**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 | | **Работа с онлайн базой данных и функциональной частью веб-приложения для поддержки образовательного процесса “StudyGo”**  **Текст программы**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.06.02-01 12 01-1-ЛУ** | | | | | |
|  | |  | | | |
| Исполнитель  студент группы БПИ197  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Яхя Я. /  «\_\_\_ »\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 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 | | **Работа с онлайн базой данных и функциональной частью веб-приложения для поддержки образовательного процесса “StudyGo”**  **Текст программы**  **RU.17701729.06.02-01 12 01-1**  **Листов 78** | | | |
|
|
|
|  | | |  |

**Москва 2021**

Оглавление

[1 Src проекта 3](#_Toc72498351)

[1.1 app 3](#_Toc72498352)

[1.1.1 store.js 3](#_Toc72498353)

[1.2 components 3](#_Toc72498354)

[1.2.1 check-works 3](#_Toc72498355)

[1.2.2 feedbacks 11](#_Toc72498356)

[1.2.3 group-setting 15](#_Toc72498357)

[1.2.4 pages 27](#_Toc72498358)

[1.2.5 sidebar 52](#_Toc72498359)

[1.2.6 user-setting 61](#_Toc72498360)

[1.2.7 Login.js 63](#_Toc72498361)

[1.2.8 MyAppBar.js 64](#_Toc72498362)

[1.2.9 Sidebar.js 66](#_Toc72498363)

[1.3 features 67](#_Toc72498364)

[1.3.1 appSlice.js 67](#_Toc72498365)

[1.4 App.js 68](#_Toc72498366)

[1.5 firebase.js 74](#_Toc72498367)

[1.6 index.js 75](#_Toc72498368)

[Список используемой литературы 76](#_Toc72498369)

[Лист регистрации изменений. 77](#_Toc72498370)

# Src проекта

## app

### store.js

import { configureStore } from '@reduxjs/toolkit';

import appReducer from '../features/appSlice';

export const store = configureStore({

  reducer: {

    app: appReducer,

  },

});

## components

### check-works

#### CheckPage.js

import React from 'react';

import Submits from './Submits';

function CheckPage() {

  return (

    <div>

      <Submits />

    </div>

  );

}

export default CheckPage;

#### Submits.js

import React from 'react';

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

import { useSelector } from 'react-redux';

import { useParams } from 'react-router';

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

import { db } from '../../firebase';

import WorkItem from './WorkItem';

function Submits() {

  const { postId } = useParams();

  const channelId = useSelector(selectChannelId);

  const [submits, loading, error] = useCollection(

    channelId &&

    postId &&

    db

      .collection('channels')

      .doc(channelId)

      .collection('posts')

      .doc(postId)

      .collection('submitedWorks')

      .orderBy('isSent')

  );

  if (!submits?.docs || loading || error) {

    return <h3>Submits loading</h3>;

  }

  return (

    <div>

      {submits?.docs.map((doc) => {

        return <WorkItem key={doc.id} id={doc.id} submitDoc={doc} />;

      })}

    </div>

  );

}

export default Submits;

#### WorkItem.js

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

import React, { useState } from 'react';

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

import { db } from '../../firebase';

import DescriptionOutlinedIcon from '@material-ui/icons/DescriptionOutlined';

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

import './WorkItem.css';

import WorkItemModal from './WorkItemModal';

import firebase from 'firebase';

import { useParams } from 'react-router';

import { useSelector } from 'react-redux';

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

function WorkItem({ id, submitDoc }) {

  const { postId } = useParams();

  const channelId = useSelector(selectChannelId);

  const submit = submitDoc?.data();

  const [modal, setModal] = useState(false);

  const [submitter, loading, error] = useDocument(

    submit?.submitter && db.collection('users').doc(submit?.submitter)

  );

  if (!submit || !submitter || loading || error) {

    return <h5>Submitter loading...</h5>;

  }

  const originalDate = new Date(submit?.time.toDate());

  const date =

    originalDate.getDate() +

    '/' +

    originalDate.getMonth() +

    '/' +

    originalDate.getFullYear() +

    ' ' +

    originalDate.getHours() +

    ':' +

    originalDate.getMinutes();

  const modalHeader = <h3>a</h3>;

  const sendReview = () => {

    db.collection('channels')

      .doc(channelId)

      .collection('posts')

      .doc(postId)

      .collection('submitedWorks')

      .doc(id)

      .update({

        isSent: 1,

        replyTime: firebase.firestore.FieldValue.serverTimestamp(),

      });

  };

  return (

    <div className='work-item-container'>

      <div className='work-item-left'>

        <Avatar src={submitter.data().urlPic}>

          {submitter.data().name[0]}

        </Avatar>

        <div className='work-item-info'>

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

          <span>{date}</span>

        </div>

      </div>

      <div className='work-item-controllers'>

        {submit?.mark && (

          <input

            type='number'

            readOnly

            value={submit?.mark ? submit?.mark : ''}

          />

        )}

        <DescriptionOutlinedIcon

          onClick={() => window.open(submit?.urlWork, '\_blank')}

        />

        {submit.isSent != 1 && (

          <EditOutlinedIcon onClick={() => setModal(true)} />

        )}

        {modal && (

          <WorkItemModal

            id={id}

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

            modalHeader={modalHeader}

            submitter={submitter}

            date={date}

            submit={submit}

          />

        )}

      </div>

    </div>

  );

}

export default WorkItem;

#### WorkItemModel.js

import React, { useState } from 'react';

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

import ReactDom from 'react-dom';

import styled from 'styled-components';

import DescriptionOutlinedIcon from '@material-ui/icons/DescriptionOutlined';

import SaveOutlinedIcon from '@material-ui/icons/SaveOutlined';

import './WorkItem.css';

import { useParams } from 'react-router';

import { useSelector } from 'react-redux';

