**Введение**

На учебной практике была поставлена задача, разработать web-ресурс на тему: Разработка электронного средства обучения, предназначенного для обучения детей и подростков футболу.

Цель проекта – предоставить удобную платформу для детей и подростков электронную платформу обучения, в которой рассказывается о всех тонкостях обучению мастерства.

Создаваемый web-ресурс будет рассчитан для обученых и начинающих спортсменов, для детей. Применить данный сайт смогут дети и подростки, решившие заняться спортом.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». Он посвящён изучению предметной области и организационно-экономической сущности поставленной задачи. Также в нем описано, как задача решается в настоящее время, перечислены входные и выходные данные. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой создаётся данный проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачи.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки web-ресурса. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет чётко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации.

«Реализация задачи» – это третий раздел отчёта по практике, в котором описываются все элементы и объекты, которые будут использованы при реализации данного сайта. В этом разделе будут чётко описаны функции пользователя и их структура. Описано руководство программиста и будет предоставлена диаграмма компонентов.

Четвёртый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данной программы. Будут смоделированы все возможные действия пользователя при работе с web-ресурсом, начиная от входа на сайт заканчивая закрытием вкладки.

В разделе «Применение» будет описано назначение, область применения, среда функционирования данного программного продукта.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, степени автоматизации процессов на различных этапах разработки.

В разделе «Список используемых источников» будет приведён список используемых при разработке источников.

В приложениях к пояснительной записке будет приведены UX и UI проектирование сайта и диаграммы.

**1 Анализ задачи**

**1.1 Постановка задачи**

**1.1.1 Организационно-экономическая сущность задачи**

Наименование задачи: Разработка электронного средства обучения, предназначенного для обучения детей и подростков футболу.

Цель разработки: увеличение информативности и доступности детям и подросткам к обучению футбольным навыкам. Сайт предоставляет качественную и ценную информацию которая собиралась годами. Сайт может помочь обучится ребенку или подростку самостоятельно.

Назначение (для каких объектов, подразделений, пользователей предназначен ПП): сайт предназначен для обучению детей и подростков футбольным навыкам.

Периодичность использования ПП: по мере необходимости.

Источники и способы получения данных: будут предоставлены администратором.

Обзор существующих аналогичных ПП: на данный момент существует множество веб-ресурсов, аналогичных поставленной теме, например, такие как .

https://fckrasnodar.ru/club/card/

**1.1.2 Функциональные требования**

К поставленной задаче были заявлены следующие функциональные требования, которые может выполнять гость:

* Просмотр главной страницы с информацией;
* Просмотр контактной информации, соц.сетей;
* Просмотр категорий тренировок;
* Поиск по сайту;
* Записаться на тренировку;

Функциональные требования для администратора:

* Редактирование страниц сайта;
* Доступ к БД;
* Добавление/удаление информации;
* Ответы на обратную связь;
* Просмотр и обработка вопросов;

**1.1.3 Описание входной, выходной и условно-постоянной информации**

Вся информация, которой оперирует пользователь в процессе решения задачи подразделяется на:

* Входную информацию;
* Выходную информацию.

К входной информации можно отнести вводимые пользователем значения, например: описание, содержание тренировок (имя, адрес электронной почты, телефон и др.), различная информация.

К выходной информации можно отнести различную информацию о тренировках.

К условно-постоянной информации можно отнести описание.

**1.1.4 Нефункциональные требования**

**Требования к применению:**

Интерфейс сайта должен быть легким, понятным, функциональным и простым в использовании. Он не должен перегружать пользователя ненужной информацией.

**Требования к производительности:**

Требования к производительности наименьшие. Для повышения производительности веб-ресурса нужно иметь хорошее подключение к интернету.

**Требования к реализации:**

Для реализации данного программного продукта будет использоваться конструктор сайтов WordPress.

**Требования к надёжности:**

Веб-сайт должен иметь возможность самовосстановления после сбоя.

**Требования к интерфейсу:**

интерфейс должен быть интуитивно понятным и легким в использовании для пользователя, следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Должны быть предусмотрены удобные способы взаимодействия с пользователем (кнопки, меню, поля ввода и т.д.).

**1.2 Диаграмма вариантов использования**

Диаграмма вариантов использования – это один из видов диаграмм UML, которая показывает, какие функции предоставляет система и как она взаимодействует с внешними сущностями, называемыми актерами.

Диаграмма вариантов использования состоит из следующих элементов:

* Вариант использования – это овал с названием, который описывает конкретную функцию или сервис, который система предоставляет актеру.
* Актер – это человек, организация или другая система, которая использует или влияет на систему.
* Связь – это отношения, которые определяют, как элементы диаграммы взаимодействуют друг с другом и с системой.
* Система – это то, что моделируется диаграммой вариантов использования.

Ниже на рисунке 1 представлена диаграмма вариантов использования для учебно-методического комплекса.

