

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра математического и компьютерного моделирования

РАБОТА ПРОВЕРЕНА

Рецензент,

_____/ А.А. Панюкова

« ____ » _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой МиКМ,

д-р физ.-мат. наук, доцент

_____/ С.А. Загребина

« ____ » _____ 2019 г.

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ
ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ–01.03.02.2019. 048.ВКР

Руководитель работы, доцент каф.
МиКМ, к.т.н.,

_____/ В.И. Дударева

« ____ » _____ 2019 г.

Автор работы,

студент группы ЕТ-416

_____/ Д.С. Юсупов

« ____ » _____ 2019 г.

Нормоконтролер,

Доцент каф. МиКМ, к.ф.-м.н.,

_____/ Т.А. Макаровских

« ____ » _____ 2019 г.

Челябинск 2019

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра математического и компьютерного моделирования

УТВЕРЖДАЮ

Заведующий кафедрой, д.ф.-м.н.
доцент

Загребина С.А.

2019 г.

ЗАДАНИЕ

на выпускную квалификационную работу студента

Юсупова Дамиля Сергеевича

Группа ЕТ-416

1 Тема работы

Разработка веб-приложения

Утверждена приказом по университету от «04» апреля 2019 г.
№580.

2 Срок сдачи студентами законченной работы

«25» июня 2019 г.

3 Исходные данные к работе

4 Перечень вопросов, подлежащих разработке:

4.1. Описать математическую модель расчета рекомендованного количества автомобилей в системе каршеринга;

4.2. Сформулировать требования к сайту;

4.3. Рассмотреть инструменты создания сайта для проверки модели;

4.4. Спроектировать веб-сайт;

4.5. Проанализировать полученные результаты и проверить адекватности модели.

5 ГРАФИЧЕСКИЙ МАТЕРИАЛ

- 5.1. Оглавление презентации
- 5.2. Титульный слайд (1 слайд)
- 5.3. Цели, задачи, актуальность (3 слайда)
- 5.4. Обзор методов (2 слайдов)
- 5.5. Представление решения задачи (5 слайда)
- 5.6. Выводы (1 слайд)

6 КАЛЕНДАРНЫЙ ПЛАН

Наименование этапов дипломной работы	Срок выполнения этапов работы	Отметка о выполнении
1. Изучение литературы по тематике выпускной квалификационной работы	11.02.2019-14.03.2019	
2. Построение математической модели	16.03.2019-10.04.2019	
3. Создание макета сайта	11.04.2019-12.04.2019	
4. Программная реализация разработанной модели	12.04.2019-05.05.2019	
5. Проверка модели	06.05.2019-13.05.2019	
6. Разработка структуры выпускной квалификационной работы	14.05.2019-20.05.2019	
7. Подготовка текста выпускной квалификационной работы	20.05.2019-28.05.2019	
8. Подготовка графического материала и доклада	29.05.2019-04.06.2019	
9. Нормоконтроль		
10. Рецензирование, представление зав. кафедрой		
11. Подготовка иллюстративного материала и доклада		

7 Дата выдачи задания « ___ » _____ 2018 г.

Руководитель работы _____ /В.И. Дударева/

Задание принял к исполнению _____ /Д.С. Юсупов/

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра «Математическое и компьютерное моделирование»

АННОТАЦИЯ

Юсупов, Д.С. Создание веб-приложения для расчета рекомендованного количества автомобилей в сервисе каршеринга/ Д.С. Юсупов. – Челябинск: ЮУрГУ, ЕТ-416, 2019. – 36 с., 14 ил., библиогр. список – 10 наим., 3 прил.

В выпускной квалифицированной работе рассмотрена задача расчета рекомендованного количества автомобилей в системе каршеринг. В ходе выполнения работы построена математическая модель расчета рекомендованного количества автомобилей в системе каршеринг и разработан веб-сайт с возможностью проводить данный расчет.

Практическая значимость разработанной модели состоит в возможности использования данного сайта при необходимости рассчитать оптимальное количество автомобилей в системе каршеринг по заданным параметрам.

Оглавление

Введение.....	7
1 Задача нахождения рекомендованного числа автомобилей в сервисе каршеринг	10
2.1 Требования к системе.....	12
2.2 Сравнение с аналогами	13
Вывод по главе один.....	16
2 Проектирование сайта	17
2.1 Математическая модель расчета рекомендованного количества автомобилей в системе каршеринга.....	17
2.2 Адекватность модели расчета рекомендованного количества автомобилей.....	18
2.3 Проектирование	19
2.4 Требования к веб-приложению.....	19
2.5 Интерфейс	20
2.6 Дизайн.....	20
Вывод по главе два.....	21
3 Реализация	22
2.1 Выбор средств реализации	22
2.2 Хостинг	25
2.3 Перенос проекта на хостинг	25
2.4 Создание базы данных	26
2.5 Настройка проекта.....	27
2.7 Подготовка базы данных	28
2.8 Настройка веб-сервера PythonAnywhere.....	28
2.9 Авторизация.....	29
2.10 Доступ к панели администратора.....	33
Заключение	34
Библиографический список	35

Введение

Актуальность темы

Рост городов приводит к удорожанию каждого участка земли, что повышает стоимость владения собственным автомобилем, который необходимо парковать, обслуживать и заправлять, но отказываться от мобильности, которую предоставляет автомобиль, не лучшее решение проблемы, так каршеринг повышает мобильность города и экономит деньги его пользователям. Понимая, что покупка и владение личным транспортным средством может быть излишне дорогой, люди начинают обращаться к более дешевым и более распределенным средствам оплаты за перевозку частного транспортного средства.

Каршеринг — вид пользования автомобилем, когда одна из сторон не является его собственником. Это вариант аренды автомобиля у профильных компаний (чаще всего для внутригородских и/или коротких поездок) или частных лиц (на любой срок и расстояние поездки — по договоренности). Такая модель аренды автомобилей удобна для редкого пользования автомобилем или в случае, когда необходим автомобиль, другой марки, типа кузова или грузоподъемности. Каршеринг является одним из глобальных трендов развития экономики совместного пользования (Sharing Economy), когда население отказывается от приобретения благ в собственность, по причине ограничения ответственности и уменьшения затрат, но имеет те же возможности, что и автовладельцы. Услуги каршеринга доступны в более чем 1000 городах в десятках стран мира.

На данный момент каршеринг продолжает набирать популярность в России, не смотря на сложность запуска и содержания появляются все новые сервисы, крупные компании, такие как Яндекс инвестируют в собственные аналоги сервиса. С начала года пользователи сервиса краткосрочной аренды автомобилей в Москве совершили более 12 млн поездок, что более чем в два раза превышает показатель за весь 2017 год (6,5 млн поездок). В столице работает

15 компаний краткосрочной аренды, чей автопарк в сумме составляет около 11,5 тыс. машин, а количество поездок превышает 60 тыс. в день.

При подобном росте возникает нужда в инструментах масштабирования бизнеса и данное веб-приложение аналогов в доступных мне источниках я не нашел позволит рассчитать рекомендованное количество автомобилей в сервисе каршеринга.

Цель работы разработать веб-приложение для расчета рекомендованного количества автомобилей в сервисе каршеринга.

Задачи:

- составить математическую модель расчета рекомендованного количества автомобилей в системе каршеринг;
- проанализировать требования к веб-приложению;
- разработать структуру и дизайн веб-приложения;
- выполнить реализацию разработанных макетов;
- проверить адекватность расчётов;
- провести тестирование пользовательского интерфейса.

Объект исследования – сервис каршеринга.

Предмет исследования – количество автомобилей в автопарке сервиса каршеринга.

В первой главе рассмотрена проблема определения рекомендованного количества автомобилей в сервисе каршеринг, были приведены требования к системе и описаны инструменты для разработки соответствующего сайта.

Во второй главе описана математическая модель расчета рекомендованного количества автомобилей в сервисе каршеринга и был составлен план проектирования сайта. Обозначены требования к дизайну и интерфейсу сайта, обозначены требования к реализации проекта.

