

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

**Московский авиационный институт**  
**(национальный исследовательский университет)**

Институт № 8  
Информационных технологий и прикладной математики

**Кафедра 813 «Компьютерная математика»**

**КУРСОВАЯ РАБОТА**  
по дисциплине «Базы данных»

на тему: Разработка клиент-серверного БД-ориентированного ПО  
«Автосалон»

**Выполнил:** студент группы М8О-311Б-19

---

Заиц Георгий Тиодорович

---

(Фамилия, имя, отчество)

---

(подпись)

---

**Принял:** доцент кафедры 813

---

Чумакова Екатерина Витальевна

---

(Фамилия, имя, отчество)

---

(подпись)

---

**Оценка:**

**Дата:**

Москва, 2021

# Содержание

<b>I</b>	<b>Теоретическая часть</b>	<b>4</b>
<b>1</b>	<b>Клиент-серверное ПО «Автосалон»</b>	<b>4</b>
1.1	Техническое задание . . . . .	4
1.2	Структура ПО . . . . .	6
<b>2</b>	<b>Анализ предметной области</b>	<b>7</b>
2.1	Информационная модель проектируемой системы . . . . .	9
<b>3</b>	<b>Функциональные задачи клиента</b>	<b>9</b>
<b>II</b>	<b>Практическая часть</b>	<b>9</b>
<b>4</b>	<b>Реляционная модель данных системы</b>	<b>10</b>
4.1	Общая схема таблиц в базе данных . . . . .	10
4.2	Таблицы . . . . .	10
4.2.1	Таблица gear_box . . . . .	10
4.2.2	Таблица capacity . . . . .	10
4.2.3	Таблица layout . . . . .	11
4.2.4	Таблица body_type . . . . .	11
4.2.5	Таблица brand . . . . .	11
4.2.6	Таблица country . . . . .	11
4.2.7	Таблица engine . . . . .	11
4.2.8	Таблица picture . . . . .	11
4.2.9	Таблица car . . . . .	11
4.2.10	Таблица roles . . . . .	11
4.2.11	Таблица customer . . . . .	11
<b>5</b>	<b>Программная реализация системы</b>	<b>12</b>
5.1	Протокол общения пользовательского клиента и сервера <b>S</b> . . . . .	12
5.2	Протокол общения клиента-админа системы и сервера <b>S</b> . . . . .	13
5.3	Протокол общения сервера <b>S</b> и базы данных <b>DB</b> . . . . .	13

5.4	Структура приложения . . . . .	15
5.5	Основные алгоритмы . . . . .	16
5.6	Руководство пользователя . . . . .	18
<b>6</b>	<b>Список использованных источников</b>	<b>23</b>
<b>7</b>	<b>Приложение</b>	<b>24</b>
7.1	Скрипты создания таблиц . . . . .	24
7.1.1	Создание таблицы body_type . . . . .	24
7.1.2	Создание таблицы layout . . . . .	24
7.1.3	Создание таблицы capacity . . . . .	24
7.1.4	Создание таблицы engine . . . . .	24
7.1.5	Создание таблицы gear_box . . . . .	25
7.1.6	Создание таблицы country . . . . .	25
7.1.7	Создание таблицы brand . . . . .	25
7.1.8	Создание таблицы picture . . . . .	25
7.1.9	Создание таблицы customer . . . . .	26
7.1.10	Создание таблицы car . . . . .	26
7.2	ПО «Автосалон» . . . . .	26
7.2.1	Код сервера . . . . .	26
7.2.2	Код пользовательского клиента . . . . .	51

# Часть I

## Теоретическая часть

### 1 Клиент-серверное ПО «Автосалон»

#### 1.1 Техническое задание

1. Необходимо выбрать предметную область для создания базы данных. Выбранная предметная область должна быть уникальной для всего потока, а не только в рамках учебной группы.
2. Необходимо описать таблицы и их назначение. Выполнить проектирование логической структуры базы данных. Описать схему базы данных.
3. Ограничения на техническую реализацию:
  - (a) Должна быть использована СУБД PostgreSQL версии 13 и выше.
  - (b) Все реальные таблицы должны иметь 3 нормальную форму или выше.
  - (c) База данных должна иметь минимум 5 содержательных таблиц. В них не входят таблицы справочники (вида ключ – значение), наследуемые таблицы ([1] п. 5.10) считаются одной таблицей, так же как все секции секционированной таблицы ([1] п. 5.11)
  - (d) В каждой таблице должен быть первичный ключ, который рекомендуется делать типом Serial или родственными ему.
  - (e) Таблицы должны быть связаны внешними ключами.
  - (f) Должен присутствовать обоснованный пример использования наследования и секционирования таблиц.
  - (g) При разумных ограничениях на значения полей необходимо в структуры таблиц прописывать ограничения на значения атрибутов.
  - (h) Должны быть содержательные запросы с применением вложенных запросов, агрегатных функций, Group By, Union и Join.
  - (i) Необходимо создать и применить свою агрегатную функцию ([1] п. 38.12).

- (j) База данных должна иметь представления ([1] п. 41.2), материализованные представления ([1] п. 41.3), триггеры ([1] п. 39) и хранимые процедуры ([1] п. 43), причём все эти объекты должны быть осмысленны, а их использование оправдано.
  - (k) Должно быть реализовано каскадное удаление или обновление данных при срабатывании триггера. Использование должно быть обосновано.
  - (l) Хотя бы в одной хранимой процедуре должен быть использован язык программирования PL/pgSQL с использованием условий и циклов.
  - (m) Клиент БД должен выполнять запросы в БД через курсоры ([1] п. 43.7).
  - (n) Таблицы должны использовать индексы ([1] п. 11).
  - (o) Доступ к таблицам должен быть разграничен на уровне прав ([1] п. 5.7) и политик доступа к строкам ([1] п. 5.8).
  - (p) Вся структура БД должна храниться в SQL файлах.
  - (q) Данные для тестовой работы с БД также должны быть представлены в SQL.
4. Необходимо разработать два клиентских приложения для доступа к базе данных. Данные приложения должны быть написаны на двух разных языках программирования и иметь разный интерфейс (например, классическое оконное приложение и web-приложение). Выбор языков программирования произволен: C/C++, python, perl, ruby, JavaScript, php, swift, C# 5.0, Java и др.
5. Необходимо организовать различные роли пользователей и права доступа к данным. Далее, необходимо реализовать возможность создания архивных копий и восстановления данных из клиентского приложения.
6. При разработке базы данных следует организовать логику обработки данных не на стороне клиента, а, например, на стороне сервера, базы данных, клиентские приложения служат только для представления данных и тривиальной обработки данных.
7. При показе вашего проекта необходимо уметь демонстрировать таблицы, представления, триггеры и хранимые процедуры базы данных, внешние ключи, ограничения целостности и др. В клиентских приложениях уметь демонстрировать

подключение к базе данных, основные режимы работы с данными (просмотр, редактирование, обновление...)

8. Необходимо реализовать корректную обработку различного рода ошибок, которые могут возникать при работе с базой данных.

## 1.2 Структура ПО

Клиент-серверное приложение должно иметь следующую структуру:

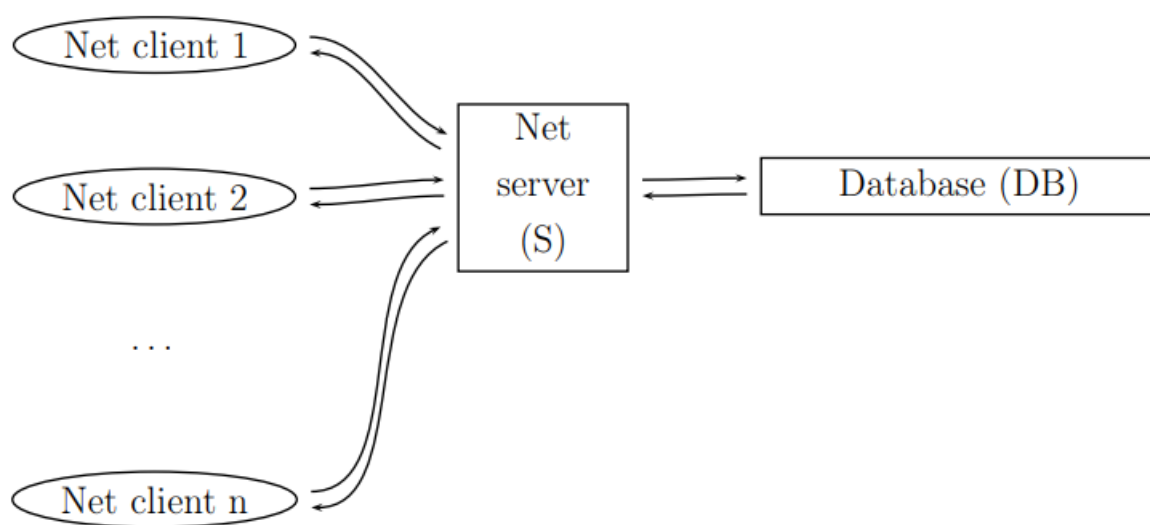


Рис. 1: Структура программного комплекса.

Сетевой сервер (S) должен принимать запросы от различных клиентов и использует базу данных (DB) для ответа на эти запросы.

Клиенты не должны иметь представление о внутренней структуре DB, структуре таблиц, хранимых процедурах и всех взаимосвязях между данными.

Технология клиент-сервер означает такой способ взаимодействия программных компонентов, при котором они образуют единую систему. Как видно из самого названия, существует некий клиентский процесс, требующий определенных ресурсов, а также серверный процесс, который эти ресурсы предоставляет.

Совсем необязательно, чтобы они находились на одном компьютере. Обычно принято размещать сервер на одном узле локальной сети, а клиентов - на других узлах.

## 2 Анализ предметной области

Базы данных являются одним из основных компонентов современных информационных систем. Информационная система - это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации.

Цель любой информационной системы - обработка информации конкретной предметной области.

Под предметной областью понимается совокупность связанных между собой функций, задач управления в некоторой области деятельности предприятия, с помощью которых достигается выполнение поставленной цели.

База данных - это приложение, обеспечивающее создание, хранение, обновление и поиск информации в базе данных, а также управление безопасностью и целостностью данных. В простейшем случае - это программа, которая обеспечивает работу с информацией. Информация в базе может быть любая, при этом, как правило, работа с информацией подразумевает следующие действия:

1. добавление новой информации;
2. поиск информации;
3. изменение информации;
4. удаление информации из базы данных;

Особенностями такой совокупности данных являются:

1. достаточно большие объемы информации;
2. максимально возможная компактность хранения данных;
3. возможность извлечения из базы данных разнообразной информации в определенной предметной области;
4. удобные для пользователя вид и форма извлекаемой информации;
5. высокая скорость доступа к данным;
6. надежность хранения информации и возможность Предоставления санкционированного доступа к данным для отдельных пользователей;

7. удобство и простота конструирования пользователем запросов, форм и отчетов для выборки данных. Создание базы данных, ее поддержка и обеспечение Доступа пользователей к ней осуществляется с помощью специального программного инструмента - системы управления базами данных.

Базы данных могут быть локальные и удаленные. Отличия в типах систем состоит в расположении программы, использующей данные, самих данных, а также способа разделения данных между несколькими пользователями.

**Локальная база данных.** Данные локальной базы данных (файлы данных) находятся на одном (локальном) устройстве, в качестве которого может выступать диск компьютера или диск другого компьютера, работающего в сети.

Для обеспечения разграниченного доступа к данным между несколькими пользователями в локальных базах данных применяется метод, получивший название блокировка файлов. Метод блокировки заключается в том, что пока данные используются одним пользователем, другой пользователь не может работать с этими данными.

**Удаленная база данных.** Данные (файлы) удаленной базы данных находятся на удаленном компьютере. При этом не следует путать сетевые диски компьютеров, работающих в сети, и каталоги удаленного компьютера.

Программа работы с удаленной базой данных состоит из двух частей: клиентской и серверной (такая технология получила название «клиент—сервер»). Клиентская часть программы, работающая на компьютере пользователя, обеспечивает взаимодействие с серверной программой с помощью запросов, передаваемых на удалённый компьютер.

Серверная часть программы, работающая на удалённом компьютере, принимает запросы, выполняет их и пересылает данные клиентской программе. Запросы представляют собой команды на специальном языке структурированных запросов (Structured Query Language, SQL).

Была смоделирована обобщенная модель автосалона и выделены возможные функции действий клиентов, а также администраторов. Предполагаемый функционал магазина: подбор клиентом машин, личный кабинет пользователя и возможность редактирования данных. Возможен поиск машин по нескольким параметрам. Создание отдельной страницы для администратора с возможностью добавления новых машин и их редактирования, а также возможность редактирования ролей пользователей.



## 2.1 Информационная модель проектируемой системы

В моделируемой системе будет несколько основных сущностей, такие как автомобиль, клиент.

Сущность автомобиля содержит атрибуты с основной информацией об автомобиле (марка, модель, тип двигателя, цена и др.).

Сущность клиента содержит атрибуты: имя пользователя, пароль и email.

