**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

|  |  |  |  |
| --- | --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  доцент департамента  Программной инженерии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Р.З. Ахметсафина  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. | |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В. В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.06.02-01 12 01-1 | | **Панель управления интерфейса веб-приложения “StudyGo” для поддержки образовательного процесса**  **Текст программы**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.02-01 12 01-1-ЛУ** | | | | | |
|  | |  | | | |
| Исполнитель  студент группы БПИ197  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Бакытбек уулу Н. /  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | | | |

**Москва 2021**

|  |  |  |  |
| --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.06.02-01 12 01-1 |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | RU.17701729.06.02-01 12 01-1 | | **Панель управления интерфейса веб-приложения “StudyGo” для поддержки образовательного процесса**  **Текст программы**  **RU.17701729.06.02-01 12 01-1**  **Листов 35** | | | |
|
|
|
|  | | |  |

**Москва 2021**

Оглавление

[1 Components 3](#_Toc72499700)

[1.1 Sidebar 3](#_Toc72499701)

[1.1.1 AddChannelPopUp.css 3](#_Toc72499702)

[1.1.2 AddChannelPopUp.js 6](#_Toc72499703)

[1.1.3 Bottom.css 9](#_Toc72499704)

[1.1.4 Bottom.js 9](#_Toc72499705)

[1.1.5 Middle.css 10](#_Toc72499706)

[1.1.6 Middle.js 10](#_Toc72499707)

[1.1.7 SidebarOption 11](#_Toc72499708)

[1.1.8 SidebarSubject.js 12](#_Toc72499709)

[1.1.9 Top.css 14](#_Toc72499710)

[1.1.10 Top.js 14](#_Toc72499711)

[1.1.11 Sidebar.css 15](#_Toc72499712)

[1.2 Group-setting 18](#_Toc72499713)

[1.2.1 CreateGroupModal.js 18](#_Toc72499714)

[1.2.2 EditGroupSettingModal.js 20](#_Toc72499715)

[1.2.3 GroupSettingMenu.css 22](#_Toc72499716)

[1.2.4 GroupSettingMenu.js 24](#_Toc72499717)

[1.2.5 InviteLink 29](#_Toc72499718)

[1.3 user-setting 30](#_Toc72499719)

[1.3.1 UserSettingSidebar.js 30](#_Toc72499720)

[1.3.2 UserSettingSidebar.js 30](#_Toc72499721)

[Список используемой литературы 34](#_Toc72499722)

[Лист регистрации изменений. 35](#_Toc72499723)

# Components

## Sidebar

### AddChannelPopUp.css

.addChennal {

background-color: white;

padding: 20px;

border-radius: 5px;

font-size: 20px;

width: 400px;

}

@media (max-width: 500px) {

.addChennal {

width: 300px;

}

}

.addChennal > form {

display: flex;

flex-direction: column;

color: black;

}

.addChennal > form > input {

margin: 10px;

font-size: 16px;

height: 35px;

}

.addChennal > form > input#file-upload-button {

background-color: rebeccapurple;

}

.addChennal > form > h5 {

margin-left: 10px;

}

.addChennal > form > h3 {

margin-bottom: 1rem;

}

.addChennal > form > p {

font-size: 11px;

margin: 0;

margin-bottom: 2rem;

color: gray;

}

.addChennal > form > button {

display: inline-block;

padding: 0.4em 1.3em;

margin: 0 0.3em 0.3em 0;

border-radius: 2px;

box-sizing: border-box;

text-decoration: none;

font-weight: 500;

color: #ffffff;

background-color: var(--slack-color);

box-shadow: inset 0 -0.6em 1em -0.35em rgba(0, 0, 0, 0.17),

inset 0 0.6em 2em -0.3em rgba(255, 255, 255, 0.15),

inset 0 0 0em 0.05em rgba(255, 255, 255, 0.12);

text-align: center;

position: relative;

margin-top: 2rem;

align-self: flex-end;

margin-right: 0.6rem;

border: none;

font-size: 16px;

}

.addChennal > form > button:active {

box-shadow: inset 0 0.6em 2em -0.3em rgba(0, 0, 0, 0.15),

inset 0 0 0em 0.05em rgba(255, 255, 255, 0.12);

}

.example-1 input[type='file'] {

outline: 0;

opacity: 0;

pointer-events: none;

user-select: none;