В третьей главе представлен процесс реализации проекта и переноса его на хостинг и обоснование использования инструментов.

В приложениях представлены листинги программного кода, техническое задание и руководство пользователя.

1 Задача нахождения рекомендованного числа автомобилей в сервисе каршеринг

Каршеринг - это модель проката автомобилей, где люди арендуют автомобили на короткие промежутки времени, часто по часам. Он отличается от традиционного проката автомобилей тем, что владельцы автомобилей часто сами являются частными лицами, а посредник по обмену автомобилями обычно отличается от владельца автомобиля. Carsharing является частью более широкой тенденции общей мобильности.

Демографические изменения, урбанизация и экономическое давление создало благодатную почву для роста каршеринга во всем мире. Миллениалы (Люди в возрасте от 15 до 34 лет) меньше ездят, живут в более урбанизированных местах и часто сталкиваются с нехваткой денег, несмотря на наличие работы. Каршеринг решает эту проблему, предлагая населению удобный и недорогой доступ к транспортному средству в любое время. В результате доля каршеринга значительно выросла.

В одной только Северной Америке насчитывалось почти 2 миллиона участников сервиса и почти 25000 транспортных средств по состоянию на январь 2017 года, согласно ежегодным прогнозам Сьюзен Шахин. Ризорсинговые услуги, такие как Lyft и Uber, также растут быстрыми темпами.

Отраслевые исследования свидетельствуют о сохранении высоких темпов роста в этой категории. Frost and Sullivan, консалтинговая компания по автомобильным и транспортным вопросам, ожидает, что мировой рынок автомобильных перевозок вырастет с 7 миллионов в 2015 году до 36 миллионов к 2025 году, и что число доступных автомобилей для обмена будет увеличено почти в 4 раза за тот же период времени. Global Market Insights Inc., глобальная группа по исследованию и анализу рынка, ожидает, что «рынок каршеринга планирует достичь рекордного двузначного роста по сравнению с аналогичным периодом прошлого года (34,8%) при прогнозируемом доходе более 16,5 млрд долларов США. к 2024 году».

Не смотря на позднее вступление России на рынок каршеринга, он показывает хорошие результаты, появляются новые компании предоставляющие услуги каршеринга, крупные компании создают свои бренды каршеринга, власти столицы продвигают идею каршеринга, что говорит о том, что рынок продолжит расти и необходимость в инструментах, упрощающих работу подобных сервисов будет расти.

В данный момент на рынке каршеринга работают такие компании, как Яндекс.Драйв, Делимобиль, BelkaCar, YouDrive, Anytime, МатрёшCar. У каждой компании своя ЦА, район и тарифы. Ниже приведена сравнительная таблица компаний, предоставляющих услуги каршеринга в Москве.

Таблица 1.1 – Сравнительная таблица сервисов каршеринга в Москве

Оператор	Минимальный тариф	Авто-парк	Модельный ряд
Яндекс. Драйв	5 руб./мин	6500	Kia Rio(включая X-Line), Renault Kaptur, Audi А3и Q3, BMW 520i, Genesis G70, Mercedes E200, Nissan Qashqai, Porsche 911 Carrera 4S и Macan, Range Rover Velar, Škoda Octavia Rapid, Volkswagen Polo, Citroën Jumpy, Ford Transit, VW Transporter
Делимобиль	3 руб./мин	4000	Hyundai Solaris, Renault Kaptur, Volkswagen Polo
BelkaCar	6 руб./мин	2775	Kia Rio (включая X-Line), Mercedes-Benz CLA и GLA
YouDrive	8 руб./мин	1200	BMW i3, BMW 2 и X2, Mini Cooper, Mercedes-Benz A, Nissan X-Trail, Smart Fortwo и Forfour
Anytime	7 руб./мин	1100	BMW 320i, Audi A3 и Q3, Nissan Qashqai, Renault Kaptur, Kia Rio, Škoda Octavia, Mercedes G500, Ferrari F430 Spider
МатрёшCar	9 руб./мин	800	BMW 320i, Kia Stinger, Mercedes Benz C180, Mazda 3, Land Rover Discovery Sport, Smart ForTwo и ForFour, Mini Countryman, Jaguar XE

Ручные расчеты рекомендованного количества автомобилей для системы каршеринг требуют больших ресурсов и остается место для ошибки.

Проблемой подобных расчетов является наличие большого количества факторов и отсутствие очевидной модели расчета. Тем ни менее, аналогов данного решения мною найдено не было, чаще всего при описании сервиса указывают приблизительное количество автомобилей для конкретных городов так, консультант по каршерингу, пишет о том, что стартап должен начинать с 10 автомобилей и прибавлять по 2 автомобиля в месяц при росте базы клиентов в 40 человек в месяц [4].

Компании не могут составить точную модель масштабирования своего бизнеса из-за отсутствия точной модели описывающей необходимость в автомобиле в описанном количестве. На данный момент для подобных расчетов пользуются MS Excel, что не всегда удобно и точно, так как остается большое пространство для человеческой ошибки.

Практическая значимость данного сайта может проявиться только при условии интеграции в веб-систему управления сервисом каршеринга. Тогда возможно использования этих данных для планирования дальнейших шагов в развитии бизнеса, записи и управлении данными полученными при помощи этого сайта.

2.1 Требования к системе

Веб-приложение реализует функционал расчета рекомендованного количества автомобилей в сервисе каршеринг, по следующим параметрам:

1. Стоимость одного автомобиля (в рублях), S_0 .
2. Количество предзаказов автомобилей (в штуках), n .
3. Количество дней использования машины (в днях), d .
4. Стоимость, которую клиент готов заплатить единовременно за срок выпуска авто (в рублях), S_{d_r} .
5. Срок выпуска авто (в днях), d_r .
6. Стоимость тех. Обслуживания (в рублях), S_{TO} .
7. Стоимость топлива (в рублях), S_f .
8. Расход топлива (в литрах на 100 километров), C_f .

Реализует следующий функционал:

1. Расчет рекомендованного количества автомобилей в сервисе каршеринг.
2. Авторизация.
3. Сохранение запросов в базу данных.

Для реализации данного функционала используются следующие средства:

1. Язык для написания функционала Python, имеет огромное количество библиотек и инструментов для работы с данными и веб-приложениями.
2. Django - фреймворк на базе python, имеет встроенные средства и решения, такие как CMS и SQL, позволяет легко экспортировать разработанное приложение в готовые проекты построенные на этом фреймворке, так же, на базе этого приложения можно выстроить систему для работы и функционирования сервиса каршеринга.
3. MySQL – База для хранения записей и пользователей.
4. Бесплатный хостинг pythonanywhere.
5. Bootstrap - свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

2.2 Сравнение с аналогами

Так как прямых аналогов данной системы нет, в работе рассмотрено решения той же тематики, рассмотрим некоторые из них.

Enterprise CarShare – калькулятор созданный с целью продвижения услуги каршеринга от компании «enterprise», считает годовые траты владения автомобилем, как для пользователя так и для бизнеса [9].

Решаемые задачи:

- расчет стоимости владения автомобилем;
- расчет стоимости владения парком автомобилей;

- продвижение сервиса каршеринга.

Calculator

Estimate Your Savings With Enterprise CarShare

Leave car costs in the dust. We offer low hourly rates that will free you from all the heavy costs of car ownership. Let us show you how much you can save.

The calculator interface is divided into two tabs: 'Personal Use' and 'Business Use'. The 'Business Use' tab is active. It lists five vehicle categories: Compact, Standard, Full Size, SUV, and Mini Van. For each category, there is an input field for the number of cars, a slider for miles (0 to 20000), and a result of '0 mi'. A green button labeled 'CALCULATE COSTS' is located at the bottom right of the form.

Рисунок 1.1 – Enterprise CarShare калькулятор

WeeShare Carsharing calculator – калькулятор так же продвигающий услуги компании по каршерингу, данный калькулятор рассчитывает во сколько вам обойдутся услуги каршеринга в год [8].

Решаемые задачи

- расчет ежегодных затрат пользования услугами каршеринга;
- продвижение сервиса.

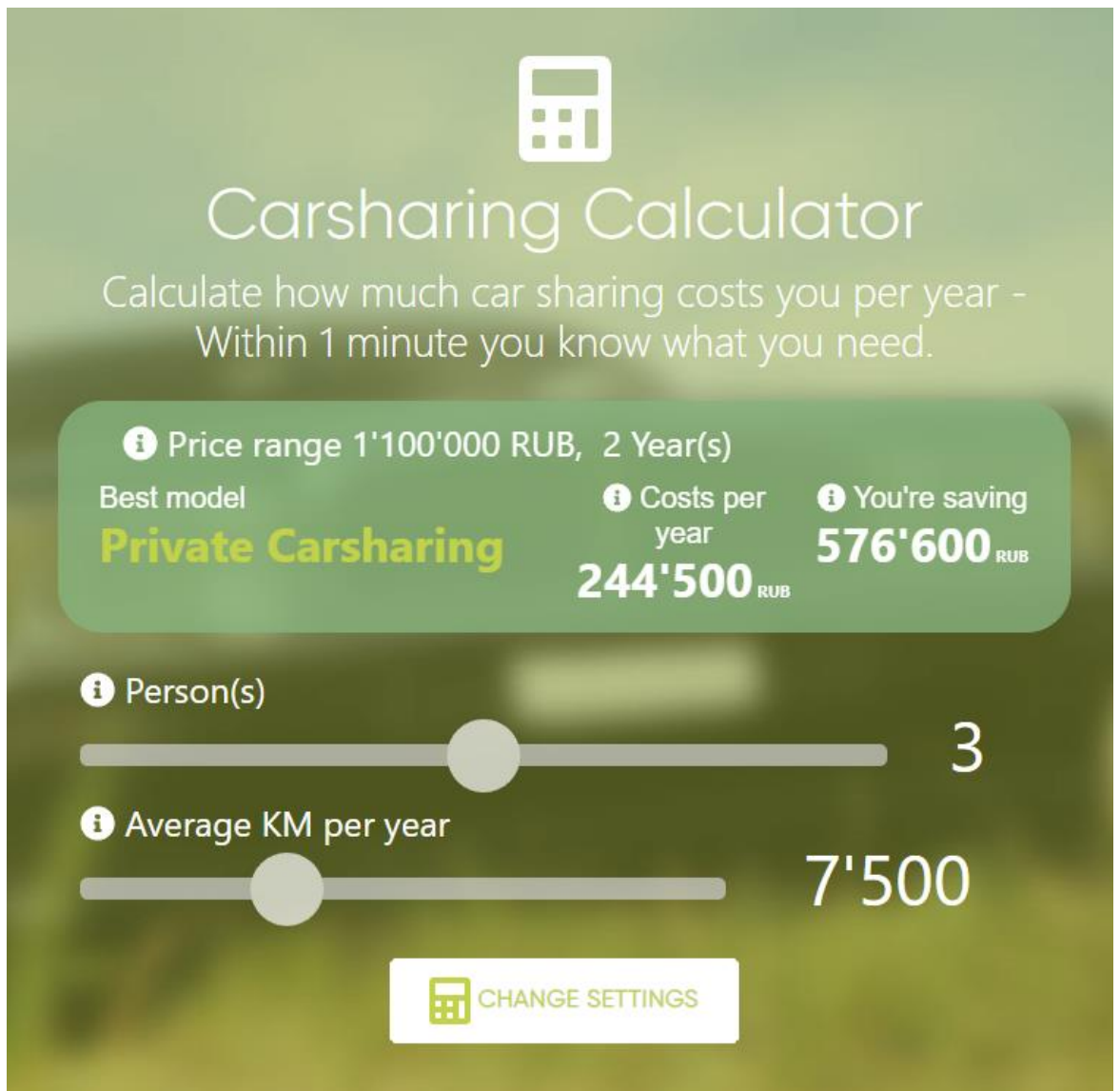


Рисунок 1.2 – WeeShare Carsharing калькулятор

Сервисы очень похожи по своим задачам и функционалу, а также дизайну, показывают выгоду каршеринга, от затрат в деньгах до помощи окружающей среде. Оба используют ползунки и выбор параметров довольно ограничен, что является минусом. В отличии от данных сервисов разрабатываемый сайт предназначен для работников этой среды. Более гибкие параметры и больший объём вводимых данных.

Инструменты разработки так же разные в обоих случаях основным языком был выбран Java как более популярный и подходящий для интеграции на сайт компании.

Вывод по главе один

Аналогов разработанного приложения в настоящий момент не найдено. Калькуляторы, разработанные Enterprise CarShare и WeeShare позволяют оценить выгоду от использования сервиса каршеринга, тем не менее, они предназначены только для потребителя.

Разработанное веб-приложение предполагается ориентировать на сотрудников каршеринговой компании. Следовательно, такая разработка является новой и востребованной.

2 Проектирование сайта

2.1 Математическая модель расчета рекомендованного количества автомобилей в системе каршеринга

Для решения поставленной задачи построим математическую модель расчета. Необходимо составить формулу по данным параметрам которая показывала бы адекватный результат расчета.

N – оптимальное количество авто,

S – стоимость

P – прибыль

Параметры:

1. Стоимость одного автомобиля (в рублях), S_0 .
2. Количество предзаказов автомобилей (в штуках), n .
3. Количество дней использования машины (в днях), d .
4. Стоимость, которую клиент готов заплатить единовременно за срок выпуска авто (в рублях), S_{dr} .
5. Срок выпуска авто (в днях), d_r .
6. Стоимость тех. Обслуживания (в рублях), S_{TO} .
7. Стоимость топлива (в рублях), S_f .
8. Расход топлива (в литрах на 100 километров), C_f .
9. Время подготовки между двумя бронированиями (в минутах) t .
10. Стоимость подготовки (в рублях), S_t .
11. Ореол подачи (в км) O .

Соотношения:

- 1) Затраты на топливо одного автомобиля на одно бронирование $d \cdot (S_f \cdot C_f)$;
- 2) Коэффициент рентабельности $\frac{S_{dr}}{S_0}$.
- 3) Затраты на то в день $\frac{S_{TO}}{365}$.

- 4) Ежедневная прибыль за срок выпуска авто $\frac{S_{dr}}{d_r}$.
- 5) Стоимость минуты подготовки $\frac{t}{S_t}$.

Основная формула расчета:

$$N = n \cdot \frac{S_{dr}}{S_0} \cdot \frac{\frac{S_{dr}}{d_r} \cdot O \cdot \frac{t}{S_t}}{\frac{STO}{365} + d \cdot (S_f \cdot C_f)}$$

2.2 Адекватность модели расчета рекомендованного количества автомобилей

Оценка адекватности модели необходима для проверки соответствия реальной системе

Оценка адекватности основана на сравнении свойств и целей реальной системы и проверяемой и может проводится разными методами:

- по средним значениям откликов модели и системы;
- по дисперсиям отклонений откликов модели от среднего значения откликов системы;
- по максимальному значению относительных отклонений откликов модели от откликов системы.

В случае с моделью расчета рекомендованного количества автомобилей в сервисе каршеринга, достаточно сравнить результат расчета веб-приложения и реально существующий проект. Сравнить по одинаковым параметрам один и тот же автомобиль.

Таблица 2.1 Оценка адекватности модели расчета рекомендованного количества автомобилей в сервисе каршеринга в сравнении с данными BelkaCar

Автомобиль	Цена руб.	Belkacar	Рассчитанное значение
Kia Rio X-line	914900	500	478
Mercedes-Benz GLA 200 OS	2170000	250	227

Окончание таблицы 2.1

Mercedes-Benz CLA sport	2900000	300	225
Автомобиль	Цена	Yandex.drive	Рассчитанное значение
Kia Rio	864900	750	498
Kia Rio X-Line	914900		478
Renault Kaptur	994990		264

Модель показывает результат, укладывающийся в разумные рамки.

2.3 Проектирование

Постановка задачи – это начало разработки веб-приложения. Постановка задачи может быть оформлена в виде ТЗ. Техническое задание — документ, описывающий требования к разрабатываемому объекту, определяющий условия, сроки и порядок ее проведения для обеспечения нужд, в соответствии с которым осуществляются выполнение работ, оказание услуг и их приемка. Это исходный документ, который учитывает основное назначение закупки товаров, работ, услуг, их характеристики, задание заказчика, описание первичных данных, целей и задач закупки, сроков поставки, выполнения работ, оказания услуг, требований к товару, работам, услугам, их результатам, к гарантиям, описание объекта закупки, объем закупаемых товаров, работ, услуг, формы отчетности, обоснование требований к товару, работам, услугам, эквивалентные показатели, экономические требования, а также специальные требования

2.4 Требования к веб-приложению

1. Задача разработчика: создание калькулятора рекомендованного количества автомобилей в сервисе каршеринга.
2. Язык: русский.
3. Аналогичные платформы: не найдены.

4. Компоненты сайта: окно авторизации, формы для ввода данных, кнопка, форма вывода информации.

5. Цветовой набор: на усмотрение разработчика.

6. Средства разработки: Python, Django, Bootstrap, MySQL.

2.5 Интерфейс

Интерфейс данного сайта содержит 11 форм заполнения, для ввода параметров расчета, кнопку инициирующую расчет и поле вывода результата.

В ходе работы над сайтом мной был разработан шаблон с использованием таких инструментов как (CSS (Каскадные таблицы стилей)) Bootstrap, HTML (HyperText Markup Language) – язык разметки текста шаблон сайта.

2.6 Дизайн

Web-дизайн – вид дизайна, удовлетворяющий потребность в проектировке веб-сайтов с использованием таких понятий как UI/UX (User interface / User experience) позволяющий создавать максимально удобный в использовании и эстетически приятный веб-сайт, основные функции веб-дизайна:

- грамотную структуру сайта;
- удобная навигация;
- красивый и хорошо подсознательно-воспринимаемый дизайн;
- правильно поданный и удобно расположенный текст.

Опираясь на вышеперечисленные свойства, мной было решено использовать следующий дизайн:

Цветовая гамма страниц состоит из четырех цветов:

- 1) Второстепенный шрифт в пустой форме: Серый (#6c757d).
- 2) Фон: белый (ffffff).
- 3) Кнопка: синий (#007bff).
- 4) Основной шрифт: черный (#000000).

Было использовано два шрифта:

- Segoe UI и Helvetica Neue.

Количество предзаказов данного автомобиля	<input type="text" value="штук"/>	Стоимость, которую клиент готов заплатить одновременно за срок выпуска авто	<input type="text" value="руб."/>
Стоимость одного автомобиля данной модели	<input type="text" value="руб."/>	Срок выпуска данного авто	<input type="text" value="дней"/>
Стоимость тех.обслуживания данного автомобиля	<input type="text" value="руб."/>	Количество дней использования выбранной машины	<input type="text" value="дней"/>
Стоимость топлива	<input type="text" value="руб."/>	Расход топлива автомобиля	<input type="text" value="литров на 100 км"/>
Ореол подачи автомобиля	<input type="text" value="км"/>	Время подготовки между двумя бронированиями	<input type="text" value="минут"/>
Стоимость подготовки одного автомобиля	<input type="text" value="руб."/>	<input type="button" value="Готово"/>	Рекомендованное количество автомобилей для заданных параметров: 4

Рисунок 2.1 – Интерфейс сайта

Вывод по главе два

После составления основных требований к веб-приложению, можно приступить непосредственно к его разработке, так как данный сайт для расчета рекомендованного количества автомобилей в сервисе каршеринга, предназначен для сотрудников сервиса каршеринга, то основными требованиями являются простота и удобный и интуитивно понятный интерфейс и читаемость текста.

Оценка адекватности расчетов показывает, что модель расчета показывает результат близкий к сервису каршеринга VelkaCar в бюджетном сегменте автомобилей.

3 Реализация

На данном этапе были рассмотрены средства реализации, обоснование их выбора и этапы реализации веб-приложения.

2.1 Выбор средств реализации

Для визуализации работы приложения использовались шаблоны, написанные на HTML5, применялись таблицы стилей CSS библиотеки Bootstrap, функционал приложения был написан на Python с использованием фреймворка Django рассмотрим каждый из элементов более подробно.

HTML (Hyper Text Markup Language) – это язык для описания веб-страниц (дословно расшифровывается как «язык гипертекстовой разметки»). Под языком разметки понимается набор тегов, каждый из которых описывает содержимое документа. Язык HTML интерпретируется браузерами и полученный в результате интерпретации форматированный текст отображается в понятном для человека виде. Браузер получает HTML документы из веб-сервера или памяти компьютера и преобразует их в веб-страницы. HTML описывает структуру веб-страницы семантически и изначально вложенными сигналами для отображения документа.

Элементы HTML строят блоки HTML страниц. С HTML структурой, изображения и другие объекты могут быть включены в преобразованные страницы. HTML предоставляет средства для создания структурированных документов путем определения структурной семантики для текста, такого как заголовки, абзацы, списки, ссылки, цитаты и другие элементы. HTML-элементы обозначены тегами, написанными с использованием угловых скобок.

HTML может встраивать программы, написанные на языке сценариев, например, JavaScript, что влияет на поведение и содержание веб-страниц. Включение CSS определяет внешний вид и расположение контента.

CSS (от англ. Cascading Style Sheets – каскадные таблицы стилей) – это набор параметров форматирования, который применяется к элементам веб-страницы для управления их видом и положением. Таблицы стилей позволяют

изменить внешний вид и расположение всех страниц веб-сайта путем редактирования всего лишь одного файла.

CSS - это язык таблиц стилей, используемый для описания представления документа, написанного на языке разметки, например, HTML. CSS является краеугольной технологией Всемирной паутины, наряду с HTML и JavaScript.

CSS предназначен для разделения представления и содержимого, включая макет, цвета и шрифты. Такое разделение может улучшить доступность контента, обеспечить большую гибкость и контроль в спецификации характеристик презентации, дать возможность нескольким веб-страницам совместно использовать форматирование, указав соответствующий CSS в отдельном файле .css, а также уменьшить сложность и повторяемость структурного контента.

Разделение форматирования и содержимого также позволяет представить одну и ту же страницу разметки в разных стилях для разных методов рендеринга, например, на экране, в печати, голосом (через речевой браузер или программу чтения с экрана) и на основе шрифта Брайля. тактильные устройства. CSS также имеет правила для альтернативного форматирования, если доступ к контенту осуществляется на мобильном устройстве.

Каскадирование имен происходит из указанной схемы приоритетов, чтобы определить, какое правило стиля применяется, если конкретному элементу соответствует несколько правил. Эта каскадная схема приоритетов предсказуема. В случае рассмотренного в данной работе веб-приложения были использованы библиотеки Bootstrap.

Python - это интерпретируемый высокоуровневый язык программирования общего назначения. Создан Гвидо ван Россумом и впервые выпущен в 1991 году, философия дизайна Python подчеркивает удобочитаемость кода с его заметным использованием значительных пробелов. Его языковые конструкции и объектно-ориентированный подход нацелены на то, чтобы помочь

программистам писать понятный, логичный код для малых и крупных проектов.

Python динамически набирается и собирает мусор. Он поддерживает несколько парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Python часто описывается как язык «с батарейками» из-за его обширной стандартной библиотеки.

Python был задуман в конце 1980-х годов как преемник языка ABC. В Python 2.0, выпущенном в 2000 году, появились такие функции, как списки и система сбора мусора, способные собирать циклы ссылок. Python 3.0, выпущенный в 2008 году, был основной ревизией языка, который не является полностью обратно совместимым, и большая часть кода Python 2 не выполняется неизменным на Python 3. Из-за беспокойства по поводу объема кода, написанного для Python 2, поддержка Python 2.7 (последний выпуск серии 2.x) был продлен до 2020 года. Разработчик языка Гвидо ван Россум взял на себя исключительную ответственность за проект до июля 2018 года, но теперь разделяет свое лидерство в качестве члена руководящего совета из пяти человек.

Интерпретаторы Python доступны для многих операционных систем. Глобальное сообщество программистов разрабатывает и поддерживает CPython, эталонную реализацию с открытым исходным кодом. Некоммерческая организация Python Software Foundation управляет и направляет ресурсы для разработки на Python и CPython.

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY

(англ. Don't repeat yourself). Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений. Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

2.2 Хостинг

Хостинг – это сервис по предоставлению инструмента для размещения веб-сайта в интернете и позволяет постоянно работать этому сайту, обычно в услуги хостинга входит пространство для хранения файлов сайта, предустановленное программное обеспечение, место для почтового сервера, DNS и т.д.

Для работы веб-приложения рассмотренного в данной работе понадобится поддержка Python, Django

В качестве хостинга был использован бесплатный вариант Pythonanywhere рассчитанный на работу с теми же инструментами, которые использовались в разработке приложения на локальном уровне. Далее будет описан процесс развертывания сайта на хостинге pythonanywhere:

2.3 Перенос проекта на хостинг

Для начала нам нужно зайти в консоль из панели управления pythonanywhere и создать виртуальную среду (virtualenv)

```
virtualenv -p /usr/bin/python3.7 simple_env
cd simple_env
```

Рисунок 3.2 - Создание виртуальной среды

Где simple название проекта, с этого момента мы находимся в виртуальном окружении python и можем устанавливать необходимые для проекта библиотеки. Такие как Django и Pillow.

```
pip install Pillow
pip install Django
```

Рисунок 3.3 - Установка библиотек

Далее мы загружаем наш проект в виде архива в папке виртуальной среды и распаковываем его и удаляем файл базы db.sqlite3.

Перейдем в распакованный каталог проекта с помощью команды `cd` (в нём должен быть файл `manage.py`, проверить список файлов в каталоге можно командой `ls`).

```
unzip my_project.zip
```

Рисунок 3.3 – Распаковка архива с проектом

2.4 Создание базы данных

База данных представляет собой организованный сбор данных, обычно хранящихся и доступных в электронном виде из компьютерной системы. Там, где базы данных более сложны, они часто разрабатываются с использованием формальных методов проектирования и моделирования.

Система управления базами данных (СУБД) - это программное обеспечение, которое взаимодействует с конечными пользователями, приложениями и самой базой данных для сбора и анализа данных. Программное обеспечение СУБД дополнительно включает в себя основные средства, предоставляемые для администрирования базы данных. Общая сумма базы данных, СУБД и связанных приложений может называться «системой базы данных». Часто термин «база данных» также используется для обозначения любой СУБД, системы баз данных или приложения, связанного с базой данных.

Перейдем в раздел Databases. Слева выберите MySQL. Найдём на странице секцию Create a database. В поле Database name введём название нашей новой базы данных. Например, `my_project`. Нажмём кнопку Create. Полное название нашей базы данных будет состоять из нашего логина на PythonAnywhere и указанного нами названия, разделённых символом `$`.

Теперь нужно создать пароль для доступа к базе данных. В секции MySQL Password ниже введём пароль в поля New password и Confirm password, после чего щёлкнём кнопку Set MySQL Password ниже. Желательно, чтобы пароль базы данных отличался от вашего пароля на PythonAnywhere. Запомните

или запишите созданный пароль. Его так же надо будет указать в настройках проекта.

2.5 Настройка проекта

На странице PythonAnywhere нужно перейти в раздел Files, и открыть каталог `simple_env`, в нём — каталог `simple`, а в нём — каталог `web_calc`. Вы увидите ваш файл `settings.py`. Щёлкните по нему. Откроется текстовый редактор с этим файлом.

Внесем в него следующие изменения:

Изменим значение параметра `DEBUG` на `False`. В список `ALLOWED_HOSTS` добавим полное доменное имя нашего сайта. Оно состоит из вашего логина на PythonAnywhere и строчки «`pythonanywhere.com`». В нашем случае это будет `damilyusupov.pythonanywhere.com`. Если этого не сделать, Django не будет работать с флагом `DEBUG = False`. Вообще, этот список содержит список доменных имён, по которым должен быть доступен наш сайт.

```
ALLOWED_HOSTS = ["damilyusupov.pythonanywhere.com", ]
```

Рисунок 3.4 – Список хостов

Укажем параметры подключения к созданной нами базе данных. Их можно посмотреть в разделе Databases сайта PythonAnywhere.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'damilyusupov$my_project',  
        'USER': 'damilyusupov',  
        'PASSWORD': 'password',  
        'HOST': 'damilyusupov.mysql.pythonanywhere-services.com',  
    }  
}
```

Рисунок 3.5 – Подключение базы данных

2.7 Подготовка базы данных

Откроем в разделе Consoles консоль Bash и примем миграции базы данных:

```
python manage.py makemigrations  
python manage.py migrate
```

Рисунок 3.6 – Миграция базы данных

2.8 Настройка веб-сервера PythonAnywhere

PythonAnywhere - это интегрированная онлайн-среда разработки (IDE) и служба веб-хостинга (платформа как услуга), основанная на языке программирования Python. Основанная Джэйлсом Томасом и Робертом Смитсоном в 2012 году, она обеспечивает доступ в браузере к серверным интерфейсам командной строки Python и Bash, а также к редактору кода с подсветкой синтаксиса. Программные файлы могут быть переданы в и из службы с помощью браузера пользователя. Веб-приложения, размещенные службой, могут быть написаны с использованием любой среды приложений на основе WSGI.

PythonAnywhere был создан Resolver Systems, который также создал Resolver One, программу для работы с электронными таблицами на основе Python. 16 октября 2012 года продукт был приобретен новой компанией, компанией PythonAnywhere LLP, которая будет разрабатывать и поддерживать продукт в будущем и взять на себя существующую команду разработчиков.

Команда разработчиков использует PythonAnywhere для разработки PythonAnywhere и говорит, что его возможности для совместной работы помогают, потому что они используют методологию экстремального программирования.

На сайте PythonAnywhere перейдем в раздел Web. Нажмем на кнопку Add a new web app. Появится мастер добавления нового веб-приложения. Смена доменного имени доступна только на платном тарифе, поэтому необходимо нажать Next. В окне «Select a Python Web framework» выбираем Manual

configuration, затем выбираем последнюю доступную версию Python. Нажимаем Next. Ждём, пока приложение будет создано.

На странице «Configuration for familyusupov.pythonanywhere.com» заполняем поля конфигурации сайта.

В секции Code для параметра Source code указываем путь до нашего проекта: /home/familyusupov/simple_env/simple. После ввода значения нужно щёлкнуть небольшую кнопку с галочкой для сохранения введённого значения. Для параметра Working directory указываем тот же путь.

Ниже в секции Virtualenv указываем путь до виртуального окружения нашего проекта: /home/familyusupov/simple_env.

Теперь в секции Code выбираем параметр WSGI configuration file.

Откроется текстовый редактор. Удаляем всё содержимое файла и копируем туда содержимое WSGI-файла нашего проекта. Его можно найти в каталоге проекта рядом с файлом settings.py. Он называется wsgi.py.

Для того, чтобы python смог загрузить модули необходимо вписать путь к проекту.

```
import os, sys

path = '/home/familyusupov/my_project_env/my_project'

if path not in sys.path:

    sys.path.append(path)

from django.core.wsgi import get_wsgi_application

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "my_project.settings")
```

Рисунок 3.7 – wsgi.py файл

После этого остается сохранить изменения и обновить веб-сервер.

2.9 Авторизация

Необходимо на базе веб-сайта, выполняющего расчет рекомендуемого количества автомобилей для сервиса каршеринга, добавить функционал аутентификации. Так как веб-сайт используется ограниченным кругом лиц,

требуется ограничить возможность производить расчет посторонним пользователям.

Для осуществления данной задачи необходимо добавить возможность аутентификации для получения доступа к веб-сайту.

Требования к системе аутентификации:

- информация пользователя добавляется администратором веб-сайта;
- пользователь без учетной записи не должен иметь доступа к основному функционалу сайта;
- необходимо добавить возможность покинуть учетную запись и функцию «восстановить пароль».

Django предоставляет систему аутентификации и авторизации ("permission") пользователя, реализованную на основе фреймворка работы с сессиями. Система аутентификации и авторизации позволяет вам проверять полномочия пользователей и определять какие кому соответствуют. Данный фреймворк включает в себя встроенные модели для пользователей и групп (основной способ применения авторизации для более чем одного пользователя), непосредственно саму систему разграничения доступа (permissions)/флаги, которые определяют какую задачу, с какой формой и отображением, инструментом отображения, может работать тот, или иной пользователь [6].

Система аутентификации является очень гибкой и позволяет формировать собственные URL-адреса, формы, отображения, а также шаблоны страниц, через простой вызов функций соответствующего API для аутентификации пользователя.

В ходе выполнения работы было использовано встроенное в фреймворк Django решение, так как оно соответствует необходимым требованиям к функционалу. В ходе выполнения работы необходимо выполнить следующие пункты:

1. Создать новый модуль, включающий в себя формы логина, выхода, восстановления пароля.
2. Добавить шаблоны форм логина и профиля.
3. Заблокировать доступ к основной части сайта, выполняющей расчет рекомендованного количества автомобилей для сервиса каршеринг для незарегистрированных пользователей.

Для реализации аутентификации необходимо создать модуль пользователя, чтобы создать модуль необходимо активировать виртуальную среду на сервере используя терминал, введем следующее:

- `Source simple_env/bin/activate` – для активации виртуальной среды;
- `python manage.py startapp users` – позволит Django сгенерировать модуль, названный `users`.

После создания модуля, необходимо включить его в список приложений на сайте, в файле `setting.py` в списке приложений, необходимо добавить `'users'`,

```
{% extends "calc/index.html" %}

{% load crispy_forms_tags %}

{% block content %}

    <div class="content-section">

        <form method="POST">

            {% csrf_token %}

            <fieldset class="form-group">

                <legend class="border-bottom mb-4">Вход</legend>

                {{ form|crispy }}

            </fieldset>

            <div class="form-group">

                <button class="btn btn-outline-info" type="submit">Вход</button>

            </div>

        </form>

    </div>

{% endblock content %}
```

это действие даст системе управления знать о модуле пользователей. Для пра-

Рисунок 3.3 – Шаблон формы входа

вильного отображения форм на сайте, требуется шаблон формы, сделанный с помощью `html`

Этот шаблон расширяет базовый шаблон и переопределяет блок контента. Остальная часть кода — это стандартный код обработки формы. На этой форме от пользователя требуется ввести свое имя и пароль, а если будут введены недопустимые значения, будет предложено ввести правильные значения, когда страница обновится.

Этот шаблон отображает сообщение, информирующее о том, что пользователь вышел из системы, и предоставляет ссылку, которую он может нажать, чтобы вернуться на экран входа в систему [7].

```
{% extends "calc/index.html" %}
{% block content %}
    <h2>Вы успешно покинули учетную запись</h2>
    <div class="border-top pt-3">
        <small class="text-muted">
            <a href="{% url 'login' %}">Вход</a>
        </small>
    </div>
{% endblock content %}
```

Рисунок 3.4 – Шаблон выхода

2.10 Доступ к панели администратора

Для доступа к панели управления хостинга необходимо использовать следующую учетную запись:

Логин: damilyusupov

Пароль: j5H9qyLbeFTQqNU

Вывод по главе три

Выбор данного фреймворка (Django) и языка программирования Python позволил разработать рабочее веб-приложение на основе составленной модели расчета рекомендованного количества машин в сервисе каршеринга

Заключение

- 1 Составлена математическая модель расчета рекомендованного количества автомобилей в сервисе каршеринг.
- 2 Проведена оценка адекватности модели, показала.
- 3 Составлены обоснованные требования к веб-приложению.
- 4 Обоснованный выбор фреймворка Django и языка Python позволяют легко интегрировать проект в более крупные системы, построенные на Django/Python
- 5 Дальнейшее развитие работы может иметь в направлениях:
 - 5.1 При организации предприятия оказывающего услуги каршеринга
 - 5.2 При масштабировании компаний занимающихся предоставлением услуг каршеринга
 - 5.3 Для интеграции во внутреннюю часть веб-сайта каршеринговой компании для работы сотрудников
- 6 В дальнейшем веб-приложение можно развивать как самостоятельный продукт или часть большей системы
- 7 Улучшить данное веб-приложение можно добавив следующий функционал:
 - Сохранение полученных данных в базу данных и составление статистики по полученным данным;
 - Добавление новых переменных, таких как класс автомобиля, стаж водителя, покупка автомобилей в лизинг и т.д.

Библиографический список

- 1 Russell, К.-М. Документация Django / К.-М. Russell, А. Holovaty, W. Simon, К.-М. Jacob, М. Wilson // <https://docs.djangoproject.com/en/2.2/>.
- 2 Ramalho, L. Fluent Python / L. Ramalho // O'REILLY, – 2015. – с. 751.
- 3 Galante, А. Документация Bootstrap / А. Galante, В. Harborow // <https://getbootstrap.com/docs/4.3/getting-started/introduction/>.
- 4 David, В Carsharing – Start Up Issues and New Operational Models / В. David // <http://www.communauto.com/%5C/images/TRB2004-003210.pdf>.
- 5 Tibus: The history of web hosting [Электронный ресурс] – Режим доступа: <https://www.tibus.com/blog/the-history-of-web-hosting-how-things-have-changed-since-tibus-started-in-1996/>, свободный. – Загл. с экрана.
- 6 Django book. Аутентификация пользователей в Django [Электронный ресурс] – Режим доступа: <https://djbook.ru/rel1.9/topics/auth/index.html>, свободный. – Загл. с экрана.
- 7 User authentication in Django. [Электронный ресурс] – Режим доступа: <https://docs.djangoproject.com/en/2.2/topics/auth/>, свободный. – Загл. с экрана.
- 8 Carsharing Calculator: Calculate how much car sharing costs you per year - Within 1 minute you know what you need. [Электронный ресурс] – Режим доступа: <https://www.weeshare.com/en/carsharing-calculator>, свободный. – Загл. с экрана.
- 9 Calculator: Estimate Your Savings With Enterprise CarShare. [Электронный ресурс] – Режим доступа: <https://www.enterprise-carshare.com/us/en/calculator.html>, свободный. – Загл. с экрана.
- 10 Каршеринг BelkaCar увеличил парк за счет 200 автомобилей KIA Rio X-Line. [Электронный ресурс] – Режим доступа: <https://www.autostat.ru/news/34069/>, свободный. – Загл. с экрана.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра математического и компьютерного моделирования

РАБОТА ПРОВЕРЕНА

Рецензент,

_____ / А.А. Панюкова

« ____ » _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой МиКМ,

д-р физ.-мат. наук, доцент

_____ / С.А. Загребина

« ____ » _____ 2019 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ К ВЫПУСКНОЙ КВАЛИФИЦИРОВАННОЙ РАБОТЕ ЮУРГУ – 01.03.02. 2019.115-048.11.000 ПЗ

Руководитель работы, доцент каф.
МиКМ, к.т.н.,

_____ / В.И. Дударева

« ____ » _____ 2019 г.

Автор работы,

студент группы ЕТ-416

_____ / Д.С. Юсупов

« ____ » _____ 2019 г.

Нормоконтролер,

Доцент каф. МиКМ, к.ф.-м.н.,

_____ / Т.А. Макаровских

« ____ » _____ 2019 г.

Челябинск 2019

1 ВВЕДЕНИЕ

1.2 Наименование программного изделия

Полное наименование программы – «Веб-сайт для расчета рекомендованного количества автомобилей в сервисе каршеринга». Краткое наименование программы – «Веб-сайт для расчета рекомендованного количества автомобилей в сервисе каршеринга».

1.3 Область применения

Веб-сайт предназначен для сотрудников и учредителей сервиса каршеринга в ходе планирования создания или расширения услуг каршеринга, для расчета рекомендованного количества автомобилей в сервисе каршеринга.

2 ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

2.1 Разработка ведется на основании задания на выпускную квалифицированную работу бакалавра по теме «Создание веб-приложения для расчета рекомендованного количества автомобилей в сервисе каршеринга».

2.2 Организация, утвердившая этот документ, и дата его утверждения

Задание утверждено руководителем работы, доцентом кафедры МиКМ, Дударевой В.И.

2.3 Наименование темы разработки – «Веб-сайт для расчета рекомендованного количества автомобилей в сервисе каршеринга»

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Разработка является частью задания к выпускной квалифицированной работе и позволит рассчитывать рекомендованное количество автомобилей сервиса каршеринга по заданным параметрам.

4 ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1 Требования к функциональным характеристикам

4.2 Состав выполняемых функций

4.3 Расчет рекомендованного количества автомобилей для сервиса

каршеринга по заданным параметрам, таким как:

- 1) стоимость одного автомобиля (в рублях);
- 2) время подготовки между двумя бронированиями (в минутах);
- 3) стоимость подготовки (в рублях);
- 4) ореол подачи (в км);
- 5) количество предзаказов автомобилей (в штуках);
- 6) количество дней использования машины (в днях);
- 7) стоимость, которую клиент готов заплатить единовременно за

срок выпуска авто (в рублях);

- 8) срок выпуска авто (в днях);
- 9) стоимость тех. обслуживания (в рублях);
- 10) стоимость топлива (в рублях);
- 11) расход топлива (в литрах на 100 километров).

4.4 Регистрация пользователей

Функция регистрации пользователей должна обеспечивать ввод регистрационной информации о пользователе и формирование учетной записи, которая должна храниться в базе данных.

Учетная запись должна содержать следующие пункты:

- имя, фамилия, отчество;
- название организации;
- должность;
- телефон;
- электронная почта;

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации должен включать следующие документы:

- технический проект программы по ГОСТ 19.404 -79;
- описание программы по ГОСТ 19.402-78;
- текст программы по ГОСТ 19:401-78;
- руководство пользователя.

6 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Разработка программы должна выполняться по следующим этапам:

- разработка модели расчета рекомендованного количества автомобилей в сервисе каршеринг – 1 неделя;
- разработка веб-сайта на базе языка Python – 6 недель;
- приемка-сдача с исправлением обнаруженных недостатков в программе – 2 недели;
- перенос веб-сайта на хостинг.

7 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Программа должна считаться годной, если она удовлетворяет всем пунктам технического задания

ПРИЛОЖЕНИЕ Б

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра математического и компьютерного моделирования

РАБОТА ПРОВЕРЕНА

Рецензент,

_____/ А.А. Панюкова

« ____ » _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой МиКМ,

д-р физ.-мат. наук, доцент

_____/ С.А. Загребина

« ____ » _____ 2019 г.

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ
К ВЫПУСКНОЙ КВАЛИФИЦИРОВАННОЙ РАБОТЕ
ЮУРГУ – 01.03.02. 2019.115-048.11.000 ПЗ

Руководитель работы, доцент
каф. МиКМ, к.т.н.,

_____/ В.И. Дударева

« ____ » _____ 2019 г.

Автор работы,

студент группы ЕТ-416

_____/ Д.С. Юсупов

« ____ » _____ 2019 г.

Нормоконтролер,

Доцент каф. МиКМ, к.ф.-м.н.,

_____/ Т.А. Макаров-
ских

« ____ » _____ 2019 г.

1 Общие сведения о веб-сайте «Расчет рекомендованного количества автомобилей в сервисе каршеринг»

Данный веб-сайт позволяет вычислить рекомендуемое количество автомобилей в сервисе каршеринга по заданным параметрам, это позволит автоматизировать планирование данного бизнеса для конкретных территорий и аудитории и сократить траты на покупку и содержания ненужных автомобилей.

В данный момент веб-сайт имеет следующий функционал:

- Расчет рекомендованного количества автомобилей в сервисе каршеринг;
- Систему аутентификации.

2 Интерфейс веб-сайта состоит из двух частей, окна аутентификации и окна ввода и вывода информации.

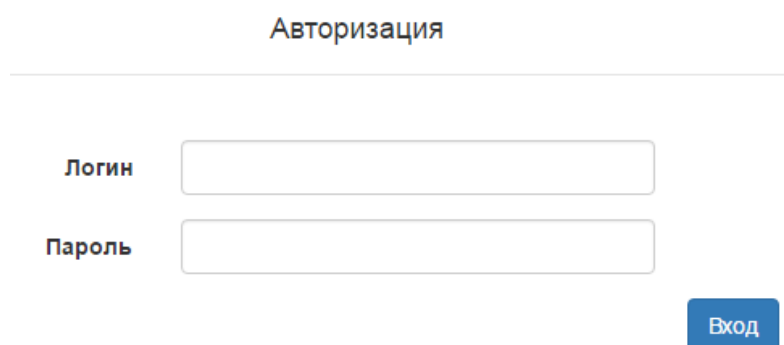
Количество предзаказов данного автомобиля	<input type="text"/> штук	Стоимость, которую клиент готов заплатить единовременно за срок выпуска авто	<input type="text"/> руб.
Стоимость одного автомобиля данной модели	<input type="text"/> руб.	Срок выпуска данного авто	<input type="text"/> дней
Стоимость тех.обслуживания данного автомобиля	<input type="text"/> руб.	Количество дней использования выбранной машины	<input type="text"/> дней
Стоимость топлива	<input type="text"/> руб.	Расход топлива автомобиля	<input type="text"/> литров на 100 км
Ореол подачи автомобиля	<input type="text"/> км	Время подготовки между двумя бронированиями	<input type="text"/> минут
Стоимость подготовки одного автомобиля	<input type="text"/> руб.		

Рисунок 2.1 - Интерфейс страницы веб-сайта

3 Работа с веб-сайтом «Расчет рекомендованного количества автомобилей в сервисе каршеринг»

3.1 Окно аутентификации

Окно аутентификации позволяет пользователю войти в систему для дальнейшего использования веб-сайта. В этом окне пользователю доступны формы ввода логина и пароля.



Авторизация

Логин

Пароль

Рисунок 3.2 - Окно аутентификации

Для входа необходимо выбрать соответствующую форму и заполнить ее предварительно выданными администрацией данными, после заполнения формы необходимо нажать на кнопку войти.

3.2 Страница ввода данных

Страница ввода данных состоит из 11 форм для ввода параметров расчета и кнопку готово, которая не будет активна до момента как все формы будут заполнены.

Данные для ввода в формы:

- 1) стоимость одного автомобиля (в рублях);
- 2) время подготовки между двумя бронированиями (в минутах);
- 3) стоимость подготовки (в рублях);
- 4) ореол подачи (в км);
- 5) количество предзаказов автомобилей (в штуках);
- 6) количество дней использования машины (в днях);
- 7) стоимость, которую клиент готов заплатить единовременно за срок выпуска авто (в рублях);
- 8) срок выпуска авто (в днях);

- 9) стоимость тех. обслуживания (в рублях);
- 10) стоимость топлива (в рублях);
- 11) расход топлива (в литрах на 100 километров).

До тех пор, пока все формы не будут заполнены соответствующими данными кнопка будет оставаться неактивной.

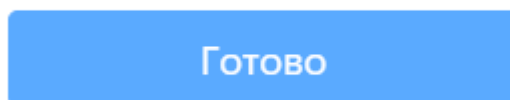


Рисунок 3.3 - Неактивная кнопка

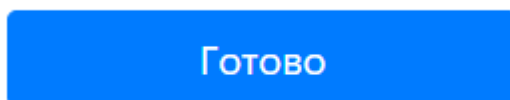


Рисунок 3.4 - Активная кнопка

3.3 Вывод результата расчета рекомендованного количества автомобилей в системе каршеринг

Как только пользователь заполнил все формы соответствующими данными он может нажать кнопку готово и результат расчетов будет выведен на том же экране. Вывод результата расчетов производится на той же странице, что и ввод входных данных.

Количество предзаказов данного автомобиля	<input type="text" value="штук"/>	Стоимость, которую клиент готов заплатить единовременно за срок выпуска авто	<input type="text" value="руб."/>
Стоимость одного автомобиля данной модели	<input type="text" value="руб."/>	Срок выпуска данного авто	<input type="text" value="дней"/>
Стоимость тех.обслуживания данного автомобиля	<input type="text" value="руб."/>	Количество дней использования выбранной машины	<input type="text" value="дней"/>
Стоимость топлива	<input type="text" value="руб."/>	Расход топлива автомобиля	<input type="text" value="литров на 100 км"/>
Ореол подачи автомобиля	<input type="text" value="км"/>	Время подготовки между двумя бронированиями	<input type="text" value="минут"/>
Стоимость подготовки одного автомобиля	<input type="text" value="руб."/>	<input type="button" value="Готово"/>	Рекомендованное количество автомобилей для заданных параметров: 4

Рисунок 3.5 - Результат расчета рекомендованного количества автомобилей в сервисе каршеринга.

ПРИЛОЖЕНИЕ В

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра математического и компьютерного моделирования

РАБОТА ПРОВЕРЕНА

Рецензент,

_____/ А.А. Панюкова

« ____ » _____ 2019 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой МиКМ,

д-р физ.-мат. наук, доцент

_____/ С.А. Загребина

« ____ » _____ 2019 г.

Создание веб-приложения для расчета рекомендованного количества автомобилей в сервисе каршеринга

ТЕКСТ ПРОГРАММЫ

К ВЫПУСКНОЙ КВАЛИФИЦИРОВАННОЙ РАБОТЕ

ЮУРГУ – 01.03.02. 2019.115-048.11.000 ПЗ

Руководитель работы, доцент
каф. МиКМ, к.т.н.,

_____/ В.И. Дударева

« ____ » _____ 2019 г.

Автор работы,

студент группы ЕТ-416

_____/ Д.С. Юсупов

« ____ » _____ 2019 г.

Нормоконтролер,

Доцент каф. МиКМ, к.ф.-м.н.,

_____/ Т.А. Макаров-

ских

« ____ » _____ 2019 г.

Челябинск 2019

4 Подключение приложений и базы данных

