

LoginPage
Renders the complete login view with form and gym illustration.
Props:
States:
Behavior: <ul style="list-style-type: none">Acts as the container for the login interface.Organizes layout into form and illustration sections.
Composition: <ul style="list-style-type: none">Parent: AppChildren: Header, AvatarIcon, LoginForm, GymIllustration
Variations:

ProfileIcon
Displays the admin's profile avatar.
Props: <ul style="list-style-type: none">userImage: {string} - URL of the admin avatar (optional, default: placeholder image)
States:
Behavior: <ul style="list-style-type: none">Displays the admin's avatar.
Composition: <ul style="list-style-type: none">Parent: LoginPageChildren: None
Variations: <ul style="list-style-type: none">Default img if no image is provided.

LoginForm
Contains login inputs and action buttons.
Props:
States: <ul style="list-style-type: none">adminLogin: stringpassword: string
Behavior: <ul style="list-style-type: none">Manages user input.Validates input when the login button is clicked.Could trigger API call for login.
Composition:

AdminMainPage
Renders the admin interface with sidebar and main content area.
Props: <ul style="list-style-type: none">user: {object} - User data (avatar, name, email)statistics: {array} - Data for charts (options, premium, vip, status)categories: {array} - List of workout categoriesonProfileClick: {function} - Navigates to profile pageonCategorySelect: {function} - Navigates to category pageonStatisticsClick: {function} - Navigates to statistics page
States:
Behavior: <ul style="list-style-type: none">Clicking on the profile triggers onProfileClickClicking on a category triggers onCategorySelectClicking on "See Full Statistics" triggers onStatisticsClick
Composition: <ul style="list-style-type: none">Parent: AppChildren: ProfileIcon, StatisticsChart, CategoryList, UserList
Variations: <ul style="list-style-type: none">Hides statistics if data is unavailable.

UserSidebar
A sidebar displaying the list of users.
Props: <ul style="list-style-type: none">users (Array) — List of usersselectedUser (Object) — Currently selected useronSelectUser (Function) — Callback for selecting a userfilterOptions (Array) — List of filter optionsonFilterChange (Function) — Callback for filter change
States: <ul style="list-style-type: none">searchQuery (string) — Search inputactiveFilter (string null) — Selected filter
Behavior: <ul style="list-style-type: none">Search users by nameFilter users by type (Premium, VIP, Status)Display avatar, username, and statusSelect user to load detailed statistics
Composition: <ul style="list-style-type: none">Parent: AdminMainPageChildren: UserListItem, FilterDropdown
Variations: <ul style="list-style-type: none">With filter dropdown (default)

ProfilePage
with a user list and statistics.
name) (required)
additional)
categories (title, image) (required)
links to "Profile Page" (required)
links to "Start Training" (required)
links to "Statistics Page" (required)
FileClick().
categorySelect(category).
onClick onStatisticsClick().
CategoryGrid
debar
with search and filter functionality.
selected user
onClick when a user is selected
filter options
onClick when filter changes
initial value
filtered filter
standard, Admin, User)
badge
when

StatisticsChart
A preview chart displaying statistics over years, months or weeks.
Props: <ul style="list-style-type: none">data (Array) — Dataset for the charttitle (string) — Chart titleonExpand (Function) — Callback to expand full statistics view
States:
Behavior: <ul style="list-style-type: none">Displays a statistics chart.
Composition: <ul style="list-style-type: none">Parent: AdminMainPageChildren: None
Variations: <ul style="list-style-type: none">Supports different chart types (bar, line, etc.).
SeeStatisticsButton
Navigates to the full statistics page.
Props: <ul style="list-style-type: none">onClick: {function} - Navigation handler (required)
States:
Behavior: <ul style="list-style-type: none">Calls onClick() when clicked.
Composition: <ul style="list-style-type: none">Parent: AdminMainPageChildren: None
Variations:
NavigationArrows
A component for navigating between different weeks or months.
Props : <ul style="list-style-type: none">onPrevious: {function} - function triggered when clicking the left arrow (required)onNext: {function} - function triggered when clicking the right arrow (required)
States:
Behavior: <ul style="list-style-type: none">Clicking the left arrow calls onPrevious(), navigating to the previous period.Clicking the right arrow calls onNext(), navigating to the next period.