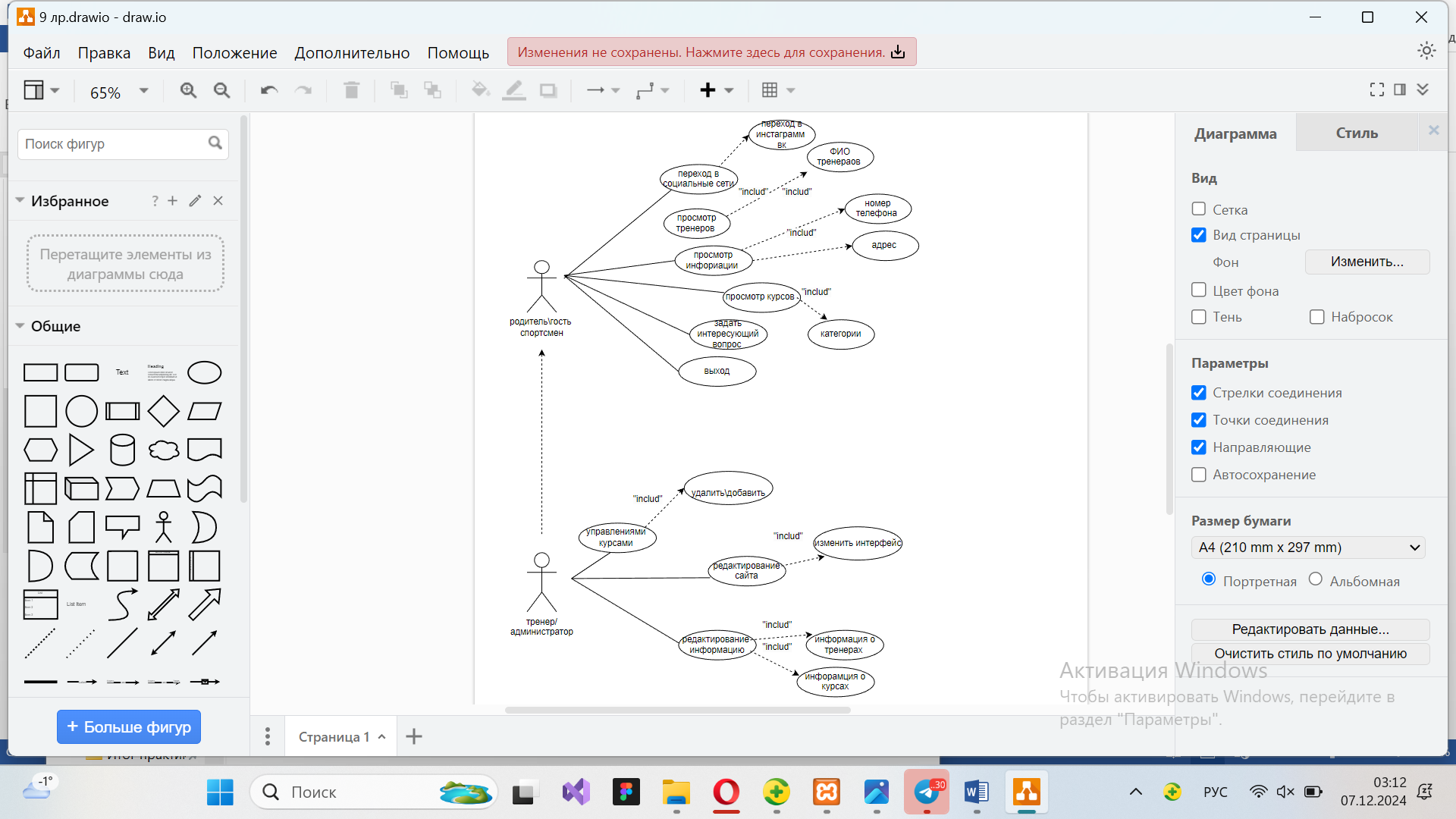


Рисунок 1 – Диаграмма вариантов использования

**1.3 Разработка плана работы над проектом**

Диаграмма Ганта — это популярный тип столбчатых диаграмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Является одним из методов планирования проектов. Используется в приложениях по управлению проектами.

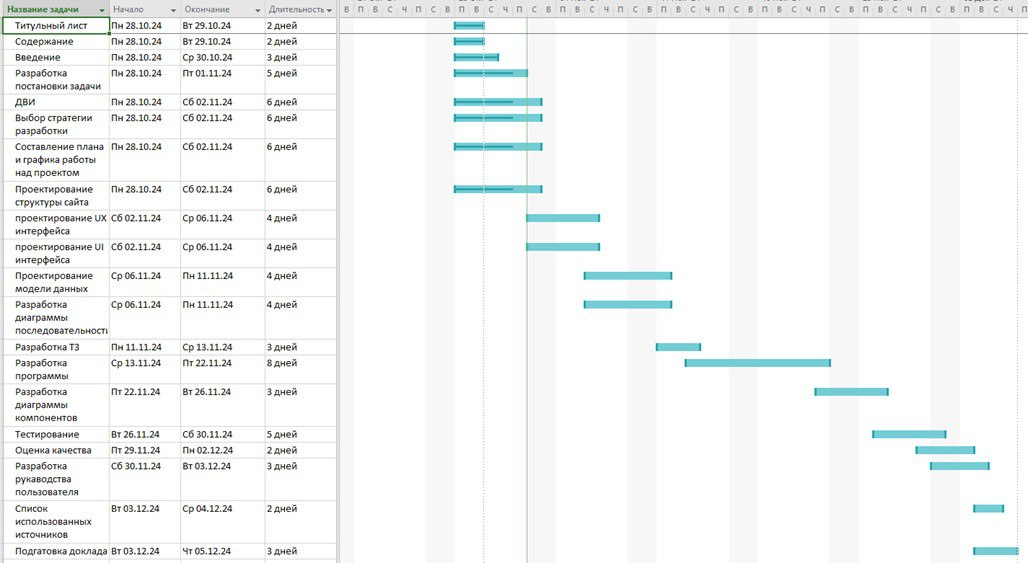
Диаграмма Ганта по разработке данного программного обеспечения представлена на рисунке 2.

Рисунок 2 – Диаграмма Ганта

**1.4 Выбор стратегии разработки и модели жизненного цикла**

Для разработки веб-ресурса следует выбрать стратегию разработки и модель жизненного цикла. Выбор модели жизненного цикла на основе характеристик требований находится в таблице 1.

Таблица 1 **–** Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории требований** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Да | Нет |
| Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Нет | Нет | Нет |
| Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Нет | Да | Да | Да |
| Требуется ли проверка концепции программного средства или системы? | Нет | Да | Да | Да | Нет | Да |
| Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Да | Да | Да | Да |
| Нужно ли реализовать основные требования на ранних этапах разработки? | Да | Да | Нет | Нет | Нет | Нет |

Вычисления: 4 за каскадную, 5 за V-образную, 7 за RAD, 4 за инкрементную, 2 за быстрого прототипирования и 3 за эволюционную.

Итог: на основе результатов заполнения таблицы 1, подходящей является RAD модель.

Таблица 2 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории команды разработчиков**  **проекта** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Нет | Нет | Да | Да | Да | Да |
| Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Нет | Нет | Нет |
| Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Да | Да | Да | Да |

Вычисления: 5 за каскадную, 5 за V-образную, 5 за RAD, 4 за инкрементную, 2 за быстрого прототипирования и 2 за эволюционную.

Итог: на основе результатов заполнения таблицы 2 подходящими являются каскадная и v-образная и RAD модели.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории коллектива пользователей** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Да | Нет | Нет | Нет |
| Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Нет | Да | Да | Да |
| Будет ли заказчик отслеживать ход выполнения проекта? | Да | Да | Да | Да | Да | Да |

Вычисления: 3 за каскадную, 3 за V-образную, 3 за RAD, 2 за инкрементную, 2 за быстрого прототипирования и 2 за эволюционную.

Итог: На основе результатов заполнения табл. 3 подходящей является каскадная, V-образная и RAD модели.

Таблица 4**–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории типов проекта и рисков** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **2** | **3** | **4** | **5** | **6** | **7** | **9** |
| Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 2 за каскадную, 3 за V-образную, 3 за RAD, 7 за инкрементную, 7 за быстрого прототипирования и 9 за эволюционную.

Итог: на основе результатов заполнения таблицы 4 подходящей является эволюционная модель.

Общий итог: Результаты заполнения табл. 3-6: каскадная – 14, V-образная – 16, RAD – 18, Инкрементная -17, Быстрого прототипирования – 13, Эволюционная – 16, в итоге заполнения табл. 1 – 4 наиболее подходящей является RAD модель.

**1.5 Инструменты разработки**

Инструменты, используемые при разработке и написании сопутствующей документации:

Figma – будет использоваться для создания UX/UI макетов проекта

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

Microsoft Office Word – для написания документации к программному продукту;

Wrike – это веб-ресурс для создания диаграммы Ганта;

Microsoft PowerPoint — программа подготовки презентаций и просмотра презентаций, являющаяся частью Microsoft Office;

Microsoft PowerPoint — программа подготовки презентаций и просмотра презентаций, являющаяся частью Microsoft Office;

WordPress – конструктор для сайта;

XAMPP Control Panel- это портативная программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий;

Веб-браузер Chrome - для нахождения информации и просмотра аналогов.

