Frontend Documentation

Table of Contents

[**1.** **Project Overview (Frontend)** 3](#_Toc195138154)

[Design Elements 3](#_Toc195138155)

[Core Technologies 3](#_Toc195138156)

[1.1 Next.js (App Router) 3](#_Toc195138157)

[1.2 TypeScript 3](#_Toc195138158)

[1.3 Tailwind CSS 3](#_Toc195138159)

[1.4 shadcn/ui 3](#_Toc195138160)

[1.5 Tiptap Text Editor 3](#_Toc195138161)

[2. Style Guide 3](#_Toc195138162)

[2.1 Visual Consistency 3](#_Toc195138163)

[2.2 Design Rationale 4](#_Toc195138164)

[3. Folder Structure and Code Organisation 4](#_Toc195138165)

[Key Directories 4](#_Toc195138166)

[4. Key Components 4](#_Toc195138167)

[4.1 DatasetsPage (in components/all\_datasets or app/datasets/all) 4](#_Toc195138168)

[4.2 DatasetCard (in components/all\_datasets/datasetCard.tsx) 5](#_Toc195138169)

[4.3 auth\_guard and withAccessControl 5](#_Toc195138170)

[4.4 Chat Components (e.g. UserChatListPage, ChatPage) 5](#_Toc195138171)

[4.5 Dashboards (e.g. admin.tsx, researcher.tsx) 5](#_Toc195138172)

[4.6 Landing Pages and stat Components 5](#_Toc195138173)

[4.7 Requests, PayPal and Organisation Directories 5](#_Toc195138174)

[5. API Documentation 5](#_Toc195138175)

[fetchWithAuth: 5](#_Toc195138176)

[Endpoints: 5](#_Toc195138177)

[Error Handling: 6](#_Toc195138178)

[6. Code Documentation 6](#_Toc195138179)

[Naming Conventions: 6](#_Toc195138180)

[Dependencies: 6](#_Toc195138181)

# **Project Overview (Frontend)**

## Design Elements

* Uses Tailwind CSS and shadcn/ui to maintain consistency and a modern aesthetic.
* The layout emphasises clarity for data exploration grid and list views, dashboards and real-time interactions such as chats and notifications.

## Core Technologies

### 1.1 Next.js (App Router)

* Route-based file structure
* Data fetching:
  + Uses fetchWithAuth to include JWT tokens in request headers.

### 1.2 TypeScript

* Ensures type safety.

### 1.3 Tailwind CSS

* Utility-first approach: classes like bg-gray-card, text-gray-600, hover:bg-primary.
* globals.css and tailwind.config.js define colour palette using shadcn’s theme extensions.

### 1.4 shadcn/ui

* Provides pre-built UI components (alerts, dropdowns, skeletons).
* Helps unify styles with Tailwind.

### 1.5 Tiptap Text Editor

* Used for rich text research submissions, storing formatting data for analysis or final reviews.
* Extensions: bold, italic, headings, bullet lists, code blocks.
* TextEditorToolbar: a custom toolbar that toggles formatting.

## 2. Style Guide

### 2.1 Visual Consistency

* Colour Palette: Defined in globals.css and tailwind.config.js. Key variables:
  + bg-gray-card, text-gray-600 and bg-primary.
  + Shade of orange #f97316 used for highlights and calls-to-action.
  + Shade of red and yellow is defined in the tailwind config file as this was not included in the shadcn-ui colour themes.
* Fonts and Spacing:
  + Tailwind utility classes ensure standard spacing.

### 2.2 Design Rationale

* Utility-first: Tailwind classes speed development and reduce custom CSS, enabling consistent spacing, text sizing, and responsiveness.
* shadcn/ui: Provides consistent styling for alerts, dropdowns, dialogs, and other UI patterns.

## 3. Folder Structure and Code Organisation

### Key Directories

1. app/
   * Next.js Route Segments using the App Router convention.
   * Subfolders correspond to route namespaces (e.g. requests, organisation, researcher).
   * Contains route-specific files (e.g. page.tsx, layout.tsx, loading.tsx) and a central globals.css.
2. components/
   * Contains reusable UI.
   * Subfolders like analyze group related logic.
   * ui/ folder integrates shadcn/ui styling for dialogues, dropdowns, forms and contains frequently used components.
3. auth\_guard/, dashboards/, requests/
   * AuthGuard: Higher-order components and hooks that guard routes based on user roles.
   * Dashboards: Contains Admin and Researcher dashboards.
   * Requests: Manages request flows to access datasets, including listing, approving, or revoking.

## 4. Key Components

### 4.1 DatasetsPage (in components/all\_datasets or app/datasets/all)

* Fetches dataset list from the backend with fetchWithAuth.
* Search, filtering, pagination logic with useMemo for performance.
* Renders either a grid or list layout.

### 4.2 DatasetCard (in components/all\_datasets/datasetCard.tsx)

* Displays each dataset’s title, description, tags, and price.
* Supports a bookmark logic when user is authenticated.

### 4.3 auth\_guard and withAccessControl

* Role-based route protection.
* Minimises code duplication by wrapping entire page components in a guard HOC.