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

import SendOutlinedIcon from '@material-ui/icons/SendOutlined';

import { db } from '../../firebase';

import firebase 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 WorkItemModal({ id, onClose, submitter, date, submit }) {

  const { postId } = useParams();

  const channelId = useSelector(selectChannelId);

  const [comment, setComment] = useState(submit.comment ? submit.comment : '');

  const [mark, setMark] = useState(submit.mark ? submit.mark : '');

  const saveReview = () => {

    if (!mark) {

      return;

    }

    db.collection('channels')

      .doc(channelId)

      .collection('posts')

      .doc(postId)

      .collection('submitedWorks')

      .doc(id)

      .update({

        mark: mark,

        comment: comment,

        isSent: 1,

        replyTime: firebase.firestore.FieldValue.serverTimestamp(),

      });

    db.collection('users').doc(submit?.submitter).collection('feedbacks').add({ channelId: channelId, postId: postId, submitId: id, time: firebase.firestore.FieldValue.serverTimestamp() });

  };

  return ReactDom.createPortal(

    <>

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

      <div style={MODAL\_STYLES}>

        <div className='work-item-modal'>

          <div className='work-item-container'>

            <div className='work-item-left'>

              <Avatar src={submitter.data().urlPic}>

                {submitter.data().name[0]}

              </Avatar>

              <div className='work-item-info'>

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

                <span>{date}</span>

              </div>

            </div>

            <div className='work-item-controllers'>

              <SidebarOption

                onClick={() => window.open(submit?.urlWork, '\_blank')}

                Icon={DescriptionOutlinedIcon}

                title='Open file'

              />

            </div>

          </div>

          <div className='work-item-comment-and-mark'>

            <div className='work-item-mark'>

              <h5>Mark</h5>

              <input

                type='number'

                value={mark}

                onChange={(e) => setMark(e.target.value)}

              />

            </div>

            <div className='work-item-comment'>

              <h5>Comments</h5>

              <TextField

                value={comment}

                onChange={(e) => setComment(e.target.value)}

                id='outlined-multiline-static'

                multiline

                rows={4}

                placeholder='Type comments...'

                variant='outlined'

              />

            </div>

            <div className='work-item-controlls'>

              <SidebarOption

                onClick={() => saveReview()}

                Icon={SendOutlinedIcon}

                title='Send'

              />

            </div>

          </div>

        </div>

      </div>

    </>,

    document.getElementById('portal')

  );

}

export default WorkItemModal;

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

  return (

    <SidebarOptionContainer onClick={onClick}>

      <Icon />

      <h3>{title}</h3>

    </SidebarOptionContainer>

  );

}

const SidebarOptionContainer = styled.div`

  display: flex;

  font-size: 12px;

  align-items: center;

  margin-top: 5px;

  padding: 5px;

  cursor: pointer;

  border-radius: 16px;

  color: gray;

  :hover {

    opacity: 0.9;

    background-color: lightgray;

  }

  > h3 {

    font-weight: 700;

  }

  > h3 > span {

    padding: 15px;

  }

  > .MuiSvgIcon-root {

    margin-right: 0.5rem;

  }

`;

### feedbacks

#### FeedbackItem.js

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

import VerifiedUserOutlinedIcon from '@material-ui/icons/VerifiedUserOutlined';

import '../pages/PostItem.css';

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

import { useState } from 'react';

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

function FeedbackItem({ channelId, postId, submitId, time }) {

  // name, urlPic, time, message, deadline, admin, postType, urlFile

  const [submitPopUp, setSubmitPopUp] = useState(false);

  const [channel] = useDocument(

    channelId && db.collection('channels').doc(channelId)

  );

  const [post] = useDocument(

    channelId &&

      postId &&

      db.collection('channels').doc(channelId).collection('posts').doc(postId)

  );

  const [submittedWork] = useDocument(

    channelId &&

      postId &&

      submitId &&

      db

        .collection('channels')

        .doc(channelId)

        .collection('posts')

        .doc(postId)

        .collection('submitedWorks')

        .doc(submitId)

  );

  if (!post || !channel || !submittedWork) {

    return null;

  }

  const { name, urlPic } = channel && channel?.data();

  const { deadline, input, urlFile } = post && post?.data();

  const { comment, mark, urlWork } = submittedWork && submittedWork?.data();

  const originalDate = new Date(time?.toDate());

  const date =

    '' +

    originalDate.getDate() +

    '/' +

    originalDate.getMonth() +

    '/' +

    originalDate.getFullYear() +

    ' ' +

    originalDate.getHours() +

    ':' +

    originalDate.getMinutes();

  return (

    name && (

      <div className='post'>

        <div className='post\_\_header'>

          <Avatar src={urlPic}>{name[0]}</Avatar>

          <div className='post\_\_info'>

            <h2>{name}</h2>

            <p>{date}</p>

          </div>

          <div className='postType'>

            <VerifiedUserOutlinedIcon style={{ color: 'green' }} />

          </div>

        </div>

        <div className='post\_\_body'>

          <p>{input}</p>

          {urlFile && (

            <p>

              <strong>Assignment: </strong>{' '}

              <a target='\_blank' rel='noreferrer noopener' href={urlFile}>

                view

              </a>

            </p>

          )}

          <p>

            <strong>Deadline: </strong> {deadline}

          </p>

          <p>

            <strong>Comment: </strong> {comment}

          </p>

          <p>

            <strong>Mark: </strong> {mark}

          </p>

          <p>

            <strong>Your submit: </strong>{' '}

            <a target='\_blank' rel='noreferrer noopener' href={urlWork}>

              view

            </a>

          </p>

        </div>

      </div>

    )

  );

}

export default FeedbackItem;

#### Feedbacks.js

import React from 'react';

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

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

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

import FeedbackItem from './FeedbackItem';

function Feedbacks() {

 const [user] = useAuthState(auth);

 const [feedback] = useCollection(user.email && db.collection('users').doc(user.email).collection('feedbacks').orderBy('time', 'desc'));

 return (

  <div>

   {feedback?.docs?.map((doc) => {

    const { channelId, postId, submitId, time } = doc?.data();

    return < FeedbackItem key={doc.id} channelId={channelId} postId={postId} submitId={submitId} time={time} />

