**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 | | **Окна пользовательского 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  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Джапаров Э. М. /  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 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 | | **Окна пользовательского интерфейса веб-приложения “StudyGo” для поддержки образовательного процесса**  **Текст программы**  **RU.17701729.06.02-01 12 01-1**  **Листов 40** | | | |
|
|
|
|  | | |  |

**Москва 2021**

Оглавление

[1 Components 3](#_Toc72502217)

[1.1 pages 3](#_Toc72502218)

[1.1.1 ChannelBuildPost.css 3](#_Toc72502219)

[1.1.2 ChannelBuildPost.js 4](#_Toc72502220)

[1.1.3 ChannelHeader.css 8](#_Toc72502221)

[1.1.4 ChannelHeader.js 8](#_Toc72502222)

[1.1.5 ChannelPage.css 9](#_Toc72502223)

[1.1.6 ChannelPage.js 9](#_Toc72502224)

[1.1.7 Deadlines.js 10](#_Toc72502225)

[1.1.8 FileAttachModal.js 11](#_Toc72502226)

[1.1.9 HomePage.css 13](#_Toc72502227)

[1.1.10 HomePage.js 13](#_Toc72502228)

[1.1.11 InputOption.css 15](#_Toc72502229)

[1.1.12 InputOption.js 15](#_Toc72502230)

[1.1.13 JoinPage.css 15](#_Toc72502231)

[1.1.14 JoinPage.js 16](#_Toc72502232)

[1.1.15 PostItem.css 17](#_Toc72502233)

[1.1.16 PostItem.js 18](#_Toc72502234)

[1.1.17 SubmitPopUp.js 21](#_Toc72502235)

[1.1.18 TaskItem.js 23](#_Toc72502236)

[1.1.19 Tasks.js 26](#_Toc72502237)

[1.2 check-works 27](#_Toc72502238)

[1.2.1 Submits.js 27](#_Toc72502239)

[1.2.2 WorkItem.css 27](#_Toc72502240)

[1.2.3 WorkItem.js 29](#_Toc72502241)

[1.2.4 WorkItemModal 31](#_Toc72502242)

[1.3 feedbacks 35](#_Toc72502243)

[1.3.1 FeedbackItem.js 35](#_Toc72502244)

[1.3.2 Feedbacks.js 37](#_Toc72502245)

[Список используемой литературы 38](#_Toc72502246)

[Лист регистрации изменений. 39](#_Toc72502247)

# Components

## pages

### ChannelBuildPost.css

#datetime-local {

font-size: small;

}

.MuiFormControlLabel-root > span {

font-size: small;

}

.channelBuildPost\_inputContainer {

background-color: white;

padding: 0;

border-radius: 4px;

margin-top: 1rem;

border: 1px solid lightgray;

}

.channelBuildPost\_input > form > .MuiTextField-root {

border: none;

flex: 1;

margin-left: 10px;

outline-width: 0;

font-weight: 600;

margin-right: 1rem;

width: 98%;

}

.channelBuildPost\_input {

display: flex;

padding: 10px;

color: gray;

}

.channelBuildPost\_input > form {

width: 100%;

}

.channelBuildPost\_options {

display: flex;

}

.channelBuildPost\_controllers {

display: flex;

align-items: center;

}

.channelBuildPost\_controllers > .MuiTextField-root {

margin-left: auto;

}

.formBtn {

display: flex;

}

.formBtn > .MuiButtonBase-root {

margin-top: 10px;

margin-left: auto;

}

.formBtn > .MuiButtonBase-root > .MuiButton-label {

margin-left: 0;

}

.MuiInputBase-root.MuiOutlinedInput-root.MuiInputBase-formControl.MuiInputBase-multiline.MuiOutlinedInput-multiline {

padding: 0.3rem 1rem;

padding-right: 4px;

}

### 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.css

.channelHeader {

position: sticky;

margin-top: 1rem;

flex: 0.2;

background-color: white;

border-radius: 4px;

border: 1px solid lightgray;

height: fit-content;

padding-bottom: 10px;

}

.channelHeader\_name {

border-bottom: 1px solid lightgray;

display: flex;

justify-content: space-between;

