Федеральное агентство по рыболовству   
ФГБОУ ВО «Калининградский государственный университет» **Калининградский морской рыбопромышленный колледж**Отчет по производственной практике   
(по профилю специальности)

Специальность 09.02.04 «**Информационные системы (по отраслям)»**

Разработал студент группы 17-ИС-14

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.И. Папков

Подпись

Руководитель:  
\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Чухарев

Подпись

г. Калининград 2020.

**Содержание**

[**Раздел 1. Предприятие – база прохождения практики** 3](#_Toc63809469)

[**Тема 1.1 Общие сведения о предприятии организации) и отделе – месте прохождения практики по профилю специальности.** 3](#_Toc63809470)

[**Тема 1.2 Виды обеспечения атоматизированных систем предприятия (организации)** 8](#_Toc63809471)

[**ПМ.01 Эксплуатация и модификация информационных систем** 10](#_Toc63809472)

[**МДК.01.01.Т1 Эксплуатация и модификация информационных систем** 10](#_Toc63809473)

[**Разработка плана по обеспечению надежности системы** 10](#_Toc63809474)

[**Инструменты, обеспечивающие надежность программных продуктов, план обеспечения надежности** 11](#_Toc63809475)

[**МДК 01.01.Т2 Безопасность и управление доступом в информционных систем** 12](#_Toc63809476)

[**Технологическая модель подсистемы информационной безопасности** 12](#_Toc63809477)

[**Защита Windows 10 встроенными средствами** 15](#_Toc63809478)

[**ПМ.03 Разработка интернет информационных систем** 18](#_Toc63809479)

[**МДК.03.01 Технологии и средства разработки Web-приложений** 18](#_Toc63809480)

[**HTML и CSS. Задание 1** 18](#_Toc63809481)

[**HTML и CSS. Задание 2** 20](#_Toc63809482)

[**Основы программирования на JavaScript. Задание 1** 22](#_Toc63809483)

[**Основы программирования на JavaScript. Задание 2** 25](#_Toc63809484)

[**Введение в программирование на PHP5. Задание 1** 27](#_Toc63809485)

[**Введение в программирование на PHP5. Задание 2** 29](#_Toc63809486)

[**Создание макета мобильной версии сайта** 31](#_Toc63809487)

[**Сверстать сайт по макету** 33](#_Toc63809488)

[**МДК.03.02 Разработка интернет информационных систем** 34](#_Toc63809489)

[**Написать сайт по теме КП с помощью конструктора сайта** 34](#_Toc63809490)

[**Разместить сайт на хостинге** 38](#_Toc63809491)

[**Использованная литература** 43](#_Toc63809492)

# **Раздел 1. Предприятие – база прохождения практики**

## **Тема 1.1 Общие сведения о предприятии организации) и отделе – месте прохождения практики по профилю специальности.**

**Рис.1.** Оргштатная структура СТЛ.

В фирме, в которой я прохожу практику, есть три отдела:

* отдел разработки
* отдел маркетинга и аналитики
* отдел внутреннего устройства

**Отдел разработки подразделяется на три подгруппы:**

1. Группа Backend разработки;

Backend-разработка — это набор аппаратно-программных средств, при помощи которых реализуется логика работы сайта. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

Например, когда вы вводите запрос на странице поисковика и жмете клавишу Enter, frontend заканчивается и начинается backend. Ваш запрос отправляется на сервер Google или Яндекса, где расположены алгоритмы поиска. Именно там случается все «волшебство». Как только на мониторе появилась информация, которую вы искали, — вновь происходит возвращение в зону frontend .

По большому счету, сервер — это тот же компьютер, только более мощный. Он хранит данные и отвечает на запросы пользователей.

Backend — это процесс объединения сервера с пользователем.

Backend-разработчик может применять любые инструменты, доступные на его сервере. Он вправе выбрать любой из универсальных языков программирования, например, Ruby, PHP, Python, Java.

Также для backend-разработки используются разные системы управления базами данных:

* MySQL;
* PostgreSQL;
* SQLite;
* MongoDB.

В зависимости от вида продукта в обязанности backend-разработчика могут сильно меняться. Такой специалист может создавать и интегрировать базы данных, обеспечивать безопасность, создавать технологии резервного копирования и восстановления.

1. Группа Frontend разработки и дизайна;

Frontend — это разработка пользовательского интерфейса и функциональности, которые работают на клиентской стороне веб-сайта или приложения. К этому виду разработки можно отнести все, что видит пользователь, открывая web-страницу. Frontend-разработчик сотрудничает с дизайнерами, программистами и UX-аналитиками, чтобы создавать удобный и востребованный продукт.

Чтобы наглядно понять, что такое frontend-разработка, откройте страницу любого сайта — перед собой вы увидите интерфейс. Щелкнув правой кнопкой мыши, можно посмотреть код этой страницы в браузере.

**К frontend-разработке относятся:**

* **HTML (HyperText Markup Language)** — язык разметки документов, при помощи которого формируется структура страницы: заголовки, абзацы, списки и так далее;
* **CSS (Cascading Style Sheets)** — язык для описания и стилизации внешнего вида документа. Благодаря CSS-коду ваш браузер понимает, как именно отображать элементы. CSS задает цвета и параметры шрифтов, определяет, как будут располагаться разные блоки сайта, и так далее. Еще он позволяет выводить один и тот же документ в разных стилях, например, для печати (обычной или шрифтом Брайля), вывода передачи на экран или чтения голосом;
* **JavaScript** — это язык, который создавался для того, чтобы оживить веб-страницы. Его задача — реагировать на действия пользователя, обрабатывать клики мышкой, перемещения курсора, нажатия клавиш. Еще он посылает запросы на сервер и загружает данные без перезагрузки страницы, позволяет вводить сообщения и многое другое.

1. Группа тестирования, сопровождения и поддержки.

Любой продукт, будь то сайт или мобильное приложение, проходит проверку на качество. В ИТ-сфере за этим следит тестировщик (Test Engineer).

В переводе с английского глагол «test» означает «проверять». Поэтому тестировать, проверять программное обеспечение с целью обнаружения ошибок (багов) в работе ПО и их исправления – вот главная задача такого специалиста.

Он имитирует ситуации, в которых могут оказаться конечные потребители, и выступает в роли эксперта по выявлению проблем в работе приложения, сайта, игры или другого программного продукта заказчика. Тестировщик удостоверяется в том, что ПО является надежным и с пользовательской, и с технической стороны.

**Обязанности тестировщика:**

* Контроль и поиск проблем. В первую очередь, специалист должен контролировать качество разрабатываемых продуктов и анализировать ошибки, которые могут возникнуть у конечных потребителей при их использовании.
* Тестирование. Следующий этап – это разработка тестовых наборов и их регулярный прогон, подготовка тестовых данных, написание методики тестирования.
* Анализ. Данные, полученные в процессе проверок, анализируются. Обнаруженные недочеты классифицируются и заносятся в базу
* Саппорт. Тестировщик не устраняет найденные проблемы и недочеты. Он регулирует и поддерживает процесс их ликвидации – находит недочеты и сообщает о них тем специалистам, которые занимаются их исправлением (например, разработчикам), а также дополняет необходимой информацией о дефекте, если такая требуется
* Документирование дефектов. Чтобы провести тестирование, зафиксировать его результаты, тестировщик должен корректно внести информацию в техническую документацию. Рекомендуется проверять документы на предмет полноты и актуальности данных.

**Отдел PR, маркетинга и аналитики подразделяется на две группы:**

1. Группа PR и маркетинга;

Интернет-маркетинг является составляющей электронной коммерции. Его также называют online-маркетингом. Он может включать такие части, как интернет-интеграция, информационный менеджмент, PR, служба работы с покупателями и продажи.

Электронная коммерция и интернет-маркетинг стали популярными с расширением доступа к интернету и являют собой неотъемлемую часть любой нормальной маркетинговой кампании. Сегмент интернет-маркетинга и рекламы растёт как в потребительском секторе, о чём свидетельствует появление с каждым днем все новых [интернет-магазинов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD), так и на рынке [B2B](https://ru.wikipedia.org/wiki/B2B).

Основными преимуществами интернет-маркетинга считаются интерактивность, возможность максимально точного [таргетинга](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D1%80%D0%B3%D0%B5%D1%82%D0%B8%D0%BD%D0%B3), возможность постклик-анализа, который ведет к максимальному повышению таких показателей как [конверсия сайта](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F_%D1%81%D0%B0%D0%B9%D1%82%D0%B0) и ROI интернет-рекламы.

1. Группа бизнес-аналитики.

Работа бизнес-аналитика в IT-компании напрямую связана со словом «требования». Аналитик выявляет их, документирует, управляет ими. Есть еще множество действий, которые можно проделать с требованиями и о которых знает только бизнес-аналитик.

Бизнес-аналитик понимает, что именно хочет заказчик и какой продукт ему нужен в результате на основе разных требований: начиная с верхнеуровневых «чего хочет бизнес» и «чего хотят клиенты этого бизнеса» и заканчивая детальным «как это все будет реализовано в системе».

**Отдел общих сведений подразделяется на три группы:**

1. Юридическая группа;
2. Отдел кадров;

**Отдел кадров** - это структура в организации, которая занимается управлением персоналом.

Отдел кадров является не только функциональной единицей, это еще и лицом компании, так как именно в отделе кадров любой соискатель начинает знакомиться с организацией.

**Цель отдела кадров**

Цель отдела кадров - это способствование достижению целей предприятия (организации) путем обеспечения предприятия необходимыми кадрами и эффективного использования потенциала работников.

Подбор работников ведется с помощью специально разработанных стратегий: подачи информации о вакансиях в СМИ и службы занятости населения, применения методик отбора, тестирования, процедур адаптации специалистов и последующего повышения квалификации.

**Задачи отдела кадров**

Основная задача отдела кадров - правильно учитывать работу сотрудников, определять количество рабочих, выходных и больничных дней для расчета зарплаты, отпусков и подачи сведений в бухгалтерию организации.

Также основными задачами отдела кадров являются:

* организация отбора, набора и найма персонала необходимой квалификации и в требуемом объеме. Подбор работников ведется с помощью специально разработанных стратегий: от подачи информации о вакансиях в СМИ и службы занятости населения до применения методик отбора, тестирования, процедур адаптации специалистов и последующего повышения квалификации;
* создание эффективной системы штатных сотрудников;
* разработка карьерных планов сотрудников;
* разработка кадровых технологий.

Кроме того, отдел кадров должен подавать сведения на сотрудников в Пенсионный фонд РФ, страховые компании, Налоговую и Миграционную службы.

**Взаимодействие с бухгалтерией организации**

Отдел кадров взаимодействует с бухгалтерией организации для решения вопросов, связанных с оплатой труда.

Так в бухгалтерию организации отдел кадров представляет документы и копии приказов об увольнении, зачислении на работу, о командировках, отпусках, поощрениях либо штрафных санкциях для сотрудников.

**Взаимодействие с юридическим отделом**

Юридический отдел снабжает сотрудников отдела кадров информацией о последних изменениях в действующем законодательстве, предоставляет всестороннюю правовую поддержку.

1. Бухгалтерия.

**Бухгалтерия** —  это штатно - структурное подразделение организации, предназначенное для сбора данных об имуществе и обязательствах предприятия.

**Основные задачи, стоящие перед бухгалтерской службой организации**

Основные задачи, стоящие перед бухгалтерской службой организации заключаются в следующем:

* формирование полной и достоверной информации о деятельности организации и ее имущественном положении, необходимой внутренним пользователям бухгалтерской отчетности — руководителям, учредителям и собственникам имущества организации, а также внешним — инвесторам, кредиторам и другим пользователям бухгалтерской отчетности;
* обеспечение информацией, необходимой внутренним и внешним пользователям бухгалтерской отчетности для контроля за соблюдением законодательства РФ при осуществлении организацией хозяйственных операций и их целесообразностью, наличием и движением имущества и обязательств, использованием материальных, трудовых и финансовых ресурсов в соответствии с утвержденными нормами, нормативами и сметами;
* предотвращение отрицательных результатов хозяйственной деятельности организации и выявление внутрихозяйственных резервов обеспечения ее финансовой устойчивости.

## **Тема 1.2 Виды обеспечения атоматизированных систем предприятия (организации)**

Автоматизироованные информационные системы применяются во всех отделах предприятия: от Frontend отдела до Бухгалтерии.

Для архивации, резервного копирования информации используются такие интернет сервисы, как **GitHub** и **Amazon Web Services (AWS).** Так же **AWS** используется для работы с СУБД **Amazon DynamoDB**.

**Эффективность использования сервисов Internet предприятием:**

**Amazon Web Services (AWS)** – это самая распространенная в мире облачная платформа с широчайшими возможностями, предоставляющая более 175 полнофункциональных сервисов для центров обработки данных по всей планете. Миллионы клиентов, в том числе стартапы, ставшие лидерами по скорости роста, крупнейшие корпорации и передовые правительственные учреждения, используют AWS для снижения затрат, повышения гибкости и ускоренного внедрения инноваций.

**AWS** предоставляет несравнимо больше сервисов и их функций, чем любой другой поставщик облачных услуг: от инфраструктурных технологий, таких как инструменты для **вычисления**, **хранилища** и **базы данных**, до инноваций, например, машинного обучения и искусственного интеллекта, озер данных и аналитики, а также Интернета вещей. С ними клиент сможет быстрее, легче и дешевле перенести текущие приложения в облако и реализовывать в нем любые возможные проекты.

**AWS** также предоставляет самые широкие функциональные возможности для своих сервисов. Например, **AWS** предлагает на выбор много **баз данных**, специально созданных для различных типов приложений, чтобы клиент мог подобрать правильный инструмент для эффективной и экономной работы.

**AWS** самая гибкая и защищенная среда для **облачных вычислений** из существующих. Базовая инфраструктура спроектирована так, чтобы удовлетворить требованиям к безопасности международных банков, учреждений в сфере обороны и других организаций с высокими требованиями к защите данных. Поддержка осуществляется за счет разнообразных инструментов для обеспечения безопасности в облаке, которые включают 230 сервисов и возможностей по обеспечению безопасности, соответствия требованиям и управлению. AWS поддерживает 90 стандартов безопасности и сертификаций на соответствие требованиям, и все 117 сервисов AWS для хранения клиентских данных предлагают возможность их шифрования.

**GitHub** — это платформа, хранящая различные **Git-репозитории** на своих многочисленных серверах. Также GitHub называют **крупнейшим веб-сервисом** для хостинга и совместной разработки IT-проектов. Гитхаб основан на системе контроля версий Git и разработан компанией GitHub на Ruby on Rails. Он бесплатен для тех проектов, которые имеют открытый исходный код. Для крупных корпоративных клиентов доступны платные тарифные планы.

Если вы являетесь пользователем GitHub, вы можете без проблем хранить удалённые репозитории на данных серверах и одновременно с этим вносить свой вклад в репозитории open-source. По сути GitHub дополняет использование Git, плюс открывает некоторые **новые возможности**.