   })}

  </div>

 )

}

export default Feedbacks

### 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;

#### EditGroupSetting.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

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

                  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

### pages

#### ChannelBuildPost.js

import React, { useState } from 'react';

import './ChannelBuildPost.css';

import TextField from '@material-ui/core/TextField';

import Radio from '@material-ui/core/Radio';

import RadioGroup from '@material-ui/core/RadioGroup';

import FormControlLabel from '@material-ui/core/FormControlLabel';

import FormControl from '@material-ui/core/FormControl';

import InputOption from './InputOption';

import Button from '@material-ui/core/Button';

import AttachFileOutlinedIcon from '@material-ui/icons/AttachFileOutlined';

import FileAttachModal from './FileAttachModal';

import { useSelector } from 'react-redux';

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

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

import firebase from 'firebase';

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

const styles = {

  container: {

    display: 'flex',

    flexWrap: 'wrap',

  },

  textField: {

    fontSize: '10px',

  },

  //style for font size

  resize: {

    fontSize: '10px',

  },

};

function ChannelBuildPost() {

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

  const [input, setInput] = useState('');

  const [deadline, setDeadline] = useState('2021-05-24T10:30');

  const [showFileUploadModal, setShowFileUploadModal] = useState(false);

  const [urlFile, setUrlFile] = useState('');

  const [selectedValue, setSelectedValue] = React.useState('post');

  const channelId = useSelector(selectChannelId)

  const groupId = useSelector(selectGroupId)

  const sendPost = (e) => {

    e.preventDefault();

    const time = firebase.firestore.FieldValue.serverTimestamp()

    if (input) {

      db.collection('channels').doc(channelId).collection('posts').add({ email: user.email, time: time, input: input, postType: selectedValue, deadline: deadline, urlFile: urlFile }).then(function (docRef) {

        if (selectedValue === 'task') {

          db.collection('groups').doc(groupId).collection('tasks').add({ channelId: channelId, postId: docRef.id, time: time })

        }

        else if (selectedValue === 'deadline') {

          db.collection('groups').doc(groupId).collection('deadlines').add({ channelId: channelId, postId: docRef.id, time: time })

        }

        setInput('');

        setUrlFile('');

      })

    }

  };

  const handleChange = (event) => {

    setSelectedValue(event.target.value);

  };

  return (

    <div className='channelBuildPost'>

      <div className='channelBuildPost\_inputContainer'>

        <div className='channelBuildPost\_input'>

          {/\* <CreateIcon /> \*/}

          <form>

            <TextField

              value={input}

              onChange={(e) => setInput(e.target.value)}

              id='outlined-multiline-static'

              multiline

              rows={4}

              placeholder='Create post...'

              variant='outlined'

            />

            <div className='channelBuildPost\_options'>

              <FormControl component='fieldset'>

                <RadioGroup

                  row

                  aria-label='position'

                  name='position'

                  defaultValue='top'

                >

                  <FormControlLabel

                    value='top'

                    control={

                      <Radio

                        color='primary'

                        checked={selectedValue === 'post'}

                        onChange={handleChange}

                        value='post'

                        name='radio-button-demo'

                        inputProps={{ 'aria-label': 'A' }}

                      />

                    }

                    label='Post'

                  />

                  <FormControlLabel

                    value='start'

                    control={

                      <Radio

                        color='primary'

                        checked={selectedValue === 'task'}

                        onChange={handleChange}

                        value='task'

                        name='radio-button-demo'

                        inputProps={{ 'aria-label': 'B' }}

                      />

                    }

                    label='Task'

                  />

                  <FormControlLabel

                    value='bottom'

                    control={

                      <Radio

                        color='primary'

                        checked={selectedValue === 'deadline'}

                        onChange={handleChange}

                        value='deadline'

                        name='radio-button-demo'

                        inputProps={{ 'aria-label': 'C' }}

                      />

                    }

                    label='Deadline'

                  />

                </RadioGroup>

              </FormControl>

            </div>

            <div className='channelBuildPost\_controllers'>

              <div

                className='fileChooser'

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

              >

                <InputOption title='File' Icon={AttachFileOutlinedIcon} />

              </div>

              {showFileUploadModal && (

                <FileAttachModal

                  onClose={() => setShowFileUploadModal(false)} setUrlFile={setUrlFile}

                />

              )}

              {selectedValue === 'deadline' && (

                <TextField

                  value={deadline}

                  onChange={(e) => setDeadline(e.target.value)}

                  id='datetime-local'

                  type='datetime-local'

                  InputLabelProps={{

                    shrink: true,

                  }}

                />

              )}

            </div>

            <div className='formBtn'>

              <Button onClick={sendPost} variant='contained' color='primary'>

                Post

              </Button>

            </div>

          </form>

        </div>

      </div>

    </div>

  );

}

export default ChannelBuildPost;

#### ChannelHeader.js

import React from 'react'

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

import './ChannelHeader.css'

function ChannelHeader({ name, description, urlPic }) {

  return (

    <div className='channelHeader'>

      <div className='channelHeader\_body'>

        <div className='channelHeader\_name'>

          <h1>{name}</h1>

          <div className='channelHeader\_avatar'>

            <Avatar src={urlPic}>{name[0]}</Avatar>

          </div>

        </div>

        <p>{description}</p>

      </div>

    </div>

  )

}

export default ChannelHeader

#### ChannelPage.js

import React from 'react';

import ChannelBuildPost from './ChannelBuildPost';

import ChannelHeader from './ChannelHeader';

import './ChannelPage.css';

import PostItem from './PostItem';

import FlipMove from 'react-flip-move';

import { useSelector } from 'react-redux';

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

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

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

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

function ChannelPage() {

  const [user] = useAuthState(auth);

  const channelId = useSelector(selectChannelId);

  const [channel] = useDocument(

    channelId && db.collection('channels').doc(channelId)

  );

  const [channelPosts] = useCollection(

    channelId &&

    db

      .collection('channels')

