1. **Pentru ce vom implementa Cookies într-o aplicație MVC ASP.net?**
2. Menținerea conexiunilor Client/Server
3. Toate enumerate
4. Controlul sesiunilor
5. Căutarea/Găsirea utilizatorilor
6. **Care este funcționalitatea modelului în ASP.NET MVC?**
7. Obiectele de model preiau și stochează starea modelului într-o bază de date
8. Toate
9. Modelul reprezintă forma datelor
10. Modelul este responsabil pentru menținerea business logic
11. Modelul este responsabil pentru menținerea datelor aplicației
12. **Indicați afirmația corectă:**
13. Controlerul redirecționează cererea de intrare către model
14. Controlerul execute o cerere de intrare
15. Toate
16. Controlerul redă html pentru vizualizare
17. Controlerul este responsabil pentru controlul datelor
18. **Web aplicația dvs. Recent dezvoltată în ASP.NET MVC are o eroare majoră pe controlerul de produse (ProductController). În momentul de față doriți doar să blocați toate paginile de produs pînă la remedierea erorii. Cum ai putea realiza această sarcină?**
19. Add route.SetItem(route.Count, “Product/”); to the RegisterRoute method
20. Nici una din ele
21. Add route.Remove(“Product/”); to the RegisterRoute method
22. Add route.IgnoreRoute(“Product/”); to the RegisterRoute method
23. **La adăugarea unui label în Razor View, care obiect din clasa HTML Helper vom folosi?**
24. @Html.CreateLabel()
25. @Html.DisplayLabel()
26. @Html.Label()
27. @Html.LabelControl()
28. **Care este clasa de bază pentru toate tipurile de rezultate care se întorc din metoda Action?**
29. ActionResult
30. BaseActionResult
31. BaseActionResults
32. ActionResults
33. **La adăugarea unei zone de text în Razor View, care obiect din clasa HTML Helper vom folosi?**
34. @Html.TextArea()
35. @Html.LargeText()
36. @Html.Text()
37. @Html.TextLarge()
38. **La reutilizarea codului partial într-un \_Layout, view se va implementa prin metoda?**
39. @Html.PartialView()
40. @Html.Partial()
41. @Html.InputView()
42. @Html.GetView()
43. **Într-o aplicație ASP.NET MVC, solicitările browser-elor sunt mapate la?**
44. Pages
45. Toate enumerate mai sus
46. Model
47. Controller
48. **Ce este un controller în ASP.NET MVC?**
49. Un coordinator între vizualizare și model
50. Controllerul captează cererile de la utilizator
51. Controllerul face ca vizualizarea corespunzătoare cu datele modelului ca răspuns
52. Toate
53. Este una dintre cele 3 componente ale MVC
54. **Dați definiție noțiunii de *Auto Property*. Oferiți un mod de integrare/utilizare**

Auto Property este utilizata atunci când nu este necesară o logică suplimentară în accesorii de proprietate. El se foloseste la incapsulare, astfel creează o proprietate o dată compilată, există un câmp privat generat în clasă, cu metode get / set care accesează câmpul, si reprezenta o variant prescurtata la:

private string name;

public string Name

{

get

{

return this.name;

}

set

{

this.name = value;

}

}

1. **Dați definiție noțiunii de *Interfață*. Utilitatea acesteia în proiecte ASP.NET**

O interfata defineste un set de metode care vor fi implementate de una sau mai multe clase. O interfata nu implementeaza metode ci doar precizeaza ce anume va contine o clasa care implementeaza interfata. Din punct de vedere sintactic, interfetele sunt similare claselor abstracte. Insa, exista mai multe aspect care le diferentiaza.

O interfata poate contine doar: metode, proprietati, evenimente si indexari. Interfetele nu pot contine variabile, constructori sau destructori. O clasa poate implementa oricate interfete. De asemenea, o interfata poate fi implementata de oricate clase. Asadar, este posibil ca doua clase sa implementeze aceeasi interfata in moduri diferite.

Daca o clasa implementeaza o interfata atunci aceasta trebuie sa implementeze toti membrii interfetei.

public interface IInterfata

{ void Metoda1();

void Metoda2(); }

1. **Enumerați acțiunile (Action) disponibile asupra unui *controller*. Explicați modul lor de utilizare**

Un **Controler** este responsabil pentru lucrul cu modelul date pentru **a efectua acțiuni**, pentru **a prelua date**. În controler sunt definite și grupate un set de acțiuni similare.

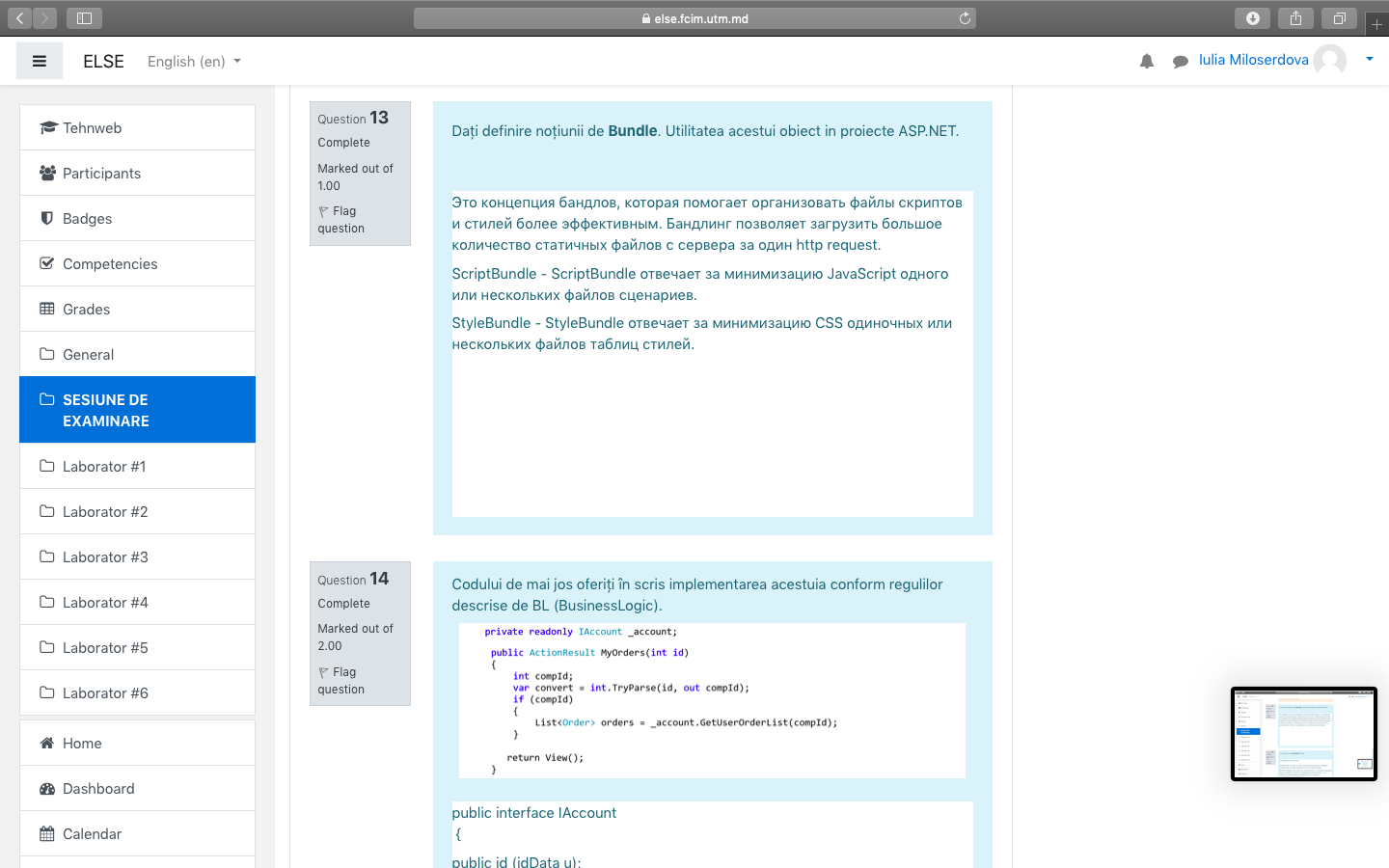
Un Controler are predefinit (default) un Action care este definit în RouteConfig.cs, de obicei aceasta e Index. Un Action poate returna mai multe tipuri de date. Toate tipurile în afară de EmptyResult necesită apelarea metodelor din cadrul Controler-ului de bază, precum ar fi View(), Content(), File(), Json(), Redirect(), JavaScript(), RedirectToRoute(), PartialView().

Un **Controler** este responsabil pentru lucrul cu modelul date pentru **a efectua acțiuni**, pentru **a prelua date**. În controler sunt definite și grupate un set de acțiuni similare.

Un controller expune actiuni, actiunea este in fapt o metoda din cadrul clasei ce defineste controller-ul.

* actiune din controller trebuie sa fie metoda publica in clasa.
* metoda folosita ca actiune nu poate fi supraincarcata si nu poate fi statica.

1. Conform codului de mai jos oferiți în scris implementarea acestuia conform regulilor descriere de BL (Business Logic)



private readonly IAccount \_account;

public ActionResult MyOrders(int id)

{

return View(\_account.GetUserOrderList(id));

}

1. **La ce folosim un ViewModel în MVC?**

Un ViewModel reprezintă datele pe care dorim să le afișăm în pagină, indiferent dacă acestea sunt utilizate pentru text static sau pentru valori de intrare (cum ar fi casetele de text și listele verticale) care pot fi adăugate în baza de date (sau editate).

1. **Dezvoltați o aplicație MVC în care doriți să permiteți accesul anumitui rol la una din metodele din controller. Care atribut sau fragment de cod veți utiliza pentru a atinge acest obiectiv?**
2. [AuthorizeAttribute(Roles=”client”)]
3. RoleProvider.GetRolesForUser(“client”)
4. Nici unul
5. [Authorize(Roles=”client”)]
6. **Ce este un View în ASP.NET MVC?**
7. Afișează datele utilizând modelele
8. Toate
9. Reprezintă interfața utilizatorului
10. Se ocupă de prezentarea datelor
11. Oferă posibilitatea utilizatorului să modifice data
12. **Care este ordinea de execuție a filtrelor de acțiune în serviciile ASP.NET?**
13. Response filters
14. Exception filters
15. Action filters
16. Authorization filters
17. CDBA
18. DCAB
19. DCBA
20. CADB
21. **În ASP.NET, atributul folosit pentru a împiedica falsificarea cererilor între request-uri este?**
22. CrossSiteForgeryToken()
23. RequestForgeryToken()
24. AntiForgeryToken()
25. ValidateAntiForgeryToken()
26. **În MVC ce component decide ce se va afișa în View?**
27. Toate enumerate
28. Model
29. Controller
30. View
31. **Care din următoarele obiecte sunt importante în crearea rutelor (Routes) în MVC?**
32. Controller name
33. Toate
34. Action method name
35. Parameter
36. **Care este descifrarea abreviaturii MVC?**
37. Model, Vision & Control
38. Model, ViewBag & Controller
39. Model, ViewData & Controller
40. Model, View & Controller
41. **Dați definiție noțiunii de *Bundle* . Utilitatea acesteia în proiecte ASP.NET**

Bundle fost introdus în MVC 4 pentru a îmbunătăți timpul de încărcare a cererii. Bundling ne permite să încărcăm o grămadă de fișiere statice de pe server într-o singură cerere http. Ne permite să încărcăm mai multe fișiere JavaScript într-o singură cerere

Bundle (Grupare) – este o caracteristică care îmbunătățește combinarea sau legarea a mai multor fișiere într-un singur fișier.

ScriptBundle - folosirea unui directoriu virtual pentru mai multe fișiere javascript

StyleBundle - folosirea unui directoriu virtual pentru mai multe fișiere de tip css