МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

**Димитровградский инженерно-технологический институт –**

филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

**(ДИТИ НИЯУ МИФИ)**

**Специальность 09.02.07 Информационные системы и программирование**

**ОТЧЕТ**

**по производственной практике**

**(по профилю специальности)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование практики** | | ПП.01.01 Производственная практика | | |
| **Профессиональный модуль:** | | ПМ.01 Разработка модулей программного обеспечения для компьютерных систем | | |
| **студента** | | Волкова Алексея Викторовича | | |
| **группы** | | 331 | | |
| **Место проведения практики:** полное наименование организации/предприятия–базы практики | | Общество с ограниченной ответственностью "СимбирСофт" | | |
| Адрес местонахождения | | ул. Куйбышева, 226Г/1, Димитровград | | |
| **Дата прохождения практики:** | с «18» мая 2023 г. по «07» июня 2023 г | | | | |
| **Руководители практики:** | |  | | |
| От организации/ предприятия -базы практики | | | Власенко Олег Федосович |
| От техникума  ДИТИ НИЯУ МИФИ | | | Надеждина Алёна Валерьевна |
| **Итог практики (оценка)** | | |  | | | |
| **Подпись руководителя практики от организации** | | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  МП ИОФ | | | |
| **Подпись преподавателя** | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Надеждина | | |
| **Дата проверки** | |  | | |

Димитровград 2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc136169558)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 4](#_Toc136169559)

[1 ЗАДАНИЕ №1 4](#_Toc136169560)

[2 ЗАДАНИЕ №2 5](#_Toc136169561)

[3 ЗАДАНИЕ №3 6](#_Toc136169562)

[4 ЗАДАНИЕ №4 7](#_Toc136169563)

[5 ЗАДАНИЕ №5 8](#_Toc136169564)

[6 ЗАДАНИЕ №6 9](#_Toc136169565)

[7 ЗАДАНИЯ РАБОТОДАТЕЛЯ 10](#_Toc136169566)

[Задание 1 10](#_Toc136169567)

[ЗАКЛЮЧЕНИЕ 11](#_Toc136169568)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc136169569)

[ПРИЛОЖЕНИЯ 13](#_Toc136169570)

# **ВВЕДЕНИЕ**

Учебная практика по профессиональному модулю ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем направлена на формирование у обучающегося общих и профессиональных компетенций, приобретение первоначального опыта по данному виду деятельности для последующего применения их по специальности 09.02.07 Информационные системы и программирование. Учебная практика реализуется в рамках профессионального модуля ОП СПО, предусмотренного ФГОС СПО специальности.

С целью овладения видом профессиональной деятельности Информационные системы и программирование по специальности 09.02.07 Программирование в компьютерных системах, обучающийся в ходе освоения программы учебной практики должен приобрести практический опыт:

1. Участия в выработке требований к программному обеспечению;
2. Участия в проектировании программного обеспечения с использованием специализированных программных пакетов;

уметь:

1. Владеть основными методологиями процессов разработки программного обеспечения;
2. Использовать методы для получения кода с заданной функциональностью и степенью качества;

знать:

1. Модели процесса разработки программного обеспечения;
2. Основные принципы процесса разработки программного обеспечения;
3. Основные подходы к интегрированию программных модулей;
4. Основные методы и средства эффект явной разработки;
5. Основы верификации и аттестации программного обеспечения;
6. Концепции и реализации программных процессов;
7. Принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программ юго обеспечения;
8. Методы организации работы в коллективах разработчиков программного обеспечения;
9. Основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов;
10. Стандарты качества программного обеспечения;
11. Методы и средства разработки программной документации. Участие в интеграции программных модулей.

Основными задачами учебной практики по профилю специальности 09.02.07 Информационные системы и программирование (базовая подготовка) являются:

1. Обучение трудовым приемам, операциям и способам выполнения трудовых процессов, характерных для соответствующей профессии и необходимых для последующего освоения ими общих и профессиональных компетенций по избранной профессии.

Учебная практика направлена на обобщение и совершенствование знаний, полученных студентами в процессе обучения, знакомство студентов с передовыми технологиями и организацией процесса компьютерной обработки различных видов информации, приобретение умений самостоятельной работы по специальности.

# **Осваиваемые компетенции**

В процессе прохождения практики по профилю специальности студентами формируются следующие компетенции:

1) общие компетенции, включающие в себя способность:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой

для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения. (в ред. Приказа Минпросвещения России от 17.12.2020 N 747)

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках. (в ред. Приказа Минпросвещения России от 17.12.2020 N 747)

ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере. (в ред. Приказа Минпросвещения России от 17.12.2020 N 747)

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

# **1 ЗАДАНИЕ №1**

* 1. **Задачи по автоматизации поставленные или планируемые в подразделении**

На производстве ООО «СимбирСофт» поставлены следующие задачи:

* автоматизация учёта данных о персональных компьютерах, имеющихся на предприятии;
* автоматизация учёта данных о заказчиках;
* автоматизация учета переферии для каждого персонального копмьютера;
* учёт сотрудников.

Их задача — помогать компаниям из различных отраслей реализовывать амбициозные программные продукты, которые приводят к росту и развитию бизнеса.

Они производят программы для самых разных задач – от страхования до медицины, от приложения по доставке продуктов до автоматического освещения в темное время суток.

ООО «СимбирСофт» -- одна из направляющих компаний в России. Она создает самые современные решения для необходимого уровня жизны; люди, работающие там, понимают, что будущее за автоматизацией процессов.

* 1. **Состав автоматизированных систем, имеющихся на предприятии**

Современное информационное предприятие не может обойтись без использования автоматизированных систем, которые значительно упрощают и оптимизируют рабочие процессы. В зависимости от специфики предприятия, на нем могут быть установлены различные типы автоматизированных систем.

1. Системы управления производственными процессами. (Enterprise Resource Planning, ERP) Это системы, которые интегрируют все процессы в одну сеть, переводят эти данные в единую форму и дают возможность анализировать и оптимизировать весь цикл деятельности предприятия.
2. Системы управления проектами (Project Management Software, PMS). Они предназначены для управления процессом создания, разработки и выполнения проектов, включая управление временем, сроками, бюджетом, ресурсами и коммуникациями.
3. Системы управления клиентскими отношениями (Customer Relationship Management, CRM). Они позволяют устанавливать и поддерживать связь с клиентами, обрабатывать данные о клиентах и их интересах, а также определять потребности и предлагать продукты и услуги, соответствующие этим потребностям.
4. Системы управления контентом (Content Management System, CMS). Это системы, которые управляют контентом на сайте предприятия, включая создание, редактирование, публикацию и удаление контента.
5. Системы управления базами данных (Database Management System, DBMS). Они используются для управления большими объемами данных на предприятии, позволяют вводить, хранить, обрабатывать и извлекать информацию, а также обеспечивают безопасность данных.

Все эти системы взаимодействуют друг с другом и обеспечивают эффективность работы информационного предприятия.

* 1. **Эффективность использования имеющихся АС**

Использование автоматизированных систем на предприятии IT имеет значительные преимущества, обеспечивая более эффективную работу и повышая продуктивность работы сотрудников. Рассмотрим основные преимущества использования автоматизированных систем:

1. Ускорение процессов. Автоматизированные системы позволяют ускорить процессы работы, так как многие задачи выполняются автоматически, без участия человека. Это позволяет сократить время на выполнение процессов и увеличить масштабность работ.
2. Увеличение точности и качества выполнения задач. При использовании автоматизированных систем снижается вероятность ошибок, так как многие задачи выполняются автоматически, по определенным алгоритмам и правилам. Это повышает точность выполнения задач, а также качество продукции/услуг предприятия.
3. Сокращение затрат и оптимизация бюджета. Автоматизация процессов позволяет снизить затраты на оплату труда сотрудников, а также на сбор и анализ данных. Это позволяет предприятию сократить бюджет на выполнение задач, увеличивая при этом эффективность работы.
4. Повышение удобства и комфорта работы сотрудников. Благодаря автоматизированным системам производственные процессы становятся более удобными и комфортными для сотрудников, что в свою очередь повышает их работоспособность и эффективность труда.
5. Более быстрый доступ к информации. Автоматизированные системы предоставляют быстрый доступ к информации и упрощают ее обработку. Это позволяет получать необходимые данные быстро и легко, что повышает эффективность работы на предприятии.

В итоге, использование автоматизированных систем на предприятиях IT приводит к повышению производительности и эффективности работы, увеличению прибыли и оптимизации использования ресурсов предприятия.

# **ЗАДАНИЕ №2**

**2.1 Описание ПО, имеющегося на предприятии и его состав**

* IntelijIDEA, для разработки на язке Java, Kotlin, Android;
* Visual Studio, для разработки на языке C#, C++;
* GO, для разработки на языке GO и работы с базами данных SQL и NoSQL;
* Microsoft Word, для создания юристом документов;
* Microsoft Excel, для создания таблиц, что бы наглядно видеть данные;
* Sympfony, для написания web-сайтов, для PHP;
* Swift, для написания мобильных проложений на системе iOS;
* Angular, для разработки дизайна на языке Angular.
  1. **Характеристика операционных систем, оболочек, сервисных приложений, языков программирования, архиваторов, антивирусных средств, применяемых на предприятии**

Поскольку есть программные продукты, должна быть и операционная система, для отображения этих продуктов. Так, в основном, тут используют два типа операционных систем:

* Microsoft Windows – операционная система от компании Microsft;
* macOS – операционная система от компании Apple.

Из архиваторов можно отметить только один:

* WinRAR – программа для архивирования от компании winrar GmbH.

Из антивирусных программных продуктов можно отметить:

* DrWeb – разработанный российской компанией Dr.Web;
* Avast – чешский разработчик антивирусных программ.
  1. **Исследование эффективность использования ПО**

Исследование эффективности использования программного обеспечения (ПО) является важной задачей для любого предприятия, которое желает оптимизировать свою работу и повысить эффективность труда. Оно позволяет оценить степень эффективности использования ПО и узнать, не является ли это затратой средств и времени без нужной отдачи.

Для проведения исследования эффективности необходимо определить следующие параметры:

1. Целевые показатели. Предприятие должно определить конкретные цели и показатели, которые хочет достигнуть при использовании ПО. Например, увеличение объема продаж, сокращение времени на выполнение задач, повышение качества продукции.
2. Используемое ПО. Необходимо выбрать ПО, которое наилучшим образом соответствует целям и задачам предприятия.
3. Инструменты сбора данных. Для сбора данных по эффективности использования ПО могут использоваться различные методы, включающие опросы сотрудников, отслеживание времени выполнения задач и т.д.
4. Анализ данных. Полученные данные необходимо обработать и проанализировать, чтобы определить степень достижения целевых показателей и оценить эффективность использования ПО.

После анализа данных можно определить, насколько эффективно использование ПО влияет на бизнес-процессы предприятия, и принять меры для повышения эффективности его использования. Например, можно провести дополнительное обучение сотрудников использованию ПО, выбрать альтернативное ПО, сменить подход к его использованию или внедрить новые технологии или инструменты.

Таким образом, исследование эффективности использования ПО является важным инструментом для предприятий, которые стремятся повысить свою производительность и эффективность рабочих процессов.

* 1. **Предложения по эффективному использованию ПО**

Для эффективного использования программного обеспечения (ПО) необходимо принимать соответствующие меры, чтобы достичь максимальной эффективности работы. Ниже приведены предложения по улучшению использования ПО.

1. Обучение сотрудников. Обучение сотрудников использованию ПО может повысить эффективность его использования. Для этого нужно предоставить сотрудникам достаточное время на ознакомление с программным обеспечением, а также предложить им различные обучающие курсы.
2. Оптимизация рабочего процесса. Оптимизация рабочего процесса может повысить эффективность использования ПО. Для этого предприятие должно провести анализ текущих рабочих процессов и выявить узкие места, которые могут быть улучшены с помощью ПО.
3. Изменение подхода к работе с ПО. Иногда для повышения эффективности работы нужно изменить подход к работе с ПО. Например, использование функциональности программного обеспечения для автоматизации простых процессов может существенно сократить время на выполнение задач.
4. Проверка совместимости ПО. Проверка совместимости программного обеспечения с другими используемыми на предприятии программами может помочь снизить ошибки и неполадки системы.
5. Проверка наличия обновлений. Проверка наличия обновлений для программного обеспечения может помочь в исправлении ошибок и улучшении работы ПО.
6. Определение показателей. Определение показателей и мониторинг эффективности использования ПО поможет предприятию оценить, насколько эффективно используется ПО, и выявить проблемы, которые могут быть решены.
7. Выбор альтернатив. В случае недостаточно эффективного использования программного обеспечения решение может быть в замене текущего ПО на альтернативное, которое лучше соответствует потребностям предприятия.

В итоге следование этим предложениям поможет предприятиям достичь максимальной эффективности работы при использовании программного обеспечения.

* 1. **Описание взаимодействия между компонентами ПО**

Как именно нужно описать взаимодействия между компонентами ПО? Могу привести общие примеры взаимодействия:

1. Веб-сервер и клиентский браузер - клиентский браузер отправляет запрос на сервер, сервер обрабатывает запрос и отправляет ответ обратно на браузер.
2. База данных и приложение - приложение отправляет запрос к базе данных, база данных обрабатывает запрос и отправляет ответ обратно на приложение.
3. Модуль шифрования и модуль дешифрования - модуль шифрования получает данные, шифрует их и отправляет зашифрованные данные на модуль дешифрования. Модуль дешифрования получает зашифрованные данные, расшифровывает их и отправляет их на следующий компонент программы.
4. Клиентское приложение и сервер приложений - клиентское приложение обращается к серверу приложений для получения доступа к соответствующей структуре данных, сервер приложений обрабатывает запрос и отправляет ответ с запрошенной структурой данных обратно клиентскому приложению.
5. Интерфейс пользователя и логика приложения - интерфейс пользователя принимает вводимые пользователем данные и передает их на логику приложения, где данные обрабатываются, а результат выводится обратно на интерфейс пользователя.
6. Асинхронный Javascript и AJAX - асинхронный Javascript отвечает за отправку запросов на сервер, AJAX в свою очередь отвечает за получение данных от сервера и выводит ее на страницу.

Это примеры типичных взаимодействий, которые могут происходить между компонентами ПО.

# **ЗАДАНИЕ №3**

**3.1 Характеристика технических средств**

Simbirsoft — это IT-компания, которая имеет современные и мощные технические средства, необходимые для разработки качественного и инновационного программного обеспечения.

Вот некоторые характеристики технических средств Simbirsoft:

* 1. Компьютеры и лаптопы. Simbirsoft использует мощные компьютеры и лаптопы, которые имеют высокую производительность и позволяют разработчикам работать с большим количеством данных и задачами.
  2. Программное обеспечение. Simbirsoft использует современное и мощное программное обеспечение, которое помогает работникам компании разрабатывать высококачественное программное обеспечение. Некоторые из главных программ, которые используются в Simbirsoft, включают в себя Microsoft Visual Studio, JetBrains ReSharper Ultimate, Adobe Creative Suite и многие другие.
  3. Серверное оборудование. Simbirsoft использует высокопроизводительное и надежное серверное оборудование, которое обеспечивает стабильную работу приложений и веб-сайтов.
  4. Инфраструктура облачных вычислений. Simbirsoft использует облачную инфраструктуру для хранения и обработки данных клиентов. Это позволяет быстро масштабировать и адаптировать ресурсы к изменяющимся потребностям клиентов.
  5. Тестовое оборудование. Simbirsoft использует различное тестовое оборудование и инструменты, которые позволяют проводить тестирование и отладку программного обеспечения перед его запуском в продакшен-среде.
  6. Инструменты для контроля версий. Simbirsoft использует инструменты для контроля версий, такие как Git и SVN, для управления кодом и версиями приложений.

В целом, технические средства Simbirsoft обеспечивают высокую производительность, надежность и качество разрабатываемого программного обеспечения.

* 1. **Исследование эффективности технических средств**

Технические средства Simbirsoft позволяют компании создавать качественное программное обеспечение с высокой эффективностью. Вот несколько причин, почему технические средства Simbirsoft являются эффективными:

* 1. Производительность. Технические средства Simbirsoft имеют высокую производительность, что позволяет разработчикам выполнять задачи намного быстрее и эффективнее, чем при использовании менее мощных технических средств.
  2. Надежность. Simbirsoft использует надежное оборудование и программное обеспечение, которое гарантирует, что создаваемое программное обеспечение работает стабильно и без сбоев.
  3. Масштабируемость. Технические средства Simbirsoft позволяют масштабировать проекты в зависимости от их размера и сложности. Это позволяет компании быстро адаптироваться к изменяющимся потребностям клиентов и выполнять проекты в срок.
  4. Тестирование. Simbirsoft использует различное тестовое оборудование и инструменты для тестирования создаваемых программных продуктов. Это позволяет компании выявлять и исправлять ошибки еще до выпуска программного обеспечения в продакшен-среду.
  5. Безопасность. Simbirsoft обеспечивает высокую степень безопасности создаваемого программного обеспечения. Компания использует современные методы шифрования данных и защиты от взлома и злоумышленного использования данных.
  6. **Исследование ресурсов интернет, использующихся в работе предприятия и эффективности их использования**

SimbirSoft активно использует ресурсы в интернете, чтобы эффективно получать информацию о новых технологиях, трендах и лучших практиках. Кроме того, они также используют социальные сети и онлайн-сервисы для общения с клиентами и демонстрации своих проектов.

Вот некоторые из основных ресурсов Интернета, которые используются компанией SimbirSoft:

* 1. Онлайн-библиотеки и ресурсы. SimbirSoft использует онлайн-библиотеки, такие как IEEE Xplore и ACM Digital Library, для поиска научных исследований, идеи и решений. Благодаря этим ресурсам компания получает доступ к самой актуальной информации, которую можно применить в своих проектах.
  2. Ресурсы для управления проектами. SimbirSoft использует такие ресурсы для управления проектами, как Trello и JIRA, которые помогают отслеживать прогресс работ, идентифицировать проблемы и принимать необходимые меры для их решения. Компания использует эти инструменты для оптимизации процессов работы и увеличения эффективности проектов.
  3. Сервисы онлайн-общения. SimbirSoft использует такие сервисы, как Zoom и Skype, для общения, обмена информацией и координации работы между различными командами. Эти ресурсы ускоряют процесс принятия решений и сокращают время, необходимое для организации рабочих встреч и конференций.
  4. Учебные ресурсы. SimbirSoft использует онлайн-курсы и тренинги, например, Codecademy и Coursera, чтобы улучшить квалификацию своих сотрудников. Это позволяет работникам улучшить свои навыки и знания, а также получить внутреннюю компетентность, что позволяет увеличить эффективность проектов.
  5. Ресурсы для мониторинга изменений в мире технологий. SimbirSoft использует ресурсы, такие как TechCrunch и TechRadar, чтобы быть в курсе актуальных новостей и трендов в мире технологий. Эта информация позволяет оценить преимущества и потенциальные угрозы конкуренции.

Эффективное использование этих ресурсов позволяет SimbirSoft находить инновационные решения и ускорять реализацию проектов, что в целом повышает производительность компании и улучшает качество продуктов.

* 1. **Предложения по оптимизации использования интернет-ресурса**
  2. Увеличение количества использования онлайн-сервисов и облачных технологий, чтобы максимально сократить затраты на аппаратное обеспечение и ресурсы.
  3. Создание программы, которая будет автоматически мониторить интернет-ресурсы компании и предупреждать о проблемах, связанных с безопасностью и другими факторами.
  4. Обучение сотрудников компании по оптимальному использованию интернет-ресурсов и технологий.
  5. Оценка и использование лучших практик в отрасли и последующее внедрение этих практик в компании

# **ЗАДАНИЕ №4**

* 1. **Исследование процесса обеспечения достоверности и безопасности информационных потоков предприятия от несанкционированного доступа**

Несанкционированный доступ к информации может быть получен разными способами: как сам персонал может выдать информацию, так и можно получить доступ к операционным системам.

Несанкционированный доступ возможен в любой организации; неважно каких размеров она будет, неважно какой будет штаб сотрудников, неважно сколько компания будет приносить прыбыли. Внимательное отношение к защите со стороны руководства, создание особых подразделений, которые будут минимизировать потери и предотвращать попытки копирования, изменения или искажения данных.

ООО «СимбирСофт» работает со следующими направлениями:

* защита веб-приложений;
* защита рабочего пространства пользователей;
* обнаружение атак и предотвращение несанкционированного доступа к данным;
* защита от DDOS-атак;
* защита от вирусного заражения;
* защита почтового трафика;
* мониторинг информационнной безопасности и т.д.

Обеспечение достоверности и безопасности информации является важной задачей для предприятия Simbirsoft. Для этого компания использует следующие методы и инструменты:

* 1. Шифрование данных. Вся информация, которая передается по сети на предприятии Simbirsoft шифруется для защиты от несанкционированного доступа к данным.
  2. Авторизация и аутентификация пользователей. Для доступа к информации на предприятии Simbirsoft необходима авторизация и аутентификация. Только зарегистрированные пользователи, которые прошли процедуру аутентификации, могут получить доступ к защищенным данным.
  3. Контроль доступа к данным. Для того, чтобы защитить конфиденциальные данные, Simbirsoft использует системы контроля доступа. Каждый пользователь имеет доступ только к определенным данным, необходимым для выполнения своих рабочих задач.
  4. Мониторинг и аудит. Simbirsoft проводит системный мониторинг и аудит пользователя для выявления и ликвидации возможных уязвимостей в системах безопасности.
  5. Правила и политики. Simbirsoft использует строгие правила и политики для обеспечения безопасности данных. Например, официально запрещены использование ненадежных паролей и передача конфиденциальных данных по электронной почте.
  6. Обучение сотрудников. Simbirsoft обучает своих сотрудников, чтобы они были осведомлены о рисках безопасности и действиях, которые необходимо предпринимать для защиты данных.

В целом, Simbirsoft применяет все необходимые методы и инструменты для обеспечения безопасности и достоверности информации, которые передаются по потокам на предприятии. Это позволяет компании сохранять конфиденциальность информации, защищать данные и предотвращать утечки информации.

* 1. **Предложения по обеспечению защиты информации**
  2. Обновление программного обеспечения - регулярное обновление систем безопасности является одним из наиболее эффективных методов защиты информации.
  3. Регулярное обновление паролей - частая смена паролей может затруднить доступ к защищенным данным.
  4. Усиление политик безопасности - усиление правил использования информации может повысить уровень безопасности.
  5. Обучение сотрудников - регулярные тренировки и постоянное обучение сотрудников повысят осведомленность об информационной безопасности и уменьшат риски нарушения безопасности.
  6. Создание резервных копий данных - резервное копирование данных является важным способом защиты от потери и повреждения информации.

# **ЗАДАНИЕ №5**

* 1. **Исследование организации и процесса разработки программного обеспечения на предприятии**

Процесс организации и процесс разработки программного обеспечения на предприятии Sibirsoft может быть описан следующим образом:

* 1. Разработка технического задания (ТЗ) - клиенты контактируют с компанией для разработки определенного программного обеспечения, и Sibirsoft начинает работу, разрабатывая техническое задание на разработку ПО.
  2. Создание проектной группы - после того, как техническое задание будет утверждено, компания создает проектную группу, которая завершит проект, состоящую из проектного менеджера, программистов, дизайнеров и других специалистов.
  3. Разработка архитектуры - компания определяет архитектуру программного обеспечения и создает функциональную спецификацию, описывающую, как будет работать программное обеспечение.
  4. Разработка кода - команда разработчиков начинает работу над написанием кода, используя выбранные инструменты и технологии.
  5. Тестирование - после завершения написания кода, собирается версия ПО, которая затем проходит через ряд тестирования, включая тестирование безопасности и производительности.
  6. Деплоймент и поддержка - после успешного прохождения тестирования, проект передается клиенту и устанавливается на соответствующих системах. После этого компания предоставляет постоянную поддержку и техническое обслуживание продукта.

Важно отметить, что процессы организации и разработки ПО на предприятии Sibirsoft могут немного отличаться в зависимости от задачи и проекта, который требуется сделать. Однако общая концепция работы такая же: предоставить высококачественное программное обеспечение, с учетом потребностей и целей клиента.

* 1. **Порядок оформления программной документации**

Программная документация является неотъемлемым компонентом программного продукта и должна оформляться в соответствии с Единой системой программной документации. Программная документация, кроме формальных документов (спецификация, ведомость держателей подлинников, формуляр и др.), включает:

* техническое задание (назначение, область применения программы, требования, предъявляемые к программе);
* текст программы (запись программы с необходимыми комментариями);
* описание программы (сведения о логической структуре и функционировании программы);
* пояснительная записка (схема алгоритма, общее описание алгоритма и/или функционирования программы, обоснование принятых решений);
* эксплуатационные документы.
  1. **Порядок внедрения ПО или передачи его в эксплуатацию**

Процесс передачи программного обеспечения в эксплуатацию в компании Simbirsoft обычно включает в себя следующие шаги:

* 1. Тестирование ПО - перед тем, как продукт будет передан клиенту, он должен пройти через хорошо разработанный процесс тестирования, включая проверку на наличие ошибок и соответствие требованиям к продукту.
  2. Подготовка документации - компания разрабатывает необходимую документацию, которая будет включать в себя руководство пользователя, техническое описание и другую документацию на продукт.
  3. Подготовка окружения клиента - компания должна подготовить окружение, в котором продукт будет запущен. Это может включать в себя установку необходимых приложений и обеспечение требований к аппаратному обеспечению.
  4. Передача кода - после завершения проверки и подготовки окружения клиента, компания передает программное обеспечение.
  5. Обучение персонала - компания обучает персонал клиента, который будет использовать продукт, о его использовании и функционале.
  6. Послепродажная поддержка - Simbirsoft оказывает постпродажную поддержку продукта, предоставляя техническую помощь и дополнительные услуги, если это необходимо.

Каждый этап передачи ПО в эксплуатацию в компании Simbirsoft тщательно прорабатывается, чтобы обеспечить высокое качество продукта и во время передать его клиенту.

* 1. **Порядок сопровождения ПО**

Сопровождение программного продукты может включать изменение программного продукта после поставки, обеспечивающее его работоспособность в измененных или изменя-ющихся условиях, то есть модернизацию, позволяющую вводить изменения в эксплуатацион-ную среду программного средства.

Также в рамках сопровождения возможна модификация продукта после поставки в це-лях обнаружения и корректировки имеющихся скрытых ошибок для предотвращения явного проявления этих ошибок при эксплуатации данного продукта.

Совпровождение также может предусматривать модификацию программного продукта после поставки для повышенгия его рабочих характеристик, и тогда оно обеспечивает модер-низацию продукта в интересах пользователя, уточнение соответствубщих программных доку-ментов и его перепрограммирование для улучшения рабочих характеристик, сопровождаемо-сти и других атрибутов программного средства.

# **ЗАДАНИЕ №6**

* 1. **Описание инструментов программирования интерфейса**

Для программирования интерфейса программного продукта «сервис Constructum» использовалась интегрированная среда разработки Visual Studio Code Insiders версии 1.79.0. Интерфейс представлен на рисунке 6.1.1.

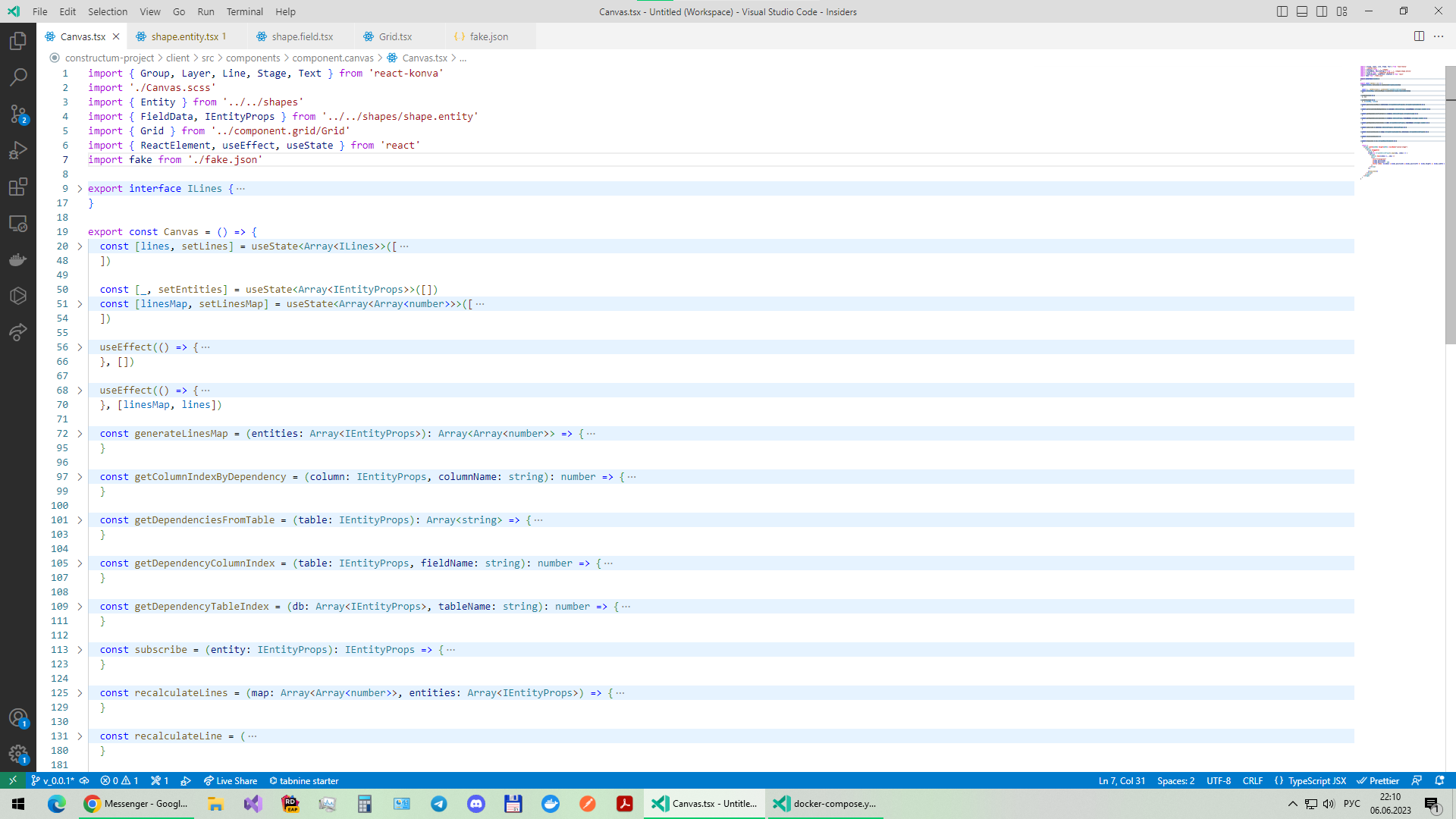


Рисунок 6.1.1 ­ Среда разработки

Для создания страниц сайта использовались следующие html теги:

* Тег <button> – создаёт на странице кнопку к которой можно назначить сценарий при нажатии на неё;
* Тег <input> – создаёт поле ввода на странице, однако есть и другие способы отображения элемента (checkbox, radiobutton, submit, button);
* Тег <form> – используется как контейнер для элементов, создающих форму. Элемент предоставляет функционал отправки данных на сервер указанным HTTP методом и обработчиком;
* Тег <script> – содержит в себе сценарии JavaScript которые будут выполняться на конкретной странице. Также возможен вариант использования в качестве подключения внешней библиотеки или скрипта.;
* Тег <header> – видимый семантический элемент шапки на странице, является блочным элементом;
* Тег <label> – видимый компонент страницы, устанавливающий связь между меткой и интерактивным элементом.
* Тег <img> – элемент, предназначенный для отображения изображений различных форматов;
* Тег <textarea> – элемент формы, предназначенный для ввода многострочного текста;
* Тег <select> – компонент предоставляющий функционал для возможности выбора конкретного пункта, содержащегося в нём.

Внешний вид приложения представлен на рисунке 6.1.2. На нём представлена страница с проектируемой в сервисе схемой базы данных.

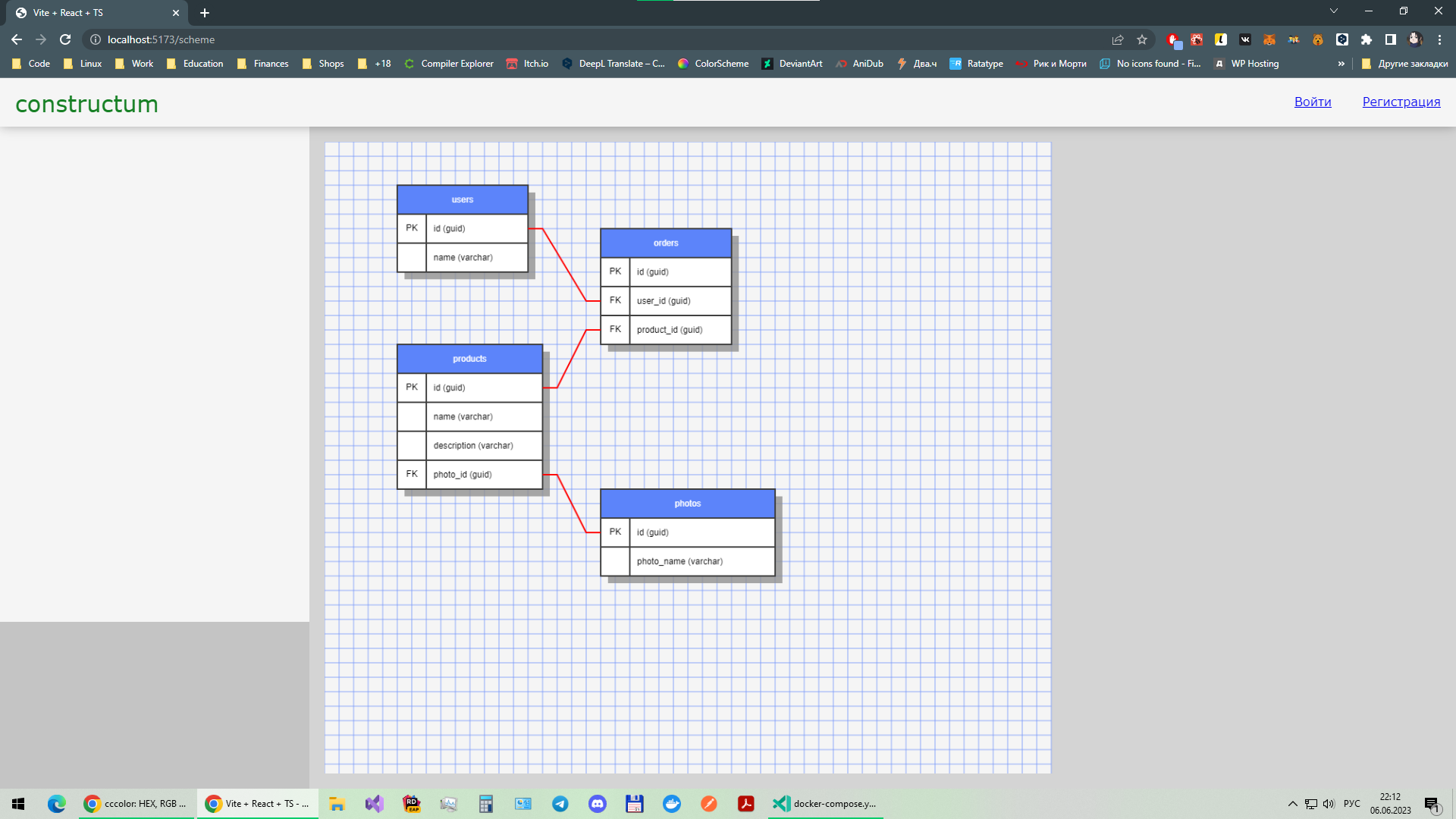


Рисунок 6.1.2 ­ Внешний вид приложения

* 1. **Технология создания и программирования структуры базы данных**

Базы данных NoSQL специально созданы для определенных моделей данных и обладают гибкими схемами, что позволяет разрабатывать современные приложения. Базы данных NoSQL получили широкое распространение в связи с простотой разработки, функциональностью и производительностью при любых масштабах. Ресурсы, представленные на этой странице, помогут разобраться с базами данных NoSQL и начать работу с ними.

Compass – это программный продукт для администрирования и разработки баз данных MongoDB. Compass позволяет выполнять задачи мониторинга, обслуживания, конфигурирования сервера MongoDB, а также создавать и выполнять запросы с документами.

Схема базы данных представлен на рисунке 6.2.1.

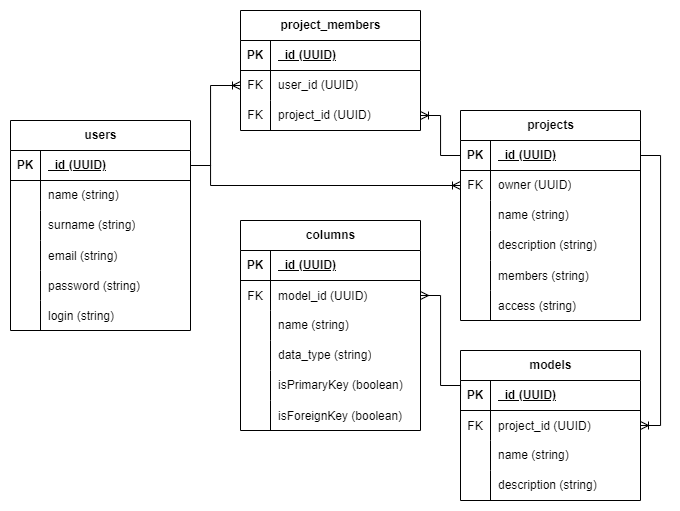


Рисунок 6.2.1 – Макет базы данных

Созданная база данных в СУБД Compass, представлена на рисунке 6.2.2.

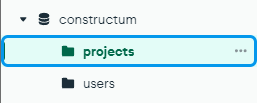


Рисунок 6.2.2 – Созданная база данных

Структура таблицы users представлена на рисунке 6.2.3.



Рисунок 6.2.3 ­ Структура таблицы users

Структура таблицы projects представлена на рисунке 6.2.4.

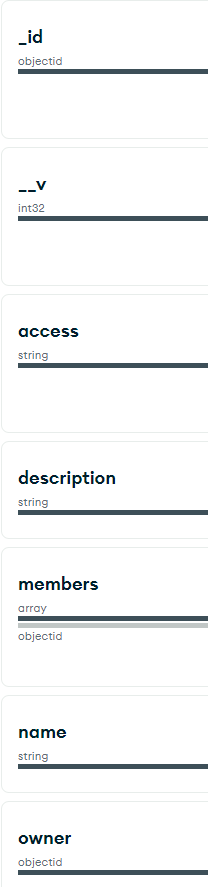


Рисунок 6.2.4 ­ Структура таблицы projects

* 1. **Технология создания интерфейса базы данных**

Для организации работы с базой данных необходимо разработать веб-интерфейс. Для этого воспользуемся интегрированной средой разработки Visual Studio Insiders.

Для получения данных из БД потребуется подключение к базе данных и соответствующий пакет для работы с MongoDB. Ниже представлен листинг для подключения к базе данных 6.3.1.

Листинг 6.3.1 ­ Подключение к базе данных

export const connect = async (): MongoConnection => {

try {

return await mongoose.connect(process.env.MONGO\_CONNECTION as string)

} catch (e) {

logger.error(e)

}

}

Для вывода информации с базы данных используется стандартнные html элементы содержащие в себе данные с модели.

* 1. **Технология создания запросов базы данных**

SQL-запросы - это наборы команд для работы с реляционными (табличными) базами данных.

По видам запросы SQL чаще всего делятся на:

* запросы, предназначенные для работы со структурой данных - для создания и модификации БД;
* запросы, используемые для непосредственно в работе с данными, с помощью которых можно добавлять, обновлять и удалять данные;
* запросы, применяемые для предоставления или отмены прав доступа к БД.

Следовательно каждый из SQL запросов можно разделить на типы:

* команды работающие со структурой БД. К ним можно отнести такие команды как CREATE - создать таблицу или пользователя, DROP - удаляет части БД;
* команды работающие с данными. К наиболее востребованным запросам относятся SELECT (выборка записей), UPDATE (обновление записей), DELETE (удаление записей), MERGE (слияние данных);
* команды работающие с правами доступа. В них входят GRANT - разрешение пользователю на проведение определённых операций с БД или данными. REVOKE - отзывает выданное разрешение пользователю. DENY - устанавливает запрет имеющего приоритет над разрешением.

Запрос реализующий функцию авторизации показан в листинге 6.4.1.

Листинг 6.4.1 ­ Запрос на авторизацию

const { login, password } = req.body

await mongoConnect()

await redisConnect()

User.findOne({ login: login })

.then(async (data) => {

const comparedPassword = await comparePassword(password.toString(), data?.password)

if (!comparedPassword) {

res.status(404).send('Пользователь не найден')

}

const { name, surname, email } = data as IUser

const payload = {

user: {

name: name,

surname: surname,

email: email,

},

}

await startSession(new Types.ObjectId(data?.\_id).toString(), JSON.stringify(payload))

await disconnect()

res.status(200).send(generateJwtSet(payload))

})

.catch((err) => {

console.error(err)

})

Запрос реализующий функцию регистрации показан в листинге 6.4.2.

Листинг 6.4.2 ­ Запрос на регистрацию

await user

.validate()

.then(async () => {

await user.save()

return res.sendStatus(200)

})

.catch((err) => {

return res.status(400).send(err)

})

# **7 ЗАДАНИЯ РАБОТОДАТЕЛЯ**

# **7.1 Упражнение 1**

**Постановка задачи:**

Найдите npm пакет animate.css и установите его в проект MyProject. Проверьте для данного пакета текущую версию и лицензию. Убедитесь, что лицензия позволяет использовать данный пакет.

**Выполнение:**

Был найден npm пакет animate.css и установлен в проект. Была проведена проверка данного пакета на текущую версию и лицензию. Лицензия данного пакета позволяет использовать его в обущающих целях. Страница с пакетом показана на рисунке 7.1

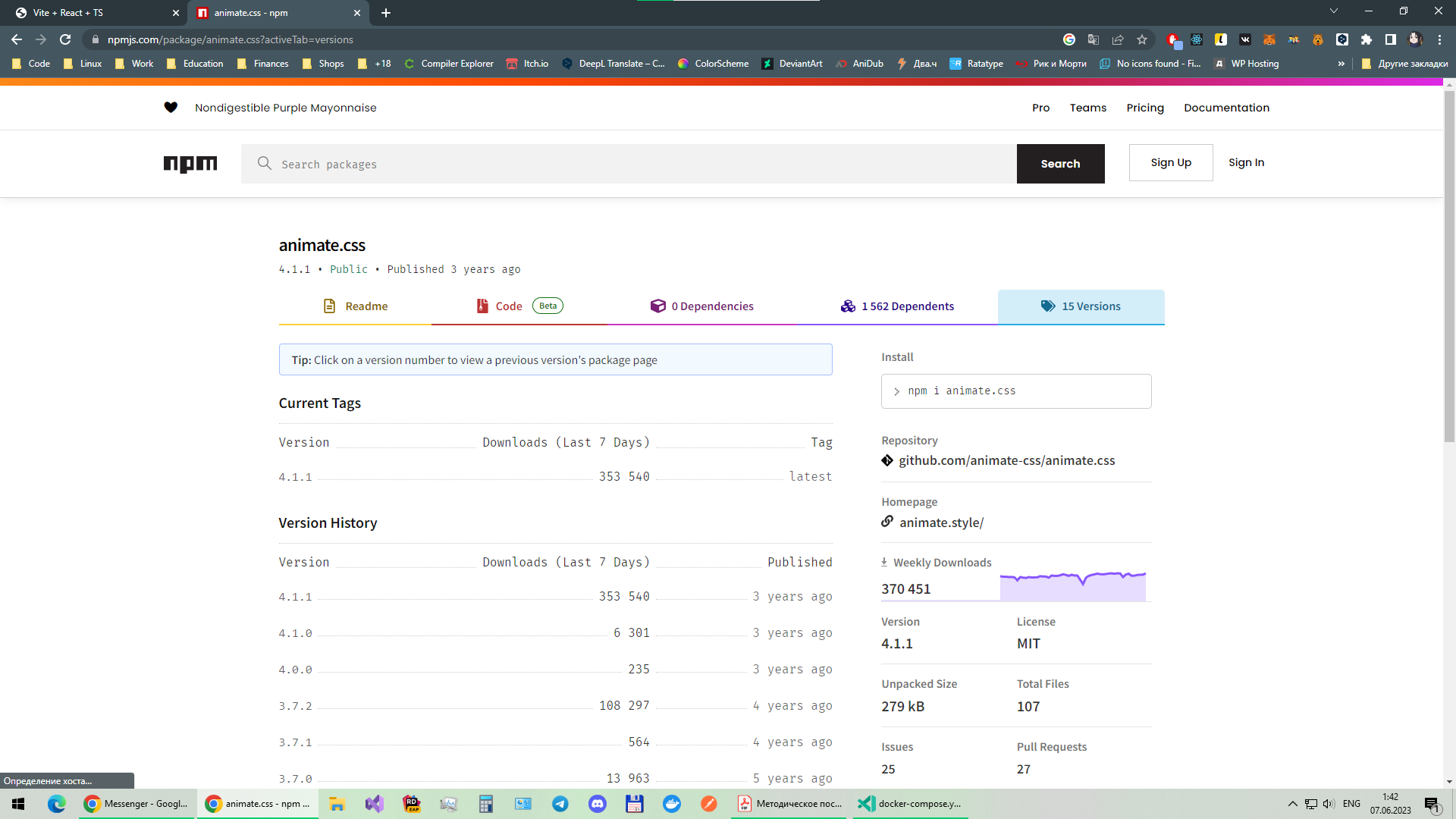


Рисунок 7.1 ­ Страница пакета animate.css

# **7.2 Упражнение 2**

**Постановка задачи:**

Установите пакет eslint@6.7.1 с указанием точной версии.

**Выполнение:**

Был установлен пакет eslint версии 6.7.1. Результат работы отображён на рисуне 7.2.



Рисунок 7.2 ­ Установленный пакет в зависимостях

# **7.3 Упражнение 3**

**Постановка задачи:**

Проверьте наличие уязвимостей всех пакетов, которые установлены в папке MyProject. Если уязвимости выявлены, то устраните их.

**Выполнение:**

С помощью команды для проверки уязвимостей в зависимостях пакетов, была проведена соответствующая проверка. Результат проверки на рисунке 7.3.

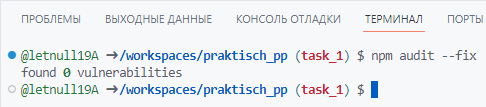


Рисунок 7.3 ­ Проверка на уязвимости зависимостей пакетов

# **7.4 Упражнение 4**

**Постановка задачи:**

Создайте простой Web-сервер на Node.js и express, который представляет собой две статичные страницы с произвольным наполнением.

**Выполнение:**

С помощью фреймворка express, был создан web-сервер с двумя страницами. Результат работы представлен на рисунке 7.4.

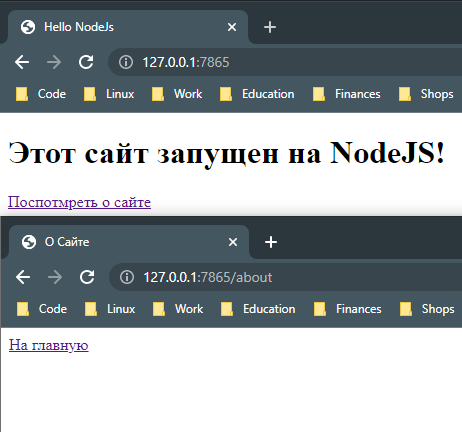


Рисунок 7.4 ­ Работа web-сервера

# **7.5 Упражнение 5**

**Постановка задачи:**

Установите редактор VSCode.

**Выполнение:**

# Был установлен редактор Visual Studio Code последней версии. Показано на рисунке 7.5.

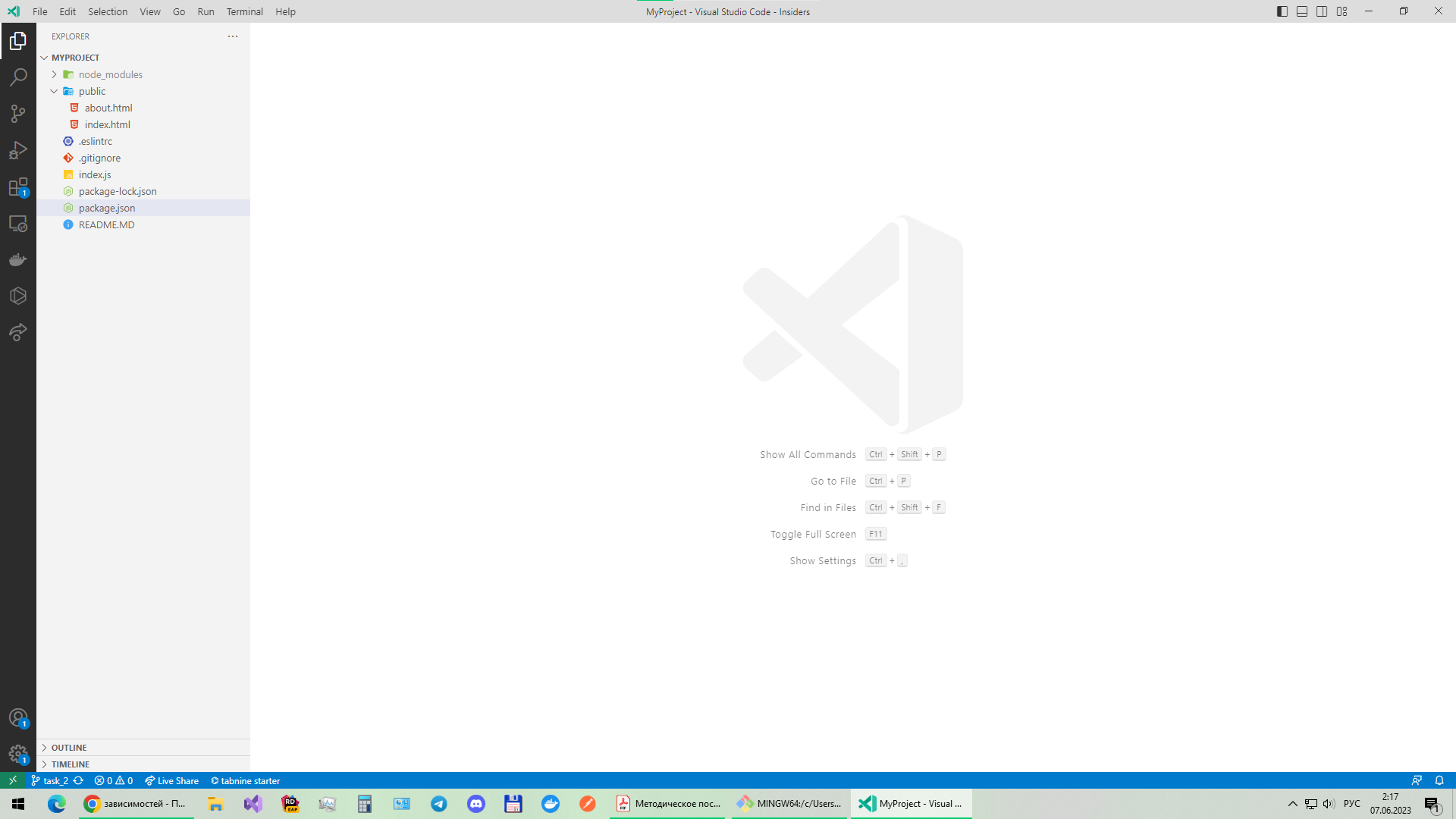


Рисунок 7.5 ­ Установленный VSCode

# **7.6 Упражнение 6**

**Постановка задачи:**

Установите пакет eslint@6.7.1 с указанием точной версии.

**Выполнение:**

Был установлен пакет eslint версии 6.7.1. Результат работы отображён на рисуне 7.6.



Рисунок 7.6 ­ Список установленных зависимостей

# **7.7 Упражнение 7**

**Постановка задачи:**

Требуется создать страницу сайта с содержимым: Статья с произвольными заголовком и содержимым, содержимое статьи обязательно должна содержать: Ссылки внутренние и внешние (по нажатию на которых открывается новая вкладка или окно), нумерованный список, таблицу с шапкой, телом и подвалом.

**Выполнение:**

Была выполнена вёрстка страницы в соответствии с заданием. Результат работы представлен на рисунке 7.7.

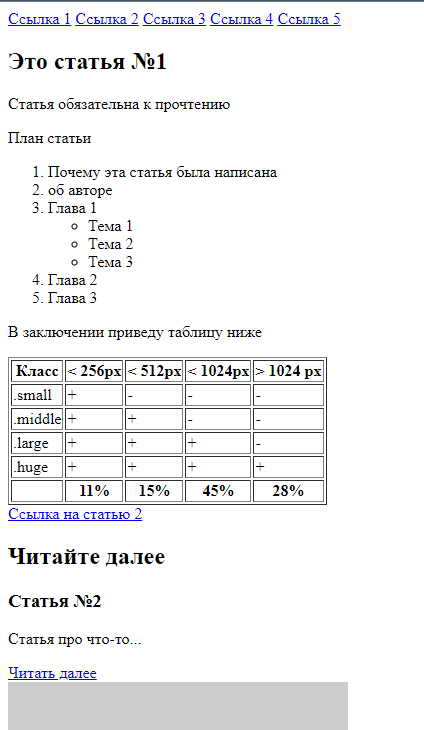


Рисунок 7.7 ­ Вёрстка сайта

# **7.8 Упражнение 8**

**Постановка задачи:**

Подключить к документу внешний файл со стилями. Применить стили к документу из прошлой главы: Высота шапки и подвала (<header> и <footer>) = 60px, все заголовки 1-го уровня имеют: размер шрифта 26px, цвет текста: #0f0f0f, все символы прописные (в верхнем регистре). Все ссылки: без нижнего подчеркивания, при наведении меняют цвет на красный (#ff0000). В похожих статьях: каждая статья, КРОМЕ ПОСЛЕДНЕЙ, имеет нижний отступ = 16px, каждая статья имеет серый фон: #cecece.

**Выполнение:**

Была выполнена вёрстка в соответствии с критериями. Результат работы представлен на рисунке 7.8.

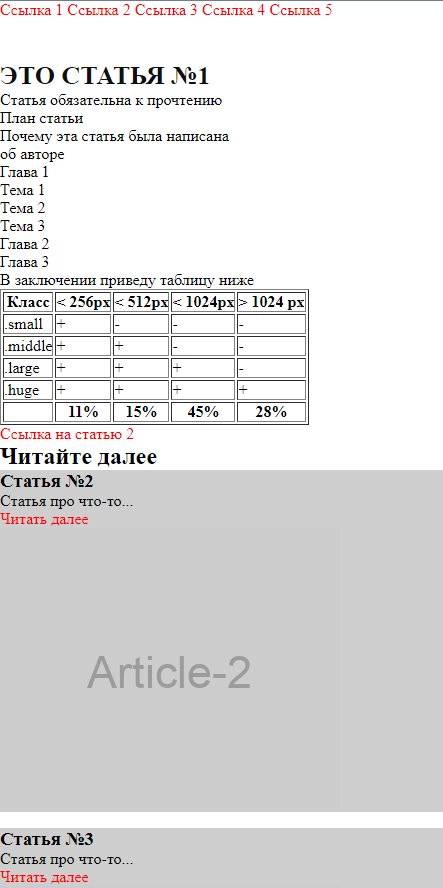


Рисунок 7.8 ­ Вёрстка страницы

# **7.9 Упражнение 9**

**Постановка задачи:**

Давайте добавим нашему сайту боковую, левую, колонку, в которую поместим виджет “Новости компании”.

Основные требования: разумеется, используем Flex, обязательно соблюдать семантически верный синтаксис (для обозначения, косвенно связанной с основным содержимым, боковой колонки, заголовка виджета и списка статей в исходном коде боковая колонка идет после основного контента (тут учитывается и семантика, и СЕО) список новостей направлен вертикально, сверху вниз.

**Выполнение:**

Была выполнена вёрстка в соответствии с критериями. Вёрстка производилась с использованием Flexbox. В Вёрстке использовались семантические теги для конкретных блоков сайта. Списку новостей было установлено вертикальное направление. Результат работы представлен на рисунке 7.9.

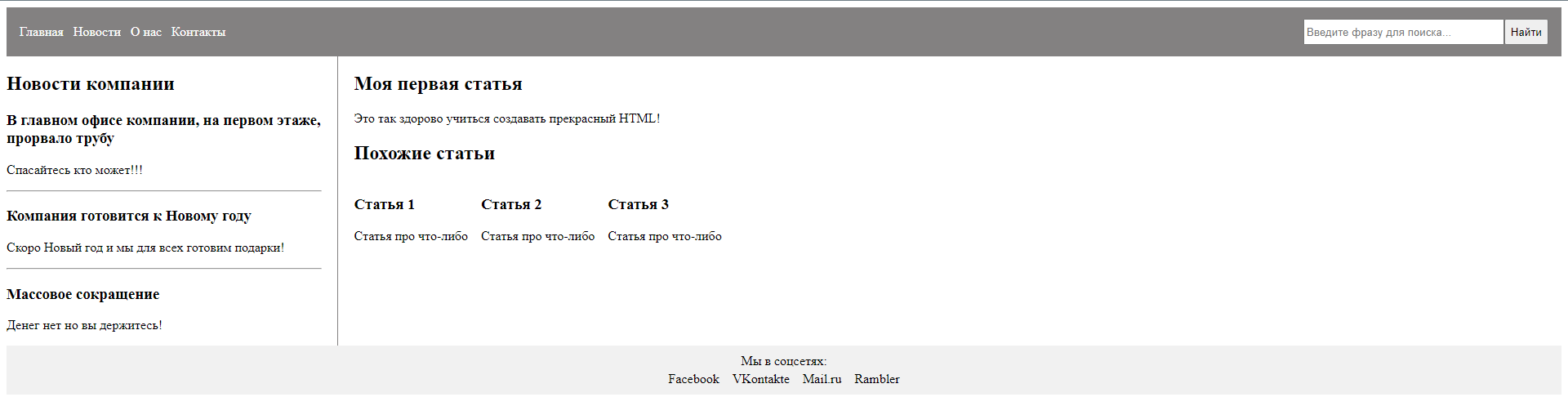


Рисунок 7.9 ­ Вёрстка сайта с использованием Flex

# **7.10 Упражнение 10**

**Постановка задачи:**

Давайте переделаем шаблон страницы нашего сайта на grid: шапку, боковую колонку и подвал. Подвал обязательно должен быть “прибит” к низу экрана, независимо от количества контента в боковой колонке и основной колонке (области контента). Также переделаем на grid список статей в “Похожих статьях”. Элементы списка должны выводиться по 3 статьи в ряд.

**Выполнение:**

Была перевыполнена вёрстка сайта на CSS Grid. Подвал сайта прикреплён к самому низу экранного пространства документа. Список статей тоже был переделан с использованием CSS Grid. Результат работы представлен на рисунке 7.10.

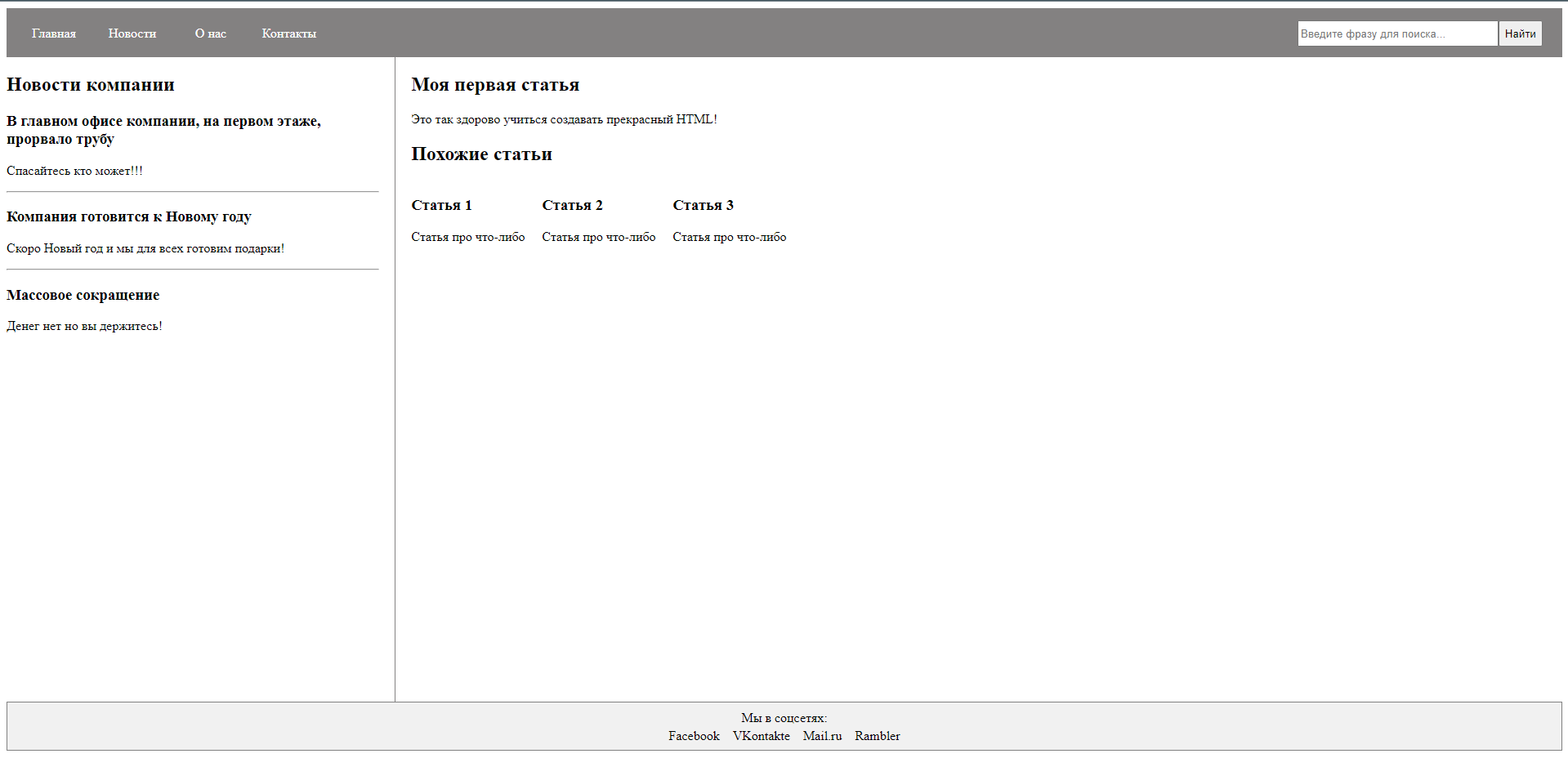


Рисунок 7.10 ­ Вёрстка сайта с использованием Grid

# **ЗАКЛЮЧЕНИЕ**

В ходе производственная практике по профессиональному модулю ПМ.01 Разработка модулей программного обеспечения для компьютерных систем направленой на формирование у обучающегося общих и профессиональных компетенций, приобретение первоначального опыта по данному виду деятельности для последующего применения их по специальности 09.02.07 Информационные системы и программирование.

Были выполнены задания от техникума, описав моменты в ходе выполнения. Прошли производственную практику в одной из ведущих компаний ООО «СимбирСофт» в г. Димитровграде с 18 мая по 7 июня 2023г.

Ещё практика предоставила возможность узнать больше о структуре предприятия, его целях и задачах.

Выполнены задания от руководителя производственой практики предприятия:

1. Изучены Git и GitHub;
2. Были получены начальные знания по вёрстке сайтов HTML
3. Изучена работа с NPM и установкой пакетов
4. Изучены основы CSS3
5. Была произведена вёрстка на FlexBox и Grid
6. Было содано приложение на фреймфорке ReactJS
7. Было разработано серверное прилежение на NodeJS с использованием ExpressJS

В ходе всей производственной практики было изучено, повторено и применено большое количество знаний и опыта в сфере программирования серверных приложений и клиентских одностраничных сайтов, а так же были получены очень полезные и ценные рекомендации и советы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 19.102-77. Единая система программной документации. стадии разработки = Unified system for program documentation. Development stages: межгосударственный стандарт: издание официальное: утв. и введ. в действие Постановлением Государсвенного комитета стандартов Совета Министров СССР от 20 мая 1977 г. №1268: введ. впервые: дата введ. 01.01.1980. – Москва: Стандартинформ, 2010 – 4 с.
2. ГОСТ 19.505-79. Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению = Unified system for program documentation. Oper-ation's guide. Requirements for contents and form of presentation : межгосударственный стандарт : издание официальное : утв. и введ. в действие Постановлением Государственного комитета CCCР по стандартам от 12 января 1979 г. N 74 : введ. впервые : дата введ. 01.01.1980. – Москва: Стандартинформ, 2010 – 2 с.
3. ГОСТ 19.105-78. Единая система программной документации. Общие требования к программным документам = Unified system for program documentation. General requirements for program documents : межгосударственный стандарт: издание официальное : утв. и введ. в дей-ствие Постановлением Государственного комитета CCCР по стандартам от 18 декабря 1978 г. N 3350 : введ. впервые : дата введ. 01.01.1980. – Москва: Стандартинформ, 2010 – 3 с.
4. ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом = Unified system for program documentation. Requirements for printed documents : межгосударственный стандарт : издание официальное : утв. и введ. в действие Постановлением Государственного комитета CCCР по стандартам от 18 декабря 1978 г. N 3350 : введ. впервые : дата введ. 01.01.1980. – Москва: Стан-дартинформ, 2010 – 12 с.
5. Amazon <https://aws.amazon.com/ru/nosql/> (дата обновления 2023 год)
6. SimbirSoft <https://www.simbirsoft.com/> (дата обновления 2023 год)
7. HTML Book <http://htmlbook.ru/> (дата обновления 2023 год)
8. MDN https://developer.mozilla.org/ (дата обновления 2023 год)
9. React Docs <https://react.dev/> (дата обновления 2023 год)
10. TypeScript Docs <https://www.typescriptlang.org/> (дата обновления 2023 год)
11. SCSS Docs <https://sass-scss.ru/> (дата обновления 2023 год)
12. Node Package Manager <https://www.npmjs.com/> (дата обновления 2023 год)
13. NodeJS официальный сайт <https://nodejs.org/en> (дата обновления 2023 год)
14. MongoDB официальный сайт <https://www.mongodb.com/> (дата обновления 2023 год)
15. Docker официальный сайт https://www.docker.com/ (дата обновления 2023 год)