Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное  
учреждение высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт заочно-вечернего обучения

Допускаю к защите

Консультант Ю.Р.Басиров

подпись, И.О. Фамилия

Проектирование приложения с использованием базы данных Наименование темы

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовому проекту по дисциплине**

«Технологии программирования»

1.049.00.00 ПЗ

|  |  |  |  |
| --- | --- | --- | --- |
| Разработал студент группы |  |  | П.А.Зыков |
| подпись |  | И. О. Фамилия |
|  |  |  |
|  |
|  |  |  |  |
| Нормоконтроль |  |  | Ю.Р.Басиров |
| подпись |  | И. О. Фамилия |

Курсовойпроект защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Иркутск 2022**

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ЗАДАНИЕ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

|  |  |
| --- | --- |
| По курсу | Технологи программирования |
|  |  |
| Студенту | Зыкову Павлу Андреевичу |
|  |  |
| Тема проекта | Проектирование приложения на react с использованием |
|  | базы данных |
| Исходные данные: | Методические указания по курсовому проектированию |
|  |  |
| Задание на курсовую | Разработать сайт используя библиотеку React. Сайт посвящен мини автосалону. Списки автомобилей хранятся в базе данных MySQL. Требуется создать сервер, который будет запускать приложения, получать и отправлять данные в базу данных. Так же разработать клиентскую часть, сделать дизайн сайта, страницы, переход между ними, вывод данных из базы данных. |
|  |  |

Графическая часть на \_\_\_ листах

Дата выдачи задания «23» апреля 2021 г.

Задание получил                             П.А.Зыков

Дата представления проекта руководителю «29» апреля 2022 г.

Руководитель курсового проектирования:                        Ю.Р. Басиров

**Содержание**

[Введение 4](#_Toc101871914)

[Основная часть 5](#_Toc101871915)

[1 Теоретическая часть 5](#_Toc101871916)

[1.1 React 5](#_Toc101871917)

[1.2 Node.js 6](#_Toc101871918)

[1.3 MySQL 7](#_Toc101871919)

[1.4 Python 9](#_Toc101871920)

[1.5 Xampp 10](#_Toc101871921)

[2 Разработка серверной части 11](#_Toc101871922)

[3 Работа приложения 13](#_Toc101871923)

[Заключение 17](#_Toc101871924)

[Список использованных источников 18](#_Toc101871925)

Введение

Невозможно представить современный мир без интернета. Веб-страницы стали более интересными, когда они смогли взаимодействовать с пользователями. На этом этапе возможности сайта расширились от простого предоставления информации до ее сбора. Пользователи могли отправлять формы, настраивать контент под себя и лучше использовать поиск. Для получения необходимой динамики началась активная работа в области программирования веб-приложений.

Прорыв в этой области был достигнут благодаря технологии Common Gateway Interface, или CGI, которая является протоколом для подключения вебсервера к внешнему приложению. Это позволило сайту генерировать основанные на вводе данных страницы и отправлять сгенерированный пользователем результат. Этот протокол был вытеснен другими, более интегрированными решениями.

Существует множество фреймворков, благодаря которым интернет приложения могут развиваться огромными темпами. Современная разработка — это многоуровневый и сложный процесс, который можно разделить на клиентскую и серверную части. А в арсенале frontend- и backend-разработчиков есть множество инструментов. Окончательный выбор средств веб-разработки зависит от сложности проекта и поставленных задач.

React это мощное средство для создания динамических и интерактивных пользовательских интерфейсов. Эта библиотека используется для разработки интерфейсов во многих известных компаниях. Среди них — BBC, Instagram, WhatsApp.

Цель данной курсовой работы разработка приложения на React с использованием базы данных.

Основная часть

1. Теоретическая часть

Готовое приложение будет написано на языке python, с использованием фреймворка react. В качестве базы данных используется —MySQL.

Приложение состоит из 2 частей. Клиент и сервер. Сервер работает с базой данных, получает и отправляет запросы. Клиентская отвечает за дизайн приложения и вывод данных из базы данных.

Приведенная ниже диаграмма отображает работу серверной части приложения с базой данных. Кроме моделей и маршрутов на диаграмме показаны "контроллеры" — функции, которые отделяют код для маршрутизации запросов от кода, который фактически обрабатывает запросы.

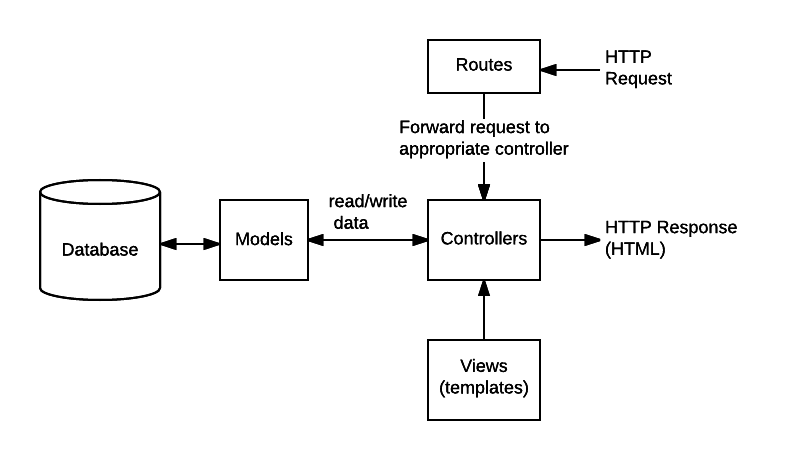


Рисунок 1.1 — Модель работы приложения с базой данных.

* 1. React

React — это библиотека для создания пользовательских интерфейсов. Обратите внимание, что это именно библиотека, а не фреймворк. React часто называют фреймворком, но это ошибка. Во-первых, его использование ни к чему вас не обязывает, не формирует «фрейм» проекта. Во-вторых, React выполняет единственную задачу: показывает на странице компонент интерфейса, синхронизируя его с данными приложения, и только этой библиотеки в общем случае недостаточно для того, чтобы полностью реализовать проект.

