

ISSN 2181-6883

PEDAGOGIK MAHORAT

Ilmiy-nazariy va metodik jurnal

**MAXSUS SON
(2020-yil, dekabr)**

Jurnal 2001-yildan chiqa boshlagan

Buxoro – 2020

Vahob Kamalov. O'qitish usulini tanlash mezonlari, fizika o'quv mavzularida pedagogik axborot texnologiyalaridan foydalanish	118
СОДИҚОВА Ш.М. Физика ўқитиш жараёнини модернизациялашнинг устувор йўналишлари	124
Мухаббат АЖИЕВА, Алишер АЖИЕВ, Гулхан КУВВАТОВА. Значение дидактического материала для закрепления межпредметных связей при изучении естественных наук	129
Mustaqim RAVSHANOV. Magnit maydon qonuniyatlariga doir masalalar yechishda talabalarda ko'nikma shakllantirish.....	132
Gulruxsor ERGASHEVA. Biologiyada 3D texnologiya elementlaridan foydalanish o'quvchilarning kasbiy kompetentsiyasini shakllantirish vositasi sifatida.....	136

MEHNAT TA'LIMI.....141

Сиддик ҚАҲҲОРОВ, Зилола РАСУЛОВА. Талабаларнинг тикувчилик компетентлигини шакллантиришда инновацион педагогик технологияларнинг илмий-методик аҳамияти	141
Шаҳноза Қулиева. Бўлажак ўқитувчиларни тизимли ёндашув асосида тайёрлаш жараёнида муаммолар вазиятлардан фойдаланиш.....	145
Гулнора ТОШЕВА. Бўлажак мутахассисларни дизайнерлик фаолиятига тайёрлашда назария ва амалиёт бирлигини таъминлашнинг назарий асослари.....	148
Улуғбек ИБРАГИМОВ. Технологик жараёнлар ва тизимларни моделлаштириш ихтисослик фанини ўқитишда мавзуга йўналтирилган компьютер дастурларидан фойдаланиш аҳамияти.....	152
Олима Мухидова. Технология фани ўқитувчиларини тайёрлашда компетенциявий ёндашувлар	158
Ma'murjon XO'JJIEV. Oliy ta'limda maxsus fanlarni masofaviy o'qitishda vizual dasturlarning fanlarni o'zlashtirishdagi imkoniyatlari	164
Алимов Абсам. Касбий таълим йўналиши талабаларини инновацион фаолиятга тайёрлаш жараёнини лойиҳалаштириш	171
Наргиза Миржанова. Формирование профессиональной компетентности будущего учителя технологии	177
Олимжон ТЎЙЧИЕВ. Ўқувчиларнинг техник ижодкорлик қобилиятларини ривожлантиришда робототехника элементларидан фойдаланиш	182

AXBOROT TEXNOLOGIYALARI.....186

Озоджон ЖАЛОЛОВ. Верхняя оценка норма функционала погрешности кубатурных формул в пространстве $\bar{L}_2^m(K_n)$	186
Нозимбек ЗАРИПОВ. Дастурлаш муҳитидан фойдаланишнинг асосий тамойиллари	191
Feruz QOSIMOV. Umimiy o'rta ta'lim maktablari informatika kursida "algoritmlar" mavzusini o'rganish bosqichlarining tahlili	194
Hamza Eshonqulov. Bulutli ma'lumotlar omborida qidirishni optimallashtirish usullari	198
Umid Hayitov. O'quvchilar bilimini baholashda ispring quizmaker dasturlarining imkoniyatlari	201
Лола ЯДГАРОВА, Гулсина АТАЕВА. Компьютер технологияларини ёшларга ўргатишда интерактив усуллардан фойдаланиш	205
Gavhar TURDIYEVA, Sohila JUMAYEVA. Bo'lajak boshlang'ich sinf o'qituvchisini tayyorlashda o'qitishning innovatsion usullari va elektron ta'lim tizimlaridan foydalanish.....	210
Ziyomat SHIRINOV. C# dasturlash tilida MVC 4 texnologiyasidan foydalanib web sahifa tayyorlash bosqichlari.	214
Nigora BOTIROVA, Shafoat IMOMOVA. Elektron ta'lim resurslarining tashkil etuvchilari va taqdim etish shakllari	224
Murodova Guli. Obyektga yo'naltirilgan fikrlashni qo'llagan holda maktabda o'quv jarayonining samaradorligini oshirish.....	227
Ҳаким РУСТАМОВ. Информатика ва математика дарсларида дидактик-дастурий кроссвордлардан фойдаланиш.....	230

C# DASTURLASH TILIDA MVC 4 TEXNOLOGIYASIDAN FOYDALANIB WEB SAHIFA TAYYORLASH BOSQICHLARI.

Ziyomat SHIRINOV

Buxoro davlat universiteti
axborot texnologiyalar kafedrası

Maqolada C# dasturlash tilida MVC texnologiyasini qo'llagan holda web sahifa tayyorlash haqida so'z yuritiladi. Bu yerda C# ni endi o'rganayotgan, ma'lumotlar bazasi va web dasturlashdan xabardor boshlagan dasturchilar uchun MVC(Model View Controller) dan foydalanib web sahifa tayyorlashda duch kelishlari mumkin bo'lgan bazi kamchilarga e'tibor qaratilgan.

Tayanch so'zlar: dasturlash, ma'lumot, jadval, ko'rinish(view), model, kontroller, sahifa, obyekt, Entity Framework, C#, loyiha, metod, xususiyat, brauzer, murojaat.

В этой статье описывается, как создать веб-страницу с использованием технологии MVC на языке программирования C#. Для разработчиков, которые уже изучают C# и знакомы с базой данных и веб-программированием, рассмотрим некоторые из недостатков, с которыми они могут столкнуться при разработке веб-страницы с использованием MVC (Model View Controller).

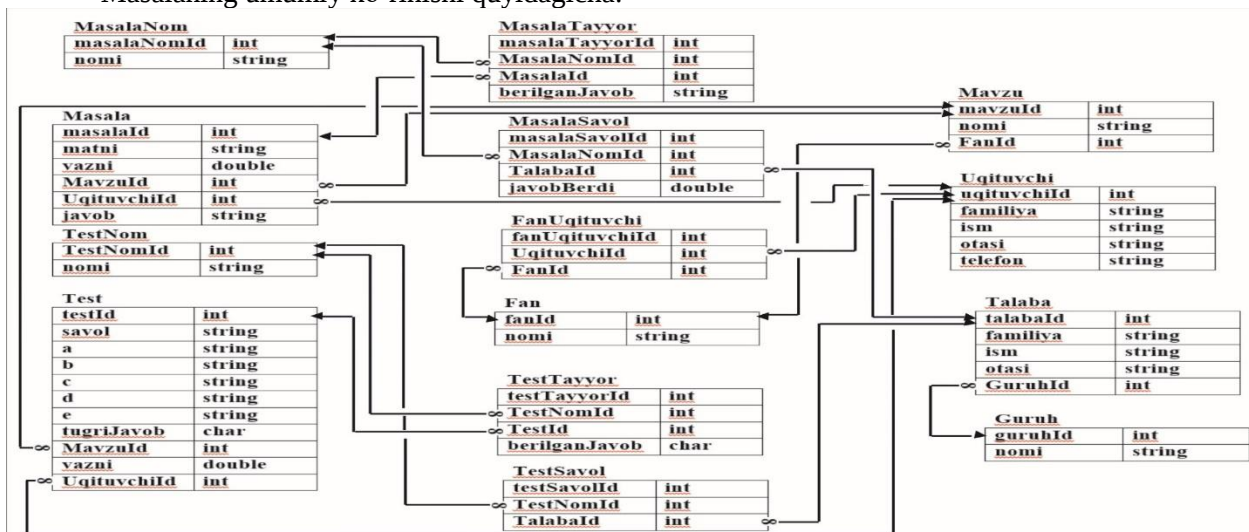
Ключевые слова: программирование, данные, таблица, представление, модель, контроллер, страница, объект, Entity Framework, C #, проект, метод, свойство, браузер, ссылка.

This article describes how to create a web page using MVC technology in the C # programming language. For developers who already learn C # and are familiar with the database and web programming, consider some of the disadvantages that they may encounter when designing a web page using MVC (Model View Controller).

Key words: programming, data, table, view, model, controller, page, object, Entity Framework, C #, project, method, property, browser, reference.

Hozirgi vaqtda barcha korxona va tashkilotlar o'zlarining web sahifalariga ega va yangi tashkil qilinayotganlarida bu bir vazifa hisoblanadi. Web sahifa tayyorlashda asosiy hisoblanuvchi ma'lumotlar doimo o'zgarib boradi. Bu o'zgarishlarni doimiy aks ettirib borishda doimo biror bir dashqi manbada ma'lumotlarni saqlashga to'g'ri keladi. Har bir dasturchi buni o'ziga xos tarzda amalga oshiradi. Hozirda bir qancha zamonaviy texnologiyalar mavjud bo'lsada, bizdagi namunaviy dasturlarimizda C# dasturlash tili va SQL Server ma'lumotlar bazasini boshqarish tizimlarida ishlashga e'tibor qaratilgan. Ushbu maqola ham talabalar bilimni baholash uchun mo'ljallab tuzilishi kerak bo'lgan web sahifaning ma'lumotlarini saqlash jaroyonini qanday amalga oshirish mumkinligini ko'rsatib o'tmoqchimiz.

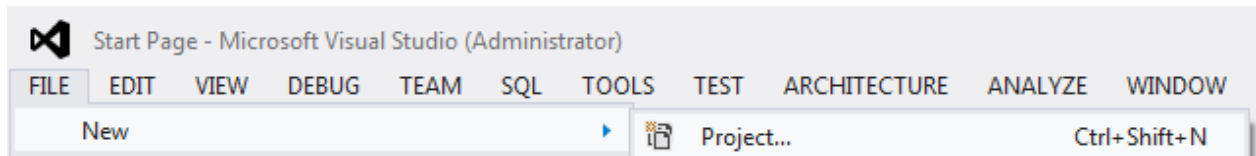
Masalaning umumiy ko'rinishi quyidagicha:



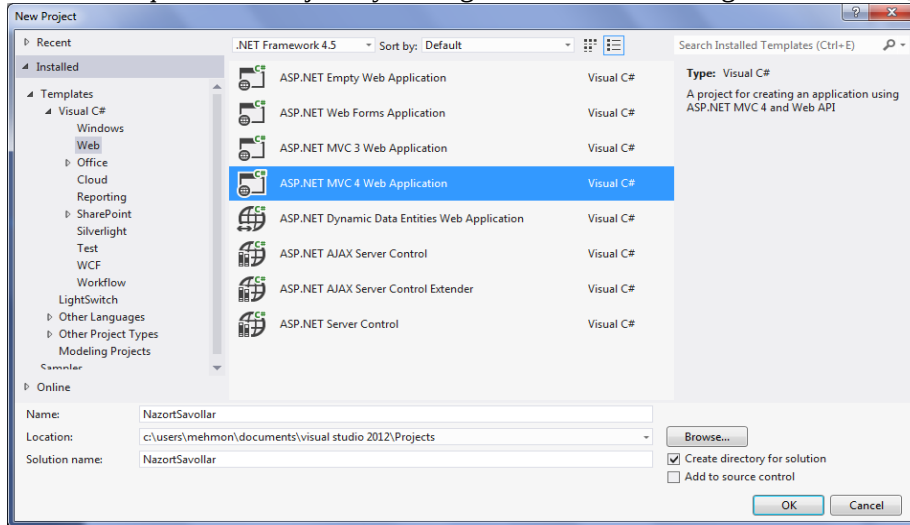
Endi Visual Studio 2012 bu jarayon qanday bo'lishini ko'rib o'tsak.

Visual Studio 2012 da loyiha yaratish quyidagicha amalga oshiriladi:

1. Visual Studio 2012 muhiti ishga tushiriladi.
2. File->New->Project tanlaniladi. [1.B.10]



3. Hosil bo'lgan oynadan **Installed->Visual C#->Web->ASP.NET MVC 4 Web Application** tanlaniladi va **Name** qatorida bo'lajak loyihaning nomi kiritilib **OK** tugmasi bosiladi. [3.B.108]

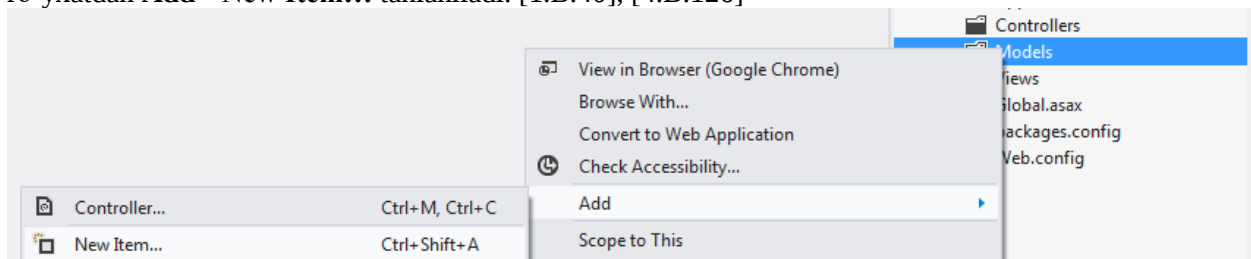


4. Keyingi oynada **Select a template** bo'limidan **Empty** bo'limini tanlab **OK** tugmasi bosiladi. Shundan so'ng bizga yangi bo'sh loyiha yaratiladi. Endi uni o'zi bu loyihani o'zimiz xohlaganimizdek tayyorlab olishimiz mumkin.

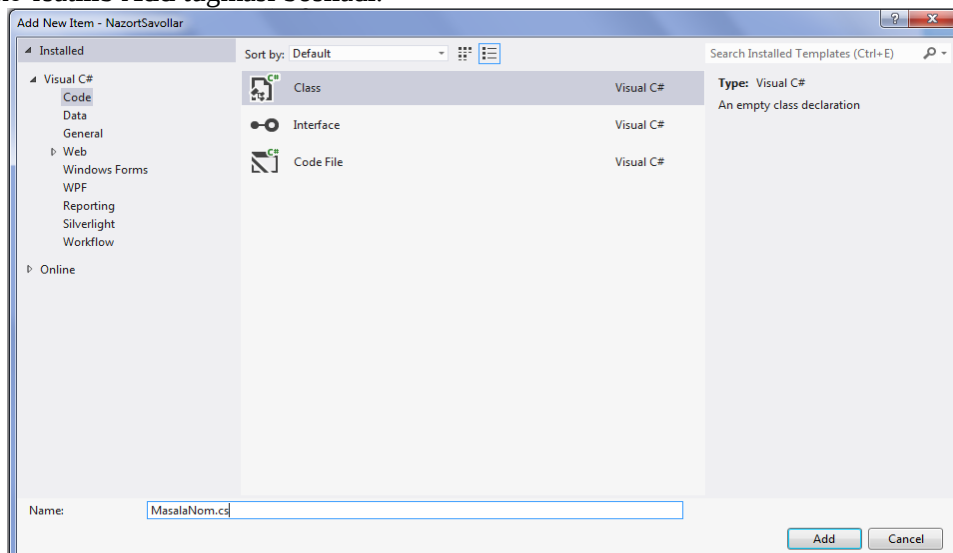
MVC texnologiyasida birinchi navbatda loyihaning modeli tayyorlanadi. Bu jarayon MasalaNom va MasalaTayyor jadvallari uchun ko'rib chiqaylik.

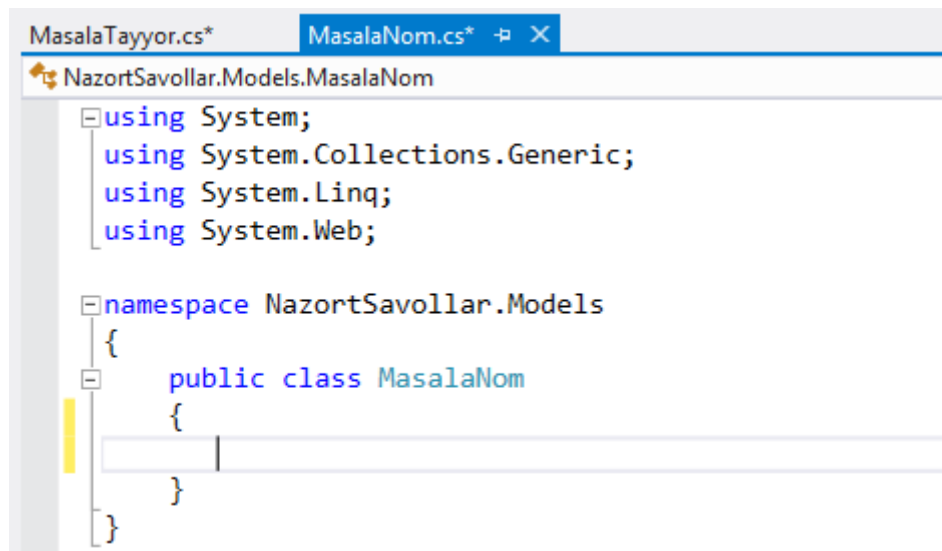
MasalaNom jadvalidagi ma'lumotlarni saqlash uchun **model** yozish quyidagi tartibda amalga oshiriladi.

Solution Explorer bo'limidan **Models** papkasi ustida sichqonchani o'ng tugmasi bosiladi va ro'yxatdan **Add->New Item...** tanlaniladi. [1.B.40], [4.B.126]



Hosil bo'lgan oynadan esa **Installed->Visual C#->Code** dan **Class** tanlaniladi va **Name** qatorida **model** nomi ko'rsatilib **Add** tugmasi bosiladi.





Shundan so'ng bizga yangi MasalaNom nomli sinf hosilqilinadi va biz yuqoridagi jadval maydonlarni ifodalovchi xususiyatlarni kiritib olamiz

```

public class MasalaNom{
    public int masalaNomId { get; set; }
    public string nomi { get; set; }
    public virtual ICollection<MasalaTayyor> MasalaTayyorTuplam { get; set; }
}

```

Yuqorida jadvalning ikkitamaydoni uchun uchun ikkita xususiyat va bitta jadvallar orasidagi bog'lanishni hosil qilish uchun maxsus to'plam xususiyati hosil qilingan. [1.B.42]

Yuqoridagi ketma-ketlikni MasalaTayyor jadvali uchun ham bajarsak, uning sinfi quyidagicha bo'ladi:

```

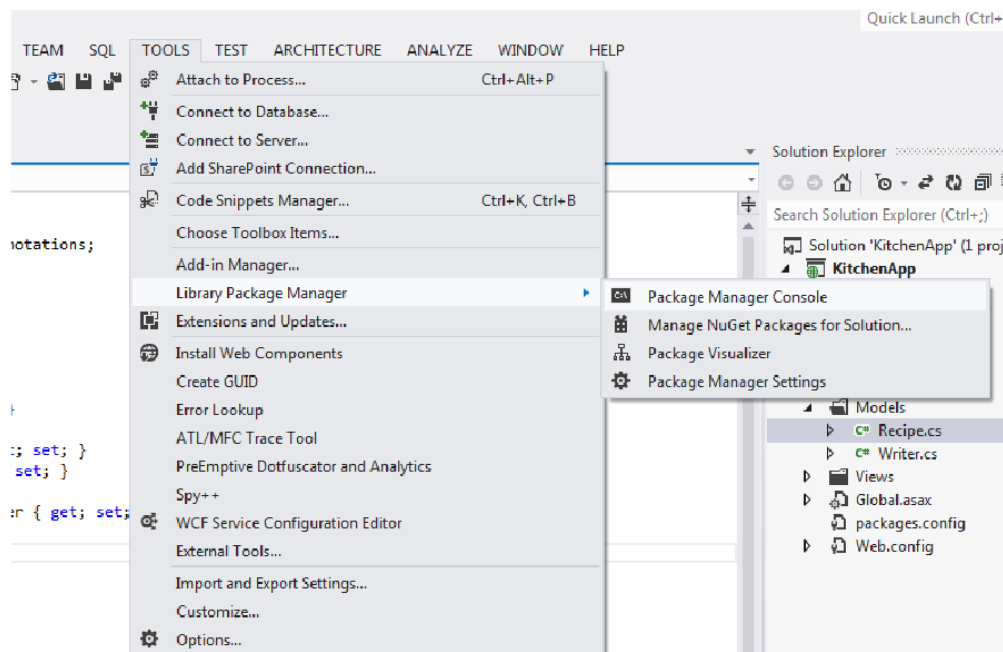
public class MasalaTayyor{
    public int masalaTayyorId { get; set; }
    public int MasalaNomId { get; set; }
    public MasalaNom MasalaNomJoriy { get; set; }
    public int MasalaId { get; set; }
    public string berilganJavob { get; set; }
}

```

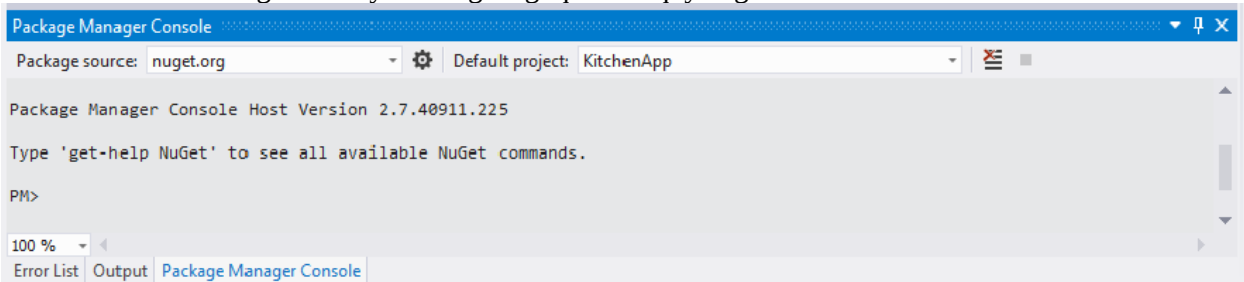
Endi loyihamizda **EntityFramework**dan foydalanish uchun quyidagi ketma-ketlikni bajaramiz:

Loyihamizga Entity Framework 6.2.0.0 ni bog'lab qo'yish uchun menyular qatoridanb **Tools ->**

Library Package Manager tanlaniladi va **Package Manager Console** ustida sichqonchani chap tugmasi bosiladi.



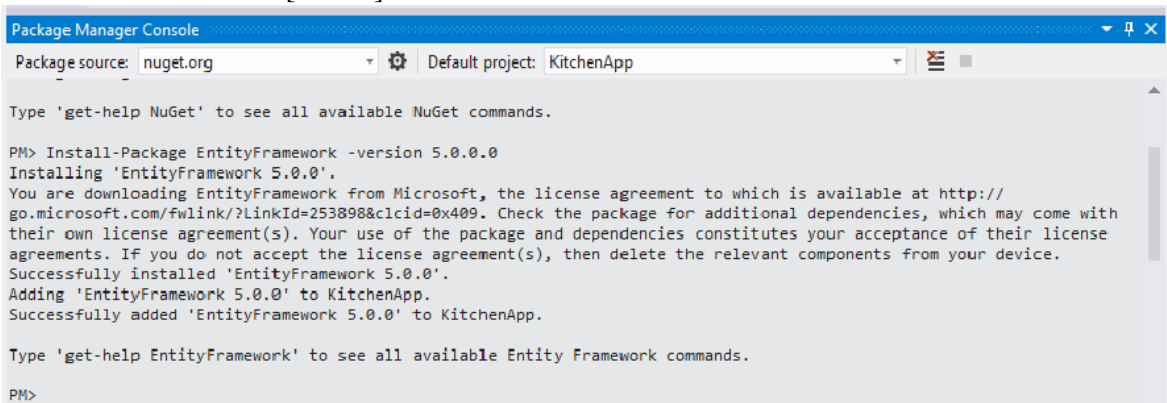
Shundan so'ng muhit oynasining ostgi qismida quyidagi ko'rinish hosil bo'ladi.



Endi Entity Framework 6.2 o'rnatish uchun quyidagi buyruqni yozishingiz kerak

PM> Install-Package EntityFramework -version 6.2.0.0

Buyruq yozilgach klaviaturadan enter tugasi bosiladi. Yana bir muhim ma'lumot Entity Framework o'rnatish jarayonida kompyuteringiz internetga ulangan bo'lishi lozim. Bir necha sekundlardan so'ng siz Visual Studio 2012 dagi loyihangizga EntityFramework 6.2 yuklanib o'rnatilganligi haqidagi tasdiqlash ma'lumotlarni ko'rasiz. [2.B.42]

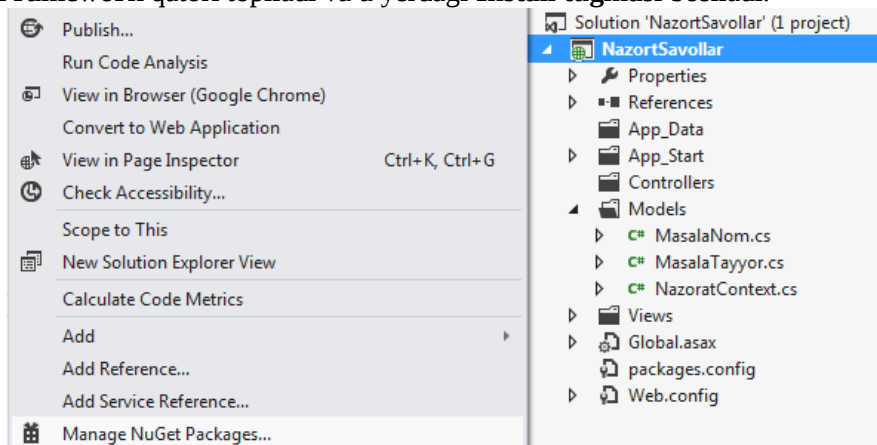


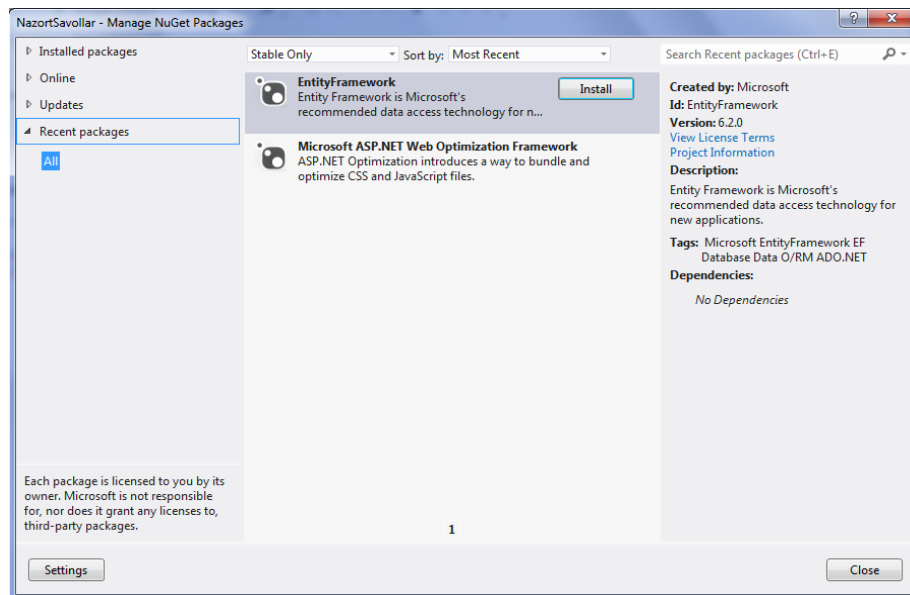
U web.config fayliga biro z o'zgartirish kiritadi, uni solution explorer oynasidan topishingiz mumkin. [2.B.47]

```
<entityFramework>
  <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
</entityFramework>
</configuration>
```

Agar bizda oldin biror loyihamizga **EntityFramework** yuklangan bo'lsa u holda bizga internet mavjud bo'lmaganda ham **EntityFramework** qo'shish imkonini bo'ladi bu esa quyidagicha bajariladi:

Solution Explorer bo'limidan loyiha nomi ustida sichqonchani o'ng tugmasi bosiladi va ro'yxatdan **Manage NuGet Packages...** tanlaniladi va hosil bo'lgan oynadan Recent packages... bo'limi talanilib **EntityFramework** qatori topiladi va u yerdagi **Install** tugmasi bosiladi.

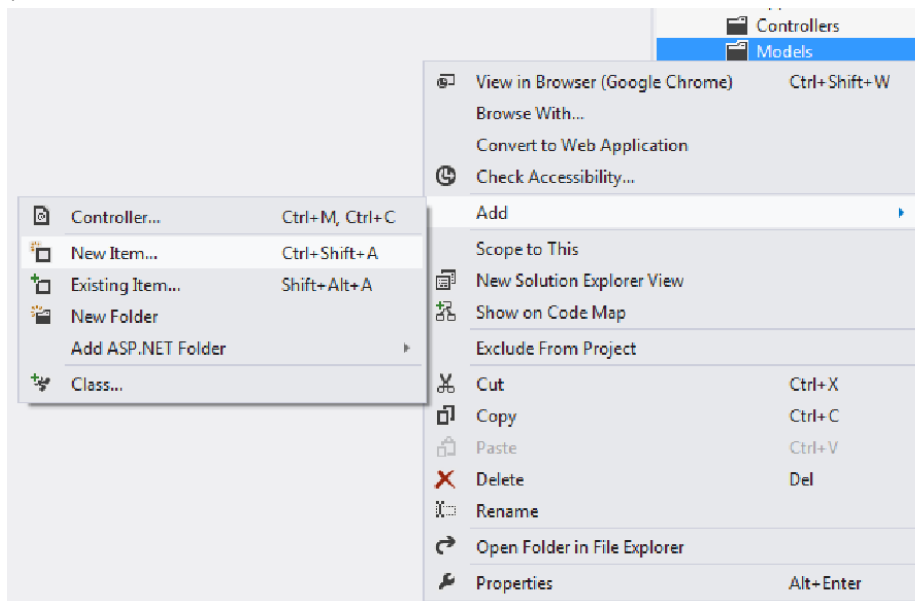




Shundan so'ng biz DbContext sinfiga merosxo'r qilib bitta sinf yaratamiz.

Sinfni qo'shish tartibi

Models papkasi ustida sichqonchani o'ng tugmasini bosamiz, u solution explorer oynasida mavjud, **Add ->New Item...** bosamiz va hosil bo'lgan muloqot oynasida sinfimiz nomini **NazoratContext** deb nomlaymiz.



Sinfni yaratgach nomlar fazosi deb ataluvchi qatorlar davomida quyidagi qatorni qo'shib qo'yamiz:
using System.Data.Entity;

Keyin sinfning tuzulishini quyidagicha shakllantiramiz.

```
public class NazoratContext : DbContext{
    public DbSet<MasalaNom> MasalaNomTuplam { get; set; }
    public DbSet<MasalaTayyor> MasalaTayyorTuplam { get; set; }
}
```

Yuqoridagi amallarni bajargandan so'ng solution explorer dan web.config file faylini oching connection string nomli qatorga ConfigSections tag(teg) ni qo'shib qo'ying. [1.B.43] Tegni configSection tegi yakunlangandan so'ng qo'shish zarur. Bu tegning ko'rinishi quyidagicha:

```
<connectionStrings>
<add name="NazoratContext" connectionString="Data Source=DOTNET-PC\MSSQLSERVER2K8;Initial
Catalog=Nazorat; User ID=sa; Password=1234" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

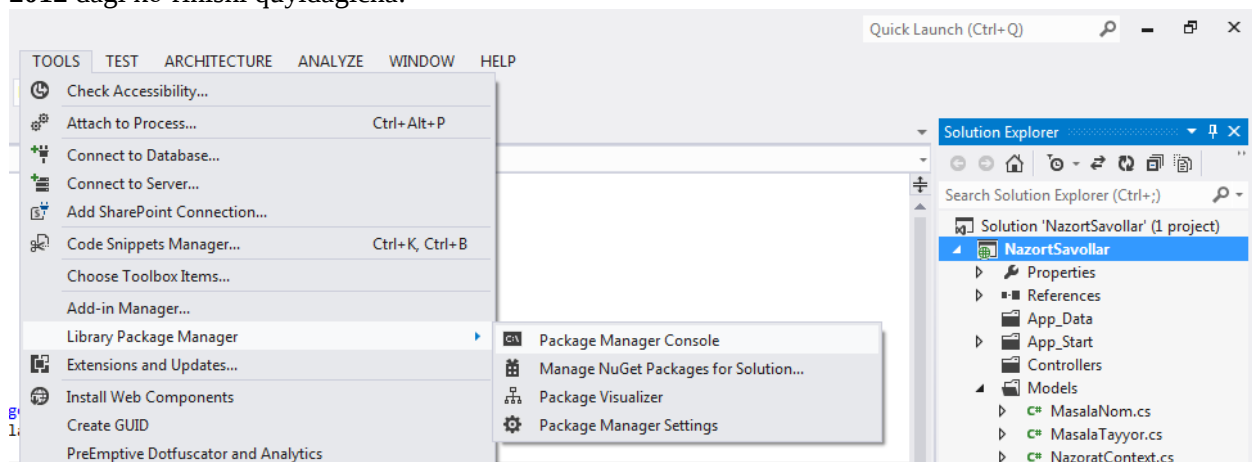
Siz yuqorida berilgan qatorlardan Data Source ning qiymatini o'zingizning kompyuteringizning nomiga moslaysiz. SQL server ochib u yerdan ID=sa va password ni 1234 Sizning kompyuteringizdagi bilan bir xil qilib qo'yishingiz kerak. Ma'lumotlarni kiritib bo'lgach web.config quyidagi ko'rinishda bo'ladi:

```
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit http://go.microsoft.com/fwlink/?LinkID=237468 -->
    <section name="entityFramework" type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=5.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
  </configSections>
  <connectionStrings>
    <add name="KitchenContext" connectionString="Data Source=DOTNET-PC\MSQLSERVER2K8;Initial Catalog=KitchenDB;User ID=sa;Password=1234" providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <appSettings>
    <add key="webpages:Version" value="2.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="PreserveLoginUrl" value="true" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
  </appSettings>
  <system.web>...</system.web>
  <system.webServer>...</system.webServer>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory, EntityFramework" />
  </entityFramework>
</configuration>
```

Yuqoridagilarni bajarib bo‘lgach ma’lumotlar bazasini yaratish so‘rovini bajaramiz. [2.B.82]

Ma’lumotlar bazasini bog‘lanishlar bilan yaratish.

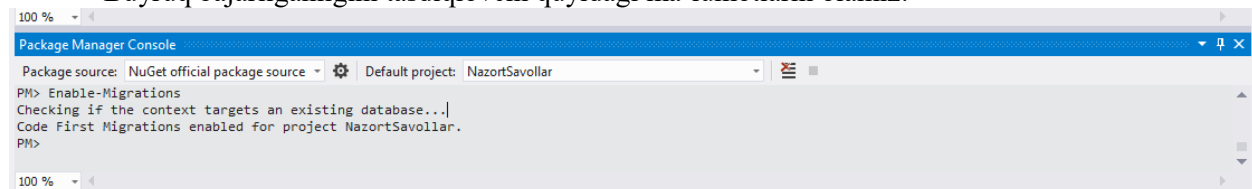
Jarayonni bajarish uchun menyular qatoridan **Tools -> Library Package Manager** va ro‘yxatdan **Package Manager Console** bo‘limi ustida sichqonchani chap tugmasi bosiladi. Uning **Visual Studio 2012** dagi ko‘rinishi quyidagicha:



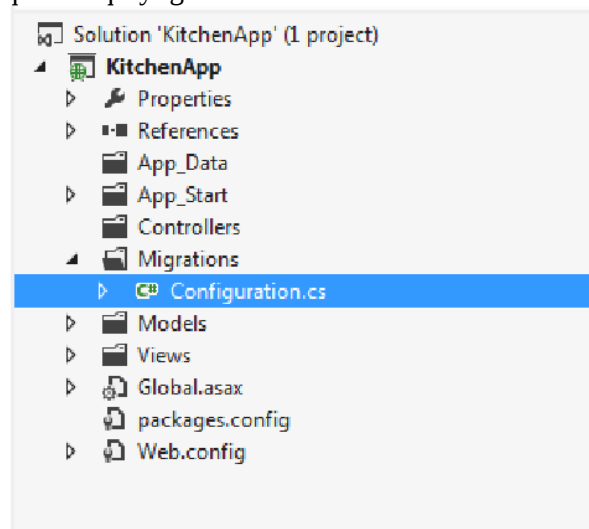
Buyruq qatorini quyidagicha yozib enter tugmasini bosamiz:

PM> Enable-Migrations

Buyruq bajarilganligini tasdiqlovchi quyidagi ma’lumotlarni olamiz:



Bu jarayon bajarilgach solution explorer yana bir Migrations nomli papka hosil bo‘ladi va unda **Configurations.cs** nomli fayl qo‘shib qo‘yilgan bo‘ladi.



Configurations.cs faylining ko‘rinishi quyidagicha bo‘ladi:

```
namespace NazortSavollar.Migrations
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Migrations;
    using System.Linq;

    internal sealed class Configuration : DbMigrationsConfiguration<NazortSavollar.Models.NazoratContext>
    {
        public Configuration()
        {
            AutomaticMigrationsEnabled = false;
        }

        protected override void Seed(NazortSavollar.Models.NazoratContext context)
        {
            // This method will be called after migrating to the latest version.

            // You can use the DbSet<T>.AddOrUpdate() helper extension method
            // to avoid creating duplicate seed data.
        }
    }
}
```

Keying qadamda siz Configuration metodi ichidagi AutomaticMigrationsEnabled ning qiymatini true ga o‘zgartirib qo‘yishingiz talab qilinadi. Bu esa keyinchalik sizga bazada qo‘shimchalar qilish imkonini beradi.

```
public Configuration(){
AutomaticMigrationsEnabled = true;
}
```

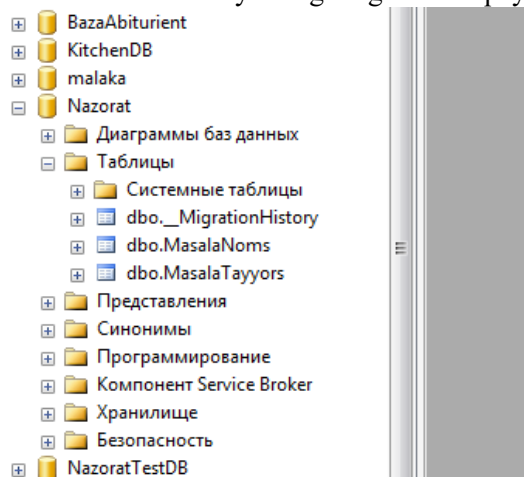
Ma’lumotlar bazasini yaratish uchun quyidagi buyruqni gajarishingiz kerak:

PM> Update-Database

```
Package Manager Console
Package source: NuGet official package source
Default project: NazortSavollar

PM> Update-Database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
No pending explicit migrations.
Applying automatic migration: 202002160700215_AutomaticMigration.
Running Seed method.
PM>
```

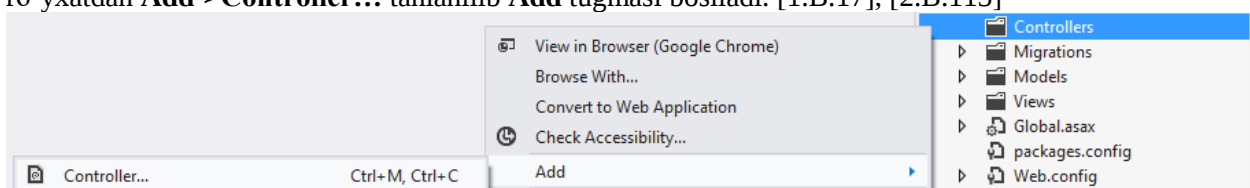
Bu sizga SQL server da ma’lumotlar bazasi yaratilganligini tasdiqlaydi:

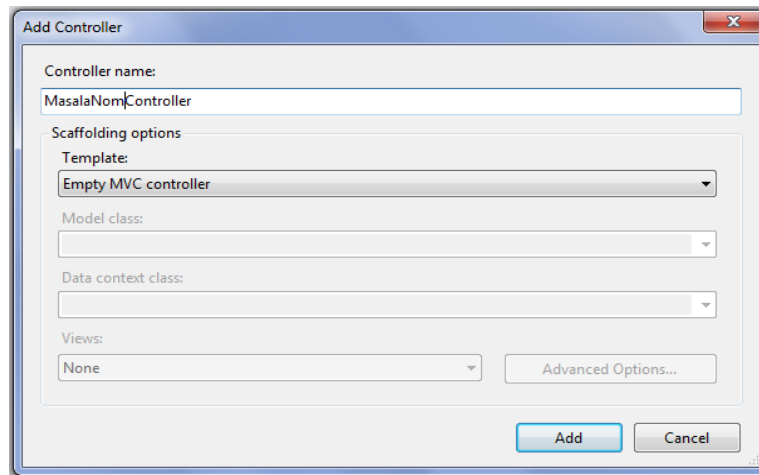


Kontrollerlar(Controllers) va ko‘rinishlar yaratish(Views)

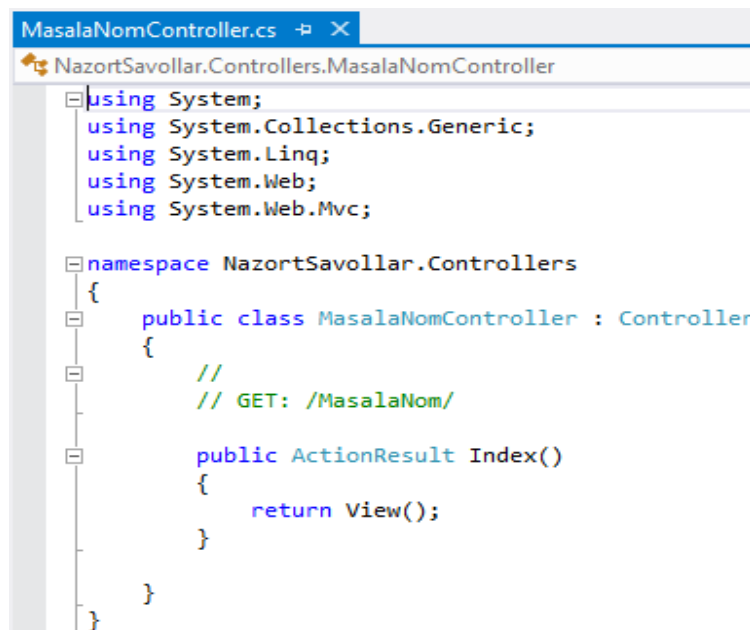
MasalaNom jadvaliga ma’lumot kiritish va unda bor ma’lumotlarni chiqarish uchun kontrollerlar yaratish yaratish quyidagicha bo‘ladi:

Solution Explorer bo‘limidan **Controllers** papkasi ustida sichqonchani o‘ng tugmasi bosiladi va ro‘yxatdan **Add->Controller...** tanlanilib **Add** tugmasi bosiladi. [1.B.17], [2.B.113]



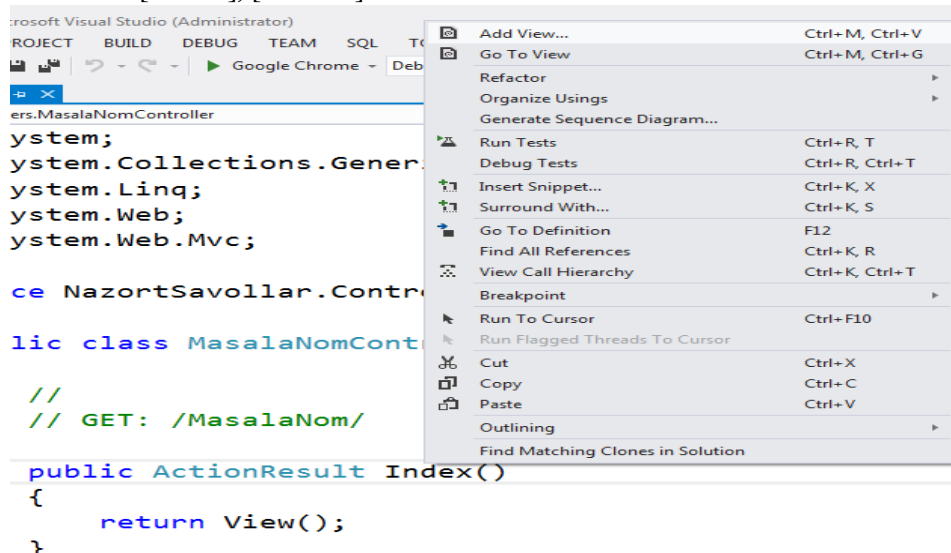


Kontroller yaratish oynasi ochilganda u yerdagi kontrollerga nom berish qatorida nom kiritilayotgan vaqtda **Controller** soʻzini oxirida saqlab qolish kerak. Bu kontroller yaratishdagi oldindan kelishilgan qoida hisoblanadi.



Index metodi barcha kontrollerlar uchun bosh sahifani ifodalovchi sahifaga oʻtishni taminlab beradi. Bu metodda MasalaNom jadvalidagi maʼlumotlarni chiqaruvchi sifatida aniqlab uning sahifasi uchun (koʻrinish)view yarataylik. [4.B.134]

Buning uchun **Index** metodi nomi ustida sichqonchani oʻng tugmasi bosiladi va roʻyxatdan **Add View...** bandi tanlanadi. [1.B.23], [3.B.116]



Ko‘rinish qo‘shish oynasi **Create a strongly-typed view** bandiga belgi qo‘yamiz, **Model class:** bandidagi ro‘yxatdan MasalaNom (NazortSavollar.Models) ni tanlaymiz va Scaffold template: **List** bandini tanlab **Add** tugmasini bosamiz.

Add View

View name: Index

View engine: Razor (CSHTML)

☒ Create a strongly-typed view

Model class: MasalaNom (NazortSavollar.Models)

Scaffold template: List ☒ Reference script libraries

☐ Create as a partial view

☒ Use a layout or master page:

(Leave empty if it is set in a Razor _viewstart file)

ContentPlaceHolder ID: MainContent

Add Cancel

```
Index.cshtml  X MasalaNomController.cs
@model IEnumerable<NazortSavollar.Models.MasalaNom>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

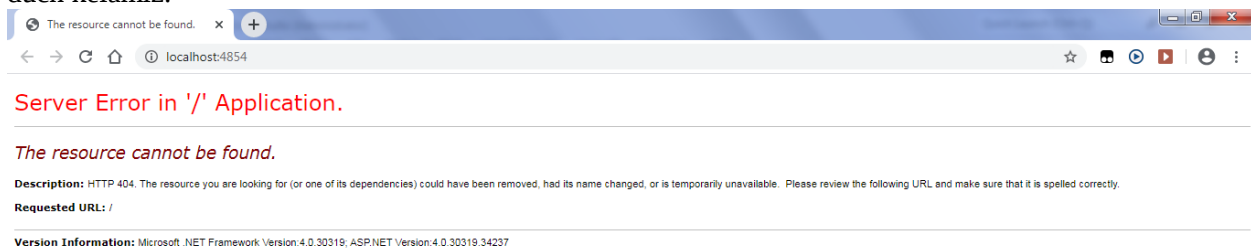
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table>
<tr>
<th>
        @Html.DisplayNameFor(model => model.nomi)
</th>
<th></th>
</tr>
@foreach (var item in Model) {
<tr>
<td>
        @Html.DisplayFor(modelItem => item.nomi)
</td>
<td>
            @Html.ActionLink("Edit", "Edit", new { id=item.masalaNomId }) |
            @Html.ActionLink("Details", "Details", new { id=item.masalaNomId }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.masalaNomId })
</td>
</tr>
}
</table>
```

2020. MS. PEDAGOGIK MAHORAT*ПЕДАГОГИЧЕСКОЕ МАСТЕРСТВО*PEDAGOGICAL SKILL

Yuqoridagi rasmda **Index.cshtml** faylining ko‘rinishi keltirilgan va bu yerda foreach qatoriga e’tibor beradigan bo‘lsak u yerda **Model** toplamining elementlariga murojaat mavjud. Shu sababdan biz yaratgan kontrollerdan uning ko‘rinishi(**view**)ga modelni argument sifatida yuborishimiz kerak. **Index** metodi ko‘rinishi quyidagicha bo‘ladi:

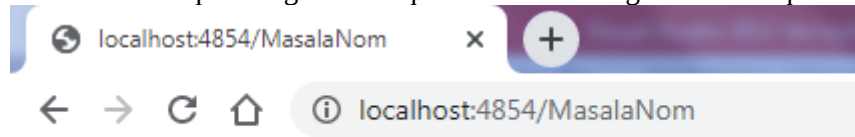
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using NazortSavollar.Models;
namespace NazortSavollar.Controllers{
    public class MasalaNomController : Controller{
        NazoratContext ns = new NazoratContext();
        public ActionResult Index(){
            return View(ns.MasalaNomTuplam.ToList());
        }
    }
}
```

Dasturni ishga tushirish uchun menyular qatorida **Debug->Start Debugging** tanlaniladi. **Visual Studio IIS Express** deb nomlanuvchi serverni ishga tushuradi va avtomatik brauzerlardan biri ishga tushib dasturni belgilangan manzil bo‘yicha yuklashga harakat qiladi. Bunda u birinchi navbatda kontroller yozishdagi kelishuvga asosan **HomeController** deb nomlanuvchi kontrollerning **Index** nomli metodini ishga tushurishga harakat qiladi. Biz bu kontrollerni yaratmaganimiz uchun quyidagi rasmda keltirilgan xatolikka duch kelamiz.



Yuqoridagi rasmda keltirilgan brauzerning manzillar qatoridagi localhost:4854 qatoriga / belgisini qo‘shib undan so‘ng kantrollerimizning nomini yozib klaviaturadan enter tugmasini bosamiz.

Eslatma: localhost:4854 qatoridagi 4854 raqamlar ketma-ketligi sizda boshqacha bo‘lishi mumkin.



Index

[Create New](#)

nomi

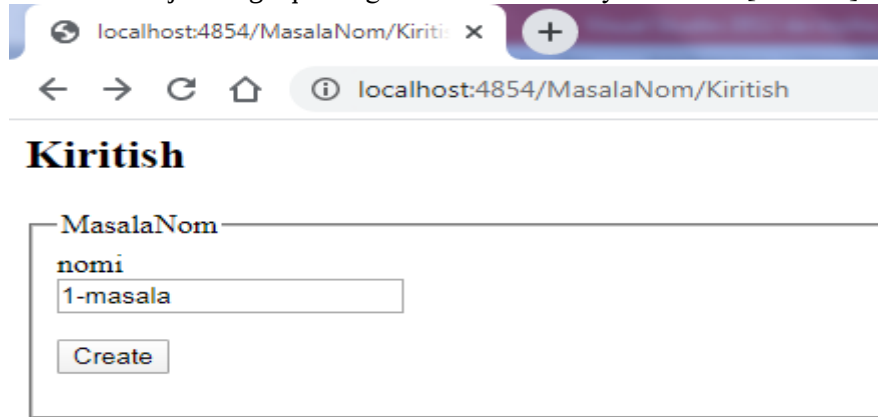
Biz bu jadvalda hechqanday ma’lumot saqlamaganligimiz uchun sahifada jadval ustunining nomidan boshqa hech narsa mavjud emas.

Endi bu jadvalimizga ma’lumot yozish uchun controller yozaylik. MasalaNomController sinfida yangi Kiritish, QushishMasalaNom va Xato nomli metodlarini yaratamiz.

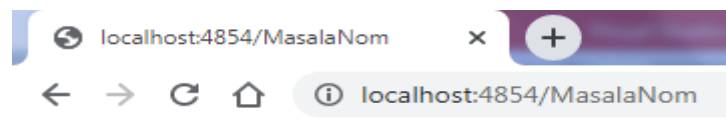
```
public ActionResult Kiritish() {
    return View();
}
[HttpPost]
public ActionResult QushishMasalaNom(MasalaNom model){
    if (ModelState.IsValid){
        ns.MasalaNomTuplam.Add(model);
        ns.SaveChanges();
        return RedirectToAction("Index");
    }
}
```

```
return RedirectToAction("Xato");  
}  
public ActionResult Xato(){  
    return View();  
}
```

Yaratilgan sahifalarni birma-bir ko‘rish uchun **Index** nomli sahifani ishga tushuramiz keyin u yerdan **Create New** bandini tanlaymiz. Kiritish sahifasi ochilgach u yerdagi maydonni to‘ldirib **Create** tugmasini bosamiz va bizga MasalaNom jadvaliga qo‘shilgan ma’lumot namoyish etiladi. [3.B.120]



[Back to List](#)



Index

[Create New](#)

nomi

1-masala [Edit](#) | [Details](#) | [Delete](#)

Adabiyotlar

1. Rick Anderson and Scott Hanselman Intro to ASP.NET MVC 4 with Visual Studio (Beta) 2012. 115pages.
2. Akhil Mittal Diving into Microsoft. NET Entity Framework 2017. 217pages.
3. Brian Driscoll, Nitin Gupta, Robert Vettor, Zeeshan and Larry Tenny Entity Framework 6 Recipes. 536pages.
4. Герберт Шилдт Полный справочник по C# 2004. 744стр.

ELEKTRON TA'LIM RESURLARINING TASHKIL ETUVCHILARI VA TAQDIM ETISH SHAKLLARI

Nigora BOTIROVA

Buxoro viloyat xalq ta'limi xodimlarini
qayta tayorlash va ularning malakasini oshirish instituti o'qituvchisi

Shafolat IMOMOVA

Buxoro davlat universiteti axborot texnologiyalari kafedrası
katta o'qituvchisi

Elektron o'quv resurslarini yaratish jarayonida psixologik-pedagogik, texnik-texnologik, estetik va ergonomik talablar qo'yiladi. Elektron darslik, Elektron ma'lumotnoma va uslubiy qo'llanma kabi dasturiy