



NATIONAL RESEARCH
UNIVERSITY

Факультет информатики, математики и компьютерных наук

Выпускная квалификационная работа - БАКАЛАВРСКАЯ РАБОТА
по направлению подготовки Программная инженерия

РАЗРАБОТКА АДАПТИВНОГО САЙТА ДЛЯ ИНТЕРНЕТ МАГАЗИНА С ИСПОЛЬЗОВАНИЕМ REACT И NODE.JS

Макаров Александр Александрович

Научный руководитель:

Марширов Виктор Викторович

Рецензент:

Улитин Борис Игоревич

Нижний Новгород, 2022

Введение

Проектная часть

Характеристики программных
средств для решения
поставленной задачи

Разработка web-приложения

Цель: Разработка адаптивного сайта для интернет магазина с использованием React и Node.js



Выбранная тема дипломной работы весьма актуальна на сегодняшний день, в связи с тем, что сегодня миллионы людей ежедневно, не выходя из дома, покупают различные товары в электронных магазинах.



Необходимость разработки состоит в том, что наилучший способ для любой торгующей компании показать себя потенциальным клиентам – это демонстрация своих товаров и цен на эти товары в сети Интернет. Использование Интернета в деятельности организации – одна из наиболее актуальных задач в современном мире.



Интернет-магазин — это форма электронной торговли, которая позволяет потребителям покупать товары или услуги с помощью веб-браузера.

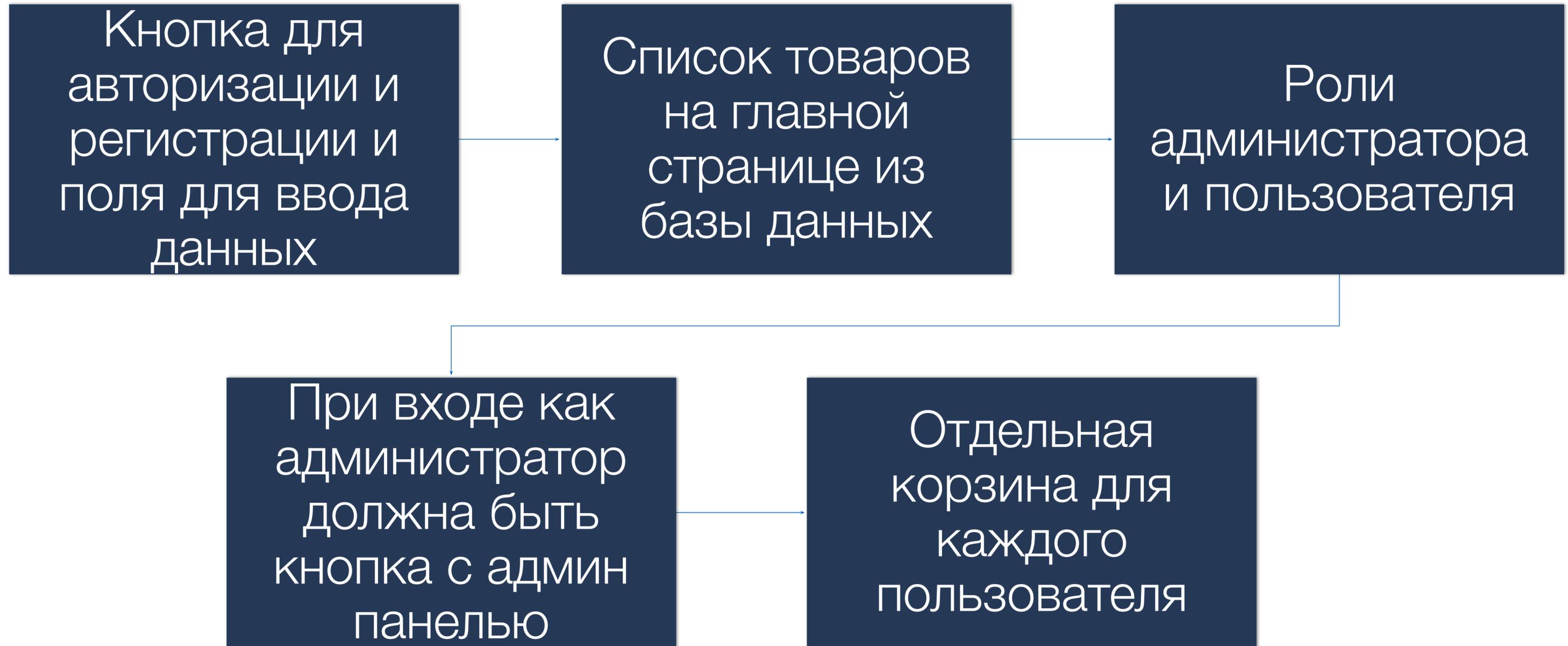
Интернет-торговля в России, прогноз на 2019-2023



Data Insight — первое российское исследовательское агентство, специализирующееся на электронной коммерции.

Дата публикации 2018 Год

ПОСТАНОВКА ЗАДАЧИ



ХАРАКТЕРИСТИКИ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ

1



2



3



4



5



6



7



База данных приложения

Для создания интернет магазина используется база данных PostgreSQL.

