

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н.П. ОГАРЁВА»

Факультет математики и информационных технологий

Кафедра фундаментальной информатики

УТВЕРЖДАЮ

Зав. кафедрой

канд. физ.-мат. наук, доц.

 А.Г. Смольянов
(подпись)

«10» 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

**РАЗРАБОТКА CRM СИСТЕМЫ И ЛИЧНОГО КАБИНЕТА С ПОМОЩЬЮ
ФРЕЙМВОРКА DJANGO**

Автор бакалаврской работы

 (подпись)

07.06.19.
(дата)

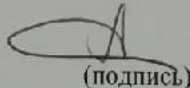
А.Н. Галяткин

Обозначение бакалаврской работы БР-02069964-02.03.02-02-19

Направление 02.03.02 Фундаментальная информатика и информационные технологии

Руководитель работы

канд. физ. – мат. наук, доц.


 (подпись)

10.06.19
(дата)

А.Г. Смольянов

Нормоконтролер

канд. тех. наук, доцент

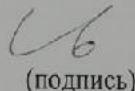
 (подпись)

10.06.19
(дата)

С.В. Гарина

Рецензент

канд. физ. – мат. наук, доц.

 (подпись)

10.06.19
(дата)

Е.В. Десяев

Саранск

2019

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н.П. ОГАРЁВА»

Факультет математики и информационных технологий

Кафедра фундаментальной информатики

УТВЕРЖДАЮ

Зав. кафедрой
канд. физ.-мат. наук, доц.

 А.Г. Смольянов
(подпись)

« 12 » 12 20 18 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

(в форме бакалаврской работы)

Студент Галяткин Андрей Николаевич

1 Тема «Разработка CRM системы и личного кабинета с помощью фреймворка DJANGO»

Утверждена приказом № 10282-С от 12.12.2018

2 Срок предоставления работы к защите 07.06.2019

3 Исходные данные для научного исследования (проектирования) научная и учебно-методическая литература, периодические издания, электронные ресурсы

4 Содержание выпускной квалификационной работы

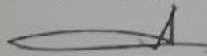
4.1 Веб-Фреймворк

4.2 CRM-система

4.3 Разработка и реализация программного веб-проекта

5 Приложение А (обязательное) Реализация входа и регистрации потенциальных заказчиков на сайте.

Руководитель работы


(подпись)

12.12.2018г.
(дата)

А. Г. Смольянов

Задание принял к исполнению


(подпись)

12.12.2018г.
(дата)

А. Н. Галяткин

РЕФЕРАТ

Выпускная квалификационная работа содержит 54 страницы, 35 рисунков, 10 использованных источников, 1 приложение.

CRM-СИСТЕМА, ФРЕЙМБОРК, DJANGO, MVT, PYTHON, JQUERY, BOOTSTRAP, PIP, БАЗА ДАННЫХ, ЛИДЫ, ЗАКАЗЧИКИ.

Объектом исследования является веб-фреймворк Django, веб-технологии HTML, jQuery, AJAX и Bootstrap в целях разработки CRM-системы.

Цель работы – разработка CRM-системы для туристического агентства и личного кабинета с помощью веб-фреймвока Django.

В результате проведенной работы были изучены технологии, применяемые в настоящее время для реализации CRM-систем, проанализированы особенности и возможности веб-фреймворка Django, языка программирования Python, веб-технологии jQuery и Bootstrap. Спроектирована и разработана база данных, реализован веб-интерфейс для работы с веб-проектом, создан механизм регистрации и входа заказчиков, реализованы функции для работы с возможностями сервиса.

Степень внедрения – частичная.

Область применения – CRM-система будет полезна небольшим турагентствам.

Эффективность – имеет прикладное значение, так как позволяет CRM-менеджерам эффективнее работать с потенциальными заказчиками, заполнять базу данных и реагировать на заказы. В свою очередь заказчики имеют доступ к простому интерфейсу сайта, где могут заказать тур и просматривать детали своих заказов в личном кабинете.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 Веб-Фреймворк	8
1.1 Общая информация о веб-фреймворках	8
1.2 Типы веб-фреймворков	8
1.2.1 Серверные веб-фреймворки	8
1.2.2 Клиентские веб-фреймворки	9
1.3 Архитектура веб-фреймворков	9
1.4 Особенности веб-фреймворков	10
1.4.1 Веб-кэширование	10
1.4.2 Скаффолдинг	11
1.4.3 Система веб-шаблонов	11
1.4.4 Сопоставление URL	11
1.4.5 Приложения	11
2 CRM-система	13
2.1 Общая информация о CRM-системе	13
2.2 Необходимость внедрения CRM	13
2.3 Классификация CRM	14
2.3.1 Операционная CRM	14
2.3.2 Аналитическая CRM	15
2.3.3 Коллаборационная CRM	15
3 Разработка и реализация программного веб-проекта	16
3.1 Постановка задачи	16
3.2 Подбор инструментов для реализации	16
3.3 Программная реализация веб-проекта	23
3.3.1 Описание концепции разрабатываемого веб-проекта	23
3.3.2 Архитектура веб-проекта	24
3.3.3 Структура веб-проекта	24
3.3.4 Структура веб-приложения	27
3.4 Описание функциональности	31
3.4.1 Общедоступная часть	31

3.4.2 Административная часть	40
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	49
ПРИЛОЖЕНИЕ А (обязательное) Реализация входа и регистрации потенциальных заказчиков на сайте	50

ВВЕДЕНИЕ

В настоящий момент конкуренция на всех уровнях рынка довольно высока. Чтобы компании выиграть конкурентную борьбу, она должна не только привлекать новых клиентов, но и удерживать уже существующих.

Для удержания клиента необходимо учитывать его интересы и пожелания. Такой подход к ведению бизнеса называется клиентоориентированным. Но при огромной клиентской базе учет интересов каждого клиента является труднодостижимой целью.

Выходом из данной ситуации является внедрение в компании CRM-системы. Расшифровка аббревиатуры CRM звучит как Customer Relationship Management, что в переводе на русский означает «Управление Взаимоотношениями с клиентами»

CRM-модель взаимодействия, полагающая, что центром всей философии бизнеса является клиент, а основными направлениями деятельности являются меры по поддержке эффективного маркетинга, продаж и обслуживания клиентов. Поддержка этих бизнес-целей включает сбор, хранение и анализ информации о потребителях, поставщиках, партнёрах, а также о внутренних процессах компании. Функции для поддержки этих бизнес-целей включают продажи, маркетинг, поддержку потребителей и взаимоотношение клиент-управление.

Целью данной бакалаврской работы является разработка CRM-системы для туристического агентства с помощью фреймвока Django, которая облегчит не только работу CRM-менеджеров агентства благодаря своим простым интерфейсом, но и заказчиков, которые смогут легко заказывать туры на сайте и отслеживать информацию о заказах в личном кабинете.

Бакалаврская работа состоит из введения, трех разделов и заключения.

В первом разделе описаны фреймворки, их типы, архитектура и особенности.

Во втором разделе рассмотрены CRM-системы в целом.

В третьем разделе рассказывается про процесс практической разработки веб-проекта, дана постановка задачи, выбраны средства для разработки, указаны плюсы данных средств, описана функциональность данного веб-приложения, как со стороны лидов и заказчиков, так и со стороны CRM-менеджеров.

1 Веб-Фреймворк

1.1 Общая информация о веб-фреймворках

Веб-фреймворк – инструмент, облегчающий процесс написания и запуска веб-приложения. Используя фреймворк не нужно самостоятельно писать большое количество кода и тратить время на поиск потенциальных просчётов и ошибок [3].

На рассвете эры веб-разработки все приложения писались вручную, и только разработчик приложения мог изменить или развернуть его. Веб-фреймворки позволили избежать этого. С 1995 года все проблемы, связанные с изменением структуры приложения, были в той или иной степени устранены благодаря появлению общего подхода к разработке веб-приложений. В это время появились языки для веба. Сейчас их разнообразие позволяет выбрать подходящий как для статических, так и для динамических страниц. В зависимости от поставленной задачи, можно выбрать один фреймворк, покрывающий все нужды, или совместить несколько.

1.2 Типы веб-фреймворков

У веб-фреймворков есть две основные функции: работа на серверной стороне (бэкенд) и работа на клиентской стороне (фронтенд).

Фронтенд-фреймворки связаны с внешней частью приложения. Простыми словами, они отвечают за внешний вид приложения. Бэкенд отвечает за внутренне устройство приложения.

1.2.1 Серверные веб-фреймворки

Правила и архитектура таких веб-фреймворков не даёт возможности создать веб-приложение с богатым интерфейсом. Они ограничены в своей функциональности, однако с их помощью всё равно можете создавать простые

страницы и разные формы. Также они могут формировать выходные данные и отвечать за безопасность в случае атак. Всё это определённо может упростить процесс разработки. Серверные веб-фреймворки в основном отвечают за отдельные, но критически важные части приложения, без которых оно не сможет нормально работать. Несколько самых популярных серверных фреймворков и языки, с которыми они работают:

- Django — Python;
- Zend — PHP;
- Express.js — JavaScript;
- Ruby on Rails — Ruby.

1.2.2 Клиентские веб-фреймворки

В отличие от серверных, клиентские веб-фреймворки никак не связаны с логикой приложения. Этот тип веб-фреймворков работает в браузере. С их помощью можно улучшить и внедрить новые пользовательские интерфейсы. Фронтенд-фреймворки позволяют создавать разные анимации и одностраничные приложения. Все клиентские фреймворки отличаются по функциональности и использованию. Вот некоторые из них:

- Backbone+Marionette;
- Angular;
- Ember.js;
- Vue.js.

Все эти веб-фреймворки используют JavaScript.

1.3 Архитектура веб-фреймворков

Архитектура любого веб-фреймворка содержит три составляющие, а именно «Модель-Представление-Контроллер» (Model-View-Controller) (MVC).

MVC — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента таким

образом, что модификация каждого компонента может осуществляться независимо. Структура MVC показана на рисунке 1.

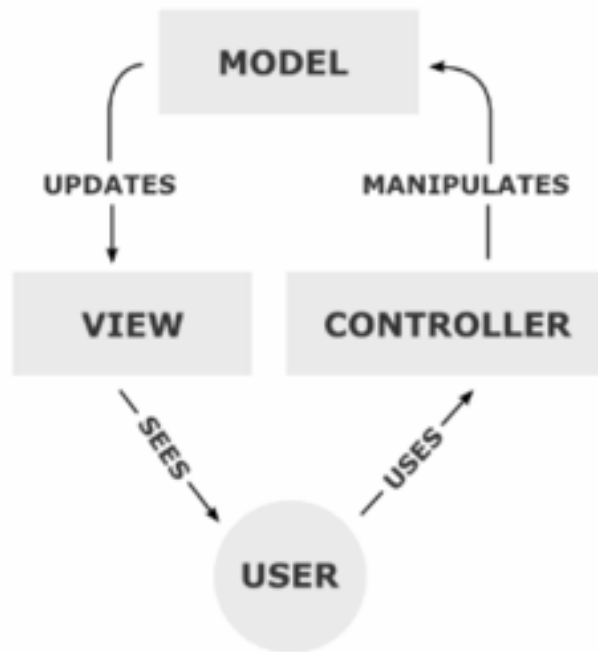


Рисунок 1 – Структура MVC

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.