Prop	
State	
Beha	
Com	
Vari	
Prop	
State	
Beha	
Com	

UserDetailsCard
Displays detailed information about the selected user, with profile info, workout categories, and statistics.
JS: <ul style="list-style-type: none">• <code>user: {object}</code> — User data (avatar, name) (required)• <code>statistics: {array}</code> — Data for user-specific charts (optional)• <code>categories: {array}</code> — List of workout categories (title, image) (required)• <code>onProfileClick: {function}</code> — Navigates to Profile Page (required)• <code>onCategorySelect: {function}</code> — Navigates to Start Training with selected category (required)• <code>onStatisticsClick: {function}</code> — Navigates to Statistics Page (required)
ES: <ul style="list-style-type: none">• <code>selectedCategory: {object null}</code> – Stores the currently selected workout category• <code>showStatistics: {boolean}</code> — Toggles the display of the statistics preview• <code>isLoading: {boolean}</code> – Indicates if data is loading (for user info or stats)• <code>error: {string null}</code> – Stores any error message (e.g., failed data fetch)
avios: <ul style="list-style-type: none">• Renders the user's avatar and basic information• Displays workout categories as selectable cards/buttons with images• Optionally visualizes user-related statistics if the <code>statistics</code> prop is provided• Clicking the avatar or name triggers <code>onProfileClick()</code> and navigates to the profile• Clicking on a workout category triggers <code>onCategorySelect(category)</code> and navigates to the training session• Includes a button or icon to view detailed statistics, triggering <code>onStatisticsClick()</code>
iposition: <ul style="list-style-type: none">• Parent: AdminMainPage• Children: AvatarComponent, UserInfoBlock, CategoryCard (mapped from <code>categories</code>), StatisticsPreview (optional), ActionButtons (Profile, Statistics)
ations: <ul style="list-style-type: none">• Could switch between read-only and editable mode

UserProfileForm
Allows editing user data.
JS: <ul style="list-style-type: none">• <code>user: {object}</code> - User data (required)• <code>onChange: {function}</code> - Updates user data (required)
ES: <ul style="list-style-type: none">• <code>formData: {object}</code> - Local form state
avios: <ul style="list-style-type: none">• Allows editing name, email, and subscription.• Sends updated data to <code>onChange</code>.
iposition: <ul style="list-style-type: none">• Parent: AdminMainPage• Children: InputField (x3), SubscriptionDropdown

AddUser/AdminPage
Renders a form to create a new user or admin.
Props: <ul style="list-style-type: none">• <code>onCreate: {function}</code> — Handles the create action (required)• <code>mode: {string}</code> — 'user' or 'admin' to switch form layout• <code>defaultValues: {object}</code> — Pre-filled form values (optional)
States: <ul style="list-style-type: none">• <code>name: {string}</code> — User's first name input• <code>surname: {string}</code> — User's last name input• <code>email: {string}</code> — Email input• <code>phone: {string}</code> — Phone number input• <code>id: {string}</code> — User ID (can be auto-generated or input)• <code>address: {string}</code> — Address input• <code>city: {string}</code> — City input• <code>subType: {string}</code> — User subscription type ("standard", "premium")• <code>status: {string}</code> — Role of the user ("user" or "admin")• <code>password: {string}</code> — Admin password input (only visible in admin mode)• <code>isLoading: {boolean}</code> — Loading state when creating• <code>error: {string null}</code> — Validation or creation error message
Behavior: <ul style="list-style-type: none">• Dynamically renders additional Password field if <code>mode</code> is 'admin'• Dropdowns control <code>subType</code> and status selection• Executes <code>onCreate</code> on "Create" button click with form data• Displays loading or error feedback when necessary
Composition: <ul style="list-style-type: none">• Parent: App / AdminPanel / Route (/add-user, /add-admin)• Children: AvatarIcon (Optional) — Default user picture, Surname, Email, Phone, etc., Dropdowns — Sub Type — Only for admin mode, CreateButton
Variations: <ul style="list-style-type: none">• Error highlights and success states