      .doc(channelId)

      .collection('posts')

      .orderBy('time', 'desc')

  );

  return (

    <div className='channelPage'>

      {channel && (

        <>

          <ChannelHeader

            name={channel?.data()?.name}

            description={channel?.data()?.description}

            urlPic={channel?.data()?.urlPic}

          />

          {channel?.data()?.creator === user.email && <ChannelBuildPost />}

          <FlipMove>

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

              const {

                email,

                deadline,

                input,

                postType,

                time,

                urlFile,

              } = doc.data();

              return (

                <PostItem

                  key={doc.id}

                  name={channel?.data()?.name}

                  urlPic={channel?.data()?.urlPic}

                  time={time}

                  message={input}

                  deadline={deadline}

                  admin={email}

                  postType={postType}

                  urlFile={urlFile}

                  postId={doc.id}

                />

              );

            })}

          </FlipMove>

        </>

      )}

    </div>

  );

}

export default ChannelPage;

#### Deadlines.js

import React, { useState } from 'react';

import './ChannelPage.css';

import TaskItem from './TaskItem';

import FlipMove from 'react-flip-move';

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

import { useSelector } from 'react-redux';

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

import { db } from '../../firebase';

////

function Deadlines() {

  const groupId = useSelector(selectGroupId);

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

  const [deadlines] = useCollection(

    groupId &&

      db

        .collection('groups')

        .doc(groupId)

        .collection('deadlines')

        .orderBy('time', 'desc')

  );

  return (

    <div className='channelPage'>

      {/\* <FlipMove> \*/}

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

        const { channelId, postId } = doc.data();

        return <TaskItem key={postId} channelId={channelId} postId={postId} />;

      })}

      {/\* </FlipMove> \*/}

    </div>

  );

}

export default Deadlines;

#### FileAttachModal.js

import React, { useState } from 'react';

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

import ReactDom from 'react-dom';

import PublishOutlinedIcon from '@material-ui/icons/PublishOutlined';

import { firebaseApp } 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 FileAttachModal({ onClose, setUrlFile }) {

  const [file, setFileName] = useState(null);

  const onChoosedFile = (e) => {

    const file = e.target.files[0];

    setFileName(file);

  }

  const onFileUploaded = (e) => {

    e.preventDefault();

    onClose();

    const storageRef = firebaseApp.storage().ref();

    const fileRef = storageRef.child(new Date().getTime() + file.name)

    fileRef.put(file).then((snapshot) => {

      snapshot.ref.getDownloadURL().then(function (downloadURL) {

        setUrlFile(downloadURL)

      });

    })

  };

  return ReactDom.createPortal(

    <>

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

      <div style={MODAL\_STYLES}>

        <div className='addChennal'>

          <form>

            <h3>Attach an assignment</h3>

            <p>

              Choose a file, which you want to attach it with your post!

            </p>

            <div className='example-1'>

              <div className='form-group'>

                <label className='label'>

                  <i className='material-icons'>

                    <PublishOutlinedIcon />

                  </i>

                  <span className='title'>Upload file</span>

                  <input type='file' onChange={onChoosedFile} />

                </label>

              </div>

            </div>

            <div className="fileName">

              {file && <input value={file.name} readOnly type='text' style={{ border: 'none', borderBottom: '1px solid lightGray', fontSize: '16px', width: '270px' }} />}

            </div>

            <button onClick={onFileUploaded}>Attach</button>

          </form>

        </div>

      </div>

    </>,

    document.getElementById('portal')

  );

}

export default FileAttachModal;

#### HomePage.js

import React from 'react';

import ReactPlayer from "react-player";

import './HomePage.css'

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

import styled from 'styled-components'

function HomePage() {

  return (

    <div className='homePage'  >

      <h1>Overview</h1>

      <p>The application is intended for creating a convenient and comfortable environment for education.

Teachers can send assignments and accept work done, share materials with students. Students can watch assignments, take assignments, and communicate about their studies.</p>

      <div className="seeIt"> <p>See it in action:</p></div>

      <ReactPlayer width='100%'

        url="https://youtu.be/8gNuXqtUXBs"

      />

      <h1>Developers</h1>

      <div className="developers\_pics">

        <div className="name"><HeaderAvatar src='https://i.postimg.cc/7YyxLcG4/photo-2021-04-11-14-33-57.jpg' alt='Yanal Yahya' />

          <h3>Yanal Yahya</h3>

          <p>yyahya@edu.hse.ru</p>

        </div>

        <div className="name">

          <HeaderAvatar src='https://sun9-69.userapi.com/impg/37ZebOIIOMFLh-bGQ3Zz2UdWhqgq44fQWMKDsw/gCon1x9pfM0.jpg?size=810x1080&quality=96&sign=e586b5ad83f8e544e9587d9164575b8b&type=album' alt='Bakytbek uulu Nurzhigit' />

          <h3>Bakytbek uulu Nurzhigit</h3>

          <p>nbakytbekuulu\_1@edu.hse.ru</p>

        </div>

        <div className="name">

          <HeaderAvatar src='https://sun9-46.userapi.com/impg/806bSU2Ogxd9Sa-cNRgfheo3fytaTB4w9f-6lw/s-t5x89toRg.jpg?size=502x502&quality=96&sign=0e21a9872223a602076fa4ed38945d1a&type=album' alt='Dzhaparov Emirkhan' />

          <h3>Dzhaparov Emirkhan</h3>

          <p>emdzhaparov@edu.hse.ru</p>

        </div>

      </div>

    </div>

  );

}

export default HomePage;

const HeaderAvatar = styled(Avatar)`

  cursor: pointer;

  :hover {

    opacity: 0.8;

  }

  width:8rem;

  height:8rem

`

#### InputOption.js

import React from 'react';

import './InputOption.css';

function InputOption({ Icon, title, color }) {

  return (

    <div className='inputOption'>

      <Icon style={{ color: color }} />

      <h5>{title}</h5>

    </div>

  );