## 3 Функциональные задачи клиента

Для обычного пользователя:

- Поиск автомобиля по заданным фильтрам
- Редактирование данных (имени пользователя, пароля и email) в личном кабинете

Для пользователя с правами администратора:

- Все возможности обычного пользователя
- Добавление новых автомобилей
- Изменение характеристик автомобилей, их удаление
- Редактирование ролей всех зарегистрированных пользователей
- Удаление пользователей

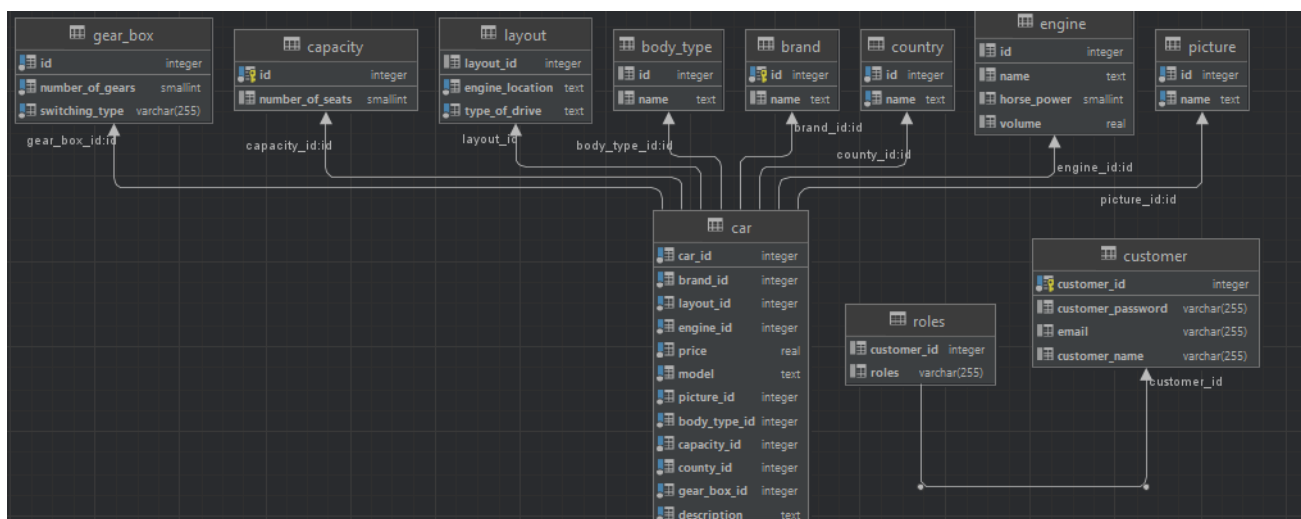


Рис. 2: Общая схема БД со связями между таблицами.

## Часть II

# Практическая часть

## 4 Реляционная модель данных системы

### 4.1 Общая схема таблиц в базе данных

### 4.2 Таблицы

DDL код создания таблиц можно найти в разделе [7.1](#).

#### 4.2.1 Таблица gear\_box

Таблица для хранения типов коробок передач. Содержит уникальный идентификатор - первичный ключ, количество передач, тип переключения.

Все последующие таблицы хранят уникальный идентификатор - первичный ключ.

#### 4.2.2 Таблица capacity

Содержит количество посадочных мест автомобиля.

#### **4.2.3 Таблица layout**

Содержит расположения двигателя, тип привода.

#### **4.2.4 Таблица body\_type**

Содержит название типа кузова.

#### **4.2.5 Таблица brand**

Содержит название марки-производителя.

#### **4.2.6 Таблица country**

Содержит название страны-производителя.

#### **4.2.7 Таблица engine**

Содержит название двигателя, количество лошадиных сил, объем.

#### **4.2.8 Таблица picture**

Содержит название изображения.

#### **4.2.9 Таблица car**

Таблица хранит полную информацию об автомобиле и внешними ключами связана с выше перечисленными таблицами.

Также в ней содержится информация об автомобиле (описание), цене, модели.

#### **4.2.10 Таблица roles**

Таблица ролей связана с таблицей пользователей внешним ключом. Содержит в себе уникальный идентификатор пользователя и его роль. Возможные роли: USER, ADMIN.

#### **4.2.11 Таблица customer**

Содержит информацию о пользователе - его имя, email и пароль.

## 5 Программная реализация системы

### 5.1 Протокол общения пользовательского клиента и сервера S

Общение клиента с сервером происходит посредством паттерна MVC.

Model-View-Controller (MVC, «Модель-Представление-Контроллер») — схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Среда Spring Web Model-View-Controller (MVC) разработана на основе DispatcherServlet, который обрабатывает все HTTP-запросы и ответы. Последовательность событий, соответствующая входящему HTTP-запросу:

- После получения HTTP-запроса DispatcherServlet обращается к интерфейсу HandlerMapping, который определяет, какой Контроллер должен быть вызван, после чего, отправляет запрос в нужный Контроллер.
- Контроллер принимает запрос и вызывает соответствующий служебный метод, основанный на GET или POST. Вызванный метод определяет данные Модели, основанные на определённой логике и возвращает в DispatcherServlet имя View.
- При помощи интерфейса ViewResolver DispatcherServlet определяет, какой Вид нужно использовать на основании полученного имени.
- После того как View создан, DispatcherServlet отправляет данные Модели в виде атрибутов во View, который в конечном итоге отображается в браузере.

Все остальные действия, которые может выполнять пользователь (например редактирование профиля) происходят по подобному сценарию, описанному выше.

## 5.2 Протокол общения клиента-админа системы и сервера S

Общение клиента-админа с сервером практически не отличается от общения обычного пользователя. Но у админа есть собственная страница, к которой закрыт доступ обычным пользователям. Разграничение доступа происходит с помощью аннотации `@PreAuthorize("hasAuthority('ADMIN')")`. Таким образом указывается, что только пользователь с ролью администратора может получить доступ к данной странице. Если на такую страницу попытается зайти обычный пользователь вернется ошибка прав доступа 403. Администратор может редактировать роли пользователей, то есть назначать новых клиентов-админов.

```
@PostMapping("/{customer}/delete")
@PreAuthorize("hasAuthority('ADMIN')")
public String delete(@PathVariable Customer customer) {
    userService.deleteUser(customer);
    return "redirect:/user";
}
```

Рис. 3: Метод GET редактирования ролей пользователей.

В методе в модель добавляются пользователи и их роли, далее они отображаются на странице редактирования.

## 5.3 Протокол общения сервера S и базы данных DB

Каждая таблица базы данных представлена в виде класса. Аннотация `@Entity` позволяет представлять таблиц в виде Java-классов. Поля повторяют поля таблиц на стороне БД. Пример представления таблицы:

```

@Entity
@Table(name = "car", schema = "public")
@Data
public class Car {
    public Car(){}

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Car car)) return false;
        return Objects.equals(getCar_id(), car.getCar_id()) && Ob
    }

    @Override
    public int hashCode() {
        return Objects.hash(getCar_id(), getBrand(), getModel(),

    public Car(Brand brand, String model, BodyType bodyType,
                Layout layout, Capacity capacity, Engine engine,
                GearBox gearBox, Country country, Double price) {
        this.brand = brand;
        this.model = model;
        this.bodyType = bodyType;
        this.layout = layout;
        this.capacity = capacity;
        this.engine = engine;

```

Рис. 4: Таблица в виде Java-класса.

Построение запросов из сервера к базе реализовано через специальные интерфейсы-репозитории. Для каждой сущности создаются отдельные репозитории с методами без тела, а создает эти методы во время компиляции Spring Data.

Такие интерфейсы помечаются специальной аннотацией `@Repository`. Эта аннотация используется в классах Java, которые напрямую обращаются к базе данных. `@Repository` работает как маркер для любого класса, который выполняет роль хранилища или доступ к данным объекта. Эта аннотация имеет функцию автоматического перевода. Например, когда возникает исключение в `@Repository`, существует обработчик для этого исключения, и нет необходимости добавлять блок `try-catch`.

Все интерфейсы - репозитории наследуют интерфейс `JpaRepository`. `JpaRepository<сущность, тип_её_ID>` — включает все заготовленные запросы, чтобы обращаться к определённой сущности. Пример использования запросов, предоставленных `JpaRepository`:

```
List<Car> findByModel(String model);
```

Рис. 5: Запрос из JpaRepository.

На вход запроса приходит строка, по которой запрос ищет в базе данных соответствующий промокод, если такой не найден возвращается null.

Также при помощи аннотации @Query можно писать собственные запросы на языке SQL. Подобный пример приведен ниже:

```
public interface CarRepo extends CrudRepository<Car, Integer> {  
    @Query("SELECT c FROM Car c JOIN Brand b on c.brand.id = b.id WHERE b.name=?1")  
    List<Car> findByBrand(String brand_id);  
  
    @Query("SELECT c FROM Car c WHERE " +  
        "(:brandId IS NULL OR c.brand.id = :brandId) AND " +  
        "(:bodyTypeId IS NULL OR c.bodyType.id = :bodyTypeId) AND " +  
        "(:capacityId IS NULL OR c.capacity.id = :capacityId) AND " +  
        "(:gearBoxId IS NULL OR c.gearBox.id = :gearBoxId)")  
    List<Car> findCarsByBrandIdAndBodyTypeIdAndCapacityIdAndGearBoxId(  
        @Param("brandId") Integer brandId,  
        @Param("bodyTypeId") Integer bodyTypeId,  
        @Param("capacityId") Integer capacityId,  
        @Param("gearBoxId") Integer gearBoxId  
    );  
}
```

Рис. 6: Запрос при помощи аннотации @Query.

Написанный вручную запрос выполняет поиск автомобиля по фильтрам.

## 5.4 Структура приложения

После создания базы данных происходит подключение к ней из сервера через localhost:5432. После запуска приложения происходит запуск сервера, и он начинает "слушать" порт 8080. Далее, введя в поисковую строку браузера localhost:8080, можно перейти на сайт книжного магазина. Если же сервер не был запущен или произошла ошибка, обращение к данному порту не даст никакой информации.

Всего используется 4 основных контроллеров.

1. CarController - обработка всех действий связанных с автомобилями.
2. RegistrationController - обработка всех действий связанных с регистрацией.
3. MainController - контроллер URL-адресов, по которым админ может добавлять автомобили и редактировать их.
4. UserController - изменение администратором ролей пользователей, а также редактирование пользователями личных данных

## 5.5 Основные алгоритмы

### 1. Регистрация и авторизация

Происходит с использованием Spring Security. При регистрации нового пользователя, производится проверка на существование такого пользователя в базе данных по email. Если метод addUser возвращает true пользователь добавляется в базу данных, ему назначается роль USER и его пароль шифруется при помощи BCryptPasswordEncoder - криптографическая хеш-функция формирования ключа.

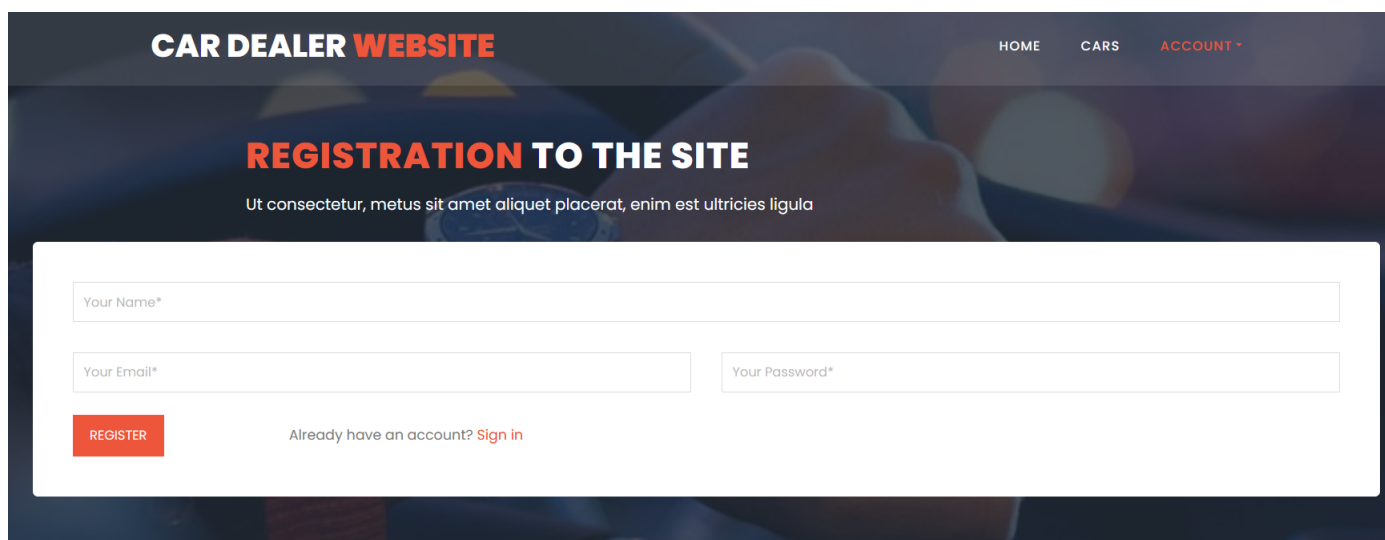


Рис. 7: Страница регистрации нового пользователя.