}

.example-1 .label {

width: 100%;

border: 2px dashed grey;

border-radius: 5px;

display: block;

padding: 1.2em;

transition: border 300ms ease;

cursor: pointer;

text-align: center;

}

.example-1 .label i {

display: block;

size: 100px;

padding-bottom: 16px;

}

.example-1 .label i > .MuiSvgIcon-root {

height: 80px;

width: 80px;

}

.example-1 .label i,

.example-1 .label .title {

color: grey;

transition: 200ms color;

}

.example-1 .label:hover {

border: 2px solid #000;

}

.example-1 .label:hover i,

.example-1 .label:hover .title {

color: #000;

}

### AddChannelPopUp.js

import React, { useState } from 'react'

import './AddChannelPopUp.css'

import ReactDom from 'react-dom'

import { useDispatch, useSelector } from 'react-redux'

import { enterChannel, enterGroup, selectGroupId } from '../../features/appSlice'

import { useAuthState } from 'react-firebase-hooks/auth'

import { auth, db } from '../../firebase'

const MODAL\_STYLES = {

position: 'fixed',

top: '50%',

left: '50%',

transform: 'translate(-50%, -50%)',

zIndex: 1000

}

const OVERLAY\_STYLES = {

position: 'fixed',

top: 0,

left: 0,

right: 0,

bottom: 0,

backgroundColor: 'rgba(0, 0, 0, .7)',

zIndex: 1000

}

function AddChannelPopUp({ onClose }) {

const groupId = useSelector(selectGroupId)

const [user, loading, error] = useAuthState(auth);

const dispatch = useDispatch();

const [name, setName] = useState('')

const [description, setDescription] = useState('')

const [url, setUrl] = useState('')

const createChannel = (e) => {

e.preventDefault()

if (!name) {

alert('Without name can\'t create new channel');

return;

}

onClose();

db.collection('channels')

.add({

name: name,

description: description,

urlPic: url,

creator: user.email,

})

.then(function (docRef) {

console.log('Document written with ID: ', docRef.id);

db.collection('groups')

.doc(groupId)

.collection('channels-id')

.add({ id: docRef.id });

dispatch(enterChannel({ channelId: docRef.id }));

//обновиить currentGroup

db.collection('users')

.doc(user.email)

.update({ currentChannel: docRef.id });

})

.catch(function (error) {

console.error('Error adding document: ', error);

});

}

return ReactDom.createPortal(

<>

<div style={OVERLAY\_STYLES} onClick={() => onClose()} />

<div style={MODAL\_STYLES}>

<div className="addChennal">

<form>

<h3>Create a Channel</h3>

<p>Channels are where your team communicates. They’re best when organized around a topic — #marketing, for example.</p>

<h5>Name</h5>

<input value={name} onChange={(e) => { setName(e.target.value); }} type="text" />

<h5>Description</h5>

<input value={description} onChange={(e) => { setDescription(e.target.value); }} placeholder='optional' type="text" />

<h5>Url Picture</h5>

<input value={url} onChange={(e) => { setUrl(e.target.value); }} placeholder='optional' type="text" />

<button onClick={createChannel}>Create</button>

</form>

</div>

</div>

</>,

document.getElementById('portal')

)

}

export default AddChannelPopUp

### Bottom.css

.bottom {

width: 100%;

}

### Bottom.js

import React, { useState } from 'react';

import './Bottom.css';

import AddIcon from '@material-ui/icons/Add';

import SidebarOption from './SidebarOption';

import SidebarSubject from './SidebarSubject';

import AddChannelPopUp from './AddChannelPopUp';

import { useSelector } from 'react-redux';

import { selectGroupId } from '../../features/appSlice';

import { useCollection } from 'react-firebase-hooks/firestore';

import { db } from '../../firebase';

import { Link } from 'react-router-dom';

function Bottom({ onClose }) {

const groupId = useSelector(selectGroupId);

const [channels] = useCollection(

groupId && db.collection('groups').doc(groupId).collection('channels-id')

);

const [popUpAddChannel, setPopUpAddChannel] = useState(false);

return (

<div className='bottom'>

<div

className='bottom\_addChannel'

onClick={() => setPopUpAddChannel(true)}

>

<SidebarOption Icon={AddIcon} title='Add Channel' />

</div>

{popUpAddChannel && (

<AddChannelPopUp onClose={() => setPopUpAddChannel(false)} />

)}

<div>

{channels?.docs.map((doc) => (

<Link to='/channel' className='middle\_link' key={doc.id} >

<SidebarSubject key={doc.id} id={doc.data().id} onClose={() => onClose()} />

</Link>

))}

</div>

</div>

);

