СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc41492667)

[1 Теоретические аспекты разработки программного продукта 7](#_Toc41492668)

[1.1 Обзор существующих Web-сайтов 7](#_Toc41492669)

[1.2 Концепция создания Web-сайта 8](#_Toc41492670)

[1.3 Инструменты и технологии создания Web-сайта 9](#_Toc41492674)

[2 Программная структура сайта 11](#_Toc41492692)

[2.1 Описание среды проектирования 11](#_Toc41492693)

[2.2 Этапы разработки сайта 12](#_Toc41492699)

[2.3 Особенности внедрения программного продукта 20](#_Toc41492729)

[2.4 Рекомендации по эксплуатации программного продукта 21](#_Toc41492737)

[ЗАКЛЮЧЕНИЕ 39](#_Toc41492750)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40](#_Toc41492752)

[ПРИЛОЖЕНИЕ А 42](#_Toc41492753)

[ПРИЛОЖЕНИЕ Б 52](#_Toc41492756)

ГРАФИЧЕСКИЙ МАТЕРИАЛ.................................................................на диске

ЛИСТ1 НПК.09.02.03.1774.25.ВО Архитектура приложения

ЛИСТ2 НПК.09.02.03. 1774.25.ВО Схема web-сайта

# ВВЕДЕНИЕ

С приходом интернета каждый пользователь может найти информацию по любому интересующему ему вопросу, получить любой ответ, а также посмотреть на любую точку планеты со снимка спутника. Исследуя интернет, часто можно встретить привлекательные изображения живописных мест в социальных сетях или рекламе, в которых хочется побывать самому. Люди все чаще стали делиться фотографиями со своих отпусков или интересных туров с приходом технологий.

Актуальность разработки Web- сайта по теме «Travelling with Honda CL 400» заключается в том, что данная тема стала популярна в 21 веке, особенно в последние годы, когда качество фотографий на телефон не отличается от фотоаппарата, благодаря чему каждый может сделать привлекательные снимки и поделиться этим с миром. Люди, интересующиеся путешествиями или собирающиеся в тур, захотят посетить данную страницу, поделиться эмоциями и повторить маршрут автора.

Цель: Разработка информационного Web-сайта о путешествии на мотоцикле.

Задачи:

* Изучить современные технологии разработки Web-сайта;
* Разработать структуру сайта;
* Разработать листинг программы;
* Осуществить отладку программного продукта;
* Поставить сайт на доверенный и доступный хостинг;
* Провести тестирование программы;
* Ввести в эксплуатацию программу;
* Осуществить расчет технико-экономических показателей разработки программного продукта;
* Оформить документацию и получить рецензию на программный продукт.

Объект исследования: Web-сайт.

Предмет исследования: технология создания программных продуктов c использованием языка гипертекстовой разметки, а также кроссплатформенной среды разработки ASP.NET Core.

Методы исследования: информационно-коммуникационные технологии.

# 1 Теоретические аспекты разработки программного продукта

## Обзор существующих Web-сайтов

## Обзор существующих web-сайтов по теме «Travelling with Honda CL 400» представляет большой перечень ресурсов как информационного плана, так и порталов по типу «Форум».

Ярким представителем является сайт TripAdvisor. Сайт наполнен информацией по будущим проектам и является источником заказа своих товаров. Так же на сайте находиться форум для обсуждения пользователями проектов разработчиков, но диалоги ведутся на иностранных языках.

Один из крупнейших туристических сайтов в мире. Главная полезность сервиса — отзывы путешественников обо всём на свете: городах, достопримечательностях, отелях, ресторанах, экскурсиях и так далее. Сайт представляет собой поисковую систему по статьям, написанными как редакцией портала и компаниями-партнерами, так и пользователями-путешественниками, которые пожелали поделиться и рассказать всему миру о своей поездке.

## 1.2 Концепция создания Web-сайта

## Как и говорилось выше, тема путешествий включает в себя все больше любителей и профессионалов по всему миру, а компании пытаются побудить новых людей к участию в турах.

## Главной проблемой пользователя становится доверие к крупным информационным порталам или проще говоря осторожность при виде слишком идеальных фотографий, недосказанных деталей или подозрительно хороших отзывов об отеле, маршруте, или придорожном кафе. Создается впечатление, что данная статья является рекламой.

## Создав Web-ресурс по теме «Travelling with Honda CL 400» можно решить проблему с недоверием: путешествие, которое вдохновило автора на создание web-страницы с информацией, касающейся не только маршрута движения, а так же способов ночлега, самочувствия после дороги и рассказе о транспорте, на котором производился тур, создает впечатление настоящего, ничем не подкупленного отзыва. Особенно на это влияют стандартные фотографии рядом с красивыми пейзажами, которые говорят о том, что автор хочет поделиться с миром информацией, будто рассказывает своему другу или знакомому, не стесняясь трудностей в процессе путешествия.

## 1.3 Инструменты и технологии создания Web-сайта

## Технологии, использованные для создания веб-ресурса разделены на подвиды:

## Разметка веб-документа;

## Языки программирования;

## Графика и файловая система;

## Программы и технологии;

## Система управления базой данных

## Разметка веб-документа.

## HTML (HyperText Markup Language) — Это стандартный язык разметки практически любого веб-ресурса. Он является заключающим элементом, который воспроизводит браузер пользователя, создает макет представления интерфейса.

## CSS (Cascading Style Sheets) — Стили CSS — файл (или несколько файлов), в содержании которого находятся стили примененные к различным HTML тегам. Стили CSS по своей сути являются технологией, которая позволяет изменять цвета, размеры и расположение блоков, текста и любых других элементов HTML-разметки.

## Языки программирования

## C#. Язык программирования C# применяется для разработки приложений любого типа для платформы .NET, выпущенной компанией Microsoft. Это объектно-ориентированный язык с С- подобным синтаксисом, поддерживающий основные парадигмы ООП. На нем реализована серверная часть Web-сайта, которая отвечает за логику приложения.

## JavaScript. Технология JavaScript используется для повышения функционала ресурса. Наиболее часто JavaScript используется для придания некой интерактивности сайту (плавающие элементы, параллакс, несложная анимация и многое другое).

## Графика и файловая система

## Файловая система в виде папок-каталогов с исполнительными файлами для сайта (.html, .css, .cs, .js документы) и графика.

## Использование картинок для поддержки любым браузером в расширениях: jpg

## Программы и технологии, упростившие технологию создания сайта:

## Adobe Photoshop CS6 – Редактор фотографий и GIF файлов, имеющий большое количество встроенных функций и сервисов. Использовался для редактирования и изменения картинок, в том числе в формате Web.

## Microsoft Visual Studio 2019 – интегрированная среда разработки, поддерживающая С-подобные языки программирования, а так же имеющая встроенный редактор для создания Web-страниц с использованием HTML, CSS, JavaScript. Использовалась версия для учебного использования.

## ASP.NET Core - представляет технологию от компании Microsoft, предназначенную для создания различного рода веб-приложений: от небольших веб-сайтов до крупных веб-порталов и веб-сервисов.

## EntityFramework Core 5. Технология доступа к данным от компании Microsoft. EF позволяет работать с базой данных на более абстрактном уровне: абстрагирование от самой базы данных и её таблиц без оперирования первичными и внешними ключами. Вместо этого идет общение с сущностями, объектами, независимо от типа хранилища. Позволяет создавать запросы к БД без использования SQL.

## Система управления базой данных:

## Microsoft SQL Server – система управления реляционными базами данных. Используемый язык запросов T-SQL. Применима при проектировании как персональных, так и крупных информационных систем. Использовалась учебная версия продукта.

## 2 Программная структура сайта

## 2.1 Описание среды проектирования

## Разработка проводилась в Microsoft Visual Studio 2019.

## При разработке HTML – страницы выполняется создание структуры и разметка элементов пользовательского интерфейса с использованием «тегов», каждый из которых является уникальным и имеющих определенный набор свойств.

## При написании, выходящих за рамки традиционной, статической методологии разработки web-страниц (то есть HTML), используется язык программирования C#, отвечающий за серверную часть приложения. Однако его возможно использовать не только на «скрытой» от пользователя стороне, но и выводить данные на пользовательский интерфейс с использованием технологии Razor pages, позволяющей добавлять к структуре html страницы код C#

## В менеджере расширений Visual Studio NuGet существует огромная коллекция библиотек, позволяющая упростить и решить задачу любой сложности. При создании web-сайта были установлены библиотеки:

## Razor.RuntimeCompilation – позволяет в режиме реального времени производить перекомпилирование страницы без перезапуска проекта, обновив страницу в браузере.

