			
admin	SchoolMonetization	Monetization	/schoolmonetization	admin			
admin	AuditLogViewer	Audit Log	/auditlogviewer	admin	admin		
admin	SchoolNew	Create School	/schoolnew	admin	admin		
LegacyMigration	Legacy Migration	/legacymigration	admin	admin			
NetworkAdmin	Network Admin	/networkadmin	admin	admin			
AdminHardening	Security Hardening	/adminhardening	admin	admin			
SchoolStaff	Staff Management	/schoolstaff	admin	admin			
Integrity	Integrity Check	/integrity	system	admin			
Dashboard	Dashboard	/dashboard	core	student,teacher,admin			
admin+others	Courses	Courses	/courses	core	student,teacher,admin		
admin+others	Reader	Smart Reader	/reader	core	student,teacher,admin		
admin+others	Feed	Community	/feed	core	student,teacher,admin		
admin+others	SchoolSearch	Search	/schoolsearch	core			
student,teacher,admin	MyQuizzes	My Quizzes	/my-quizzes	core	student,teacher,admin		
core	Account	My Account	/account	core	student,teacher,admin		
core	Community	Community	/community	core	student,teacher,admin		
CourseDetail	Course Detail	/coursedetail	core	student,teacher,admin			
Downloads	Downloads	/downloads	core	student,teacher,admin			
admin+others	Forums	Forums	/forums	core	student,teacher,admin		
admin+others	LearningPaths	Learning Paths	/learningpaths	core			
student,teacher,admin	LessonViewer	Lesson Viewer	/lessonviewer	core	student,teacher,admin		
LessonViewerPremium	Lesson Viewer Premium	/lessonviewerpremium	core				
student,teacher,admin	Messages	Messages	/messages	core			
student,teacher,admin	Achievements	Achievements	/achievements	labs	student,teacher,admin		
AdaptiveLearning	Adaptive Learning	/adaptivelearning	labs	student,teacher,admin			
admin+others	Affiliate	Affiliate Program	/affiliate	labs			
student,teacher,admin	AlumniNetwork	Alumni Network	/alumninetwork	labs	student,teacher,admin		
Analytics	Analytics	/analytics	labs	student,teacher,admin			
CareerPaths	Career Paths	/careerpaths	labs	student,teacher,admin			



admin+others | Challenges | Challenges | /challenges | labs | student,teacher,admin  
 | | | admin+others | CohortDetail | Cohort Detail | /cohortdetail | labs |  
 student,teacher,admin | | | admin+others | Cohorts | Cohorts | /cohorts | labs |  
 student,teacher,admin | | | admin+others | Events | Events | /events | labs |  
 student,teacher,admin | | | admin+others | LanguageLearning | Languages |  
 /languagelearning | labs | student,teacher,admin | | | admin+others | Languages |  
 Languages (Legacy) | /languages | labs | student,teacher,admin | | | admin+others  
 | Leaderboard | Leaderboard | /leaderboard | labs | student,teacher,admin | | |  
 admin+others | LiveStreams | Live Streams | /livestreams | labs |  
 student,teacher,admin | | | admin+others | Marketplace | Marketplace |  
 /marketplace | labs | student,teacher,admin | | | admin+others | Mentorship |  
 Mentorship | /mentorship | labs | student,teacher,admin | | | admin+others |  
 StudySetNew | New Study Set | /studyssetnew | labs | student,teacher,admin | | |  
 admin+others | Offline | Offline | /offline | labs | student,teacher,admin | | |  
 admin+others | StudySetPractice | Practice | /studyssetpractice | labs |  
 student,teacher,admin | | | admin+others | Skills | Skills | /skills | labs |  
 student,teacher,admin | | | admin+others | StudySet | Study Set | /studysset | labs  
 | student,teacher,admin | | | admin+others | StudySets | Study Sets | /studyssets |  
 labs | student,teacher,admin | | | admin+others | AffiliateProgram | Affiliate  
 Program Info | /affiliateprogram | marketing | public,student,teacher,admin | | |  
 admin+others | SchoolCheckout | Checkout | /schoolcheckout | marketing |  
 public,student,teacher,admin | | | admin+others | SchoolCourses | Course Catalog  
 | /schoolcourses | marketing | public,student,teacher,admin | | | admin+others |  
 SchoolLanding | Landing Page | /schoollanding | marketing |  
 public,student,teacher,admin | | | admin+others | SchoolPricing | Pricing |  
 /schoolpricing | marketing | public,student,teacher,admin | | | admin+others |  
 CourseSales | Sales Page | /coursesales | marketing | public,student,teacher,admin  
 | | | admin+others | SchoolThankYou | Thank You | /schoolthankyou | marketing |  
 public,student,teacher,admin | | | admin+others | Integrations | Integrations |  
 /integrations | system | student,teacher,admin | | | admin+others | SchoolJoin |  
 Join School | /schooljoin | system | student,teacher,admin | | | admin+others |  
 Portfolio | Profile | /portfolio | system | student,teacher,admin | | | admin+others |  
 SchoolSelect | School Select | /schoolselect | system | student,teacher,admin | | |  
 admin+others | Subscription | Subscription | /subscription | system |  
 student,teacher,admin | | | admin+others | Vault | Vault | /vault | system |  
 student,teacher,admin | | | admin+others | Teach | Teach | /teach | teach |  
 teacher,admin | | | admin+others | TeachAnalytics | Teaching Analytics |  
 /teachanalytics | teach | teacher,admin | | | admin+others | TeachQuizzes | Quizzes |  
 /teach/quizzes | teach | teacher,admin | | | admin+others | TeachCourse | Course  
 Builder | /teachcourse | teach | teacher,admin | | | admin+others | TeachLesson |  
 Lesson Editor | /teachlesson | teach | teacher,admin | | | admin+others |  
 TeachCourseNew | New Course | /teachcoursenew | teach | teacher,admin | | | public  
 | CertificateVerify | Certificate Verification | /certificateverify | marketing |  
 public | | | public | InviteAccept | Accept Invite | /inviteaccept | system | public |  
 | | | public | oauth2callback | OAuth Callback | /oauth2callback | system | public | |  
 | | | student | MyProgress | My Progress | /myprogress | core | student | | | student |  
 HabitTracker | Habit Tracker | /habittracker | labs | student | | | student |  
 Microlearning | Microlearning | /microlearning | labs | student | | | student |

```
RewardsShop | Rewards Shop | /rewardsshop | labs | student | | | student |  
Scholarships | Scholarships | /scholarships | labs | student | | | student |  
StudyBuddies | Study Buddies | /studybuddies | labs | student | | | student |  
Tournaments | Tournaments | /tournaments | labs | student | | |
```

## **PR\_CHECKLIST.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/PR\_CHECKLIST.md

### **PR Checklist (Must Pass) — v9+**

Prepared: January 01, 2026 (Asia/Jerusalem)

- ☐ No features removed; legacy routes still work via aliases.
- ☐ Vault + Feature Registry lists everything accessible.
- ☐ Tenant scoping everywhere (activeSchoolId portals; slug→school\_id storefront).
- ☐ RBAC enforced server/app-side; wrong portal blocked.
- ☐ LOCKED/DRIP\_LOCKED never fetch/render/mount materials.
- ☐ Download URLs revealed only after entitlement verification + logged.
- ☐ Search is metadata-only for unauthorized.
- ☐ Checkout idempotent (no duplicate entitlements/commissions/payouts).
- ☐ QA checklist updated for impacted flows.
- ☐ CHANGELOG/RECOVERY updated if behavior changed.

## **QA\_RELEASE\_CHECKLISTS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/QA\_RELEASE\_CHECKLISTS.md

# **QA & Release Checklists — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

Use this as the “ship gate.” If any check fails, v9.0 is not stable.

---

## **1) Pre-flight**

- Clean install works (node\_modules removed -> reinstall).
  - Build succeeds in production mode.
  - No console errors on initial load.
- 

## **2) Public Marketing**

- / loads quickly and renders hero + CTA
  - All marketing pages render on mobile
  - Links to login + signup are correct
- 

## **3) Auth + Portal Routing**

- Student login -> student portal
  - Teacher login -> teacher portal
  - Admin login (if exists) -> admin portal
  - Wrong portal role -> friendly “switch portal” UI
- 

## **4) Tenant isolation**

- Switching active school changes all scoped pages
  - No data from another tenant appears in lists
  - Storefront slug resolves to correct school
- 

## **5) Content protection (most important)**

- Guest: no full materials visible anywhere
  - LOCKED: lesson viewer shows AccessGate only; no materials fetched
  - DRIP\_LOCKED: countdown only; no materials fetched
  - UNLOCKED: materials render; watermark/policy applied
- 

## **6) Downloads**

- Unauthorized: no URL shown
  - Authorized: “Get download” reveals URL and logs audit event
  - Attempt to brute-force a file id is blocked or yields no URL
- 

## **7) Storefront + Checkout**

- Coupon works and is tracked
  - Purchase grants entitlements (idempotent)
  - Thank-you page shows purchase summary
  - Affiliate referral attribution works (if enabled)
- 

## **8) Ops/Admin tools**

- Invites: create -> accept -> cannot reuse -> revoke
- Audit logs: events recorded for admin actions
- Integrity page: report export works

## **9) Performance + UX**

- Tables collapse to cards on mobile
  - Skeleton loaders used for slow lists
  - Reduced motion respected
- 

## **10) Release outputs**

- CHANGELOG updated with v9.0 entry
- RECOVERY updated with:
  - where registry lives
  - how to confirm all features are present
  - how to run integrity report
- A sample exported integrity report stored in repo (if allowed)

## **RBAC\_TENANCY\_MODEL.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/RBAC\_TENANCY\_MODEL.md

# **RBAC + Tenant Scoping Model — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

---

## **1) Roles**

Baseline roles:

- Guest (public only)
- Student
- Teacher
- School Admin (tenant admin)
- Org Owner (tenant owner; billing + high privilege)
- Platform Admin / Super Admin (global)

Notes:

- A user may have multiple tenant memberships (multi-school).
  - A user may be a student in one tenant and teacher/admin in another.
- 

## **2) Authorization principles**

- Deny by default; allow only via explicit role.
  - UI may hide actions; server/app-layer guards are authoritative.
  - All data operations must be tenant-scoped.
- 

## **3) Tenant scoping rules**

### **3.1 For authenticated portal requests**

- Determine activeSchoolId from session.

- All entity queries must include `school_id == activeSchoolId` (or equivalent tenant field).

