МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Интеллектуальные информационные технологии»

«К защите допускаю» Заведующий кафедрой

В.А. Головко

« » 2024 г.

РАЗРАБОТКА ВЕБ-ПЛАТФОРМЫ ДЛЯ ЭФФЕКТИВНЫХ ЗАНЯТИЙ В СПОРТИВНОМ ЗАЛЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ

ДП.АС59.200067- 05 81 00

Листов 55

|  |  |
| --- | --- |
| Руководитель | А.М. Соловчук |
| Выполнил | А.С. Сахацкий |
| Консультанты: |  |
| по экономическому разделу | Н.П. Четырбок |
| по ЕСПД | Е.В. Булей |
| Рецензент | Л.В. Лизун |

2024

## АННОТАЦИЯ

55 с./59 с., 38 рис., 9 табл., 5 ист. лит., 1 прил., 5 л. граф. матер.

Данный дипломный проект посвящен разработке веб-платформы для эффективных занятий в спортивном зале. Система представляет собой веб- приложение, которое работает на устройствах с наличием современного браузера и доступа в интернет.

Дипломный проект включает в себя введение и пять разделов. В первом разделе выполняется постановка задачи и проводится системный анализ. Второй раздел содержит данные о проектировании системы. В третьем разделе описывает- ся реализация системы, а также описывается этапы ее тестирования. В четвертом разделе производится технико-экономическое обоснование производимой работы. В пятом разделе производятся энерго- и русурсосбережение производимой работы.

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Факультет |  | ЭИС | Кафедра | ИИТ |
| УТВЕРЖДАЮ | | | | |
| Зав. кафедрой | | | | |
|  |  | (подпись) | |  |
| « » |  | 2024 г. |  |  |
| ЗАДАНИЕ | | | | |
| по дипломному проектированию | | | | |
| Студенту | Сахацкому Александру Сергеевичу | | | |
| 1. Тема проекта | | Разработка веб-платформы для эффективных тренировок в спортивном зале | | |
| (Утверждена приказом по вузу от | | | № | ) |
| 2. Сроки сдачи студентом законченного проекта | | | | |
| 3. Исходные данные к проекту | | | | |
| Особенности разработки: - язык программирования: Java, JavaScript; - фреймворк: | | | | |
| Spring Boot, Spring Security, Spring Boot Mail, - язык разметки: Html, Freemarker, - язык стилизации: CSS. | | | | |
|  | | | | |
|  | | | | |
| Платформа должна обеспечивать: | | | | |
| - удобная административная панель; | | | | |
| - возможность авторизации администратора; | | | | |
| - отправка своих данных клиентом. | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| Необходимо разработать веб-платформу с необходимым функционалом и интуитивно понятным | | | | |
| пользовательским интерфейсом. | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| 4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов) | | | | |
| Введение |  |  |  |  |
| 1. Системный анализ и постановка задачи | | | | |
| 2. Проектирование системы | | | | |
| 3. Реализация и тестирование веб-платформы | | | | |
| 1. Технико-экономическое обоснование 2. Энерго- и ресурсосбережение | | | | |
| Заключение | | | | |
| Список сокращений | | | | |
| Список использованных источников | | | | |
| Приложение А. Текст программы | | | | |

