

Toodle – Component Inventory

Frontend Architecture Reference & Implementation Status

Project: Toodle Tutor Management System

Framework: React 19 + Vite 6 + TailwindCSS v4

Auth: Auth0 (@auth0/auth0-react)

Total Components: 28

28
TOTAL

4
IMPLEMENTED

3
PARTIAL

21
STUB (TODO)

Auth Components AUTH src/components/auth/

LoginButton
auth/LoginButton.jsx

STUB – TODO

Triggers the Auth0 Universal Login redirect flow. Renders a customizable button with loading state while Auth0 SDK initializes. Accepts `variant`, `size`, and `className` props.

PROPS

Prop	Type	Description
variant	string?	Button visual variant (primary, ghost, outline)
size	string?	Button size: sm md lg
className	string?	Additional TailwindCSS class overrides
label	string?	Button label text (default: "Log In")

DEPENDENCIES

useAuth() → login() / loginWithRedirect()

<LoginButton size="lg" className="w-full" />

LogoutButton
auth/LogoutButton.jsx

STUB – TODO

Terminates the user's Auth0 session and redirects to `window.location.origin`. Cleans up cached session data and `toodle_dev_user` from `localStorage` on logout.

PROPS

Prop	Type	Description
variant	string?	Button visual variant
size	string?	sm md lg

DEPENDENCIES

useAuth() → logout()

<LogoutButton variant="ghost" size="sm" />

ProtectedRoute
auth/ProtectedRoute.jsx

IMPLEMENTED

Route guard wrapper that redirects unauthenticated users to the login page. Wraps all protected app routes in React Router. Shows a loading spinner while Auth0 resolves session state.

PROPS

Prop	Type	Description
children	ReactNode	Child route elements to render when authenticated

DEPENDENCIES

useAuth() → isAuthenticated, isLoading · react-router-dom Navigate

<Route element={<ProtectedRoute><PageLayout /></ProtectedRoute>}>

RoleGate
auth/RoleGate.jsx

IMPLEMENTED

Conditional render guard for RBAC-restricted UI blocks. Renders children only when the current user's role matches the `allowedRoles` list. Returns null or a fallback otherwise.

PROPS

Prop	Type	Description
allowedRoles	string[]	Roles permitted to see children (ORGANISER TUTOR STUDENT)
children	ReactNode	Content rendered when role check passes
fallback	ReactNode?	Optional element rendered when role check fails

DEPENDENCIES

useAuth() → role, hasRole()

<RoleGate allowedRoles={['organiser']}><AllocationBoard /></RoleGate>

PageLayout

layout/PageLayout.jsx

STUB – TODO

Master layout wrapper for all protected internal pages. Coordinates mobile sidebar state and composes Navbar, Sidebar, React Router `<Outlet />`, and Footer. Manages `lg:p1-64` offset so main content never overlaps the fixed sidebar.

COMPOSED OF

Navbar · Sidebar · `<Outlet />` (from react-router-dom) · Footer

INTERNAL STATE

`isSidebarOpen` (boolean) – mobile drawer toggle state

```
<Route element={<ProtectedRoute><PageLayout /></ProtectedRoute>}>
  <Route path="/dashboard" element={<DashboardPage />} />
</Route>
```

Navbar

layout/Navbar.jsx

STUB – TODO

Top navigation bar. Displays Toodle logo, Wits SDP badge, mobile hamburger toggle, and authenticated user section (avatar, display name, role badge, quick logout). Sticky header with translucent backdrop blur.

PROPS

Prop	Type	Description
<code>isSidebarOpen</code>	<code>boolean</code>	Current mobile sidebar drawer state
<code>onToggleSidebar</code>	<code>() => void</code>	Callback to toggle mobile sidebar

DEPENDENCIES

`useAuth()` · `Badge` · `LogoutButton` · `lucide-react` icons

Sidebar

layout/Sidebar.jsx

STUB – TODO

Side navigation drawer with role-filtered navigation links. Highlights active route via React Router `NavLink`. Responsive slide-in mobile drawer (<1024px) with backdrop overlay.

ROLE-BASED NAVIGATION LINKS

Role	Visible Links
<code>Organiser</code>	Dashboard · Allocation Board · Courses · Tutor Directory
<code>Tutor</code>	Dashboard · Courses · My Availability · Profile
<code>Student</code>	Dashboard · Courses

PROPS

Prop	Type	Description
<code>isOpen</code>	<code>boolean</code>	Mobile drawer open state
<code>onClose</code>	<code>() => void</code>	Callback to close mobile drawer

DEPENDENCIES

`useAuth()` · `react-router-dom` `NavLink` · `lucide-react`

Footer

layout/Footer.jsx

STUB – TODO

Standard application footer displaying Wits University copyright, SDP project attribution, and helpful links. Fixed or pinned at bottom of layout. Responsive mobile/desktop layout.

NO EXTERNAL PROPS

Badge

ui/Badge.jsx

STUB – TODO

Accessible status chip and pill component for communicating roles, course statuses, and constraint states. Supports an optional status dot indicator.

PROPS

Prop	Type	Description
variant	string	neutral primary success warning danger purple
size	string?	sm md lg (default: md)
dot	boolean?	Show filled status dot before label
children	ReactNode	Badge label content

```
<Badge variant="success" dot>Full Quota</Badge>
<Badge variant="danger">Timetable Clash</Badge>
```

Button

ui/Button.jsx

STUB – TODO

Accessible, reusable button primitive. Supports multiple visual variants, sizes, built-in loading state with an embedded Spinner, left/right Lucide icon slots, and forwards all HTML button attributes and refs.

PROPS

Prop	Type	Description
variant	string	primary secondary accent danger ghost outline
size	string?	sm md lg (default: md)
isLoading	boolean?	Shows embedded Spinner and disables button
leftIcon	ReactNode?	Lucide icon element shown left of label
rightIcon	ReactNode?	Lucide icon element shown right of label
children	ReactNode	Button label text

```
<Button variant="primary" leftIcon={<Plus />} onClick={handleCreate}>Add Course</Button>
```

Card / CardHeader / CardBody / CardFooter

ui/Card.jsx

STUB – TODO

Surface container primitive for content grouping. Provides consistent border, background, shadow, padding, and hover elevation. Composed of three subcomponents.

SUBCOMPONENTS

Subcomponent	Responsibilities
CardHeader	Title text, description, and optional action button slot (top-right)
CardBody	Main scrollable content container with padding
CardFooter	Bottom action row separated by a top divider

CARD PROPS

Prop	Type	Description
hover	boolean?	Enable elevated box-shadow on hover
className	string?	Additional class overrides
children	ReactNode	CardHeader / CardBody / CardFooter content

```
<Card hover>
  <CardHeader title="Course Info" action={<Button size="sm">Edit</Button>} />
  <CardBody>...</CardBody>
  <CardFooter>...</CardFooter>
</Card>
```

Modal / ModalFooter

ui/Modal.jsx

STUB – TODO

Accessible dialog modal using the native HTML <dialog> element. Handles backdrop click, Escape key cancel, focus trapping, and body scroll locking.

PROPS

Prop	Type	Description
isOpen	boolean	Controls dialog open/closed state
onClose	() => void	Called on backdrop click or Escape key
title	string	Dialog header title
children	ReactNode	Modal body content

```
<Modal isOpen={isOpen} onClose={() => setIsOpen(false)} title="Assign Tutor">
  <p>Modal body content</p>
  <ModalFooter><Button onClick={() => setIsOpen(false)}>Close</Button></ModalFooter>
</Modal>
```

Spinner

ui/Spinner.jsx

STUB – TODO

Accessible SVG/CSS loading indicator for async pending operations. Includes role="status" and hidden screen-reader label text.

PROPS

Prop	Type	Description
size	string?	sm (16px) md (24px) lg (32px) xl (48px)
className	string?	Color override, e.g. text-primary

```
<Spinner size="md" className="text-primary" />
```

useAuth()

hooks/useAuth.js

IMPLEMENTED

Central authentication and RBAC hook. Wraps `@auth0/auth0-react`, extracts the user role from the custom Auth0 namespace claim (`{VITE_AUTH0_AUDIENCE}/roles`), and provides dev-mode bypass via `localStorage`.