Вскоре после появления React и подобные ему решения (Vue.js, Svelte) практически захватили мир фронтенда: потому что они помогают решать проблемы, основываясь на идее декларативного программирования, а не на императивном подходе.

С React-приложением всегда можно смешивать обычное HTML-наполнение, а также другие библиотеки JavaScript. На одной и той же странице также могут находиться сразу несколько React-приложений.[3, с24]

React — это не универсальный инструмент, который подойдёт для любого проекта. Чтобы понять, решит ли React ваши задачи, важно знать о его преимуществах и недостатках. Эта библиотека действительно может упростить жизнь разработчикам:

* С её помощью можно построить интерфейс из отдельных компонентов, которые легко поддерживать.
* Она добавляет удобный слой абстракции, избавляя от необходимости работать с DOM напрямую.
* React — это уже не новая библиотека, за ней стоит крупная компания Facebook и большое сообщество разработчиков. Поэтому она отлично протестирована, регулярно поддерживается и стабильно обновляется, а переход к новым версиям проходит максимально гладко.
* Благодаря сообществу у React хорошо проработанная документация и большой опыт, накопленный в статьях, курсах и конференциях. Это значительно облегчает не только изучение библиотеки новичками, но и поиск ответов на всевозможные вопросы в процессе работы.
* На Гитхабе можно найти готовые React-компоненты почти на все случаи жизни. А если их нет, но есть нужные независимые библиотеки, то можно поискать интеграцию либо сделать её самостоятельно.
* React — это проект с открытым исходным кодом. Благодаря этому его можно безопасно использовать даже в коммерческих приложениях.

Для запуска приложения на react надо иметь node.js.

* 1. Node.js

Node.js — среда выполнения кода вне браузера. Эта платформа позволяет писать серверный код для динамических веб-страниц и веб-приложений, а также для программ командной строки. Она предполагает использование одного языка программирования для разработки веб-приложений вместо применения разных языков для работы над фронтендом и бэкендом.[1]

Node.js — не отдельный язык программирования, а платформа для использования языка программирования на стороне сервера.

Платформа Node.js была представлена в 2009 году. Её создал инженер Райан Дал, а спонсором разработки выступила компания Joyent. Компания известна поддержкой опенсорсных проектов, включая Node.js, Illumos, SmartOS.

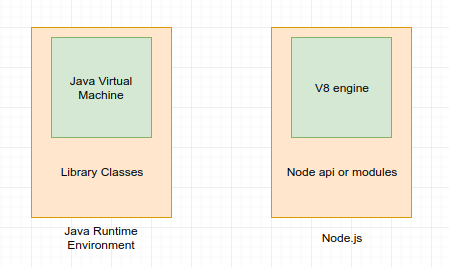


Рисунок 1.2 node.js

Райан Дал использовал для создания Node.js движок V8. Платформа реализована с низкоуровневой неблокирующей моделью ввода/вывода, которая построена на событийно — ориентированной модели. Рисунок 1.2

Node.js применяется для бэкенд-разработки на JavaScript. Если в браузере JavaScript тотально доминирует, и конкурентов этого языка на фронтенде не видно даже на горизонте, то в бэкенд — разработке ситуация другая. Здесь JS конкурирует с PHP, Python и другими языками.

Выбор серверного python для бэкенда обеспечивает проекту ряд преимуществ: рост эффективности разработки благодаря использованию одного языка для фронт — и бэкенда и возможности переиспользования кода; возможность использовать npm — самый большой пакетный менеджер; более простой по сравнению с другими стеками поиск исполнителей, так как python входит в число самых популярных языков программирования.

Node.js хорошо подходит для разработки RTA — веб-приложений, реагирующих на действия пользователя в режиме реального времени. Например, это может быть онлайн-редактор типа Google Docs, который позволяет работать над одним документом нескольким пользователям одновременно.

Node.js легко обрабатывает большое количество запросов одновременно и обеспечивает быстродействие приложения. Поэтому серверный python часто используют для создания SPA — одностраничных веб-приложений, в которых рендеринг выполняется на стороне клиента. Node.js на бэкенде используют Netflix, Uber, eBay, Groupon, Yahoo и другие известные организации, и проекты.

* 1. MySQL

MySQL — это система управления базами данных (СУБД), распространяемая как свободное программное обеспечение (пользователи имеют право на неограниченную установку, запуск, свободное использование). База данных (БД) — набор некоторых данных, которые хранятся в упорядоченной форме.[2]

Часто вебмастера путают понятия базы данных и СУБД. Если база данных — это набор данных, то СУБД — это специальное программное обеспечение, разработанное для управления этими базами данных. Существует множество СУБД, например MySQL, которую я использую в данном проекте.

MySQL — СУБД с открытым исходным кодом, основой которого был код, написанный в Беркли. Она поддерживает большую часть стандарта SQL и предлагает множество современных функций:

* сложные запросы
* внешние ключи
* триггеры
* изменяемые представления
* транзакционная целостность
* многоверсионность

Кроме того, пользователи могут всячески расширять возможности MySQL, например создавая свои

* типы данных
* функции
* операторы
* агрегатные функции
* методы индексирования
* процедурные языки

А благодаря свободной лицензии, MySQL разрешается бесплатно использовать, изменять и распространять всем и для любых целей — личных, коммерческих или учебных.

Говоря техническим языком, MySQL реализован в архитектуре клиент-сервер. Рабочий сеанс включает следующие взаимодействующие процессы (программы):

* Главный серверный процесс, управляющий файлами баз данных, принимающий подключения клиентских приложений и выполняющий различные запросы клиентов к базам данных. Эта программа сервера БД называется postgres.
* Клиентское приложение пользователя, желающее выполнять операции в базе данных. Клиентские приложения могут быть очень разнообразными: это может быть текстовая утилита, графическое приложение, веб-сервер, использующий базу данных для отображения веб-страниц, или специализированный инструмент для обслуживания БД. Некоторые клиентские приложения поставляются в составе дистрибутива PostgreSQL, однако большинство создают сторонние разработчики.

