

account__manager

account__manager

Modules

- (screens)/(auth)/login/page
- (screens)/(auth)/register/page
- (screens)/(base)/chat/[chatType]/[id]/page
- (screens)/(base)/group-settings/[groupId]/page
- (screens)/(base)/layout
- (screens)/(base)/page
- (screens)/(base)/profile/[id]/page
- (screens)/(base)/settings/page
- (screens)/layout
- colors
- components
- components/Aside
- components/Button
- components/Divider
- components/Header
- components/Icon
- components/Message
- components/ProfileImage
- components/Section
- components/TextInput
- components/UserCard
- config
- layout
- socket

account__manager

ChatApp

Project: Web Application for User Communication

- The application allows user database management through a login system. Users can only edit or delete their own accounts. They can send private messages to each other and create group chats. The project emphasizes simplicity, security, and an intuitive interface.
Key Features:

- **User Database Management:** Secure login/authentication system.
- **Account Control:** Users can only modify or delete their own accounts.
- **Communication Tools:** Private messaging and group chat functionality.
- **Design Principles:** Focus on simplicity, security, and user-friendly interface. ## Instalation and run ### Frontend: Instalation:

```
cd website
npm i
```

Run for development `npm run dev`

Backend

Requirements

- Python 3.9+
- SQLite

Installation

```
cd backend\utils
pip install -r requirements.txt
```

Run Server

```
flask run --port=5000 --debug
```

`account_manager`

`account_manager / colors`

colors

Variables

- default

`account_manager`

`account_manager / colors / default`

Variable: default

```
const default: object
```

Defined in: colors.tsx:5

Obiekt colors zawierający stałe kolory używane w aplikacji

Type declaration

border

```
border: string = "#7B7B7B"
```

on_surface_gray

```
on_surface_gray: string = "#565656"
```

on_surface_light_gray

```
on_surface_light_gray: string = "#A0A0A0"
```

on_surface_white

```
on_surface_white: string = "#EAE5E5"
```

primary

```
primary: string = "#680B8A"
```

red

```
red: string = "#FF0000"
```

account_manager

account_manager / components

components

Functions

- Header
- Icon
- Message
- ProfileImage
- Section
- TextInput

- `UserCard`

References

Aside

Renames and re-exports default

Button

Renames and re-exports default

Divider

Renames and re-exports default

`account__manager`

`account__manager` / `components/Aside`

`components/Aside`

Functions

- `default`
- `FriendsGroupButton`
- `UserCard`

`account__manager`

`account__manager` / `components/Aside` / `default`

Function: `default()`

`default(props): Element`

Defined in: `components/Aside.tsx:21`

Komponent `Aside` wyświetlający listę użytkowników lub grup oraz przycisk do tworzenia grup

Parameters

props

obiekt właściwości komponentu

groups `any`

lista grup do wyświetlenia

users `any`

lista użytkowników do wyświetlenia

Returns

Element

- komponent

account__manager

account__manager / components/Aside / FriendsGroupButton

Function: FriendsGroupButton()

FriendsGroupButton(props): Element

Defined in: components/Aside.tsx:101

Przycisk wyboru pomiędzy listą przyjaciół a listą grup

Parameters

props

obiekt właściwości komponentu

callback `Function`

funkcja zmieniająca widok (przyjaciele/grupy)

checked `boolean`

czy przycisk jest aktualnie wybrany

label `string`

etykieta przycisku (“Friends” lub “Groups”)

Returns

Element

- komponent

account__manager

account__manager / components/Aside / UserCard

Function: UserCard()