При авторизации посредством метода loadUserByUsername в базе данных ищется пользователь. Если найден - переход на главную страницу, иначе сообщение, что данный пользователь не найден.



The screenshot shows the login page of a 'CAR DEALER WEBSITE'. The header includes navigation links: HOME, CARS, and ACCOUNT. The main heading is 'LOGIN TO THE SITE' with a placeholder text 'Ut consectetur, metus sit amet aliquet placerat, enim est ultricies ligula'. Below this are two input fields: 'Your Name\*' and 'Your Password\*'. A red 'SIGN IN' button is on the left, and a link 'Don't have an account? Register now' is on the right.

Рис. 8: Страница авторизации.

## 2. Добавление нового автомобиля

The screenshot shows the 'ADD CAR PANEL' page. The header includes navigation links: HOME, CARS, ADD CAR, USERS LIST, ADMIN, and a red 'Sign Out' button. The main heading is 'ADD CAR PANEL' with the same placeholder text as the login page. The form contains several input fields: Brand, Model, Body Type, Layout, Capacity, Engine, Horse Power, Engine displacement, GearBox, Number of gears, Country, and Price. There is also a file upload section with a 'Browse...' button and the text 'No file selected.'. A large text area for 'Description' is at the bottom, followed by a red 'ADD' button.

Рис. 9: Страница добавления автомобиля

Входными параметрами в метод `addCar` являются все поля автомобиля (марка, модель, цена и т.д.). Каждый параметр проверяется на корректность (например не отрицательность цены и количества посадочных мест). Поля тип кузова, страна-производитель, изображение, двигатель, коробка передач, количество посадочных мест, тип привода, марки сначала ищутся в своих таблицах при помощи репозитория, если не найдены то они добавляются туда и возвращается

объект каждой сущности. Далее автомобиль создается, добавляется в базу данных и выводится на странице. В случае некорректности или незаполнения какого либо поля выводится сообщение об ошибке.

### 3. Изменение данных в личном кабинете

На вход в метод `updateProfile` подается новый email и пароль (либо пустая строка если ничего не было введено). Происходит проверка на равенство старого email новому, и в случае их различия, новая почта пользователя сохраняется. Новый пароль сразу заносится зашифрованным. И обновленные данные о пользователе сохраняются в базу данных.

Реализация алгоритмов представлена в разделе приложения [7.2](#).

## 5.6 Руководство пользователя

1. Регистрация и авторизация Для того, чтобы авторизоваться, необходимо иметь аккаунт, который можно зарегистрировать на соответствующей странице.



**REGISTRATION TO THE SITE**

Ut consectetur, metus sit amet aliquet placerat, enim est ultricies ligula

admin

awesomeAdmin@admin.ru

.....

REGISTER

Already have an account? [Sign in](#)

Рис. 10: Страница регистрации с заполненными данными аккаунта.

После успешной регистрации необходимо войти в аккаунт.

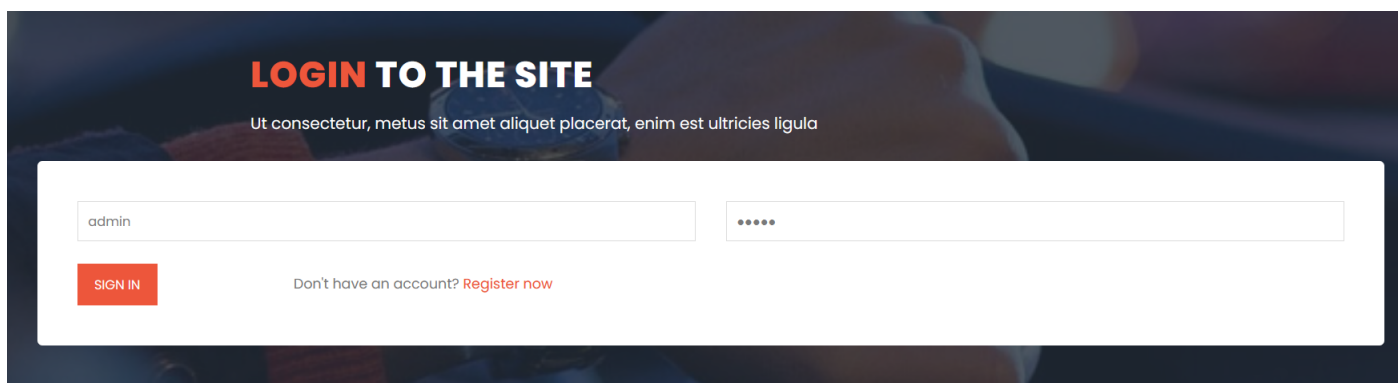


Рис. 11: Страница авторизации с заполненными данными аккаунта.

После успешной авторизации у пользователей с правами администратора, появятся дополнительные пункты меню.

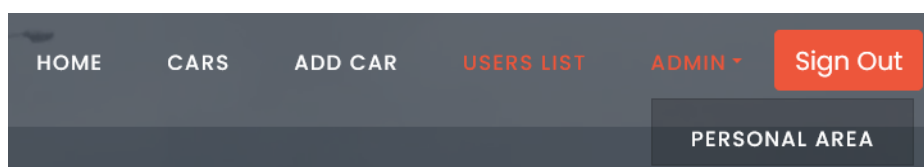


Рис. 12: Страница авторизации с заполненными данными аккаунта.

Закрытые для обычных пользователей являются пункты "ADD CAR" и "USER LIST" а странице "ADD CAR" происходит управление доступными автомобилями, находящимся в базе данных (управление доступно только пользователям с правами "ADMIN" ).

Body Type:

Make:

Gearbox:

Number of seats:

All

All

All

All

SEARCH

**\$64,700**

**Nissan Skyline GT-R X (R34)**

RB26DETT Manual (6 speed) Japan Coupe

+ View Car + Edit Car [Delete Car](#)

**\$99,000**

**Volkswagen Touareg III**

CASA 3.0 TDI Automatic (6 speed) Germany SUV

+ View Car + Edit Car [Delete Car](#)

**\$16,175**

**Subaru Impreza WRX STI**

EJ257 Automatic (6 speed) Japan Hatchback

+ View Car + Edit Car [Delete Car](#)

Рис. 13: Страница отображения автомобилей.

На это странице также присутствует фильтрация автомобилей, есть возможность выбрать любую комбинацию из 4 фильтров и получить необходимый результат.

Также приватной страницей является страница со списком пользователей и их редактированием. При нажатии на гиперссылку "edit" на странице будет предоставлен выбор ролей для выбранного участника. Также с этой страницы, есть возможность удалить пользователя.

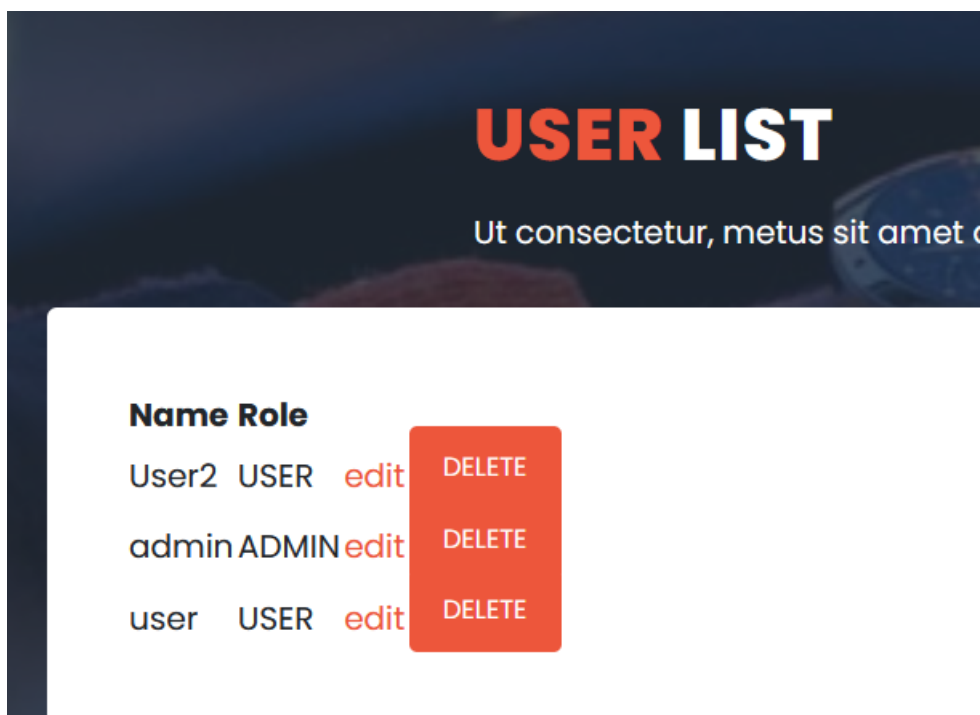



Рис. 14: Страница списка пользователей.


The screenshot shows a form for editing a user. It includes a text input field containing "User2", a checkbox labeled "USER" which is checked, and an unchecked checkbox labeled "ADMIN". Below these is a "Save" button.

Рис. 15: Редактирование пользователя.

Также для всех пользователей доступен личный кабинет, где есть возможность сменить email, имя пользователя и пароль.

---

**HELLO ADMIN**  
  
awesomeAdmin@admin.ru

**PERSONAL AREA**  
  

Nunc urna sem, laoreet ut metus id, aliquet consequat magna. Sed viverra ipsum dolor, ultricies fermentum massa consequat eu.

**Change Email**

**Change Username**

**Change Password**


Email:


awesomeAdmin@admin.ru

Change

Рис. 16: Редактирование email.

---

**HELLO ADMIN**  
  
awesomeAdmin@admin.ru

**PERSONAL AREA**  
  

Nunc urna sem, laoreet ut metus id, aliquet consequat magna. Sed viverra ipsum dolor, ultricies fermentum massa consequat eu.

**Change Email**

**Change Username**

**Change Password**

User Name :

admin

Change

Рис. 17: Редактирование имени пользователя.

HELLO ADMIN PERSONAL

awesomeAdmin@admin.ru

Change Email

Change Username

Change Password

Password:

Confirm password:

Change

Рис. 18: Редактирование пароля.

## 6 Список использованных источников

1. Документация PostgresPro 14. <http://postgrespro.ru/docs/14/>.
2. П. В. Лузанов, Е. В. Рогов, И. В. Лёвшин. Postgres. Первое знакомство. <http://postgrespro.ru/education/books/introbook>.
3. Дж. Р. Грофф, Пол Н. Вайнберг, Эндрю Оппель. SQL полное руководство. Третье издание. 2015
4. К. Дж. Дейт. Введение в системы баз данных. — 8-е изд. — М.: Вильямс, 2006. — С. 1328.
5. Д. С. Боуман, С. Л. Эмерсон, М. Дарновски. Практическое руководство по SQL. “Вильямс”, 2001, 4 изд.

## 7 Приложение

### 7.1 Скрипты создания таблиц

#### 7.1.1 Создание таблицы body\_type

Файл body\_type.sql

```
CREATE TABLE body_type(  
    id serial primary key,  
    name text unique not null  
);
```

---

#### 7.1.2 Создание таблицы layout

Файл layout.sql

```
CREATE TABLE layout(  
    layout_id serial primary key,  
    engine_location text unique ,  
    type_of_drive text unique not null  
);
```

---

#### 7.1.3 Создание таблицы capacity

Файл capacity.sql

```
CREATE TABLE capacity(  
    capacity_id serial primary key,  
    number_of_seats smallint CHECK ( number_of_seats > 0 )  
);
```

---

#### 7.1.4 Создание таблицы engine

Файл engine.sql

```
CREATE TABLE engine(  
    id serial primary key,  
    name text unique not null,  
    volume real,  
    horse_power smallint CHECK ( horse_power > 0 )  
);
```

---



### 7.1.5 Создание таблицы gear\_box

Файл gear\_box.sql

```
CREATE TABLE engine(  
    id serial primary key,  
    name text unique not null,  
    volume real,  
    horse_power smallint CHECK ( horse_power > 0 )  
);
```

---

### 7.1.6 Создание таблицы country

Файл country.sql

```
CREATE TABLE country(  
    id serial primary key,  
    name text unique not null  
);
```

---

### 7.1.7 Создание таблицы brand

Файл brand.sql

```
CREATE TABLE brand(  
    id serial primary key,  
    name text unique not null  
);
```

---

### 7.1.8 Создание таблицы picture

Файл picture.sql

```
CREATE TABLE picture(  
    picture_id serial primary key,  
    picture_name text unique not null  
);
```

---

## 7.1.9 Создание таблицы customer

### Файл customer.sql

```
CREATE TABLE customer(  
    customer_id serial primary key,  
    customer_name text not null,  
    phone_number int unique not null,  
    role_id int not null ,  
    foreign key (role_id) references role(role_id)  
);
```

---

## 7.1.10 Создание таблицы car

### Файл car.sql

```
CREATE TABLE car(  
    car_id serial primary key,  
    brand_id int not null,  
    model text not null,  
    body_type_id int not null,  
    layout_id int not null,  
    capacity_id int not null,  
    engine_id int not null,  
    gear_box_id int not null,  
    county_id int not null,  
    picture_id int not null,  
    price real,  
    foreign key (brand_id) references brand(id),  
    foreign key (body_type_id) references body_type(id),  
    foreign key (layout_id) references layout(layout_id),  
    foreign key (capacity_id) references capacity(capacity_id),  
    foreign key (engine_id) references engine(id),  
    foreign key (gear_box_id) references gear_box(id),  
    foreign key (picture_id) references gear_box(picture_id),  
    foreign key (county_id) references country(id)  
);
```

---

## 7.2 ПО «Автосалон»

### 7.2.1 Код сервера

#### Файл MvcConfig.java

```
package com.example.kursachdb.config;  
  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.ViewControllerRegistry;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;
```

```

@Configuration
public class MvcConfig implements WebMvcConfigurer {

    public void addViewControllers(ViewControllerRegistry registry) {
        registry.addViewController("/login").setViewName("login");
    }

}

```

---

## Файл WebSecurityConfig.java

```

package com.example.kursachdb.config;

import com.example.kursachdb.service.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
@RequiredArgsConstructor
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    private final UserService userService;

    private final PasswordEncoder passwordEncoder;

    @Bean
    public static PasswordEncoder getPasswordEncoder() {
        return new BCryptPasswordEncoder(8);
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
                .antMatchers("/", "/login", "/registration").permitAll()
                .antMatchers("/resources/**").permitAll()
                .antMatchers("/*.js").permitAll()
                .antMatchers("/api/**").authenticated()
            .and()
                .formLogin()
                .loginPage("/login").passwordParameter("customer_password").usernameParameter("customer_name")
                .permitAll()
            .and()
                .logout()
                .permitAll();
    }
}

```

```

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.userDetailsService(userService)
        .passwordEncoder(passwordEncoder);
}
}

```

---

## Файл CarController.java

```

package com.example.kursachdb.controllers;

import com.example.kursachdb.domain.Car;
import com.example.kursachdb.service.CarService;
import lombok.RequiredArgsConstructor;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;

@Controller
@RequiredArgsConstructor
@RequestMapping("/cars")
public class CarController {
    private final CarService carService;

    @GetMapping
    public String cars(
        @RequestParam(required=false,name="makeTypeSelect") Integer makeTypeSelect,
        @RequestParam(required=false,name="bodyTypeSelect") Integer bodyTypeSelect,
        @RequestParam(required=false,name="numSeatsTypeSelect") Integer numSeatsTypeSelect,
        @RequestParam(required=false,name="gearBoxTypeSelect") Integer gearBoxTypeSelect,
        Model model
    ) {
        carService.findCar(model, makeTypeSelect, bodyTypeSelect, numSeatsTypeSelect, gearBoxTypeSelect);

        return "cars";
    }

    @GetMapping("/{car}")
    public String aboutCar(@PathVariable Car car, Model model) {
        model.addAttribute("car", car);

        return "carDetails";
    }

    @GetMapping("/add")
    @PreAuthorize("hasAuthority('ADMIN')")
    public String add() {
        return "addCar";
    }

    @PostMapping("/add")
    @PreAuthorize("hasAuthority('ADMIN')")

```

```

public String add(
    @RequestParam String brand,
    @RequestParam String carModel,
    @RequestParam String bodyType,
    @RequestParam String layout,
    @RequestParam Short capacity,
    @RequestParam String engine,
    @RequestParam Short engineHP,
    @RequestParam Double engineVolume,
    @RequestParam String gearBox,
    @RequestParam Short numberOfGears,
    @RequestParam String country,
    @RequestParam Double price,
    @RequestParam("imageFile") MultipartFile imageFile,
    @RequestParam String description
) throws IOException {
    carService.addCar(
        brand,
        carModel,
        bodyType,
        layout,
        capacity,
        engine,
        engineHP,
        engineVolume,
        gearBox,
        numberOfGears,
        country,
        price,
        imageFile,
        description
    );
    return "addCar";
}

@PostMapping("/{car}/delete")
@PreAuthorize("hasAuthority('ADMIN')")
public String delete(@PathVariable Car car) {
    carService.deleteCar(car);
    return "redirect:/cars";
}

@GetMapping("/{car}/edit")
@PreAuthorize("hasAuthority('ADMIN')")
public String edit(@PathVariable Car car, Model model) {
    model.addAttribute("car", car);
    return "editCar";
}

@PostMapping("/{car}/edit")
@PreAuthorize("hasAuthority('ADMIN')")
public String carEditUpdate(
    @PathVariable Car car,
    @RequestParam String brand,
    @RequestParam String carModel,
    @RequestParam String bodyType,
    @RequestParam String layout,
    @RequestParam Short capacity,
    @RequestParam String engine,

```

```

        @RequestParam Short engineHP,
        @RequestParam Double engineVolume,
        @RequestParam String gearBox,
        @RequestParam Short numberOfGears,
        @RequestParam String country,
        @RequestParam("imageFile") MultipartFile imageFile,
        @RequestParam String description
    ) throws IOException {
        carService.editCar(
            car,
            brand,
            carModel,
            bodyType,
            layout,
            capacity,
            engine,
            engineHP,
            engineVolume,
            gearBox,
            numberOfGears,
            country,
            price,
            imageFile,
            description
        );
        return "redirect:/cars";
    }
}

```

---

## Файл MainController.java

```

package com.example.kursachdb.controllers;

import com.example.kursachdb.repos.CarRepo;
import com.example.kursachdb.service.CarService;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;

@Controller
@RequiredArgsConstructor
public class MainController {
    private final CarRepo carRepo;

    private final CarService carService;

    @GetMapping("/index")
    public String index(Model model) {
        model.addAttribute("cars", carRepo.findAll());
        return "index";
    }
}

```

```

    }

    @PostMapping("/uploadImage")
    public String uploadImage(@RequestParam("imageFile") MultipartFile imageFile, Model model) {
        try {
            carService.saveImage(imageFile);
        } catch (IOException e) {
            e.printStackTrace();
        }

        return "addCar";
    }
}

```

---

## Файл RegistrationController.java

```

package com.example.kursachdb.controllers;

import com.example.kursachdb.domain.Customer;
import com.example.kursachdb.service.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

@Controller
@RequiredArgsConstructor
public class RegistrationController {

    private final UserService userService;

    @GetMapping("/registration")
    public String registration() {
        return "registration";
    }

    @PostMapping("/registration")
    public String addUser(Customer customer, Model model) {
        if (!userService.addUser(customer)) {
            model.addAttribute("usernameError", "User exists!");
            return "registration";
        }

        return "redirect:/login";
    }
}

```

---

## Файл UserController.java

```

package com.example.kursachdb.controllers;

import com.example.kursachdb.domain.Customer;

```

```

import com.example.kursachdb.domain.Role;
import com.example.kursachdb.repos.CustomerRepo;
import com.example.kursachdb.service.UserService;
import lombok.RequiredArgsConstructor;
import org.springframework.security.access.prepost.PreAuthorize;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import java.util.Map;

@Controller
@RequiredArgsConstructor
@RequestMapping("/user")
public class UserController {
    private final UserService userService;
    private final CustomerRepo customerRepo;

    @GetMapping
    @PreAuthorize("hasAuthority('ADMIN')")
    public String userList(Model model) {
        model.addAttribute("users", customerRepo.findAll());

        return "customerList";
    }

    @GetMapping("/profile/{customer}")
    public String userProfile(@PathVariable Customer customer, Model model) {
        model.addAttribute("customer", customer);
        return "customerProfile";
    }

    @GetMapping("/{customer}")
    @PreAuthorize("hasAuthority('ADMIN')")
    public String userEditForm(@PathVariable Customer customer, Model model) {
        model.addAttribute("roles", Role.values());
        model.addAttribute("customer", customer);

        return "customerEdit";
    }

    @PostMapping("/{customer}/delete")
    @PreAuthorize("hasAuthority('ADMIN')")
    public String delete(@PathVariable Customer customer) {
        userService.deleteUser(customer);
        return "redirect:/user";
    }

    @PostMapping("/profile/{customer}/editEmail")
    public String userChangeEmail(@RequestParam String email, @PathVariable Customer customer) {
        userService.changeEmail(email, customer);
        return "redirect:/user/profile/" + customer.getCustomer_id();
    }

    @PostMapping("/profile/{customer}/editUsername")
    public String userChangeUsername(@RequestParam String username, @PathVariable Customer customer) {
        userService.changeUsername(username, customer);
        return "redirect:/user/profile/" + customer.getCustomer_id();
    }
}

```



```

@PostMapping("/profile/{customer}/editPassword")
public String userChangePassword(@RequestParam String up2, @PathVariable Customer customer) {
    userService.changePassword(up2, customer);
    return "redirect:/user/profile/" + customer.getCustomer_id();
}

@PostMapping
@PreAuthorize("hasAuthority('ADMIN')")
public String userSave(
    @RequestParam String username,
    @RequestParam Map<String, String> form,
    @RequestParam("customerId") Customer customer) {
    userService.changeUser(username, form, customer);

    return "redirect:/user";
}
}

```

---

## Файл BodyType.java

```

package com.example.kursachdb.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class BodyType {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    private String name;

    public BodyType(String name) {
        this.name = name;
    }

    public BodyType() {}

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }
}

```

```
}  
}
```

---

## Файл Brand.java

```
package com.example.kursachdb.domain;  
  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
  
@Entity  
public class Brand {  
    @Id  
    @GeneratedValue(strategy = GenerationType.AUTO)  
    private Integer id;  
  
    public Brand(String name) {  
        this.name = name;  
    }  
  
    private String name;  
  
    public Brand() {  
  
    }  
  
    public Integer getId() {  
        return id;  
    }  
  
    public void setId(Integer id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

---

## Файл Capacity.java

```
package com.example.kursachdb.domain;  
  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;
```

```

@Entity
public class Capacity {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    private Short numberOfSeats;

    public Capacity(Short numberOfSeats) {
        this.numberOfSeats = numberOfSeats;
    }

    public Capacity() {

    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public short getNumberOfSeats() {
        return numberOfSeats;
    }

    public void setNumberOfSeats(short numberOfSeats) {
        this.numberOfSeats = numberOfSeats;
    }
}

```

---

## Файл Car.java

```

package com.example.kursachdb.domain;

import javax.persistence.*;
import lombok.Data;

import java.util.Objects;

@Entity
@Table(name = "car", schema = "public")
@Data
public class Car {
    public Car(){}

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Car car)) return false;
        return Objects.equals(getCar_id(), car.getCar_id()) &&
            Objects.equals(getBrand(), car.getBrand()) &&
            Objects.equals(getModel(), car.getModel()) &&
    }
}

```

```

Objects.equals(getBodyType(), car.getBodyType()) &&
Objects.equals(getLayout(), car.getLayout()) &&
Objects.equals(getCapacity(), car.getCapacity()) &&
Objects.equals(getEngine(), car.getEngine()) &&
Objects.equals(getGearBox(), car.getGearBox()) &&
Objects.equals(getCountry(), car.getCountry()) &&
Objects.equals(getPicture(), car.getPicture()) &&
Objects.equals(getPrice(), car.getPrice());
}

@Override
public int hashCode() {
    return Objects.hash(getCar_id(), getBrand(), getModel(), getBodyType(),
        getLayout(), getCapacity(), getEngine(), getGearBox(), getCountry(),
        getPicture(), getPrice());
}

public Car(Brand brand, String model, BodyType bodyType,
    Layout layout, Capacity capacity, Engine engine,
    GearBox gearBox, Country country, Double price) {
    this.brand = brand;
    this.model = model;
    this.bodyType = bodyType;
    this.layout = layout;
    this.capacity = capacity;
    this.engine = engine;
    this.gearBox = gearBox;
    this.country = country;
    this.price = price;
}

@PrePersist
void preInsert() {
    if (this.picture == null)
    {
        this.picture = new Picture();
        this.picture.setId(1);
    }
    if (this.picture.getId() == null) {
        this.picture.setId(1);
    }
    if (this.description == null || this.description.length() < 1)
    {
        this.description = "Description was not added by the user";
    }
}

@Id
@Column(name = "car_id")
@GeneratedValue(strategy = GenerationType.AUTO)
private Integer car_id;

@ManyToOne
@JoinColumn(name = "brand_id")
private Brand brand;

@Column(name = "model", unique = true)
private String model;

```

```

@ManyToOne
@JoinColumn(name = "body_type_id")
private BodyType bodyType;

@ManyToOne
@JoinColumn(name = "layout_id")
private Layout layout;

@ManyToOne
@JoinColumn(name = "capacity_id")
private Capacity capacity;

@ManyToOne
@JoinColumn(name = "engine_id")
private Engine engine;

@ManyToOne
@JoinColumn(name = "gear_box_id")
private GearBox gearBox;

@ManyToOne
@JoinColumn(name = "county_id")
private Country country;

@ManyToOne
@JoinColumn(name = "picture_id")
private Picture picture;

@Column(name = "price")
private Double price;

@Column(name = "description")
private String description;

public Double getPrice() {
    return price;
}

public void setPrice(Double price) {
    this.price = price;
}

public Picture getPicture() {
    return picture;
}

public void setPicture(Picture picture) {
    this.picture = picture;
}
}

```

---

## Файл Country.java

```

package com.example.kursachdb.domain;

import javax.persistence.Entity;

```

```

import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Country {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    public Country(String name) {
        this.name = name;
    }

    private String name;

    public Country() {
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

---

## Файл Customer.java

```

package com.example.kursachdb.domain;

import lombok.*;
import org.hibernate.Hibernate;
import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import javax.persistence.*;
import java.util.Collection;
import java.util.Objects;
import java.util.Set;

@Entity
@Table(name = "customer")
@Getter
@Setter
@ToString

```

```

@RequiredArgsConstructor
public class Customer implements UserDetails {
    @Id
    @Column(name = "customer_id")
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer customer_id;

    @Column(name = "customer_name")
    private String username;

    @Column(name = "customer_password")
    private String customer_password;

    @Column(name = "email")
    private String email;

    @ElementCollection(targetClass = Role.class, fetch = FetchType.EAGER)
    @CollectionTable(name = "roles", joinColumns = @JoinColumn(name = "customer_id"))
    @Enumerated(EnumType.STRING)
    private Set<Role> roles;

    public boolean isAdmin(){
        return roles.contains(Role.ADMIN);
    }

    public Integer getCustomer_id() {
        return customer_id;
    }

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return getRoles();
    }

    @Override
    public String getPassword() {
        return customer_password;
    }

    @Override
    public boolean isAccountNonExpired() {
        return true;
    }

    @Override
    public boolean isAccountNonLocked() {
        return true;
    }

    @Override
    public boolean isCredentialsNonExpired() {
        return true;
    }

    @Override
    public boolean isEnabled() {
        return true;
    }
}

```

```

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null || Hibernate.getClass(this) != Hibernate.getClass(o)) return false;
    Customer customer = (Customer) o;
    return customer_id != null && Objects.equals(customer_id, customer.customer_id);
}

@Override
public int hashCode() {
    return getClass().hashCode();
}
}

```

---

## Файл Engine.java

```

package com.example.kursachdb.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Engine {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    private String name;

    private Short horsepower;

    private Double volume;

    public Engine(String name) {
        this.name = name;
    }

    public Engine() {

    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {

```



```

        this.name = name;
    }

    public Double getVolume() {
        return volume;
    }

    public void setVolume(Double capacity) {
        this.volume = capacity;
    }

    public Short getHorsePower() {
        return horsePower;
    }

    public void setHorsePower(Short horsePower) {
        this.horsePower = horsePower;
    }
}

```

---

## Файл GearBox.java

```

package com.example.kursachdb.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class GearBox {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    private Short numberOfGears;

    private String switchingType;

    public GearBox() {
    }

    public GearBox(String switchingType) {
        this.switchingType = switchingType;
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public Short getNumberOfGears() {
        return numberOfGears;
    }
}

```

```

    }

    public void setNumberOfGears(Short numberOfGears) {
        this.numberOfGears = numberOfGears;
    }

    public String getSwitchingType() {
        return switchingType;
    }

    public void setSwitchingType(String switchingType) {
        this.switchingType = switchingType;
    }
}

```

---

## Файл Layout.java

```

package com.example.kursachdb.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Layout {

    public Layout(String type_of_drive) {
        this.type_of_drive = type_of_drive;
    }

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer layout_id;

    private String engine_location;

    private String type_of_drive;

    public Layout() {}

    public String getEngine_location() {
        return engine_location;
    }

    public void setEngine_location(String engine_location) {
        this.engine_location = engine_location;
    }

    public String getType_of_drive() {
        return type_of_drive;
    }

    public void setType_of_drive(String type_of_drive) {
        this.type_of_drive = type_of_drive;
    }
}

```

---

## Файл Picture.java

```
package com.example.kursachdb.domain;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Picture {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;

    private String name;

    public Picture(){}

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

---

## Файл Role.java

```
package com.example.kursachdb.domain;

import org.springframework.security.core.GrantedAuthority;

public enum Role implements GrantedAuthority {
    USER, ADMIN;

    @Override
    public String getAuthority() {
        return name();
    }
}
```

---

## Файл CarRepo.java

```

package com.example.kursachdb.repos;

import com.example.kursachdb.domain.Car;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface CarRepo extends CrudRepository<Car, Integer> {
    @Query("SELECT c FROM Car c WHERE " +
        "(:brandId IS NULL OR c.brand.id = :brandId) AND " +
        "(:bodyTypeId IS NULL OR c.bodyType.id = :bodyTypeId) AND " +
        "(:capacityId IS NULL OR c.capacity.id = :capacityId) AND " +
        "(:gearBoxId IS NULL OR c.gearBox.id = :gearBoxId)")
    List<Car> findCarsByBrandIdAndBodyTypeIdAndCapacityIdAndGearBoxId(
        @Param("brandId") Integer brandId,
        @Param("bodyTypeId") Integer bodyTypeId,
        @Param("capacityId") Integer capacityId,
        @Param("gearBoxId") Integer gearBoxId
    );
}

```

---

## Файл CarService.java

```

package com.example.kursachdb.service;

import com.example.kursachdb.domain.*;
import com.example.kursachdb.repos.*;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.ui.Model;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.util.UUID;

@Service
@RequiredArgsConstructor
public class CarService implements ICarService{
    private final BodyTypeRepo bodyTypeRepo;

    private final BrandRepo brandRepo;

    private final CapacityRepo capacityRepo;

    private final CarRepo carRepo;

    private final CountryRepo countryRepo;

    private final EngineRepo engineRepo;

    private final GearBoxRepo gearBoxRepo;
}

```

```

private final LayoutRepo layoutRepo;

private final PictureRepo pictureRepo;

public void deleteCar(Car car) {
    carRepo.delete(car);
}

public void findCar(
    Model model,
    Integer makeTypeSelect,
    Integer bodyTypeSelect,
    Integer numSeatsTypeSelect,
    Integer gearBoxTypeSelect
) {
    Iterable<Car> cars;

    if (makeTypeSelect == null && bodyTypeSelect == null && numSeatsTypeSelect == null && gearBoxTypeSelect == null)
    {
        cars = carRepo.findAll();
    }
    else
    {
        cars = carRepo.findCarsByBrandIdAndBodyTypeIdAndCapacityIdAndGearBoxId(makeTypeSelect, bodyTypeSelect, numSeatsTypeSelect, gearBoxTypeSelect);
    }

    model.addAttribute("makes", brandRepo.findAll());
    model.addAttribute("bodyTypes", bodyTypeRepo.findAll());
    model.addAttribute("layouts", layoutRepo.findAll());
    model.addAttribute("gearBoxes", gearBoxRepo.findAll());
    model.addAttribute("capacities", capacityRepo.findAll());

    model.addAttribute("cars", cars);
}

private void innerAddCar(
    Car car,
    String brand,
    String carModel,
    String bodyType,
    String layout,
    Short capacity,
    String engine,
    Short engineHP,
    Double engineVolume,
    String gearBox,
    Short numberOfGears,
    String country,
    Double price,
    MultipartFile imageFile,
    String description
) throws IOException {
    Brand brandFromDB = brandRepo.findBrandByName(brand);
    BodyType bodyTypeFromDB = bodyTypeRepo.findByName_body_type(bodyType);
    Layout layoutFromDB = layoutRepo.findByType_of_drive(layout);
    Capacity capacityFromDB = capacityRepo.findByNumber_of_seats(capacity);
    Engine engineFromDB = engineRepo.findEngineByNameAndHorsePowerAndVolume(engine, engineHP, engineVolume);
    GearBox gearBoxFromDB = gearBoxRepo.findBySwitchingTypeAndNumberOfGears(gearBox, numberOfGears);
}

```

```

Country countryFromDB = countryRepo.findByName(country);
Picture pictureFromDB = null;

if (!imageFile.isEmpty())
{
    String resultFilename = UUID.randomUUID() + imageFile.getOriginalFilename();
    pictureFromDB = pictureRepo.findPictureByName(resultFilename);

    if (pictureFromDB == null)
    {
        String folder = "src/main/resources/static/images/";
        byte[] bytes = imageFile.getBytes();
        pictureFromDB = new Picture();
        Path path = Paths.get(folder + resultFilename);
        Files.write(path, bytes);
        pictureFromDB.setName(resultFilename);
        pictureRepo.save(pictureFromDB);
    }
}

if (brandFromDB == null)
{
    brandFromDB = new Brand(brand);
    brandRepo.save(brandFromDB);
}
if (bodyTypeFromDB == null)
{
    bodyTypeFromDB = new BodyType(bodyType);
    bodyTypeRepo.save(bodyTypeFromDB);
}
if (layoutFromDB == null)
{
    layoutFromDB = new Layout(layout);
    layoutRepo.save(layoutFromDB);
}
if (capacityFromDB == null)
{
    capacityFromDB = new Capacity(capacity);
    capacityRepo.save(capacityFromDB);
}
if (engineFromDB == null)
{
    engineFromDB = new Engine(engine);
    engineFromDB.setHorsePower(engineHP);
    engineFromDB.setVolume(engineVolume);
    engineRepo.save(engineFromDB);
}
if (gearBoxFromDB == null)
{
    gearBoxFromDB = new GearBox(gearBox);
    gearBoxFromDB.setNumberOfGears(numberOfGears);
    gearBoxRepo.save(gearBoxFromDB);
}
if (countryFromDB == null)
{
    countryFromDB = new Country(country);
    countryRepo.save(countryFromDB);
}

```

```

    if (pictureFromDB == null)
    {
        pictureFromDB = new Picture();
        pictureFromDB.setId(1);
    }

    if (pictureFromDB.getId() == null) {
        pictureFromDB.setId(1);
    }

    if (car.getPicture() == null || car.getPicture().getId() == null)
    {
        car.setPicture(pictureFromDB);
    }

    car.setBrand(brandFromDB);
    car.setBodyType(bodyTypeFromDB);
    car.setLayout(layoutFromDB);
    car.setCapacity(capacityFromDB);
    car.setEngine(engineFromDB);
    car.setGearBox(gearBoxFromDB);
    car.setCountry(countryFromDB);
    car.setModel(carModel);
    car.setPrice(price);
    car.setDescription(description);

    carRepo.save(car);
}

@Override
public void addCar(
    String brand,
    String carModel,
    String bodyType,
    String layout,
    Short capacity,
    String engine,
    Short engineHP,
    Double engineVolume,
    String gearBox,
    Short numberOfGears,
    String country,
    Double price,
    MultipartFile imageFile,
    String description
) throws IOException {
    innerAddCar(
        new Car(),
        brand,
        carModel,
        bodyType,
        layout,
        capacity,
        engine,
        engineHP,
        engineVolume,
        gearBox,
        numberOfGears,
        country,

```

```

        price,
        imageFile,
        description
    );
}

@Override
public void saveImage(MultipartFile imageFile) throws IOException {
    String folder = "src/main/resources/static/images/";
    byte[] bytes = imageFile.getBytes();
    Path path = Paths.get(folder + imageFile.getOriginalFilename());
    Files.write(path, bytes);
}

public void editCar(
    Car car,
    String brand,
    String carModel,
    String bodyType,
    String layout,
    Short capacity,
    String engine,
    Short engineHP,
    Double engineVolume,
    String gearBox,
    Short numberOfGears,
    String country,
    Double price,
    MultipartFile imageFile,
    String description
) throws IOException {
    innerAddCar(
        car,
        brand,
        carModel,
        bodyType,
        layout,
        capacity,
        engine,
        engineHP,
        engineVolume,
        gearBox,
        numberOfGears,
        country,
        price,
        imageFile,
        description
    );
}
}

```

---

## Файл ICarService.java

```

package com.example.kursachdb.service;

import org.springframework.web.multipart.MultipartFile;

```



```

import java.io.IOException;

public interface ICarService {
    void addCar(
        String brand,
        String carModel,
        String bodyType,
        String layout,
        Short capacity,
        String engine,
        Short engineHP,
        Double engineVolume,
        String gearBox,
        Short numberOfGears,
        String country,
        Double price,
        MultipartFile imageFile,
        String description
    ) throws IOException;

    void saveImage(MultipartFile imageFile) throws IOException;
}

```

---

## Файл UserService.java

```

package com.example.kursachdb.service;

import com.example.kursachdb.domain.Customer;
import com.example.kursachdb.domain.Role;
import com.example.kursachdb.repos.CustomerRepo;
import lombok.RequiredArgsConstructor;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;

import java.util.Arrays;
import java.util.Collections;
import java.util.Map;
import java.util.Set;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
public class UserService implements UserDetailsService {
    private final CustomerRepo customerRepo;

    private final PasswordEncoder passwordEncoder;

    @Override
    public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
        return customerRepo.findByUsername(username);
    }
}

```

```

public void changeUser(String username, Map<String, String> form, Customer customer) {
    customer.setUsername(username);