}

export default Bottom;

### Middle.css

.middle {

width: 100%;

border-bottom: 1px solid lightgray;

}

.middle\_item {

margin: 20px;

}

.middle\_link {

text-decoration: none;

}

### Middle.js

import React from 'react';

import './Middle.css';

import EventIcon from '@material-ui/icons/Event';

import ListAltIcon from '@material-ui/icons/ListAlt';

import WhatshotIcon from '@material-ui/icons/Whatshot';

import SidebarOption from './SidebarOption';

import { Link } from 'react-router-dom';

function Middle({ onClose }) {

return (

<div className='middle'>

{/\* <Link to='/schedule' className='middle\_link' onClick={() => onClose()}>

<SidebarOption

className='middle\_item'

Icon={EventIcon}

title='Schedule'

/>

</Link> \*/}

<Link to='/tasks' className='middle\_link' onClick={() => onClose()}>

<SidebarOption

className='middle\_item'

Icon={ListAltIcon}

title='Tasks'

/>

</Link>

<Link to='/deadlines' className='middle\_link' onClick={() => onClose()}>

<SidebarOption

className='middle\_item'

Icon={WhatshotIcon}

title='Deadlines'

/>

</Link>

</div>

);

}

export default Middle;

### SidebarOption

import React from 'react';

import styled from 'styled-components';

function SidebarOption({ Icon, title }) {

return (

<SidebarOptionContainer>

<Icon />

<h3>{title}</h3>

</SidebarOptionContainer>

);

}

export default SidebarOption;

const SidebarOptionContainer = styled.div`

display: flex;

font-size: 12px;

align-items: center;

padding: 10px;

cursor: pointer;

width: 100%;

:hover {

opacity: 0.9;

background-color: #340e36;

}

> h3 {

font-weight: 500;

color: white;

}

> h3 > span {

padding: 15px;

}

> .MuiSvgIcon-root {

margin-right: 0.5rem;

color: white;

}

`;

### SidebarSubject.js

import React from 'react';

import { useAuthState } from 'react-firebase-hooks/auth';

import { useDocument } from 'react-firebase-hooks/firestore';

import { useDispatch } from 'react-redux';

import styled from 'styled-components';

import { enterChannel } from '../../features/appSlice';

import { auth, db } from '../../firebase';

import ChannelPage from '../pages/ChannelPage';

function SidebarSubject({ id, onClose }) {

const dispatch = useDispatch();

const [user] = useAuthState(auth);

const [channel] = useDocument(db.collection('channels').doc(id));

return (

<SidebarOptionContainer

onClick={() => {

dispatch(enterChannel({ channelId: id }));

onClose();

db.collection('users').doc(user.email).update({ currentChannel: id });

}}

>

<h3>

<span>#</span>

{channel?.data()?.name}

</h3>

</SidebarOptionContainer>

);

}

export default SidebarSubject;

const SidebarOptionContainer = styled.div`

display: flex;

font-size: 12px;

align-items: center;

padding: 10px;

padding-left: 30px;

cursor: pointer;

:hover {

opacity: 0.9;

background-color: #340e36;

}

> h3 {

font-weight: 500;

color: white;

}

> h3 > span {

padding: 7px;

}

> .MuiSvgIcon-root {

margin-right: 0.5rem;

color: white;

}

`;

const SidebarOptionChannel = styled.h3`

padding: 10px 0;

font-weight: 300;

`;

### Top.css

.top {

display: flex;

padding: 20px;

padding-left: 22px;

width: 100%;

border-bottom: 1px solid lightgray;

}

.top\_info {

flex: 1;

}

.top\_info > h2 {

font-size: 15px;

font-weight: 900;

margin-bottom: 5px;

color: white;

}

.top\_info > h3 {

display: flex;

font-size: 13px;

font-weight: 400;

align-items: center;

color: white;

}

.top\_info > h3 > .MuiSvgIcon-root {

font-size: 14px;

margin-top: 1px;

margin-right: 2px;

color: green;