UserCard(props): Element

Defined in: components/Aside.tsx:78

Komponent karty użytkownika

Parameters

props

obiekt właściwości komponentu

onClick? () => void

funkcja wywoływana po kliknięciu na kartę (opcjonalna)

user any

obiekt reprezentujący użytkownika lub grupę

Returns

Element

- komponent

account__manager

account__manager / components/Button

components/Button

Functions

- default

account__manager

account__manager / components/Button / default

Function: default()

default(props): Element

Defined in: components/Button.tsx:12

Komponent Button wyświetlający przycisk z trzema wariantami stylów

Parameters

props

obiekt właściwości komponentu

label string

tekst na przycisku

onClick () => void

funkcja wywoływana po kliknięciu przycisku

type? "filled" | "outlined" | "outlined2" = "filled"

typ przycisku: “filled” | “outlined” | “outlined2” (domyślnie “filled”)

Returns

Element

- komponent

account__manager

account__manager / components/Divider

components/Divider

Functions

- default

account__manager

account__manager / components/Divider / default

Function: default()

default(props): Element

Defined in: components/Divider.tsx:12

Komponent Divider wyświetlający poziomy lub pionowy pasek podziału

Parameters

props

obiekt właściwości komponentu

color string

kolor tła paska

horizontal? boolean

jeśli true, pasek jest poziomy

vertical? boolean

jeśli true, pasek jest pionowy

Returns

Element

- komponent

account__manager

account__manager / components / Header

Function: Header()

Header(props): Element

Defined in: components/Header.tsx:21

Komponent Header wyświetlający nagłówek strony

Parameters

props

obiekt właściwości komponentu

backArrow? `boolean`

czy wyświetlać strzałkę wstecz (opcjonalne)

header `any`

zawartość nagłówka

userProfileSrc? `string`

adres URL zdjęcia profilowego użytkownika (opcjonalne)

Returns

`Element`

- komponent

account__manager

`account__manager` / `components` / `Icon`

Function: `Icon()`

`Icon(props): Element`

Defined in: `components/Icon.tsx:12`

Komponent `Icon` wyświetlający ikonę

Parameters

props

obiekt właściwości komponentu

onClick `() => void`

funkcja wywoływana po kliknięciu na ikonę

size `number`

rozmiar ikony

src? `string`

źródło obrazka (jeśli pusty, wyświetlany jest obraz domyślny)

Returns

`Element`

- komponent

account__manager

account__manager / components / Message

Function: Message()

`Message(props): Element`

Defined in: components/Message.tsx:15

Komponent Message wyświetlający pojedynczą wiadomość

Parameters

props

obiekt właściwości komponentu

children `any`

zawartość wiadomości

isFirst? `Boolean`

czy jest to pierwsza wiadomość w grupie wiadomości (opcjonalne)

isLast? `Boolean`

czy jest to ostatnia wiadomość w grupie wiadomości (opcjonalne)

isUserAuthor? `Boolean`

czy autorem wiadomości jest użytkownik (domyślnie false)

name? `string`

imię autora wiadomości (opcjonalne)

src? `string`

adres URL zdjęcia profilowego autora wiadomości

Returns

Element

- komponent

account__manager

account__manager / components / ProfileImage

Function: ProfileImage()

ProfileImage(props): Element

Defined in: components/ProfileImage.tsx:10

Komponent ProfileImage wyświetlający okrągłe zdjęcie profilowe z oznaczeniem aktywności

Parameters

props

obiekt właściwości komponentu

isActive? `Boolean`

czy użytkownik jest aktywny (opcjonalne)

size `number`

rozmiar obrazka (szerokość i wysokość w px)

src `string`

adres URL zdjęcia profilowego

Returns

Element

- komponent

account__manager

account__manager / components / Section

Function: Section()

Section(props): Element

Defined in: components/Section.tsx:10

Komponent Section wyświetlający sekcję z nagłówkiem i zawartością

Parameters

props

obiekt właściwości komponentu

children any

zawartość sekcji

header String

nazwa sekcji

onClick? () => void

funkcja wywoływana po kliknięciu na sekcję (opcjonalne)

Returns

Element

- komponent

account__manager

account__manager / components / TextInput

Function: TextInput()