1.4 Особенности веб-фреймворков

1.4.1 Веб-кэширование

Кэширование просто помогает хранить разные документы и позволяет избежать надоедливой перегрузки сервера. Пользователи могут использовать его в разных системах при определённых условиях. Также оно работает на серверной стороне. Например, можно заметить кэшированные веб-страницы на странице результатов поисковой выдачи Google.

1.4.2 Скаффолдинг

Это ещё одна технология, поддерживаемая некоторыми MVC-фреймворками. Фреймворк может автоматически сгенерировать типичные части приложения или даже всю структуру проекта (если речь идёт о инициализации). Это позволяет существенно увеличить скорость разработки и стандартизирует кодовую базу.

1.4.3 Система веб-шаблонов

Система веб-шаблонов представляет собой набор разных методологий и программного обеспечения, реализованных для создания и развёртывания веб-страниц. Для обработки веб-шаблонов используются шаблонизаторы. Они являются инструментом фреймворка, отвечающим за веб-публикацию.

1.4.4 Сопоставление URL

Если необходимо упростить индексацию сайта поисковыми движками, в то же время создавая привлекательное название для сайта, то используется эта функция фреймворков. Также сопоставление URL может облегчить доступ к страницам сайта.

1.4.5 Приложения

Множество типов веб-приложений поддерживаются веб-фреймворками. В основном они применяются для создания таких приложений, как блоги, форумы, CMS и т. д.

Вся эта функциональность свойственна всем веб-фреймворкам. Однако, как ни парадоксально, с таким широким ассортиментом разработчик теряется и не может ничего выбрать. Поэтому нужно придумать критерии, согласно которым можно выбрать лучший инструмент, облегчающий разработку. Например, выбор может зависеть от предпочитаемого языка программирования. Кроме того, нужно обратить внимание на круг

возможностей набора инструментов фреймворка. Говоря о предпочтениях, стоит отметить, что они могут служить как во благо, так и во вред. Конечно, лучше использовать фреймворки, которые проще изучить.

2 CRM-система

2.1 Общая информация о CRM-системе

Система управления взаимоотношениями с клиентами (CRM, CRM-система, Customer Relationship Management) – прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов [1].

На самом деле, CRM-системой можно считать любой вариант контроля и учета, который поможет улучшить взаимодействие с клиентами. Даже если ведется история звонков и контактов на бумаге или в Excel – это тоже можно считать CRM-системой в том случае, если разработанная схема учета и контроля работает и позволяет контролировать все варианты взаимодействия с клиентами. Конечно, такие методы ведения учета уходят в прошлое, ведь в современном мире без эффективной автоматизации сложно представить себе работу любого бизнеса. А потому, когда говорят о CRM-системе, обычно подразумевают специальное программное обеспечение.

Другими словами, CRM-система – это любое программное обеспечение, которое помогает успешно контролировать работу с клиентами, упорядочить и планировать её.

2.2 Необходимость внедрения CRM

CRM нужна для того, чтобы в первую очередь не потерять потенциального клиента. В малом и среднем бизнесе нашей страны конкуренция очень высокая, поэтому компании прикладывают значительные усилия для того, чтобы

привлечь клиентов. На такие типы затрат компанией выделяется значительный бюджет и очень важно, чтобы эти затраты оправдали себя.

Во-вторых, для контроля сотрудников компании и стандартизации работы с клиентами. Без общей стандартизированной CRM-системы каждый сотрудник будет работать так, как он привык.

В-третьих, для накопления статистической базы, что очень важно для развития любого бизнеса. Благодаря использованию CRM-системы вся рабочая информация собирается в одной общей базе в стандартизированном виде.

2.3 Классификация CRM

Существуют разные типы CRM-систем, которые различаются уровнем обработки информации и решаемых задач. Например, турагенству необходимо вести базу данных потенциальных клиентов, фиксировать в базе их телефонные номера для связи и рекламы предложений.

2.3.1 Операционная CRM

Главная задача операционных CRM-систем – повысить лояльность клиента в процессе непосредственного контакта с ним.

Такие системы упрощают взаимодействие с клиентами, систематизируют данные о заявках и сделках, сами выставляют счета, напоминают перезвонить клиенту и могут сами отправить ему sms-уведомление, записывают телефонные звонки и т. д. [2].

Все операционные CRM умеют:

- Регистрировать входящий трафик (звонки, письма, заявки на сайт);
- Хранить в базе данные о клиентах, заявках, сделках, задачах и т. д.;
- Автоматизировать документооборот внутри компании;
- Фиксировать продвижение сделок по воронке продаж;
- Напоминать о запланированных звонках, письмах, встречах;
- Ставить задачи и контролировать работу сотрудников.

2.3.2 Аналитическая CRM

Цель аналитических CRM – это анализ накопленной информации о клиентах и продажах для выстраивания более эффективной стратегии.

В отличие от операционных систем, аналитические не только фиксируют историю взаимодействия с клиентом, но и помогают проследить закономерности в продажах: клиенты из каких источников покупают чаще всего, на каком этапе срывается большинство сделок, как распределены клиенты по воронке продаж – и все эти данные обновляются в онлайн-режиме, в разрезе любого параметра.

Аналитические CRM умеют:

- Сегментировать клиентскую базу;
- Определять ценность клиента;
- Анализировать их рентабельность;
- Отслеживать поведение клиентов на каждом этапе сделки;
- Показывать распределение сделок по воронке продаж;
- Анализировать динамику продаж;
- Анализировать эффективность маркетинговых инструментов;
- Прогнозировать объем продаж.

2.3.3 Коллаборационная CRM

Такие CRM налаживают коммуникации с клиентами для сбора обратной связи. Информация, полученная с их помощью, помогает скорректировать ассортимент товаров, ценовую политику, а также процесс обслуживания покупателей. Например, сотрудники call-центра турагенства обзванивают клиентов, задавая вопросы о качестве проведенной поездки в ту или иную страну и фиксируя ответы, корректируют цены на предлагаемый тур.

3. Разработка и реализация программного веб-проекта

3.1 Постановка задачи

Требуется разработать CRM-систему для турагентства и личный кабинет с помощью веб-фреймворка Django, где CRM-менеджеры могут добавлять новые туры, отслеживать потенциальных заказчиков и вести заказы. Заказчики, в свою очередь, могут заказывать туры на сайте и просматривать свои заказы в личном кабинете.

Программное обеспечение должно обеспечивать следующие функциональные возможности:

1. Добавление данных в базу данных;
2. Считывание данных из базы данных;
3. Стилизация шаблонов с помощью CSS-фреймворка Bootstrap;
4. Динамическое изменение количества добавляемых заказов, количества человек в заказе и цены заказа, а также динамическое удаление заказа с помощью JQuery;
5. Отображение данных о заказе в личном кабинете;
6. Отображение данных о клиентах и потенциальных клиентах в административной части.

3.2 Выбор инструментов для реализации

На сегодняшний день существует множество способов реализации CRM-системы с помощью веб-фреймворка, например, AllcountJS, Zend, Yii, Django и т. д. Для реализации своего проекта я выбрал именно веб-фреймворк Django версии 1.11. Django это свободный фреймворк для веб-приложений написанный на языке Python.

Python представляет популярный высокоуровневый язык программирования, который предназначен для создания приложений различных типов. Это и веб-приложения, и игры, и настольные программы, и

работа с базами данных. Довольно большое распространение Python получил в области машинного обучения и исследований искусственного интеллекта. В своей работе я выбрал версию языка Python 2.7.9.

Изначально Django был создан в 2005 году для управления новостными сайтами LJWorld.com, lawrence.com и KUsports.com, принадлежащими компании The World Company. В 2008 году вышел публичный первый релиз фреймворка. На сегодняшний день он продолжает развиваться. Текущей версией фреймворка является версия 2.2 вышедшая 1 апреля 2019 года. Также постоянно выходят подверсии, который дополняют фреймворк и исправляют малочисленные ошибки [4].

Django довольно популярен. Он используется на многих сайтах, таких как Instagram, Disqus, Mozilla, The Washington Times, Pinterest, YouTube, Google и др.

Также Django используется в качестве веб-компонента в различных проектах, таких как Graphite - система построения графиков и наблюдения, FreeNAS - свободная реализация системы хранения и обмена файлами и др.

Фреймворк Django реализует архитектурный паттерн Model-View-Template или сокращенно MVT, который по факту является модификацией распространенного в веб-программировании паттерна MVC (Model-View-Controller, Модель-Представление-Контроллер).

Схематично можно представить архитектуру MVT в Django следующим образом (рисунок 2):

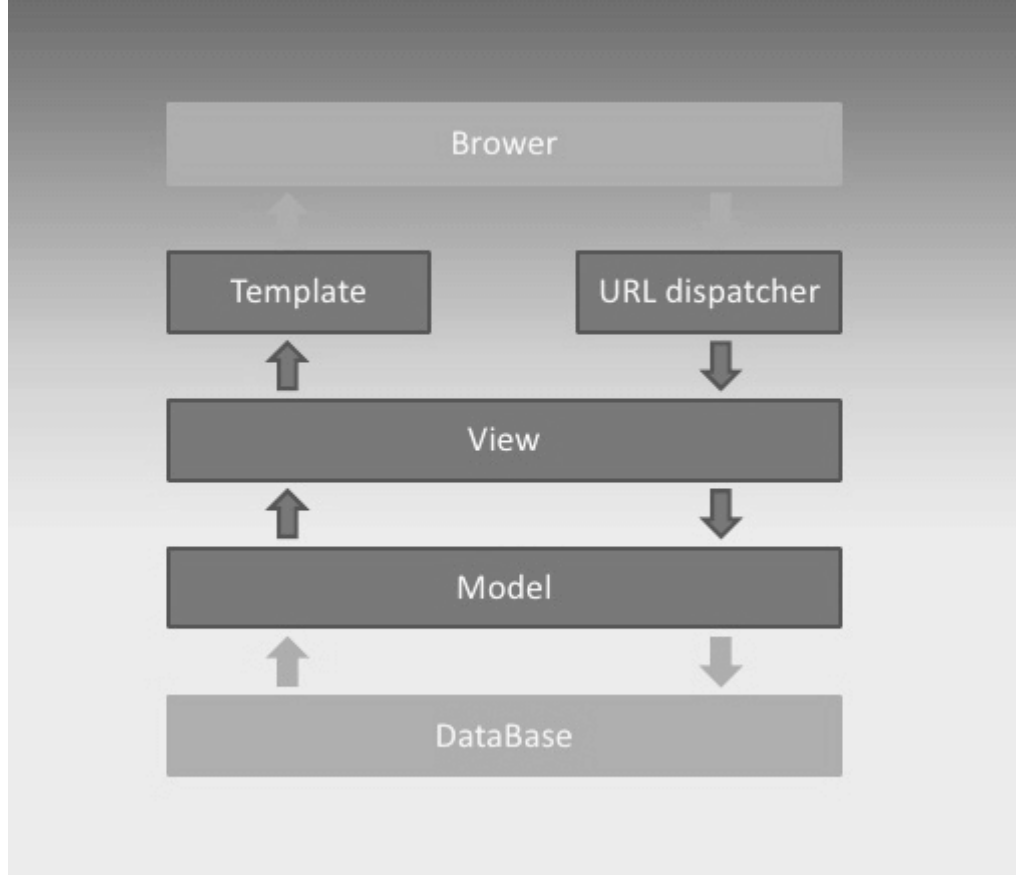


Рисунок 2 – Архитектура MVT

Основные элементы паттерна:

- URL dispatcher: при получении запроса на основании запрошенного адреса URL определяет, какой ресурс должен обрабатывать данный запрос;
- View: получает запрос, обрабатывает его и отправляет в ответ пользователю некоторый ответ. Если для обработки запроса необходимо обращение к модели и базе данных, то View взаимодействует с ними. Для создания ответа может применять Template или шаблоны. В архитектуре MVC этому компоненту соответствуют контроллеры (но не представления);
- Model: описывает данные, используемые в приложении. Отдельные классы, как правило, соответствуют таблицам в базе данных;
- Template: представляет логику представления в виде сгенерированной разметки html. В MVC этому компоненту соответствует View, то есть представления.

Когда к приложению приходит запрос, то URL dispatcher определяет, с каким ресурсом сопоставляется данный запрос и передает этот запрос выбранному ресурсу. Ресурс фактически представляет функцию или View, который получает запрос и определенным образом обрабатывает его. В

процессе обработки View может обращаться к моделям и базе данных, получать из нее данные, или, наоборот, сохранять в нее данные. Результат обработки запроса отправляется обратно, и этот результат пользователь видит в своем браузере. Как правило, результат обработки запроса представляет сгенерированный html-код, для генерации которого применяются шаблоны (Template).

Из плюсов данного веб-фреймворка можно выделить:

- Быстрота: Django был разработан, чтобы помочь разработчикам создать приложение быстро настолько, насколько это возможно. Это включает в себя формирование идеи, разработку и выпуск проекта, где Django экономит время и ресурсы на каждом из этих этапов. Таким образом, его можно назвать идеальным решением для разработчиков, для которых вопрос дедлайна стоит в приоритете;
- Полная комплектация: Django работает с десятками дополнительных функций, которые заметно помогают с аутентификацией пользователя, картами сайта, администрированием содержимого и многим другим. Данные аспекты помогают осуществить каждый этап веб разработки;
- Безопасность: Работая в Django, вы получаете защиту от ошибок, связанных с безопасностью и ставящих под угрозу проект. Я имею ввиду такие распространенные ошибки, как инъекции SQL, кросс-сайт подлоги, clickjacking и кросс-сайтовый скриптинг. Для эффективного использования логинов и паролей, система пользовательской аутентификации является ключом;
- Масштабируемость: фреймворк Django наилучшим образом подходит для работы с самыми высокими трафиками. Следовательно, логично, что великое множество загруженных сайтов используют Django для удовлетворения требований, связанных с трафиком;
- Разносторонность: менеджмент контента, научные вычислительные платформы, даже крупные организации – со всем этим можно эффективно справляться при помощи Django.

Также стоит отметить, что для работы с данным веб-фреймворком нет необходимости устанавливать IDE редакторы. Достаточно установить интерпретатор Python и установить в него Django.

Для того чтобы установить Django, потребуется пакетный менеджер `pip`.

`pip` – система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python.

Начиная с версии Python 2.7.9 и Python 3.4, они содержат пакет `pip` (или `pip3` для Python 3) по умолчанию.

Для стилизации шаблонов в проекте мной был использован CSS-фреймворк Bootstrap версии 3.3.7.

Bootstrap – это CSS фреймворк, который изначально создавался для внутреннего использования компанией «Twitter» с рабочим названием «Twitter Blueprint», но в итоге был опубликован в открытый доступ и стал хорошим набором инструментов для front-end разработки под названием «Bootstrap» [8].

Преимущества Bootstrap:

- Высокая скорость разработки макетов страниц сайта. Bootstrap содержит огромный набор готовых решений и элементов;
- Кроссбраузерность и адаптивность сайта. Все элементы фреймворка адаптивны под все устройства и корректно отображаются во всех современных браузерах;
- Легкость в использовании. Даже человек, имеющий базовые знания о HTML и CSS, может свободно создавать web-страницы с использованием фреймворка;
- Простота в обучении. У Bootstrap очень хорошая документация с большим количеством примеров готового кода.

Основные инструменты Bootstrap:

- Сетки – заранее заданные размеры колонок, которые можно сразу же использовать, например, ширина колонки 140 px относится к классу `.span2` (`.col-md-2` в третьей версии фреймворка), который можно использовать в CSS-описании документа;
- Шаблоны – фиксированный или резиновый шаблон документа;

- Типографика – описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.;
- Медиа – предоставляет некоторое управление изображениями и видео.
- Таблицы – средства оформления таблиц, вплоть до добавления функциональности сортировки;
- Формы – классы для оформления форм и некоторых событий, происходящих с ними;
- Навигация – классы оформления для панелей, вкладок, перехода по страницам, меню и панели инструментов;
- Алерты – оформление диалоговых окон, подсказок и всплывающих окон.

Для динамического изменение количества добавляемых заказов, количества человек в заказе и цены заказа, а также динамического удаление заказа в проекте был использован jQuery версии 1.11.1.

jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX. Разработка jQuery ведётся командой добровольцев на пожертвования [5].

Возможности jQuery:

- Движок кросс-браузерных CSS-селекторов Sizzle, выделившийся в отдельный проект;
- Переход по дереву DOM, включая поддержку XPath как плагина;
- События;
- Визуальные эффекты;
- AJAX-дополнения;
- JavaScript-плагины.

Для хранения данных как правило необходима база данных.

База данных – совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними.

Существует огромное количество разновидностей баз данных, отличающихся по различным критериям.

Классификация по модели данных:

- Иерархическая;
- Объектная и объектно-ориентированная;
- Объектно-реляционная;
- Реляционная;
- Сетевая;
- Функциональная.

Django проектировался для работы под управлением Apache с модулем `mod python` и с использованием PostgreSQL в качестве базы данных.

В настоящее время, помимо базы данных PostgreSQL, Django может работать с другими СУБД: MySQL, SQLite и Oracle.

В данной работе используется база данных SQLite. SQLite — это база данных, чем-то похожая на MySQL. Принципиальное отличие SQLite от других БД в том, что вся база представляет собой один файл. Если в MySQL база хранится где-то в дебрях сервера и недоступна для переноса, то в SQLite просто: один файл — одна база.

3.3 Программная реализация веб-проекта

3.3.1 Описание концепции разрабатываемого веб-проекта

Концепция разрабатываемого программного продукта представляет собой веб-проект, построенный на веб-фреймворке Django, предоставляющий следующие возможности:

- Отображение категорий туров на главной странице;
- Отображение туров на главной странице;
- Отображение страниц туров, включающих в себя описание туров;
- Отображение форм регистрации и входа для незарегистрированных пользователей;
- Отображение заказов зарегистрированных пользователей;
- Отображение личного кабинета зарегистрированных пользователей;
- Отображение деталей заказа в личном кабинете зарегистрированных пользователей;
- Добавление новых категорий туров в административной части веб-приложения CRM-менеджерами;
- Добавление новых стран в административной части веб-приложения CRM-менеджерами;
- Добавление туров в административной части веб-приложения CRM-менеджерами;
- Просмотр зарегистрированных пользователей в административной части веб-приложения CRM-менеджерами;
- Просмотр заказов пользователей в административной части веб-приложения CRM-менеджерами.

Так как веб-проект служит в первую очередь как CRM-система, в качестве критериев эффективности выступят основные функции, которые должны присутствовать в CRM-системе. В список основных функций CRM-системы моего веб-проекта входят:

- Отслеживание потенциальных клиентов;
- Отслеживание заказов клиентов.

Со стороны клиентов турагентства реализованы следующие функции:

- Возможность регистрации и входа;
- Заказ туров на сайте;
- Просмотр деталей заказа в личном кабинете.

3.3.2 Архитектура веб-проекта

Проект состоит из двух частей:

- Общедоступная часть, которая позволяет заказчикам заказывать туры и пользоваться личным кабинетом;
- Административная часть, которая позволяет CRM-менеджерам заполнять и редактировать базу данных сайта, а также отслеживать потенциальных клиентов и просматривать заказы клиентов.

Эти две части реализованы, следуя шаблону MVT. Примем, что MVT определяет способ разработки программного обеспечения, при котором код для определения и доступа к данным (модель, файл `models.py`) отделен от логики приложения (управление, файл `views.py`), которая в свою очередь отделена от интерфейса пользователя (представление, файл `html`).

При создании веб-проекта на Django происходит разделение на сам веб-проект и веб-приложение, которое создается после создания проекта.

3.3.3 Структура веб-проекта

В основу проекта входят:

- Внешний каталог `DjangoTravel/` - это просто контейнер для проекта. Его название никак не использует Django, и его можно переименовать во что угодно;
- Внутренний каталог `DjangoTravel/` - это пакет Python проекта. Его название - это название пакета Python, которое будет использоваться для импорта чего-либо из проекта.

Во внутреннем находятся каталоги:

1. Каталог `Travel` – это веб-приложение проекта;

2. Каталог `Media` – каталог, в котором находятся изображения, используемые в шаблонах веб-проекта;

3. Каталог `Static` – каталог, в котором находятся статические файлы и скрипты, используемые в веб-проекте.

- База данных `db.sqlite3` находящаяся в каталоге `DjangoTravel`, которая используется в проекте. В Django база данных создается после того как выполнится первая миграция в веб-приложении;

- Файл `manage.py`, находящийся в каталоге `DjangoTravel` – это скрипт, который позволяет взаимодействовать с проектом Django.

Также во внутреннем каталоге `DjangoTravel` создается каталог с одноименным названием содержащий файлы:

- Файл `settings.py` – это файл, содержащий конфигурации проекта;

Рассмотрим некоторые конфигурации:

1. `DATABASES` - словарь содержащий настройки для всех баз данных, которые будут использоваться Django. Словарь содержит псевдонимы используемых баз данных и словарь с настройками для каждой. Настройка `DATABASES` должна определять базу данных с псевдонимом `default` и любое количество дополнительных баз данных. В данной работе представлен самый простой вариант настройки это одна `SQLite` база данных.

`ENGINE` – Бэкенд базы данных (так как использовалась СУБД `SQLite`, то был взят бэкенд `django.db.backends.sqlite3`).

`NAME` – название базы данных. Значение по умолчанию, `os.path.join(BASE_DIR, 'db.sqlite3')`, создаст файл в каталоге проекта;

2. `INSTALLED_APPS` – Список строк, который указывает на все приложения Django, используемые в проекте.

По умолчанию, `INSTALLED_APPS` содержит следующие приложения, которые предоставляются Django:

1. `django.contrib.admin` – интерфейс администратора;
2. `django.contrib.auth` – система авторизации;
3. `django.contrib.contenttypes` – фреймворк типов данных;
4. `django.contrib.sessions` – фреймворк сессии;
5. `django.contrib.messages` – фреймворк сообщений;

6. `django.contrib.staticfiles` – фреймворк для работы со статическими файлами.

Также были добавлены: `Crispy_forms` – приложение необходимое для работы пакета `Crispy_forms`, используемого для стилизации форм в Django, и используемое приложение `Travel`;

3. `STATIC_URL` - URL, указывающий на каталог со статическими файлами `STATIC_ROOT`;

4. `STATIC_ROOT` - Абсолютный путь к каталогу, в который собираются все статические файлы;

5. `MEDIA_URL` – URL, указывающий на каталог с медиа-файлами `MEDIA_ROOT`;

6. `MEDIA_ROOT` – Абсолютный путь к каталогу, в который собираются все медиа-файлы;

- Файл `urls.py`, находящийся в каталоге `DjangoTravel` – конфигурация URL-ов проекта Django. Это «содержание» всех Django-сайтов. Для определения URL-ов приложения, используется модуль Python неофициально названный `URLconf` (конфигурация URL-ов). Этот модуль содержит код Python, который отображает URL-шаблоны (регулярные выражения) и связанные функции Python.

Для добавления URL-ов из модуля `django.urls` берутся атрибуты `url` и `include`;

- Файл `wsgi.py`, находящийся в каталоге `DjangoTravel` – это точка входа проекта для WSGI – совместных веб-серверов.

3.3.4 Структура веб-приложения

В веб-приложение входят:

- Каталог `Travel` – это каталог, в котором находятся файлы и каталоги приложения. В каталог `Travel` входит каталог `Migrations`. Это каталог, который содержит все файлы миграций. Django создаст миграции при любых изменениях модели или полей – даже тех параметров, которые не влияют на базу данных – так как единственный способ восстановить состояние моделей – это хранить все изменения в истории. «Начальные миграции» приложения – это

миграции, которые создают первую версию таблиц для приложения. Обычно приложение содержит только одну начальную миграцию, но для сложной структуры моделей их может быть несколько. «Начальные миграции» помечаются атрибутом `initial = True` в классе миграции. Если атрибут `initial` не указан, миграция будет считаться «начальной», если это первая миграция в приложении (то есть она не зависит от другой миграции текущего приложения).

Также в каталог `Travel` входит каталог `templates` в котором хранятся шаблоны веб-приложения;

- `models.py` - файл отвечающий за модели в Django. Модели отображают информацию о данных. Они содержат поля и поведение данных. Обычно одна модель представляет одну таблицу в базе данных.

В данной работе было сделано шесть моделей с названиями `Category`, `Country`, `Tour`, `CartItem`, `Cart`, `Order`.

Каждая модель представлена классом, унаследованным от `django.db.models.Model`. Каждая модель содержит несколько атрибутов, каждый из которых отображает поле в таблице базы данных.

Каждое поле представлено экземпляром класса `Field` – например, `CharField` для текстовых полей. Это указывает Django какие типы данных хранят эти поля.

Названия каждого экземпляра `Field` - это название поля, в “машинном” (`machine-friendly`) формате. Эти названия будут использоваться в коде, а база данных будет использовать их как названия колонок.

Некоторые классы, унаследованные от `Field`, имеют обязательные аргументы. Например, `CharField` требует, чтобы ему передали `max_length`. Это используется не только в схеме базы данных, но и при валидации. Также в моделях присутствует необязательный аргумент `verbose_name = ""` - это читабельное название модели, в единственном числе. Если этот параметр не указан, то в административной части отобразится имя поля. Дополнительные настройки для модели можно определить через класс `Meta`, например название в единственной и множественной форме (`verbose_name` и `verbose_name_plural`). Все они не обязательны и добавлять класс `Meta` тоже не обязательно.

Также в некоторых моделях представлены два метода `__str__` и `get_absolute_url`. Метод `__str__` определяет название модели в административной части сайта. В данном примере в качестве названия берется название категории тура. Метод `get_absolute_url()` необходим, чтобы указать Django как вычислить URL для объекта. Метод должен вернуть строку, которая может быть использована в HTTP запросе, для некоторых моделей применяется функция `pre_save_category_slug` необходимая для автоматического заполнения поля `slug`, функция `image_folder`, необходимая для переименования картинок (в качестве нового названия берется `slug` модели), функция `add_to_cart`, необходимая для добавления заказов в базу данных, функция `remove_to_cart`, необходимая для удаления заказов из базы данных, функция `change_qty`, необходимая для изменения количества человек, которые оправляются в поездку, функция `image_folder`, необходимая для переименования картинок и класс `TourManager`, предназначенный для переопределения менеджера `objects.all()`;

- `admin.py` – файл необходимый для использования моделей в административной части проекта. Код файла `admin.py` указан на рисунке 3.

```
# -*- coding: utf-8 -*-
from __future__ import unicode_literals

from django.contrib import admin
from ecomapp.models import Category, Country, Tour, Order

admin.site.register(Category)
admin.site.register(Country)
admin.site.register(Tour)
admin.site.register(Order)
```

Рисунок 3 – Код файла `admin.py`

В файл импортируются все используемые модели и с помощью команды `admin.site.register` регистрируются для отображения в административной части;

- `forms.py` – файл необходимый для работы с формами в общедоступной части. В файле описаны формы для регистрации, входа на сайт и форма для заказа тура;

- `views.py` – файл который отвечает за отображение общедоступной части. По сути это логика приложения. Отображение является функцией,

которая обрабатывает HTTP-запрос, получает данные из базы данных, которые применяются для генерации страницы HTML. Затем функция отображения возвращает сгенерированную страницу пользователю в виде HTTP-ответа;

- `urls.py` – файл отвечающий за связь с файлом `urls.py`, находящимся в основе проекта. Функция `url()` определяет URL-паттерн и функцию отображения, которая будет вызвана, если введенный адрес будет соответствовать данному паттерну.

Данная функция `url()`, кроме того, определяет параметр `name`, который уникально определяет это частное URL-преобразование.

- `apps.py` – файл который определяет конфигурацию приложения. Код файла `apps.py` представлен на рисунке 4.

```
# -*- coding: utf-8 -*-
from __future__ import unicode_literals

from django.apps import AppConfig

class TravelConfig(AppConfig):
    name = 'Travel'
```

Рисунок 4 – Код файла `apps.py`

Django содержит реестр установленных приложений, который содержит текущие настройки и предоставляет интроспекцию. Также предоставляет список доступных моделей.

Этот реестр называется просто `apps` и находится в модуле `django.apps`.

Полная схема работы веб-приложения представлена на рисунке 5.

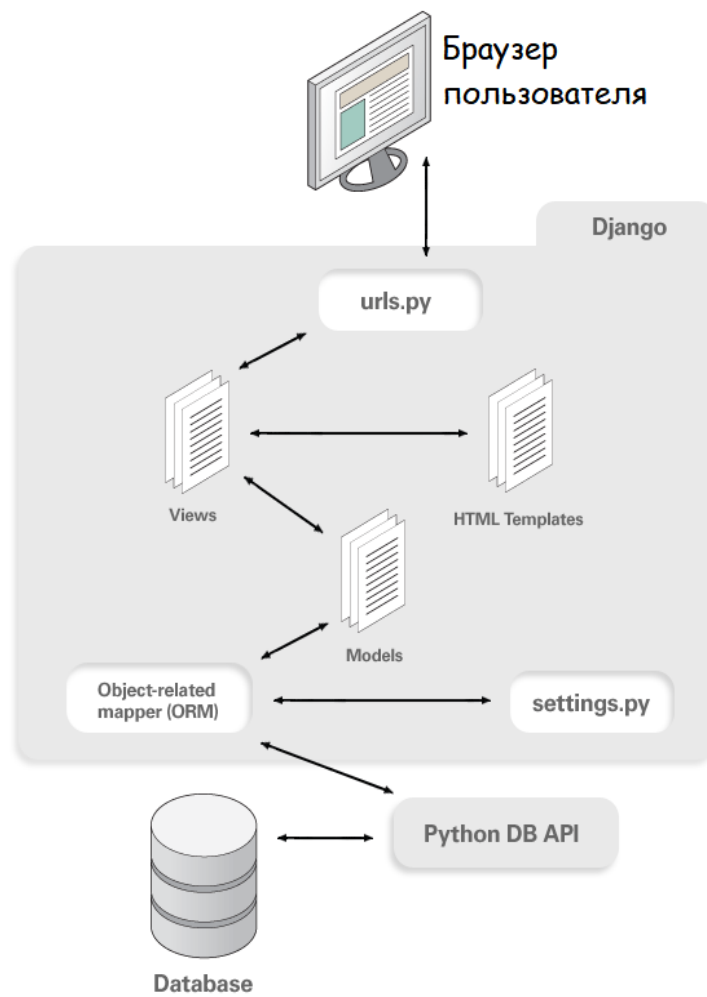


Рисунок 5 – Схема работы веб-приложения

3.4 Описание функциональности

3.4.1 Общедоступная часть

При входе на сайт заказчик видит его главную страницу, на которой расположены категории туров и туры. Также в верхней правой части главной страницы расположено приветствие «Здравствуйте, гость» и ссылки на форму входа и регистрации. В верхней левой части расположено количество заказов. Вид главной странице показан на рисунке 6.

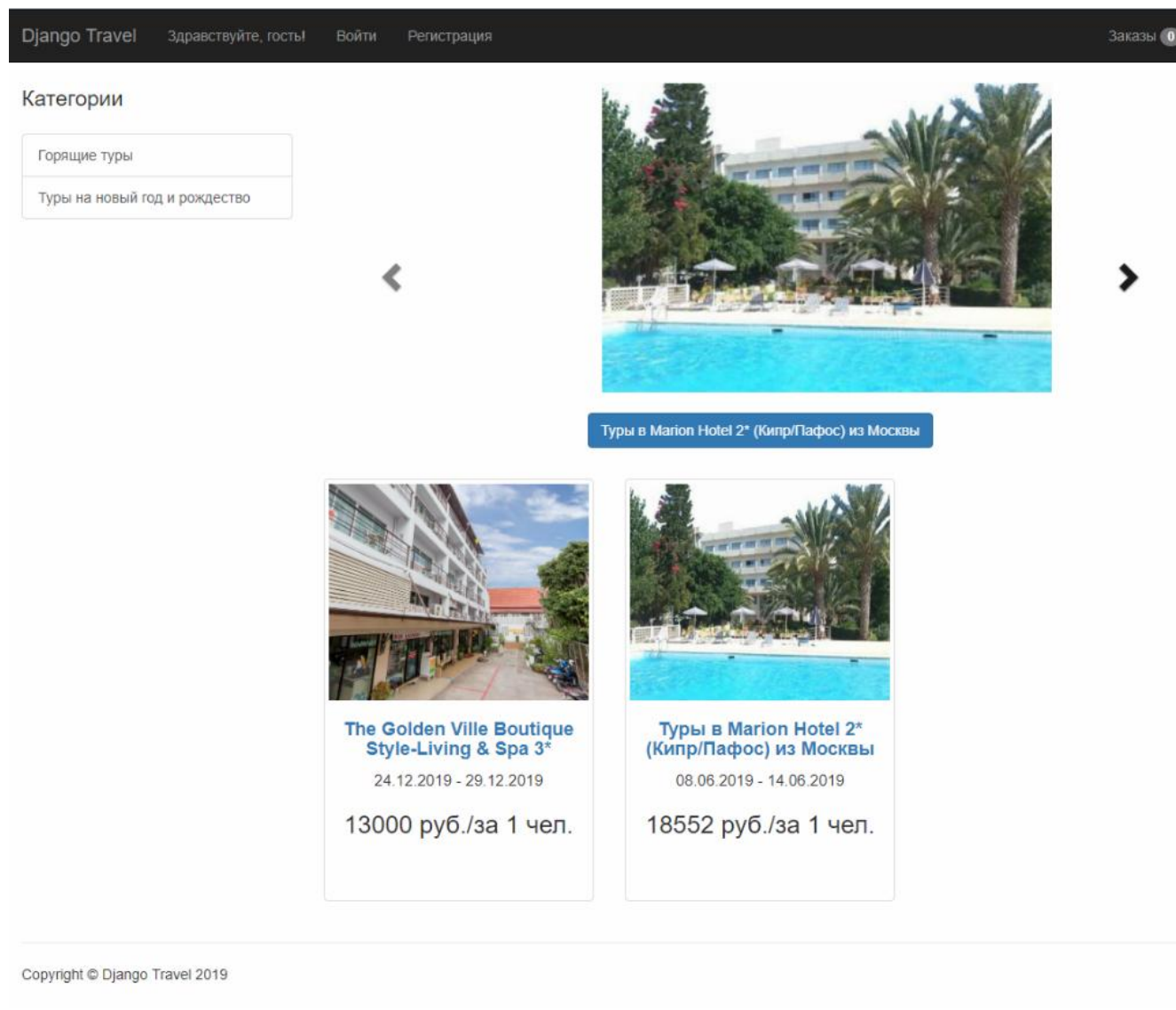


Рисунок 6 – Главная страница

Так как заказчик не зарегистрирован в системе, он не может заказать тур, поэтому кнопка заказать не отображается. Информация о каждом туре на главной странице минимальна. Для подробной информации о туре необходимо

перейти по ссылке на этот тур. Перейдя по ссылке, заказчик увидит подробную информацию о туре (рисунок 7).



The Golden Ville Boutique Style-Living & Spa 3*

Таиланд, Паттайа

Отель: The Golden Ville Boutique Style-Living & Spa 3*

Откуда:

Москва

Куда:

Паттайа

Начало тура:

24 декабря 2019 г.

Возвращение:

29 декабря 2019 г.

Тип размещения:

SNGL

Тип питания:

RO

Цена за одного человека: **13000 руб.**

Заказывать товар могут только авторизованные пользователи. Пожалуйста, [войдите](#) или [зарегистрируйтесь](#)

Рисунок 7 – Подробная информация о туре

В подробной информации заказчик видит:

- Название тура;
- Страну;
- Город;
- Город отбытия;
- Город прибытия;
- Дату начала тура;
- Дату возвращения;
- Тип размещения;
- Тип питания;
- Цену за одного человека.

Кнопка «Заказать» отсутствует. Вместо неё появляется сообщение о том, что заказать тур могут только авторизованные пользователи, и предложение войти или зарегистрироваться. Вернутся на главную страницу можно нажав на кнопку «Django Travel», расположенную сверху слева.

На главной странице заказчик может нажать на нужную ему категорию тура. Тогда отобразится другой шаблон, в котором заказчик увидит только

нужную ему категорию. Для примера выберем категорию «Туры на новый год и рождество» (рисунок 8).

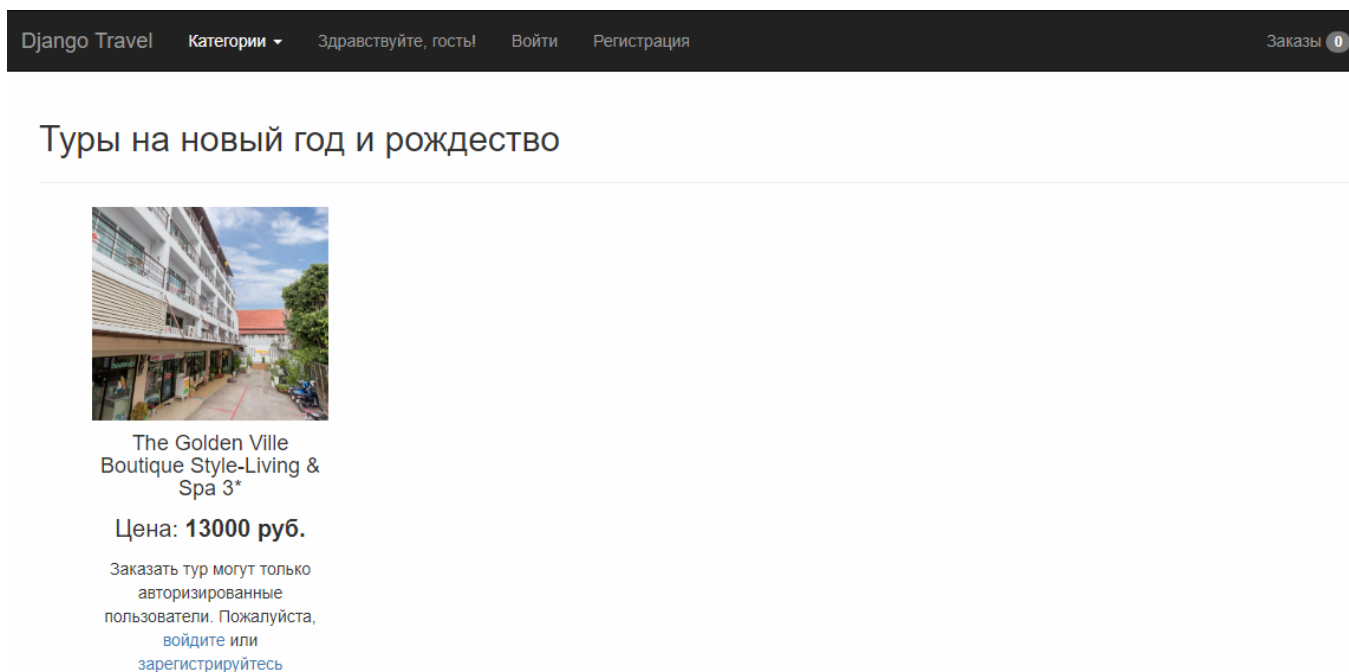


Рисунок 8 – Пример выбора категории «Туры на новый год и рождество»

Кнопка «Заказать» также отсутствует.

Перейдя по ссылке войти, заказчик попадёт на страницу входа (рисунок 9).

Авторизация

Рисунок 9 – Форма входа

Для входа заказчику необходимо ввести логин и пароль. Если заказчика с введённым логином не существует, то появится соответствующее уведомление (рисунок 10).

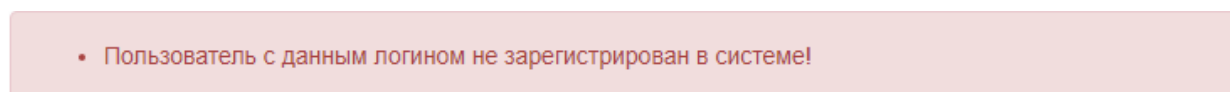


Рисунок 10 – Ошибка входа

Если заказчик неверно введёт свой пароль, появится соответствующее уведомление (рисунок 11).

Рисунок 11 – Уведомление о неверном пароле.

Перейдя по ссылке «Зарегистрируйтесь», заказчик попадёт на страницу регистрации (рисунок 12).

Регистрация нового пользователя

Логин*

Обязательное поле. Не более 150 символов. Только буквы, цифры и символы @/./+/_/.

Пароль*

Придумайте пароль

Повторите пароль*

Имя

Фамилия

Ваша почта

Пожалуйста, указывайте реальный адрес

Зарегистрироваться

Рисунок 12 – Форма регистрации

Для регистрации заказчику необходимо ввести:

- Логин;
- Пароль;
- Имя;
- Фамилию;
- Адрес электронной почты.

Все поля проверяются на правильность заполнения. Обязательными поля указаны со звёздочкой. Минимальная длина пароля, предусмотренная в Django 8 символов, при этом символы не должны состоять только из букв или цифр.

Также присутствует проверки полей «Логин», «Ваша почта». Если заказчик попытается зарегистрироваться под логином или почтой, которые уже есть в базе данных то появится уведомление о том, что: «Пользователь с данным логином или почтовым адресом уже зарегистрирован в системе!» (рисунок 13).

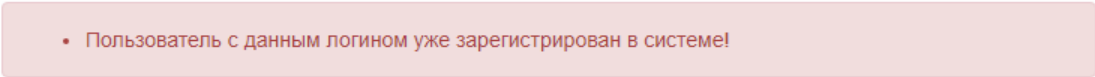
- 
- Пользователь с данным логином уже зарегистрирован в системе!

Рисунок 13 – Уведомление о зарегистрированном пользователе

Если пароли не совпадают, то появится о том, что: «Ваши пароли не совпадают! Попробуйте снова!» (рисунок 14).

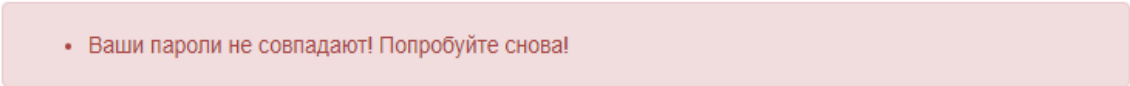
- 
- Ваши пароли не совпадают! Попробуйте снова!

Рисунок 14 – Уведомление о несовпадающих паролях

Когда заказчик успешно регистрируется на сайте или войдет на сайт, то его перебросит на главную страницу, где будут доступны кнопки заказать и личный кабинет заказчика, в верхней правой части изменится приветствие, появится кнопка выйти (рисунок 15). Зарегистрированный заказчик становится лидером.

Категории

Горящие туры

Туры на новый год и рождество



Туры в Marion Hotel 2* (Кипр/Пафос) из Москвы

**The Golden Ville Boutique
Style-Living & Spa 3***

24.12.2019 - 29.12.2019

13000 руб./за 1 чел.

Заказать

**Туры в Marion Hotel 2*
(Кипр/Пафос) из Москвы**

08.06.2019 - 14.06.2019

18552 руб./за 1 чел.

Заказать

Copyright © Django Travel 2019

Рисунок 15 – Обновлённая главная страница

Для каждого заказчика создается новая «корзина» заказов.

Нажав на кнопку «Заказать» изменится количество заказов в верхнем правом углу.

Для перехода на страницу заказа нужно нажать на кнопку «Заказы».

Если заказчик заказал тур то, на странице заказов отобразится таблица заказов, где указаны название тура, количество человек, отправляющихся в поездку и цена, кнопка «Убрать из заказов» и кнопка «Предварительный заказ» (рисунок 16).

Ваши заказы			
Тип	Кол-во чел.	Цена	
The Golden Ville Boutique Style-Living & Spa 3*	<input type="text" value="1"/>	13000,00 руб.	<button>Убрать из заказов</button>
Итого:		13000 руб.	

Рисунок 16 – Заказ пользователя

Если заказчик укажет количество человек больше одно, то цена изменится. Цена меняется динамически посредством выполнения скрипта. Скрипт написан при помощи JQuery. Код скрипта показан на рисунке 17.

```
$(document).ready(function() {
    $('.cart-item-qty').on('click', function() {
        qty = $(this).val()
        item_id = $(this).attr('data-id')
        data = {
            qty: qty,
            item_id: item_id
        }
        $.ajax({
            type: "GET",
            url: '{% url "change_item_qty" %}',
            data: data,
            success: function(data) {
                $('#cart-item-total-'+item_id).html(parseFloat(data.item_total).toFixed(2) + ' руб.')
                $('#cart-total-price').html('<strong>' + parseFloat(data.cart_total_price).toFixed(2) + ' руб. </strong>')
            }
        })
    })
})
```

Рисунок 17 – Скрипт изменения цены

Если заказчик захочет убрать данный тур из заказов, то ему необходимо нажать на кнопку «Убрать из заказов». Тур убирается динамически посредством выполнения скрипта, написанного при помощи JQuery (рисунок 18).

```

$(document).ready(function() {
    $('#cart-empty').css('display', 'none')
    $('#remove_from_cart').on('click', function(e) {
        e.preventDefault()
        tour_slug = $(this).attr('data-slug')
        item_product_id = $(this).attr('data-id')
        data = {
            tour_slug: product_slug
        }

        $.ajax({
            type: "GET",
            url: '{% url "remove_from_cart" %}',
            data: data,
            success: function(data) {
                $('#cart_count').html(data.cart_total)
                $('#cart-item-'+item_product_id).css('display', 'none')
                $('#cart-total-price').html('<strong>' + parseFloat(data.cart_total_price).toFixed(2) + ' руб. </strong>')
                if(parseInt(data.cart_total) == 0) {
                    $('#my-cart').css('display', 'none')
                    $('#cart-empty').css('display', 'block')
                    $('#cart-empty').html('<h3 class="text-center">Ваша корзина пуста</h3>')
                }
            }
        })
    })
})

```

Рисунок 18 – Скрипт удаления тура из заказов

Нажав на кнопку «Предварительный заказ», заказчик перейдет на страницу предварительного заказа, где будут показаны его предварительные заказы в виде таблицы где указаны название тура, количество человек, отправляющихся в поездку и цена тура (рисунок 19).

Ваш предварительный заказ		
Тип	Кол-во чел	Цена
The Golden Ville Boutique Style-Living & Spa 3*	1	13000,00 руб.
Итого:		13000 руб.
<div>Оформить заказ</div>		

Рисунок 19 – Предварительный заказ

Нажав на кнопку «Оформить заказ», заказчик перейдет на страницу оформления заказа (рисунок 20), где необходимо указать:

- Имя;
- Фамилию;
- Контактный телефон.

Поле «Комментарии к заказу» не является обязательным к заполнению, и заказчик может ничего там не писать.

Имя*	Фамилия	Контактный телефон*
<input type="text" value="Андрей"/>	<input type="text" value="Галяткин"/>	<input type="text" value="+79999999999"/>
Пожалуйста, указывайте реальный номер телефона, по которому с Вами можно связаться		

Комментарии к заказу

Комментарий к заказу.

Оформить заказ

Рисунок 20 – Оформление заказа

Заполнив указанную форму и нажав на кнопку «Оформить заказ», заказчик увидит уведомление о том, что заказ принят в обработку (рисунок 21).

Спасибо за Ваш заказ! Он принят в обработку. В ближайшее время менеджер с Вами свяжется

Рисунок 21 – Уведомление об обработки заказа

Перейдя в личный кабинет (рисунок 22) в верхней части сайта, заказчик увидит таблицу заказов, в которой указаны номер заказа, дата заказа, стоимость заказа и детали заказа.

Личный кабинет пользователя User1			
Ваши заказы			
Номер заказа	Дата заказа	Стоимость заказа	Детали заказа
6	6 июня 2019 г. 18:42	13000,00 руб.	<div>Детали заказа</div>

Рисунок 22 – Личный кабинет

Для отображения деталей заказа, заказчик должен нажать на кнопку «Детали заказа». После нажатия, откроется форма, где указан номер заказа, а также в таблицах указаны название тура, количество человек, стоимость тура, имя заказчика, фамилия заказчика, номер телефона заказчика и комментарии к заказу.

Тип

Тип в The Golden Ville Boutique Style-Living & Spa 3*	1 чел.	13000,00 руб.
Итого		13000 руб.

Заказчик

Имя: Андрей

Фамилия: Галяткин

Телефон: +79999999999

Комментарии к заказу

Комментарий к заказу.

Заккрыть

Рисунок 23 – Форма деталей заказа

3.4.2 Административная часть

Для входа в административную часть необходимо перейти на страницу администратора. Переход на страницу администратора осуществляется открытием ссылки /admin. После чего, вводится «имя пользователя» и «пароль» в поля имя пользователя и пароль соответственно. Необходимые «имя пользователя» и «пароль» были созданы при выполнении команды `python manage.py createsuperuser`. Страница входа показана на рисунке 24.

Администрирование Django

Имя пользователя:

Admin

Пароль:

.....

Войти

Рисунок 24 – Вход в административную часть

Если ввести некорректные данные, то появится сообщение об ошибке (рисунок 25).

Пожалуйста, введите корректные имя пользователя и пароль учётной записи. Оба поля могут быть чувствительны к регистру.

Рисунок 25 – Сообщение об ошибке входа в административную часть

После того как были введены корректные данные необходимо нажать на кнопку Войти. После этого CRM-менеджер увидит стартовый интерфейс Администрирования Django (рисунок 26).

TRAVEL		
Элементы заказов	+ Добавить	Изменить
Корзины	+ Добавить	Изменить
Категории	+ Добавить	Изменить
Страны	+ Добавить	Изменить
Заказы	+ Добавить	Изменить
Туры	+ Добавить	Изменить
NOTIFICATIONS		
Notifications	+ Добавить	Изменить
ПОЛЬЗОВАТЕЛИ И ГРУППЫ		
Группы	+ Добавить	Изменить
Пользователи	+ Добавить	Изменить

Рисунок 26 – Стартовый интерфейс административной части

В этой части сайта отображаются все модели, сгруппированные по установленному приложению. CRM-менеджер может кликнуть на названии модели, чтобы получить список всех связанных записей, далее может кликнуть на этих записях, для их редактирования. Также можно непосредственно кликнуть на ссылку «Добавить», расположенную рядом с каждой моделью, чтобы начать создание записи этого типа. Добавить новую запись можно нажав на кнопку «Добавить +», находящуюся на каждой из последующих страниц.

Для просмотра лидов CRM-менеджер должен кликнуть на название модели «Пользователи». После этого CRM-менеджера перебросит на страницу (рисунок 27), где в таблице показаны:

- Логин;
- Адрес электронной почты;
- Имя;
- Фамилия;
- CRM-менеджер.

<input type="checkbox"/>	ЛОГИН	▲ АДРЕС ЭЛЕКТРОННОЙ ПОЧТЫ	ИМЯ	ФАМИЛИЯ	CRM-МЕНЕДЖЕР
<input type="checkbox"/>	ManagerCRM1				
<input type="checkbox"/>	User1	agn15061997@yandex.ru	Андрей	Галяткин	

Кликнув по ссылке «Страны», CRM-менеджер перейдет на страницу (рисунок 28) где увидит все имеющиеся записи стран.



Рисунок 28 – Записи стран

Нажав на название страны, CRM-менеджер может отредактировать запись (рисунок 29).

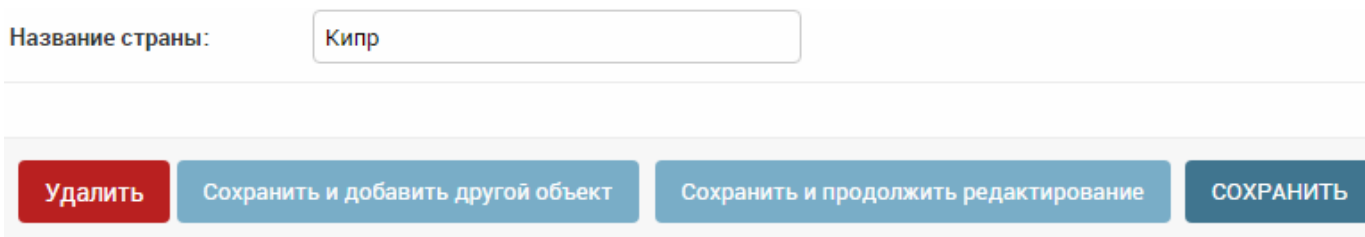


Рисунок 29 – Редактирование записи

Здесь же можно удалить запись, сохранить запись и продолжить редактирование или сохранить запись и добавить новую запись. При сохранении, в соответствующей таблице базы данных изменится название страны. Запись в базу данных происходит по соответствующей технологии.