### 3.2 For public storefront requests

- Resolve `schoolSlug` -> `school_id` once at layout/root.
- Use that `school_id` to scope all storefront queries.

### 3.3 Prohibited patterns

- Never query tenant-scoped entities without a tenant filter.
  - Never accept client-supplied `tenant_id` as “truth.”
- 

## 4) Permissions matrix (minimum viable)

Legend: R=read, W=write, A=approve/admin, X=none

Capability	Guest	Student	Teacher	School Admin	Owner	SuperAdmin
Browse marketing/storefront	R	R	R	R	R	R
View course sales pages	R	R	R	R	R	R
Enroll/purchase	W	W	W	W	W	W
View lesson materials (unlocked)	X	R	R	R	R	R
Create/edit courses/lessons	X	X	W	W	W	W
Grade assignments	X	X	W	A	A	A
View analytics	X	R (own)	R/W	A	A	A
Manage users/invites	X	X	X	A	A	A
Configure branding/terminology	X	X	X	A	A	A
Configure protection policy	X	X	X	A	A	A
Monetization config	X	X	X	A	A	A
Payouts batching	X	X	X	A	A	A
Cross-tenant access	X	X	X	X	X	X
Impersonation	X	X	X	X	X	A

---

## 5) Route guards

- `/student/*` requires authenticated + membership role includes Student (or higher) in `activeSchool`.
  - `/teacher/*` requires Teacher/Admin/Owner.
  - `/admin/*` requires School Admin/Owner.
  - `/superadmin/*` requires SuperAdmin.
  - `/integrity`, `/migration-tools`, `/audit-log-viewer` are admin-only and ideally `vaultOnly`.
-



## **6) Audit logging requirements**

Log these events with actor + tenant + timestamp + metadata:

- Role changes, membership changes
- Invite created/accepted/revoked
- Tenant settings changes
- Protection policy changes
- Offer/coupon changes
- Transaction approvals/refunds
- Download URL reveal events
- Impersonation start/stop

## README.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/README.md

# Breslov Academy LMS — v9.0 Spec Pack (Text Docs)

Prepared: January 01, 2026 (Asia/Jerusalem)

This folder is the single source of truth to continue coding in a fresh chat using the v9.0 zip.

## Non-negotiables

- No features removed. Everything that exists stays; reorganize + upgrade only.
- No destructive refactors. Keep legacy routes working via aliases/adapters.
- Vault always. Every feature must remain discoverable via Vault + Feature Registry.
- Tenant scoping everywhere. All school-owned data must be scoped by `activeSchoolId` (private portals) or `schoolSlug` -> `school_id` (public storefront).
- Content safety: In `LOCKED` or `DRIP_LOCKED` states, lesson materials must not fetch, render, mount, or be passed into downstream components (including AI tutor).

## What you do first when you upload the v9.0 zip

- 1 Unzip locally and run the project once to confirm it builds.
- 2 Run the Phase 0 Audit in `ROADMAP_V9_DETAILED.md` to discover:
  - router entry file(s)
  - layout/nav shell
  - Vault page
  - session/auth provider
  - tenancy helper(s)
  - where entities live (`entities/*.json` in Base44-style projects)
- 3 Populate/verify the Feature Registry from real routes (never guess).

## Files in this pack

- `ROADMAP_V9_DETAILED.md` — extremely detailed, step-by-step roadmap and backlog.
- `ARCHITECTURE_NORTH_STAR.md` — target architecture (session, tenancy, gating, registry).
- `IA_ROUTES_PORTALS.md` — complete route map (public + portals + storefront + legacy).

- RBAC\_TENANCY\_MODEL.md — roles/permissions matrix + tenant-scoping rules.
- DESIGN\_SYSTEM\_UI\_UX.md — tokens, components, patterns (WOW + calm + academic).
- FEATURE\_REGISTRY\_AND\_VAULT.md — registry schema + nav + Vault + command palette.
- CONTENT\_PROTECTION\_MATERIALS\_ENGINE.md — gating states + materials engine contracts.
- STORE\_MONEY\_SUBSCRIPTIONS.md — storefront, offers, coupons, affiliates, payouts.
- OPS\_INTEGRITY\_QA.md — integrity contract, diagnostics, regression net, release checks.
- MIGRATIONS\_DATA\_MODEL.md — conceptual data model + “extend-not-replace” migrations.
- NEXT\_CHAT\_CONTEXT.md — copy/paste into a new ChatGPT coding chat.

## Output discipline during coding

Every major step ends with:

- What changed
  - Where it changed (file paths)
  - How to run/test
  - What’s next
- 

## Repo reality integration (required before heavy coding)

Because GitHub repos can be private, Phase 0 Audit must use REPO\_REALITY\_AUDIT.md to extract:

- actual router file paths
  - actual entity storage (Base44 JSONs etc.)
  - actual auth/session provider
  - actual storefront + monetization modules
- 

## GitHub repository (Breslov Academy)

Remote:

- HTTPS: <https://github.com/yeshivabachur/breslov-academy.git>

- SSH: `git@github.com:yeshivabachur/breslov-academy.git`

Clone:

- Git (HTTPS): `git clone https://github.com/yeshivabachur/breslov-academy.git`
- Git (SSH): `git clone git@github.com:yeshivabachur/breslov-academy.git`
- GitHub CLI: `gh repo clone yeshivabachur/breslov-academy`

## **RELEASE\_TRAINS\_FUTURE\_BUILDS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/RELEASE\_TRAINS\_FUTURE\_BUILDS.md

# **Release Trains & Future Builds (Continuous Improvement Plan)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **Build channels**

- Stable: tagged releases only (must pass QA + integrity export)
- Beta (optional): release candidates for risky UI/onboarding changes
- Nightly (optional): build artifact on every merge to main

## **Versioning (practical)**

- v9.0.0 milestone stable
- v9.0.x patches
- v9.1.0 feature release
- v10.0.0 major planned change (rare)

## **Definition of a “build”**

A build includes:

- code
- CHANGELOG entry
- RECOVERY updates (if needed)
- integrity report sample (stable)
- known issues list updated

## **Future build epics (suggested)**

- v9.1: Onboarding & tenant excellence
- v9.2: Student learning WOW
- v9.3: Teacher power tools
- v9.4: Storefront conversion & trust
- v9.5: Observability + QA + perf budgets



## **RELEASE\_TRAIN\_CONTINUOUS\_BUILDS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/RELEASE\_TRAIN\_CONTINUOUS\_BUILDS.md

# **Release Train — Continuous Builds (v9.0 → v10)**

Prepared: January 01, 2026 (Asia/Jerusalem)

This is the *long-term* plan for “continuous improvement builds” that never stop. It is written to work with your workflow: you give me a ZIP; I produce PR-ready code in the next chat.

---

## **Repo & cloning**

- HTTPS: <https://github.com/yeshivabachur/breslov-academy.git>
- SSH: `git@github.com:yeshivabachur/breslov-academy.git`
- GH CLI: `gh repo clone yeshivabachur/breslov-academy`

Branching baseline:

- `main` = always deployable (only stable releases)
- `dev` = optional (if you prefer a single integration branch)
- feature branches per PR (recommended)

Release artifact:

- Build ZIP produced by `scripts/release.ps1`
  - Must include: `BUILDINFO.json`, `CHECKSUMS.txt`
- 

## **Versioning policy (SemVer with “train” meaning)**

- MAJOR (v10, v11...): architecture milestones (portalization complete, new tenancy model, etc.)
- MINOR (v9.1, v9.2...): shippable feature sets
- PATCH (v9.1.1, v9.1.2...): stability, polish, security, perf

## **Labels**

- Stable: passes healthchecks, no critical issues, safe to deploy.
  - Preview: large refactors; behind feature flags; no production deploy.
  - Hotfix: small targeted patch off latest stable.
- 

## **Release cadence (recommended)**

- Weekly patch releases during active build:  $v9.x.(n)$  — stability + polish
  - Monthly minor releases:  $v9.(x+1).0$  — bigger features
  - 1-2 major releases per year max
- 

## **Train Map (high-level)**

### **Train A — Portalization & Public Website (v9.1.x)**

Goal: landing site + separate logins + portal routes without breaking legacy routes.

Planned patches:

- v9.1.0: public landing + `/login/student` `/login/teacher`
- v9.1.1: portal route groups `/student/*` `/teacher/*` `/admin/*`
- v9.1.2: audience persistence + portal switching UI
- v9.1.3: onboarding routing cleanup (no membership cases)
- v9.1.4: accessibility + performance polish

### **Train B — Multi-tenant Onboarding Excellence (v9.2.x)**

Goal: school/entity signup workflow + approvals + invite hardening.

### **Train C — Student “WOW Learning” (v9.3.x)**

Goal: calm premium lesson flow, fast, safe, delightful.

### **Train D — Teacher Tools (v9.4.x)**

Goal: course builder, grading, analytics.

### **Train E — Storefront + Checkout Reliability (v9.5.x)**



Goal: conversion + trust + idempotency.

## **Train F — Observability + QA Automation (v9.6.x)**

Goal: CI, tests, performance budgets.

## **Train G — Major Platform Upgrade (v10.0)**

Goal: everything above + feature flags + robust admin.

---

## **PR template (use every time)**

Each PR must include: 1) What changed 2) Where it changed (file paths) 3) How to run/test 4) What's next

And must not violate:

- tenancy scoping invariants
  - content protection invariants
  - “no features removed” rule
- 

## **Definition of “Done” per build**

A build is “done” only when:

- All healthchecks pass
  - No regressions to existing features
  - Portal nav is consistent
  - Access gating remains airtight
  - ZIP release contains checksums + BUILDINFO
- 

## **“ZIP → PR” workflow (your process)**

1) You send me the new ZIP (or commit hash). 2) I extract, audit changes, and update:

- `KNOWN_ISSUES_TECH_DEBT.md`
- roadmap status

3) I produce a patch plan for the next PR (in the chat) and then code it.

## REPO\_REALITY\_AUDIT.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/REPO\_REALITY\_AUDIT.md

# Repo Reality Audit (Run Locally) — Breslov Academy

Prepared: January 01, 2026 (Asia/Jerusalem)

You asked me to “read the GitHub repo and integrate details.” Important limitation: I can’t reliably fetch private GitHub repo contents from here. So this file defines a zero-guess protocol to extract the repo “truth” from your machine, then we incorporate it into the roadmap before writing pages of code.