## EntityFramework Proxies – использование «ленивой» загрузки при отправке запроса на получение данных из БД. Ленивая загрузка означает, что данные будут поступать на сервер только по мере необходимости (выполнение запроса, принимающий выходные параметры)

## EntityFramework SqlServer – позволяет применять технологию EF Core к СУБД MS SQL Server. Содержит встроенный набор функций для упрощенного написания запросов.

## Среда разработки позволяет поставить «точку останова» в конкретной строке программы для отладки и просмотра значения переменных.

## 2.2 Этапы разработки сайта

## 1 этап. Определение целей разработки сайта, Понимание зачем и для кого разрабатывается web-сайт?

## Первоочередное значение для меня имеет авторский стиль рассказа, надежность работы сайта и легкое донесение информации, а также привлекательный интерфейс. Важно рассказать пользователю о своем путешествии так непринужденно, что бы процесс чтения был увлекательным. Так же сделать визуальное оформление, соответствующее тематике для создания правильной атмосферы рассказа. Задача воодушевить пользователей на прохождение аналогичного маршрута или в целом путешествовать, с последующим обсуждением. В дополнении к основному материалу мой опыт использования мотоцикла Honda с фотографиями, характеристиками и проблемами эксплуатации.

## Подготовить наполненность сайта. Предоставить контактную информацию с разработчиком.

## 2 этап. Создание макета сайта.

## Разработка и согласование выбранного варианта концепции web-сайта;

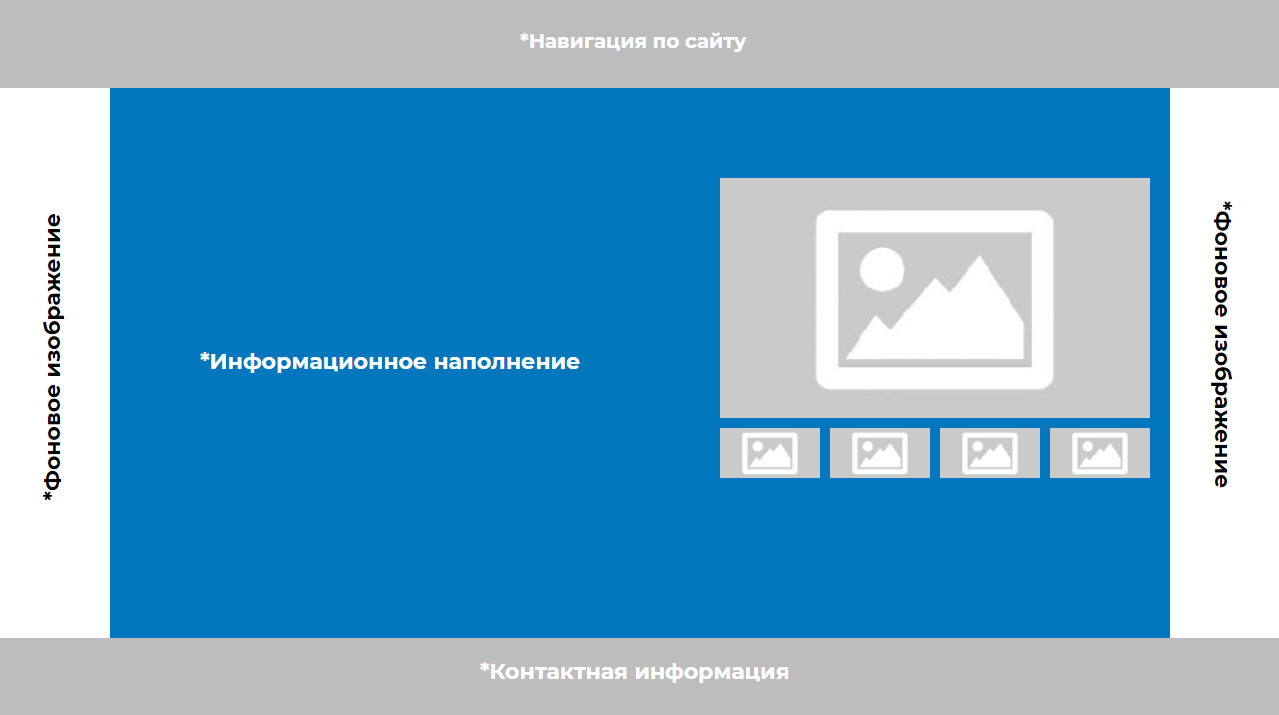
## Разработка структуры web-сайта (ПРИЛОЖЕНИЕ Б);

## Создание макетов внутренних страниц;

## 

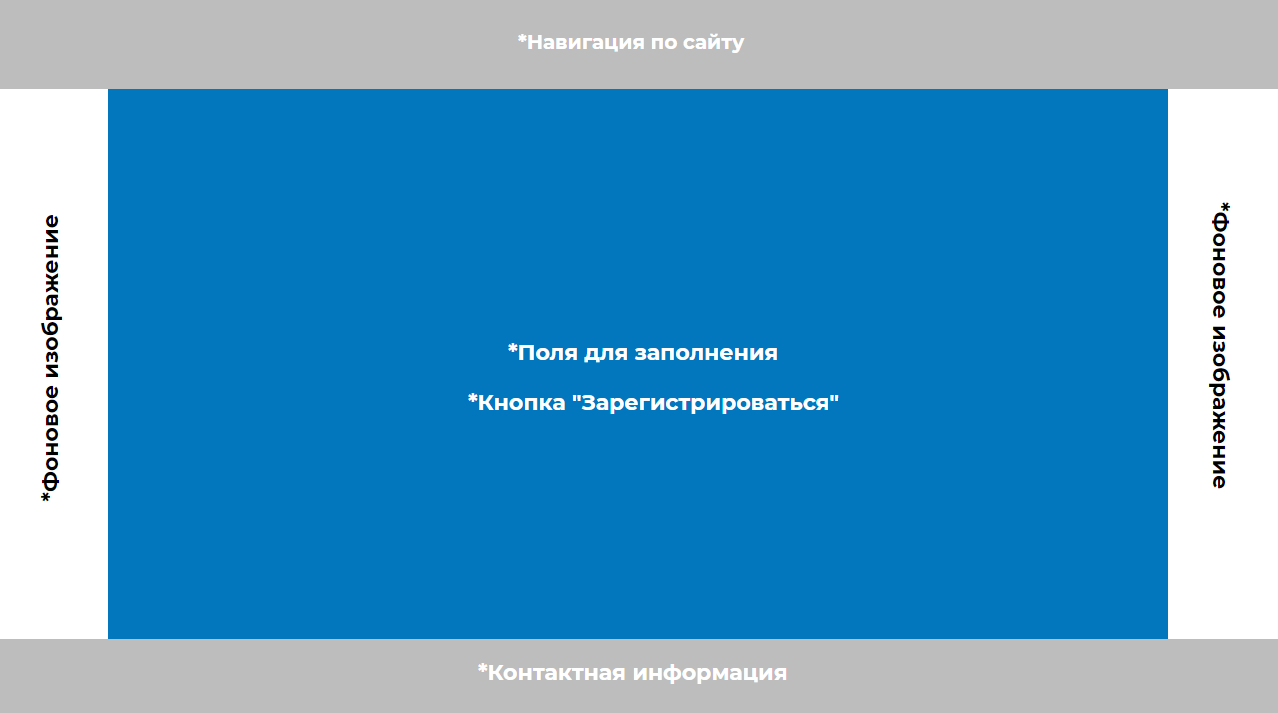
Изображение 1. Макет начальной страницы «О Туре»

## Данный макет был разработан на портале Wilda.ru. Для страницы «О туре» использовались блоки, а также вставка текста и его поворот по горизонтали.