Разработка ведется на ноутбуке Asus ZenBook 14. У данного ноутбука следующие

параметры:

- процессор Core Ultra 5;

- объем ОЗУ 16 гб;

- оперативная система 16 ГБ

- объем места на HDD – 512 гб ;

- видеоподсистема NVIDIA GeForce RTX 3050 Ti + Intel UHD Graphics;

- ОС – Windows 11.

Данный программный продукт не очень требователен к аппаратным ресурсам.

**2. Проектирование задачи**

**2.1 Разработка структуры сайта**

Для успеха сайта очень важно его проектирование, так как оно влияет на то, как пользователи взаимодействуют с сайтом и насколько он эффективен. Качественный сайт должен быть простым в использовании, иметь удобную навигацию, быстро загружаться и удовлетворять запросы пользователей.

Структура сайта будет представлена в приложении А.

**2.2 Разработка UML-диаграмм**

**2.2.1 Диаграмма деятельности**

Диаграмма деятельности – это UML - диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Диаграммы деятельности предоставляют графическое представление процессов, что позволяет команде разработки и заинтересованным сторонам легко понять, как осуществляется работа. Это особенно полезно для сложных процессов, где трудно отслеживать последовательность действий.

Диаграмма деятельности будет представлена в приложении Б. В данной диаграмме представлены функции пользователя в личном кабинете.

**2.2.2 Диаграмма последовательности**

Диаграмма последовательности — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта и взаимодействие актеров информационной системы в рамках прецедента.

**Диаграммы последовательностей используются для уточнения диаграмм прецедентов**, более детального описания логики сценариев использования. Это отличное средство документирования проекта с точки зрения сценариев использования.

Диаграммы последовательностей обычно содержат **объекты**, которые **взаимодействуют в рамках сценария**, **сообщения**, которыми они обмениваются, и **возвращаемые результаты**, связанные с сообщениями. Впрочем, часто возвращаемые результаты обозначают лишь в том случае, если это не очевидно из контекста.

**Объекты** обозначаются прямоугольниками с подчеркнутыми именами (чтобы отличить их от классов).

**Сообщения (вызовы методов)** - линиями со стрелками.

**Возвращаемые результаты** - пунктирными линиями со стрелками.

Прямоугольники на вертикальных линиях под каждым из объектов показывают «время жизни» (фокус) объектов. Впрочем, довольно часто их не изображают на диаграмме, все это зависит от индивидуального стиля проектирования.

Диаграмма последовательности будет представлена в приложении В. На данной диаграмме показан процесс.

**2.2.3 Модель данных**

Модель данных — это абстракция, которая описывает структуру, взаимосвязи, ограничения и правила работы с данными в системе или приложении. Она служит основой для проектирования баз данных и играет ключевую роль в различных этапах разработки системы.

Модель данных будет представлена в приложении Г. На данной диаграмме показана база данных проекта.

**2.3 Разработка пользовательского интерфейса**

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемого интернет-ресурса.

Для разработки визуального дизайна использовались сдержанные, мягкие цвета для удобства использования программного продукта.

В ходе разработки был спроектирован дизайн Разработка электронного средства обучения, предназначенного для обучения детей и подростков футболу для «Doublekik».

На прототипах для данного проекта показаны такие страницы как:

- Главная страница

- о нас

- тренера

- блог о футболе

-доп.информация

-записаться на тренировку

Прототипы для мобильной версии веб-сайта:

- Главная страница

**UX-прототипы**

UX-прототип — это модель продукта, а не полноценный дизайн-макет. В нем важна прежде всего структура и контент (содержание), а не дизайн и красота (форма). На этапе прототипирования сайта и проектировщику, и заказчику важно сконцентрироваться на пользовательском опыте, юзабилити.

UX-прототипы (прототипы пользовательского опыта) представляют собой интерактивные модели или наброски интерфейсов, используемые для визуализации и тестирования дизайна программных продуктов перед их полной разработкой. Они помогают дизайнерам и разработчикам понять, как пользователи взаимодействуют с продуктом, выявить потенциальные проблемы и улучшить общий пользовательский опыт.

Для данного веб-сайта показаны примеры главной страницы.

Разработанные прототипы для данного веб-сайта можно просмотреть по ссылке:

https://www.figma.com/design/kv0GLXd3ClmzRfJMjQP5Sd/Untitled?node-id=0-1&t=GEH4sbWI6GURrJ6q-1

С прототипами UX вы можете ознакомиться в приложении Д.

**2.3.2 UI-прототипы**

UI прототипы (прототипы пользовательского интерфейса) — это визуальные представления экранов, элементов и взаимодействий для цифровых продуктов, таких как веб-сайты и мобильные приложения. Они помогают дизайнерам, разработчикам и заинтересованным сторонам визуализировать, как будет выглядеть конечный продукт, и проверять его функциональность до этапа окончательной разработки. UI-прототипы играют важную роль в дизайне пользовательских интерфейсов, поскольку позволяют визуализировать и тестировать идеи до их реализации. Процесс прототипирования способствует более эффективной разработке и улучшению качества конечного продукта. Использование соответствующих инструментов и методов поможет командам создать успешные и удобные интерфейсы для пользователей. Если у вас есть дополнительные вопросы или нужна помощь, не стесняйтесь спрашивать!

При разработке прототипа UI для главной страницы сайта использовались такие шрифты как: ISOCPEUR размер 64 для меню сайта, для какой-либо информации ISOCPEUR размер 80, для подвала сайта ISOCPEUR размер 70

Разработанные прототипы для данного веб-сайта можно просмотреть по ссылке:

https://www.figma.com/design/kv0GLXd3ClmzRfJMjQP5Sd/Untitled?node-id=0-1&t=GEH4sbWI6GURrJ6q-1

С прототипом UI вы можете ознакомиться в приложении Е.

**3. Реализация**

**3.1 Руководство программиста**

Программный продукт разработан с использованием конструктора WordPress 6.7.1. WordPress предлагает интуитивно понятный интерфейс, что делает его доступным даже для пользователей без технических навыков. Установка и управление контентом не требуют глубоких знаний программирования. Существуют тысячи бесплатных и платных тем и плагинов, которые позволяют настраивать внешний вид сайта и добавлять функциональность. Темы определяют дизайн, а плагины расширяют возможности.

**3.1.1 Создание сайта**

Для того чтобы создать свой веб-ресурс в WordPress необходимо сначала зарегистрироваться, создать свою учетную запись и подключить базу данных. Для создания своей учетной записи вам необходимо зайти на сайт phpMyAdmin. Чтобы подключить базу данных достаточно придумать ее название (doublekik) и пароль (doublekik). При регистрации в самом WordPress также надо ввести имя пользователя и придумать ему пароль.

Данные для входа на сайт для администратора:

- Имя пользователя: doublekik;

- пароль: doublekik.