Repo:

- HTTPS: `https://github.com/yeshivabachur/breslov-academy.git`
- SSH: `git@github.com:yeshivabachur/breslov-academy.git`
- GH CLI: `gh repo clone yeshivabachur/breslov-academy`

---

## 0) One-time: clone + open

Code (bash):

```
gh repo clone yeshivabachur/breslov-academy
cd breslov-academy
```

---

## 1) Generate a “Repo Snapshot” text file

Run this and paste the output into chat (or commit it under docs/repo-snapshot.md).

### 1.1 Identity

Code (bash):

```
echo "## REPO ID"
git remote -v
git rev-parse HEAD
git status -sb
git log -1 --oneline
```

### 1.2 Tech stack + scripts

Code (bash):

```
echo "## PACKAGE.JSON"
cat package.json
```

### 1.3 High-level tree (first 6 levels)

Mac/Linux:

Code (bash):

```
echo "## TREE"
find . -maxdepth 6 -type d \( -name node_modules -o -name .git -o -name dist -o -name build \) -prune
```

Windows PowerShell:

Code (powershell):

```
"## TREE"
Get-ChildItem -Recurse -Depth 6 -File | Where-Object { $_.FullName -notmatch '\\node_modules\\|\\.git\\
```

### 1.4 Router + pages discovery (search terms)

Mac/Linux (recommended: ripgrep rg):

Code (bash):

```
echo "## ROUTER HITS"
rg -n "react-router|createBrowserRouter|Routes\\b|Route\\b|next/router|app\\/" -S . | sed -n '1,200p'
echo "## VAULT/REGISTRY HITS"
rg -n "Vault|featureRegistry|registry|Command.*Palette" -S . | sed -n '1,200p'
echo "## TENANT HITS"
rg -n "tenant|schoolSlug|school_id|activeSchool" -S . | sed -n '1,200p'
echo "## GATING HITS"
rg -n "LOCKED|DRIP_LOCKED|UNLOCKED|materials|downloadUrl" -S . | sed -n '1,200p'
```

---

## 2) Integrating the snapshot into this Spec Pack

Once you paste the snapshot:

- We will replace “assumptions” in ROADMAP\_V9\_DETAILED.md with exact file paths.
- We will create a “Legacy Route Map” from the real router list.
- We will adjust the plan to your actual stack (Vite vs Next, Base44 entities, etc.).

## **ROADMAP\_V9\_DETAILED.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/ROADMAP\_V9\_DETAILED.md

# **ROADMAP v9.0 — Extremely Detailed Execution Plan**

Prepared: January 01, 2026 (Asia/Jerusalem)

This roadmap is written to be executed after you upload the v9.0 zip. It assumes the project already includes v7 security + v8 monetization foundations, and v9 adds:

- Public marketing site
  - Split portals/logins
  - Tenant signup/onboarding
  - Premium design system shell + module conversion
- 

## **Phase 0 — Zip Intake, Audit, and “Truth Map” (NO DELETIONS)**

### **Objective**

Build a factual map of the codebase (routes, layouts, entities, hooks) so we can refactor safely.

### **Tasks**

#### **0.1 Project bootstrap**

- Unzip v9.0 locally
- Install deps
- Run dev server
- Confirm build passes (or capture exact error list)

#### **0.2 Identify canonical files (write them down)**

Find:

- Router entry file(s)
- Layout/nav shell file(s)
- Vault page file

- Session/auth provider/hook (e.g., useSession)
- Tenant scope helpers (e.g., useStorefrontContext, activeSchoolId)
- Lesson access hook (e.g., useLessonAccess)
- Entities folder (Base44 often uses entities/\*.json)

### **0.3 Create/verify append-only docs (create-if-missing)**

- CHANGELOG.md (append-only)
- RECOVERY.md
- ARCHITECTURE.md
- KNOWN\_ISSUES\_TECH\_DEBT.md

### **0.4 Build the “Truth Map” tables (in docs)**

In ARCHITECTURE.md, add:

- Where is Feature Registry?
- Where is Vault?
- Where is content protection policy?
- Where is monetization configuration?
- Where are tenant settings (terminology, branding)?
- Where is integrity page?

### **0.5 Risk scan**

- Search for any queries that fetch lessons/materials without gating
- Search for any entities queried without tenant scoping
- Locate any old/legacy routes that must remain

### **Acceptance criteria**

- App builds or we have a logged exact error list
  - We have a written map of the files above
  - Docs exist and are append-only
-

## **Phase 1 — Aliases + Import Hardening (Fix root causes)**

### **Objective**

Prevent “auto-fix” regressions by ensuring @/ imports resolve reliably.

### **Tasks**

- Confirm Vite alias @ -> ./src
- Confirm jsconfig/tsconfig paths @/\* -> src/\*
- Replace incorrect relative imports with @/ . . . consistently

### **Acceptance criteria**

- No “module not found” errors for central imports
  - Feature registry is importable from one stable module
- 

## **Phase 2 — Central Feature Registry + Vault Enforcement**

### **Objective**

Re-establish one source of truth for navigation + Vault + integrity.

### **Tasks**

#### **2.1 Registry module**

- Create/restore registry module file
- Populate registry from actual router map
- For each feature:
  - define canonical route
  - define aliases (legacy)
- assign area, audiences, vaultOnly, showInMainNav

#### **2.2 Vault**

- Remove any inline registry list
- Vault renders from registry
- Add search/filter UI
- Add badges (area, audience)

### **2.3 Layout nav**

- Build nav groups:
- Core (always)
- Teach (teacher/admin)
- Admin (admin/owner)
- Labs (optional)
- Vault (always)

### **2.4 Integrity**

- Ensure integrity page exists and reads registry
- Add export JSON report

### **Acceptance criteria**

- Vault shows every feature (permission-filtered)
  - No inline feature lists remain
  - Legacy routes work via aliases/redirects
  - Integrity report exports without crashing
- 

## **Phase 3 — v9 Surface Split: Public Site + Separate Logins + Portal Layouts**

### **Objective**

Implement new product surfaces while keeping old routes working.

### **Tasks**

#### **3.1 Public marketing site**



- Implement required marketing routes with premium layout
- Add top nav links to the correct login/signup pages

### **3.2 Separate login pages**

- Implement `/login/student`, `/login/teacher` (+ optional admin)
- Maintain existing `/login` as legacy redirect chooser
- Ensure post-login routing sends user to correct portal based on role + chosen audience

### **3.3 Portal shells**

Create distinct shells:

- StudentShell
- TeacherShell
- AdminShell
- SuperAdminShell

### **3.4 Route guards**

- Add guard wrappers per portal
- Ensure tenant is selected/derived before rendering portal pages

### **3.5 Legacy compatibility**

- Keep old portal routes accessible (aliases), progressively migrate canonical to new route groups

## **Acceptance criteria**

- Login entry points land in correct portal
  - Guards block unauthorized access
  - Old deep links still function
- 

## **Phase 4 — Signup + Onboarding (Individuals + Schools/Entities)**

## Objective

Enable controlled onboarding with approvals + invites while preserving security.

## Tasks

### 4.1 Individual signup

- /signup chooser, /signup/student, /signup/teacher
- Student join rules: invite code OR domain match OR approval queue
- Teacher join rules: invite-only OR approval

### 4.2 School/entity signup

- /signup/school
- Create pending application + tenant record
- Create owner/admin account or link existing
- Onboarding wizard in /admin/onboarding

### 4.3 Invitations lifecycle

- Create invitations with expiry + role + tenant
- Accept flow:
- token -> attach membership -> mark accepted -> invalidate token
- Revoke flow:
- remove membership -> audit log

## Acceptance criteria

- Tenant signup submission works end-to-end (pending -> approved -> onboard)
  - Invites work and cannot be reused
  - Approvals queue visible to admins
- 

## Phase 5 — Design System Shell + UI Revamp (No feature breakage)

## Objective

Deliver the new premium UI without breaking functionality.

## Tasks

- Implement tokens + light/dark/night themes
- Build/standardize base components
- Build LMS components (course cards, access gates, progress)
- Replace global shell first, then convert modules in safe order

## Acceptance criteria

- UI consistent across portals
  - No regressions in gating/monetization
  - Mobile-first behavior for heavy tables
- 

# Phase 6 — Security + Correctness Hardening (v9 stability)

## Objective

Lock down leakage vectors and correctness in money flows.

## Tasks

- Confirm lesson materials never fetched while locked/drip-locked
- Confirm downloads never expose URLs unauthorized
- Confirm search is metadata-only
- Rate limit auth + checkout + downloads
- Expand audit logs coverage

## Acceptance criteria

- Integrity checks pass or show actionable warnings
  - Manual smoke tests pass
-

## **Phase 7 — Storefront + Monetization polish (if needed)**

### **Objective**

Ensure storefront is premium and operationally correct.

### **Tasks**

- School branding polish
  - Offers/bundles/subscriptions UX
  - Coupon UX improvements
  - Affiliate dashboard + payouts polish
  - Account portal improvements
- 

## **Phase 8 — QA, Release Packaging, and “v9.0 Stable” Cut**

### **Objective**

Ship a stable build and recovery playbook.

### **Tasks**

- Final regression checklist
- Update CHANGELOG with v9.0 entry
- Update RECOVERY with “how to restore”
- Export integrity report JSON sample
- Package stable zip

## **Phase 0 Add-on — Repo Reality Snapshot (mandatory)**

- Run the protocol in `REPO_REALITY_AUDIT.md`.
- Paste the snapshot into the coding chat (or commit under `docs/repo-snapshot.md`).

- Replace any assumption placeholders in docs with exact file paths and routes.

## **SECURITY\_THREAT\_MODEL.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/SECURITY\_THREAT\_MODEL.md

# **Security Threat Model (Practical) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This is a pragmatic checklist for the app-layer security model.

---

## **1) Primary threats**

- Unauthorized lesson material viewing
  - Download URL leakage
  - Cross-tenant data leakage
  - Privilege escalation (role spoofing)
  - Replay/duplication in checkout flows
  - Invite token reuse
  - Bot abuse (signup/login/checkout)
- 

## **2) Controls**

### **2.1 Tenant scoping**

- Every query includes school\_id filter.
- Storefront resolves slug -> school\_id at root and reuses it.

### **2.2 RBAC**

- Route guards + data guards.
- Superadmin tools are locked down and audited.

### **2.3 Content gating**

- LOCKED/DRIP\_LOCKED never fetch materials.
- Preview rendering uses separate “safe preview” payload.

## **2.4 Download reveal**

- A “reveal” action fetches URL only after entitlement verification.
- Log reveal event.
- Prefer ephemeral URLs if supported; otherwise best-effort patterns.

## **2.5 Idempotency**

- Payment intent id unique
- Entitlement grants dedupe
- Commission + payout dedupe

## **2.6 Abuse protections**

- Rate limit login/signup/reveal/download/checkout calls.
  - Basic bot mitigation on public forms (where feasible).
- 

## **3) Security regression tests**

- Manually test locked/drip locked states
- Verify search never shows full content
- Try cross-tenant switching and ensure no data bleed

## **STORE\_MONEY\_SUBSCRIPTIONS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/STORE\_MONEY\_SUBSCRIPTIONS.md

# **Storefront + Monetization (Teachable/Kajabi mode) — v9.0**

Prepared: January 01, 2026 (Asia/Jerusalem)

This describes the storefront experience while keeping multi-tenant security intact.

---

## **1) Storefront surfaces (per school slug)**

- School landing: `/s/:schoolSlug`
- Catalog: `/s/:schoolSlug/catalog`
- Sales page: `/s/:schoolSlug/course/:courseId`
- Pricing/offers: `/s/:schoolSlug/pricing`
- Checkout: `/s/:schoolSlug/checkout`
- Thank you: `/s/:schoolSlug/thank-you`

All storefront data must be scoped using `schoolSlug` -> `school_id`.

---

## **2) Offer model**

Offer types:

- Single course
- Bundle
- Subscription plan/tier
- Add-ons:
- Copy license
- Download license

Offer fields:

- price, currency, active
- included content references



- limits (seat count, expiry)
  - upsells/order bumps
  - coupon eligibility rules
- 

### **3) Checkout flow (idempotent)**

1) Guest enters email (or logged-in uses account) 2) Cart/offer summary 3) Coupon application 4) Payment 5) Create purchase record 6) Grant entitlements (idempotent) 7) Track analytics event 8) Redirect to thank-you

Idempotency safeguards:

- payment intent id / transaction id unique constraint
  - entitlement grants dedupe
  - affiliate commission dedupe
  - payout batch dedupe
- 

### **4) Entitlements as source of truth**

Rules:

- Any access checks consult entitlements.
  - Expired entitlements are ignored everywhere.
  - Entitlements must be tenant-scoped.
- 

### **5) Affiliates**

Per-tenant affiliates:

- referral code or link
- click tracking
- conversion tracking
- earnings ledger
- payout batches + export + mark-paid workflow

## **6) Analytics instrumentation**

Track:

- page views on storefront
  - checkout start
  - coupon applied
  - purchase completed
  - upsell accepted/rejected
  - churn/renewal events for subscriptions (best-effort)
- 

## **7) Admin dashboards**

School admin:

- offers/coupons/transactions approvals
- payouts batching
- affiliate performance
- conversion funnel

## **V9\_AUTH\_ONBOARDING\_REALITY.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_AUTH\_ONBOARDING\_REALITY.md

# **Auth, Audience, and Onboarding (Reality + Plan)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **Current authentication model (v9.0)**

- Auth provider: Base44 SDK
- Auth state lives in:
  - `src/lib/AuthContext.jsx`
- Login redirect:
  - `base44.auth.redirectToLogin(window.location.href)`
- Logout:
  - `base44.auth.logout(...)`

Important nuance:

- Base44 login is not implemented as an in-app route page in this ZIP.
- The app can be configured to require auth; storefront routes attempt to remain guest-safe.

## **Current session model (v9.0)**

- `src/components/hooks/useSession.jsx` is the “single source of truth” for:
  - user
  - memberships
  - activeSchoolId
  - role
  - audience (student/teacher/admin)

Audience is derived from role via:

- `normalizeAudienceFromRole()` in `src/components/config/features.jsx`

## **New requirement: separate login pages**

We will implement in-app routes that still use Base44 auth, but provide:

- distinct URLs
- distinct messaging and “what happens next”
- persistent audience intent

### **Proposed new routes (add explicit routes in `src/App.jsx`)**

- `/login/student`
- `/login/teacher`
- `/login/admin` (optional)
- `/signup` (router)
- `/signup/student`
- `/signup/teacher`
- `/signup/school`

### **How these routes work (no guessing)**

1) Render a lightweight page explaining the path (student vs teacher). 2) Store “intended audience” in `localStorage`:

- `ba_intended_audience = student|teacher|admin`

3) Call `base44.auth.redirectToLogin()` when user clicks continue. 4) After return, `useSession` loads user + memberships:

- If user has no memberships → route to onboarding wizard
- If multiple schools → route to `SchoolSelect`
- Else → route to portal home

### **Onboarding (already exists in parts)**

Existing pages you can reuse/extend:

- `SchoolSelect` (`src/pages/SchoolSelect.jsx`)
- `SchoolJoin` (`src/pages/SchoolJoin.jsx`)
- `SchoolNew` (`src/pages/SchoolNew.jsx`)
- Invite acceptance:
- `InviteAccept` (`src/pages/InviteAccept.jsx`)

### **New requirement: school/entity signup**

We will extend `SchoolNew` into a true “application” workflow:

- `SchoolApplication` entity (new) OR reuse an existing application-style entity if present in backend.
- status lifecycle: `pending` → `approved` → `active` (or `rejected`)
- approval UI: place in Admin portal (see roadmap)

## **V9\_CODEBASE\_REALITY\_MAP.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_CODEBASE\_REALITY\_MAP.md

# **V9.0 Stable — Codebase Reality Map**

Prepared: January 01, 2026 (Asia/Jerusalem)

This document is generated from the uploaded ZIP: breslov-academy\_v9.0-stable.zip.

## **Repo identity**

- App: base44-app 9.0.0
- BUILDINFO: v9.0-stable created 2026-01-01T00:50:23Z
- Source zip in BUILDINFO: breslov-academy-main00\_v89.zip

GitHub:

- HTTPS: <https://github.com/yeshivabachur/breslov-academy.git>
  - SSH: <git@github.com:yeshivabachur/breslov-academy.git>
  - GH CLI: `gh repo clone yeshivabachur/breslov-academy`
- 

## **Stack (confirmed)**

- Vite + React (vite.config.js, src/main.jsx)
- React Router DOM (src/App.jsx)
- Tailwind + shadcn/ui + Radix
- TanStack Query (@tanstack/react-query)
- Base44 SDK (@base44/sdk) with runtime tenancy enforcement

Scripts:

- `npm run dev, npm run build, npm run lint, npm run typecheck, npm run preview`
  - Release packaging: `scripts/release.ps1`
  - Update repo from ZIP: `scripts/update-github-from-zip.ps1`
-

## Key files (where the “truth” lives)

- App entry: `src/main.jsx`
  - Router: `src/App.jsx`
  - Global layout/nav: `src/Layout.jsx`
  - Pages registry: `src/pages.config.js`
  - Feature registry (nav + vault): `src/components/config/features.jsx`
  - Route helper: `src/utils/index.ts`
  - Auth context: `src/lib/AuthContext.jsx`
  - Session context (audience/role/tenant): `src/components/hooks/useSession.jsx`
  - Tenancy bridge: `src/components/api/TenancyBridge.jsx`
  - Scoped CRUD helpers: `src/components/api/scoped.jsx`
  - Scoped entities list: `src/components/api/scopedEntities.js`
  - Tenancy runtime enforcer install: `src/api/base44Client.js`
  - Lesson access hook: `src/components/hooks/useLessonAccess.jsx`
  - Materials engine: `src/components/materials/materialsEngine.jsx`
  - Access gate UI: `src/components/security/AccessGate.jsx`
  - Vault: `src/pages/Vault.jsx`
  - Integrity: `src/pages/Integrity.jsx`
  - Storefront landing: `src/pages/SchoolLanding.jsx`
  - Release script: `scripts/release.ps1`
  - Update GitHub from ZIP: `scripts/update-github-from-zip.ps1`
- 

## High-level source tree (first 120 entries, depth≤3)

- `App.jsx`
- `Layout.jsx`
- `api/`
- `api/base44Client.js`
- `components/`
- `components/UserNotRegisteredError.jsx`
- `components/academic/`
- `components/academic/quizEngine.jsx`

- components/accessibility/
- components/accessibility/AccessibilityMenu.jsx
- components/admin/
- components/admin/ContentProtectionSettings.jsx
- components/ai/
- components/ai/AITutor.jsx
- components/ai/AcademicAdvisorChat.jsx
- components/ai/AiTutorPanel.jsx
- components/ai/CourseRecommendations.jsx
- components/analytics/
- components/analytics/ConversionFunnel.jsx
- components/analytics/EngagementPredictor.jsx
- components/analytics/LearningStyleAssessment.jsx
- components/analytics/RevenueChart.jsx
- components/analytics/StudentDashboard.jsx
- components/analytics/attribution.jsx
- components/analytics/track.jsx
- components/announcements/
- components/announcements/AnnouncementsPanel.jsx
- components/api/
- components/api/TenancyBridge.jsx
- components/api/contracts.js
- components/api/scoped.jsx
- components/api/scopedEntities.js
- components/api/tenancyEnforcer.js
- components/api/tenancyRuntime.js
- components/api/tenancyWarnings.js
- components/auth/
- components/auth/roles.jsx
- components/automation/
- components/automation/AutomationBuilder.jsx
- components/career/
- components/career/ResumeBuilder.jsx
- components/career/SalaryNegotiator.jsx



- components/certificates/
- components/certificates/CertificateView.jsx
- components/certificates/certificatesEngine.jsx
- components/collaboration/
- components/collaboration/Whiteboard.jsx
- components/community/
- components/community/ReportButton.jsx
- components/config/
- components/config/features.jsx
- components/courses/
- components/courses/CourseCard.jsx
- components/creator/
- components/creator/BulkOperations.jsx
- components/dashboard/
- components/dashboard/StatCard.jsx
- components/drip/
- components/drip/dripEngine.jsx
- components/enterprise/
- components/enterprise/TeamDashboard.jsx
- components/focus/
- components/focus/PomodoroTimer.jsx
- components/gamification/
- components/gamification/AchievementBadge.jsx
- components/gamification/PowerUpShop.jsx
- components/hooks/
- components/hooks/useLessonAccess.jsx
- components/hooks/useSession.jsx
- components/hooks/useStorefrontContext.jsx
- components/hooks/useTerminology.jsx
- components/insights/
- components/insights/LearningInsights.jsx
- components/instructor/
- components/instructor/CourseBuilder.jsx
- components/instructor/InstructorDashboard.jsx