ProfileForm
Main form for creating a new user or admin profile. Includes fields for name, email, phone, address, and choosing role/subscription.
Props: <ul style="list-style-type: none">• <code>mode ("user" "admin")</code> — determines the form mode• <code>onSubmit (Function)</code> — callback to create a new profile• <code>onUploadPhoto (Function)</code> — callback for profile photo upload
States:
Behavior: <ul style="list-style-type: none">• Expands the dropdown to choose between User or Admin

in account.
quired) gic (required) tional)
it manually) rd" "vip" in") ible in admin mode) g user/admin ssages
=== 'admin' n data
-admin) re, InputFields — Name, ie, Status, PasswordField
uploading a profile picture, ption type.
ile to upload
dmin

<ul style="list-style-type: none">• Parent: LoginPage• Children: InputField (x2), LoginButton, HelpIcon
Variations:
InputField
Reusable input field component.
Props: <ul style="list-style-type: none">• placeholder: {string}• type: {string} - e.g., text/password• onChange: {function}
States:
Behavior: <ul style="list-style-type: none">• Renders an input box.• Captures user typing and triggers the onChange callback.
Composition: <ul style="list-style-type: none">• Parent: LoginForm• Children: None
Variations: <ul style="list-style-type: none">• Type “text”, “password”, “email”.• With or without a visibility toggle for passwords.
LoginButton
Submits the login form.
Props: <ul style="list-style-type: none">• onClick: {function}• text: {string} - Defaults to "Log In"
States:
Behavior: <ul style="list-style-type: none">• Triggers form submission logic.• Can show a spinner or loading state on click.
Composition: <ul style="list-style-type: none">• Parent: LoginForm• Children: None
Variations: <ul style="list-style-type: none">• Could be primary, secondary or loading state.

AddUser
Adds a new user
Props: <ul style="list-style-type: none">• onClick: {function} - Handler to add a
States:
Behavior: <ul style="list-style-type: none">• Calls onClick() when clicked.• Opens a modal or form to add a new u
Composition: <ul style="list-style-type: none">• Parent: AdminMainPage• Children: None
Variations:

Search
Enables filtering
Props: <ul style="list-style-type: none">• onSearch: function• placeholder: string
States: <ul style="list-style-type: none">• searchQuery: string
Behavior: <ul style="list-style-type: none">• Filters the list based on search input.
Composition: <ul style="list-style-type: none">• Parent: AdminMainPage• Children: None
Variations: <ul style="list-style-type: none">• Could include filter tags or dropdowns

UserList
Lists user
Props: <ul style="list-style-type: none">• users: array• onUserSelect: function
States:
Behavior: <ul style="list-style-type: none">• Displays the list of users.• Handles user selection.
Composition: <ul style="list-style-type: none">• Parent: AdminMainPage

rButton
to the system.
new user (required)
user.
hBar
g the user list.
s for advanced search.

List
r cards.