|  |  |
| --- | --- |
| 5. Перечень графического материала (с точным указанием обязательных чертежей и | |
| графиков) | |
| 1. Постановка задачи (плакат – формат A1) | |
| 2. Структура личного кабинета (плакат – формат A1) | |
| 3. Результаты тестирования (плакат – формат А 1) | |
| 4. Схема алгоритма программы (чертеж – формат А1) | |
| 5. Схема алгоритма авторизации администратора (чертеж – формат А1) | |
|  | |
|  | |
|  | |
|  | |
|  | |
| 6. Консультанты по проекту (с указанием относящихся к ним разделов проекта) | |
| ст. преподаватель Соловчук А.М. – по основным разделам | |
| ассистент Булей Е.В. – нормоконтроль | |
| доцент Четырбок Н.П. – экономический раздел | |
|  | |
| 7. Дата выдачи задания | 20.04.2024 г. |
| 8. Календарный график работы над проектом на весь период проектирования (с указанием | |
| сроков выполнения и трудоемкость отдельных этапов) | |
|  | |
| Раздел 1: 21.04 – 27.04: 20% | |
| Раздел 2: 28.04 – 18.05: 30% | |
| Раздел 3: 19.05 – 31.05: 30% | |
| Раздел 4, 5: 01.06 – 6.06: 10% | |
| Оформление проекта: 7.06 – 10.05: 10% | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | Руководитель |
|  | (подпись) |
| Задание принял к исполнению (дата) | |
| (подпись студента) | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **СОДЕРЖАНИЕ**  **ВВЕДЕНИЕ** **5**   1. **СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ** **6**    1. Характеристика предметной области 6    2. Выбор инструментов и методов реализации 7    3. Фреймворк Spring Boot 10    4. Постановка задачи 12 2. **ПРОЕКТИРОВАНИЕ СИСТЕМЫ** **14**    1. Структура и архитектура веб-платформы 14    2. Архитектура сервера 16    3. Архитектура сервиса 17    4. Создание макета веб-платформы 18    5. Проектирование базы данных 25 3. **РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ВЕБ-ПЛАТФОРМЫ** **30**    1. Реализация клиентской части 30    2. Тестирование веб-платформы 39 4. **ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ** **43**    1. Исходные данные для расчета экономического эффекта 43    2. Расчётобъема функций программного модуля 43    3. Расчёт полной себестоимости программного модуля 45    4. Расчёт цены и прибыли по программному продукту 49 5. **ЭНЕРГО- И РЕСУРСОСБЕРЕЖЕНИЕ** **50**    1. Энергосбережение и государственное регулирование 50    2. Ресурсосбережение и государственные требования к ресурсосбережению 50   **ЗАКЛЮЧЕНИЕ** **52**  **СПИСОК СОКРАЩЕНИЙ** **53**  **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** **54**  **ПРИЛОЖЕНИЕ А ТЕКСТ ПРОГРАММЫ** | | | | | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | | | | | |
|  |  |  |  |  |
| *Изм* | *Лист* | *№ докум.* | *Подп.* | *Дата* |
| *Разраб.* | | *Сахацкий А.С.* |  |  | *Разработка веб – платформы для эффективных занятий в спортивном зале* | *Лит* | | | *Лист* | *Листов* |
| *Проверил* | | *Соловчук А.М.* |  |  | *Д* |  |  | *4* | *55* |
|  | |  |  |  | *БрГТУ* | | | | |
| *Н контр.* | | *Булей Е.В.* |  |  |
| *Утв.* | | *Головко В.А.* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ВВЕДЕНИЕ**  На данный момент многие спорт-комплексы и спортивные организации, предлагающие услуги в спортивных залах, ищут единую веб-платформу, которая объединила бы всех заинтересованных в обласи спорта.  Веб-платформа – это программное решение, которое устанавливает связь между сервером и клиентом для передачи данных по сети. Пользовательский интерфейс обрабатывается на стороне клиента, а серверная часть занимается приемом запросов, выполнением различных оперций и формирования контена для веб-страниц, которые потом отправляются клиенту с использованием различных протоколов, в основном HTTP протоколе.  Веб-платформы имеют несколько особенностей:   * работают независимо от операционной системы клиента, но требуют кроссбраузерности из-за различных стандартов в каждом браузере, а именно: HTML, CSS; * тестирование и обслуживание веб-платформы оптимальней чем обслуживание обычных приложений; * веб-платформы способны выдерживать большое количество обращений пользователей.   Объектом исследования дипломной работы является построение веб-платформы с удобным и интуективно понятным интерфейсом для пользователя и для администратора.  Предмет исследования дипломной работы – разработка веб-платформы для эффективных тренировок в спортивном зале.  Целью дипломного проекта является разработка веб-платформы для эффективных тренировок в спортивном зале. Веб-платформа идет уже настроена для работы администратором. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *5* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **СИСТЕМНЫЙ АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ**    1. **Характеристика предметной области**   Все информационные ресурсы доступные в интернете хранятся на веб-серверах. Веб-сервер относится как к аппаратной начинке, так и к программному обеспечению. С точки зрения аппаратного обеспечения, веб-сервер предоставляет собой компьютер, который хранит содержимое сайта, такик как HTML-документы и CSS-стили, и доставляет их на устройства конечных пользователей, такие как веб-браузеры. Этот компьютер подключен к Интернету и доступен по доменному имени. С точки зрения программного обеспечения, веб-сервер состоит из различных компонентов, которые управляют доступом пользователей к размещенным на сервере файлам, например HTTP- сервер. HTTP-сервер – это часть программного обеспечения, которая понимает URL- адреса и HTTP протоколы.  Веб-платформа состоит из одной или нескольких веб-страниц, которые состоят из текстовых файлов с расширением \*.html или \*.ftlh. Эти файлы содержат HTML- код или Freemaker шаблон, который определяет внешний вид и структуру отображаемой информации. Графические, аудио и видео файлы, не входят в состав веб-страницы, но могут быть включены через ссылки в HTML-коде.  Для хранения информации использую БД либо JSON файлы. В зависимости от назначения и скорости доступа к файлам используется определенный вариант:   * + - Если требуется передать маленькое количество информации или сохранить на не долгий срок информацию использую JSON файлы. JSON это стандартный формат текстового представления структурированных данных;     - Если требуется сохранить информацию на долгое время то использутся реляционная база данных. Реляционная база данных – это тип базы данных, организованный на реляционной модели данных. В такой базе данных информация предоставлена в виде таблиц, где каждая таблица состоит из строк и столбцов. Строка это запись, столбец это поле. Отношение между таблицами устанавливаются с помощью ключе, что позволяе эффективно организовывать и обрабатывать данные, а также выполнять операции объединения и запросы на выборку данных. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *6* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * 1. **Выбор инструментов и методов реализации**   Выбор инструментов и методов для создания веб-платформы зависит от требования к проекту.Существует множество платформ и инструментов для создания макета сайта.  Некоторые из наиболее популярных включают в себя:   * + - Adobe Photoshop – это известный инструмент для создания веб-дизайна. Он позволяет создавать макеты сайтов, редактировать изображения. Минусом данной программы это долгое проектирование дизайна.     - Canva – это онлайн-платформа для создания веб-дизайна, которая включает в себя множество готовых решений и макетов, фигур и иконок. Макет с данного сайта можно импортировать в различные форматы, такие как: \*.jpg, \*.png,   \*.pdf. При использовании Pro версии, можно сохранить в формате \*.psd.   * + - Figma – это онлайн-инструмент, предназначенный для создания дизайна и прототипа веб-платформы. Он имеет множество готовых вариантов. Данная программа самая популярная, тк имеет поддержу почти всех разрешений.   Выбор инструмента для создания макета сайта зависит от ваших потребностей и опыта веб-разработки. Кроме того, многие из этих инструментов могут быть комбинированы, чтобы создавать более сложные проекты.  Для своего веб-приложения я буду использовать платформу Canva.  Canva – это веб-приложение для графического дизайна, которая позволяет пользователям создавать различные визуальные материалы, такие как презентации, постеры, баннеры, инфографику и многое другое. Это качественный инструмент для создания дизайна и прототипирования веб-платформ, которая позволяет легко и быстро воплощать свою идею.  При разработке веб-платформ требуются различные языки: языки верстки и фреймворки, также база данных. Ниже я перечислил языки и технологии, которые я использовал при разработке своей веб-платформы:  **Java** – это объектно-ориентированный язык программиования, разработанный компанией Sun Microsystems, далее компания Oracle. Данный язык один из самых популярных, т.к. он применяется в различных областях разработки: мобильные приложения Android, игры, серверные финансовые приложения, веб-приложения, встраиваемые системы. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *7* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Вот основные характеристики и возможности языка Java:   * Объектно-ориентированный подход: Java полностью объектно- ориентированный язык, это означет, что все элементы являются объектами, а программа состоит из взаимодействия между объектами. Это помогает создавать модульный и гибкий код. * Платформенная независимость: Одна из ключевых особенностей Java является ее платформенная независимость. Код Java компилируется в байт- код, который может быть запущен на любой платформе, поддерживающей виртуальную машину Java (Java Virtual Machine - JVM). Это делает Java идеальным выбором для создания кросс-платформенных приложений. * Безопасность: Java обеспечивает высокий уровень безопасности благодаря механизмам, таким как контроль доступа, проверка типов во время выполнения и многое другое. Это делает Java предпочтительным выбором для разработки прилодений, требующих высокого уровня защиты данных. * Многопоточность: Java поддерживает многопоточность, что позволяет создавать параллельные и асинхронные приложения. Это особенно важно для разработки высокопроизводительных и масштабируемых приложений, таких как серверы и многозадачные системы.   Java-код пишется в текстовом редакторе или в IDE и сохраняется в файле с расширением .java.  **Spring** – это фреймворк для разработки приложений на языке Java. Он предоставляет обширный набор инструментов и функциональных возможностей, которые помогают упростить процесс создания сложных и масштабируемых приложений. Spring широко используется в корпоративной разработке для создания веб- приложений, микросервисов, обработки данных и многое другое.  **Spring Boot** – это проект в рамках Spring Framework, который предназначен для упрощения процесса создания и настройки приложений на основе Spring. Он предоставляет множество автоматических конфигураций, встроенных зависимостей и удобных инструментов, что позволяет быстро создавать самостоятельные, автономные веб-платформы с минимальным усилием. Sprin Boot позволяет разработчикам сосредотточиться на бизнес логике приложения, а не на конфигурации и настройке фреймворка.  **Spring Mail** – это модуль в Spring Framework, который предоставляет возможности для отправки электронной почты из приложения Java. Он упрощает отправку электронных сообщений, предоставляя высокоуровневые абстракции и API для работы с почтовыми протоколами. Spring Mail позволяет отправлять электронные письма с использованием различных протоколов, таких как SMTP, POP3 и IMAP. Так же Spring Mail интегрируется с другими компонентами Spring, такими как Spring Boot. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *8* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Freemarker** – это шаблонный движок для языка программирования Java, который предоставляет простые и эффективные средства для генерации текстового вывода на основе шаблонов. Он позволяет разработчикам создавать динамические веб- страницы, электронные письма, отчеты и другие текстовые документы, интегрируя в них данные из ращличных источников, таких как: базы данных, объекты Java или другие источники данных.  Основные особенности и возможности Freemarker:   * Простота использования: Freemarker предоставляет простой и интуитивно понятный синтаксис шаблонов, который легко читается и поддерживается разработчиками. Он позволяет быстро создавать и модифицировать шаблоны без необходимости изучения сложных языков разметки. * Большие возможности: Freemarker поддерживает различные функции и операции, такие как условные операторы, циклы, вложенные шаблоны, макросы и многое другое. Это позволяет создавать сложные и динамические шаблоны для разных типов приложений. * Интеграция с Java и другими языками: Freemarker легко интегрируется с приложениями Java, позволяя испольщовать объекты Java в шаблонах и обрабатывать их встроенными функциями. Данный движок поддерживает интеграцию с другими языками программирования, такими как Python и JavaScript. * Расширяемость: Freemarker обладает гибкой архитектурой, которая позволяет создавать собственные функции и макросы для расширения его функциональности и адаптации под конкретные потребности проекта.   Freemarker является популярным инструментом в мире Java-разработки и широко используется для создания динамических веб-страниц.  **HTML (Hyper Text Markup Language)** – это стандартный язык разметки, используемый для создания веб-страниц. Он предоставляет структурные элементы, необходимые для организации содержимого веб-страницы и определение ее труктуры и внешнего вида.  Основные концепции HTML:   * Теги: язык разметки основан на использовании тегов, которые определяют структуру и содержание веб-страницы. Теги обычно имеют открывающий и закрывающий элемент, между которыми размещается содержимое. * Атрибуты: HTML элемент может содержать атрибуты, которые предоставляют дополнительную информацию о элементе. Например атрибут “src” используется для указания источника изображения. * Семантика: язык HTML предоставляет семантические эдементы, которые позволяют разработчика определить значение и назначение различных частей веб-страницы: <header>, <footer>, <body> и т.д. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *9* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| HTML является основой веб-разработки и часто используется в сочетании с CSS (Carcade Style Sheets) для стилизации и JavaScript для добавления интерактивности. С помощью HTML можно создавать разообразные веб-страницы, начиная от простых блогов и информационных сайтов до сложных веб-приложений.  **CSS (Cascading Style Sheets) –** это язык таблиц стилей, используемый для определения внешнего вида и форматирования содержимого веб-страницы, написанного на языке разметки HTML. Он позволяет разработчикам задавать стили, такие как цвета, шрифты, отступы, размеры и многое другое, чтобы придать странице желаемый внешний вид.  Основные возможности CSS:   * Селекторы: с помощью CSS можно применять стили к HTML-элементам с помощью селекторов. Селекторы могут быть эдементами, классами, идентификаторами или псевдоклассами, что позволяет точно выбирать элементы для применения стилей. * Свойства и значения: CSS определяет свойства например: color, font-size, margin и т.д. и их значения, которые применяются к выбранным элементам. * Единицы измерения: CSS поддерживает различные единицы измерения, такие как пиксели, проценты, em, rem и другие, что позволяет разработчикам задавать размеры и расстояния с учетом требуемой точности и адаптивности. * Медия-запросы: с помощью медиа-запросов CSS можно создавать адаптивные стили, которые изменяются в зависимости от характеристик устройства, таких как размер экрана или ориентация. Это позволяет создавать отзывчивые веб- дизайны, которые выглядят хорошо на различных устройствах и разрешениях экранов.   CSS играет важную роль в веб-разработкеб позволяя разработчикам создавать внешне приятные и функциональные веб-страницы. Он часто используется в сочетании с HTML и JavaScript для создания полноценных веб-платформ.  **1.3 Фреймворк Spring Boot**  Spring Boot – это фреймворк для разработки приложений на языке Java, который был разработан компанией Pivotal Software (теперь VMware) и выпущен в 2014 году. Он был создан для упрощения процесса создания приложений на основе платформы Spring Framework. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *10* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

