

Технички опис на веб страната E-Index

Оваа веб страна е креирана во Visual Studio 2022. При креирањето користен е ASP.Net Core Web App(Model-View-Controller) шаблон(template).

Јазици кои се користени во оваа веб страна се:

- JavaScript
- C#
- HTML

Освен програмските јазици, користени се и CSS и Bootstrap за стилизирање на страницата.

Податоците кои се прикажуваат на WEB страницата се сместени во SQL Server Management Studio 19. Во овој SQL сервер се креирани сите табели потребни за WEB страницата, како и процедури кои се користат за управување на тие табели.

Имињата на табелите кои се креирани одговараат со имињата на моделите кои се креирани во *Models* фолдерот во проектот. Исто така и имињата на колоните во SQL одговараат со променливите во моделот. Креирани се табелите:

- Gradovi
- Korisnik
- Predmeti
- Student

Процедури кои се креирани за да ни помагаат со управување на податоците во табелите се:

- DeleteKorisnik
- DeleteStudent
- InsertKorisnik
- InsertStudent
- SelectGradovi
- SelectKorisnik
- SelectKorisnikById
- SelectStudentByIndex
- SelectStudents
- SelectUsers
- SelectUspeh
- UpdateGrade
- UpdateKorisnik
- UpdateKvota
- UpdatePassword
- UpdateProsek
- UpdateStudent

Со стартување на програмата се повикува почетниот *Layout* односно *_LoginLayout.cshtml* во кој имаме форма за најавување и регистрација на корисниците. Прикажувањето на формата која овозможува најавување и регистрација се прави со помош на *get* методот *Index* од контролерот *LoginController*.

Со кликање на копчето за регистрација се префрламе во нов View(со помош на *get* методот во *LoginController,Registration*) каде имаме *input* делови за внес на корисничко име и лозинка. Се прави проверката дали корисничкото име кое е внесено веќе постои во базата на податоци и се испишува порака доколку постои со помош на javascript код, во view *Registration.cshtml*(сместен во папката *Views/Login*). Освен javascript кодот во view-то се, користи и процедурата *SelectUsers* за да се земат сите корисници кои се во базата на податоци. Доколку корисничкото име не е пред тоа употребено се додава новиот корисник во базата на податоци со помош на *post* метод во *LoginController* контролерот. Во овој метод регистрацијата се прави со помош на процедурата *InsertKorisnik* кои прима параметри кои ги внесуваме во view за корисничкото име и лозинка. Бидејќи во процедурата како параметри се бараат и дали корисникот е активен и дали е админ, автоматски се внесуваат од методот *RegistrationUser(IsAdmin=0 и IsActive=1)*.

Кај формата за најава, со внесување на корисничкото име и лозинка се испраќаат податоците до *LoginController* контролерот и со помош на *post* методот *Login*, како и процедурата *SelectKorisnik* се прави проверка дали корисникот со тоа корисничко име и лозинка постои и доколку не постои се испишува порака. Ако постои таков корисник со тоа корисничко име и лозинка се префрламе на ново view, *Index*(од контролерот *HomeController*) каде се прикажува табелата со студенти.

Доколку корисникот кои што се најавил е админ, има дополнителни привелегии за разлика од другите корисници. Привелегиите за админ се доделуваат така што, доколку корисникот кои што е најавен, во базата на податоци за *IsAdmin* има вредност 1.

Табелата на студенти која ни се прикажува може да се сортира по индекс на студентот, име и презиме на студентот, просек на студентот и градот во кој е роден студентот. Сортирањето се прави со помош на кодот што ни е напишан во методот *Index* од *HomeController* контролерот. Исто така и кодот за пребарување по индекс на студентот е напишан во овој метод. Влезните параметри за методот се добиваат од view *Index* кој се наоѓа во папката *Views/Home*.

Во методот *Index* од *HomeController* е напишан и кодот за страничење на табелата. За таа цел се користи и класата *PaginatedList.cs* во која се сместени методите и својства кои ги користиме.

Со кликање на линкот за преглед на предметите на студентот каде сме го кликнуле линкот, се отвара ново view, *ViewSubjects*, каде со помош на методот со истото име од *HomeController* се прикажуваат оценките и предметите за

студентот. Во овој метод во контролерот се употребува и процедурат *SelectUspeh* која треба да прима параметри за индексот на студентот за кои сакаме да ги прикажеме предметите и оценките. Параметарот се добива преку URL со кликање на линкот за преглед на предметите. По завршувањето на прегледот се враќаме на почетната табела со копчето *откажи*.

Доколку сакаме да едитираме некој од студентите кликаме на линкот за уредување и се отвара ново view *Details* кој го користи истоимениот *get* метод од *HomeController*. Во овој view се прикажуваат информациите за студентот каде сме го кликнуле линкот за едитирање, во *input* полиња. Секои од полињата се задолжител. Ова се овозможува со javascript код кој е напишан во view-то. Полињата за индекс, квотата и просекот се оневозможени при прегледот на информациите на студентот. По завршување на прегледот, до целата листа од студенти се враќаме со копчето *откажи*. Доколку сме направиле некакви промени, за зачувување на тие промени се притиска копчето *зачувај*. Со тоа се повикува формата, преку која ќе се повика методот за уредување, чиј код е напишан во *HomeController* контролерот во *post* методот *UpdateDetails*. Во овој метод се повикува и процедурата од SQL базата, *UpdateStudent*, која прима параметри од view за индексот на студентот, името и презимето на студентот, просекот на студентот, квотата на студентот и градот во кој е роден студентот.

За да ги видиме деталите за студентот и да го избришеме, кликаме на линкот за бришење на студент во редот на студентот кои сакаме да го избришеме. Се повикува нов view *IndexDelete* со помош на истоимениот *get* методот од *HomeController* контролерот, кај кои со помош на процедурата *SelectStudentByIndex* и параметарот кои се добива од URL се селектира студентот со тој индекс од табелата на студенти и се прикажуваат информациите за студентот. Доколку сме сигурни дека сакаме да го избришеме студентот кликаме на копчето *избриши* и со помош на формата која е во view-то *IndexDelete* се повикува *post* методот *Delete* за бришење на студентот. Во овој метод и се извршува процедурата *DeleteStudent* кои како параметар добива индекс на студентот кои сакаме да го избришеме од URL. По бришењето на студентот, повторно се враќаме на табелата со сите останати студенти.

Креирањето на нов студент се прави со кликање на копчето *креирај нов студент*. На тој начин се отвара ново view *Create.cshtml*. За да се прикажат информациите во ова view се користи *get* методот од *HomeController* контролерот, *Create*. Во овој метод се извршуваат и процедураите *SelectStudents* и *SelectGradovi*.

Сите *input* полиња се задолжителни во ова view, а просекот мора да биде вредност помеѓу 6 и 10. Тоа го постигнуваме со помош на javascript код во view. Доколку не сакаме да креираме нов студент само кликаме на копчето *откажи*. Кога ќе се внесат сите информации за студентот, се клика на копчето *креирај* и со тоа се повикува формата која ја имаме во *Create* view. Со оваа форма се користи *post* методот *CreateNewStudent* од *HomeController* контролерот и се извршува процедурата *InsertStudent* која се наоѓа во овој метод. Како параметри

од view *Create* оваа процедура ги прима името и презимето на студентот, индексот на студентот, просекот на студентот, градот на раѓање на студентот кој се добива од листа на градови како и во која квота е запишан студентот. По креирањето на студентот се отвара view каде се прикажани сите студенти заедно со новиот студент.

Секој од корисниците може да добие уверение за положени испити со кликање на симболот за pdf документ, во колоната УППИ. За реализација на ова се користи методот *Report* кои пренасочува кон *Report.cshtml* view. Во овој метод се прави колекција од повеќе модели во еден модел, бидејќи во view може да се употребува само еден модел. Пристапувањето до податоците на секој од студентите се прави со помош на процедурата *SelectStudentByIndex* која како аргумент го добива индексот на студентот кои се праќа преку URL. За да пристапиме до оценките на студентот, се употребува процедурата *SelectUspeh* која како аргумент го добива исто така индексот на студентот за да знае за кој студент се работи. Во подолжение на овој метод, се прави конвертирање на кодот кој што ни е во *Report.cshtml* view-то, во pdf формат со помош на *IronPdf* библиотеката. За да се овозможи веднаш да се симнува pdf документ, како резултат од овој метод враќаме *File()*.

Доколку корисникот кој што е најавен е админ (има администраторски привилегии) во *Index* view се прикажува и табела во која се сместени сите предмети заедно со просекот на оценките од секој студент по тој предмет. Тоа е овозможено со помош на методот *Index* од *HomeController* контролерот. Оваа табела може да биде сортирана по името на предметот како и по големина на просекот. Освен оваа табела, кај админ корисникот се прикажува и *Mode* и *Mediana*. Кодот кој го овозможува тоа е сместен во *Index* методот од *HomeController* контролерот.

Во табелата во која се прикажани сите студенти, дополнителни привилегии доколку се најави админ е тоа што може да се има увид и за квотата на студентот, како и за сумата која ја плаќа студентот. Квотата на студентот може да биде променета од страна на админот со помош на листа, чиј код е сместен во *Index* методот од *HomeController* контролерот. Со промена на квотата, автоматски се менува и сумата која ја плаќа студентот.

Бидејќи со ова видовме дека методот *Index* од *HomeController* контролерот треба да пренесе повеќе информации до еден view, а не можеме да употребуваме повеќе од еден метод во еден view. Креираме нов модел *MyViewModels* каде ги сместуваме сите модели кои треба да се користат во *Index* view и со повикување на *MyViewModels* моделот, можеме да ги користиме сите модели во тој модел.

Заглавието на страната е слична и кај обичните корисници и кај админите. Разликата е во тоа што админите имаат привилегии да ја видат и табелата со сите корисници на WEB страната преку заглавието. Со кликање на админ делот од заглавието се отвара нов view *Admin*, кај кој со користење на истоимениот метод од *HomeController* контролерот се прикажуваат во табела сите корисници кои се логирани на WEB страната. Во

овој метод се извршува процедурата *SelectUsers* за прикажување на корисниците.

Админот кои што е најавен може да ги менува своите корисничкото име и лозинка и на сите останати корисници, освен на админите кои се регистрирани на WEB страната. Може да поставува други админи и да го прави корисникот да биде активен или не. При кликање на линкот за уредување на корисниците се отвара ново view *EditKorisnik* каде се прикажуваат информациите на корисникот со помош на *get* методот *EditKorisnik* од контролерот *HomeController*. Доколку направиме некој промени за корисникот, за да ги зачуваме овие промени кликаме на копчето *зачувај* и се повикува формата која го користи *post* методот *UpdateKorisnik* од *HomeController* контролерот. Во овој метод во контролерот се извршува и процедурата *UpdateKorisnik* кои како параметри од *input* деловите од view, пратени преку формата, ги прима корисничкото име на корисникот, лозинката на корисникот, дали е активен и дали е админ. По извршувањето на процедурата се зачувуваат промените кои се направени за корисникот и се пренасочува страната кај view-то кое ги прикажува сите корисници во табела. Ако не сакаме да направиме никаква промена и да се вратиме назад, кликаме на копчето *откажи*.

Доколку админот сака да избрише некој од корисниците, притиска на линкот за бришење во редот на корисникот кој сака да го избрише. Со кликање на линкот се отвара ново view *IndexDeleteKorisnik* каде се сместени податоците кои админот може да ги види пред да го избрише корисникот. Ова view се овозможува од *get* методот *IndexDeleteKorisnik* од контролерот *HomeController*. Во овој метод се извршува и процедурата *SelectKorisnikById* која како параметар го добива индексот на студентот кои сакаме да го избришеме. Индексот се праќа од view, преку URL. Со сигурност ако треба да се избрише корисникот, се клика на копчето *избриши*. Копчето *избриши* ја повикува формата која се поврзува со *post* методот *DeleteKorisnik* од *HomeController* контролерот. Во овој метод се извршува и процедурата која го брише корисникот, *DeleteKorisnik*. Како аргумент оваа процедура го добива индексот исто како и процедурата за прикажување на информациите на корисникот пред да биде избришан.

Во заглавието, исто за сите корисници е тоа што имаме дел кој се користи за промена на јазикот на страната. Со кликање на едно од знамињата, соодветно на знамето се мењава јазикот на страната. Ова е овозможено со помош на локализација. Кодот кој ја прави промената на јазикот е сместен во *_Layout.cshmtl*.

Во горниот десен агол се наоѓа стрелка која кога ќе ја притисмене се појавува листа со дел за промена на лозинката и за одјавување.

Одјавувањето се прави со помош на *get* методот *Logout* од *HomeController* контролерот. Со притискање на копчето за одјава се враќаме на view кое го користиме за најава.

Со кликање на делот за промена на лозинката, WEB страната се пренасочува на view *ChangePassword*. Ова се овозможува со методот *ChangePassword* од

LoginController контролерот. Во овој view имаме *input* елементи за внесување на старата лозинка на корисникот и внесување на новата лозинка. Проверката за тоа дали лозинката што сме ја внеле како стара лозинка се поклопува со онаа лозинка која е во базата на податоци за тој корисник, ја правиме во *LoginController* контролерот, во методот *UpdatePassword*. Доколку старата лозинка е во ред, кога ќе го притиснеме копчето за регистрација, се повикува формата која го содржи *post* методот *UpdatePassword* од *LoginController* контролерот. Во овој метод во контролерот се извршува и процедурата *UpdatePassword*. Како параметри ги добива, корисничкото име каде треба да се направи промената на лозинката и новата лозинка за тој корисник. По завршувањето на промената, се враќаме на view каде се прикажани сите студенти.

Како посебни делови во заглавието се сместени и линковите од каде можеме да ги превземеме фајловите кои ни помагаат да ја разбереме оваа WEB страна и техничкиот дел за изработувањето на страната. Фајлот со техничкиот дел, само админ корисниците можат да го превземат, кај останатите корисници тој дел нема да биде видлив. Со притискање на некој од линковите за превземање се повикуват методите *DownloadFile* и *DownloadSingleFile* кои се сместени во *HomeController* контролерот. Фајловите кои можат да бидат превземени се сместени во *MyFile* фолдерот во *wwwroot*. Освен овој фолдер, во *wwwroot* се наоѓаат и други фолдери во кои може да се сместат *css*, *javascript* фајлови, како и слики кои ќе бидат користени во проектот.

Поврзаноста со базата на податоци, WEB страната ја прави во *appsettings.json* фајлот. Во овој фајл се користи *json* каде поставуваме конекциски *string* на *DefaultConnection*. Како линк за *DefaultConnection* го поставуваме линкот од базата на податоци со која сакаме да се поврземе.