}

### Top.js

import React, { useState } from 'react';

import FiberManualRecordIcon from '@material-ui/icons/FiberManualRecord';

import './Top.css';

import styled from 'styled-components'

import { Avatar, Button } from '@material-ui/core'

import { useSelector } from 'react-redux';

import { selectGroupId } from '../../features/appSlice';

import { useDocument } from 'react-firebase-hooks/firestore';

import { auth, db } from '../../firebase';

import { useAuthState } from 'react-firebase-hooks/auth';

function Top() {

const groupId = useSelector(selectGroupId)

const [group] = useDocument(groupId && db.collection('groups').doc(groupId));

const [user, loading, error] = useAuthState(auth);

return (

<div className='top'>

<div className='top\_info'>

<h2>{group?.data()?.name}</h2>

<h3>

<FiberManualRecordIcon />

{user.displayName}

</h3>

</div>

<HeaderAvatar src={group?.data()?.urlPic} alt='БПИ197' />

</div>

);

}

export default Top;

const HeaderAvatar = styled(Avatar)`

cursor: pointer;

:hover {

opacity: 0.8;

}

width:3.5rem;

height:3.5rem

`

### Sidebar.css

.navbar {

background-color: var(--slack-color);

height: 60px;

display: flex;

justify-content: start;

align-items: center;

}

.menu-bars {

margin-left: 2rem;

font-size: 2rem;

background: none;

}

.nav-menu {

/\* margin-top: var(--app-bar-height); \*/

background-color: var(--slack-color);

width: var(--sidebar-width);

/\* height: calc(100vh - var(--app-bar-height)); \*/

justify-content: center;

position: fixed;

top: 0;

left: -100%;

transition: 350ms;

overflow-y: auto;

z-index: 10001;

height: 100vh;

}

.nav-menu.active {

left: 0;

transition: 250ms;

}

.nav-text {

display: flex;

justify-content: start;

align-items: center;

padding: 8px 0px 8px 16px;

list-style: none;

height: 60px;

}

.nav-text a {

text-decoration: none;

color: #f5f5f5;

font-size: 18px;

width: 95%;

height: 100%;

display: flex;

align-items: center;

padding: 0 16px;

border-radius: 4px;

}

.nav-text a:hover {

background-color: #4a154b;

}

.nav-menu-items {

width: 100%;

padding: 0;

}

.navbar-toggle-close {

align-items: center;

padding: 0.5rem;

display: flex;

height: 4rem;

}

.studyGoLogo {

align-self: center;

color: white;

}

span {

margin-left: 16px;

}

## Group-setting

### CreateGroupModal.js

import React, { useState } from 'react';

import { auth, db } from '../../firebase';

import { useDispatch } from 'react-redux';

import '../sidebar/AddChannelPopUp.css';

import ReactDom from 'react-dom';

import { useAuthState } from 'react-firebase-hooks/auth';

import { enterGroup } from '../../features/appSlice';

const MODAL\_STYLES = {

position: 'fixed',

top: '50%',

left: '50%',

transform: 'translate(-50%, -50%)',

zIndex: 1000,

};

const OVERLAY\_STYLES = {

position: 'fixed',

top: 0,

left: 0,

right: 0,

bottom: 0,

backgroundColor: 'rgba(0, 0, 0, .7)',

zIndex: 1000,

};

function CreateGroupModal({ onClose }) {

const [user, loading, error] = useAuthState(auth);

const [name, setName] = useState('');

const [url, setUrl] = useState('');

const dispatch = useDispatch();

if (loading) {

return <h3>user loading</h3>;

}

if (error) {

return <h3>error auth</h3>;

}

const createGroup = (e) => {

e.preventDefault();

if (!name) alert("Without name can't create new group");

else {

onClose();

db.collection('groups')

.add({

name: name,

urlPic: url,

})

.then(function (docRef) {

console.log('Document written with ID: ', docRef.id);

db.collection('users')

.doc(user.email)

.collection('groups-id')

.add({ id: docRef.id });

dispatch(enterGroup({ groupId: docRef.id }));

//обновиить currentGroup

db.collection('users')

.doc(user.email)

.update({ currentGroup: docRef.id });

})

.catch(function (error) {

console.error('Error adding document: ', error);

});

}

};

return ReactDom.createPortal(

<>

<div style={OVERLAY\_STYLES} onClick={() => onClose()} />

<div style={MODAL\_STYLES}>

<div className='addChennal'>

<form>

<h3>Create a group</h3>

<p>

From here you can create a new group, where you can communicate with friends, teachers and сolleagues.