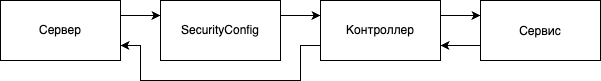
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| История создания фреймворка началась с осознания разработчиками необходимости упростить процесс настройки и развертывания приложений на основе Spring Framework. Ранее для создания приложений на Spring требовалась значительная настройка, включая конфигурацию XML (eXtensible Markup Language – язык разметки, который используется для хранения и паередачи данных) файлов и других параметров. Основная идея Spring Boot заключается в том, чтобы предоставить “конвенции,  а не конфигурации” (convention over configuration), что означает, что большинство настроек и конфигураций по умолчанию должны быть автоматизированы, чтобы разработчики могли сосредоточиться на бизнес-логике своих приложений, а не на настройке фреймворка.  Кроме того, Spring Boot включает в себя множество встроенных функций, таких как встроенные серверы приложений (например Tomcat или Jetty), автоматическое управление зависимостями через инструменты сборки (например Maven или Gradle), автоматическую конфигурацию на основе классов Java и аннотация, интеграцию с базами данных и другими сервисами, что дополнительно ускоряет и упрощает разработку.  Исторически Spring Boot стал ответом на запрос сообщества разработчиков Java, которое требовало более простых и эффективных средств для разработки приложений на базе Spring. Появление Spring Boot привело к росту популярности Spring Framework в целом, особенно в контексте разработки микросервисных архитектур, где простота и скорость разработки являются ключевыми факторами успеха.  Вот основные преимущества Spring Boot по отношению к другим фреймворкам:   * Простота использования и быстрый старт: Spring Boot предоставляет множество удобных инструментов и абстракций, которые значительно упрощают процесс разработки. Благодаря автоматической конфигурации и встроенным серверам приложений, разработчики могут создавать приложения быстрее и без необходимости тратить много времени на начальную настройку проекта. * Конвенции перед конфигурацией: Одним из ключевых преимуществ Spring Boot является использование “конвенций перед конфигурацией”, что позволяет уменьшить количество шаблонного кода и рутинных операций. * Встроенные функции и интеграции: Spring Boot включает в себя множество встроенных функций, таких как встроенные серверы приложений, системы управления зависимостями, автоматическая конфигурация и многое другое. Это упрощает разработку, управление и масштабирование приложений. Кроме того, Spring Boot легко интегрруется с другими инструментами и библиотеками, что делает его гибким и мощным фреймворком для разработки различных типов приложений. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *11* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * Большое сообщество и поддержка : Данный фреймворк имеет большое и активное сообщество разработчкиво, что означает, что вы можете легко найти поддержку, документацию, учебные материалы и решения своих проблем в интернете. Это также означает, что фреймворк постоянно развивается и совершенствуется. * Управление зависимостями и совместимость: Spring Boot интегрирует в себя систему управления зависимостями Maven или Gradle, что упрощает управление библиотеками и зависимостями проекта. * Поддержка облачных технологий: Spring Boot имеет нативную поддержку облачных технологий и платформ, таких как Kubernetes, Docker, AWS, Azure и другие. Это делает его идеальным выбором для разработки облачных приложений и микросервисов, а также обеспечивает высокую масштабируемость и доступность приложений. * Безопасность: Spring Boot обеспечивает высокий уровень безопасности благодаря мощному механизму аутентификации и авторизации, а также интеграции с различными инструментами и библиотеками для защиты приложений от различных угроз, таких как атаки XSS, CSRF токены, инъекции SQL и т. д.   С момента своего выпуска Spring Boot стал одним из наиболее популярных фреймворков адля разработки Java-приложений, особенно в области микросервисной архитектуры. Его активное сообщество разработчиков и широкий спектр интеграции и расширений делают его предпочтительным выбором для многих разработчиков и организаций.  **1.4 Постановка задачи**  Для обеспечения оптимального удобства использования веб-платформы нам необходимо определить список функций, которые улучшат опыт как для клиентов так и для администраторов приложения. Составление технического задания представляет собой следующий этап в жизненном цикле программного обеспечения после сбора и анализа требований. Правильно составленное техническое задание содержит подробную информацию о требованиях к разрабатываемому продукту, его функциональности, качеству и архитектуры. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *12* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Техническое задание веб-платформы для эффективных тренировок в спортивном зале:   * текст на сайте не должен содержать орфографических, пунктуационных, синтаксических, грамматических ошибок; * возможность изменения роли пользователя в административной панели; * возможность начисления или списывания денег с баланса пользователей в административной панели; * возможность измененить аватарку пользователя в панели пользователя; * возможность измененить пароль от аккаунта пользователя в панели пользователя; * возможность изменить имя пользователя в панели пользователя; * возможность обработки запросов на роль тренера в панели администратора; * возможность авторизации в системе; * навигационное меню должно содержать кнопки “Главная ”, “Программы ”, “Питание”, “Упражнения”, “Тренера”; * при нажатии кнопки “ Главная ” должен производиться редирект на главную страницу; * навигационное меню личного кабинета должно содержать кнопки “На главную”, “Информация”, “Программы”, “Питание”, “Выйти из аккаунта”, если пользователь авторизован как администратор, то будет отображаться кнопка “Панель админа”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *13* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **ПРОЕКТИРОВАНИЕ СИСТЕМЫ**    1. **Структура и архитектура веб-платформы**   Веб-платформа для эффективных занятий в тренажерном зале выступает в разных ролях:   * + - Трекер прогресса: Пользователь может отмечать свои тренировки и платформа будет автоматически подисчитывать количество повторений, подходов и т.п;     - Социальная сеть: Платформа объединяет пользователей с похожими целями и интересами, позволяя им обмениваться опытомб мотивировать друг друга и даже соревноваться;     - Интерактивный тренажер: На плафторме представлены 3д упражнения, которые позволяют пользователю понять как правильно делать упражнение;     - Информационный ресурс: Платформа предоставляет программы питания и тренировок;   Каждая их этих ролей помогает улучшить эффективность занятий пользователей и обеспечить им лучшие результаты.  В данном проекте реализована клиент – серверная архитектура.  Клиент-серверная архитектура заключается в том, что один компьютер (сервер) предоставляет ресурсы или инофрмацию другому компьютеру (клиенту) через сеть. Клиенты отправляют запросы на сервер для выполнения определенных действий или получения информации, а серверы обрабатывают эти запросы и отправляют обратно ответы. Эта модель обеспечивает эффективное взаимодействие в сети и используется в различных типах приложений и сервисов.  На сервере обычно располагается основная част данных и ресурсов: файлы, базы данных и т. д. Клиенты же обращаются к серверу с запросами на получение данных иди выполнение определенных операций.  Клиенты и серверы могут взаимодействовать различными способами, используя различные сетевые протоколы. Например, веб-браузер на компьютере может быть клиентом, а веб-сервер, хранящий веб-платформу будет сервером. Когда клиент вводит URL (uniform Resource Locator – адрес ресурса в сети Интернет) в браузере, браузер отправляет запрос на сервер, который затем отвечает, отправляя запрошенную веб- страницу обратно на ваш компьютер.  Клиент-серверная архитектура широко используетя в современных информационных системаб и веб-платформах, обеспечивая эффективное взаимодейстиве между компьютерами в сети. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *14* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2.2 Архитектура сервера**  Рассмотрим архитектуру сервера (см. рисунок 2.1).  Рисунок 2.1 – Архитектура сервера Сервер веб-платформы работает в следующем порядке:   1. Получение HTTP запроса: Сервер получает запрос от пользователя который включает в себе заголовок и тело запроса. 2. Обработка запроса в SecurityConfig: Запрос проходит через конфигурацию безопасноит (SecurityConfig) которая настроена в приложении с использованием Spring Security. В SecurityConfig определены правила доступа к различным ресурсам и страницам. Если страница требует авторизации и пользователь не авторизирован, то securityConfig перенаправляет его на страницу аутентификации или регистрации. 3. Перенаправление на страницу аутентификации или регистрации: Если страница доступна без авторизации и пользователь пытается выполнить запрос к ней, то сервер перенаправит запрос в соответствующий контроллер. Контроллер – это компонент приложения, который отвечает за обработку запросов и взаимодействие с сервисами для выполения различной логики. 4. Обработка запроса в контроллере: Контроллер отправляет данные или параметры в сервис, который отвечает за выполнение логики обработки передаваемых ему данных. Сервис обрабатывает полученные данные, выполняет необходимые операции (например, сохранение данных в базе данных или выборку по определенным параметрам) и возвращает эти данные в контроллер. 5. Формирование шаблона страницы: После получения обработанных данных от сервиса, контроллер использует шаблонизатор FreeMarker для формирования HTML страницы. В этот шаблон вставляются данные, полученные из сервиса. После формирования шаблона страницы, контроле отправляет сформированную HTML страницу клиенту | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *16* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |



Таким образом, процесс обработки запроса от пользователя до возвращения страницы пользователю включает в себя несколько этапов: проверку доступа, обработку запроса в контроллере, выполнение логики в сервисе, формирование HTML страницы и отправка ее пользователю.