Composition: <ul style="list-style-type: none">Parent: StatisticsGraphChildren: None
Variations:

Scroll
A component that enables smooth scrolling between multiple charts or content sections.
Props: <ul style="list-style-type: none">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: <ul style="list-style-type: none">isScrolling: boolean - Tracks if the user is currently scrollingscrollPosition: number - Stores current scroll position (optional, if needed)
Behavior: <ul style="list-style-type: none">Provides scrollable area for child charts or sectionsSupports horizontal or vertical scroll based on the direction propOptional snap behavior for smooth chart-to-chart scrollTriggers onScrollEnd() after scroll finishes (if provided)
Composition: <ul style="list-style-type: none">Parent: StatisticsGraphChildren: None
Variations:

Varia
Prop
State
Beha
Com
Varia
Prop
State
Beha
Com

ations:
InputField
Input field for user data.
JS:
<ul style="list-style-type: none">value: {string} - Current value (required)onChange: {function} - Updates the data (required)type: {string} - Field type (text, email)
es:
avios:
<ul style="list-style-type: none">Captures user input and passes it to onChange.
iposition:
<ul style="list-style-type: none">Parent: UserProfileFormChildren: None
ations:
SubscriptionDropdown
Allows changing the user's subscription type.
JS:
<ul style="list-style-type: none">value: {string} - Current subscription type (required)onChange: {function} - Updates the subscription type (required)
es:
avios:
<ul style="list-style-type: none">Allows selecting a new subscription type.
iposition:
<ul style="list-style-type: none">Parent: UserProfileFormChildren: None
ations:
EditButton
Enables editing mode.
JS:
<ul style="list-style-type: none">onClick: {function} - Activates editing mode (required)
es:
avios:
<ul style="list-style-type: none">Sets isEditing to true.
iposition:
<ul style="list-style-type: none">Parent: UserProfileFormChildren: None

<ul style="list-style-type: none">If "Admin" is selected → switch to Admin Registration appears)Upload and preview profile photoOn Create click → profile is created with all the entered data
Composition:
<ul style="list-style-type: none">Parent: AdminMainPage / AddUserPageChildren: InputField, Dropdown, ProfilePhotoUpload
Variations:
<ul style="list-style-type: none">User Creation Form (default)With Profile Photo Upload (default)Without Photo Upload (minimal mode)

InputField
Reusable component for form input fields
Props:
<ul style="list-style-type: none">label (string)placeholder (string)value (string)onChange (function)type (text, number, email, password, etc.)
States:
Behavior:
<ul style="list-style-type: none">Allows the admin to edit name, email, phone, address and password.Calls onChange(updatedField, value) when the user changes the value
Composition:
<ul style="list-style-type: none">Parent: AddUserPageChildren: None
Variations:
<ul style="list-style-type: none">Text input (default)Password inputEmail input

ProfileIcon
Displays the user's profile avatar.
Props:
<ul style="list-style-type: none">userImage: {string} - URL of the user avatar placeholder image
States:
Behavior:
<ul style="list-style-type: none">Displays the user's/admin's avatar.
Composition:

Form (password input

ed data

als.

s, city, membership type,
user types in an input field.

ir (optional, default:

<ul style="list-style-type: none">Children: <code>UserListItem</code> (repeated)
Variations: <ul style="list-style-type: none">Can highlight active/selected user.

UserList
Represents a
Props: <ul style="list-style-type: none"><code>user</code>: object<code>onClick</code>: function
States:
Behavior: <ul style="list-style-type: none">Shows user details.Calls <code>onClick</code> when selected.
Composition: <ul style="list-style-type: none">Parent: <code>UserList</code>Children: None
Variations:

stItem
single user.

Varia
Prop
State
Beha
Com
Varia
Prop
State
Beha
Com
Varia

ations:

SaveButton

Saves user changes.

)S:

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

es:

avios:

- Calls `onSaveUser(updatedUser)`.

position:

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

ations:

DeleteButton

Deletes a user.

)S:

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

es:

avios:

- Calls `onDeleteUser(user)`.

position:

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

ations:

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 clicked (required)
- variant: {string} - button style (primary, etc.) (optional, default: "primary")

States:

Behavior:

- Calls `onClick()` when clicked.

Composition:

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

Variations:

- Primary, secondary, disabled states.

When the button is secondary, danger,