01.11.2024

Тема: Создание проекта на основе шаблона *MVC.*

Цель работы: Ознакомиться с созданием проекта на основе шаблона *MVC.*

Ход работы:

***ASP.NET*** – это фреймворк от Microsoft для разработки веб-приложений и веб-сервисов. Он позволяет создавать динамичные сайты, поддерживает различные языки программирования, предлагает встроенные функции безопасности и оптимизирован для высокой производительности. Включает технологии, такие как *ASP.NET MVC и ASP.NET Web API.*

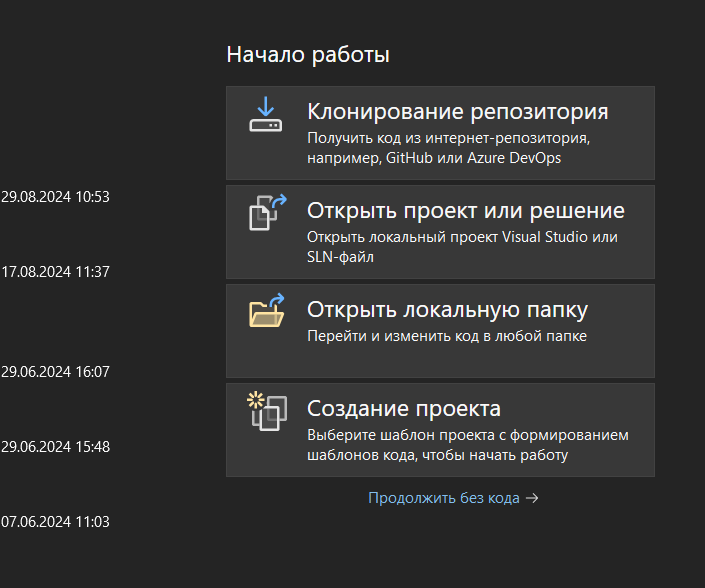


Рисунок 1 – Создание проекта

Создание веб-приложения на *ASP.NET Core MVC* начинается с установки *Visual Studio* и .*NET SDK*. Затем создайте новый проект, выбрав шаблон "*Web Application (Model-View-Controller*)". В структуре проекта будут папки для контроллеров, моделей и представлений. Создайте модель, например, *User,* и контроллер *UsersController* с методом *Index,* который возвращает представление. Добавьте представление *Index.cshtml* в папку *Views/Users*. Наконец, запустите приложение, нажав "Запуск" в *Visual Studio*. Это основные шаги для начала работы!

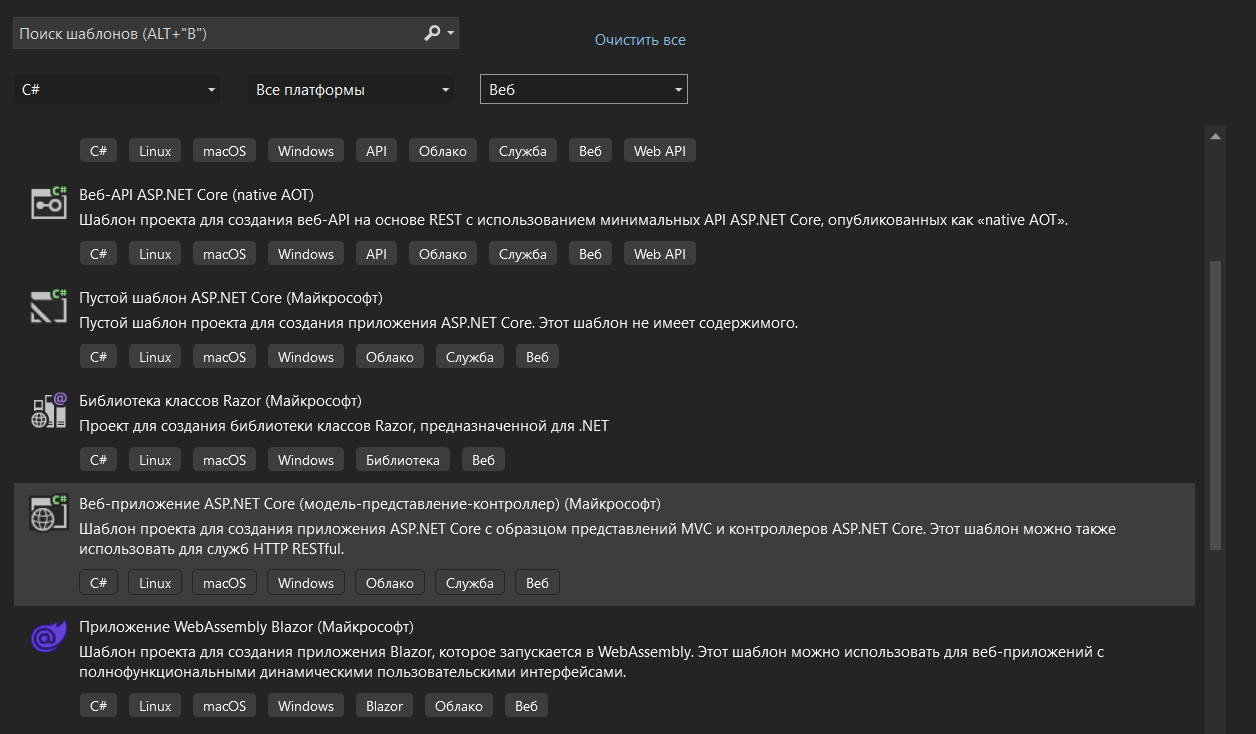


Рисунок 2 – Шаблоны

*MVC* — это архитектурный шаблон, который делит приложение на три основных компонента. Модель отвечает за данные и бизнес-логику, управляя взаимодействием с базой данных. Представление отвечает за отображение информации пользователю, формируя интерфейс. Контроллер обрабатывает входящие запросы, взаимодействует с моделью и выбирает, какое представление отобразить. Эта структура позволяет улучшить организацию кода, облегчает тестирование и масштабирование приложения, а также повышает его гибкость и поддержку.

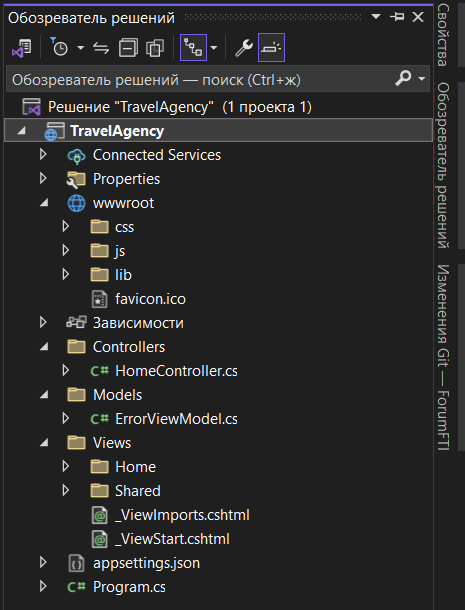


Рисунок 3 – Обозреватель решений

В проекте есть несколько ключевых папок для работы. Папка ***wwwroot***включает в себя директорию для стилей, папку для скриптов, обеспечивающих динамическое взаимодействие, и раздел для дополнительных файлов, которые могут упростить разработку при правильном использовании. В данном проекте будет применяться чистый *JavaScript.* Папка ***Controllers***отвечает за обработку входящих запросов. Здесь определяются конкретные URL-адреса, по которым пользователи попадают на страницы приложения при переходе по ссылкам или нажатии кнопок. Папка ***Models***предназначена для управления данными, которые хранятся и обрабатываются на сервере.

Папка ***Views*** содержит *HTML*-шаблоны, которые сервер возвращает после обработки запроса. В обозревателе решений можно выбрать файл ***Views/Shared/\_Layout.cshtml***. Файлы макетов, или мастер-страницы, позволяют задать единый шаблон для страниц *Razor*, создавая унифицированный вид приложения. Они используют код *Razor* и *HTML*, как и обычные страницы, но упрощают процесс разработки. С помощью мастер-страниц можно определить общие элементы, такие как меню и другие компоненты, а также подключить общие стили и скрипты. Это избавляет от необходимости указывать путь к файлам стилей на каждой странице и упрощает их изменение в будущем. Тег <*head*>, как показано на рисунке, предназначен для хранения служебной информации о странице и располагается первым внутри тега <*html*>, сразу перед <*body*>. Внутри <*head*> обычно находятся заголовок, ключевые слова, описание страницы и другие служебные данные.

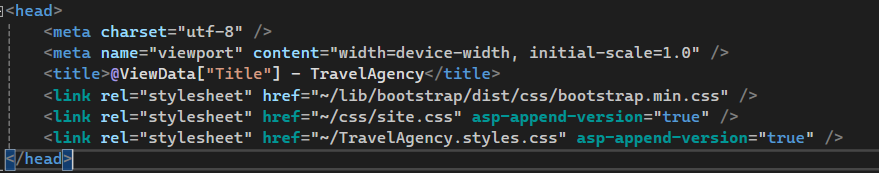


Рисунок 4 – тег <*head*>

Элемент <*header*> является семантическим и служит для отделения вводного содержимого или навигации от основного контента страницы или секционного блока. Например, он может использоваться для создания "шапки" с навигацией для всего сайта, а также для отображения имени автора и даты публикации в превью статьи или новости.

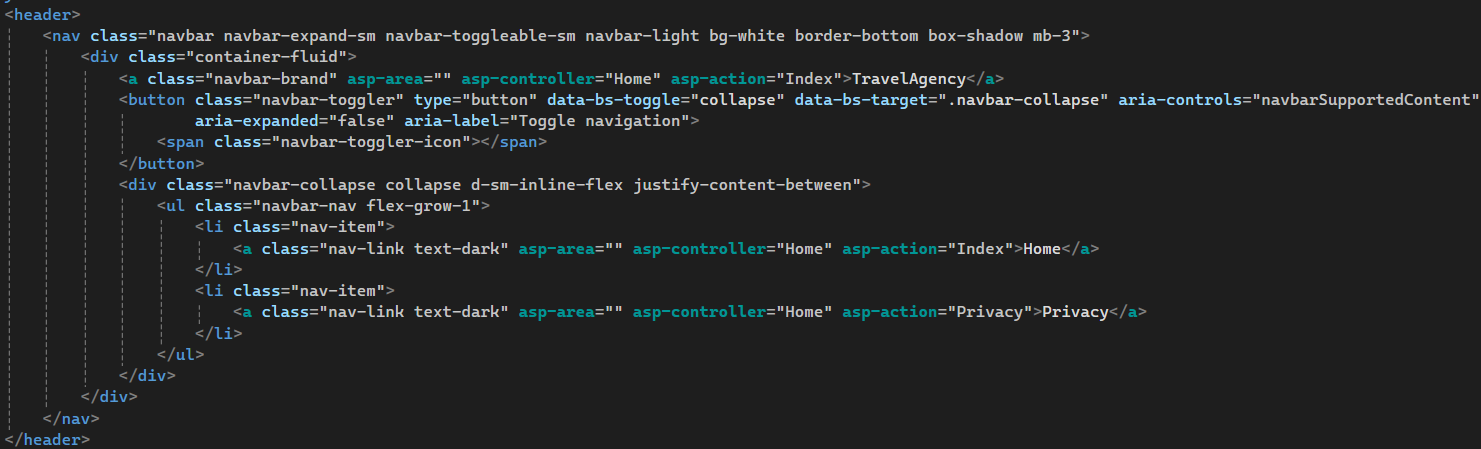


Рисунок 5 – тег <*header*>

Основной особенностью файлов макетов является использование метода @*RenderBody*(). Этот метод вставляет содержимое страниц *Razor*, которые используют данную мастер-страницу. Благодаря этому можно легко установить единообразный стиль оформления для всех страниц приложения.

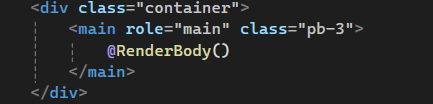


Рисунок 6 – функция @*RenderBody()*

Тег <*footer*> служит для создания футера сайта, располагающегося в нижней части страницы. В этом разделе обычно размещаются такие элементы, как контактная информация, нижнее меню и полезные ссылки. Скрипты чаще всего подключаются из внешних файлов с расширением .*js*. Для этого используется тег <*script*> с атрибутом *src*, в котором указывается путь к файлу. Рекомендуется размещать подключение внешних скриптов перед закрывающим тегом </*body*>, чтобы обеспечить правильную загрузку и работу страниц.

В папке ***Views/Shared*** также находятся файлы \_*ViewImports.cshtml* и \_*ViewStart.cshtml*. Файл \_*ViewImports.cshtml* содержит две строки кода, которые предназначены для глобального подключения пространств имен и дополнительных функций, применяемых ко всем представлениям проекта *ASP.NET Core MVC.* Это упрощает разработку и обеспечивает согласованность.

Файл \_*ViewStart.cshtml*, как правило, тоже содержит лишь пару строк. Он используется для указания файла макета (*layout)*, который будет применяться к текущему представлению. Это позволяет установить общий стиль для всех страниц, значительно упрощая поддержку и изменения в будущем.

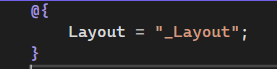


Рисунок 7 – содержимое файла \_*ViewStart.cshtml*

В разделе представлений есть еще одна важная папка — ***Home***. В ней хранятся представления, связанные с действиями контроллера ***HomeController***. Этот контроллер обычно отвечает за основные страницы веб-приложения, такие как главная страница, страница контактов и другие.

По умолчанию при запуске проекта вызывается именно этот контроллер, и действие ***Index***. Если необходимо изменить поведение по умолчанию, это можно сделать в файле ***Program.cs***, в соответствующих строках, как показано на рисунке.

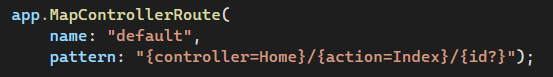


Рисунок 8 – фрагмент кода из файла *Program cs*