}

export default InputOption;

#### JoinPage.js

import React from 'react';

import './JoinPage.css';

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

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

import { useHistory, useParams } from 'react-router';

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

import Login from '../Login';

import styled from 'styled-components';

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

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

import { useDispatch } from 'react-redux';

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

import Button from '@material-ui/core/Button';

import { queryAllByAltText } from '@testing-library/dom';

function JoinPage() {

  const dispatch = useDispatch();

  const [user] = useAuthState(auth);

  const { groupId } = useParams();

  const history = useHistory();

  console.log(groupId);

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

  if (!user) {

    return <Login />;

  }

  if (!group || !group?.data()) {

    return <h3>Group doesn't exist...</h3>;

  }

  const join = (e) => {

    e.preventDefault();

    db.collection('users')

      .doc(user.email)

      .collection('groups-id')

      .where('id', '==', groupId)

      .get()

      .then((query) => {

        if (query.size === 0) {

          db.collection('users')

            .doc(user.email)

            .collection('groups-id')

            .add({ id: groupId });

        }

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

      });

    history.push('/');

  };

  return (

    <div className='join'>

      <HeaderAvatar src={group?.data()?.urlPic} alt={group?.data()?.name[0]} />

      <h3>{group.data().name} </h3>

      <Button onClick={join} variant='contained' color='primary'>

        Join

      </Button>

    </div>

  );

}

export default JoinPage;

const HeaderAvatar = styled(Avatar)`

  cursor: pointer;

  :hover {

    opacity: 0.8;

  }

  width: 10rem;

  height: 10rem;

  margin-bottom: 2rem;

`;

#### PostItem.js

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

import PublishOutlinedIcon from '@material-ui/icons/PublishOutlined';

import AssignmentOutlinedIcon from '@material-ui/icons/AssignmentOutlined';

import PictureAsPdfOutlinedIcon from '@material-ui/icons/PictureAsPdfOutlined';

import NotificationsNoneOutlinedIcon from '@material-ui/icons/NotificationsNoneOutlined';

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

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

import {

  ChatOutlined,

  SendOutlined,

  ShareOutlined,

  ThumbUpAltOutlined,

} from '@material-ui/icons';

import React, { forwardRef, useState } from 'react';

import InputOption from './InputOption';

import './PostItem.css';

import SubmitPopUp from './SubmitPopUp';

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

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

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

// const { email, deadline, input, postType, time, urlFile } = doc.data()

const PostItem = forwardRef(

  (

    { name, urlPic, time, message, deadline, admin, postType, urlFile, postId },

    ref

  ) => {

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

    const [submitPopUp, setSubmitPopUp] = useState(false);

    const isAdmin = user.email === admin;

    const originalDate = new Date(time?.toDate());

    deadline = postType === 'deadline' ? deadline : '';

    const date =

      '' +

      originalDate.getDate() +

      '/' +

      originalDate.getMonth() +

      '/' +

      originalDate.getFullYear() +

      ' ' +

      originalDate.getHours() +

      ':' +

      originalDate.getMinutes();

    return (

      <div ref={ref} className='post'>

        <div className='post\_\_header'>

          <Avatar src={urlPic}>{name[0]}</Avatar>

          <div className='post\_\_info'>

            <h2>{name}</h2>

            <p>{date}</p>

          </div>

          <div className='postType'>

            {postType === 'post' ? (

              <NotificationsNoneOutlinedIcon />

            ) : postType === 'task' ? (

              <ListAltIcon />

            ) : (

              <WhatshotIcon />

            )}

          </div>

        </div>

        <div className='post\_\_body'>

          <p>{message}</p>

          {urlFile && (

            <p>

              <strong>Assignment: </strong>{' '}

              <a target='\_blank' rel='noreferrer noopener' href={urlFile}>

                link

              </a>

            </p>

          )}

          {deadline && postType === 'deadline' && (

            <p>

              <strong>Deadline: </strong> {deadline}

            </p>

          )}

        </div>

        <div className='post\_\_buttons'>

          <InputOption Icon={ThumbUpAltOutlined} title='Like' color='gray' />

          <InputOption Icon={ChatOutlined} title='Comment' color='gray' />

          {deadline && !isAdmin && (

            <div onClick={() => setSubmitPopUp(true)} className='submitWork'>

              <InputOption

                Icon={PublishOutlinedIcon}

                title='Submit'

                color='gray'

              />

            </div>

          )}

          {submitPopUp && <SubmitPopUp postId={postId} onClose={() => setSubmitPopUp(false)} />}          {deadline && isAdmin && (

            <Link to={'/check/' + postId} style={{ textDecoration: 'none' }}>

              <InputOption

                Icon={AssignmentOutlinedIcon}

                title='Submitted works'

                color='gray'

              />

            </Link>

          )}

        </div>

      </div>

    );

  }

);

export default PostItem;

#### SubmitPopUp.js

import React, { useState } from 'react';

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

import ReactDom from 'react-dom';

import PublishIcon from '@material-ui/icons/Publish';

import PublishOutlinedIcon from '@material-ui/icons/PublishOutlined';

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

import { useSelector } from 'react-redux';

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

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

import firebase 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 SubmitPopUp({ postId, onClose, channekIdTask }) {

  const channelId = useSelector(selectChannelId);

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

  const [file, setFileName] = useState(null);

  const onChoosedFile = (e) => {

    const file = e.target.files[0];

    setFileName(file);

  };

  const onFileUploaded = (e) => {

    e.preventDefault();

    onClose();

    const storageRef = firebaseApp.storage().ref();

    const fileRef = storageRef.child(new Date().getTime() + file.name);

    fileRef.put(file).then((snapshot) => {

      snapshot.ref.getDownloadURL().then(function (downloadURL) {

        postId &&

          db

            .collection('channels')

            .doc(channelId ? channelId : channekIdTask)

            .collection('posts')

            .doc(postId)

            .collection('submitedWorks')

