**IGameCatalogService.cs**Game catalog contract. Supports paged search and fetching game details.

**IDealsService.cs**Deals contract. Supports fetching deals by game title and a list of top deals.

**BaseHttpService.cs**Centralizes HTTP calls and converts network failures into a single exception for consistent handling. Services inherit it to share logging and no-network behaviour.

**RawgService.cs**Implements the catalog. Sends RAWG requests with Metacritic ordering, caches paged results and returns normalized models. Fetches game details and screenshots separately and strips HTML from the description.

**CheapSharkService.cs**Implements deals. Resolves a game by title, loads its deals, and sorts by savings and price; builds top deals via a separate request.

**LocalStorage.cs**Provides a simple key-value store in the browser via JS interop with JSON serialization. Used for persistence.

**AppState.cs**Holds shared state. Stores the backlog in the browser plus the latest search parameters and results with pagination, and supports lazy one-time load, toggle, presence check, and save.

**Program.cs**Configures DI, Blazor render modes, typed HttpClient with base addresses, and component routing.

**AppErrorBoundary.cs**Catches component-level exceptions and redirects to 404 if the failure is network related.

**Routes.razor**Defines the root router, wires focus management and wraps navigation with the global error boundary.

**Home.razor**Shows the search box, then renders results, writes them to AppState, and includes pagination.

**GameDetails.razor**Loads game details and screenshots and shows related deals for the title.

**Deal.razor**Displays a list of top deals with links to stores.

**Backlog.razor**After first render, loads the backlog from local storage and lets the user manage saved games.

**SearchBar.razor**Encapsulates the input and search events and passes them to the parent component.

**Pager.razor**Controls page navigation and returns focus to the heading.

**GameCard.razor**Shows a compact game summary with add or remove actions for the backlog.

**DealCard.razor**Renders a single deal with price, savings, store, and a direct link.

**NotFound.razor**Displays a 404 state and suggests returning to Home.