Как и в других типичных клиент-серверных приложениях, клиент и сервер могут располагаться на разных компьютерах. В этом случае они взаимодействуют по сети TCP/IP. Важно не забывать это и понимать, что файлы, доступные на клиентском компьютере, могут быть недоступны (или доступны только под другим именем) на компьютере-сервере.

Сервер MySQL может обслуживать одновременно несколько подключений клиентов. Для этого он запускает («порождает») отдельный процесс для каждого подключения. Можно сказать, что клиент и серверный процесс общаются, не затрагивая главный процесс postgres. Таким образом, главный серверный процесс всегда работает и ожидает подключения клиентов, принимая которые, он организует взаимодействие клиента и отдельного серверного процесса. (Конечно всё это происходит незаметно для пользователя, а эта схема рассматривается здесь только для понимания.)

* 1. Python

Python ‒ мощный компьютерный язык программирования, поддерживающий множество парадигм, который оптимизирован для обеспечения высокой продуктивности программистов, читабельности кода и качества программного обеспечения. [4]

Разработка языка Python началась в конце восьмидесятых годов двадцатого столетия. Для распределенной операционной системы «Amoeba» понадобился расширяемый скриптовый язык, и сотрудник голландского института Гвидо ван Россум начал писать такой язык в свободное время. Уже в тысяча девятьсот девяносто первом году Гвидо опубликовал первый код.

Python — это активно развивающийся скриптовый язык, который используют для решения большого объема самых разноплановых проблем и задач. Python пригодится в создании компьютерных и мобильных приложений, его применяют в работе с большим объемом информации, при разработке web-сайтов и других разнообразных проектов, используют в машинном обучении. Данный язык программирования используют крупные известные корпорации, такие как Spotify и Амазон (например, для анализа данных и создания алгоритма рекомендаций), YouTube и даже Walt Disney. Таким образом, Python нашел свое место в различных областях — с его помощью можно решить множество задач разной сложности.

Разработчики со всего мира полюбили этот язык программирования за множество возможностей:

‒ Объектно-ориентированное программирование (ООП). Реализация ООП в Пайтон хоть и специфична по сравнению с иными объектно-ориентированными языками, но одновременно является неплохо продуманной.

‒ Обобщенное и функциональное программирование.

‒ Модули и пакеты. Программное обеспечение (ПО) на Питоне оформляется в виде модулей, которые могут быть собраны в пакеты.

‒ Интроспекция. Это возможность запросить тип и структуру объекта во время выполнения программы. В Питоне для любого объекта можно узнать всю информацию о его внутренней структуре.

‒ Обработка исключений, итераторы и генераторы.

‒ Управление контекстом выполнения.

‒ Декораторы, регулярные выражения.

Преимущества Python

‒ Язык характеризуется логичным синтаксисом, вследствие чего исходный код программ, написанных «на питоне», легко читается и воспринимается.

‒ Еще одно из преимуществ данного языка программирования — его условная легкость. Он считается наиболее подходящим для начинающих специалистов: разрабатывать несложные программы можно научиться уже спустя пару-тройку дней изучения.

‒ Большое интернет-сообщество. Если разработчик сталкивается с вопросами и трудностями, он всегда может спросить совета у коллег, что значительно ускоряет решение проблем.

‒ Гибкость и масштабируемость. Пайтон позволяет разработчикам адаптировать высокоуровневую логику приложения, что позволяет легко расширять сложные приложения по мере необходимости.

‒ Разработка «на питоне» идет быстрее, чем на большинстве других языках.

‒ Python является интерпретируемым языком программирования. Это значит, что до запуска он представляет собой обычный текстовый файл. Соответственно, программировать можно почти на всех платформах.

Таким образом, Пайтон отлично подойдет для начинающих специалистов. С помощью этого языка можно реализовать практически любой проект.

* 1. Xampp

XAMPP — бесплатный кроссплатформенный дистрибутив для сборки локального веб-сервера. Содержит Apache, MySQL, язык программирования Perl, интерпретатор скриптов PHP и дополнительные библиотеки. Имеет открытый исходный код, простой в установке и использовании. XAMPP - это всего лишь набор программ. Это как бы "пакет программ", среди которых есть Apache, PHP и MySQL. На рисунке 1.3 изображено окно программы XAmpp.

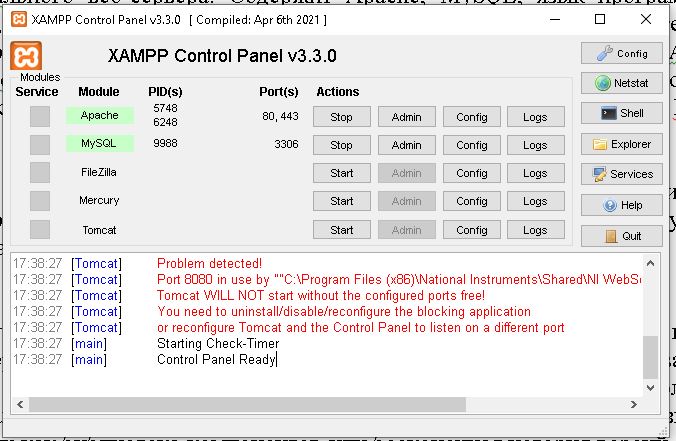


Рисунок 1.3 Окно программы XAMPP

1. Разработка серверной части

В листинге программы прописаны роуты, при http запросе на которые выполняется нужный sql-скрипт и добавляются инструкции если необходимо.