TextInput(props): Element

Defined in: components/TextInput.tsx:15

Komponent Input wyświetlający pole do wprowadzania tekstu z etykietą

Parameters

props

obiekt właściwości komponentu

label `string`

etykieta pola

long? `boolean`

czy użyć pola textarea zamiast input (opcjonalne)

password? `boolean`

czy pole jest typu hasło (opcjonalne)

placeholder? `string`

tekst zastępczy w polu (opcjonalny)

setValue `Function`

funkcja ustawiająca nową wartość pola

value `string`

aktualna wartość pola

Returns

`Element`

- komponent

account__manager

`account__manager` / `components` / `UserCard`

Function: `UserCard()`

`UserCard(props): Element`

Defined in: `components/UserCard.tsx:18`

Komponent `UserCard` wyświetlający kartę użytkownika

Parameters

props

obiekt właściwości komponentu

buttons? any

opcjonalne przyciski do wyświetlenia obok nazwy (opcjonalne)

children? any

kontent karty (opcjonalne)

className? string

klasy (opcjonalne)

isActive Boolean

czy użytkownik jest aktywny

name String

imię użytkownika

onClick? () => void

funkcja wywoływana po kliknięciu na element z imieniem i nazwiskiem (opcjonalne)

onClickCard? () => void

funkcja wywoływana po kliknięciu na całą kartę (opcjonalne)

style? any

dodatkowe stylowanie (opcjonalne)

surname String

nazwisko użytkownika

Returns

Element

- komponent

account__manager

account_manager / components/Header

components/Header

References

default

Renames and re-exports Header

account_manager

account_manager / components/Icon

components/Icon

References

default

Renames and re-exports Icon

account_manager

account_manager / components/Message

components/Message

References

default

Renames and re-exports Message

account_manager

account_manager / components/ProfileImage

components/ProfileImage

References

default

Renames and re-exports ProfileImage

account__manager

[account__manager](#) / [components/Section](#)

components/Section

References

default

Renames and re-exports Section

account__manager

[account__manager](#) / [components/TextInput](#)

components/TextInput

References

default

Renames and re-exports TextInput

account__manager

[account__manager](#) / [components/UserCard](#)

components/UserCard

References

default

Renames and re-exports UserCard

account__manager

account__manager / config

config

Variables

- default

account__manager

account__manager / config / default

Variable: default

const default: object

Defined in: config.tsx:4

Konfiguracja aplikacji zawierająca adres API

Type declaration

api

api: string = "http://localhost:5000"

account__manager

account__manager / layout

layout

Variables

- metadata

Functions

- default

account__manager

account__manager / layout / default

Function: default()

default(props): Element

Defined in: layout.tsx:21

Główny layout

Parameters

props

Readonly<{ children: ReactNode; }>

obiekt właściwości komponentu

Returns

Element

Layout

account__manager

account__manager / layout / metadata

Variable: metadata

const metadata: Metadata

Defined in: layout.tsx:12

account__manager

account__manager / (screens)/(auth)/login/page

(screens)/(auth)/login/page

Functions

- default

account__manager

account__manager / (screens)/(auth)/login/page / default

Function: default()

default(): Element

Defined in: (screens)/(auth)/login/page.tsx:19

Strona logowania. Zarządza formularzem logowania użytkownika.

Returns

Element

Strona logowania.

account__manager

account__manager / (screens)/(auth)/register/page

(screens)/(auth)/register/page

Functions

- default

account__manager

account__manager / (screens)/(auth)/register/page / default

Function: default()

default(): Element

Defined in: (screens)/(auth)/register/page.tsx:19

Strona rejestracji użytkownika. Zarządza formularzem rejestracji oraz przesyłaniem danych do API.

Returns

Element

Strona rejestracji.

account__manager

account__manager / (screens)/(base)/chat/[chatType]/[id]/page

(screens)/(base)/chat/[chatType]/[id]/page

Functions

- default
- LabeledContent

account__manager

account__manager / (screens)/(base)/chat/[chatType]/[id]/page / default

Function: default()

default(props): Element

Defined in: (screens)/(base)/chat/[chatType]/[id]/page.tsx:17

Strona chatu Umożliwia chatowanie prywatne oraz grupowe

Parameters

props

any

Właściwości komponentu (nieużywane)

Returns

Element

Strona czatu

account__manager

account__manager / (screens)/(base)/chat/[chatType]/[id]/page / LabeledContent

Function: LabeledContent()

LabeledContent(props): Element

Defined in: (screens)/(base)/chat/[chatType]/[id]/page.tsx:196

Parameters

props

content `string`

Zawartość tekstowa

label `string`

Tekst etykiety

long? `boolean`

Flaga do wyświetlania dłuższego tekstu

Returns

`Element`

Komponent

account__manager

`account__manager / (screens)/(base)/group-settings/[groupId]/page`

`(screens)/(base)/group-settings/[groupId]/page`

Functions

- `default`
- `LabeledContent`
- `LineInput`
- `ParticipantCard`

account__manager

`account__manager / (screens)/(base)/group-settings/[groupId]/page / default`

Function: `default()`

`default(props): Element`

Defined in: `(screens)/(base)/group-settings/[groupId]/page.tsx:22`

Strona ustawień grupy `GroupSettings` Umożliwia zarządzanie ustawieniami grupy, w tym: - wyświetlanie i edycję nazwy i opisu grupy, - dodawanie i usuwanie

członków, - zmiana ról członków, - filtrowanie członków według ról (OWNER, ADMIN, USER).

Parameters

props

Właściwości komponentu

params `Promise<{ groupId: string; }>`

Promise zawierający ID grupy

Returns

`Element`

JSX element strony ustawień grupy

account__manager

account__manager / (screens)/(base)/group-settings/[groupId]/page / Labeled-Content

Function: LabeledContent()

LabeledContent(props): `Element`

Defined in: (screens)/(base)/group-settings/[groupId]/page.tsx:268

Parameters

props

children? `any`

Dzieci komponentu

content? `string`

Opcjonalna zawartość tekstowa

label `string`

Tekst etykiety

long? `boolean`

Flaga do wyświetlania dłuższego tekstu (większa czcionka)

Returns

Element

Komponent

account__manager

account__manager / (screens)/(base)/group-settings/[groupId]/page / LineInput

Function: LineInput()

LineInput(props): Element

Defined in: (screens)/(base)/group-settings/[groupId]/page.tsx:283

Parameters

props

placeholder string

Tekst zastępczy

setValue Function

Funkcja aktualizująca wartość inputa

value string

Wartość inputa

Returns

Element

Komponent

account__manager

account__manager / (screens)/(base)/group-settings/[groupId]/page / ParticipantCard

Function: ParticipantCard()

ParticipantCard(props): Element

Defined in: (screens)/(base)/group-settings/[groupId]/page.tsx:236

Parameters

props

Propsy komponentu

id `string`

ID użytkownika

isActive `boolean`

Status aktywności użytkownika

name `string`

Imię użytkownika

removeMember `(id) => void`

Funkcja do usuwania członka z grupy

role `string`

Rola użytkownika (OWNER, ADMIN, USER)

setRole `(id, role) => void`

Funkcja do zmiany roli użytkownika

surname `string`

Nazwisko użytkownika

userRole `string`

Rola aktualnego użytkownika

Returns

Element

account_manager

`account_manager / (screens)/(base)/layout`

(screens)/(base)/layout

Functions

- default

account__manager

account__manager / (screens)/(base)/layout / default

Function: default()

default(props): null | Element

Defined in: (screens)/(base)/layout.tsx:18

Layout strongy

Parameters

props

Właściwości komponentu

children any

Elementy potomne renderowane wewnątrz layoutu

Returns

null | Element

Layout

account__manager

account__manager / (screens)/(base)/page

(screens)/(base)/page

Functions

- default

account__manager

account__manager / (screens)/(base)/page / default

Function: default()

default(): Element

Defined in: (screens)/(base)/page.tsx:16

Strona główna Wyświetla dashboard użytkownika z listą znajomych, grup i zaproszeń do znajomych.

Returns

Element

Strona główna

account__manager

account__manager / (screens)/(base)/profile/[id]/page

(screens)/(base)/profile/[id]/page

Functions

- default
- Input
- LineInput

account__manager

account__manager / (screens)/(base)/profile/[id]/page / default

Function: default()

default(props): Element

Defined in: (screens)/(base)/profile/[id]/page.tsx:18

Strona profilu użytkownika. Umożliwia przeglądanie profili użytkowników

Parameters

props

Właściwości komponentu

params { id: string; }

Parametry trasy zawierające ID użytkownika

params.id string

Returns

Element

Strona profilu użytkownika

account__manager

account__manager / (screens)/(base)/profile/[id]/page / Input

Function: Input()

Input(props): Element

Defined in: (screens)/(base)/profile/[id]/page.tsx:103

Komponent Input wyświetlający etykietę oraz pole tekstowe lub tekst tylko do odczytu.

Parameters

props

Właściwości komponentu

editable? boolean

Flaga czy pole jest edytowalne

label string

Etykieta pola

long? boolean

Flaga czy pole ma być długie (textarea) czy jednolinijkowe (input)

placeholder string

Tekst zastępczy pola

setValue Function

Funkcja aktualizująca wartość

value string

Wartość pola

Returns

Element

JSX z etykietą oraz odpowiednim polem (textarea/input lub span)

account__manager

account__manager / (screens)/(base)/profile/[id]/page / LineInput

Function: LineInput()

LineInput(props): Element

Defined in: (screens)/(base)/profile/[id]/page.tsx:85

Komponent wyświetlający jednoliniowy tekst (tylko do odczytu).

Parameters

props

Właściwości komponentu

placeholder string

Tekst zastępczy wyświetlany, gdy wartość jest pusta

setValue Function

Funkcja ustawiająca wartość (nieużywana, bo komponent jest tylko do odczytu)

value string

Wyświetlana wartość tekstowa

Returns

Element

JSX z tekstem wyświetlanym w divie

account__manager

account__manager / (screens)/(base)/settings/page

(screens)/(base)/settings/page

Functions

- default

account__manager

account__manager / (screens)/(base)/settings/page / default

Function: default()

default(): Element

Defined in: (screens)/(base)/settings/page.tsx:16

Strona ustawień profilu użytkownika Umożliwia edycję danych osobowych i społecznościowych użytkownika

Returns

Element

Strona ustawień

account__manager

account__manager / (screens)/layout

(screens)/layout

Functions

- default

account__manager

account_manager / (screens)/layout / default

Function: default()

default(props): Element

Defined in: (screens)/layout.tsx:11

Layout z providerem socket.io

Parameters

props

obiekt właściwości komponentu

children `ReactNode`

elementy potomne renderowane wewnątrz providera

Returns

`Element`

Layout

account_manager

account_manager / socket

socket

Functions

- `SocketProvider`
- `useSocket`

account_manager

account_manager / socket / SocketProvider

Function: SocketProvider()

SocketProvider(children): Element

Defined in: socket.tsx:21

Provider zarządzający połączeniem socket.io i udostępniający je komponentom potomnym

Parameters

children

elementy potomne, które mogą korzystać z kontekstu socket

children `ReactNode`

Returns

`Element`

account__manager

account__manager / socket / useSocket

Function: useSocket()

useSocket(): any

Defined in: socket.tsx:67

Hook zwracający kontekst socket.io

Returns

`any`

Obiekt z właściwościami socket oraz isConnected

Throws

Błąd jeśli hook jest używany poza kontekstem