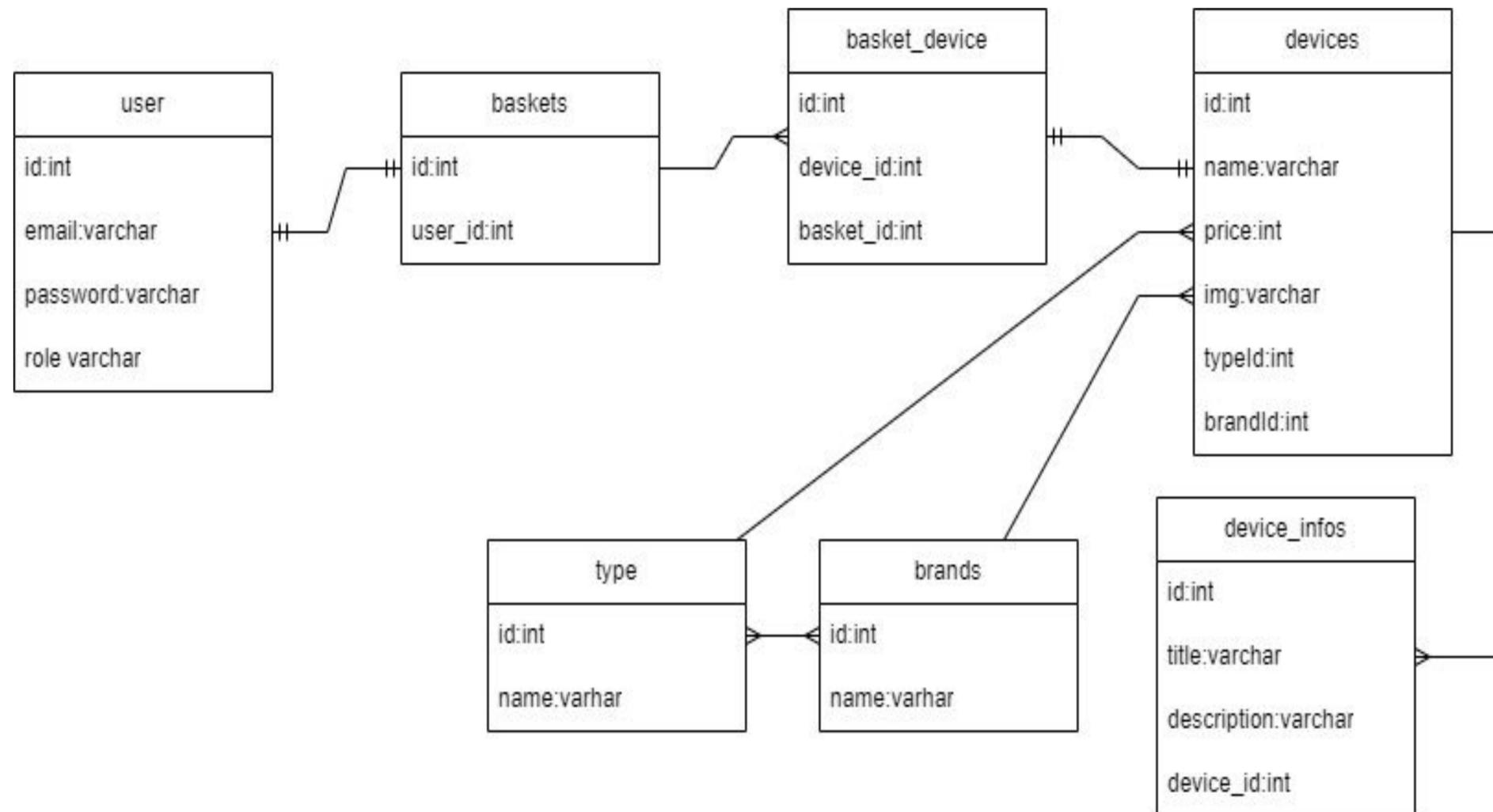


Схема базы данных.

Чтобы создать таблицы в базе данных необходимо в файле `models.js` серверной части приложения с помощью библиотеки `sequelize`, фреймворка `node.js`, который создает и заполняет таблицу полями, написать программный код.

```
models.js
1  const sequelize = require('../db')
2  const {DataTypes} = require('sequelize')
3
4  const User = sequelize.define( modelName: 'user', attributes: {
5    id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
6    email: {type: DataTypes.STRING, unique: true},
7    password: {type: DataTypes.STRING},
8    role: {type: DataTypes.STRING, defaultValue: "USER"},
9  })
10
11 const Basket = sequelize.define( modelName: 'basket', attributes: {
12   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
13 })
14
15 const BasketDevice = sequelize.define( modelName: 'basket_device', attributes: {
16   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
17 })
18
19 const Device = sequelize.define( modelName: 'device', attributes: {
20   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
21   name: {type: DataTypes.STRING, unique: true, allowNull: false},
22   price: {type: DataTypes.INTEGER, allowNull: false},
23   img: {type: DataTypes.STRING, allowNull: false},
24 })
25
26 const Type = sequelize.define( modelName: 'type', attributes: {
27   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
28   name: {type: DataTypes.STRING, unique: true, allowNull: false},
29 })
30
31 const Brand = sequelize.define( modelName: 'brand', attributes: {
32   id: {type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true},
33   name: {type: DataTypes.STRING, unique: true, allowNull: false},
34 })
```

Заполнение таблиц полями



РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ

1
1

Серверная часть приложения

Для начала присваивается переменным `name` (Название товара), `price` (стоимость товара), `brandId`, `typeId`, `info` (информация о товаре), `img` (фото товара) значения из модального окна пользователя, фото товара сохраняется в статичную папку проекта `static`, далее происходит валидация этих переменных если данные получены успешно то они передаются и сохраняются в базу данных, если нет то выводится ошибка.

```
deviceController.js
1  const uuid = require('uuid')
2  const path = require('path');
3  const {Device, DeviceInfo} = require('../models/models')
4  const ApiError = require('../error/ApiError');
5
6  class DeviceController {
7    async create(req, res, next) {
8      try {
9        let {name, price, brandId, typeId, info} = req.body
10       const {img} = req.files
11       let fileName = uuid.v4() + ".jpg"
12       img.mv(path.resolve(__dirname, '..', 'static', fileName))
13       const device = await Device.create({values: {name, price, brandId, typeId, img: fileName}});
14       if (info) {
15         info = JSON.parse(info)
16         info.forEach(i =>
17           DeviceInfo.create({values: {
18             title: i.title,
19             description: i.description,
20             deviceId: device.id
21           }})
22         )
23       }
24       return res.json(device)
25     } catch (e) {
26       next(ApiError.badRequest(e.message))
27     }
28   }
29 }
```