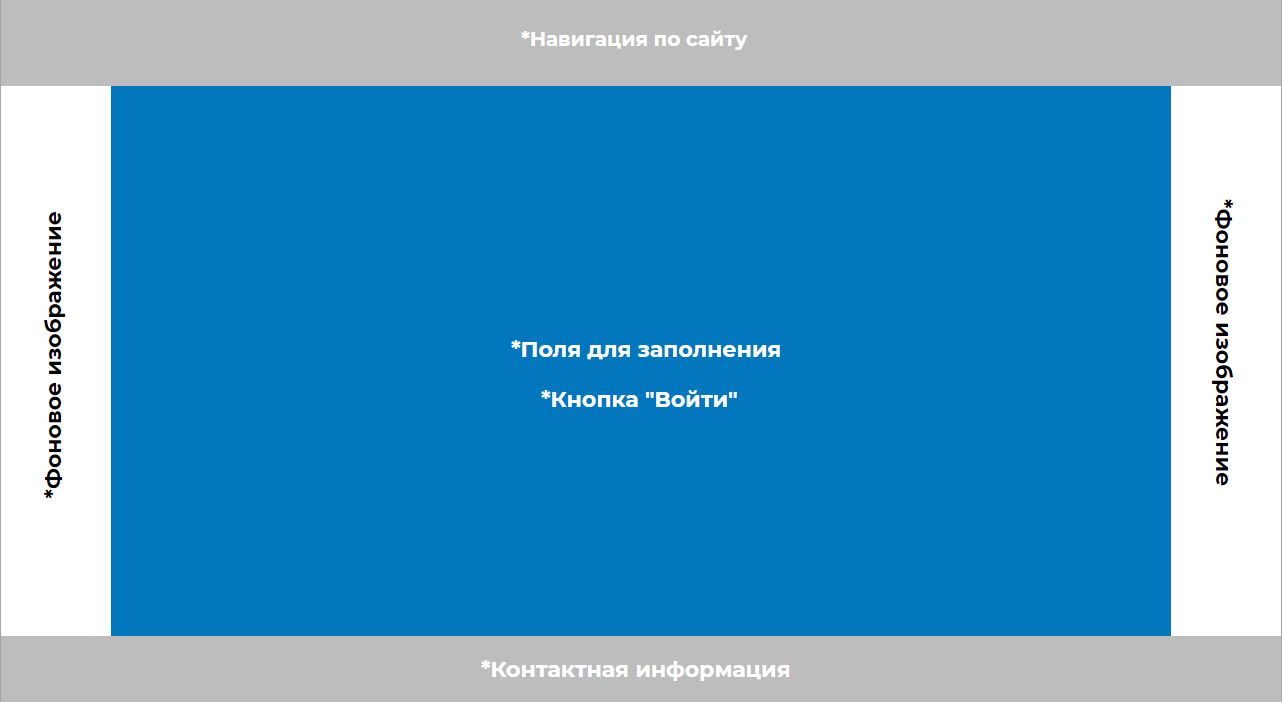
Изображение 2. Макет страницы “Галерея моего мотоцикла”

## Данный макет был разработан на портале Wilda.ru. Для страницы «Галерея моего мотоцикла» использовались блоки, изображения, а также вставка текста и его поворот по горизонтали.



Изображение 3. Макет страницы «Зарегистрироваться»

## Данный макет был разработан на портале Wilda.ru. Для страницы «Зарегистрироваться» использовались блоки, а также вставка текста и его поворот по горизонтали.



Изображение 4. Макет страницы «Войти»

## Данный макет был разработан на портале Wilda.ru. Для страницы «Войти» использовались блоки, а также вставка текста и его поворот по горизонтали.

## 

Изображение 5. Макет страницы «Форум»

## Данный макет был разработан на портале Wilda.ru. Для страницы «Форум» использовались блоки, а также вставка текста и его поворот по горизонтали.

## Основа макета каждой из страниц прописана в файле Layout.html: эта часть интерфейса отображается на всех страницах приложения по умолчанию.

## Доработка и согласование каждой из типовых страниц;

## 3 этап. Верстка, программирование и подключение базы данных

## Web-приложение построено с применением паттерна «MVC». Данный подход разделяет архитектуру приложения на три составляющие

## Model - этот компонент отвечает за данные, а также определяет структуру приложения.

## View - этот компонент отвечает за взаимодействие с пользователем. То есть код компонента view определяет внешний вид приложения и способы его использования.

## Controller - интерпретирует действия пользователя, оповещая модель о необходимости изменений

## Благодаря такой архитектуре изменение и дополнение программы происходит модульно, не затрагивая остальные части программы.

## Компонент Model:

## Данный компонент содержит в себе модели, определяющие структуру приложения. Для реализации регистрации, аутентификации и авторизации пользователя были добавлены следующие модели: UserModel, описывающая свойства пользователя; RoleModel, содержащая в себе информацию о правах пользователя «Администратор» или «Пользователь»; RegisterModel, описывающая поля для регистрации, а так же атрибуты обязательного наличия; LoginModel, описывающая обязательные для заполнения поля при входе; MessageModel, содержащая информацию о сообщениях для форума.

## Далее необходимо создать базу данных и её архитектуру. Для упрощения взаимодействия с базой данных использовалась технология EntityFramework, в частности, подход Code First. Данный подход предполагает создание базы данных на основе моделей, которые расположены в папке «Models». Для того, чтобы EF правильно распознал иерархию классов, добавим в каждую из моделей уникальный номер ID. Так же вложим в класс Пользователя ссылку на класс Ролей, в класс Сообщение – ссылку на класс пользователя. Необходимая иерархия классов построена.

## Модель ApplicationContext необходима для явного указания таблиц, с которыми мы будем работать. Создаем наборы данных по каждой из моделей UserModel, RoleModel, MessageModel. В методе OnModelCreating прописываем параметры создания БД, а именно: при первом запуске создается новая БД, но если она уже существует, то перезаписи не происходит. Так же для упрощения тестирования создадим по экземпляру администратора и пользователя и занесем их в БД при создании.

## В файле Startup.cs создадим метод ConfigureServices, отвечающий за подключение к БД. Так же добавим куки файлы для запоминания пользователя, что избавит от постоянного входа в приложение. Теперь наше приложение имеет базу данных и подключение к ней.

## Компонент View:

## Данный компонент реализует в себе визуальное отображение страницы, а также пользовательские элементы и различные эффекты при взаимодействии со страницей. Отдельно были добавлены страницы AdminPanel и UserPanel, содержащие в себе личный кабинет пользователя. Разделение необходимо в силу разных возможностей. В файле Layout добавлена фоновая картинка, а также шапка сайта. Элементы, содержащиеся в данном файле, видны на любом странице Web-приложения.

## Стиль и наполнение основных страниц сайта находится в файлах About\_Motorcycle, About\_Tour, My\_Experience, Forum.

## Кнопки для перехода по страницам, а также кнопки входа и отправки сообщений посылают запрос на определенный для каждой из целей контроллер.

## Компонент Controller:

## Данный компонент оповещает модель о том, что ей необходимо изменяться. Так же используется для перенаправления по страницам сайта. AccountController описывает логику регистрации, авторизации и аутентификации пользователя. Метод Register с атрибутом Post добавляет нового пользователя в базу данных после проверки на количество введенных полей при регистрации. Метод Login производит аутентификацию пользователя по его роли для выдачи определенных прав.

## В AdminController реализована логика удаления пользователя из базы данных методом DeleteUser с атрибутом Post.

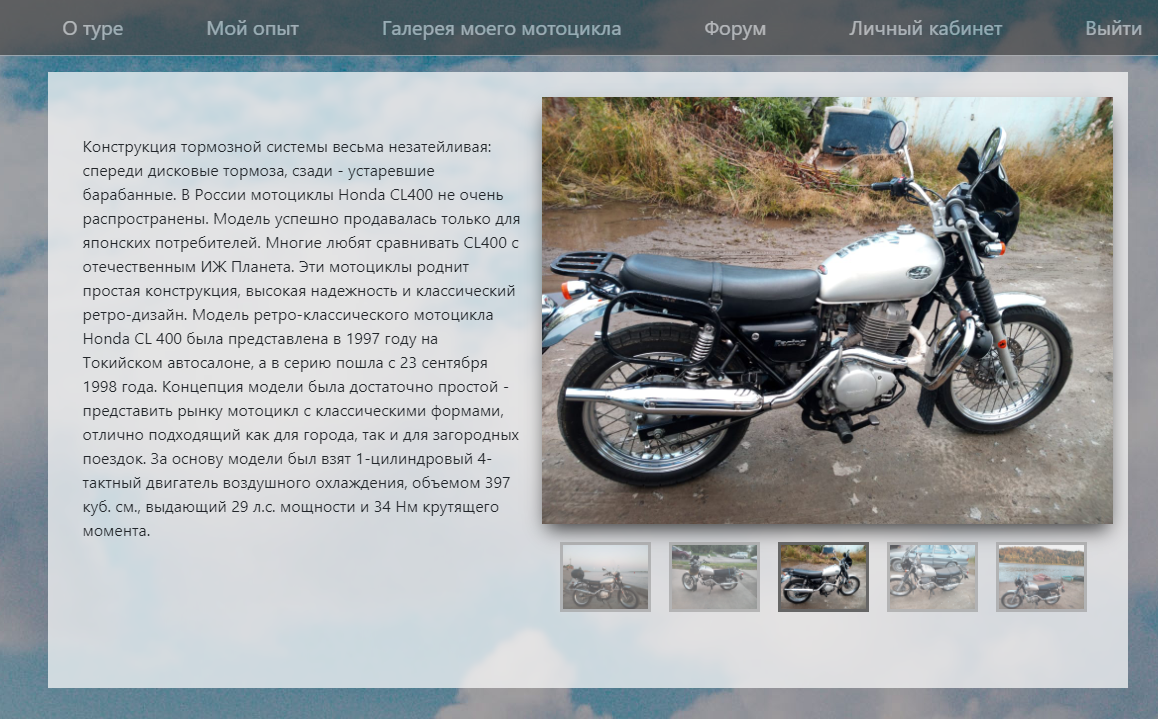
## UserController вызывается лишь при авторизации пользователя и необходим для переключения между страницами сайта.

## 4 этап. Разработка интерфейса сайта.

## Наполнение сайта осуществляется согласно выбранной тематики. В интерфейсе предполагаются следующие страницы: туре, мой опыт, галерея моего мотоцикла. Каждая страница содержит текстовую информацию о разделе и фотографии.



Изображение 5. Страница «О туре»



Изображение 6. Страница «Галерея моего мотоцикла»

## 5 этап. Запуск сайта в Интернете.

## Для того, чтобы сайт стал доступен любому пользователю, его необходимо опубликовать на хостинг-сервисе. В качестве такового был выбран somee.com. Данный сервис предоставляет возможность бесплатно опубликовать 1 сайт, построенном на технологии .NET/ASP.NET, поэтому он подходит как нельзя лучше. Для того, чтобы это сделать, необходимо выбрать в среде разработки Visual Studio 19 меню «Сборка> Опубликовать». Указываем FTP адрес и ждем публикации. После небольшого ожидания вбиваем в строку браузера www.TravelingWithHonda.somee.com и переходим на сайт.

## 6 этап. Тестирование сайта.

## Тестирование проходило как серверной части, так и визуальной составляющей. Встречались ошибки отказа в регистрации, неправильный занос данных в БД. Благодаря использованию встроенного в среду разработки средства «Точка останова» нахождение ошибок и выявление причины занимало гораздо меньше времени. С визуальной частью чаще всего встречались проблемы неправильного указания свойств стилей, неправильные ссылки, некорректное отображение фотографий, плохая адаптация сайта.

## 2.3 Особенности внедрения программного продукта

## Свой сайт я решил внедрить в интернет и распространить слух про него среди знакомых единомышленников.

## Ввод остатков в программу – это первый этап моей работы по внедрению моего программного продукта. Он решает перечень задач:

## Наглядность сайта.

## Пользователь сразу увидит слегка приглушенное крымское небо. Начав знакомиться с рассказом о моем путешествии, он буквально с первых строк погрузиться в увлекательный рассказ о начале путешествия из Анапы в Крым. По мене чтения материала гость увидит большое количество фотографий, передающие живописность и красоту природы России.

## Возможно, он бегло прочитает и перейдет к следующему разделу, а возможно и воодушевится моим приключением и с удовольствием захочет повторить его, пусть и не на мотоцикле. После этого он может зайти в раздел «Форум» и задать интересующие вопросы не только мне, но и всем пользователям сайта. Так же он может поинтересоваться о транспорте, на котором я преодолел немалое расстояние и посмотреть на его внешний вид.

## - Изучение нюансов работы. В процессе переноса сайта на хост, я изучил разного рода нюансы. Адаптивность сайта, его работоспособность в разных браузерах и исключение возникновения ошибок, способ добавления базы данных.

## - Сотрудники и администрирование. Для администрирования сайта достаточно одного человека, так как сайт представляет собой личный блог.

## 2.4 Рекомендации по эксплуатации программного продукта

## Проверка состояния сайта в данный период времени

## Анализировать мощности хостинга при большом количестве одновременного захода пользователей: в случае проблем поменять хостинг

## Прохождение новых маршрутов путешествий для нового контента;

## Наполнение сайта информацией;

## Реагирование на типичные ошибки:

## - Визуальное оформление некорректное – это значит, что новой информации на странице не дали адаптацию в файле CSS, исправить соответственно можно в этом же файле адаптировав информацию, либо страницу.

## - Ошибка 404 - стандартный код ответа HTTP о том, что клиент был в состоянии общаться с сервером, но сервер не может найти данные согласно запросу. Для решения проблемы стоит связаться с хостом сервера и уточнить, что за проблема и когда она решится.

## - Ошибка 502 - результат запроса, который означает, что на стороне сайта возникли проблемы. Проблема чаще всего вызвана неправильной работой оборудования хостера или его неправильной настройкой. Типичной проблемой может быть неправильная настройка DNS или прокси сервера.

## - Ошибка 503 - Service Unavailable? Данная ошибка означает, что сервер с сайтом не доступен на данный момент (по техническим причинам).

## - Ошибка 504 gateway timeout – это код состояния HTTP, который появляется, когда один сервер не получает своевременный ответ от другого сервера, который действует как шлюз или прокси. Проще говоря, это означает, что сервер не смог выполнить ваш запрос в течение заданного периода времени.

2.5 Экономическое обоснование разработки программного продукта

Экономическое обоснование разработки программного продукта

Техник-программист относится к категории специалистов, принимается и увольняется на работу приказом начальника ВЦ (начальника производственного отдела ВЦ (ИВЦ), иного структурного подразделения). Техник-программист подчиняется начальнику ВЦ (ИВЦ), начальнику производственного отдела ВЦ (ИВЦ), иному должностному лицу. В своей деятельности техник-программист руководствуется: нормативно-правовыми актами, регулирующими соответствующие вопросы; методическими материалами, касающимися вопросов его деятельности; правилами трудового распорядка; приказами, распоряжениями директора предприятия (непосредственного руководителя); должностной инструкцией.

На техника-программиста возлагаются следующие функции: механизированная и автоматизированная обработка поступающей в вычислительный центр информации; участие в проектировании систем обработки данных и систем математического обеспечения машины; разработка программ; учет использования машинного времени, объемов выполненных работ. Данные виды работ регламентируются согласно Постановления Минтруда РФ от 23.07.1998 N 28"Об утверждении Межотраслевых типовых норм времени на работы по сервисному обслуживанию персональных электронно-вычислительных машин и организационной техники и сопровождению программных средств" и Приказа Минздравсоцразвития РФ от 14.10.2011 N 1175н "Об утверждении Межотраслевых типовых норм времени на работы по сервисному обслуживанию оборудования телемеханики, сопровождению и доработке программного обеспечения".

При производстве программных продуктов предприятие несёт определённые затраты. Эти производственные затраты называют годовыми издержками производства или эксплуатационными расходами, либо текущими затратами.

Себестоимость продукции - совокупность прямых издержек, связанных с производством изделия; все виды затрат, понесенных при производстве и реализации определенного вида продукции.

В себестоимости продукции следует различать затраты, обусловленные производством заданного объёма продукции и транспортно-сбытовые затраты на реализацию произведённой продукции. В связи с этим различают фабрично-заводскую себестоимость и полную себестоимость. В полную себестоимость входят затраты на передачу и реализацию продукции. Затраты, включённые в себестоимость, классифицируются:

- по экономическим элементам;

- по статьям калькуляции.

В расчёте себестоимости программного продукта учитывается: количество человек участвующих в разработке программного продукта; количество используемой компьютерной техники; количество времени, затраченного на производство программного продукта; тариф за потребление электроэнергии; эксплуатационные материалы; тарифы за использование коммуникации; транспортные расходы.

При расчетах себестоимости программного продукта используются следующие формулы:

Полезный фонд времени работы ПК

Фп = T \* Ku \* n \* t, (2.1)

где: Фп - полезный фонд, ч;

Т - количество дней работы над программой;

Ku - коэффициент использования ПК;

n - количество ПК;

t - продолжительность смены, ч.

Стоимость потребляемой энергии

Uэ\э = Фп \* Ц \* Р, (2.2)

где: Uэ\э- стоимость потребляемой энергии, руб.;

Ц - тариф за 1 кВт\*ч, руб.;

Р - потребляемая мощность ПК, кВт.

Материальные затраты - часть издержек производства, затрат на производство продукции, товаров, услуг, в которую включаются затраты на сырье, основные и вспомогательные материалы, топливо, энергию и другие затраты, приравниваемые к материальным. Материальные затраты (расходы) образуют часть себестоимости продукции.

Uмз= Uэ\э + Uэм, (2.3)

где: Uмз - материальные затраты, руб.;

Uэ\э- стоимость потребляемой энергии, руб.;

Uэм- стоимость эксплуатационных материалов, руб.

Фонд оплаты труда включает затраты на оплату труда основного производственного персонала, включая премиальные выплаты за производственные затраты, оплата ежегодных и учебных отпусков, выплаты работникам, высвобождаемых с предприятия в результате сокращения штатов и реорганизации предприятий.

Основная заработная плата за время работы над программой:

Заработная плата программиста исчисляемая в руб./мес.

ЗПосн = Т \* З, (2.4)

где: Т - время работы над программой, ч;

З - заработная плата программиста за один день, руб.

Дополнительная заработная плата включает в себя: премии, доплату за сверхурочное время, в праздничные дни, по совместительству.

ЗПдоп = 0,4 \* ЗПосн, (2.5)

где: ЗПдоп - дополнительная заработная плата, руб.;

ЗПосн - основная заработная плата за время работы над программой, руб.

Uзп = ЗПосн + ЗПдоп, (2.6)

где: Uзп - оплата труда всего, руб.;

ЗПдоп - дополнительная заработная плата, руб.;

ЗПосн - основная заработная плата за время работы над программой

В 2021 году страховые взносы по-прежнему рассчитывают, исходя из начислений физическим лицам, установленных лимитов облагаемой базы и ставок. Процентная ставка взносов напрямую зависит от суммы, облагаемой страховыми взносами.

В таблице ниже приведены актуальные для начисления в 2021 году страховые взносы, ставки с выплат работников (ст. 426 НК РФ).

Таблица 1 – Страховые взносы, ставки с выплат работников

|  |  |  |  |
| --- | --- | --- | --- |
| Отчисления | Ставка (%) | Условие | Лимит (рублей) |
| Пенсионные | 22 | До достижения | 1 465 000 |
| 10 | После превышения |  |
| Социальные на ВНиМ | 2,9 | До достижения | 966 000 |
| 0 | После превышения |  |
| Медицинские | 5,1 | Лимита нет | |
| Травматизм | Индивидуально от 0,2% до 8,5%, лимитирование отсутствует | | |

Тарифы страховых взносов в 2021 году составляет (ст.425, 426 НК РФ):

22 % — на пенсионное страхование;

5,1 % — на медицинское страхование;

2,9 % — на социальное страхование;

0,2 % ‑ по травматизму.

При этом сумма взносов к уплате зависит от того, превысил доход установленный лимит или нет.

Пенсионные взносы в пределах лимита надо платить по ставке 22 %, с доходов сверх лимита — по ставке 10 % с суммы превышения лимита.

Социальные взносы платят по ставке 2,9 % в пределах лимита, свыше ‑0% , а медицинские — по общему тарифу 5,1 %.

Реквизиты для перечисления налогов и сборов расположены на сайтах государственных органов и фондов.

Если допустить, что у организации нет права на использование пониженных тарифов, то размер отчислений в фонды составит:

22% – отчисления в ПФР, страховая и накопительная части;

0,2% – отчисления в ФСС по травматизму;

2,9% – отчисления в ФСС по временной нетрудоспособности и в связи с материнством;

5,1% – отчисления в ФФОМС0% – отчисления в ТФОМС)

Uсн = Нсн/100 \* Uзп, (2.7)

где: Uсн - отчисления на социальные нужды, руб.;

Uзп - оплата труда всего, руб.;

Нсн - норма отчисления на социальные нужды, руб.

Льготные ставки страховых взносов для IT-компаний с 1 января 2021 года для организаций из сферы информационных технологий снизили страховые взносы. Вместо 14% совокупный тариф составил 7,6%, из них:

* 6% — на пенсионное страхование;
* 1,5% — на соцстрахование в связи с нетрудоспособностью и материнством;
* 0,1% — на медицинское страхование.

Право на пониженные тарифы получили организации, которые занимаются проектированием и разработкой изделий электронной компонентной базы и электронной (радиоэлектронной) продукции (пп. 18 п. 1 ст. 427 НК РФ).

IT-компании могут получить льготу по страховым взносам при соблюдении условий:

* госаккредитация в Минкомсвязи (включение в спецреестр);
* доля доходов от IT-деятельности — не менее 90% от общего дохода;
* среднесписочная численность — не менее 7 человек.

В письме № 03-07-07/111669 от 18.12.2020 Минфин разъяснил как определить долю доходов и привёл примеры применения налоговых льгот. Если IT-компания соответствует нужным критериям, можно применять новые тарифы, не сообщая об этом в налоговую.

Амортизационные отчисления ‑ отчисления части стоимости основных фондов для возмещения их износа. Амортизационные отчисления включаются в издержки производства или обращения. Производятся всеми коммерческими организациями на основе установленных норм и балансовой стоимости основных фондов, на которые начисляется амортизация, руб.

В соответствии с классификацией основных средств, электронно-вычислительная техника относится ко 2 амортизационной группе с полезным сроком использования от 2 до 3 лет.

Учетной политикой организации установлено, что при определении сроков полезного использования, используется нижняя граница установленных сроков плюс один месяц. Таким образом, срок полезного использования данного основного средства 2 года 1 месяц или 25 месяцев.

Также учетной политикой организации установлено право на единовременное признание расходов на капитальные вложения (амортизационную премию) в размере 10% по 1,2,8,9,10 амортизационным группам.

В соответствии с учетной политикой организация рассчитывает амортизацию линейным способом. Таким образом, норма амортизации для объектов основных средств, отнесенных ко 2 амортизационной группе равна 4% (1/25 х 100).

На = На% / 100 % \* Соф, (2.8)

где: На - амортизационные отчисления, руб.;

На% - норма амортизационных отчислений, руб.;

Соф - стоимость основных фондов, руб.

Прочие затраты - элемент себестоимости продукта (работ, услуг), в котором отражаются налоги, сборы, платежи, включая необязательные виды страхования, отчисления в страховые фонды (резервы) и другие обязательные отчисления, производимые в соответствии с установленным законодательством порядком.

Полные издержки

U = Uмз + Uзп + Uсн + На + Uпр, (2.9)

где: U – полные издержки, руб.;

Uмз - материальные затраты, руб.;

Uзп - оплата труда всего, руб.;

Uсн - отчисления в различные фонды на социальные нужды, руб.;

На - амортизационные отчисления, руб.;

Uпр - прочие затраты, руб.

Цена программного продукта для покупателя составляет сумму себестоимости создания программного продукта, добавочной стоимости и налога на добавочную стоимость. Вычисляется по формуле:

Спр = Ссб + Сдб + Сндс, (2.10)

где: Спр – цена программного продукта, руб.;

Ссб – себестоимость программного продукта, руб.;

Сдб – добавочная стоимость (20 %), руб.;

Сндс – налог на добавочную стоимость, руб.

Расчет стоимости web- сайта «Travelling with Honda CL 400»

1 Количество компьютеров - 1 шт.

2 Количество дней работы над программой – 44 дней.

3 Коэффициент использования ПК – 0,98

4 Средняя продолжительность рабочей смены – 8 ч

5 Тариф за потребление электроэнергии для юридических лиц - 4,98 руб./ кВ\*ч.

6 Потребляемая мощность ПК - 1 кВт\*ч.

Полезный фонд времени работы ПК

Фп = 345 ч

Стоимость потребляемой энергии

Uэ\э = 1718,10 руб.

Таблица 2 - Виды работ

|  |  |  |  |
| --- | --- | --- | --- |
| № п\п | Вид работы | Время, ч | В том числе работа на ПК и оргтехнике, ч |
| 1 | Разработка концепции | 13 | 10 |
| 2 | Анализ рынка ПП | 20 | 20 |
| 3 | Поиск и сравнение программ для создания ПП | 11 | 11 |
| 4 | Анализ доп. материалов | 15 | 15 |
| 5 | Разработка макета ПП | 27 | 27 |
| 6 | Разработка сайта | 186 | 186 |
| 7 | Отладка программы | 25 | 25 |
| 8 | Анализ результативности работы | 23 | 23 |
| 9 | Расчет экономических обоснований | 6 | 6 |
| 10 | Оформление документации | 26 | 22 |
|  | Итого: | 352 | 345 |

Таблица 3 - Аппаратное обеспечение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п\п | Наименование оборудования | Стоимость единицы,  руб. | Кол-во,шт. | Стоимость,  руб. |
| 1 | Монитор LG 34UC79G-B | 46 799,00 | 1 | 46 799,00 |
| 2 | Мышь проводная HyperX Pulsefre Surge | 5 399,00 | 1 | 5 399,00 |
| 3 | Клавиатура Zet Gaming Blade | 4 399,00 | 1 | 4 399,00 |
| 4 | Наушники HyperX Cloud Revolver S | 11 999,00 | 1 | 11 999,00 |
| 5 | Системный блок | 242 698,00 | 1 | 242 698,00 |
| 6 | Принтер (МФУ) Samsung SCX-4220 | 5 799,00 | 1 | 5 799,00 |
|  | Итого: |  |  | 317 093,00 |

Таблица 4 - Программное обеспечение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п\п | Наименование оборудования | Стоимость единицы,  руб. | Кол-во,шт. | Стоимость,  руб. |
| 1 | Windows 10 Pro | 13 699,00 | 1 | 13 699,00 |
| 2 | Office 365 | 2 821,00 | 1 | 2 821,00 |
| 3 | Microsoft Visual Studio Professional | 37 176,00 | 1 | 37 176,00 |
| 4 | Sublime Text 3 | 8 322,00 | 1 | 8 322,00 |
| 5 | Adobe Photoshop CC | 19 468,80 | 1 | 19 468,80 |
| 6 | GlobalScape - CuteFTP Home Edition | 3 412,94 | 1 | 3 412,94 |
|  | Итого: |  |  | 84 899,74 |

Таблица 5 - Эксплуатационные материалы

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| №  п/п | Наименование | | Стоимость единицы, руб. | | Кол-во, шт. | | Сумма,  руб. | |
| 1 | | USB Flash Kingston DataTraveler 32 ГБ | | 550,00 | | 2 | | 1 100,00 |
| 2 | | Бумага для офисной техники SvetoCopy (A4, марка C, 80 г/кв.м, 500 листов) | | 280,00 | | 2 | | 560,00 |
| 3 | | Картридж NetProduct (N-SCX-D4200A) для Samsung SCX-D4200/4220 | | 913,00 | | 1 | | 913,00 |
| 4 | | Ручка шариковая PILOT BPS-GP синяя 0,5мм | | 49,00 | | 3 | | 147,00 |
| 6 | | Блок для записей Brauberg | | 300,00 | | 1 | | 300,00 |
|  | | Итого: | |  | |  | | 3 020,00 |

Таблица 6 - Прочие расходы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п\п | Наименование расходных услуг | Стоимость единицы, руб. | Единица измерения | Стоимость, руб. |
| 1 | Проезд | 150,00 | 10 | 1500,00 |
| 2 | Интернет | 1 000,00 | 3 | 3 000,00 |
| 3 | Мобильная связь | 200,00 | 3 | 600,00 |
|  | Итого: |  |  | 5100,00 |

Материальные затраты:

Uмз = 3 020,00 + 1718,10 = 4738,10 руб.

Основная заработная плата за время работы над программой:

Расчет заработной платы техника программиста 11980 руб./мес. (за 20 рабочих дней):

Оклад за 44 рабочих дня – 26 356,00 руб.;

Премия 20 % – 5 271,20 руб.;

Районный коэффициент 70% – 22 139,04 руб.;

Северные надбавки 50% –15 813,60 руб.;

Начислено за месяц – 69 579,84 руб.;

Отчисления в различные фонды:

Отчисления в ПФР, страховая и накопительная части (22%) – 15 307,56 руб.;

Отчисление в ФСС по травматизму (0,2%) – 139,16 руб.;

Отчисления в ФСС по временной нетрудоспособности и в связи с материнством (2,9%) – 2 017,82 руб.;

Отчисления в ФФОМС (5,1%) – 3 548,57 руб.;

Фонд оплаты труда и налоговых выплат за период создания программного продукта (44 дня) составляет:

Таблица 7

|  |  |  |
| --- | --- | --- |
| № п/п | Экономический показатель | Сумма, руб. |
| 1 | Оплаты труда | 69 579,84 |
| 2 | Отчисления в различные фонды | 21 013,11 |
| 3 | ИТОГО: | 90 592,95 |

Расчет амортизационных отчислений

Амортизационная премия по приобретенной компьютерной технике рассчитывается в сумме 31 709,30 руб. (317 093,00 х 10%). Первоначальная стоимость компьютера, уменьшенная на сумму амортизационной премии, с которой в дальнейшем будет производиться расчет ежемесячной амортизации 285 383,70 руб. (317 093,00 – 31 709,30).

Расчет ежемесячной амортизации:

4% от 317 093,00 руб. составляет – 11 415,35 руб.

Соответственно, в целях налогообложения, и исчисления амортизации в период создания программного продукта за 44 рабочих дня составит 25 113,77 руб.

Расчет использования программного обеспечения осуществляется аналогично амортизации компьютерной техники:

Расчет единовременной премии 8 489,97 (84 899,74 х 10%)

Первоначальная стоимость программного обеспечения, уменьшенная на сумму амортизационной премии, с которой в дальнейшем будет производиться расчет ежемесячной амортизации 76 409,77 (84 899,74 – 8 489,97).

Расчет ежемесячной амортизации программного обеспечения:

4 % от 84 899,74 руб. - 3 056,40 руб.

Соответственно, в целях налогообложения, и исчисления амортизации в период создания программного продукта за 44 рабочих дня составит 6 724,06 руб.

Таблица 8 ‑ Структура себестоимости

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование | Издержки,  руб. | Структура себестоимости, % |
| 1 | Материальные затраты | 4 738,10 | 3,58 |
| 2 | Фонд оплаты труда | 69 579,84 | 52,60 |
| 3 | Налоговые отчисления | 21 013,11 | 15,89 |
| 4 | Амортизационные отчисления | 31 837,83 | 24,07 |
| 5 | Прочие затраты | 5 100,00 | 3,86 |
|  | ИТОГО: | 132 268,88 | 100 |

Sn - Сумма по статьям издержек.

Sn = 4 738,10 + 69 579,84 + 21 013,11 + 31 837,83 + 5 100,00 =

132 268,88 руб.

Цена программного продукта для покупателя составляет сумму себестоимости создания программного продукта, добавочной стоимости (20 %) и налога на добавочную стоимость (18%). Цена программного продукта для покупателя составляет:

132 268,88 + 26 453,78 + 4 761,68 = 163 484,34 руб.

В цену программного продукта можно включить рецензирование, регистрацию авторских прав и прочее.

Экономическое обоснование разработки программных продуктов позволяет оптимизировать затраты производства и эффективно аргументировать свою работу.

# ЗАКЛЮЧЕНИЕ

Если необходимо донести информацию максимально быстро, доступно для большого количества людей, то лучше, с использованием Web технологий, которые стали доступны в любых цифровых устройствах, никак не выйдет.

Веб ресурс позволяет поделиться с миром своей уникальной историей путешествия, рассказать о своем средстве передвижения, а также создает возможность обсуждения прочитанного материала.

Разработка сайта обусловлена следующими факторами:

* Быстрота подачи информации широкому кругу лиц;
* Возможность организовать обратную связь с пользователями;

При разработке Web-сайта были решены следующие задачи:

* изучены современные технологии разработки Web-сайта;
* разработана структура сайта;
* разработан листинг программы;
* осуществлена отладка программного продукта;
* сайт размещен на доверенный и доступный хостинг;
* проведено тестирование программы;
* ведена в эксплуатацию программа;
* осуществлен расчет технико-экономических показателей разработки программного продукта.

## Эффективность создания и внедрения сайта очевидна: представлен интерактивный инструмент, позволяющий сообщить пользователям путешествии по России на собственном транспорте, привлечь единомышленников. Расходы на содержание сайта незначительны и сводятся лишь к платежам за раскрутку и поддержание сайта в актуальном виде.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем. Учебное пособие. – СПб: Университет ИТМО, 2018.
2. Стасышин В.М. Проектирование информационных систем и баз данных: учебное пособие. – Новосибирск: НГТУ, 2019.
3. Горлушкина Н.Н. Системный анализ и моделирование информационных процессов и систем. Учебное пособие. – СПб: Университет ИТМО, 2020.
4. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. – Проектирование информационных систем: учебное пособие / 2-е изд., испр. – М.: Интернет-Университет информационных технологий (ИНТУИТ.РУ): БИНОМ. Лаборатория знаний, 2019.
5. Осетрова И.С. Администрирование MS SQL Server 2014. – СПб: Университет ИТМО, 2019.
6. Осетрова И.С. Разработка баз данных в MS SQL Server 2014. – СПб: Университет ИТМО, 2019.
7. Троелсен Э. Язык программирования C# 5.0 и платформа .NET 4.5. 6-е изд. – М.: ООО «И.Д. Вильямс», 2020.
8. Фрэйн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. ‑ СПб.:Питер, 2020.
9. Кузнецов С.А.Хостинг для вашего сайта., - СПб.: Gudzon- 2018.
10. ГаевскийА.Ю., Романовский В.А. Создание Web-страниц и Web-сайтов HTML, -М.: [Триумф](http://spisok-literaturi.ru/publisher/triumf.html?id=27526), 2018.
11. Поддержка Microsoft [Электронный ресурс]. - <https://docs.microsoft.com>
12. Вопросы и ответы программистов [Электронный ресурс]. - <https://ru.stackoverflow.com/>
13. Руководство по ASP [Электронный ресурс]. - <https://metanit.com/sharp/aspnet5/>
14. Конструктор макетов [Электронный ресурс] - <https://wilda.ru/>

## Паттерны для новичков [Электронный ресурс] - <https://habr.com/ru/post/215605/>

## ПРИЛОЖЕНИЕ А

## (обязательное)

public class RoleModel

{

public int Id { get; set; }

public string Name { get; set; }

public virtual List<UserModel> Users { get; set; } = new List<UserModel>();

## }

public class RegisterModel

{

[Required(ErrorMessage = "Не указан логин")]

public string Login { get; set; }

[Required(ErrorMessage = "Не указан пароль")]

[DataType(DataType.Password)]

public string Password { get; set; }

[DataType(DataType.Password)]

[Compare("Password", ErrorMessage = "Пароли не совпадают")]

public string ConfirmPassword { get; set; }

//-------------------------------------------------------------

[Required(ErrorMessage = "Не указана фамилия")]

public string Surname { get; set; }

[Required(ErrorMessage = "Не указано имя")]

public string Name { get; set; }

## }

public class UserModel

{

[Key, Column(Order = 1)]

public int Id { get; set; }

public string Name { get; set; }

public string Surname { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public int RoleId { get; set; }

public virtual RoleModel Role { get; set; }

## }

public class LoginModel

{

[Required(ErrorMessage = "Не указан логин")]

public string Login { get; set; }

[Required(ErrorMessage = "Не указан пароль")]

[DataType(DataType.Password)]

public string Password { get; set; }

## }

public class ApplicationContext : DbContext

{

public DbSet<UserModel> Users { get; set; }

public DbSet<RoleModel> Roles { get; set; }

public DbSet<Message> Messages { get; set; }

public ApplicationContext(DbContextOptions<ApplicationContext> options)

: base(options)

{

Database.EnsureCreated();

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

string adminRoleName = "admin";

string userRoleName = "user";

string adminLogin = "admin";

string adminPassword = "admin";

// создаем роли

RoleModel adminRole = new RoleModel { Id = 1, Name = adminRoleName };

RoleModel userRole = new RoleModel { Id = 2, Name = userRoleName };

// и юзера

UserModel adminUser = new UserModel { Id = 1, Login = adminLogin, Password = adminPassword, RoleId = adminRole.Id, Name = "Кирилл" };

modelBuilder.Entity<RoleModel>().HasData(new RoleModel[] { adminRole, userRole });

modelBuilder.Entity<UserModel>().HasData(new UserModel[] { adminUser });

base.OnModelCreating(modelBuilder);

}

## }

public class AccountController : Controller

{

private ApplicationContext \_context;

public AccountController(ApplicationContext context)

{

\_context = context;

}

[HttpGet]

public IActionResult Register()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Register(RegisterModel model)

{

if (ModelState.IsValid)

{

UserModel user = await \_context.Users.FirstOrDefaultAsync(u => u.Login == model.Login);

if (user == null)

{

// добавляем пользователя в бд

user = new UserModel

{

Login = model.Login,

Password = model.Password,

Name = model.Name,

Surname = model.Surname,

};

RoleModel userRole = await \_context.Roles.FirstOrDefaultAsync(r => r.Name == "user");

if (userRole != null)

{

user.Role = userRole;

}

\_context.Users.Add(user);

await \_context.SaveChangesAsync();

//-------------------------------

await Authenticate(user); // аутентификация

if (user.Role.Name == "admin") return RedirectToAction("AdminPanel", "Admin");

else if (user.Role.Name != "admin") return RedirectToAction("User", "User");

}

else

ModelState.AddModelError("", "Некорректные логин и(или) пароль");

}

return View(model);

}

[HttpGet]

public IActionResult Login()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Login(LoginModel model)

{

if (ModelState.IsValid)

{

var passHash = model.Password;

UserModel user = await \_context.Users.FirstOrDefaultAsync(u => u.Login == model.Login && u.Password == passHash);

if (user != null)

{

await Authenticate(user); // аутентификация

if(user.Role.Name=="admin") return RedirectToAction("AdminPanel", "Admin");

else if (user.Role.Name != "admin") return RedirectToAction("UserPanel", "User");

}

ModelState.AddModelError("", "Некорректные логин и(или) пароль");

}

return View(model);

}

[HttpGet]

public async Task<IActionResult> Logout()

{

await HttpContext.SignOutAsync();

return RedirectToAction("Index", "Home");

}

private async Task Authenticate(UserModel user)

{

// создаем один claim

var claims = new List<Claim>

{

new Claim(ClaimsIdentity.DefaultNameClaimType, user.Login),

new Claim(ClaimsIdentity.DefaultRoleClaimType, user.Role?.Name)

};

// создаем объект ClaimsIdentity

ClaimsIdentity id = new ClaimsIdentity(claims, "ApplicationCookie", ClaimsIdentity.DefaultNameClaimType,

ClaimsIdentity.DefaultRoleClaimType);

// установка аутентификационных куки

await HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new ClaimsPrincipal(id));

}

## }

[Authorize(Roles = "admin")]

public class AdminController : Controller

{

private UserModel \_User { get { return \_context.Users.FirstOrDefault(u => u.Login == User.Identity.Name); } }

private ApplicationContext \_context;

public AdminController(ApplicationContext context)

{

\_context = context;

}

[HttpGet]

public IActionResult AdminPanel()

{

return View(\_User);

}

## }

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

private UserModel \_User { get { return \_context.Users.FirstOrDefault(u => u.Login == User.Identity.Name); } }

private ApplicationContext \_context;

public HomeController(ILogger<HomeController> logger, ApplicationContext context)

{

\_logger = logger;

\_context = context;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

public IActionResult About\_Motorcycle()

{

return View();

}

public IActionResult About\_Tour()

{

return View();

}

public IActionResult My\_Experience()

{

return View();

}

[HttpGet]

public IActionResult Forum()

{

return View(\_context.Messages.ToArray());

}

[HttpPost]

public IActionResult Forum(string InputedMsg)

{

\_context.Messages.Add(new Message { messageText = InputedMsg, Departure = DateTime.Now, Sender = \_User });

\_context.SaveChanges();

return View(\_context.Messages.ToArray());

}

## }

<div class="motorcycle" style="display: flex;">

@\* TEXT \*@

<div class="leftcol">

<div id="1" name="enableble" class="text-left">

<p class="tour\_\_content-text">

В 1998 году Honda выводит на внутренний рынок достаточно уникальный байк CL400.

Эта модель продолжает оригинальные традиции мотоциклов 70-х годов.

В основе Honda CL400 лежит классическая конструкция.

Мотоцикл получил стильный ретро-дизайн, при этом является непростым "дорожником".

Увеличенный клиренс, высокопрофильные шины с глубоким рисунком протектора ставят Honda CL400 в один ряд с представителями мотокасты эндуро.

</pclass="tour\_\_content-text">

</div>

<div id="2" hidden="hidden" name="enableble" class="text-left">

<p class="tour\_\_content-text">

1-цилиндровый двигатель Honda CL400 заводится кикстартером, как классические ретро-мотоциклы.

Воздушная система охлаждения справляется со своей задачей.

Предельно простая конструкция подвески с телескопической вилкой спереди и двумя стойками сзади не делает Honda CL400 гоночным мотоциклом и не способствует хорошей управляемости.

Однако большой ход амортизаторов обуславливает невероятно легкое поведение байка на бездорожье, а легкий вес и специальные шины помогают раскрыть весь потенциал.

</p>

</div>

<div id="3" hidden="hidden" name="enableble" class="text-left">

<p class="tour\_\_content-text">

Конструкция тормозной системы весьма незатейливая: спереди дисковые тормоза, сзади - устаревшие барабанные.

В России мотоциклы Honda CL400 не очень распространены.

Модель успешно продавалась только для японских потребителей.

Многие любят сравнивать CL400 с отечественным ИЖ Планета.

Эти мотоциклы роднит простая конструкция, высокая надежность и классический ретро-дизайн.

Модель ретро-классического мотоцикла Honda CL 400 была представлена в 1997 году на Токийском автосалоне,

а в серию пошла с 23 сентября 1998 года.

Концепция модели была достаточно простой - представить рынку мотоцикл с классическими формами,

отлично подходящий как для города, так и для загородных поездок.

За основу модели был взят 1-цилиндровый 4-тактный двигатель воздушного охлаждения,

объемом 397 куб. см., выдающий 29 л.с. мощности и 34 Нм крутящего момента.

</p>

</div>

<div id="4" hidden="hidden" name="enableble" class="text-left">

<p class="tour\_\_content-text">

Данный мотор был позаимствован от эндуро Honda XR 400 и немного перенастроен для повышения крутящего момента на низах.

Из особенностей Honda CL 400 следует отметить стальную дуплексную раму, кик-стартер, систему смазки - сухой картер,

классические подвески в виде обычной телескопической вилки и двойного амортизатора, задние барабанные тормоза,

5-ступенчатую КПП и большие спицованные колеса на 19' и 18'. Модель Honda CL 400 предназначалась, в первую очередь,

для японского рынка, поэтому этот мотоцикл широко представлен на японских аукционах.

