Автономная некоммерческая образовательная организация

высшего образования Центросоюза Российской Федерации

СИБИРСКИЙ УНИВЕРСИТЕТ

ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ

Кафедра информатики

**ОТЧЕТ О ПРОХОЖДЕНИИ ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКИ**

Место прохождения практики: *Общество с ограниченной ответственностью «Инженерное бюро»*

Обучающегося 4 курса

Тропина Даниила Александровича

ПИБ-11, ПИБ-О-01-21-009

Руководитель практики

преподаватель (практик)

Блок И.Н.

Оценка после защиты

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Сроки прохождения практики:

с 02.04.2025

по 16.05.2025

Новосибирск

2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc197269872)

[РАЗДЕЛ 1. ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ ОРГАНИЗАЦИИ. 4](#_Toc197269873)

[1.1 Анализ организации 4](#_Toc197269874)

[1.2 Организационная структура 5](#_Toc197269875)

[1.3 Технологический процесс 6](#_Toc197269876)

[РАЗДЕЛ 2. ИССЛЕДОВАНИЕ ТЕХНОЛОГИЙ ДЛЯ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ. 10](#_Toc197269877)

[2.1 Обзор программных решений для веб-разработки 10](#_Toc197269878)

[2.2 Анализ IDE для разработки веб-приложений 15](#_Toc197269879)

[2.3 Сравнение и обоснование выбора СУБД 18](#_Toc197269880)

[2.4 Анализ инструментов объектно-реляционного отображения 21](#_Toc197269881)

[РАЗДЕЛ 3. ЭТАПЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ 23](#_Toc197269882)

[3.1 Мероприятия по созданию ИС 23](#_Toc197269883)

[3.2 Мероприятия по реализации веб-приложения и введения в эксплуатацию 25](#_Toc197269884)

[ЗАКЛЮЧЕНИЕ 29](#_Toc197269885)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 30](#_Toc197269886)

# ВВЕДЕНИЕ

Целью технологической (проектно-технологической) практики является расширение и углубление профессиональных практических знаний, умений, навыков применения самостоятельных решений на конкретном участке работы путем выполнения в условиях производства различных обязанностей, свойственных будущей профессиональной деятельности, а также формирование практических умений и навыков ведения исследовательской работы для подготовки и защиты выпускной квалификационной работы.

Задачи преддипломной практики включают:

- овладение современными методами сбора, анализа и обработки научной информации в области информатики и вычислительной техники;

- овладение основами компьютерной обработки информации с помощью современных прикладных программ;

- получение опыта оформления технической документации;

- изучение структурных и функциональных схем предприятия, организации деятельности подразделений, требований к техническим и программным средствам, используемым на предприятии;

- изучение основных характеристик и параметров производственных и технологических процессов;

- систематизация и углубление теоретических знаний и практических навыков в области проектирования информационных систем;

- получение теоретических и практических результатов, необходимых для успешного выполнения и защиты выпускной квалификационной работы.

# 

# РАЗДЕЛ 1. ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ПРЕДПРИЯТИЯ ОРГАНИЗАЦИИ.

## 1.1 Анализ организации

За период прохождения практики была проанализирована работа организации ООО «Инженерное Бюро». Компания специализируется на продаже, монтаже и поставке систем вентиляции и кондиционирования. С момента основания в 2012 году основной доход компания получает от участия в тендерах и заключения прямых договоров с клиентами.

Деятельность компании включает весь цикл работы с системами вентиляции и кондиционирования. Сначала проводится обследование объекта и разработка проекта системы. Затем выполняется монтаж и запуск оборудования в эксплуатацию. После установки компания обеспечивает техническое обслуживание и ремонт систем. Также доступна услуга по модернизации существующих систем для повышения их эффективности.

Компания предлагает качественные решения для создания комфортного микроклимата в помещениях. Оборудование устанавливается профессионально, а регулярное обслуживание гарантирует его долгую и надежную работу. Клиенты могут быть уверены в высоком уровне сервиса на всех этапах сотрудничества.

Стоит отметить, что компания использует современное оборудование и технологии, что позволяет ей оставаться конкурентоспособной на рынке. Процессы внутри компании организованы четко, начиная от планирования проекта и заканчивая его реализацией. Это помогает поддерживать высокий уровень качества услуг для клиентов.

## 

## 1.2 Организационная структура

Организация «Инженерное Бюро» функционирует по принципу линейной структуры управления, которая отличается ясным распределением обязанностей и четкой иерархией. Организационная модель управления включает несколько ключевых управленческих блоков:

На вершине организационной структуры находится Правление, данное подразделение осуществляет руководство компанией в соответствии с законодательством РФ. Директор несет полную ответственность за принимаемые решения и определяет стратегическое направление развития бизнеса.

Административный отдел оказывает поддержку руководителю на бизнес-встречах, контролирует исполнение его поручений и обеспечивает взаимодействие между различными отделами. Одной из задач отдела является контроль выполнения задач на уровне не менее 90% от месячных планов. Также административный отдел ведет журнал всей корреспонденции и внедряет инструменты, облегчающие управление организацией для генерального директора.

Тендерный отдел занимается анализом рынка, поиском выгодных тендеров и подготовкой необходимой документации для участия в них. Специалисты отдела следят за регистрацией и продлением аккредитации на площадках государственных закупок и коммерческих сайтов. Кроме того, они собирают и формируют все нужные документы, такие как декларации, платёжные поручения, независимая гарантия, письма о деловой репутации и другие.

Производственный отдел, находящийся под руководством главного инженера, выполняет монтажные работы. Этот отдел тесно сотрудничает с административным отделом для координации вопросов закупок и обеспечения необходимыми материалами.

Отдельного внимания заслуживает новый отдел по работе с клиентами, создание которого стало частью практики. Этот отдел займется выполнением заказов клиентов, взаимодействуя с производственным отделом для реализации технических заданий и координации проектов.

Структура организации «Инженерное Бюро» наглядно представлена на Рисунке 1.

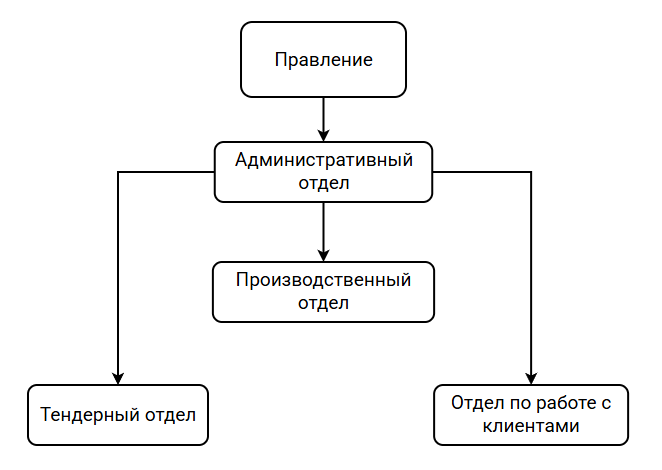


Рис. 1 - Организационная структура ООО «Инженерное Бюро»

## 

## 1.3 Технологический процесс

В результате анализа работы организации были выявлены основные бизнес-процессы, среди которых особое внимание уделяется следующим направлениям: управление заказами и взаимодействие с клиентами, производственный процесс монтажа систем вентиляции и кондиционирования, закупка оборудования и материалов, а также финансовый учет и бухгалтерия.

На данный момент в компании отсутствует автоматизация внутренних процессов. При поступлении заказа от клиента информация фиксируется вручную в CRM-системах (Yougile, Контур Закупки). Однако такая практика приводит к потере данных со временем, создавая ряд проблем, которые можно решить с помощью внедрения веб-приложения. Заказы принимаются как на личных встречах, так и через телефон или электронную почту. После этого административный отдел обрабатывает каждую заявку, занося данные в систему вручную и координируясь с директором и производственным отделом для уточнения деталей заказа.

Текущий подход имеет несколько существенных недостатков:

1. **Отсутствие автоматизации.** Ручной ввод данных в CRM-системы увеличивает вероятность ошибок и снижает эффективность работы. Административному отделу требуется значительное время для обработки каждого заказа, что замедляет процесс и может привести к задержкам.

2. **Централизация коммуникаций.** Все взаимодействия между отделами проходят через административный отдел, что усложняет процесс и повышает риск недопонимания.

Для решения этих проблем предлагается внедрение веб-приложения, которое позволит автоматизировать процессы управления заказами. Это решение потребует обучения сотрудников отдела по работе с клиентами, но в долгосрочной перспективе принесет значительные преимущества. Автоматизация рабочих процессов позволит компании не только ускорить выполнение заказов, но и увеличить их количество, что положительно скажется на прибыли (Рис. 2).

Требования, предъявленные к веб-приложению:

**- Реализация системы регистрации пользователей.** Потребуется разработать и внедрить функционал регистрации пользователей на сайте, а также настроить систему учета объема заказов для каждого клиента в дальнейшем данный механизм позволит предоставлять персональные предложения клиентам.

**- Создание адаптивного интерфейса для оформления заказов.** Необходимо создать адаптивный интерфейс для самостоятельного оформления заказов. Этот интерфейс должен быть удобным и интуитивно понятным, чтобы клиенты могли легко выбирать товары или услуги и оформлять заказы без помощи операторов. Важно обеспечить автоматическую фиксацию всех данных заказа, чтобы минимизировать ошибки, возникающие при ручном вводе информации. Это повысит точность обработки заказов и улучшит общее качество обслуживания.

**- Статус заказа.** Потребуется внедрить систему отслеживания статуса заказов в реальном времени. Это позволит клиентам всегда быть в курсе текущего состояния их заказов, начиная от момента оформления и заканчивая доставкой. Для этого требуется разработать функционал, который будет отображать статус заказа на каждом этапе его обработки, а также настроить уведомления для клиентов. Уведомления могут отправляться по email, SMS или через личный кабинет на сайте, что повысит уровень доверия и удовлетворенности пользователей.

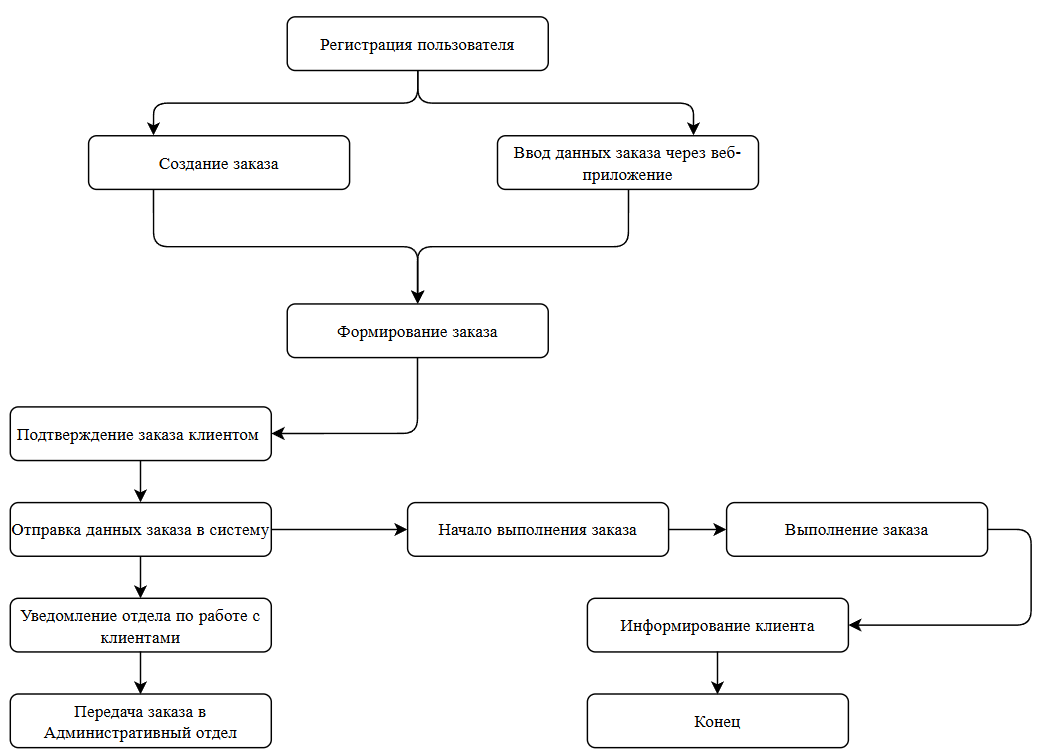


Рис. 2 - Динамические аспекты поведения системы

Внедрение веб-приложения поможет компании «Инженерное Бюро» оптимизировать внутренние процессы, сократить время на обработку заказов и минимизировать количество ошибок. Улучшение качества обслуживания клиентов сделает компанию более конкурентоспособной на рынке.

Кроме того, клиенты смогут использовать удобный интерфейс для самостоятельного оформления заказов, проверки наличия оборудования и отслеживания статуса. Это создаст дополнительные возможности для развития бизнеса и привлечения новых клиентов [5].

# РАЗДЕЛ 2. ИССЛЕДОВАНИЕ ТЕХНОЛОГИЙ ДЛЯ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ.

## 2.1 Обзор программных решений для веб-разработки

Сфера вентиляции и кондиционирования требует современных решений для эффективного взаимодействия с клиентами и управления проектами. Существует ряд специализированных программных продуктов, которые предлагают широкий функционал для этой отрасли, включая каталоги оборудования и услуг, возможность онлайн-оформления заказов, интеграцию с внутренними системами управления, а также инструменты для продвижения услуг и аналитики.

В ходе исследования рынка были изучены следующие программные решения [6]:

1. **Drupal** — это система управления контентом (CMS) с открытым исходным кодом, написанная на PHP. Она используется для создания сайтов различной сложности: от простых блогов до крупных корпоративных порталов, интернет-магазинов, социальных сетей и даже приложений. Drupal известен своей гибкостью, безопасностью и масштабируемостью.

При помощи Drupal можно создавать уникальные структуры сайтов, при помощи системы категорий и типов контента, уникальные поля дают возможность гибко адаптироваться к нестандартным требованиям заказчика, позволяя реализовывать уникальные идеи и задачи (Рис. 3). Drupal в своём арсенале имеет огромное хранилище модулей (более 40 000), благодаря этому расширяется функциональность системы. При помощи модулей у пользователя появляется возможность интеграции с внешними сервисами, такими как CRM системы, платежные системы, аналитические инструменты и другие.

Drupal принято считать одной из безопасных CMS (Content Management System) — система управления контентом, платформа для создания и управления сайтами.

Разработчики регулярно выпускают обновления и устраняют уязвимости. В системе предусмотрены встроенные механизмы защиты от атак, таких как SQL-инъекции. Благодаря своей надежности и высокой репутации, Drupal активно используется многими государственными учреждениями. Пропускная способность, отлично справляется с большим трафиком контента, что делает Drupal востребованным среди новостных порталов и социальных сетей. Drupal поддерживает создание многоязычных сайтов «из коробки», что делает его удобным для международных проектов. Регулярно проводятся конференции (DrupalCon), где обсуждаются новые тенденции и разработки.



Рис. 3 - Интерфейс Drupal

2. **TYPO3** — это профессиональная система управления контентом (CMS) с открытым исходным кодом, которая широко используется для создания сложных, многофункциональных и масштабируемых веб-сайтов. Она особенно популярна среди крупных организаций, корпораций и разработчиков, которым требуется высокая гибкость, безопасность и возможность управления большими объемами контента. Следует подробнее рассмотреть её особенности, преимущества, недостатки и сферы применения (Рис. 4).

TYPO3 позволяет создавать сайты любой сложности – от крупных корпоративных порталов до мультисайтовых платформ (преимущество для крупных организаций с множеством проектов). Система поддерживает множество языков и позволяет создавать многоязычные сайты. TYPO3 мощная система расширений, которая позволяет добавлять функции: интеграции с CRM и ERP-системы, электронной коммерции расширение (TYPO3 Commerce), сложные опросы и формы. Разработчикам предоставляется возможность создавать расширения самостоятельно, это приветствуется, так как появляется возможность создать уникальные проекты. TYPO3 известен своей высокой безопасностью. Регулярные обновления и строгая архитектура делают его подходящим для проектов, где безопасность данных критически важна (например, банки).



Рис. 4 - Интерфейс TYPO3

3. Специализированные CRM-системы: Такие системы как Bitrix24, amoCRM, HtbStop CRM, предлагают широкий спектр функций для управления клиентскими отношениями и заказами. Преимуществами являются: интеграция с различными бизнес-процессами, автоматизация взаимодействия с клиентами, мощные аналитические инструменты. Недостатками являются: Высокая стоимость для малых и средних предприятий, сложность в настройке и интеграции.

4. Альтернативные платформы для онлайн-заказа услуг: Сервисы такие как **Tilda**, **WordPress.com** и **Webflow** предоставляют инструменты для создания сайтов с функционалом по приему заказов и бронированию услуг.

Рассмотренные выше платформы, такие как Tilda, WordPress.com и Webflow, предоставляют удобные инструменты для создания сайтов с функционалом онлайн-заказов и бронирования услуг. Они отличаются простотой использования, интуитивным интерфейсом и позволяют быстро запустить сайт даже без глубоких знаний в программировании. Кроме того, эти платформы предлагают встроенные возможности для интеграции с платежными системами и календарями, что упрощает процесс настройки. Однако у них есть и ограничения: многие продвинутые функции доступны только через платные подписки или дополнительные модули, а бесплатные версии часто ограничивают возможности кастомизации дизайна и функционала.

Эти ограничения могут стать препятствием для реализации сложных или специфических требований бизнеса. В таких случаях целесообразно рассмотреть разработку собственного решения, которое будет полностью соответствовать нуждам проекта. Одним из таких решений может стать использование технологии ASP.NET. Это современная платформа для создания веб-приложений, которая позволяет разрабатывать интерактивные и масштабируемые решения с использованием языка программирования C# и экосистемы .NET. ASP.NET MVC.

В частности, ASP.NET MVC является привлекательным выбором благодаря следующим преимуществам [3].

**Краткое описание преимуществ ASP.NET MVC:**

**1. Архитектура MVC:** Четкое разделение на Модель (данные и логика), Представление (интерфейс) и Контроллер (управление), упрощающее разработку, тестирование и поддержку, особенно в крупных проектах.

**2. Серверный рендеринг:** Генерация HTML на сервере обеспечивает excellent SEO-оптимизацию и быструю начальную загрузку страниц.

**3. Интеграция с .NET:** Полная совместимость с экосистемой .NET, включая Entity Framework, Dependency Injection и технологии Microsoft (Azure, SQL Server).

**4. Гибкость:** Поддержка RESTful API, Razor для динамических страниц, фильтров, маршрутизации и валидации моделей.

**5. Современные стандарты:** Встроенные инструменты для HTTP/2, WebSockets, gRPC и SignalR для реального времени.

**6. Кроссплатформенность:** Возможность разработки и запуска приложений на Windows, Linux и macOS благодаря .NET Core (.NET 5+).

**7. Дружественный для .NET-разработчиков:** Простота освоения для команд, работающих с C# и .NET, что снижает затраты на обучение.

**8. Поддержка legacy-систем:** Хорошо подходит для модернизации и поддержки старых приложений, созданных на классическом ASP.NET.

Таким образом, ASP.NET MVC становится идеальным решением для разработки веб-приложений, сочетая в себе гибкость, масштабируемость и удобство как в процессе разработки, так и в дальнейшей поддержке проекта. Благодаря четкой архитектуре, широким возможностям настройки и интеграции с современными технологиями, этот фреймворк позволяет создавать надежные, производительные и легко тестируемые приложения. Кроме того, его совместимость с актуальными стандартами разработки и кроссплатформенность делают ASP.NET MVC особенно привлекательным выбором. Это обеспечивает не только высокое качество продукта, но и возможность его дальнейшего развития без значительных затрат на модификацию или переписывание кода [4].

## 

## 2.2 Анализ IDE для разработки веб-приложений

Поскольку для разработки веб-приложения был выбран фреймворк ASP.NET MVC, важно подобрать среду разработки, которая максимально соответствует его особенностям и предоставляет все необходимые инструменты для эффективной и удобной работы. Правильный выбор IDE позволит не только упростить процесс разработки, но и повысить производительность за счет интегрированных возможностей и оптимизированных инструментов, специально адаптированных для работы с этим фреймворком.

Ниже приведен обзор некоторых из существующих на данный момент сред разработки для написания кода [8].

**Microsoft Visual Studio —** это интегрированная среда разработки (IDE) от Microsoft, которая оптимально подходит для работы с фреймворком ASP.NET MVC. Она предоставляет обширный набор инструментов для разработки веб-приложений на платформе .NET и поддерживает язык программирования C# и другие технологии, используемые в экосистеме .NET. К преимуществам можно то, что бесплатная версия «Community Edition» для рядового пользователя будет иметь вполне себе обширный функционал, который можно расширить, подключив плагины (Рис. 5).

****

Рис. 5 - Интерфейс Microsoft Visual Studio

**Visual Studio Code —** кроссплатформенный редактор кода, который при помощи плагинов можно превратить в подобие IDE. Кроссплатформенным его делает то, что он работает на MacOS, Ubuntu и Windows. Данный редактор кода является open-source редактором и плата за плагины не требуется. Также для того, чтобы приступить к работе не потребуется много ресурсов, так как он является достаточно легковесным. Несмотря на поддержку .NET-платформы, Visual Studio Code неудобен для сложных проектов. К минусам можно отнести сомнительную надежность, так как многие надстройки имеют низкое качество сборки и не всегда выполняют даже основные функции (Рис. 6).

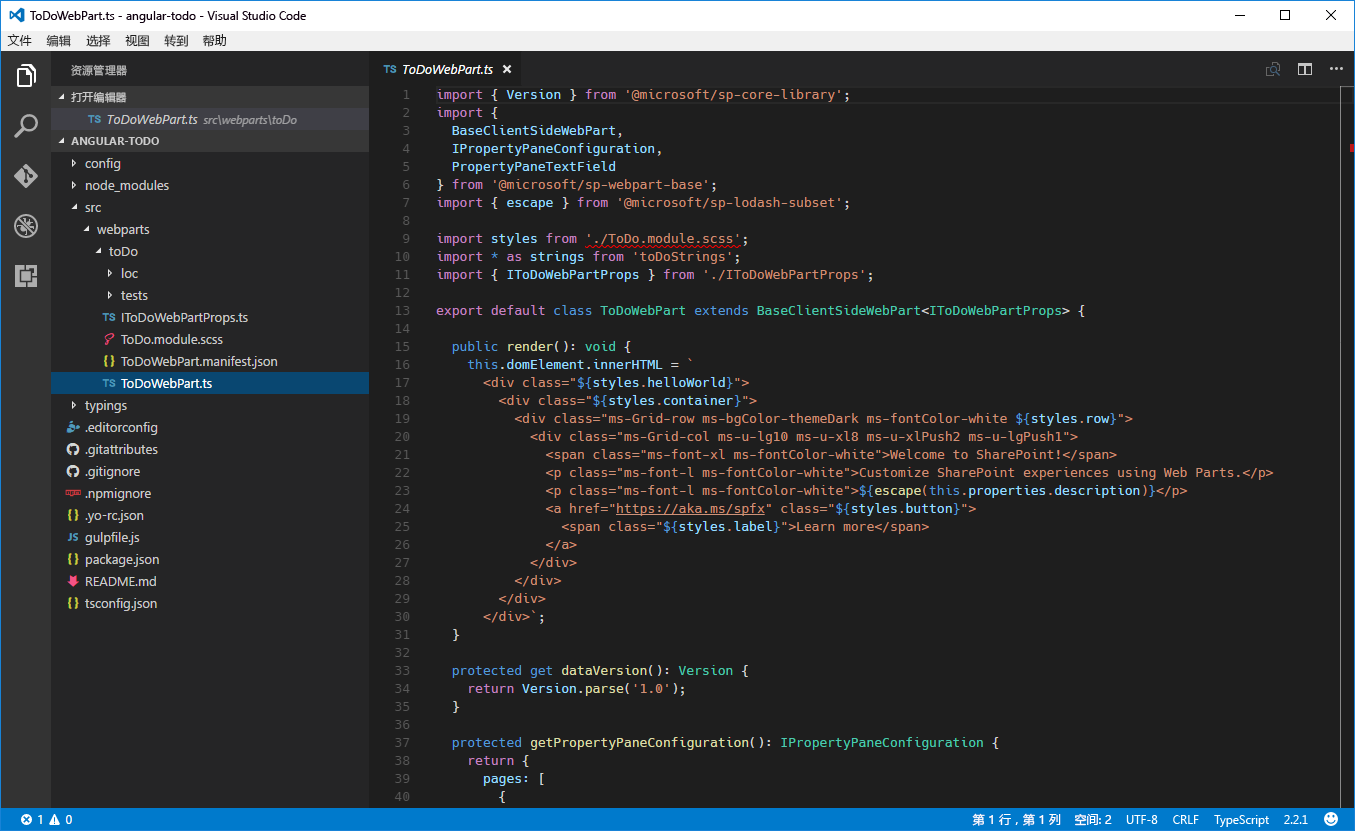


Рис. 6 - Интерфейс Visual Studio Code

**IntelliJ IDEA** – мощная интегрированная среда разработки (IDE) для написания программного обеспечения, данная среда разработана крупным игроком на рынке JetBrains. Благодаря своей функциональности и удобству пользования активно используется в мире Java-разработки. К основным особенностям можно отнести: изначально среда была заточена под Java, со временем она начала поддерживать множество других языков программирования, таких как Kotlin, Scala, Groovy, Python, JavaScript, TypeScript, HTML/CSS, SQL и многие другие. IntelliJ IDEA имеет отличную поддержку множества популярных фреймворков и библиотек, таких как Spring, Hibernate, Android SDK, React, Angular и Vue.js. Это позволяет разработчикам использовать эти технологии без необходимости дополнительной настройки. К минусам можно отнести: IntelliJ IDEA требует значительных вычислительных ресурсов (CPU, RAM), особенно если проекты большие и используются много плагинов, а так же Ultimate Edition является платной, что может быть недоступно для некоторых разработчиков или маленьких компаний (Рис. 7).

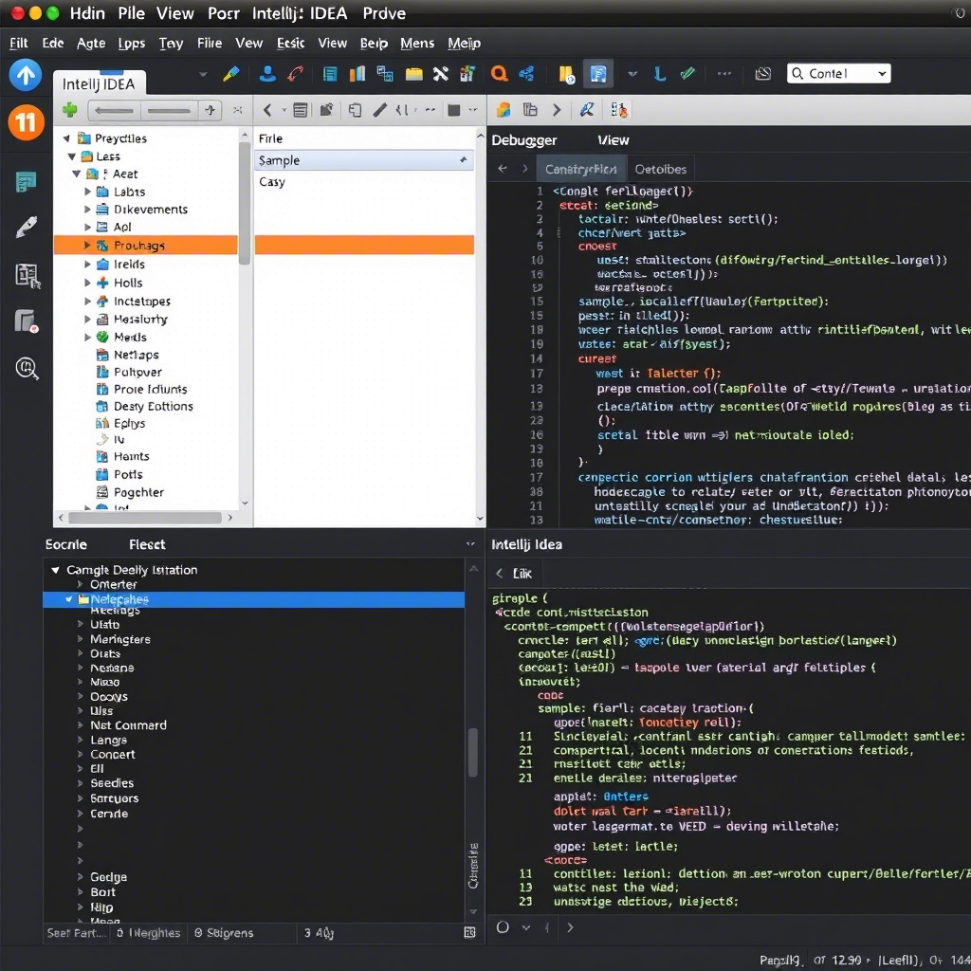


Рис. 7 - Пользовательский интерфейс IntelliJ IDEA

**MonoDevelop –** бесплатная свободная среда разработки от Xamarin, аналогичная IDE для создания приложений на множестве языков, а так же поддержка основных операционных систем. Возможность кастомизации позволяет пользователям настраивать расположение функций и окон в соответствии с их личными предпочтениями. К сожалению, несмотря на то, что у MonoDevelop есть собственный отладчик и инструменты для работы с кодом, в части поддержки разных платформ и проектов – это пока сырая IDE (Рис. 8).

****

Рис. 8 - Интерфейс MonoDevelop

На основе этих факторов выбор среды разработки пал на Microsoft Visual Studio по причине хорошей интеграции работы с фреймворком ASP.NET MVC.

## 

## 2.3 Сравнение и обоснование выбора СУБД

В процессе разработки веб-приложения одним из ключевых этапов является выбор подходящей системы управления базами данных (СУБД). СУБД будет отвечать за хранение, управление и обработку данных, таких как информация о клиентах, заказах и услугах. При этом особое внимание уделяется поддержке транзакций, обеспечивающих безопасность операций, например, при обработке платежей.

**Анализ существующих решений:**

1. **MS SQL Server** представляет собой реляционную СУБД с высокой надежностью и строгим соответствием стандартам SQL. Она обеспечивает полную поддержку транзакций ACID (атомарность, согласованность, изолированность, устойчивость), а также расширяемость благодаря множеству доступных модулей и инструментов. MS SQL Server идеально подходит для выполнения сложных запросов и аналитических задач, хотя может потребовать дополнительных усилий для настройки и администрирования.

2. **MySQL и MariaDB** это популярные реляционные СУБД, известные своей простотой использования, высокой производительностью и широкой поддержкой сообщества. Несмотря на свои преимущества не соответствуют задуманному проекта. Основные причины - MySQL и MariaDB менее эффективны при работе с многоуровневыми запросами и аналитическими задачами, что может стать ограничением для обработки данных о клиентах, заказах и услугах. Они уступают другим решениям (например, MS SQL Server) в строгом соблюдении стандартов SQL, что может привести к проблемам при реализации сложной бизнес-логики. Хотя поддерживают транзакции, их возможности в плане надежности и масштабируемости транзакций ACID менее развиты по сравнению с MS SQL Server.

3. **MongoDB**

Как одна из популярных NoSQL-систем, MongoDB предлагает гибкость в хранении данных в формате JSON-подобных документов, что делает ее удобной для динамических схем и быстрой разработки приложений. Начиная с версии 4.0, MongoDB поддерживает транзакции ACID, но все еще имеет ограничения в работе со сложными запросами и операциями.

4. **Cassandra** отличается высокой масштабируемостью и доступностью, что делает ее привлекательной для распределенных систем с большим объемом записей и чтений. Однако настройка и управление этой СУБД могут быть достаточно сложными, а поддержка сложных запросов и транзакций остается ограниченной.

Учитывая требования проекта, был принято решение выбрать MS SQL Server Express.

MS SQL Server обеспечивает полную поддержку транзакций ACID, что критично для обработки данных о клиентах, заказах и платежах. Это гарантирует, что все операции будут выполнены корректно, а данные останутся целостными даже в случае сбоев. В отличие от MySQL и MariaDB, которые также поддерживают транзакции, но уступают в надежности и строгости соблюдения стандартов SQL, MS SQL Server предоставляет более продвинутые инструменты для работы с транзакциями и аналитическими задачами.

Кроме того, MS SQL Server идеально подходит для выполнения сложных запросов, что важно для анализа данных о клиентах и заказах. MySQL и MariaDB показывают себя хуже при работе с многоуровневыми запросами, а NoSQL-решения, такие как MongoDB и Cassandra, вообще не предназначены для таких задач.

Еще одно преимущество MS SQL Server — его глубокая интеграция с ASP.NET MVC, который используется в проекте. Это позволяет разработчикам легко взаимодействовать с базой данных через Entity Framework Core, упрощая процесс разработки и сокращая время настройки.

Наконец, MS SQL Server Express предлагает удобные инструменты администрирования, развитую экосистему и хорошую масштабируемость. Хотя он требует больше усилий для настройки по сравнению с MySQL или MariaDB, эти затраты оправданы за счет более высокой производительности и надежности.

Таким образом, MS SQL Server Express стал оптимальным выбором благодаря своей надежности, поддержке сложных запросов, интеграции с ASP.NET MVC и удобству использования в рамках проекта.

## 

## 2.4 Анализ инструментов объектно-реляционного отображения

После выбора системы управления базами данных (СУБД), следующим важным шагом становится выбор инструмента для объектно-реляционного отображения, или ORM. Этот инструмент позволяет работать с базой данных через объекты, что делает код более понятным и упрощает процесс разработки. Благодаря автоматическому преобразованию данных между базой и объектами, а также управлению транзакциями, ORM существенно облегчает жизнь разработчикам проекта [7].

**Описание нескольких популярных решений:**

**1. Entity Framework Core** — это современный и мощный инструмент от Microsoft, который отлично работает с платформой .NET и фреймворком Blazor. Он поддерживает множество СУБД, таких как SQL Server, MySQL, PostgreSQL и SQLite, и предлагает высокую производительность благодаря встроенным асинхронным операциям. Однако его настройка может быть сложной для небольших проектов, где часть функций может оказаться избыточной.

**2. Dapper** — это легковесный микрофреймворк, который отличается своей простотой и скоростью работы. Он требует минимальной настройки и позволяет писать оптимизированные SQL-запросы. Dapper подходит для тех случаев, когда нужно получить максимальную производительность без лишних наворотов. Однако он не предоставляет автоматического управления миграциями и других высокоуровневых возможностей, которые есть у более продвинутых ORM.

**3. NHibernate** — это гибкий инструмент с широкими возможностями, поддерживающий различные паттерны проектирования, кэширование и оптимизацию запросов. Он идеально подходит для крупных корпоративных приложений, но его освоение и настройка могут быть довольно трудоемкими. Плюс ко всему, его интеграция с современными технологиями .NET и Blazor менее глубокая, чем у Entity Framework Core.

**4. Linq to SQL** — это простой инструмент, хорошо подходящий для небольших проектов. Он легко интегрируется с C# и поддерживает LINQ-запросы. Однако его возможности ограничены, и он работает только с Microsoft SQL Server, что делает его непригодным для масштабируемых и сложных решений.

На основе анализа всех этих вариантов был выбран Entity Framework Core, так как он предлагает лучший баланс между универсальностью, производительностью и удобством использования. Глубокая интеграция с .NET и Blazor обеспечивает плавное взаимодействие всех компонентов приложения. Поддержка различных СУБД дает возможность выбрать наиболее подходящую базу данных для конкретного проекта. Кроме того, наличие инструментов для миграции и асинхронные операции позволяют создавать быстрые и масштабируемые приложения. Обширная документация и активное сообщество помогают быстро находить решения для возникающих проблем.

# РАЗДЕЛ 3. ЭТАПЫ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ

## 3.1 Мероприятия по созданию ИС

Разрабатываемая система должна решать задачу автоматизации поступления заказов клиента и выполнять следующие функции:

- ввод, хранение, изменение и удаление информации о товарах и услугах;

- оформление заказа;

- расчет стоимости заказа.

Веб-приложение будет выполнять все функции, необходимые для оформления заказов в организации «Инженерное Бюро».

Для создания веб-приложения требуется разработать концептуальную модель данных в виде ER-диаграммы. Эта модель поможет определить основные сущности и установить возможные связи между ними.

Концептуальная модель базы данных представлена на Рисунке 9:

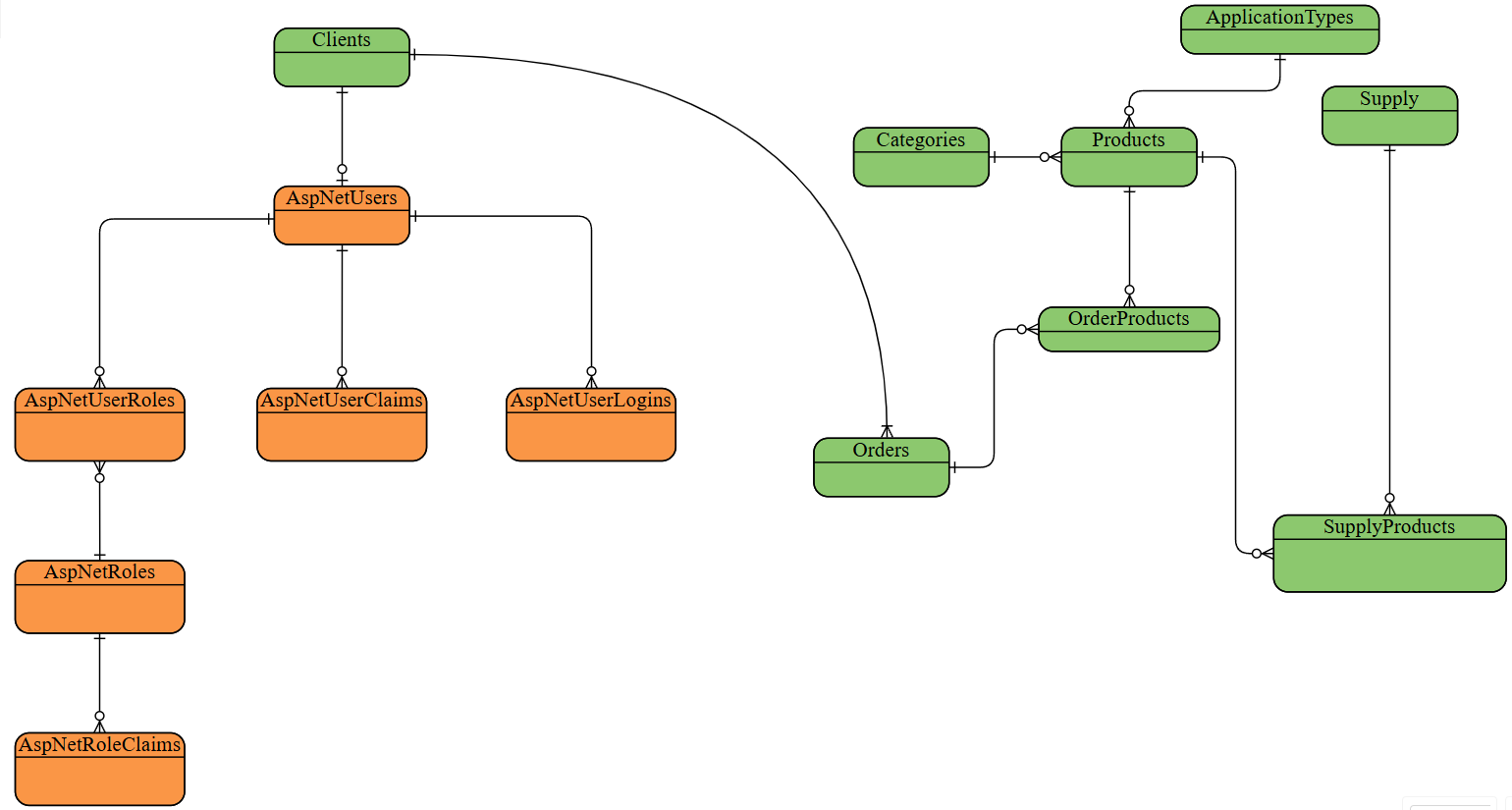


Рис. 9 - Концептуальная модель базы данных

На основе концептуальной диаграммы необходимо разработать логическую модель базы данных. Эта модель будет отображать, какие данные хранятся в базе, включая объекты предметной области, их атрибуты и связи между ними. Структура логической модели представлена на Рисунке 10:

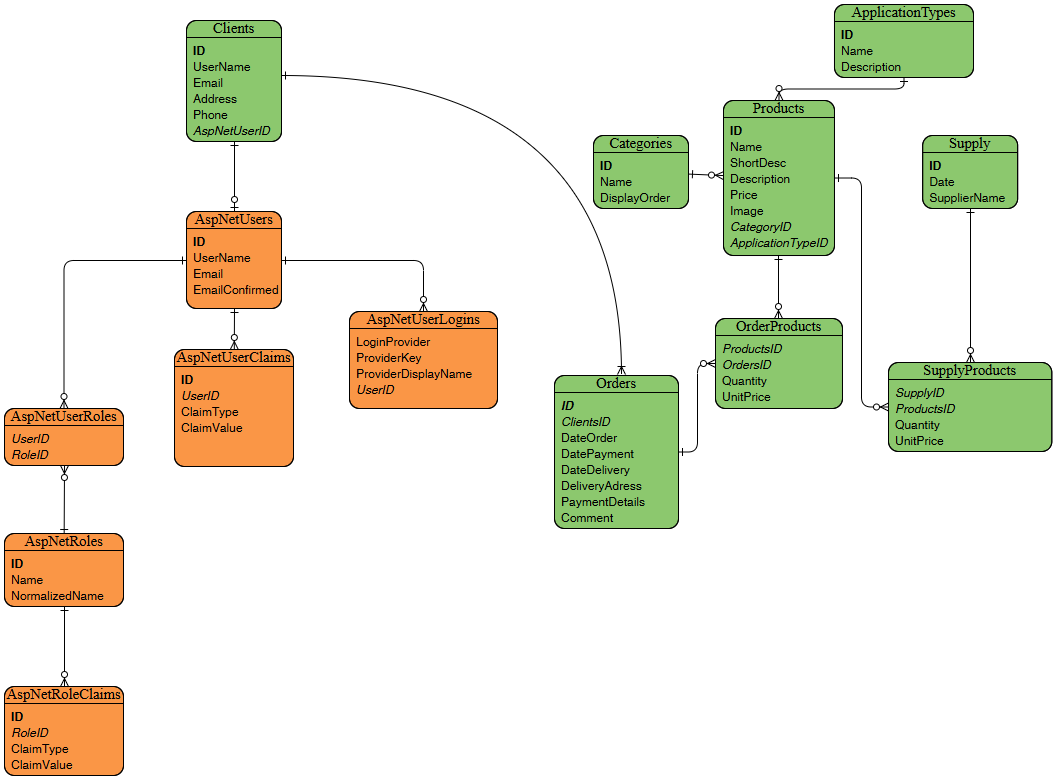


Рис. 10 - Логическая модель базы данных

Технологический процесс обработки информации состоит из таких операций, как:

- сбор данных, представляющий собой процесс регистрации, фиксации, записи информации о связях, событиях, объектах, действиях, признаках;

- обработка данных, к которой относятся следующие действия: расчеты, выборка, слияние и другие;

- генерирование информации, которое заключается в организации, переорганизации и преобразовании данных в форму, необходимую пользователям;

- хранение данных и информации, включающее размещение, накопление, копирование и выработку данных и информации с целью их последующего применения;

передача информации и данных, которая представляет собой распространение информации и данных между всеми пользователями при помощи средств и систем коммуникаций и путем перемещения данных от источника (отправителя) к приемнику (получателю).

## 

## 3.2 Мероприятия по реализации веб-приложения и введения в эксплуатацию

Выбор клиент-серверной архитектуры для реализации проектируемого приложения обоснован тем, что данная модель отвечает всем необходимым требованиям, к примеру: централизованное управления, улучшенная безопасность, а также возможность масштабирования (Рис. 11). Реализация распределенной системы на основе SQL-сервера обеспечит оптимальное распределение вычислительных ресурсов между клиентским приложением и центральным хранилищем данных [1]. Централизованное хранение информации в SQL-сервере позволит поддерживать актуальность данных и обеспечить их согласованность во всех точках системы. SQL-сервер будет выполнять функции координатора взаимодействия клиентского приложения, обеспечивая эффективную обработку сложных запросов с использованием хранимых процедур и триггеров. Такой подход оптимизирует сетевое взаимодействие за счет минимизации объема передаваемых данных и исключения конфликтных ситуаций при одновременном доступе к данным.

Клиентская часть представляет собой интерфейс, доступный через браузер, что позволяет пользователям выполнять все нужные действия независимо от их местоположения. Интерфейс включает главную страницу с каталогом продукции, а также удобную навигационную панель для быстрого перехода к детальной информации по каждому товару. После достижения базовой готовности проект переходит на этап внедрения, который является ключевым для команды разработчиков. На этом этапе проверяется соответствие созданного продукта первоначальным требованиям заказчика. После успешного завершения внедрения система передается в эксплуатацию, где могут возникнуть новые задачи, связанные с развитием бизнес-процессов или изменениями в области технологий. На этом этапе продолжается работа над оптимизацией приложения, включая повышение производительности, исправление недочетов и добавление требуемых функций [2].

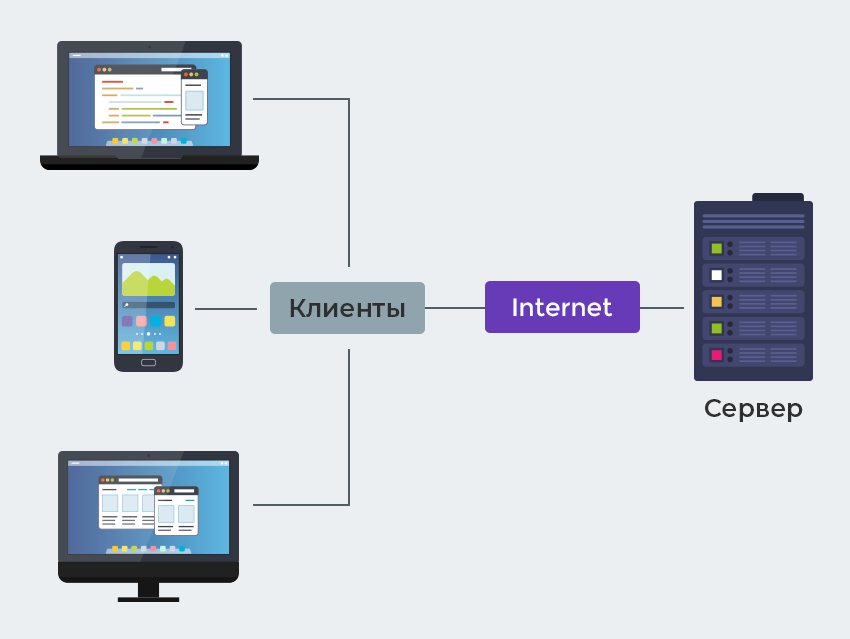


Рис. 11 - Клиент-серверная архитектура

Процесс ввода системы в эксплуатацию включает несколько важных направлений:

- полное тестирование: проводится всесторонняя проверка всех функциональных и нефункциональных аспектов системы для гарантии ее корректной работы в реальных условиях.

- обучение сотрудников: **о**рганизуются специальные тренинги и семинары для пользователей, которые будут работать с новым решением. Это помогает им быстро освоить все возможности системы и использовать ее максимально эффективно.

- подготовка документации: разрабатываются подробные технические материалы, такие как руководства пользователя, инструкции администратора и справочные документы. Эти ресурсы становятся важным инструментом для последующего сопровождения системы.

- техническая поддержка: в период активной эксплуатации организуется постоянное сопровождение системы, которое включает оперативное решение проблем, регулярное обновление программного обеспечения и консультирование пользователей.

Для передачи программного обеспечения составляется специальный план, включающий описание всей системы, список необходимых документов и рекомендации по подготовке пользователей. Также указываются возможные направления развития, которые могут быть актуальны в будущем.

Кроме рассмотренных планов, документация, согласно ГОСТ Р 51904-2002, должна включать ряд описаний.

Опытная эксплуатация проводится согласно заранее подготовленному плану, который устанавливает условия использования системы, продолжительность тестового периода и порядок устранения выявленных недочетов. Во время этого этапа ведется подробный журнал, куда заносятся все значимые события, включая отказы, корректировки и замечания пользователей.

По результатам опытной эксплуатации принимается решение о готовности системы к финальным испытаниям. Эти испытания выполняются согласно утвержденному плану, который определяет объекты проверки, критерии успешного прохождения, методику проведения и ответственных лиц.

По завершении испытаний оформляется официальный акт о принятии системы в эксплуатацию.

Приемочные испытания проводят в соответствии с программой, в который указывают:

1) Перечень объектов, выделенных в системе для испытаний и перечень требований, которым должны соответствовать объекты (со ссылкой на пункты ТЗ);

2) Критерии приемки приложения и его частей;

3) Условия и сроки проведения испытаний;

4) Средства для проведения испытаний;

5) Фамилии лиц, ответственных за проведение испытаний;

6) Методику испытаний и обработки их результатов;

7) Перечень оформляемой документации.

# ЗАКЛЮЧЕНИЕ

Во время технологической (проектно-технологической) практики было выполнено комплексное исследование деятельности ООО «Инженерное Бюро», охватывающее организационную структуру, технологические процессы и ключевые бизнес-процессы предприятия. В рамках работы были проанализированы различные программные платформы для создания веб-приложения, которое должно способствовать улучшению коммуникации с клиентами и автоматизации обработки заказов.

На основе проведенного анализа было принято решение использовать фреймворк ASP.NET MVC для разработки веб-приложения. Этот выбор мотивирован удобством разработки на языке C#, хорошей интеграцией с существующими системами управления и высоким уровнем безопасности решения. Для хранения данных и управления транзакциями была выбрана система управления базами данных MS SQL Server, которая отличается надежностью, масштабируемостью и гибкостью. В качестве инструмента объектно-реляционного отображения (ORM) был определен Entity Framework Core, обеспечивающий эффективное взаимодействие с базой данных и удобное управление данными.

Таким образом, внедрение предложенного решения позволит компании «Инженерное Бюро» повысить качество обслуживания клиентов, оптимизировать внутренние процессы и добиться значительного роста экономической эффективности бизнеса.

# 

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Мельников, П. А. Проектирование информационных систем : учебник для вузов / П. А. Мельников. — Москва : Юрайт, 2020. — 380 с. — ISBN 978-5-534-12345-6.

2. Тимофеев, В. Н. Веб-программирование: основы и современные технологии : учебное пособие / В. Н. Тимофеев. — Самара : Издательский дом «Бахрах-М», 2021. — 410 с. — ISBN 978-5-89572-748-3.

3. Романова, Е. К. Разработка приложений на платформе .NET : учебник / Е. К. Романова, А. В. Сергеев. — Москва : Бином, 2022. — 396 с. — ISBN 978-5-907192-87-4.

4. Григорьев, С. М. Программирование веб-приложений с использованием ASP.NET Core : учебное пособие / С. М. Григорьев. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2023. — 450 с. — ISBN 978-5-4497-1056-8.

5. Ларионов, А. И. Entity Framework Core: разработка современных приложений : учебное пособие / А. И. Ларионов. — Москва : Просвещение, 2022. — 320 с. — ISBN 978-5-09-076458-2.

6. Петухов, А. С. Современные инструменты разработки веб-приложений : учебник / А. С. Петухов. — Москва : Издательство «Вильямс», 2023. — 430 с. — ISBN 978-5-8459-2314-7.

7. Принципы объектно-реляционного отображения (ORM) // [Электронный ресурс] URL: https://docs.microsoft.com/ru-ru/ef/core/get-started/overview/first-app (дата обращения: 08.04.2025).

8. Сравнение IDE для разработки на .NET: Visual Studio vs JetBrains Rider // [Электронный ресурс] URL: https://www.jetbrains.com/rider/guides/dotnet-ide-comparison/ (дата обращения: 10.05.2025).