## 2.3 Архитектура сервиса

Рассмотрим архитектуру сервиса (см. рисунок 2.2).

Рисунок 2.2 – Архитектура сервиса

Репозиторий – Spring Data JPA которые представляют собой интерфейсы, которые можно создавать для доступа к данным. Запросы JPA автоматически генерируются на основе названий методов. Обычно репозтории Sping Data расширяют интерфейсы Repository или CrudRepository. При использовании автоконфигурации посик репозиториев начинается с пакета, содержащего главный конфигурационный класс (тот, который аннотирован @EnanleAutoConfiguration или @SpringBootApplication) и продолжается по иерархии вниз.

Сервис работает в следующем порядке:

1. Отправка запроса в репозиторий: Сервис отправляет параметры в репозиторий.
2. Выполнение SQL запроса в базе данных: Репозиторий использует SQL запросы либо автоматические запросы для взаимодействия с базой данных. Этот запрос использует параметры передаваемые сервисом для выборки данных (SELECT), изменения данных (INSERT, UPDATE, DELETE) или другими операциями.
3. Обработка запроса в базе данных: База данных получает запрос от репозитория и выполняет его. В результате выполнения запроса, возвращаются данные, которые соответствуют параметрам и типу запроса.
4. Обработка данных в сервисе: Сервис получает данные от репозитория и начинает их обработку в соответствии с логикой приложения.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2.4 Cоздание макета веб-платформы**  Для создания макета будет использован такой инструмент для дизайна интерфейсов как Canva.  Основные особенности Сanva:   1. Гибкий доступ: Canva предоставляет возможность работать над проектами из любой точки с доступом в Интернет. Это особенно удобно, т.к. нет привязки к устройству. Благодаря этой гибкости пользователь может легко переключаться между различными мастами работы и устройствами. 2. Большой набор шаблонов: Canva предлагает обширную коллекцию готовых шаблонов для различных типов дизайна, включая баннеры, цветовые палитры и многое другое. Это упрощает процесс создания дизайна, поскольку пользватель может выбрать подходящий шаблон и настроить его под свои нужды, экономя время и усилия. 3. Простой интерфейс: Интерфейс Canva обладает интуитивной структурой и простыми инструментами редактирования, что делает процесс создания дизайна бфстрым и эффективным. Пользователь может легко изменить цвет, шрифт и его размеры, расположение элементов, добавлять текст и видео без необходимости обширных навыков дизайна. 4. Безопасное хранение и доступ к файлам: Canva обеспечивает безопасное хранение всех созданных проектов в облаке. Это позволяет пользователю сохранять свои дизайны и иметь к ним доступ из любого устройства, что облегчает работу. 5. Обучающие ресурсы: Canva предоставляет обширную базу знаний, видеоуроки и руководства по использованию инструментов, созданию дизайна и улучшению UI/UX дизайна. Это помогает пользователю быстро освоить программу и сделать свои дизайны быстро и качественно.   Для начала, обозначим основные модули для пользовательского интерфейса. Веб-приложение будет состоять из нескольких страниц. Также, будет реализовано меню сайта в верхней части, для более удобной ориентации по сайту (см. рисунок 2.4.1).    Рисунок 2.3 – Структура меню в верхней части. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *18* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| При нажатии кнопки “Главная страница” пользователь переходит на главную страницу. Данная страница содержит общую информацию о платформе и тренировках.  При нажатии кнопки “Программы” пользователь переходит на страницу с программами, на которой будут расписаны программы по различным типам тренировок (например, кардио, йога, пауэрлифтинг, фулл-бади). Пользователь может ознакомиться с деталями каждой программы и выбрать для себя подходящую.  При нажатии кнопки “Питание” пользователь переходит на страницу с программами питания, на которой будет отображен список программ с питанием, который можно отфильтровать по типам тренировок (например, кардио, йога, пауэрлифтинг, фулл-бади). Пользователь может ознакомиться с деталями каждой программы питания и выбрать для себя подходящую.  При нажатии кнопки “Упражнения” пользователь переходит на страницу с упражнениями, на которой будет отображен список упражнений, который множно будет отфильтровать по типам тренировок (например, кардио, йога, пауэрлифтинг, фулл-бади). Пользователь может ознакомиться с деталями каждого упражнения и просмотреть 3д выполнение данного упражнения.  При нажатии кнопки “Тренера” пользователь переходит на страницу с тренерами, на которой будет отображен список тренеров, который можно отфильтровать по типам тренировок (например, кардио, йога, пауэрлифтинг, фулл-бади). Пользователь может посмотреть фотку, опыт работы, почту и информацию о каждом тренере.  При нажатии кнопки “Личный кабинет” пользователь переходит на страницу с личным кабинетом. Если пользователь не авторизирован, то его редиректит на страницу с авторизацией. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *19* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рассмотрим структуру интерфейса главной страницы (см. рисунок 2.4)    Рисунок 2.4 – Структура интерфейса главной страницы.  Модуль “Выбор программы” отображен в виде списка программ, каждая из которых включает название и кнопку подробнее. Здесь пользователи могут быстро ознакомиться с различными доступными программами, чтобы выбрать наиболее подходящую для своих нужд.  Модуль “О нас” отображен в виде списка, который включает основные плюсы моей веб-платформы и наши предложения пользователю. В этом модуле подробно изложены ключевые преимущества, которые делают мою платформу уникальной и привлекательной для клиентов.  Модуль “Контакты” отображен в виде списка, который включает контактную информацию, такую как ссылки на наши страницы в социальных сетях и адрес электронной почты. Приведенные ссылки на социальные сети позволяют пользователям легко найти и подписаться на наши страницы, чтобы быть в курсе новостей и обновлений.  Модуль “Подписка” отображен в виде списка с видео на заднем плане. Список включает в себя типы подписок доступные для пользователей и характеристику каждой подписки. В этом модуле подробно описаны различные варианты подписок, которые можно оформить, начиная от базовых пакетов до премиум-опций. Видео на заднем плане создает динамичный и привлекательный визуальный эффект, привлекая внимание пользователей. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *20* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Контакты” отображен в виде сетки фоток, которая включает фотку главного разработчика, главного дизайнера и главного фитнес-тренера. Так же в этом модуле присутствует кнопка, при нажатии которой пользователь переходит на страницу с анкетой, после заполнения и одобрения пользователь может стать частью команды, а именно фитнес-тренером.  Рассмотрим структуру интерфейса страницы “Программы” (см. рисунок 2.5)    Рисунок 2.5 – Структура интерфейса страницы программ.  Модуль “Больше о программе” отображен в виде списка, который структурирован так, что в левой части отображается краткое описание прогрммы, а справа находится мотивационное видео о данной программе. Краткое описание включает основные элементы программы, такие как цели, тип тренировок, ожидаемые результаты. Это помогает пользователю быстро понять, о чем программа и как она может соответствовать его потребностям и целям. Мотивационное видео расположенное справа, визуально демонстрирует преимущество программы, ее атмосферу и потенциал для достижения результатов. Таким образом пользователи могут более глубоко ознакомиться с программой и принять обоснованное решение о ее выборе.  Модуль “Тренажеры” отображен в виде слайдера (сетки), где каждый элемент включает в себя две части. Левая часть содержит 3D модель упражнения, которая динамически показывает технику выполнения упражнения и демонстрирует, какие группы мышц задействованы. Эта визаулизация помогает пользователям понять правильную форму и движение, что полезно для навичков. Правая часть элемента состоит из названия упражнения или тренажера и краткого описания. Описание включает информацию о пользе упражнения, целевых мышечных группах и ключевых моментах выполнения. Такой подход позволяеи пользователям не только ознакомиться с тренажерами и упражнениями, но и понять их правильное предназначение. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *21* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Программы” отображен в виде слайдера (сетки), где каждый элемент содержит название программы и её подробное описание. Описание включает название упражнений, количество подходов или повторений для каждой программы. Такой формат позволяет легко просматривать различные программы и сравнивать их, чтобы выбрать ту, которая лучге всего соответствует цели и уровню подготовки. Описание каждой программы детализировано таким образом, чтобы пользователи могли следовать программе с максимальной эффективностью.  Рассмотрим структуру интерфейса страницы “Питание” (см. рисунок 2.6).    Рисунок 2.6 – Структура интерфейса страницы “Питание”.  Модуль “Программы питания” отображен в виде списка, который можно фильтровать по типам тренировок, таким как все программы, кардио, йога, пауэрлифтинг и фулл-бади. На этой странице отображается перечень различных программ питания, предоставляющих детальную информацию о каждой из них. Пользователи могут просматривать и анализировать каждую программу, чтобы выбрать ту, которая наилучшим образом соответствует их потребностям и целям тренировок.  Рассмотрим структуру интерфейса страницы “Упражнения” (см. pисунок 2.7)    Рисунок 2.7 – Структура интерфейса страницы “Упражнения”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *22* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Упражнения” отображен в виде списка, который можно фильтровать по типам тренировок, таким как все программы, кардио, йога, пауэрлифтинг и фулл-бади. На этой странице отображается перечень различных упражнений, предоставляющих детальную информацию о каждой из них. Пользователи могут просматривать и анализировать каждое упражнение, чтобы выбрать то, которое наилучшим образом соответствует их потребностям и целям тренировки.  Рассмотрим структуру интерфейса страницы “Тренера” (см. рисунок 2.8)    Рисунок 2.8 – Структура интерфейса страницы “Тренера”.  Модуль “Тренера” представлен в виде списка, которые можно отфильтровать по типам тренировок, таким как кардио, йога, пауэрлифтинг и фулл-бади. В этом модуле отображается перечень тренеров, где пользователь может увидеть фотографию каждого тренера, ознакомиться с их опытом работы, контактной почтой и дополнительной информацией.  Рассмотрим структуру интерфейса страницы “Личный кабинет” (см. рисунок 2.9)    Рисунок 2.9– Структура интерфейса страницы “Личный кабинет”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *23* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Информация” представлен в виде списка, содержащего фотографию пользователя, его имя, электронную почту и статус. В модуле также имеются кнопки для изменения имени, пароля, и фотографии профиля. В нижней части отображается краткая сттатистика по тренировкам, включая количество повторений и подходов.  Модуль “Программы” представлен в виде списка с выпадающим меню, где можно выбрать тренировку для просмотра подробной информации о ней. В этом модуле также хранится список завершенных тренировок. Дополнительно есть таймер для отслеживания времени на тренировке, который можно ставить на паузу, запускать, сбрасывать и сохранять данные тренировок  Модуль “Питание” представлен в виде выпадающего меню со всеми программами питания, при выборе которой открывается подробная информация.  Модуль “Оплата” отображает список с адресами крипто-кошельков для пополнения баланса.  Рассмотрим структуру интерфейса страницы “Панель администратора” (см. рисунок 2.10)    Рисунок 2.10 – Структура интерфейса страницы “Панель администратора”.  Модуль “Пользователи” представлен в виде таблицы которая состоит из столбцов: почта, номер телефона, активность, роли, баланс, бан, кнопки изменить роль, кнопки изменить баланс. В этом модуле администратор может банить пользователей, те они не смогут авторизироваться в системе, изменять роль пользователя (например, пользователь, администратор, тренер), изменить баланс пользователя (увеличить или уменьшить).  Модуль “Заявки” представлен в виде таблицы которая состоит из столбцов: почта, о себе, опыт работы, тип тренировки, и кнопки принять и отказать. В этом модуле администратор может удтвердить роль тренера, либо отказать ему в этом. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *24* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Определим основные цвета, которые будут использованы при разработке веб- платформы (см. рисунок 2.11).    Рисунок 2.11 – Основные цвета.  Определим основные шрифты, которые будут использоваться при разработке веб- платформы (см. рисунок 2.12).    Рисунок 2.12 – Основные шрифты.  **2.5 Проектирование базы данных**  В своем проекте я использовал базу данных PostgreSQL.  PostgreSQL (Postgres) – это мощная, открытая объектно-реляционная систему управления базами данных (далее СУБД), которая поддерживает стандарты SQL и расширения. Разработка PostgreSQL началась в 1986 году как проект в Калифорнийском университете в Беркли и с тех пор эволюционировала в одну из самых надежных и популярных СУБД в мире.  Рассмотрим плюсы PostgreSQL перед остальными СУБД:   * Открытый исходный код: PostgreSQL является бесплатным и с открытым исходным кодом, что позволяет пользователям изменять и улучшать его под свои нужды. * Расширяемость: PostgreSQL поддерживает пользовательские функции, типы данных, операторы и индексы. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *25* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * Продвинутая поддержка данных: помимо стандартных типов данных, PostgreSQL поддерживает JSON, XML и другие. * Масштабируемость: PostgreSQL хорошо масштабируется как по вертикали (увеличение мощности одного сервера), так и по горизонтали (разделение нагрузки между несколькими серверами). * Надежность и восстановление: система транзакций PostgreSQL обеспечивает целостность данных, а механизмы резервного копирования и восстановления помогают минимизировать потери данных.   Настраивать базу данных буду через pgAdmin – это наиболее популярный и функциональный инструмент для управления и разработки PostgreSQL баз данных. pgAdmin предоставляет графический пользовательский интерфейс и множество возможностей для администраторов и разработчиков баз данных.  Рассмотрим основные функции pgAdmin:   * Управление базами данных: Создание, изменение и удаление баз данных, таблиц, индексов и других объектов. * Визуализация данных: Просмотр и редактирование данных в таблицах. * Консоль запросов: Выполнение SQL-запросов и скриптов с функцией автодополнения. * Мониторинг производительности: Инструменты для отслеживания производительности серверов и запросов. * Резервное копирование и восстановление: Легкое создание бэкапов и восстановление данных.   pgAdmin является кроссплатформенным инструментом и поддерживает операционные системы Windows, macOS и Linux.  Рассмотрим структуры таблиц, которые используются в проекте (см. таблицы 2.13 – 2.21):  Таблица 2.13 – Структура таблицы “О нас”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *26* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный  ключ |
| 2 | Заголовок | title | Символьный | 255 |  |
| 3 | Информация | info | Символьный | 255 |  |

