**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ ПМР**

**ГОУ СПО «ТИРАСПОЛЬСКИЙ ТЕХНИКУМ ИНФОРМАТИКИ И ПРАВА»**

**ДНЕВНИК**

**ПРОХОЖДЕНИЯ практики**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**УП.01.01\_Учебная практика**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

вид практики

Профессиональный модуль \_\_\_ПМ.01 Разработка модулей программного\_\_\_ обеспечения для компьютерных систем \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

индекс, наименование профессионального модуля

Междисциплинарный курс \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_МДК 01.01. Разработка программных модулей\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_МДК 01.02. Поддержка и тестирование программных модулей\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

индекс, наименование междисциплинарного курса

Ф.И.О. студента(ки)\_\_\_\_Сиглов Артем Витальевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность \_\_2.09.02.07 Информационные системы и программирование\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

код, наименование специальности

Курс \_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ группа \_\_\_315\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Время прохождения практики: с «\_1\_» \_\_11\_\_ 20\_24\_ г. по «\_12\_» \_\_12\_\_ 20\_24\_ г.

База практики \_\_ГОУ СПО «Тираспольский техникум информатики и права»\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **п/п** | **Дата** | **Описание выполненной работы** | **Количество часов** | **Оценка и подпись руководителя практики** |
|  | 01.11.2024 | Роль практического обучения при формировании компетенций обучающихся.  Инструктаж по технике безопасности и правилам поведения на учебной практике. | 2 |  |
|  | 01.11.2024 | Создание проекта на основе шаблона MVC. | 2 |  |
|  | 01.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 02.11.2024 | Использование Git для контроля версий. | 2 |  |
|  | 02.11.2024 | Использование Git для контроля версий. | 2 |  |
|  | 02.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 04.11.2024 | Определение структуры MVC-приложения, взаимодействие моделей, контроллеров и представлений. | 2 |  |
|  | 04.11.2024 | Определение структуры MVC-приложения, взаимодействие моделей, контроллеров и представлений. | 2 |  |
|  | 04.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 05.11.2024 | Frontend разработка основная страница: написание шапки сайта, секции «О нас». | 2 |  |
|  | 05.11.2024 | Frontend разработка основная страница: написание шапки сайта, секции «О нас». | 2 |  |
|  | 05.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 06.11.2024 | Frontend разработка основная страница: написание секций «Услуги», «Контакты». | 2 |  |
|  | 06.11.2024 | Frontend разработка основная страница: написание секций «Услуги», «Контакты». | 2 |  |
|  | 06.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 07.11.2024 | Frontend разработка основная страница: написание форм входа/регистрации. | 2 |  |
|  | 07.11.2024 | Frontend разработка основная страница: написание форм входа/регистрации. | 2 |  |
|  | 07.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 08.11.2024 | Frontend разработка основная страница: написание секции «Написать сообщение» и подвала сайта. | 2 |  |
|  | 08.11.2024 | Frontend разработка основная страница: написание секции «Написать сообщение» и подвала сайта. | 2 |  |
|  | 08.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 09.11.2024 | Использование fetch-запросов для обновления данных без перезагрузки страницы. | 2 |  |
|  | 09.11.2024 | Использование fetch-запросов для обновления данных без перезагрузки страницы. | 2 |  |
|  | 09.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 11.11.2024 | Создание адаптивного интерфейса с использованием медиазапросов CSS. | 2 |  |
|  | 11.11.2024 | Создание адаптивного интерфейса с использованием медиазапросов CSS. | 2 |  |
|  | 11.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 12.11.2024 | Знакомство с PostgreSQL. | 2 |  |
|  | 12.11.2024 | Знакомство с PostgreSQL. | 2 |  |
|  | 12.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 13.11.2024 | Создание базы данных в СУБД PostgreSQL. | 2 |  |
|  | 13.11.2024 | Создание базы данных в СУБД PostgreSQL. | 2 |  |
|  | 13.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 14.11.2024 | Подключение PostgreSQL через appsettings.json. Определение сущностей и их атрибутов в C# для базы данных с использованием Entity Framework. | 2 |  |
|  | 14.11.2024 | Подключение PostgreSQL через appsettings.json. Определение сущностей и их атрибутов в C# для базы данных с использованием Entity Framework. | 2 |  |
|  | 14.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 15.11.2024 | Разработка базовых операций CRUD для всех сущностей. Добавление моделей. | 2 |  |
|  | 15.11.2024 | Разработка базовых операций CRUD для всех сущностей. Добавление моделей. | 2 |  |
|  | 15.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 16.11.2024 | Разработка сервисов для авторизации и аутентификации. | 2 |  |
|  | 16.11.2024 | Разработка сервисов для авторизации и аутентификации. | 2 |  |
|  | 16.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 18.11.2024 | Подключение ASP.NET Identity для управления доступом пользователей и роли. | 2 |  |
|  | 18.11.2024 | Подключение ASP.NET Identity для управления доступом пользователей и роли. | 2 |  |
|  | 18.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 19.11.2024 | Использование C#, FluentValidation и JavaScript для валидации вводимых данных при авторизации.  Оформление отчетной документации. | 2 |  |
|  | 19.11.2024 | Использование C#, FluentValidation и JavaScript для валидации вводимых данных при регистрации. | 2 |  |
|  | 19.11.2024 | Использование C#, FluentValidation и JavaScript для валидации вводимых данных при регистрации. | 2 |  |
|  | 20.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 20.11.2024 | Реализация аутентификации пользователя при регистрации. | 2 |  |
|  | 20.11.2024 | Реализация аутентификации пользователя при регистрации. | 2 |  |
|  | 21.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 21.11.2024 | Интеграция сторонних провайдеров, таких как Google, для регистрации и входа пользователей. | 2 |  |
|  | 21.11.2024 | Интеграция сторонних провайдеров, таких как Google, для регистрации и входа пользователей. | 2 |  |
|  | 22.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 22.11.2024 | Frontend разработка страницы с профилем пользователя: написание вкладки с запросами. | 2 |  |
|  | 22.11.2024 | Frontend разработка страницы с профилем пользователя: написание вкладки с запросами. | 2 |  |
|  | 23.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 23.11.2024 | Frontend разработка страницы с профилем пользователя: написание вкладки с корзиной. | 2 |  |
|  | 23.11.2024 | Frontend разработка страницы с профилем пользователя: написание вкладки с корзиной. | 2 |  |
|  | 25.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 25.11.2024 | Реализация функционала редактирования профиля и обновления данных пользователя. | 2 |  |
|  | 25.11.2024 | Реализация функционала редактирования профиля и обновления данных пользователя. | 2 |  |
|  | 26.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 26.11.2024 | Frontend разработка страницы со странами. | 2 |  |
|  | 26.11.2024 | Frontend разработка страницы со странами. | 2 |  |
|  | 27.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 27.11.2024 | Frontend разработка страницы с турами: реализация фильтров и сортировки данных на страницах с турами. | 2 |  |
|  | 27.11.2024 | Frontend разработка страницы с турами: реализация фильтров и сортировки данных на страницах с турами. | 2 |  |
|  | 28.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 28.11.2024 | Разработка сервисов для реализации фильтров и сортировки данных на страницах с турами. | 2 |  |
|  | 28.11.2024 | Разработка сервисов для реализации фильтров и сортировки данных на страницах с турами. | 2 |  |
|  | 29.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 29.11.2024 | Frontend разработка страницы с турами: секция отображения туров. | 2 |  |
|  | 29.11.2024 | Frontend разработка страницы с турами: секция отображения туров. | 2 |  |
|  | 30.11.2024 | Оформление отчетной документации. | 2 |  |
|  | 30.11.2024 | Frontend разработка конкретная страница с туром. | 2 |  |
|  | 30.11.2024 | Frontend разработка конкретная страница с туром. | 2 |  |
|  | 02.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 02.12.2024 | Реализация поискового механизма с использованием JS. | 2 |  |
|  | 02.12.2024 | Реализация поискового механизма с использованием JS. | 2 |  |
|  | 03.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 03.12.2024 | Разделение прав доступа для администраторов, обычных пользователей. | 2 |  |
|  | 03.12.2024 | Разделение прав доступа для администраторов, обычных пользователей. | 2 |  |
|  | 04.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 04.12.2024 | Разработка интерфейса для администрирования пользователей и управление запросами. | 2 |  |
|  | 04.12.2024 | Разработка интерфейса для администрирования пользователей и управление запросами. | 2 |  |
|  | 05.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 05.12.2024 | Разработка интерфейса для администрирования странами и турами. | 2 |  |
|  | 05.12.2024 | Разработка интерфейса для администрирования странами и турами. | 2 |  |
|  | 06.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 06.12.2024 | Разработка функций для управления контентом сайта, таких как создание новых туров. | 2 |  |
|  | 06.12.2024 | Разработка функций для управления контентом сайта, таких как создание новых туров. | 2 |  |
|  | 07.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 07.12.2024 | Разработка модульных тестов на xUnit для проверки корректности работы логики приложения при аутентификации. | 2 |  |
|  | 07.12.2024 | Разработка модульных тестов на xUnit для проверки корректности работы логики приложения при аутентификации. | 2 |  |
|  | 09.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 09.12.2024 | Разработка модульных тестов на xUnit для проверки корректности работы логики приложения для сервисов, связанных с профилем. | 2 |  |
|  | 09.12.2024 | Разработка модульных тестов на xUnit для проверки корректности работы логики приложения для сервисов, связанных с профилем. | 2 |  |
|  | 10.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 10.12.2024 | Разработка модульных тестов на xUnit для проверки корректности работы логики приложения для сервисов, связанных с турами. | 2 |  |
|  | 10.12.2024 | Разработка модульных тестов на xUnit для проверки корректности работы логики приложения для сервисов, связанных с турами. | 2 |  |
|  | 11.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 11.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 11.12.2024 | Оформление отчетной документации. | 2 |  |
|  | 12.12.2024 | Оформление индивидуального проекта и презентации к защите работы. | 2 |  |
|  | 12.12.2024 | Оформление индивидуального проекта и презентации к защите работы. | 2 |  |
|  | 12.12.2024 | Дифференцированный зачет. | 2 |  |

Содержание объемов выполненных работ подтверждаю

Руководитель практики

от техникума (предприятия) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_С.Г. Патеров\_\_/

подпись Ф.И.О.

**01.11.2024**

**Тема:** Роль практического обучения при формировании компетенций обучающихся. Инструктаж по технике безопасности и правилам поведения на учебной практике. Создание проекта на основе шаблона MVC. Оформление отчетной документации.

**Ход Работы:**

**Шаг 1.** Для начала необходимо зайти в *Visual studio* и выбрать пункт «Создание проекта» как показано на рисунке 1.

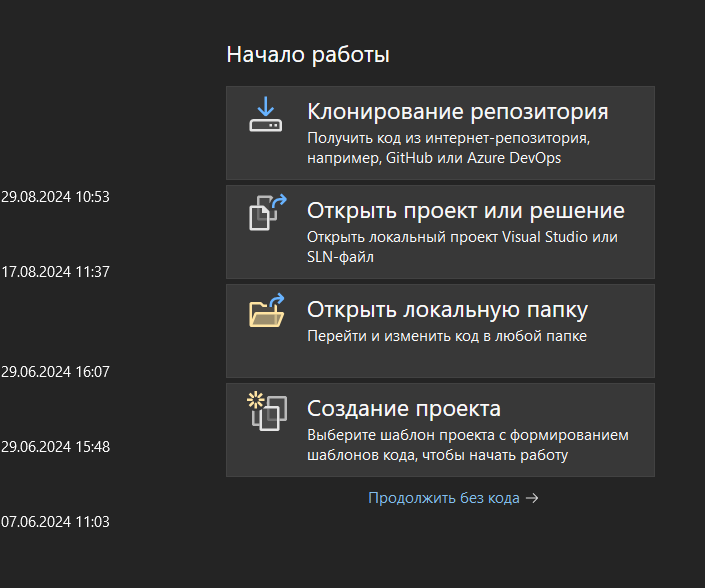


Рисунок 1 – Создание проекта

**Шаг 2.** После выбираем необходимые параметры для поиска нужного шаблона либо вводим название «Веб-приложение *ASP.NET Core* (модель-представление-контроллер» в строку поиска как можно увидеть на рисунке 2. Данный проект будет написан с использованием архитектуры *MVC*.

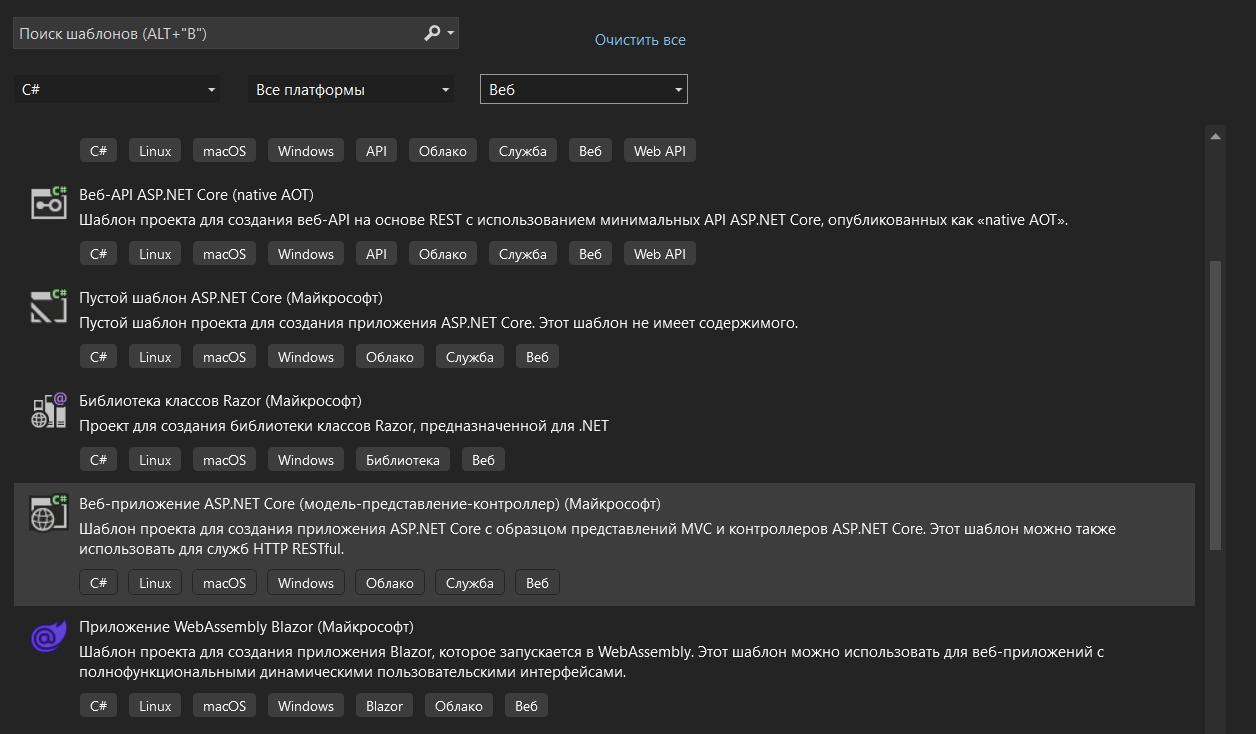


Рисунок 2 – Шаблоны

**Шаг 3.**  *MVC* расшифровывается как «модель-представление-контроллер» (от англ. *model-view-controller*). Это способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач. Один блок отвечает за данные приложения, другой отвечает за внешний вид, а третий контролирует работу приложения.

Когда нужный шаблон был выбран указываем имя проекта «*TravelAgency*» и путь к проекту. После нажимаем кнопку «Далее» в появившемся окне необходимо выбрать платформу .*NET* для нашего проекта (.Net 6 или выше).

В обозревателе решений на рисунке 3 показана структура данного шаблона на начальном этапе разработки. Если обозреватель решений не закреплен на панели его можно найти в Вид/Обозреватель решений.

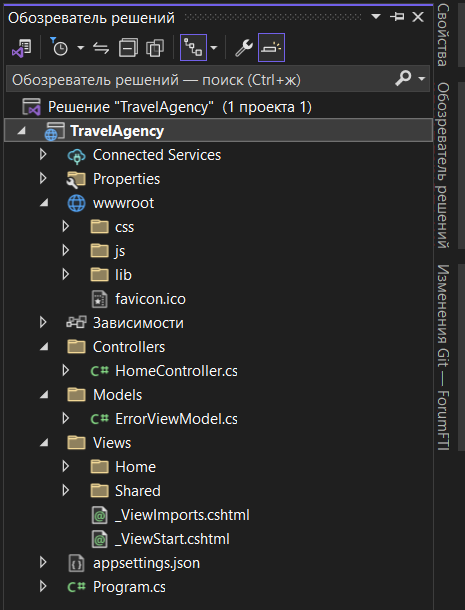


Рисунок 3 – Обозреватель решений

**Шаг 4.** Здесь есть несколько основных папок с которыми мы будем работать.

Папка *wwwroot* содержит в себе папку для стилей проекта, папку, содержащую скрипты для динамического взаимодействия с проектом, и папку с дополнительными файлами которые могут упростить работу если знать, как их использовать. В данном проекте мы будем использовать чистый JavaScript.

Папка *Controllers* – обрабатывает входящие запросы. Во фреймворке это может заключаться в определении конкретных *URL*, на которые попадает пользователь при переходе по ссылке или при нажатии кнопки.

Папка *Models* – отвечает за данные, которые хранятся и обрабатываются на сервере.

Папка *Views* – это *HTML*-шаблон, который возвращает сервер после обработки запроса.

В обозревателе решений выбираем *Views/Shered/\_Layout.cshtml*. Файлы *layout* или мастер-страницы позволяют определить единый шаблон для страниц *Razor* и применяются для создания единообразного, унифицированного вида приложения. Для определения интерфейса мастер-страницы также применяют код *Razor* и *html*, как обычные страницы *razor*, но при этом упрощают создание приложения. Например, можно определить на мастер-странице общие для всех остальных страниц *Razor* меню и другие элементы, а также подключить общие стили и скрипты. В итоге нам не придется на каждой отдельной странице прописывать путь к файлам стилей, а потом при необходимости его изменять.

Тег <*head*> показанный на рисунке 4 предназначен для хранения служебной информации о странице. Он располагается первым в теге <*html*>, сразу перед <*body*>. Внутри <*head*> обычно содержится заголовок, ключевые слова, описание страницы и другие служебные данные.

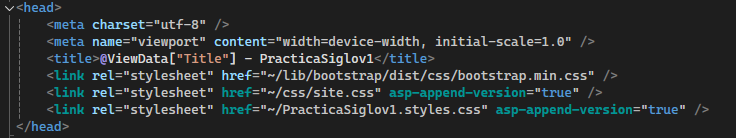


Рисунок 4 – тег <head>

<*header*> – это семантичный элемент, который нужен для отделения вводного содержимого или навигации от основного содержимого целой страницы или секционного блока. Например, «шапка» с навигацией для всего сайта или имя автора и дата публикации в превью статьи или новости.

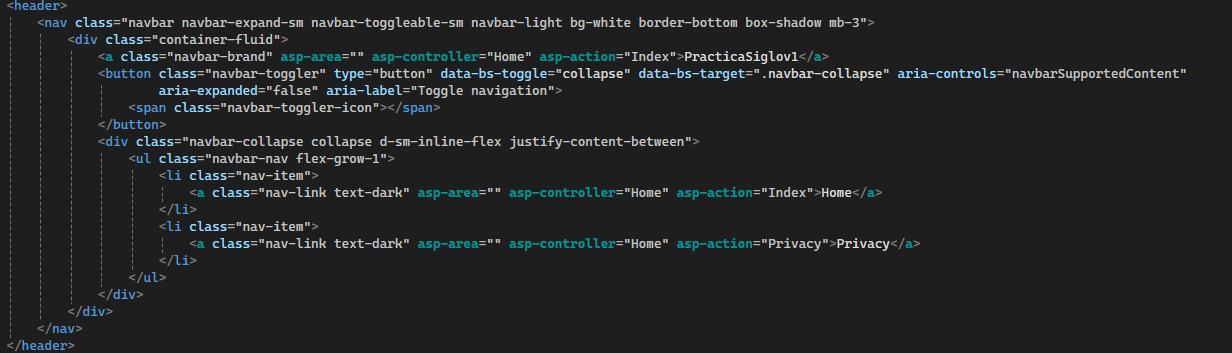


Рисунок 5 – тег <header>

Отличительной особенностью файлов *layout* является использование метода @*RenderBody()*. Этот метод будет вставлять содержимое страниц *Razor*, которые используют данную мастер-страницу. В итоге мы сможем легко установить для всех страниц единообразный стиль оформления.

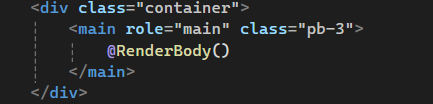


Рисунок 6 – функция *@RenderBody*()

Tег <*footer*> служит для создания футера сайта или же подвала сайта. Это нижняя часть сайта, в которой может находиться какая-нибудь контактная информация, нижнее меню, ссылки или же что-то в этом духе.

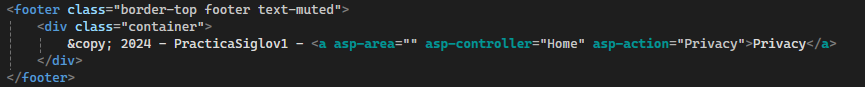


Рисунок 7 – тег <*footer*>

Скрипты чаще всего подключают из внешних файлов с расширением .*js*. Для этого используют тег <*script*> с атрибутом *src*, в котором указывается путь к файлу. Внешние скрипты лучше подключать перед закрывающим тегом </*body*>.

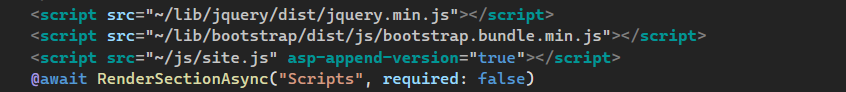


Рисунок 8 – подключение скриптов

**Шаг 5.** Рассмотрим еще два файла находящихся в папке *Views/Shered/* *\_ViewImports.cshtml* и *\_ViewStart.cshtml.* В первом файле находятся всего две строки кода показанные на рисунке 9. Эти немногочисленные строки предназначены для глобального подключения пространств имен и дополнительных возможностей, которые применяются ко всем представлениям в вашем проекте *ASP.NET Core MVC*.

https://lh7-rt.googleusercontent.com/docsz/AD_4nXeFuPF3Qr9MzC3AiEd8WMh79bEOYUYoJ2riWK0I9t5x_8ME8FjFHVA8NONU5XDaUyapZgsYgbn6U7jiqY9U0wD-Ebevd8kuLkeEpviT5tKkxXPLt-MNy4x8SvzrEhKYn-P_y5k3FZrlWl25ZkgQrKMVUviP0NZBVs5tSPW80nigxsk?key=dVzRd2Dt4PuAbWPPkpYmh6OO

Рисунок 9 – содержимое файла *\_ViewImports*

В файле *\_ViewStart.cshtml* так же всего пару строк, как и показано на рисунке 10. Эта строка в *Razor*-представлении *ASP.NET Core* используется для указания файла макета (*layout*), который будет применен к текущему представлению.

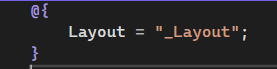


Рисунок 10 – содержимое файла *\_ViewStart.cshtml*

**Шаг 6.** Рассмотрим еще одну папку в разделе представлений – папка *Home.* В этой папке хранятся представления, которые связаны с действиями *(Action Methods)* контроллера *HomeController.* Этот контроллер обычно отвечает за основные страницы веб-приложения, такие как главная страница, страница контактов и т.д. Так же по умолчанию при запуске проекта вызывается именно этот контролер и действие *Index* это можно изменить в файле *Program.cs* в строках показанных на рисунке 11.

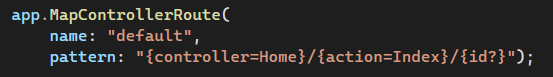


Рисунок 11 – фрагмент кода из файла *Program.cs*

В результате создания проекта при запуске проекта открывается страница браузера показанная на рисунке 12

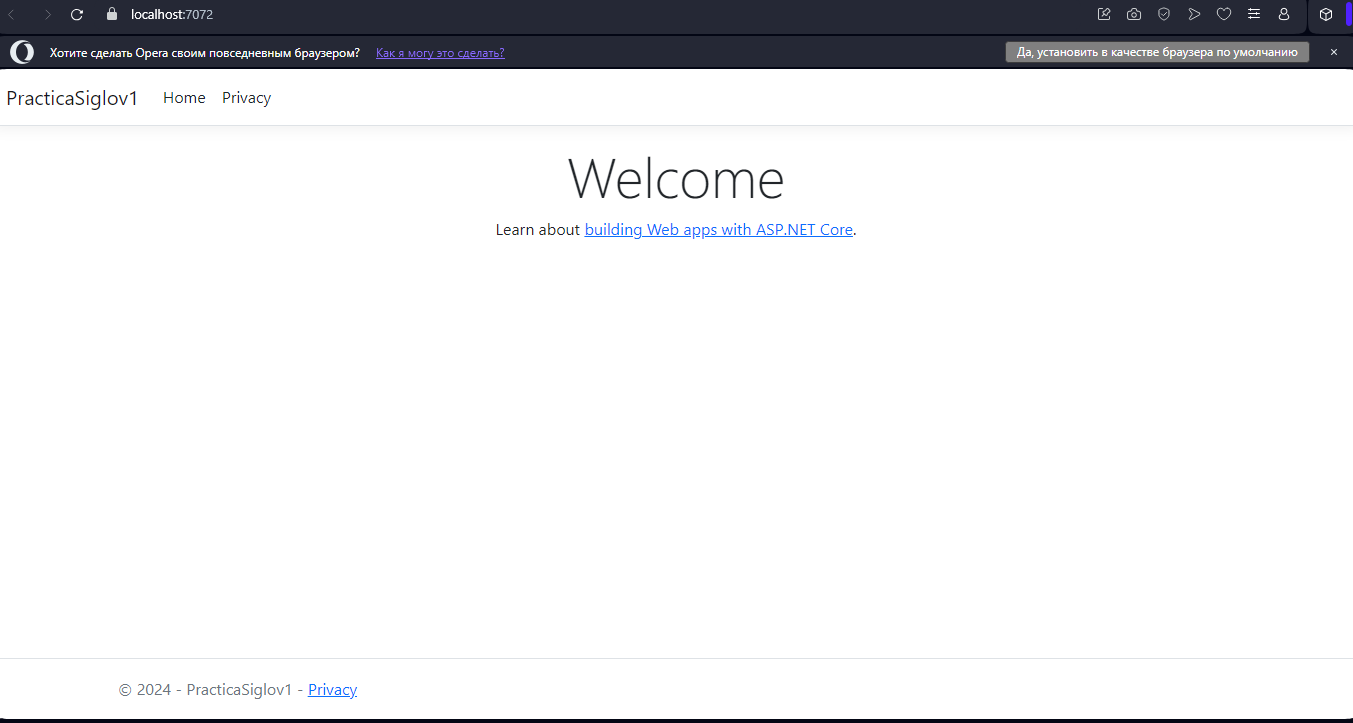


Рисунок 12 – результат работы кода при создании

**Вывод:** Создал проект на основе MVC ознакомился с архитектурой проекта.

**02.11.2024**

**Тема:** Использование Git для контроля версий. Оформление отчетной документации.

**Ход работы:**

**Шаг 1.** Во время регистрации вам будет предложено проверить адрес электронной почты. Без проверенного адреса электронной почты вы не сможете выполнить некоторые основные задачи *GitHub*, например, создание репозитория. Для начала создадим репозиторий для этого в правом верхнем углу открываем меню пользователя и выбираем пункт *«Your repositories».*

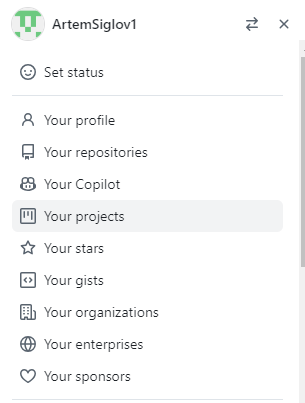


Рисунок 12 – ваш профиль

**Шаг 2.** На странице с репозиториями нажимаем на кнопку *«New».* Откроется страница с созданием репозитория. Задаем название репозитория *TravelAgency* как показано на рисунке 13. Остальные поля и пункты можно оставить в исходном положении.

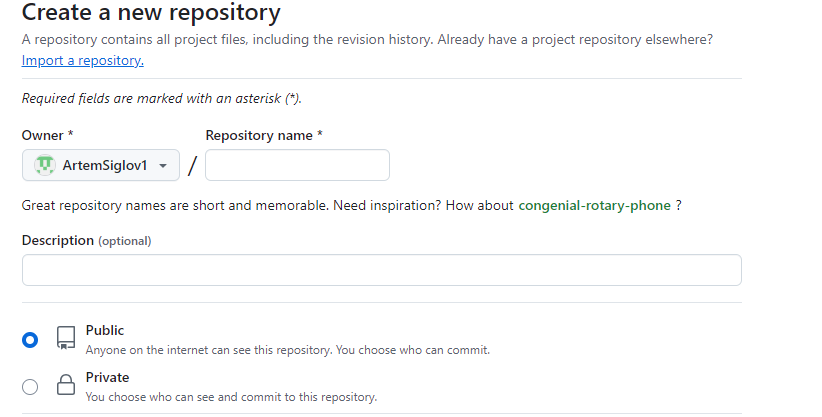
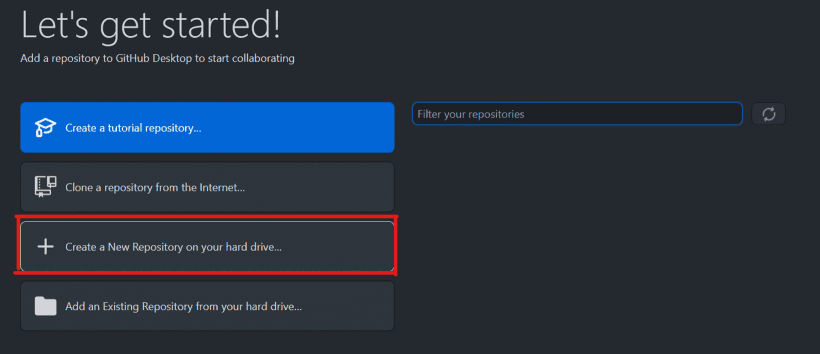


Рисунок 13 – создание удаленного репозитория

**Шаг 3.** После создания, удаленного репозитория необходимо создать локальный репозиторий. Для того что бы разобраться как работать с *GitHub Desktop* можно перейти на сайт [*https://htmlacademy.ru/blog/git/github-desktop*](https://htmlacademy.ru/blog/git/github-desktop)либо просмотреть информацию ниже, она взята с этого сайта.

После первого входа в *GitHub Desktop* вас попросят ввести ваши [логин и пароль от *GitHub.com*](https://htmlacademy.ru/blog/git/github-account). После этого у вас появится доступ ко всем репозиториям, сохранённым в профиле.

Создание репозитория: если вы никогда не пользовались *GitHub*-ом, нужно будет создать репозиторий для работы над проектом. На главном экране *GitHub Desktop* выберите пункт *«Create a New Repository on your hard drive».*

 Рисунок 14 – начало работы

**Шаг 4.** Нужно будет ввести название репозитория, его описание и выбрать папку на компьютере, куда будут сохраняться файлы.

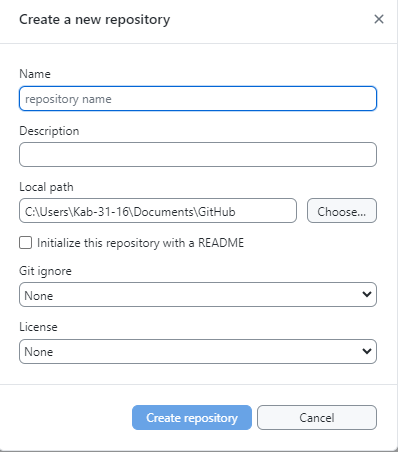


Рисунок 15 – создание локального репозитория

**Шаг 5.** После этого нажимаем на *Create repository*, ждём несколько секунд и готово – на компьютере появилась папка, которой можно пользоваться для разработки вашего проекта.

Клонирование репозитория: если у вас уже какой-нибудь репозиторий на Гитхабе, его можно клонировать. Клонировать – это скачать все файлы к себе на компьютер, чтобы можно было их изменять и потом загружать обратно.

Выбираем *Add -> Clone Repository...*

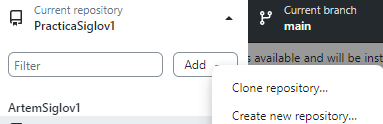


Рисунок 16 – клонирование репозитория

**Шаг 6.** В открывшемся окне выбираем один из имеющихся репозиториев. В данном случае он называется *zaverstai,* но у вас может быть любой другой.

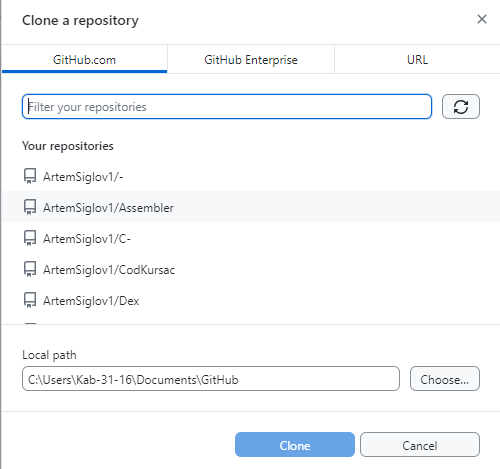


Рисунок 17 – выбор репозитория

После этого файлы репозитория начнут скачиваться – если их много, то это займет некоторое время.

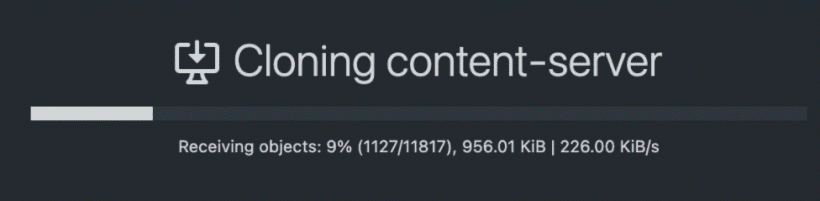


Рисунок 18 – скачивание файлов

**Шаг 7.** Работа с репозиторием. Меняем файлы и сохраняем обратно: вне зависимости от того, создали вы репозиторий или клонировали его, так выглядит *GitHub Desktop* с открытым репозиторием, в котором мы пока ничего не меняли.

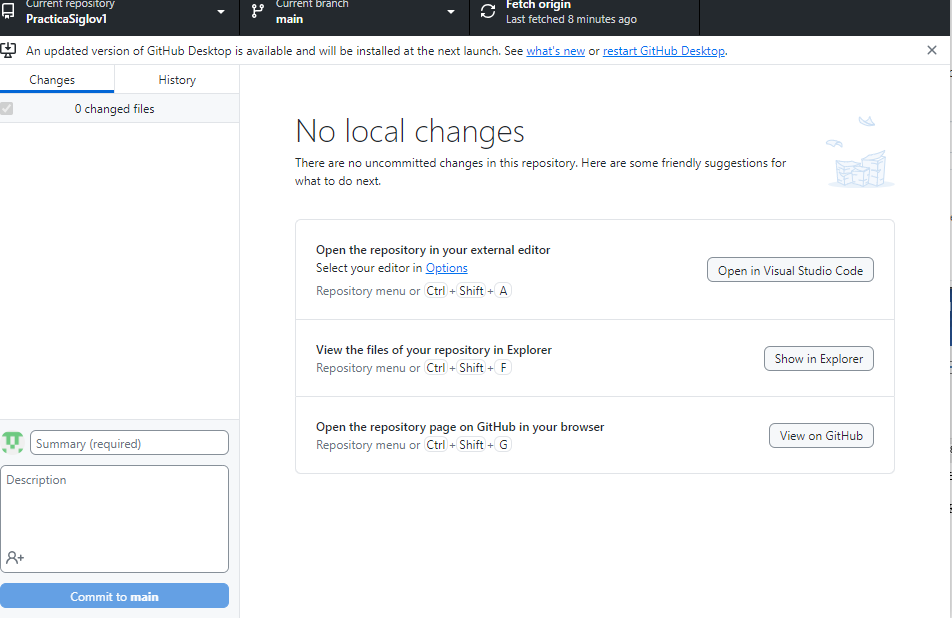


Рисунок 19 – вид приложения с открыты репозиторием

*Слева* – *поле для измененных файлов, справа* – *служебная информация. Слева снизу* – *поле для коммитов.*

**Шаг 8.** Давайте добавим какой-нибудь файл. Например, я добавил в локальный репозиторий (скопировал в папку) файл index.html. Вы можете добавить текстовый файл.

Сразу после добавления или изменения файла в окне GitHub Desktop будет видно, что изменилось – если мы добавили целый новый файл, то все строчки будут с плюсиками и зелёные. Это значит, что они были добавлены в файл и GitHub Desktop раньше их никогда не видел.

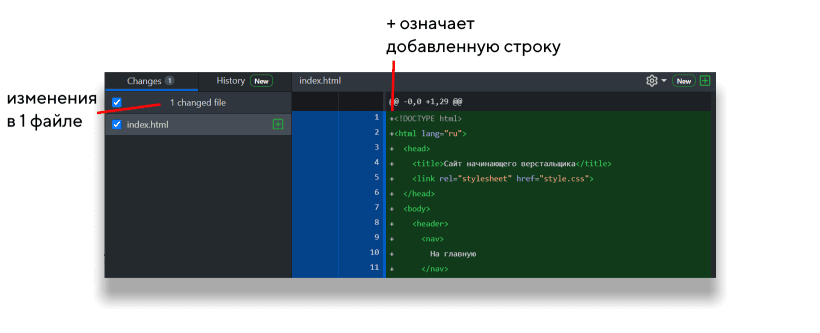


Рисунок 20 – вид при добавлении файла

**Шаг 9.** Загружаем новый репозиторий на *GitHub*: после того, как мы добавили какой-то код в свежесозданный репозиторий, нужно сделать коммит, то есть зафиксировать все сохранённые изменения и дать им название. Текст должен быть лаконичным и в то же время сообщать о том, что делает коммит. Например, «добавляет имя наставника в *Readme*», «вводит функцию сортировки изображений», «правит ошибку в поиске городов на карте». Вводим имя жмём большую синюю кнопку *«Commit to main»*

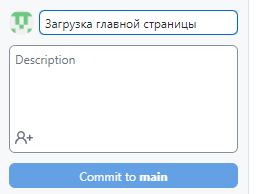


Рисунок 21 – создание коммита

**Шаг 10.** Изменения, которые мы внесли и сохранили, пока локальны. Их нужно послать на *GitHub.* Чтобы опубликовать свежесозданный репозиторий на *GitHub*, нажмите *Publish repository*.

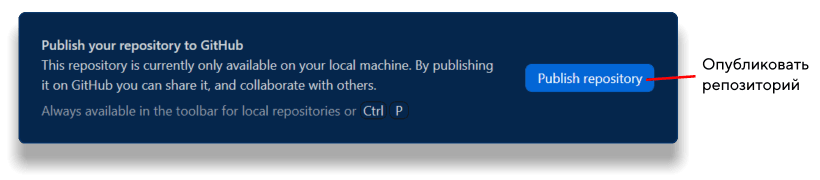


Рисунок 22 – опубликовать репозиторий

**Шаг 11.** Появится подтверждение о публикации репозитория – проверяем название и описание, если нужно, ставим галочку о том, что код приватный, и публикуем.

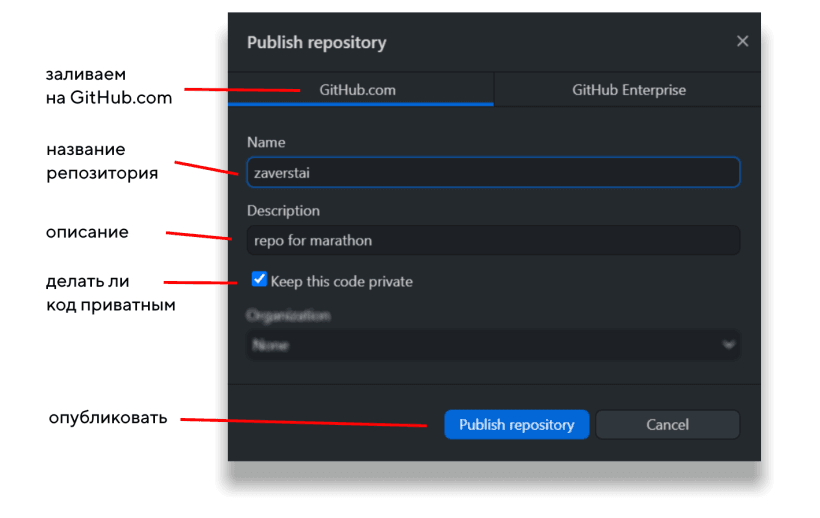


Рисунок 23 – подтверждение о публикации репозитория

Готово – после этого репозиторий появится в вашем профиле на *GitHub. com.*

**Шаг 12.** Добавляем код и коммитим изменения: репозиторий создан и загружен на GitHub, теперь нужно добавить немного кода.

Когда вы допишете код в файлы, которые находятся в репозитории, вы сможете просмотреть все их изменения в окне *GitHub Desktop*. Вот здесь, например, мы изменили «второй» на «третий» в тексте страницы – и изменения сразу видны, можно проверить, что всё исправленное будет загружено.

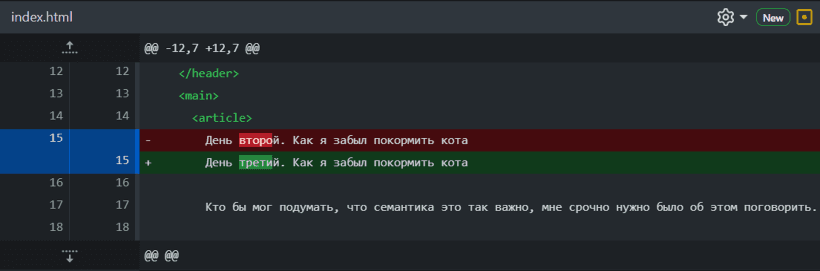


Рисунок 24 – просмотр исправлений

**Шаг 13.** Дальше действуем по проверенной схеме – коммитим изменения.

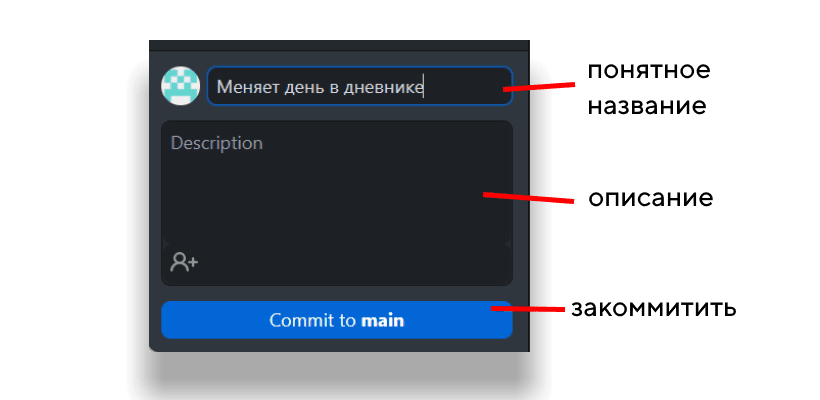


Рисунок 25 – коммит изменений

**Шаг 14.** В центре главного экрана появится предложение запушить коммит в удалённый репозиторий. Соглашаемся и жмём *Push origin*.

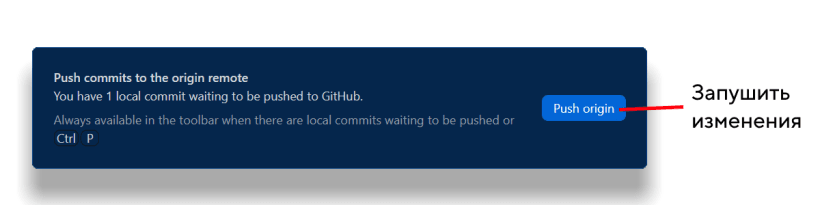


Рисунок 26 – отправка изменений

Готово! Теперь, если зайти на [*GitHub.com*](http://github.com/), в наш репозиторий, увидим изменённый файл, который мы только что отправили.

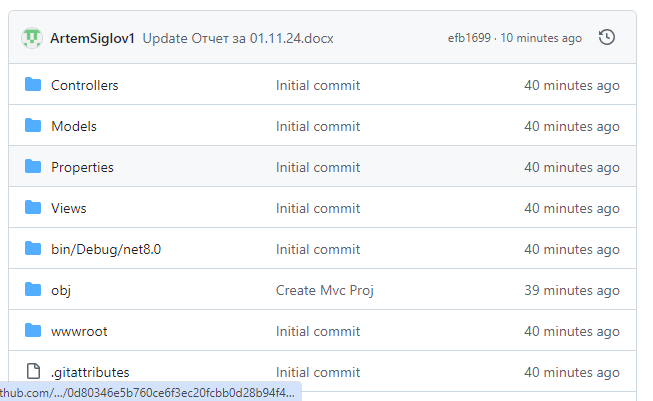


Рисунок 27 – проверка отправки изменений

**Вывод:** Научился и закрепил навыки создания или клонирования репозиториев, чтобы предоставить доступ к проекту в любом месте где есть доступ к интернету.

**04.11.24**

**Тема:** Определение структуры *MVC*-приложения, взаимодействие моделей, контроллеров и представлений. Оформление отчетной документации.

**Ход работы:**

**Шаг 1.** Изучим такие вопросы как: чистая архитектура, правило зависимостей, сущности, сценарии, интерфейс-адаптеры, фреймворки и драйверы, пересечение границ. Разберем может ли быть более четырех кругов, а также как данные пересекают границы.

**Шаг 2.** В обозревателе решений ПКМ кликаем на решение. Выбираем пункт «Добавить», «Создать папку решения». Как показано на рисунке 28.

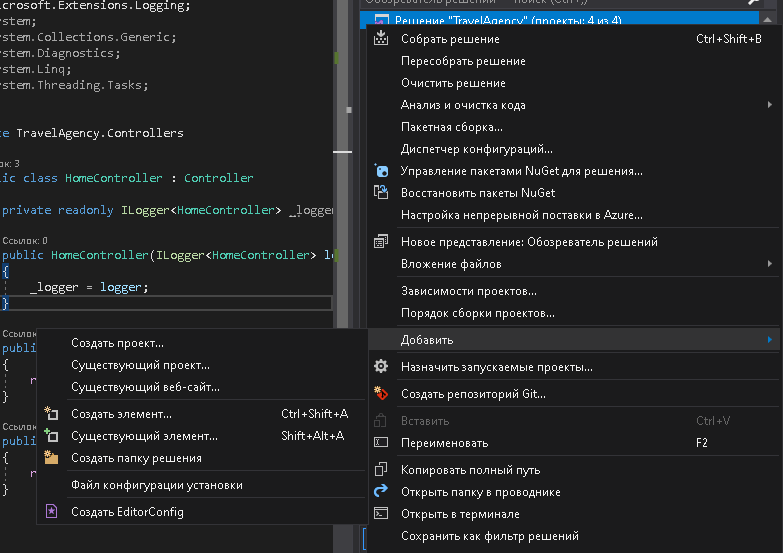


Рисунок 28 –­ создание папки решения.

**Шаг 3.** В обозревателе решений найдем созданную папку в предыдущем шаге. Кликаем на нее ПКМ. Выбираем пункт «Добавить», «Создать проект» проект должен быть типа «Библиотека классов» как показано на рисунке 29.

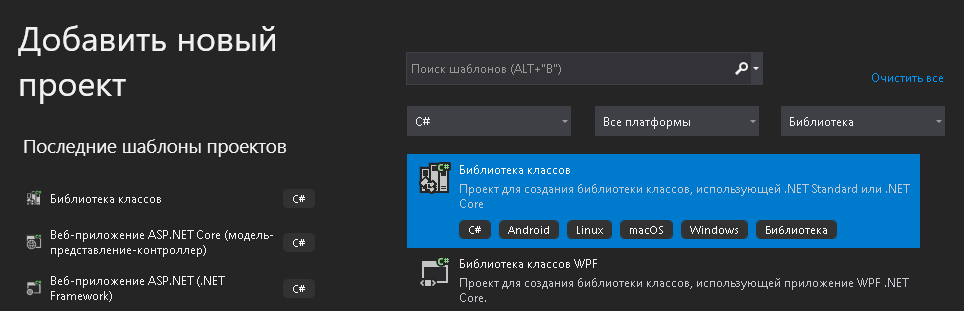


Рисунок 29 – создание проекта «Библиотека классов»

Добавить что получилось в результате

**Вывод:** Определил структуры *MVC*-приложения, взаимодействие моделей, контроллеров и представлений. Оформил отчетную документацию.

**05.11.24**

**Тема:** Frontend разработка основная страница: написание шапки сайта, секции «О нас».

Оформление отчетной документации.

**Ход работы:**

Продолжим с разметки главной страницы, а именно в файле *\_Layot.cshtml* приведем содержимое тега *<body>* к следующему виду в соответствии с рисунком 30. В данном фрагменте представлена разметка шапки сайта.

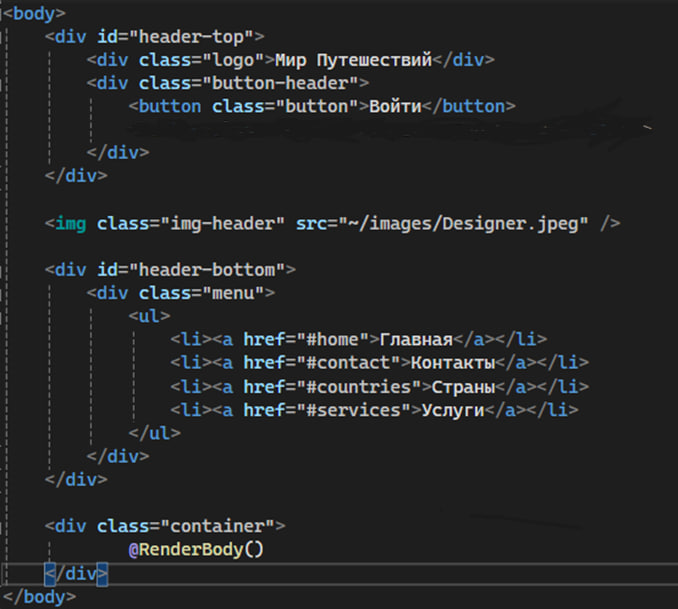


Рисунок 30 – разметка шапки сайта

На данный момент если вы запустите приложение, то оно будет выглядеть ужасно так как проект еще не содержит никаких *CSS* свойств. Что бы это исправить в папке «*css*» удалим имеющийся файл «*site.css*» и создадим новый файл *«shared.css».* Нажмем правой кнопкой мыши по папке в которую хотим добавить элемент и выберем Добавить/Создать элемент… как показано на рисунке 31.

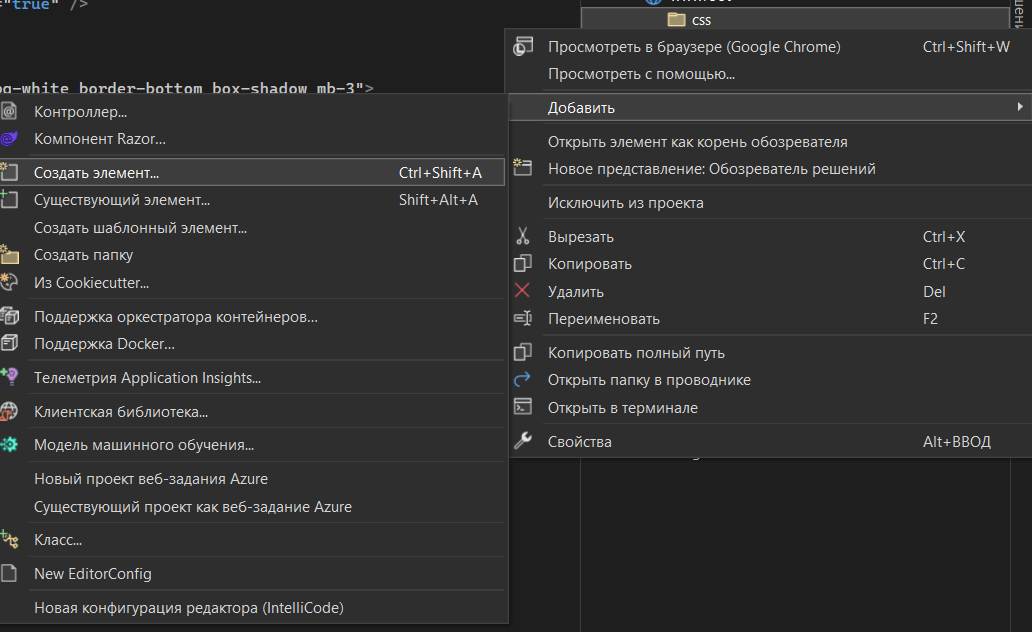


Рисунок 31 – открытие окна «добавление нового элемента»

В открывшемся окне необходимо выбрать в левом меню *C#/ASP.NET Core*/Веб/Содержимое/Таблица стилей. Выбранный файл нужно назвать *shared.css* как показано на рисунке 32.

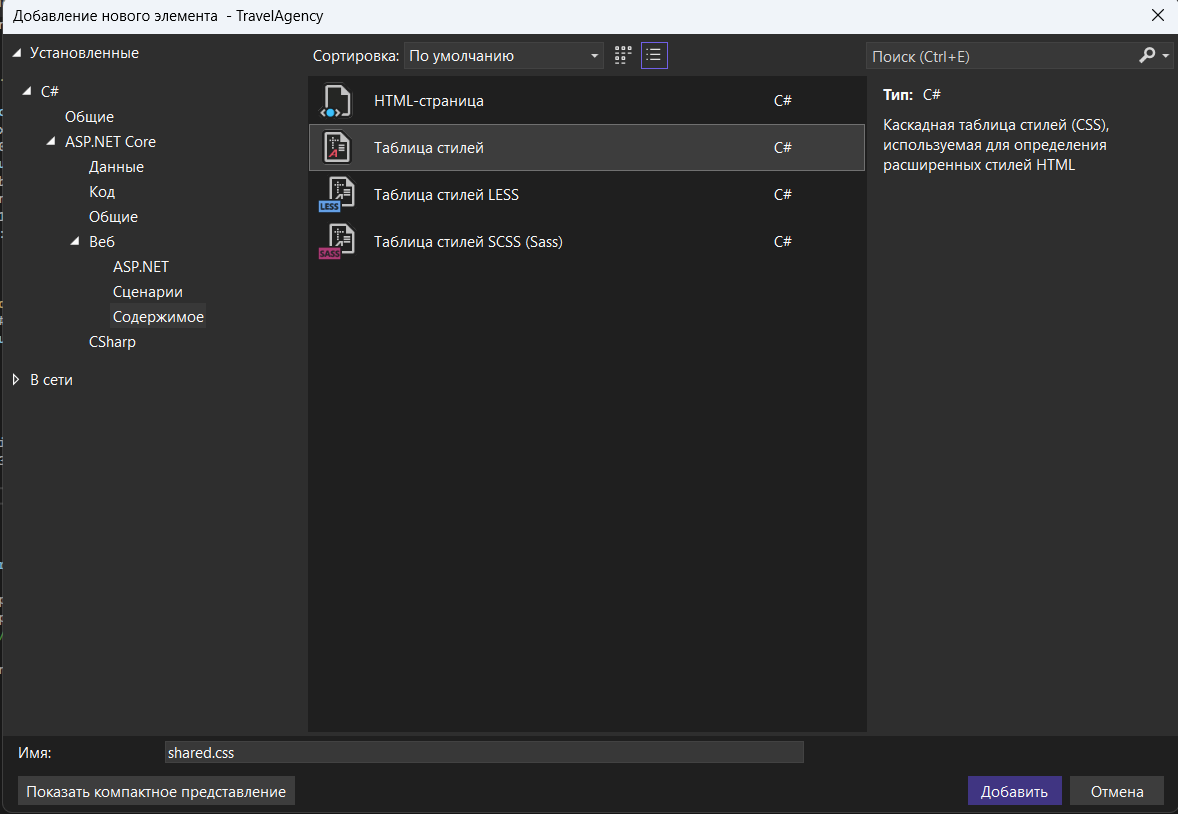


Рисунок 32 – добавление файла таблицы стилей

В появившемся файле прописываем пару стилей для тега *<html>, \*,\*::before,\*::after* и *body* в соответствиис рисунком 33.

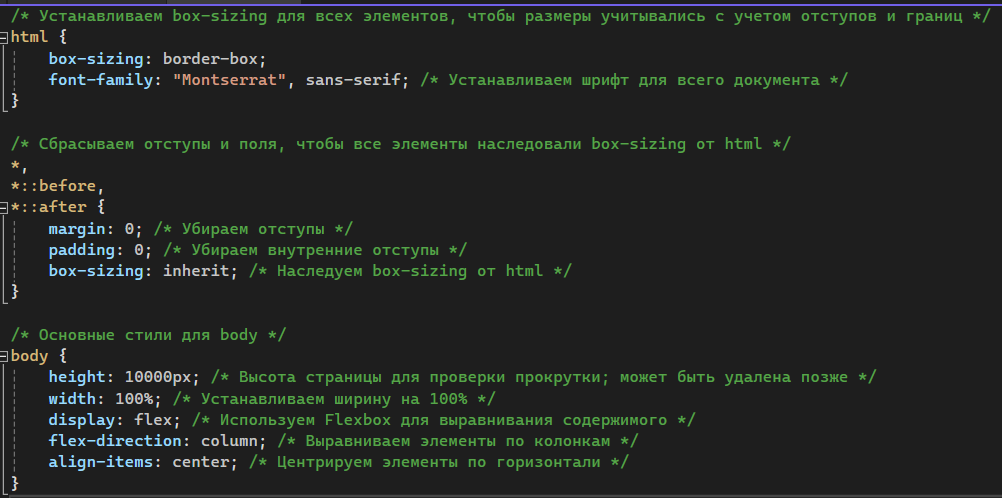


Рисунок 34 – стили *<html>*, *\*,\*::before,\*::after* и *body*

Далее прописываем стили для логотипа как показано на рисунке 35 в нашем случае это просто текст.

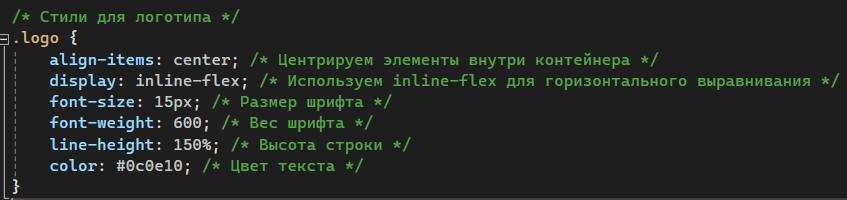


Рисунок 35 – стили для логотипа сайта

В данном проекте было принято решение разделить шапку сайта на две части. Это несет в себе исключительно эстетический смысл. В верхней части будут располагаться слева логотип, а справа две кнопки «Войти» и «Зарегистрироваться», далее по центру расположена картинка и под ней вторая часть шапки, где расположено меню. Для данного элемента сайта показаны стили на рисунке 36.

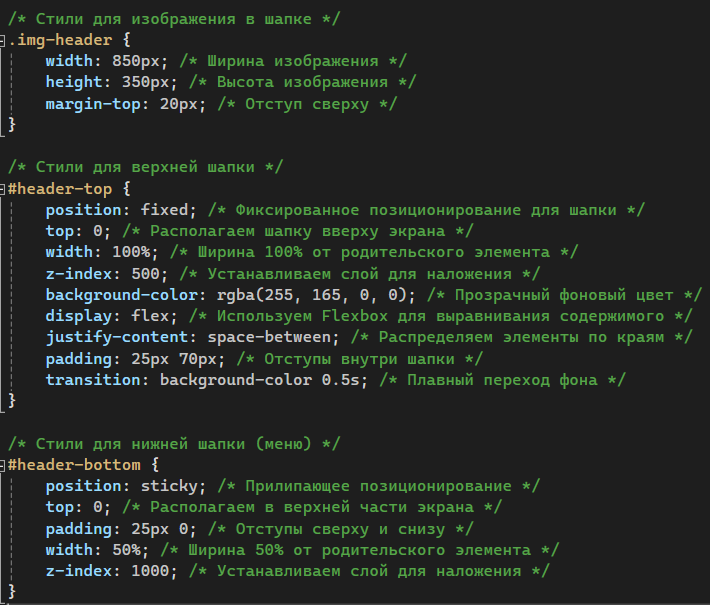


Рисунок 36 – стили шапки сайта

Ниже на рисунке 37 представлены стили для кнопок в верхней шапке.

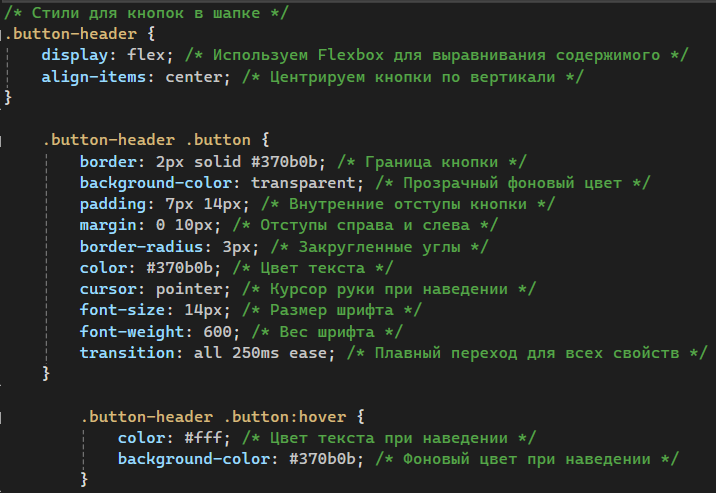


Рисунок 37 – стили для кнопок верхней шапки

Далее на рисунке 38 продемонстрированы стили для самого меню в нижней шапке.

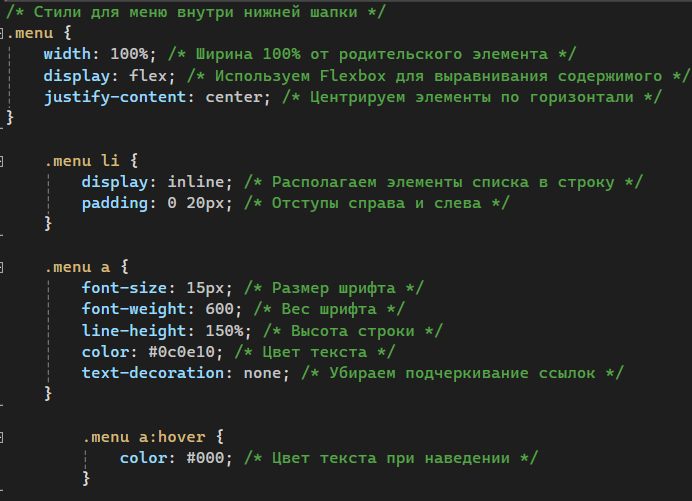


Рисунок 38 – стили для меню нижней шапки

Чтобы создать эффект, при котором шапка изначально прозрачная, а затем постепенно становится оранжевой при прокрутке, мы будем использовать *CSS* вместе с *JavaScript*. Для создания *js* файла откроем окно «Добавление нового элемента» и выберем *C#/ASP.NET Core*/Веб/Сценарии/Файл *JavaScript.* Выбранный файл необходимо назвать «*shared\_script.js»* как показано на рисунке 39.

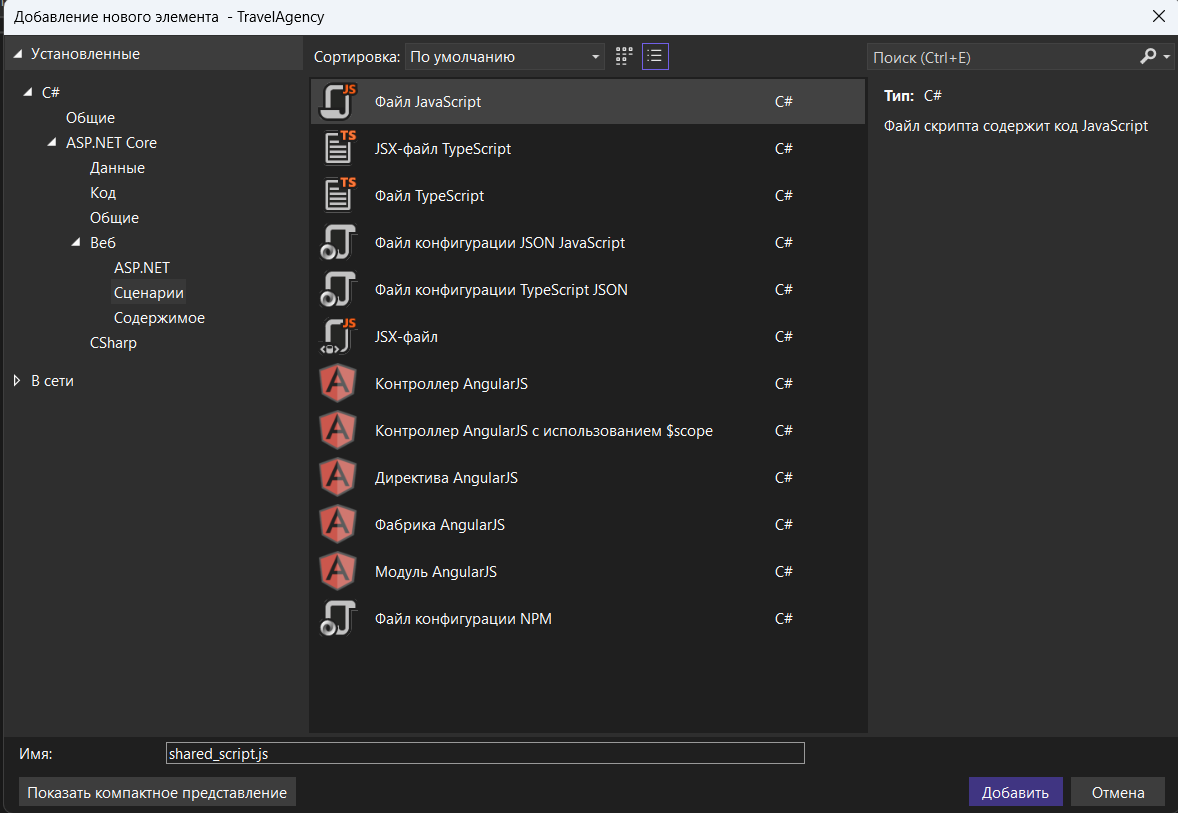


Рисунок 39 – создание файла *JavaScript*

Далее в созданном файле прописываем следующий скрипт как показано на рисунке 40. Теперь если запустите проект, то при прокрутке шапка будет постепенно изменять свой цвет.

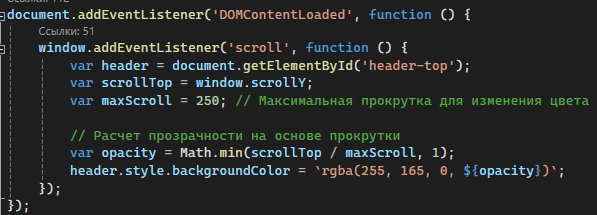


Рисунок 40 – скрипт для изменения цвета шапки с прозрачного на оранжевый

Определим стиль контейнера для тела сайта как показанно на рисунк 41.

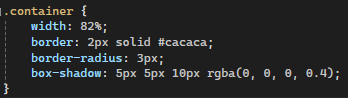


Рисунок 41 – стили для контейнера

Теперь можно перейти к реализации основного контента для этого создадим файл *SiteInformation.cshtml* в паке Home как показано на рисунке 42.

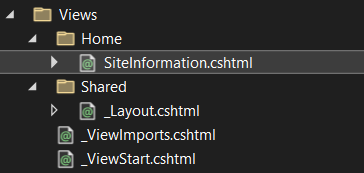


Рисунок 42 – структура папки Views на данном этапе разработки

Начнем с информации «О нас» пропишем разметку страницы как показано на рисунке как показано на рисунке 43. В выделенное место необходимо скопировать текст 1 из приложения А.

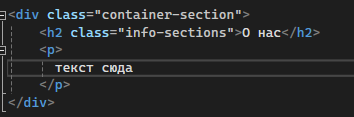


Рисунок 43 – «О нас» *html-*разметка

Применим стили к информации «О нас» в соответствии с рисунком 44

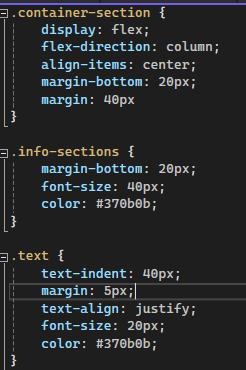
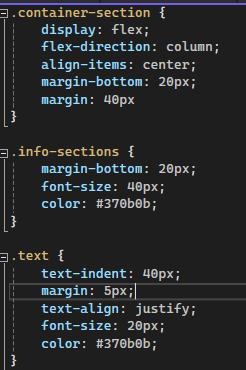


Рисунок 44 – «О нас» стили

В конце данной главы у вас должен получится вот такой результат.

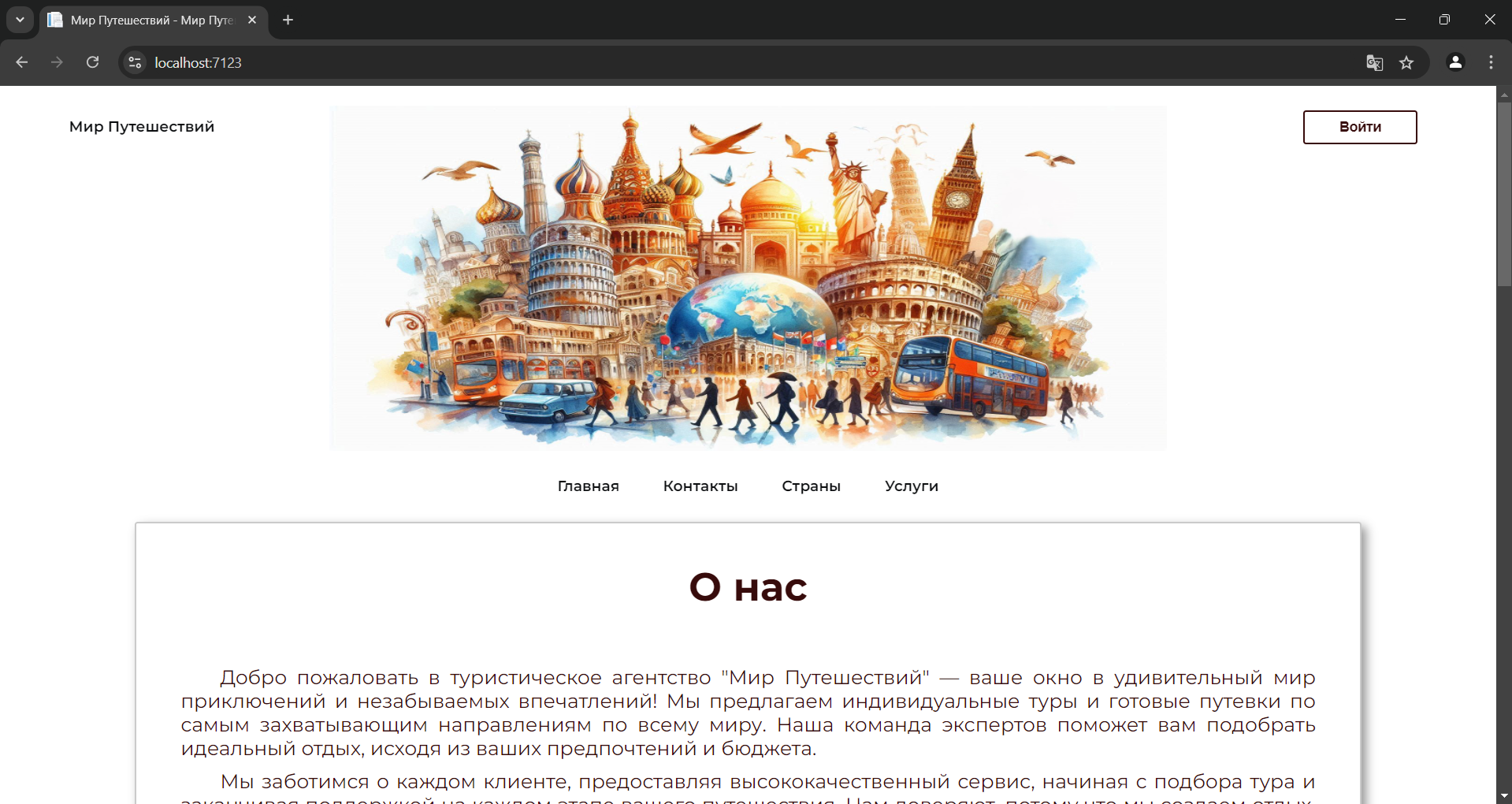


Рисунок 45 – Результат