- `components/instructor/TeachCourseCurriculum.jsx`
- `components/instructor/TeachCoursePricing.jsx`
- `components/instructor/TeachCourseSettings.jsx`
- `components/instructor/TeachCourseStudents.jsx`
- `components/integrations/`
- `components/integrations/GoogleServiceConnector.jsx`
- `components/language/`
- `components/language/FlashcardPractice.jsx`
- `components/language/LanguageSelector.jsx`
- `components/learning/`
- `components/learning/BookmarkPanel.jsx`
- `components/learning/CodePlayground.jsx`
- `components/learning/DiscussionThread.jsx`
- `components/learning/NotesPanel.jsx`
- `components/learning/PremiumVideoPlayer.jsx`
- `components/learning/QuizCard.jsx`
- `components/learning/TranscriptPanel.jsx`
- `components/marketing/`
- `components/marketing/CourseLandingPage.jsx`
- `components/materials/`
- `components/materials/materialsEngine.jsx`
- `components/mobile/`
- `components/mobile/OfflineMode.jsx`
- `components/moderation/`
- `components/moderation/ContentReportForm.jsx`
- `components/monetization/`
- `components/monetization/OfferCard.jsx`
- `components/navigation/`
- `components/navigation/CommandPalette.jsx`
- `components/notifications/`
- `components/notifications/NotificationCenter.jsx`
- `components/onboarding/`
- `components/onboarding/OnboardingFlow.jsx`
- `components/parent/`

- `components/parent/StudentProgressReport.jsx`
- `components/payouts/`
- `components/payouts/PayoutBatchManager.jsx`
- `components/proctoring/`
- `components/proctoring/ProctorView.jsx`
- `components/protection/`
- `components/protection/ProtectedContent.jsx`
- `components/scheduling/`
- `components/scheduling/StudyCalendar.jsx`
- `components/school/`

> Note: this is intentionally shallow. When coding, we use targeted search (rg) rather than browsing.

---

## Existing invariants docs (already in repo)

These are already present in the codebase and must remain true:

### SECURITY\_INVARIANTS.md

## Security Invariants

These are non-negotiable rules enforced by architecture.

### Tenancy

- Any entity listed in `src/components/api/scopedEntities.js` is school-scoped.
- All creates/updates/deletes for school-scoped entities must use:
  - `scopedCreate`, `scopedUpdate`, `scopedDelete`
- Direct calls to `base44.entities.<SchoolScopedEntity>.create/update/delete` are forbidden.
- The runtime enforcer will block missing/incorrect `school_id`.

### Access

- Expiring entitlements must be checked with `isEntitlementActive`.
- Lesson access uses `useLessonAccess` (drip + preview + license add-ons).

## **Content protection**

- Locked users must not receive full premium lesson content.
- Preview content must be strictly limited by policy.
- Watermarks must include user identity (email) when protected.

## **Quizzes**

- If questions are stored in `QuizQuestion`, do not fetch them when access is LOCKED.

## **ROADMAP\_CODEX.md**

# **Breslov Academy — Roadmap (Codex / Implementation)**

This repo follows a strict principle:

- No features are deleted. Features may be upgraded, merged, or hidden from main nav, but must stay discoverable in Vault.

## **P0 — Make it stable**

- 1 Tenancy invariants: all school-scoped CRUD uses `scoped*` helpers (never direct `base44.entities.X.create/update/delete`).
- 2 Access invariants: access checks are expiry-aware (`isEntitlementActive`) and drip-aware (`dripEngine`).
- 3 Reader invariants: locked users must not receive full premium content.
- 4 Build invariants: no Hook rule violations; no `CommonJS` require in ESM.

## **P1 — Quizzes (v9.0)**

- Teacher can create/edit/publish quizzes.
- Students can take quizzes.
- Attempts recorded.
- Question fetch gating: do not fetch questions when access is LOCKED.

- Preferred storage: Quiz (meta) + QuizQuestion (questions). Fallback to inline questions if entity absent.

## **P2 — Monetization + White-label**

- Multi-school storefront per school.
- Course access levels + bundles + add-ons.
- Stripe checkout flows.

## **P3 — Quality**

- Automated healthchecks.
- Release packaging with checksums.
- CI (typecheck, lint, tests).

## **ROADMAP\_STATUS.md**

# **Roadmap Status**

Date: 2026-01-01

### **Done**

- v7 foundation: Protected content system (copy/print/download blocking, watermarking, preview limits, license add-ons).
- Tenancy scaffolding: TenancyBridge, tenancyEnforcer, scoped\* helpers.
- Feature registry: Vault discoverability for all pages.

### **In progress (v9.0)**

- Quizzes: canonical routes + teacher editor + student take flow + attempt recording.
- Access hardening: fix useLessonAccess hook to be hook-rule compliant and expiry/drip aware.

### **Next**

- Refactor remaining legacy access\_tier checks to use entitlement-based access (backwards compatible).

- Add healthcheck scripts + CI.
- Release automation (PowerShell): generate stable ZIP + checksums + GitHub branch update.

## **HEALTHCHECKS.md**

# **Healthchecks**

Run these before tagging a stable ZIP.

## **Local**

- 1 Install deps
  - `npm ci` (recommended) or `npm install`
- 2 Build
  - `npm run build`
- 3 Lint
  - `npm run lint`
- 4 Dev smoke
  - `npm run dev`
  - Open: Dashboard, Courses, CourseDetail, LessonViewer, Reader, Vault
  - Teacher: Teach → Course Builder → Lesson Editor
  - Quizzes: `/teach/quizzes` → create quiz → publish → `/quiz/:id`

## **Tenancy smoke**

- Create a quiz/attempt only when an active school is set.
- Verify cross-school reads are denied unless explicitly allowed.

## **Release artifacts**

- Generate a ZIP that excludes `node_modules`, `.git`, `dist`, `coverage`.
- Include checksums and a BUILDINFO record.

## **V9\_CONTENT\_PROTECTION\_REALITY.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_CONTENT\_PROTECTION\_REALITY.md

# **Content Protection & Access Control (Reality + Hard Rules)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **Access levels used in code**

Source of truth:

- `src/components/hooks/useLessonAccess.jsx`

Access levels:

- FULL — user has active entitlement for course (or ALL\_COURSES)
- PREVIEW — no entitlement but previews allowed by policy
- LOCKED — no access; no preview
- DRIP\_LOCKED — user has entitlement but lesson not yet available via drip schedule

## **Content protection policy**

Entity:

- `ContentProtectionPolicy` (school-scoped)

Fetches by:

- `useLessonAccess()` `queryKey` `['protection-policy', schoolId]`

Default policy lives in hook.

## **“No leakage” rule**

If `accessLevel` is `LOCKED` or `DRIP_LOCKED`:

- Do NOT fetch premium materials.
- Do NOT fetch quiz questions (see invariants).

Enforcement helpers:

- `src/components/materials/materialsEngine.jsx`
- `shouldFetchMaterials(accessLevel)` must gate any material retrieval
- `src/components/security/AccessGate.jsx`

- Unified lock UI for LOCKED/DRIP\_LOCKED

## **Secure downloads**

- `getSecureDownloadUrl()` in `materialsEngine.jsx`
- Must return URL only when:
- school policy allows it
- entitlement/license allows it
- event is logged (best-effort)

## **Preview mode requirements**

- Preview must be limited by policy:
- time: `max_preview_seconds`
- text: `max_preview_chars`
- Watermarking must include user identity (email) when protected.

## **Tech debt to eliminate (in order)**

1) Any direct “`access_tier`” checks in pages → replace with entitlement-based access 2) Any material fetch calls that don’t use `shouldFetchMaterials` 3) Any components that render full content in PREVIEW without trimming



## V9\_FEATURE\_REGISTRY\_REALITY.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_FEATURE\_REGISTRY\_REALITY.md

# V9.0 Stable — Feature Registry (Reality Dump)

Prepared: January 01, 2026 (Asia/Jerusalem)

Source of truth:

- `src/components/config/features.jsx`

This registry powers:

- Left navigation groups (`src/Layout.jsx` via `getNavGroupsForAudience`)
- Vault discoverability (`src/pages/Vault.jsx`)
- Canonical route selection (`src/utils/index.ts`)

## Key policies

- Never delete a feature from the registry.
- If a page exists but is missing from the registry, add it as hidden (and/or `vaultOnly`) rather than omitting it.

## Known mismatch in v9.0 ZIP

The following page components are present in `pages.config` but are not in the feature registry:

- `TeachQuizEditor`
- `QuizTake`

Fix plan:

- Add registry entries for both (hidden), and reference their canonical routes:
- `TeachQuizEditor` → `/teach/quizzes/new` (and `/teach/quizzes/:quizId`)
- `QuizTake` → `/quiz/:quizId`