К примеру, появляется возможность сделать **форк удалённого репозитория**, создав свою копию репозитория на GitHub-сервере. Это может быть полезным, если у вас отсутствуют права на создание ветви в оригинальном репозитории. После того, как вы используете команду **git clone**, ваш локальный репозиторий сможет отслеживать оригинальный репозиторий как **upstream**, а удалённый форк как **origin**.

После этого может возникнуть необходимость слить тематическую ветвь удалённого репозитория в основную ветвь оригинального. Для этого создаётся новый запрос на внесение изменений (Pull Request), причём **GitHub** проверит наличие конфликтов перед выполнением слияния. Также в запросе можно обсуждать код, а все коммиты, отправляемые вами в удалённую ветвь, автоматически добавятся в запрос.

Локальных вычислительных систем на предприятии не имеется, так как всю нужную информации сотрудники могут получить друг от друга при помощи выше указанных облачных сервисов. В Backend отделе сотрудники используют на своих компьютерах ПО **Linux**, в остальных отделах **Windows 10**.

# **ПМ.01 Эксплуатация и модификация информационных систем**

## **МДК.01.01.Т1 Эксплуатация и модификация информационных систем**

### **Задание 1. Разработка плана по обеспечению надежности системы**

Общая структура данного документа обычно включает три раздела:

* Общие положения;
* Перечень мероприятий по обеспечению надёжности;
* Порядок контроля реализации и корректировки программы обеспечения надёжности.

В Общихположениях отражается:

* основание для разработки Программы обеспечения надёжности. Основанием для разработки ПОН автоматизированной информационной системы (АИС) является тактико-техническое задание (ТТЗ) на ОКР;
* нормативно-технические и организационно-методические документы, используемые при выполнении мероприятий ПОН. При выполнении мероприятий ПОН АИС используются следующие нормативно-технические и организационно-методические документы:

1. Международный стандарт МЭК 300-1 (1993)/ ИСО 9000-4. Часть 1. Управление программой надежности;
2. Международный стандарт НЭК 300-2 (проект). Управление надежностью. Часть 2. Элементы и задачи программы надежности;
3. ГОСТ РВ 15.211-2002 Система разработки и постановки на производство военной техники. Порядок разработки программ и методик испытаний опытных образцов изделий. Основные положения;
4. ГОСТ В 15.206-84 Система разработки и постановки на производство военной техни­ки. Программы обеспечения надёжности. Общие требования;
5. ГОСТ 27.002-89 Надежность в технике. Основные понятия. Термины и определения;
6. ГОСТ 27.003-90 Надежность в технике. Состав и общие правила задания требований по надежности;
7. ГОСТ 27.410-87 Надежность в технике. Методы контроля показателей надежности и планы контрольных испытаний на надежность;
8. ГОСТ 40.9001-88 (ИСО 9001-87) Системы качества. Модель для обеспечения качества при проектировании и (или) разработке, производстве, монтаже и обслуживании;

* основные сведения о назначении, конструкции и эксплуатации автоматизированной информационной системы;
* требования по надёжности. В данном подразделе отражаются общие сведения по надежности, требования по безотказности, долговечности, ремонтопригодности, сохраняемости и специальные требования по надежности программного обеспечения (ПО). Программное обеспечение при выходе из строя или отключении отдельных АРМ должно обеспечить функционирование остальных АРМ в составе сети. Ошибочные действия пользователей не должны приводить к потере информации и к «зависанию» ПО. Программное обеспечение АИС должно обеспечивать обнаружение и диагностику ошибок с выдачей соответствующих сообщений пользователю. ПО при отсутствии сбоев и отказов технических средств должно функционировать в режимах, регламентированных требованиями программной документации, а также обеспечивать обработку информации вида и в объемах, регламентированных требованиями программной документации.

### **Задание 2. Инструменты, обеспечивающие надежность программных продуктов, план обеспечения надежности**

**Методы проектирования надежного программного обеспечения**

Методы проектирования надежного программного обеспечения можно разбить на следующие группы:

1. Предупреждение ошибок - методы, позволяющие минимизировать или исключить появление ошибки.
2. Обнаружение ошибок - методы направленные на разработку дополнительных функций программного обеспечения, помогающих выявить ошибки.
3. Устойчивость к ошибкам - дополнительные функции программного обеспечения, предназначенные для исправления ошибок и их последствий, обеспечивающие функционирование системы при наличии ошибок.

Методы предупреждения ошибок

Методы предупреждения ошибок концентрируются на отдельных этапах процесса проектирования программного обеспечения и включают в себя:

* методы достижения большей точности при преобразовании информации;
* методы улучшения обмена информацией;
* методы немедленного обнаружения и устранения ошибок.

Методы обнаружения ошибок

Методы обнаружения ошибок базируются на введении в программное обеспечение системы различных видов избыточности:

* Временная избыточность. Использование части производительности аппаратных средств для контроля исполнения и восстановления работоспособности ПО после сбоя.
* Информационная избыточность. Дублирование части данных информационной системы для обеспечения надёжности и контроля достоверности данных.
* Программная избыточность включает в себя: «взаимное недоверие» - компоненты системы проектируются, исходя из предположения, что другие компоненты и исходные данные содержат ошибки, и должны пытаться их обнаружить; немедленное обнаружение и регистрацию ошибок; выполнение одинаковых функций разными модулями системы и сопоставления результатов обработки; контроль и восстановление данных с использованием других видов избыточности.

**Методы обеспечения устойчивости к ошибкам**

Методы обеспечения устойчивости к ошибкам направлены на минимизацию ущерба, вызванного появлением ошибок, и включают в себя:

* обработку сбоев аппаратуры;
* повторное выполнение операций;
* динамическое изменение конфигурации;
* сокращенное обслуживание в случае отказа отдельных функций системы;
* копирование и восстановление данных;
* изоляцию ошибок.

## **МДК 01.01.Т2 Безопасность и управление доступом в информционных систем**

### **Задание 3. Технологическая модель подсистемы информационной безопасности**

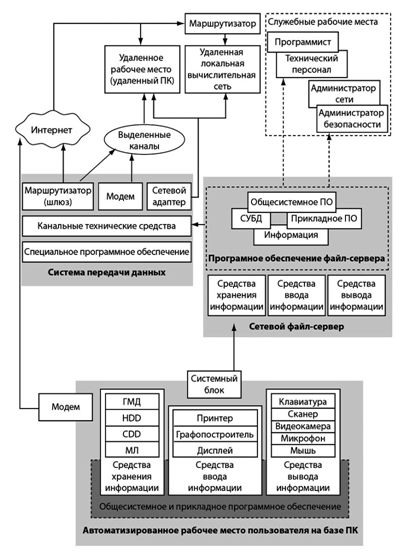
Современные распределенные корпорации, имеющие подразделения на разных континентах, имеют сложную техническую, инженерную и информационную инфраструктуру. Создание информационной сети такой корпорации и её эффективная защита является чрезвычайно сложной концептуальной и технологической задачей.

Первоначальное решение, характерное для последнего десятилетия прошлого века, использовать для формирования сети телефонные линии быстро привело к нагромождению коммуникаций и к невозможности эффективной защиты. Последующее создание и сопровождение собственных корпоративных сетей для обеспечения информационного обмена данными на базе таких линий связи стало обходиться в миллионы долларов.

Быстрое развитие технологий Internet, образование, рост и развитие "всемирной паутины" позволили создать достаточно дешевые и надежные коммуникации. Однако техническая надежность связи вовсе не означала безопасности корпоративных сетей, имеющих выходы в Интернет. Общие принципы построения Интернет и его использование как общедоступной сети с публичными сервисами привели к тому, что стало очень трудно обеспечить надежную защиту от проникновения в корпоративные и государственные сети, построенные на базе протоколов TCP/IP и Internet -приложений — Web, FTP, e-mail и т.д.

Целевое назначение любой корпоративной информационной системы состоит в обеспечении пользователей необходимой информацией в режиме "On Line" и адекватном информационном сопровождении деятельности предприятия.

Базисом КИС является общесистемное программное обеспечение, которое включает операционную систему и программные оболочки, программы общего и прикладного назначения: автоматизированные рабочие места (АРМ) и Web-сервисы общего и специального назначения, СУБД и управление интегрированными вычислительными и мультимедийными приложениями, а также доступом в локальные и внешние сети.

  
**Рис. 2.** Схема корпоративной информационной системы, включающей локальные сети и выход в Internet

Физически нижний уровень КИС базируется на серверах, рабочих станциях, персональных компьютерах различного назначения и коммуникационных устройствах, а также на программном обеспечении, реализующем работу перечисленных устройств. В связи с этим подсистема ИБ начинается с защиты именно этого программно-аппаратного оборудования. С этой целью можно использовать известные защитные средства операционных систем, антивирусные пакеты, средства и устройства аутентификации пользователя, средства криптографической защиты паролей и данных прикладного уровня. Все эти средства образуют базу для реализации первого уровня технологической модели подсистемы ИБ.



**Рис. 3.**Четырехуровневая технологическая модель подсистемы информационной безопасности

Второй физический уровень КИС — рабочие станции, серверы и персональные компьютеры объединятся в локальные сети, которые организуют внутреннее Intranet-пространство предприятия и могут быть иметь выходы во внешнее Internet-пространство.

В этом случае речь идет о средствах информационной защиты (СЗИ) второго уровня — уровня защиты локальных сетей, который обычно включает:

* средства безопасности сетевых ОС;
* средства аутентификации пользователей (User Authentication Facilities — UAF);
* средства физического и программного разграничения доступа к распределенным и разделяемым информационным ресурсам;
* средства защиты домена локальной сети (Local Area Network Domain — LAND);
* средства промежуточного доступа (Proxy Server) и межсетевые экраны (Firewall);
* средства организации виртуальных локальных подсетей (Virtual Local Area Network — VLAN);
* средства обнаружения атаки и уязвимостей в системе защиты локальных сетей.

Следующий уровень реализации КИС — объединение нескольких локальных сетей географически распределенного предприятия в общую корпоративную Intranet-сеть через открытую сеть на базе современных технологий поддержки и сопровождения таких сетей (Quality of Service — QoS) с использованием открытой среды Internet в качестве коммутационной среды.

В этом случае на третьем уровне защиты КИС используются технологии защищенных виртуальных сетей (Virtual Private Networks — VPN). VPN-технологии часто интегрируются со средствами первого и второго уровней. Такой защищенный VPN-канал может простираться не только до маршрутизаторов доступа и пограничных Firewall'лов, но и до серверов и рабочих станций локальной сети.

Четвертый уровень защиты КИС — организация защищенного межкорпоративного обмена в среде электронного бизнеса (eBusiness). Методологической и технологической основой такой защиты являются методы и технологии управления публичными ключами и сертификатами криптографической защиты (Public Key Infrastructure — PKI). Суть этих технологий состоит в реализации двух глобальных функций: генерации и корректном распространении ключей и сертификатов и отслеживании их жизненного цикла. Базой для реализации средств защиты будут электронная цифровая подпись (Electronic Digital Signature — EDS) и VPN-технологии.

Отметим, что два нижних уровня защиты являются достаточно традиционными, так как они предназначены для обеспечения безопасности конкретной физически реализованной КИС. Верхние два уровня относятся к обеспечению безопасности передачи данных и электронного бизнеса, который осуществляется уже не в физическом, а в виртуальном пространстве, при этом VPN-технологии обеспечивают защищенный обмен данными в межкорпоративном пространстве, а PKI-технологии обеспечивают VPN-устройства ключами и сертификатами. В настоящее время на рынке имеется достаточное число технических и программных решений для защиты данных, информации, систем и сетей. Ниже рассмотрены некоторые базовые технологии на примере криптографической защиты данных, технологий межсетевых экранов, защищенных VPN-каналов связи, антивирусных и биометрических методов.

### **Задание 4. Защита Windows 10 встроенными средствами**

В состав Windows 10 входит "Безопасность Windows", которая и предоставляет последние обновления для антивирусной защиты. Ваше устройство получит активную защиту с момента запуска Windows 10. "Безопасность Windows" регулярно проверяет устройство на наличие вредоносных программ, вирусов и угроз безопасности. В дополнение к этой защите в режиме реального времени обновления скачиваются автоматически, чтобы обеспечить безопасность устройства и защитить его от угроз.

**Важные сведения о безопасности**

* Система "Безопасность Windows" встроена в Windows 10 и использует антивирусную программу под названием «Антивирусная программа в Microsoft Defender». (В предыдущих версиях Windows 10 приложение "Безопасность Windows" называлось "Центр безопасности Защитника Windows").
* Антивирусная программа в Microsoft Defender автоматически отключается при установке и включении другого антивирусного приложения. При удалении другого приложения антивирусная программа в Microsoft Defender включится автоматически.
* Если у вас возникли проблемы с получением обновлений для системы "Безопасность Windows", см. разделы **Устранение ошибок Центра обновления Windows** и **Вопросы и ответы по обновлению Windows**.
* Инструкции по удалению приложения см. в разделе **Восстановление работы** или **удаление приложения в Windows 10.**
* Инструкции по изменению учетной записи пользователя на учетную запись администратора см. в разделе **Создание локально учетной записи пользователя** или **администратора в Windows 10**.

**Знакомство с функциями "Безопасности Windows" и их настройка**

Безопасность Windows — это место для управления инструментами, которые защищают ваше устройство и ваши данные:

* **Защита от вирусов и угроз**. Позволяет отслеживать угрозы для вашего устройства, запускать проверку и устанавливать обновления для обнаружения новейших угроз. (Некоторые из этих параметров недоступны, если вы используете Windows 10 в S-режиме.)
* **Защита учетных записей**. Доступ к параметрам входа в систему и параметрам учетной записи, включая Windows Hello и динамическую блокировку.
* **Брандмауэр и защита сети**. Позволяет управлять параметрами брандмауэра и отслеживать события, происходящие с вашими сетями и подключениями к Интернету.
* **Управление приложениями и браузером**. Обновление параметров фильтра SmartScreen в Microsoft Defender для защиты вашего устройства от потенциально опасных приложений, файлов, сайтов и загрузок, а также обеспечение защиты от эксплойтов, где вы можете настроить параметры защиты для ваших устройств.
* **Безопасность устройств.** Просмотр встроенных параметров безопасности, которые помогают защитить устройство от атак вредоносного программного обеспечения.
* **Производительность и работоспособность устройства**. Просмотр состояния работоспособности и производительности устройства и поддержание устройства в чистом и актуальном состоянии с помощью последней версии Windows 10.
* **Семья**. Отслеживание действий ваших детей в Интернете и устройств вашей семьи.

Можно настроить, каким образом эти функции "Безопасности Windows" будут защищать ваше устройство. Чтобы получить к ним доступ, нажмите кнопку **Пуск** > **Параметры** > **Обновление и безопасность** > **Безопасность Windows**. Затем выберите функцию, которую вы хотите изучить.

**Запуск проверки наличия вредоносных программ вручную**

Если вы хотите проверить безопасность определенного файла или папки, щелкните правой кнопкой мыши файл или папку в проводнике, а затем выберите **Проверить с помощью Microsoft Defender**.

