Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчет по лабораторной работе №8

Студент Смыслов Максим Алексеевич
фамилия, имя, отчество

Группа ИУ5-55Б

Студент 18.12.2021 **Смыслов М.А.**
подпись, дата фамилия, и.о.

Преподаватель 18.12.2021 **Гапанюк Ю.Е.**
подпись, дата фамилия, и.о.

2021г.

Описание задания:

1. Создать стартовый React-проект. Удалить неиспользуемый код. Организовать директории для страниц, компонентов, утилит и работы с сетью.
2. Организовать роутниг в веб-приложении.
3. Разработать базовые страницы, на которых будут отображаться сущности из выбранной вами предметной области:
    i. Стартовая страница.
    ii. Страница просмотра списка объектов.
    iii. Страница просмотра конкретного объекта.
4. Вынести переиспользуемые компоненты в отдельные файлы:
    i. Для навигации по приложению можно добавить header.
    ii. Для отображения дополнительной информации (данные о студенте и предметной области) можно использовать footer.
    iii. Источники ввода-вывода (поля ввода (inputs)/формы/текстовые блоки).
    iv. Переиспользуемые таблицы/гриды.
5. Добавить асинхронные запросы в разработанный API, чтобы страница получала данные с сервера.
6. Если в Вашем проекте реализована сложная логика работы с состоянием приложения, то рекомендуется добавить пользовательские хуки.
7. Страницы приложения должны хорошо отображаться как на больших, так и на маленьких экранах.

Текст программы:
About.js

```
import { observer } from 'mobx-react-lite'
import React from 'react'
import { Card, Container } from 'react-bootstrap'

const About = observer(() => {
    return (
        <Container>
            <Card className='align-items-center mt-3 mb-2'>

                    <h1 className='mt-2'>О сервисе</h1>
                    <div className='p-4 pt-0 pb-0' style={{fontSize:'20px', textAlign:'justify'}}>
                        <p >
                        <span style={{fontWeight:'bold'}}>Добрый день или вечер!</span>  Вы попали на сайт сервиса с очередями.
                        Для того, чтобы полностью пользоваться его функционалом пожалуйста зарегистрируйтесь или авторизуйтесь.
                        </p>
                        <h4>О функционале</h4>
                        <p>
                        Этот сервис поможет вам и вашим коллегам по несчастью с гордостью отстоять возникшую очередь.
                        Здесь вы можете найти нужную вам очередь и встать в неё.
                        Пожалуйста не забывайте выходить из неё когда закончите.
                        </p>
                        <h4>Примечание</h4>
                        <p>
                        Данный сайт пока находится в далёкой pre alpha версии,
                        и, если вы нашли какие-то недочёты, обязательно свяжитесь со мной.
                        </p>
                        <h4>Контакты</h4>
                        <p>
                            Почта: <span style={{fontStyle:'italic'}}>test@mail.ru</span><br></br>
                            Телефон: <span style={{fontStyle:'italic'}}>+7(999)555-55-55</span><br></br>

                        </p>
                    </div>

            </Card>
        </Container>
```

Create.js

```
import { observer } from 'mobx-react-lite'
import React, {useState} from 'react'
import { Container,Form,Card, Button } from 'react-bootstrap'
import { useHistory } from 'react-router-dom'
import { createQ } from '../http/queueAPI'
import { QUEUE_ROUTE } from '../utils/consts'

const clearInput = (idElem) =>{
    try {
        document.getElementById(idElem).value = ""
    } catch (error) {
       console.log(error)
    }
}

const Create = observer(() => {

    const [name, setName] = useState('')
    const [description, setDescription] = useState('')
    const history = useHistory()

    const checkInputs = () => {

        if (name && description) return false
        else return true
    }

    const create = async () => {
        try {
            const res = await createQ(name,description)
            history.push(QUEUE_ROUTE + `/${res.id}`)
        } catch (error) {
            alert(error.response.data.message)
        }

    }
```

```
return (
    <Container>
        <Card className='align-items-center mt-4 p-4'>
            <Form style={{width:'75%'}}>
                <Form.Label>Название</Form.Label>
                <Form.Control
                    value={name}
                    onChange={str=>setName(str.target.value)}
                    id='nameInput'
                    style={{width:'98.5%'}}
                    placeholder={'Введите название'}
                    className='mb-3'
                />
                <Form.Label>Описание</Form.Label>
                <Form.Control
                    value={description}
                    onChange={str=>setDescription(str.target.value)}
                    id='descriptionInput'
                    style={{width:'98.5%'}}
                    placeholder={'Напишите описание для вашей очереди'}
                    as="textarea"
                    className='mb-3'
                />
                <Container className='d-flex justify-content-end'>
                <Button

                        variant={'outline-secondary'}
                        className='m-2'
                        onClick={() => {
                            clearInput('descriptionInput')
                            clearInput('nameInput')
                        }}
                    >
                        Очистить
                </Button>
                <Button
```

Find.js

```
    const [number, setNumber] = useState('')
    const [foundVisible, setFoundVisible] = useState(false)

    const findByName = async () => {
        try {

            searchByName(name).then(data=>{
                queues.setQueues(data)
                console.log(queues.queues)
            })

        } catch (error) {
            alert(error.response.data.message)
        }
    }
    const findByNumber = async () => {
        try {

            searchById(number).then(data=>{
                queues.setQueues(data)
                console.log(queues.queues)
            })

        } catch (error) {
            alert(error.response.data.message)
        }
    }


    return (
        <Container >
            <Card className='align-items-center mt-4 p-4'>
                <h4 className='ml-0'>Поиск по номеру очереди</h4>
                <Form style={{width:'75%'}}>
                    <Form.Control
                        id='number'
                        value={number}
                        onChange={str=>setNumber(str.target.value)}
                        placeholder='Введите номер'
                        style={{width:'98.5%'}}
                    />
                    <Container className='d-flex justify-content-end'>
                    <Button
```

MyQs.js

```
const MyQs = observer(() => {

    const history = useHistory()
    const {queues,user} = useContext(Context)
    useEffect(() => {
        getQs(user.user.id).then((data)=>{
            try {
                queues.setQueues(data[0].queues)
                console.log(queues.queues)
            } catch (error) {

            }

        })
    }, [])

    return (
        <Container>
            <Card className='p-4 pt-2 mt-3'>
                <h1>Мои очереди</h1>
                {queues.queues.length > 0? queues.queues.map(data=>{
                    return(
                        <Card
                        style={{cursor:'pointer'}}
                        className='m-1'
                        key={data.id}
                        onClick={()=>history.push(QUEUE_ROUTE+ '/' + data.id)}
                        >
                            <h4 className='m-2 mt-0'>{data.name}</h4>
                            <h5 className='m-2 mt-0' style={{font:'italic 20px  Arial', opacity:'0.75'}}>{data.description}</h5>
                        </Card>
                    )
                }): <h3>У тебя нет очередей, попробуй
                    <Button
                        onClick={() => history.push(FIND_ROUTE)}
                    >поискать</Button>
                        </h3>}
            </Card>

        </Container>
    )
})

export default MyQs
```

Profile.js

```
import React, {useContext, useEffect, useState} from 'react'
import { Card, Container } from 'react-bootstrap'
import { useParams } from 'react-router-dom/cjs/react-router-dom.min'
import { Context } from '..'
import { getUser } from '../http/userAPI'

const Profile = () => {
    const {id} = useParams()
    const {curUsers} = useContext(Context)
    const [username, setUsername] = useState()

    useEffect(() => {
       getUser(id).then((data)=>{
        curUsers.setUsers(data)
        setUsername(curUsers.users.username)
        console.log(curUsers.users.username)
       })
    }, [curUsers, id])
    return (
        <Container>
            <Card className='mt-3 p-2'>
                <h1>Профиль пользователя</h1>
                <h2>{username}</h2>

            </Card>
        </Container>
    )
}

export default Profile
```

Queue.js

```
:onst Queue = observer(() => {

    const [nameQueue, setNameQueue] = useState('')
    const [descQueue, setDescQueue] = useState('')
    const {user, curUsers} = useContext(Context)
    const {id} = useParams()
    const history = useHistory()
    const update = async () => {
        let response = await fetch(process.env.REACT_APP_API_URL + 'api/queue?id='+id)

        let parse = await response.json()
       try {
           console.log(parse)
           if(!parse.message){
            document.getElementById('nobody').innerHTML =''
            if(!(parse[0].users.length === curUsers.users.length) || !(parse[0].users.every(function(element, index) {
                return curUsers.users.filter((data)=>{
                    return data.id === element.id
                }).length > 0
            }))) {
                clearInterval(timerId);
                curUsers.setUsers(parse[0].users)
            }
           } else {
               document.getElementById('nobody').innerHTML = "В очереди никого нет, станьте первым!"
               clearInterval(timerId);
                curUsers.setUsers([])
            }
       } catch (error) {
           return
       }




    }

    const getIn = async () => {
        try {
            const res = await addToQ(user.user.id, id)
            await update()
            return res
        } catch (error) {
            alert(error.response.data.message)
        }
```

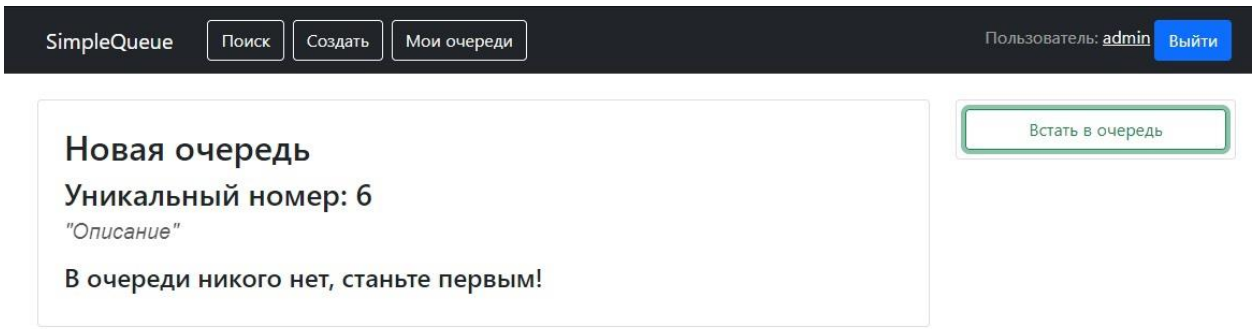
```
<Row>
    <Col className='mt-4' sm={9}>
        <Card className='p-4'>
            <h2 >{nameQueue}</h2>
            <h3>Уникальный номер: {id}</h3>
            {descQueue? <p style={{font:'italic 20px  Arial', opacity:'0.75'}}>"{descQueue}"</p> : null}

            {curUsers.users? curUsers.users.map((data,index)=>{
                return(
                    <Card
                    onClick={()=>history.push(PROFILE_ROUTE+'/'+data.id)}
                    key={index}
                    style={{height:'50px', background:'#212529', color:'white', textAlign:'center', fontSize:'20px', border:'2px solid white', cursor:'pointer'
                    >{`${index+1}. ${data.username}`}</Card>
                )
            }) : null}
            <h4 id='nobody'></h4>
        </Card>
    </Col>
    <Col className='mt-4' sm={3}>
        <Card className='p-2'>
            {(() =>{
                if(!user.isAuth) return <h4>Вы не можете встать в очередь, пожалуйста авторизуйтесь</h4>
                else if(curUsers.users.filter((data)=>{
                        return data.id === user.user.id
                    }).length > 0)
                    return(
                        <Button
                            variant='outline-danger'
                            onClick={getOut}
                        >
                            Выйти
                        </Button>
                    )
                    else return(
                        <Button
                            variant='outline-success'
                            onClick={getIn}
                        >
                            Встать в очередь
                        </Button>
                    )
```

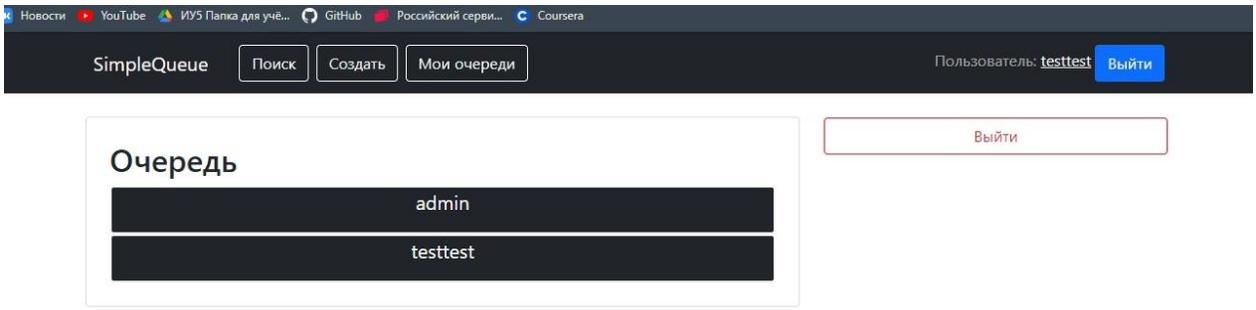
userAPI.js

```
import { $authHost, $host } from "./index"


export const getUsers = async (id) => {
    const {data} = await $host.get('api/queue',{
        params:{
            id
        }
    })
    return data

}
export const searchById = async (id) => {
    const {data} = await $host.get('api/queue/searchById',{
        params:{
            id
        }
    })
    return data

}
export const searchByName = async (name) => {
    const {data} = await $host.get('api/queue/searchByName',{
        params:{
            name
        }
    })
    return data

}

export const addToQ = async (userId, queueId) => {
    const {data} = await $authHost.post('api/queue/addToQ',{
            userId,
            queueId
        })
    return data

}
export const createQ = async (name, description) => {
```

queueAPI.js

```
import { $authHost, $host } from "./index"


export const getUsers = async (id) => {
    const {data} = await $host.get('api/queue',{
        params:{
            id
        }
    })
    return data

}
export const searchById = async (id) => {
    const {data} = await $host.get('api/queue/searchById',{
        params:{
            id
        }
    })
    return data

}
export const searchByName = async (name) => {
    const {data} = await $host.get('api/queue/searchByName',{
        params:{
            name
        }
    })
    return data

}

export const addToQ = async (userId, queueId) => {
    const {data} = await $authHost.post('api/queue/addToQ',{
            userId,
            queueId
        })
    return data

}
export const createQ = async (name, description) => {
```


SimpleQueue
Поиск
Создать
Мои очереди
Пользователь: admin
Выйти
Встать в очередь
Новая очередь
Уникальный номер: 6
"Описание"
В очереди никого нет, станьте первым!


Новости
YouTube
ИУ5 Папка для учё...
GitHub
Российский серви...
Coursera
SimpleQueue
Поиск
Создать
Мои очереди
Пользователь: testtest
Выйти
Выйти
Очередь
admin
testtest