

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ЮЖНО-РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени М.И. Платова
(НОВОЧЕРКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ)»

ФАКУЛЬТЕТ _____ ИТУ _____
КАФЕДРА _____ ИИСТ _____
НАПРАВЛЕНИЕ _____ ПРИЛАДНАЯ ИНФОРМАТИКА В ЭКОНОМИКЕ _____

КУРСОВАЯ РАБОТА

ПО ДИСЦИПЛИНЕ _____ Разработка приложений для Интернет _____

НА ТЕМУ: _____ Web-сайт кулинарных рецептов _____

Выполнил студент _____ 3 _____ курса, Группы _____ 2а _____ Федотова А.П. _____
Фамилия, имя, отчество

Принял _____ доцент _____ Панфилов Александр Николаевич _____
Должность, звание Фамилия, имя, отчество

«_____» _____ 2021 г. _____
Подпись

Новочеркасск, 2021 г.

**Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего образования
«ЮЖНО-РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
(НОВОЧЕРКАССКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ)» имени М.И.ПЛАТОВА**

ФАКУЛЬТЕТ
КАФЕДРА
НАПРАВЛЕНИЕ

Информационных технологий и управления
Информационные и измерительные системы и технологии
09.03.03 - "Прикладная информатика"

УТВЕРЖДАЮ:

Заведующий кафедрой ИИСТ
_____ Горбатенко Н.И.

(подпись)

«__» _____ 2020 г.

ЗАДАНИЕ

на курсовую работу по дисциплине «РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ ИНТЕРНЕТ»

Студенту **ФЕДОТОВА**

(Фамилия, имя, отчество)

1. Тема курсовой работы: **Web - сайт кулинарных рецептов**

Выполнение курсовой работы включает следующие этапы:

Этап первый. Проектирование сайта:

1. Сформулировать целевую аудиторию сайта, дать описание, определить ключевые слова тематического ядра. Оценить объем сайта (сколько страниц). Отразить в отчете.
2. Разработать структуру сайта. Отобразить в виде схемы в отчете.
3. Разработать систему связей между страницами сайта. Отобразить в виде схемы в отчете.
4. Разработать структуру страницы (отразить в отчете).
5. Определить используемую технологию (отразить в отчете).

Этап второй. Создание базовой структуры:

1. Создать структуру сайта в корневом каталоге.
2. Разработать и создать базовый шаблон страницы с учетом требований спецификации языка HTML.
3. Разработать систему навигации.
4. Доработать шаблон с учетом панелей навигации.
5. Создать индексные страницы сайта и разделов.
6. Создать дополнительные страницы.

Этап третий. Создание базового наполнения:

1. Создать базовое наполнение индексных страниц.
2. Создать базовое наполнение разделов.
3. Создать расширенное наполнение.

Этап четвертый. Подготовка документации:

1. Подготовить краткую информацию по сайту: содержание, ключевые слова, используемая технология и др.
2. Исчислить показатели эффективности сайта
3. Подготовить отчет по курсовой работе

Срок сдачи работы

Руководитель Панфилов А.Н.

Задание принял к исполнению

« 04 » сентября 2020 г.

Дата выдачи задания

« 25 » декабря 2020 г.

(Подпись)

(Подпись)

« 04 » сентября 2020 г.

СОДЕРЖАНИЕ

| | |
|--|----|
| СОДЕРЖАНИЕ..... | 4 |
| ВВЕДЕНИЕ..... | 5 |
| ПОСТАНОВКА ЗАДАЧИ..... | 6 |
| 1.ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ИНТЕРНЕТ-ПРИЛОЖЕНИЯ..... | 7 |
| 2.ПРОЕКТИРОВАНИЕ WEB-САЙТА..... | 11 |
| 3. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ САЙТА | 17 |
| 4.ПРОГРАММНАЯ РЕАЛИЗАЦИЯ САЙТА..... | 24 |
| 5.ОЦЕНКА ЭФФЕКТИВНОСТИ И ТЕСТИРОВАНИЕ САЙТА..... | 25 |
| ЗАКЛЮЧЕНИЕ..... | 27 |
| БИБЛИОГРАФИЧЕСКИЙ СПИСОК..... | 28 |
| ПРИЛОЖЕНИЯ | 29 |

ВВЕДЕНИЕ

В современном мире, все большее значение имеет потребление качественной еды, красивой подачи, эстетики питания.

Если в прошлом, для создания того или иного блюда, люди использовали газеты и журналы, советы друзей и знакомых, то сегодня, интернет стремительно поглощает внимание людей. Именно через интернет современный человек узнает большую часть необходимой ему информации. В том числе – о кухне, способах приготовления блюд, продуктах и подаче. Как следствие спроса, появилось массу сайтов, которые доносят нужную информацию. Однако, этого недостаточно. Сегодня для успешности сайта необходимо учитывать массу факторов.

Особенно нужно отметить и структуру сайта. Он должен содержать простое, удобное в использовании навигацию, быть понятным целевой аудитории, но вместе с тем вызывать доверие и расположение пользователя. И в этом отношении, важны все нюансы – цвета, шрифт и, конечно, общее настроение, которое создает дизайн сайт.

Целью данной работы является создание удобного, максимально информативного сайта, посвященного кулинарным рецептам.

ПОСТАНОВКА ЗАДАЧИ

В ходе выполнения курсовой работы был разработан Web-сайт кулинарных рецептов. На сайте можно просматривать рецепты, оставлять комментарии к ним, поставить оценку «нравится», а также можно поделиться собственным рецептом. Материал сайта доступен каждому.

Для достижения поставленной цели необходимо решить следующие задачи:

- Построить модель Интернет-приложения;
- Составить техническое задание на разработку Web-сайта;
- Спроектировать функциональную структуру и модули сайта;
- Спроектировать дизайн сайта;
- Разработать информационное обеспечение сайта;
- Разработать программную часть сайта;
- Протестировать сайт и оценить его эффективность.

Разработанный сайт должен соответствовать следующим требованиям к программной среде и средствам разработки:

- формат представления Web-документов – HTML версии 4.0 и выше;
- язык разработки серверных приложений – PHP версии 5.5 и выше;
- операционная система – Windows 8,8.1,10;
- Web-сервер – OpenServer;
- сервер баз данных – MySQL;

1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ ИНТЕРНЕТ- ПРИЛОЖЕНИЯ

1.1. Описание информационных и бизнес процессов в рамках предметной области Интернет- приложения.

Сайт используется для публикации кулинарных рецептов, которые распределены по категориям. Поделиться рецептом может любой пользователь.

Таким образом, в рамках данной предметной области можно выделить следующие информационные и бизнес процессы:

ВЗАИМОДЕЙСТВИЕ С ПОЛЬЗОВАТЕЛЕМ. Включает в себя регистрацию пользователя и предоставление ему возможность публиковать рецепты. При регистрации требуется указать имя, фамилию, логин, город, а также адрес электронной почты.

ФОРМИРОВАНИЕ ПУБЛИКАЦИИ. Основной контент web-сайта – рецепты. И при создании публикаций, помимо самого текста, пользователю так же необходимо указать название рецепта, кухню, категорию блюда и способ приготовления.

Документооборот Web-сайта (Рисунок.1.1) небольшой и ограничивается регистрацией или входом на сайт, просмотром рецептов, комментариев к ним, а также созданием рецептов.



Рисунок 1.1 – Схема документооборота сайта.

1.2 Построение модели Интернет- приложения

Разрабатываемый web-сайт должен содержать следующие страницы:

— Главная страница;

- Просмотр рецептов;
- Регистрация/Вход на сайт;
- Просмотр комментариев;
- Страница «О приложении».

Страницы сайта должны содержать элементы: Заголовок (Header), Подвал(Footer) и Навигационное меню (Nav). На рисунке 1.2 представлена модель сайта на примере страницы списка публикаций.

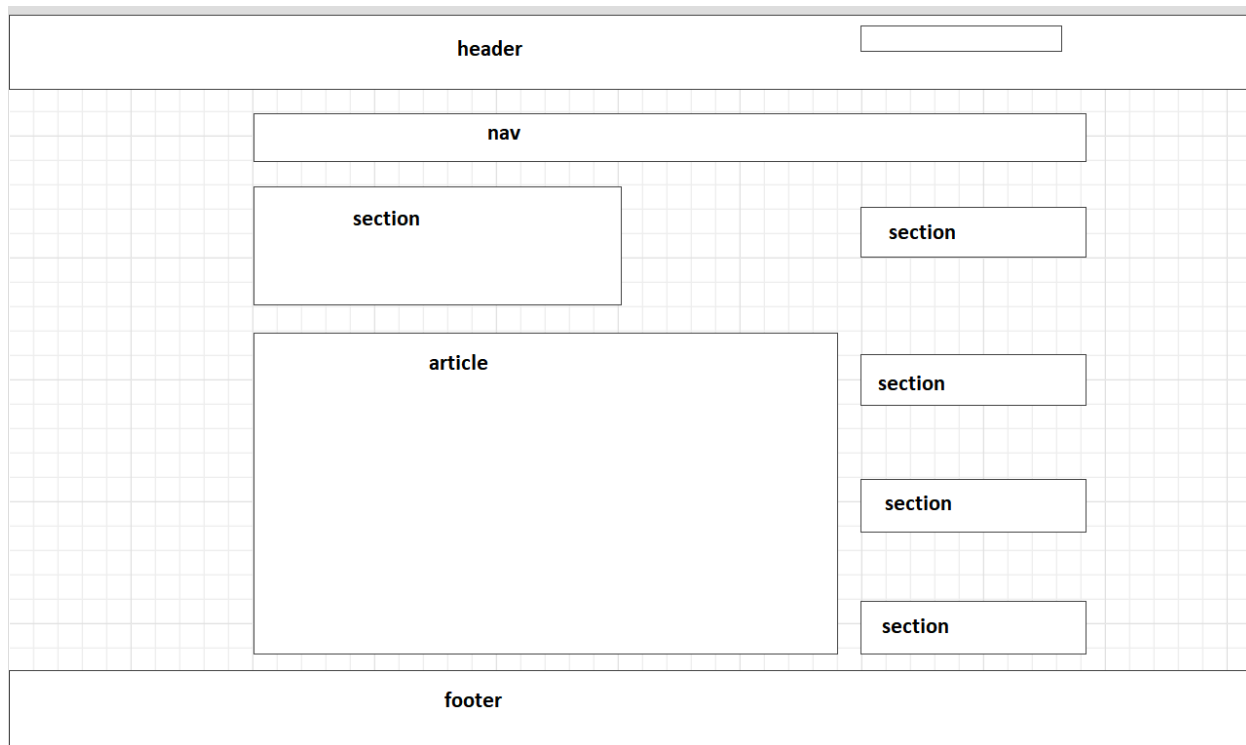


Рисунок 1.2 – Модель главной страницы.

Горизонтальное меню в заголовке сайта будем использовать для перехода между страницами сайта. Расположим на нём ссылки на основные страницы сайта. (Рисунок 1.3).

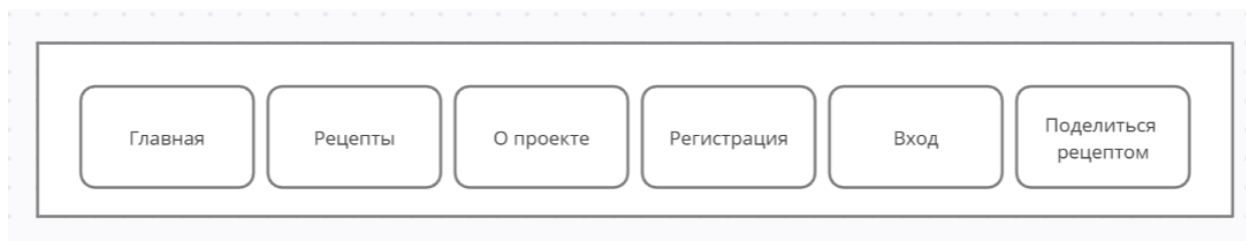


Рисунок 1.3 – Схема навигационного меню сайта.

Панель «Категории», расположенная в правой части сайта, предназначена для выбора способа сортировки рецептов.

В основной области контента отображаются рецепты сайта взятые из базы данных.

На странице «О приложении» - отображается статичная HTML-страница. А с страниц «Регистрация» и «Вход» отображаются формы регистрации или входа на сайт.

1.3 Техническое задание на разработку Web- сайта Интернет – проекта.

Техническое задание на разработку Web-сайта электронного журнала по системам разработки Интернет приложений прикреплено в приложении А.

2. ПРОЕКТИРОВАНИЕ WEB-САЙТА

2.1 Функциональная структура и модули сайта.

Функциональная структура сайта отображает функции, которые должны быть реализованы в разрабатываемом сайте. На её основе в дальнейшем и происходит разработка функционала всего сайта.

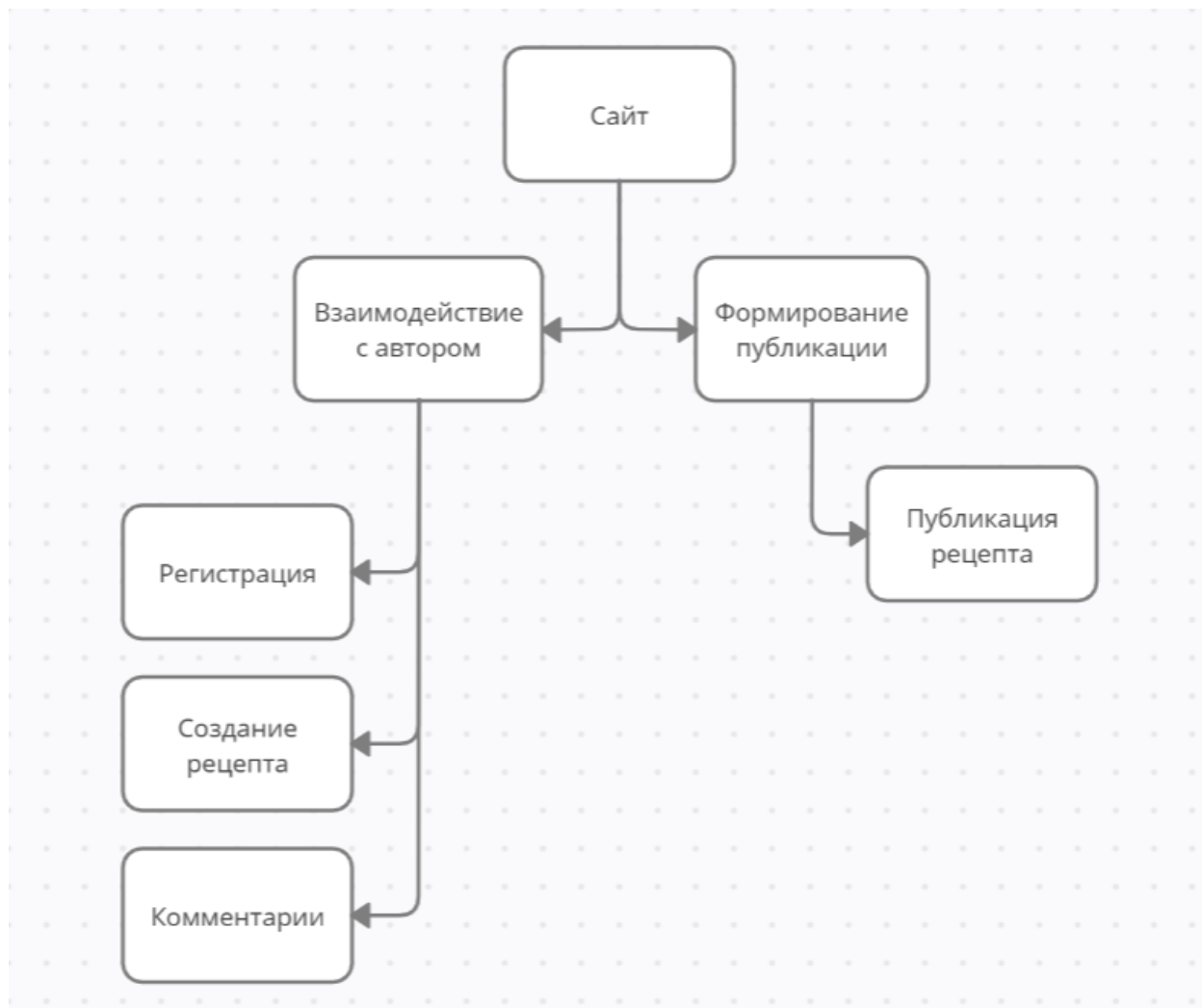


Рисунок 2.1 –Функциональная схема сайта.

К основным модулям разрабатываемого сайта можно отнести следующие модули:

1. Вывод рецептов.

Все публикации сайта, сохранённые в базе данных, выводятся на соответствующей странице сайта.

2. Вывод категорий.

Пользуясь боковой панелью, пользователь может выбрать способ сортировки рецептов по категориям.

3. Просмотр рецепта.

С помощью данного модуля можно просматривать отдельно каждый рецепт. Он отображает название публикации и содержание публикации.

4. Регистрация/Вход.

С помощью данного модуля пользователь может войти на сайт, либо зарегистрироваться. При регистрации он заполняет форму и указывает информацию о себе, после чего эти данные сохраняются в базу данных. При входе пользователь указывает свою электронную почту и пароль.

К дополнительным модулям сайта относятся:

1. Создание рецептов.

Данный модуль подразумевает создание публикаций в виде статических страниц, имеющих в составе как текстовую, так и графическую информацию.

2. Оценка рецептов.

Данный модуль подразумевает оценивание опубликованных рецептов.

2.2 Дизайн сайта

Дизайн сайта был разработан на основе модели и технического задания для сайта.

Заголовок (*header*) страницы сайта (Рисунок 2.2) представляет собой логотип сайта (сделанный с помощью программы Adobe Photoshop) формата png. и форму поиска по сайту.

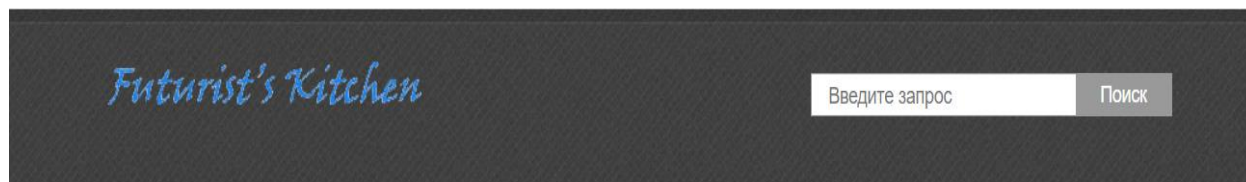


Рисунок 2.2 – Заголовок сайта.

Ниже (Рисунок 2.3) располагается навигационная панель. Она представляет собой HTML список, стилизованный в CSS как горизонтальный и включает в себя ссылки на основные страницы сайта.



Рисунок 2.3 – Навигационная панель.

В нижней части (*Footer*) сайта (Рисунок 2.4) находятся: краткое описание сайта с заголовком, маркированный список с ссылками и логотип сайта формата png.

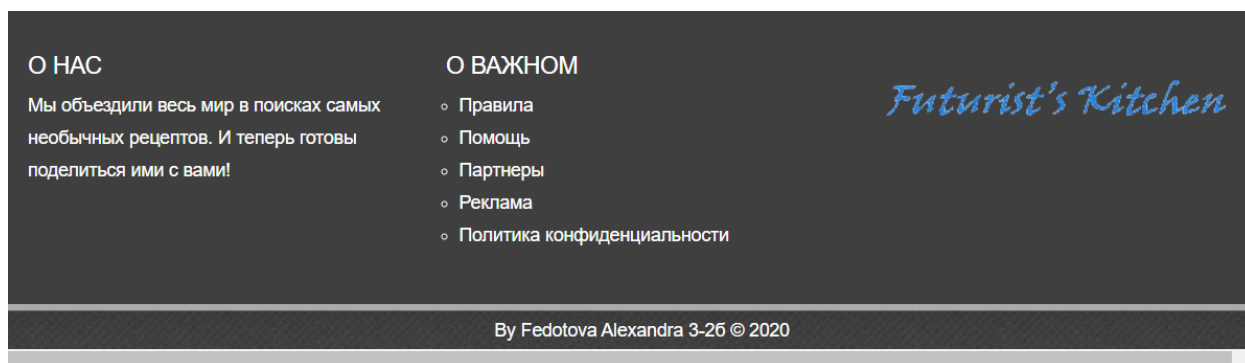


Рисунок 2.4 – Footer сайта.

Данные элементы присутствуют на каждой странице сайта. Остальной контент располагается между ними. На главной странице (Рисунок 2.5) с правой стороны присутствует боковая панель «Категории», на которые разбиты рецепты.

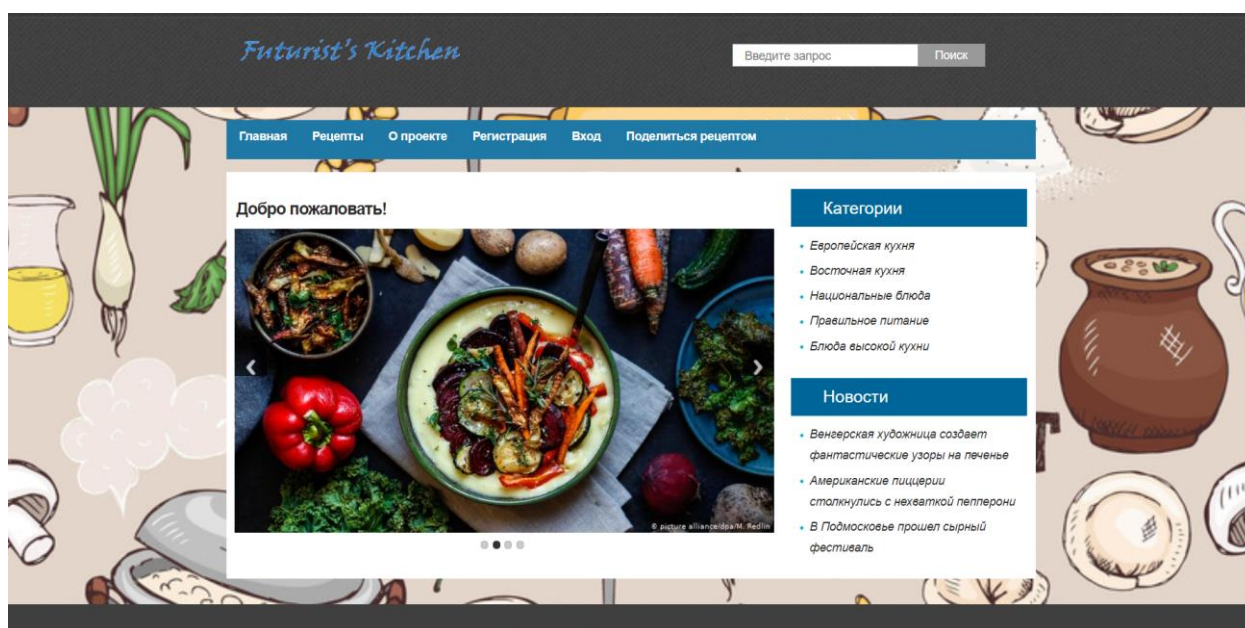


Рисунок 2.5 – Главная страница.

На некоторых страницах сайта боковая панель отсутствует, так как на таких страницах нет никаких функций, которыми можно было управлять с помощью отдельного меню. В качестве примера можно привести страницу «Регистрация» (Рисунок 2.6).

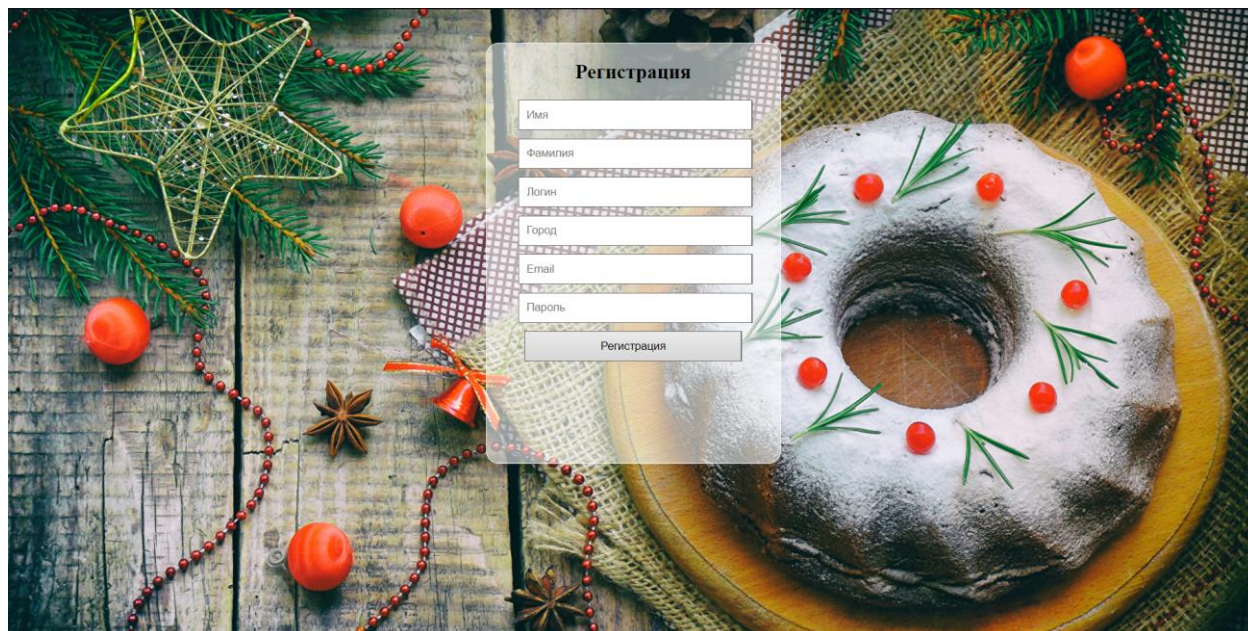


Рисунок 2.6 – Страница «Регистрация».

На сайте есть несколько форм для следующих действий:

- Регистрация (Рисунок 2.6).
Форма содержит 6 полей и 1 кнопку.
- Создание нового рецепта.

Данная форма (Рисунок 2.7) доступна всем пользователям (авторам) и предназначена для создания новых рецептов.

Создание нового рецепта

Название

Кухня

- выберите кухню - ▾

Категория

- выберите категорию - ▾

Способ

- выберите способ - ▾

Приготовление

Опубликовать

Рисунок 2.7 – Форма создания рецептов.

- Форма для написания комментариев.

Данной формой (Рисунок 2.8) может воспользоваться любой пользователь, пожелавший оставить комментарий к рецепту.

Имя

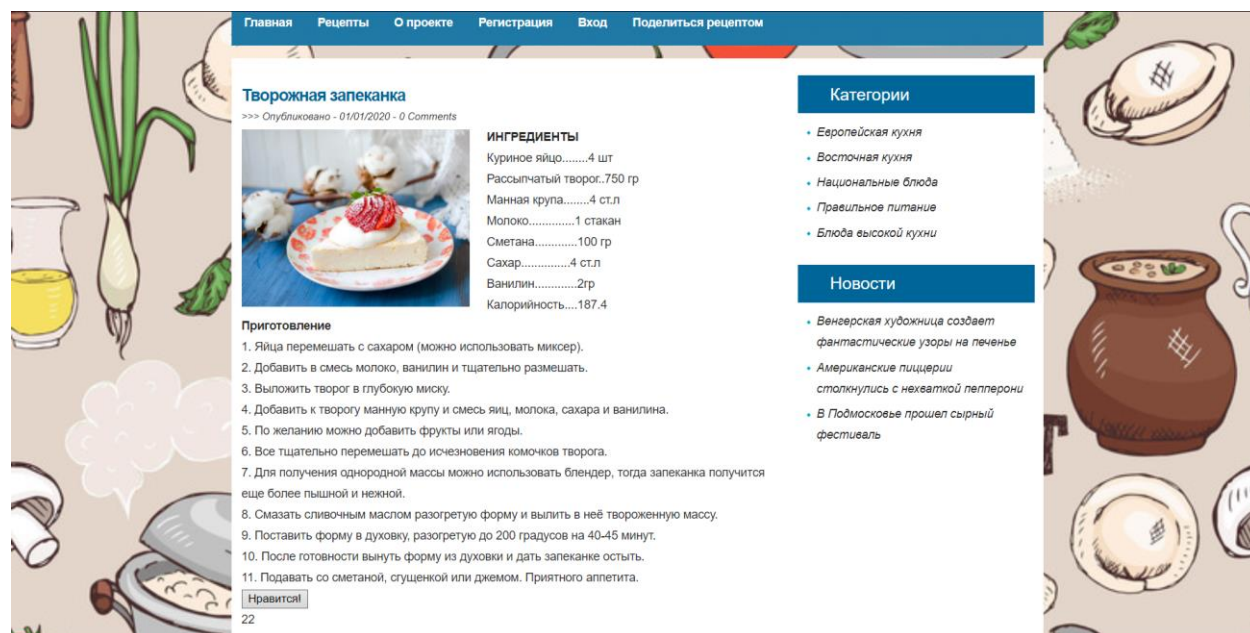
Ваш отзыв:

Комментарий

Отправить

Рисунок 2.8 – Форма для отправки комментария


Страница «Рецепты» (Рисунок 2.9) содержит публикации, располагающиеся по центру. Публикация может содержать в себе не только текстовую, но еще и графическую информацию. Само содержание публикации хранится уже в виде *HTML*-текста в базе данных.



Главная Рецепты О проекте Регистрация Вход Поделиться рецептом

Творожная запеканка

>>> Опубликовано - 01/01/2020 - 0 Comments



ИНГРЕДИЕНТЫ

- Куриное яйцо.....4 шт
- Рассыпчатый творог.....750 гр
- Манная крупа.....4 ст.л
- Молоко.....1 стакан
- Сметана.....100 гр
- Сахар.....4 ст.л
- Ванилин.....2гр
- Калорийность.....187.4

Приготовление

1. Яйца перемешать с сахаром (можно использовать миксер).
2. Добавить в смесь молоко, ванилин и тщательно размешать.
3. Выложить творог в глубокую миску.
4. Добавить к творогу манную крупу и смесь яиц, молока, сахара и ванилина.
5. По желанию можно добавить фрукты или ягоды.
6. Все тщательно перемешать до исчезновения комочков творога.
7. Для получения однородной массы можно использовать блендер, тогда запеканка получится еще более пышной и нежной.
8. Смазать сливочным маслом разогретую форму и вылить в неё творожную массу.
9. Поставить форму в духовку, разогретую до 200 градусов на 40-45 минут.
10. После готовности вынуть форму из духовки и дать запеканке остыть.
11. Подавать со сметаной, сгущенкой или джемом. Приятного аппетита.

22

Категории

- Европейская кухня
- Восточная кухня
- Национальные блюда
- Правильное питание
- Блюда высокой кухни

Новости

- Венгерская художница создает фантастические узоры на печенье
- Американские пиццерии столкнулись с нехваткой пепперони
- В Подмосковье прошел сырный фестиваль

Рисунок 2.9 – Страница «Рецепты».

3. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ САЙТА

В ходе выполнения работы была спроектирована структура базы данных согласно варианту (Таблица 3.1). На основании данных этой таблицы приступаем к построению логической модели.

| Элемент данных | Параметры элемента |
|----------------|--|
| Ингредиенты | Id рецептов, название |
| Категории | Название |
| Кухня | Название |
| Рецепты | Название, id кухни, id категории, id способа, описание |
| Способ | Название |
| Пользователи | Имя пользователя, фамилия, город, e-mail, логин, пароль. |

Таблица 3.1 – Описание массивов данных базы сайта.

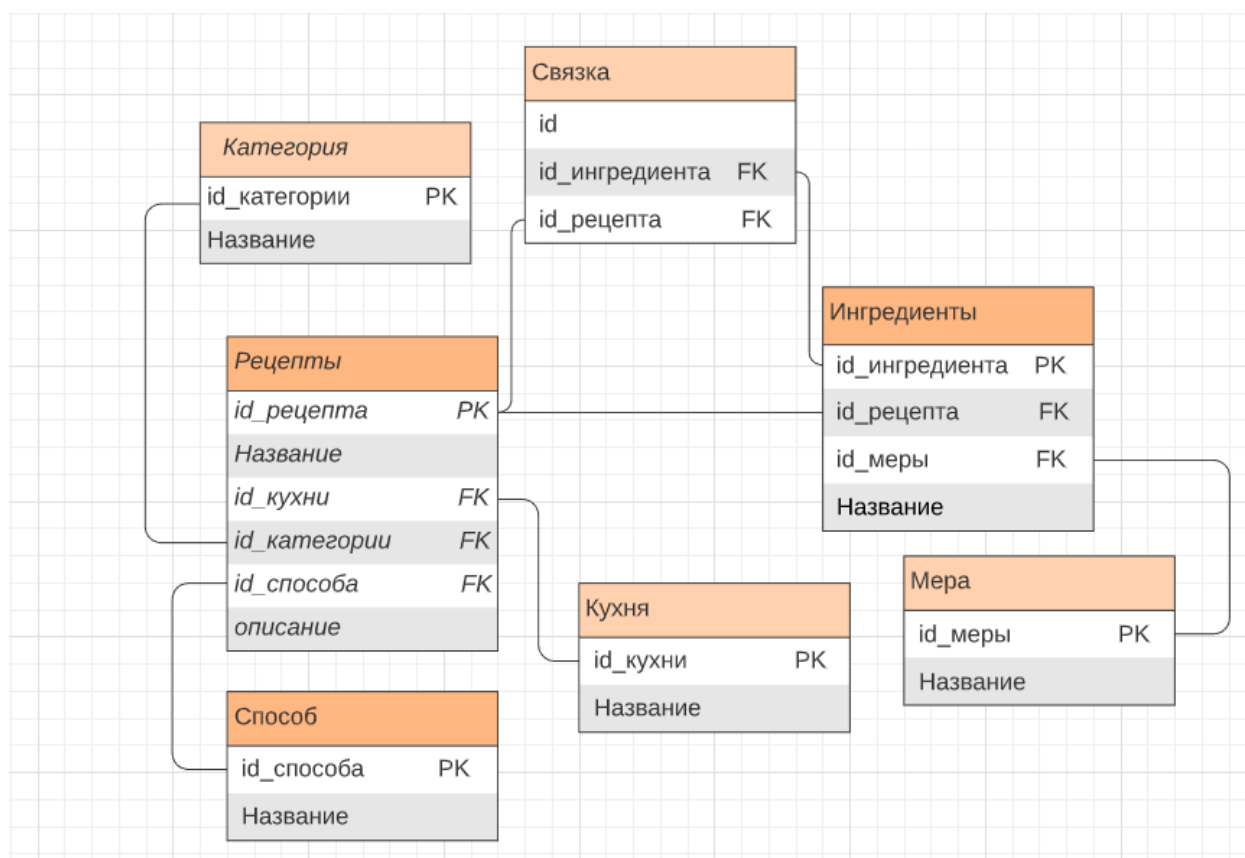


Рисунок 3.1 – Логическая базы данных.

Логическая модель разрабатываемой базы данных (Рисунок 3.1) подразумевает отношения, соответствующие каждому элементу данных. В отношении

ИНГРЕДИЕНТЫ атрибут Рецепт является внешним ключом, связанным с первичным ключом таблицы РЕЦЕПТЫ.

Переходим к физической реализации базы данных. Для этого воспользуемся программной средой phpMyAdmin.

Структура базы данных выглядит следующим образом (Рисунки 3.2 – 3.7):

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---|---|---------|--------------------|----------|------|--------------|-------------|----------------|---|
| <input type="checkbox"/> | 1 | ing_id  | int(11) | | | Нет | Нет | | AUTO_INCREMENT |    |
| <input type="checkbox"/> | 2 | res_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 3 | mera_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 4 | name | text | utf8mb4_general_ci | | Нет | Нет | | |    |

Рисунок 3.2 – Структура таблицы *ing*.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---|--|----------|--------------------|----------|------|--------------|-------------|---------------|---|
| <input type="checkbox"/> | 1 | kategor_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 2 | name | char(15) | utf8mb4_general_ci | | Нет | Нет | | |    |

Рисунок 3.3 – Структура таблицы *kategor*.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---|--|---------|--------------------|----------|------|--------------|-------------|----------------|---|
| <input type="checkbox"/> | 1 | kuhnia_id  | int(11) | | | Нет | Нет | | AUTO_INCREMENT |    |
| <input type="checkbox"/> | 2 | name | text | utf8mb4_general_ci | | Нет | Нет | | |    |

Рисунок 3.4 – Структура таблицы *kuhnia*.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---|--|----------------|--------------------|----------|------|--------------|-------------|----------------|---|
| <input type="checkbox"/> | 1 | res_id  | int(11) | | | Нет | Нет | | AUTO_INCREMENT |    |
| <input type="checkbox"/> | 2 | name | varchar(40) | utf8mb4_general_ci | | Нет | Нет | | |    |
| <input type="checkbox"/> | 3 | kuhnia_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 4 | kategor_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 5 | sposob_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 6 | description | varchar(10000) | utf8mb4_general_ci | | Нет | Нет | | |    |

Рисунок 3.5 – Структура таблицы *res*.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---|---|---------|--------------------|----------|------|--------------|-------------|---------------|---|
| <input type="checkbox"/> | 1 | sposob_id  | int(11) | | | Нет | Нет | | |    |
| <input type="checkbox"/> | 2 | name | text | utf8mb4_general_ci | | Нет | Нет | | |    |

Рисунок 3.6 – Структура таблицы *sposob*.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|--------------------------|---|---|-------------|--------------------|----------|------|--------------|-------------|----------------|---|
| <input type="checkbox"/> | 1 | user_id  | int(11) | | | Нет | Нет | | AUTO_INCREMENT |    |
| <input type="checkbox"/> | 2 | user_name | varchar(50) | utf8mb4_general_ci | | Нет | Нет | | |    |
| <input type="checkbox"/> | 3 | surname | varchar(50) | utf8mb4_general_ci | | Нет | Нет | | |    |
| <input type="checkbox"/> | 4 | town | varchar(15) | utf8mb4_general_ci | | Нет | Нет | | |    |
| <input type="checkbox"/> | 5 | user_email | varchar(50) | utf8mb4_general_ci | | Нет | Нет | | |    |
| <input type="checkbox"/> | 6 | login | varchar(15) | utf8mb4_general_ci | | Нет | Нет | | |    |
| <input type="checkbox"/> | 7 | parol | varchar(10) | utf8mb4_general_ci | | Нет | Нет | | |    |

Рисунок 3.7 – Структура таблицы *user*.

Также схему спроектированной базы данных можно посмотреть в том же phpMyAdmin во вкладке Дизайн (Рисунок 3.8). На схеме так же показаны и связи между таблицами.

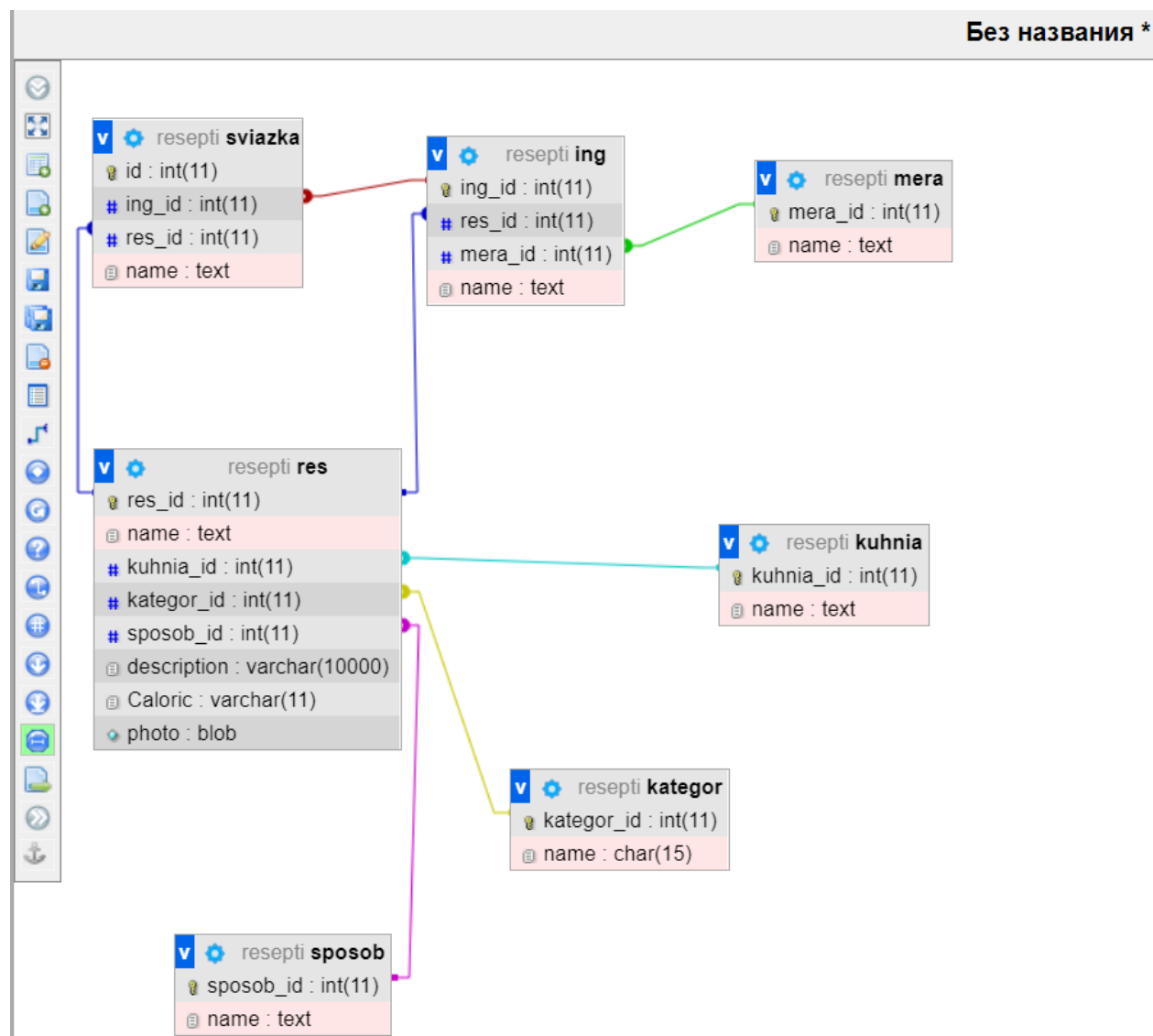


Рисунок 3.8 – Схема базы данных в phpMyAdmin.

Теперь заполняем макетными данными все созданные таблицы. Результат заполнения представлен в виде таблиц данных из *phpMyAdmin* на рисунках 3.9-3.13.

| ←T→ | | | | ing_id | res_id | mera_id | name |
|--------------------------|--|--|--|--------|--------|---------|------------------------|
| <input type="checkbox"/> | | | | 1 | 2 | 5 | Куриное яйцо |
| <input type="checkbox"/> | | | | 2 | 2 | 1 | Творог |
| <input type="checkbox"/> | | | | 3 | 2 | 1 | Манная крупа |
| <input type="checkbox"/> | | | | 4 | 2 | 2 | Молоко |
| <input type="checkbox"/> | | | | 5 | 2 | 1 | Сахар |
| <input type="checkbox"/> | | | | 6 | 6 | 1 | Репчатый лук |
| <input type="checkbox"/> | | | | 7 | 6 | 1 | Мука пшеничная |
| <input type="checkbox"/> | | | | 8 | 6 | 2 | Говяжий бульон |
| <input type="checkbox"/> | | | | 9 | 6 | 2 | Куриный бульон |
| <input type="checkbox"/> | | | | 10 | 6 | 1 | Сыр грюйгер |
| <input type="checkbox"/> | | | | 11 | 6 | 4 | Соль |
| <input type="checkbox"/> | | | | 12 | 6 | 4 | Перец черный молотый |
| <input type="checkbox"/> | | | | 13 | 3 | 1 | Свёкла |
| <input type="checkbox"/> | | | | 14 | 3 | 1 | Картофель |
| <input type="checkbox"/> | | | | 15 | 3 | 1 | Морковь |
| <input type="checkbox"/> | | | | 16 | 3 | 1 | Капуста квашеная |
| <input type="checkbox"/> | | | | 17 | 3 | 1 | Солёные огурцы |
| <input type="checkbox"/> | | | | 18 | 3 | 3 | Масло растительное |
| <input type="checkbox"/> | | | | 19 | 4 | 1 | Рис |
| <input type="checkbox"/> | | | | 20 | 4 | 1 | Говядина |
| <input type="checkbox"/> | | | | 21 | 4 | 2 | Вода |
| <input type="checkbox"/> | | | | 22 | 3 | 1 | Горох консервированный |
| <input type="checkbox"/> | | | | 23 | 5 | 1 | Мясной фарш |
| <input type="checkbox"/> | | | | 24 | 5 | 1 | Томатная паста |
| <input type="checkbox"/> | | | | 25 | 5 | 1 | Масло сливочное |
| <input type="checkbox"/> | | | | 26 | 5 | 3 | Масло оливковое |
| <input type="checkbox"/> | | | | 27 | 5 | 5 | Листы лазаньи |
| <input type="checkbox"/> | | | | 28 | 5 | 1 | Сыр пармезан |
| <input type="checkbox"/> | | | | 29 | 2 | 1 | Сметана |
| <input type="checkbox"/> | | | | 30 | 1 | 1 | Горох |

Консоль

Рисунок 3.9 – Заполненная макетными данными таблица ing.

| ←T→ | | | | kategor_id | name |
|--------------------------|--|--|--|------------|--------------|
| <input type="checkbox"/> | | | | 1 | Первые блюда |
| <input type="checkbox"/> | | | | 2 | Вторые блюда |
| <input type="checkbox"/> | | | | 3 | Салаты |
| <input type="checkbox"/> | | | | 4 | Закуски |
| <input type="checkbox"/> | | | | 5 | Десерты |
| <input type="checkbox"/> | | | | 6 | Консервация |

Рисунок 3.10 – Заполненная макетными данными таблица kategor.

| | | | | kuhnia_id | name |
|--|--|--|--|-----------|-------------------------|
| | | | | 1 | Европейская кухня |
| | | | | 2 | Азиатская кухня |
| | | | | 3 | Национальная кухня |
| | | | | 4 | Средиземноморская кухня |
| | | | | 5 | Вегетарианская кухня |
| | | | | 6 | Пост |
| | | | | 7 | Правильное питание |
| | | | | 8 | Диета |

Рисунок 3.11 – Заполненная макетными данными таблица kuhnia.

| | | | | res_id | name | kuhnia_id | kategor_id | sposob_id | description |
|--|--|--|--|--------|-----------------------|-----------|------------|-----------|----------------|
| | | | | 1 | Гороховый суп | 1 | 1 | 2 | |
| | | | | 2 | Твороженная запеканка | 1 | 5 | 4 | |
| | | | | 3 | Винегрет | 1 | 3 | 7 | |
| | | | | 4 | Плов | 2 | 2 | 3 | |
| | | | | 5 | Лазанья | 1 | 2 | 4 | |
| | | | | 6 | Луковый суп | 1 | 1 | 2 | |
| | | | | 7 | проба | 4 | 2 | 6 | fdhsdfhsgghsgs |

Рисунок 3.12 – Заполненная макетными данными таблица res.

| | | | | sposob_id | name |
|--|--|--|--|-----------|-----------|
| | | | | 1 | Жарка |
| | | | | 2 | Варка |
| | | | | 3 | Тушение |
| | | | | 4 | Запекание |
| | | | | 5 | На пару |
| | | | | 6 | Соление |
| | | | | 7 | Нарезка |

Рисунок 3.13 – Заполненная макетными данными таблица sposob.

| | | | | user_id | user_name | surname | town | user_email | login | parol |
|--|--|--|--|---------|-----------|------------|--------------|----------------|-------|---------|
| | | | | 5 | Елена | Прекрасная | Чита | 123@mail.ru | helen | 1234567 |
| | | | | 7 | Алиса | Чехова | Новгород | al@mail.ru | alice | 345678 |
| | | | | 8 | Женя | Власова | Новочеркасск | qwerty@mail.ru | 1234 | asdfgh |
| | | | | 9 | Софья | Рябова | Волгоград | sofa@mail.ru | sofa | asdfg |

Рисунок 3.14 – Заполненная макетными данными таблица user.

Теперь переходим к проектированию всех запросов к базе данных, которые потребуются при разработке сайта. Для доступа с сайта к базе данных используются запросы, написанные на языке *SQL*. Запросы делятся на три типа:

- Функциональные;
- Информационные - поисковые;
- Информационные – агрегативные;

1. Функциональные запросы.

К функциональным запросам, реализованным для работы веб-сайта можно отнести следующие запросы:

1.1 Зарегистрировать пользователя.

```
INSERT INTO `user` (`user_name`, `surname`, `login`, `town`, `user_email`, `parol`)
VALUES('".$user_name."', ".$surname."', ".$login."', ".$town."', ".$user_email."', ".$parol."");
```

1.2 Создать публикацию.

```
INSERT INTO res(name, kuhnia_id, kategor_id, sposob_id, description) VALUES('".$name."', ".$kuhnia."',
".$kategor."', ".$sposob."', ".$description."");
```

2. Информационные – поисковые запросы.

2.1 Показать список всех рецептов.

```
SELECT * FROM res;
```

2.2 Показать список всех рецептов с сортировкой по указанному типу.

```
SELECT * FROM res ORDER BY @sort_type_value;
```

2.3 Показать рецепт по id.

```
SELECT * FROM publication WHERE id = @id_value;
```

3. Информационные – агрегативные запросы.

3.1 Показать количество рецептов по конкретной категории:

```
SELECT COUNT(*) FROM `res` WHERE kategor_id=@ kategor_id_value;
```

3.2 Показать количество рецептов по конкретному способу

```
SELECT COUNT(*) FROM `res` WHERE sposob_id=@ sposob_id_value;
```


4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ САЙТА

Аналогичным образом создаём скрипт *check_form.js* для проверки заполнения формы регистрации на сайте. После того, как в форме входа на сайт были указаны данные и после успешной проверки данных *javascript*-скриптом, запускается файл *login.php*.

Регистрация автора на сайте происходит с помощью скрипта *registration.php*. Сначала данный скрипт принимает все данные методом *POST*, а затем проверяет их на полноту. Если были переданы не все требуемые данные, процесс прервётся и выведется сообщение «Не все данные введены, проверьте все еще раз!». Если все данные были переданы, то они запишутся в соответствующие переменные. Далее создаётся соединение с базой данных, пол чего формируется *SQL*-запрос *INSERT* в строке и выполняется с помощью функции *mysqli_query*. Если запрос будет выполнен успешно, то будет выведено соответствующее сообщение.

Также есть страница добавления рецептов. После создания публикации и проверки формы запускается программный код, вставляющий публикацию в базу данных. Еще есть страница добавления комментариев. Программный код вставляет полученные данные в базу данных, а потом выводит их на сайт.

5. ОЦЕНКА ЭФФЕКТИВНОСТИ И ТЕСТИРОВАНИЕ САЙТА

На данном этапе, когда Web-сайт уже разработан, его необходимо протестировать и оценить его эффективность. Для этого воспользуемся инструментом *Google Lighthouse*, входящей в состав браузера *Google Chrome*. Данный инструмент позволяет оценить доступность и эффективность веб-сайта. Запускаем его нажатием клавиши *F12* в браузере и переходя на вкладку *Lighthouse*. После обработки сайта, получаем информацию (Рисунок 5.1).

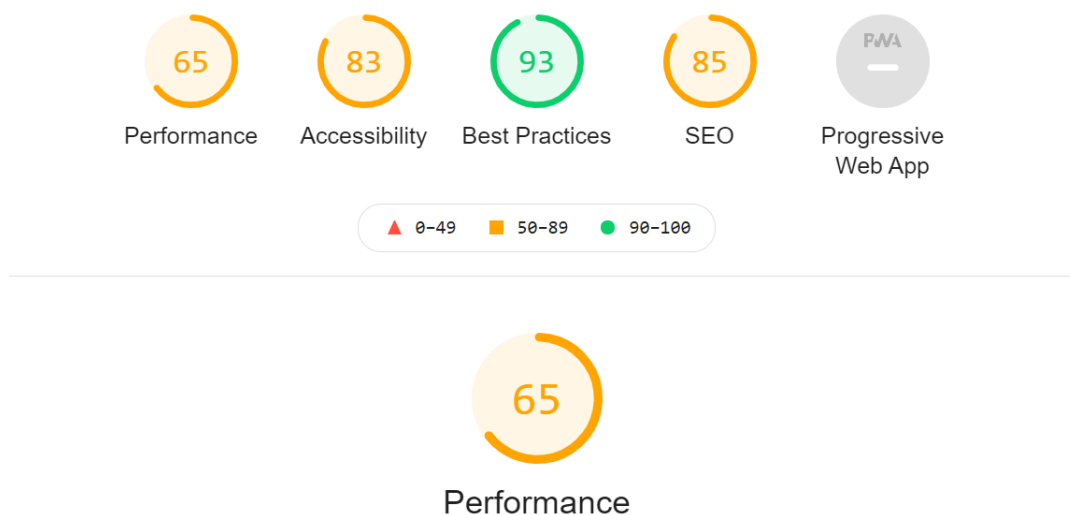


Рисунок 5.1 - Результат работы инструмента Google Lighthouse.

В результате чего получаем следующие данные:

- Производительность: 100;
- Доступность: 70;
- Соответствия рекомендациям Google: 93;
- SEO: 78.

С помощью клавиши *F12* в браузере *Opera* переходим на вкладку сеть для того, чтобы узнать размер каждой страницы и время её загрузки (Рисунок 5.2). Полученные результаты представим в виде таблицы.

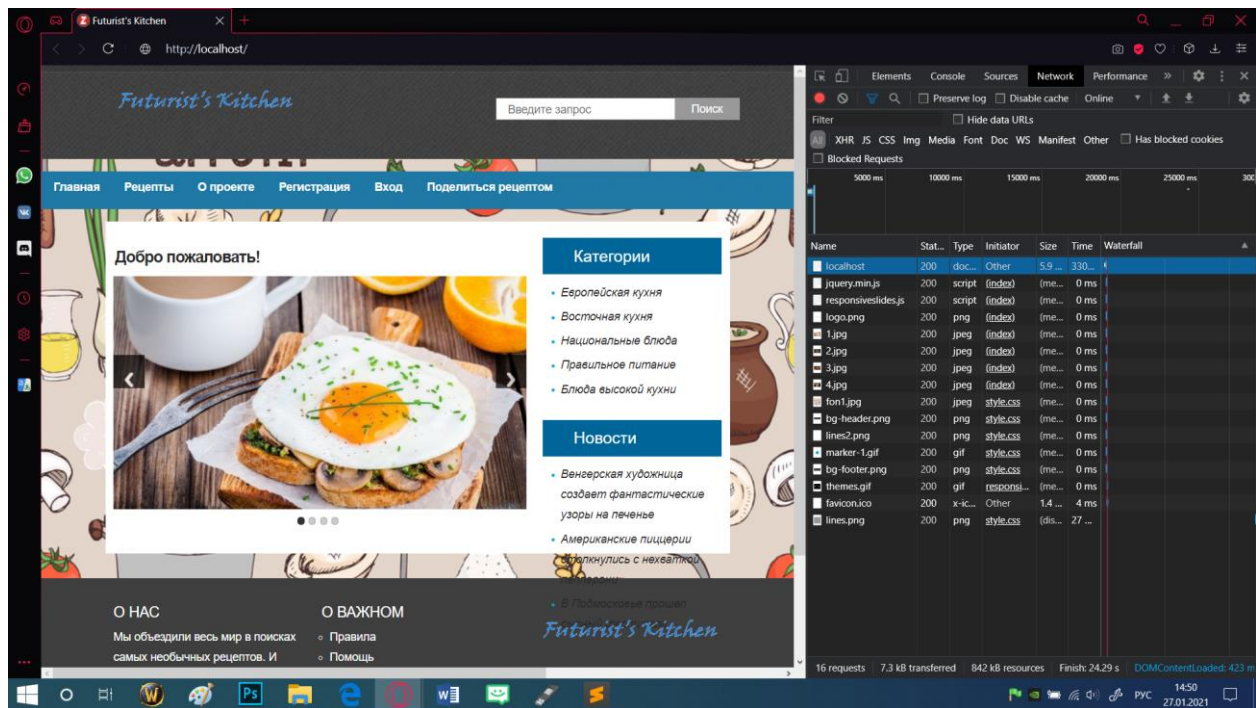


Рисунок 5.2 – Мониторинг сети в браузере *Opera*.

| Страница | Размер страницы |
|---------------------|-----------------|
| Главная | 7.3 КБ |
| Рецепты | 12.5 КБ |
| О проекте | 6.9 КБ |
| Регистрация | 3.8 КБ |
| Поделиться рецептом | 9.6 КБ |

Таблица 5.1 – Размеры html-страниц.

ЗАКЛЮЧЕНИЕ

Таким образом, в ходе выполнения курсовой работы был разработан Web-сайт электронного журнала по системам разработки Интернет - приложений.

Разработанный сайт соответствует требованиям, описанным в техническом задании, а значит, поставленная цель была выполнена.

Так же в процессе выполнения работы и разработки сайта были получены навыки описания информационных и бизнес процессов, построения модели Интернет – приложения, опыт составления технического задания на разработку Web-сайта. Также был получен опыт проектирования Web-сайта, разработки базы данных и программной реализации сайта.

Также разработанный сайт был протестирован, и также была оценена его эффективность.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Руководство по HTML5 и CSS3 // METANIT.COM - Сайт о программировании
URL: <https://metanit.com/web/html5/> (дата обращения: 26.12.2020).
2. Руководство по JavaScript // METANIT.COM - Сайт о программировании URL:
<https://metanit.com/web/javascript/> (дата обращения: 26.12.2020).
3. Руководство по MySQL // METANIT.COM - Сайт о программировании URL:
<https://metanit.com/sql/mysql/> (дата обращения: 26.12.2020).
4. Руководство по PHP // php URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 26.12.2020).
5. Lighthouse // Google Developers URL:
<https://developers.google.com/web/tools/lighthouse?hl=ru> (дата обращения: 26.12.2020).

ПРИЛОЖЕНИЯ

Приложение А – Техническое задание на разработку Web-сайта.

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1 Полное наименование системы и ее условное обозначение:

Web-сайт кулинарных рецептов.

1.2 Краткое наименование системы: Futurist's Kitchen.

1.3 Имя сайта (название домена): www.F'K.ru

1.4 Наименования организации-заказчика и организаций-участников работ:

Заказчиком системы является ООО «Futurist's Kitchen».

1.5 Перечень документов, на основании которых создается система: Основанием для разработки является типовой договор на разработки Web- проекта.

1.6 Плановые сроки начала и окончания работы по созданию системы Плановый срок начала работ по созданию – 4 сентября 2020года. Плановый срок окончания работ - 20 декабря 2020 года.

1.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы:

Система передается в виде функционирующего комплекса на базе средств вычислительной техники Заказчика и Исполнителя в сроки, установленные Договором. Приемка системы осуществляется комиссией в составе уполномоченных представителей Заказчика и Исполнителя.

1.8Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ:

При разработке автоматизированной системы и создании проектноэксплуатационной документации Исполнитель руководствуется требованиями следующих нормативных документов:

- ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;
- ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем;
- Панфилов А.Н. Разработка приложений для Интернет. Методические указания по выполнению курсовой работы.
- Постановление Правительства РФ от 10.07.2013 № 582

2. НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы.

Предоставление информации из мира кухни : сборник кулинарных рецептов, новости, списки лучших ресторанов и поваров.

2.2 Цели создания системы:

Сайт должен способствовать привлечению интереса аудитории к контенту. А также нахождению деловых партнеров (владельцы крупных ресторанов, кулинарные школы и Академии и т.д.).

3. ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

3.1 Краткие сведения об объекте автоматизации.

Объектом автоматизации является кулинарный Web-сайт «Futurist's Kitchen». Основной деятельностью сайта является обеспечение пользователей контентом.

4. ТРЕБОВАНИЯ К СИСТЕМЕ

4.1 Язык сайта.

Русский и английский.

4.2 Объем и состав текстовой информации.

Объём и состав текстовой информации, вносимой на одну страницу сайта взамен старой информации, не должен превышать более чем на 30% общий объём информации, уже имеющейся на обновляемой странице.

4.3 Основные ключевые слова, по которым сайт должны находить по запросам в поисковых системах и Интернет – каталогах.

«Рецепт», «кулинария», «кухня», «еда», «суп», «салат», «приготовить», «ингредиенты», «сварить», «книга рецептов», «»

4.4 Объём и состав графической информации.

Объём и состав графической информации, вносимой на одну страницу сайта взамен старой информации, не должен превышать более чем на 30% общий объём информации, уже имеющейся на обновляемой странице.

4.5 Общий объём и состав текстовой и графической информации.

10 л.

4.7 Количество страниц сайта.

Сайт должен содержать следующие отдельные html страницы: 1 – Главная (домашняя) страница; 2 – Публикации; 3 – Категории; 4 – Публикация нового рецепта; 5 – О приложении; 6 – Регистрация; 7 – Вход.

4.6 Предполагаемая возрастная аудитория сайта

Все возрастные категории.

4.8 Кнопки управления (навигация сайта)

С каждой страницы сайта должен быть обеспечен переход (установлена гиперссылка) на начальную страницу сайта.

4.9 Схема сайта

Начальная страница сайта должна содержать гиперссылки, обеспечивающие переход с нее на не менее чем 80 % страниц сайта, но не более чем 150 гиперссылок.

4.10 Общий объём сайта, Мб

Не оговаривается.

4.11 Оформление рисунков

Не оговаривается.

4.12 Пропускная способность линии связи

Среднее время загрузки страниц не должно превышать 28 секунд при скорости соединения 28.8 Кбит/сек.

4.13 Основной диапазон разрешения мониторов, на которых будет просматриваться сайт

От 1024x768 до 1240x1024 пикселей

4.14 Минимальное разрешение монитора, на котором будет просматриваться сайт 1024x768 пикселей

4.15 Основной браузер, которым будет просматриваться сайт, и его минимальная версия.

Google Chrome 86.0.4240.75

Mozilla Firefox 51.0.1

4.16 Цветовая палитра

На усмотрение исполнителя.

4.17 Общий фон сайта.

На усмотрение исполнителя.

4.18 Размер и вид шрифта сайта.

Размер шрифта основного текста сайта - 14. шрифт – Helvetica, serif. Для заголовков используются стандартные размеры шрифта, предусмотренные заголовками в языке HTML. Шрифт предназначен как для основного текста сайта, так и для заголовков и меню сайта.

4.19 Срок разработки сайта.

См. пункт 1.6.

4.20 Порядок передачи сайта.

Исполнитель передает сайт Заказчику на внешнем носителе, а также логин, пароль и название (код передачи данных) по протоколу ftp. Заказчик обязан проверить наличие грамматических и орфографических ошибок на сайте в течение трех рабочих дней. Обнаруженные ошибки Исполнитель обязан устранить в течение трех рабочих дней.

4.21 Дополнительные условия.

Каждая страница сайта должна содержать логотип и название компании. Внизу на каждой странице сайта должна быть указана контактная информация.

Приложение Б – Исходный код страниц Web-сайта и Web-приложений.

Главная страница – index.php

```
<?php
require_once("db_connect.php");
require ('header.php');
?>

<section id="content">
    <div class="zerogrid">
        <div class="row block">
            <div class="main-content col11">
                <?php require ('content-home.php'); ?>
            </div>
            <?php require ('sidebar.php'); ?>
        </div>
    </div>
</section>
<?php require ('footer.php'); ?>
```

Заголовок сайта - header.php

```
<?php
    session_start();
?>
<?php $url = $_SERVER["REQUEST_URI"];?>

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Futurist's Kitchen</title>
    <meta name="description" content="meta name="description">
    <meta name="author" content="www.Futurist'sKitchen.com">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <link rel="stylesheet" href="css/zerogrid.css">
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/responsive.css">
    <link rel="stylesheet" href="css/responsiveslides.css" />

    <link href='./images/favicon.ico' rel='icon' type='image/x-icon'/>

    <script src="js/jquery.min.js"></script>
    <script src="js/responsiveslides.js"></script>
    <script>
$(function () {
    $("#slider").responsiveSlides({
        auto: true,
        pager: true,
```

```

        nav: true,
        speed: 500,
        maxwidth: 800,
        namespace: "centered-btns"
    });
});
</script>

</head>
<body>
<header>
    <div class="zerogrid">
        <div class="row">
            <div class="col05">
                <div id="logo"></div>
            </div>
            <div class="col06 offset05">
                <div id='search-box'>
                    <form action="search_result.php" id='search-form' method='POST'
target='_top'>
                        <input id='search-text' name='q' placeholder='Введите запрос'
type='text'/>
                        <button id='search-button'
type='submit'><span>Поиск</span></button>
                    </form>
                </div>
            </div>
        </div>
    </div>
</header>
<nav>
    <ul>
        <li><a href="/index.php">Главная</a></li>
        <li><a href="/res.php">Рецепты</a></li>
        <li><a href="/about.php">О проекте</a></li>
        <li><a href="/login.php">Регистрация</a></li>
        <li><a href="/vhod.php">Вход</a></li>
        <li><a href="/published.php">Поделиться рецептом</a></li>
    </ul>
</nav>
Контент на главной странице – content-home.php
<div class="rslides_container">
<article>
<div class="heading">
<h2>Добро пожаловать!</h2>

</article>

```



```
<ul class="rslides" id="slider">
<li></li>
<li></li>
<li></li>
<li></li>
</ul>
</div>
```

Боковая панель страницы – sidebar.php

```
<div class="sidebar col05">
<section>
<div class="heading">Категории</div>
<div class="content">
<ul class="list">
<li><a href="http://www.zerotheme.com"><i>Европейская кухня</i></a></li>
<li><a href="http://www.zerotheme.com"><i>Восточная кухня</i></a></li>
<li><a href="http://www.zerotheme.com"><i>Национальные блюда</i></a></li>
<li><a href="http://www.zerotheme.com"><i>Правильное питание</i></a></li>
<li><a href="http://www.zerotheme.com"><i>Блюда высокой кухни</i></a></li>
</ul>
</div>
</section>
<section>
<div class="heading">Новости </div>
<div class="content">
<ul class="list">

<li><a href="#"></a><i>Венгерская художница создает фантастические узоры на печенье</i></li>

<li><a href="#"></a><i>Американские пиццерии столкнулись с нехваткой пепперони</i></li>

<li><a href="#"></a><i>В Подмоскowie прошел сырный фестиваль</i></li>
</ul>
</div>
</section>

</div>
```

Подвал страницы – footer.php

```
<footer>
<div class="zerogrid">
<div class="row">
<section class="col-1-3">
<div class="heading">О нас</div>
<div class="content">
Мы объездили весь мир в поисках самых необычных рецептов. И теперь готовы поделиться ими с вами!
</div>
</section>
<section class="col-1-3">
<div class="heading">О важном</div>
<div class="content">
<ul>
<li><a href="http://www.zerotheme.com">Правила</a></li>
```

```
<li><a href="http://www.zeroheme.com">Помощь</a></li>
<li><a href="http://www.zeroheme.com">Партнеры</a></li>
<li><a href="http://www.zeroheme.com">Реклама</a></li>
<li><a href="http://www.zeroheme.com">Политика конфиденциальности</a></li>
</ul>
</div>
</section>
<section class="col-1-3">
<div class="zerogrid">
<div class="row">
<div class="col05">
<div id="logo"><a href=""></a></div>
</div>
</div>
</div>
</div>
</section>
</div>
</div>
</footer>
```

<p align="justify"><i>"Futurist's Kitchen"("Кухня футуриста")</i> - это сайт, основанный на огромном архиве различных кулинарных рецептов стран мира, включая национальную кухню. Естественно, всё с фотографиями самих блюд и подробного описания процессов приготовлений.

Помимо поиска кулинарных изысков, Вы можете делиться своими рецептами и хитростями с другими пользователями, обмениваться опытом, а также оценивать чужие творения.

Хотите поделиться с миром своим талантом? Создав профиль, Вы сможете публиковать рецепты непосредственно на самом сайте.

Рецепты с наибольшим количеством положительных оценок будут представлены на главной странице сайта в колонке "Рецепты месяца".

</p>

</div>

</article>

Страница «Рецепты» - res.php

```
<?php require ('header.php');
```

```
require ('db_connect.php');
```

```
?>
```

```
<section id="content">
```

```
    <div class="zerogrid">
```

```
        <div class="row block">
```

```
            <div class="main-content col11">
```

```
                <?php require ('content-home2.php'); ?>
```

```
            </div>
```

```
        <?php require ('sidebar.php'); ?>
```

```
    </div>
```

```
</div>
```

```
</section>
```

```
<?php require ('footer.php'); ?>
```

Контент страницы «Рецепты» - content-home2.php

```
<article>
```

```
<div class="heading">
```

```
<h2><a href="/r1.php">Творожная запеканка</a></h2>
```

```
<p class="info">>>> Опубликовано - 01/01/2020 - 0 Comments</p>
```

```
</div>
```

```
<div class="content">
```

```

```

```
<p> <p><B>ИНГРЕДИЕНТЫ<br></B></p>
```

```
Куриное яйцо.....4 шт<br>
```

```
Рассыпчатый творог..750 гр<br>
```

```
Манная крупа.....4 ст.л<br>
```

```
Молоко.....1 стакан<br>
```

```
Сметана.....100 гр<br>
```

```
Сахар.....4 ст.л<br>
```

```
Ванилин.....2гр<br>
```

```
Калорийность....187.4<br>
```

```
<p><B>Приготовление<br></B></p>
```

```
1. Яйца перемешать с сахаром (можно использовать миксер).<br>
```

```
2. Добавить в смесь молоко, ванилин и тщательно размешать.<br>
```

3. Выложить творог в глубокую миску.

4. Добавить к творогу манную крупу и смесь яиц, молока, сахара и ванилина.

5. По желанию можно добавить фрукты или ягоды.

6. Все тщательно перемешать до исчезновения комочков творога.

7. Для получения однородной массы можно использовать блендер, тогда запеканка получится еще более пышной и нежной.

8. Смазать сливочным маслом разогретую форму и вылить в неё твороженную массу.

9. Поставить форму в духовку, разогретую до 200 градусов на 40-45 минут.

10. После готовности вынуть форму из духовки и дать запеканке остыть.

11. Подавать со сметаной, сгущенкой или джемом. Приятного аппетита.
</p>

<!-- like -->

<button class="btn">Нравится!</button>

<div class="likes">22</div>

<script>

const btn = document.querySelector('.btn');

let like = true,

likeCount = document.querySelector('.likes').innerHTML;

btn.addEventListener('click', () => {

likeCount = like ? ++likeCount : --likeCount;

like = !like;

document.querySelector('.likes').innerHTML = likeCount;

});

</script>

</article>

<article>

<div class="heading">

<h2>Луковый суп</h2>

<p class="info">>>> Опубликовано - 02/01/2020 - 0 Comments</p>

</div>

<div class="content">

<p> <p>ИНГРЕДИЕНТЫ
</p>

Репчатый лук..... 4 головки

Пшеничная мука.....1 столовая ложка

Красное сухое вино.....1/2 стакана

Куриный бульон.....1,5 л

Сыр грюйер.....150 г

Калорийность....76.8
</p>

<p>Приготовление
</p>

1. В большой толстостенной кастрюле растопить сливочное масло и отправить туда лук, нарезанный полукольцами. Обжаривать, помешивая, пока он не станет прозрачным.

2. Убавить огонь и продолжать жарить лук, время от времени помешивая и следя, чтобы он не подгорал. Слегка обжарить на сухой сковороде муку. Как только мука начнет менять цвет и пахнуть жареным орехом, тут же сыпать ее в кастрюлю с луком и тщательно вмешать.

3. Теперь остается вылить в кастрюлю сначала вино, смешанное с сахаром, дать луку минуту покипеть вместе с вином, а потом разбавить бульоном. Довести до кипения и варить на очень медленном огне 15-20 минут. Перемешать и добавить оставшуюся воду. Довести до кипения и варить на медленном огне еще 15 минут.

4. Разложить куски багета на противне и посыпать их тертым сыром. Поставить в духовку на три-четыре минуты, чтобы сыр расплавился и слегка подрумянился. Подавать суп в глубоких тарелках с плавающими и постепенно набухающими гренками.

</p>

<button class="btn">Нравится!</button>

<div class="likes">22</div>

<script>

const btn = document.querySelector('.btn');

let like = true,

likeCount = document.querySelector('.likes').innerHTML;

btn.addEventListener('click', () => {

likeCount = like ? ++likeCount : --likeCount;

like = !like;

document.querySelector('.likes').innerHTML = likeCount;

});

</script>

</article>

<article>

<div class="heading">

<h2>Винегрет</h2>

<p class="info">>>> Опубликовано - 02/01/2020 - 0 Comments</p>

</div>

<div class="content">

<p><p>ИНГРЕДИЕНТЫ
</p>

Свекла....1 шт.

Картошка..2 шт.

Зеленый горошек..1 банка.

Морковь.....3 шт.

Капуста квашеная...100 гр.

Соленые огурцы...2 шт.

Растительное масло...3 ст.л.

Калорийность....130.1
</p>

<p>Приготовление
</p>

1. Отварить овощи, остудить и нарезать кубиками.

2.Соединить все овощи, добавить растительное масло и хорошо перемешать

3. Приятного аппетита!

</p>

</div>

<button class="btn">Нравится!</button>

<div class="likes">22</div>

<script>

const btn = document.querySelector('.btn');

let like = true,

likeCount = document.querySelector('.likes').innerHTML;

btn.addEventListener('click', () => {

likeCount = like ? ++likeCount : --likeCount;

```

like = !like;
document.querySelector('.likes').innerHTML = likeCount;
});
</script>
</article>

```

Страница с комментариями – comm.php

```

<?php require ('header.php'); ?>
<section id="content">
    <div class="zerogrid">
        <div class="row block">
            <div class="main-content col11">

                <style>
                textarea {
                    border: 1px solid #888;
                }
                input {
                    border: 1px solid #888;
                    width: 80px;
                    padding: 8px;
                    margin-bottom: 10px;
                }
                </style>

<?php
$page_id = 150; // Уникальный идентификатор страницы
$mysqli = new mysqli("localhost", "mysql", "mysql", "resepti"); // Подключается к базе данных
$result_set = $mysqli->query("SELECT * FROM `comments` WHERE `page_id`"); // Вытаскиваем все
комментарии для данной страницы
while ($row = $result_set->fetch_assoc()) {
    echo "<div><p><b>".$row['name']."</b></p></div>";
    print_r($row['text_comment']); // Вывод комментариев
    echo "<br />";
}
?>

        <form action="comment.php" method="post">
            <label>Имя</label>
            <p><input type="text" name="name"/></p>
            <p>Ваш отзыв:</p>
            <p><textarea
name="text_comment"></textarea></p>
                placeholder="Комментарий"
                rows="10"
                cols="45"
                <input type="hidden" name="page_id" value="150" />
                <p><input type="submit" value="Отправить"></p>
            </form>
        </div>
        <?php require ('sidebar.php'); ?>
    </div>
</div>

```

</section>

<?php require ('footer.php'); ?>

Скрипт, принимающий данные из формы – comment.php

<?php

/* Принимаем данные из формы */

\$name = \$_POST["name"];

\$page_id = \$_POST["page_id"];

\$text_comment = \$_POST["text_comment"];

\$name = htmlspecialchars(\$name);// Преобразуем спецсимволы в HTML-сущности

\$text_comment = htmlspecialchars(\$text_comment);// Преобразуем спецсимволы в HTML-сущности

\$mysqli = new mysqli("localhost", "mysql", "mysql", "resepti");// Подключается к базе данных

\$mysqli->query("INSERT INTO `comments` (`name`, `page_id`, `text_comment`) VALUES ('\$name', '\$page_id', '\$text_comment')");// Добавляем комментарий в таблицу

header("Location: ".\$_SERVER["HTTP_REFERER"]);// Делаем реридект обратно

?>

Страница регистрации – login.php

<?php

session_start();

?>

<?php \$url = \$_SERVER["REQUEST_URI"];?>

<!DOCTYPE HTML>

<html lang="ru">

<head>

<meta charset="utf-8">

<title>Регистрация</title>

</head>

<style>

body {

background: url(../images/sugar.jpg) no-repeat center center fixed;

background-size: cover;

margin: 0;

}

#window {

width: 350px;

height: 500px;

margin: 40px auto;

background: rgba(255,255,255,0.56);

border: 1px solid #ffff;

border-radius: 15px;

}

.form {

width: 275px;

margin: 0px auto 20px auto;

text-align: center;

}

.input {

width: 260px;

padding: 8px;

```

margin-bottom: 10px;
}
</style>
<body>
<div id="window">
<div class="form">
<h2>Регистрация</h2>
<form action="registration.php" method="post">
<input type="text" placeholder="Имя" name="user_name" class="input"required>
<input type="text" placeholder="Фамилия" name="surname" class="input"required>
<input type="text" placeholder="Логин" name="login" class="input"required>
<input type="text" placeholder="Город" name="town" class="input"required>
<input type="email" placeholder="Email" name="user_email" class="input"required>
<input type="password" placeholder="Пароль" name="parol" class="input"required>
<input type="submit" value="Регистрация" name="sab" class="input">
</form>
</div>
</div>
</body>
</html>

```

Скрипт, принимающий данные из формы – registration.php

```

<?php

if(!isset($_POST['user_name'])||
!isset($_POST['surname'])||
!isset($_POST['login'])||
!isset($_POST['town'])||
!isset($_POST['user_email'])||
!isset($_POST['parol'])){
die("Не все данные введены, проверьте все еще раз!");
}

$host = 'localhost';
$user = 'mysql';
$password = 'mysql';
$database = 'resepti';

$user_name = filter_var(trim($_POST['user_name']));
$surname = filter_var(trim($_POST['surname']));
$login = filter_var(trim($_POST['login']));
$town = filter_var(trim($_POST['town']));
$user_email = filter_var(trim($_POST['user_email']));
$parol = filter_var(trim($_POST['parol']));
$connect = mysqli_connect($host, $user, $password, $database) or die ("Невозможно подключение к MySQL");
$query="INSERT INTO `user` (`user_name`, `surname`, `login`, `town`, `user_email`, `parol`)
VALUES('".$user_name."', '".$surname."', '".$login."', '".$town."', '".$user_email."', '".$parol"');

```



```
$result = mysqli_query($connect, $query);  
if ($result) echo "Регистрация прошла успешно."  
mysqli_close($connect);  
?>
```

Страница «Поделиться рецептом» - published.php

```
<?php require ('header.php');
```

```
$host = 'localhost';  
$user = 'mysql';  
$password = 'mysql';  
$database = "resepti";  
$query_kuhnia = "SELECT * FROM kuhnia;";  
$query_kategor = "SELECT * FROM kategor;";  
$query_sposob = "SELECT * FROM sposob;";  
$link = mysqli_connect($host, $user, $password, $database) or die ("Невозможно подключение к  
MySQL");  
$result_kuhnia = mysqli_query($link, $query_kuhnia);  
$result_kategor = mysqli_query($link, $query_kategor);  
$result_sposob = mysqli_query($link, $query_sposob);  
mysqli_close($link);  
?>
```

```
<section id="content">  
    <div class="zerogrid">  
        <div class="row block">  
            <div class="main-content col11">  
                <div id="content">  
                    <style>  
                        #window {  
                            width: 350px;  
                            height: 600px;  
                            margin: 40px auto;  
                            background: rgba(255,255,255,0.56);  
                            border: 1px solid #ffff;  
                            border-radius: 15px;  
                        }  
                        form {  
                            width: 275px;  
                            margin: 0px auto 20px auto;  
                            text-align: center;  
                        }  
                        textarea {  
                            border: 1px solid #888;  
                        }  
                        input {  
                            border: 1px solid #888;  
                            width: 260px;  
                            padding: 5px;  
                            margin-bottom: 10px;
```

```

    }
</style>
<div id="window">
  <div class="form">
    <form action="save.php" method="post">
      <p><h2>Создание нового рецепта</h2> </p>
      <p>
        <label for="name">Название</label>
        <input class="text" type="text" name="name" />
      </p>
      <p><label for="kuhnia">Кухня</label></p>
      <select class="text" type="text" name="kuhnia">
        <option>- выберите кухню -</option>
        <?php
          if (mysqli_num_rows($result_kuhnia) > 0) {
            while ($row = mysqli_fetch_assoc($result_kuhnia)) {
              echo "<option value='".$row['kuhnia_id']."'>".$row['name']."'</option>";
            }
          }
        ?>
      </select>
      <p><label for="kategor">Категория</label></p>
      <select class="text" type="text" name="kategor">
        <option>- выберите категорию -</option>
        <?php
          if (mysqli_num_rows($result_kategor) > 0) {
            while ($row = mysqli_fetch_assoc($result_kategor)) {
              echo "<option value='".$row['kategor_id']."'>".$row['name']."'</option>";
            }
          }
        ?>
      </select>
      <p><label for="sposob">Способ</label></p>
      <select class="text" type="text" name="sposob">
        <option>- выберите способ -</option>
        <?php
          if (mysqli_num_rows($result_sposob) > 0) {
            while ($row = mysqli_fetch_assoc($result_sposob)) {
              echo "<option value='".$row['sposob_id']."'>".$row['name']."'</option>";
            }
          }
        ?>
      </select>
      <p>
        <label for="description">Приготовление</label>
      </p>
      <p> <textarea cols="60" rows="20" name="description"></textarea>
    </p>
  </div>
</div>

```

```

<p> <button type="submit" value="save">Опубликовать</button></p>
    </form>
  </div>
</div>
</div>
</div>

```

```

    <?php require ('sidebar.php'); ?>

```

```

</section>

```

```

<?php require ('footer.php'); ?>

```

Скрипт, принимающий данные из формы – save.php

```

<?php
    require('header.php');
    $host = 'localhost';
    $user = 'mysql';
    $password = 'mysql';
    $database = "resepti";
    // Переменные с формы
    if(!isset($_POST['name'])||
        !isset($_POST['kuhnia'])||
        !isset($_POST['kategor'])||
        !isset($_POST['sposob'])||
        !isset($_POST['description'])){
        die("Не все данные введены, проверьте все еще раз!");
    }
    $name = trim($_POST['name']);
    $kuhnia = trim($_POST['kuhnia']);
    $kategor = trim($_POST['kategor']);
    $sposob = trim($_POST['sposob']);
    $description = trim($_POST['description']);
    $link = mysqli_connect($host, $user, $password, $database) or die ("Невозможно подключение к
MySQL");
    $result = mysqli_query($link, $query);
    $query="INSERT INTO res(name, kuhnia_id, kategor_id, sposob_id, description) VALUES('".$name."',
    '".$kuhnia."', '".$kategor."', '".$sposob."', '".$description."')";
    $result = mysqli_query($link, $query);
    if ($result) echo "Рецепт успешно опубликован.";
    mysqli_close($link);
?>

```

Файл стилей style.css

```

a,abbr,acronym,address,applet,article,aside,audio,b,blockquote,big,body,center,canvas,caption,cite,code,com
mand,datalist,dd,del,details,dfn,dl,div,dt,em,embed,fieldset,figcaption,figure,font,footer,form,h1,h2,h3,h4,h5
,h6,header,hgroup,html,i,iframe,img,ins,kbd,keygen,label,legend,li,meter,nav,object,ol,output,p,pre,progress,
q,s,samp,section,small,span,source,strike,strong,sub,sup,table,tbody,tfoot,thead,th,tr,tdvideo,tt,u,ul,var{ back
ground:transparent;border:0 none;font-size:100%;margin:0;padding:0;border:0;outline:0;vertical-
align:top;}ol, ul { list-style:none;}blockquote, q { quotes:none;}table, table td
{ padding:0;border:none;border-collapse:collapse;}img { vertical-align:top;}embed { vertical-align:top;}
article, aside, audio, canvas, command, datalist, details, embed, figcaption, figure, footer, header, hgroup,
keygen, meter, nav, output, progress, section, source, video { display:block;}

```

```

mark, rp, rt, ruby, summary, time {display:inline;}
input, textarea {border:0; padding:0; margin:0; outline: 0;}
iframe {border:0; margin:0; padding:0;}
input, textarea, select {margin:0; padding:0px;}

/* ----- */
/* -----Style----- */
/* ----- */
html, body {width:100%; padding:0; margin:0;}
body {background: url(../images/fon1.jpg) no-repeat center center fixed;background-size:100%; color:
#292929;font: 14px/25px Arial, Helvetica, sans-serif;}

/*-----*/
a{color: #006699;text-decoration: none;}
a:hover {text-decoration: none;}

a.button, input.button {      cursor: pointer;      color: #808080;      line-height: 12px;      font-family:
Arial, Helvetica, sans-serif; font-size: 12px;      font-weight: bold;      background: #E4E4E4;
      border:1px solid #DFDFDF;}
a.button {      display: inline-block; text-decoration: none;padding: 6px 12px 6px 12px;}
input.button { margin: 0;      border: 0;      height: 24px; padding: 0px 12px 0px 12px;}
a.button:hover, input.button:hover {background: #006699;color:#ffffff;}

/*-----header */
header {height: 115px;width:100%; background:url('../images/bg-header.png') repeat-x;margin-
bottom:15px;}

header #logo {margin-top:10px;}

header #search-box {position: relative;width: 300px;margin-top:30px;}
header #search-form {height: 25px;border: 1px solid #999;background-color: #fff;overflow: hidden;}
header #search-text {font-size: 14px;color: #ddd;background: transparent;}
header #search-box input[type="text"] {width: 80%;padding: 5px 0 5px 1em;color: #333;outline: none;}
header #search-button {position: absolute;top: 0;right: 0;height: 27px;width: 80px;font-size: 14px;color:
#fff;text-align: center;line-height: 26px;background-color: #999;cursor: pointer;border:0px;}

/*-----menu */
nav {width:960px;height: 47px;margin: 0 auto 15px auto;background:#006699 url(../images/lines2.png);}
nav ul {list-style: none;margin: 0;padding: 0;}
nav ul li {position: relative;float: left;padding: 6px 5px 0px 5px}
nav ul li:hover {background: #006699 url('../images/lines.png');}
nav ul li a {font-size: 14px; line-height:14px;color: #ffffff;display: block;padding: 7px 10px 6px
10px;margin-bottom: 6px;z-index: 6;position: relative;font-weight:bold;}
nav ul li:hover a{ }

/*-----content */
#content { }

```

```

.block {padding: 10px 0px;background:#ffffff;}

.main-content{ }
.main-content article{ margin: 10px 0; padding: 1px 2px; overflow:hidden; position:relative}
.main-content .heading h2{ font-size:20px; line-height:25px; letter-spacing:-1px; font-weight:bold;}
.main-content .heading .info{ font-size:12px; font-style:italic;}
.main-content .content{ }
.main-content .content img{ float:left; margin:3px 20px 0px 0; border:none;}
.main-content .footer{ }
.main-content .footer .more{ float:right;}
.main-content .footer .more a{ text-decoration:none;}

.sidebar section{ margin-bottom: 10px;}
.sidebar .heading{ color:#ffffff; background:#006699 no-repeat left; font-size:20px; font-weight:normal;
padding: 10px 10px 10px 38px;}
.sidebar .content{padding:10px;}
.sidebar .content img{ float:left; margin:5px; border:none;}

.featured{ width:100%; margin-bottom: 20px;}

ul { margin:0; padding:0;list-style-image:none;}
ul.list li {padding:0px 0 5px 12px; line-height:25px; background:url(../images/marker-1.gif) 0 11px no-repeat;}
ul.list li a { font-size:14px; color:#292929; line-height:25px;}
ul.list li a:hover { color:#006699; text-decoration:none;}

/*-----footer */
footer {padding: 20px 10px; background:#3F3F3F;margin-top:20px;}
footer .heading{ color:#ffffff; padding-bottom:5px;font-size:18px; font-weight:normal; text-transform:uppercase;}
footer .content{color:#ffffff;}
footer .content img{ float:left; margin:5px; border:none;}
footer ul { margin:0; padding:0;list-style-image:none;}
footer ul li { list-style:circle;margin-left:10px;}
footer ul li a { font-size:14px; color:#ffffff;}
footer ul li a:hover { color:#006699; text-decoration:none; cursor:pointer;}

#copyright{ display:inline-block; width:100%; text-align: center; background:url(../images/bg-footer.png);padding:8px 0px;color:#ffffff;}
Файл стилей slides.css
.rslides {
  position: relative;
  list-style: none;
  overflow: hidden;
  width: 100%;
  padding: 0;
  margin: 0;
}

```

```
.rslides li {  
  position: absolute;  
  display: none;  
  width: 100%;  
  left: 0;  
  top: 0;  
}
```

```
.rslides li:first-child {  
  position: relative;  
  display: block;  
  float: left;  
}
```

```
.rslides img {  
  display: block;  
  height: auto;  
  float: left;  
  width: 100%;  
  border: 0;  
}
```

```
.rslides {  
  margin: 0 auto;  
}
```

```
.rslides_container {  
  
  position: relative;  
}
```

```
.centered-btns_nav {  
  position: absolute;  
  -webkit-tap-highlight-color: rgba(0,0,0,0);  
  top: 50%;  
  left: 0;  
  opacity: 0.7;  
  text-indent: -9999px;  
  overflow: hidden;  
  text-decoration: none;  
  height: 41px;  
  width: 38px;  
  background: transparent url("../images/themes.gif") no-repeat left top;  
  margin-top: -45px;  
}
```

```
.centered-btns_nav:active {
```

```
opacity: 1.0;
}
```

```
.centered-btns_nav.next {
  left: auto;
  background-position: right top;
  right: 0;
}
```

```
.transparent-btns_nav {
  position: absolute;
  -webkit-tap-highlight-color: rgba(0,0,0,0);
  top: 0;
  left: 0;
  display: block;
  background: #fff; /* Fix for IE6-9 */
  opacity: 0;
  filter: alpha(opacity=1);
  width: 48%;
  text-indent: -9999px;
  overflow: hidden;
  height: 91%;
}
```

```
.transparent-btns_nav.next {
  left: auto;
  right: 0;
}
```

```
.large-btns_nav {
  position: absolute;
  -webkit-tap-highlight-color: rgba(0,0,0,0);
  opacity: 0.6;
  text-indent: -9999px;
  overflow: hidden;
  top: 0;
  bottom: 0;
  left: 0;
  background: #000 url("../images/themes.gif") no-repeat left 50%;
  width: 38px;
}
```

```
.large-btns_nav:active {
  opacity: 1.0;
}
```

```
.large-btns_nav.next {
  left: auto;
```

```
background-position: right 50%;
right: 0;
}
```

```
.centered-btns_nav:focus,
.transparent-btns_nav:focus,
.large-btns_nav:focus {
  outline: none;
}
```

```
.centered-btns_tabs,
.transparent-btns_tabs,
.large-btns_tabs {
  margin-top: 10px;
  text-align: center;
}
```

```
.centered-btns_tabs li,
.transparent-btns_tabs li,
.large-btns_tabs li {
  display: inline;
  float: none;
  _float: left;
  *float: left;
  margin-right: 5px;
}
```

```
.centered-btns_tabs a,
.transparent-btns_tabs a,
.large-btns_tabs a {
  text-indent: -9999px;
  overflow: hidden;
  -webkit-border-radius: 15px;
  -moz-border-radius: 15px;
  border-radius: 15px;
  background: #ccc;
  background: rgba(0,0,0, .2);
  display: inline-block;
  _display: block;
  *display: block;
  -webkit-box-shadow: inset 0 0 2px 0 rgba(0,0,0,.3);
  -moz-box-shadow: inset 0 0 2px 0 rgba(0,0,0,.3);
  box-shadow: inset 0 0 2px 0 rgba(0,0,0,.3);
  width: 9px;
  height: 9px;
}
```

```
.centered-btns_here a,
```



```
.transparent-btns_here a,  
.large-btns_here a {  
  background: #222;  
  background: rgba(0,0,0, .8);  
}
```