Листинг программы:

1. Импорт настроек и конфиг для подключения к базе данных mysql:

from flask import Flask, jsonify, request

from flask\_mysqldb import MySQL

from flask\_cors import CORS, cross\_origin

import MySQLdb.cursors

app = Flask(\_\_name\_\_)

CORS(app)

app.config['MYSQL\_USER'] ='root'

app.config['MYSQL\_PASSWORD'] = ''

app.config['MYSQL\_HOST'] = '127.0.0.1'

app.config['MYSQL\_DB'] = 'cars'

app.config['MYSQL\_CURSORCLASS'] = 'DictCursor'

app.config['JSON\_AS\_ASCII'] = False

app.secret\_key = 'some secret key'

mysql = MySQL(app)

1. Роут для получения данных авто в наличии с привязкой второй таблицы для определения какие авто уже забронированы:

@app.route("/shop", methods=['GET','OPTIONS'])

@cross\_origin(origin='\*')

def lyrics():

cur = mysql.connection.cursor()

cur.execute('select \* from (SELECT s.\*, o.id o\_id, ROW\_NUMBER() OVER(PARTITION BY s.id) AS row\_num FROM cars.shop s left join cars.orders o on s.id = o.carId) t where row\_num = 1;')

res = cur.fetchall()

return jsonify(res)}

1. Роут для регистрации брони на авто, если таковая отсутствует:

@app.route('/createorder', methods=['POST'])

@cross\_origin(origin='\*')

def createorder():

request.get\_json(force=True)

json = request.get\_json(force=True)

if not json['name'] or not json['phone'] or not json['carId'] or not json['carModel'] or not json['price']:

return jsonify("Не все поля были заполнены!")

else:

name = json['name']

phone = json['phone']

carId = json['carId']

carModel = json['carModel']

price = json['price']

cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

cursor.execute('insert into `cars`.`orders` (name, phone, carId, carModel, price) values(% s, % s, % s, % s, % s)', (name, phone, carId, carModel, price))

mysql.connection.commit()

return jsonify('OK')

1. Роут для регистрации заявки на выкуп авто:

@app.route('/createsale', methods=['POST'])

@cross\_origin(origin='\*')

def createsale():

request.get\_json(force=True)

json = request.get\_json(force=True)

if not json['name'] or not json['phone'] or not json['marka'] or not json['model'] or not json['year']:

return jsonify("Не все поля были заполнены!")

else:

name = json['name']

phone = json['phone']

marka = json['marka']

model = json['model']

year = json['year']

cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

cursor.execute('insert into `cars`.`sale` (name, phone, marka, model, year) values(% s, % s, % s, % s, % s)', (name, phone, marka, model, year))

mysql.connection.commit()

return jsonify('OK')

1. Два роута для отображения всех имеющихся броней и всех заявок на выкуп. Для страницы «Статусы»:

@app.route('/getorderslist', methods=['GET'])

@cross\_origin(origin='\*')

def getorderslist():

cur = mysql.connection.cursor()

cur.execute('select \* from cars.orders o left join cars.shop s on s.id = o.carId')

res = cur.fetchall()

return jsonify(res)

@app.route('/getsaleslist', methods=['GET'])

@cross\_origin(origin='\*')

def getsaleslist():

cur = mysql.connection.cursor()

cur.execute('select \* from cars.sale')

res = cur.fetchall()

return jsonify(res)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

1. Работа приложения

При переходе на ресурс localhost:3000 мы попадаем на главную страницу, на которой у нас есть доступ к оформлению заказа, списку заказов и дополнительной странице, с описанием продукции. Главная страница показана на рисунке 3.1

