LoginPage

Renders the complete login view with form and gym illustration.

Props:

States:

Behavior:

- Acts as the container for the login interface.
- Organizes layout into form and illustration sections.

Composition:

- **Parent:** App
- **Children:** Header, AvatarIcon, LoginForm, GymIllustration

Variations:

ProfileIcon

Displays the admin's profile avatar.

Props:

- `userImage: {string} – URL of the admin avatar (optional, default: placeholder image)`

States:

Behavior:

- Displays the admin’s avatar.

Composition:

- **Parent:** LoginPage
- **Children:** None

Variations:

- Default img if no image is provided.

LoginForm

Contains login inputs and action buttons.

Props:

States:

- `adminLogin: string`
- `password: string`

Behavior:

- Manages user input.
- Validates input when the login button is clicked.
- Could trigger API call for login.

Composition:

- **Parent:** LoginPage
- **Children:** InputField (x2), LoginButton, HelpIcon

Variations:

InputField

Reusable input field component.

Props:

- `placeholder: {string}`
- `type: {string} - e.g., text/password`
- `onChange: {function}`

States:

Behavior:

- Renders an input box.
- Captures user typing and triggers the onChange callback.

Composition:

- **Parent:** LoginForm
- **Children:** None

Variations:

- Type “text”, “password”, “email”.
- With or without a visibility toggle for passwords.

LoginButton

Submits the login form.

Props:

- `onClick: {function}`
- `text: {string} - Defaults to "Log In"`

States:

Behavior:

- Triggers form submission logic.
- Can show a spinner or loading state on click.

Composition:

- **Parent:** LoginForm
- **Children:** None

Variations:

- Could be primary, secondary or loading state.

AdminManePage

Renders the admin interface with a user list and statistics.

Props:

- user: {object} - User data (avatar, name) (required)
- statistics: {array} - Data for charts (optional)
- categories: {array} - List of workout categories (title, image) (required)
- onProfileClick: {function} - Navigates to "Profile Page" (required)
- onCategorySelect: {function} - Navigates to "Start Training" (required)
- onStatisticsClick: {function} - Navigates to "Statistics Page" (required)

States:

Behavior:

- Clicking on the profile triggers `onProfileClick()`.
- Clicking on a category triggers `onCategorySelect(category)`.
- Clicking on "See Full Statistics" triggers `onStatisticsClick()`.

Composition:

- Parent: App
- Children: ProfileIcon, StatisticsChart, CategoryGrid

Variations:

- Hides statistics if data is unavailable.

User Sidebar

A sidebar displaying the list of users with search and filter functionality.

Props:

- `users` (Array) — List of users
- `selectedUser` (Object) — Currently selected user
- `onSelectUser` (Function) — Callback when a user is selected
- `filterOptions` (Array) — List of filter options
- `onFilterChange` (Function) — Callback when filter changes

States:

- `searchQuery` (string) — Search input value
- `activeFilter` (string | null) — Selected filter

Behavior:

- Search users by name
- Filter users by type (Premium, VIP, Standard, Admin, User)
- Display avatar, username, and status badge
- Select user to load detailed statistics

Composition:

- **Parent:** AdminMainPage
- **Children:** UserListItem, FilterDropdown

Variations:

- With filter dropdown (default)

AddUserButton

Adds a new user to the system.

Props:

- onClick: {function} - Handler to add a new user (required)

States:

Behavior:

- Calls `onClick()` when clicked.
- Opens a modal or form to add a new user.

Composition:

- Parent: AdminMainPage
- Children: None

Variations:

SearchBar

Enables filtering the user list.

Props:

- `onSearch: function`
- `placeholder: string`

States:

- `searchQuery: string`

Behavior:

- Filters the list based on search input.

Composition:

- Parent: AdminMainPage
- Children: None

Variations:

- Could include filter tags or dropdowns for advanced search.

UserList

Lists user cards.

Props:

- `users: array`
- `onUserSelect: function`

States:

Behavior:

- Displays the list of users.
- Handles user selection.

Composition:

- Parent: AdminMainPage
- Children: UserListItem (repeated)

Variations:

- Can highlight active/selected user.

UserListItem

Represents a single user.

Props:

- `user: object`
- `onClick: function`

States:

Behavior:

- Shows user details.
- Calls `onClick` when selected.

