

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	1
ВВЕДЕНИЕ	2
ГЛАВА 1. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	5
1.1. Формирование списка критерия сравнения существующих решений в области доставки еды и продуктов	5
1.2. Обзор существующих решений в области доставки еды и продуктов	7
1.3. Обзор языка и среды программирования	12
1.4. Выводы к главе 1	19
ГЛАВА 2. РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ РАЗМЕЩЕНИЯ ОБЪЯВЛЕНИЙ О КУЛИНАРНОЙ ПРОДУКЦИИ ДОМАШНЕГО ПРИГОТОВЛЕНИЯ	20
2.1. Разработки структуры сервиса	20
2.2. База данных и административная часть веб-сервиса для размещения объявлений о кулинарной продукции домашнего производства	22
2.3. Описание полученного web-приложения	28
2.3.1. Структура сервиса	28
2.3.2. Программно-аппаратные требования	30
2.3.3. Функциональные возможности	30
2.4. Выводы к главе 2	31
ЗАКЛЮЧЕНИЕ	32
СПИСОК ЛИТЕРАТУРЫ	33

ВВЕДЕНИЕ

В настоящее время онлайн-сервисы по доставке, распространению и популяризации готовой еды стали особенно популярны. Такие сервисы позволяют пользователям со всего мира заказывать, находить и приобретать желаемый продукт проще и эффективнее. Также активно развиваются сервисы, позволяющие размещать объявления по продажам продукции собственного производства для всеобщего публичного доступа.

Задача разрабатываемого сервиса объединить в себе задачи по распространению и приобретению пищевой продукции собственного производства.

Существует множество, как методологий, так и структур построения подобных сервисов, поэтому выбор зависит от конкретного разработчика и его предпочтений.

Приложение, разработанное в рамках приведенной работы, представляет собой веб-платформу, которая позволяет размещать объявления о кулинарной продукции.

Реализация основана на использовании технологии MVC, фреймворка Django и языка Python.

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. Это принцип разработки программного обеспечения, нацеленный на снижение повторения информации различного рода, особенно в системах со множеством слоев абстрагирования.

Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений, а не выводятся

автоматически из структуры моделей контроллеров.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Проект Django представляет собой настраиваемую пользователем среду разработки. Он берёт начало из Web-сайта одной онлайн-газеты и был выпущен как система с открытым исходным кодом в июле 2005 года.

Главные компоненты среды разработки Django следующие:

- Объектно-реляционное отображение (object-relational mapping) для создания моделей;
- Безупречный интерфейс администратора, специально созданный для конечных пользователей;
- Элегантно спроектированный механизм адресования (URL);
- Язык шаблонов, для дизайнеров;
- Система кэширования.

Объектом исследования представленной работы служит деятельность сервиса размещения объявлений.

Предметом исследования представленной работы служит веб-сервис для размещения объявлений о кулинарной продукции.

Основной целью представленной работы является разработка веб-платформы для размещения объявлений о кулинарной продукции домашнего производства, отвечающей современным требованиям к программно-технической части.

Для достижения цели необходимо решить следующие задачи:

1. Проанализировать существующие веб-сервисы в области доставки еды и продуктов.
2. Определить подходящую СУБД с учетом выбранного фреймворка разработки.
3. Разработать структуру и требования к веб-платформе.

4. Реализовать веб-платформу для размещения объявлений о кулинарной продукции домашнего производства.

ГЛАВА 1. АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

1.1. Формирование списка критерия сравнения существующих решений в области доставки еды и продуктов

Перед тем, как создавать собственный веб-сервис, необходимо провести анализ того, как с этой задачей на данный момент справились другие разработчики web-сервисов. Это позволит проанализировать плюсы и минусы уже существующих сайтов пищевой промышленности, а также специализированных ресурсов для размещения объявлений.

Выбранные площадки делятся на две категории - сервисы полного цикла (производство и распространения продукции) и торговые площадки предоставляющие услуги распространения и доставки продукции. Первые производят пищевой продукт, упаковывают, продают и доставляют. Вторые же выполняют роль агрегатора по продаже и доставке готовой продукции.

Схема, по которой был проведен анализ сайтов-конкурентов:

- доменное имя;
- навигационная структура;
- отображение в различных веб-браузерах;
- быстрота доступа к сайту;
- система регистрации и авторизации

Доменное имя анализируется по следующим критериям:

- Длина домена.

Длина домена должна содержать не более трех-четырех слов, так как короткие адреса лучше запоминаются, находятся и проходят оптимизацию в поисковых системах.

- Ключевые слова.

Хороший домен содержит ключевые слова которые дают первое понимание о смысле сайта для пользователя и помогают правильно отображаться в запросах поисковых систем.

- Зона доменного имени.

Зона доменного имени должна совпадать с местоположением потенциальных пользователей. Это необходимо, во-первых, для того чтобы пользователям было проще найти ресурс, во-вторых, для сокращения незаинтересованных посетителей.

- Название бренда.

Название бренда адресе хорошо выделяется на фоне конкурентов если бренд узнаваемый и имеет хорошую репутацию, предпочтительно использование доменов верхнего уровня, так как они дают возможность лучше обозначить тематику сайта.

Навигационная структура сайта отвечает за все переходы пользователя по сайту и, соответственно, является значимой частью интерфейса. Навигация подразделяется на основную — чаще всего представленную в виде горизонтальных или вертикальных меню, и дополнительную — для удобства перемещения пользователя. К дополнительной навигации относятся кнопки, ссылки, активные слайдеры и прочие элементы интерфейса, расположенные вне основного меню. Согласно общим правилам для навигационных структур, необходимо соблюдение правила перехода с любой страницы сайта на главную страницу. Кроме этого, частой практикой является создание дополнительной ссылки на главную в виде логотипа.

Ссылки должны иметь интуитивно понятный вид, выделяться подчеркиваниями, цветом, рамками и т.д. Важно, чтобы на любой странице пользователь понимал, где он находится и как ему перейти в нужное место. Если для перехода с одной страницы на другую пользователь совершает более 3-4 «кликов» мышью, то это говорит о низком юзабилити сайта.

Отображение в различных веб-браузерах, или кроссбраузерность, имеет для разработчиков большое значение и означает корректность отображения сайта в различных браузерах. Основная цель кроссбраузерности — не допустить значительных изменений дизайна сайта в разных веб-браузерах. Для проведения тестов на кроссбраузерность существует множество сервисов, воспользуемся онлайн сервисом browserling.com.

Одним из основных показателей доступности сайта является скорость его загрузки. Скорость загрузки сайта — это время которое необходимо для полной загрузки сайта от перехода по ссылке до полного отображения страницы. На скорость влияет множество интернет показателей — скорость загрузки основного контента (LCP), время задержки до первого ввода (FID), накопительный сдвиг макета (CLS) и т.д. Проверить эти показатели можно при помощи популярного сервиса для тестов скорости pagespeed.web.dev.

Регистрация и авторизация в классических веб-приложениях открывают множество новых возможностей, которые не всегда доступны без наличия авторизованного аккаунта.

Формы должны располагаться в традиционных для этого местах, чтобы пользователю было легко их найти. Регистрация и авторизация должны быть максимально простыми, чтобы пользователю не приходилось тратить много времени на их прохождения, наличие авторизации через телефон или соцсети в современных веб-приложениях является важным атрибутом. Плюсом будет наличие возможности прохождения этапа регистрации и авторизации из любой части сайта.

1.2. Обзор существующих решений в области доставки еды и продуктов

Проведём обзор, анализ и тесты популярных платформ, включая проект по доставке готовых рационов питания — «Grow Food», крупнейшие сервисы по доставке еды — «Яндекс.Еда» и «Dostaevsky».

Популярным сервисом полного цикла является проект «Grow Food», проект позиционирует себя как компания, занимающаяся доставкой здорового питания по подписке через интернет.

Подписка «Grow Food» — это еженедельная доставка питания на протяжении 45, 60 или 90 дней. Пользователь выбирает нужную линейку из представленных на сайте — Super Fit, Fit, Daily, Balance, Power. Далее ему необходимо оформить подписку в «Личном кабинете» (см. Рис. 1.1), заполнить необходимые поля, оплатить заказ картой онлайн, после чего еженедельно человек будет получать готовый рацион питания.

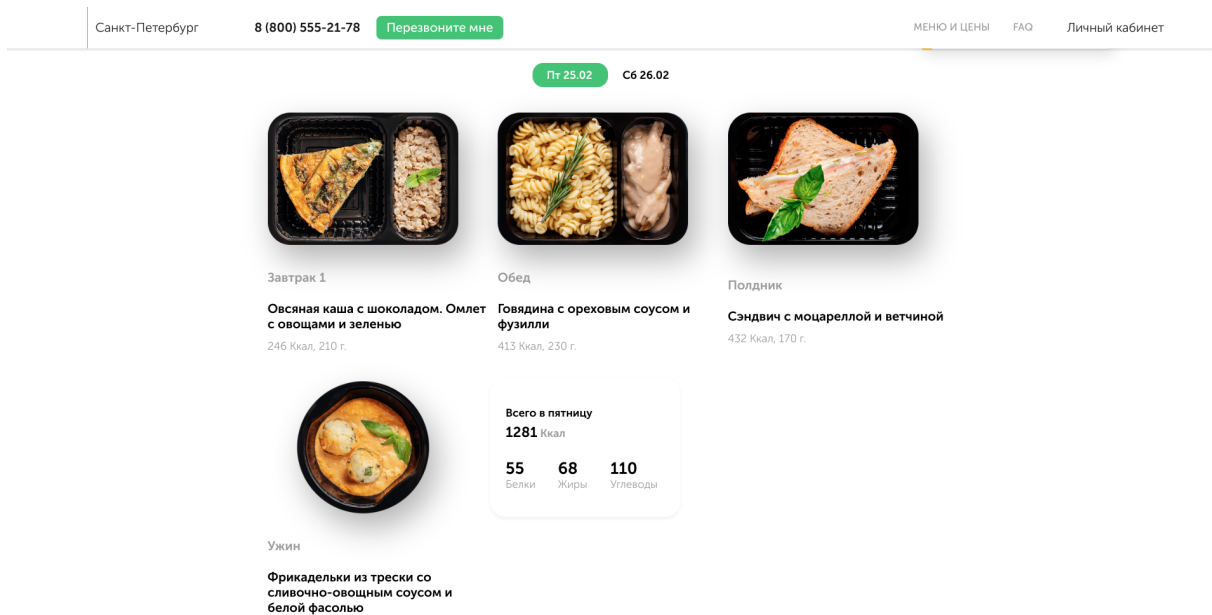


Рисунок 1.1. Личный кабинет «Grow Food»

Доменное имя ресурса расположенного в зоне .pro это общий домен верхнего уровня, название которого происходит от англ. professional и означает что домен предназначен для профессионалов в своей деятельности. Домен короткий, отражающий название бренда, имеет ключевые слова, не содержит лишних СИМВОЛОВ.

Веб-ресурс обладает удобной навигационной структурой, что удобно для пользователя. Вся необходимая информация для пользователя располагается на главной странице, при этом по сайту можно передвигаться при помощи навигации, при нажатии курсором на нужный раздел в меню фокус переносится на нужную информацию.

Проведя тесты кроссбраузерности Grow Food показал достойный результат корректно отобразив свою домашнюю страницу в браузерах Chrome, Firefox, Internet Explorer, Opera и Safari, в разных операционных системах, например Windows и Android

По скорости доступа к сайту выявились проблемы, проводя тесты был выявлен ряд проблем, проверка основных интернет-показателей не была пройдена, в частности скорость загрузки основного контента заняла 5.3 сек. что

является отрицательным результатом, также проверки на первую прорисовку контента и совокупное изменение макета дали удовлетворительный результат.

Регистрация на сайте выполняется с помощью номера телефона это единственный способ зарегистрироваться на данной платформе, введя номер в специальной форме на телефон мгновенно поступает смс с кодом доступа, введя код остается только заполнить небольшую анкету с личными данными. После регистрации, авторизация на сайте будет проходить аналогичным путем через смс на телефон. Такой способ отличается быстротой использования но имеет минус невозможности воспользоваться сервисом без телефона.

Лидером рынка торговых площадок является сервис «Яндекс.Еда», сервис занимается доставкой продукции из точек общепита — кафе, ресторанов, пекарен и кулинарий.

В первом квартале 2022 показатель валовой стоимости товаров направлений по распространению потребительских товаров и доставки продуктов из «Еды» по России вырос в 2,1 раза — до 20,5 млрд руб. Доставка продуктов в «Еде» также продолжает развиваться — в первом квартале 2022 количество заказов выросло в 3,2 раза по сравнению с первым кварталом 2021, а показатель валовой стоимости товаров увеличился в 3,9 раза [1].

Заказ можно сделать в мобильном приложении или на сайте сервиса (см. Рис. 1.2), приложение «Яндекс.Еда» есть под системы IOS и Android.

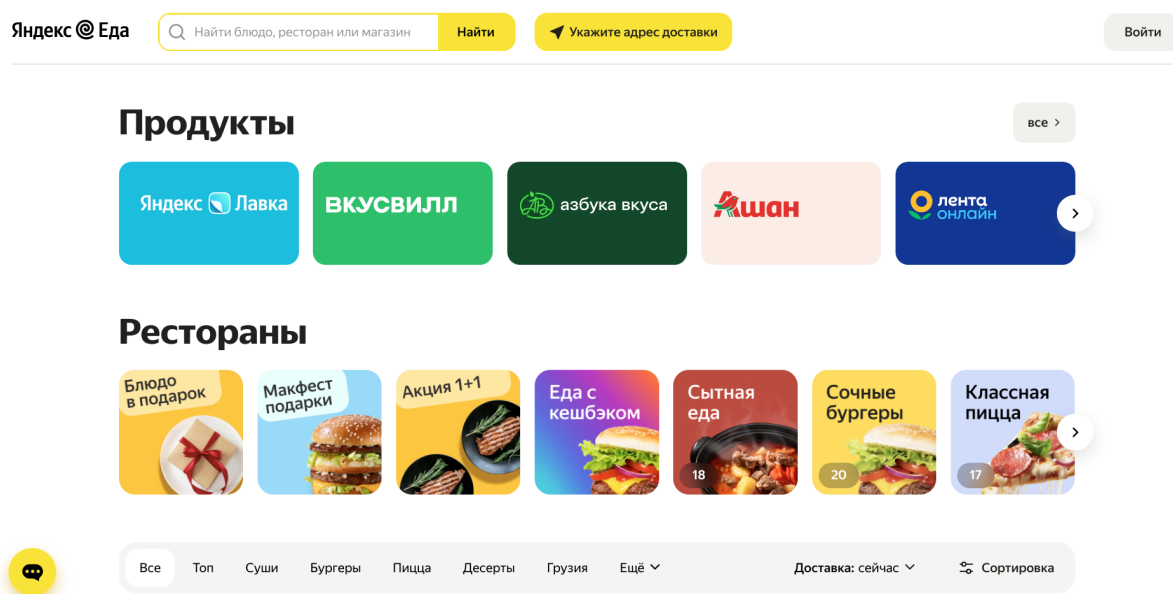


Рисунок 1.2. Главная страница сайта «Яндекс.Еда»

Чтобы сделать заказ через сайт или приложение, пользователь выбирает ресторан из ленты на главной странице, выбирает блюда, добавляет их в корзину, выбирает способ оплаты и оформляет заказ. Если ресторан работает со службой доставки «Яндекс.Еда», то пользователь получает уведомление, когда ресторан начинает готовить заказ. Когда заказ забирает из ресторана курьер сервиса, сотрудничающего с «Яндекс.Еда», пользователь получает уведомления об изменении статуса заказа, а также может отслеживать свой заказ на карте в режиме реального времени.

Так как «Яндекс.Еда» входит в группу компании «Яндекс» ее адрес расположен на поддомене `eda.` домена `yandex.ru`. Поддомены часто используются организациями для создания уникальных имен сайтов для своих подразделений или сервисов, связанных с организацией.

Навигация по сайту осуществляется прямыми переходами в нужные раздел указанный в меню на странице и с помощью продвинутого поиска в котором можно указать любую связанную с запросом информацию, сайт обладает динамическим обновлением веб-страницы что ускоряет и упрощает использование.

Тесты на отображение «Еда» от «Яндекс» прошла на максимальные показатели, корректно отображаясь не только в популярных браузерах и стандартных размерах экрана, но и в самых нестандартных задачах сайт создал максимально корректное изображение.

Сайт загружается за 3.4 секунды, первая прорисовка контента происходит за 0.4 секунды. First Input Delay, Interaction to Next Paint, Time to First Byte и другие сетевые показатели дают отличную оценку.

Пользователю «Едой» придется создать аккаунт для всей экосистемы «Яндекс». У «Яндекса», как и у других крупных систем, регистрация и последующая авторизация осуществляется любым удобным для пользователя способом: можно воспользоваться привычным вариантом логин или email, или

войти по номеру телефона, через множество соцсетей. Доступна также авторизация через сканирование QR-кода.

«Dostaevsky» — это круглосуточная доставка готовой еды работающая в крупных городах Москва, Санкт-Петербург, Сочи, Краснодар и Новосибирск доставляющая итальянскую, японскую, китайскую, мексиканскую, осетинскую и русскую кухню.

В меню на сайте представлены множества блюд, приобрести и заказать которые можно путем перетаскивания желаемого блюда в корзину с последующим оформлением заказа, минимальная сумма заказа — 500 рублей. Осуществляет приготовление и реализацию продукции с доставкой клиенту производитель, список производителей можно найти на сайте компании (см. Рис. 1.3).

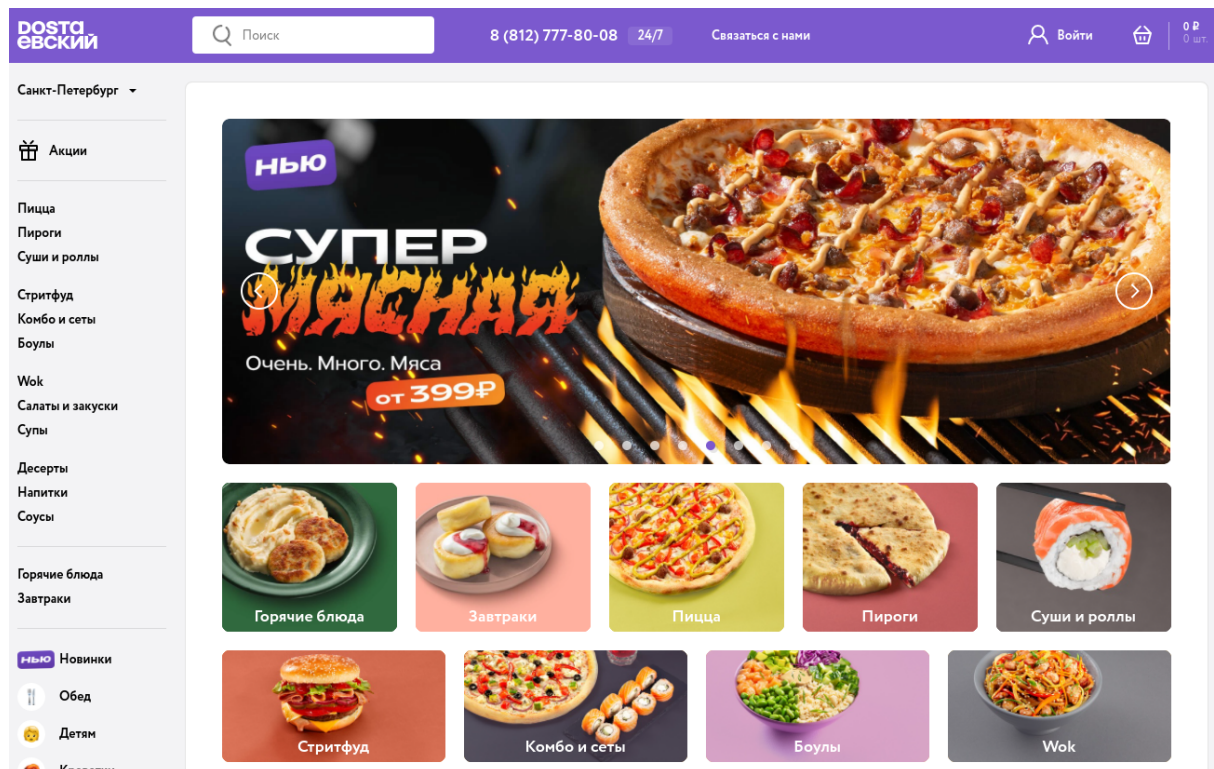


Рисунок 1.3. Главная страница сайта ««Dostaevsky»»

Доменное имя `dostaevsky.ru` содержит название компании в зоне `.ru`. Домен `.ru` национальный домен верхнего уровня для России. «Dostaevsky» применяет множество маркетинговых инструментов и название бренда в домене выгодно отличает ресурс.

Структура навигации разбита на несколько пунктов:

1. В верхнем меню поле поиска, контактный номер, кнопка входа и корзина.
2. В левом сайдбаре который будет зафиксирован на любой странице сайта расположены категории блюд, акции и выбор города.
3. Футер сайта содержит переходы на дополнительную информацию в виде соглашений, условий и политики компании.

На главной странице сайта находится слайдер с акционными предложениями, выбор категории блюд и сами блюда которые можно сразу же добавить в корзину.

В популярном но проблемном браузере Internet Explorer ресурс отрисовал изображения расплывчато и плохом качестве, работа платформы была корректной но над отображением разработчикам стоит поработать. Chrome, Firefox и другие популярные веб-обозреватели показали приятную глазу картинку и полный доступный функционал.

Анализ скорости дал низкую оценку производительности хотя основные интернет характеристики были пройдены с неплохими показателями, в основном проблемы с использованным на странице контентом форматы и размеры изображений, неиспользуемый CSS код, ошибки по JavaScript.

Регистрации в классическом ее понимании нету вообще, при первой авторизации по номеру телефона ресурс просто запоминает новый аккаунт, схема стандартная номер телефона, капча, код из смс и авторизация успешно пройдена. При первом заказе потребуется указать имя, адрес доставки и пару стандартных настроек по заказу. Процесс максимально простой и клиентоориентированный, возможно стоило добавить больше способов так называемой регистрации.

1.3. Обзор языка и среды программирования

Для разработки был выбран язык программирования Python и использован фреймворк Django.

Фреймворк Django написан разработчиками на языке программирования Python, именно поэтому его структура соответствует особенностям данного языка. Создатели смогли реализовать в Django паттерн MVC, и он применяется в

текущей версии фреймворка.

В Django также реализован мощный движок различных шаблонов и собственный язык для разметки. Шаблоны представляют собой файлы с HTML-кодом, с помощью которого отображаются данные. Содержимое файлов может быть статическим или динамическим. Шаблоны не содержат бизнес-логики. Поэтому они только отображают данные.

Такая архитектура позволяет Django успешно решать разные задачи, о которых ниже.

Неопытные специалисты считают Django одной из многих систем управления контентом (CMS). На самом деле это программный инструмент, с помощью которого можно создавать и запускать веб-приложения.

Фреймворк Django справляется с большим количеством задач и повышенными нагрузками. Его применяют для создания:

1. CRM-систем.
2. CMS.
3. Коммуникационных платформ.
4. Сервисов бронирования номеров.
5. Платформ управления документооборотом.

Также Django подходит для создания алгоритмических генераторов, платформ для электронных рассылок, систем верификации, систем фильтрации с динамическими правилами и сложными параметрами, платформ для анализа данных и сложных вычислений, машинного обучения.

Тысячи сайтов в разных странах мира созданы на Django. Этот фреймворк отлично подходит для разработки веб-приложений. Данный фреймворк использован следующими популярными веб-ресурсами:

- YouTube;
- DropBox;
- Google;
- Quora;
- Bitly;

- Reddit;
- Instagram;
- Pinterest.

Ниже описаны основные преимущества данного фреймворка, благодаря которым его выбирают ведущие разработчики со всего мира.

1. Развита экосистема.

Опытные разработчики рекомендуют воспринимать Django как систему. Это значит, что фреймворк обычно используется с большим количеством сторонних приложений. Их можно выбирать в зависимости от потребностей конкретного проекта.

2. Зрелость.

Django был представлен в 2005 году. За 14 лет существования он сильно изменился и усовершенствовался. В фреймворке постоянно появляются новые возможности, а старые совершенствуются.

3. Административная панель.

Административная панель Django автоматически генерируется при создании приложения. Это избавляет разработчика от необходимости создавать админку вручную.

4. SEO-дружественность.

Написанный на Python код получается читабельным и понятным даже неподготовленным людям. Это один из факторов, благодаря которым веб-приложения на Python считаются SEO-дружественными. Django генерирует семантические URL. Их также называют человеко-понятными URL или ЧПУ. В приложениях на Django легко реализуются другие функции, необходимые для поисковой оптимизации.

5. Расширяемость.

Функциональность Django расширяется с помощью плагинов. Это программные модули, которые позволяют быстро добавить на сайт нужную функцию. В официальном каталоге есть сотни плагинов, которые позволяют легко реализовать на сайте sitemap.xml, управлять доступами, подключить платёжную

систему Stripe и так далее. При необходимости вы можете отключать или заменять плагины, чтобы адаптировать приложение к текущим нуждам проекта.

6. Библиотеки.

В популярных языках программирования есть библиотеки, с помощью которых удобно решать специальные задачи. В библиотеках можно найти готовые решения: функции, классы, конфигурации и так далее. Благодаря таким решениям расширяются возможности языка, а также упрощается создание приложений.

7. Настройка.

Настройки фреймворка Django производится в файле `settings.py`, файл содержит все необходимые настройки для проекта. Основные настройки для корректной работы фреймворка это временная зона, язык кода, используемая база данных и настройки шаблонизатора.

8. ORM.

В Django реализовано объектно-реляционное отображение (ORM), которое обеспечивает взаимодействие приложения с базами данных (БД). ORM автоматически передаёт данные из БД, например, PostgreSQL или MySQL, в объекты, которые используются в коде приложения.

Также ORM позволяет быстро переключаться между базами данных с минимальными изменениями кода. Например, разработчик может использовать SQLite на локальном сервере, а потом переключиться на MySQL на production-сервере. Однако, для минимизации ошибок лучше использовать одну базу данных во время разработки и в продакшене.

Система управления базами данных (СУБД) — комплекс лингвистических и программных средств специального или общего назначения, которые позволяют управлять созданием, использованием баз данных.

СУБД выполняет ряд функций:

1. Управление данными на внешних устройствах памяти (дисках).
2. Управление данными, которые находятся в оперативной памяти, при помощи кэша диска.

3. Ведения журнала изменений, резервное копирование и восстановление данных после сбоев в системе.
4. Поддержка языков баз данных.

СУБД содержит компоненты:

1. Ядро — управляет данными на внешних и внутренних устройствах памяти, ведет журнал изменений.
2. Процессор языка базы данных — оптимизирует запросы на изменение и извлечение данных.
3. Подсистема поддержки времени исполнения — создает пользовательский интерфейс с СУБД при манипуляции данными.
4. Сервисные программы (внешние утилиты) — обеспечивают дополнительные возможности обслуживания информационной системы.

СУБД систематизируют в зависимости от классификационного признака:

1. По модели данных:
 - a. Иерархические.
 - b. Сетевые.
 - c. Реляционные.
 - d. Объектно-ориентированные.
2. По степени распределенности:
 - a. Локальные.
 - b. Распределенные.
3. По способу доступа к базе данных:
 - a. Файл-серверные.
 - b. Клиент-серверные.
 - c. Встраиваемые.

Для работы с Web-приложением подходят клиент-серверные СУБД. Такие СУБД располагают на сервере вместе с базой данных, осуществляя непосредственный (монопольный) доступ к базе данных. При этом все запросы

клиента на обработку данных СУБД обрабатывает централизованно. Рассмотрим основные типы СУБД, с которыми может взаимодействовать Django.

1. Microsoft SQL Server

Microsoft SQL Server — СУБД для управления реляционными базами данных. Данная СУБД реализует такие ключевые требования:

- Производительность.
- Стабильность.
- Масштабируемость.

Microsoft SQL Server отличается от других СУБД тем, что осуществляет интеграцию с Microsoft Office, выступает гарантом высокой безопасности и производительности.

Основные достоинства Microsoft SQL Server:

- Высокая производительность и масштабируемость.
- Низкая уязвимость базы данных.
- Поддержка самообслуживания при работе с инструментами бизнес-аналитики.

Недостатки Microsoft SQL Server:

- Высокая цена для юридических лиц.
- Для работы приложения требуются значительные ресурсы.

2. MySQL

MySQL — одна из самых популярных СУБД для управления реляционными базами данных. Данная СУБД эффективно функционирует в купе с Web-приложениями и сайтами. При этом данная СУБД обладает простотой в освоении.

MySQL отличается от других СУБД тем, что является универсальной и довольно распространенной.

Основные достоинства MySQL:

- Легкость в использовании.
- Богатый функционал.
- Высокий уровень безопасности.

- Масштабируемость.
- Высокая производительность.

Недостатки MySQL:

- Надежность недостаточного уровня.
- Низкая скорость разработки.

3. Oracle database

Oracle database — объектно-реляционная СУБД, поддерживающая технологии, осуществляющие объектно-ориентированный подход, обеспечивая управление использованием и созданием баз данных.

Ключевые возможности Oracle database: обеспечение работы каждого экземпляра базы данных на нескольких узлах GRID, что обеспечивает управление нагрузкой и масштабированием; автоматическое распределение данных между ресурсами хранения данных.

Основные достоинства Oracle database:

- Высокая производительность.
- Простота средств разработки.
- Высокий уровень безопасности.
- Масштабируемость.
- Недорогие системы серверов.

Недостатки Oracle database:

- Высокая стоимость.
- Для работы приложения требуются значительные ресурсы .

Таким образом, изучив разные клиент-серверные системы управления базами данных и сопоставив достоинства и недостатки каждой системы, выбрана СУБД — MySQL.

Рассмотрим основные компоненты Django.

1. Шаблонизатор.

У Django есть собственный шаблонизатор, шаблонизатор позволяет применять html-шаблоны для генерации конечных html-страниц. Шаблон

содержит статические статические части html-кода и специальный синтаксис для динамического вывода.

Django может работать с одним или несколькими шаблонизаторами,

Django предоставляет встроенные серверные модули для собственной системы шаблонов Django Template Language, и популярную альтернативу Jinja2. Jinja2 позволяет подставлять переменные из Python напрямую на страницы в браузере.

2. Представления

Представления в Django отвечают на web-запросы, это сообщение посылаемое клиентом серверу, и возвращают web-ответы сообщение от сервера к клиенту. Ответ может быть HTML-содержимым веб-страницы, перенаправлением, ошибкой 404, XML-документом или изображением. Представление содержит любую логику для возврата ответа, код может быть любым и хранится в различных местах по соглашению в файле views.py.

1.4. Выводы к главе 1

В первой главе были рассмотрены сервисы по продаже готовой еды, с целью выделения ключевых характеристик современных конкурентных решений. Также был произведён анализ различных СУБД, имеющих возможность интеграции с фреймворком Django. В результате было выявлено, что использование СУБД MySQL является наиболее подходящим для реализации веб-платформы.

ГЛАВА 2. РАЗРАБОТКА WEB-СЕРВИСА ДЛЯ РАЗМЕЩЕНИЯ ОБЪЯВЛЕНИЙ О КУЛИНАРНОЙ ПРОДУКЦИИ ДОМАШНЕГО ПРИГОТОВЛЕНИЯ

2.1. Разработки структуры сервиса

На рисунке 2.1 изображена концептуальная модель веб-сервиса для размещения объявлений о кулинарной продукции домашнего производства.



Рисунок 2.1 Концептуальная модель веб-сервиса для размещения объявлений о
кулинарной продукции домашнего производства

Web-браузер получает от пользователя запрос на выдачу информации и передает его на Web-сервер.

Web-сервер определяет: запрос динамический или статический? Если статический, то Web-сервер получает файл HTML-страницы и передает ответ на запрос Web-браузеру, если динамический — отправляет данный запрос Web-приложению.

Web-приложение формирует ответ на запрос, используя информацию из базы данных, вставляя ее в нужные динамические поля файла HTML-страницы

В данном случае, база данных служит для накопления и хранения информации о проектах, заказчиках и сотрудниках.

При разработке веб-сервиса будет использован паттерн MVC.

MVC (model — view — controller) — шаблон, которые делит веб-сервис на три взаимосвязанных компонента:

- Контроллер — класс веб-сервиса, при помощи которого связываются пользователь и система, база данных и представление. Данный класс получает запросы от пользователя и выводит сформированную в результате обработки запроса HTML-страницу.
- Представление — интерфейс пользователя, который представляет собой HTML-страницу.
- Модель — класс, который служит для описания логики хранения и использования данных.

Взаимодействия между компонентами системы представлены на Рисунке 2.2:

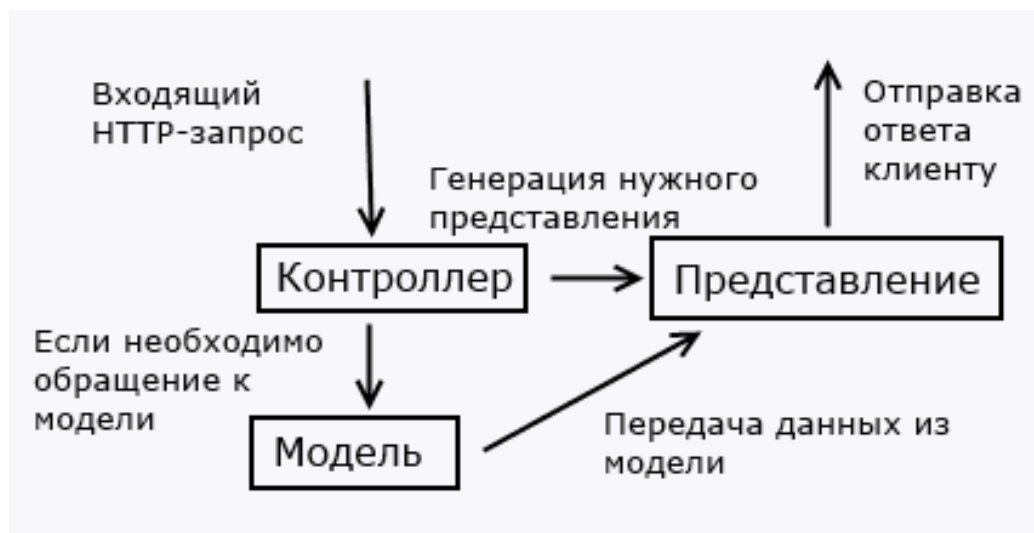


Рисунок 2.2 Схема взаимодействия компонентов в шаблоне MVC

Данная схема показывает, что модель базы данных не изменяется при изменении контроллера или представления. А изменения контроллера или представления не взаимосвязаны и также не влияют друг на друга.

Использование данного шаблона позволяет разделить ответственность при разработке приложения, а также облегчает тестирование.

Таким образом, веб-сервис будет состоять из компонентов:

- База данных, в которой будут накапливаться данные веб-сервиса.
- Контроллер (серверная часть), который будет анализировать запрос пользователя.

- HTML-страницы, которые будут служить визуальным компонентом системы, пользовательским интерфейсом.

2.2. База данных и административная часть веб-сервиса для размещения объявлений о кулинарной продукции домашнего производства

Незарегистрированные пользователи имеют возможность просматривать объявления о кулинарной продукции домашнего приготовления, оставленные зарегистрированными пользователями-кулинарами, зарегистрироваться, ранжировать всех зарегистрированных пользователей по: Id, имени, фамилии, году рождения, рейтингу, кулинарному стажу. Также имеется возможность выбора метода ранжирования: плотное (DENSE_RANK) и неплотное (DENSE).

При регистрации, пользователю автоматически назначается роль «customer». Также в системе предусмотрены роли «admin» и «chef». У каждой роли имеются права на выполнение каких-либо операций.

Пользователи-покупатели могут:

1. Редактировать свой профиль.
2. Просматривать добавленные пользователями объявления.
3. Сортировать объявления по категориям.
4. Просматривать личные данные пользователей-кулинаров.
5. Добавлять комментарии к объявлениям.
6. Искать объявления по названию, краткой и полному описанию, по категориям.

Пользователи-кулинары, кроме всех вышеперечисленных действий, также имеют возможность добавлять новые объявления.

Администраторы обладают все теми же правами, что и другие пользователи, но в дополнение имеют возможность:

1. Изменить роль пользователя.
2. Просматривать список всех пользователей веб-сервера.

ER-моделирование — моделирование предметной области при помощи модели «сущность-связь». Структура данных изображается при помощи ER-диаграмм [6].

В ER-моделировании существуют четыре базовых понятия:

1. Сущность (Объект).
2. Атрибут сущности — свойство, которое характеризует объект с какой-либо точки зрения.
3. Ключевой атрибут — атрибут, который позволяет однозначно идентифицировать объект.
4. Связь. Связи, которые устанавливаются между разными сущностями, бывают:
 - a. Один к одному (1:1).
 - b. Один ко многим (1:M).
 - c. Многие ко многим (N:M).

Размещение объявлений о кулинарной продукции домашнего приготовления в своей деятельности оперирует следующими сущностями:

- пищевая ценность,
- продукция,
- заказ,
- покупатель,
- регистрация,
- кулинар,
- категория,
- платеж,
- продукция в заказе.

На рисунке 2.3 отображена ER-модель для предметной области размещения объявлений о кулинарной продукции домашнего приготовления, которая построена в результате анализа сущностей и связей.

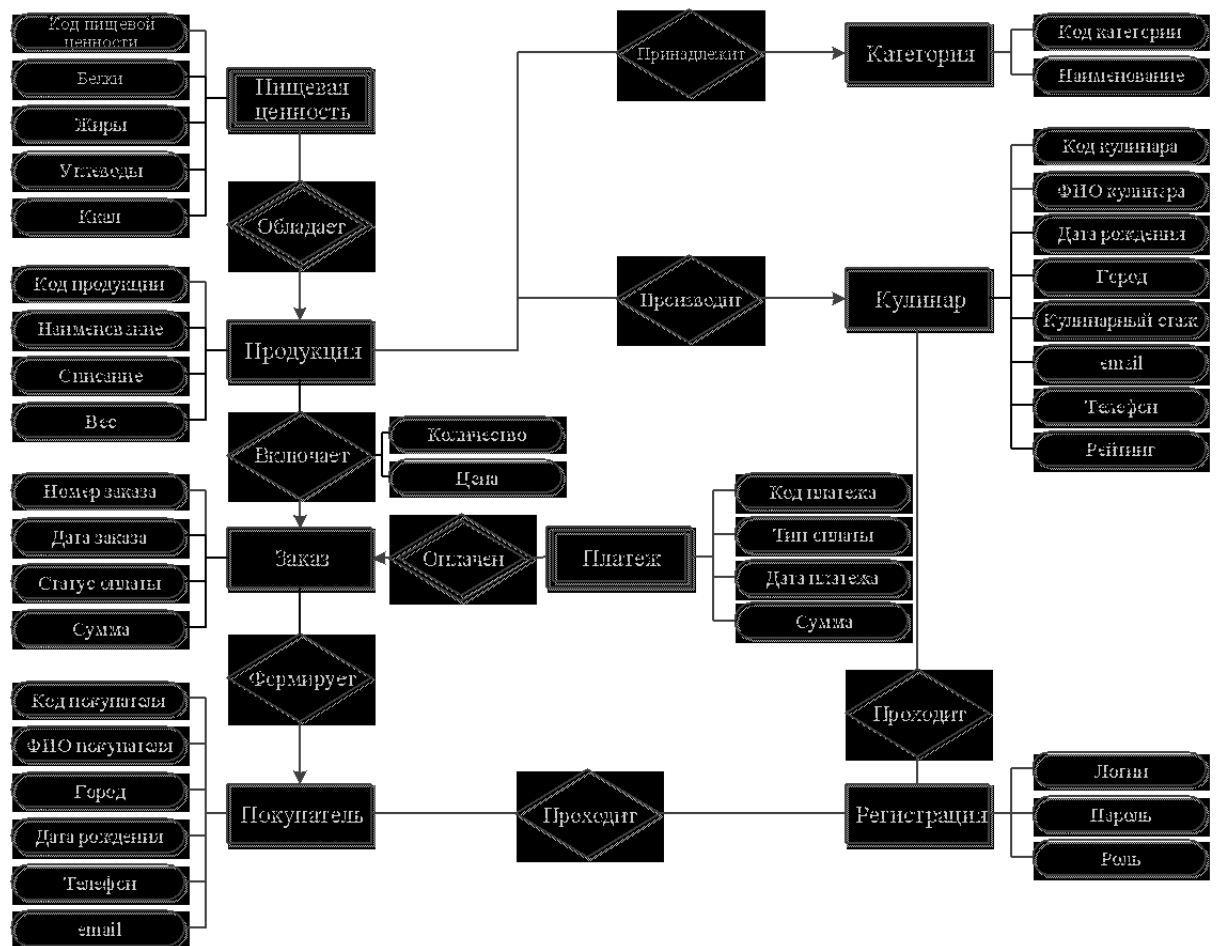


Рисунок 2.3 Построенная ER-модель

Описание таблицы базы данных Customer (Покупатель) представлено в таблице 2.1.1. Данная сущность используется в БД для регистрации пользователей-покупателей.

Таблица 2.1.1. Таблица «Customer»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое	Обязательно заполнять
Код покупателя	cust_ID	char(10)	+	+
ФИО	cust_FIO	nvarchar(155)	-	+
Город	cust_city	nvarchar(100)	-	+
Дата рождения	cust_bdate	date	-	+
Телефон	cust_tel	nvarchar(10)	-	+
Email	cust_email	nvarchar(100)	-	+

Описание таблицы базы данных Category (Категория) представлено в таблице 2.1.2. Данная сущность используется в БД для хранения справочной информации о категории продукта.

Таблица 2.1.2. Таблица «Category»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое	Обязательно заполнять
Код категории	cat_ID	char(10)	+	+
Наименование	cat_name	nvarchar(155)	-	+

Описание таблицы базы данных Chef (Кулинар) представлено в таблице 2.1.3. Данная сущность используется в БД для регистрации пользователей-кулинаров.

Таблица 2.1.3. Таблица «Chef»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое поле	Обязательно заполнять
Код кулинара	chef_ID	char(10)	+	+
ФИО	chef_FIO	nvarchar(155)	-	+
Город	chef_city	nvarchar(100)	-	+
Дата рождения	chef_bdate	date	-	+
Кулинарный стаж	chef_herd	char(10)	-	+
Телефон	chef_tel	nvarchar(10)	-	+
Email	chef_email	nvarchar(100)	-	+
Рейтинг	rating	char(10)	-	+

Описание таблицы базы данных Registration (Регистрация) представлено в таблице 2.1.4. Данная сущность используется в БД для хранения данных для входа на веб-сервис пользователей.

Таблица 2.1.4. Таблица «Registration»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое поле	Обязательно заполнять
----------------------	-----------------------	---------------------	------------------	--------------------------

Логин	login	char(10)	+	+
Пароль	password	nvarchar(155)	-	+
Роль	role	nvarchar(100)	-	+
Код кулинера	chef_ID	char(10)	-	-
Код покупателя	cust_ID	char(10)	-	-

Описание таблицы базы данных Order (Заказ) представлено в таблице 2.1.5. Данная сущность используется в БД для хранения данных о заказе пользователем продукции.

Таблица 2.1.5. Таблица «Order»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое поле	Обязательно заполнять
Номер заказа	ord_num	char(10)	+	+
Дата заказа	ord_date	date	+	+
Статус оплаты	ord_status	nvarchar(100)	-	+
Сумма	ord_sum	double	-	+

Описание таблицы базы данных Payment (Платеж) представлено в таблице 2.1.6. Данная сущность используется в БД для хранения данных для оплаты заказа.

Таблица 2.1.6. Таблица «Payment»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое поле	Обязательно заполнять
Номер заказа	ord_num	char(10)	+	+
Код платежа	ord_ID	char(10)	+	+
Тип оплаты	pay_type	nvarchar(100)	-	+
Дата платежа	ord_date	date	-	+
Сумма	ord_sum	double	-	+

Описание таблицы базы данных Products (Продукция) представлено в таблице 2.1.7. Данная сущность используется в БД для хранения данных о

кулинарной продукции домашнего приготовления.

Таблица 2.1.7. Таблица «Products»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое поле	Обязательно заполнять
Код продукции	prod_ID	char(10)	+	+
Наименование	prod_name	nvarchar(55)	-	+
Описание	prod_desc	nvarchar(255)	-	+
Вес	prod_weight	char(10)	-	+
Код кулинара	chef_ID	char(10)	-	+
Код категории	cat_ID	char(10)	-	+

Описание таблицы базы данных ProductsinOrder (Продукция в заказе) представлено в таблице 2.1.8. Данная сущность используется в БД для хранения данных о продукции, которая входит в определенный заказ.

Таблица 2.1.8. Таблица «ProductsinOrder»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое поле	Обязательно заполнять
Номер заказа	ord_numb	char(10)	+	+
Код продукции	prod_ID	char(10)	+	+
Количество	number	double	-	+
Цена	price	double	-	+

Описание таблицы базы данных NutriValue (Пищевая ценность) представлено в таблице 2.1.9. Данная сущность используется в БД для хранения данных о пищевой ценности кулинарной продукции домашнего приготовления.

Таблица 2.1.9. Таблица «NutriValue»

Наименование поля	Идентификатор поля	Тип и длина поля	Ключевое	Обязательно заполнять
Код продукции	prod_ID	char(10)	+	+

Код пищевой ценности	nv_ID	char(10)	+	+
Белки	nv_proteins	nvarchar(30)	-	+
Жиры	nv_fats	nvarchar(30)	-	+
Углеводы	nv_carb	nvarchar(30)	-	+
Ккал	nv_kcal	nvarchar(30)	-	+

2.3. Описание полученного web-приложения

В приложении, разработанном в рамках представленной курсовой работы, есть все функции для быстрого и удобного добавления объявлений о кулинарной продукции собственного приготовления. Используя функцию поиска, можно моментально отсортировать и найти объявление, необходимое пользователю.

2.3.1. Структура сервиса

Разработанное приложение состоит из моделей, контроллеров и представлений. Ниже дадим краткое описание компонентов системы и их названия в коде.

Контроллеры:

1. AccountController — используется для регистрации и авторизации пользователей, изменения профиля пользователя, добавления объявления.
2. HomeController — используется для получения информации, отображаемой на главной странице, получения информации о пользователе, поиска, ранжирования пользователей, получения статистики.

Модели:

1. Customer — модель, представляющая собой сущность пользователя.
2. Category — модель, представляющая собой сущность категории продукции.
3. Chef — модель, представляющая собой сущность пользователя-кулинара.
4. Registration — модель, представляющая собой сущность регистрации.
5. Order — модель, представляющая собой сущность заказа.

6. Payment — модель, представляющая собой сущность платежа.
7. Products — модель, представляющая собой сущность продукции.
8. ProductsinOrder — модель, представляющая собой сущность продукции в заказе.
9. NutriValue — модель, представляющая собой сущность пищевой ценности.

Представления:

1. Account

- a. CreateDescription — представление создания объявления о кулинарной продукции собственного приготовления.
- b. EditProfile — представление редактирования профиля пользователя.
- c. Login — представление авторизации.
- d. Profile — представление просмотра профиля.
- e. Register— представление регистрации.

2. AdminModer

- a. AddCategory — представление добавления категории объявления.
- b. GetDeclarationsForUser — представление, отображающее список всех объявлений выбранного пользователя.
- c. GetUsers — представление, отображающее список всех пользователей.

3. Declaration

Details — представление, отображающее подробную информации об объявлении.

4. Home

- a. DetailsUser — представление, отображающее подробную информацию о выбранном пользователе.
- b. Index — представление главной страницы.
- c. Rank — представление ранжирования пользователей.

- d. Search— представление поиска.
- e. Statistics— представление статистики.
- f. 5. Shared
- g. _Footer — представление футера.
- h. _Header — представление хедера.
- i. _Layout — представление мастер страницы.
- j. _DetailUserScript — представление, содержащее скрипт для формы отображения информации о пользователе.

2.3.2. Программно-аппаратные требования

Приложение является веб-сервисом, поэтому программные аппаратные требования для сервера и для клиента будут немного отличаться.

Для сервера:

1. ОС (операционная система): Любая операционная система, которая поддерживает установку HTTP-сервера;
2. ОЗУ объемом 16 ГБ;
3. 25 Гб доступного пространства на жестком диске;
4. Платформа .NET 4.5 и выше;
5. Любой HTTP-сервер;
6. СУБД: Microsoft SQL Server 2008 и выше.

Для клиента:

1. ОС: Любая операционная система, поддерживающая интернет соединение, и имеющая возможность установки интернет - браузера;
2. Любой интернет — браузер;
3. Клавиатура, мышь/тачпад.

2.3.3. Функциональные возможности

Незарегистрированный пользователь может просматривать добавленные объявления, комментарии к ним, ранжировать пользователей по различным характеристикам, смотреть статистику, смотреть карту.

Работа с системой начинается с главной страницы, которая содержит

1. Навигационную панель со страницами:

- a. Главная страница
 - b. Поиск
 - c. Избранное
 - d. Рейтинг
 - e. Вход
- 2. Объявления, добавленные пользователями
 - 3. Категории объявлений о кулинарной продукции домашнего приготовления, при нажатии на которые отображаются объявления, соответствующие выбранной категории.

Незарегистрированный пользователь может просматривать информацию о любом пользователе. Для этого он должен нажать на логин любого пользователя на главной странице.

Кроме этого пользователь может просматривать полную информацию по каждому объявлению, а также комментарии выбранного объявления.

Также незарегистрированный пользователь может воспользоваться поиском объявлений по названию, краткому или полному описанию.

2.4. Выводы к главе 2

Во второй главе была реализована структура будущей системы и разработана база данных, отвечающая ролям выделенных пользователей. Также в данной главе была выполнена четвёртая задача выпускной квалификационной работы — реализация веб-платформы для размещения объявлений о кулинарной продукции домашнего производства.

ЗАКЛЮЧЕНИЕ

В результате была достигнута основная цель данной работы — разработка веб-платформы для размещения объявлений о кулинарной продукции домашнего производства, отвечающей современным требованиям к программно-технической части.

Для достижения цели были решены следующие задачи:

1. Проанализированы существующие веб-сервисы в области доставки еды и продуктов.
2. Определена подходящая СУБД с учетом выбранного фреймворка разработки.
3. Разработана структура и требования к веб-платформе.
4. Реализована веб-платформу для размещения объявлений о кулинарной продукции домашнего производства.

В рамках выпускной квалификационной работы было разработано веб-приложение, представляющее собой информационную систему, позволяющую пользователям оставлять объявления о кулинарной продукции собственного приготовления.

Данное приложение позволяет быстро найти интересующие их объявления и оставить свои. Для управления сайтом предусмотрено три управляющие роли: пользователи-покупатели, пользователи-кулинары и администраторы.

Пользователи, относящиеся к первой группе, могут, помимо основных функций зарегистрированных пользователей, добавлять новости и создавать темы объявлений. Самая привилегированная группа пользователей — администраторы. Помимо функций остальных пользователей, они также имеют возможность изменять роли пользователей и экспортировать различные данные из базы данных. Стоит также отметить возможность ранжирования (сортировки) всех зарегистрированных пользователей по: имени, фамилии, году рождения, рейтингу. Данная функция позволяет быстро и удобно отсортировать пользователей по необходимому полю, объединив однотипных (по этому полю) пользователей в одну группу.

СПИСОК ЛИТЕРАТУРЫ

1. Васильев, Александр Николаевич Python на примерах. Практический курс по программированию. Руководство / Васильев Александр Николаевич. — М.: Наука и техника, 2017. — 752 с. (дата обращения: 06.02.2022).
2. Вишневская, Т. И. Методология программной инженерии: методические указания / Т. И. Вишневская, Т. Н. Романова. — Москва: МГТУ им. Н.Э. Баумана, 2017. — 60 с. — ISBN 978-5-7038-4522-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103504> (дата обращения: 06.02.2022).
3. Дронов, Владимир Django: практика создания Web-сайтов на Python / Владимир Дронов. — М.: БХВ — Петербург, 2016. — 865 с. (дата обращения: 06.02.2022).
4. Коломейченко, А. С. Информационные технологии: учебное пособие для вузов / А. С. Коломейченко, Н. В. Польшакова, О. В. Чеха. — 2-е изд., перераб. — Санкт-Петербург: Лань, 2021. — 212 с. — ISBN 978-5-8114-7564-3. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/177030> (дата обращения: 06.02.2022).
5. Куусинен, М.Э. Говорим по-фински. Puhutaan suomea / М.Э. Куусинен. — М.: Петрозаводск: Карелия, 2017. — 356 с. (дата обращения: 06.02.2022).
6. Лучано, Р. Python. К вершинам мастерства / Р. Лучано ; перевод с английского А. А. Слинкин. — Москва: ДМК Пресс, 2016. — 768 с. — ISBN 978-5-97060-384-0. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/93273> (дата обращения: 06.02.2022).
7. Россум, Г. Язык программирования Python / Г. Россум, Ф.Л.Дж. Дрейк, Д.С. Откидач и др. — М.: [не указано], 2020. — 578 с. (дата обращения: 06.02.2022).

8. Форсье, Джефф Django. Разработка веб-приложений на Python / Джефф Форсье, Пол Биссекс, Уэсли Чан. — М.: Символ-плюс, 2018. — 456 с. (дата обращения: 06.02.2022).
9. Классификация таблиц в реляционных базах данных по признакам целостности и избыточности данных: [Электронный ресурс]: — URL: <https://habr.com/ru/post/250177/> (дата обращения: 06.02.2022).
10. Системы управления базами данных: [Электронный ресурс]: — URL: <https://lecturesdb.readthedocs.io/databases/dbms.html> (дата обращения: 06.02.2022).
11. Система управления базами данных: [Электронный ресурс]: — URL: <https://studfiles.net/preview/6085230/> (дата обращения: 06.02.2022).
12. Система управления базами данных MySQL: [Электронный ресурс]: — URL: https://depix.ru/articles/sistema_upravleniya_bazami_dannyh_mysql (дата обращения: 06.02.2022).
13. Советов, Б. Я. Информационные технологии: теоретические основы: учебное пособие / Б. Я. Советов, В. В. Цехановский. — 2-е изд., стер. — Санкт-Петербург: Лань, 2022. — 444 с. — ISBN 978-5-8114-1912-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/209876> (дата обращения: 06.02.2022).
14. Сравнение современных СУБД: [Электронный ресурс]: — URL: <http://drach.pro/blog/hi-tech/item/145> (дата обращения: 06.02.2022).
15. Ружников, В. А. Основы сетевого программирования на языке высокого уровня Python: учебно-методическое пособие / В. А. Ружников, М. А. Вержаковская. — Самара: ПГУТИ, 2019. — 136 с. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/223331> (дата обращения: 06.02.2022).
16. Тарасов, С. В. СУБД для программиста. Базы данных изнутри / С. В. Тарасов. — Москва: СОЛОН-Пресс, 2015. — 320 с. — ISBN 978-2-7466-7383-0. — Текст: электронный // Лань: электронно-библиотечная

- система. — URL: <https://e.lanbook.com/book/64959> (дата обращения: 06.02.2022).
17. Как выбрать доменное имя. — Текст: электронный // Справка Google: [сайт]. — URL: <https://support.google.com/domains/answer/6388481?hl=ru> (дата обращения: 06.02.2022).
18. Как сделать кроссбраузерность сайта. — Текст: электронный // webformymself: [сайт]. — URL: <https://webformymself.com/kak-sdelat-krossbrauzernost-sajta/> (дата обращения: 06.02.2022).
19. Скорость загрузки сайта: как проверить и увеличить. — Текст: электронный // ashmanov: [сайт]. — URL: <https://www.ashmanov.com/education/articles/skorost-zagruzki-sajta/> (Дата обращения: 06.02.2022)
20. Яндекс объявляет финансовые результаты за I квартал 2022 года. — Текст: электронный // Яндекс: [сайт]. — URL: https://yandex.ru/company/press_releases/2022/2022-27-04 — (Дата обращения: 30.04.2022).