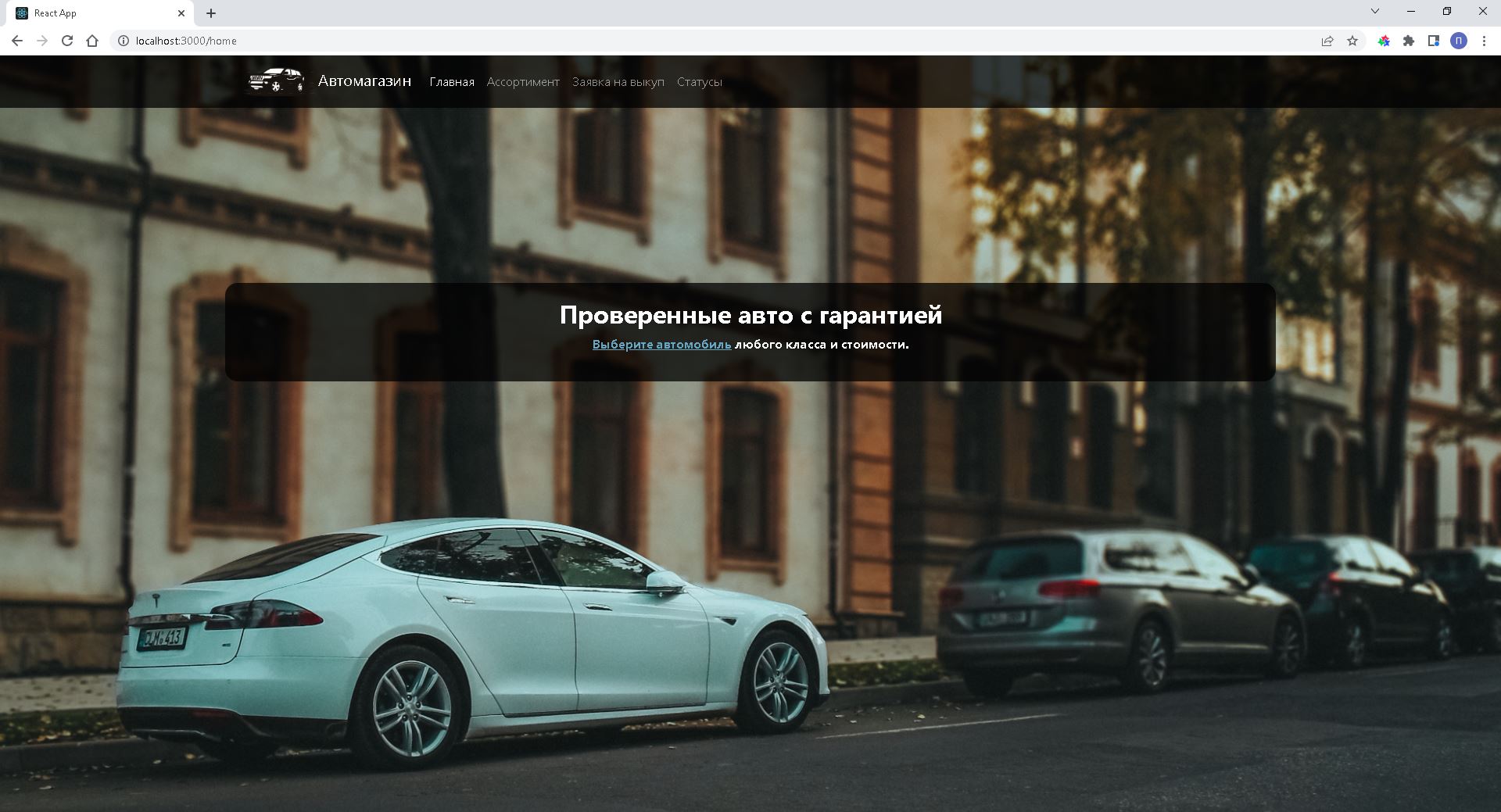


Рисунок 3.1 – Главная страница.

При нажатии на кнопку «Ассортимент», появляется список из базы данных автомобилей в наличии. Страница со списком автомобилей показана на рисунке 3.2.

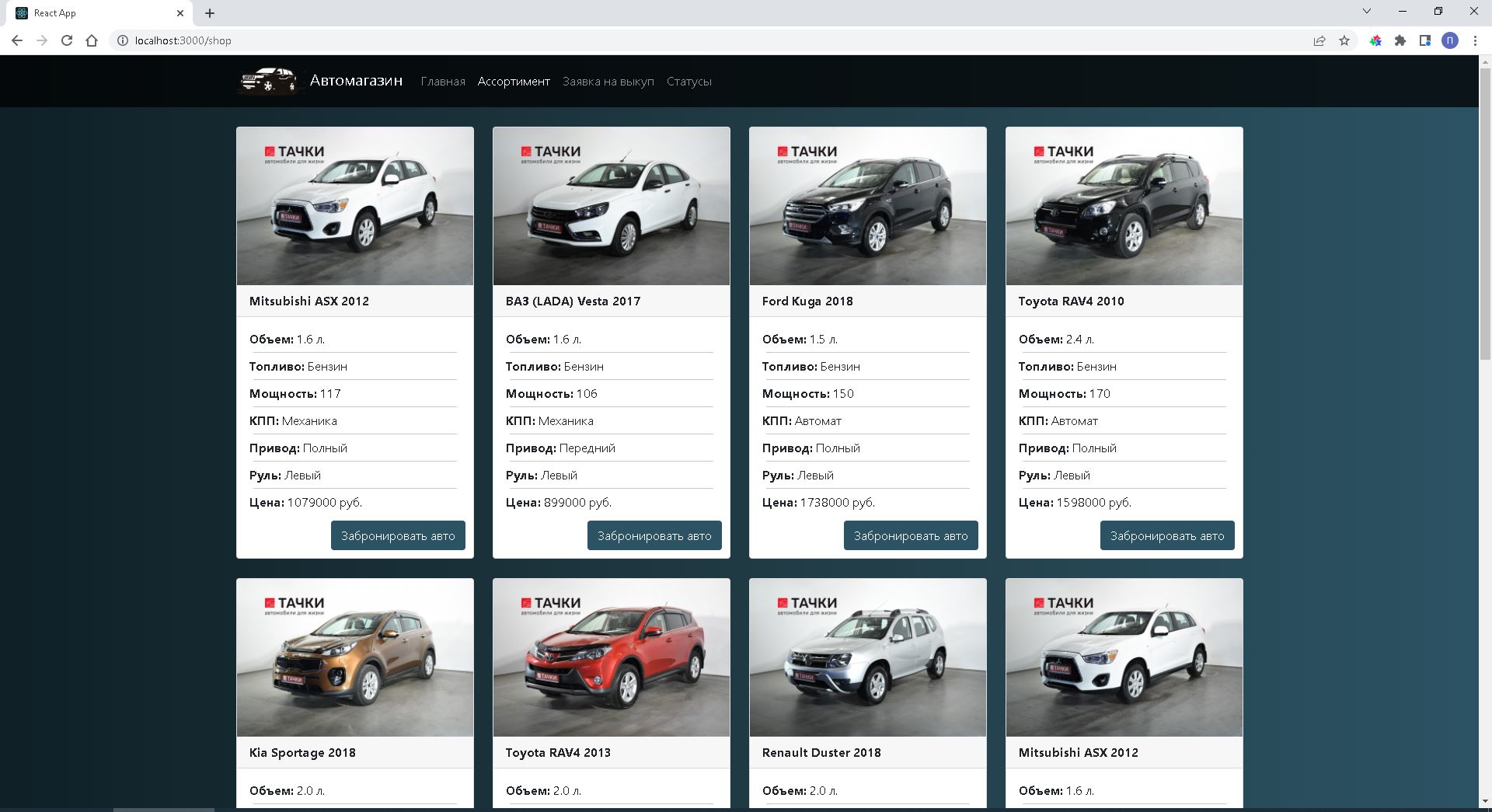


Рисунок 3.2 – Список автомобилей

При нажатии кнопки «Забронировать авто» появляется всплывающее окно с формой в которую нужно ввести контакты для обратной связи. Всплывающее окно «Забронировать авто» показано на рисунке 3.3.

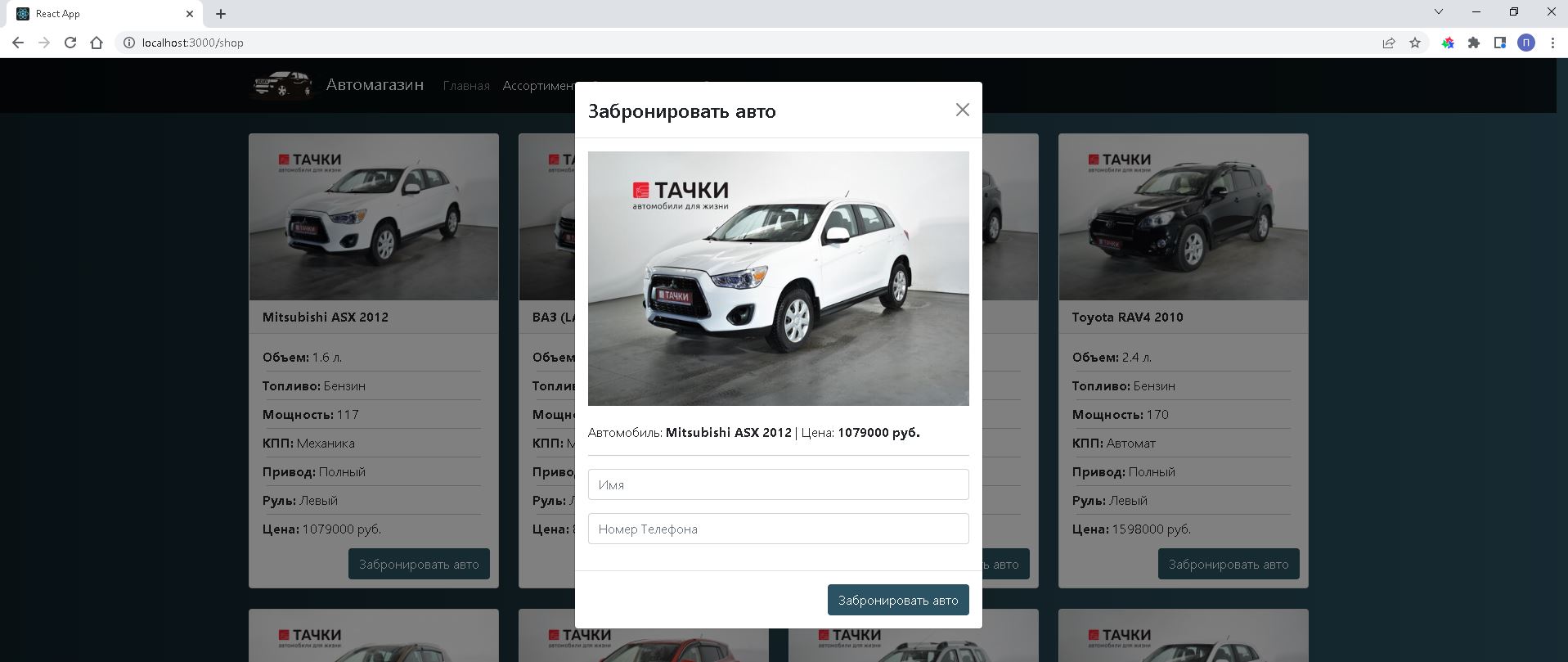


Рисунок 3.3 – Окно «Забронировать авто»

После ввода данные проверяются на правильность и в окне появляется сообщение об успешном бронировании автомобиля. Результат изображен на рисунке 3.4.

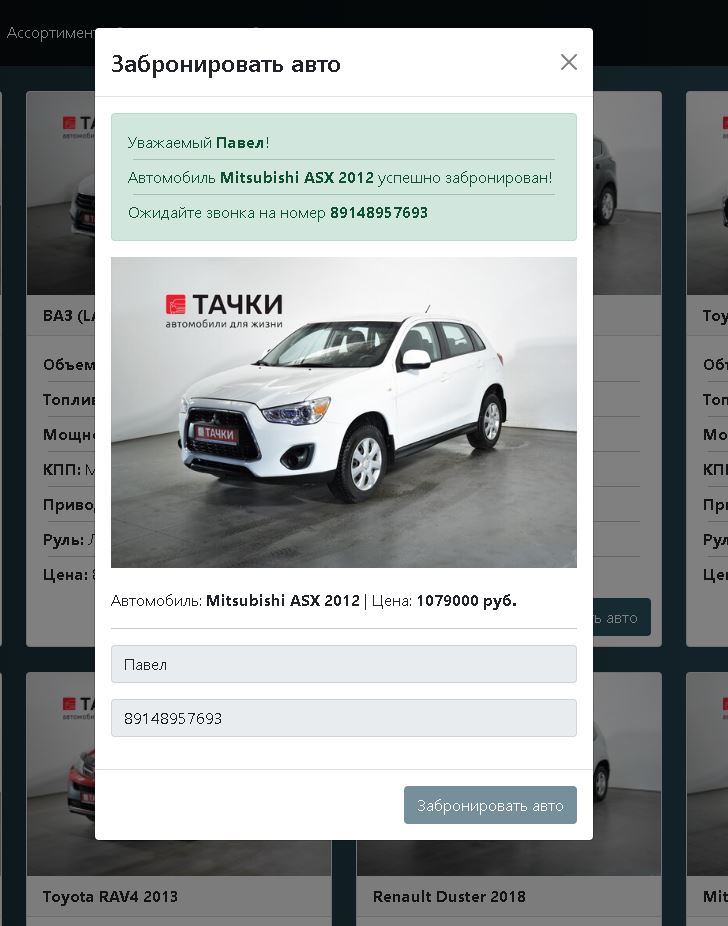


Рисунок 3.4 – успешное бронирование

В списке автомобилей меняется статус автомобиля на «автомобиль забронирован», повторное бронирование невозможно. Результат обновления списка авто изображен на рисунке 3.5.

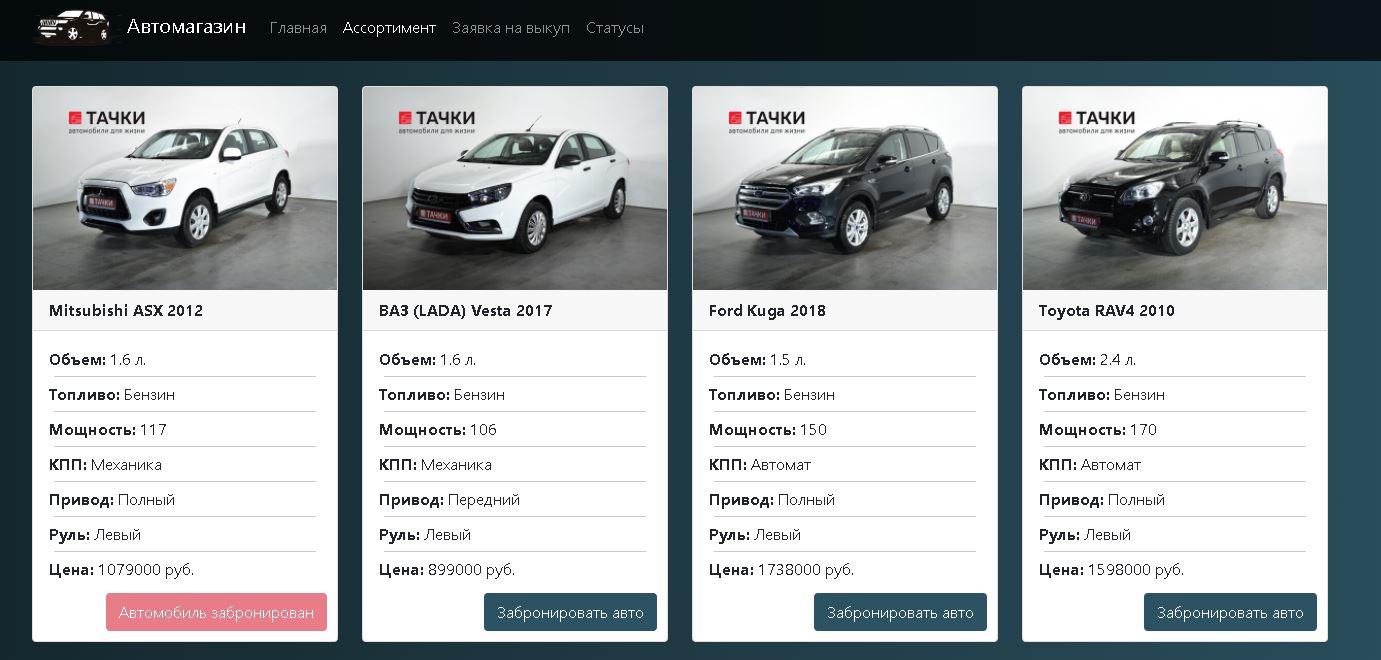


Рисунок 3.5 – Список автомобилей после бронирования

При нажатии на пункт меню «Заявка на выкуп» открывается страница с обязательными для заполнения полями содержащими данные о автомобиле и его владельце. Результат заполнения формы изображен на рисунке 3.6.

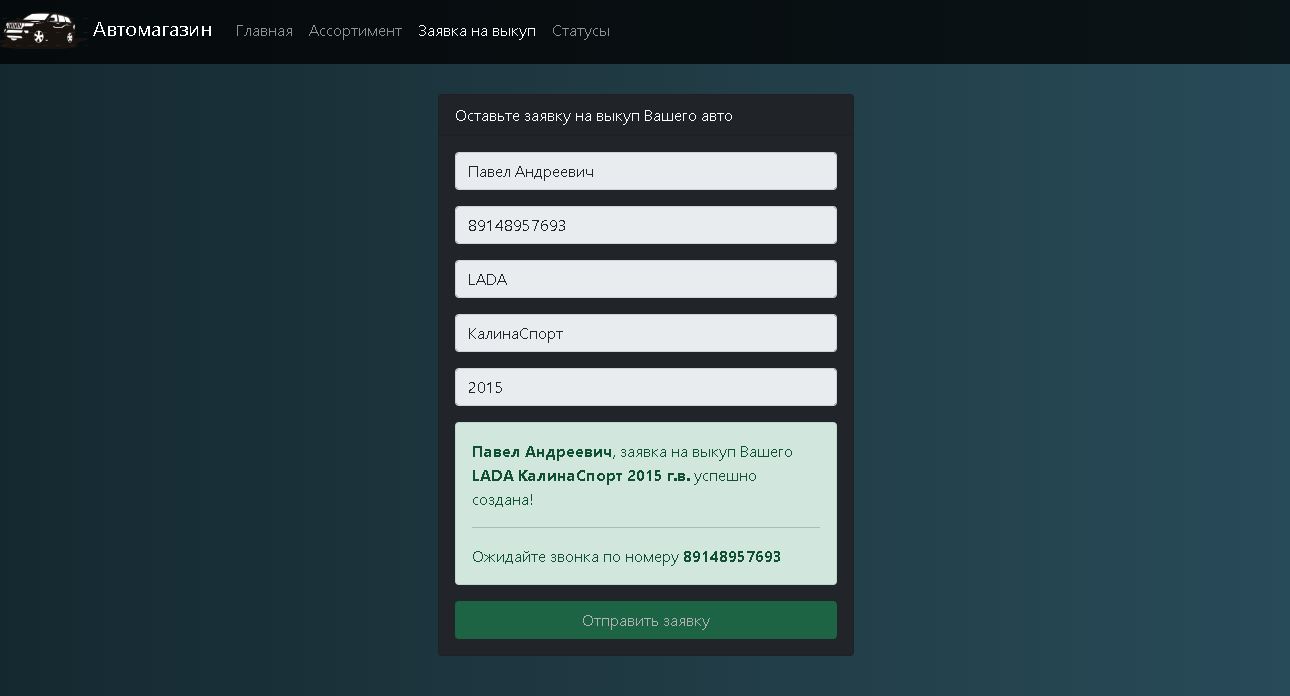


Рисунок 3.6 – заполнение формы «Заявка на выкуп».

При нажатии на пункт меню «Статусы» открывается страница на которой отображаются все забронированные автомобили и все заявки на выкуп авто. Страница «Статусы» изображена на рисунке 3.7.

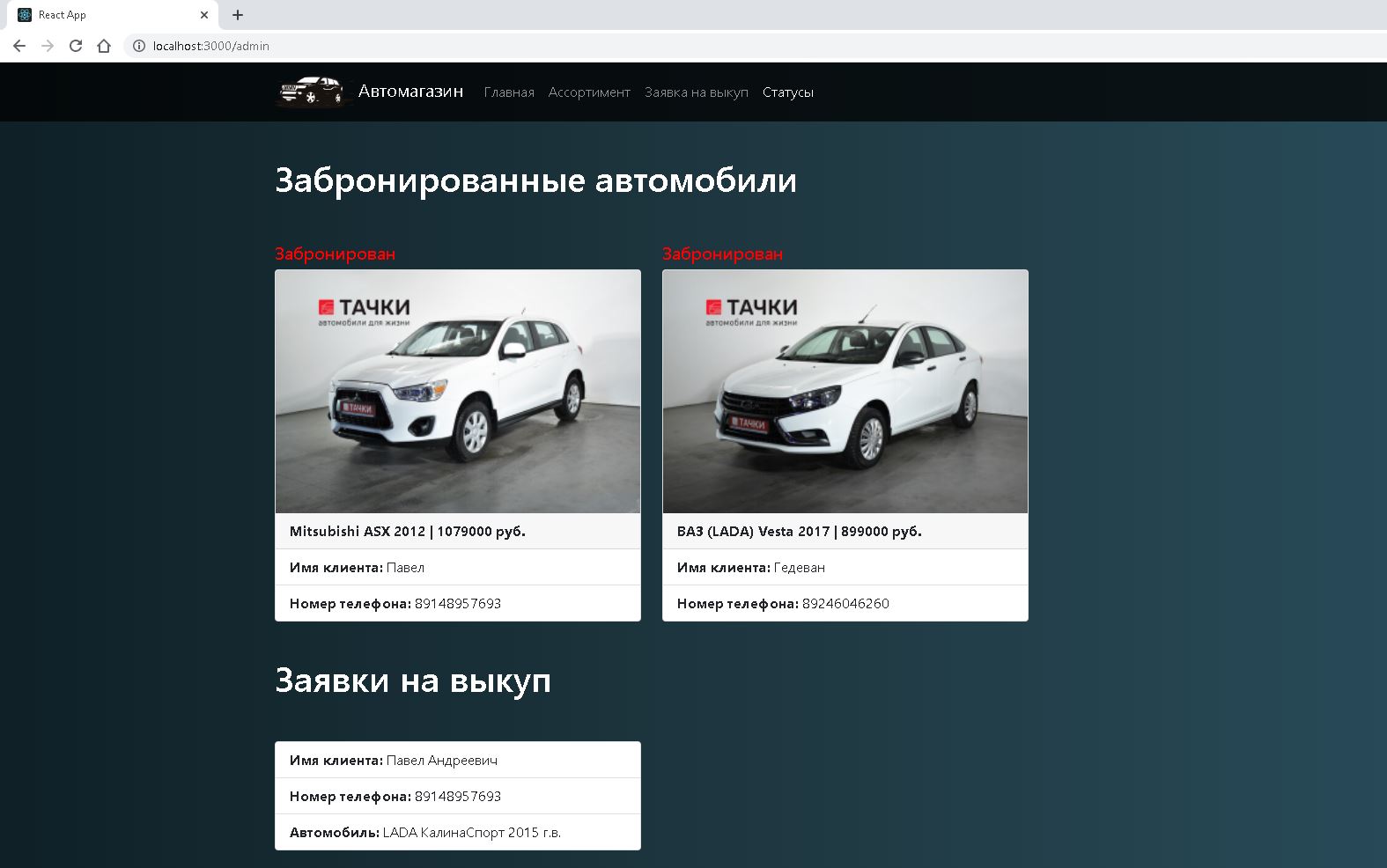


Рисунок 3.7 – Страница «Статусы»

При нажатии на пункт меню «Главная» происходит возврат к главной странице изображенной на рисунке 3.1.

Заключение

В процессе выполнения курсового проекта, нами было реализовано приложение с помощью фреймворка React с использованием базы данных MySQL.

Были получены знания и навыки работы принципа разделения ответственности между внутренней реализацией и внешним представлением.

Разработана клиентская сторона пользовательского интерфейса по отношению к программно-аппаратной части сервиса, разработан пользовательский интерфейс с красивым дизайном, удобной навигацией.

Так же была разработана программно-аппаратная часть сервиса с помощью React, Python и MySQL.

Результаты проведенной работы говорят об освоении теоретического материала и применения его на практике.

Список использованных источников

1. Habr «Node.js» [Электронный ресурс]: Сайт. - URL: https://habr.com/ru/post/420123/ (дата обращения 01.04.2022);
2. Макхост «Что такое MySql» [Электронный ресурс]: Сайт. - URL: https://mchost.ru/articles/chto-takoe-mysql/ (дата обращения 01.04.2022);
3. Стоян Стефанов, «React.js. Быстрый старт.» —Пер.сангл. СПб.: Питер, 2017
4. Лутц,Марк, «ИзучаемPython», том 1 5-еизд.: Пер. с англ.— СПб. : ООО“Диалектика”,2019
5. Flask Documentation [Электронный ресурс]: Сайт. - URL: https://flask.palletsprojects.com/en/2.1.x/ (дата обращения 01.04.2022);
6. Node.js Documentation [Электронный ресурс]: Сайт. - URL: https://nodejs.org/en/docs/ (дата обращения 01.04.2022);
7. React Documentation [Электронный ресурс]: Сайт. - URL: https://ru.reactjs.org/docs/getting-started.html (дата обращения 01.04.2022);