Если вы подозреваете, что устройство может содержать вирусы или вредоносные программы, необходимо незамедлительно проверить ваше устройство, выполнив быструю проверку. Это намного быстрее выполнения полной проверки всех файлов и папок.

**Запуск быстрой проверки в системе "Безопасность Windows"**

1. Выберите **Пуск** > **Параметры** > **Обновление и безопасность** > **Безопасность Windows** и затем нажмите **Защита от вирусов и угроз**.
2. В разделе **Текущие угрозы** выберите **Быстрая проверка**(или в предыдущих версиях Windows 10 в разделе **Журнал угроз** выберите **Проверить сейчас**).

Если проверка не обнаружила никаких проблем, но вы все еще обеспокоены, возможно проверить устройство более тщательно.

**Запуск расширенной проверки в "Безопасности Windows"**

1. Выберите **Пуск** > **Параметры** > **Обновление и безопасность** > **Безопасность Windows**  и затем нажмите **Защита от вирусов и угроз**.
2. В разделе **Текущие угрозы** выберите **Параметры проверки** (или в предыдущих версиях Windows 10 в разделе**Журнал угроз** выберите **Запустить новую расширенную проверку**).
3. Выберите один из вариантов проверки.

* **Полная проверка**(проверка файлов и программ, выполняемых на вашем устройстве)
* **Выборочная проверка**(проверка определенных файлов или папок )
* **Проверка автономного Microsoft Defender** (запустите эту проверку, если ваше устройство было заражено или может быть заражено вирусом или вредоносной программой).

1. Выберите **Проверить сейчас**.

**Создание собственного расписания проверок**

Несмотря на то что функция "Безопасность Windows" регулярно проверяет устройства для обеспечения их безопасности, можно также задать время и периодичность проведения проверок.

Планирование сканирования

1. Нажмите кнопку **Пуск**, введите **планировщик заданий** в строке **Поиск** и в списке результатов выберите **Планировщик задач**.
2. В области слева выберите стрелку (>) рядом с разделом **Библиотека планировщика заданий**, чтобы развернуть его. Затем сделайте то же самое с пунктами **Microsoft** > **Windows**, прокрутите список вниз и выберите папку **Windows Defender**.
3. На центральной панели выберите пункт **Запланированная проверка в Защитнике Windows.**(Наведите указатель мыши на соответствующие пункты, чтобы просмотреть их полные названия.)
4. В области **Действия** справа прокрутите список вниз, а затем выберите **Свойства**.
5. В открывшемся окне перейдите на вкладку **Триггеры** и нажмите кнопку **Создать**.
6. Задайте время и периодичность, а затем нажмите кнопку **OK**.
7. Просмотрите расписание и выберите **OK**.

**Включение и отключение защиты, обеспечиваемой антивирусной программой в Microsoft Defender в режиме реального времени**

Иногда может потребоваться быстро остановить работу защиты в режиме реального времени. Когда защита в режиме реального времени отключена, открываемые или скачиваемые файлы не проверяются на наличие угроз. Однако скоро защита в режиме реального времени будет включена автоматически для возобновления защиты вашего устройства.

**Временное отключение защиты в режиме реальном времени**

1. Выберите **Пуск** > **Параметры** > **Обновление и безопасность** > **Безопасность Windows** и затем **Защита от вирусов и угроз** > **Управление параметрами**. (В предыдущих версиях Windows 10 выберите раздел **Защита от вирусов и угроз** > **Параметры защиты от вирусов и угроз**.)
2. Задайте параметру **Защита в режиме реального времени** значение **Выкл.** и нажмите **Да** для подтверждения.

**ПМ.03 Разработка интернет информационных систем**

## **МДК.03.01 Технологии и средства разработки Web-приложений**

### **Задание 5. HTML и CSS**

Создайте html- и css- файлы, результат которых показан на рисунке.

Шаг 1. В html файле создаем таблицу и подключаем css файл:

|  |  |
| --- | --- |
|  |  |
|  |  |

**Рис. 4.** Html файл

Шаг 2. Создаем css файл со свойствами таблицы:

|  |  |
| --- | --- |
|  |  |

**Рис. 5.** Css файл

Шаг 3. Результат:



**Рис. 6.** Результат

### **Задание 6. HTML и CSS**

Поворот на 360 при наведении. Сделайте все изображения круглыми. Задайте для изображения свойство transition со значением 1s. Для :hover сделайте поворот на 360 по часовой стрелке.

Шаг 1. Создаем html файл, в котором создадим контейнер для картинок и поместим сами картинки:



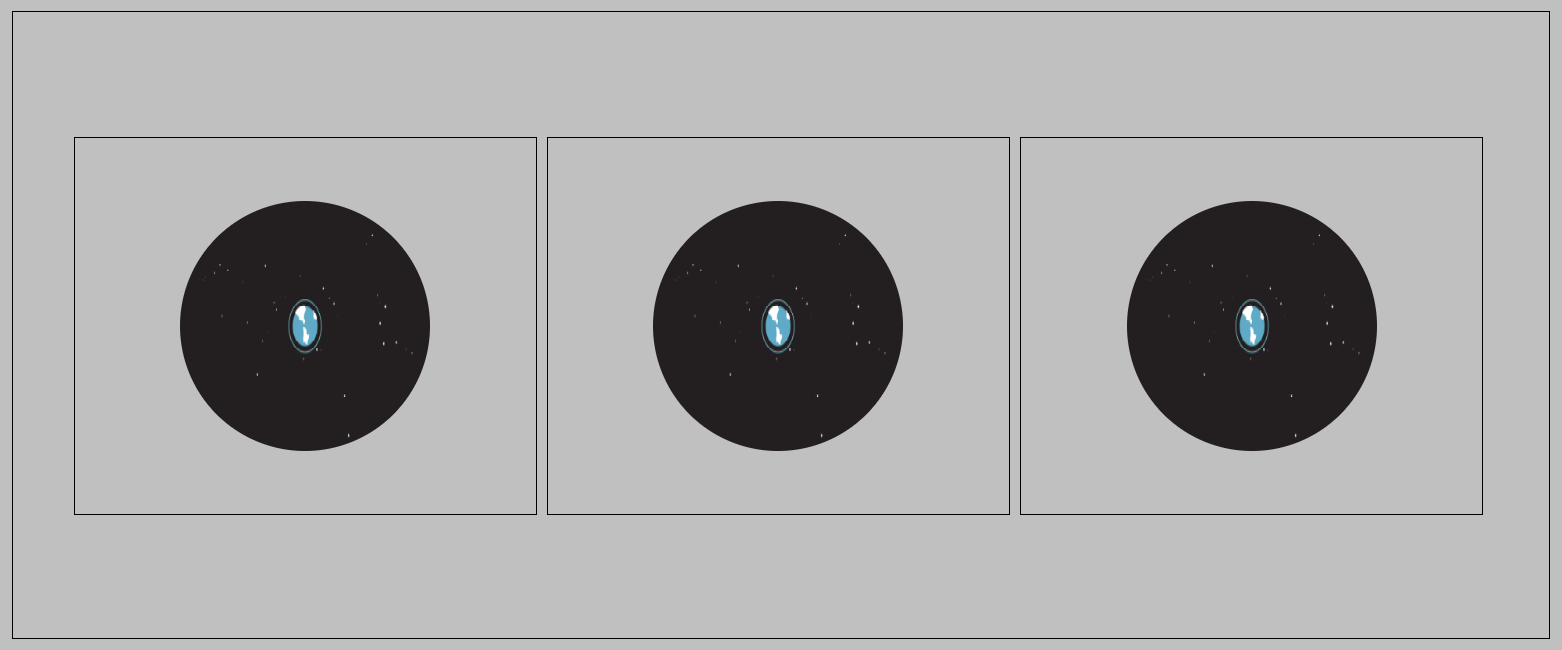
**Рис. 7.** Html файл

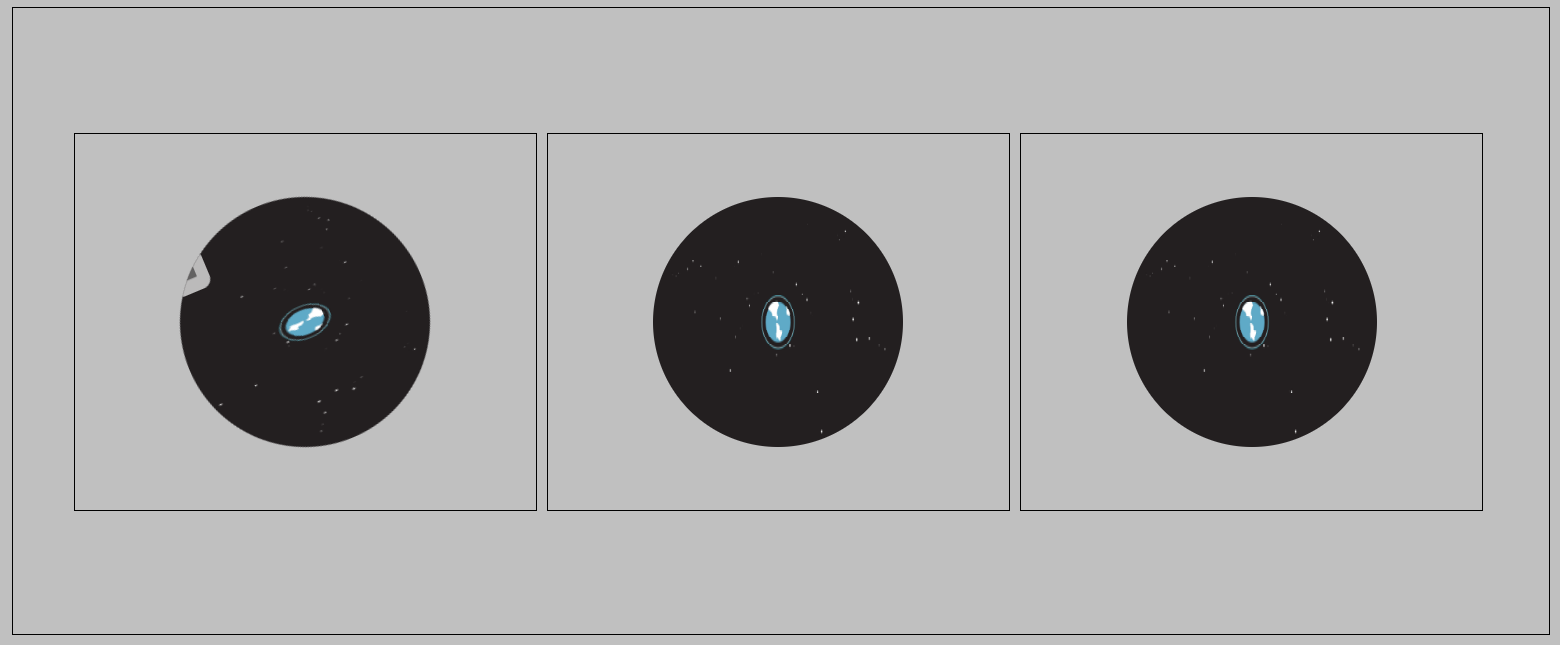
Шаг 2. Создадим css файл в котором зададим свойства для контейнера и картинок и свойства по условию:



**Рис. 8.** Css файл

Шаг 3. Результат:



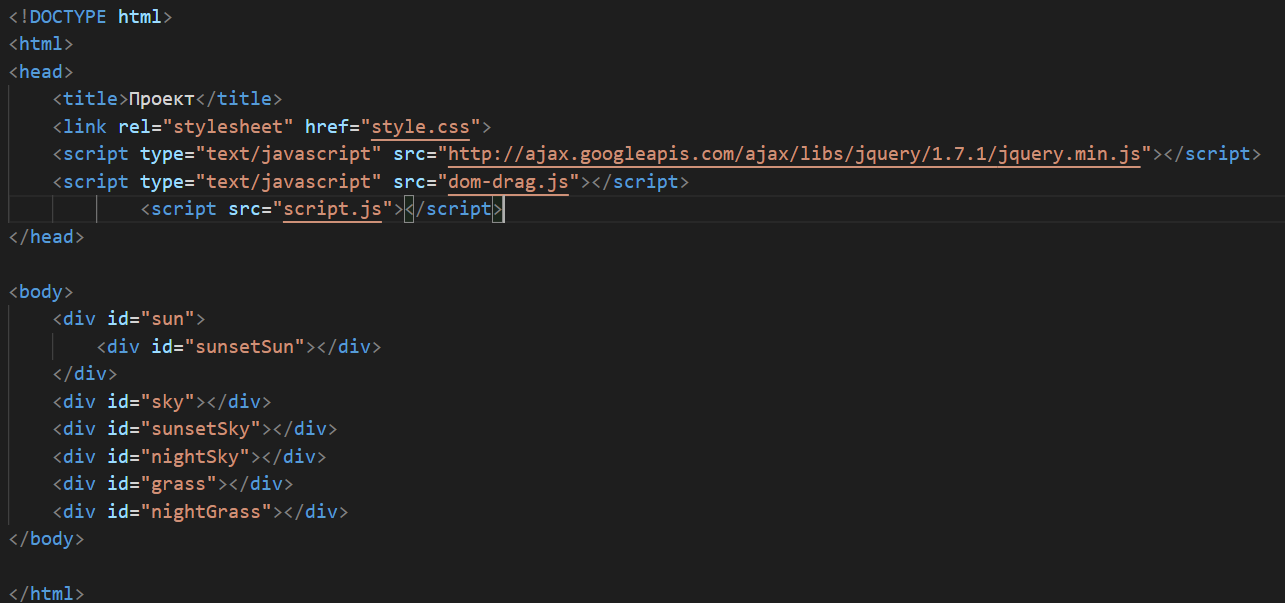


**Рис. 9-10.** Результат

### **Задание 7. Основы программирования на JavaScript**

С помощью JS сделайте анимацию заката и рассвета

Шаг 1. Создадим html файл со всей «структурой» заката: небо, лужайку и само солнце. Подключим к нему css и js файлы:



**Рис. 11.** Html файл

Шаг 2. Создадим css файл со всеми свойствами будущего заката:

|  |  |
| --- | --- |
|  |  |

**Рис. 12-13.** Css файлы

Шаг 3. Создадим JS файл с анимацией заката:

|  |  |
| --- | --- |
|  |  |

**Рис. 14-15.** JS файл

Шаг 4. Результат:



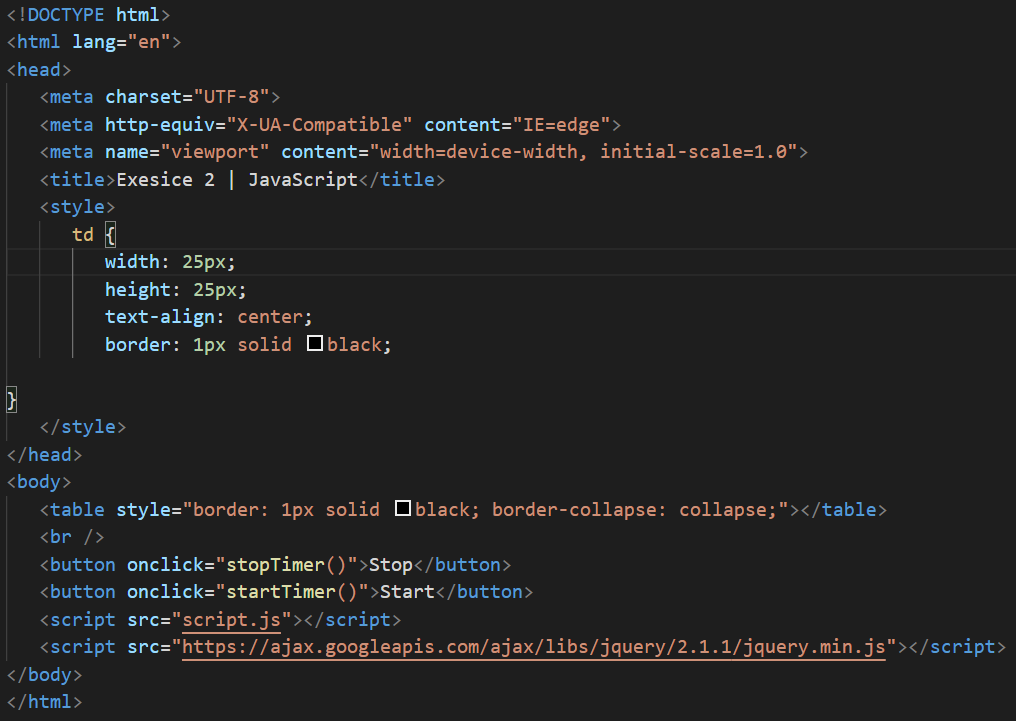


**Рис. 16-17.** Результат

### **Задание 8. Основы программирования на JavaScript**

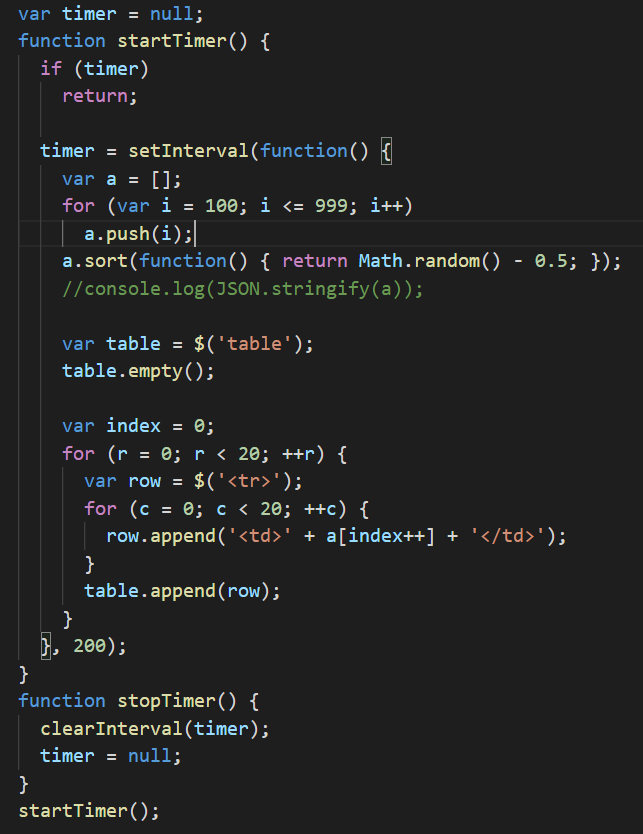
Создать таблицу размером 20 на 20, заполненную случайными целыми числами от 100 до 999.

Шаг 1. Создаем html файл, в котором будет генерироваться таблица и подключаем js файлы:



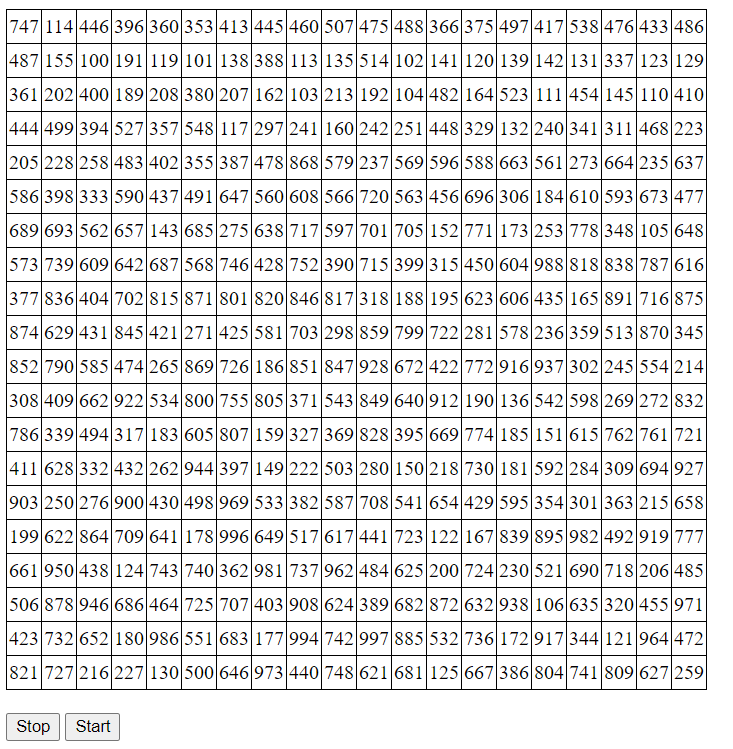
**Рис. 18.** Html файл

Шаг 2. Создаем js файл:



**Рис. 19.** JS файл

Шаг 3. Результат:

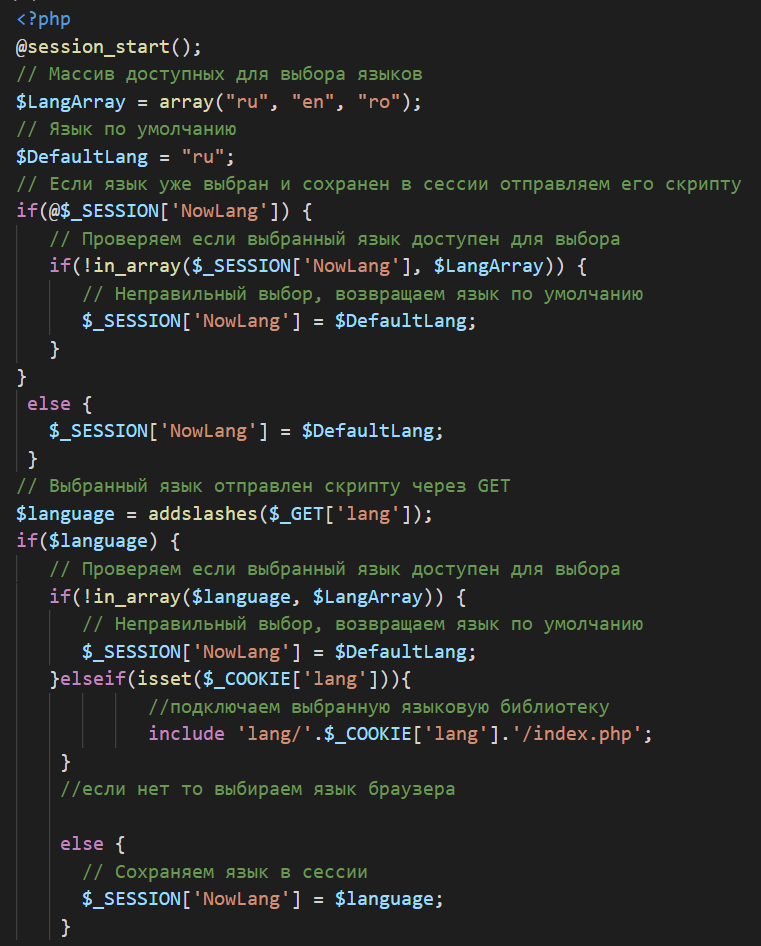


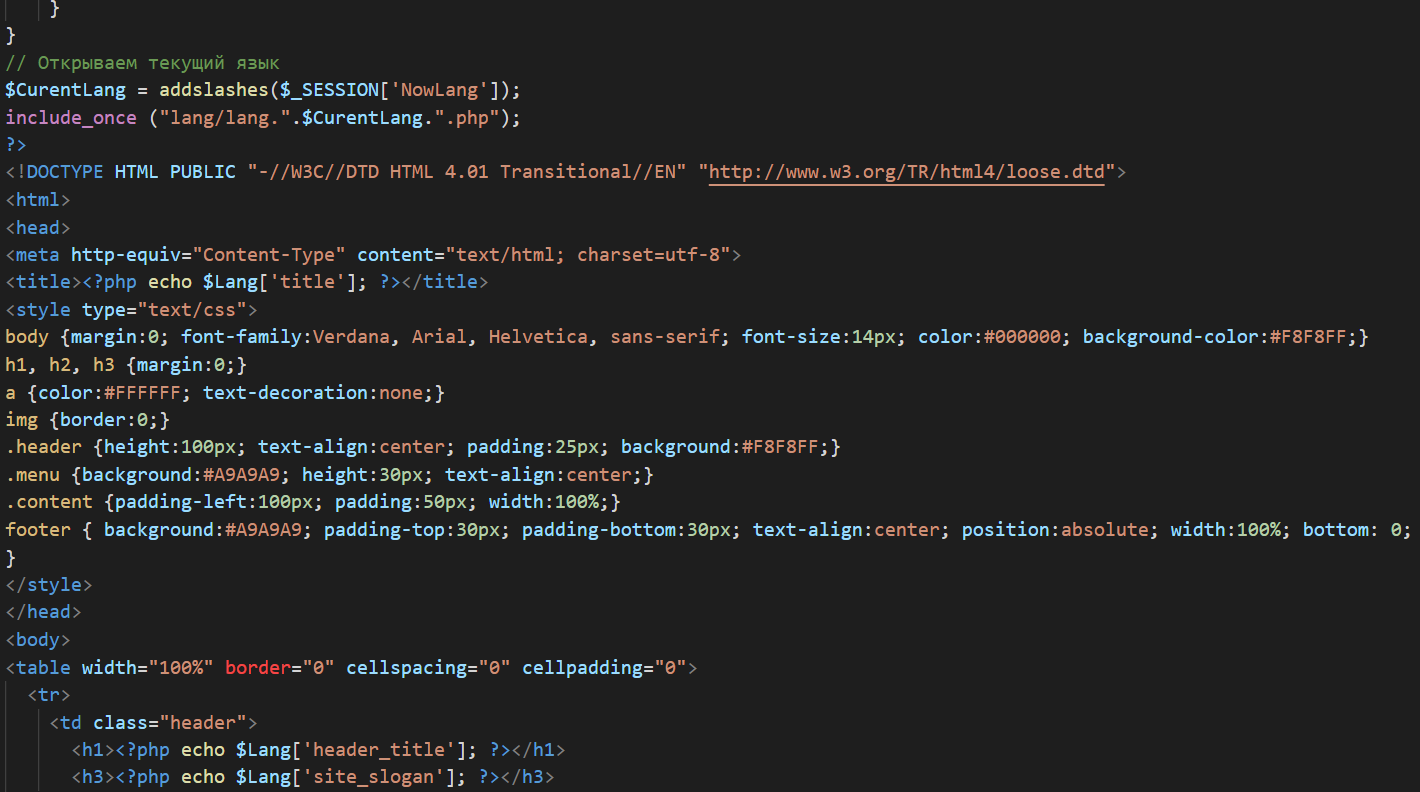
**Рис. 20.** Результат

### **Задание 9. Введение в программирование на PHP5**

Создайте куку, отвечающую за язык сайта. Создайте несколько гиперссылок, которые будут изменять текущий язык сайта, путем изменения куки.

Шаг 1. Создаем php файл со структурой сайта-примера и куки файлом:





**Рис. 21.** Основной php файл

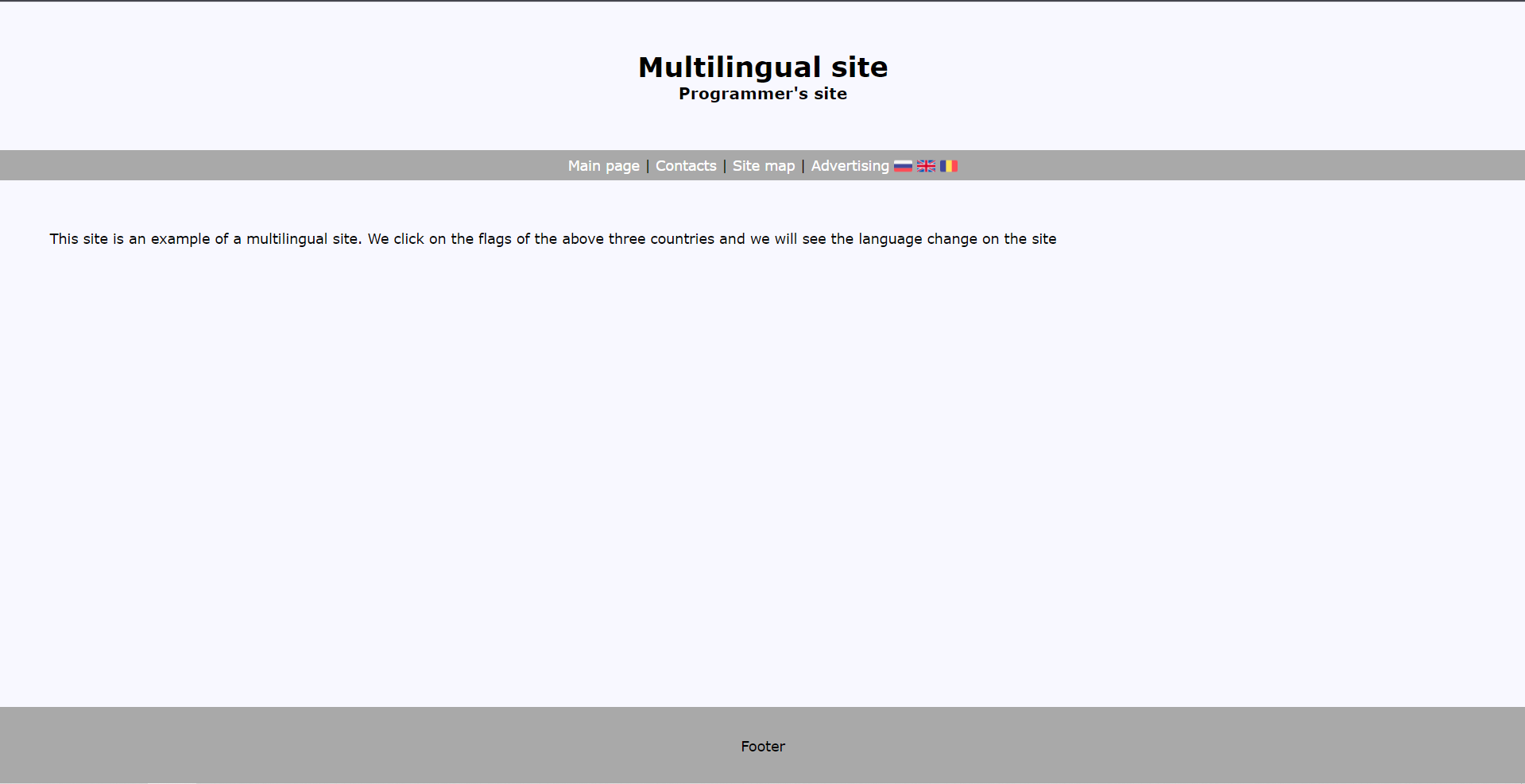
Шаг 2. Создаем три php файла для трех языков (Русский, Английский, Румынский):

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Рис. 22-24.** Языковые php файлы

Шаг 3. Результат:







**Рис. 25-27.** Результат

### **Задание 10. Введение в программирование на PHP5**

Напишите простое приложение для тестирования так, чтобы все промежуточные результаты тестирования записывались в сессионную переменную.

Шаг 1. Создадим html с тестом, который будет через форму подключаться к файлу php:



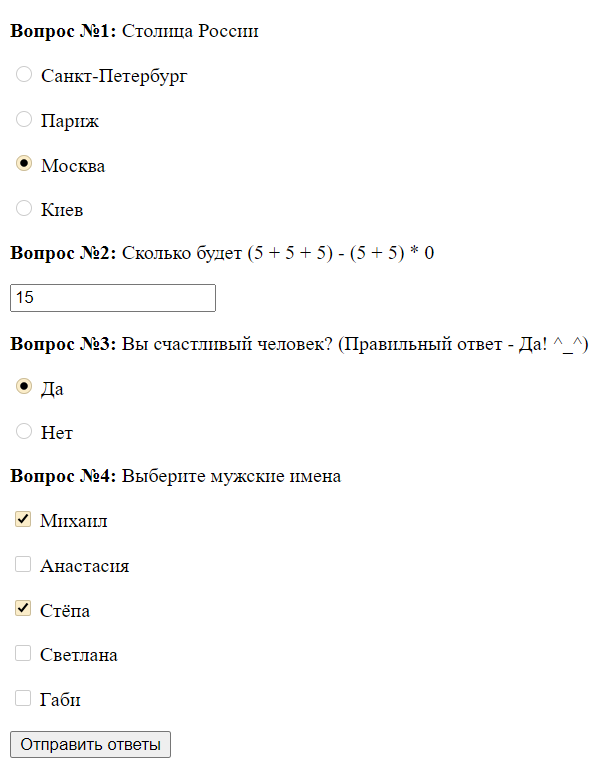
**Рис. 28.** Html файл

Шаг 2. Создаем php файл, в котором будет вестись подсчет баллов:

|  |  |
| --- | --- |
|  |  |

**Рис. 29-30.** Php файл

Шаг 3. Результат:

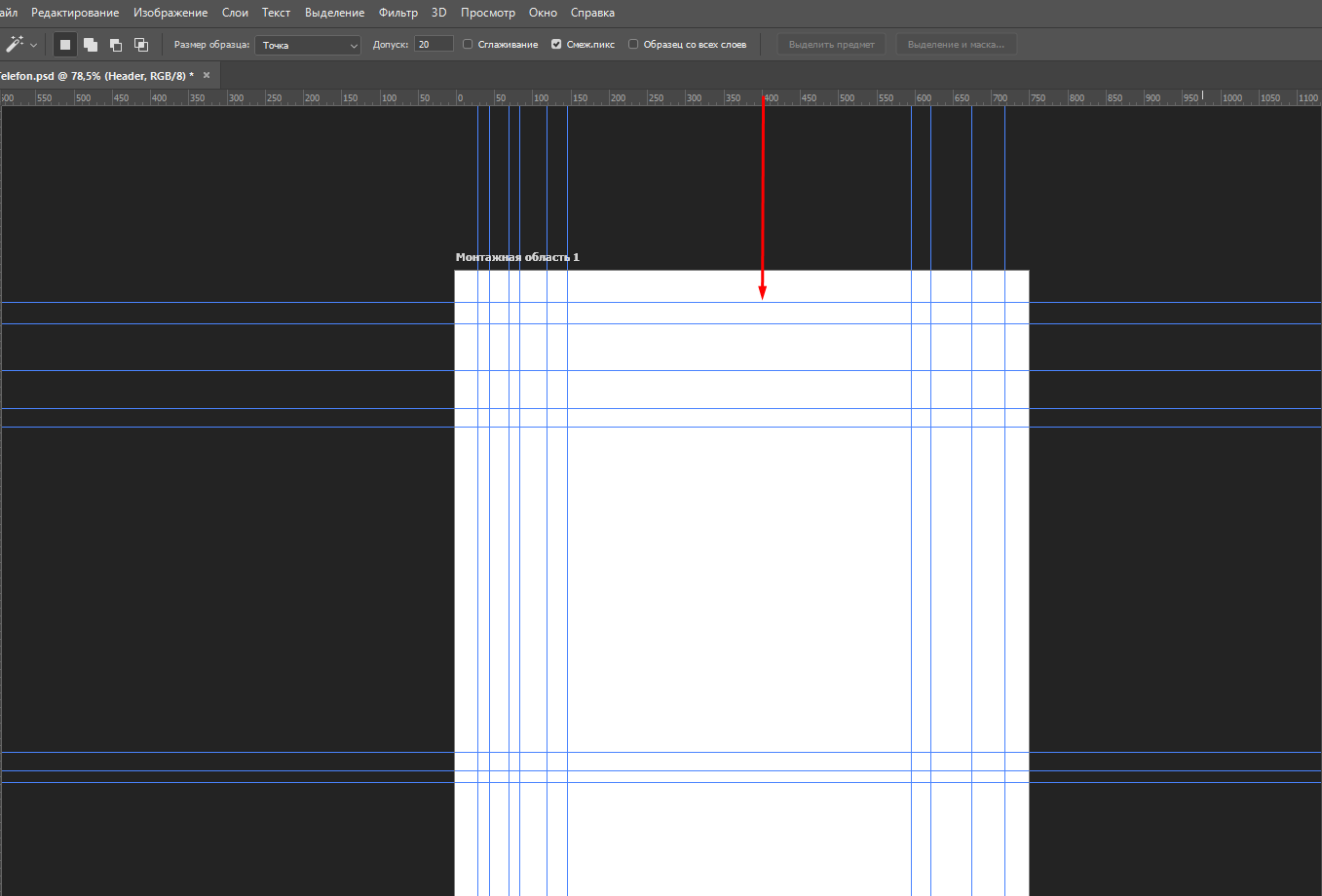




**Рис. 31-32.** Результат

### **Задание 11. Создание макета мобильной версии сайта**

Шаг 1. Для создания макета мобильной версии сайта я использовал приложение PhotoShop. Для точного расположения элементов сайта мне потребовалось построить сетку. Для этого достаточно зажать кнопку мыши на линейке и перетащить в монтажную область:



**Рис. 33.** Монтажная область

Шаг 2. Затем я поэтапно начал составлять макет сайта:

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Рис. 34-36.** Слои макета

Шаг 3. Итоговый результат:

|  |  |
| --- | --- |
|  |  |

**Рис. 37-38.** Готовый макет

### **Задание 12. Сверстать сайт по макету**

По созданному мной макету я сверстал мобильную версию сайта:

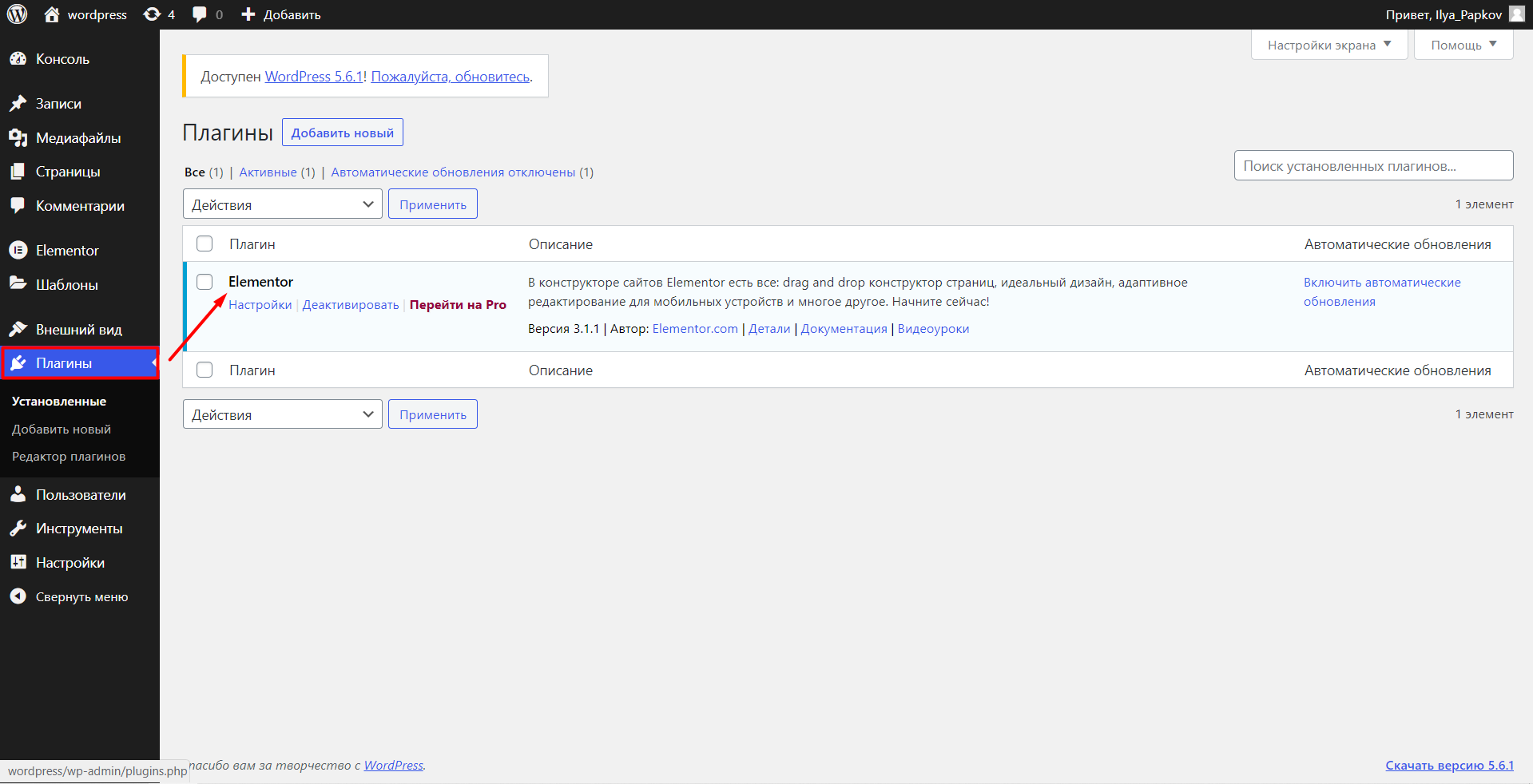
|  |  |
| --- | --- |
|  |  |
|  |  |

**Рис. 39-42.** Мобильная версия сайта по макету

## **МДК.03.02 Разработка интернет информационных систем**

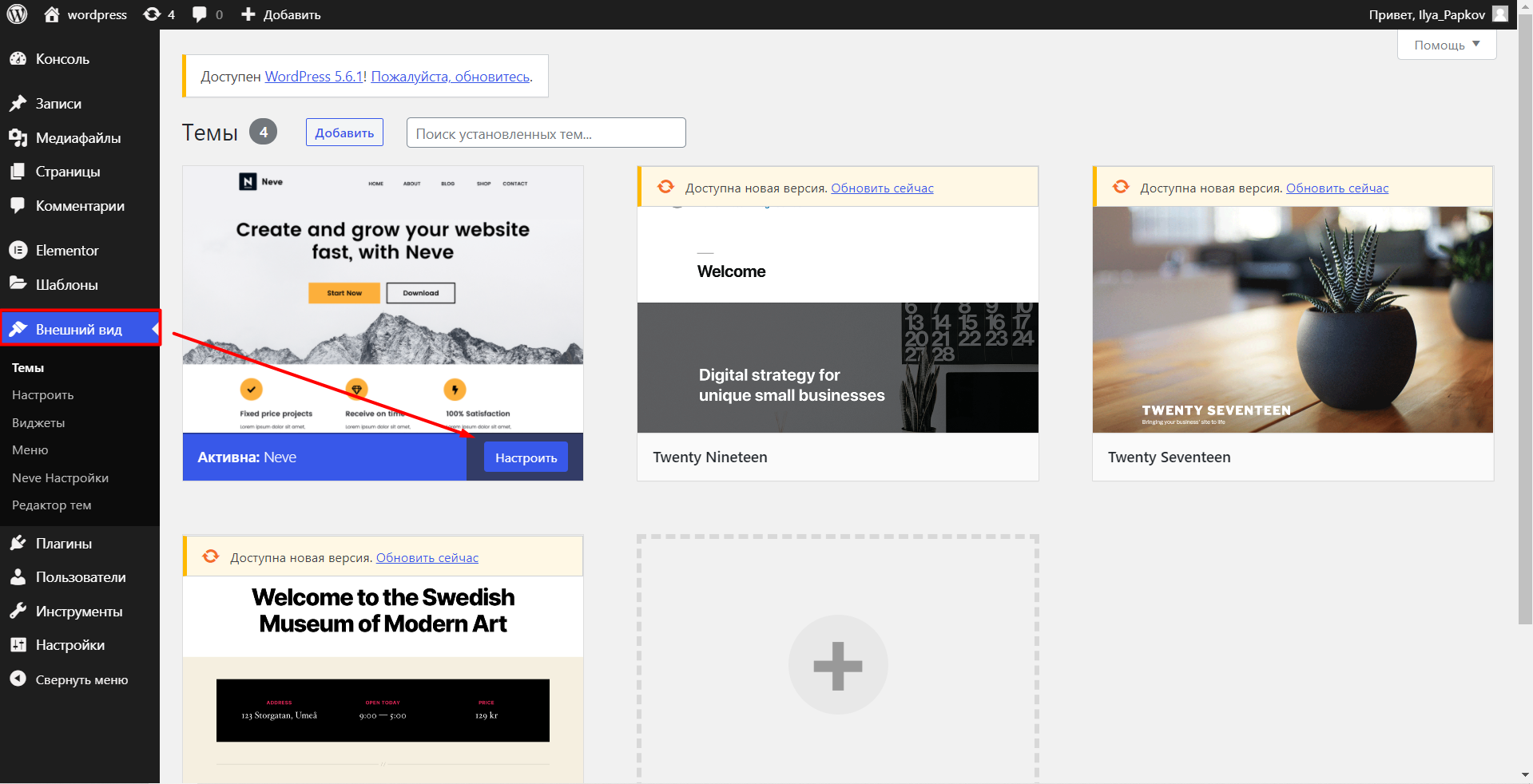
### **Задние 13. Написать сайт по теме КП с помощью конструктора сайта**

Шаг 1. Я решил выбрать конструктор сайта WordPress, из-за его популярности и довольно большего количества плагинов, доступных на данную CMS. Для работы с моим сайтом я установил плагин Elementor:



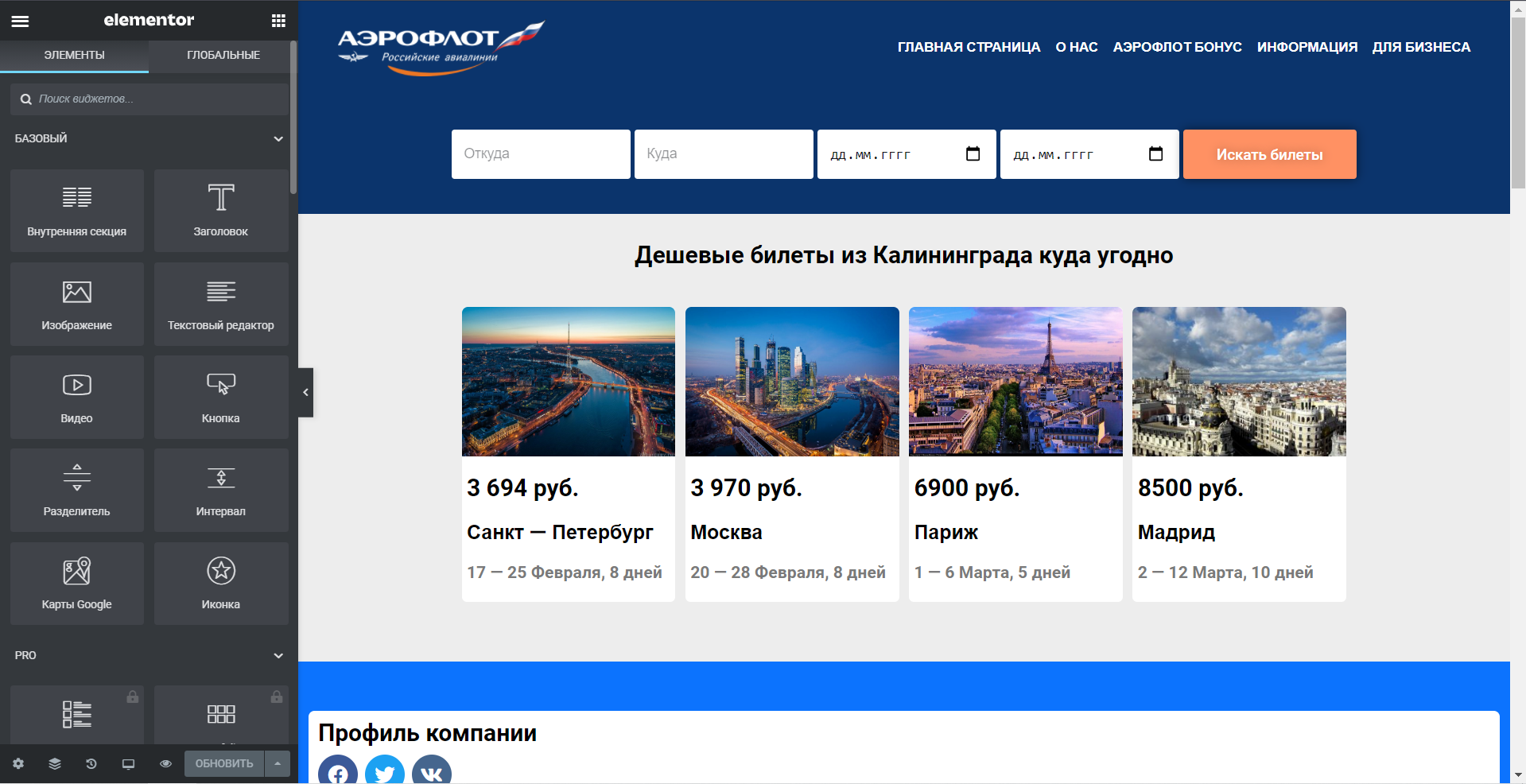
**Рис. 43.** Установка elementor

Так же я установил шаблон Neve, чтобы на его базе создать свой сайт:



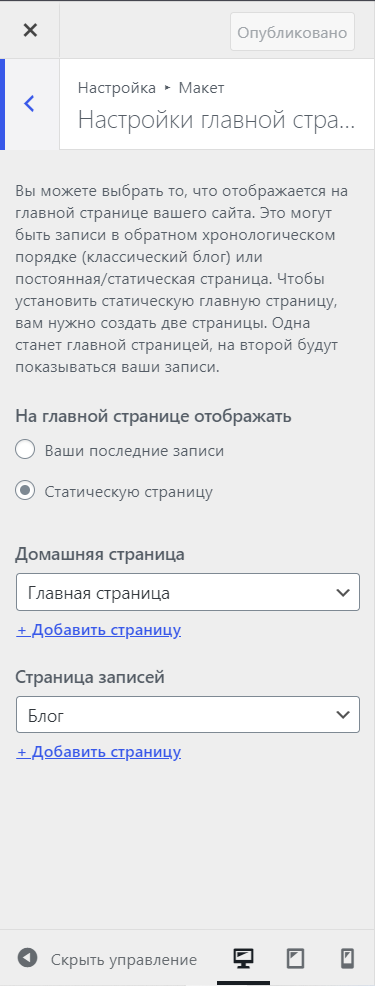
**Рис. 43.** Выбор шаблона для сайта

Шаг 2. Далее я создал свой сайт с помощью ранее установленного плагина Elementor. Он очень удобен тем, что в нем предустановлены все нужные мне кейсы для создания сайта:



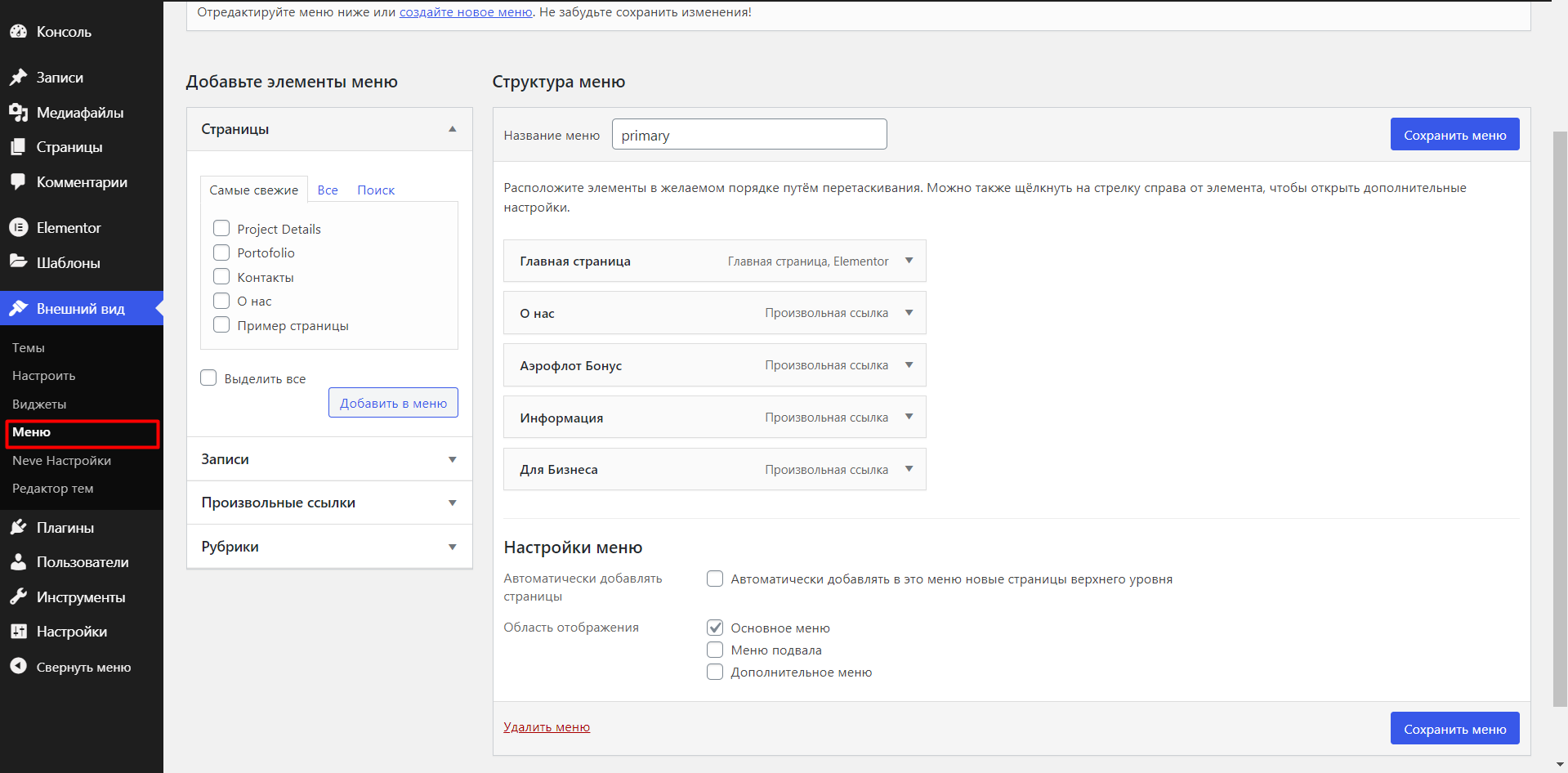
**Рис. 44.** Создание сайта через elementor

Далее перехожу во вкладку Внешний вид => Настроить и выбираю главной страницей нашу статичную страничку, которую мы создали.



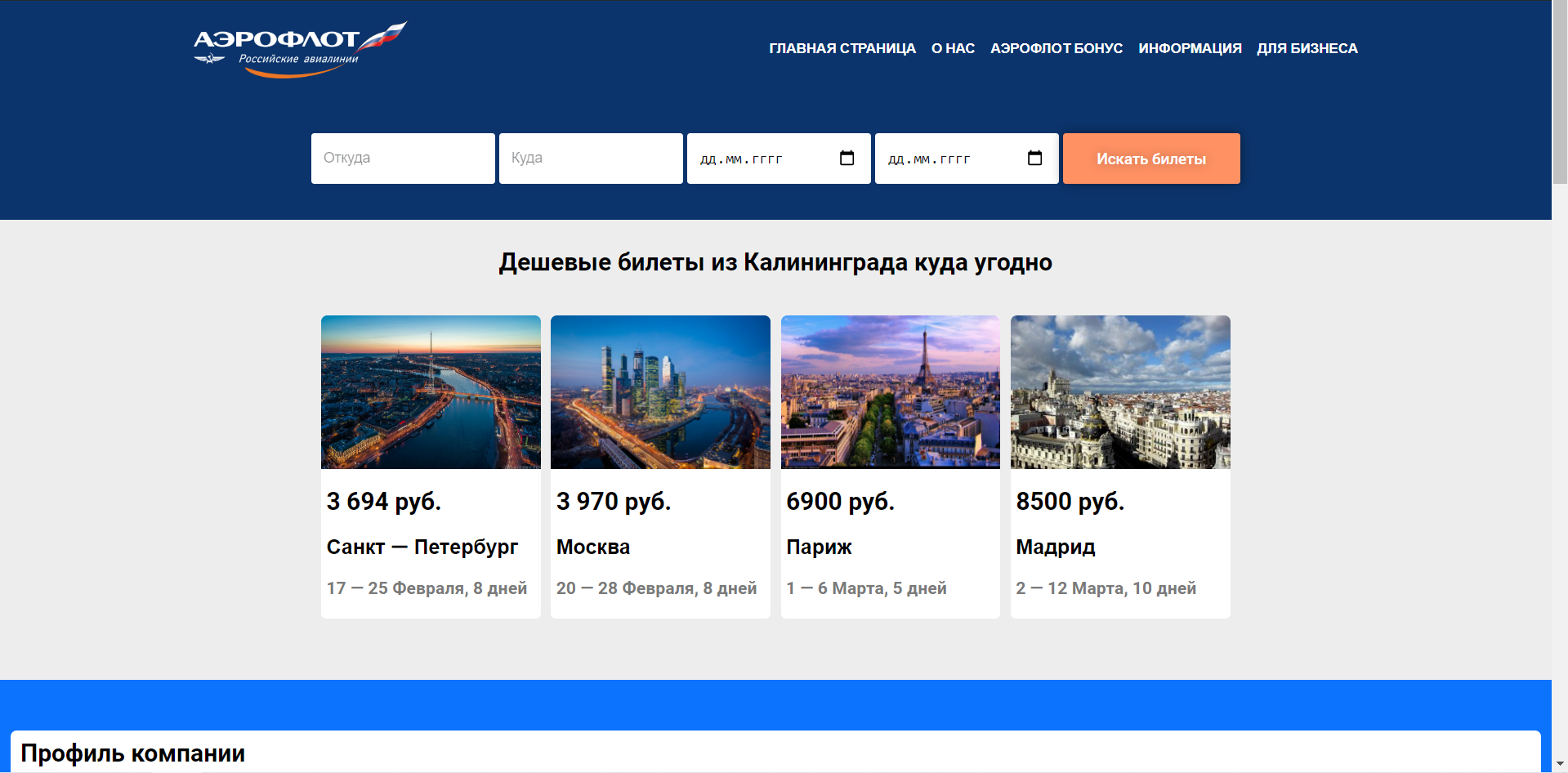
**Рис. 45.** Настройка главной страниц

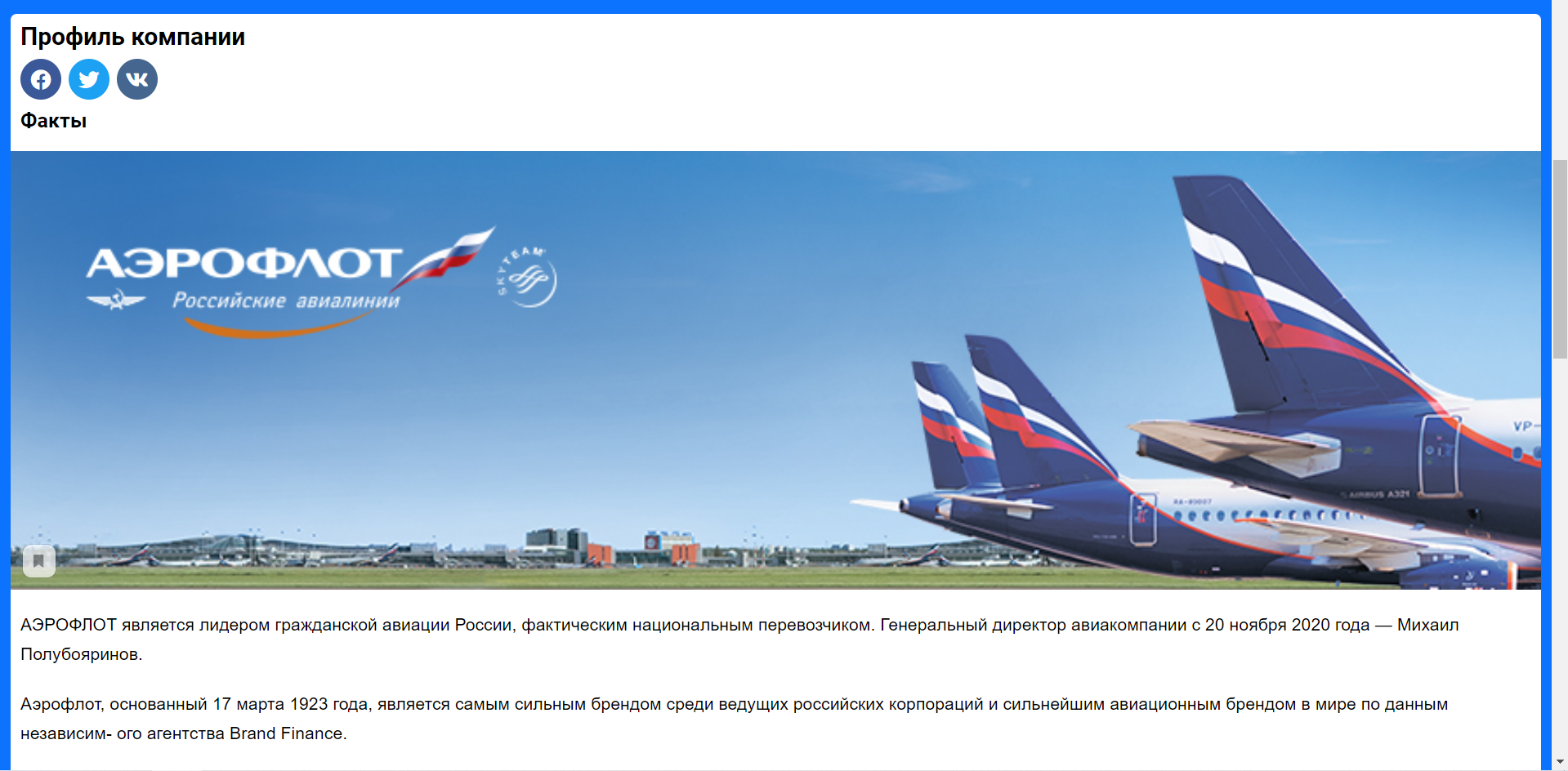
После этого я перешел во вкладку Внешний вид => Меню и настраиваю меню моего сайта:

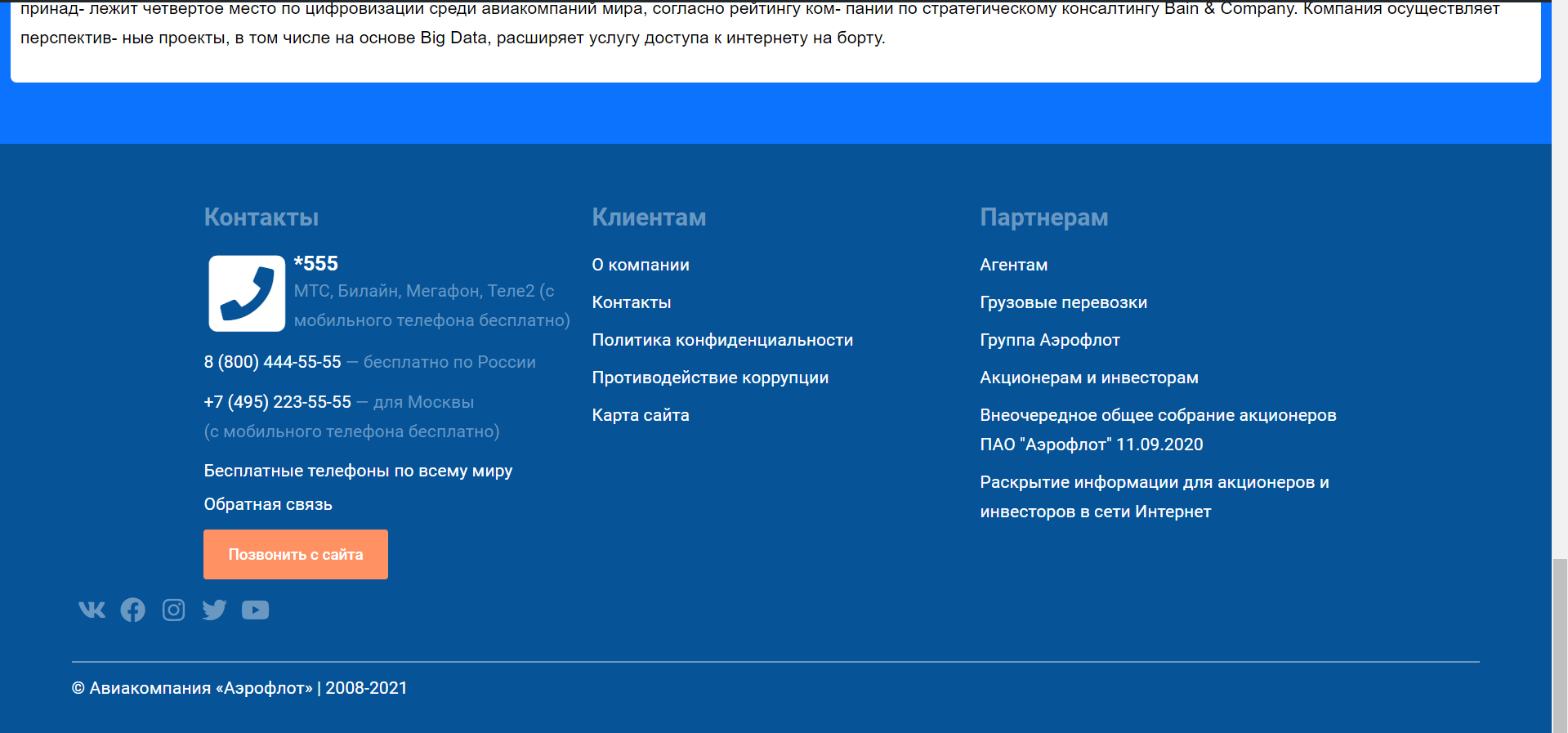


**Рис. 45.** Настройка меню сайта

Шаг 3. Готовый вариант сайта:







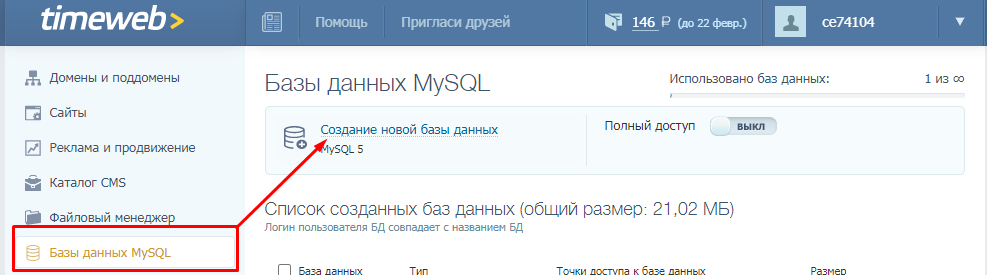
**Рис. 46-48.** Готовый вариант сайта

### **Задание 14. Разместить сайт на хостинге**

Шаг 1. Выбираем хостинг.

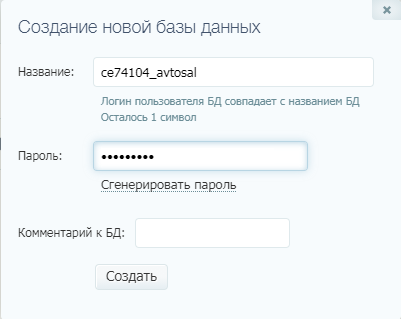
Я выбрал хостинг TimeWeb потому, что данный хостинг предоставляет бесплатное доменное имя и бесплатный период на 10 дней использования. Так же данный хостинг имеет простой и интуитивно понятный интерфейс и позволяет производить настройку среды, в которой будет находиться наш сайт.

Шаг 2. Подготовка хоста.

1. Первым делом я создаю Базу Данных где будут храниться файлы WordPress, для этого я перехожу во вкладку Базы Данных MySQL и нажимаем создание новой базы данных. 

**Рис. 49.** Создание БД

1. Вводим название и пароль для нашей БД и нажимаем на кнопку создать.



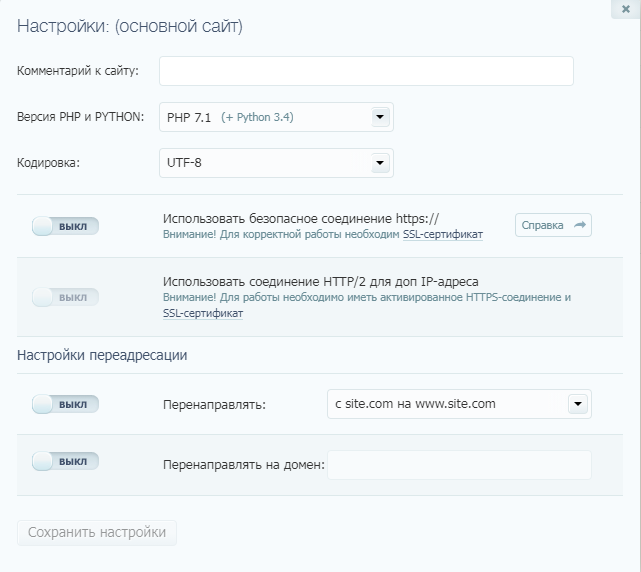
**Рис. 50.** Настройка БД

1. Далее мы переходим во вкладку сайты и создаём новый сайт. Для этого нам нужно ввести его название.



**Рис. 51.** Создание сайта

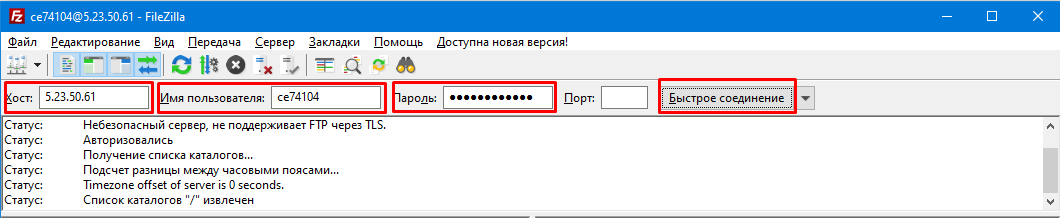
1. Настраиваем наш сайт и подвязываем к нему домен.



**Рис. 52.** Настройка сайта и привязка домена

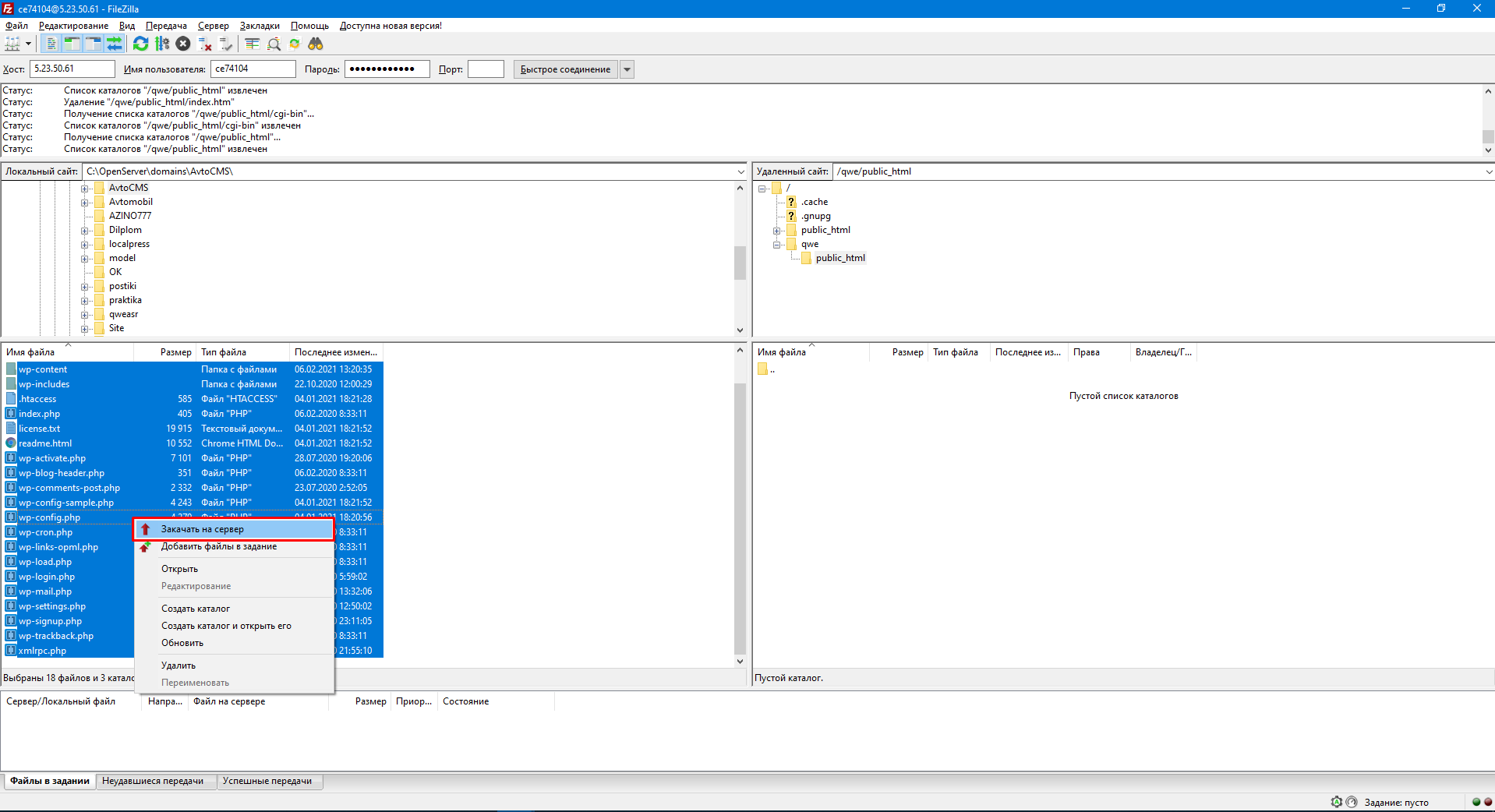
Шаг 3. Переносим локальные файлы на сервер.

1. Для переноса файлов я использовал программу FileZilla. В ней мы должны ввести IP нашего сервера, логин и пароль от нашего личного кабинета. После этого мы нажимаем на кнопку быстрое соединение.