---

## Feature list (76)

Feature key	Label	Canonical route	Area	Audiences	Nav order	Hidden	Vault only
--- --- --- --- --- ---	SchoolAdmin	School Admin	/schooladmin	admin	admin	1	

```

| | | | SchoolAnalytics | School Analytics | /schoolanalytics | admin | admin | 2 | | | |
SchoolMonetization | Monetization | /schoolmonetization | admin | admin | 3 | | | |
AuditLogViewer | Audit Log | /auditlogviewer | admin | admin | | | | | SchoolNew |
Create School | /schoolnew | admin | admin | | | | | LegacyMigration | Legacy Migration |
/legacymigration | admin | admin | | | | | NetworkAdmin | Network Admin |
/networkadmin | admin | admin | | | | | AdminHardening | Security Hardening |
/adminhardening | admin | admin | | | | | SchoolStaff | Staff Management |
/schoolstaff | admin | admin | | | | | Dashboard | Dashboard | /dashboard | core |
student,teacher,admin | 1 | | | | Courses | Courses | /courses | core |
student,teacher,admin | 2 | | | | Reader | Smart Reader | /reader | core |
student,teacher,admin | 3 | | | | Feed | Community | /feed | core |
student,teacher,admin | 4 | | | | SchoolSearch | Search | /schoolsearch | core |
student,teacher,admin | 5 | | | | MyQuizzes | My Quizzes | /my-quizzes | core |
student,teacher,admin | 6 | | | | Account | My Account | /account | core |
student,teacher,admin | 10 | | | | Community | Community | /community | core |
student,teacher,admin | | | | | CourseDetail | Course Detail | /coursedetail | core |
student,teacher,admin | | | | | Downloads | Downloads | /downloads | core |
student,teacher,admin | | | | | Forums | Forums | /forums | core |
student,teacher,admin | | | | | LearningPaths | Learning Paths | /learningpaths |
core | student,teacher,admin | | | | | LessonViewer | Lesson Viewer | /lessonviewer |
core | student,teacher,admin | | | | | LessonViewerPremium | Lesson Viewer Premium |
/lessonviewerpremium | core | student,teacher,admin | | | | | Messages | Messages |
/messages | core | student,teacher,admin | | | | | MyProgress | My Progress |
/myprogress | core | student | | | | | Achievements | Achievements | /achievements |
labs | student,teacher,admin | | | | | AdaptiveLearning | Adaptive Learning |
/adaptivelearning | labs | student,teacher,admin | | | | | Affiliate | Affiliate
Program | /affiliate | labs | student,teacher,admin | | | | | AlumniNetwork | Alumni
Network | /alumninetwork | labs | student,teacher,admin | | | | | Analytics |
Analytics | /analytics | labs | student,teacher,admin | | | | | CareerPaths | Career
Paths | /careerpaths | labs | student,teacher,admin | | | | | Challenges | Challenges |
/challenges | labs | student,teacher,admin | | | | | CohortDetail | Cohort Detail |
/cohortdetail | labs | student,teacher,admin | | | | | Cohorts | Cohorts | /cohorts |
labs | student,teacher,admin | | | | | Events | Events | /events | labs |
student,teacher,admin | | | | | HabitTracker | Habit Tracker | /habittracker | labs |
student | | | | | LanguageLearning | Languages | /languagelearning | labs |
student,teacher,admin | | | | | Languages | Languages (Legacy) | /languages | labs |
student,teacher,admin | | | | | Leaderboard | Leaderboard | /leaderboard | labs |
student,teacher,admin | | | | | LiveStreams | Live Streams | /livestreams | labs |
student,teacher,admin | | | | | Marketplace | Marketplace | /marketplace | labs |
student,teacher,admin | | | | | Mentorship | Mentorship | /mentorship | labs |
student,teacher,admin | | | | | Microlearning | Microlearning | /microlearning | labs
| student | | | | | StudySetNew | New Study Set | /studyssetnew | labs |
student,teacher,admin | | | | | Offline | Offline | /offline | labs |
student,teacher,admin | | | | | StudySetPractice | Practice | /studyssetpractice | labs
| student,teacher,admin | | | | | RewardsShop | Rewards Shop | /rewardssshop | labs |
student | | | | | Scholarships | Scholarships | /scholarships | labs | student | | | | |
Skills | Skills | /skills | labs | student,teacher,admin | | | | | StudyBuddies | Study
Buddies | /studybuddies | labs | student | | | | | StudySet | Study Set | /studysset |

```

```

labs | student,teacher,admin | | | | StudySets | Study Sets | /studysets | labs |
student,teacher,admin | | | | Tournaments | Tournaments | /tournaments | labs |
student | | | | AffiliateProgram | Affiliate Program Info | /affiliateprogram |
marketing | public,student,teacher,admin | | | | CertificateVerify | Certificate
Verification | /certificateverify | marketing | public | | | | SchoolCheckout |
Checkout | /schoolcheckout | marketing | public,student,teacher,admin | | | |
SchoolCourses | Course Catalog | /schoolcourses | marketing |
public,student,teacher,admin | | | | SchoolLanding | Landing Page | /schoollanding
| marketing | public,student,teacher,admin | | | | SchoolPricing | Pricing |
/schoolpricing | marketing | public,student,teacher,admin | | | | CourseSales |
Sales Page | /coursesales | marketing | public,student,teacher,admin | | | |
SchoolThankYou | Thank You | /schoolthankyou | marketing |
public,student,teacher,admin | | | | InviteAccept | Accept Invite | /inviteaccept |
system | public | | | | Integrations | Integrations | /integrations | system |
student,teacher,admin | | | | Integrity | Integrity Check | /integrity | system |
admin | | | | SchoolJoin | Join School | /schooljoin | system |
student,teacher,admin | | | | oauth2callback | OAuth Callback | /oauth2callback |
system | public | | | | Portfolio | Profile | /portfolio | system |
student,teacher,admin | | | | SchoolSelect | School Select | /schoolselect | system |
student,teacher,admin | | | | Subscription | Subscription | /subscription | system |
student,teacher,admin | | | | Vault | Vault | /vault | system |
student,teacher,admin | | | | Teach | Teach | /teach | teach | teacher,admin | 1 | | |
TeachAnalytics | Teaching Analytics | /teachanalytics | teach | teacher,admin | 2 | | |
| TeachQuizzes | Quizzes | /teach/quizzes | teach | teacher,admin | 3 | | |
TeachCourse | Course Builder | /teachcourse | teach | teacher,admin | | | |
TeachLesson | Lesson Editor | /teachlesson | teach | teacher,admin | | | |
TeachCourseNew | New Course | /teachcoursenew | teach | teacher,admin | | | |

```

## **V9\_PUBLIC\_SITE\_AND\_PORTALS\_PLAN.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_PUBLIC\_SITE\_AND\_PORTALS\_PLAN.md

# **Public Website + Separate Portals (Design + Implementation Plan)**

Prepared: January 01, 2026 (Asia/Jerusalem)

This document translates the “unified master plan” into this exact v9.0 codebase.

## **Goal**

Add: 1) Public marketing site (trust + conversion) 2) Separate portal experiences:

- Student (/student/\*)
- Teacher (/teacher/\*)
- School Admin (/admin/\*)
- Super Admin (/superadmin/\*)

3) Separate login URLs:

- /login/student
- /login/teacher
- optional /login/admin

WITHOUT breaking:

- Existing Base44 legacy routes (/ :pageName)
  - Vault discoverability
  - Tenancy invariants
  - Access/entitlement gating
- 

## **A) Public marketing site**

### **New routes**

- / (public landing if not signed in; otherwise route to portal)
- /about
- /how-it-works

- /faq
- /contact
- /privacy
- /terms

### **Implementation approach (Vite + React Router)**

1) Add explicit routes in `src/App.jsx` above the generic pages map. 2) Create new lightweight page components under:

- `src/pages/public/` (recommended) OR `src/pages/` (keep simple)

3) Add feature registry entries (audiences: `public`, `hidden: true` for legal pages if desired) 4) Reuse existing design system:

- `src/components/theme/tokens`
- `GlassCard`, `PageShell`, `SectionHeader`

### **Root routing logic**

/ should:

- show Landing if user is not authenticated
  - show “select school” if user has multiple memberships and no active school
  - show portal home if active school is set
- 

## **B) Portals**

### **New route groups**

- `/student/*`
- `/teacher/*`
- `/admin/*`
- `/superadmin/*`

### **The “smart” way (uses existing session audience)**

Create a tiny router bridge:

- `src/components/routing/PortalGate.jsx`

Responsibilities: 1) Read current session: `useSession()` 2) Enforce allowed roles:

- student portal: any membership role
- teacher portal: instructor/TA/admin/owner
- admin portal: admin/owner
- superadmin: platform global admin (see roles helper)

3) Set audience in session (or localStorage intent) so nav renders correct items. 4) Render the existing Layout with portal-specific default route.

## Portal home pages

Define these defaults:

- Student: /student/dashboard → renders existing Dashboard module (or new StudentDashboard)
- Teacher: /teacher/dashboard → renders existing Teach / TeachAnalytics
- Admin: /admin/dashboard → renders existing SchoolAdmin / SchoolAnalytics
- Superadmin: /superadmin/network → renders NetworkAdmin + audit log

## Keep legacy routes

All existing routes remain working:

- /dashboard, /Dashboard
- /teach, /Teach

etc.

Over time we add gentle redirects:

- /teach → /teacher/teach (optional)

but only after stability is proven.

---

## C) Separate login pages

Add explicit routes:

- /login/student
- /login/teacher
- /login/admin (optional)

These pages:

- set intended audience

- explain what happens
  - trigger `base44.auth.redirectToLogin()`
- 

## **D) School/entity signup**

We already have `SchoolNew` and `SchoolJoin`. We will evolve them into:

- `/signup/school` (application wizard)
  - admin approvals inside `/admin/*`
- 

## **E) Navigation**

The existing layout uses:

- `getNavGroupsForAudience(audience)` from the registry

So portals become “real” immediately once audience is set.

We will add:

- Portal switch links (if user has multiple roles)
- Distinct portal topbars (colors/icons) without duplicating logic

## **V9\_QA\_RELEASES.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_QA\_RELEASES.md

# **QA, Healthchecks, Releases (Reality + Future)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **Existing healthchecks**

From repo root HEALTHCHECKS.md:

## **Healthchecks**

Run these before tagging a stable ZIP.

### **Local**

#### 1 Install deps

- `npm ci` (recommended) or `npm install`

#### 2 Build

- `npm run build`

#### 3 Lint

- `npm run lint`

#### 4 Dev smoke

- `npm run dev`
- Open: Dashboard, Courses, CourseDetail, LessonViewer, Reader, Vault
- Teacher: Teach → Course Builder → Lesson Editor
- Quizzes: `/teach/quizzes` → create quiz → publish → `/quiz/:id`

## **Tenancy smoke**

- Create a quiz/attempt only when an active school is set.
- Verify cross-school reads are denied unless explicitly allowed.

## **Release artifacts**



- Generate a ZIP that excludes node\_modules, .git, dist, coverage.
- Include checksums and a BUILDINFO record.

## **Existing release tooling (Windows PowerShell)**

- scripts/release.ps1
- stages repo files
- writes checksums
- writes BUILDINFO
- zips output
- scripts/update-github-from-zip.ps1
- clones repo
- expands zip
- robocopy /MIR into repo (excluding .git/node\_modules/dist/build/coverage)
- pushes branch

## **Future CI plan (GitHub Actions)**

- PR: lint, typecheck, build
- Optional: small test suite for invariants
- Tag: attach zipped artifact + checksums to GitHub Release

## **V9\_QUIZZES\_REALITY.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_QUIZZES\_REALITY.md

# **Quizzes (v9.0 Reality + Future Extensions)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **Canonical quiz routes (already implemented in router)**

Defined in `src/App.jsx`:

- `/my-quizzes` → `Pages.MyQuizzes`
- `/quiz/:quizId` → `Pages.QuizTake`
- `/teach/quizzes` → `Pages.TeachQuizzes`
- `/teach/quizzes/new` → `Pages.TeachQuizEditor`
- `/teach/quizzes/:quizId` → `Pages.TeachQuizEditor`

## **Storage model (from scoped entities list)**

School-scoped entities include:

- Quiz (meta)
- QuizQuestion (questions)
- QuizAttempt (attempts)

Confirm in backend + usage:

- search in code for `base44.entities.Quiz`, `QuizQuestion`, `QuizAttempt`
- enforce scoping via `scoped*`

## **Security invariant**

From `SECURITY_INVARIANTS.md`:

- If questions are stored in `QuizQuestion`, do not fetch them when access is `LOCKED`.

## **v9.1+ extensions (planned)**

- Question bank / reusable items
- Randomization by difficulty
- Anti-cheating: time windows, per-lesson gating, watermarking

- Analytics: item discrimination, mastery tracking

## **V9\_ROADMAP\_EXTREME\_DETAIL.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_ROADMAP\_EXTREME\_DETAIL.md

### **Roadmap — Extreme Detail (v9.1 → v10+)**

Prepared: January 01, 2026 (Asia/Jerusalem)

This roadmap is designed to be used as the “coding script” in the next chat. It assumes v9.0-stable is the starting baseline (this ZIP).

---

### **Guiding rules (non-negotiable)**

1) No features removed — Vault remains the complete registry. 2) Tenancy is enforced everywhere:

- use scoped\* for school-scoped entities
- keep runtime enforcer enabled

3) Access gating is enforced everywhere:

- never fetch premium materials in LOCKED/DRIP\_LOCKED
- never fetch quiz questions in LOCKED

4) Prefer production-grade solutions:

- route guards + server checks
  - audit logs for admin actions
  - idempotency for money
- 

### **Phase 0 — Intake & Reality Lock (1 PR)**

#### **Deliverables**

- [ ] “Reality Map” docs committed under docs/:
- Codebase map
- Route map
- Feature registry dump

- Tenancy/access invariants
- [ ] Add missing FEATURES registry entries:
- TeachQuizEditor (hidden)
- QuizTake (hidden)
- [ ] Ensure npm run build passes (typecheck + lint)

## Files

- src/components/config/features.jsx
- docs/V9\_CODEBASE\_MAP.md (new)
- docs/V9\_ROUTE\_MAP.md (new)

## Acceptance tests

- [ ] Vault still renders and lists all features
  - [ ] /teach/quizzes/new and /quiz/:id still work
  - [ ] No new console errors in Layout
- 

# Phase 1 (v9.1) — Public Site + Login Split + Portal Shells

Goal: get the big UX architecture in place without touching deep LMS logic.

## 1.1 Public landing + trust pages

### Tasks

- [ ] Create pages:
- Landing
- About
- HowItWorks
- FAQ
- Contact
- Privacy
- Terms

- [ ] Add explicit routes in `src/App.jsx` (above generic map)
- [ ] Add FEATURE registry entries (audience: public) where appropriate
- [ ] Add SEO basics (title/meta) using `react-helmet-async` (new dep) OR minimal `document.title` per page

## Acceptance

- [ ] Guest sees landing at /
- [ ] Authenticated user is routed to portal home
- [ ] Lighthouse is “good enough” (no huge images; lazy load)

## 1.2 Separate login URLs

### Tasks

- [ ] Add routes:
- `/login/student`
- `/login/teacher`
- optional `/login/admin`
- [ ] Store intended audience (`localStorage`)
- [ ] Call `base44.auth.redirectToLogin()`

## Acceptance

- [ ] Each login page works and returns user correctly
- [ ] Intended portal experience loads after return

## 1.3 Portal route groups (no feature breakage)

### Tasks

- [ ] Add route groups:
- `/student/*`
- `/teacher/*`
- `/admin/*`
- `/superadmin/*`
- [ ] Add `PortalGate` component:
- checks session
- enforces roles

- sets audience
- [ ] Add portal home routes:
- /student/dashboard
- /teacher/dashboard
- /admin/dashboard

## **Acceptance**

- [ ] Teacher cannot access student portal routes that require teacher privileges
  - [ ] Admin portal blocks non-admin
  - [ ] Legacy routes still work unchanged
- 

# **Phase 2 (v9.2) — Onboarding & School/Entity Signup (Multi-tenant excellence)**

## **2.1 School signup “application”**

- [ ] Route: /signup/school
- [ ] Wizard steps:  
1) org info 2) owner info 3) confirm
- [ ] Create pending school record (or SchoolApplication entity)
- [ ] Send confirmation + show status page

## **2.2 Invites + approvals**

- [ ] Admin portal:
- approvals queue
- invite teacher/student
- [ ] Invite accept flow hardening:
- InviteAccept page (already exists)

## **2.3 Tenant settings**

- [ ] ContentProtectionSettings component already exists:
- wire into Admin portal settings

- [ ] add audit logs on every policy change
- 

## **Phase 3 (v9.3) — Student Learning WOW**

Focus: learning flow clarity and delight.

- [ ] Student dashboard redesign (KPI cards, next lesson, streak)
  - [ ] Course detail page:
    - clean curriculum
    - progress
    - locked vs preview states
  - [ ] Lesson viewer:
    - make AccessGate first-class
    - preview trimming enforced
    - better media player UX
  - [ ] Reader:
    - night mode
    - highlights/notes (if entity exists)
- 

## **Phase 4 (v9.4) — Teacher Power Tools**

- [ ] Course builder UX (drag/drop lessons, bulk actions)
  - [ ] Quiz authoring upgrades:
    - question bank
    - import/export
    - analytics
  - [ ] Grading workflow improvements (assignments/submissions)
- 

## **Phase 5 (v9.5) — Storefront Conversion & Trust**



- [ ] School landing improvements (hero, testimonials, FAQs)
  - [ ] Pricing page:
  - bundles + add-ons display
  - [ ] Checkout reliability:
  - idempotency
  - post-checkout entitlement verification
  - [ ] Affiliate tracking + attribution hardening
- 

## **Phase 6 (v9.6) — Observability + QA + Performance Budgets**

- [ ] Add GitHub Actions CI:
  - lint + typecheck + build
  - [ ] Add minimal unit tests:
  - tenancy scoping
  - access gating (no fetch when locked)
  - money idempotency guards
  - [ ] Add performance budgets:
  - route-based code splitting
  - image size checks
- 

## **Phase 7 (v10) — Major upgrades (only after stable loop works)**

- Feature flags framework
- Multi-language content authoring
- Integrations marketplace

## V9\_ROUTES\_REALITY\_MAP.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_ROUTES\_REALITY\_MAP.md

# V9.0 Stable — Route Map (Reality)

Prepared: January 01, 2026 (Asia/Jerusalem)

## How routing works in this codebase

The router is defined in `src/App.jsx`.

### 1) Main page

- `/` renders `pagesConfig.mainPage` (currently Dashboard) via `src/pages.config.js`.

### 2) Legacy Base44 page routes

For every entry in `pagesConfig.PAGES` (78 pages), the router registers:

- `/ {PageKey}` (e.g. `/Dashboard`, `/Vault`)

Additionally, the router includes a case-insensitive fallback:

- `/:pageName` (e.g. `/dashboard` → matches Dashboard)

So both `/Dashboard` and `/dashboard` work.

### 3) Canonical human URLs (preferred)

Canonical URLs are defined in the Feature Registry at:

- `src/components/config/features.jsx`

Routing preference is applied by:

- `src/utils/index.ts` (`createPageUrl()` checks the registry first)

### 4) Explicit “special” routes (hard-coded)

These are registered explicitly before the generic page map:

Quizzes (v9.0 canonical):

- `/my-quizzes` → `Pages.MyQuizzes`
- `/quiz/:quizId` → `Pages.QuizTake`
- `/teach/quizzes` → `Pages.TeachQuizzes`

- /teach/quizzes/new → Pages.TeachQuizEditor
- /teach/quizzes/:quizId → Pages.TeachQuizEditor

Storefront (guest-safe):

- /s/:schoolSlug → SchoolLanding
- /s/:schoolSlug/courses → SchoolCourses
- /s/:schoolSlug/course/:courseId → CourseSales
- /s/:schoolSlug/pricing → SchoolPricing
- /s/:schoolSlug/checkout → SchoolCheckout
- /s/:schoolSlug/thank-you → SchoolThankYou
- /s/:schoolSlug/certificate/:certificateId → CertificateVerify

---

## Page registry → routes table (78 pages)

PageKey	Canonical route (FEATURES)	Legacy route	Notes	--- --- --- ---	Account
/account	/Account		Achievements	/achievements	/Achievements
AdaptiveLearning	/adaptivelearning	/AdaptiveLearning	vaultOnly		
AdminHardening	/adminhardening	/AdminHardening	vaultOnly		Affiliate
/affiliate	/Affiliate	vaultOnly		AffiliateProgram	/affiliateprogram
/AffiliateProgram	vaultOnly		AlumniNetwork	/alumninetwork	/AlumniNetwork
vaultOnly		Analytics	/analytics	/Analytics	vaultOnly
	AuditLogViewer	/auditlogviewer	/AuditLogViewer	vaultOnly	
CareerPaths	/careerpaths	/CareerPaths	vaultOnly		CertificateVerify
/CertificateVerify	hidden		Challenges	/challenges	/Challenges
	CohortDetail	/cohortdetail	/CohortDetail	hidden	
Cohorts	/cohorts	/Cohorts		Community	/community
/Community	vaultOnly		CourseDetail	/coursedetail	/CourseDetail
hidden		CourseSales	/coursesales	/CourseSales	hidden
	Courses	/courses	/Courses		Dashboard
/dashboard	/Dashboard		Downloads	/downloads	/Downloads
vaultOnly		Events	/events	/Events	vaultOnly
	Feed	/feed	/Feed		Forums
/forums	/Forums	vaultOnly		HabitTracker	/habittracker
/HabitTracker	vaultOnly		Integrations	/integrations	/Integrations
	Integrity	/integrity	/Integrity	vaultOnly	
InviteAccept	/inviteaccept	/InviteAccept	hidden		LanguageLearning
/languagelearning	/LanguageLearning		Languages	/languages	/Languages
vaultOnly		Leaderboard	/leaderboard	/Leaderboard	vaultOnly
	LearningPaths	/learningpaths	/LearningPaths	vaultOnly	
LegacyMigration	/legacymigration	/LegacyMigration	vaultOnly		LessonViewer
/lessonviewer	/LessonViewer	hidden		LessonViewerPremium	/lessonviewerpremium
/LessonViewerPremium	hidden		LiveStreams	/livestreams	/LiveStreams
vaultOnly		Marketplace	/marketplace	/Marketplace	vaultOnly
	Mentorship	/mentorship	/Mentorship	vaultOnly	
Messages	/messages	/Messages	vaultOnly		Microlearning

```

/microlearning | /Microlearning | vaultOnly | | MyProgress | /myprogress |
/MyProgress | vaultOnly | | MyQuizzes | /my-quizzes | /MyQuizzes | | | NetworkAdmin |
/networkadmin | /NetworkAdmin | vaultOnly | | Offline | /offline | /Offline | | |
Portfolio | /portfolio | /Portfolio | | | QuizTake | ` | /QuizTake | not in feature
registry | | Reader | /reader | /Reader | | | RewardsShop | /rewardsshop |
/RewardsShop | vaultOnly | | Scholarships | /scholarships | /Scholarships |
vaultOnly | | SchoolAdmin | /schooladmin | /SchoolAdmin | | | SchoolAnalytics |
/schoolanalytics | /SchoolAnalytics | | | SchoolCheckout | /schoolcheckout |
/SchoolCheckout | hidden | | SchoolCourses | /schoolcourses | /SchoolCourses | |
| | SchoolJoin | /schooljoin | /SchoolJoin | hidden | | SchoolLanding | /schoollanding
| /SchoolLanding | | | SchoolMonetization | /schoolmonetization |
/SchoolMonetization | | | SchoolNew | /schoolnew | /SchoolNew | hidden | |
SchoolPricing | /schoolpricing | /SchoolPricing | hidden | | SchoolSearch |
/schoolsearch | /SchoolSearch | | | SchoolSelect | /schoolselect | /SchoolSelect |
hidden | | SchoolStaff | /schoolstaff | /SchoolStaff | vaultOnly | |
SchoolThankYou | /schoolthankyou | /SchoolThankYou | hidden | | Skills | /skills |
/Skills | vaultOnly | | StudyBuddies | /studybuddies | /StudyBuddies | vaultOnly
| | StudySet | /studysset | /StudySet | hidden | | StudySetNew | /studyssetnew |
/StudySetNew | hidden | | StudySetPractice | /studyssetpractice | /StudySetPractice
| hidden | | StudySets | /studyssets | /StudySets | | | Subscription | /subscription
| /Subscription | hidden | | Teach | /teach | /Teach | | | TeachAnalytics |
/teachanalytics | /TeachAnalytics | | | TeachCourse | /teachcourse | /TeachCourse
| hidden | | TeachCourseNew | /teachcoursenew | /TeachCourseNew | hidden | |
TeachLesson | /teachlesson | /TeachLesson | hidden | | TeachQuizEditor | |
/TeachQuizEditor | not in feature registry | | TeachQuizzes | /teach/quizzes |
/TeachQuizzes | | | Tournaments | /tournaments | /Tournaments | vaultOnly | |
Vault | /vault | /Vault | | | oauth2callback | /oauth2callback | /oauth2callback` |
hidden |

```

---

## Important routing implication for the “new portal split”

Because the current router depends on:

- `/ {PageKey}` and `/:pageName`

...we must add new portal routes without breaking these.

Recommended approach:

- Add explicit route groups:
- `/student/*`
- `/teacher/*`
- `/admin/*`
- `/superadmin/*`

- Keep existing routes working forever (or alias them to the new groups).

## **V9\_TENANCY\_REALITY.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_TENANCY\_REALITY.md

# **Tenancy & Data Scoping (Reality)**

Prepared: January 01, 2026 (Asia/Jerusalem)

## **Core principle**

If an entity is “school-scoped”, every query + write must be scoped to `school_id`.

## **Source of truth for scoping**

- `src/components/api/scopedEntities.js`
- `SCHOOL_SCOPED_ENTITIES` (must include `school_id`)
- `GLOBAL_ENTITIES` (no `school_id` requirement)

## **Safe query/write helpers**

- `src/components/api/scoped.jsx`
- `scopedList(entity, schoolId, sort, limit)`
- `scopedFilter(entity, schoolId, filter)`
- `scopedCreate(entity, payload, schoolId)`
- `scopedUpdate(entity, id, payload, schoolId, strict?)`
- `scopedDelete(entity, id, schoolId, strict?)`

## **Runtime enforcement (automatic guard)**

- Installed in `src/api/base44Client.js` via `installTenancyEnforcer(...)`
- Runtime context is kept in sync by:
- `src/components/api/TenancyBridge.jsx`
- Tenancy context lives in:
- `src/components/api/tenancyRuntime.js` (module state)

## **Session source of truth**

- `src/components/hooks/useSession.jsx`
- `activeSchoolId`

- `activeSchool`
- `memberships`
- `role` (per active school)
- `audience` (student/teacher/admin) derived from role

## **Practical coding rules**

1) When you touch any page that reads/writes scoped entities:

- replace direct `base44.entities.X.*` calls with `scoped*`

2) For global admins:

- use explicit “global escape hatch” functions (as already provided by `tenancyEnforcer`)

3) Never rely on UI-only tenancy enforcement — the API layer must enforce too.

## **Smoke checks to run after tenant changes**

(See `HEALTHCHECKS.md` in repo root.)

## V9\_UI\_UX\_REVAMP\_PLAN.md

Breslov\_Academy\_Coding\_Book\_v9\_1/SPECPACK\_V5/V9\_UI\_UX\_REVAMP\_PLAN.md

# Premium UI/UX Revamp Plan (University-grade, Calm, WOW)

Prepared: January 01, 2026 (Asia/Jerusalem)

This codebase already has:

- Tailwind
- shadcn/ui + Radix
- Theme tokens: src/components/theme/tokens
- Theme toggle: src/components/theme/ThemeToggle

## Strategy: “Shell first, then modules”

1) Upgrade tokens + themes (light/dark/night) 2) Make Layout + PageShell consistent 3) Standardize reusable components (cards, tables, forms, empty states) 4) Convert pages module-by-module without feature loss 5) Final polish: micro-interactions, accessibility audit, perf budgets

## Hard requirements

- No harsh whites (use surfaces)
- Large readable typography
- Reduced motion support
- Responsive layouts, especially dashboard + course pages

## Deliverables (concrete)

- Token updates:
- src/components/theme/tokens.\*
- New component primitives:
- GlassCard (already exists)
- DataTable, StatCard, KPIGrid
- WizardShell (for onboarding)
- Landing hero:
- subtle gradients, strong type, CTAs





## Part V — Stable Root Docs

### HEALTHCHECKS.md

Breslov\_Academy\_Coding\_Book\_v9\_1/STABLE\_ROOT\_DOCS/HEALTHCHECKS.md

## Healthchecks

Run these before tagging a stable ZIP.

### Local

- 1 Install deps
  - `npm ci` (recommended) or `npm install`
- 2 Build
  - `npm run build`
- 3 Lint
  - `npm run lint`
- 4 Dev smoke
  - `npm run dev`
  - Open: Dashboard, Courses, CourseDetail, LessonViewer, Reader, Vault
  - Teacher: Teach → Course Builder → Lesson Editor
  - Quizzes: `/teach/quizzes` → create quiz → publish → `/quiz/:id`

### Tenancy smoke

- Create a quiz/attempt only when an active school is set.
- Verify cross-school reads are denied unless explicitly allowed.

### Release artifacts

- Generate a ZIP that excludes `node_modules`, `.git`, `dist`, `coverage`.
- Include checksums and a BUILDINFO record.

## **README.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/STABLE\_ROOT\_DOCS/README.md

## **Base44 App**

## **ROADMAP\_CODEX.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/STABLE\_ROOT\_DOCS/ROADMAP\_CODEX.md

# **Breslov Academy — Roadmap (Codex / Implementation)**

This repo follows a strict principle:

- No features are deleted. Features may be upgraded, merged, or hidden from main nav, but must stay discoverable in Vault.

## **P0 — Make it stable**

- 1 Tenancy invariants: all school-scoped CRUD uses scoped\* helpers (never direct `base44.entities.X.create/update/delete`).
- 2 Access invariants: access checks are expiry-aware (`isEntitlementActive`) and drip-aware (`dripEngine`).
- 3 Reader invariants: locked users must not receive full premium content.
- 4 Build invariants: no Hook rule violations; no CommonJS require in ESM.

## **P1 — Quizzes (v9.0)**

- Teacher can create/edit/publish quizzes.
- Students can take quizzes.
- Attempts recorded.
- Question fetch gating: do not fetch questions when access is LOCKED.
- Preferred storage: Quiz (meta) + QuizQuestion (questions). Fallback to inline questions if entity absent.

## **P2 — Monetization + White-label**

- Multi-school storefront per school.
- Course access levels + bundles + add-ons.
- Stripe checkout flows.

## **P3 — Quality**

- Automated healthchecks.

- Release packaging with checksums.
- CI (typecheck, lint, tests).

## **ROADMAP\_STATUS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/STABLE\_ROOT\_DOCS/ROADMAP\_STATUS.md

# **Roadmap Status**

Date: 2026-01-01

## **Done**

- v7 foundation: Protected content system (copy/print/download blocking, watermarking, preview limits, license add-ons).
- Tenancy scaffolding: TenancyBridge, tenancyEnforcer, scoped\* helpers.
- Feature registry: Vault discoverability for all pages.

## **In progress (v9.0)**

- Quizzes: canonical routes + teacher editor + student take flow + attempt recording.
- Access hardening: fix useLessonAccess hook to be hook-rule compliant and expiry/drip aware.

## **Next**

- Refactor remaining legacy access\_tier checks to use entitlement-based access (backwards compatible).
- Add healthcheck scripts + CI.
- Release automation (PowerShell): generate stable ZIP + checksums + GitHub branch update.

## **SECURITY\_INVARIANTS.md**

Breslov\_Academy\_Coding\_Book\_v9\_1/STABLE\_ROOT\_DOCS/SECURITY\_INVARIANTS.md

# **Security Invariants**

These are non-negotiable rules enforced by architecture.

## **Tenancy**

- Any entity listed in `src/components/api/scopedEntities.js` is school-scoped.
- All creates/updates/deletes for school-scoped entities must use:
  - `scopedCreate`, `scopedUpdate`, `scopedDelete`
- Direct calls to `base44.entities.<SchoolScopedEntity>.create/update/delete` are forbidden.
- The runtime enforcer will block missing/incorrect `school_id`.

## **Access**

- Expiring entitlements must be checked with `isEntitlementActive`.
- Lesson access uses `useLessonAccess` (drip + preview + license add-ons).

## **Content protection**

- Locked users must not receive full premium lesson content.
- Preview content must be strictly limited by policy.
- Watermarks must include user identity (email) when protected.

## **Quizzes**

- If questions are stored in `QuizQuestion`, do not fetch them when access is LOCKED.