Клиентская часть приложения

```
index.html x userController.js x
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1" />
6     <link
7       rel="stylesheet"
8       href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
9       integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApmC26EwA0H8WgZl5MYyxFfc+NcPb1dKGj7Sk"
10      crossorigin="anonymous"
11     />
12     <link rel="stylesheet" type="text/css" href="main.css">
13     <title>Интернет Магазин</title>
14   </head>
15   <body>
16     <noscript>You need to enable JavaScript to run this app.</noscript>
17     <div id="root"></div>
18   </body>
19 </html>
20
```

`npm install react-bootstrap bootstrap`

Клиентская часть приложения

```
Shop.js
DevicePage.js
userAPI.js
Admin.js
Auth.js
NavBar.js
App.js
index.js

15
16 useEffect( effect: () => {
17   fetchTypes().then(data => device.setTypes(data))
18   fetchBrands().then(data => device.setBrands(data))
19   fetchDevices( typelid: null, brandid: null, page: 1, limit: 2).then(data => {
20     device.setDevices(data.rows)
21     device.setTotalCount(data.count)
22   })
23 }, deps: [])
24
25 useEffect( effect: () => {
26   fetchDevices(device.selectedType.id, device.selectedBrand.id, device.page, limit: 2).then(data => {
27     device.setDevices(data.rows)
28     device.setTotalCount(data.count)
29   })
30 }, deps: [device.page, device.selectedType, device.selectedBrand,])
31
32 return (
33   <Container>
34     <Row className="mt-2">
35       <Col md={3}>
36         <TypeBar />
37       </Col>
38       <Col md={9}>
39         <BrandBar />
40         <DeviceList />
41         <Pages />
42       </Col>
43     </Row>
44   </Container>
45 );
46
47
48 export default Shop;
```

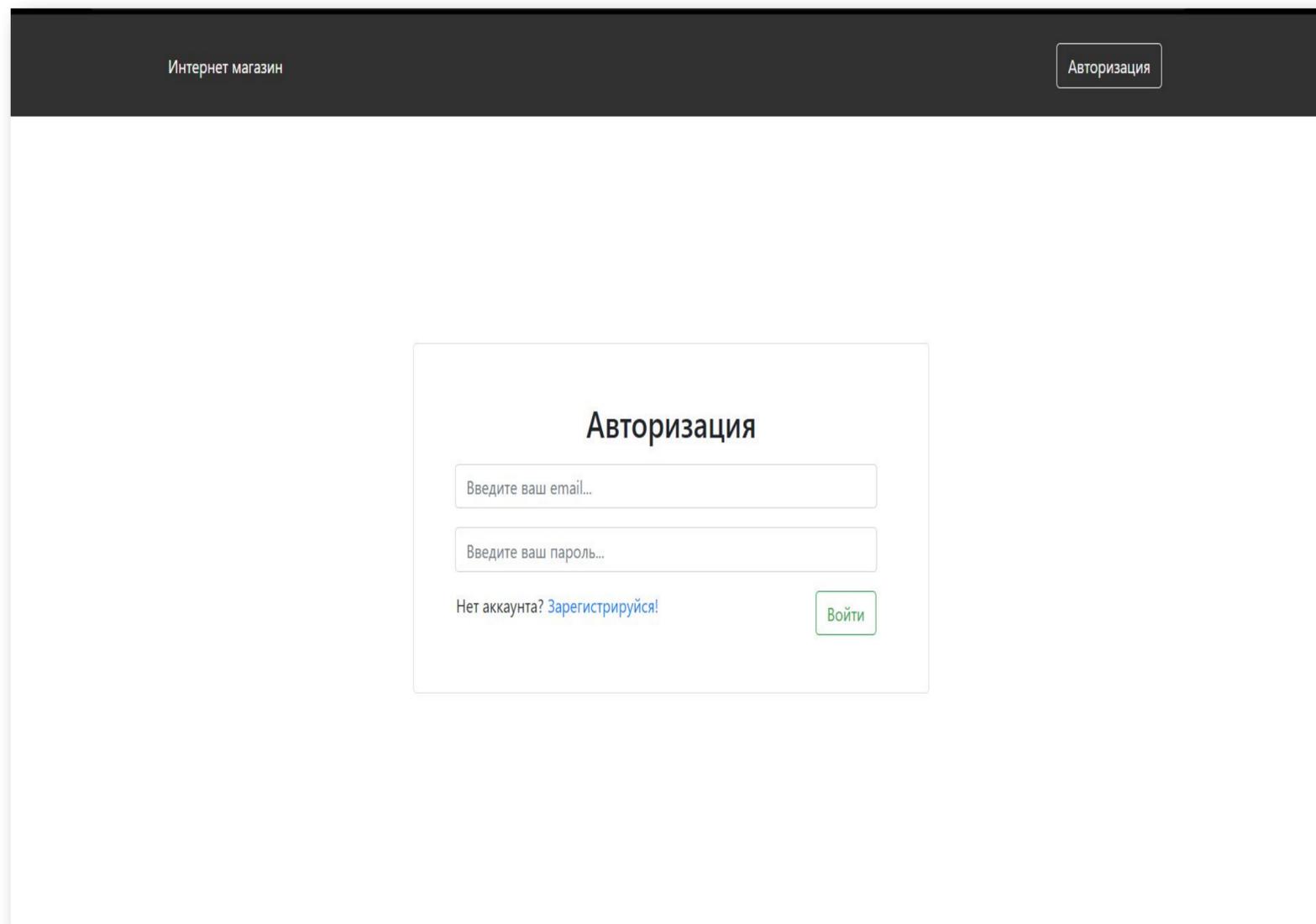
Главная страница

```
TypeBar.js

1 import React, {useContext} from 'react';
2 import {observer} from "mobx-react-lite";
3 import {Context} from "../index";
4 //import Col from "react-bootstrap/Col";
5 import ListGroup from "react-bootstrap/ListGroup";
6
7 const TypeBar = observer( baseComponent: () => {
8   const {device} = useContext(Context)
9   return (
10     <ListGroup>
11       {device.types.map(type =>
12         <ListGroup.Item
13           style={{cursor: 'pointer'}}
14           active={type.id === device.selectedType.id}
15           onClick={() => device.setSelectedType(type)}
16           key={type.id}
17         >
18           {type.name}
19         </ListGroup.Item>
20       )}
21     </ListGroup>
22   );
23 });
24
25 export default TypeBar;
```