После ввода этих данных вы перейдете в административную панель WordPress, где сможете настраивать свой сайт, устанавливать темы и плагины, а также добавлять контент. Для создания сайта я выбрала тему «Neve».

Рисунок 3 - Тема «Neve»

**3.1.2 Настройка функциональности и дизайна**

Все элементы, видимые на экране, можно изменить, переместить или удалить. При нажатии на кнопку «Настроить» вверху экрана появляется панель инструментов редактора, которая состоит из 10 разделов:

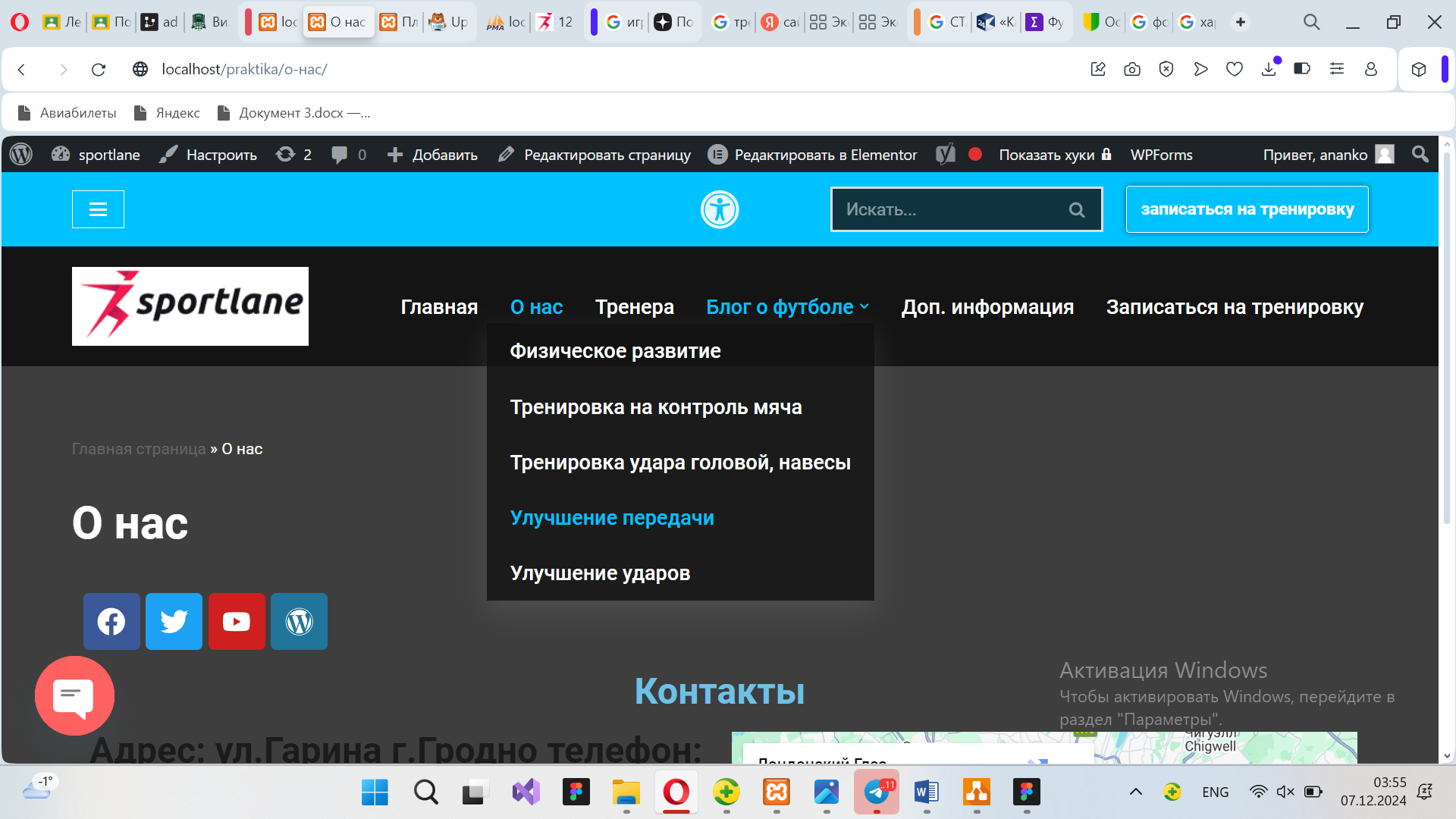
1. «Настройки темы» - этот раздел позволяет вам изменять общие настройки темы, такие как основные функции, макеты страниц и доступные опции кастомизации.
2. «Свойства сайта» - здесь вы можете задать название сайта, описание (теглайн) и иконку (favicon).
3. «Цвета» - в этом разделе вы можете устанавливать основные цветовые схемы для вашего сайта, включая цвета текста, заголовков, фона и ссылок.
4. «Изображение заголовка» - вы можете загружать и устанавливать изображение заголовка, которое отображается в верхней части страницы.
5. «Фоновое изображение» - этот раздел позволяет устанавливать фон для вашего сайта, включая фоновое изображение или цвет.
6. «Меню» - здесь вы можете создать и управлять меню навигации вашего сайта. Вы можете добавлять страницы, категории и произвольные ссылки в меню, а также задавать их порядок.
7. «Виджеты» - этот раздел позволяет добавлять и настраивать виджеты, которые можно разместить в областях виджетов вашей темы — боковых панелях, подвалах и других местах.
8. «Настройки главной страницы» - вы можете выбрать, какую страницу отображать как главную: статическую страницу (по вашему выбору) или страницу с записями блога. Это позволяет настроить главную страницу в соответствии с потребностями вашего сайта.
9. «Дополнительные стили» - здесь вы можете добавлять свои собственные CSS-стили для кастомизации внешнего вида вашего сайта. Это полезно для более точной настройки дизайна, если стандартные опции вам не позволяют достичь желаемого результата.

**3.1.3 Редактирование страниц**

Первым делом, после открытия шаблона(темы) для будущего веб-ресурса, нужно изменить его название. Для этого необходимо войти в админ панель сайта, на панели инструментов слева найдите раздел «Настройки» и выберите «Общие». На странице "Общие настройки" вы увидите два поля, где нужно изменить название сайта.

