**Коллоквиум 2, СТРWП-ИСиТ-3-2023**

***Вопросы не пишем, пишем номер и ответ***.

1. В правом верхнем углу листа:

***2-СТРWП-ИСиТ-3/группа, Фамилия И.О.***

1. Поясните термин «программная платформа».

Программная платформа представляет собой общую организацию исполнения прикладных программ, задавая, например, порядок запуска программы, схему использования ею адресного пространства, зафиксированные в архитектуре операционной системы плюс API на уровне операционной системы45.

1. Состав платформы .NET CORE.

В состав .NET Core входят CoreCLR (.NET runtime c низкоуровневыми классами, сборщиком мусора, JIT-компилятором RyuJIT, средствами для загрузки байткода и его компиляции в машинный код), стандартные библиотеки, библиотеки CoreFX и утилиты, предоставляющие средства для разработки микросервисов, библиотек, серверных и консольных приложений на базе платформы.NET9.

1. Что такое OWIN?

OWIN (Open Web Interface for .NET) – это спецификация, определяющая интерфейс, который устраняет сильную связанность веб-приложения с конкретной реализацией сервера.

1. Что такое Nuget?

NuGet — это бесплатный пакетный менеджер с открытым исходным кодом, служащий для .NET и .NET Core механизмом совместного использования кода, поддерживаемым Microsoft.

1. Назначение платформы ASP.NET CORE.

ASP.NET Core предоставляет платформу для создания веб-приложений и служб, приложений Интернета вещей (IoT) и серверных частей для мобильных приложений.

1. Основные свойства платформы ASP.NET CORE.

ASP.NET Core предоставляет следующие преимущества: единое решение для создания пользовательского веб-интерфейса и веб-API, разработано для тестируемости, Razor Pages упрощает написание кода для сценариев страниц и повышает его эффективность, Blazor позволяет использовать в браузере язык C# вместе с JavaScript.

1. Четыре компонента приложения ASP.NET CORE и их назначение.

Middleware: компоненты для обработки запросов и формирования ответов.

Routing: маршрутизация запросов к соответствующим обработчикам.

Model-View-Controller (MVC): шаблон проектирования для построения веб-приложений.

Entity Framework Core: ORM для работы с базами данных.

1. За счет чего достигается кроссплатформенность .NET CORE.

Кроссплатформенность .NET Core достигается благодаря тому, что .NET Core может работать поверх кросс-платформенной среды .NET Core, которая может быть развернута на основных популярных операционных системах: Windows, Mac OS, Linux.

1. Поясните термин «статический ресурс» приложения ASP.NET CORE. Приведите примеры.

Статический ресурс в приложении ASP.NET CORE - это файл, который не требует обработки на сервере при каждом запросе. Примеры: изображения, таблицы стилей (CSS), клиентские скрипты (JavaScript).

1. Поясните суть приложения ASP.NET CORE HTTP-Handler.

HTTP-Handler в ASP.NET Core - это процесс, который выполняется в ответ на запрос, сделанный к приложению ASP.NET. Каждый запрос обрабатывается HTTPHandler на основе расширения файла запроса (.aspx, .asmx и т.д.). Наиболее распространенным обработчиком является обработчик страниц ASP.Net, который обрабатывает файлы .aspx.

1. Поясните назначение класса HttpContext.

Класс HttpContext в ASP.NET предоставляет ссылку на объект текущего HTTP-запроса. Этот объект дает доступ к встроенным объектам Request, Response и Server свойствам запроса.

1. Расшифруйте аббревиатуру MVC и поясните каждый компонент.

MVC - это аббревиатура для Model-View-Controller:

Model (Модель):Отвечает за данные, методы работы с ними и структуру программы.

View (Представление):Отображает данные на уровне пользовательского интерфейса.

Controller (Контроллер):Обеспечивает взаимодействие с системой, обрабатывает действия пользователя, проверяет полученную информацию и передает ее модели.

1. Поясните назначение компонента «маршрутизатор» приложения ASP.NET CORE MVC. Время жизни маршрутизатора.

В ASP.NET Core MVC, маршрутизатор отвечает за обработку входящих запросов и определение, какой контроллер и действие должны быть вызваны для обработки запроса.

1. Поясните назначение атрибута Route.

Атрибут Route используется для определения маршрута к конкретному действию контроллера.

1. Поясните назначение компонента «котроллер» приложения ASP.NET CORE MVC. Время жизни котроллера.

Контроллер в ASP.NET Core MVC обрабатывает запросы, получает данные модели, вызывает шаблоны представления и возвращает ответ.

Время жизни контроллера ограничено временем обработки конкретного запроса.

1. Поясните назначение объектов ViewBag и ViewData и принцип их применения.

Объекты ViewData и ViewBag доступны в методах контроллера и в представлениях и позволяют передавать данные из контроллера в представление.

1. Назначение и применение атрибутов: HttpGet, HttpPost.

Атрибуты HttpGet и HttpPost используются для указания типа HTTP-запроса, который может обрабатывать метод действия.

1. Назначение и применение атрибутов: NonAction, NonController, ActionName.

Атрибуты NonAction, NonController, ActionName используются для управления поведением контроллеров и действий в ASP.NET Core MVC.

1. Назначение и применение метода контроллера OnAction.

Метод OnAction в контроллере ASP.NET Core MVC выполняется после выполнения метода действия.

1. Поясните назначение и принцип работы технологии Dependency Injection.

Dependency Injection - это техника программирования, которая делает класс независимым от его зависимостей. Он достигает этого, разделяя использование объекта от его создания.

1. Dependency Injection: жизненный цикл Transient-объекта.

Dependency Injection: жизненный цикл Transient-объекта. Transient-объекты в Dependency Injection создаются при каждом обращении к сервису. Это значит, что в течение одного запроса может быть несколько обращений к сервису, и при каждом обращении будет создаваться новый объект.

1. Dependency Injection: жизненный цикл Singleton-объекта.

Dependency Injection: жизненный цикл Singleton-объекта. Singleton-объекты в Dependency Injection создаются при первом обращении к нему, и все последующие запросы используют один и тот же ранее созданный объект сервиса4.

1. Поясните назначение компонента «представление» приложения ASP.NET CORE MVC. Жизненный цикл представления.

Назначение компонента «представление» приложения ASP.NET CORE MVC. Жизненный цикл представления. В ASP.NET MVC Core, представления - это файлы с расширением cshtml, которые содержат код пользовательского интерфейса в основном на языке html, а также конструкции Razor24. Представление отвечает за отображение данных модели пользователю, реагируя на изменения модели.

1. Поясните назначение компонента «Razor Engine» приложения ASP.NET CORE MVC.

Назначение компонента «Razor Engine» приложения ASP.NET CORE MVC. Razor Engine - это движок представлений по умолчанию для приложений ASP.NET Core. Он ищет разметку Razor в файле представления, анализирует ее и создает HTML-ответ.

1. Назначение Razor-директивы model.

Назначение Razor-директивы model. Директива @model в Razor используется для указания типа модели для представления или страницы.

1. Назначение Razor-директивы inherits.

Назначение Razor-директивы inherits. Директива @inherits в Razor позволяет указать базовый класс для представления или страницы.

1. Назначение Razor-директивы inject.

Назначение Razor-директивы inject. Директива @inject в Razor используется для внедрения сервиса в представление или страницу.

1. Назначение Razor-директив: RenderSection, RenderBody, section.

Назначение Razor-директив: RenderSection, RenderBody, section. @RenderBody() используется в макете для отображения содержимого дочерних представлений. @RenderSection() используется для отображения определенной секции из дочернего представления. @section используется в дочернем представлении для определения секции.

1. Поясните понятие «html helper».

Понятие «html helper». HTML Helper - это метод, используемый для генерации HTML-разметки.

1. Поясните понятие «tag helper».

Понятие «tag helper». Tag Helper - это новая функция MVC, которую удобно использовать для генерации HTML кода. Они выглядят как обычные HTML элементы и атрибуты, но обрабатываются движком Razor.

1. Назначение фреймворка Identity.

Фреймворк Identity в ASP.NET Core представляет систему аутентификации и авторизации. Он облегчает разработку и объединение разных компонентов большого программного проекта. Identity предоставляет функциональность для управления пользователями, включая хранение паролей, управление ролями, аутентификацию и авторизацию.

1. Каким образом передается Identity-токен авторизации.

Identity-токен авторизации обычно передается через HTTP-заголовки. В случае использования схемы "Bearer Token", токен будет передан в заголовке Authorization с префиксом Bearer.

1. Перечислите 5 свойств пароля, которые можно установить в Identity.

В Identity можно установить следующие свойства пароля: RequiredLength: минимальная длина пароля.

RequireNonAlphanumeric: требование наличия хотя бы одного не алфавитно-цифрового символа.

RequireDigit: требование наличия хотя бы одной цифры. RequireLowercase: требование наличия хотя бы одного символа в нижнем регистре.

RequireUppercase: требование наличия хотя бы одного символа в верхнем регистре.

1. Поясните понятие роль в фреймворке Identity.

Роль в фреймворке Identity представляет группу пользователей с определенными правами. Роли позволяют создавать группы пользователей с определенными правами и в зависимости от принадлежности к той или иной группе, разграничивать доступ к ресурсам приложения.

1. Назначение атрибута Authorize.

Атрибут Authorize в ASP.NET Core используется для ограничения доступа к контроллеру или действию пользователям, прошедшим проверку подлинности. Этот атрибут проверяет, прошел ли пользователь проверку подлинности, и если нет, то перенаправляет его на страницу входа.