Composition:

- **Parent:** UserList

  Children: None

Variations:

StatisticsChart

A preview chart displaying statistics over years, months or weeks.

Props:

- `data` (Array) — Dataset for the chart
- `title` (string) — Chart title
- `onExpand` (Function) — Callback to expand full statistics view

States:

Behavior:

- Displays a statistics chart.

Composition:

- Parent: AdminMainPage
- Children: None

Variations:

- Supports different chart types (bar, line, etc.).

SeeStatisticsButton

Navigates to the full statistics page.

Props:

- onClick: {function} - Navigation handler (required)

States:

Behavior:

- Calls `onClick()` when clicked.

Composition:

- Parent: AdminMainPage
- Children: None

Variations:

NavigationArrows

A component for navigating between different weeks or months.

Props:

- `onPrevious: {function} – function triggered when clicking the left arrow (required)`
- `onNext: {function} – function triggered when clicking the right arrow (required)`

States:

Behavior:

- Clicking the left arrow calls onPrevious(), navigating to the previous period.
- Clicking the right arrow calls onNext(), navigating to the next period.

Composition:

- Parent: StatisticsGraph
- Children: None

Variations:

Scroll

A component that enables smooth scrolling between multiple charts or content sections.

Props:

- `children`: {ReactNode} - Chart or content elements to scroll through (required)
- `direction`: {"horizontal" | "vertical"} - Sets scroll direction (optional, default: `"horizontal"`)
- `onScrollEnd`: {function} - Callback triggered when the user finishes scrolling (optional)

States:

- `isScrolling`: boolean - Tracks if the user is currently scrolling
- `scrollPosition`: number - Stores current scroll position (optional, if needed)

Behavior:

- Provides scrollable area for child charts or sections
- Supports horizontal or vertical scroll based on the `direction` prop
- Optional snap behavior for smooth chart-to-chart scroll
- Triggers `onScrollEnd()` after scroll finishes (if provided)

Composition:

- Parent: StatisticsGraph
- Children: None

Variations:

UserDetailsCard

Displays detailed information about the selected user, with profile info, workout categories, and statistics.

Props:

- `user`: {object} — User data (avatar, name) (required)
- `statistics`: {array} — Data for user-specific charts (optional)
- `categories`: {array} — List of workout categories (title, image) (required)
- `onProfileClick`: {function} — Navigates to Profile Page (required)
- `onCategorySelect`: {function} — Navigates to Start Training with selected category (required)
- `onStatisticsClick`: {function} — Navigates to Statistics Page (required)

States:

- `selectedCategory: {object|null} – Stores the currently selected workout category`
- `showStatistics`: {boolean} — Toggles the display of the statistics preview
- `isLoading: {boolean} – Indicates if data is loading (for user info or stats)`
- `error: {string|null} – Stores any error message (e.g., failed data fetch)`

Behavios:

- Renders the user’s avatar and basic information
- Displays workout categories as selectable cards/buttons with images
- Optionally visualizes user-related statistics if the `statistics` prop is provided
- Clicking the avatar or name triggers `onProfileClick()` and navigates to the profile
- Clicking on a workout category triggers `onCategorySelect(category)` and navigates to the training session
- Includes a button or icon to view detailed statistics, triggering `onStatisticsClick()`

Composition:

- **Parent:** AdminMainPage
- **Children:** AvatarComponent, UserInfoBlock, CategoryCard (mapped from `categories`), StatisticsPreview (optional), ActionButtons (Profile, Statistics)

Variations:

- Could switch between read-only and editable mode

UserProfileForm

Allows editing user data.

Props:

- user: {object} - User data (required)
- onChange: {function} - Updates user data (required)

States:

- `formData: {object} - Local form state`

Behavios:

- Allows editing name, email, and subscription.
- Sends updated data to `onChange`.

Composition:

- **Parent:** AdminMainPage
- **Children:** InputField (x3), SubscriptionDropdown

Variations:

InputField

Input field for user data.

Props:

- value: {string} - Current value (required)
- onChange: {function} - Updates the data (required)
- type: {string} - Field type (text, email)

States:

Behavios:

- Captures user input and passes it to `onChange`.

Composition:

- **Parent:** UserProfileForm
- **Children:** None

Variations:

SubscriptionDropdown

Allows changing the user's subscription type.

Props:

- value: {string} - Current subscription type (required)
- onChange: {function} - Updates the subscription type (required)

States:

Behavios:

- Allows selecting a new subscription type.