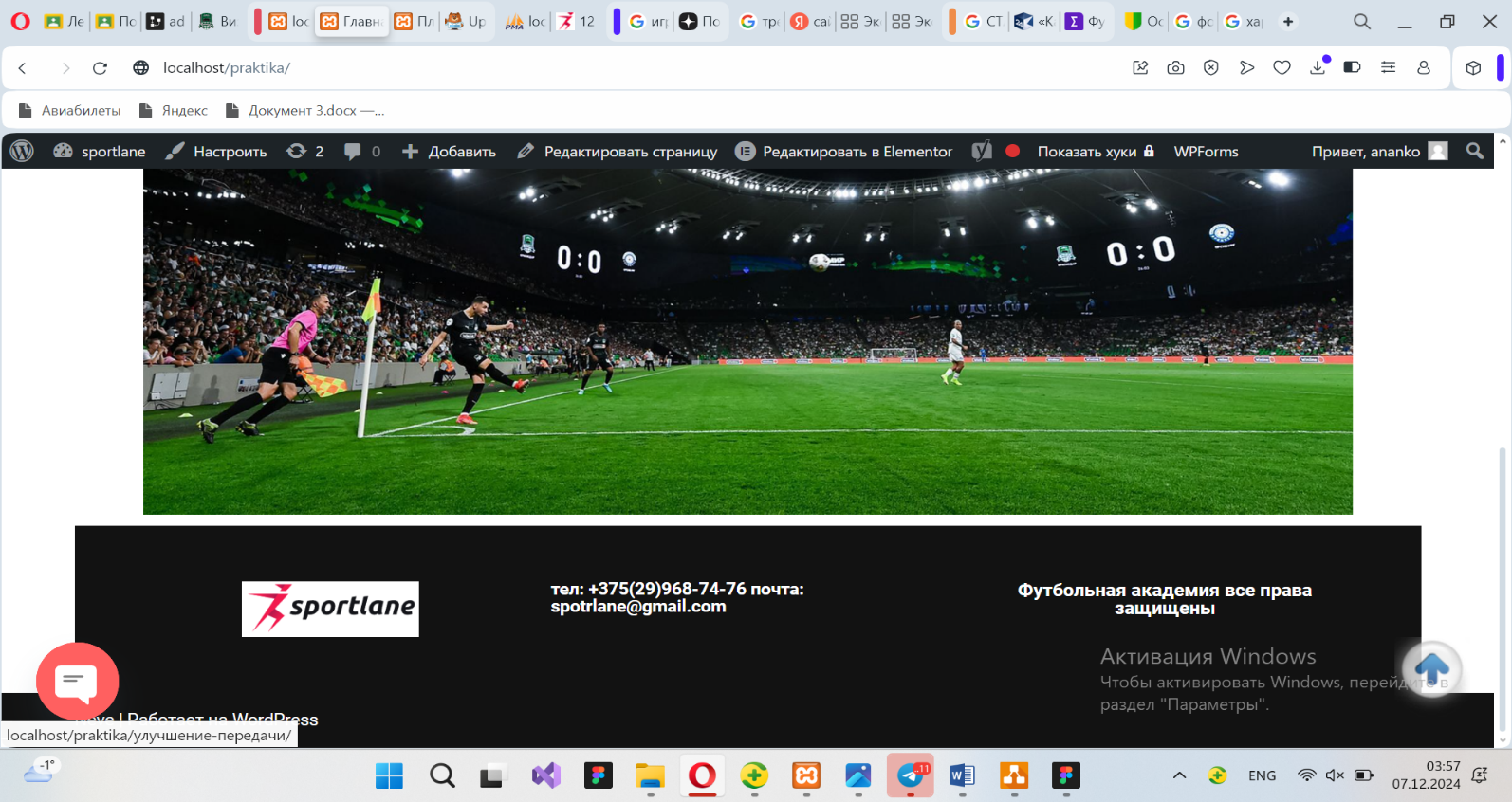
Теперь следует разобраться с настройкой меню сайта. Тема «» имеет ненужные страницы, поэтому оставляем только главную. Далее чтобы создать остальные разделы перейдите в раздел "Внешний вид": В левой боковой панели выберите "Внешний вид", а затем нажмите на "Меню". Создайте новое меню (введите его название, нажмите «Создать меню»). Добавьте элементы меню: выберите страницы, записи, категории или пользовательские ссылки, которые вы хотите добавить в меню, из ресурсов, представленных слева, установите галочки рядом с элементами, которые хотите добавить, и нажмите "Добавить в меню".

Рисунок 4 - Меню сайта



Переходим к редактированию страниц сайта. Для начала нужно настроить подвал сайта и хедер. Подвал данного сайта будет содержать логотип сайта, соц.сети для связи, режим работы магазина, краткое описание оплаты при заказе.

Рисунок 5 - подвал сайта



Хедер данного сайта будет содержать логотип, горизонтальное меню сайта. Хедер редактировался с помощью ElementsKit.

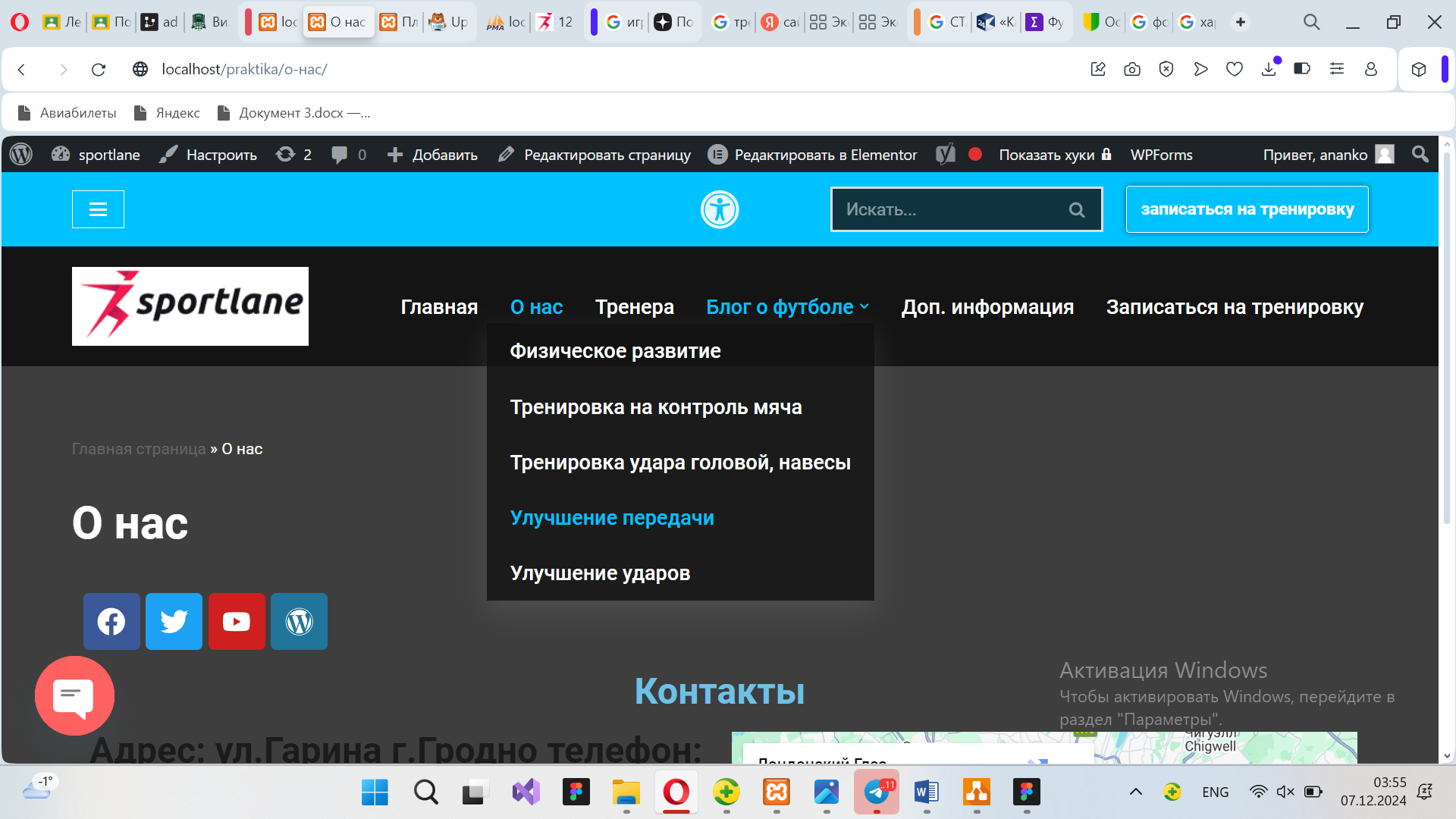


Рисунок 6 - Хедер сайта

Далее переходим к настройке главной страницы. На главной размещен паттерн.

Также вся информация, которая вставлена , например это информация «О нас». Чтобы добавить какую-то информацию необходимо зайти на нужную страницу, и вверху экрана нажать «Редактировать страницу».

****

Рисунок 7 - Редактировать

После этого в левой части экрана появится набор элементов, которые можно добавить и после их редактировать как нужно.

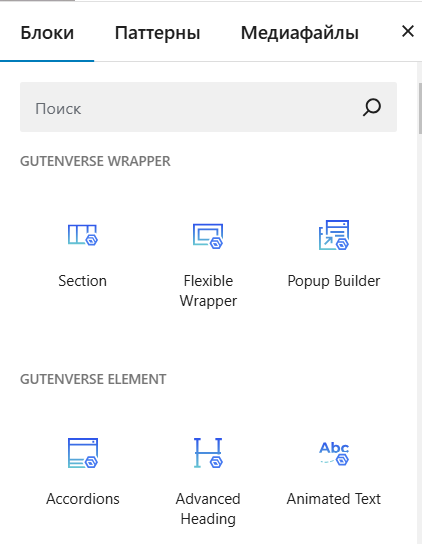


Рисунок 8 - Набор элементов

**4 Тестирование**

**4.1 Тесты на использование**

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

Разработанные тест-кейсы и статус их выполнения представлены в приложение И.

Расписание работ над проектом представлено в таблице 5.

По результатам тестирования все элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах. Таким образом, программный продукт можно использовать, не испытывая особых проблем или неудобств, связанных с взаимодействием с программным продуктом.

**5 Применение**

**5.1 Общие сведения о программном продукте.**

Цель данного проекта – предоставить удобную платформу для детей и подростков, где пользователи смогут находить необходимую информацию, получать советы и делиться опытом. В процессе работы мы столкнулись с различными задачами, решения которых способствовали не только развитию проекта, но и углублению наших знаний в области веб-разработки.

Создаваемый web-ресурс будет рассчитан для профессиональных и начинающих спортсменов. Применить данный сайт смогут люди, заинтересовавшиеся в данной теме.

**5.2 Руководство пользователя**

**5.2.1 Руководство пользователя для гостя**

Для того, чтобы открыть сайт необходимо перейти по ссылке. Затем достаточно следовать приведенной инструкции установки приложения. После открытия сайта нас встречает главная страница, где предоставлены все товары находящиеся на сайте.