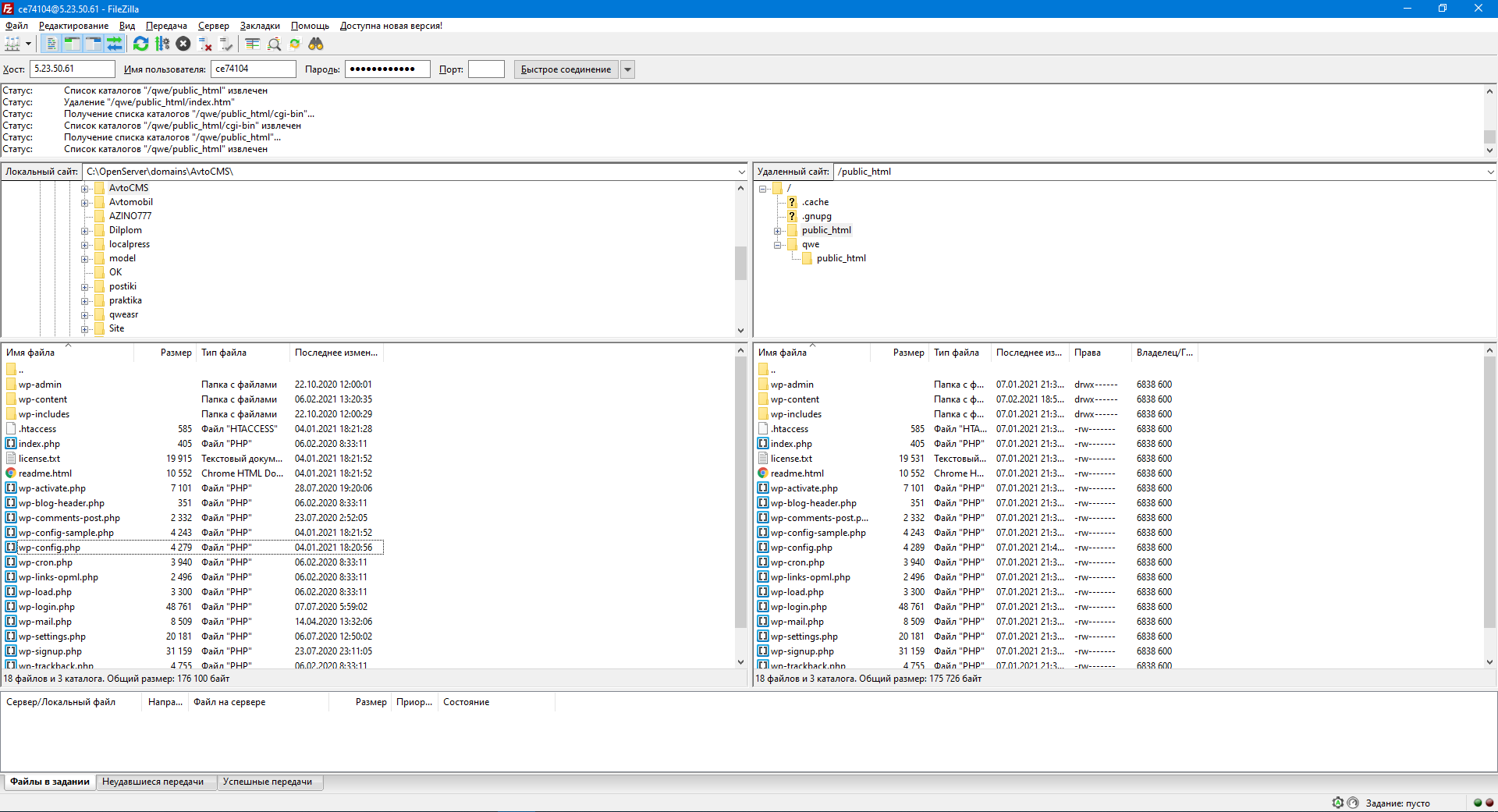
**Рис. 53.** Перенос файлов

1. После успешного подключения в левом окне программы мы должны выделить все файлы нашего сайта, а в правом окне мы должны перейти в директорию на хосте в которую будут переноситься файлы. Далее мы должны нажать в левом окне на правую кнопку мыши и выбрать пункт “Закачать на сервер”. После этого будет произведено копирование файлов в директорию хоста.



**Рис. 54.** Загрузка файлов на хостнг

1. После успешного завершения копирования мы увидим все наши файлы в правом окне.

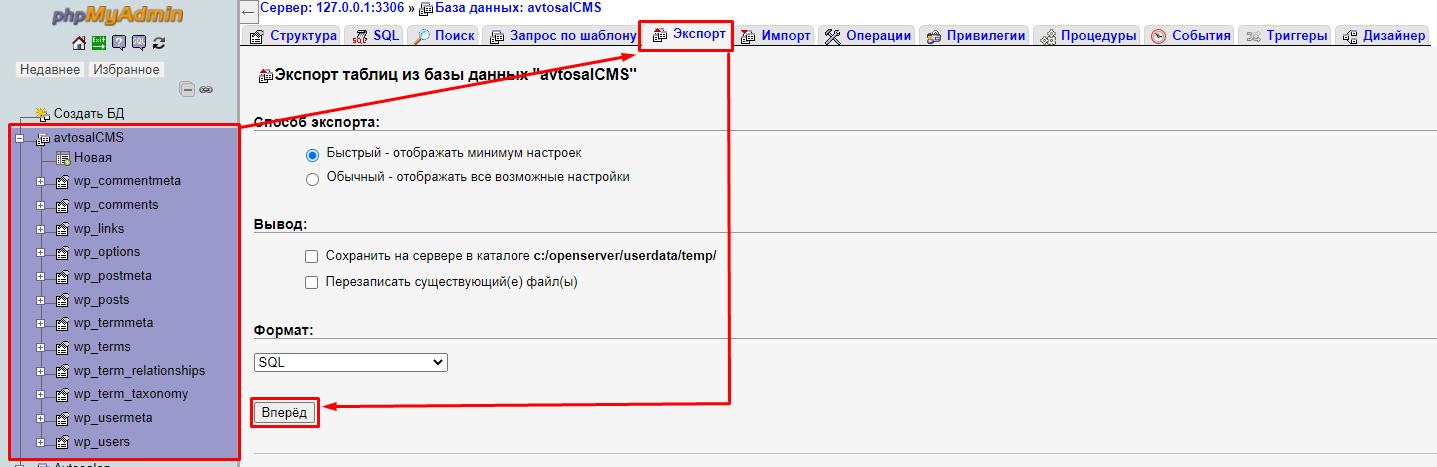


**Рис. 55.** Успешная загрузка файлов

Шаг 4. Настройка сайта на хостинге.

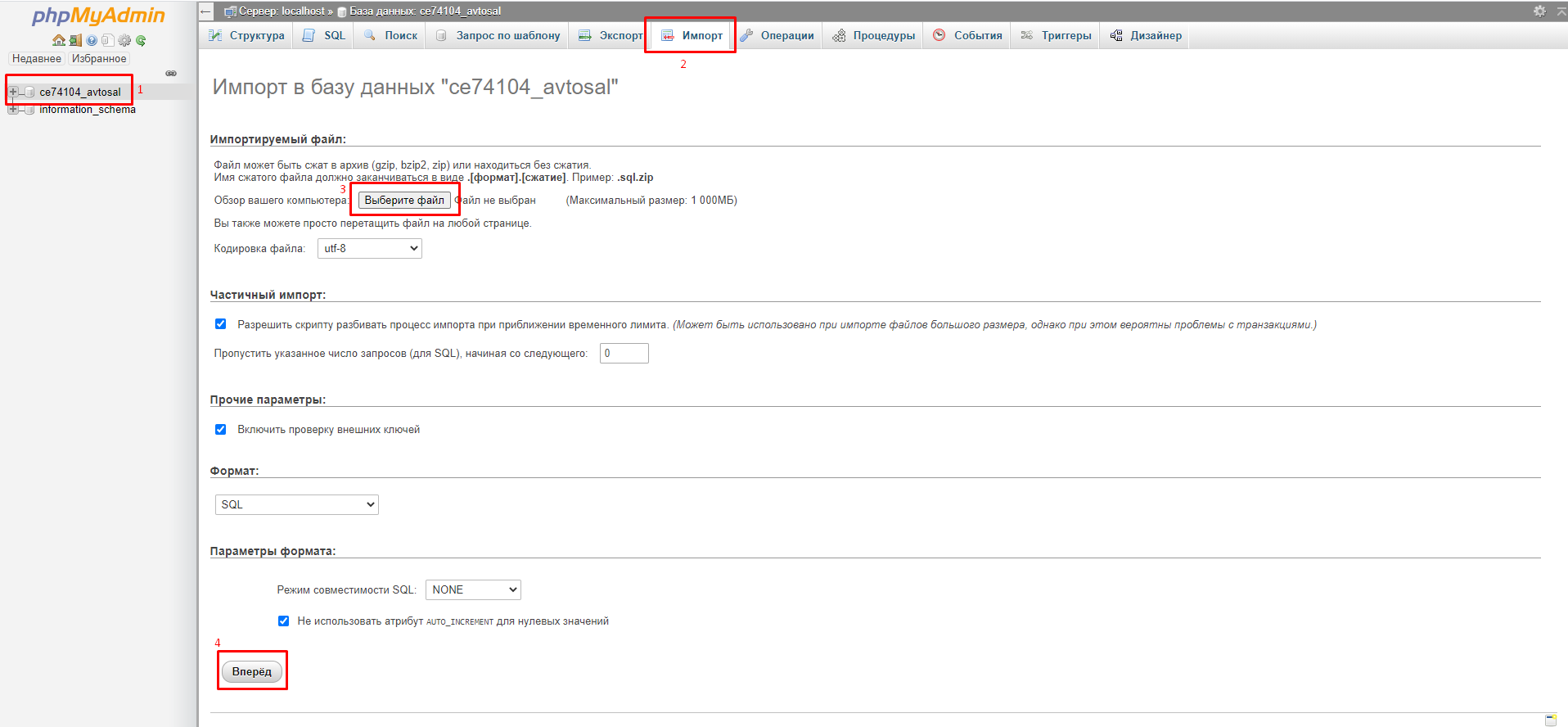
Для успешного запуска сайта на хостинге мы должны исправить некоторые файлы и импортировать локальную БД.

1. Мы должны экспортировать БД с локального компьютера и импортировать её на хостинг. Для этого я запустил OpenServer и перешел в phpMyAdmin. Затем я выбрал свою БД и нажал



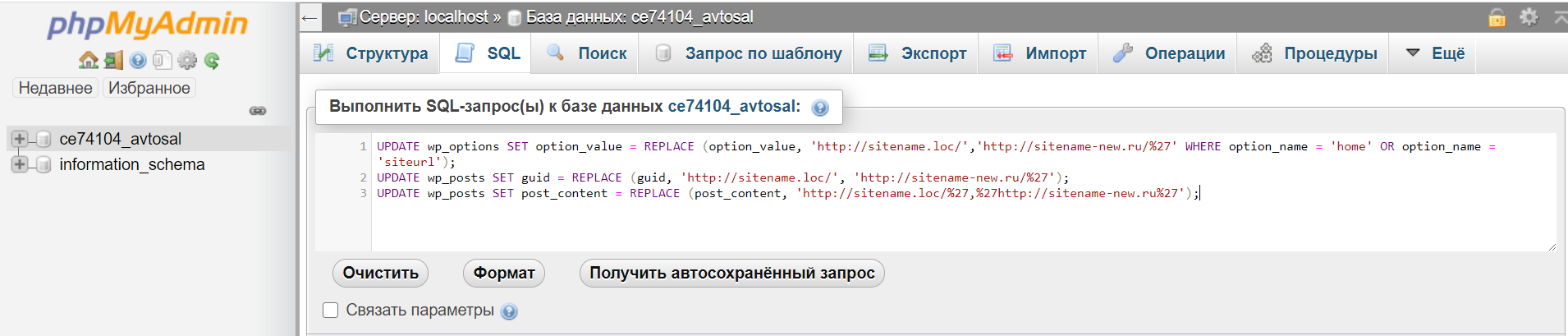
**Рис. 56.** Экспорт БД

1. Далее мы переходим в БД на хостинге, которую мы создали ранее и выбираем пункт импорт. Затем мы выбираем файл с нашей БД и импортируем его.



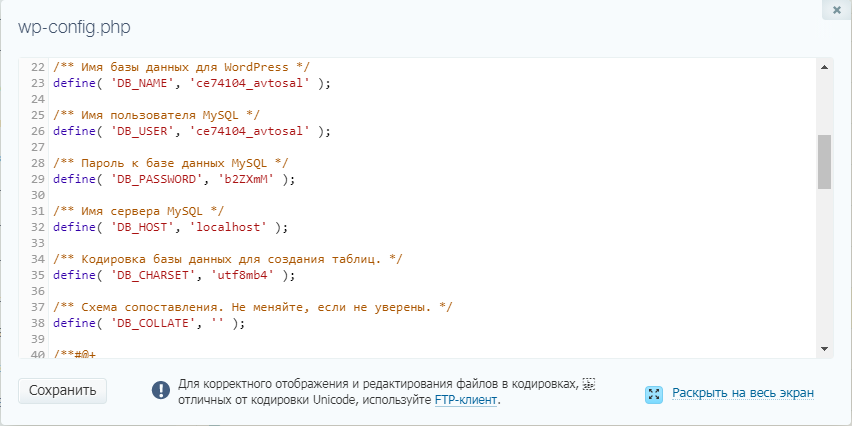
**Рис. 57.** Импорт БД н ахостинг

1. Далее мы должны подправить ссылки в нашей БД. Ниже приведён шаблон запросов, которые необходимо выполнить для корректной работы сайта.



**Рис. 58.** Настройка БД

1. После изменения БД мы должны внести изменения в wp-config.php файл где мы указываем данные для подключения к БД. Для этого мы переходим во вкладку Сайты и нажимаем на Управление файлами сайтов, выбираем директорию нашего сайта и находим файл. После этого мы исправляем поля: имя БД, имя пользователя, пароль и нажимаем на кнопку сохранить.



**Рис. 59.** Настройка БД

Шаг 5. Проверка сайта.

После того как мы проделали все шаги, мы должны перейти на наш сайт и убедиться, что всё работает корректно или подправить какие ошибки.

# **Использованная литература**

1. Джон Дакетт "Javascript и jQuery. Интерактивная веб-разработка", год: 2017, Издательство: Издательство Э;  
2. Книга Git для профессионального программиста, C. Чакон, Б. Штрауб, год 2016, издатель: Питер;  
3. <https://www.filezilla.ru/documentation/> - Документация по программе FileZilla;  
4. <https://codex.wordpress.org/> - Документация по пользованию конструктора сайта Wordpress’a;  
5. <https://bootstrap-4.ru/docs/4.0/layout/grid/> - Пользование сеткой bootstrap для создания сайтов;  
6. <https://itchief.ru/lessons/html-and-css/css-media-que>.. - Работа с медиа-запросов для верстки мобильной версии сайта;  
7. Книга PHP 7 в подлиннике, Дмитрий Котеров, Год издания 2016, Издательство: БХВА-Петербург