```
import os
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

SECRET_KEY = ''

DEBUG = 'FALSE'

ALLOWED_HOSTS = ['damilyusupov.pythonanywhere.com',]

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'calc',
]

ROOT_URLCONF = 'web_calc.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
WSGI_APPLICATION = 'web_calc.wsgi.application'
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'damilyusupov$wcalc',
        'USER': 'damilyusupov',
        'PASSWORD': 'YW5wvadPfTVMA5h',
        'HOST': 'damilyusupov.mysql.pythonanywhere-services.com',
    }
}
]
```


8 Расчет рекомендованного количества автомобилей в сервисе каршеринг

```
from django.shortcuts import render, redirect
from django.urls import reverse
from django.contrib import messages

def index(request):
    if request.method == 'POST':
        try:
            x1 = int(request.POST['x1'])
            x2 = int(request.POST['x2'])
            x3 = int(request.POST['x3'])
            x4 = int(request.POST['x4'])
            x5 = int(request.POST['x5'])
            x6 = int(request.POST['x6'])
            x7 = int(request.POST['x7'])
            x8 = int(request.POST['x8'])

        except ValueError:
            messages.error(request, 'Заполните все поля')
            return redirect('index')

        else:
            result = round((x1 * (x2 / x3) * ((x2 / x4) / ((x5 / 365)
+ x6 * (x7 * x8))))))

            template = 'calc/index.html'
            return render(request, template, {'result': result})
```

5 Аутентификация

```
from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth.decorators import login_re-
quired
from .forms import UserRegisterForm

def register(request):
    if request.method == 'POST':
        form = UserRegisterForm(request.POST)
        if form.is_valid():
            form.save()
            username =
form.cleaned_data.get('username')
            messages.success(request, Ваш профиль был
создан')
            return redirect('login')
        else:
            form = UserRegisterForm()
            return render(request, 'users/register.html',
{'form': form})

@login_required
def profile(request):
    return render(request, 'users/profile.html')
```

9 Шаблон сайта

```
<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boot-
strap/4.3.1/css/bootstrap.min.css" integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

<title>Расчет рекомендуемого количества автомобилей в системе карше-
ринг</title>

</head>

<body>

    <div class="form-group">
        <form method="post" action = "{% url 'index' %}">
            {% csrf_token %}
            <div class="container">
                <div class="row" style="margin-top: 5%">
                    <div class="col-sm">
                        Количество предзаказов данного автомобиля
                    </div>
                    <div class="col-sm">
                        <input name="x1" class="form-control"
placeholder="штук"></input>
                    </div>
                    <div class="col-sm">
                        Стоимость, которую клиент готов заплатить
единовременно за срок выпуска авто
                    </div>
                    <div class="col-sm">
                        <input name="x2" class="form-control"
placeholder="руб." aria-required="true" data-ok="false"></input>
                    </div>
                </div>
            </div>
        </form>
    </div>
```

```

<div class="row" style="margin-top: 5%">
  <div class="col-sm">
    Стоимость одного автомобиля данной модели
  </div>
  <div class="col-sm">
    <input name="x3" class="form-control"
placeholder="руб."></input>
  </div>
  <div class="col-sm">
    Срок выпуска данного авто
  </div>
  <div class="col-sm">
    <input name="x4" class="form-control"
placeholder="дней"></input>
  </div>
</div>
<div class="row" style="margin-top: 5%">
  <div class="col-sm">
    Стоимость тех.обслуживания данного автомо-
биля
  </div>
  <div class="col-sm">
    <input name="x5" class="form-control"
placeholder="руб."></input>
  </div>
  <div class="col-sm">
    Количество дней использования выбранной
машины
  </div>
  <div class="col-sm">
    <input name="x6" class="form-control"
placeholder="дней"></input>
  </div>
</div>

```

```

<div class="row" style="margin-top: 5%">
  <div class="col-sm">
    Стоимость топлива
  </div>
  <div class="col-sm">
    <input name="x7" class="form-control"
placeholder="руб."></input>
  </div>
  <div class="col-sm">
    Расход топлива автомобиля
  </div>
  <div class="col-sm">
    <input name="x8" class="form-control"
placeholder="литров на 100 км"></input>
  </div>
</div>
<div class="row" style="margin-top: 5%">
  <div class="col-sm mr-auto">
    Ореол подачи автомобиля
  </div>
  <div class="col-sm">
    <input name="x9" class="form-control"
placeholder="км"></input>
  </div>
  <div class="col-sm">
    Время подготовки между двумя бронировани-
ями
  </div>
  <div class="col-sm">
    <input name="x10" class="form-control"
placeholder="минут"></input>
  </div>
</div>
<div class="row" style="margin-top: 5%">

```

```

        <div class="col-sm">
            Стоимость подготовки одного автомобиля
        </div>
        <div class="col-sm">
            <input name="x11" class="form-control"
placeholder="руб."></input>
        </div>
        <div class="col-sm">
            <button type="submit" class="btn btn-pri-
mary btn-lg btn-block" name="btn" value="add">Готово</button>
        </div>
        <div class="col-sm">
            Рекомендованное количество автомобилей для
заданных параметров: 4
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</form>
</div>
</body>
</html>

```