Модель выпускалась вплоть до 2001 года, после чего была снята с производства и заменена.

</p>

</div>

</div>

<div class="rightcol">

<div class="slider slider-pi">

<input type="radio" name="slide\_switch" id="id1" checked="checked" />

<label for="id1">

<img src="~/Images/о мотике/1.jpeg" width="85" />

</label>

<img src="~/Images/о мотике/1.jpeg" width="571" height="427" />

<input type="radio" name="slide\_switch" id="id2" />

<label for="id2">

<img src="~/Images/доп/WhatsApp Image 2021-05-24 at 23.58.47.jpeg" width="85" height="64" />

</label>

<img src="~/Images/доп/WhatsApp Image 2021-05-24 at 23.58.47.jpeg" width="571" height="427" />

<input type="radio" name="slide\_switch" id="id3" />

<label for="id3">

<img src="~/Images/о мотике/3.jpeg" width="85" />

</label>

<img src="~/Images/о мотике/3.jpeg" width="571" height="427" />

<input type="radio" name="slide\_switch" id="id4" />

<label for="id4">

<img src="~/Images/о мотике/4.jpeg" width="85" />

</label>

<img src="~/Images/о мотике/4.jpeg" width="571" height="427" />

<input type="radio" name="slide\_switch" id="id5" />

<label for="id5">

<img src="~/Images/доп/WhatsApp Image 2021-05-16 at 15.48.33 (1).jpeg" width="85"/>

</label>

<img src="~/Images/доп/WhatsApp Image 2021-05-16 at 15.48.33 (1).jpeg" width="571" height="427" />

</div>

</div>

</div>

<script>

let el = document.getElementsByName("enableble");

function enable(id) {

for (let i = 0; i < el.length; i++) {

el[i].hidden = true;

}

document.getElementById(id).hidden = false;

}

document.getElementById("id1").onclick = function () { enable(1) };

document.getElementById("id2").onclick = function () { enable(2) };

document.getElementById("id3").onclick = function () { enable(3) };

document.getElementById("id4").onclick = function () { enable(4) };

## </script>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Motorcycle</title>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<link rel="stylesheet" href="~/css/site.css" />

<link rel="stylesheet" href="~/css/ImageSlider.css" />

</head>

<body>

<div class="bg">

<img src="~/Images/bg60temniy.jpg" alt="Alternate Text" />

</div>

<header class="custom-header">

<nav class="custom-header navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container">

@\*<a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Motorcycle</a>\*@

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="About\_Tour">О туре</a>

<a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="My\_Experience">Мой опыт</a>

<a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="About\_Motorcycle">Галерея моего мотоцикла</a>

@if (User.Identity.IsAuthenticated)

{

<a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="Forum">Форум</a>

if (User.IsInRole("user"))

{<a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="Index">Личный кабинет</a>}

else if (User.IsInRole("admin"))

{ <a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="Index">Личный кабинет</a>}

<a class="navbar-brand" **asp-area**="" **asp-controller**="Account" **asp-action**="Logout">Выйти</a>

}

else

{

<a class="navbar-brand" **asp-area**="" **asp-controller**="Account" **asp-action**="Register">Зарегистрироваться</a>

<a class="navbar-brand" **asp-area**="" **asp-controller**="Account" **asp-action**="Login">Войти</a>

}

</div>

</nav>

<div class="ErrorsMessage" **asp-validation-summary**="All"></div>

</header>

<div class="container custom-container">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer">

<div class="container">

&copy; 2021 - Motorcycle - Путешествиия это круто!

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

<script **src**="~/js/site.js" **asp-append-version**="true"></script>

@await RenderSectionAsync("Scripts", required: false)

<script>

const img = document.querySelector('.bg');

document.addEventListener('scroll', () => {

const scroll = window.pageYOffset;

console.log(img)

img.style.transform = `translateY(${scroll / 5}px)`

})

</script>

</body>

## </html>

@model IEnumerable<Message>

<form **asp-action**="Forum" **asp-controller**="Home" method="post">

<div style=" padding-top: 50px; ">

<div id="Chat" class="Chat">

@foreach (var msg in Model)

{

<div class="ChatItem" style="min-height:100px">

<p> @(msg.Sender.Name+ " "+ msg.Sender.Surname + " - " + msg.Departure.ToShortDateString())</p>

<p>@msg.messageText</p>

</div>

}

<div class="MsgInput">

<input style="width:100%" type="text" name="InputedMsg" value="" placeholder="Введите сообщение..." />

<button type="submit">Отправить</button>

</div>

</div>

</div>

</form>

<script>

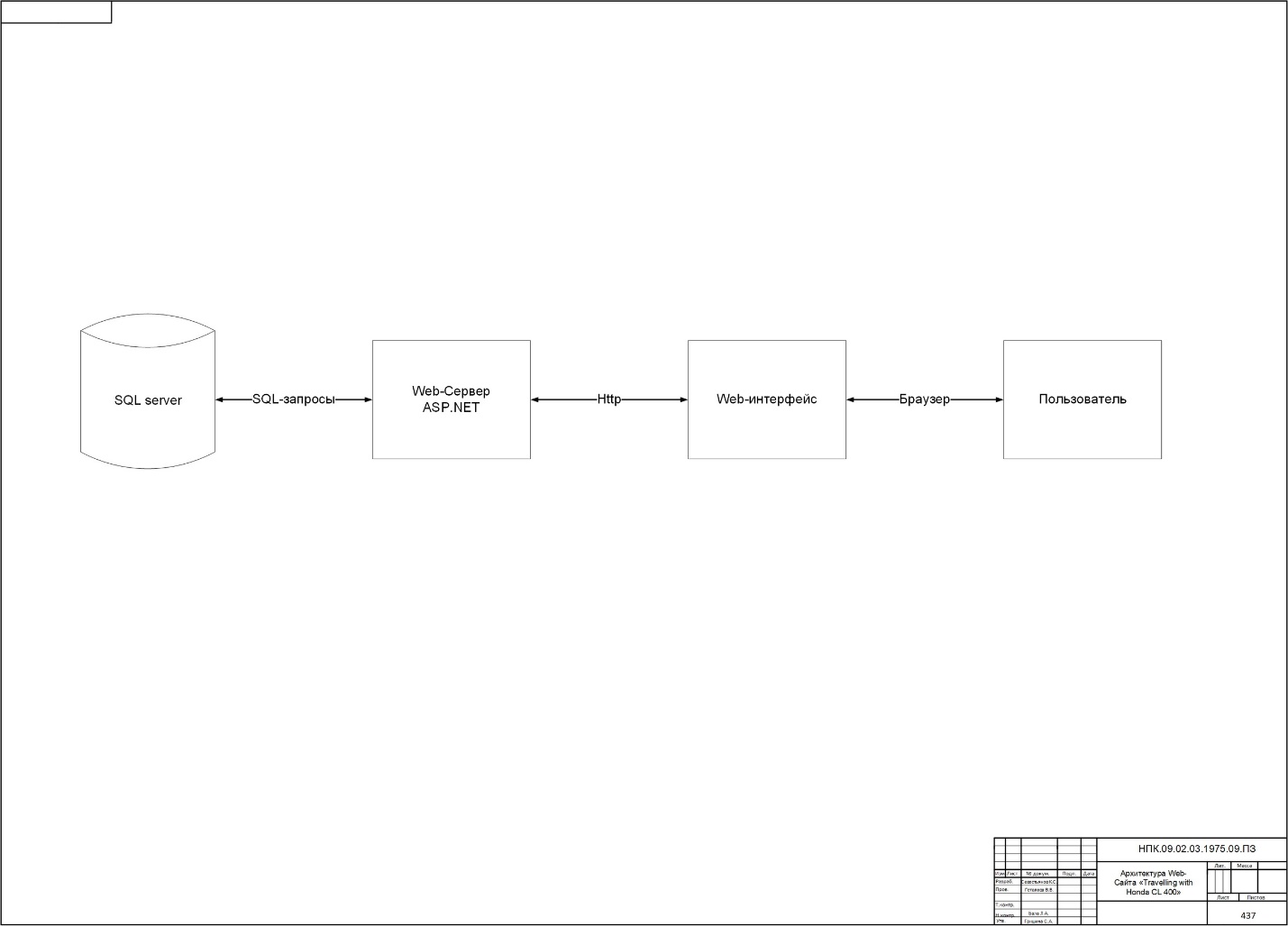
let chat = document.getElementById("Chat");

chat.scrollTop = chat.scrollHeight;

## </script>

## ПРИЛОЖЕНИЕ Б

Архитектура приложения



Структура сайта