Элемент html Typebar(Тип товара).

Результат разработки приложения



Интернет магазин

Авторизация

Авторизация

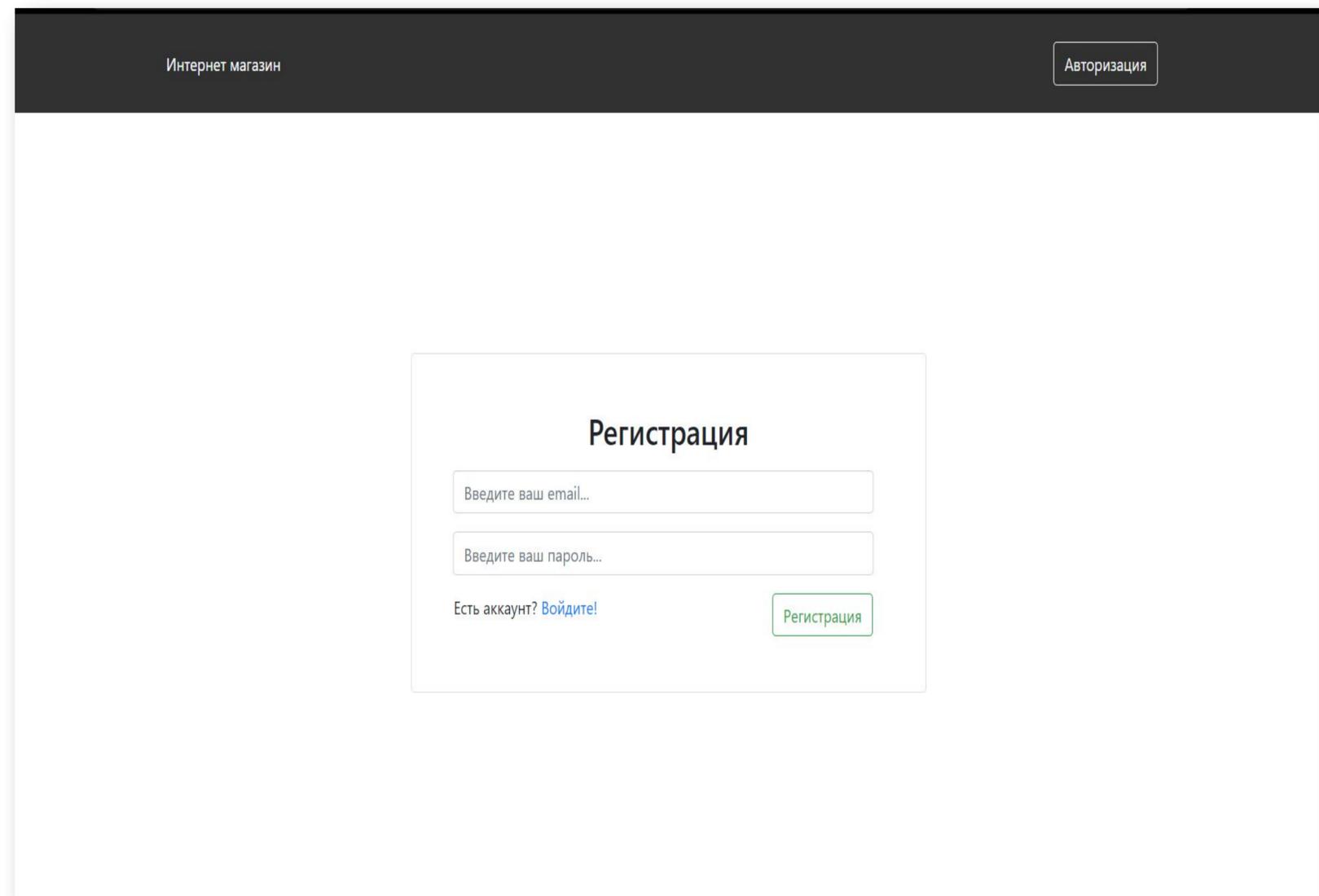
Введите ваш email...

Введите ваш пароль...

Нет аккаунта? [Зарегистрируйся!](#)

Войти

Авторизация



Интернет магазин

Авторизация

Регистрация

Введите ваш email...

Введите ваш пароль...

Есть аккаунт? [Войдите!](#)

Регистрация

Регистрация

Результат разработки приложения

Интернет магазин

Подтвердите действие на странице localhost:3000
Пользователь не найден

Авторизация

OK

Авторизация

Нет аккаунта? [Зарегистрируйся!](#)

Войти

Ошибочный ввод данных

Интернет магазин

Подтвердите действие на странице localhost:3000
Пользователь с таким email уже существует

Авторизация

OK

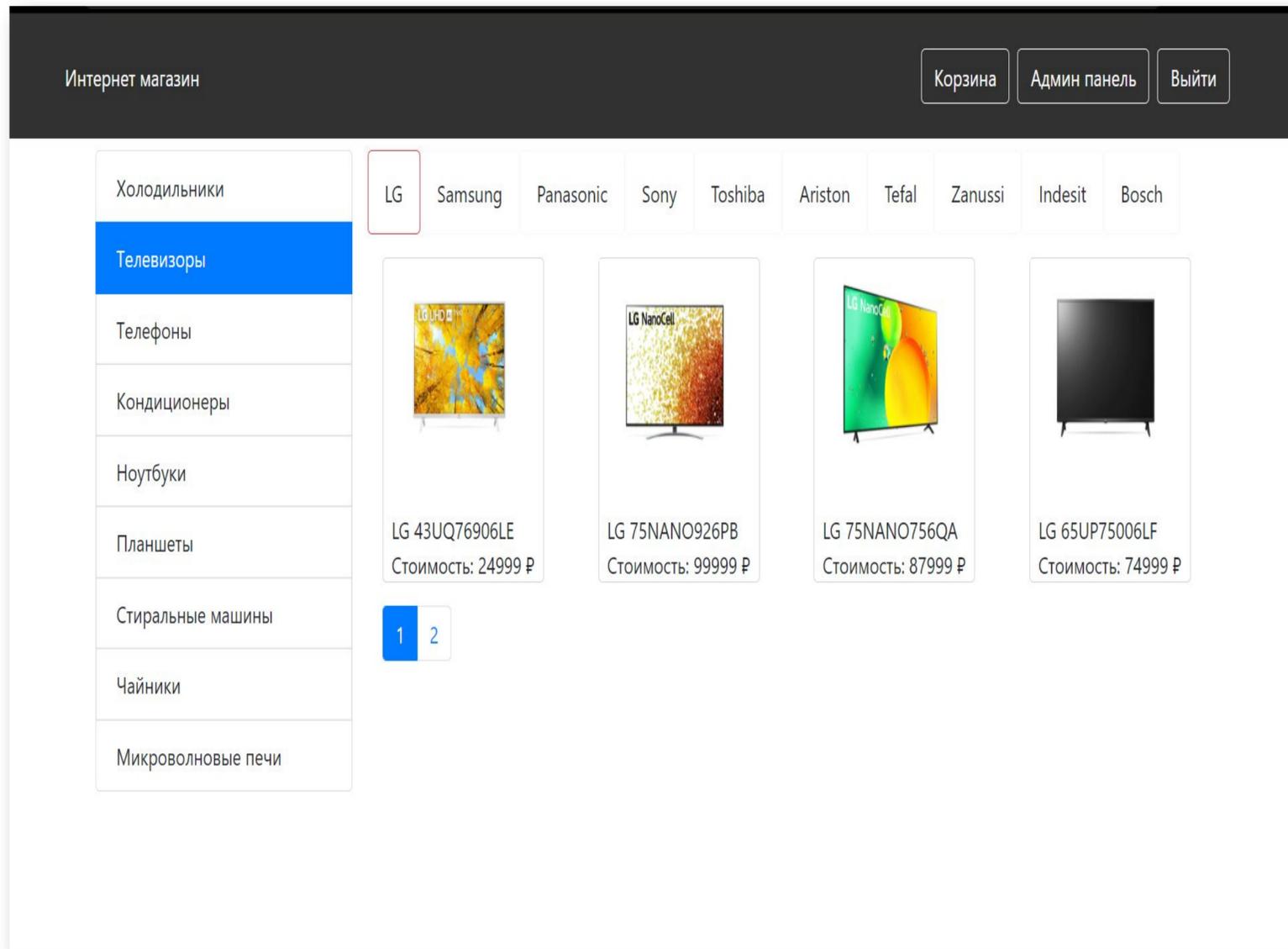
Регистрация

Есть аккаунт? [Войдите!](#)