### 4.4 Chat Components (e.g. UserChatListPage, ChatPage)

* Manages WebSocket connections for real-time chat.
* Tracks unread messages, typing and conversation lists.
* useEffect sets up the socket, handles onmessage, onclose.

### 4.5 Dashboards (e.g. admin.tsx, researcher.tsx)

* Admin: shows total datasets, pending requests and analytics.
* Researcher: displays user’s datasets, submissions and approvals.
* Condition based rendering via user role.

### 4.6 Landing Pages and stat Components

* Introduce new or unregistered users to the platform.
* Includes dynamic charts for usage stats, e.g. number of active users or newly uploaded datasets.

### 4.7 Requests, PayPal and Organisation Directories

* requests/: Handles user requests for dataset access, listing them in AllRequests.tsx, and approving in ApprovalRequest.tsx.
* paypal/: Payment integration flows; includes success logic from PayPal.
* organisation/: Contains admin detail routes and sign-up flows for organisations.

## 5. API Documentation

### fetchWithAuth:

* + Wraps fetch, automatically attaching an Authorization header with the user’s token.

### Endpoints:

* + GET /datasets/all – retrieve all datasets.
  + GET /datasets/detail/:id – get details for a single dataset.
  + POST /datasets/perform/:id – run statistical analysis on a dataset.
  + POST /requests/requestaction/:request\_id/ – Approve/Reject data access.
  + POST /research/submissions/pending/ – Submitting new research analysis.

### Error Handling:

* + If response.ok is false, code checks response.status and logs or displays errors in UI components.

## 6. Code Documentation

## Naming Conventions:

* + Components mostly use PascalCase.
  + Hooks use camelCase with use prefix.
  + Constants like ITEMS\_PER\_PAGE are UPPER\_CASE.

## Dependencies:

#### Core Framework and Language:

* next ^15.1.6 – React framework supporting SSR/SSG and routing.
* react ^19.1.0 – Core library for building user interfaces.
* react-dom ^19.1.0 – DOM-specific methods for React.
* typescript ^5.7.3 – Typed superset of JavaScript used for type safety.

#### Styling and Theming:

* tailwindcss ^3.4.1 – Utility-first CSS framework for rapid UI development.
* @tailwindcss/typography ^0.5.16 – Tailwind plugin for rich text content.
* tailwindcss-animate ^1.0.7 – Adds pre-built animations to Tailwind.
* tailwind-merge ^3.0.2 – Merges conflicting Tailwind utility classes.
* clsx ^2.1.1 – Conditionally joins classNames.

#### UI Component Libraries:

* @shadcn/ui ^0.0.4 – Component generator using Tailwind and Radix primitives.
* lucide-react ^0.474.0 – Icon library used alongside ShadCN UI.
* @radix-ui/react-alert-dialog ^1.1.6 – Accessible alert dialogs.
* @radix-ui/react-dropdown-menu ^2.1.6 – Dropdown menu primitives.
* @radix-ui/react-label ^2.1.2 – Form label primitives.
* @radix-ui/react-select ^2.1.6 – Custom select menus.
* @radix-ui/react-separator ^1.1.2 – Visual separators for layout.
* @radix-ui/react-slot ^1.1.1 – Composable slot component.
* @radix-ui/react-switch ^1.1.3 – Toggle switch primitive.

#### State Management and Data Fetching:

* @tanstack/react-query ^5.67.1 – Asynchronous data fetching and caching.
* swr ^2.3.3 – Lightweight data fetching and caching library.

#### Rich Text Editing

Used for implementing a rich text editor in research submissions.

* @tiptap/react ^2.11.5 – React bindings for the Tiptap editor.
* @tiptap/starter-kit ^2.11.5 – Preconfigured set of common extensions.
* @tiptap/pm ^2.11.5 – ProseMirror bindings.
* @tiptap/extension-\* ^2.11.5 – Full suite of 19+ extensions including:
  + bold, italic, underline, strike, highlight, text-style, blockquote, heading, bullet-list, ordered-list, list-item, code-block, image, link, colour, character-count, table, table-row, table-header, table-cell, text-align

#### Charts and Data Visualisation:

Used in dashboards and landing pages to visualise metrics and trends.

* apexcharts ^4.4.0 – Core charting library.
* react-apexcharts ^1.7.0 – React wrapper for ApexCharts.
* chart.js ^4.4.8 – Flexible charting library.
* react-chartjs-2 ^5.3.0 – React bindings for Chart.js.

#### Parsing, Markdown and CSV Export:

* html-react-parser ^5.2.2 – Safely parses HTML strings into React components.
* react-markdown ^10.0.0 – Markdown-to-React renderer.
* turndown ^7.2.0 – Converts HTML to Markdown.
* react-csv ^2.2.2 – CSV export functionality for research data.

#### Payment and Transactions

* @paypal/react-paypal-js ^8.8.2 – Integration with PayPal for dataset payments.

#### Utility Libraries

* lodash ^4.17.21 – JavaScript utility functions.
* date-fns ^4.1.0 – Date formatting and manipulation.
* jwt-decode ^4.0.0 – JWT parsing for authentication.
* swiper ^11.2.5 – Carousel and slider components.
* bleach ^0.3.0 – Input sanitisation