

Izdelajte spletno aplikacijo s tehnologijo **ASP.NET Core MVC**. Aplikacija naj vsebuje:

- Novo stran za vsako izmed vaših glavnih entitet, kjer izpišete v statični obliki podatke, ki ste jih določili za posamezno entiteto.
- Glavo, nogo, sekcijo za navigacijo (meni) in sekcijo za osrednji (vsebinski) del. Slednji se bo dinamično spreminjal, ostali deli pa ostanejo na vseh podstraneh enaki. Uporabite `MasterPage!`
- Navigacija naj vsebuje povezave do vseh podstrani.
- Glede na vaš projekt smiselno prikažite uporabo jQuery UI accordion
- Vsakič ko naložite spletno stran, naj se na smiselnem mestu izpiše trenutni čas in datum
- Na vseh straneh mora imeti vsebina obojestransko poravnavo
- Vse poravnave, barve pisav, ipd. morajo biti definirane v skupni datoteki `css`.
- Pri oblikovanju si lahko pomagate z UI Frameworki kot je npr. Bootstrap

Za vašo aplikacijo zagotovite tudi odzivno oblikovanje - **Responsive design**

3. Vaja - Spletni obrazec

Rok oddaje: sredo, 17. marec 2021, 23.55

Spletni strani, izdelani v drugi nalogi (2. Vaja - Uvod v MVC) dodajte spletni obrazec (ASP.NET Core MVC) iz 4 korakov (pogledov) **registracije uporabnika**. Obrazec naj vsebuje:

- korak: Vpis imena, priimka, roj. Dne, EMŠO, starost
- korak: Vpis naslova, poštne številke, pošte in države
- korak: Vpis e-naslova in gesla za vpis (geslo vpišemo dvakrat)
- korak: Izpis vseh vpisanih podatkov

Prvi trije koraki imajo gumb "Naprej".

Na strani dodajte povezavo do registracije uporabnika. Smiselno uporabite tudi komponenti jQueryUI Slider in DatePicker.

Implementirajte tudi **modele**, ki jih pri prejšnji nalogi še niste potrebovali.

Pri preusmerjanju upoštevajte vzorec PRG.

4. Vaja - Napredni spletni obrazec

Rok oddaje: sredo, 24. marec 2021, 23.55

Preuredite obrazec iz vaje 3 v strogo tipiziran spletni obrazec (ASP.NET MVC), ki bo vseboval tudi validacijo, in sicer:

1. Vnosno polje za ime (dovoljene so le črke)
2. Vnosno polje za priimek (dovoljeni so le črke)
3. Vnosno polje za datum rojstva
 - Datum mora biti v pravilnem formatu (dd.mm.lllll)
 - Datum mora biti večji od leta 1900 in manjši od današnjega dne
4. Dodajte vnosno polje "kraj rojstva" (padajoči meni z vsaj 5 kraji)
 - Uporabite kolekcijo, ki vsebuje izraz »yield«
5. Vnosno polje za enotno matično številko (EMŠO)
 - EMŠO sestavlja 13 zaporednih števil
6. Vnosno polje za e-pošto
 - Validacija preverja znak @, domeno (npr. domena.com) ter ime pred @. Primer: ime@domena.com
7. Pri vpisu gesel se morata le-ta ujemati. Prav tako mora geslo vsebovati vsaj eno število in vsaj en poseben znak
8. Vsa polja so obvezna in jih je potrebno izpolniti

Celoten projekt je potrebno narediti v pogledu Razor.

Obrazec preuredite tako, da bodo posamezne poglede zamenjali zavihki. Zavihke uredite s pomočjo kontrole Tabs v knjižnici jQuery UI (ali Bootstrap tabs).

Ko izpolnimo obrazec, lahko kliknemo na gumb "Naprej" in izpišejo se nam vpisane vrednosti. Prikaz obrazca in kasneje podatkov, mora biti implementiran z vzorci Post-Redirect-Get.

Validacija ne sme biti implementirana v JavaScriptu (razen kontrola Tabs)!

Ob nepravilni validaciji se izpiše obvestilo ob kontroli!

5. Vaja - Napredna predloga

Rok oddaje: sredo, 31. marec 2021, 23.55

Obrazec iz prejšnje naloge spremenite tako, da boste obrazec za urejanje (torej dodajanje) podatkov izrisovali z metodo »EditorForModel()« in obrazec za izpis podatkov z metodo »DisplayForModel()«. Poleg obrazca iz prejšnje naloge po enakem postopku izdelajte še obrazec za dodajanje in ogled izdelkov (storitev) vašega informacijskega sistema. Za oba obrazca naj veljajo naslednja pravila:

1. Osnovni tipi "Int32", "Double" in "DateTime" imajo lastne predloge (*primer*)! Tipu "Int32" in "Double" se pri urejanju izriše "slider", tipu "DateTime" koledar. Pri ogledu (DisplayForModel) se izpišejo le vrednosti. Predloge osnovnih tipov potem kasneje uporabite v skupni predlogi za celoten model.
2. Prikaz modela lahko poljubno oblikujete.
3. Na prvem koraku boste uredili model. Po kliku na gumb se bo le-ta prikazal (izpisal) na zaslon v predogled.

V primeru težav s tipom Double ("," namesto ".") si pomagajte s tem ali naslednjim virom

Validacijo EMŠA iz prejšnje naloge še dopolnite z anotacijsko validacijo po meri (ValidationAttribute). Pri tem si pomagajte z naslednjim virom. Za samo validacijo EMŠA pa si pomagajte s tem virom (predvsem izračun kontrolne številke).

Pomagajte si s tem virom.

6. Vaja - Povezovanje s podatkovno bazo

Rok oddaje: sreda, 7. april 2021, 23.55

Za shranjevanje podatkov vaših modelov postavite podatkovno bazo (Code First – Entity Framework). Ko vzpostavite podatkovno bazo, potem za modele, ki **ne služijo** obdelavi podatkov omogočite naslednje funkcionalnosti:

- Dodajanje elementov (uporabite že obstoječe predloge)
- Izpis vseh elementov modela (v primeru kjer je gre za modele elementov, ki jih nudi vaša spletna aplikacije, le-te izpišite tudi na predstavitvenih straneh – torej preoblikujete poprej statične strani)
- Izpis poljubnega elementa modela - izbira v seznamu vseh ali podrobnosti na predstavitveni strani (uporabite že obstoječe predloge)
- Urejanje elementov modela (uporabite že obstoječe predloge)
- Brisanje elementov modela

Za urejanje in prikaz elementov modelov (v kolikor še niste pri prejšnjih nalogah) izdelajte Display in Editor predloge.

POMEMBNO

Preden izdelate podatkovno bazo poskrbite da vaš razred "Uporabnik" **ne bo vseboval atributov povezanih z gesli**, čeprav so prejšnje naloge (in tudi ta) od vas zahtevale da vnašate geslo za uporabnika (to še vedno ostaja enako). Sprememba je potrebna zavoljo zmanjšanja kompleksnosti naslednje naloge, ki bo sledila tej -> Avtorizacija.

Da zagotovite vse funkcionalnosti in zadostite zgoraj navedenim navodilom predlagamo naslednje:

- Vašega uporabnika razdelite na dva modela/razreda (npr. Uporabnik in UporabnikZGesli, kjer bo slednji dedoval od razreda Uporabnik in vseboval podatke povezane z gesli)
- Za omenjeni razred prav tako nastavite anotacijo [**NotMapped**], ki bo podatkovnemu kontekstu sporočila, da naprednejšega razreda ne potrebuje hraniti v bazi (torej podatkov o geslih).

8. Vaja - Web API

Rok oddaje: sreda, 12. maj 2021, 23.59

Projekt dopolnite z naslednjimi funkcionalnostmi, in sicer tako, da boste za to uporabili **Web Api**. S pomočjo omenjene tehnologije dodajte naslednje funkcionalnosti:

1. Vašemu projektu dodajte model "nakup/naročnino/rezervacijo"
2. Projektu prav tako dodajte Web API controller, ki bo skrbel za delo z vašim modelom
3. Uporabniku (in administratorju) dodajte možnost, da izdelek kupi/naroči/rezervira, in sicer tako, da ima možnost izdelke najprej dodati v "košarico", nato pa storitev opravi tako, da izpolni še dodatne manjkajoče podatke (ki ste jih definirali v vašem modelu). Storitve nato zaključite tako da kličete spletno storitev, ki ste jo definirali z Web API.
4. Uporabniku (in administratorju) dodajte možnost pregleda preteklih nakupov.

Za normalno delovanje strani uporabite naslednje podatke:

Administrator:

Up. lme: demo@admin.com

Geslo: admin

Uporabnik:

Up. lme: demo@kupec.com

Geslo: kupec

10. Vaja - Testiranje s Selenium-om

Rok oddaje: sreda, 26. maj 2021, 23.59

Po vzoru spodnjega primera testirajte vašo spletno aplikacijo. Stestirajte vsaj 4 funkcionalnosti, ki vključujejo vnos/urejanje podatkov, uporabo gumbov, navigacijo, ipd. Primeri takih funkcionalnosti so:

- Pravilna prijava - stran naloži potrebne elemente, gumbe,... , ki jih imajo samo prijavljene osebe
- Nepravilna prijava - testiranje delovanja prijave, če vrne pravi pogled, izpiše pravilna opozorila,...
- Nakup - dodajanje artiklov v košarico in preverjanje, če se košarica pravilno napolni na zaslonu
- Urejanje uporabnikov - sprememba podatkov in preverjanje ali se funkcionalnost pravilno izvede
- Pravilna registracija - vnos podatkov v vnosna polja in potem preverjanje če je bila registracija uspešna
- Nepravilna registracija - preverjanje pravih odzivov na napačno vnesena polja
- ...

11. Vaja - MVC področja in lokalizacija

Rok oddaje: sreda, 2. junij 2021, 23.59

S pomočjo področij (**Areas**) ločite vašo aplikacijo, na **2 ločeni** področji. Področji naj imata vsaka svoj krmilnik ter vsaj 2 različna pogleda. Eno izmed področij naj bo omogočeno **vsem uporabnikom**, drugo pa **samo administratorjem** (Se pravi ločeno področje za administratorja in ločeno za preostale). Za vsako področje izdelajte svoj Master Page (_Layout.cshtml) in ga tudi različno oblikujte. S pomočjo lokalizacije naredite tudi stran večjezično, tako da bo odvisno od uporabnikove izbire na voljo **angleški** (en) ali **slovenski** jezik (sl-SI).