RETURNS

Key	Type	Description
user	<code>object null</code>	Authenticated user profile (devUser or Auth0 user)
role	<code>string null</code>	Active role: <code>organiser tutor student</code>
isAuthenticated	<code>boolean</code>	True if auth0 authenticated or devUser present
isLoading	<code>boolean</code>	True while Auth0 resolves initial session
error	<code>Error null</code>	Auth0 initialization or authentication error
isOrganiser	<code>boolean</code>	<code>role === 'organiser'</code>
isTutor	<code>boolean</code>	<code>role === 'tutor'</code>
isStudent	<code>boolean</code>	<code>role === 'student'</code>
login()	<code>function</code>	Triggers Auth0 <code>loginWithRedirect()</code>
logout()	<code>function</code>	Clears session and redirects to origin
getToken()	<code>async function</code>	Returns JWT Bearer access token string
hasRole(roles)	<code>function</code>	Checks if current role is in the provided array
loginAsDevRole(role)	<code>function</code>	Dev bypass: stores mock user in <code>localStorage</code>

DEV BYPASS STORAGE KEY

`toodle_dev_user` in `localStorage` — persists mock user across page reloads during local development.

useApi(apiFn, options?)

hooks/useApi.js

IMPLEMENTED

Standardized async data-fetching hook. Manages data, loading, and error state. Supports immediate-on-mount or deferred manual triggering. Prevents state updates on unmounted components via a mounted ref.

PARAMETERS

Param	Type	Description
apiFn	<code>async function</code>	Async API function to invoke (e.g. <code>coursesApi.getCourses</code>)
options.immediate	<code>boolean</code>	Execute on mount (default: <code>true</code>)
options.initialData	<code>any</code>	Initial data state value (default: <code>null</code>)
options.params	<code>any[]</code>	Arguments array passed to <code>apiFn</code> on auto-execute

RETURNS

Key	Type	Description
data	<code>any</code>	API response data
loading	<code>boolean</code>	True while request is in-flight
error	<code>string null</code>	Normalized error message from response or Error object
execute(...args)	<code>async function</code>	Manually trigger the API call with optional args
refetch	<code>async function</code>	Alias for <code>execute()</code> — re-runs the last call
setData	<code>function</code>	Directly update data state (optimistic updates)

ERROR NORMALIZATION

Resolves: `err?.response?.data?.message` → `err?.message` → `'An unexpected error occurred'`

```
const { data: courses, loading, error, refetch } = useApi(coursesApi.getCourses);
```

BASE URL: VITE_API_URL (DEFAULT: HTTP://LOCALHOST:3000/API/V1) · AUTH: BEARER TOKEN INJECTED BY SETUPAUTHINTERCEPTOR

usersApi

api/users.js — users & auth profile endpoints

IMPLEMENTED

Function	Method	Endpoint	Description
getCurrentUser()	GET	/auth/me	Fetch authenticated user's profile & DB role
syncUser(userData)	POST	/auth/callback	Upsert Auth0 user into PostgreSQL on login
deleteCurrentUser()	DELETE	/auth/me	Delete own account
getUsers(params?)	GET	/users	List all users (Organiser only)
getUser(id)	GET	/users/:id	Get specific user details
updateUser(id, data)	PATCH	/users/:id	Update user capacity / profile fields

coursesApi

api/courses.js — course & session endpoints

IMPLEMENTED

Function	Method	Endpoint	Description
getCourses(params?)	GET	/courses	List courses with staffing metadata
getCourse(id)	GET	/courses/:id	Single course with allocations & sessions
createCourse(data)	POST	/courses	Create course offering (Organiser)
updateCourse(id, data)	PATCH	/courses/:id	Update quotas / thresholds (Organiser)
deleteCourse(id)	DELETE	/courses/:id	Delete course (Organiser)
getCourseSessions(id)	GET	/courses/:id/sessions	List contact sessions for a course
createCourseSession(id, data)	POST	/courses/:id/sessions	Add a session slot
updateSession(id, data)	PATCH	/sessions/:id	Update session timing / venue
deleteSession(id)	DELETE	/sessions/:id	Remove session slot

tutorsApi

api/tutors.js — tutor directory, marks & availability

IMPLEMENTED

Function	Method	Endpoint	Description
getTutors(params?)	GET	/tutors	List tutors with marks & current hours
getTutor(id)	GET	/tutors/:id	Detailed tutor profile
addOrUpdateMark(id, data)	POST	/tutors/:id/marks	Add/update course mark (Organiser)
setAvailability(id, slots)	PUT	/tutors/:id/availability	Set weekly availability (Tutor self-service). Body: '{"slots: [...]}"

allocationsApi

api/allocations.js — allocation board & constraint validation

IMPLEMENTED

Function	Method	Endpoint	Description
getAllocations(params?)	GET	/allocations	List allocations (filterable by course/tutor)
createAllocation(data)	POST	/allocations	Create assignment; runs server constraint validation
updateAllocation(id, data)	PATCH	/allocations/:id	Update hours or status
deleteAllocation(id)	DELETE	/allocations/:id	Remove tutor from course
validateAllocation(params)	GET	/allocations/validate	Dry-run validation before saving (no DB write)

Component	Route	Status	Lines	Roles	Description
LandingPage	/	IMPL	314	Public	Public marketing page. Hero section, feature cards, login CTA. Renders LoginButton for unauthenticated users.
LoginPage	/login	STUB	18	Public	Dedicated login page. Triggers Auth0 Universal Login redirect.
DashboardPage	/dashboard	IMPL	173	All roles	Role-adaptive home screen. Organiser: staffing stats + recent allocations. Tutor: own assignments + hours. Student: enrolled courses.
CoursesPage	/courses	IMPL	149	All roles	Course catalogue list with search/filter. Organiser sees create/delete controls. Uses coursesApi + useApi.
CourseDetailPage	/courses/:id	STUB	19	All roles	Single course detail view with sessions, allocations, and tutor list.
TutorsPage	/tutors	IMPL	200	Organiser	Tutor directory with search. Displays marks, hours, availability summary. Organiser-only via RoleGate.
AllocationBoardPage	/allocations	STUB	23	Organiser	Drag-and-drop allocation board (@dnd-kit). Tutors column ↔ Courses column. Live constraint validation badges.
SessionSwapPage	/swap	PARTIAL	559	Tutor	Largest page. Session swap request workflow — browse available swap slots, submit requests, view pending/approved swaps.
TimesheetsPage	/timesheets	PARTIAL	174	Tutor Organiser	Timesheet submission (Tutor) and approval/export (Organiser). Weekly hours grid view.
ReportsPage	/reports	PARTIAL	200	Organiser	Staffing reports dashboard. Allocation coverage metrics, understaffed courses, hours summary.
VolunteersPage	/volunteers	IMPL	169	Organiser	Student volunteer applications list. Review and promote students to tutors.
ProfilePage	/profile	STUB	21	All roles	User profile view/edit. Tutor: set availability slots. Student: view enrolled courses. Account deletion.
NotFoundPage	*	IMPL	15	All	404 fallback page with home navigation link.

helpers.js

utils/helpers.js

IMPLEMENTED

Function	Signature	Description
<code>cn()</code>	<code>cn(...inputs) → string</code>	Merges TailwindCSS classes with <code>clsx</code> + <code>tailwind-merge</code> (conflict resolution).
<code>formatTime()</code>	<code>formatTime(timeStr) → string</code>	Converts 24h HH:MM string (e.g. "14:15:00") → 12h AM/PM (e.g. "2:15 PM").
<code>formatDay()</code>	<code>formatDay(day) → string</code>	Title-cases weekday enum (e.g. "MONDAY" → "Monday").
<code>getRoleBadgeStyle()</code>	<code>getRoleBadgeStyle(role) → string</code>	Returns TailwindCSS bg+border+text class string per role. Organiser → blue, Tutor → emerald, Student → purple.
<code>getConstraintBadgeStyle()</code>	<code>getConstraintBadgeStyle(severity) → string</code>	Returns badge classes per severity: error → rose, warning → amber, info → sky.

constants.js

utils/constants.js

IMPLEMENTED

Constant	Value	Description
<code>AUTH0_NAMESPACE</code>	<code>{"{VITE_AUTH0_AUDIENCE}"/roles</code>	Namespace key for reading role from Auth0 JWT custom claims. Default audience: <code>https://toodle-api</code> .
<code>ROLES</code>	<code>ORGANISER TUTOR STUDENT</code>	Lowercase role string enum used throughout the app for comparison.
<code>ALLOCATION_STATUS</code>	<code>ACTIVE PENDING REMOVED</code>	Matches Prisma AllocationStatus enum.
<code>SESSION_TYPES</code>	<code>LECTURE TUTORIAL LAB</code>	Matches Prisma SessionType enum.
<code>DAYS_OF_WEEK</code>	<code>MONDAY ... FRIDAY</code>	Ordered weekday array for availability matrix and session scheduling.
<code>CONSTRAINT_TYPES</code>	<code>MARK_BELOW_THRESHOLD TIMETABLE_CLASH HOURS_EXCEEDED</code>	Allocation constraint violation type identifiers used in validation responses.