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

Таблица 2.14 – Структура таблицы “Все программы”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный  ключ |
| 2 | Название программы | progrname | Символьный | 255 |  |
| 3 | Информация о  программе | progrinfo | Символьный | 255 |  |
| 4 | Путь к картинке | imgurl | Символьный | 255 |  |
| 5 | Тип программы | type | Символьный | 255 |  |
| 6 | Путь к json  файлу | jsonurl | Символьный | 255 |  |
| 7 | Путь к видео | videourl | Символьный | 255 |  |

Таблица 2.15 – Структура таблицы “Тренажеры”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный  ключ |
| 2 | Название  тренажера | name | Символьный | 255 | Вторичный  ключ |
| 3 | Путь к 3д  модели | imageurl | Символьный | 255 |  |
| 4 | Информация о тренажере | info | Символьный | 255 |  |
| 5 | Тип программы | type | Символьный | 255 |  |
| 6 | Группа мышц | musclegroup | Символьный | 255 |  |

Таблица 2.16 – Структура таблицы “Программы питания”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный  ключ |
| 2 | Завтрак | morning | Символьный | 255 |  |
| 3 | Перекус | refreshment | Символьный | 255 |  |
| 4 | Обед | lunch | Символьный | 255 |  |
| 5 | Полдник | afternoon | Символьный | 255 |  |
| 6 | Программа  тренировок | progrtype | Символьный | 255 |  |
| 7 | Калории | calories | Символьный | 255 |  |
| 8 | БЖУ | pfc | Символьный | 255 |  |

***Изм. Лист***

***№ докум.***

***Подп.***

***Дата***

# ДП.АС59.200067 – 05 81 00

***Лист***

*27*

Таблица 2.17 – Структура таблицы “Подписки”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный ключ |
| 2 | Тип подписки | type | Символьный | 255 |  |
| 3 | Цена | price | Числовой | 9 |  |
| 4 | Срок подписки | time | Символьный | 255 |  |
| 5 | Информация о  подписке | info | Символьный | 255 |  |

Таблица 2.18 – Структура таблицы “Слайдер”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный  ключ |
| 2 | Путь к картинке | imgurl | Символьный | 255 |  |
| 3 | Тип слайдера | type | Символьный | 255 |  |
| 4 | Информация | info | Символьный | 255 |  |

Таблица 2.19 – Структура таблицы “Тренера”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный ключ |
| 2 | Почта | email | Символьный | 255 | Вторичный  ключ |
| 3 | О себе | aboutme | Символьный | 255 |  |
| 4 | Опыт работы | workexperiens | Символьный | 255 |  |
| 5 | Подтверждение  статуса тренера | confirmation | Булевый |  |  |
| 6 | Тип программ  тренера | training\_type | Символьный | 255 |  |

Таблица 2.20 – Структура таблицы “Роль пользователей”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Вторичный  ключ |
| 2 | Роль  пользователя | roles | Символьный | 255 |  |

***Изм. Лист***

***№ докум.***

***Подп.***

***Дата***

# ДП.АС59.200067 – 05 81 00

***Лист***

*28*

Таблица 2.21 – Структура таблицы “Пользователи”.

