**Введение**

В современном мире цифровизация играет ключевую роль в развитии любого бизнеса, включая сферу услуг. Парикмахерские не являются исключением, и наличие удобного и функционального веб-сайта становится важным инструментом для привлечения новых клиентов. Цель данного проекта заключается в создании веб-сайта для парикмахерской, который позволит предоставлять клиентам актуальную информацию о предоставляемых услугах, а также запись на прием онлайн.

Разработка сайта будет осуществлена с использованием современных технологий веб-разработки, таких как HTML, CSS, JavaScript, а также фреймворков, обеспечивающих адаптивность и удобство использования.

Далее приведено краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ предметной области и формулировка требований к программе». В нем можно ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой создаётся данный проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе «Проектирование» будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки.

«Реализация» – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации игрового приложения. В этом разделе будут описаны функции пользователя и их структура.

Четвёртый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данного проекта, т.е. будет протестирована каждая операция, которая выполняется приложением.

В разделе «Применение» будет описано назначение, область применения, среда функционирования игрового приложения. «Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки. В разделе «Список использованных источников» будет приведён список используемых при разработке источников.

В приложении к пояснительной записке будет приведен листинг программы, диаграмма вариантов использования и модель данных.

**1 Анализ задачи**

* 1. **Описание предметной области**

Сайт для парикмахерской — это веб-приложение, разработанное для предоставления информации о парикмахерской, её услугах, мастерах и акциях, а также для удобной онлайн-записи клиентов. Сайт служит важным инструментом для привлечения новых клиентов, улучшения взаимодействия с существующими и повышения уровня сервиса.

Основные элементы сайта включают:

1 Информация о парикмахерской: описание заведения, его преимущества, контактные данные.

2 Онлайн-запись: возможность для клиентов выбирать услуги, мастера и удобное время для посещения.

3 График работы: отображение расписания работы парикмахерской и доступности мастеров.

4 Список услуг: перечень предоставляемых услуг с описанием и ценами.

5 Новости и акции: раздел для информирования клиентов о текущих акциях, скидках и событиях.

6 Категории услуг: структурированное представление услуг по категориям (например, мужские стрижки, женские стрижки и т.д.).

Наименование задачи: разработка программного обеспечения по учёту клиентов в парикмахерской

Цель разработки: Создание удобного и функционального сайта для парикмахерской, который позволит клиентам легко записываться на услуги, узнавать актуальную информацию.

Назначение: Сайт предназначен для широкой аудитории, включая постоянных и новых клиентов парикмахерской, которые хотят быстро и удобно записаться на услуги, ознакомиться с перечнем услуг и получить актуальную информацию о работе заведения.

Основные элементы сайта:

1 Онлайн-запись: выбор услуги, мастера и удобного времени

2 График работы: отображение расписания работы парикмахерской

3 Список услуг: описание услуги с указанием цены

4 Новости и акции:

5 Категории услуг: представление услуг по категориям для удобства навигации.

6 Информация о мастерах: фото и описание мастера

Аналоги: Zoon – парикмахерская г.Гродно, платформы для онлайн-записи: сервисы вроде "Яндекс.Записи" или "Booksy". Они предлагают удобный интерфейс для записи, интеграцию с календарем и уведомления.

Исследование включает следующие элементы функциональности:

1 Онлайн-запись: клиент выбирает услугу, мастера и удобное время, система автоматически проверяет доступность выбранного времени.

2 График работы: отображение расписания работы парикмахерской.

3 Список услуг: навигация по категориям услуг.

4 Новости и акции

Сайт для парикмахерской станет важным инструментом для привлечения клиентов и улучшения качества обслуживания. Он должен быть удобным, информативным и функциональным, чтобы удовлетворить потребности как новых, так и постоянных клиентов.

* 1. **Диаграмма вариантов использования**

Диаграмма вариантов использования, разработанная для данного проекта, представлена в Приложении А.

Диаграмма вариантов использования UML описывает функциональность и требования системы с помощью актеров (пользователей или других систем) и вариантов использования (рисунок 1). Варианты использования представляют собой функциональные возможности системы и то, как актеры будут взаимодействовать с системой. Эти диаграммы моделируют сервисы, задачи и функции, которые система должна выполнять, и отражают динамическое поведение системы в процессе её работы.



Рисунок 1 – Актёр

Компоненты ДВИ:

1 Актеры представляют собой пользователей или другие системы, которые взаимодействуют с целевой системой.

2 Варианты использования описывают конкретные функции или задачи, которые система должна выполнять для удовлетворения потребностей актеров.

3 Связи между актерами и вариантами использования показывают, как они взаимодействуют друг с другом.

**1.3 Модель данных**

Цель моделирования данных состоит в определении концептуальной схемы совокупности данных в форме одной или нескольких локальных моделей, которые относительно легко могут быть отображены в любую форму представления данных.

Первый шаг моделирования – извлечение информации и выделение сущностей. Сущность – это реальный объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению в проекте.

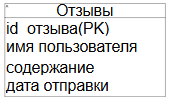


Рисунок – Сущность

Применимо к разрабатываемой задаче можно выделить следующие

сущности:

Таблица 1 – Описание сущностей

|  |  |
| --- | --- |
| Сущность | Описание |
| Админ | Доступ к админ панели, контенту сайта |
| Пользователь | Информация о пользователе |
| Запись | Информация о записи |
| Парикмахер | Информация о парикмахере |
| Услуги | Информация об услуге |

Следующим шагом в моделировании данных является идентификация

связей. Связь – это ассоциация между двумя сущностями, значимая для рассматриваемой предметной области. Степень связи и обязательность графически изображаются так, как показано на рисунке

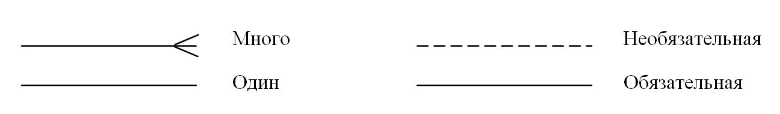


Рисунок – Связи

Ознакомиться со связями в нашей модели данных можно в таблице 2.