Кликнув по ссылке «Категории», CRM-менеджер перейдет на страницу где увидит все имеющиеся записи категорий (рисунок 30).

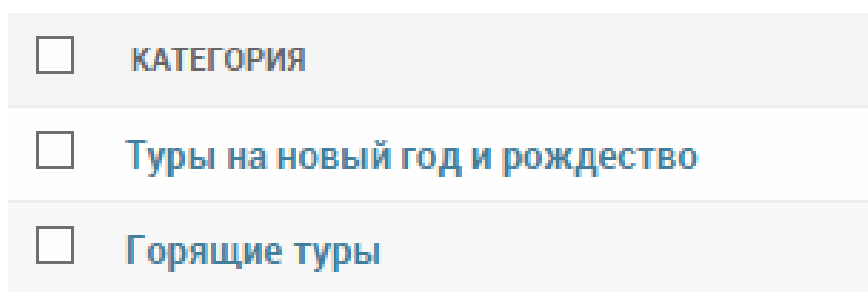


Рисунок 30 – Записи категорий

Также, как и на странице «Страны», здесь можно изменить, сохранить или удалить запись (рисунок 31).

Название тура: Туры на новый год и рождество

Slug: туры-na-novyj-god-i-rozhdestvo

Удалить Сохранить и добавить другой объект Сохранить и продолжить редактирование СОХРАНИТЬ

Рисунок 31 –Редактирование категорий

Кликнув по ссылке «Туры», CRM-менеджер перейдет на страницу где увидит все имеющиеся у него записи касающиеся туров.

<input type="checkbox"/>	ТУР
<input type="checkbox"/>	Тур в Marion Hotel 2* (Кипр/Пафос)
<input type="checkbox"/>	Тур в The Golden Ville Boutique Style-Living & Spa 3*

Рисунок 32 – Записи туров

На странице тура (рисунок 33) содержатся следующие записи:

- Категория. Данное поле представлено в model.py как поле ForeignKey и поэтому CRM-менеджер при заполнении записи тура просто выбирает из добавленных им категорий;
- Страна. Данное поле представлено в базе данных также, как и поле Категория;
- Название тура;
- Slug;
- Город;
- Отель;
- Город отправления;
- Город прибытия;
- Начало тура;
- Конец тура;
- Тип размещения. В models.py для данного поля предусмотрен заготовленный список. Поэтому CRM-менеджер просто выбирает значение в данном поле;
- Тип питания. Представлено также, как и поле Тип размещения;

- Картинка. В models.py поле представлено как ImageField и имеет необязательный параметр upload_to, который загружает изображение в выбранную папку для последующего отображения в шаблоне. По-умолчанию, экземпляр ImageField создается как колонка varchar в базе данных.

Здесь можно также редактировать и удалить запись.

Категория:	Туры на новый год и рождество	✎ +
Страна:	Таиланд	✎ +
Название тура:	Тур в The Golden Ville Boutique Style-Living	
Slug:	The-Golden-Ville-Boutique-Style-Living-Anc	
Город:	Паттайа	
Отель:	The Golden Ville Boutique Style-Living & Spa	
Город отправления:	Москва	
Город прибытия:	Паттайа	
Цена:	13000 <small>Цена указана за одного человека</small>	
Начало тура:	24.12.2019	Сегодня 📅
Конец тура:	29.12.2019	Сегодня 📅
Тип размещения:	SNGL	
Тип питания:	RO	
Картинка:	На данный момент: The-Golden-Ville-Boutique-Style-Living-And-Spa/The-Golden-Ville-Boutique-Style-Living-And-Spa.jpg Изменить: <input type="button" value="Выберите файл"/> Файл не выбран	

Рисунок 33 – Редактирование туров

Кликнув по ссылке «Заказы», CRM-менеджер перейдет на страницу где увидит все имеющиеся у него записи касающиеся заказов заказчиков (рисунок 34).

<input type="checkbox"/>	ЗАКАЗ
<input type="checkbox"/>	Заказ №6

Нажав на ссылку заказа CRM-менеджер, увидит данные заказа и заказчика (рисунок 35), а именно:

- Логин заказчика, сделавшего заказ;
- Название заказа;
- Итоговую стоимость заказа;
- Имя заказчика;
- Фамилию заказчика;
- Номер телефона для связи с заказчиком;
- Комментарии к заказу.

Как и на других страницах поля можно редактировать и удалить запись полностью.

Логин:	User1	✎ +
Заказ:	Тур в The Golden Ville Boutique Style-Living & Spa 3*	✎ +
Стоимость заказа:	13000,00	
Имя заказчика:	Андрей	
Фамилия заказчика:	Галяткин	
Телефон заказчика:	+79999999999	
Комментарии к заказу:	<div>Комментарий к заказу.</div>	

Удалить

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ

Рисунок 35 – Данные заказчика

ЗАКЛЮЧЕНИЕ

Внедрение CRM-системы является одним из приоритетных направлений развития любой компании, поскольку сможет обеспечить повышение качества обслуживания клиентов, уменьшить трудозатраты на сопровождение и освободить сотрудников от рутинной работы.

CRM-системы автоматизирует процессы взаимоотношений с клиентами, тем самым помогая внедрять, измерять и контролировать внутренние стандарты и методики работы с клиентами.

Каждая CRM-система - это воплощение видения разработчиков того, как нужно работать с клиентом. В ней заложено множество готовых инструментов, которые позволяют перевести работу на качественно новый уровень.

Руководителю не на что опираться в этом вопросе, а потому и отдел продаж часто работает далеко не лучшим образом. Внедрение CRM-системы позволяет получить не только инструмент, но и помощь, взгляд разработчиков на то, как должен работать отдел продаж. В свою очередь при разработке CRM-системы обычно опираются на лучшие практики, на экспертов в вопросах работы с клиентами.

Целью написания данной бакалаврской работы являлась разработка CRM-системы для турагентства и личного кабинета с помощью фреймворка Django. Самостоятельное использование этой CRM-системы значительно облегчает работу CRM-менеджеров.

В ходе разработки была изучена предметная область, проанализированы требования к веб-приложению. Был спроектирован и разработан сайт с личным кабинетом для заказчиков в общедоступной части веб-проекта, а также была спроектирована база данных и разработан интерфейс для административной части веб-проекта.

Планируется дальнейшее развитие проекта, в частности, добавление новых категорий и туров в базу данных, будет улучшена форма регистрации и входа. Всё это и не только возможно благодаря большим мощностям

фреймворка Django. Следовательно, приложение не утратит актуальности и будет в дальнейшем гораздо полезнее.

В ходе разработки мною был изучен большой объем документации по технологиям фреймворка Django. Получены навыки разработки веб-приложений с помощью данного фреймворка и верстки веб-страниц с использованием современных технологий.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гринберг П. CRM со скоростью света / П. Гринберг – СПб.: Символ Плюс, 2007. – 528 с.
2. Кривошеева Р. Что такое CRM-системы и как их правильно выбирать? / Хабр [Электронный ресурс] / Р. Кривошеева // Habrahabr: Habrahabr.ru. – [Б. м. : б. и.], 2015. – Режим доступа: <https://habr.com/post/249633/>. – Загл. с экрана.
3. Головатый А. Django. Подробное руководство /А. Головатый, Дж. Каплан-Мосс – СПб.: Символ-Плюс, 2010. – 560 с.
4. Чан У. Django. Разработка веб-приложений на Python/ У. Чан, П. Биссекс, Д. Форсье – СПб.: Символ-Плюс, 2009. – 456 с.
5. Фримен А. jQuery для профессионалов/ А. Фримен – М.: «Вильямс», 2012. — 960 с.
6. Самков Г. jQuery. Сборник рецептов / Г. Самков – СПб.: БХВ-Петербург, 2010. – С. 416.
7. Шевчук А. jQuery для начинающих / Хабр [Электронный ресурс] / А. Шевчук // Habrahabr: Habrahabr.ru. – [Б. м. : б. и.], 2008. – Режим доступа: <https://habr.com/post/38208/>. – Загл. с экрана.
8. Bootstrap 3. Документация на русском языке [Электронный ресурс]: документация для Bootstrap 3. – [Б. м. : б. и.], [2018]. – Режим доступа: <http://bootstrap-3.ru/index.php>. – Загл. с экрана.
9. Кузнецов М. Как сверстать веб-страницу. Часть 2 — Bootstrap/ Хабр [Электронный ресурс] / М. Кузнецов // Habrahabr: Habrahabr.ru. – [Б. м. : б. и.], 2014. – Режим доступа: <https://habr.com/post/211032/>. – Загл. с экрана.
10. Документация Django [Электронный ресурс]: Документация Django. – [Б. м. : б. и.], [2018]. – Режим доступа: <http://www.djbook.ru>. – Загл. с экрана.

ПРИЛОЖЕНИЕ А

(обязательное)

Реализация входа и регистрации потенциальных заказчиков на сайте.

#Код из файла views.py

#Функция регистрации

```
def registration_view(request):
    form = RegistrationForm(request.POST or None)
    categories = Category.objects.all()
    if form.is_valid():
        new_user = form.save(commit=False)
        username = form.cleaned_data['username']
        password = form.cleaned_data['password']
        email = form.cleaned_data['email']
        first_name = form.cleaned_data['first_name']
        last_name = form.cleaned_data['last_name']
        new_user.username = username
        new_user.set_password(password)
        new_user.first_name = first_name
        new_user.last_name = last_name
        new_user.email = email
        new_user.save()
        login_user = authenticate(username=username, password=password)
        if login_user:
            login(request, login_user)
            return HttpResponseRedirect(reverse('base'))
    context = {
        'form': form,
        'categories': categories
    }
    return render(request, 'registration.html', context)
```

Продолжение ПРИЛОЖЕНИЯ А

#Функция входа

```
def login_view(request):
    form = LoginForm(request.POST or None)
    categories = Category.objects.all()
    if form.is_valid():
        username = form.cleaned_data['username']
        password = form.cleaned_data['password']
        login_user = authenticate(username=username, password=password)
        if login_user:
            login(request, login_user)
            return HttpResponseRedirect(reverse('base'))
    context = {
        'form': form,
        'categories': categories
    }
    return render(request, 'login.html', context)
```

#Код из файла forms.py

```
from django import forms
from django.utils import timezone
from django.contrib.auth.models import User
```

#Форма входа

```
class LoginForm(forms.Form):

    username = forms.CharField()
    password = forms.CharField(widget=forms.PasswordInput)

    def __init__(self, *args, **kwargs):
        super(LoginForm, self).__init__(*args, **kwargs)
        self.fields['username'].label = 'Логин'
        self.fields['password'].label = 'Пароль'
```

Продолжение ПРИЛОЖЕНИЯ А

```
def clean(self):
    username = self.cleaned_data['username']
    password = self.cleaned_data['password']
    if not User.objects.filter(username=username).exists():
        raise forms.ValidationError('Пользователь с данным логином не
зарегистрирован в системе!')
    user = User.objects.get(username=username)
    if user and not user.check_password(password):
        raise forms.ValidationError('Неверный пароль!')

#Форма регистрации

class RegistrationForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput)
    password_check = forms.CharField(widget=forms.PasswordInput)

    class Meta:
        model = User
        fields = [
            'username',
            'password',
            'password_check',
            'first_name',
            'last_name',
            'email'
        ]

    def __init__(self, *args, **kwargs):
        super(RegistrationForm, self).__init__(*args, **kwargs)
        self.fields['username'].label = 'Логин'
        self.fields['password'].label = 'Пароль'
        self.fields['password'].help_text = 'Придумайте пароль'
        self.fields['password_check'].label = 'Повторите пароль'
        self.fields['first_name'].label = 'Имя'
        self.fields['last_name'].label = 'Фамилия'
```

```

self.fields['email'].label = 'Ваша почта'
self.fields['email'].help_text = 'Пожалуйста, указывайте реальный адрес'

```

```

def clean(self):
    username = self.cleaned_data['username']
    password = self.cleaned_data['password']
    password_check = self.cleaned_data['password_check']
    email = self.cleaned_data['email']
    if User.objects.filter(username=username).exists():
        raise forms.ValidationError('Пользователь с данным логином уже
зарегистрирован в системе!')
    if User.objects.filter(email=email).exists():
        raise forms.ValidationError('Пользователь с данным почтовым
адресом уже зарегистрирован!')
    if password != password_check:
        raise forms.ValidationError('Ваши пароли не совпадают! Попробуйте
снова!')

```

#Код шаблона входа login.html

```

{% extends 'base.html' %}
{% load static %}
{% load crispy_forms_tags %}
{% block content %}
<h3 class='text-center'>Авторизация</h3>
<hr>
<div class='col-sm-8 col-sm-offset-2'>
<form action="" method='POST'>
    {% csrf_token %}
    {{ form|crispy }}
    <input type='submit' class='btn btn-success btn-block' value="Войти в
систему">
</form>
</div>

```

Продолжение ПРИЛОЖЕНИЯ А

```
{% endblock content % }
```

```
#Код шаблона регистрации registration.html
```

```
{% extends 'base.html' % }
```

```
{% load static % }
```

```
{% load crispy_forms_tags % }
```

```
{% block content % }
```

```
<h3 class='text-center'>Регистрация нового пользователя</h3>
```

```
<div class='col-sm-8 col-sm-offset-2'>
```

```
<hr>
```

```
<form action="" method="POST">
```

```
    {% csrf_token % }
```

```
    {{ form|crispy }}
```

```
    <input type='submit' class='btn btn-success btn-block'
value='Зарегистрироваться'>
```

```
</form>
```

```
</div>
```

```
{% endblock content % }
```

О Т З Ы В

на выпускную квалификационную работу
студента 4 курса направления подготовки 02.03.02

«Фундаментальная информатика и информационные технологии»

Галяткина Андрея Николаевича на тему

«Разработка CRM системы и личного кабинета
с помощью фреймворка DJANGO»

В современных условиях конкуренция на всех уровнях рынка весьма высока, что заставляет компании бороться не только за привлечение новых клиентов, но и за удержание уже существующих. Для этого компаниям необходимо гибко учитывать интересы и пожелания клиентов. Такой подход к ведению бизнеса называется клиентоориентированным. При большой клиентской базе учет интересов каждого клиента является труднодостижимой целью. Выходом из данной ситуации может быть внедрение в компании CRM-систем (Customer Relationship Management или системы управления взаимоотношениями с клиентами).

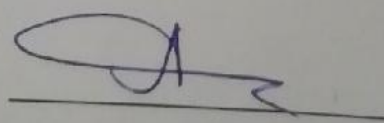
Целью настоящей бакалаврской работы является разработка CRM-системы для туристического агентства с помощью фреймворка Django. Предполагается, что такая система облегчит не только работу CRM-менеджеров агентства, но и заказчиков, которые смогут легко заказывать туры на сайте и отслеживать информацию о заказах с помощью личного кабинета.

В ходе исследования автором была изучена предметная область, проанализированы требования, предъявляемые к современному веб-приложению. В результате автором был спроектирован и разработан сайт с личным кабинетом для заказчиков в общедоступной части веб-проекта, а также была спроектирована и реализована база данных и интерфейс для административной части веб-проекта.

Проделанное исследование и результаты практической части его работы характеризуют автора выпускной квалификационной работы как программиста высокой квалификации, владеющего методами и инструментами современного программирования и способного к реализации своих знаний в рамках реальных проектов, связанных с разработкой программного обеспечения.

Считаю, что настоящая работа удовлетворяет требованиям, предъявляемым к выпускным квалификационным работам, а её автор, Галяткин А.Н., заслуживает оценки «отлично».

Научный руководитель
зав. кафедрой
фундаментальной информатики,
доцент



А. Г. Смольянов

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.П. ОГАРЁВА»

ОТЗЫВ РЕЦЕНЗЕНТА
О ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Студента Галяткина Андрея Николаевича
Факультет Математики и информационных технологий
Кафедра Фундаментальной информатики и информационных технологий
Группа 402
Направление подготовки Фундаментальная информатика и информационные технологии
Квалификация (степень) бакалавр
Наименование темы: Разработка CRM системы для турагентства и личного кабинета с помощью фреймворка DJANGO
Рецензент Десяев Е.В., МГУ им. Н.П. Огарева, кандидат физико-математических наук, доцент

ОЦЕНКА ВЫПУСКНОЙ РАБОТЫ

№ п/п	Показатели		Оценка				
			5	4	3	2	0*
1.	Актуальность тематики работы		+				
2.	Степень полноты обзора состояния вопроса и корректность постановки задачи		+				
3.	Уровень и корректность использования в работе методов исследования, математического моделирования			+			
4.	Степень комплексности работы, применение в ней знаний естественнонаучных, социально-экономических, общепрофессиональных и специальных дисциплин			+			
5.	Ясность, четкость, последовательность и обоснованность изложения		+				
6.	Применение современного математического и программного обеспечения, компьютерных технологий в работе		+				
7.	Качество оформления (общий уровень грамотности, стиль изложения, качество иллюстраций, соответствие требованиям стандарта)		+				
8.	Оригинальность и новизна полученных результатов (научных, конструкторских и технологических решений)		+				
9.	Тип работы	фундаментальная с оригинальными результатами					
		реферативная					
		прикладная	+				
10.	Рекомендации	к опубликованию					
		к внедрению	+				
ИТОГОВАЯ ОЦЕНКА			ОТЛИЧНО				

* – не оценивается (трудно оценить)

Отмеченные достоинства: Бакалаврская работа А. Г. Галяткина посвящена реализации CRM-системы и личного кабинета для турагентства, разработанных при помощи фреймворка Django. Актуальность данной работы заключается в том, что веб-проект кроссплатформенный. Для работы веб-приложения достаточно установить Python версии 2.7 и Django 1.11. Для работы с базой данных веб-приложения не нужно устанавливать дополнительного ПО, что является несомненным плюсом данной работы.

Отмеченные недостатки: Приложение не размещено на хостинге. Привязанность к версии Python и Django.

Заключение: Считаю, что работа А.Г. Галяткина отвечает требованиям, предъявляемым к выпускным квалификационным работам и заслуживает оценки «отлично».

10 июня

2019 г.

Рецензент

Уб

(подпись)

Заявление о самостоятельном характере выполнения работы

Я, Галяткин Андрей Николаевич, обучающийся 4 курса направления подготовки 02.03.02 – Фундаментальная информатика и информационные технологии, заявляю, что в моей работе на тему Разработка компьютерной системы тестирования для оценки текущего уровня знаний студентов, представленной в Государственную экзаменационную комиссию для публичной защиты, не содержится элементов неправомерных заимствований.

Все прямые заимствования из печатных и электронных источников, а также ранее защищённых письменных работ, кандидатских и докторских диссертаций имеют соответствующие ссылки.

Я ознакомлен с действующим в Университете Положением о проверке работ, обучающихся ФГБОУ ВО «МГУ им. Н.П. Огарёва» на наличие заимствований, в соответствии с которым обнаружение неправомерных заимствований является основанием для отрицательного отзыва руководителя работы.

Подпись обучающегося

07.06.2019г.

Дата

Работа представлена для проверки в Системе «Антиплагиат.ВУЗ»

07.06.2019г.

Дата представления работы

подпись руководителя

ОТЧЕТ
о результатах проверки работы обучающегося
на наличие заимствований

Ф.И.О. автора работы: Галяткин Андрей Николаевич

Тема работы: Разработка CRM системы для турагентства и личного кабинета с помощью фреймворка DJANGO

Руководитель: Смольянов Андрей Григорьевич

Представленная работа прошла проверку на наличие заимствований в системе «Антиплагиат.ВУЗ» (или иной аналогичной системе, анализа текстов на наличие заимствований, выбранной Университетом)

Результаты автоматической проверки: оригинальность – 62,44%

цитирования – 1,31%

заимствования – 36,25%

Результаты анализа полного отчета на наличие заимствований:

правомерные заимствования: да, 36,25%

корректные цитирования: да, 1,31%

неправомерные заимствования: нет

признаки обхода системы: нет

Общее заключение об итоговой оригинальности работы и возможности ее допуска к защите:

Выпускная квалификационная работа Галяткина А.Н. Разработка CRM системы для турагентства и личного кабинета с помощью фреймворка DJANGO допускается к защите.

Руководитель

Должность канд. физ. – мат. наук, доц.



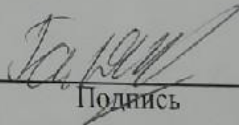
А.Г. Смольянов

Заведующему кафедрой
фундаментальной информатики
А. Г. Смольянову
студента 4 курса
очной формы обучения
на бесплатной основе
направления подготовки
02.03.02 – Фундаментальная
информатика и информационные
технологии факультета математики
и информационных технологий
ФГБОУ ВО «МГУ им. Н. П.
Огарёва»
Галяткина Андрея Николаевича

заявление.

Прошу разместить мою выпускную квалификационную работу на тему
«Разработка CRM системы для турагентства и личного кабинета с помощью
фреймворка DJANGO» в электронной библиотечной системе университета в
полном объеме.

07.06.19.
Дата


Подпись