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №:  п/п | Наименование  реквизита | Обозначение  реквизита | Тип данных | Длина  поля | Признак  ключа |
| 1 | Идентификатор | id | Числовой | 9 | Первичный ключ |
| 2 | Активность | active | Булевый |  |  |
| 3 | Путь к аватарке | avatarurl | Символьный | 255 |  |
| 4 | Дата  регистрации | date\_of\_created | Символьный | 255 |  |
| 5 | Почта | email | Символьный | 255 |  |
| 6 | Имя | name | Символьный | 255 |  |
| 7 | Пароль | password | Символьный | 255 |  |
| 8 | Номер телефона | phonenumber | Символьный | 255 |  |
| 9 | Список  завершенных программ | last\_program\_array | Символьный | 255 |  |
| 10 | Баланс  пользователей | user\_balance | Числовой | 9 |  |

***Изм. Лист***

***№ докум.***

***Подп.***

***Дата***

# ДП.АС59.200067 – 05 81 00

***Лист***

*29*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ВЕБ – ПЛАТФОРМЫ**    1. **Реализация клиентской части**   Пользователь взаимодействует с системой, вводя необходимые данные с помощью клавиатуры и нажимая нужные кнопки мышью. Проект представляет собой многостраничную веб-платформу. При входе на сайт пользователя приветствует модуль “Главная страница” (см. рисунок 3.1).  Рисунок 3.1 – Модуль “Главная страница” В верхней части модуля расположены:   * + - логотип компании;     - навигационное меню в котором при нажатии на кнопку, пользователя перенесёт на нужную страницу;   Так же в этом модуле расположен слайдер с самой узнаваемой и популярной фразой известного бодибилдера. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *30* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Выбор программы” отображен в виде списка программ, каждая из которых включает название и кнопку подробнее (см. рисунок 3.2).    Рисунок 3.2 – Модуль “Выбор программы”.  Модуль “О нас” отображен в виде списка, который включает основные плюсы моей веб-платформы и наши предложения пользователю (см. рисунок 3.3).    Рисунок 3.3 – Модуль “О нас”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *31* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Контакты” отображен в виде списка, который включает контактную информацию, такую как ссылки на наши страницы в социальных сетях и адрес электронной почты (см. рисунок 3.4).    Рисунок 3.4 – Модуль “Контакты”.  Модуль “Подписка” отображен в виде списка с видео на заднем плане (см. рисунок 3.5).    Рисунок 3.5 – Модуль “Подписка”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *32* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Модуль “Команда” отображен в виде сетки фоток, которая включает фотку главного разработчика, главного дизайнера и главного фитнес-тренера (см. рисунок 3.6).    Рисунок 3.6 – Модуль “Команда”.  Рассмотрим страницу программы на примере типа тренировок пауэрлифтигн.  Модуль “Больше о программе” отображен в виде списка, который структурирован так, что в левой части отображается краткое описание прогрммы, а справа находится мотивационное видео о данной программе (см. рисунок 3.7).    Рисунок 3.7 – Модуль “Больше о программе”.  Модуль “Тренажеры” отображен в виде слайдера (сетки), где каждый элемент включает в себя две части. Левая часть содержит 3D модель упражнения, которая динамически показывает технику выполнения упражнения и демонстрирует, какие группы мышц задействованы. Правая часть элемента состоит из названия упражнения или тренажера и краткого описания (см. рисунок 3.8). | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *33* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.8 – Модуль “Тренажеры”.  Модуль “Программы” отображен в виде слайдера (сетки), где каждый элемент содержит название программы и её подробное описание (см. рисунок 3.9).    Рисунок 3.9 – Модуль “Программы”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *34* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рассмотрим страницу питания. Модуль “Программы питания” отображен в виде списка, который можно фильтровать по типам тренировок, таким как все программы, кардио, йога, пауэрлифтинг и фулл-бади (см. рисунок 3.10).    Рисунок 3.10 – Модуль “Программы питания”.  Рассмотрим страницу упражнений. Модуль “Упражнения” отображен в виде списка, который можно фильтровать по типам тренировок, таким как все программы, кардио, йога, пауэрлифтинг и фулл-бади (см. рисунок 3.11).    Рисунок 3.11 – Модуль “Упражнения”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *35* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рассмотрим страницу “Тренера”. Модуль “Тренера” представлен в виде списка, которые можно отфильтровать по типам тренировок, таким как кардио, йога, пауэрлифтинг и фулл-бади (см. рисунок 3.12).    Рисунок 3.12 – Модуль “Тренера”.  Рассмотрим страницу “Личный кабинет”. Модуль “Информация” представлен в виде списка, содержащего фотографию пользователя, его имя, электронную почту и статус (см. рисунок 3.13).    Рисунок 3.13 – Модуль “Информация”.  Модуль “Программы” представлен в виде списка с выпадающим меню, где можно выбрать тренировку для просмотра подробной информации о ней (см. рисунок 3.14). | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *36* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.14 – Модуль “Программы”.  Модуль “Питание” представлен в виде выпадающего меню со всеми программами питания, при выборе которой открывается подробная информация (см. рисунок 3.15).    Рисунок 3.15 – Модуль “Питание”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *37* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рассмотрим страницу “Панель администратора”. Модуль “Пользователи” представлен в виде таблицы которая состоит из столбцов: почта, номер телефона, активность, роли, баланс, бан, кнопки изменить роль, кнопки изменить баланс (см. рисунок 3.16).    Рисунок 3.16 – Модуль “Пользователи”.  Модуль “Заявки” представлен в виде таблицы которая состоит из столбцов: почта, о себе, опыт работы, тип тренировки, и кнопки принять и отказать (см. рисунок 3.17).    Рисунок 3.17 – Модуль “Заявки”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *38* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **3.2 Тестирование веб-платформы**  Рассмотрим модуль “Регистрация”. Модуль “Регистрация” включает в себя поля для ввода электронной почты, имени, номера телефона и пароля (см. рисунок 3.18).    Рисунок 3.18 – Модуль “Регистрация”.  Введенные данные отправляются на сервер для обработки, после чего сервер обрабатывает их и выводит ответ в консоль (см. рисунок 3.19).    Рисунок 3.19 – Ответ в консоли на введенные данные.  Если пользователь с указанной электронной почтой уже существует, выводится соответствующее сообщение в консоль (см. рисунок 3.20).    Рисунок 3.20 – Сообщение в конмсоле при регистрации пользователя с уже существующей электронной почтой.  Модуль включает проверку ввода данных: если в поле электронной почты отсутствует знак “@”, выводится соответствующая подсказка (см. рисунок 3.21). | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *39* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.21 – Подсказка при отсутствии в поле почты знака “@”.  Рассмострим модуль “Забыли пароль”. Модуль “Забыли пароль” включает себя поле ввода почты. Пользователь может восстановить забытый пароль, введя привязанную к аккаунту почту (см. рисунок 3.22). Почта проверяется по базе данных,и если такая почта найдена, на нее отправляется специальный код (см. рисунок 3.23). Также в консоли выводится сообщение об отправке кода (см. рисунок 3.24).    Рисунок 3.22 – Модуль “Забыли пароль”. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *40* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 3.23 – Новый пароль который получает пользователь.    Рисунок 3.24 – Сообщение в консоли об отправке сообщения пользователю на почте.  Рассмотрим ответ пользователю на заявку стать тренером. Сообщение пользователю если приняли завяку на тренерство (см. рисунок 3.25).    Рисунок 3.25 – Сообщение пользователю о решении компании по заявке стать тренером. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *41* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сообщение пользователю если не приняли завяку на тренерство (см. рисунок  3.26).    Рисунок 3.26 – Сообщение пользователю о решении компании по заявке стать тренером. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *42* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ЗАКЛЮЧЕНИЕ**  В данном дипломном проекте была разработана веб-платформа для эффективных тренировок в спортивном зале. Для реализации использовались языки HTML, CSS, JavaScript, Java и фреймворк Spring.  Во время решения задачи дипломного проектирования были пройдены все этапы разработки такие как: обследование систем в данной предметной области, обследование объекта автоматизации, проектирование системы, реализация, технико-экономическое обоснование, энерго- и ресурсосбережение.  По результатам разработки была успешно достигнута цель разработать приемлемую по себестоимости систему, обеспечивающую весь вышеописанный требуемый функционал.  Одни из основных требований к системе – масштабируемость, возможность улучшения и удобство в использования как со стороны клиента, так и со стороны администратора, а также низкие требования к ресурсам ПК для работы. Все вышеназванные требования были достигнуты.  Таким образом все задачи дипломного проектирования были выполнены. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *52* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **СПИСОК СОКРАЩЕНИЙ**  БД – база данных;  Веб (от англ.Web) – World Wide Web; ПО – программное обеспечение;  ПК – персональный компьютер; CSS – Cascading Style Sheets;  HTML – HyperText Markup Language; HTTP – HyperText Transfer Protocol; XML – eXtensible Markup Language. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *53* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**   1. Горнаков, С. Осваиваем популярные системы управления сайтом / С. Горнаков, Издательство “ДМК-Пресс”, 2009. – 336с. 2. Хвещук В., Муравьев Г.Л., Крапивин Ю. Б. Методическое пособие по курсовому проектированию по дисциплине “Базы и банки данных”. 3. Хеник, Б. HTML и CSS. Путь к совершенству / Б. Хеник – Санкт-Петербург, Издательство “Питер”, 2013. – 336с. 4. Стандарт БрГТУ по оформлению курсовых и дипломных работ. 5. “Технико-экономическое обоснавание дипломных проектов для студентов специальности: автоматизированные системы обработки информации, искусственный интелект, вычислительные машины системы и сети, промышленная электроника для дневной и заочной формы обучения”, Брест, 2014. – 15c. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | ***Лист*** |