}

.channelHeader\_name > h1 {

margin: 20px;

}

.channelHeader\_avatar {

margin: 20px;

}

p {

margin: 20px;

}

### 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.css

.channelPage {

margin-bottom: 1rem;

}

### 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.css

.developers\_pics {

display: flex;

justify-content: space-around;

align-items: center;

margin-top: 2rem;

}

.seeIt > p {

margin: 0;

}

.homePage > h1 {

margin-top: 3rem;

text-align: center;

color: var(--slack-color);

}

.name > h3 {

margin-top: 10px;

text-align: center;

}

.name > p {

margin: 0;

text-align: center;

color: gray;

font-size: small;

}

.name {

display: flex;

flex-direction: column;

align-items: center;

width: 12rem;

padding: 10px;

}

@media (max-width: 700px) {

.developers\_pics {

display: flex;

flex-direction: column;

}

}

### 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.css

.inputOption {

display: flex;

align-items: center;

color: gray;

margin-top: 10px;

padding: 0 10px;

cursor: pointer;

}

.inputOption:hover {

background-color: whitesmoke;

border-radius: 10px;

}

.inputOption > h5 {

margin-left: 3px;

}

### 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.css

.join {

/\* position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%); \*/

display: flex;

flex-direction: column;

align-items: center;

margin-top: 23%;

}

.join > h3 {

margin-bottom: 1rem;

}

.join > .MuiButtonBase-root {

width: 7rem;

}

.join > .MuiButtonBase-root > .MuiButton-label {

margin-left: 0;

}

### 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.css

.post {

border: 1px solid lightgray;

padding: 1rem;

margin-top: 1rem;

border-radius: 4px;

background-color: white;

}

.post\_\_header {

display: flex;

margin-bottom: 10px;

}

.post\_\_info {

margin-left: 10px;

}

.post\_\_header > p {

font-size: 12px;

margin-left: auto;

color: gray;

}

.post\_\_info > p {

font-size: 12px;

margin: 0;

color: gray;

}

.post\_\_body {

overflow-wrap: anywhere;

}

.post\_\_body > p {

margin: 1rem;

}

.post\_\_info > h2 {

font-size: 15px;

}

.post\_\_buttons {

border-top: 1px solid lightgray;

margin-top: 0;

display: flex;

}

.postType {

margin-right: 10px;

margin-left: auto;

}

### 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;

## check-works

### 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.css

.work-item-container {

margin-top: 1rem;

display: flex;

justify-content: space-between;

padding: 1rem;

background-color: white;

align-items: center;

border-radius: 4px;

}

.work-item-left {

display: flex;

align-items: center;

}

.work-item-info {

margin-left: 0.5rem;

}

.work-item-info > span {

margin: 0;

font-size: 0.7rem;

color: gray;

font-weight: 300;

}

.work-item-controllers {

display: flex;

vertical-align: center;

}

.work-item-controlls {

display: flex;

justify-content: flex-end;

}

.work-item-controllers > input {

margin-left: 1rem;

font-size: large;

width: 2.5rem;

border: none;

color: green;

text-align: center;

}

.work-item-controllers > .MuiSvgIcon-root {

margin-left: 1rem;

color: gray;

cursor: pointer;

}

.work-item-controllers > .MuiSvgIcon-root:hover {

color: #6d6969;

}

.work-item-modal {

width: 500px;

background-color: white;

}

.work-item-comment-and-mark {

padding: 0 5%;

padding-bottom: 1rem;

}

.work-item-mark {

display: inline;

}

.work-item-mark > input {

width: 3.5rem;

font-size: 1rem;

border: 1px solid lightgray;

margin-bottom: 1rem;

}

.work-item-comment > .MuiTextField-root {

width: 100%;

}

.work-item-controlls {

display: flex;

}

@media (max-width: 600px) {

.work-item-modal {

width: 300px;

}

}

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

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}

/>

)}

{/\* {submit.isSent != 1 && (

<SendOutlinedIcon onClick={() => sendReview()} />

)} \*/}

</div>

</div>

);

}

export default WorkItem;

### WorkItemModal

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 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';

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 TaskItem from '../pages/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 { 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

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