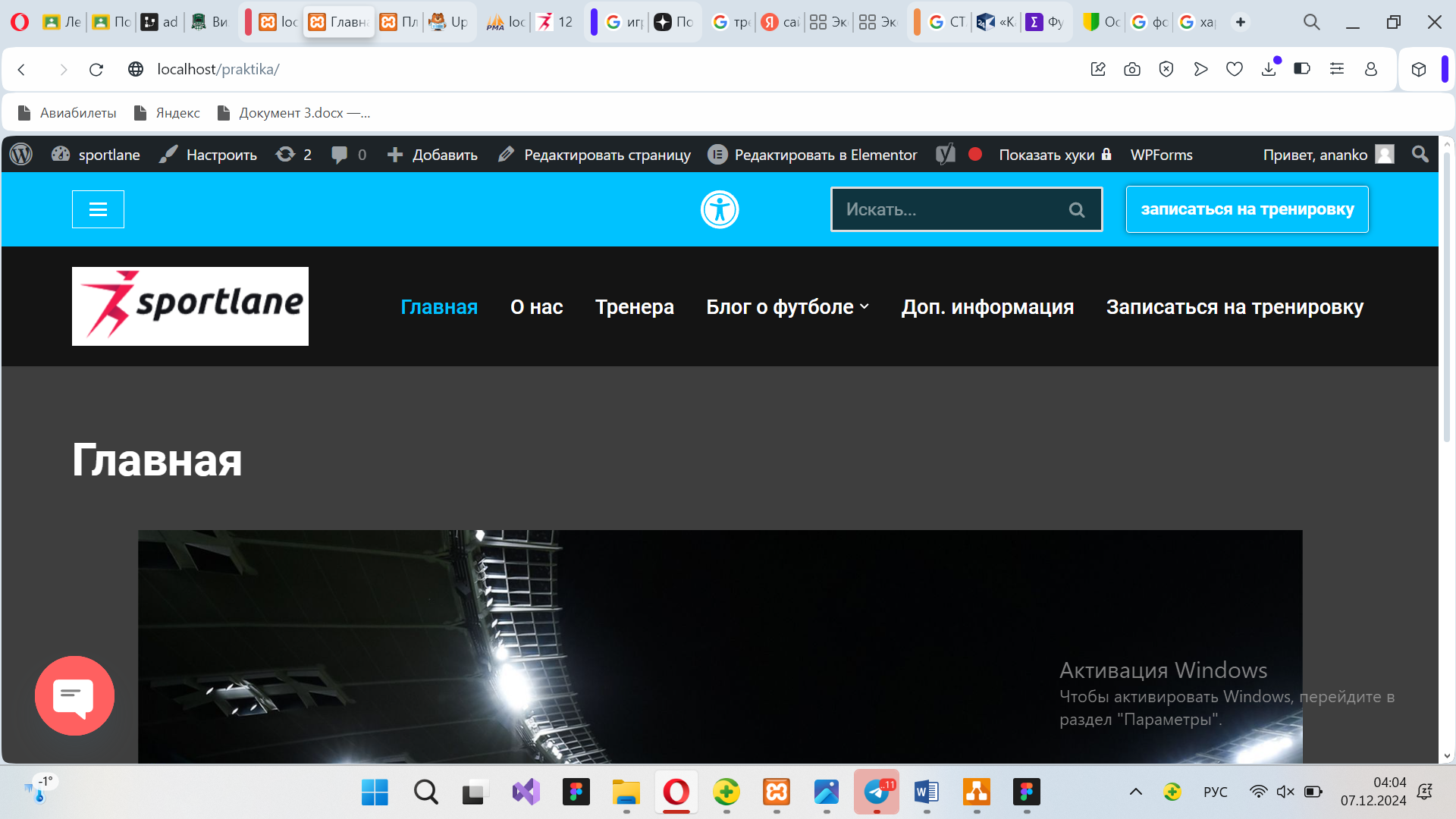


Рисунок 9 - Главная страница

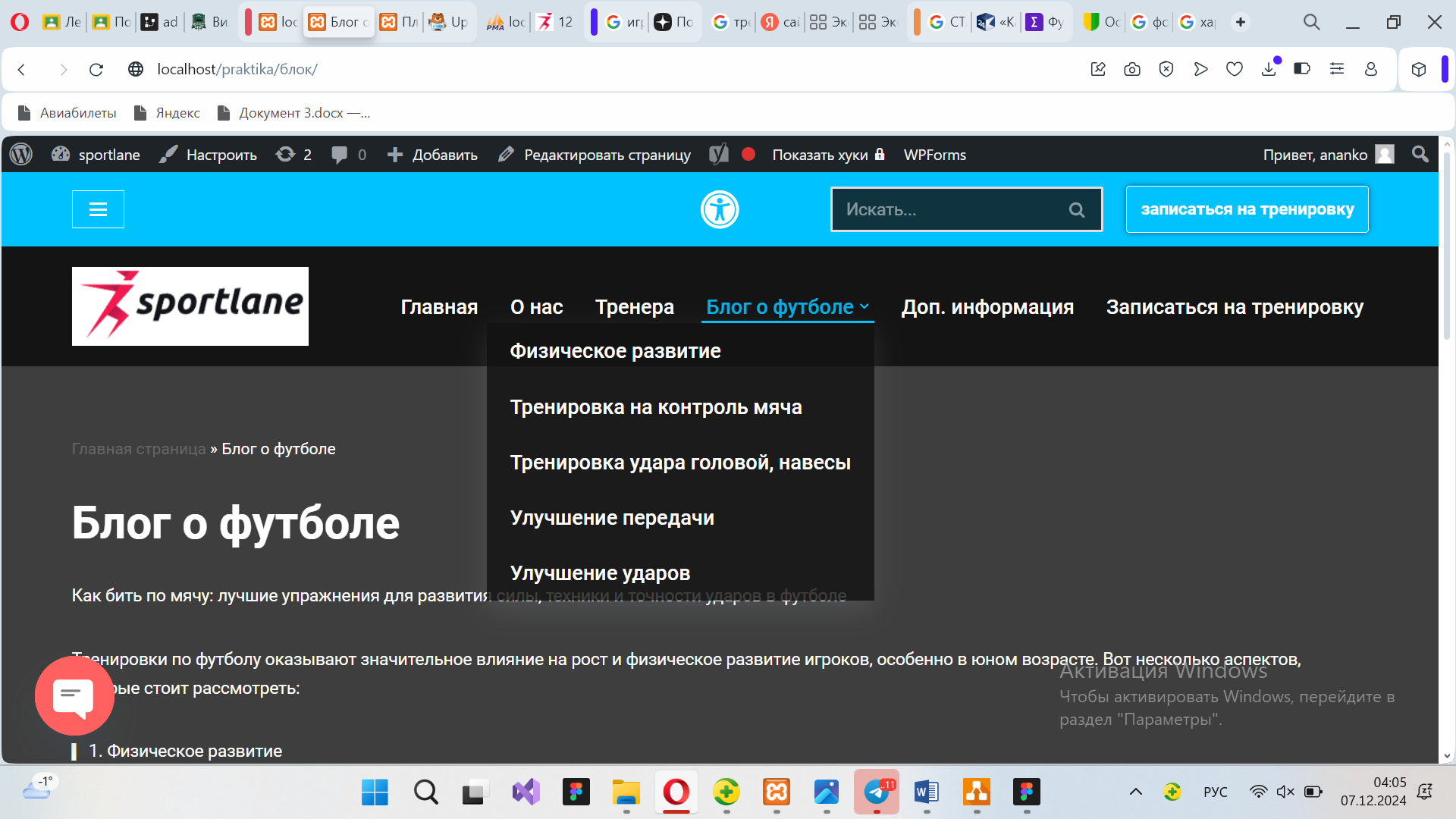


Рисунок 10 – Описания тренировок

После прочитанной информации можно записаться на тренировку.

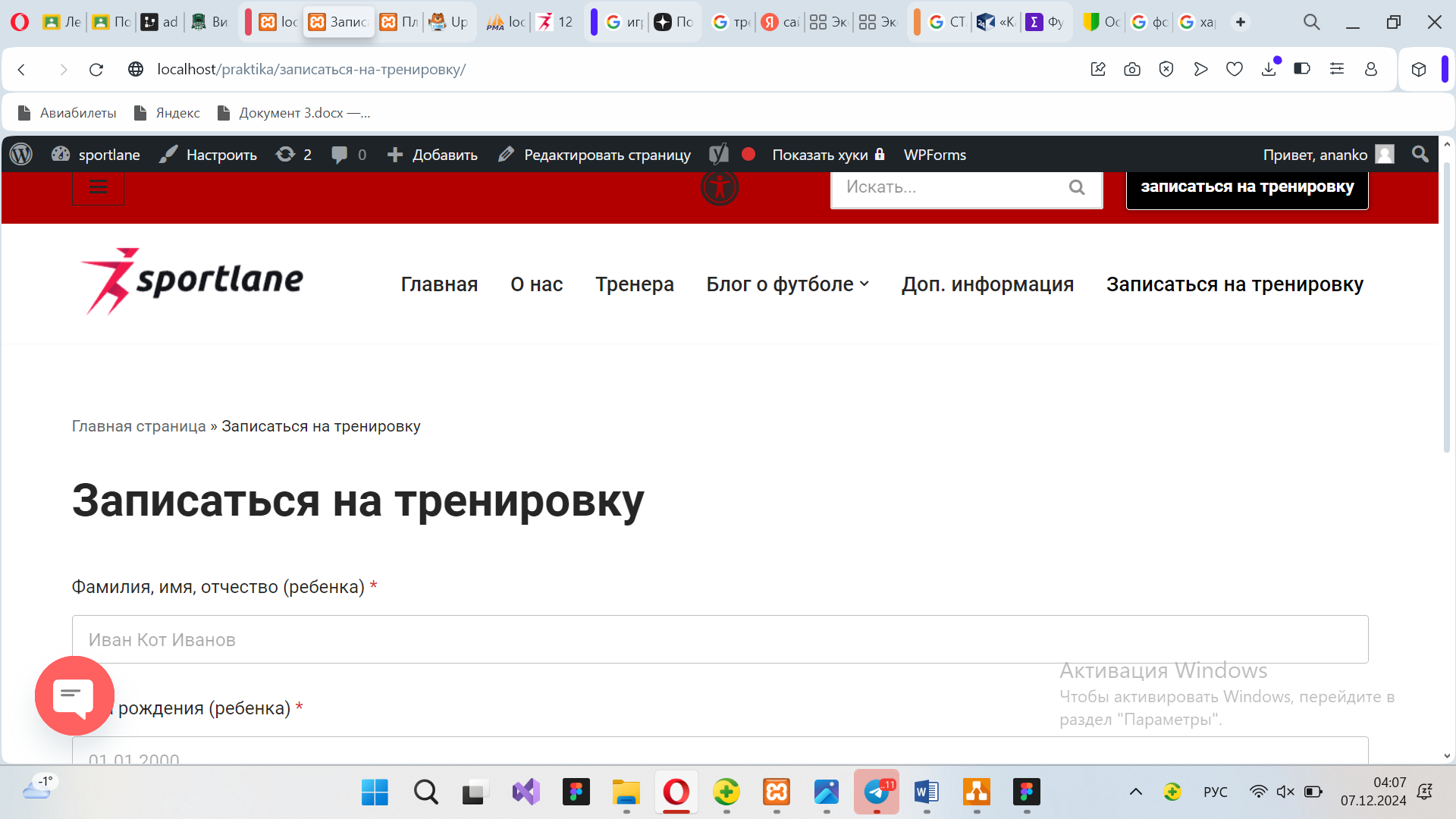
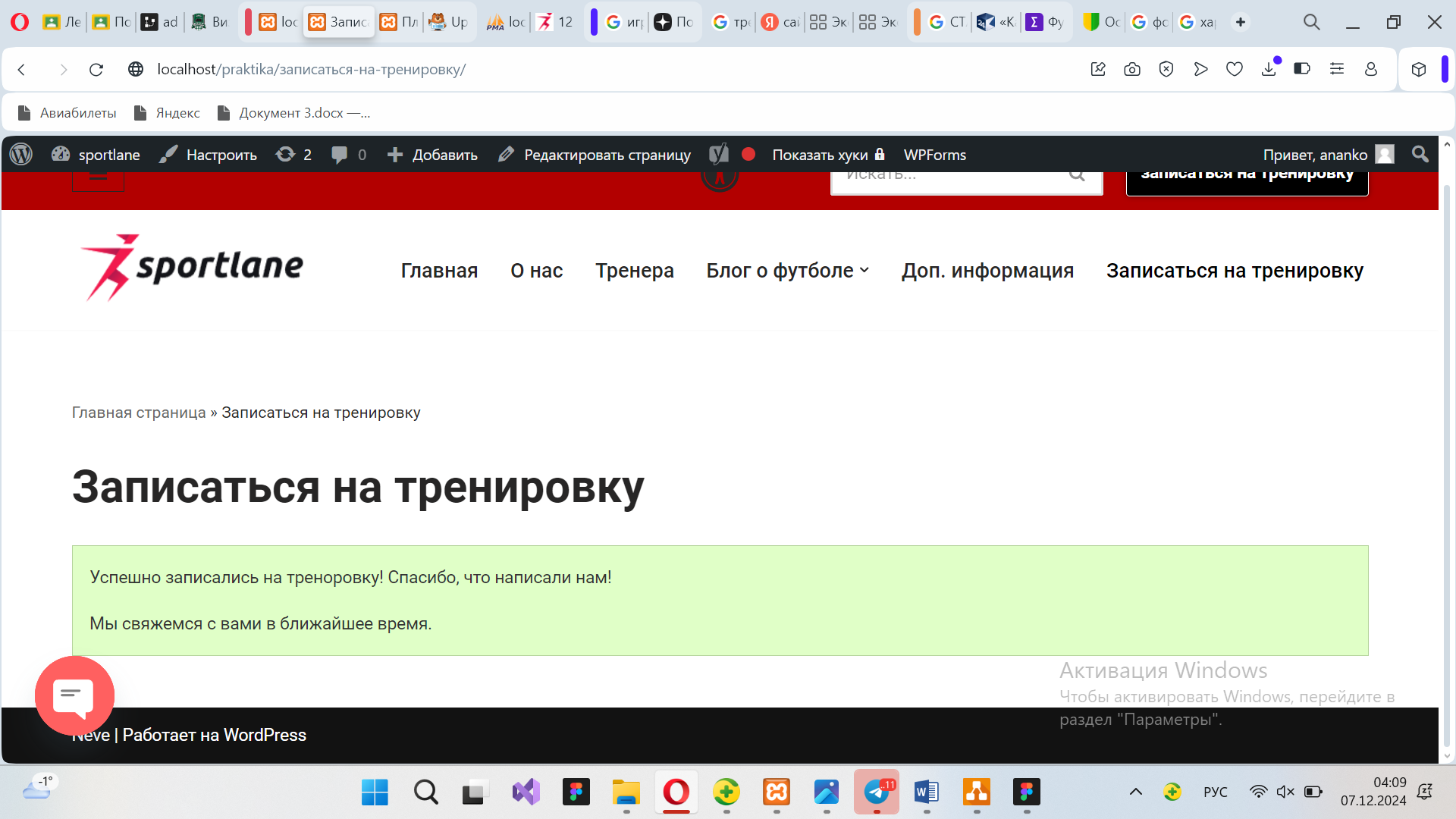


Рисунок 11 – запись на тренировку



**5.2.3 Руководство пользователя для администратора**

Для использования функционала администратора необходимо зайди в админ панел WordPress

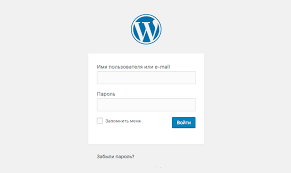


Рисунок 13 – Вход в админ панель

После входа в админ панель админ сможет редактировать сайт с помощью инструментов

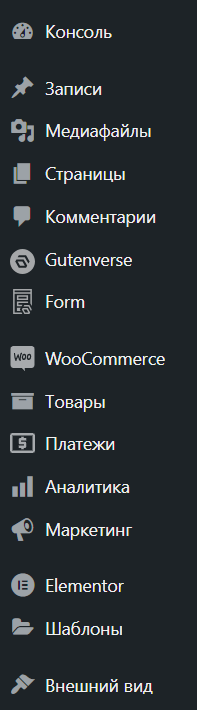


Рисунок 14 – Инструменты

**Заключение**

Создание сайта на платформе WordPress — это отличный выбор для многих пользователей, благодаря простоте использования, гибкости и расширяемости. WordPress предоставляет интуитивно понятный интерфейс и широкий выбор тем и плагинов, что позволяет адаптировать сайт под любые нужды и предпочтения.

В процессе разработки важно уделить внимание не только дизайну, но и функциональности, а также обеспечению безопасности и оптимизации для поисковых систем. Регулярное обновление контента и поддержка актуальности сайта способствуют его успешному продвижению и привлечению аудитории.

Кроме того, сообщество WordPress активно делится знаниями и опытом, что делает поиск решений для возникающих проблем более доступным. Используя данную платформу, вы получаете возможность создавать как простые блоги, так и сложные корпоративные сайты, что подтверждает универсальность WordPress. В итоге, инвестиции в создание сайта на WordPress могут стать надежным фундаментом для вашего онлайн-присутствия и бизнеса в целом.

**Список используемых источников**

1. <https://twins.by/>
2. https://fckrasnodar.ru/club/card/