</p>

<h5>Name</h5>

<input

value={name}

onChange={(e) => {

setName(e.target.value);

}}

type='text'

/>

<h5>Url Picture</h5>

<input

value={url}

onChange={(e) => {

setUrl(e.target.value);

}}

placeholder='optional'

type='text'

/>

<button onClick={createGroup}>Create</button>

</form>

</div>

</div>

</>,

document.getElementById('portal')

);

}

export default CreateGroupModal;

### EditGroupSettingModal.js

import React, { useState } from 'react';

import '../sidebar/AddChannelPopUp.css';

import ReactDom from 'react-dom';

import { db } from '../../firebase';

import { useSelector } from 'react-redux';

import { selectGroupId } from '../../features/appSlice';

import { useDocument } from 'react-firebase-hooks/firestore';

const MODAL\_STYLES = {

position: 'fixed',

top: '50%',

left: '50%',

transform: 'translate(-50%, -50%)',

zIndex: 1000,

};

const OVERLAY\_STYLES = {

position: 'fixed',

top: 0,

left: 0,

right: 0,

bottom: 0,

backgroundColor: 'rgba(0, 0, 0, .7)',

zIndex: 1000,

};

function EditGroupSettingModal({ onClose }) {

const groupId = useSelector(selectGroupId);

const [group] = useDocument(groupId && db.collection('groups').doc(groupId));

const [name, setName] = useState('');

const [url, setUrl] = useState('');

if (!group) {

return null;

}

if (!name && group?.data()?.name) setName(group?.data()?.name);

if (!url && group?.data()?.urlPic) setUrl(group?.data()?.urlPic);

const editGroup = (e) => {

e.preventDefault();

if (!name) {

alert("Without name can't edit group setting");

return;

}

onClose();

db.collection('groups').doc(groupId).update({ name: name, urlPic: url });

};

return ReactDom.createPortal(

<>

<div style={OVERLAY\_STYLES} onClick={() => onClose()} />

<div style={MODAL\_STYLES}>

<div className='addChennal'>

<form>

<h3>Group setting</h3>

<p>

From here you can manage the setting of current group.

</p>

<h5>Name</h5>

<input

value={name}

onChange={(e) => {

setName(e.target.value);

}}

type='text'

/>

<h5>Url Picture</h5>

<input

value={url}

onChange={(e) => {

setUrl(e.target.value);

}}

placeholder='optional'

type='text'

/>

<button onClick={editGroup}>Save</button>

</form>

</div>

</div>

</>,

document.getElementById('portal')

);

}

export default EditGroupSettingModal;

### GroupSettingMenu.css

.group-menu.active {

margin-top: var(--app-bar-height);

background-color: var(--slack-color);

width: var(--sidebar-width);

position: fixed;

right: auto;

left: 2rem;

color: white;

z-index: 999;

border-radius: 8px;

padding-top: 1rem;

}

.group-menu-item > input {

}

.group-menu-hide {

position: fixed;

top: -1000px;

}

.group-menu-items {

display: flex;

align-items: center;

flex-direction: column;

}

.group-menu-items > h4 {

margin: 5px;

margin-bottom: 1rem;

}

.group-signOut\_btn {

border-top: 1px solid lightgray;

width: 100%;

display: flex;

align-items: center;

flex-direction: column;

}

.group-signOut\_btn > .MuiButtonBase-root > .MuiButton-label {

margin: 0;

}

.groupOptions {

width: 100%;

padding-top: 0rem;

border-bottom: 1px solid lightgray;

border-top: 1px solid lightgray;

}

.groupsController {

width: 100%;

max-height: 250px;

overflow-y: auto;

}

### GroupSettingMenu.js

import React, { useState } from 'react';

import './GroupSettingMenu.css';

import { useAuthState } from 'react-firebase-hooks/auth';

import { auth, db } from '../../firebase';

import styled from 'styled-components';

import { Avatar, Button } from '@material-ui/core';

import EditIcon from '@material-ui/icons/Edit';

import EditOutlinedIcon from '@material-ui/icons/EditOutlined';

import PeopleOutlineIcon from '@material-ui/icons/PeopleOutline';

import PeopleAltIcon from '@material-ui/icons/PeopleAlt';

import LinkIcon from '@material-ui/icons/Link';

import ExitToAppIcon from '@material-ui/icons/ExitToApp';

import AddOutlinedIcon from '@material-ui/icons/AddOutlined';

import CreateGroupModal from './CreateGroupModal';

import EditGroupSettingModal from './EditGroupSettingModal';

import InviteLink from './InviteLink';

import ReactDom from 'react-dom';

import { useCollection, useDocument } from 'react-firebase-hooks/firestore';

import { useDispatch, useSelector } from 'react-redux';

import {

enterChannel,

enterGroup,

selectGroupId,

} from '../../features/appSlice';

import { useHistory } from 'react-router';

function SidebarOption({ Icon, title, action }) {

return (

<SidebarOptionContainer onClick={action}>

<Icon />

<h3>{title}</h3>

</SidebarOptionContainer>

);