            .add({

              submitter: user.email,

              urlWork: downloadURL,

              time: firebase.firestore.FieldValue.serverTimestamp(),

              isSent: 0,

            });

        onClose();

      });

    });

  };

  return ReactDom.createPortal(

    <>

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

      <div style={MODAL\_STYLES}>

        <div className='addChennal'>

          <form>

            <h3>Submit task</h3>

            <p>Send your solution to be checked from the teacher!</p>

            <div className='example-1'>

              <div className='form-group'>

                <label className='label'>

                  <i className='material-icons'>

                    <PublishOutlinedIcon />

                  </i>

                  <span className='title'>Upload file</span>

                  <input type='file' onChange={onChoosedFile} />

                </label>

              </div>

            </div>

            <div className='fileName'>

              {file && (

                <input

                  value={file.name}

                  readOnly

                  type='text'

                  style={{

                    border: 'none',

                    borderBottom: '1px solid lightGray',

                    fontSize: '16px',

                    width: '270px',

                  }}

                />

              )}

            </div>

            <button onClick={onFileUploaded}>Send</button>

          </form>

        </div>

      </div>

    </>,

    document.getElementById('portal')

  );

}

export default SubmitPopUp;

#### TaskItem.js

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

import PublishOutlinedIcon from '@material-ui/icons/PublishOutlined';

import AssignmentOutlinedIcon from '@material-ui/icons/AssignmentOutlined';

import PictureAsPdfOutlinedIcon from '@material-ui/icons/PictureAsPdfOutlined';

import NotificationsNoneOutlinedIcon from '@material-ui/icons/NotificationsNoneOutlined';

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

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

import {

  ChatOutlined,

  SendOutlined,

  ShareOutlined,

  ThumbUpAltOutlined,

} from '@material-ui/icons';

import InputOption from './InputOption';

import './PostItem.css';

import SubmitPopUp from './SubmitPopUp';

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

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

import { useState } from 'react';

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

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

function TaskItem({ channelId, postId }) {

  // name, urlPic, time, message, deadline, admin, postType, urlFile

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

  const [submitPopUp, setSubmitPopUp] = useState(false);

  const [channel] = useDocument(

    channelId && db.collection('channels').doc(channelId)

  );

  const [post] = useDocument(

    channelId &&

      postId &&

      db.collection('channels').doc(channelId).collection('posts').doc(postId)

  );

  if (!post || !channel) {

    return null;

  }

  const { name, urlPic } = channel && channel?.data();

  var { email, deadline, input, postType, time, urlFile } =

    post && post?.data();

  deadline = postType === 'deadline' ? deadline : '';

  const isAdmin = user.email === email;

  const originalDate = new Date(time?.toDate());

  const date =

    '' +

    originalDate.getDate() +

    '/' +

    originalDate.getMonth() +

    '/' +

    originalDate.getFullYear() +

    ' ' +

    originalDate.getHours() +

    ':' +

    originalDate.getMinutes();

  return (

    name && (

      <div className='post'>

        <div className='post\_\_header'>

          <Avatar src={urlPic}>{name[0]}</Avatar>

          <div className='post\_\_info'>

            <h2>{name}</h2>

            <p>{date}</p>

          </div>

          <div className='postType'>

            {postType === 'post' ? (

              <NotificationsNoneOutlinedIcon />

            ) : postType === 'task' ? (

              <ListAltIcon />

            ) : (

              <WhatshotIcon

                style={{

                  color: '#E25822',

                }}

              />

            )}

          </div>

        </div>

        <div className='post\_\_body'>

          <p>{input}</p>

          {urlFile && (

            <p>

              <strong>Assignment: </strong>{' '}

              <a target='\_blank' rel='noreferrer noopener' href={urlFile}>

                link

              </a>

            </p>

          )}

          {deadline && postType === 'deadline' && (

            <p>

              <strong>Deadline: </strong> {deadline}

            </p>

          )}

        </div>

        <div className='post\_\_buttons'>

          <InputOption Icon={ThumbUpAltOutlined} title='Like' color='gray' />

          <InputOption Icon={ChatOutlined} title='Comment' color='gray' />

          {deadline && !isAdmin && (

            <div onClick={() => setSubmitPopUp(true)} className='submitWork'>

              <InputOption

                Icon={PublishOutlinedIcon}

                title='Submit'

                color='gray'

              />

            </div>

          )}

          {submitPopUp && (

            <SubmitPopUp

              postId={postId}

              channelIdTask={channelId}

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

            />

          )}

          {deadline && isAdmin && (

            <Link to={'/check/' + postId} style={{ textDecoration: 'none' }}>

              <InputOption

                Icon={AssignmentOutlinedIcon}

                title='Submitted works'

                color='gray'

              />

            </Link>

          )}

        </div>

      </div>

    )

  );

}

export default TaskItem;

#### Tasks.js

import React, { useState } from 'react';

import './ChannelPage.css';

import TaskItem from './TaskItem';

import FlipMove from 'react-flip-move';

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

import { useSelector } from 'react-redux';

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

import { db } from '../../firebase';

////

function Tasks() {

  const groupId = useSelector(selectGroupId);

  const [tasks] = useCollection(

    groupId &&

      db

        .collection('groups')

        .doc(groupId)

        .collection('tasks')

        .orderBy('time', 'desc')

  );

  return (

    <div className='channelPage'>

      {/\* <FlipMove> \*/}

      {tasks?.docs?.map((doc) => {

        const { channelId, postId } = doc.data();

        return <TaskItem key={postId} channelId={channelId} postId={postId} />;

      })}

      {/\* </FlipMove> \*/}

    </div>

  );

}

export default Tasks;

### sidebar

#### 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.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.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.js

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;

`;

#### StudyGoLogo.js

import React from 'react'

function StudyGoLogo() {

  return (

    <div className='studyGoLogo'>

      <h1>StudyGo</h1>

    </div>

  )

}

export default StudyGoLogo

#### 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

`

### user-setting

#### 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

`

### Login.js

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

import React from 'react';

import styled from 'styled-components';

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

function Login() {

  const signIn = (e) => {

    e.preventDefault();

    auth.signInWithPopup(provider).catch((error) => alert(error.message));

  };

  return (

    <LoginContainer>

      <LoginInner>

        <img src='https://i.pinimg.com/originals/20/7a/2b/207a2b17fb9dfb995097e923c53eec1f.gif' />

        <h2>Sign in StudyGo</h2>

        <Button type='submit' onClick={signIn}>

          Sign in with Google

        </Button>

      </LoginInner>

    </LoginContainer>

  );

}

export default Login;

const LoginContainer = styled.div`

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

  display: grid;

  place-items: center;

  height: 100vh;

`;

const LoginInner = styled.div`

  padding: 5rem;

  text-align: center;

  background-color: white;

  border-radius: 10px;

  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.12), 0 1px 2px rgba(0, 0, 0, 0.24);

  > img {

    object-fit: contain;

    height: 10rem;

    margin-bottom: 2rem;

  }

  > button {

    margin-top: 2rem;

    text-transform: inherit !important;

    background-color: #0a8d48 !important;

    color: white;

  }

`;

### MyAppBar.js

import React, { useState } from 'react';

import CloseIcon from '@material-ui/icons/Close';

import Button from '@material-ui/core/Button';

import MenuIcon from '@material-ui/icons/Menu';

import '../styles/Sidebar.css';

import Sidebar from './Sidebar';

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

import { auth } from '../firebase';

import styled from 'styled-components';

export default function MyAppBar() {

  const [user] = useAuthState(auth);

  const [sidebar, setSidebar] = useState(false);

  const [sidebarUserSetting, setSidebarUserSetting] = useState(false);

  const showSidebar = () => setSidebar(!sidebar);

  return (

    <div>

      <AppBarContainer>

        {sidebar ? (

          <CloseIcon onClick={showSidebar} />

        ) : (

          <MenuIcon onClick={showSidebar} />

        )}

        <Button color='inherit' onClick={() => auth.signOut()}>

          Sign out

        </Button>

      </AppBarContainer>

      <Sidebar sidebar={sidebar} showSidebar={showSidebar} />

    </div>

  );

}

const AppBarContainer = styled.div`

  display: flex;

  width: 100%;

  height: 4rem;

  position: fixed;

  align-items: center;

  justify-content: space-between;

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

  color: white;

  padding-left: 0.5rem;

  border-bottom: 1px solid lightgray;

`;

### Sidebar.js

import React, { useState } from 'react';

import '../styles/Sidebar.css';

import CloseIcon from '@material-ui/icons/Close';

import Top from './sidebar/Top';

import Middle from './sidebar/Middle';

import Bottom from './sidebar/Bottom';

import ReactDom from 'react-dom';

const OVERLAY\_STYLES = {

  position: 'fixed',

  marginLeft: 'var(--sidebar-width)',

  top: 0,

  left: 0,

  right: 0,

  bottom: 0,

  zIndex: 101,

};

const MODAL\_STYLES = {

  position: 'fixed',

  marginTop: 'var(--app-bar-height)',

  top: '50%',

  left: '50%',

  width: '100%',

  height: '100%',

  backgroundColor: 'rgba(0, 0, 0, .5)',

  transform: 'translate(-50%, -50%)',

  zIndex: 100,

};

function Sidebar({ sidebar, onClose }) {

  return sidebar && ReactDom.createPortal(

    <>

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

      <div style={MODAL\_STYLES}> <nav className={sidebar ? 'nav-menu active' : 'nav-menu'}>

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

          <Top />

          <Middle onClose={onClose} />

          <Bottom onClose={onClose} />

        </div>

      </nav>

      </div>

    </>

    ,

    document.getElementById('portal'));

}

export default Sidebar;

## features

### appSlice.js

import { createSlice } from '@reduxjs/toolkit';

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

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

const initialState = {

  groupId: null,

  channelId: null,

};

export const appSlice = createSlice({

  name: 'app',

  initialState,

  // The `reducers` field lets us define reducers and generate associated actions

  reducers: {

    enterGroup: (state, action) => {

      state.groupId = action.payload.groupId;

    },

    enterChannel: (state, action) => {

      state.channelId = action.payload.channelId;

    },

  },

});

export const { enterGroup: enterGroup } = appSlice.actions;

export const selectGroupId = (state) => state.app.groupId;

export const { enterChannel: enterChannel } = appSlice.actions;

export const selectChannelId = (state) => state.app.channelId;

export default appSlice.reducer;

## App.js

import React, { useState } from 'react';

import './styles/Sidebar.css';

import './App.css';

import { BrowserRouter as Router, Switch, Route, Link } from 'react-router-dom';

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

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

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

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

import Button from '@material-ui/core/Button';

import CloseIcon from '@material-ui/icons/Close';

import MenuIcon from '@material-ui/icons/Menu';

import Login from './components/Login';

import Schedule from './components/pages/Schedule';

import Tasks from './components/pages/Tasks';

import Deadlines from './components/pages/Deadlines';

import HomePage from './components/pages/HomePage';

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

import styled from 'styled-components';

import Sidebar from './components/Sidebar';

import StudyGoLogo from './components/sidebar/StudyGoLogo';

import UserSettingSidebar from './components/user-setting/UserSettingSidebar';

import GroupSettingMenu from './components/group-setting/GroupSettingMenu';

import CheckPage from './components/check-works/CheckPage';

import AssignmentReturnedOutlinedIcon from '@material-ui/icons/AssignmentReturnedOutlined';

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

import {

  enterChannel,

  enterGroup,

  selectChannelId,

  selectGroupId,

} from './features/appSlice';

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

import JoinPage from './components/pages/JoinPage';

import Feedbacks from './components/feedbacks/Feedbacks';

import MenuBookIcon from '@material-ui/icons/MenuBook';

function App() {

  const dispatch = useDispatch();

  let history = useHistory();

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

  const usersRef = db.collection('users').doc(user?.email);

  usersRef.get().then((docSnapshot) => {

    if (!docSnapshot.exists && user) {

      usersRef.set({ name: user.displayName }); // create the document

    }

  });

  const groupId = useSelector(selectGroupId);

  const channelId = useSelector(selectChannelId);

  if (!groupId && user) {

    const docRef = db.collection('users').doc(user.email);

    docRef

      .get()

      .then((doc) => {

        if (doc.exists) {

          const id = doc.data().currentGroup;

          if (id) {

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

          }

        } else {

          // doc.data() will be undefined in this case

          console.log('No such document!');

        }

      })

      .catch((error) => {

        console.log('Error getting document:', error);

      });

  }

  if (!channelId && !groupId && user) {

    const docRef = db.collection('users').doc(user.email);

    docRef

      .get()

      .then((doc) => {

        if (doc.exists) {

          const id = doc.data().currentChannel;

          if (id) {

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

          }

        } else {

          // doc.data() will be undefined in this case

          console.log('No such document!');

        }

      })

      .catch((error) => {

        console.log('Error getting document:', error);

      });

  }

  const [groupDetails] = useDocument(

    groupId && db.collection('groups').doc(groupId)

  );

  const [sidebar, setSidebar] = useState(false);

  const [sidebarUserSetting, setSidebarUserSetting] = useState(false);

  const [groupSetting, setGroupSetting] = useState(false);

  const showSidebar = () => {

    setSidebar(!sidebar);

    setSidebarUserSetting(false);

    setGroupSetting(false);

  };

  const showUserSetting = () => {

    setSidebarUserSetting(!sidebarUserSetting);

    setGroupSetting(false);

    setSidebar(false);

  };

  const showGroupSetting = () => {

    setGroupSetting(!groupSetting);

    setSidebarUserSetting(false);

    setSidebar(false);

  };

  if (loading) {

    return <></>;

  }

  if (error) {

    return (

      <>

        <h3>Auth error...</h3>

      </>

    );

  }

  const showFeedbacks = () => {

    console.log('his');

    history?.push('/feedbacks');

  };

  return (

    <div className='app' style={{ overflowY: 'auto' }}>

      <CssBaseline />

      <Router>

        {!user ? (

          <Login />

        ) : (

          <>

            <AppBarContainer>

              {sidebar ? (

                <div className='sidebar-control'>

                  <CloseIcon onClick={showSidebar} />

                </div>

              ) : (

                <div className='sidebar-control'>

                  <MenuIcon onClick={showSidebar} />

                </div>

              )}

              <HeaderAvatar

                style={{ marginLeft: '1rem', marginRight: 'auto' }}

                onClick={() => showGroupSetting()}

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

                alt='БПИ197'

              />

              <StudyGoLogo

                style={{ marginLeft: 'auto', marginRight: 'auto' }}

              />

              <Link

                to='/feedbacks'

                style={{ marginLeft: 'auto', marginRight: '1rem' }}

              >

                <MenuBookIcon

                  style={{ textDecoration: 'none', color: 'white' }}

                />

              </Link>

              <HeaderAvatar

                onClick={() => showUserSetting()}

                src={user?.photoURL}

                alt={user?.displayName}

              />

              {/\* <Button color='inherit' onClick={() => auth.signOut()}>

                Sign out

              </Button> \*/}

            </AppBarContainer>

            <Sidebar sidebar={sidebar} onClose={() => setSidebar(false)} />

            <UserSettingSidebar

              userSettingSidebar={sidebarUserSetting}

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

            />

            <GroupSettingMenu

              groupSetting={groupSetting}

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

            />

            <div

              className={

                sidebar ? 'content-container-moved' : 'content-container'

              }

            >

              <Switch>

                <Route path='/' exact component={HomePage} />

                <Route path='/schedule' component={Schedule} />

                <Route path='/tasks' component={Tasks} />

                <Route path='/deadlines' component={Deadlines} />

                <Route path='/channel' component={ChannelPage} />

                <Route path='/check/:postId' component={CheckPage} />

                <Route path='/join/:groupId' component={JoinPage} />

                <Route path='/feedbacks' component={Feedbacks} />

              </Switch>

            </div>

          </>

        )}

      </Router>

    </div>

  );

}

export default App;

const AppBarContainer = styled.div`

  display: flex;

  width: 100%;

  height: var(--app-bar-height);

  position: fixed;

  align-items: center;

  justify-content: space-between;

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

  color: white;

  padding-left: 0.5rem;

  border-bottom: 1px solid lightgray;

`;

const HeaderAvatar = styled(Avatar)`

  cursor: pointer;

  :hover {

    opacity: 0.8;

  }

  margin-right: 1rem;

`;

## firebase.js

import firebase from 'firebase'

const firebaseConfig = {

  apiKey: 'AIzaSyBoQX3twU5\_zKzqsOF871\_Cn3LYHWsZzns',

  authDomain: 'yen-studygo.firebaseapp.com',

  projectId: 'yen-studygo',

  storageBucket: 'yen-studygo.appspot.com',

  messagingSenderId: '549095636780',

  appId: '1:549095636780:web:9a6528517a2d2697787689',

}

const firebaseApp = firebase.initializeApp(firebaseConfig)

const db = firebaseApp.firestore()

const auth = firebase.auth()

const provider = new firebase.auth.GoogleAuthProvider()

export { auth, db, provider, firebaseApp }

## index.js

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import { store } from './app/store';

import { Provider } from 'react-redux';

import \* as serviceWorker from './serviceWorker';

import { createMuiTheme } from '@material-ui/core/styles';

import { ThemeProvider } from '@material-ui/core/styles';

const theme = createMuiTheme({

  palette: {

    primary: {

      main: '#4a154b',

    },

  },

});

ReactDOM.render(

  <React.StrictMode>

    <Provider store={store}>

      <ThemeProvider theme={theme}>

        <App />

      </ThemeProvider>

    </Provider>

  </React.StrictMode>,

  document.getElementById('root')

);

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

serviceWorker.unregister();

# 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 |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |