Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

УНИВЕРСИТЕТ ИТМО

Факультет инфокоммуникационных технологий

Образовательная программа 09.03.03

Направление подготовки (специальность) Мобильные и сетевые технологии

О Т Ч Е Т

о курсовой работе

Тема задания: разработка клиентской части сервиса доставки еды внутри студенческих общежитий средствами фреймворка React.JS

Обучающийся Кривошапкина Айталина Сергеевна, К33402

Руководитель: Добряков Д. И., преподаватель

Оценка за курсовую работу ____

Подписи членов комиссии:

_____ (Добряков Д. И.)
(подпись)

Дата ____

Санкт-Петербург
2020

ВВЕДЕНИЕ

Актуальность

На данный момент наблюдается большой рост сервисов доставки еды, продуктов и готовых рационов. Связано это с наступившей два года назад пандемией: она создала новые привычки и подстегнула людей переносить больше повседневных дел в онлайн. Только за 2020 год оборот интернет-продаж еды в целом вырос более чем в 4 раза [1].

Несмотря на разнообразие и насыщенность рынка, нет еще сервиса, в котором клиент и курьер могли бы самостоятельно договориться о доставке. Проще такой сервис запустить в студенческих общежитиях, где сами студенты могут играть роль не только клиента, но и курьера. Доставка в таком случае осуществляется своему же “соседу”, что сводит затраты на время доставки до минимума. Благодаря лояльному ценообразованию у студентов будет желание заказывать, а у других - желание получить прибыль.

Цели и задачи

1. Определение средств разработки
2. Определение функциональных требований
3. Прототипирование
4. Проектирование и реализация клиентской части

ГЛАВА 1. СРЕДСТВА РАЗРАБОТКИ И ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1.1 Средства разработки

Клиентская часть приложения реализуется на фреймворке React.JS [2]. React обеспечивает повышенную гибкость благодаря использованию «компонентов» — коротких изолированных участков кода, которые помогают разработчикам создавать сложную логику и UI. React взаимодействует с HTML через virtual DOM — копию реального DOM-дерева элементов страницы. В копии все элементы представлены как объекты JavaScript. Эти элементы, вместе с декларативным стилем программирования React и односторонним связыванием данных, упрощают и ускоряют разработку.

В качестве инструмента для прототипирования был выбран онлайн-сервис Figma [3] — это программа для веб-дизайнеров, с помощью которой можно создавать не только прототипы, но и конечные интерфейсы сайтов и приложений. Процесс работы в программе интуитивно понятен, а возможность совместной одновременной работы над проектом привлекает все больше и больше команд к реализации прототипов именно в Figma.

1.2 Функциональные требования

Функциональные требования по этому проекту заключались в реализации нескольких страниц, обеспечивающих минимальную жизненную способность приложения:

1. Лэндинг
    a. Краткая информация о проекте и о команде.
2. Регистрация и аутентификация
    a. При регистрации пользователь вводит свой телефон, выбирает общежитие, номер комнаты, в которой он живет, персональную информацию и задает пароль доступа к своему личному кабинету.
    b. При аутентификации требуется только номер телефона и пароль.
3. Список ресторанов
    a. Список ресторанов и возможность фильтрации по ним.

4. Список товаров в ресторане
    a. Позиции, предоставляемые выбранным рестораном.
    b. Рядом с каждой позицией его цена и текущее количество выбранного товара в корзине пользователя.
5. Корзина
    a. Список выбранных товаров
    b. Поле ввода предлагаемой стоимости доставки
    c. Добавление заказа в историю заказов
6. Личный кабинет
    a. Информация о пользователе
    b. История заказов
    c. Возможность закрыть действующий заказ
7. Список курьеров
    a. Список курьеров в текущем общежитии
8. Страница курьера
    a. Персональная информация о курьере
    b. Календарь рабочих часов.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

2.1. Прототипирование

Перед тем, как начать проектирование и реализацию, следует продумать макет будущего сайта для проверки гипотез и логики, отработки пользовательских сценариев. Для данной задачи был выбран онлайн-сервис для разработки интерфейсов и прототипирования Figma.

Для будущего сайта был выбран минималистичный стиль в фиолетовых тонах. В хедере размещен логотип сервиса, название города, ссылка для авторизации или регистрации пользователя, а также корзина с отображением количества товаров в ней. У каждой страницы присутствует соответствующий ей заголовок, а само содержимое разложено по “карточкам”. Например, на странице списка ресторанов каждый отдельный ресторан вынесен в свою карточку с основной информацией: логотип, название, адрес ближайшего к общежитию ресторана, время пути и средняя цена товара. В футер вынесены логотип сервиса, меню сайта, а также предложение скачать мобильное приложение на Android.

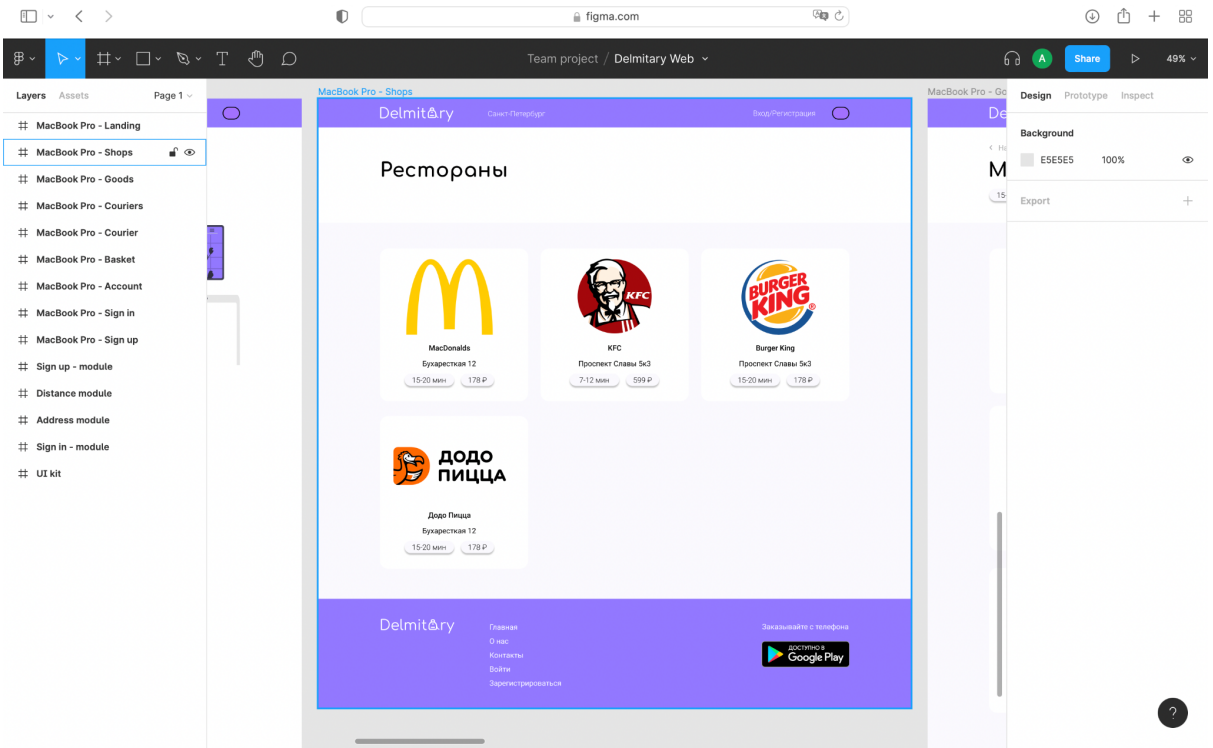

Рисунок 1. Макет страницы списка ресторанов в Figma

После отрисовки макета каждой страницы был создан прототип, с помощью которого были отработаны пользовательские сценарии.

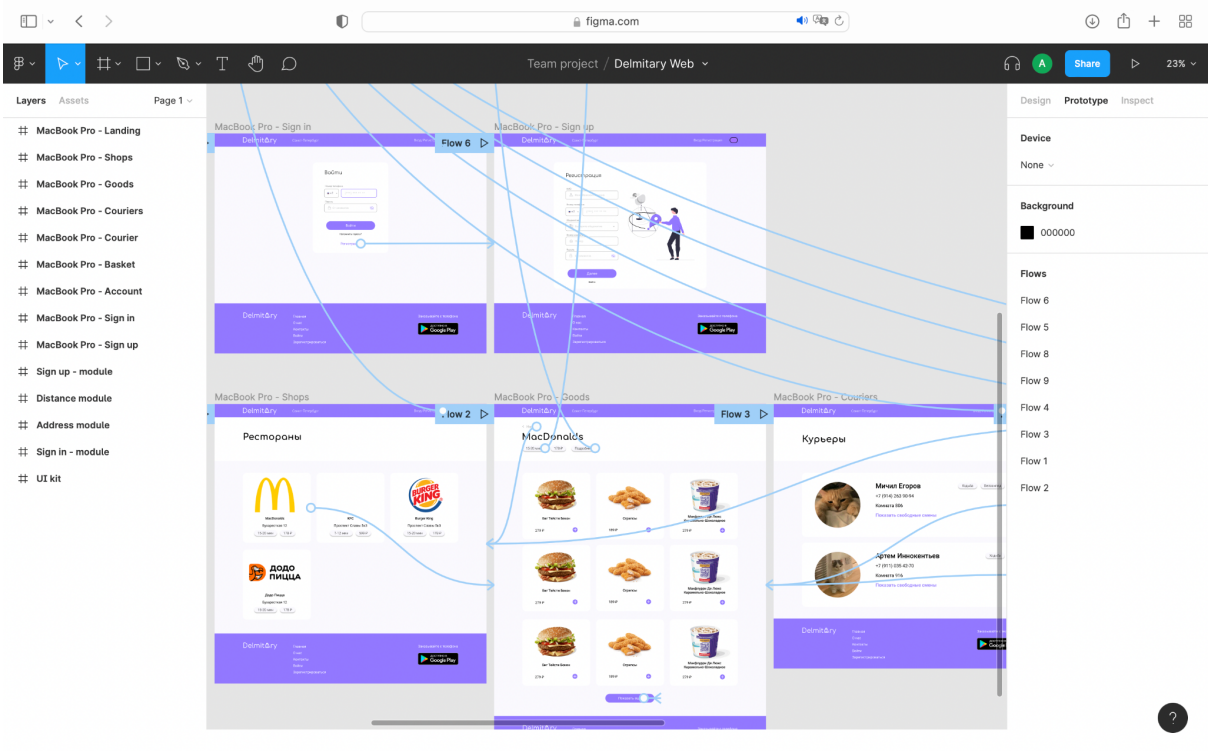

Рисунок 2. Проектирование пользовательских сценариев в Figma

Благодаря прототипированию были существенно снижены затраты на разработку, так как структура сайта, карта взаимодействия пользователя с продуктом и конечный дизайн уже были готовы.

2.2. Проектирование моделей

Теперь спроектируем модели данных и их взаимосвязи.

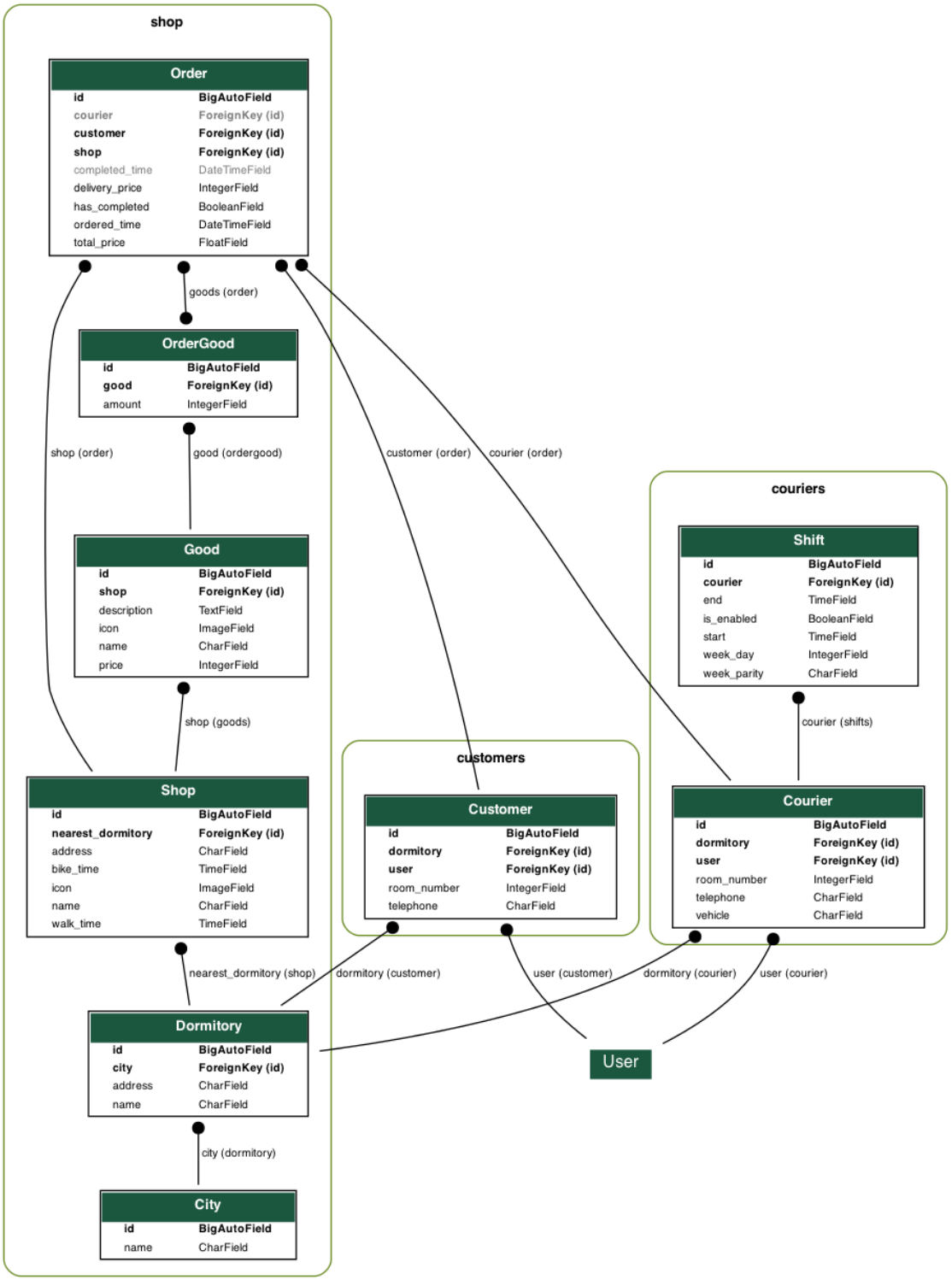

Рисунок 3. Модели данных

2.3. Проектирование и реализация клиентской части

2.3.1 Страница списка ресторанов

На данной странице отображаются все рестораны, товары которых можно заказать. Каждый ресторан имеет свою карточку с изображением логотипа, названием, адресом ближайшего филиала к общежитию заказчика, примерным временем пути до него, а также средней стоимостью товара. Также на странице есть возможность поиска ресторана по названию. Для отрисовки карточек и формы поиска были использованы шаблоны Bootstrap [4].

Листинг 1. Функция рендеринга страницы ресторанов из файла Shop.js

```
    render() {
        return (
            <div>
                <HeaderTitle title="Рестораны"/>

                <form action="" onSubmit={this.filterShops}>
                    <div class="row">
                        <div class="col-md-9">
                            <input class="form-control" type="text"
onChange={(e) => this.setState({filter_name: e.target.value})}
placeholder="Название магазина"/>
                        </div>
                        <div class="col-md-3">
                            <input class="btn btn-delmitary" type="submit"
value="Поиск" style={{height: "38px"}} />
                        </div>
                    </div>
                </form>

                <div className="mt-5 row row-cols-1 row-cols-md-3 g-4">
                    {this.state.shops.map((shop) =>
                        <CompactShop key={shop.id} shop={shop}/>
                    )}
                </div>
            </div>
        )
    }
```

Отображение страницы:

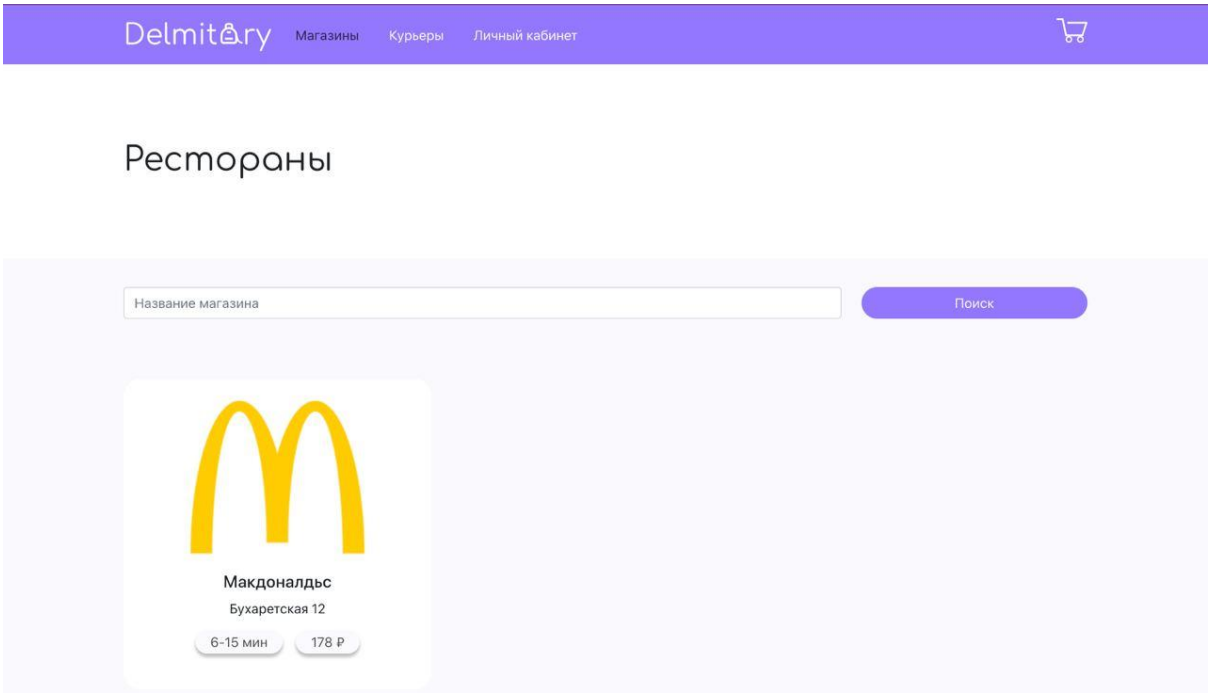

Рисунок 4. Страница списка ресторанов

2.3.2 Страница списка товаров

На данной странице в заголовок вынесено название ресторана. Ниже представлены товары в виде карточек с основной информацией: изображение товара, его название и цена. Также на каждой карточке присутствуют кнопки добавления или удаления товара с указанием количества данного товара в корзине.

Листинг 2. Функция рендеринга страницы товаров из файла Goods.js

```
    render() {
        return (
            <div>
                <HeaderTitle title={this.state.shop.name} />
                <div class="row">
                    {
                        this.state.goods.length > 0
                        ? this.state.goods.map((good) => <Good good={good}
cart={this.props.cart} addGood={this.addGood}
decreaseGood={this.decreaseGood}/>)
                        : <p style={{
                                position: 'absolute', left: '50%', top:
'50%',
                                transform: 'translate(-50%, -50%)'
                            }}>Загрузка...
                        </p>
```

```
                    }
                </div>

            </div>
        );
    }
```

Отображение страницы:

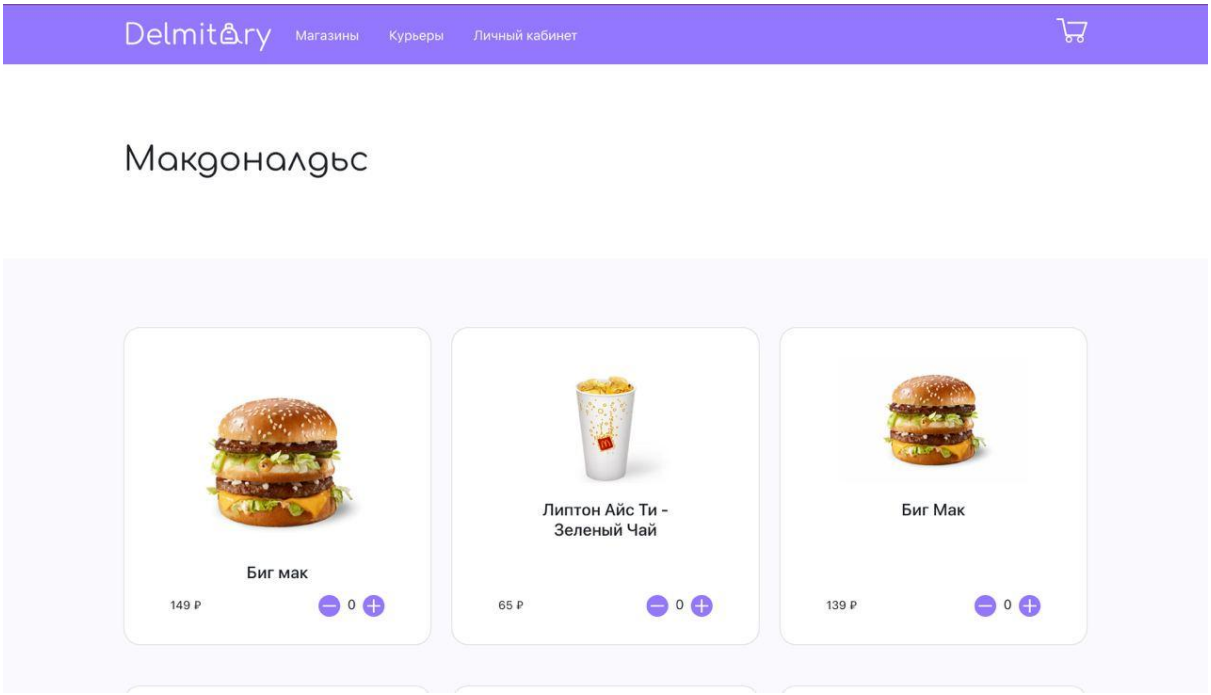

Рисунок 5. Страница товаров выбранного ресторана

2.3.3 Страница регистрации

Для регистрации были реализованы формы с вводом текста и формы с выпадающим списком. В каждой форме присутствует своя плавающая метка с подсказкой какого формата данные нужно вводить. На случай, если у пользователь уже зарегистрирован, внизу страницы есть кнопка, ведущая на страницу авторизации.

Листинг 3. Функция рендеринга страницы регистрации из файла Registration.js

```
    render() {
        if(this.state.isSuccess) {
            return <Navigate replace to='/shops/'/>
        }

        return (
            <div className="grid-wrapper">
```

```
                <div className="regform">
                    <h3 class="comfortaaTitle"> Регистрация </h3>
                    <form class="row g-3 needs-validation" novalidate>
                        <div class="mb-3">
                            <label for="validationTooltip01"
class="form-label">ФИО</label>
                            <input type="text" class="form-control"
id="validationTooltip01" placeholder="Иванов Иван Иванович" required/>
                            <div class="valid-tooltip">
                            Looks good!
                            </div>
                        </div>
                        <div class="mb-3 column-1">
                            <label for="validationTooltip02"
class="form-label">Номер телефона</label>
                            <input type="text" class="form-control"
id="validationTooltip02" placeholder="* (***) *** ** **" required/>
                            <div class="valid-tooltip">
                            Looks good!
                            </div>
                        </div>
                        <div class="mb-3">
                            <label for="validationTooltip04"
class="form-label">Общежитие</label>
                            <select class="form-select"
id="validationTooltip04" required>
                            <option selected disabled value="">Выберите
общежитие</option>
                            <option>Альпийский пер., 15к2</option>
                            <option>Вяземский пер., 5-7</option>
                            <option>Ленсовета, 23</option>
                            </select>
                            <div class="invalid-tooltip">
                            Please select a valid state.
                            </div>
                        </div>
                        <div class="mb-3">
                            <label for="validationTooltip03"
class="form-label">Номер комнаты</label>
                            <input type="text" class="form-control"
id="validationTooltip03" placeholder="Номер комнаты" required/>
                            <div class="invalid-tooltip">
                            Please provide a valid city.
                            </div>
                        </div>
                        <div className="mb-3">
                            <label htmlFor="exampleInputPassword1"
className="form-label">Пароль</label>
                            <input onChange={(e) =>
this.setPassword(e.target.value)} type="password" className="form-control"
id="exampleInputPassword1" placeholder="Придумайте пароль"/>
                        </div>
                        <div class="our-button">
```

```
                            <button class="btn btn-primary"
type="submit">Далее</button>
                        </div>
                        <Link to={`/auth/`}><p>Есть аккаунт? Войти</p></Link>
                    </form>
                </div>

                <div className="reg-icon">
                    <img src={regIcon} style={{height: "366px"}} />
                </div>

            </div>

        )
    }
```

2.3.4 Страница аутентификации

После ввода данных для авторизации, логина и пароля, следует чекбокс “Запомнить меня”. На случай, если у пользователь еще не зарегистрирован, внизу страницы есть кнопка, ведущая на страницу регистрации.

Листинг 4. Функция рендеринга страницы регистрации из файла Auth.js

```
    render() {
        if(this.state.isSuccess) {
            return <Navigate replace to='/shops/'/>
        }

        return (
            <div className="authform">
                <h3 class="comfortaaTitle"> Войти </h3>

                <form onSubmit={this.login}>
                    <div className="mb-3">
                        <label htmlFor="exampleInputEmail1"
className="form-label">Логин</label>
                        <input onChange={(e) =>
this.setLogin(e.target.value)} type="text" className="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp"/>
                    </div>
                    <div className="mb-3">
                        <label htmlFor="exampleInputPassword1"
className="form-label">Пароль</label>
                        <input onChange={(e) =>
this.setPassword(e.target.value)} type="password" className="form-control"
id="exampleInputPassword1"/>
                    </div>
                    <div className="mb-3 form-check">
                        <input type="checkbox" className="form-check-input"
id="exampleCheck1"/>
```

```
                        <label className="form-check-label"
htmlFor="exampleCheck1">Запомнить меня</label>
                    </div>
                    <div className="our-button" style={{padding: "0 61px"}}>
                        <button type="submit" className="btn
btn-delmitary">Submit</button>
                    </div>
                    <Link to={`/registration/`}><p>Регистрация</p></Link>
                </form>
            </div>
        )
    }
```

Отображение страницы:

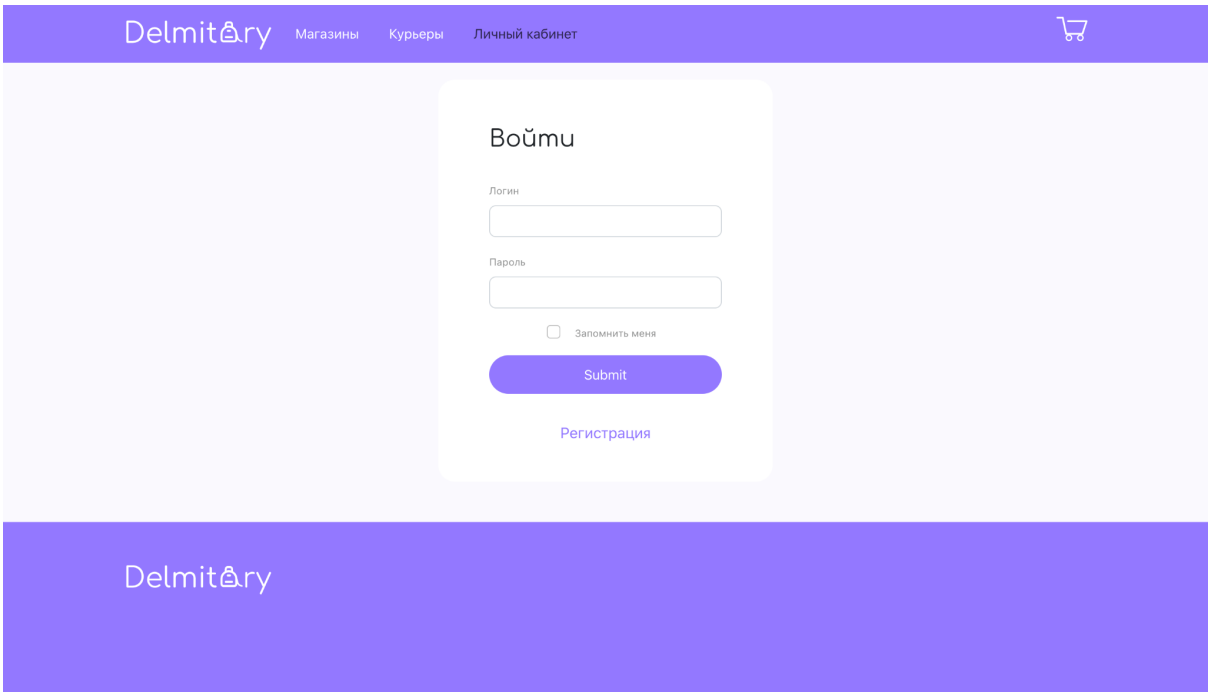

Рисунок 6. Страница авторизации

2.3.5 Страница личного кабинета

В личном кабинете пользователя содержатся персональные данные такие как логин и почта, а также история совершенных заказов. Каждый заказ вынесен в карточку с его основной информацией. По умолчанию отображаются номер заказа, дата оформления, сумма и текущий статус с возможностью завершить заказ. С помощью кнопок “Подробнее” и “Скрыть” можно отображать или скрывать информацию с содержимым заказа в виде изображений.

Листинг 5. Функция рендеринга страницы личного кабинет из файла Account.js

```
    render() {
        if(sessionStorage.getItem('auth_token') === null) {
            return <Auth />
        }

        return (
            <div>
                <HeaderTitle title="Личный кабинет"/>

                <div className="row">
                    <div className="col-sm-3">
                        <h2>Профиль</h2>
                    </div>
                    <div className="col-sm-9">
                        <h2>Заказы</h2>
                    </div>
                </div>

                <div className="row">
                    <div className="col-sm-3">
                        <div className="card border-0" >
                        <div className="card-body" >
                            <p>Имя: {this.state.user.username} </p>
                            <p>Почта: {this.state.user.email} </p>
                        </div>
                        </div>
                    </div>
                    <div className="col-sm-9" >
                        {this.state.orders.map((order) => <Order
order={order}/>)}
                    </div>
                </div>
            </div>
        )
    }
```

Отображение страницы:

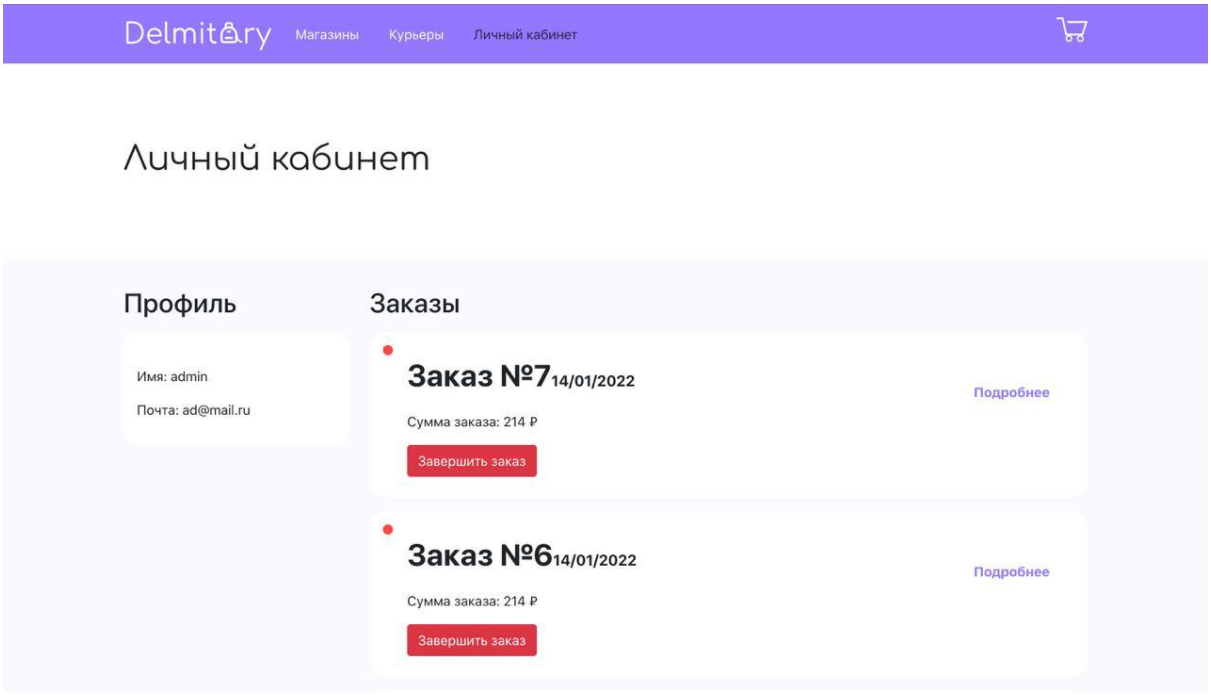

Рисунок 7. Личный кабинет пользователя по умолчанию

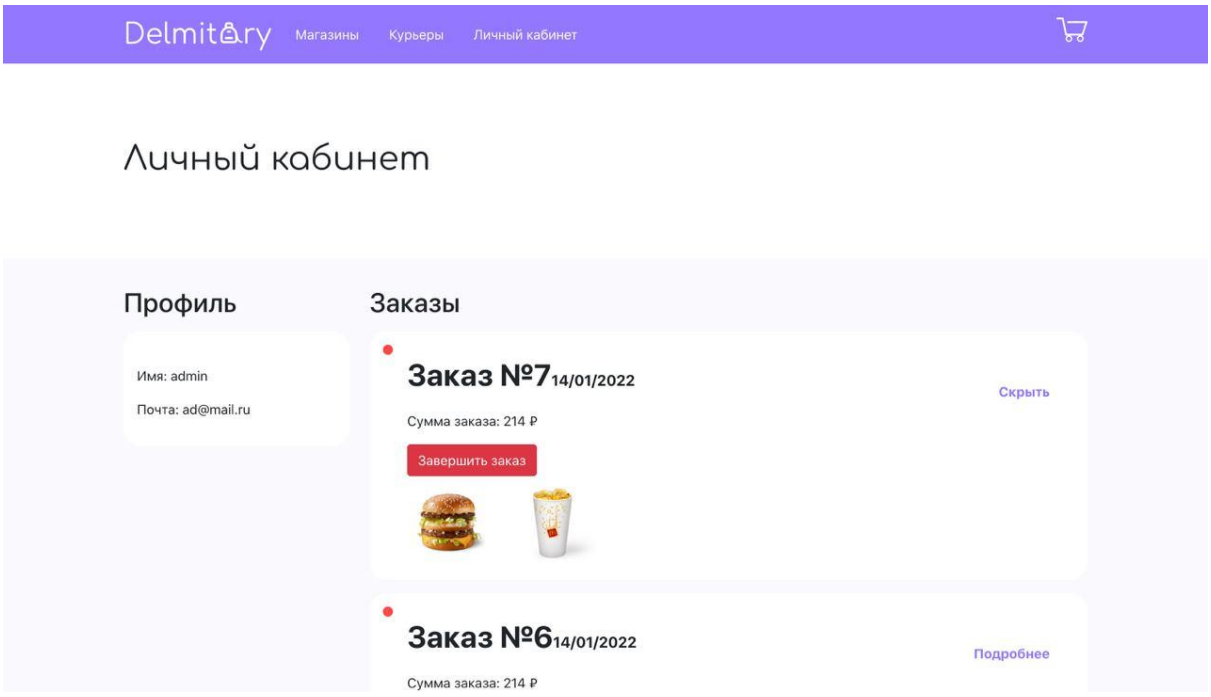

Рисунок 8. Личный кабинет пользователя с раскрытой карточкой заказа

2.3.6 Страница списка курьеров

Здесь отображается список всех курьеров в отдельных карточках. По умолчанию мы видим изображение профиля, имя и фамилию, номер телефона, номер комнаты в

общежитии и способ передвижения: езда на велосипеде или ходьба. Также можно посмотреть рабочее расписание, то есть свободные смены - данная кнопка приведет на личную страницу курьера.

Листинг 6. Функция рендеринга страницы курьеров из файла Couriers.js

```
    render() {
        return (
            <div>
                <HeaderTitle title="Курьеры"/>

                {this.state.couriers.map((courier) => <CourierCompact
courier={courier}/>)}
            </div>
        )
    }
```

Отображение страницы:

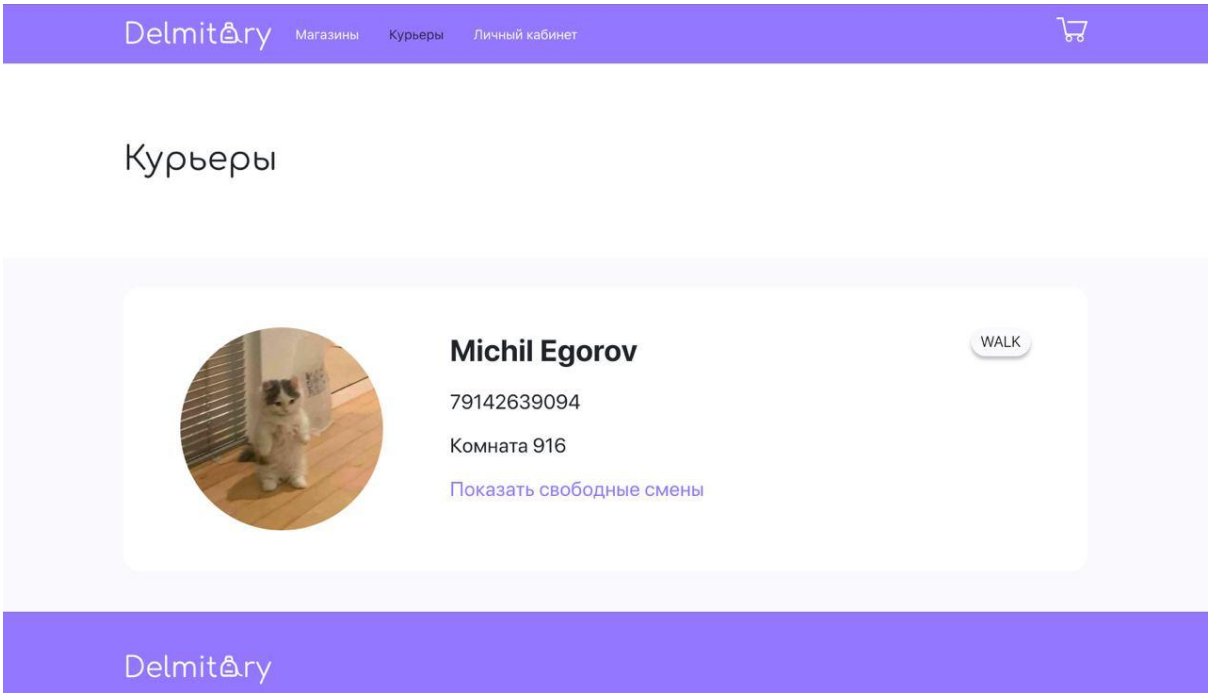

Рисунок 9. Страница списка курьеров

2.3.7 Страница курьера

График работы поделен на нечетную и четную неделю ввиду учебного расписания в университете. Для каждого дня недели отрисована своя карточка, где указываются свободные промежутки времени.

Листинг 7. Функция рендеринга страницы курьера из файла Courier.js

```
    render() {
        return (
            <div>
                <HeaderTitle title={this.state.courier.first_name + " " +
this.state.courier.last_name}/>

                <h1>Нечетная неделя</h1>
                <div className="mt-3 row">
                    {this.state.even_shifts.map((child) => child)}
                </div>

                <h1 class="mt-5">Четная неделя</h1>
                <div className="mt-3 row">
                    {this.state.odd_shifts.map((child) => child)}
                </div>
            </div>
        )
    }
```

Отображение страницы:

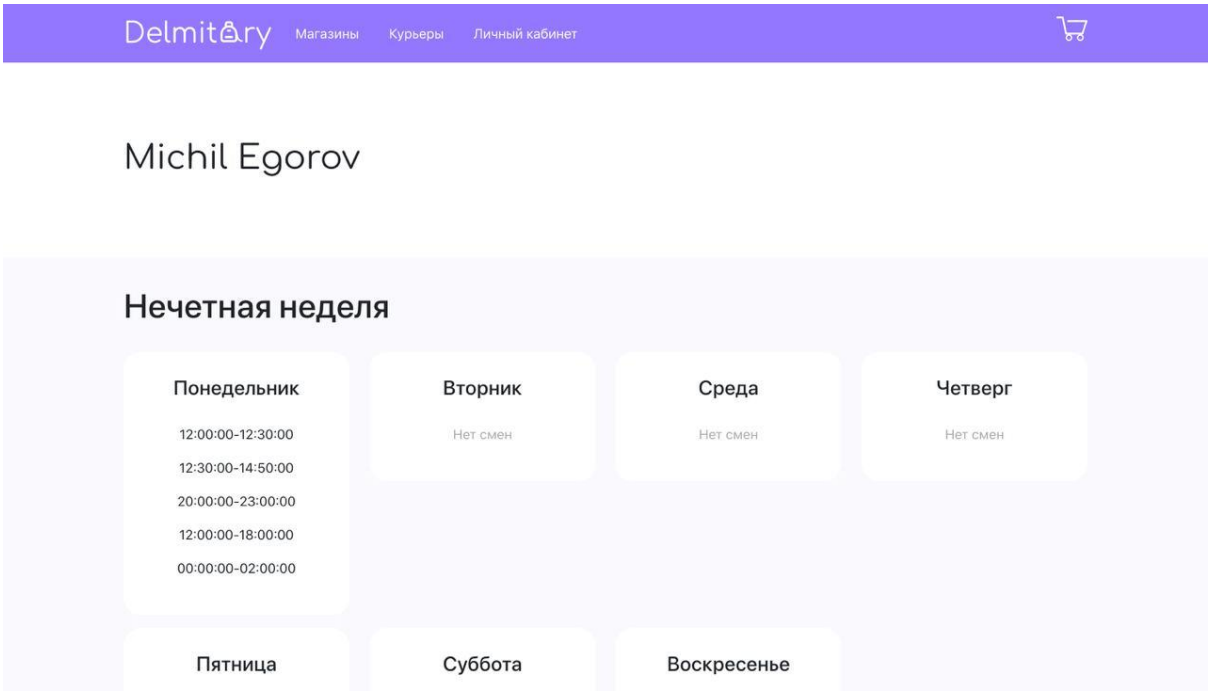

Рисунок 10. Страница курьера с графиком работы

2.3.8 Страница корзины

В корзине отображаются все добавленные товары с тем же содержимым, что и на странице товаров. Количество добавленного товара можно регулировать кнопками

плюса и минуса, либо удалить его совсем, нажав на крест в углу. Справа отображается сумма заказа с полем для предложения стоимости доставки и кнопка для оформления.

Листинг 8. Функция рендеринга страницы корзины из файла Cart.js

```
    render() {
        return (
            <div>
                <div>
                    <HeaderTitle title={"Моя корзина"} />
                </div>

                <div class="row">
                    <div className="col-md-8">
                        <h2>Заказы</h2>

                        {this.props.cart.goods.length != 0 ?
                        this.props.cart.goods.map((good) => <CartItem
addGood={this.addGood} decreaseGood={this.decreaseGood} good={good}/>)
                        : <Link to='/shops'>Выберите товар из списка</Link>}
                    </div>
                    <div className="col-md-4">
                        <h2>Итого</h2>
                        <div class="order-info card mt-4">
                            <div class="card-body" style={{padding: "0px 36px
18px 36px"}}>
                                <div class="d-flex justify-content-between">
                                    <p> Всего:</p>
                                    <p> {this.props.cart.totalPrice} ₽</p>
                                </div>
                                <label> Доставка: </label>
                                <input class="form-control" type="number"
value={this.state.delivery_price ? this.state.delivery_price : ''}
onChange={(e) => this.changePrice(e.target.value)}/>
                                <div class="mt-3 d-flex
justify-content-between">
                                    <p> Итого:</p>
                                    <p> {this.props.cart.totalPrice +
this.state.delivery_price} ₽</p>
                                </div>
                                <div class="text-center">
                                    <button class="btn btn-delmitary"
onClick={this.makeOrder}>Оформить заказ</button>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        )
    }
```

Отображение страницы:

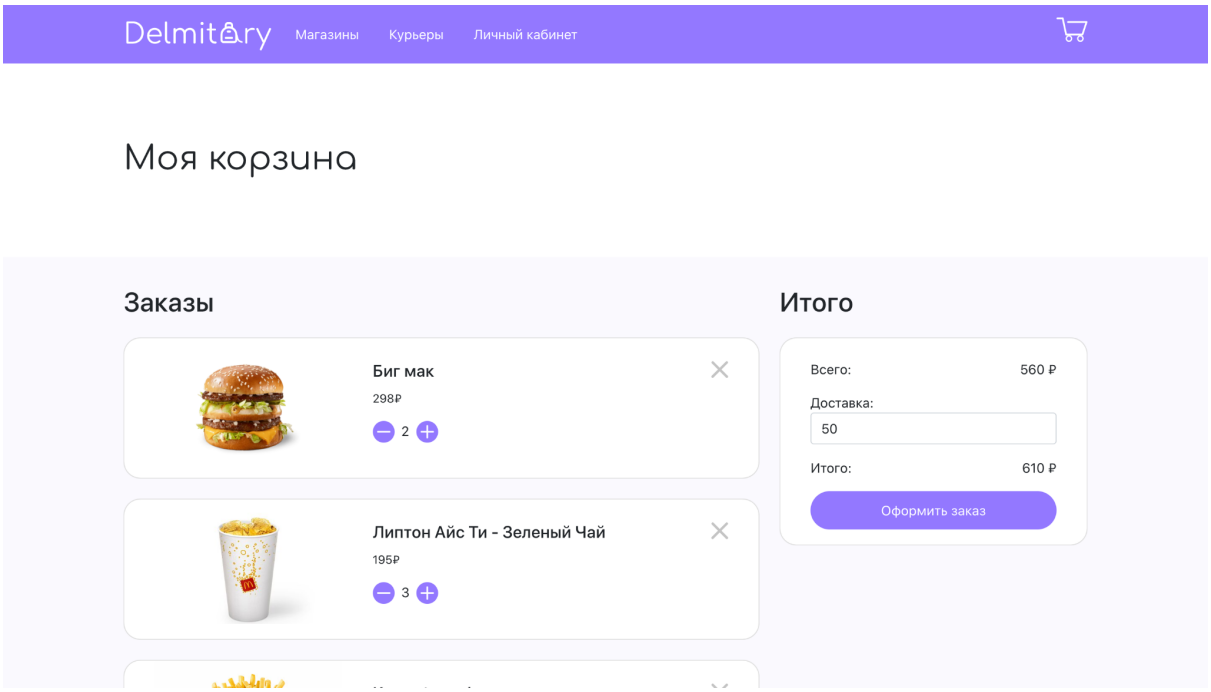

Рисунок 11. Страница корзины

ЗАКЛЮЧЕНИЕ

Выводы по работе

В результате данной работы был изучен фреймворк React.JS для реализации клиентской части сервиса доставки еды внутри студенческих общежитий.

СПИСОК ЛИТЕРАТУРЫ

1. Исследование Тинькофф “Рынок доставки еды, продуктов и готовых рационов” [Электронный ресурс] — https://acdn.tinkoff.ru/static/documents/market-for-delivery-food-groceries-and-ready-made-rations.pdf. Дата обращения 20.01.2022.
2. Документация React [Электронный ресурс] — https://ru.reactjs.org/docs/getting-started.html. Дата обращения 18.11.2021.
3. Документация Figma [Электронный ресурс] — https://www.figma.com/community. Дата обращения 18.11.2021.
4. Документация Bootstrap [Электронный ресурс] — https://bootstrap-4.ru. Дата обращения 18.11.2021.