}

function GroupItem({ id }) {

const dispatch = useDispatch();

const [group] = useDocument(id && db.collection('groups').doc(id));

const [user, loading, error] = useAuthState(auth);

const groupId = useSelector(selectGroupId);

const history = useHistory();

return (

<SidebarOptionContainer

onClick={() => {

if (id !== groupId) {

dispatch(enterGroup({ groupId: id }));

dispatch(enterChannel({ channelId: null }));

db.collection('users')

.doc(user.email)

.update({ currentGroup: id, currentChannel: null });

history?.push('/');

}

}}

>

<Avatar

src={group?.data()?.urlPic}

style={{

width: '30px',

height: '30px',

marginRight: '10px',

marginLeft: '30px',

}}

/>

<h3>{group?.data()?.name}</h3>

</SidebarOptionContainer>

);

}

const OVERLAY\_STYLES = {

position: 'fixed',

top: 0,

left: 0,

right: 0,

bottom: 0,

zIndex: 998,

};

const MODAL\_STYLES = {

position: 'fixed',

top: '50%',

left: '50%',

width: '100%',

height: '100%',

transform: 'translate(-50%, -50%)',

};

function GroupSettingMenu({ groupSetting, onClose }) {

const dispatch = useDispatch();

const [user, loading] = useAuthState(auth);

const [createGroupModal, setCreateGroupModal] = useState(false);

const [editGroupModal, setEditGroupModal] = useState(false);

const [inviteLink, setInviteLink] = useState(false);

const groupId = useSelector(selectGroupId);

const [group] = useDocument(groupId && db.collection('groups').doc(groupId));

const [groups] = useCollection(

db.collection('users').doc(user.email).collection('groups-id')

);

const leaveGroup = () => {

db.collection('users')

.doc(user.email)

.collection('groups-id')

.where('id', '==', groupId)

.get()

.then((query) => {

query.forEach((doc) => {

db.collection('users')

.doc(user.email)

.collection('groups-id')

.doc(doc.id)

.delete();

});

dispatch(enterGroup({ groupId: null }));

dispatch(enterChannel({ channelId: null }));

db.collection('users')

.doc(user.email)

.update({ currentGroup: null, currentChannel: null });

});

};

return ReactDom.createPortal(

<>

{groupSetting && (

<div style={MODAL\_STYLES}>

<div style={OVERLAY\_STYLES} onClick={() => onClose()} />

<nav

className={groupSetting ? 'group-menu active' : 'group-menu-hide'}

>

<div className='group-menu-items'>

<HeaderAvatar

src={group?.data()?.urlPic}

alt={group?.data()?.name[0]}

/>

<h4>{group?.data()?.name}</h4>

<div className='groupOptions'>

<SidebarOption

action={() => setEditGroupModal(true)}

title='Group setting'

Icon={EditOutlinedIcon}

/>

{editGroupModal && (

<EditGroupSettingModal

onClose={() => setEditGroupModal(false)}

/>

)}

{/\*<SidebarOption title='Members' Icon={PeopleOutlineIcon} />\*/}

<SidebarOption

action={() => setInviteLink(true)}

title='Invite link'

Icon={LinkIcon}

/>

{inviteLink && (

<InviteLink onClose={() => setInviteLink(false)} />

)}

<SidebarOption

action={() => leaveGroup()}

title='Leave group'

Icon={ExitToAppIcon}

/>

</div>

<div className='groupsController'>

<SidebarOption

action={() => setCreateGroupModal(true)}

title='Create new group'

Icon={AddOutlinedIcon}

/>

{createGroupModal && (

<CreateGroupModal

onClose={() => setCreateGroupModal(false)}

/>

)}

{groups?.docs.map((doc) => (

<GroupItem key={doc.id} id={doc.data().id} />

))}

</div>

</div>

</nav>

</div>

)}

</>,

document.getElementById('portal')

);

}

export default GroupSettingMenu;

const HeaderAvatar = styled(Avatar)`

cursor: pointer;

:hover {

opacity: 0.8;

}

width: 3.5rem;

height: 3.5rem;

`;

const SidebarOptionContainer = styled.div`

display: flex;

font-size: 12px;

align-items: center;

padding: 10px 20px;

cursor: pointer;

width: 100%;

:hover {

opacity: 0.9;

background-color: #340e36;

}

> h3 {

font-weight: 500;

color: white;

}

> h3 > span {

padding: 15px;

}

> .MuiSvgIcon-root {

margin-right: 0.5rem;

color: white;

}

`;

### InviteLink

import React, { useState } from 'react'

import '../sidebar/AddChannelPopUp.css'

import ReactDom from 'react-dom'

import { useSelector } from 'react-redux'

import { selectGroupId } from '../../features/appSlice'

const MODAL\_STYLES = {

position: 'fixed',

top: '50%',

left: '50%',

transform: 'translate(-50%, -50%)',

zIndex: 1000

}

const OVERLAY\_STYLES = {

position: 'fixed',

top: 0,

left: 0,

right: 0,

bottom: 0,

backgroundColor: 'rgba(0, 0, 0, .7)',

zIndex: 1000

}

function InviteLink({ onClose }) {

const groupId = useSelector(selectGroupId);

const [url, setUrl] = useState(groupId && (window.location.origin + '/join/' + groupId))

return ReactDom.createPortal(

<>

<div style={OVERLAY\_STYLES} onClick={() => onClose()} />

<div style={MODAL\_STYLES}>

<div className="addChennal">

<form>

<h3>Invite Link</h3>

<p>Send this link to your friends to add them into this group.</p>

<h5>Link</h5>

<input value={url} type="text" readOnly />

</form>

</div>

</div>

</>,

document.getElementById('portal')

)

}

export default InviteLink

## user-setting

### UserSettingSidebar.js

.menu.active {

margin-top: var(--app-bar-height);

background-color: var(--slack-color);

width: var(--sidebar-width);

position: fixed;

right: 2rem;

left: auto;

color: white;

z-index: 999;

border-radius: 8px;

}

.menu-hide {

position: fixed;

background-color: red;

top: -100%;

}

.menu-items {

padding-top: 1rem;

display: flex;

align-items: center;

flex-direction: column;

}

.menu-items > p {

margin: 0;

color: lightgray;

margin-bottom: 30px;

}

.signOut\_btn {

border-top: 1px solid lightgray;

width: 100%;

display: flex;

align-items: center;

flex-direction: column;

}

.signOut\_btn > .MuiButtonBase-root > .MuiButton-label {

margin: 0;

}

### UserSettingSidebar.js

import React, { useState } from 'react';

import './UserSettingSidebar.css'

import { useAuthState } from 'react-firebase-hooks/auth'

import { auth } from '../../firebase'

import styled from 'styled-components'

import { Avatar, Button } from '@material-ui/core'

import ReactDom from 'react-dom'

const OVERLAY\_STYLES = {

position: 'fixed',

top: 0,

left: 0,

right: 0,

bottom: 0,

zIndex: 998

}

const MODAL\_STYLES = {

position: 'fixed',

top: '50%',

left: '50%',

width: '100%',

height: '100%',

transform: 'translate(-50%, -50%)'

}

function UserSettingSidebar({ userSettingSidebar, onClose }) {

const [user, loading] = useAuthState(auth)

return ReactDom.createPortal(

<>

{userSettingSidebar &&

<div style={MODAL\_STYLES}>

<div style={OVERLAY\_STYLES} onClick={() => onClose()} />

<nav className={userSettingSidebar ? 'menu active' : 'menu-hide'}>

<div className='menu-items'>

<HeaderAvatar src={user?.photoURL} alt={user?.displayName} />

<h3>{user?.displayName}</h3>

<p>{user?.email}</p>

<div className='signOut\_btn'>

<Button color='inherit' onClick={() => {

onClose();

auth.signOut();

}}>

Sign out</Button>

</div>

</div>

</nav>

</div>

}

</>

, document.getElementById('portal'));

}

export default UserSettingSidebar;

const HeaderAvatar = styled(Avatar)`

cursor: pointer;

:hover {

opacity: 0.8;

}

width:3.5rem;

height:3.5rem

`

# Список используемой литературы

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

# Лист регистрации изменений.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | Новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |