

1. Методите на контролерот се нарекуваат:
 - a. Рути
 - b. Акции**
 - c. Пакети
 - d. Параметри
2. Кој е одговорен за пренос на податоци до погледот?
 - a. Контролерот**
 - b. Погледот
 - c. Рутерот
 - d. Моделот
3. Кое од следниве тврдења е точно?
 - a. Акциски метод е приватен метод
 - b. Акциски метод е protected
 - c. Акциски метод е static
 - d. Акциски метод е public**
4. Кое од слените е тип на поглед во MVC
 - a. Designer view
 - b. Partial view**
 - c. Data view
 - d. Executable view
5. Кое од следниве погледи содржи заеднички споделени делови на корисничкиот интерфејс?
 - a. Layout view**
 - b. HTML view
 - c. Partial view
 - d. Razor view
6. Што користи Razor View Engine-от за да рендерира серверска содржина?
 - a. <%= %>
 - b. @**
 - c. /* */
 - d. \$
7. Која валидациска анотација овозможува да се специфицира минимум и максимум ограничувања на нумеричка вредност
 - a. Range**
 - b. Required
 - c. Regularexpression
 - d. String length
8. Доколку за објектот кој се превзема податоци со LINQ постојат поврзани објекти, истите _____.
 - a. Автоматски се ПРЕЗЕМААТ
 - b. Се преземаат делумно, за поврзаните објекти
 - c. Автоматски не се преземаат**
 - d. Ниедно од наведените
9. Што прави следниот JQuery селектор \$("#main")?
 - a. Ги наоѓа сите елементи со класа main
 - b. Ги наоѓа сите <main> елементи
 - c. Ги наоѓа сите наследници на елементи со ид вредност main
 - d. Ги наоѓа сите елементи со ид вредност main**
10. OAuth 2.0 е
 - a. Апликациски протокол
 - b. Делегациски протокол**
 - c. Ниедно од наведените
 - d. Автентикациски протокол
11. OAuth 2.0
 - a. Дозволува корисникот да има пристап и изведува акции во име на друг корисник
 - b. Ниедно од наведените
 - c. Дозволува пристап на друга личност или апликација да изведува акции во име на корисник**
 - d. Автентифицира корисник на апликација
12. Кој од следните HTTP методи овозможува креирање на ресурс на серверот?
 - a. HTTP DELETE
 - b. HTTP PUT
 - c. HTTP GET
 - d. HTTP POST**
13. Default HTTP метод во акциите од контролерите во Web API е:
 - a. HTTP PUT
 - b. HTTP GET
 - c. HTTP POST**
 - d. HTTP DELETE
14. Потребно е дадена MVC апликација да се направи да биде пристаплива само за логирани корисници од било која улога со исклучок на почетната страница, кои од следните акции треба да се преземат?
 - a. Да се додаде filters.Add(new AuthorizeAttribute()); во BundleConfig
 - b. Да се дефинира [AllowAnonymous] над Index акцијата во Home контролерот**
 - c. Да се додаде [Authorize(Roles='Administrator')] над Home контролерот

- d. Да се додаде `filters.Add(new AuthorizeAttribute());` во `FilterConfig` фајлот

15. Нека е даден следниот код:

```
[Authorize(Roles = "Administrator,Editor")]
0 references
public class MoviesController : Controller
{
    private ApplicationDbContext db = new ApplicationDbContext();

    // GET: Movies
    [Authorize(Roles = "Editor")]
    0 references
    public ActionResult Index()
    {
        return View(db.Movies.ToList());
    }
}
```

Кои од наведените улоги имаат пристап до Index ?

- a. Ниедно од наведените
 - b. Само Administrator
 - c. **Само Editor**
 - d. Administrator и Editor
16. Кое од наведените тврдења е точно за инсталација и прикажување на `bootstrap` дијалог?
- a. Потребно е да се повика `bootstrap.confirm("Are you sure you want to delete?", function(result){});` во `document.ready`
 - b. Потребно е да се промени `Layout.cshtml` фајлот за вклучување на `bootstrap` ViewModel
 - c. Потребно е да се вклучи референца до скриптите од `Bootstrap` во `BundleConfig`
 - d. Потребно е да се инсталира `Bootstrap` преку `NuGet Console` со командата `install-package bootstrap`
17. Кои од наведените тврдења е точно за прикажување на `jQuery DataTable` компонента со `id mytable`
- a. Табелата мора да содржи `<tbody>`
 - b. Треба да се повика `$("#myTable").DataTable()` во `document.ready` функцијата
 - c. Табелата мора да содржи `<thead>`
 - d. Табелата мора да содржи помалку од 100 редици
18. Со која од следниве анотации во моделот може да се промени описот на дадено поле во погледите?
- a. `[Name = "download URL"]`
 - b. `[Display(Name="Download URL")]`
 - c. `[Name="Download URL"]`
 - d. `[Requires (Name="Download URL")]`
19. Со која од наведените компоненти можеме да доделуваме улоги на даден корисник?
- a. Identity Manager
 - b. Role Manager
 - c. Claim Manager
 - d. **User Manager**
20. Кои од наведените тврдења се точни за ViewModels?
- a. ViewModel-от се користи со цел да се креираат табели во базата на податоци
 - b. ViewModel-от може да содржи конкретни правила за валидација на податоците
 - c. **ViewModel-от може да содржи повеќе ентитети или објекти од различни изворни модели**
 - d. **ViewModel-от содржи полиња кои се потребни за репрезентација во рамките на view компонентата**
21. Што претставува последниот параметар во следниот код:

```
<div id="dailydeal">
    @Ajax.ActionLink("Today's special!",
        "DailyDeal",
        null,
        new AjaxOptions
        {
            UpdateTargetId = "dailydeal",
            InsertionMode = InsertionMode.Replace,
            HttpMethod = "GET"
        },
        new { @class = "btn btn-primary" })
</div>
```

- a. Името на акцијата
- b. Специфицира примена на стил за изглед
- c. Текстот на линкот
- d. **Специфицира како да се прати барањето и што ќе се случи со резултатот**

22. Нека е даден модел како на сликата:

```
public class Author
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}

public class Book
{
    public int Id { get; set; }
    public string Title { get; set; }
    public List<Author> Authors { get; set; }
    public double Price { get; set; }
}
```

Дополнете ја дадената акција така што ќе враќа поглед со податоци за книги, податоците да се дефинираат од ваша страна (не од база). Почетниот код е даден:

```
public class BookController : Controller
{
    public ActionResult Index()
    {
        /*
        {
            Id = 1,
            Title = "C# 8.0 and .NET Core 3.0",
            Author = {
                Id = 1,
                FirstName = Mark,
                LastName = J. Price
            },
            Price = 19.50
        }
        */
        // Вашиот код тука!!
    }
}
```

23. Нека е даден следниот модел

```
public class Author
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}

public class Book
{
    public int Id { get; set; }
    public string Title { get; set; }
    public List<Author> Authors { get; set; }
    public double Price { get; set; }
}
```

И нека е даден поглед:

```
@model List<MyModels.Book>

@{
    ViewBag.Title = "User Login";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<div class="container">
    <!-- Вашиот код тука!! -->
</div>
```

пополнете го погледот така што ќе се овозможи приказ на податоците со горенаведениот модел

24. Нека е дадена форма како на сликата:

Add user with this email to a role:

Email:

Admin:

Save

Преку дадената форма, корисник со улога администратор може да доделува различни улоги на останатите корисници, при POST барањето испратено при клик на SAVE копчето од дадената форма се испраќаат податоци репрезентирани со следниот модел:

```

public class AddToRoleModel
{
    [Required]
    public string Email { get; set; }
    public List<string> roles { get; set; }
    public string selectedRole { get; set; }
}

```

Во продолжение е дадена акцијата која го прифаќа POST барањето:

```

[Authorize]
public class AccountController : Controller
{
    private ApplicationSignInManager _signInManager;
    private ApplicationUserManager _userManager;

    [HttpPost]
    [Authorize(Roles = "Admin")]
    public ActionResult AddUserToRole(AddToRoleModel model)
    {
        //Вашиот код тука!!!
        return RedirectToAction("Index", "Clients");
    }
}

```

Довршете ја имплементацијата на акцијата така што ќе се овозможи додавање на корисник во соодветна улога сместена во променливата selectedRole во AddToRoleModel моделот

Решенија на задачите (ne odgovaram za tochnost):

```

22)
public ActionResult Index(){
    Author author = new Author() {Id=1, firstName
    = "Mark", LastName = "Price"};

    List<Book> books = new List<Book>();

    Books.Add(new Book(){id = 1, Title = "test",
    author = author, price = 14.55})

    ...

    Return View(books);
}

```

23)

```

<div class=container>

    <ul>

        @foreach(var item in Model){

            <li>

                @item.Title by @item.Author.FirstName
                @item.Author.LastName

                Price: @item.Price

            </li>

        }

    </ul>

</>

```

24)

```

public ActionResult....(AddToRoleModel model){

    If(ModelState.IsValid){

        var user = db.Users.ToList().Find(u => u.Email ==
        model.Email)

        if (user != null){

            await
            UserManager.AddToRoleAsync(u.Id,
            model.selectedRole);

            return RedirectToAction("Index",
            "Home")

        }

    }

    Return HTTPNotFound(); // ako ne e najden korisnik ili
    ne e validen modelot

}

```