Таблица 2 – Сущности и их связи

|  |  |  |
| --- | --- | --- |
| Сущность | Связь | Сущность |
| Пользователь | Один ко многим | Запись |
| Услуги | Многие ко многим | Запись |
| Парикмахер | Один ко многим | Запись |

Последним шагом моделирования является идентификация атрибутов.

Атрибут – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибуты сущностей могут быть двух типов: обязательный (\*) и необязательный (0).

Таблица 3 – Описание атрибутов сущностей

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Описание атрибута |
| Пользователь | id | Идентификатор |
| Пользователь | Логин | Для входа в систему |
| Пользователь | Пароль | Для входа в систему |
| Пользователь | Фамилия | Информация о пользователе |
| Пользователь | Имя | Информация о пользователе |
| Пользователь | Отчество | Информация о пользователе |
| Пользователь | Роль | Роль пользователя |
| Запись | id | Идентификатор |
| Запись | id\_пользователя | Информация о пользователе |
| Запись | Id\_парикмахера | Информация о парикмахере |
| Запись | Id\_услуги | Информация об услуге |
| Запись | Время | Назначенное время и дата |
| Запись | Номер телефона | Номер тел. пользователя |
| Запись | Оплачено | Статус записи на услугу |
| Услуги | Id | Идентификатор |
| Услуги | Название | Название услуги |
| Услуги | Описание | Описание услуги |
| Парикмахер | id | Идентификатор |
| Парикмахер | Фамилия | Информация о парикмахере |
| Парикмахер | Имя | Информация о парикмахере |
| Парикмахер | Отчество | Информация о парикмахере |
| Парикмахер | Фото | Фото парикмахера |
| Отзывы | id | Идентификатор |
| Отзывы | Имя пользователя | Имя отправителя |
| Отзывы | Содержание | Содержание отзыва |
| Отзывы | Оценка | Оценка от 1 до 5 |
| Отзывы | Дата отправки | Дата его отправки |

В итоге трех этапов моделирования получим модель данных, представленную в графической части проекта.

**1.4 Инструменты разработки**

Для разработки сайта выбран язык разметки гипертекста HTM, который является популярным инструментом для создания сайтов. Для работы с БД выбрана утилита PHP My Admin, которая реализует интерфейс для управления БД MySQL через браузер.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

- WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм,

- Microsoft Office Word – для написания документации к программному продукту,

- Visual Studio Code – среда разработки.

Разработка проекта будет происходить на компьютере со следующими параметрами:

- процессор Ryzen 5;

- объем ОЗУ 16 GB;

- объем места: 512 GB;

- ОС: Windows 10 Pro.

**1.5 Требования к приложению**

Требование к интерфейсу:

- простота использования: пользователи должны легко находить необходимые функции;

- адаптивный дизайн: интерфейс должен корректно отображаться на различных устройствах

- эстетика и визуальная привлекательность: Использование современного дизайна, который соответствует текущим тенденциям и создает положительное первое впечатление.

Для реализации интерфейса будут использоваться чёрный цвет и акцентный светло-фиолетовый. Элементы не должны перекрывать и налаживаться друг на друга.

Средства защиты:

* контроль доступа: ограничение доступа к личным данным и функциям приложения в зависимости от ролей пользователей (например, администраторы, обычные пользователи);
* шифрование данных: Использование протоколов шифрования (например, HTTPS) для защиты передаваемой информации и хранения паролей в зашифрованном виде.

**2 Проектирование задачи**

**2.1 Организация данных**

Для организации данных используется утилита PHPMyAdmin, графический интерфейс которой отображается с помощью браузера. Для работы с данными используется SQL. PHPMyAdmin – это бесплатный кроссплатформенный инструмент. В таблице 4 представлено описание таблиц БД.

Таблица 4 – Описание таблиц БД

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Поле | Описание | Тип данных |
| Клиент | id | Идентификатор пользователя | INT (Primary Key) |
| Пароль | Пароль пользователя | VARCHAR(255) |
| Логин | Логин пользователя | VARCHAR(255) |
| Фамилия | Фамилия пользователя | TEXT |
| Имя | Имя пользователя | TEXT |
| Отчество | Отчество пользователя | TEXT |
| Роль | Определение роли пользователя | CHAR(255) |
| Запись | id | Идентификатор записи | INT (Primary Key) |
| Id\_пользователя | Идентификатор пользователя | INT (Foreign Key) |
| Id\_услуги | Идентификатор услуги | INT (Foreign Key) |
| Id\_парикмахера | Идентификатор парикмахера | INT (Foreign Key) |
| Дата/время | Дата и время, на которое назначена услуга | DATETIME |
| Номер телефона | Номер пользователя | VARCHAR(12) |
| Оплачено | Логическое значение, указывающее на статус записи | BOOLEAN |
| Парикмахер | id | Идентификатор парикмахера | INT (Primary Key) |
| Фамилия | Фамилия парикмахера | TEXT |
| Имя | Имя парикмахера | TEXT |
| Отчество | Отчество парикмахера | TEXT |
| Фото | Ссылка на фото пакримахера | VARCHAR(255) |

Таблица 4 – Продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Поле | Описание | Тип данных |
| Услуги | id | Идентификатор услуги | INT (Primary Key) |
| Название | Название услуги | TEXT |
| Описание | Описание услуги | TEXT |
| Id\_категории | Идентификатор категории | INT (Foreign Key) |
| Цена | Цена услуги | DECIMAL(10,2) |
| Категория | id | Идентификатор категории | INT (Primary Key) |
| Название | Название категории | TEXT |
| Отзывы | id | Идентификатор отзыва | INT (Primary Key) |
| Имя пользователя | Имя пользователя | VARCHAR(255) |
| Содержание | Содержание отзыва | TEXT |
| Оценка | Оценка от 1 до 5 | INT |
| Дата отправки | Дата отправки отзыва | TIMESTAMP |

**2.2 Процессы**

Описание действий диаграммы вариантов использования включает в себя визуализацию взаимодействий между пользователями и системой. Пользователи могут выполнять основные действия. Администраторы имеют доступ к управлению контентом, управлению аккаунтами, записями на услугу и мастерами.

Варианты использования:

- пользователь создаёт аккаунт/входит в аккаунт;

- пользователь просматривает новости, услуги, мастеров;

- пользователь записывается онлайн, оставляет свои данные;

- пользователь оставляет отзыв;

- пользователь выходит из аккаунта;

- админ работает с данными о мастерах, услугах, онлайн записях (удаление, редактирование, экспорт в exel).

**2.3 Средства защиты и разграничения доступа к данным**

Средства защиты и разграничения доступа к данным включают в себя аутентификацию пользователей и шифрование данных. Использование паролей обеспечивает безопасность учетных записей. Регулирование прав доступа позволяет пользователям выполнять только разрешенные действия, защищая конфиденциальность и целостность информации в системе.

- гость: просматривает страницы сайта, не может пользоваться основными функциями;

- основной пользователь;

- админ: управление контентом и аккаунтами.

**2.4 Тесты на использование**

Проектирование задачи – это очень важный и ответственный этап в разработке любого приложения.

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения этапа написания программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Таблица 5 – Тест-кейсы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Заглавие и шаги выполнения | Ожидаемый результат | Фактический результат | Статус |
| 1 |  | Регистрация 1 Открыть форму регистрации 2 Заполнить поля 3 Нажать на кнопку «Зарегистрироваться» | 1 Форма загрузилась  2 Ввод и валидация полей работают  3 Создаётся аккаунт, загружается страница indexsesion.php | 1 Форма загрузилась  2 Ввод и валидация полей работают  3 Создаётся аккаунт, загружается страница indexsesion.php | Выполнено |
| 2 |  | Вход  1 Открыть форму входа 2 Заполнить поля 3 Нажать на кнопку «Войти» | 1 Форма загрузилась  2 Ввод и валидация полей работают  3 Вход в систему, загружается страница indexsesion.php | 1 Форма загрузилась  2 Ввод и валидация полей работают  3 Создаётся аккаунт, загружается страница indexsesion.php | Выполнено |
| 3 |  | Онлайн запись  1 Нажать кнопку «Онлайн запись»  2 Заполнить поля  3 Нажать кнопку «Записаться» | 1 Загрузка формы регистрации  2 Ввод и валидация работают | 1 Успешный вход  2 Загрузка формы регистрации  3 Ввод и валидация работают | Выполнено |
| 4 |  | Отызвы  1 Загрузить форму  2 Заполнить поля  3 Нажать кнопку «Отправить» | 1 Загрузка формы  2 Ввод и валидация полей работают  3 Отзыв сохраняется в БД | 1 Загрузка формы  2 Ввод и валидация полей работают  3 Отзыв сохраняется в БД | Выполнено |

Таблица 5 – Продолжение

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Заглавие и шаги выполнения | Ожидаемый результат | Фактический результат | Статус |
| 5 |  | Добавление мастеров  1 Загрузить админ-панель  2 Заполнить поля в таблице «Мастера»  3 Нажать кнопку «Добавить» | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новый мастер отображается в таблице и сохраняется в БД | 1 Форма загрузилась  2 Ввод и валидация полей работают  3 Создаётся аккаунт, загружается страница indexsesion.php | Выполнено |
| 6 |  | Добавить категорию  1 Загрузить админ-панель  2 Заполнить поля в таблице «Категория»  3 Нажать кнопку «Добавить» | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новая категория отображается в таблице и сохраняется в БД | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новая категория отображается в таблице и сохраняется в БД | Выполнено |
| 7 |  | Добавить услугу  1 Загрузить админ-панель  2 Заполнить поля в таблице «Услуги»  3 Нажать кнопку «Добавить» | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новая услуга отображается в таблице и сохраняется в БД | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новая услуга отображается в таблице и сохраняется в БД | Выполнено |
| 8 |  | Добавить записи на услугу  1 Загрузить админ-панель  2 Заполнить поля в таблице «Записи на услугу»  3 Нажать кнопку «Добавить» | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новая запись отображается в таблице и сохраняется в БД | 1 Админ-панель загрузилась  2 Ввод и валидация полей работают  3 Новая запись отображается в таблице и сохраняется в БД | Выполнено |

Таблица 5 – Продолжение

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Заглавие и шаги выполнения | Ожидаемый результат | Фактический результат | Статус |
| 9 |  | Изменение статуса  1 Загрузить админ-панель  2 Поставить галочку «Оплачено» в таблице «Записи на услугу» | 1 Админ-панель загрузилась  2 Изменение статуса | 1 Админ-панель загрузилась  2 Изменение статуса | Выполнено |
| 10 |  | Сумма услуг за период  1 Загрузить админ-панель  2 Выбрать период  3 Нажать кнопку | 1 Админ-панель загрузилась  2 Период отображается корректно  3 Отображение суммы оплаченных услуг | 1 Админ-панель загрузилась  2 Период отображается корректно  3 Отображение суммы оплаченных услуг | Выполнено |
| 11 |  | Редактирование таблиц  1 Загрузить админ- панель 2 Нажать кнопку «Редактировать» в таблицах «Записи на услугу» и «Услуги» | 1 Админ-панель загрузилась  2 Загрузились страницы редактирования | 1 Админ-панель загрузилась  2 Загрузились страницы редактирования | Выполнено |
| 12 |  | Удаление записей в таблице  1 Загрузить админ-панель  2 Нажать кнопку «Удалить» в таблицах | 1 Админ-панель загрузилась  2 Запись удалена из таблицы и из БД | 1 Админ-панель загрузилась  2 Запись удалена из таблицы и из БД | Выполнено |
| 13 |  | Экспорт Exel  1 Загрузить админ панель  2 Нажать кнопку «Экспорт Exel» в таблице «Записи на услуги»  3 Открыть файл exel | 1 Админ-панель загрузилась  2 Запись экспортировалась  3 Корректное отображение в файле | 1 Админ-панель загрузилась  2 Запись экспортировалась  3 Корректное отображение в файле | Выполнено |
| 14 |  | Аккаунт  1 Загрузить account.php  2 Проверить отображение записей | 1 Страница загрузилась  2 Записи отображаются | 1 Страница загрузилась  2 Записи отображаются | Выполнено |

Таблица 5 – Продолжение

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Заглавие и шаги выполнения | Ожидаемый результат | Фактический результат | Статус |
| 15 |  | Поиск  1 Загрузить страницу услуг  2 Ввести название услуги | 1 Страница услуг загрузилась  2 Отображение искаемой услуги | 1 Страница услуг загрузилась  2 Отображение искаемой услуги | Выполнено |
| 16 |  | Фильтр  1 Загрузить страницу услуг  2 Выбрать фильтр-категорию | 1 Страница услуг загрузилась  2 Отображение услуг по фильтру | 1 Страница услуг загрузилась  2 Отображение услуг по фильтру | Выполнено |
| 17 |  | Мастер  1 Загрузить страницу мастеров  2 Отображение фото мастера  3 Отображение ФИО мастера | 1 Страница мастеров загрузилась  2 Фото и ФИО мастеров отображается | 1 Страница мастеров загрузилась  2 Фото и ФИО мастеров отображается | Выполнено |
| 18 |  | Слайдер  1 Загрузка главной страницы  2 Проверить отображение всех слайдов | Слайдер переключается, все слайды отображаются | Слайдер переключается, все слайды отображаются | Выполнено |
| 19 |  | Регистрация на одну почту  1 Загрузка формы регистрации  2 Ввести почту зарегистрированного аккаунта | 1 Загрузилась форма регистрации  2 Отображение сообщения о использовании почты другим аккаунтом | 1 Загрузилась форма регистрации  2 Отображение сообщения о использовании почты другим аккаунтом | Выполнено |

**3 Реализация**

Элементы управления находятся в .php, а обрабатываются в .js.

Таблица 6 – Описание методов

|  |  |  |  |
| --- | --- | --- | --- |
| Функция | Модуль | Назначение | Метод |
| Авторизация | Autification.php | Авторизвация пользователя | POST |
| Регистрация | Registration.php | Регистрация нового аккаунта | POST |
| Отправка отзыва | Index.php indexsesion.php | Отправка отзыва | POST |
| Онлайн запись | indexsesion.php | Запись на услугу | openModal, closeModal,  submitBooking |
| Выход из аккаунта | Аccount.php | Выход из аккаунта | session\_destroy  exit |
| Поиск по названию | Index4.php | Поиск услуги по её названию | filterServices |
| Фильтр по категории | Index4.php | Фильтрация услуг по выбранной категории | filterServices |
| Удаление записей | Admin.php | Удаление записей из таблиц в админ-панели | DELETE |
| Добавление записей | Admin.php | Добавление записей в таблицу в админ-панели | POST |
| Редактирование записей | Admin.php | Редактирование записей в таблице в админ-панели | POST, EDIT |
| Просмотр итоговой суммы за период | Admin.php | Просмотр итоговой суммы выполненных услуг в админ-панели за указанный период |  |
| Обновление статуса записи | Admin.php | Изменение статуса «Оплачено» записи на услугу | updateNoteStatus |

**4 Тестирование**

Тестирование помогает разработчикам лучше понять процесс тестирования и обеспечить высокое качество продукта. Он также может служить руководством для новых членов команды по выполнению задач в области тестирования.

При тестировании программного продукта, в первую очередь нужно обратить внимание на правильную работу страниц. Отображается ли текст, работаю ли выпадающие списки.

Таким образом, основная часть ошибок и неполадок была выявлена и исправлена на этапе реализации проекта. После завершения этапа реализации, было проведено тщательное тестирование.

Таблица 7 – Расписание проведения тестов

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Дата | Описание | Длительность, ч |
| Алексейчук В.И. | 03.03.2024 | Выполнение тест-кейсов | 1 |
| Алексейчук В.И. | 03.03.2024 | Анализ выполнения тест-кейсов | 1,5 |
| Алексейчук В.И. | 05.06.2024 | Повторное выполнение тест кейсов | 1 |

Далее проводилась статистика по тестированию (таблица 3).

Таблица 8 – Статистика по ошибкам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Статус | Количество ошибок | Важность | | | |
| Критическая | Высокая | Средняя | Минимальная |
| Проверено | 2 | 1 | 1 | 0 | 0 |
| Исправлено | 2 | 1 | 1 | 0 | 0 |

Таким образом, после проведения тестирования, были выявлена ошибки при обновлении данных.

Опираясь на всё вышеперечисленное, можно сделать вывод о стабильности работы сайт. Тестирование прошло успешно, программный продукт можно характеризовать как реализованный и готовый к работе.

**5 Применение**

**5.1 Назначение**

Цель данного проекта заключается в разработке сайта, который предоставит пользователям возможность записаться онлайн на услуги парикмахерской. Сайт станет удобным способом быстро записаться к парикмахеру, без необходимости звонков.

На сайте можно узнать актуальную информацию о мастерах, акциях и новостях парикмахерской, о услугах и их цены.

**5.2 Среда функционирования**

Операционная система может быть любой, включая Windows, macOS, Linux, а также мобильные платформы, такие как Android и iOS. Главное, чтобы устройство имело стабильное подключение к интернету и установленный веб-браузер. Браузер, например, Chrome, Firefox или Safari, необходим для просмотра веб-страниц

**5.3 Сообщения системы**

Сообщения об ошибках:

Если пользователь допустил ошибку при заполнении формы (например, не указал имя или выбрал недоступное время), система выводит соответствующее сообщение с подсказкой для исправления.

Информационные сообщения:

В случае технических сбоев или временной недоступности сервиса система информирует пользователя о проблеме и предлагает повторить попытку позже.

**5.4 Обучение**

Сайт для парикмахерской разработан с учетом простоты и интуитивности интерфейса, что делает его доступным для пользователей без необходимости дополнительного обучения.

**Заключение**

В рамках проекта была поставлена задача разработки веб-сайта для парикмахерской, который позволит клиентам получать актуальную информацию о предоставляемых услугах, мастерах, акциях и осуществлять онлайн-запись на удобное время. Целью разработки являлось создание удобного, функционального инструмента для привлечения новых клиентов и улучшения взаимодействия с существующими.

В процессе работы был проведен анализ предметной области, определены основные функциональные требования к сайту, такие как онлайн-запись, структурированный список услуг, раздел новостей и акций, а также информация о мастерах. Были изучены аналогичные решения, такие как платформы для онлайн-записи и сайты парикмахерских, что позволило выделить ключевые элементы, необходимые для успешной реализации проекта.

Сайт для парикмахерской стал важным инструментом для автоматизации процессов записи клиентов и улучшения качества обслуживания. Он позволяет клиентам быстро и удобно записываться на услуги, получать актуальную информацию о работе заведения и текущих акциях. Разработка сайта способствует повышению конкурентоспособности парикмахерской и улучшению взаимодействия с клиентами.

В результате проделанной работы были достигнуты поставленные цели: создан функциональный и удобный сайт, который соответствует современным требованиям и потребностям клиентов. Проект демонстрирует эффективное использование современных технологий веб-разработки и может быть использован в качестве основы для дальнейшего развития и масштабирования.

**Список используемых источников**

1. Metanit [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://metanit.com/?ysclid=m7nuvjslqa511292434>

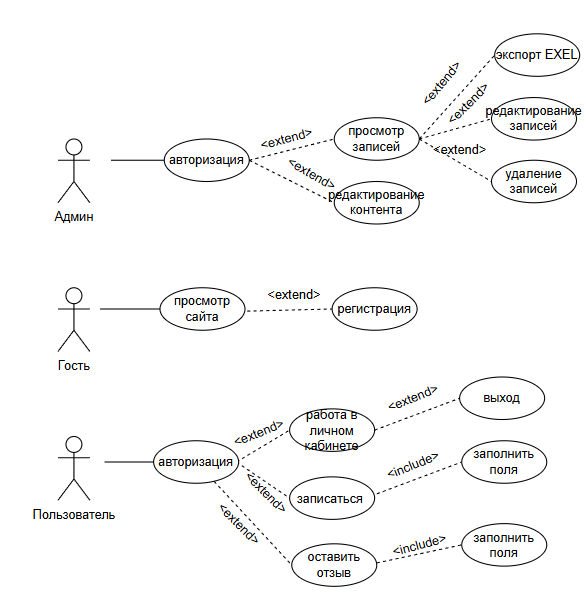


Рисунок А.1 – ДВИ

<?php

session\_start();

// Проверка роли пользователя

if (!isset($\_SESSION['role'])) { // Ошибка была здесь: лишняя скобка

header("Location: autification.php");

exit();

}

if ($\_SESSION['role'] !== 'admin') {

die("Доступ запрещён.");

}

// Подключение к базе данных

$host = 'localhost';

$dbname = 'stylist';

$user = 'root';

$password = '8308';

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $user, $password);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Ошибка подключения к базе данных: " . $e->getMessage());

}

// Обработка добавления мастера

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['add\_master'])) {

$fam = $\_POST['fam'];

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

$lastfam = $\_POST['lastfam'];

$foto = $\_POST['foto'];

$stmt = $pdo->prepare("INSERT INTO master (Fam, Name, Lastfam, Fhoto) VALUES (?, ?, ?, ?)");

$stmt->execute([$fam, $name, $lastfam, $foto]);

}

// Обработка удаления мастера

if (isset($\_GET['delete\_master'])) {

$id = $\_GET['delete\_master'];

$stmt = $pdo->prepare("DELETE FROM master WHERE Id = ?");

$stmt->execute([$id]);

}

// Обработка добавления категории

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['add\_category'])) {

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

$stmt = $pdo->prepare("INSERT INTO category (Name) VALUES (?)");

$stmt->execute([$name]);

}

// Обработка удаления категории

if (isset($\_GET['delete\_category'])) {

$id = $\_GET['delete\_category'];

$stmt = $pdo->prepare("DELETE FROM category WHERE Id = ?");

$stmt->execute([$id]);

}

// Обработка добавления услуги

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['add\_service'])) {

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

$description = $\_POST['description'];

$price = $\_POST['price'];

$category\_id = $\_POST['category\_id'];

$stmt = $pdo->prepare("INSERT INTO service (Name, description, Price, Id\_category) VALUES (?, ?, ?, ?)");

$stmt->execute([$name, $description, $price, $category\_id]);

}

// Обработка удаления услуги

if (isset($\_GET['delete\_service'])) {

$id = $\_GET['delete\_service'];

$stmt = $pdo->prepare("DELETE FROM service WHERE Id = ?");

$stmt->execute([$id]);

}

// Обработка удаления услуги

if (isset($\_GET['delete\_note'])) {

$id = $\_GET['delete\_note'];

$stmt = $pdo->prepare("DELETE FROM note WHERE Id = ?");

$stmt->execute([$id]);

}

// Обработка добавления записи на услугу

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['add\_note'])) {

$clientId = $\_POST['client\_id'];

$serviceId = $\_POST['service\_id'];

$masterId = $\_POST['master\_id'];

$time = $\_POST['time']; // Формат: YYYY-MM-DDTHH:MM

$phone = $\_POST['phone'];

// Преобразование формата времени

$time = DateTime::createFromFormat('Y-m-d\TH:i', $time);

if (!$time) {

die("Неверный формат времени.");

}

$time = $time->format('Y-m-d H:i:s'); // Преобразуем в формат MySQL

// Валидация данных

if (empty($clientId) || empty($serviceId) || empty($masterId) || empty($time) || empty($phone)) {

die("Все поля обязательны для заполнения.");

}

// Сохранение записи в базу данных

$stmt = $pdo->prepare("INSERT INTO note (Id\_client, Id\_service, Id\_master, Time, Phone\_client) VALUES (?, ?, ?, ?, ?)");

$stmt->execute([$clientId, $serviceId, $masterId, $time, $phone]);

}

// ПОЛУЧЕНИЕ МАСТЕРОВ КАТЕГОРИЙ УСЛУГ ЗАПИСЕЙ

$masters = $pdo->query("SELECT \* FROM master")->fetchAll(PDO::FETCH\_ASSOC);

$categories = $pdo->query("SELECT \* FROM category")->fetchAll(PDO::FETCH\_ASSOC);

$services = $pdo->query("SELECT \* FROM service")->fetchAll(PDO::FETCH\_ASSOC);

//ОБЪЕДИНЕНИЕ

$notes = $pdo->query("

SELECT note.\*, client.Fam AS client\_fam, client.Name AS client\_name

FROM note

JOIN client ON note.Id\_client = client.Id

")->fetchAll(PDO::FETCH\_ASSOC);

// Обработка экспорта записей в Excel

if (isset($\_GET['export\_notes'])) {

header('Content-Type: text/csv; charset=utf-8');

header('Content-Disposition: attachment; filename="notes.csv"');

// Открываем поток вывода

$output = fopen('php://output', 'w');

// BOM для корректного отображения кириллицы в Excel

fwrite($output, "\xEF\xBB\xBF");

fputcsv($output, array('ID', 'Клиент', 'Услуга', 'Мастер', 'Время', 'Телефон'), ';');

foreach ($notes as $note) {

$clientName = $note['client\_fam'] . ' ' . $note['client\_name'];

$serviceName = $services[array\_search($note['Id\_service'], array\_column($services, 'Id'))]['Name'];

$masterName = $masters[array\_search($note['Id\_master'], array\_column($masters, 'Id'))]['Fam'];

$row = array(

$note['Id'],

$clientName,

$serviceName,

$masterName,

$note['Time'],

$note['Phone\_client']

);

// Записываем строку в CSV

fputcsv($output, $row, ';');

}

// Закрываем поток вывода

fclose($output);

exit();

}

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Админ-панель</title>

<link rel="stylesheet" href="admin.css">

<script>

function updateNoteStatus(checkbox) {

const noteId = checkbox.getAttribute('data-note-id');

const statusType = checkbox.getAttribute('name');

const statusValue = checkbox.checked ? 1 : 0;

fetch('update\_note\_status.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

noteId: noteId,

statusType: statusType,

statusValue: statusValue

})

})

.then(response => response.json())

.then(data => {

if (!data.success) {

alert('Ошибка при обновлении статуса.');

checkbox.checked = !checkbox.checked; // Откат изменения

}

})

.catch(error => {

console.error('Ошибка:', error);

checkbox.checked = !checkbox.checked; // Откат изменения

});

}

</script>

</head>

<body>

<h1>Админ-панель</h1>

<!-- Управление мастерами -->

<div class="section">

<h2>Мастера</h2>

<form method="POST" novalidate>

<input type="text" name="fam" placeholder="Фамилия" required>

<input type="text" name="name" placeholder="Имя" required>

<input type="text" name="lastfam" placeholder="Отчество" required>

<input type="text" name="foto" placeholder="Ссылка на фото" required>

<button type="submit" name="add\_master">Добавить мастера</button>

</form>

<table>

<tr>

<th>Фамилия</th>

<th>Имя</th>

<th>Отчество</th>

<th>Действия</th>

</tr>

<?php foreach ($masters as $master): ?>

<tr>

<td><?= htmlspecialchars($master['Fam']) ?></td>

<td><?= htmlspecialchars($master['Name']) ?></td>

<td><?= htmlspecialchars($master['LastFam']) ?></td>

<td>

<a href="?delete\_master=<?= $master['Id'] ?>" class="delete">Удалить</a>

</td>

</tr>

<?php endforeach; ?>

</table>

</div>

<!-- Управление категориями -->

<div class="section">

<h2>Категории услуг</h2>

<form method="POST">

<input type="text" name="name" placeholder="Название категории" required>

<button type="submit" name="add\_category">Добавить категорию</button>

</form>

<table>

<tr>

<th>Название</th>

<th>Действия</th>

</tr>

<?php foreach ($categories as $category): ?>

<tr>

<td><?= htmlspecialchars($category['Name']) ?></td>

<td>

<a href="?delete\_category=<?= $category['Id'] ?>" class="delete">Удалить</a>

</td>

</tr>

<?php endforeach; ?>

</table>

</div>

<!-- Управление услугами -->

<div class="section">

<h2>Услуги</h2>

<form method="POST">

<input type="text" name="name" placeholder="Название услуги" required>

<input type="text" name="description" placeholder="Описание услуги" required>

<input type="number" name="price" step="0.01" placeholder="Цена" required>

<select name="category\_id" required>

<?php foreach ($categories as $category): ?>

<option value="<?= $category['Id'] ?>"><?= htmlspecialchars($category['Name']) ?></option>

<?php endforeach; ?>

</select>

<button type="submit" name="add\_service">Добавить услугу</button>

</form>

<table>

<tr>

<th>Название</th>

<th>Описание</th>

<th>Цена</th>

<th>Категория</th>

<th>Действия</th>

</tr>

<?php foreach ($services as $service): ?>

<tr>

<td><?= htmlspecialchars($service['Name']) ?></td>

<td><?= htmlspecialchars($service['description']) ?></td>

<td><?= htmlspecialchars($service['Price']) ?> руб.</td>

<td><?= htmlspecialchars($categories[array\_search($service['Id\_category'], array\_column($categories, 'Id'))]['Name']) ?></td>

<td>

<a href="?delete\_service=<?= $service['Id'] ?>" class="delete">Удалить</a>

<a href="edit\_service.php?id=<?= $service['Id'] ?>">Редактировать</a>

</td>

</tr>

<?php endforeach; ?>

</table>

</div>

<!-- Управление записями на услуги -->

<div class="section">

<h2>Записи на услуги</h2>

<form method="POST">

<select name="client\_id" required>

<option value="">Выберите клиента</option>

<?php

$clients = $pdo->query("SELECT Id, Fam, Name, Lastfam FROM client")->fetchAll(PDO::FETCH\_ASSOC);

foreach ($clients as $client): ?>

<option value="<?= $client['Id'] ?>">

<?= htmlspecialchars($client['Fam'] . ' ' . $client['Name'] . ' ' . $client['Lastfam']) ?>

</option>

<?php endforeach; ?>

</select>

<select name="service\_id" required>

<option value="">Выберите услугу</option>

<?php foreach ($services as $service): ?>

<option value="<?= $service['Id'] ?>">

<?= htmlspecialchars($service['Name']) ?> (<?= htmlspecialchars($categories[array\_search($service['Id\_category'], array\_column($categories, 'Id'))]['Name']) ?>)

</option>

<?php endforeach; ?>

</select>

<select name="master\_id" required>

<option value="">Выберите мастера</option>

<?php foreach ($masters as $master): ?>

<option value="<?= $master['Id'] ?>">

<?= htmlspecialchars($master['Fam'] . ' ' . $master['Name']) ?>

</option>

<?php endforeach; ?>

</select>

<input type="datetime-local" name="time" required>

<input type="tel" name="phone" placeholder="Номер телефона" required>

<button type="submit" name="add\_note">Добавить запись</button>

</form>

<?php //подсчёт суммы за месяц

if (isset($\_GET['month']) && isset($\_GET['year'])) {

$month = $\_GET['month'];

$year = $\_GET['year'];

$stmt = $pdo->prepare("

SELECT SUM(service.Price) AS TotalAmount

FROM note

JOIN service ON note.Id\_service = service.Id

WHERE IsPaid = 1

AND YEAR(note.Time) = ? AND MONTH(note.Time) = ?

");

$stmt->execute([$year, $month]);

$totalAmount = $stmt->fetchColumn();

echo "<p>Сумма выполненных и оплаченных услуг за выбранный месяц: <strong>$totalAmount руб.</strong></p>";

}

//сумма за год

if (isset($\_GET['year'])) {

$year = $\_GET['year'];

$stmt = $pdo->prepare("

SELECT SUM(service.Price) AS TotalAmount

FROM note

JOIN service ON note.Id\_service = service.Id

WHERE IsPaid = 1

AND YEAR(note.Time) = ?

");

$stmt->execute([$year]);

$totalAmount = $stmt->fetchColumn();

echo "<p>Сумма выполненных и оплаченных услуг за год: <strong>$totalAmount руб.</strong></p>";

}

?>

<table>

<tr>

<th>Клиент</th>

<th>Услуга</th>

<th>Мастер</th>

<th>Время</th>

<th>Телефон</th>

<th>Оплачено</th>

<th>Действия</th>

</tr>

<?php foreach ($notes as $note): ?>

<tr>

<td><?= htmlspecialchars($note['client\_fam'] . ' ' . $note['client\_name']) ?></td>

<td><?= htmlspecialchars($services[array\_search($note['Id\_service'], array\_column($services, 'Id'))]['Name']) ?></td>

<td><?= htmlspecialchars($masters[array\_search($note['Id\_master'], array\_column($masters, 'Id'))]['Fam']) ?></td>

<td><?= htmlspecialchars($note['Time']) ?></td>

<td><?= htmlspecialchars($note['Phone\_client']) ?></td>

<td>

<input type="checkbox" name="is\_paid" data-note-id="<?= $note['Id'] ?>"

<?= $note['IsPaid'] ? 'checked' : '' ?> onchange="updateNoteStatus(this)">

</td>

<td>

<a href="edit\_note.php?id=<?= $note['Id'] ?>">Редактировать</a>

<a href="?delete\_note=<?= $note['Id'] ?>" class="delete">Удалить</a>

<a href="?export\_notes=1">Экспорт в Excel</a>

</td>

</tr>

<?php endforeach; ?>

</table>

<form method="GET" action="">

<label for="month">Месяц:</label>

<select name="month" id="month">

<option value="">Выберите месяц</option>

<?php for ($i = 1; $i <= 12; $i++): ?>

<option value="<?= $i ?>" <?= isset($\_GET['month']) && $\_GET['month'] == $i ? 'selected' : '' ?>>

<?= DateTime::createFromFormat('!m', $i)->format('F') ?>

</option>

<?php endfor; ?>

</select>

<label for="year">Год:</label>

<input type="number" name="year" id="year" value="<?= $\_GET['year'] ?? date('Y') ?>" required>

<button type="submit">Показать</button>

</form>

</div>

</body>

</html>

<?php

// Подключение к базе данных

$host = 'localhost';

$dbname = 'stylist';

$user = 'root';

$password = '8308';

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8", $user, $password); //подключение к PMA

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Ошибка подключения к базе данных: " . $e->getMessage());

}

$registrationSuccess = isset($\_GET['registration']) && $\_GET['registration'] === 'success'; //успешная регистрация

// Получение списка услуг

$services = $pdo->query("SELECT \* FROM service")->fetchAll(PDO::FETCH\_ASSOC);

// Получение списка мастеров

$masters = $pdo->query("SELECT \* FROM master")->fetchAll(PDO::FETCH\_ASSOC);

$registrationSuccess = isset($\_GET['registration']) && $\_GET['registration'] === 'success'; // Успешная регистрация

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>StyList</title>

<link rel="stylesheet" href="slide.css">

<link rel="stylesheet" href="index.css">

<script src = "indexsesion.js"></script>

</head>

<body>

<?php if ($registrationSuccess): ?>

<div class="notification">Регистрация прошла успешно!</div>

<?php endif; ?>

<?php if (isset($\_GET['review']) && $\_GET['review'] === 'success'): ?>

<div class="notification">Спасибо за ваш отзыв!</div>

<?php endif; ?>

<header>

<div class="logo">StyList</div>

<nav>

<div class="burger" onclick="toggleMenu()">

<div></div>

<div></div>

<div></div>

</div>

<ul id="nav-menu">

<li><a href="account.php">Аккаунт</a></li>

<li><a href="index4.php">Услуги</a></li>

<li><a href="index3.php">Мастера</a></li>

</ul>

</nav>

</header>

<!-- Слайдер для новостей и акций -->

<div id="slider">

<div class="slides">

<!-- First slide -->

<div class="slider">

<div class="legend"></div>

<div class="content">

<div class="content-txt">

<h1>40%</h1>

<h2>На первое окрашивание</h2>

</div>

</div>

<div class="images">

<img src="img/1.jpg">

</div>

</div>

<!-- Second slide -->

<div class="slider">

<div class="legend"></div>

<div class="content">

<div class="content-txt">

<h1>ПОДАРОЧНЫЕ СЕРТИФИКАТЫ</h1>

<h2>покупайте в салоне</h2>

</div>

</div>

<div class="images">

<img src="img/2.jpg">

</div>

</div>

<!-- Third slide -->

<div class="slider">

<div class="legend"></div>

<div class="content">

<div class="content-txt">

<h1>Новые мастера</h1>

<h2>Новая услуга: завивка</h2>

</div>

</div>

<div class="images">

<img src="img/3.png">

</div>

</div>

</div>

</div>

<main>

<div class="welcome-message">

<h1>Добро пожаловать, дорогой гость!</h1>

</div>

</main>

<footer>

&copy; 2023 Парикмахерская. Все права защищены.

</footer>

<button class="online-booking" onclick="openModal()">Онлайн запись</button>

<div class="overlay" id="overlay"></div>

<div class="modal" id="modal">

<h2>Онлайн запись</h2>

<form method="POST" action="book\_service.php">

<select name="service\_id" required>

<option value="">Выберите услугу</option>

<?php foreach ($services as $service): ?>

<option value="<?= $service['Id'] ?>">

<?= htmlspecialchars($service['Name']) ?> (<?= htmlspecialchars($service['Price']) ?> руб.)

</option>

<?php endforeach; ?>

</select>

<select name="master\_id" required>

<option value="">Выберите мастера</option>

<?php foreach ($masters as $master): ?>

<option value="<?= $master['Id'] ?>">

<?= htmlspecialchars($master['Fam'] . ' ' . $master['Name']) ?>

</option>

<?php endforeach; ?>

</select>

<input type="date" name="date" required>

<input type="time" name="time" required>

<input type="tel" name="phone" placeholder="Номер телефона" required>

<button type="submit">Записаться</button>

<button type="button" onclick="closeModal()">Закрыть</button>

</form>

</div>

<!-- Форма для отзыва -->

<section class="review-section">

<h2>Оставьте отзыв</h2>

<form id="reviewForm" method="POST">

<input type="text" name="client\_name" placeholder="Ваше имя" required>

<textarea name="review\_text" placeholder="Ваш отзыв" required></textarea>

<select name="rating" required>

<option value="">Оцените нас</option>

<option value="5">5 - Отлично</option>

<option value="4">4 - Хорошо</option>

<option value="3">3 - Удовлетворительно</option>

<option value="2">2 - Плохо</option>

<option value="1">1 - Ужасно</option>

</select>

<button type="submit" id="submitReview">Отправить отзыв</button>

</form>

<div id="reviewMessage"></div> <!-- Сообщение об успешной отправке или ошибке -->

</section>

<!-- Стили для формы отзыва -->

<style>

.review-section {

margin: 20px auto;

padding: 20px;

max-width: 600px;

background: rgb(213, 164, 202);

border-radius: 10px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.review-section h2 {

text-align: center;

margin-bottom: 20px;

}

.review-section form {

display: flex;

flex-direction: column;

gap: 10px;

}

.review-section input,

.review-section textarea,

.review-section select {

padding: 10px;

border: 1px solid #ccc;

border-radius: 5px;

font-size: 16px;

}

.review-section textarea {

resize: vertical;

height: 100px;

}

.review-section button {

padding: 10px;

background: #cf54bf;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

}

.review-section button:hover {

background: #cf54bf;

}

.notification {

position: fixed;

top: 20px;

right: 20px;

padding: 15px;

background: #4CAF50;

color: white;

border-radius: 5px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

z-index: 1000;

}

</style>

<?php if (isset($\_GET['booking']) && $\_GET['booking'] === 'success'): ?>

<div class="notification">Запись на услугу успешно оформлена!</div>

<?php endif; ?>

</body>

</html

function openModal() {

document.getElementById('modal').style.display = 'block';

document.getElementById('overlay').style.display = 'block';

}

function closeModal() {

document.getElementById('modal').style.display = 'none';

document.getElementById('overlay').style.display = 'none';

}

//запись онлайн

function submitBooking() {

const date = document.getElementById('date').value;

const time = document.getElementById('time').value;

const phone = document.getElementById('phone').value;

if (date && time && phone) {

alert(`Запись на ${date} в ${time} успешно оформлена! Мы свяжемся с вами по номеру ${phone}.`);

closeModal();

} else {

alert('Пожалуйста, заполните все поля.');

}

}

function toggleMenu() {

const navMenu = document.getElementById('nav-menu');

navMenu.classList.toggle('active');

}