    Set<String> roles = Arrays.stream(Role.values())
        .map(Role::name)
        .collect(Collectors.toSet());

    customer.getRoles().clear();

    for(String key : form.keySet())
    {
        if(roles.contains(key))
        {
            customer.getRoles().add(Role.valueOf(key));
        }
    }
    customerRepo.save(customer);
}

public void deleteUser(Customer customer) {
    customerRepo.delete(customer);
}

public void changeEmail(String email, Customer customer) {
    customer.setEmail(email);
    if (customerRepo.findCustomerByEmail(email) == null)
    {
        customerRepo.save(customer);
    }
}

public void changeUsername(String username, Customer customer) {
    customer.setUsername(username);
    if (customerRepo.findByUsername(username) == null)
    {
        customerRepo.save(customer);
    }
}

public void changePassword(String password, Customer customer) {
    customer.setCustomer_password(passwordEncoder.encode(customer.getPassword()));
}

public boolean addUser(Customer customer) {
    Customer customerFromDb = customerRepo.findByUsername(customer.getUsername());

    if (customerFromDb != null) {
        return false;
    }

    customer.setRoles(Collections.singleton(Role.USER));
    customer.setCustomer_password(passwordEncoder.encode(customer.getPassword()));

    customerRepo.save(customer);

    return true;
}
}

```

---

## Файл KursachDbApplication.java

```
package com.example.kursachdb;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@SpringBootApplication
public class KursachDbApplication implements WebMvcConfigurer {

    public static void main(String[] args) {

        SpringApplication.run(KursachDbApplication.class, args);
    }

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/static/**").addResourceLocations("classpath:/static/");
    }
}
```

---

## 7.2.2 Код пользовательского клиента

### Файл footer.ftlh

```
<#macro page>
    <!-- ***** Footer Start ***** -->
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <p>
                    Copyright © 2020 Company Name
                    - Template by: <a href="https://www.phpjabbers.com/">PHPJabbers.com</a>
                </p>
            </div>
        </div>
    </div>
    <!-- ***** Footer End ***** -->
</#macro>
```

---

### Файл header.ftlh

```
<#macro page>
    <#include "security.ftlh">
    <!-- ***** Header Area Start ***** -->
    <div class="container">
        <div class="row">
            <div class="col-12">
```

```

<nav class="main-nav">
  <!-- ***** Logo Start ***** -->
  <a href="/index" class="logo">Car Dealer<em> Website</em></a>
  <!-- ***** Logo End ***** -->
  <!-- ***** Menu Start ***** -->
  <ul class="nav">
    <li><a href="/index">Home</a></li>
    <li><a href="/cars">Cars</a></li>
    <#if isAdmin>
      <li><a href="/cars/add">Add car</a></li>
      <li><a href="/user">Users List</a></li>
    </#if>
    <#if known>
      <#--<li><a href="#">${name}</a></li>-->
      <li class="dropdown">
        <a class="dropdown-toggle active"
          data-toggle="dropdown" href="#" role="button"
          aria-haspopup="true"
          aria-expanded="false">${name}</a>
        <div class="dropdown-menu">
          <a class="dropdown-item"
            href="/user/profile/${id}">Personal area
          </a>
        </div>
      </li>
    <#else>
      <li class="dropdown">
        <a class="dropdown-toggle active"
          data-toggle="dropdown" href="#" role="button"
          aria-haspopup="true"
          aria-expanded="false">Account</a>
        <div class="dropdown-menu">
          <a class="dropdown-item" href="/login">Login</a>
          <a class="dropdown-item"
            href="/registration">Registration</a>
        </div>
      </li>
    </#if>
    <#if known>
      <div class="main-button">
        <form action="/logout" method="post">
          <button class="btn btn-primary"
            type="submit">Sign Out</button>
          <input type="hidden" name="_csrf"
            value="${_csrf.token}" />
        </form>
      </div>
    </#if>
  </ul>
  <a class='menu-trigger'>
    <span>Menu</span>
  </a>
  <!-- ***** Menu End ***** -->
</nav>
</div>
</div>
</div>
<!-- ***** Header Area End ***** -->
</#macro>

```

---

## Файл security.ftlh

```
<#assign known = Session.SPRING_SECURITY_CONTEXT??>
<#if known>
    <#assign
        user = Session.SPRING_SECURITY_CONTEXT.authentication.principal
        name = user.getUsername()
        id = user.getCustomer_id()
        isAdmin = user.isAdmin()
    >
<#else>
    <#assign
        name = ""
        isAdmin = false
    >
</#if>
```

---

## Файл addCar.ftlh

```
<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    0i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display=
    swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
    href="../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"
    href="../static/css/font-awesome.css">

    <link rel="stylesheet" href="../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
        </div>
    </div>
</div>
```

```

        <span></span>
    </div>
</div>
<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Add Area Starts ***** -->
<section class="section" id="contact-us" style="margin-top: 0">
    <div class="container-fluid">
        <div class="contact-form section-bg" style="background-image:
            url(../static/images/contact-1-720x480.jpg)">
            <div class="container">
                <div class="row">
                    <div class="col-lg-10 offset-lg-1">
                        <div class="cta-content">
                            <br>
                            <br>
                            <h2><em>Add</em> car panel</h2>
                            <p>Ut consectetur, metus sit amet aliquet placerat,
                                enim est ultricies ligula</p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="row">
                <div class="col-md-3 col-sm-12">
                    <fieldset>
                        <input type="text" name="brand" placeholder="Brand"
                            required="">
                    </fieldset>
                </div>
                <div class="col-md-3 col-sm-12">
                    <fieldset>
                        <input type="text" name="carModel"
                            placeholder="Model" required="">
                    </fieldset>
                </div>
                <div class="col-md-3 col-sm-12">
                    <fieldset>
                        <input type="text" name="bodyType"
                            placeholder="Body Type" required="">
                    </fieldset>
                </div>
                <div class="col-md-3 col-sm-12">
                    <fieldset>
                        <input type="text" name="layout"
                            placeholder="Layout" required="">
                    </fieldset>
                </div>
            </div>
        </div>
    </div>

```

```

        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="number" name="capacity"
                placeholder="Capacity" required="" min="0"
                oninput="validity.valid||(value=value.replace(/\D+
                    /g, ''))">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="text" name="engine"
                placeholder="Engine" required="">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="number" name="engineHP"
                placeholder="Horse Power" required="" min="0"
                oninput="validity.valid||(value=value.replace(/\D+
                    /g, ''))">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="number" name="engineVolume"
                placeholder="Engine displacement" required=""
                min="0" step="0.1"
                oninput="validity.valid||(value=value.replace(/\D+
                    /g, ''))">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="text" name="gearBox"
                placeholder="GearBox" required="">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="number" name="numberOfGears"
                placeholder="Number of gears" required="" min="0"
                oninput="validity.valid||(value=value.replace(/\D+
                    /g, ''))">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="text" name="country"
                placeholder="Country" required="">
        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="number" name="price"
                placeholder="Price" min="0"
                oninput="validity.valid||(value=value.replace(/\D+
                    /g, ''))" required="">

```

```

        </fieldset>
    </div>
    <div class="col-md-3 col-sm-12">
        <fieldset>
            <input type="file" name="imageFile"
                accept="image/png, image/jpeg, image/jpg"/>
        </fieldset>
    </div>
    <div class="col-lg-12">
        <fieldset>
            <textarea name="description" rows="6"
                placeholder="Description" required=""></textarea>
        </fieldset>
    </div>
    <div class="col-md-2">
        <fieldset>
            <button type="submit" id="form-submit"
                class="main-button">Add</button>
        </fieldset>
    </div>
</div>
<input type="hidden" name="_csrf" value="{_csrf.token}" />
</form>
</div>
</div>

</section>
<!-- ***** Add Area Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    {f.page}</f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```



## Файл carDetails.ftlh

```
<!DOCTYPE html>
<html lang="en">

<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">
  <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,200i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display=swap" rel="stylesheet">

  <title>PHPJabbers.com | Free Car Dealer Website Template</title>

  <link rel="stylesheet" type="text/css" href="../static/css/bootstrap.min.css">

  <link rel="stylesheet" type="text/css" href="../static/css/font-awesome.css">

  <link rel="stylesheet" href="../static/css/style.css">

</head>

<body>

  <!-- ***** Preloader Start ***** -->
  <div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
      <span class="dot"></span>
      <div class="dots">
        <span></span>
        <span></span>
        <span></span>
      </div>
    </div>
  </div>
  <!-- ***** Preloader End ***** -->

  <!-- ***** Header Area Start ***** -->
  <#import "blocks/header.ftlh" as h>

  <header class="header-area header-sticky">
    <@h.page>
    </@h.page>
  </header>
  <!-- ***** Header Area End ***** -->

  <!-- ***** Call to Action Start ***** -->
  <section class="section section-bg" id="call-to-action"
  style="background-image:
  url(../static/images/banner-image-1-1920x500.jpg)">
    <div class="container">
      <div class="row">
        <div class="col-lg-10 offset-lg-1">
          <div class="cta-content">
            <br>
```

```

        <br>
        <h2><em>${car.price}</em></h2>
        <p>Lorem ipsum dolor sit amet, consectetur.</p>
    </div>
</div>
</div>
</section>
<!-- ***** Call to Action End ***** -->

<!-- ***** Fleet Starts ***** -->
<section class="section" id="trainers">
    <div class="container">
        <br>
        <br>

        <div class="text-xs-center">
            
        </div>

        <!--<div id="carouselExampleIndicators" class="carousel slide"
data-ride="carousel">
            <ol class="carousel-indicators">
                <li data-target="#carouselExampleIndicators" data-slide-to="0"
                    class="active"></li>
                <li data-target="#carouselExampleIndicators"
                    data-slide-to="1"></li>
                <li data-target="#carouselExampleIndicators"
                    data-slide-to="2"></li>
            </ol>
            <div class="carousel-inner">
                <div class="carousel-item active">
                    
                </div>
                <div class="carousel-item">
                    
                </div>
                <div class="carousel-item">
                    
                </div>
            </div>
            <a class="carousel-control-prev"
                href="#carouselExampleIndicators" role="button"
                data-slide="prev">
                <span class="carousel-control-prev-icon"
                    aria-hidden="true"></span>
                <span class="sr-only">Previous</span>
            </a>
            <a class="carousel-control-next"
                href="#carouselExampleIndicators" role="button"
                data-slide="next">
                <span class="carousel-control-next-icon"

```

```

        aria-hidden="true"></span>
        <span class="sr-only">Next</span>
    </a>
</div>-->

<br>
<br>

<div class="row" id="tabs">
    <div class="col-lg-4">
        <ul>
            <li><a href='#tabs-1'><i class="fa fa-cog"></i> Vehicle
            Specs</a></li>
            <li><a href='#tabs-2'><i class="fa fa-info-circle"></i>
            Vehicle Description</a></li>
            <li><a href='#tabs-3'><i class="fa fa-plus-circle"></i>
            Vehicle Extras</a></li>
            <li><a href='#tabs-4'><i class="fa fa-phone"></i> Contact
            Details</a></li>-->
        </ul>
    </div>
    <div class="col-lg-8">
        <section class='tabs-content' style="width: 100%;">
            <article id='tabs-1'>
                <h4>Vehicle Specs</h4>

                <div class="row">
                    <div class="col-sm-6">
                        <label>Make</label>

                        <p>${car.brand.name}</p>
                    </div>

                    <div class="col-sm-6">
                        <label> Model</label>

                        <p>${car.model}</p>
                    </div>

                    <div class="col-sm-6">
                        <label>Car body style</label>

                        <p>${car.bodyType.name}</p>
                    </div>

                    <div class="col-sm-6">
                        <label>Country</label>

                        <p>${car.country.name}</p>
                    </div>

                    <div class="col-sm-6">
                        <label>Fuel</label>

                        <p>Diesel</p>
                    </div>

                    <div class="col-sm-6">
                        <label>Engine displacement</label>

```

```

        <p>${car.engine.volume}</p>
    </div>

    <div class="col-sm-6">
        <label>Power</label>

        <p>${car.engine.horsePower} hp</p>
    </div>

    <div class="col-sm-6">
        <label>Gearbox</label>

        <p>${car.gearBox.switchingType}
        (${car.gearBox.numberOfGears} speed)</p>
    </div>

    <div class="col-sm-6">
        <label>Type of drivetrain layouts</label>

        <p>${car.layout.type_of_drive}</p>
    </div>

    <div class="col-sm-6">
        <label>Number of seats</label>

        <p>#${car.capacity.numberOfSeats}</p>
    </div>
</div>
</article>
<article id='tabs-2'>
    <h4>Vehicle Description</h4>

    <p>${car.description}</p>
</article>
<!--<article id='tabs-3'>
    <h4>Vehicle Extras</h4>

    <div class="row">
        <div class="col-sm-6">
            <p>ABS</p>
        </div>
        <div class="col-sm-6">
            <p>Leather seats</p>
        </div>
        <div class="col-sm-6">
            <p>Power Assisted Steering</p>
        </div>
        <div class="col-sm-6">
            <p>Electric heated seats</p>
        </div>
        <div class="col-sm-6">
            <p>New HU and AU</p>
        </div>
        <div class="col-sm-6">
            <p>Xenon headlights</p>
        </div>
    </div>
</div>

```

```

</article>-->
<#--<article id='tabs-4'>
  <h4>Contact Details</h4>

  <div class="row">
    <div class="col-sm-6">
      <label>Name</label>

      <p>John Smith</p>
    </div>
    <div class="col-sm-6">
      <label>Phone</label>

      <p>123-456-789 </p>
    </div>
    <div class="col-sm-6">
      <label>Mobile phone</label>
      <p>456789123 </p>
    </div>
    <div class="col-sm-6">
      <label>Email</label>
      <p><a href="#">john@carsales.com</a></p>
    </div>
  </div>
</article>-->
</section>
</div>
</div>
</div>
</section>
<!-- ***** Fleet Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
  <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл cars.ftlh

```
<!DOCTYPE html>
<html lang="en">
<#include "blocks/security.ftlh">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,200i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display=swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
href="../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"
href="../static/css/font-awesome.css">

    <link rel="stylesheet" href="../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>
</div>
<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Call to Action Start ***** -->
<section class="section section-bg" id="call-to-action"
style="background-image: url(../static/images/banner-image-1-1920x500.jpg)">
    <div class="container">
```

```

<div class="row">
  <div class="col-lg-10 offset-lg-1">
    <div class="cta-content">
      <br>
      <br>
      <h2>Our <em>Cars</em></h2>
      <p>Ut consectetur, metus sit amet aliquet placerat, enim
        est ultricies ligula</p>
    </div>
  </div>
</div>
</section>
<!-- ***** Call to Action End ***** -->

<!-- ***** Fleet Starts ***** -->
<section class="section" id="trainers">
  <div class="container">
    <br>
    <br>
    <div class="contact-form">
      <form action="#" id="contact">
        <div class="row">
          <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
            <div class="form-group">
              <label>Body Type:</label>

              <select name="bodyTypeSelect">
                <option value="">All</option>
                <#if bodyTypes?has_content>
                  <#list bodyTypes as bodyType>
                    <option
                      value="{bodyType.id}">{bodyType.name}
                    </option>
                  </#list>
                <#else>
                  No body types
                </#if>
              </select>
            </div>
          </div>

          <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
            <div class="form-group">
              <label>Make:</label>

              <select name="makeTypeSelect">
                <option value="">All</option>
                <#if makes?has_content>
                  <#list makes as make>
                    <option
                      value="{make.id}">{make.name}</option>
                  </#list>
                <#else>
                  No make
                </#if>
              </select>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>

```

```

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
  <div class="form-group">
    <label>Gearbox:</label>

    <select name="gearBoxTypeSelect">
      <option value="">All</option>
      <#if gearBoxes?has_content>
        <#list gearBoxes as gearBox>
          <option
            value="{gearBox.id}">{gearBox.switchi
              ngType} ({gearBox.numberOfGears}
                speed)</option>
        </#list>
      <#else>
        No make
      </#if>
    </select>
  </div>
</div>

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
  <div class="form-group">
    <label>Number of seats:</label>

    <select name="numSeatsTypeSelect">
      <option value="">All</option>
      <#if capacities?has_content>
        <#list capacities as capacity>
          <option
            value="{capacity.id}">{capacity.numbe
              rOfSeats}</option>
        </#list>
      <#else>
        No make
      </#if>
    </select>
  </div>
</div>
</div>

<div class="col-sm-4 offset-sm-4">
  <div class="main-button text-center">
    <button type="submit">Search</button>
  </div>
</div>
<br>
<br>
<input type="hidden" name="_csrf" value="{_csrf.token}" />
</form>
</div>

<div class="row">
  <#if cars?has_content>
    <#list cars as car>
      <div class="col-lg-4">
        <div class="trainer-item">
          <div class="image-thumb">
            <#if car.picture.name?has_content>

```



```

        
    <#else>
        
    </#if>

</div>
<div class="down-content">
<span>
    <sup>${car.price}</sup>
</span>

    <h4>${car.brand.name} ${car.model}</h4>

    <p>
        <i class="fa fa-dashboard"></i>
        ${car.engine.name}
        <i class="fa fa-cog"></i>
        ${car.gearBox.switchingType}
        (${car.gearBox.numberOfGears} speed)
        <i class="fa fa-flag"></i>
        ${car.country.name}
        <i class="fa fa-car"></i>
        ${car.bodyType.name}
    </p>

    <ul class="social-icons">
        <li><a href="/cars/${car.car_id}">+ View
            Car</a></li>
        <#if isAdmin>
            <li><a
                href="/cars/${car.car_id}/edit">+ Edit
                Car</a></li>
            <li>
                <a>
                    <form
                        action="/cars/${car.car_id}/delete"
                        method="post">
                        <button class="btn
                            btn-primary"
                            type="submit">Delete
                            Car</button>
                        <input type="hidden"
                            name="_csrf"
                            value="${_csrf.token}" />
                    </form>
                </a>
            </li>
        </#if>
    </ul>
</div>
</div>
</div>
<#list>
<#else>
    <section class="section section-bg" id="call-to-action"

```

```

        style="background-image:
        url(../static/images/banner-image-1-1920x500.jpg)">
        <div class="container">
            <div class="cta-content">
                <h2>The search has not given any
                <em>results</em></h2>
                <p>We are definitely expanding our range.</p>
            </div>
        </div>
    </section>
</#if>
</div>

<br>

</div>
</section>
<!-- ***** Fleet Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../static/js/popper.js"></script>
<script src="../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../static/js/scrollreveal.min.js"></script>
<script src="../static/js/waypoints.min.js"></script>
<script src="../static/js/jquery.counterup.min.js"></script>
<script src="../static/js/imgfix.min.js"></script>
<script src="../static/js/mixitup.js"></script>
<script src="../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл customerEdit.ftlh

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
<meta name="description" content="">
<meta name="author" content="">
<link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display
=swap" rel="stylesheet">

<title>PHPJabbers.com | Free Car Dealer Website Template</title>

<link rel="stylesheet" type="text/css"
href="../static/css/bootstrap.min.css">

<link rel="stylesheet" type="text/css"
href="../static/css/font-awesome.css">

<link rel="stylesheet" href="../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>
</div>
<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Contact Us Area Starts ***** -->
<section class="section" id="contact-us" style="margin-top: 0">
    <div class="container-fluid">
        <div class="contact-form section-bg" style="background-image:
url(../static/images/contact-1-720x480.jpg)">
            <div class="container">
                <div class="row">
                    <div class="col-lg-10 offset-lg-1">
                        <div class="cta-content">
                            <br>
                            <br>
                            <h2><em>User</em> list</h2>
                            <p>Ut consectetur, metus sit amet aliquet placerat,
enim est ultricies ligula</p>

```



```

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display
    =swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
    href="../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"
    href="../static/css/font-awesome.css">

    <link rel="stylesheet" href="../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>

<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>

<!-- ***** Header Area End ***** -->

<!-- ***** Contact Us Area Starts ***** -->
<section class="section" id="contact-us" style="margin-top: 0">
    <div class="container-fluid">
        <div class="contact-form section-bg" style="background-image:
        url(../static/images/contact-1-720x480.jpg)">
            <div class="container">
                <div class="row">
                    <div class="col-lg-10 offset-lg-1">
                        <div class="cta-content">
                            <br>

```

```

        <br>
        <h2><em>User</em> list</h2>
        <p>Ut consectetur, metus sit amet aliquet placerat,
        enim est ultricies ligula</p>
    </div>
</div>
</div>
</div>
<form id="contact" action="" method="post">
    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Role</th>
                <th></th>
            </tr>
        </thead>
        <tbody>
            <#list users as customer>
                <tr>
                    <td>${customer.username}</td>
                    <td><#list customer.roles as role>${role}<#sep>,</td>
                    <td><a
                        href="/user/${customer.customer_id}">edit</a></td>
                    <td>
                        <a>
                            <form
                                action="/user/${customer.customer_id}/delete" method="post">
                                    <button class="btn btn-primary"
                                        type="submit">Delete</button>
                                    <input type="hidden" name="_csrf"
                                        value="${_csrf.token}" />
                                </form>
                        </a>
                    </td>
                </tr>
            </#list>
        </tbody>
    </table>
    <input type="hidden" name="_csrf" value="${_csrf.token}" />
</form>
</div>
</div>
</section>
<!-- ***** Contact Us Area Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->

```

```

<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл customerProfile.ftlh

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display
    =swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css" href="../../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="../../static/css/font-awesome.css">

    <link rel="stylesheet" href="../../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>
</div>

```

```

<!-- ***** Preloader End ***** -->

<!-- ***** Blog Start ***** -->
<section class="section" id="our-classes">
  <div class="container">
    <div class="row">
      <div class="col-md-4">
        <div class="section-heading">
          <h2>Hello <em>${customer.username}</em></h2>
          
          <h4>${customer.email}</h4>
        </div>
      </div>
      <div class="col-md-8">
        <div class="section-heading">
          <h2>Personal <em>Area</em></h2>
          
          <p>Nunc urna sem, laoreet ut metus id, aliquet consequat
            magna. Sed viverra ipsum dolor, ultricies fermentum massa
            consequat eu.</p>
        </div>
      </div>
    </div>
    <div class="row" id="tabs">
      <div class="col-lg-4">
        <ul>
          <li><a href='#tabs-1'>Change email</a></li>
          <li><a href='#tabs-2'>Change username</a></li>
          <li><a href='#tabs-3'>Change password</a></li>
        </ul>
      </div>
      <div class="col-lg-8">
        <section class='tabs-content'>
          <article id='tabs-1'>
            <form
              action="/user/profile/#{customer.customer_id}/editEmail"
              method="post">
              <div class="form-group row">
                <label class="col-sm-4
                  col-form-label">Email:</label>
                <div class="col-sm-8">
                  <input name="email" type="text"
                    pattern="^[^\s@]+@([^\s@.,]+\.)+[^\.s@.,]{2,}$"
                    class="form-control"
                    placeholder="some@some.com (Your new
                      Email)"
                    value="<#if
                      customer??>${customer.email}</#if>"
                    required="">
                </div>
              </div>
              <input type="hidden" name="_csrf"
                value="${_csrf.token}" />
              <button class="btn btn-primary"
                type="submit">Change</button>
            </form>
          </article>
        </section>
      </div>
    </div>
  </div>

```



```

<article id='tabs-2'>
  <form
    action="/user/profile/#{customer.customer_id}/editUser
    name" method="post">
    <div class="form-group row">
      <label class="col-sm-4 col-form-label">User
      Name :</label>
      <div class="col-sm-8">
        <input type="text" name="username"
        value="<#if
        customer??>${customer.username}</#if>"
        class="form-control"
        placeholder="User name" />
      </div>
    </div>
    <input type="hidden" name="_csrf"
    value="${_csrf.token}" />
    <button class="btn btn-primary"
    type="submit">Change</button>
  </form>
</article>
<article id='tabs-3'>
  <form
    action="/user/profile/#{customer.customer_id}/editPass
    word" method="post"
    oninput='up2.setCustomValidity(up2.value !=
    up.value ? "Passwords do not match." : "")'>
    <div class="form-group row">
      <div class="col-sm-8">
        <p>
          <label for="password1">Password:</label>
          <input id="password1" class="form-control
          " type="password required name=up>
        </p>
        <p>
          <label for="password2">Confirm
          password:</label>
          <input id="password2" class="form-control"
          type="password name=up2>
        </p>
      </div>
    </div>
    <input type="hidden" name="_csrf"
    value="${_csrf.token}" />
    <button class="btn btn-primary"
    type="submit">Change</button>
  </form>
</article>
</section>
</div>
</div>
</div>
</section>
<!-- ***** Blog End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
  <@h.page>

```

```

</@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл editCar.ftlh

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display=
    swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
    href="../../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"

```

```

href="../../../static/css/font-awesome.css">

<link rel="stylesheet" href="../../../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>
<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Add Area Starts ***** -->
<section class="section" id="contact-us" style="margin-top: 0">
    <div class="container-fluid">
        <div class="contact-form section-bg" style="background-image:
url(../../../static/images/contact-1-720x480.jpg)">
            <div class="container">
                <div class="row">
                    <div class="col-lg-10 offset-lg-1">
                        <div class="cta-content">
                            <br>
                            <br>
                            <h2><em>Edit</em> car panel</h2>
                            <p>Ut consectetur, metus sit amet aliquet placerat,
enim est ultricies ligula</p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="row">
                <div class="col-md-3 col-sm-12">
                    <fieldset>
                        <input type="text" name="brand"
value="{car.brand.name}" placeholder="Brand"
required="">
                    </fieldset>
                </div>
                <div class="col-md-3 col-sm-12">

```

```

        <fieldset>
            <input type="text" name="carModel"
                value="{car.model}" placeholder="Model"
                required="">
        </fieldset>
    </div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="text" name="bodyType"
            value="{car.bodyType.name}" placeholder="Body
            Type" required="">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="text" name="layout"
            value="{car.layout.type_of_drive}"
            placeholder="Layout" required="">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="number" name="capacity"
            placeholder="Capacity"
            value="{car.capacity.numberOfSeats}" required=""
            min="0"
            oninput="validity.valid||(value=value.replace(/\D+
            /g, ''))">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="text" name="engine"
            placeholder="Engine" value="{car.engine.name}"
            required="">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="number" name="engineHP"
            placeholder="Horse Power"
            value="{car.engine.horsePower}" required=""
            min="0"
            oninput="validity.valid||(value=value.replace(/\D+
            /g, ''))">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="number" name="engineVolume"
            placeholder="Engine displacement"
            value="{car.engine.volume}" required="" min="0"
            step="0.1"
            oninput="validity.valid||(value=value.replace(/\D+
            /g, ''))">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>

```

```

        <input type="text" name="gearBox"
        placeholder="GearBox"
        value="{car.gearBox.switchingType}" required="">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="number" name="numberOfGears"
        placeholder="Number of gears"
        value="{car.gearBox.numberOfGears}" required=""
        min="0"
        oninput="validity.valid||(value=value.replace(/\\D
        +/g, ''))">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="text" name="country"
        placeholder="Country" value="{car.country.name}"
        required="">
    </fieldset>
</div>
<div class="col-md-3 col-sm-12">
    <fieldset>
        <input type="file" name="imageFile"
        value="{car.picture.name}" accept="image/png,
        image/jpeg, image/jpg"/>
    </fieldset>
</div>
<div class="col-lg-12">
    <fieldset>
        <textarea name="description" rows="6"
        placeholder="Description"
        required="">{car.description}</textarea>
    </fieldset>
</div>
<div class="col-md-2">
    <fieldset>
        <button type="submit" id="form-submit"
        class="main-button">Edit</button>
    </fieldset>
</div>
</div>
<input type="hidden" name="_csrf" value="{_csrf.token}" />
</form>
</div>
</div>

</section>
<!-- ***** Add Area Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->

```

```

<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл index.ftlh

```

<!DOCTYPE html>
<html lang="en">
<#include "blocks/security.ftlh">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display
    =swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
    href="../../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"
    href="../../static/css/font-awesome.css">

    <link rel="stylesheet" href="../../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">

```

```

        <span></span>
        <span></span>
        <span></span>
    </div>
</div>
<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Main Banner Area Start ***** -->
<div class="main-banner" id="top">
    <video autoplay muted loop id="bg-video">
        <source src="../../static/images/video.mp4" type="video/mp4" />
    </video>

    <div class="video-overlay header-text">
        <div class="caption">
            <h6>Lorem ipsum dolor sit amet</h6>
            <h2>Best <em>car dealer</em> in town!</h2>
            <div class="main-button">
                <a href="">Contact Us</a>
            </div>
        </div>
    </div>
</div>
<!-- ***** Main Banner Area End ***** -->

<!-- ***** Cars Starts ***** -->
<section class="section" id="trainers">
    <div class="container">
        <div class="row">
            <div class="col-lg-6 offset-lg-3">
                <div class="section-heading">
                    <h2>Featured <em>Cars</em></h2>
                    
                    <p>Nunc urna sem, laoreet ut metus id, aliquet consequat magna. Sed viverra ipsum dolor, ultricies fermentum massa consequat eu.</p>
                </div>
            </div>
        </div>
        <div class="row">
            <#if cars?has_content>
                <#list cars as car>
                    <#if car?index < 3>
                        <div class="col-lg-4">
                            <div class="trainer-item">
                                <div class="image-thumb">
                                    <#if car.picture.name?has_content>
                                        
    </#else>
    
    </#if>

</div>
<div class="down-content">
<span>
    <sup>${car.price}</sup>
</span>

<h4>${car.brand.name} ${car.model}</h4>

<p>
    <i class="fa fa-dashboard"></i>
    ${car.engine.name}
    <i class="fa fa-cog"></i>
    ${car.gearBox.switchingType}
    (${car.gearBox.numberOfGears} speed)
    <i class="fa fa-flag"></i>
    ${car.country.name}
    <i class="fa fa-car"></i>
    ${car.bodyType.name}
</p>

<ul class="social-icons">
    <li><a href="/cars/${car.car_id}">+
    View Car</a></li>
    </#if isAdmin>
    <li><a
        href="/cars/${car.car_id}/edit">+
        Edit Car</a></li>
    <li>
        <a>
            <form action="/cars/
            ${car.car_id}/delete"
            method="post">
                <button class="btn
                btn-primary"
                type="submit">Delete
                Car</button>
                <input type="hidden"
                name="_csrf"
                value="${_csrf.token}"
                />
            </form>
        </a>
    </li>
    </#if>
</ul>
</div>
</div>
</div>
</#if>
</#list>
</#else>

```



```

<section class="section section-bg" id="call-to-action"
style="background-image:
url(../static/images/banner-image-1-1920x500.jpg)">
  <div class="container">
    <div class="cta-content">
      <h2>The search has not given any
      <em>results</em></h2>
      <p>We are definitely expanding our range.</p>
    </div>
  </div>
</section>
</#if>
</div>

<br>

<div class="main-button text-center">
  <a href="cars">View Cars</a>
</div>
</div>
</section>
<!-- ***** Cars Ends ***** -->

<section class="section section-bg" id="schedule" style="background-image:
url(../static/images/about-fullscreen-1-1920x700.jpg)">
  <div class="container">
    <div class="row">
      <div class="col-lg-6 offset-lg-3">
        <div class="section-heading dark-bg">
          <h2>Read <em>About Us</em></h2>
          
          <p>Nunc urna sem, laoreet ut metus id, aliquet consequat
          magna. Sed viverra ipsum dolor, ultricies fermentum massa
          consequat eu.</p>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-lg-12">
        <div class="cta-content text-center">
          <p>Lorem ipsum dolor sit amet, consectetur adipisicing
          elit. Labore deleniti voluptas enim! Provident consectetur
          id earum ducimus facilis, aspernatur hic, alias, harum
          rerum velit voluptas, voluptate enim! Eos, sunt,
          quidem.</p>

          <p>Lorem ipsum dolor sit amet, consectetur adipisicing
          elit. Iusto nulla quo cum officia laboriosam. Amet tempore,
          aliquid quia eius commodi, doloremque omnis delectus
          laudantium dolor reiciendis non nulla! Doloremque maxime
          quo eum in culpa mollitia similique eius doloribus
          voluptatem facilis! Voluptatibus, eligendi, illum.
          Distinctio, non!</p>
        </div>
      </div>
    </div>
  </div>
</section>

```





```

<section class="section" id="features">
  <div class="container">
    <div class="row">
      <div class="col-lg-6 offset-lg-3">
        <div class="section-heading">
          <h2>Read our <em>Testimonials</em></h2>
          
          <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit. Voluptatem incididunt alias minima tenetur nemo
necessitatibus?</p>
        </div>
      </div>
      <div class="col-lg-6">
        <ul class="features-items">
          <li class="feature-item">
            <div class="left-icon">
              
            </div>
            <div class="right-content">
              <h4>John Doe</h4>
              <p><em>"Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Dicta numquam maxime
voluptatibus, impedit sed! Necessitatibus
repellendus sed deleniti id et!"</em></p>
            </div>
          </li>
          <li class="feature-item">
            <div class="left-icon">
              
            </div>
            <div class="right-content">
              <h4>John Doe</h4>
              <p><em>"Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Dicta numquam maxime
voluptatibus, impedit sed! Necessitatibus
repellendus sed deleniti id et!"</em></p>
            </div>
          </li>
        </ul>
      </div>
      <div class="col-lg-6">
        <ul class="features-items">
          <li class="feature-item">
            <div class="left-icon">
              
            </div>
            <div class="right-content">
              <h4>John Doe</h4>
              <p><em>"Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Dicta numquam maxime
voluptatibus, impedit sed! Necessitatibus
repellendus sed deleniti id et!"</em></p>
            </div>
          </li>
          <li class="feature-item">
            <div class="left-icon">

```

```

        
    </div>
    <div class="right-content">
        <h4>John Doe</h4>
        <p><em>"Lorem ipsum dolor sit amet, consectetur
        adipiscing elit. Dicta numquam maxime
        voluptatibus, impedit sed! Necessitatibus
        repellendus sed deleniti id et!"</em></p>
    </div>
</li>
</ul>
</div>
</div>

<br>

<div class="main-button text-center">
    <a href="">Read More</a>
</div>
</div>
</section>
<!-- ***** Testimonials Item End ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл login.ftlh

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display
    =swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
    href="../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"
    href="../static/css/font-awesome.css">

    <link rel="stylesheet" href="../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>

<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->
<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
    <@h.page>
    </@h.page>
</header>

<!-- ***** Header Area End ***** -->

<!-- ***** Contact Us Area Starts ***** -->
<section class="section" id="contact-us" style="margin-top: 0">
    <div class="container-fluid">
        <div class="contact-form section-bg" style="background-image:
        url(../static/images/contact-1-720x480.jpg)">
            <div class="container">
                <div class="row">
                    <div class="col-lg-10 offset-lg-1">
                        <div class="cta-content">
                            <br>

```

```

        <br>
        <h2><em>Login</em> to the site</h2>
        <p>Ut consectetur, metus sit amet aliquet placerat,
        enim est ultricies ligula</p>
    </div>
</div>
</div>
</div>
<form id="contact" action="/login" method="post">
    <div class="row">
        <div class="col-md-6 col-sm-12">
            <fieldset>
                <input name="customer_name" type="text"
                placeholder="Your Name*" required="">
            </fieldset>
        </div>
        <div class="col-md-6 col-sm-12">
            <fieldset>
                <input name="customer_password" type="password"
                placeholder="Your Password*" required="">
            </fieldset>
        </div>
        <div class="col-md-2">
            <fieldset>
                <button type="submit" id="form-submit"
                class="main-button">Sign in</button>
            </fieldset>
        </div>
        <div class="col-md-7">
            <p>
                Don't have an account? <a
                href="registration">Register now</a>
            </p>
        </div>
        <input type="hidden" name="_csrf" value="{_csrf.token}" />
    </div>
</form>
</div>
</div>
</section>
<!-- ***** Contact Us Area Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>

```

```

<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---

## Файл registration.ftlh

```

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,2
    00i,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,900,900i&display
    =swap" rel="stylesheet">

    <title>PHPJabbers.com | Free Car Dealer Website Template</title>

    <link rel="stylesheet" type="text/css"
    href="../../static/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css"
    href="../../static/css/font-awesome.css">

    <link rel="stylesheet" href="../../static/css/style.css">

</head>

<body>

<!-- ***** Preloader Start ***** -->
<div id="js-preloader" class="js-preloader">
    <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
            <span></span>
            <span></span>
            <span></span>
        </div>
    </div>
</div>

<!-- ***** Preloader End ***** -->

<!-- ***** Header Area Start ***** -->

```



```

<#import "blocks/header.ftlh" as h>

<header class="header-area header-sticky">
  <@h.page>
</@h.page>
</header>
<!-- ***** Header Area End ***** -->

<!-- ***** Contact Us Area Starts ***** -->
<section class="section" id="contact-us" style="margin-top: 0">
  <div class="container-fluid">
    <div class="contact-form section-bg" style="background-image:
url(../static/images/contact-1-720x480.jpg)">
      <div class="container">
        <div class="row">
          <div class="col-lg-10 offset-lg-1">
            <div class="cta-content">
              <br>
              <br>
              <h2><em>Registration</em> to the site</h2>
              <p>Ut consectetur, metus sit amet aliquet placerat,
enim est ultricies ligula</p>
            </div>
          </div>
        </div>
      </div>

      <form id="contact" action="/registration" method="post">
        <div class="row">
          <div class="col-md-12 col-sm-12">
            <fieldset>
              <input name="username" type="text"
placeholder="Your Name*" required="">
            </fieldset>
          </div>
          <div class="col-md-6 col-sm-12">
            <fieldset>
              <input name="email" type="text" id="email"
pattern="^[^\s@]+@[^\s@.]+\.[^\s@.]{2,}$"
placeholder="Your Email*" required="">
            </fieldset>
          </div>
          <div class="col-md-6 col-sm-12">
            <fieldset>
              <input name="customer_password" type="password"
placeholder="Your Password*" required="">
            </fieldset>
          </div>
          <div class="col-md-2">
            <fieldset>
              <button type="submit" id="form-submit"
class="main-button">Register</button>
            </fieldset>
          </div>
          <div class="col-md-7">
            <p>
              Already have an account? <a href="login">Sign
in</a>
            </p>
          </div>
        </div>
      </form>
    </div>
  </div>

```

```

        </div>
    </div>
    <input type="hidden" name="_csrf" value="{_csrf.token}" />
</form>
</div>
</div>
</section>
<!-- ***** Contact Us Area Ends ***** -->

<!-- ***** Footer Start ***** -->
<#import "blocks/footer.ftlh" as f>
<footer>
    <@f.page></@f.page>
</footer>
<!-- ***** Footer End ***** -->

<!-- jQuery -->
<script src="../../static/js/jquery-2.1.0.min.js"></script>

<!-- Bootstrap -->
<script src="../../static/js/popper.js"></script>
<script src="../../static/js/bootstrap.min.js"></script>

<!-- Plugins -->
<script src="../../static/js/scrollreveal.min.js"></script>
<script src="../../static/js/waypoints.min.js"></script>
<script src="../../static/js/jquery.counterup.min.js"></script>
<script src="../../static/js/imgfix.min.js"></script>
<script src="../../static/js/mixitup.js"></script>
<script src="../../static/js/accordions.js"></script>

<!-- Global Init -->
<script src="../../static/js/custom.js"></script>

</body>
</html>

```

---