Composition:

- **Parent:** UserProfileForm
- **Children:** None

Variations:

EditButton

Enables editing mode.

Props:

- onClick: {function} - Activates editing mode (required)

States:

Behavios:

- Sets `isEditing` to `true`.

Composition:

- **Parent:** UserProfileForm
- **Children:** None

Variations:

SaveButton

Saves user changes.

Props:

- onClick: {function} - Saves updated user data (required)

States:

Behavios:

- Calls `onSaveUser(updatedUser)`.

Composition:

- **Parent:** UserProfileForm
- **Children:** None

Variations:

DeleteButton

Deletes a user.

Props:

- onClick: {function} - Deletes the user (required)

States:

Behavios:

- Calls `onDeleteUser(user)`.

Composition:

- **Parent:** UserProfileForm
- **Children:** None

Variations:

AddUser/AdminPage

Renders a form to create a new user or admin account.

Props:

- `onCreate`: {function} — Handles the create action (required)
- `mode`: {string} — `'user'` or `'admin'` to switch form logic (required)
- `defaultValues`: {object} — Pre-filled form values (optional)

States:

- `name`: {string} — User's first name input
- `surname`: {string} — User's last name input
- `email`: {string} — Email input
- `phone`: {string} — Phone number input
- `id`: {string} — User ID (can be auto-generated or input manually)
- `address`: {string} — Address input
- `city`: {string} — City input
- `subType`: {string} — User subscription type (`"standard" | "vip" | "premium"`)
- `status`: {string} — Role of the user (`"user"` or `"admin"`)
- `password`: {string} — Admin password input (only visible in admin mode)
- `isLoading`: {boolean} — Loading state when creating user/admin
- `error`: {string|null} — Validation or creation error messages

Behavior:

- Dynamically renders additional Password field if `mode === 'admin'`
- Dropdowns control subType and status selection
- Executes `onCreate` on "Create" button click with form data
- Displays loading or error feedback when necessary

Composition:

- **Parent:** App / AdminPanel / Route (`/add-user`, `/add-admin`)
- **Children:** AvatarIcon (Optional) — Default user picture, InputFields — Name, Surname, Email, Phone, etc., Dropdowns — Sub Type, Status, PasswordField — Only for admin mode, CreateButton

Variations:

- Error highlights and success states

ProfileForm

Main form for creating a new user or admin profile. Includes uploading a profile picture, filling user details and choosing role/subscription type.

Props:

- `mode` ("user" | "admin") — determines the form mode
- `onSubmit` (Function) — callback to create a new profile
- `onUploadPhoto` (Function) — callback for profile photo upload

States:

Behavior:

- Expands the dropdown to choose between `User` or `Admin`
- If "Admin" is selected → switch to Admin Registration Form (password input appears)
- Upload and preview profile photo
- On Create click → profile is created with all the entered data

Composition:

- **Parent:** AdminMainPage / AddUserPage
- **Children:** InputField, Dropdown, ProfilePhotoUpload

Variations:

- User Creation Form (default)
- With Profile Photo Upload (default)
- Without Photo Upload (minimal mode)

InputField

Reusable component for form input fields.

Props:

- `label` (string)
- `placeholder` (string)
- `value` (string)
- `onChange` (function)
- `type` (text, number, email, password, etc.)

States:

Behavior:

- Allows the admin to edit name, email, phone, address, city, membership type, and password.
- Calls `onChange(updatedField, value)` when the user types in an input field.

Composition:

- **Parent:** AddUserPage
- **Children:** None

Variations:

- Text input (default)
- Password input
- Email input

ProfileIcon

Displays the user’s profile avatar.

Props:

- `userImage: {string} – URL of the user avatar (optional, default: placeholder image)`

States:

Behavior:

- Displays the user's/admin's avatar.

Composition:

- **Parent:** AddUserPage
- **Children:** None

Variations:

- Default img if no image is provided.

Button

A reusable button component.

Props:

- `label: {string} – button text (required)`
- `onClick: {function} – function triggered when the button is clicked (required)`
- `variant: {string} – button style (primary, secondary, danger, etc.) (optional, default: "primary")`

States:

Behavior:

- Calls onClick() when clicked.

Composition:

- **Parent:** AddUserPage
- **Children:** None

Variations:

- Primary, secondary, disabled states.