|  |  |  |  |  |
| *54* |
| ***Изм.*** | ***Лист*** | ***№ докум.*** | ***Подп.*** | ***Дата*** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ**    1. **Исходные данные для расчёта экономического эффекта**   Данный дипломный проект направлен на создание веб-платформы для эффективных тренировок в спортивном зале.  Разработка программного продукта предусматривает проведение практически всех стадий проектирования и относится к первой группе сложности.  Последовательность расчётов:   * + - Расчёт объёма функций программного модуля.     - Расчёт полной себестоимости программного модуля.     - Расчёт отпускной цены и прибыли программного модуля.   **4.2 Расчёт объёма функций программного модуля**  Общий объём ПО определяется исходя из объёма функций, реализуемых программой, по формуле (4.1):  𝑉0 = ∑𝑛 𝑉𝑖, (4.1)  𝑖=0  где 𝑉0 – общий объём ПО;  𝑉𝑖 – объём функций ПО;  n – общее число функций.  Расчёт общего объёма ПО предполагает определение объёма по каждой функции в строках кода. Если на стадии технико-экономического обоснования рассчитать точный объём функций невозможно, то его объём можно получить на основании прогнозной оценки фактических данных по аналогичным проектам, выполненным ранее, или с использованием нормативов по каталогу функций.  Благодаря информации о функциях разрабатываемого ПО, по каталогу определяется общий объём ПО. Этот объём корректируется на основе экспертных оценок в зависимости от условий, в которых разрабатывается ПО.  Уточнённый объём ПО (𝑉𝑦) определяется по формуле (4.2):  𝑉𝑦 = ∑𝑛 𝑉𝑦𝑖, (4.2)  𝑖=0 | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *43* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| где 𝑉𝑦𝑖 – уточнённый объём функции в строках исходного кода; n – общее число функций.  Перечень и объём функций ПО представлены в таблице 4.1.  Таблица 4.1 – Перечень и объём функций программного обеспечения | | | | | | | | | |
|  | **№ функции** | | | **Наименование (содержание) функции** | | | **Объём функции строк исходного кода** | |  |
| **По каталогу**  **(**𝑽𝒊**)** | **Уточнённый (**𝑽𝒚𝒊**)** |
| 107 | | | Организация ввода/вывода информации в интерактивном  режиме | | | 280 | 448 |
| 202 | | | Формирование базы данных | | | 1980 | 120 |
| 507 | | | Обеспечение интерфейса между  компонентами | | | 1680 | 1379 |
| 602 | | | Вспомогательные и сервисные программы | | | 470 | 300 |
| 109 | | | Управление вводом/выводом | | | 1970 | 1964 |
| 207 | | | Организация поиска и поиск в базе данных | | | 4720 | 4678 |
| 506 | | | Обработка  ошибочных и сбойных ситуаций | | | 1540 | 1510 |
| 707 | | | Графический вывод результатов | | | 420 | 510 |
| Итого | | | | | | 13060 | 10909 |
| Исходя из информации, полученной из таблицы 4.1, можно сделать вывод, что уточнённый объём ПО составил 10909 строки вместо предполагаемого количества 13060. | | | | | | | | | |
|  |  |  |  | |  | *ДП.АС59.200067 – 05 81 00* | | | *Лист* |
|  |  |  |  | |  | *44* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| * 1. **Расчёт полной себестоимости программного модуля**   Стоимостная оценка программного обеспечения у разработчика предполагает составление смены затрат, которая включает следующие статьи расходов:   * + - заработную плату исполнителей (основную – ЗПо и дополнительную – ЗПд);     - отчисления на социальные нужды (Рсоц);     - материалы и комплектующие изделия (Рм);     - спецоборудование (Рс);     - машинное время (Рмв);     - расходы на научные командировки (Рнк);     - прочие прямые расходы (Рпр);     - накладные расходы (Рнр);     - затраты на освоение и сопровождение программного средства (Ро и Рсо).   Полная себестоимость (Сп) разработки программного продукта рассчитывается как сумма расходов по всем статьям с учётом рыночной стоимости аналогичных продуктов.  Основной статьёй расходов на создание программного продукта является заработная плата разработчиков (исполнителей), в число которых принято включать инженеров-программистов, руководителей проекта, системных архитекторов, дизайнеров, разработчиков баз данных, Web-мастеров и других специалистов, необходимых для решения специальных задач в команде.  Для расчёта заработной платы разработчиков программного продукта необходимо предварительно определить:   * + - продолжительности времени разработки ( Фрв ), которое устанавливается экспертным путём с учётом сложности, новизны программного модуля и фактически затраченного времени. В данном дипломном проекте Фрв = 1 месяц;     - количества разработчиков программного обеспечения. В данном дипломном проекте – один разработчик (инженер-программист 7 разряда).   Заработная плата разработчиков определяется как сумма основной и дополнительной заработной платы всех разработчиков проекта.  Основная заработная плата каждого исполнителя определяется по формуле (4.3):  ЗП = Т ∗ Тк ∗ Ф ∗ К , (4.3)  осн ст.1р Фэфф.р.в. рв пр  где Тст.1р – ставка 1 разряда рабочего (на дату написания дипломного проекта – 250 бел.руб.);  Тк – тарифный коэффициент согласно разряду исполнителя; | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *45* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Фэфф.р.в. – среднее количество рабочих дней – 22 дня;  Фр.в. – фонд рабочего времени исполнителя (продолжительность разработки программного модуля, дни);  Кпр – коэффициент премии, Кпр = 1,4.  Проведём расчёт основной заработной платы по формуле 4.3. Тарифный коэффициент согласно 7 разряду инженера-программиста Тк = 1,47. Продолжительность разработки информационной системы – 1 месяц.  Дополнительная заработная плата каждого исполнителя ( Ндоп.зп. = 15%).  Она рассчитывается от основной заработной платы по формуле (4.5):  Ндоп.зп  ЗПдоп = ЗПосн ∗ . (4.5)  100%  Результаты расчёта заработной платы внесём в таблицу 4.2.  Таблица 4.2 – Расчет заработной платы | | | | | | | | | | | | | |
|  | | **Категория работников** | | **Разряд** | **Тарифные коэффициент (**Тк) | | Фэфф.р.в.**, (дн.)** | **Коэффициент премии**  (Кпр**)** | **Заработная плата, бел.руб.** | | |  | |
| **Основная** | **Дополнительная** | **Всего** |
| Инженер-  программист | | 7 | 1,47 | | 22 | 1,4 | 701,59 | 105,24 | 806,83 |
| Исходя из таблицы, заработная плата инженера-программиста составляет 806 бел. руб. 83 коп.  Отчисления на социальные нужды ( Рсоц ) определяются по формуле (4.6) в соответствии с действующим законодательством по нормативу (29% - отчисления в ФСЗН + 6% отчисления по обязательному страхованию):  Р = (ЗП + ЗП ) ∗ 35% . (4.6)  соц осн доп 100%  Расходы по статье специальное оборудование ( Рс ) включает затраты на приобретение технических и программных средств для разработки конкретного ПП, включая расходы на проектирование, изготовление, отладку и т.д. Поскольку в данном дипломном проекте спецоборудование не применялось, данная статья не рассчитывается. | | | | | | | | | | | | | |
|  |  |  |  | |  | *ДП.АС59.200067 – 05 81 00* | | | | | | | *Лист* |
|  |  |  |  | |  | *46* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| По статье «Материалы и комплектующие изделия» (Рм) отражаются расходы на магнитные носители, бумагу, красящие ленты и другие материалы, необходимые для разработки программного продукта. Норма расхода материалов в суммарном выражении определяется в расчете на 100 строк исходного кода по формуле (4.7).  Р = Н ⋅ 𝑉0 , (4.7)  м м 100  где 𝑉0 – уточнённый общий объём функций строк исходного кода. Согласно расчётам пункта 4.2 данное значение равно 10909 строки;  Нм – норма расхода материалов в расчете на 100 строк исходного кода  программного продукта. Принимаем равной 0,5 бел. руб.  Расходы по статье «Машинное время» включают в себя оплату машинного времени, необходимого для разработки и отладки ПП. Они определяются в машино-часах по нормативам на 100 строк исходного кода машинного времени. Рмв определяется по формуле (4.8).  Р = Ц ⋅ 𝑉0 ⋅ Н , (4.8)  мв мв𝑖 100 мв  где Цмв𝑖 – цена одного машинного часа (принимается в размере 0,5 бел. руб.);  V0 – уточнённый общий объём машинного кода;  Нмв – норматив расхода машинного времени на отладку 100 строк кода, машино-часов. Принимается в размере 0,7.  Расходы на научные командировки ( Рнк ) берутся либо по смете научных командировок, разрабатываемой на предприятии, либо в процентах от основной заработной платы исполнителей (10-15%). В данном дипломном проекте данная статья не рассчитывается, так как научные командировки не предусмотрены.  