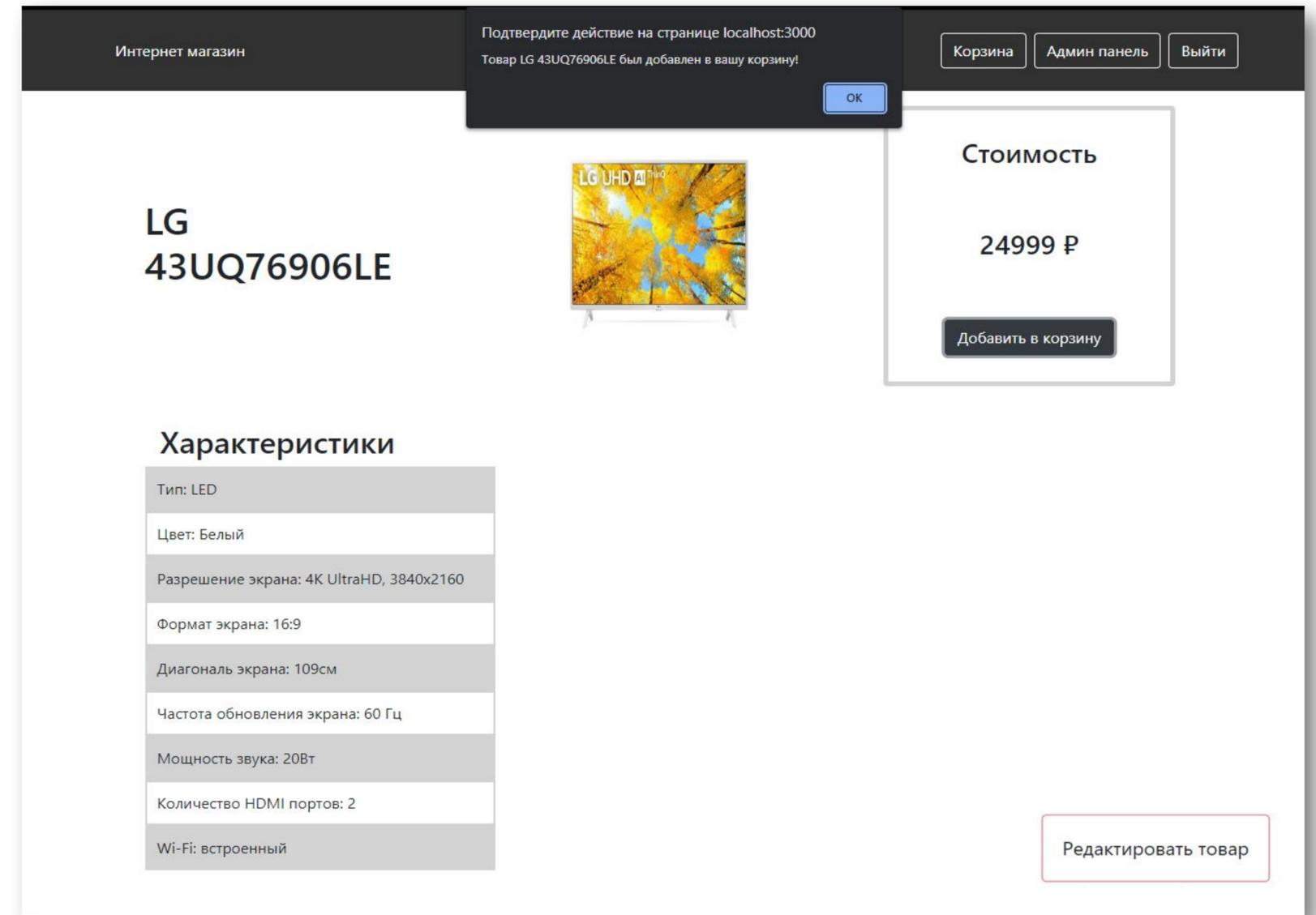
Регистрация

Уже существующий email

Результат разработки приложения



Главная страница сайта



Характеристика товара

Результат разработки приложения

Интернет магазин

[Корзина](#) [Админ панель](#) [Выйти](#)

Корзина

Итого: 170998 ₺

	LG 43UQ76906LE	24999 ₺
	LG 75NANO756QA	87999 ₺
	LG GA-B459MAUM	58000 ₺

[Оплата товара](#)

Корзина

Интернет магазин

[Корзина](#) [Админ панель](#) [Выйти](#)

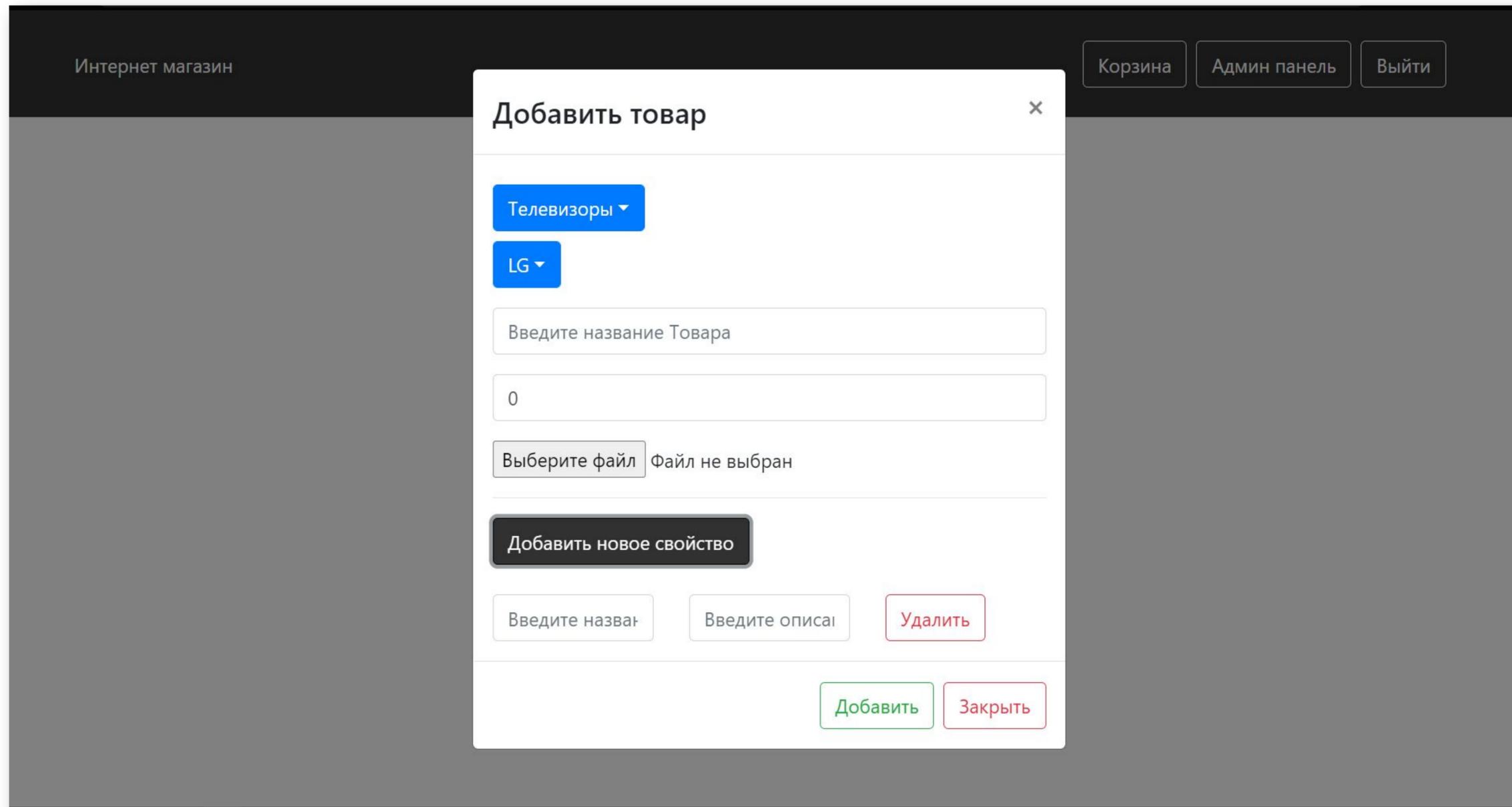
[Добавить тип](#)

[Добавить бренд](#)

[Добавить товар](#)

Админ панель

Результат разработки приложения



Интернет магазин

Корзина Админ панель Выйти

Добавить товар

Телевизоры ▾

LG ▾

Введите название Товара

0

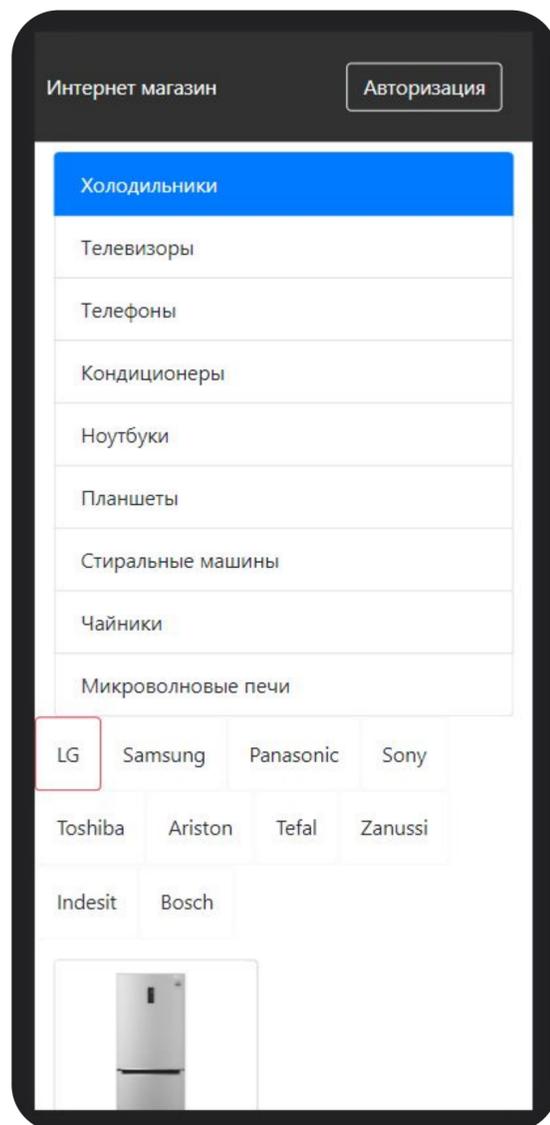
Выберите файл Файл не выбран

Добавить новое свойство

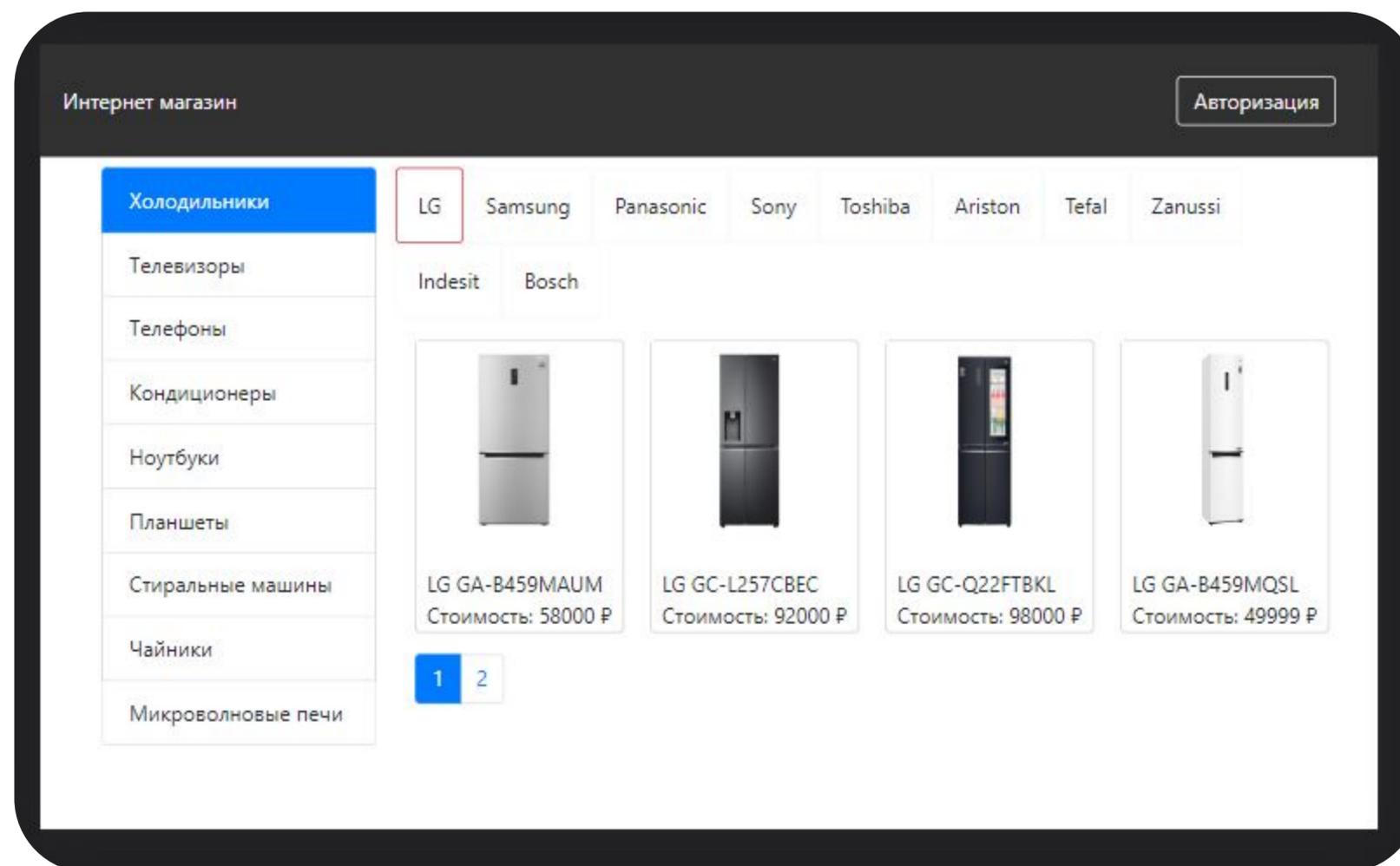
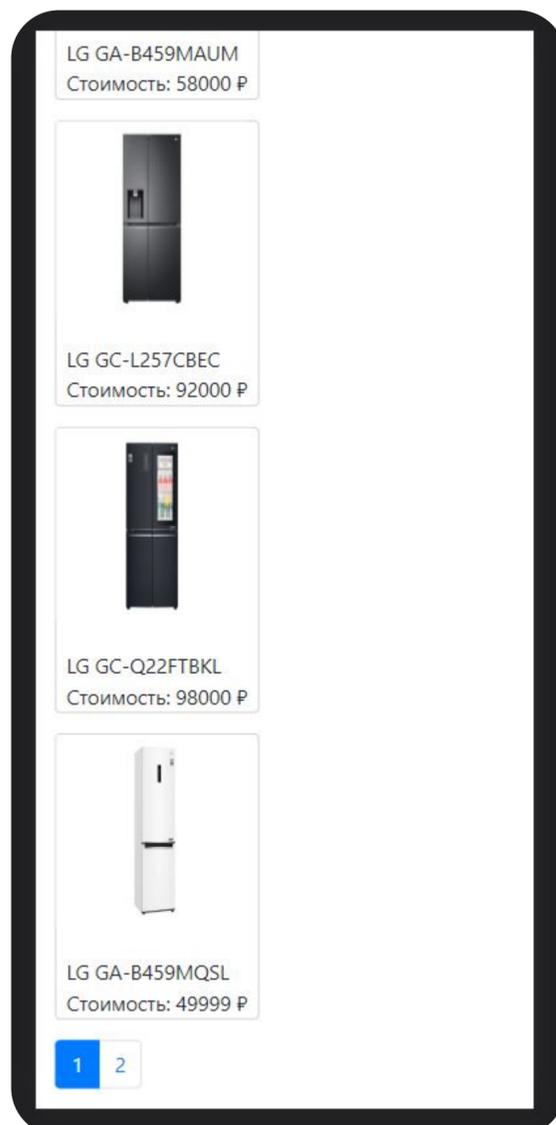
Введите назва Введите описа Удалить

Добавить Заккрыть

Модальное окно с добавлением товара



Смартфон



Планшет



ЗАКЛЮЧЕНИЕ

2
0

В результате выполнения дипломной работы был проведен анализ популярных современных технологий для разработки веб-приложений, при выполнении веб-приложения была выбрана среда разработки WebStorm, платформа Node.js, языки JavaScript, HTML, CSS, фреймворки React и Express.

В ходе реализации проекта осуществлена адаптивная верстка веб-страниц, разработана база данных для приложения, реализована регистрация и авторизация пользователя.

Таким образом, был разработан адаптивный сайт для интернет магазина. Реализована полноценная рабочая система с клиентской и серверной частями

Разработанный интернет-магазин проходил тестирование в компании ООО "Ларус", занимающейся продажей бытовой техники, в целом приложение удовлетворило потребности, но также есть предложения по доработке: добавление поиска товара, сравнения и вкладки избранные товары. После полной доработки приложения и подключения функции оплаты оно может быть внедрено в работу компании.

Цель и задачи в ходе разработки проекта были достигнуты.

- 1 Beginning JavaScript® 4 th Edition Published by Paul Wilton, Jeremy McPeak Inc. 10475 Crosspoint Boulevard Indianapolis.
- 2 Learning React by Alex Banks and Eve Porcello Copyright © 2017 Alex Banks and Eve Porcello. All rights reserved. Printed in the United States of America. Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol.
- 4 Web Development with Node and Express by Ethan Brown Copyright © 2014 Ethan Brown. All rights reserved. Printed in the United States of America. Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol.
- 5 WebStorm документация URL: [https://www.jetbrains.com/help/webstorm /getting-started-with-webstorm.html](https://www.jetbrains.com/help/webstorm/getting-started-with-webstorm.html) (дата обращения 11.12.2022).
- 6 Закутаева К. В., Смагина И. В. Современные технологии разработки сайта и управления им //Экономическая среда. – 2019. – №. 2. – С. 15-24.
- 7 Крокфорд Дуглас К83. Как устроен JavaScript. — СПб.: Питер, 2019. — 304 с. — (Серия «Для профессионалов»). ISBN 978-5-4461-1260-9.
- 8 Express.js Guide the Comprehensive Book on Express.js Azat Mardan ©2013 - 2014 Azat Mardan.
- 9 Основы HTML URL: [https://developer.mozilla.org/ru/docs/ Learn/Getting_started_with_the_web/HTML_basics](https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics) (дата обращения 12.12.2022).

Спасибо за внимание!

GitHub репозиторий- <https://github.com/kRON513/WebShop>

Видео работы веб-приложения -https://drive.google.com/drive/folders/1WsNhSbeRuXsJmZsOL6E9_Qxbncp4TxE7?usp=sharing.