|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**КУРСОВАЯ РАБОТА**

по дисциплине: Шаблоны программных платформ языка Джава

по профилю: Разработка программных продуктов и проектирование информационных систем

направления профессиональной подготовки: 09.03.04 «Программная инженерия»

Тема: Приложение «Каршеринговая компания»

Студент: Утенков Юрий Юрьевич

Группа: ИКБО-36-22

Работа представлена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/Утенков Ю. Ю./

(подпись и ф.и.о. студента)

Руководитель: старший преподаватель Рачков Андрей Владимирович

Работа допущена к защите\_\_\_\_\_\_\_\_\_\_\_(дата)\_\_\_\_\_\_\_\_\_/Рачков А.В./

(подпись и ф.и.о. рук-ля)

Оценка по итогам защиты: \_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

М. РТУ МИРЭА. 2024

**Здесь должно быть задание КР.**

**РЕФЕРАТ**

Отчёт 32 с., 25 рис., 17 источников.

ИНТЕРНЕТ, РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ПРИЛОЖЕНИЯ, SPRING FRAMEWORK, SPRING BOOT, SPRING DATA JPA, SPRING MVC, SPRING SECURITY, КАРШЕРИНГОВАЯ КОМПАНИЯ, ВЕБ-ПРИЛОЖЕНИЕ, MAVEN, GRADLE.

Данная курсовая работа посвящена созданию веб-приложения на тему “Каршеринговая компания” на языке программирования Java с использованием фреймворка Spring. В ходе работы проведен анализ предметной области, обоснован выбор технологий разработки, реализовано приложение и проведено его тестирование.

С помощью метода сравнительного анализа выявлены ключевые решения, которые определяют стиль и функциональность приложения.

Результатом работы стало успешное создание функционального и надежного приложения, способного эффективно решать задачи аренды автомобилей в Москве..

Report, 32 pages, 25 figures, 17 sources.

INTERNET, SERVER APPLICATION DEVELOPMENT, SPRING FRAMEWORK, SPRING BOLT, SPRING DATA JPA, SPRING MVC, SPRING SECURITY, CARSHARING COMPANY, WEB APPLICATION, MAVEN, GRADLE.

This course work is devoted to the development of a web application on the topic of "Carsharing company" using the Java programming language and the Spring framework platform. Throughout the project, the subject area was analyzed, the choice of development technologies was justified, the application was implemented and tested.

Using the method of comparative analysis, key solutions were identified that determine the style and functionality of the application.

The result of the work was the successful creation of a functional and reliable application capable of effectively solving car rental tasks in Moscow.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc166358539)

[1 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ 6](#_Toc166358540)

[1.1 Анализ предметной области 6](#_Toc166358541)

[1.2 Структура базы данных 9](#_Toc166358542)

[2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ 11](#_Toc166358543)

[2.1 Используемое прикладное программное обеспечение 11](#_Toc166358544)

[2.2 Используемые технологии 12](#_Toc166358545)

[3 РАЗРАБОТКА 15](#_Toc166358546)

[3.1 Структура проекта 15](#_Toc166358547)

[3.2 Разработка серверной части приложения 16](#_Toc166358548)

[3.2.1 Структура серверной части приложения 16](#_Toc166358549)

[3.2.2 Файл application.properties 17](#_Toc166358550)

[3.2.3 Класс SecurityConfig 17](#_Toc166358551)

[3.2.3 Класс ClientsService 18](#_Toc166358552)

[3.2.4 Класс AdminsService 19](#_Toc166358553)

[3.2.5 Класс CarsService 20](#_Toc166358554)

[3.3 Разработка клиентской части приложения 21](#_Toc166358555)

[3.3.1 Главная страница приложения 22](#_Toc166358556)

[3.3.2 Страница профиля клиента 22](#_Toc166358557)

[3.3.3 Страница доступных автомобилей 23](#_Toc166358559)

[3.3.4 Страница профиля автомобиля 24](#_Toc166358560)

[3.3.5 Страница о нас 24](#_Toc166358561)

[3.3.6 Страница входа и регистрации 25](#_Toc166358562)

[4 КОНТЕЙНЕРИЗАЦИЯ 27](#_Toc166358563)

[4.1 Использование Docker 27](#_Toc166358564)

[ЗАКЛЮЧЕНИЕ 30](#_Toc166358565)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc166358566)

ВВЕДЕНИЕ

В современном мире каршеринговые компании играют ключевую роль в формировании удобной и доступной системы передвижения для жителей городов. С ростом городов и повышением уровня жизни, спрос на удобные и экономичные способы передвижения становится все более актуальным. В условиях такой динамики развития, каршеринговые компании играют важную роль, обеспечивая мобильность населения современными, функциональными и экономически эффективными услугами.

Цель данной курсовой работы – разработка полноценного приложения для каршеринговой компании, которое позволит пользователям легко находить автомобили в их распоряжении и бронировать их по своему усмотрению. Основными инструментами при разработке данного приложения были база данных PostgreSQL, язык программирования Java, фреймворк для создания веб-приложений Spring Framework, а также Spring Data JPA для работы с ORM-отображением данных.

Спроектированная работа включает в себя фронтенд-часть приложения, написанную с помощью HTML, CSS, и бэкенд-часть, созданную с использованием вышеперечисленных технологий. Приложение поможет пользователям легко находить автомобили в различных районах города, сравнивать условия аренды и фильтровать их под свои предпочтения.

1 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

* 1. Анализ предметной области

Перед тем, как начать разработку приложения, необходимо провести анализ предметной области и ознакомиться с сайтами, предоставляющими услуги каршеринга, чтобы полностью покрыть требования пользователей к функциональности разрабатываемого сайта. Для этого были изучены три сайта каршеринг-компаний:

– <https://delimobil.ru>,

– <https://belkacar.ru>,

– <https://yandex.ru/drive>.

Веб-ресурс <https://delimobil.ru> [1] имеет лаконичный и интуитивно понятный интерфейс, что позволяет пользователям быстро ориентироваться на сайте и находить нужную информацию. Данный ресурс дает возможность просмотреть и основную информацию об автомобилях, а также предоставляет специализированную фильтрацию поиска подходящего автомобиля, что делает информацию персонализированной (рисунок 1).

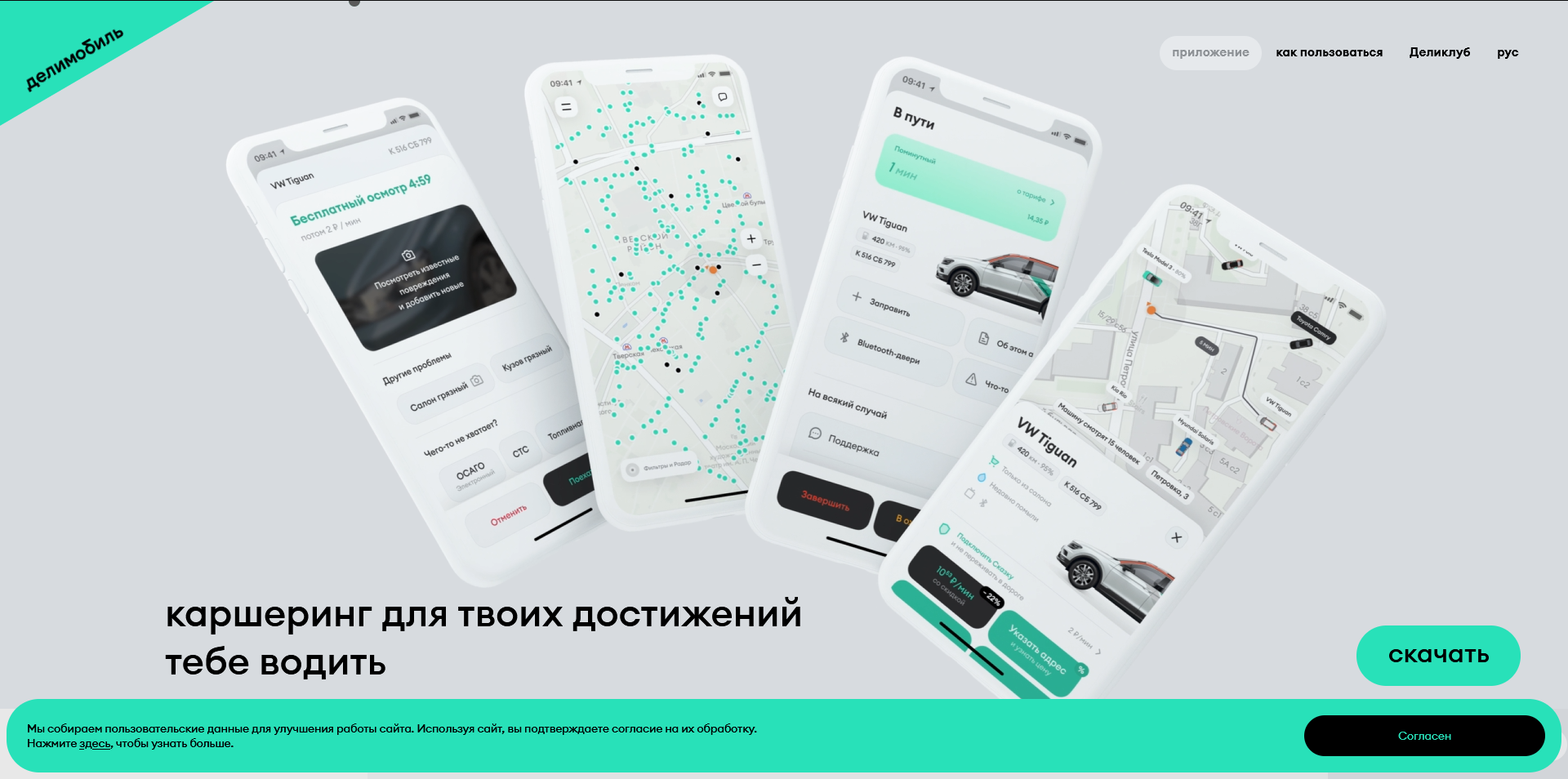


Рисунок 1 – Главная страница сайта <https://delimobil.ru>

Веб-ресурс <https://belkacar.ru> [2] представляет собой платформу каршеринговой компании BelkaCar, предлагающую услуги краткосрочной аренды автомобилей через мобильное приложение. Сервис доступен в Москве, Сочи, Краснодарском крае, Санкт-Петербурге, Казани и Калининграде. На сайте можно найти информацию о различных тарифах, правилах использования, а также контактные данные компании. BelkaCar входит в число крупнейших каршеринговых сервисов России, предлагая удобные условия аренды и широкий выбор автомобиле (рисунок 2).

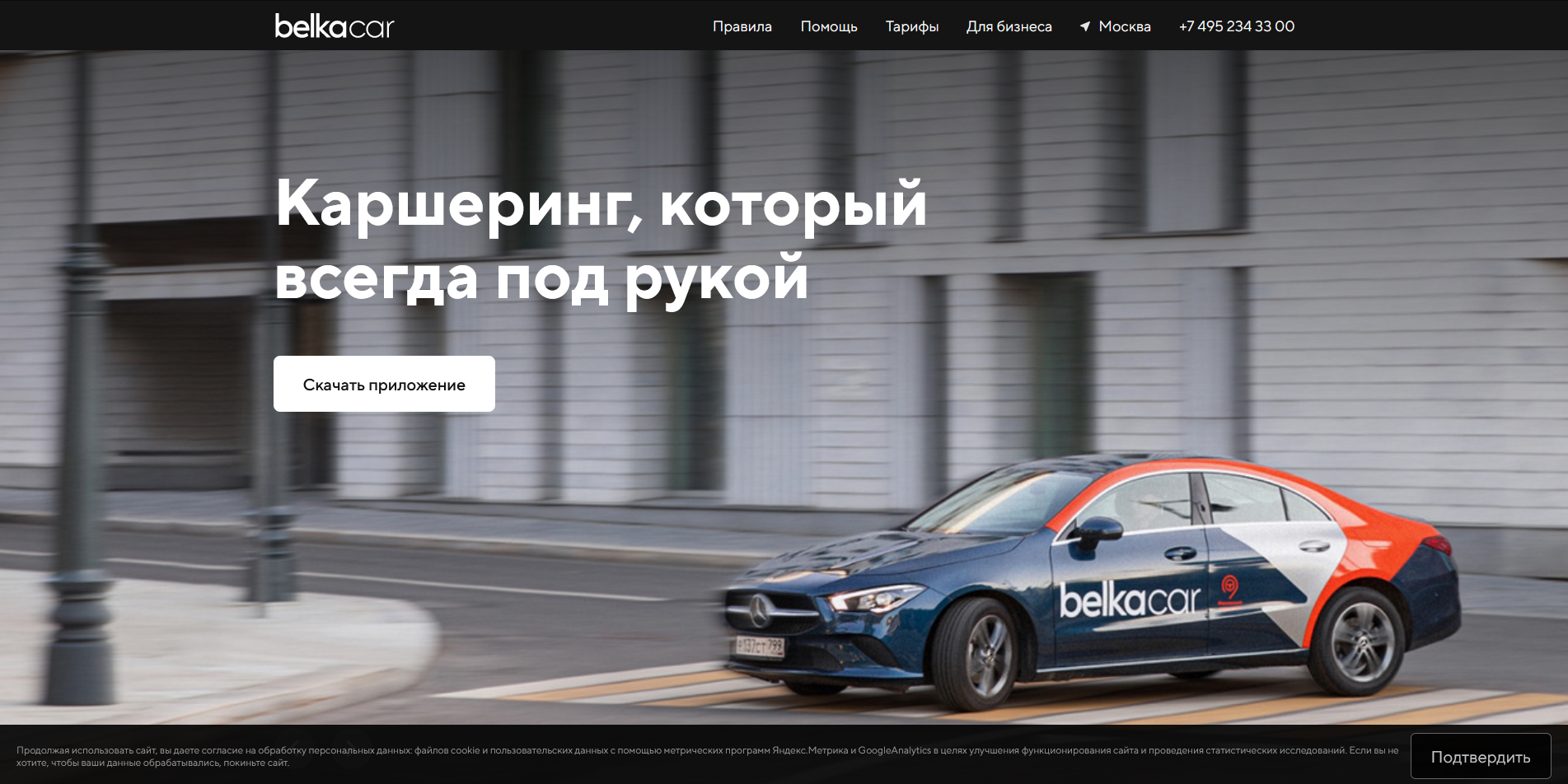


Рисунок 2 – Главная страница сайта [https://belkacar.ru](https://belkacar.ru/)

Веб-ресурс <https://yandex.ru/drive> [3] представляет собой сервис Яндекса, предназначенный для организации и управления файлами и документами в облаке. Он позволяет пользователям хранить, обмениваться и совместно работать над файлами, используя облачное хранилище Яндекса. Сервис обеспечивает доступ к файлам из любого устройства с интернет-соединением, что делает его удобным инструментом для работы с документами в режиме реального времени (рисунок 3).

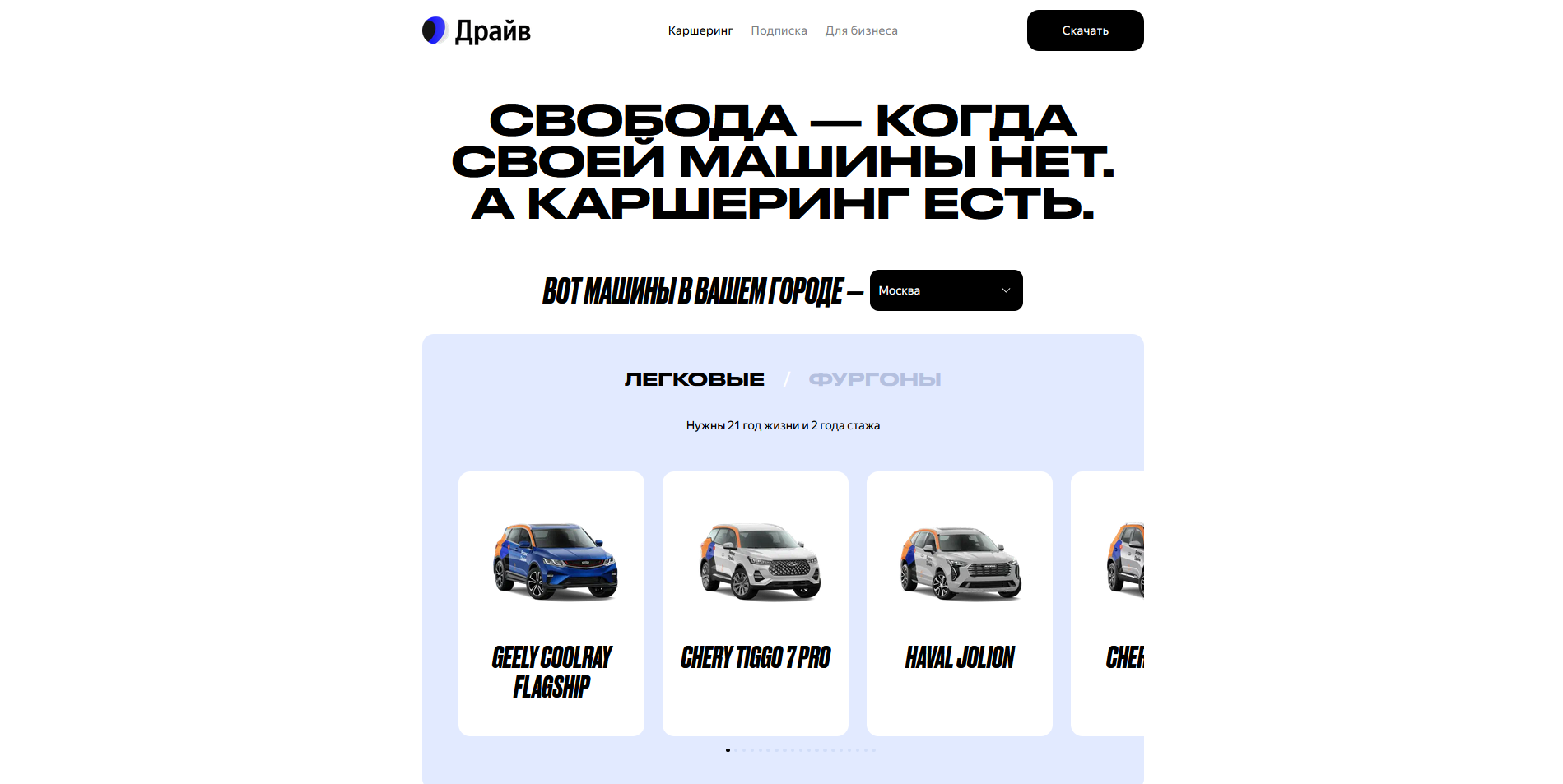


Рисунок 3 – Главная страница веб-ресурса <https://yandex.ru/drive>

Для наглядного сравнения трех различных сайтов на общую тематику, построим таблицу 1, в которой отражены основные критерии оценки и их характеристики.

Таблица 1 – Сравнение характеристик сайтов аналогов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сравнительные показатели | Просмотр доступных автомобилей | Фильтрация автомобилей | Возможность связи с оператором | Возможность выбора места забора/возврата | Возможность онлайн оформления заявки |
| [https://delimobil.ru](https://delimobil.ru/) | 5 | 4 | 5 | 3 | 4 |
| [https://belkacar.ru](https://belkacar.ru/) | 3 | 5 | 5 | 3 | 5 |
| <https://yandex.ru/drive> | 4 | 5 | 3 | 5 | 2 |

Исходя из сопоставительного анализа особенностей схожих веб-сайтов, можно сделать вывод, что проектируемое приложение для каршеринга должно иметь следующие функции:

1. просмотр доступных автомобилей,
2. фильтрация,
3. возможность удобно арендовать,
4. регистрация в личном кабинете.

1.2 Структура базы данных

Для проектирования системы следует определить ключевые сущности.

Таблица clients содержит захешированный пароль пользователя, его email, а также поле name, отвечающее за имя клиента (рисунок 4).

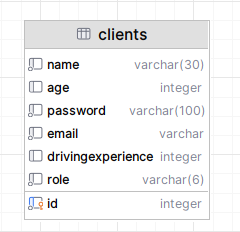


Рисунок 4 – Сущность базы данных для хранения информации о пользователе

Для хранения информации о автомобилях существует таблица car, которая содержит поле clients\_id, которое является внешним ключом к таблице clients, поле images, отвечающее за уникальную картинку для данного автомобиля, поле name, отвечающее за название конкретного автомобиля, поле description, необходимое для описания автомобиля, поле price\_per\_min, необходимое для хранения информации о цене аренды автомобиля поминутно, поле drive, нужное для хранения типа привода автомобиля, и остальные поля для полноценного описания сущности car. (рисунок 5).

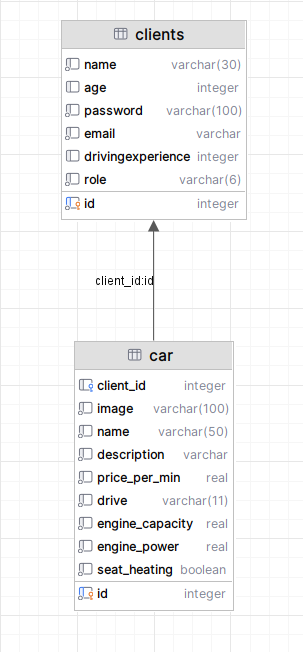


Рисунок 5 – Сущность базы данных информации о автомобилях

2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИЙ

2.1 Используемое прикладное программное обеспечение

Использование IntelliJ IDEA для разработки курсовой работы предоставляет множество преимуществ [4]. Эта интегрированная среда разработки обеспечивает удобство настройки проекта, интеграцию с Spring Framework и поддержку различных технологий, таких как Maven и Gradle. С помощью IntelliJ IDEA легко создавать и настраивать проекты, автоматически добавлять зависимости и конфигурировать контекст приложения. Кроме того, данная среда предоставляет широкий набор инструментов для рефакторинга кода и отладки приложений, что повышает производительность и облегчает разработку. Все необходимые инструменты доступны в одной удобной среде, что упрощает процесс разработки и позволяет сосредоточиться на создании качественного программного обеспечения.

В данной курсовой работе также используется Docker – инструмент для контейнеризации приложений [5]. С его помощью приложения становятся независимыми от инфраструктуры, что делает возможным масштабируемость и перенос кода на другую систему без необходимости переносить все компоненты. Docker создает изолированный контейнер с приложением, который легко можно развернуть в системах, на которых установлена данная технология.

Также при создании проекта курсовой работы были использованы браузеры Yandex, Firefox.

Для распределенного контроля версий использовалась система Git, поскольку она обеспечивает ветвление, помогающее в ведении параллельной разработки [6]. Ветвление позволяет сохранить полную историю изменений в проекте и восстановить данные при необходимости из старых версий.

Выбор данных инструментов обеспечивает полную гибкость, масштабируемость и безопасность при разработке проекта, а также создает максимально удобный способ развернуть приложение.

2.2 Используемые технологии

В данной курсовой работе использование языка программирования Java обусловлено факторами кроссплатформенности и обширной поддержки.

Java – это кроссплатформенный язык программирования, что означает, что программы, написанные на Java [7], могут выполняться на различных операционных системах без необходимости переписывания кода. Это обеспечивает широкую доступность и переносимость программ.

Одним из ключевых преимуществ Java является его надежность и безопасность [8]. Java обеспечивает строгую типизацию и проверку на ошибки во время выполнения, что помогает предотвращать многие типичные ошибки программирования. Благодаря системе управления памятью, основанной на сборке мусора, Java также обеспечивает автоматическое освобождение памяти и предотвращает утечки памяти [9].

Maven – это инструмент управления проектами и сборки программного обеспечения, который позволяет разработчикам эффективно управлять зависимостями, сборкой и документацией проекта [10]. Он основан на концепции проектно-ориентированного управления зависимостями (Project Object Model, POM), что обеспечивает структурирование проекта и автоматическую загрузку необходимых библиотек и плагинов из центрального репозитория.

Spring Framework является большим набором библиотек, а также инструментов для разработки веб-приложений [11]. Spring является контейнером, или контекстом приложения, внутри которого создаются и взаимодействуют друг с другом компоненты. Инверсия управления и внедрение зависимостей делают фреймворк Spring масштабируемым, гибким и удобным в использовании.

В данной курсовой работе также используются дополнительные части фреймворка Spring, такие как Spring MVC, Spring Security и Spring Data JPA.

Spring MVC — это ключевой компонент Spring Framework, предназначенный для создания веб-приложений, который организует разработку вокруг трех основных компонентов: модели (отвечающей за обработку данных и бизнес-логику), представления (отображающего данные пользователю) и контроллера (обрабатывающего HTTP-запросы и взаимодействующего с моделью и представлением) [12]. Этот подход позволяет разработчикам четко структурировать проекты, упрощая их масштабирование и поддержку. Spring MVC не только обеспечивает базовую архитектуру для веб-приложений, но и предоставляет богатый набор инструментов и возможностей, включая поддержку RESTful сервисов, интеграцию с базами данных, механизмы безопасности, интернационализацию и многое другое. Это делает его мощным и гибким решением для создания современных, масштабируемых и надежных веб-приложений.

Spring Security отвечает за процесс аутентификации и авторизации в приложении, а также за другие функции, нацеленные на безопасность [13]. С помощью широкого набора инструментов Spring Security защищает приложение от атак, SQL инъекций, подделок запросов. Данная технология необходима в приложении для конфиденциальности данных пользователей, распределения доступа к различным частям приложения.

Spring Data JPA – это также часть фреймворка Spring, конкретнее – Java Persistence API [14]. Она предназначена для высокоуровневой работы с SQL запросами и взаимодействием с базой данных. Репозитории Spring Data JPA позволяют избежать написания SQL запросов самостоятельно, и предлагают уже готовый набор основных методов, осуществляющих эти запросы.

Lombok [15] является инновационным инструментом для разработчиков на Java, который значительно упрощает процесс написания кода за счет автоматической генерации стандартного кода. Этот инструмент использует аннотации для указания компилятору на необходимость автоматического создания методов, таких как геттеры, сеттеры, конструкторы и т.д. Это позволяет программистам сосредоточиться на логике приложения, а не на формальностях, делая код более чистым и легким для понимания. Одним из ключевых преимуществ использования Lombok является значительное уменьшение объема кода. Вместо того чтобы вручную писать множество шаблонных методов, разработчики могут просто добавить соответствующие аннотации к классу или полям, и Lombok автоматически сгенерирует необходимый код. Lombok предлагает ряд дополнительных аннотаций, которые могут быть полезны в различных сценариях разработки. Использование Lombok позволяет значительно улучшить качество кода и повысить производительность разработки. Однако, как и любой инструмент, он требует понимания его возможностей и правильного применения для достижения наилучших результатов. Lombok стал помощником для многих разработчиков Java, помогая им создавать более чистый, эффективный и поддерживаемый код.

PostgreSQL – это база данных с открытым исходным кодом [16]. Это объектно-реляционная система управления базами данных, в которой данные хранятся в виде строк, а столбцы являются различными атрибутами данных. Она позволяет безопасно хранить, обрабатывать и извлекать данные. В курсовой работе база данных хранит данные о жилых комплексах, квартирах, пользователях сайта.

Клиентская часть приложения написана основана преимущественно на HTML и CSS [17]. Они позволяют красиво и качественно отображать различные элементы для удобства пользователя.

3 РАЗРАБОТКА

3.1 Структура проекта

Проект содержит в себе две части: клиентскую и серверную. Клиентская часть приложения реализована при помощи HTML, CSS, серверная часть написана на языке Java с использованием Spring Framework и PostgreSQL для бизнес-логики и хранения данных соответственно.

Проект содержит файлы Dockerfile и docker-compose, которые обеспечивают контейнеризацию и сборку приложения, что делает проект легко масштабируемым и гибким. В проекте также содержатся файлы для сборки с помощью Maven, отвечающие за тестирование, управление зависимостями кода (рисунок 8).

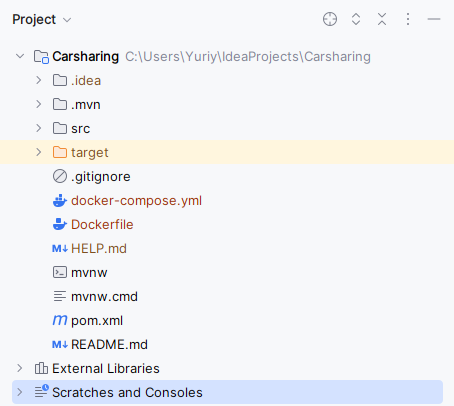


Рисунок 8 – Структура проекта

3.2 Разработка серверной части приложения

3.2.1 Структура серверной части приложения

Серверная часть приложения состоит из следующих пакетов.

В пакете services содержатся классы, отвечающие за бизнес-логику приложения. Сервисы обеспечивают взаимодействие с хранилищем данных, реализацию бизнес-логики, обработку запросов и валидацию данных и другие операции над ними.

Пакет models содержит классы, являющиеся сущностями, на основе которых создаются таблицы в базе данных. Они отвечают за содержание таблиц, их поля и валидацию данных, поступающих от пользователя сайта.

Пакет enums содержит вспомогательные перечисления для сущностей, а также для удобства конфигурации Spring Security. Они применяются для удобства передачи данных в нужном для клиента формате.

Пакет repository предназначен для непосредственного взаимодействия с базой данных. Данный паттерн проектирования является широко используемым в проектах, подобного формата.

Пакет controllers содержит классы-контроллеры, которые обеспечивают корректную обработку HTTP-запросов со стороны клиента. Каждый из контроллеров служит для обработки одного конкретного запроса, а также взаимодействует с сервисами.

В пакете config содержатся классы, отвечающие за конфигурацию компонентов Spring, таких как безопасность, взаимодействие с базой данных, отображение шаблонов.

В пакете validator находится один класс, предоставляющий более продвинутую валидацию, введённых данных пользователем, он позволяет избежать ошибок в программе, и сообщать пользователю, если он введёт некорректные данные.

3.2.2 Файл application.properties

В файле application.properties содержатся данные, необходимые для настройки работы приложения. В данном файле определены способы взаимодействия с базой данных, почтовым сервисом, а также порт приложения и прочие необходимые параметры (рисунок 9).

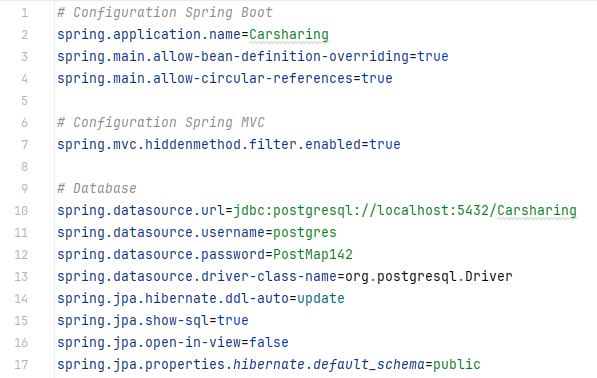


Рисунок 9 – Файл application.properties

3.2.3 Класс SecurityConfig

Данный класс отвечает за настройку и корректную работу Spring Security в приложении. В нем реализован метод, в котором определяются необходимые параметры для аутентификации и авторизации пользователя, а также безопасность приложения (рисунок 10).

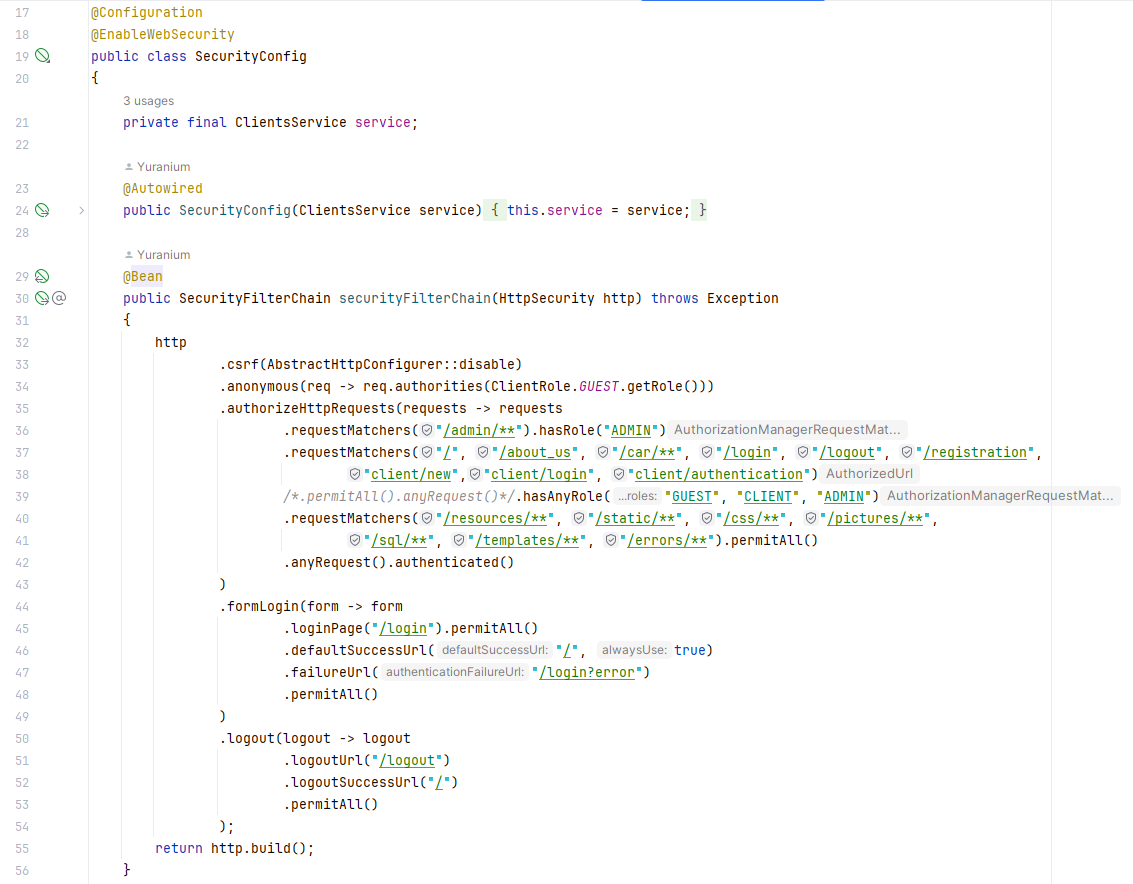


Рисунок 10 – Реализация класса SecurityConfig

3.2.3 Класс ClientsService

В классе-сервисе ClientsService реализованы методы, предназначенные для совершения различных действий с регистрацией новых клиентов, аутентификацией, идентификацией. Данный класс предоставляет операции поиска клиента по его имени, ID, сохранение клиентов базу данных. планировок по названию жилого комплекса, поиск всех планировок, а также их фильтрацию по всем параметрам. Класс ClientsService использует Spring Data JPA, что делает его абстрактным для работы с базой данных и обеспечивает инкапсуляцию логики кода (рисунок 11).



Рисунок 11 – Код класса ClientstService

3.2.4 Класс AdminsService

Класс AdminsService также является сервисным компонентом приложения. Он обеспечивает работу с клиентам-админами данного сервиса. В нем реализованы похожие методы, которые помогают реализовывать слой логики для данных пользователей (рисунок 12).

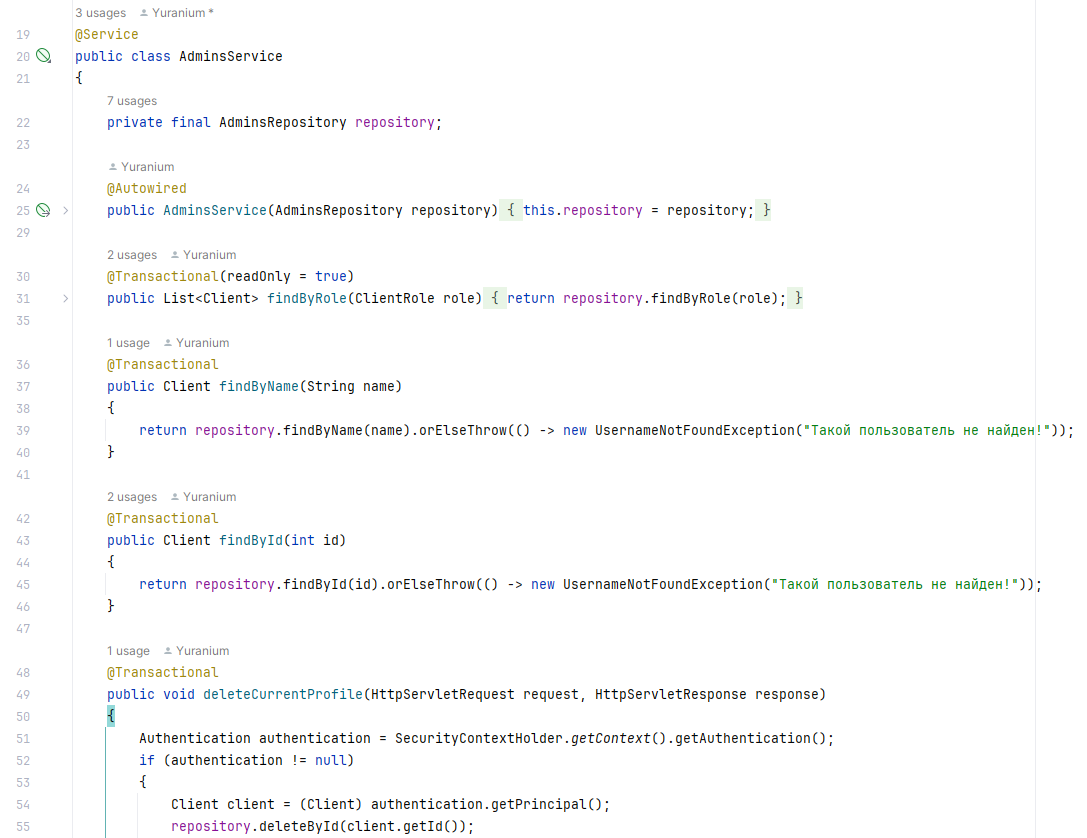


Рисунок 12 – Код класс AdminsService

3.2.5 Класс CarsService

Класс-сервис CarsService отвечает за осуществление основных операций взаимодействия с базой данных для автомобилей. Доступные в этом классе методы предоставляют возможность поиска всех автомобилей на данный момент, поиска определённого автомобиля по его ID, а также возможность арендовать автомобиль (рисунок 13).

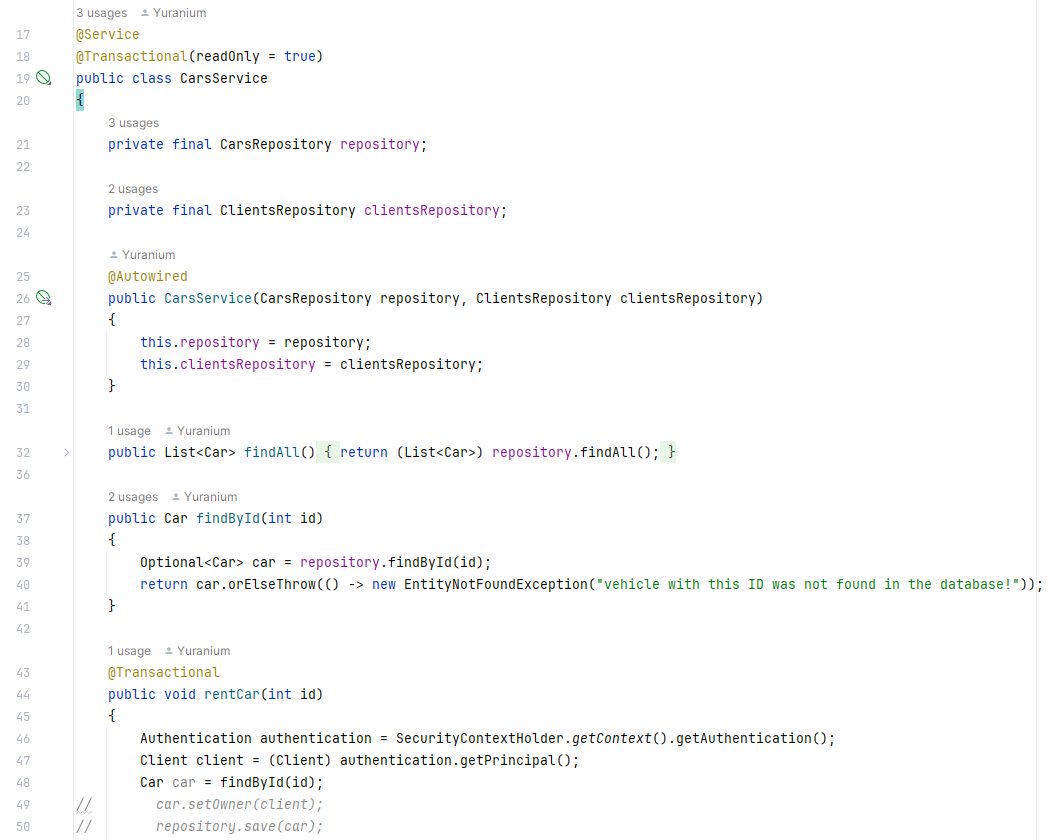


Рисунок 13 – Код класса CarsService

3.3 Разработка клиентской части приложения

Клиентская часть приложения расположена в папках resources/static и resources/templates. Все материалы, расположенные в данной директории, представляют собой изображения, файлы с кодом HTML и CSS.

На главной странице предоставлена информация о каршеринг-компании, а также удобная навигация по доступным страницам. На странице «Просмотр авто» доступны все автомобили, находящиеся на данный момент в каталоге. Страница «О нас» показывает пользователю краткую информацию, о данной каршеринг-компании.

Страница «Зарегистрироваться» дает пользователю возможность зарегистрироваться или войти в учетную запись и получить доступ к заполнению заявки на аренду автомобиля. Благодаря настройке Spring Security, доступ к аренде автомобиля возможен только аутентифицированным пользователям.

3.3.1 Главная страница приложения

На главной странице представлена навигация по сайту и информация о каршеринг-компании (рисунок 16).

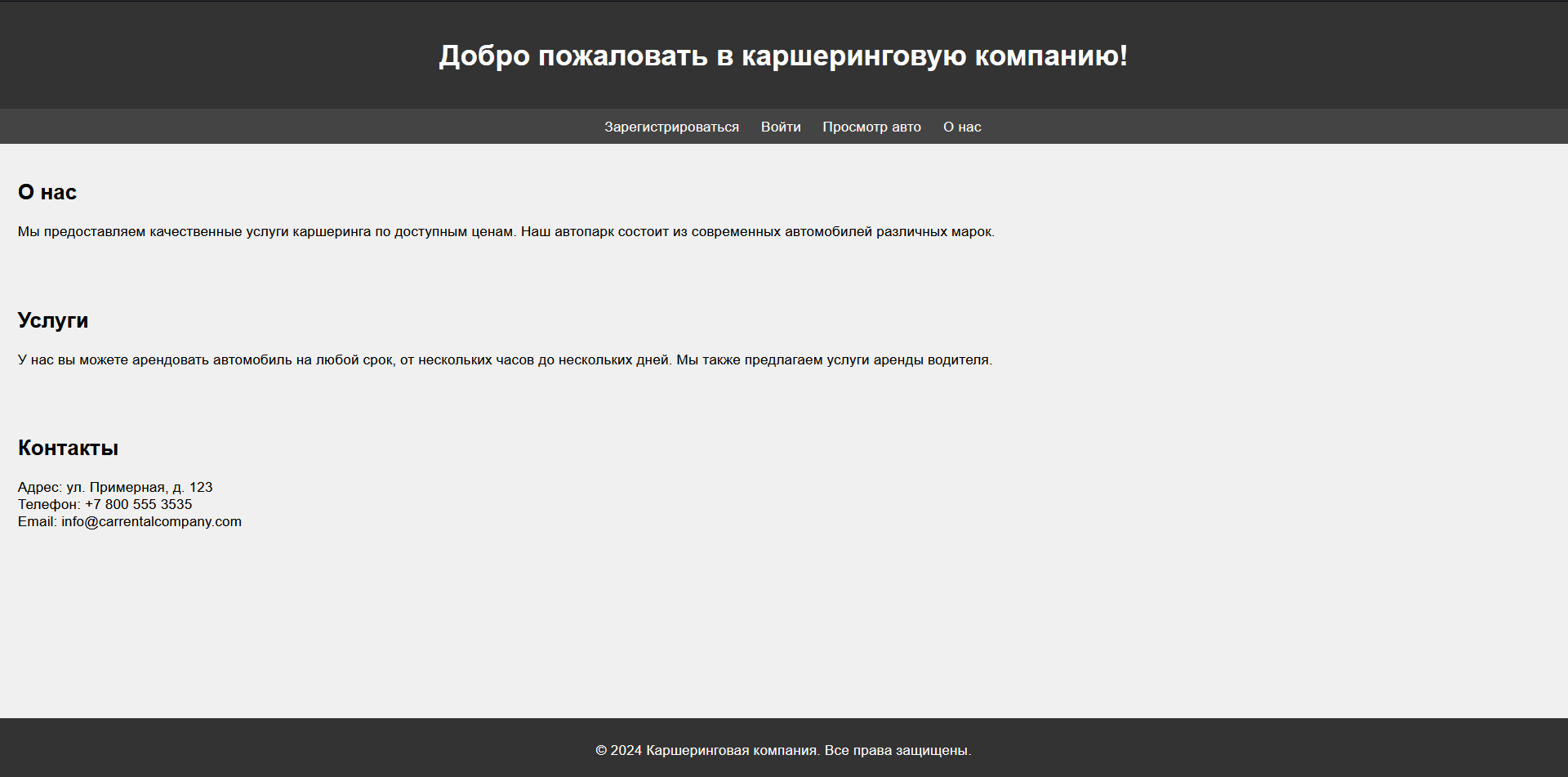


Рисунок 16 – Главная страница сайта

3.3.2 Страница профиля клиента

На данной странице представлена информация о профиле клиента, с возможность удалить профиль или изменить данные, введёные при регистрации. У клиента, с ролью админ, доступна функция просмотра всех зарегистрированных пользователей, а также возможность их удалить. У клиента, с ролью клиент, будет показана информация обо всех арендованных автомобилях (рисунок 17).

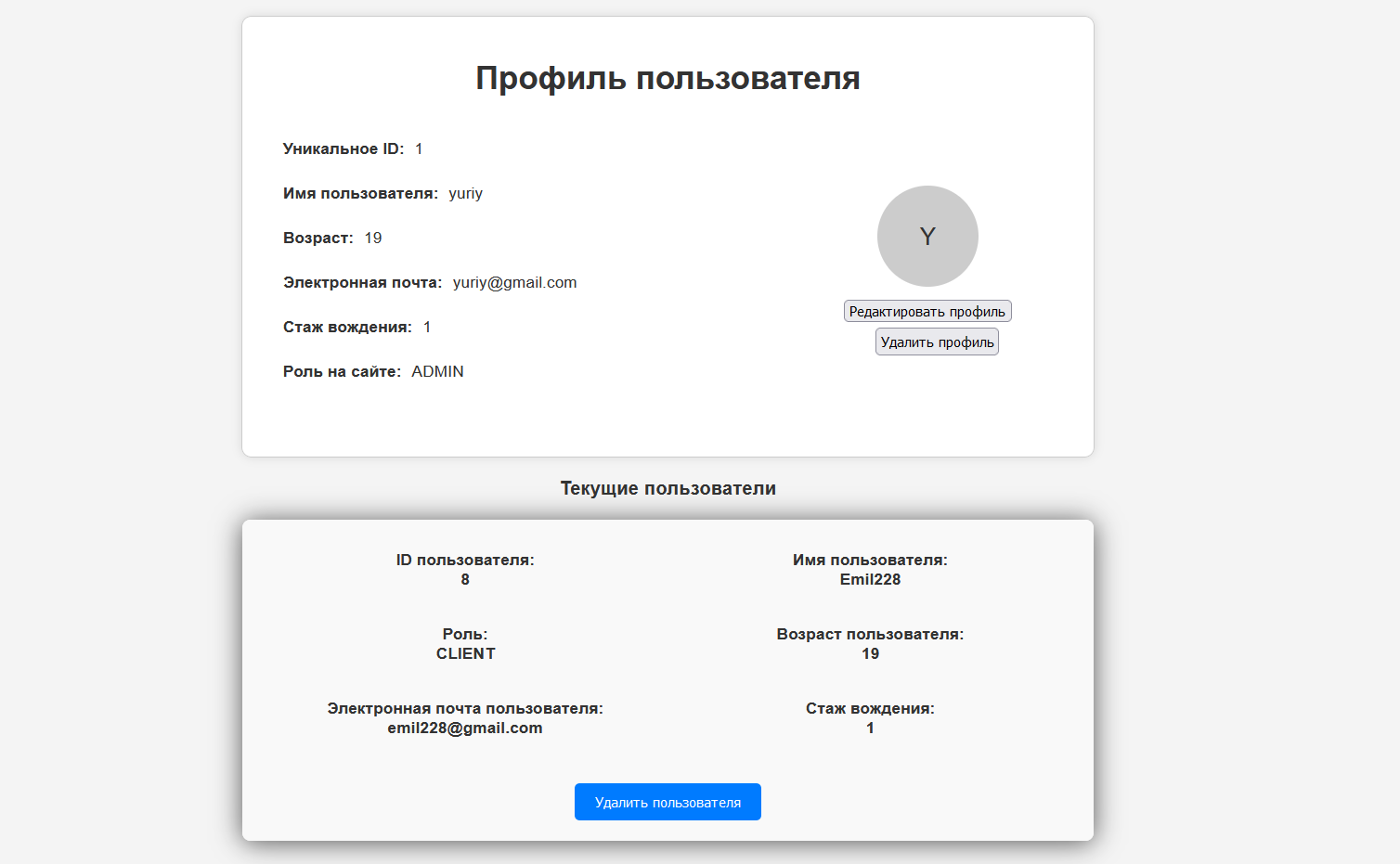


Рисунок 17 – Страница профиля пользователя

3.3.3 Страница доступных автомобилей

На странице доступных автомобилей представлен список доступных автомобилей (рисунок 18).



Рисунок 18 – Страница с каталогом всех автомобилей

3.3.4 Страница профиля автомобиля

На странице профиля автомобиля представлена информация о цене по минутам, вид привода, объём двигателя, и прочая информация, касающаяся данного автомобиля (рисунок 19).

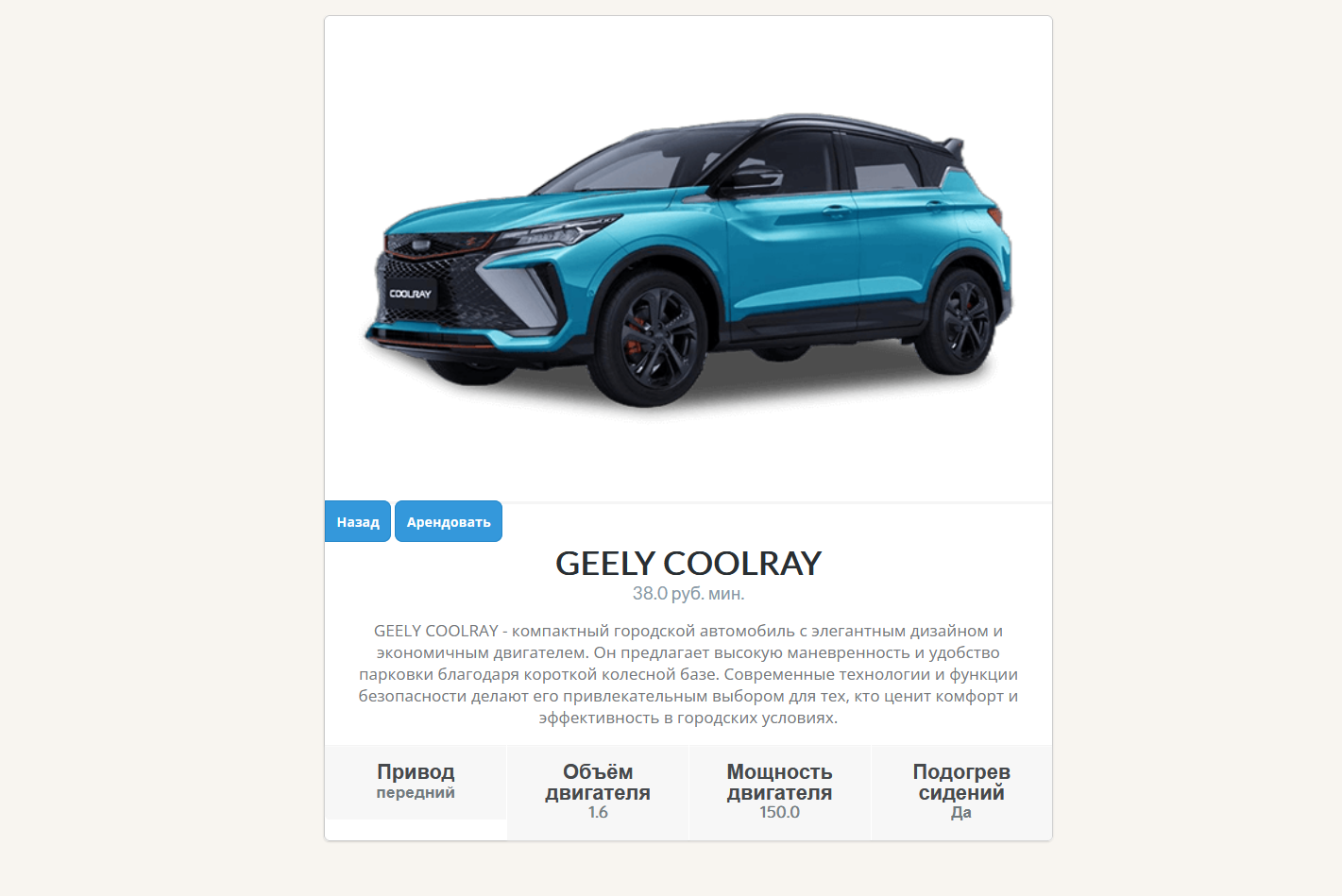


Рисунок 19 – Страница профиля автомобиля

3.3.5 Страница о нас

На данной странице представлена информация о данной каршерин-компании, контактные данные и краткое описание (рисунок 20).

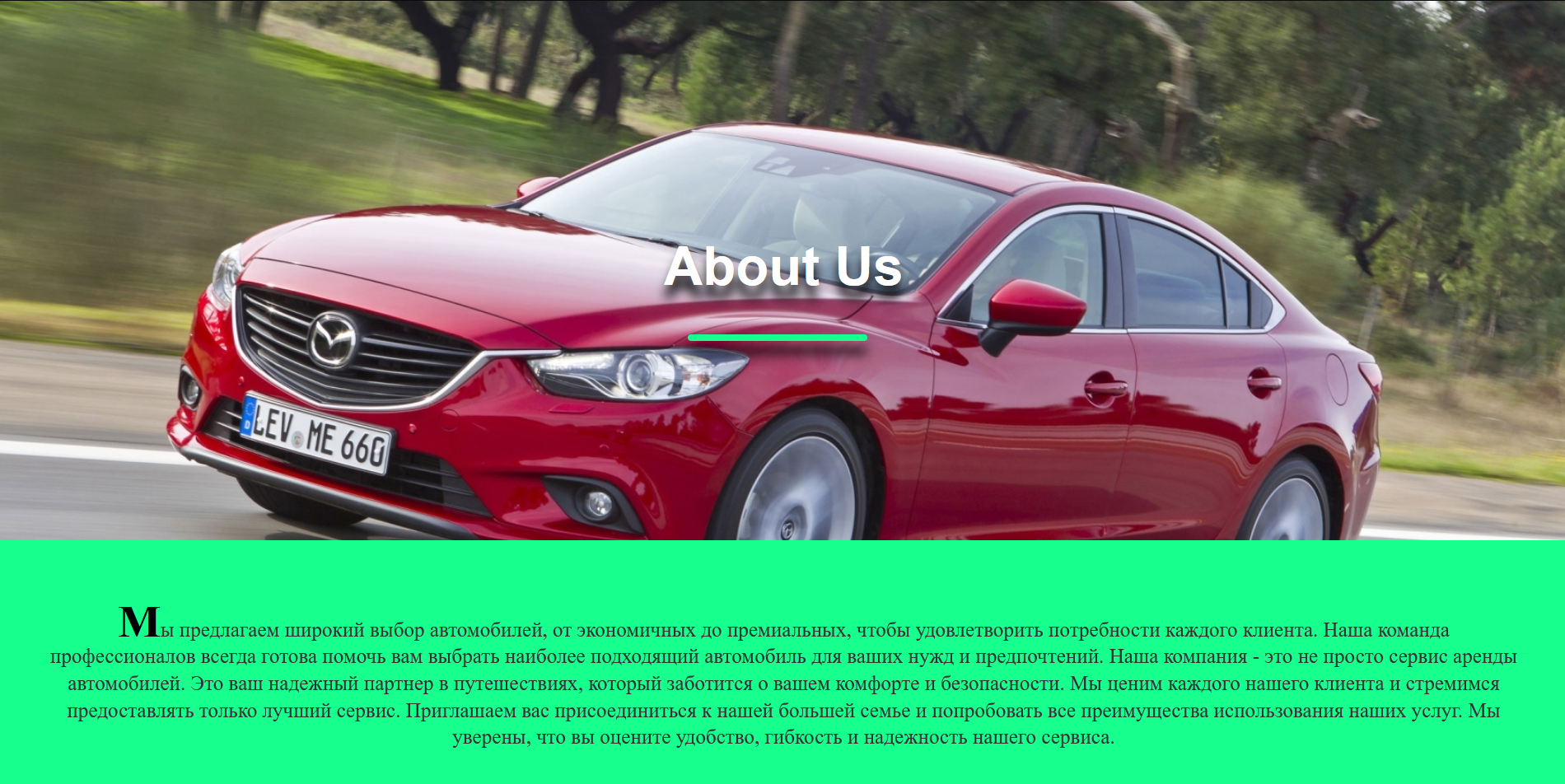


Рисунок 20 – Страница о нас

3.3.6 Страница входа и регистрации

На данной странице пользователь может зарегистрироваться или войти в свой аккаунт (рисунок 21 - 22).

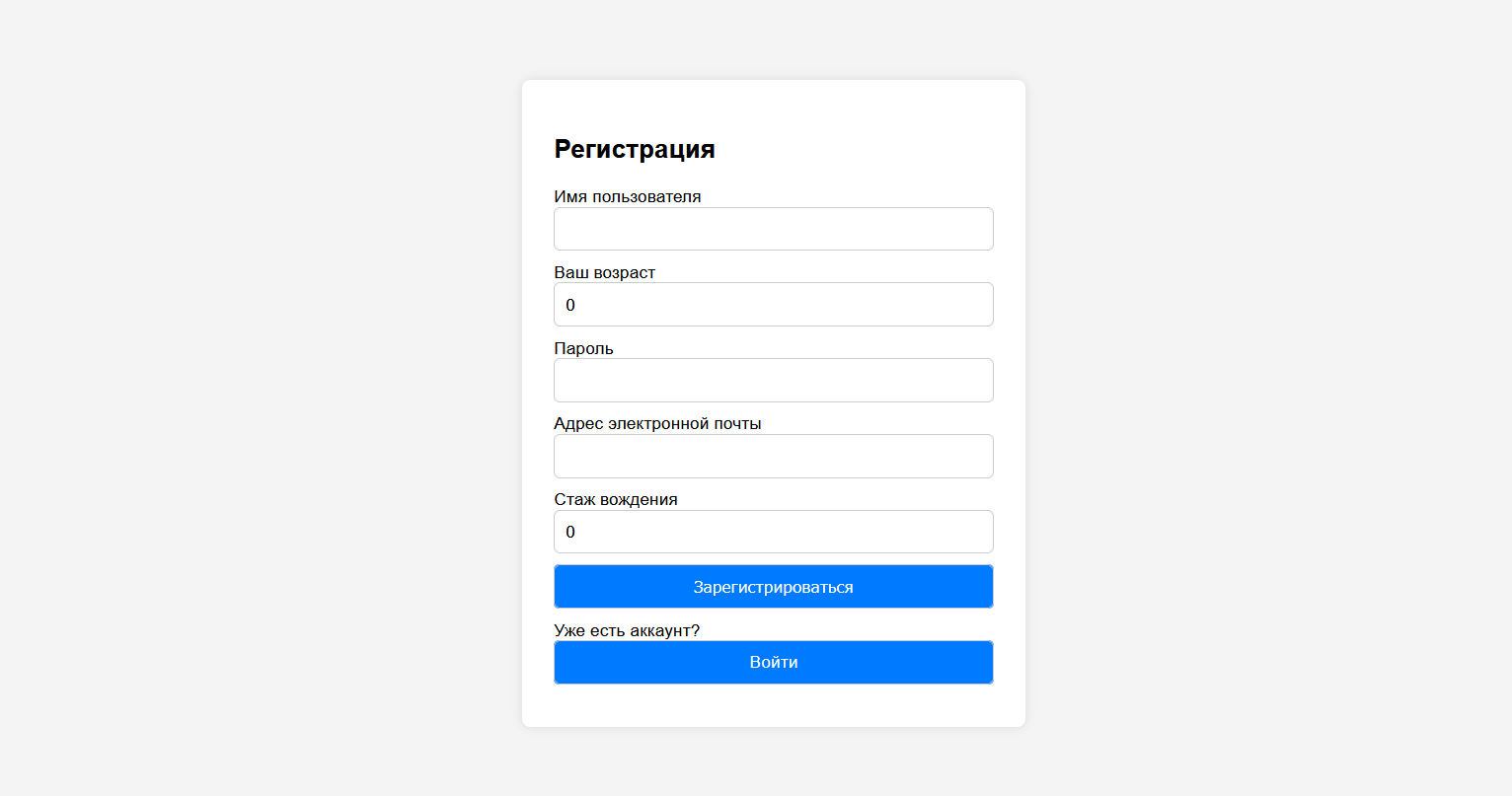


Рисунок 21 – Страница регистрации



Рисунок 22 – Страница входа

4 КОНТЕЙНЕРИЗАЦИЯ

4.1 Использование Docker

Docker позволяет создавать изолированные контейнеры для приложений и упаковывать окружение, необходимое для успешного запуска кода. В проекте был создан Dockerfile для серверной части приложения (рисунки 25-26).

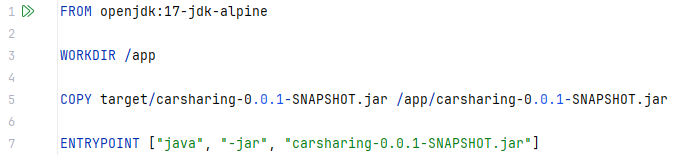


Рисунок 23 – Файл Dockerfile

Поскольку в приложении необходимо взаимодействие нескольких образов, был использован docker-compose файл, написанный в файловом формате YAML. Он обеспечивает взаимодействие между контейнерами и их правильную и последовательную отработку (рисунок 24).

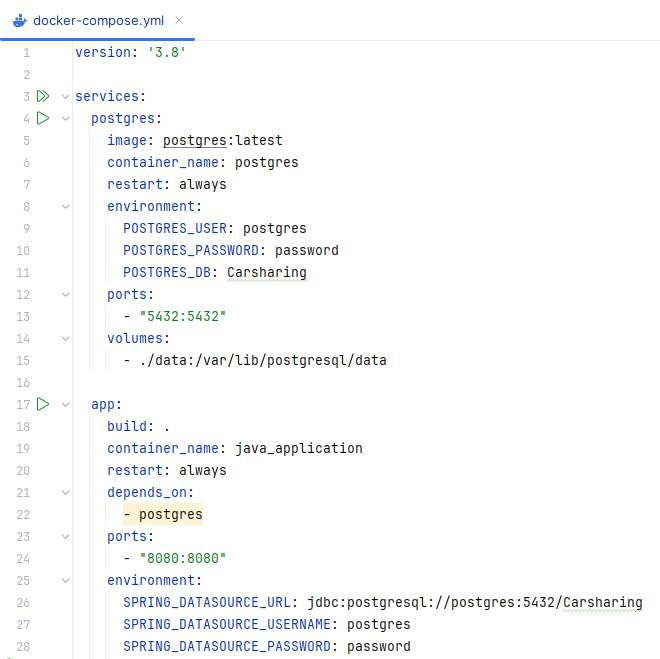


Рисунок 24 – Файл docker-compose.yml

Результат запуска спроектированного приложения представлен ниже (рисунок 25).

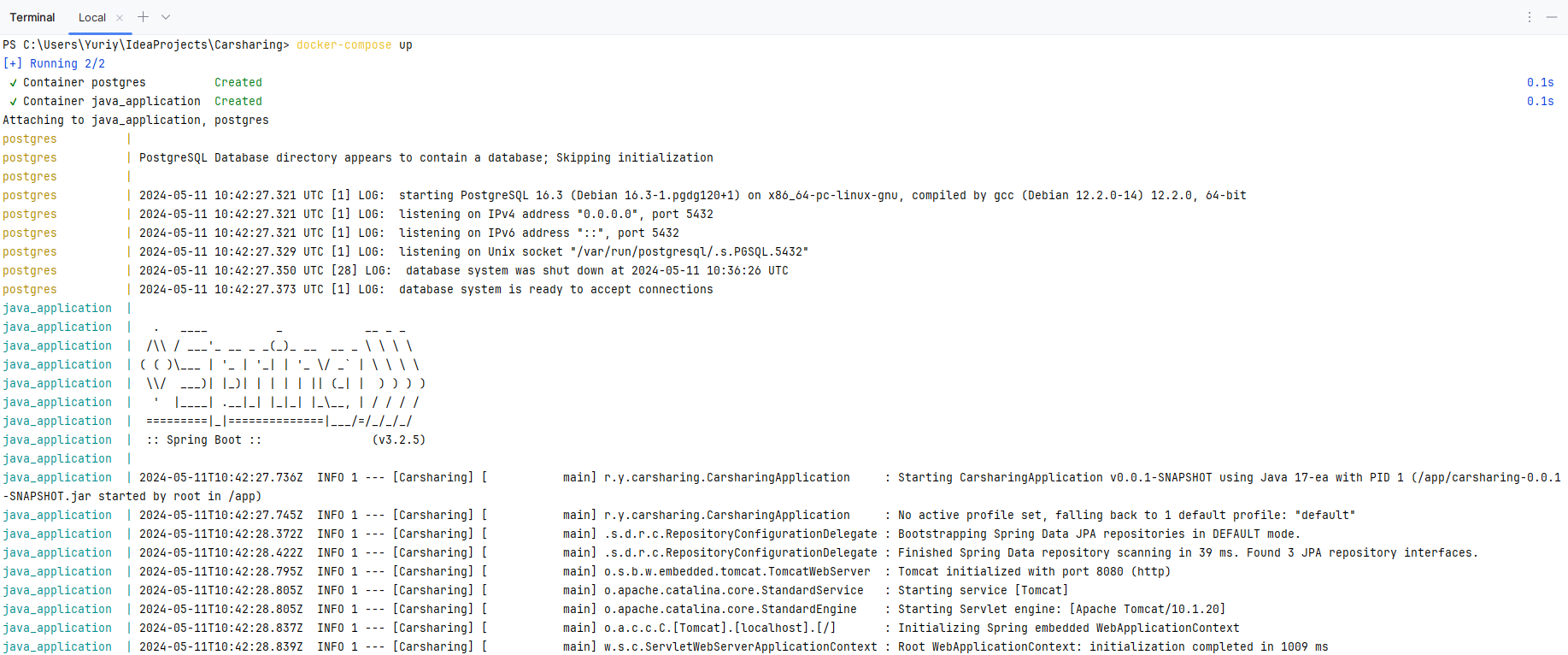
****

Рисунок 25 – Запуск приложения

ЗАКЛЮЧЕНИЕ

Во время выполнения данной курсовой работы было спроектировано приложение на тему «Каршеринговая компания».

Благодаря тщательному анализу предметной области были выявлены необходимые для удобства пользователей функции и особенности приложения. После проведения анализа и полученных результатов был выбран набор технологий и инструментов для разработки приложения, включающий в себя язык программирования Java, Spring Framework и PostgreSQL. Данные инструменты являются удобными, многофункциональными и масштабируемыми. Использованные технологии позволили создать веб-приложение, полностью отвечающее требованиям пользователей.

Во время разработки были применены полученные ранее знания о предпочтениях клиентов и реализован необходимый функционал. Приложение было протестировано несколькими клиентами, с помощью чего удалось подтвердить безошибочную работу программы. Результатом данной курсовой работы является проект, разработанный на языке Java с использованием Spring Framework, который полностью удовлетворяет потребности пользователей. Исходный код проекта и инструкция по запуску доступны по данной ссылке: <https://github.com/Yuranium/Carsharing>

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Делимобиль – каршеринг для твоих достижений. [Электронный ресурс] – URL: <https://delimobil.ru> (дата обращения 05.05.2024).
2. BelkaCar [Электронный ресурс] – URL: <https://belkacar.ru> (дата обращения 05.05.2024).
3. Яндекс.Драйв — каршеринг в Москве. [Электронный ресурс] – URL: <https://realty.ya.ru> (дата обращения 06.05.2024).
4. IntelliJ IDEA [Электронный ресурс] – URL: <https://www.jetbrains.com/idea/> (дата обращения 10.04.2024).
5. Docker [Электронный ресурс] – URL: <https://www.docker.com/> (дата обращения 10.04.2024).
6. Git [Электронный ресурс] – URL: <https://git-scm.com/> (дата обращения 10.04.2024).
7. Брюс Эккель. Философия Java. 4-е полное изд. — СПб.: Питер, 2015. — 1168 c.
8. Герберд Шилдт. Java. Полное руководство, 10-е изд. — СПб.: ООО "Альфакнига', 2018. — 1488 с.
9. Кэти Сьерра, Берт Бейтс. Изучаем Java. Основы. 2-е изд. — М.: ООО "Head First", 2012. — 720 с.
10. Maven – Википедия [Электронный ресурс] – URL: <https://ru.wikipedia.org/wiki/Apache_Maven> (дата обращения 10.04.2024).
11. Spring Framework [Электронный ресурс] – URL: <https://www.baeldung.com/> (дата обращения 10.04.2024)
12. Spring MVC - Spring [Электронный ресурс] – URL: <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html> (дата обращения 10.04.2024).
13. Spring Security - Wikipedia [Электронный ресурс] – URL: <https://en.wikipedia.org/wiki/Spring_Security> (дата обращения 10.04.2024).
14. Spring Data JPA [Электронный ресурс] – URL: <https://spring.io/projects/spring-data-jpa> (дата обращения 20.04.2024).
15. Project Lombok – Project Lombok [Электронный ресурс] – URL: [https://projectlombok.org/](https://projectlombok.org/%20) (дата обращения 01.05.2024).
16. PostgreSQL [Электронный ресурс] – URL: <https://postgresql-tutorial.com/> (дата обращения 20.04.2024).
17. WebReference [Электронный ресурс] – URL: [https://webref.ru/layout/learn-html-css](https://webref.ru/layout/learn-html-css%20) (дата обращения 11.04.2024).