Расходы по статье «Прочие затраты» (Рпр) включают затраты на приобретение специальной научно-технической информации и специальной литературы. В данном дипломном проекте данная статья не рассчитывается, так как специальная научно- техническая информация и специальная литература не приобреталась.  Затраты по статье «Накладные расходы» ( Рнр ) связаны с содержанием  вспомогательных хозяйств, а также с расходами на общехозяйственные нужды. Определяется по нормативу в процентах к основной заработной плате по формуле (4.9):  Ннр  Рнр = ⋅ ЗПосн, (4.9)  100%  где Ннр – норматив накладных расходов, в данном дипломном проекте норматив накладных расходов равен 80%. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *47* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сумма вышеперечисленных расходов по статься на программный продукт служит исходной базой для расчёта затрат на освоение и сопровождение программного продукта. Они рассчитываются по формуле (4.10).  СЗ = ЗПосн + ЗПдоп + Рсоц + Рм + Рмв + Рнк + Рпр + Рнр. (4.10)  Затраты на освоение программного продукта ( Ро ). Организация-разработчик участвует в освоении программного продукта и несёт соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по установленному нормативу от суммы затрат по формуле (4.11).  Р = СЗ ⋅ Но , (4.11)  о 100%  где СЗ – сумма вышеперечисленных расходов по статьям на разработку программного продукта;  Но – установленный норматив затрат на освоение. Для данного дипломного  проекта принимается равной 8%.  Затраты на сопровождение программного продукта ( Рсо ). Организация- разработчик осуществляет сопровождение программного продукта и несёт расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Эти расходы рассчитываются по формуле (4.12).  Р = СЗ ∗ Нсо , (4.12)  со 100%  где СЗ – сумма вышеперечисленных расходов по статьям на разработку программного продукта;  Нсо – установленный норматив затрат на сопровождение программного  продукта. Для данного дипломного проекта принимается равной 7%.  Полная себестоимость (СП) разработки программного продукта рассчитывается как сумма расходов по всем статьям. Она определяется по формуле (4.13).  СП = СЗ + Ро + Рсо, (4.13)  Результаты вычислений внесём в таблицу 4.3. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *48* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица 4.3 – Себестоимость программного продукта | | | | | | | | | | |
|  | | | **Наименование статей**  **затрат** | | | | **Норматив, %** | **Сумма затрат,**  **бел.руб.** |  | |
| 1. Заработная плата, всего | | | | – | 806,83 |
| 2. Отчисления на  социальные нужды | | | | 35 | 282,39 |
| 3. Cпецоборудование | | | | Не применялось | – |
| 4. Материалы | | | | – | 54,55 |
| 5. Машинное время | | | | – | 38,18 |
| 6. Научные  командировки | | | | Не планировались | – |
| 7. Прочие затраты | | | | – | – |
| 8. Накладные расходы | | | | 80 | 561,27 |
| 9. Сумма затрат | | | | – | 1743,22 |
| 10. Затраты на  освоение | | | | 8 | 139,46 |
| 11. Затраты на  сопровождение | | | | 7 | 122 |
| 12. Полная  себестоимость | | | | – | 2004,68 |
| В результате всех расчётов полная себестоимость программного продукта составила 2004 бел. руб. 68 коп.  **4.4 Расчёт цены и прибыли по программному продукту**  Для определения цены программного продукта необходимо рассчитать плановую прибыль, которая рассчитывается по формуле (4.14).  П = СП ∗ 𝑅 , (4.14)  100%  где СП – полная себестоимость программного модуля, бел. руб.;  R – уровень рентабельности программного модуля. В данном дипломном проекте уровень рентабельности равен 15%. | | | | | | | | | | |
|  |  |  | |  |  | *ДП.АС59.200067 – 05 81 00* | | | | *Лист* |
|  |  |  | |  |  | *49* |
| *Изм.* | *Лист* | *№ докум.* | | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| После расчета прибыли от реализации по формуле (4.15) определяется прогнозируемая цена программного продукта без налогов.  Цп = СП + П, (4.15)  где СП – полная себестоимость информационной системы учёта ресурсов веб- платформы, бел. руб.;  П – плановая прибыль от реализации программного модуля, бел. руб.  Отпускная цена (цена реализации) программного продукта включает налог на добавленную стоимость и рассчитывается по формуле (4.16):  Цо = СП + П + НДСпп, (4.16)  где СП – полная себестоимость информационной системы учёта ресурсов веб- платформы, бел. руб.;  П – плановая прибыль от реализации информационной системы, бел. руб.;  НДСпп – налог на добавленную стоимость.  Для данного программного продукта НДСпп рассчитывается по формуле (4.17).  НДС = Ц ∗ НДС , (4.17)  пп п 100%  где Цп – прогнозируемая цена, бел. руб.;  НДС – налог на добавленную стоимость. В настоящее время он составляет 20%.  Прибыль от реализации программного продукта за вычетом налога на прибыль является чистой прибылью (ПЧ). Чистая прибыль остаётся организации-разработчику и представляет собой экономический эффект от создания нового программного продукта. Она рассчитывается по формуле (4.18):  ПЧ = П ∗ (1 − Нп ), (4.18)  100%  где П – плановая прибыль от реализации информационной системы, бел. руб.;  Нп – ставка налога на прибыль. В настоящее время она равна 20%.  Все расчёты цены и прибыли по программному продукту сведены в таблицу 4.4. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *50* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица 4.4 – Расчёт отпускной цены и чистой прибыли программного модуля | | | | | | | | | | |
|  | | **Наименование статей**  **затрат** | | | | | **Норматив, %** | **Сумма затрат, бел.**  **руб.** |  | |
| 1. Полная себестоимость | | | | | – | 2004,68 |
| 2. Прибыль | | | | | 15 | 300,72 |
| 3. Цена без НДС | | | | | – | 2305,4 |
| 4. НДС | | | | | 20 | 461,08 |
| 5. Отпускная цена | | | | | – | 2766,48 |
| 6. Налог на прибыль | | | | | 20 | 60,14 |
| 7. Чистая прибыль | | | | | – | 240,57 |
| В ходе произведенных расчетов определены основные экономические показатели:   * полная себестоимость – 2004 бел. руб. 68 коп.; * прогнозируемая цена – 2766 бел. руб. 48 коп.; * чистая прибыль – 240 бел. руб. 57 коп.   Разработанный программный продукт имеет малое количество конкурентов с более высокими ценами на их услуги. Таким образом, рассчитанная отпускная цена на программный продукт, разрабатываемой в рамках данного дипломного проекта, является конкурентоспособной. При расчете цены учтены отчисления в фонд социальной защиты, а также налоги, необходимые к уплате. К конечному итогу получаем окончательную цену продукта, равную 2766 бел. руб. 48 коп. | | | | | | | | | | |
|  |  | |  |  |  | *ДП.АС59.200067 – 05 81 00* | | | | *Лист* |
|  |  | |  |  |  | *51* |
| *Изм.* | *Лист* | | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. **ЭНЕРГО- И РЕСУРСОСБЕРЕЖЕНИЕ**    1. **Энергосбережение и государственное регулирование**   Энергосбережение, или экономия электроэнергии, включает правовые, организационные, научные, производственные, технические и экономические меры, направленные на эффективное использование и экономное расходование топливно- энергетических ресурсов, а также на вовлечение в хозяйственный оборот возобновляемых источников энергии. Это является важной задачей для сохранения природных ресурсов.  Повышение энергоемкости производства, увеличение количества техники в производственных процессах и рост цен на энергоносители делают вопрос экономии электроэнергии особенно актуальным. Хотя универсальных способов экономить электроэнергию не существует, разработаны методики, технологии и устройства, которые выводят энергосбережение на новый уровень.  Экономия электроэнергии требует стратегического подхода для эффективного использования производственных мощностей с минимальными энергетическими затратами. Это включает использование энергосберегающих технологий, направленных на снижение потерь электроэнергии при работе оборудования. Применение таких технологий и устройств не только сокращает потребление электроэнергии, но и приносит дополнительные преимущества. Особенно важно внедрение энергосберегающего оборудования на производственных комплексах, где нерациональное использование электроэнергии приводит к значительным финансовым потерям.  Для укрепления экономической безопасности государства 14 июня 2007 года Президент Республики Беларусь подписал Директиву № 3 “Экономия и бережливость – главные факторы экономической безопасности государства”.   * 1. **Ресурсосбережение и государственные требования к ресурсосбережению**   Ресурсосбережение подразумевает снижение материалоемкости продукции, увелич ение выхода конечной продукции и сокращение потерь в производственном проце  ссе с применением новейших технологий и техники. В Республике Беларусь дейс твует межгосударственный стандарт, разработанный Межгосударственным тех ническим комитетом по стандартизации МТК 111. Этот стандарт является ос новополагающим и определяет цель, задачи, объекты, принципы, термины и к лассификацию требований рационального использования и экономного | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *50* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| расходованияматериальных ресурсов на всех этапах жизненного цикла веществ, материалов, изделий и продукции при выполнении работ и оказаний услуг юридическим и физическим лицам.  Цель стандартизации в области ресурсосбережения заключается в создании организационно-методической и нормативной базы для государственной технической политики, направленной на снижение ресурсоемкости без ухудшения условий экономического развития страны и при обеспечении высоких потребительских свойств продукции. | | | | | | |
|  |  |  |  |  | *ДП.АС59.200067 – 05 81 00* | *Лист* |
|  |  |  |  |  | *51* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |