Spring 2015

Recipe Suggestion Tool

Tejaswi Patha
Governors State University

Deepika Sandhadi
Governors State University

Recommended Citation

Patha, Tejaswi and Sandhadi, Deepika, "Recipe Suggestion Tool" (2015). *All Capstone Projects*. 98.
http://opus.govst.edu/capstones/98

ABSTRACT

There is currently a great need for a tool to search cooking recipes based on ingredients, country and recipetype. Current search engines do not provide this feature. Most of the recipe search results in current websites are not efficiently clustered based on relevance or categories resulting in a user getting lost in the huge search results presented. They also do not provide links to view images of the ingredients of a recipe.

My project aims to combine the features like search based on ingredients, suggestions for similar recipes, and images for the ingredients under one search engine and provide an intuitive interface for the same. I explored different clustering algorithms to find an efficient algorithm that can be used to cluster recipe data matching user's queries. As part of this project, I also built FreeText search it help users can search Recipes by ingredients, country and recipe type. I created few charts for users to understand which ingredients are used more in recipes and which country ingredients are more. This website also provides articles to users for making tasty recipes. In this article page users can comment and rate the article. Our website is deployed to Microsoft azure platform.

Table of Contents

1.0 INTRODUCTION

In this project, we worked on a recipe suggestion tool, which suggests similar recipes. Users can also search Recipes based on ingredients, Country and RecipeType from Database and provide recipes list by order. We also provide the facility for the user to view the images of the recipes. We use Entity framework to retrieve the data from SQL server database. For keyword search we used SQL server FreeTextTable function and ranked the results based on the relevance.

We used ASP.net identity provider for login component and registration component. Only logged in users can view the dashboard page, comment on articles and rate articles.

We used Microsoft chart control to create charts for ingredients vs. count and country vs. count. I wrote few stored procedures to feed the data to charts. We also provided articles to the users by stored procedures. Users can read an article, rate the articles and comment on article if the user is logged in.

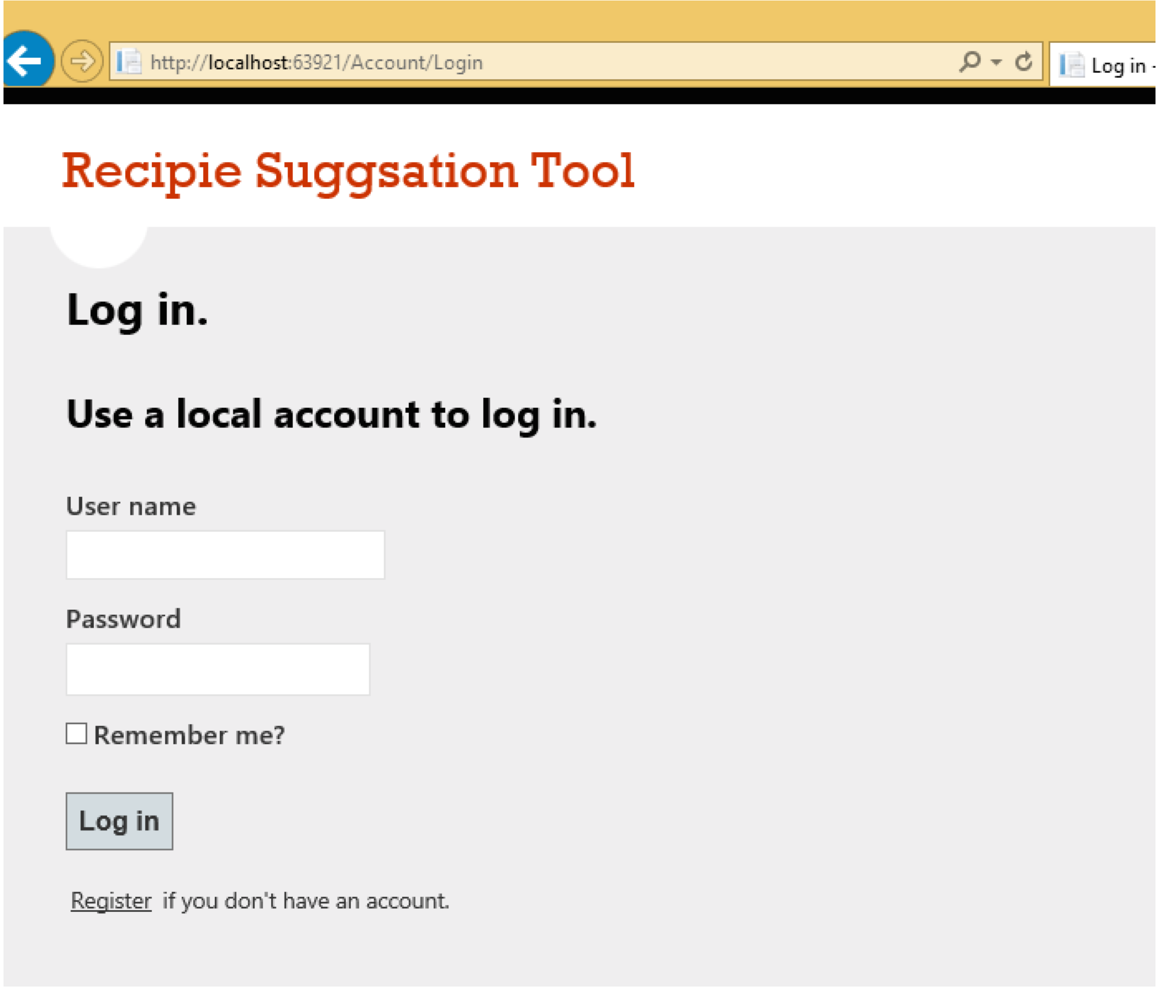
FreeTextTable Search helps users to access information easily. One only needs to get connected to the Internet to get the information one needs. When searching for cooking recipes, sometimes user may prefer to search based on ingredients, country and RecipeType.

We have deployed the website and SQL database on windows azure. We have created a Virtual machine in widows azure installed IIS, .NET framework 4.5, and set up the IIS to host the web site in the virtual machine.

2.0 USER INTERFACE

2.1 Login Page:

The Login Screen will remain in the right column of the web page.
The log in screen will have an option to enter username and password for existing users.



Recipie Suggsation Tool

Log in.

Use a local account to log in.

User name

Password

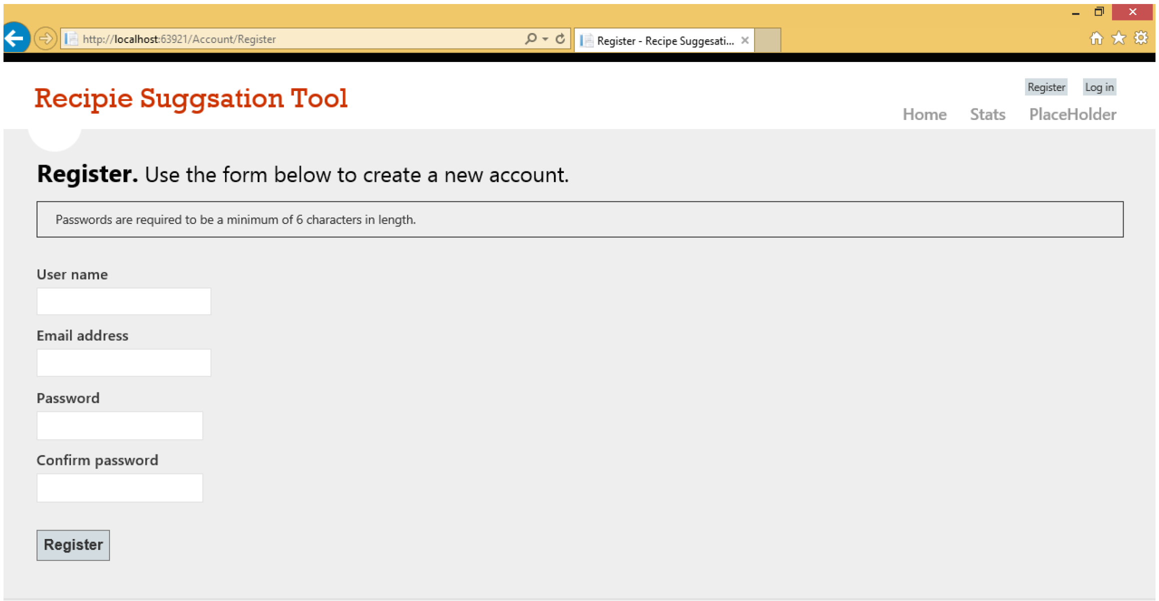
□ Remember me?

Register if you don't have an account.

2.2 Register Page:

Register screen will allow users to create new account.
The account information section consists of fields as shown below, which are

required to create new account.


http://localhost:63921/Account/Register
Register - Recipe Suggesati...
Recipie Suggsation Tool
Register
Log in
Home
Stats
PlaceHolder

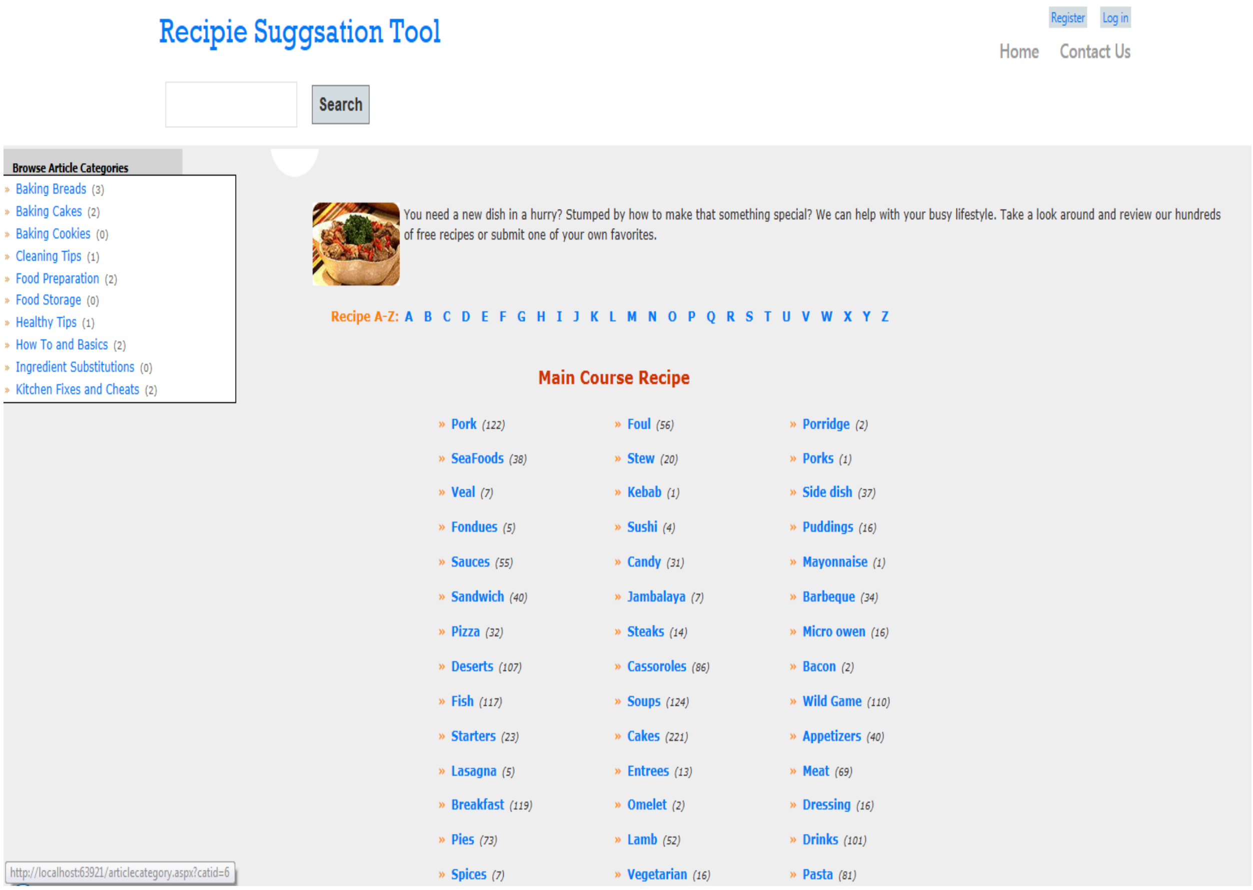
Register. Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

2.3 Home Page:

Once user login to the website they can search recipes in multiple ways like,
By selecting Ingredient,
By selecting country name,
By Keyword,
By alphabet

The search engine returns recipe image, recipe id, recipe name, category, and country.



Recipie Suggsation Tool

- Baking Breads (3)
- Baking Cakes (2)
- Baking Cookies (0)
- Cleaning Tips (1)
- Food Preparation (2)
- Food Storage (0)
- Healthy Tips (1)
- How To and Basics (2)
- Ingredient Substitutions (0)
- Kitchen Fixes and Cheats (2)

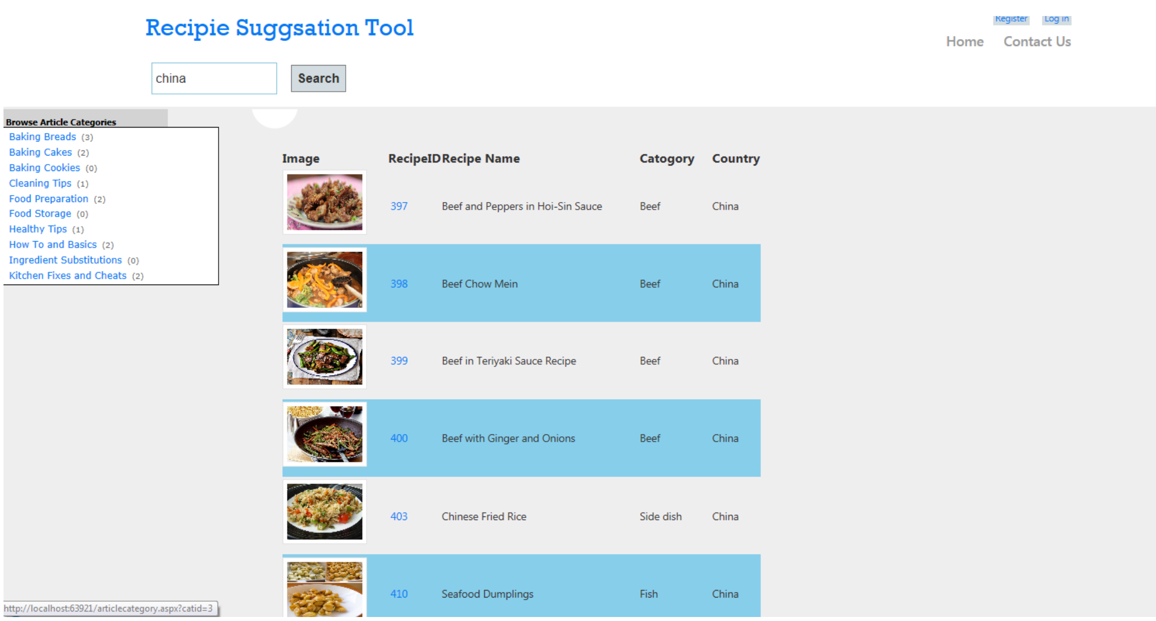
You need a new dish in a hurry? Stumped by how to make that something special? We can help with your busy lifestyle. Take a look around and review our hundreds of free recipes or submit one of your own favorites.

Recipe A-Z: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Main Course Recipe

- Pork (122)
- SeaFoods (38)
- Veal (7)
- Fondues (5)
- Sauces (55)
- Sandwich (40)
- Pizza (32)
- Deserts (107)
- Fish (117)
- Starters (23)
- Lasagna (5)
- Breakfast (119)
- Pies (73)
- Spices (7)
- Foul (56)
- Stew (20)
- Kebab (1)
- Sushi (4)
- Candy (31)
- Jambalaya (7)
- Steaks (14)
- Cassoroles (86)
- Soups (124)
- Cakes (221)
- Entrees (13)
- Omelet (2)
- Lamb (52)
- Vegetarian (16)
- Porridge (2)
- Porks (1)
- Side dish (37)
- Puddings (16)
- Mayonnaise (1)
- Barbeque (34)
- Micro owen (16)
- Bacon (2)
- Wild Game (110)
- Appetizers (40)
- Meat (69)
- Dressing (16)
- Drinks (101)
- Pasta (81)

http://localhost:63921/articlecategory.aspx?catid=6



Recipie Suggsation Tool

china

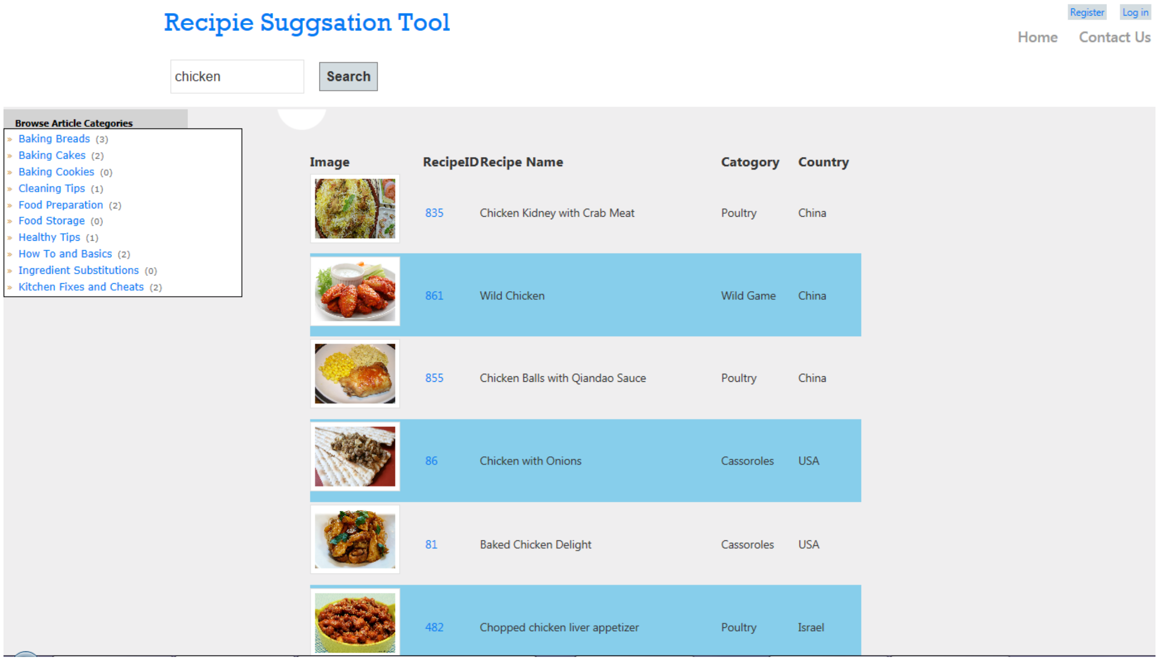
Browse Article Categories

- Baking Breads (3)
- Baking Cakes (2)
- Baking Cookies (0)
- Cleaning Tips (1)
- Food Preparation (2)
- Food Storage (0)
- Healthy Tips (1)
- How To and Basics (2)
- Ingredient Substitutions (0)
- Kitchen Fixes and Cheats (2)

http://localhost:63921/articlecategory.aspx?catid=3

Image	RecipeID	Recipe Name	Catogory	Country
	397	Beef and Peppers in Hoi-Sin Sauce	Beef	China
	398	Beef Chow Mein	Beef	China
	399	Beef in Teriyaki Sauce Recipe	Beef	China
	400	Beef with Ginger and Onions	Beef	China
	403	Chinese Fried Rice	Side dish	China
	410	Seafood Dumplings	Fish	China

\



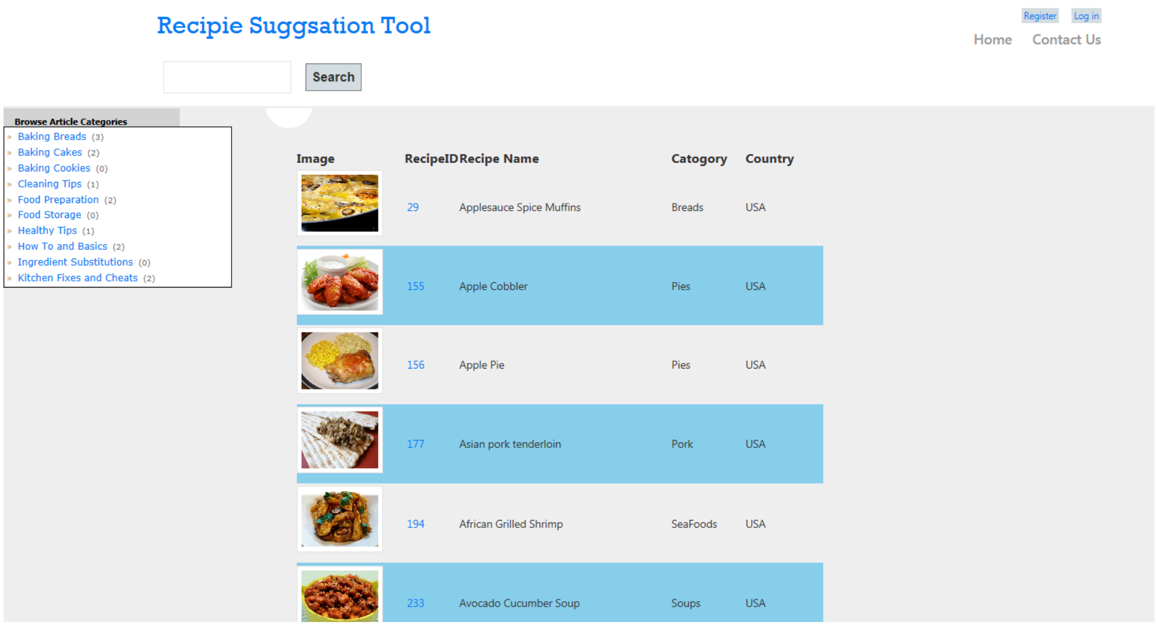
Recipie Suggsation Tool

chicken

Browse Article Categories

- Baking Breads (3)
- Baking Cakes (2)
- Baking Cookies (0)
- Cleaning Tips (1)
- Food Preparation (2)
- Food Storage (0)
- Healthy Tips (1)
- How To and Basics (2)
- Ingredient Substitutions (0)
- Kitchen Fixes and Cheats (2)

Image	RecipeID	Recipe Name	Catogory	Country
	835	Chicken Kidney with Crab Meat	Poultry	China
	861	Wild Chicken	Wild Game	China
	855	Chicken Balls with Qiandao Sauce	Poultry	China
	86	Chicken with Onions	Cassoroles	USA
	81	Baked Chicken Delight	Cassoroles	USA
	482	Chopped chicken liver appetizer	Poultry	Israel



Recipie Suggsation Tool

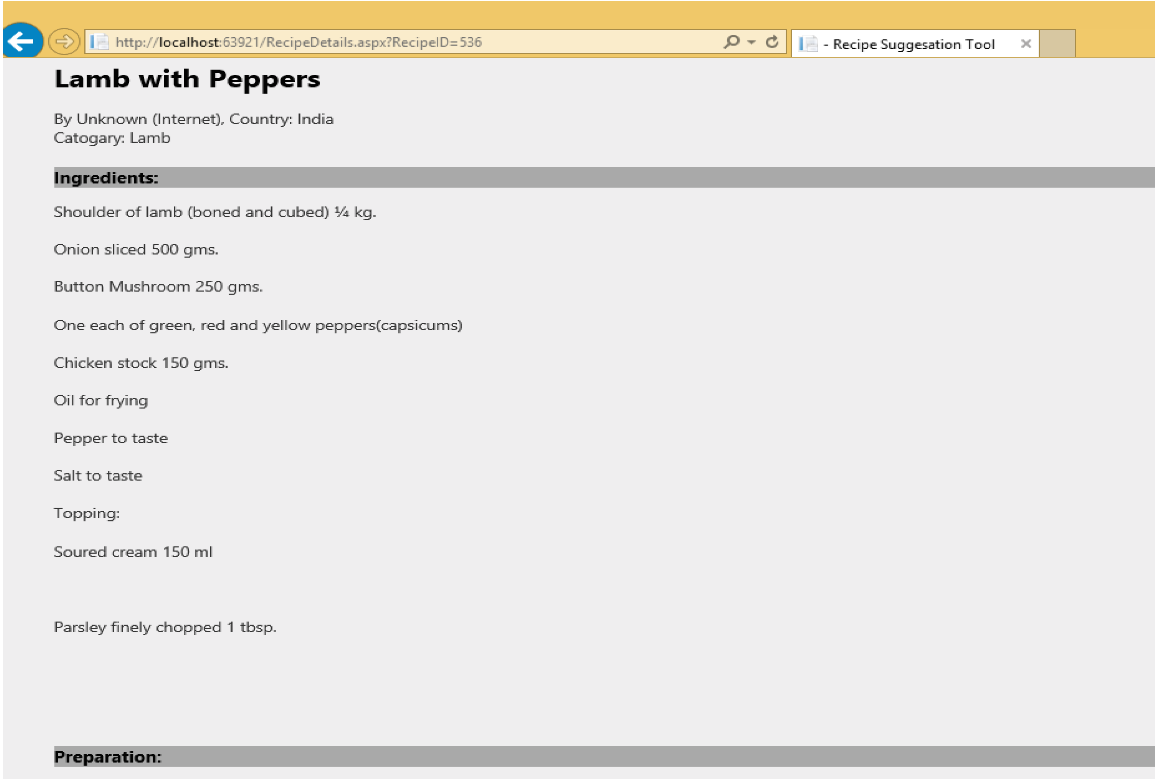
Browse Article Categories

- Baking Breads (3)
- Baking Cakes (2)
- Baking Cookies (0)
- Cleaning Tips (1)
- Food Preparation (2)
- Food Storage (0)
- Healthy Tips (1)
- How To and Basics (2)
- Ingredient Substitutions (0)
- Kitchen Fixes and Cheats (2)

Image	RecipeID	Recipe Name	Catogory	Country
	29	Applesauce Spice Muffins	Breads	USA
	155	Apple Cobbler	Pies	USA
	156	Apple Pie	Pies	USA
	177	Asian pork tenderloin	Pork	USA
	194	African Grilled Shrimp	SeaFoods	USA
	233	Avocado Cucumber Soup	Soups	USA

Description Page:

Users can see ingredients and preparation method of the recipe once they click on recipe id.


http://localhost:63921/RecipeDetails.aspx?RecipeID=536
- Recipe Suggesation Tool

Lamb with Peppers

By Unknown (Internet), Country: India
Catogary: Lamb

Ingredients:

Shoulder of lamb (boned and cubed) ¾ kg.

Onion sliced 500 gms.

Button Mushroom 250 gms.

One each of green, red and yellow peppers(capsicums)

Chicken stock 150 gms.

Oil for frying

Pepper to taste

Salt to taste

Topping:

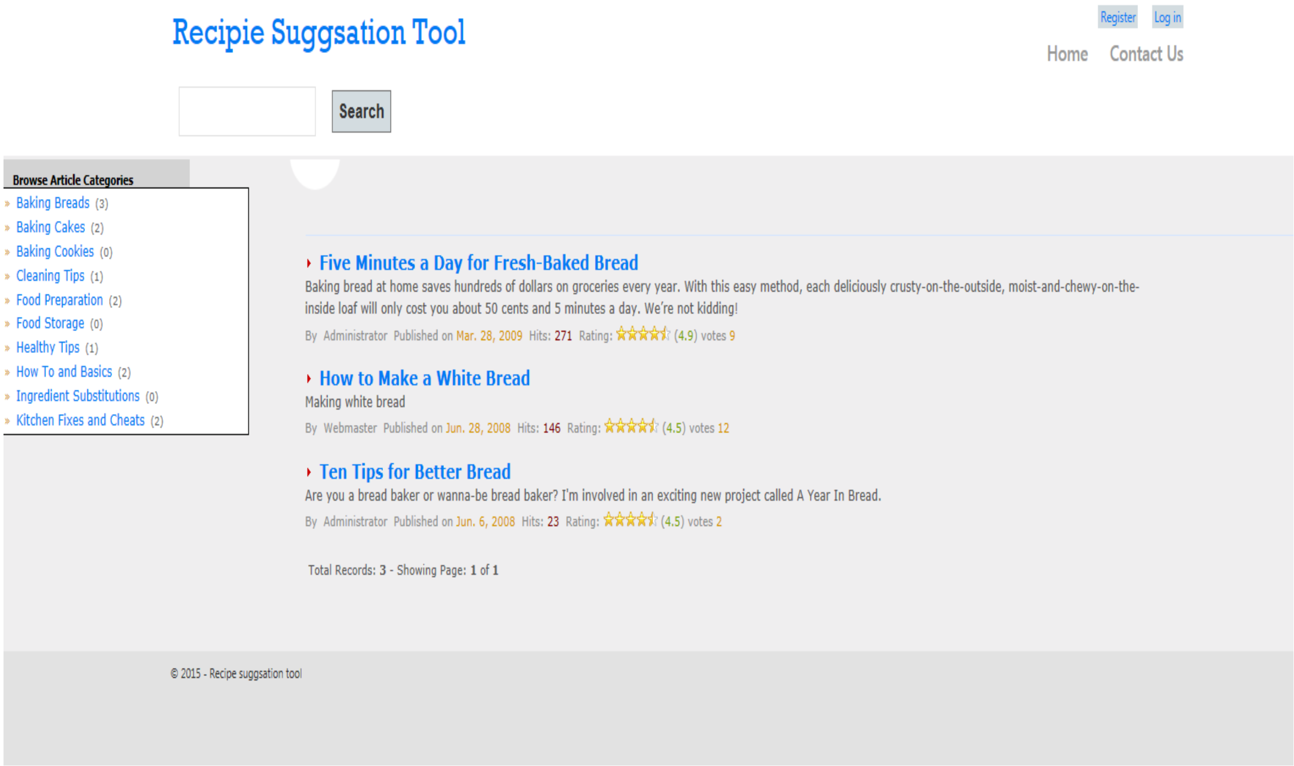
Soured cream 150 ml

Parsley finely chopped 1 tbsp.

Preparation:

2.4 Article Page:

In this page user can read articles, can give rating and can enter comments. We have used user control for left section of the article and repeater to show the list on right section.



Browse Article Categories

- » Baking Breads (3)
- » Baking Cakes (2)
- » Baking Cookies (0)
- » Cleaning Tips (1)
- » Food Preparation (2)
- » Food Storage (0)
- » Healthy Tips (1)
- » How To and Basics (2)
- » Ingredient Substitutions (0)
- » Kitchen Fixes and Cheats (2)

Five Minutes a Day for Fresh-Baked Bread

Baking bread at home saves hundreds of dollars on groceries every year. With this easy method, each deliciously crusty-on-the-outside, moist-and-chewy-on-the-inside loaf will only cost you about 50 cents and 5 minutes a day. We're not kidding!

By Administrator Published on Mar. 28, 2009 Hits: 271 Rating: ★★★★☆ (4.9) votes 9

How to Make a White Bread

Making white bread

By Webmaster Published on Jun. 28, 2008 Hits: 146 Rating: ★★★★☆ (4.5) votes 12

Ten Tips for Better Bread

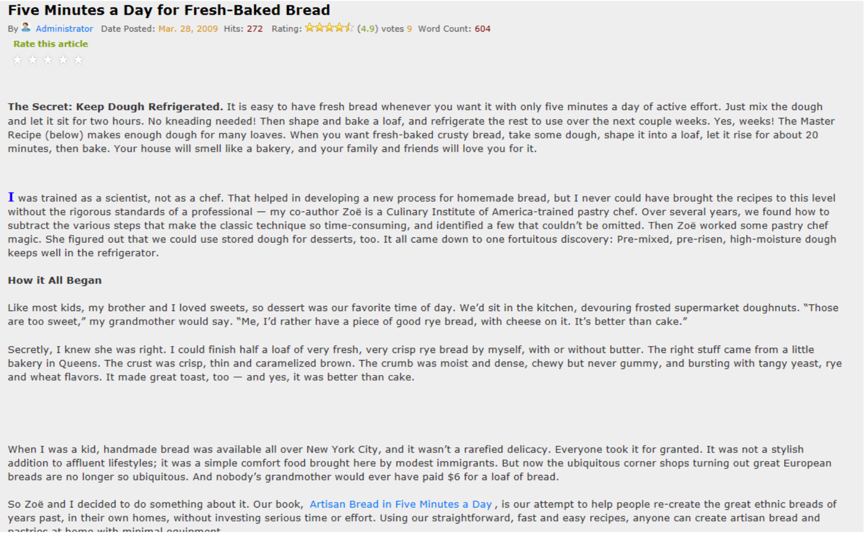
Are you a bread baker or wanna-be bread baker? I'm involved in an exciting new project called A Year In Bread.

By Administrator Published on Jun. 6, 2008 Hits: 23 Rating: ★★★★☆ (4.5) votes 2

Total Records: 3 - Showing Page: 1 of 1

© 2015 - Recipe suggsation tool

By clicking on the name you can go to full article page where logged in users can enter comments and rate the article.



Five Minutes a Day for Fresh-Baked Bread

By Administrator Date Posted: Mar. 28, 2009 Hits: 272 Rating: ★★★★☆ (4.9) votes 9 Word Count: 604

Rate this article

☆☆☆☆☆

The Secret: Keep Dough Refrigerated. It is easy to have fresh bread whenever you want it with only five minutes a day of active effort. Just mix the dough and let it sit for two hours. No kneading needed! Then shape and bake a loaf, and refrigerate the rest to use over the next couple weeks. Yes, weeks! The Master Recipe (below) makes enough dough for many loaves. When you want fresh-baked crusty bread, take some dough, shape it into a loaf, let it rise for about 20 minutes, then bake. Your house will smell like a bakery, and your family and friends will love you for it.

I was trained as a scientist, not as a chef. That helped in developing a new process for homemade bread, but I never could have brought the recipes to this level without the rigorous standards of a professional — my co-author Zoë is a Culinary Institute of America-trained pastry chef. Over several years, we found how to subtract the various steps that make the classic technique so time-consuming, and identified a few that couldn't be omitted. Then Zoë worked some pastry chef magic. She figured out that we could use stored dough for desserts, too. It all came down to one fortuitous discovery: Pre-mixed, pre-risen, high-moisture dough keeps well in the refrigerator.

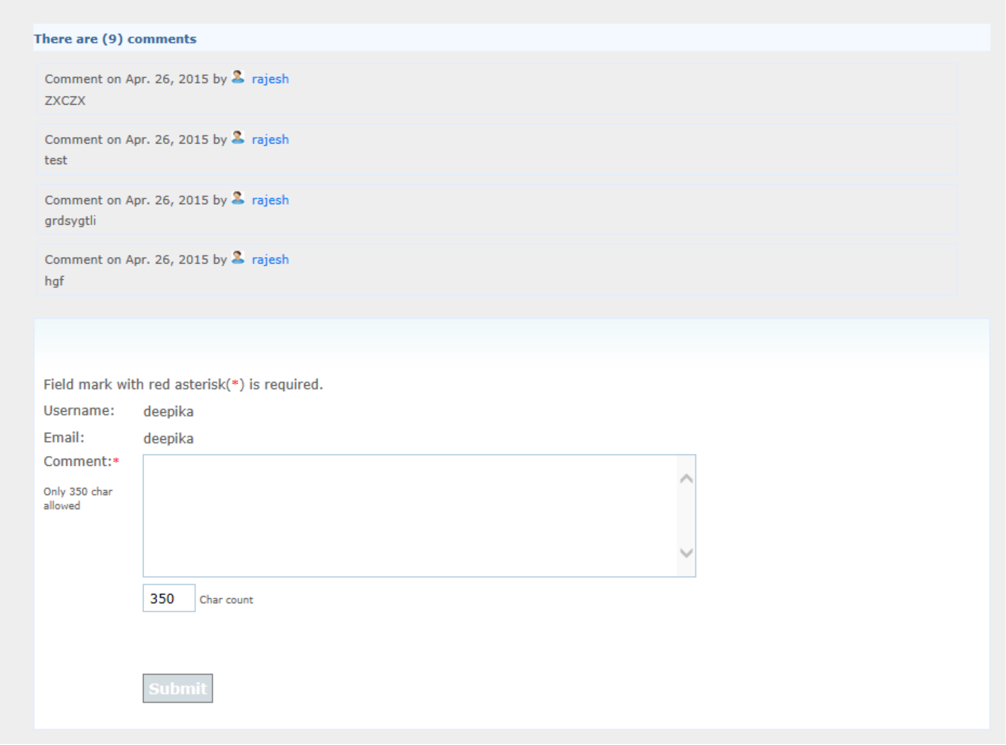
How it All Began

Like most kids, my brother and I loved sweets, so dessert was our favorite time of day. We'd sit in the kitchen, devouring frosted supermarket doughnuts. "Those are too sweet," my grandmother would say. "Me, I'd rather have a piece of good rye bread, with cheese on it. It's better than cake."

Secretly, I knew she was right. I could finish half a loaf of very fresh, very crisp rye bread by myself, with or without butter. The right stuff came from a little bakery in Queens. The crust was crisp, thin and caramelized brown. The crumb was moist and dense, chewy but never gummy, and bursting with tangy yeast, rye and wheat flavors. It made great toast, too — and yes, it was better than cake.

When I was a kid, handmade bread was available all over New York City, and it wasn't a rarefied delicacy. Everyone took it for granted. It was not a stylish addition to affluent lifestyles; it was a simple comfort food brought here by modest immigrants. But now the ubiquitous corner shops turning out great European breads are no longer so ubiquitous. And nobody's grandmother would ever have paid $6 for a loaf of bread.

So Zoë and I decided to do something about it. Our book, Artisan Bread in Five Minutes a Day, is our attempt to help people re-create the great ethnic breads of years past, in their own homes, without investing serious time or effort. Using our straightforward, fast and easy recipes, anyone can create artisan bread and pastries at home with minimal equipment.



There are (9) comments

Comment on Apr. 26, 2015 by rajesh
ZXCZX

Comment on Apr. 26, 2015 by rajesh
test

Comment on Apr. 26, 2015 by rajesh
grdsygtli

Comment on Apr. 26, 2015 by rajesh
hgf

Field mark with red asterisk(*) is required.

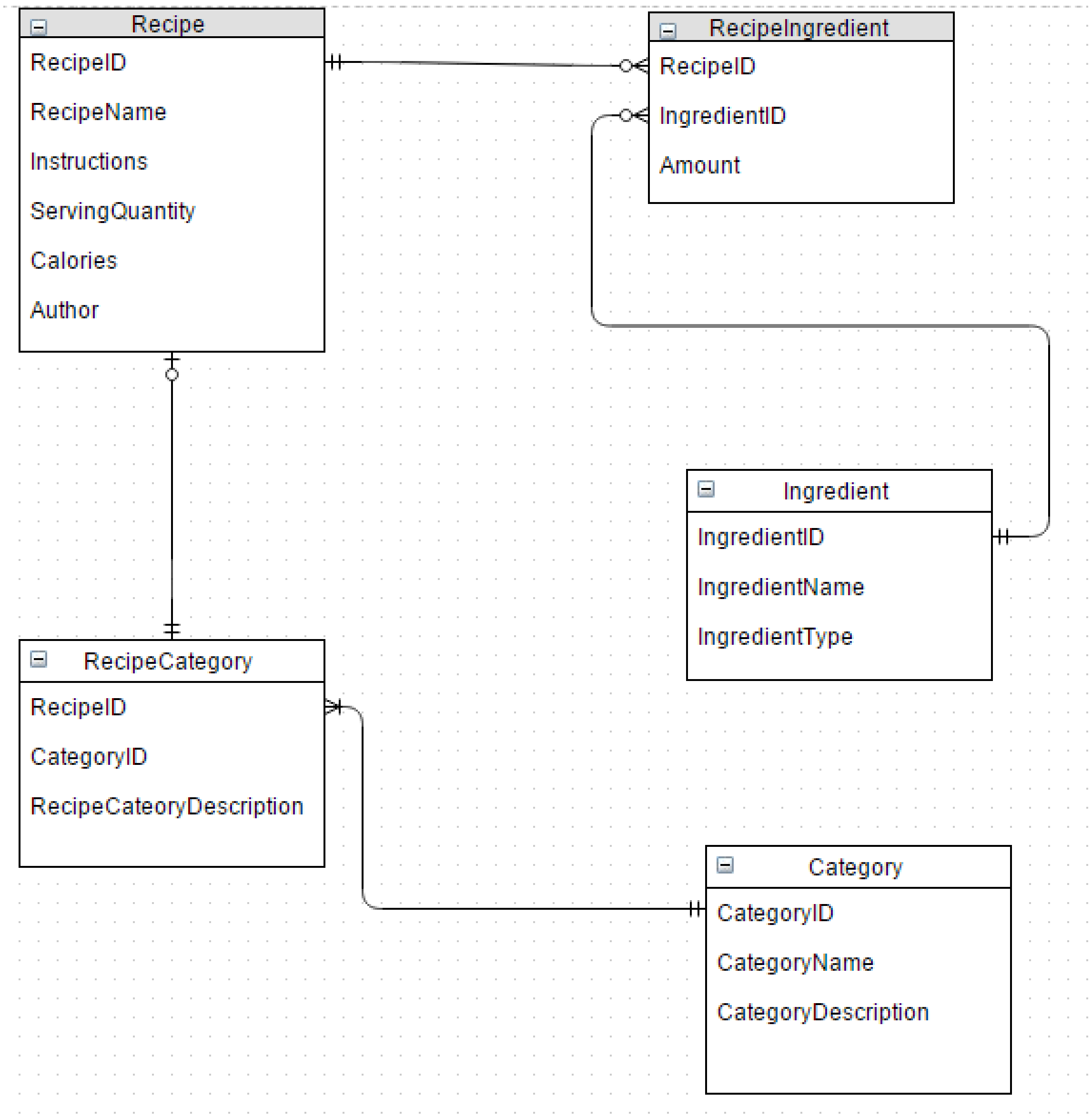
Username:	deepika
Email:	deepika
Comment:* Only 350 char allowed	
	350 Char count

3.0 DATABASE DESIGN

3.1 Data Definition

The main repository for Recipe data is a relational database utilizing Microsoft SQL server 2008 as the management system and hosted on a Microsoft Azure. This repository is where all Recipe data is stored and maintained. There are two instances, which exist to support the development and production Recipe sites. The Recipe database is normalized and converted to a MS SQL database, which is hosted on Windows Virtual machine in Azure. The External website uses this MSSQL database. The renormalization of the main Recipe database allows for faster query performance on the external web site.

3.2 E-R DIAGRAM

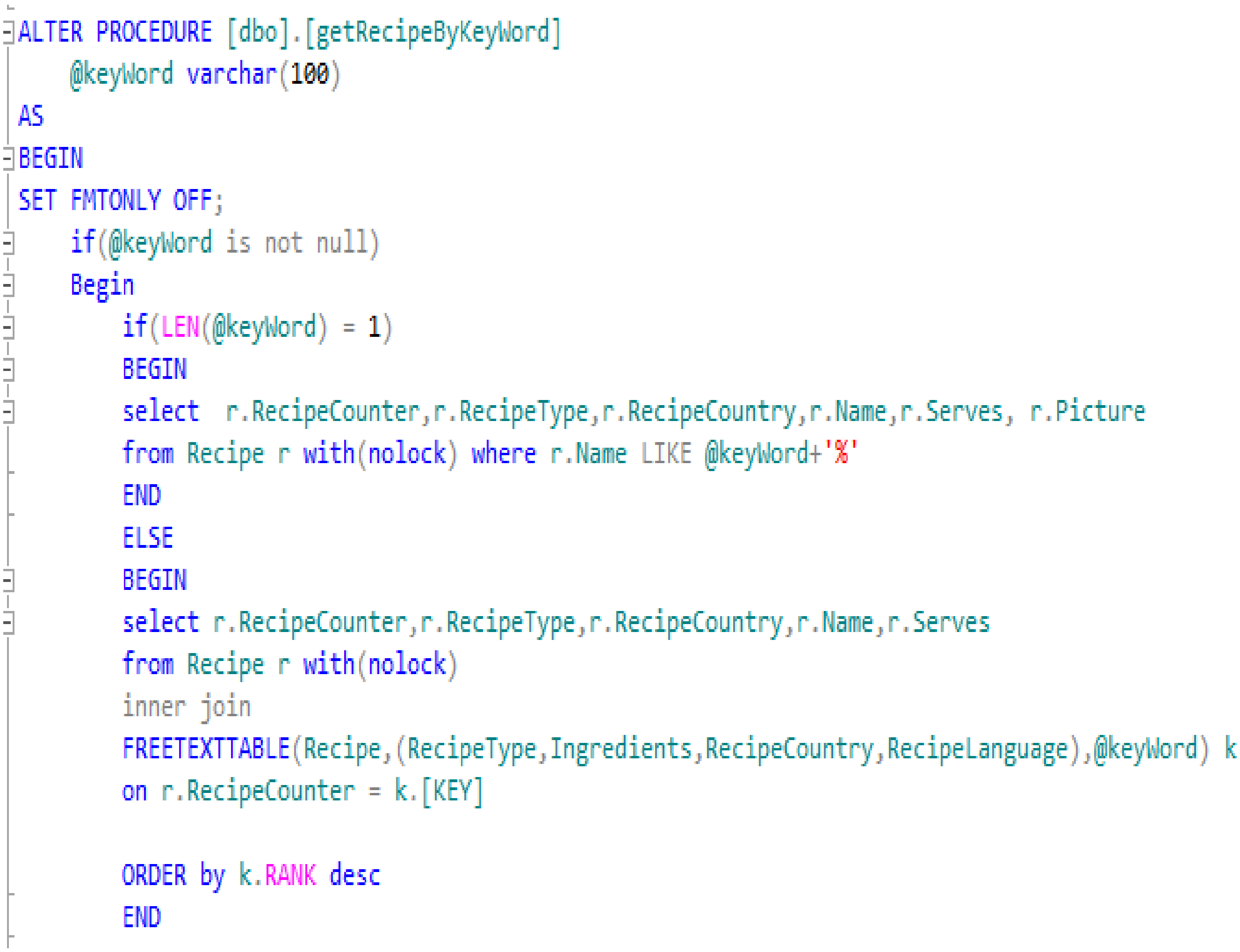


4.0 DATA VIZUALIZATION

- GetRecipeByKeyWord
- GetRecipeCountByIngredients
- GetRecipeCountByCountry

4.1 GetRecipeByKeyWord:

- We used FreeTextTable for free text search on the table.
- Search all the columns based on our priority and it will give ranking.



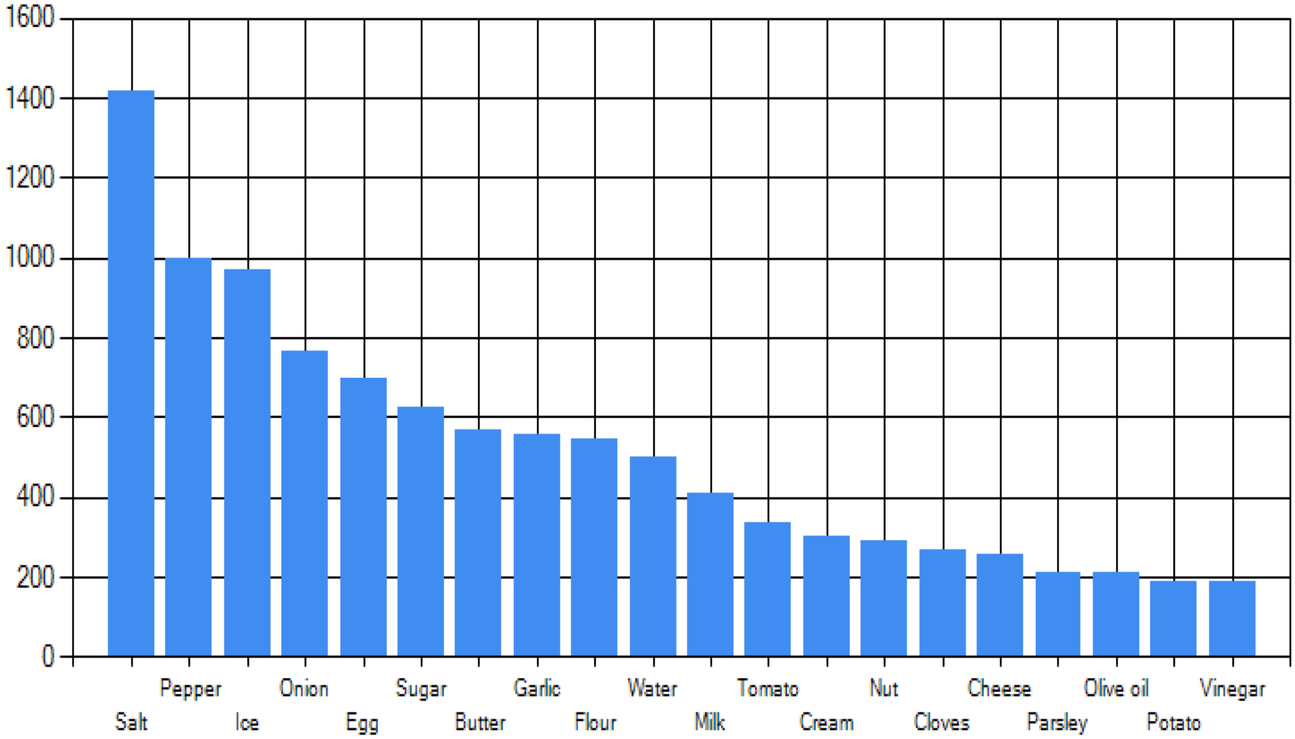
```
ALTER PROCEDURE [dbo].[getRecipeByKeyWord]
    @keyWord varchar(100)
AS
BEGIN
SET FMTONLY OFF;
    if(@keyWord is not null)
    Begin
        if(LEN(@keyWord) = 1)
        BEGIN
        select  r.RecipeCounter,r.RecipeType,r.RecipeCountry,r.Name,r.Serves, r.Picture
        from Recipe r with(nolock) where r.Name LIKE @keyWord+'%'
        END
        ELSE
        BEGIN
        select r.RecipeCounter,r.RecipeType,r.RecipeCountry,r.Name,r.Serves
        from Recipe r with(nolock)
        inner join
        FREETEXTTABLE(Recipe,(RecipeType,Ingredients,RecipeCountry,RecipeLanguage),@keyWord) k
        on r.RecipeCounter = k.[KEY]

        ORDER by k.RANK desc
        END
```

4.2 GetRecipeCountByIngredients:

Get recipe count by Ingredients

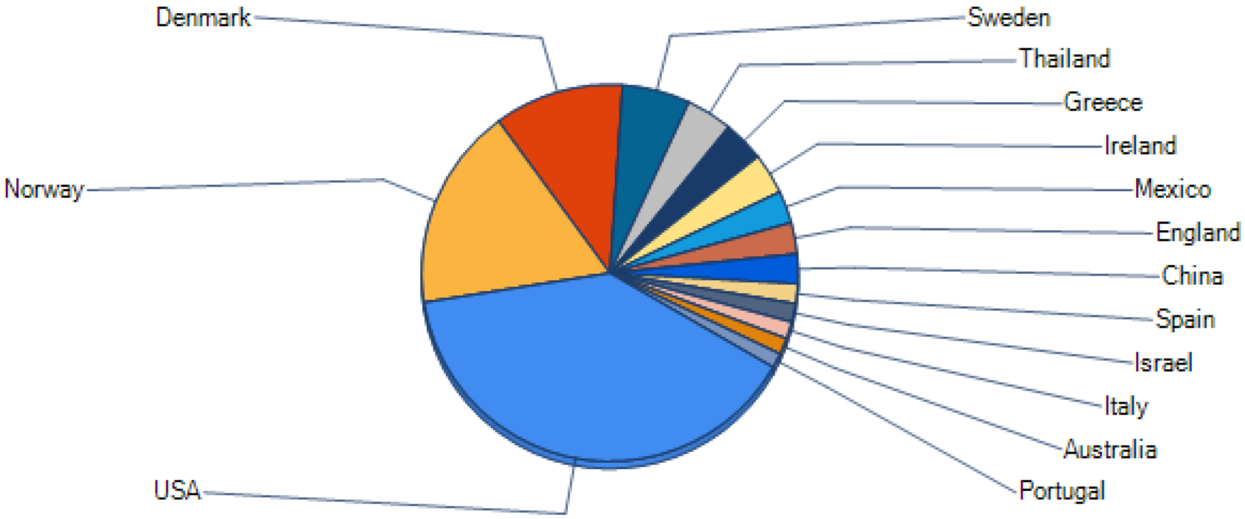
We created chart for top 20 ingredients vs. count. Collected data for this graph from stored procedure. We used ASP.NET chart controller for drawing charts.
This is the graph for top 20 ingredients vs. count by descending order



In out of 1600 recopies top 20 ingredients are Salt, Pepper, Ice, Onions, Egg, Sugar, Butter, Garlic, Flour, Water, Milk, Tomato, Cream, Nut, Cloves, Cheese, Parsley, Olive oil, Potato, Vinegar.

4.3 Get recipe count by country:

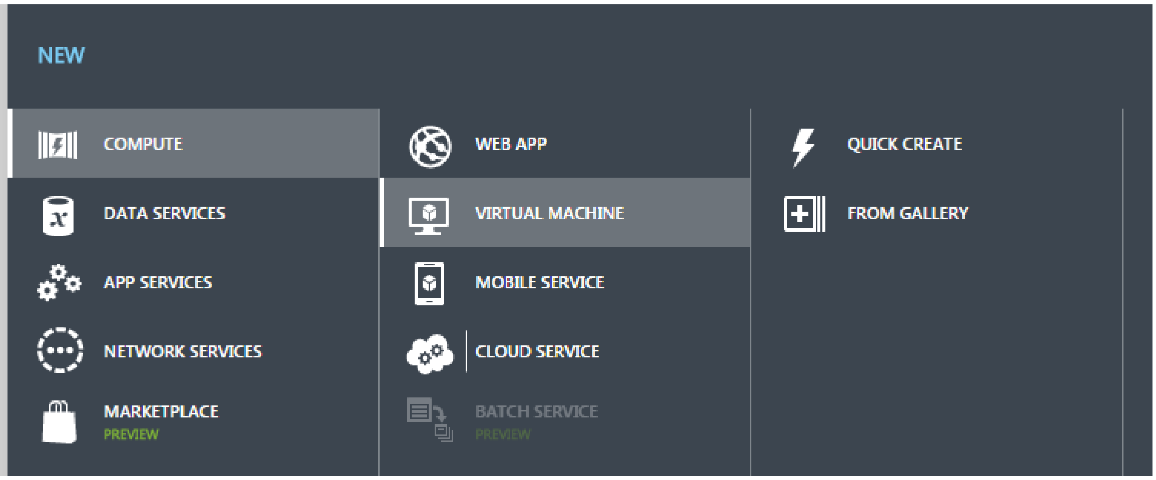
This is a Pie chart from asp.net chart controller
This chart shows top 15 counties by recipe cunt count in recipe database. Stored procedure use for this chart is getRecipeCountByCountry.



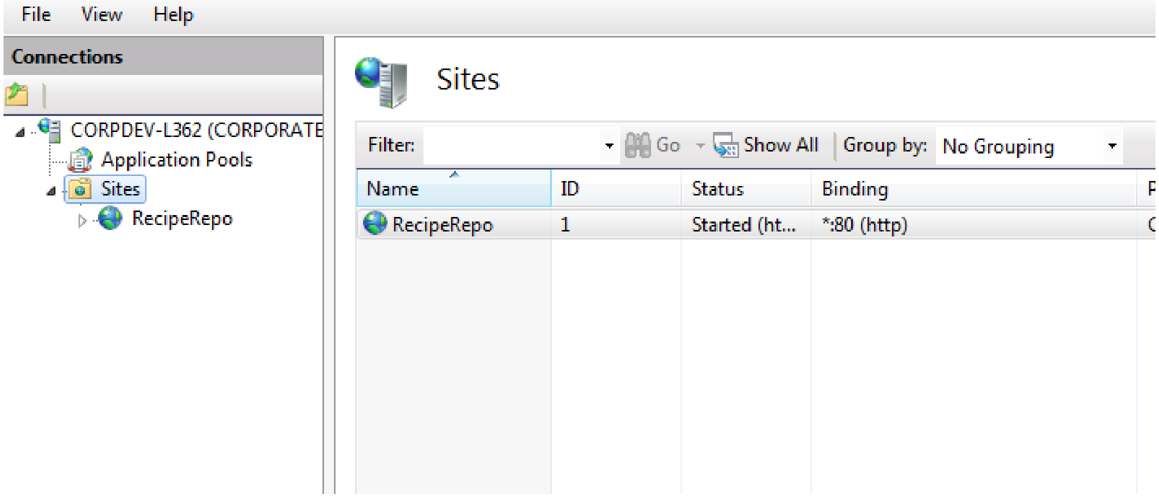
5.0 AZURE CLOUD:

We have deployed our code in Microsoft azure by creating a new virtual machine and setting up the IIS for hosting the website in that machine.

5.1 Creating Virtual machine:



5.2 Host website on IIS:



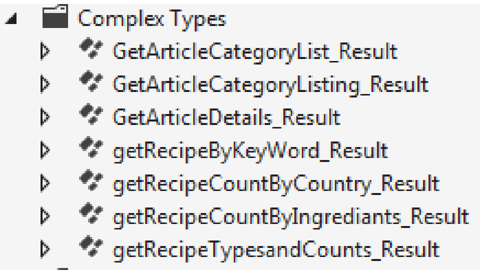
Name	ID	Status	Binding	P
RecipeRepo	1	Started (ht...	*:80 (http)	C

6.0 Techinical Environments

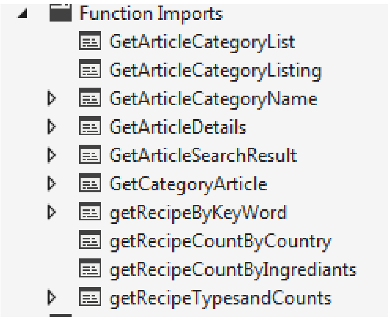
6.1 Development

1) Entity framework:

The Entity Framework is a set of technologies in ADO.NET that support the development of data-oriented software applications. We have used entity framework to read data from database and execute stored procedures. We have created complex types for stored procedures and used them throughout the project



Complex Types
GetArticleCategoryList_Result
GetArticleCategoryListing_Result
GetArticleDetails_Result
getRecipeByKeyWord_Result
getRecipeCountByCountry_Result
getRecipeCountByIngrediants_Result
getRecipeTypesandCounts_Result



Function Imports
GetArticleCategoryList
GetArticleCategoryListing
GetArticleCategoryName
GetArticleDetails
GetArticleSearchResult
GetCategoryArticle
getRecipeByKeyWord
getRecipeCountByCountry
getRecipeCountByIngrediants
getRecipeTypesandCounts

2) Microsoft ASP.NET

3) SQL Server 2008

4) Microsoft Azure

References

https://msdn.microsoft.com/en-us/data/ef.aspx
http://azure.microsoft.com/en-us/documentation/articles/web-sites-dotnet-get-started/
http://www.vahrehvah.com/
http://www.recipe.com/
http://www.ingredientpairings.com/

Appendix 1

Sample Asp.net page

```
<%@ Page Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="Home.aspx.cs" Inherits="RecipeSuggestionTool.Home" %>
<%@ Register TagPrefix="ucl" TagName="alphaletter" Src="Control/alphaletter.ascx"
%>


<asp:Content ID="Content2" ContentPlaceHolderID="LeftPanel" Runat="Server">



</asp:Content>

<asp:Content runat="server" ID="BodyContent" ContentPlaceHolderID="MainContent">

    <div style="padding: 2px; text-align: left; margin-left: 40px; margin-bottom:
15px; margin-top: 16px; margin-right: 40px;">
    <asp:Image id="Myranimage" runat="server"
     Width = 107 Height = 74
     AlternateText = "Recipe Random Image" Style="float:left; padding-right: 5px;"
    />
    <span class="drecipe">
    You need a new dish in a hurry? Stumped by how to make that something special?
We can help with your busy lifestyle. Take a look around and review our hundreds
of free recipes or submit one of your own favorites.
    </span>
    </div>
    <br />
    <br />
    <div style="float:left">
    <div style="padding: 2px; text-align: center; margin-left: 26px; margin-
bottom: 12px; margin-right: 26px;">
    <ucl:alphaletter id="alpha1" runat="server"></ucl:alphaletter>
    </div>
    <div style="text-align: center; padding-top: 3px;"><asp:Label
cssClass="content2" runat="server" id="lbltotalRecipe" />
    </div>
    <br />
    <div style="text-align: center;  padding-bottom: 5px;"><span style="font-
family: verdana,arial; font-size: 17px; color: #CC3300;"><b>Main Course
Recipe</b></span></div>
    <div class="MainCourseCat">
    <asp:DataList id="MainCourseCategory" RepeatColumns="3"
RepeatDirection="Horizontal" runat="server" HorizontalAlign="Center">
          <ItemTemplate>
         <div style="margin-left: 60px; margin-top: 3px; margin-bottom: 3px;
margin-right: 10px;">
    <span class="bluearrow">&raquo;</span> <a class="catlink" title="Browse all
<%# Eval("Catagory") %> recipe" href='<%# Eval("Catagory",
"Default.aspx?keyword={0}") %>'><%# Eval("Catagory")%></a> <span
class="catcount"><i>(<%# Eval("cnt")%>)</i></span>
           </div>
          </ItemTemplate>
      </asp:DataList>
      </div>
      <br />
      <div style="clear:both;"></div>
      <div style="text-align: center;  padding-bottom: 5px;"><span style="font-
family: verdana,arial; font-size: 17px; color: #CC3300;"><b>Ethnic &amp; Regional
Cuisine</b></span></div>
    <div class="EthnicCat">
    <asp:DataList id="EthnicRegionalCat" RepeatColumns="3"
RepeatDirection="Horizontal" runat="server" HorizontalAlign="Center">
          <ItemTemplate>
         <div style="margin-left: 60px; margin-top: 3px; margin-bottom: 3px;
margin-right: 10px;">
    <span class="bluearrow">&raquo;</span> <a class="catlink" title="Browse all
<%# Eval("Catagory") %> recipe" href='<%# Eval("Catagory",
"Default.aspx?keyword={0}") %>'><%# Eval("Catagory")%></a> <span
class="catcount"><i>(<%# Eval("cnt")%>)</i></span>
           </div>
          </ItemTemplate>
      </asp:DataList>
      </div>
        </div>
      <div style="clear:both; margin-top: 16px;"></div>
</asp:Content>
```

Sample c# code behind

```
using RecipeSuggestionTool.DataLayer;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace RecipeSuggestionTool
{
    public partial class _Default : Page
    {
        BusinessLayer businessLayer = new BusinessLayer();
        protected void Page_Load(object sender, EventArgs e)
        {
            string keyword = String.Empty;
            if (Request.QueryString["keyword"] != null)
            {
                keyword = Request.QueryString["keyword"];
                grdResults.DataSource =
businessLayer.searchByKeyWord(keyword.Trim());
                grdResults.DataBind();
            }
        }
        protected void grdResults_RowDataBound(object sender, GridViewRowEventArgs
e)
        {
            if (e.Row.RowType == DataControlRowType.DataRow)
            {
                ImageButton img = (ImageButton)e.Row.FindControl("btnImg");

                getRecipeByKeyWord_Result item = new getRecipeByKeyWord_Result();
                item = (getRecipeByKeyWord_Result)e.Row.DataItem;
                    img.ImageUrl =
"~/Images/RecipeImage/"+item.RecipeCounter+".jpg";
                //}
                e.Row.Cells[0].Controls.Add(img);
            }
        }



        protected void grdResults_RowCommand(object sender,
GridViewCommandEventArgs e)
        {
            if (e.CommandName == "item")
            {
                Response.Redirect("~/RecipeDetails.aspx?RecipeID=" +
((LinkButton)e.CommandSource).Text);

            }
        }

        protected void grdResults_SelectedIndexChanged(object sender, EventArgs e)
        {

        }
    }
}
```

Sample business layer

```
using RecipeSuggestionTool.DataLayer.Model;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;

namespace RecipeSuggestionTool.DataLayer
{
    public  class BusinessLayer
    {
        public List<getRecipeByKeyWord_Result> searchByKeyWord(string keyWord)
        {
            RecipiesRepoEntities recipeRepo = new RecipiesRepoEntities();
            List<getRecipeByKeyWord_Result> searchResult = new 
List<getRecipeByKeyWord_Result>();
            searchResult = recipeRepo.getRecipeByKeyWord(keyWord).ToList();
            return searchResult;
        }
        public List<getRecipeTypesandCounts_Result> getCategories(int type)
        {
            RecipiesRepoEntities recipeRepo = new RecipiesRepoEntities();
            List<getRecipeTypesandCounts_Result> types = new 
List<getRecipeTypesandCounts_Result>();
           types =  recipeRepo.getRecipeTypesandCounts(type).ToList();
           return types;
        }
        public List<GetArticleCategoryListing_Result> GetArticleCatogories()
        {
            RecipiesRepoEntities recipeRepo = new RecipiesRepoEntities();
            List<GetArticleCategoryListing_Result> result = new 
List<GetArticleCategoryListing_Result>();
            result = recipeRepo.GetArticleCategoryListing().ToList();
            return result;
        }

        #region Get article details, Update hit counter, Get Category list, Add, 
Update and Delete Article

        /// <summary>
        /// Returns article category list.
        /// </summary>
        /// <returns></returns>
        public IDataReader GetArticleCategoryList
        {
            get { return DataAccess.GetFromReader("GetArticleCategoryList"); }
        }

        /// <summary>
        /// Returns 10 newest cooking articles in the right side panel.
        /// </summary>
        /// <returns></returns>
        public IDataReader GetNewestArticleSidePanel(int Top)
        {
            SqlParameter prmTop = new SqlParameter("@Top", SqlDbType.Int, 4);
            prmTop.Value = Top;

            return DataAccess.GetFromReader("GetNewestArticleSidePanel", prmTop);
        }

        /// <summary>
        /// Returns Article Category
        /// </summary>
        public IDataReader GetArticlesInCategory(int CATID, int OrderBy, int
SortBy, int PageIndex, int PageSize)
        {
            SqlParameter prmCatId = new SqlParameter("@CATID", SqlDbType.Int, 4);
            prmCatId.Value = CATID;

            SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int,
4);
            prmOrderBy.Value = OrderBy;

            SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int,
4);
            prmSortBy.Value = SortBy;

            SqlParameter prmPageIndex = new SqlParameter("@PageIndex",
SqlDbType.Int, 4);
            prmPageIndex.Value = PageIndex;

            SqlParameter prmPageSize = new SqlParameter("@PageSize",
SqlDbType.Int, 4);
            prmPageSize.Value = PageSize;

            return DataAccess.GetFromReader("GetCategoryArticle", prmCatId,
prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
        }

        /// <summary>
        /// Returns top 10 articles submitted by author/user
        /// </summary>
        public IDataReader GetTop10OtherArticlesByThisAuthor(int UserID, int AID)
        {
            SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int,
4);
            prmUserID.Value = UserID;

            SqlParameter prmAID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmAID.Value = AID;

            return DataAccess.GetFromReader("spSelectOtherArticlesByThisAuthor",
prmUserID, prmAID);
        }

        /// <summary>
        /// Returns 10 related articles
        /// </summary>
        public IDataReader GetRelatedArticle(int CatID, int AID)
        {
            SqlParameter prmCatId = new SqlParameter("@CatID", SqlDbType.Int, 4);
            prmCatId.Value = CatID;
            SqlParameter prmAID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmAID.Value = AID;

            return DataAccess.GetFromReader("spSelectRelatedArticle", prmCatId, 
prmAID);
        }

        /// <summary>
        /// Returns IDataReader Article Category
        /// </summary>
        public IDataReader GetArticleCategory(int CATID, int OrderBy)
        {
            SqlParameter prmCatId = new SqlParameter("@CATID", SqlDbType.Int, 4);
            prmCatId.Value = CATID;

            SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int, 
4);
            prmOrderBy.Value = OrderBy;

            return DataAccess.GetFromReader("CategoryPage_Article", prmCatId, 
prmOrderBy);
        }

        /// <summary>
        /// Returns article category name
        /// </summary>
        /// <returns></returns>
        public string GetArticleCategoryName(int CAT_ID)
        {
            SqlParameter prmCatID = new SqlParameter("@CAT_ID", SqlDbType.Int, 4);
            prmCatID.Value = CAT_ID;

            return DataAccess.GetString("GetArticleCategoryName", prmCatID);
        }

        /// <summary>
        /// Update article rating
        /// </summary>
        /// <returns></returns>
        public static int AddArticleRating(int ID, int Rating)
        {
            SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmID.Value = ID;

            SqlParameter prmRating = new SqlParameter("@Rating", SqlDbType.Int, 
4);
            prmRating.Value = Rating;

            return DataAccess.Execute("AddArticleRating", prmID, prmRating);
        }

        /// <summary>
        /// Insert article
        /// </summary>
        /// <returns></returns>
        public int AddArticle(article article)
        {
            SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 
4);
            prmUserID.Value = article.UID;

            SqlParameter prmTitle = new SqlParameter("@Title", SqlDbType.VarChar, 
200);
            prmTitle.Value = article.Title;

            SqlParameter prmContent = new SqlParameter("@Content", 
SqlDbType.VarChar, 8000);
            prmContent.Value = article.Content;

            SqlParameter prmAuthor = new SqlParameter("@Author", 
SqlDbType.VarChar, 50);
            prmAuthor.Value = article.Author;

            SqlParameter prmCatID = new SqlParameter("@CAT_ID", SqlDbType.Int, 4);
            prmCatID.Value = article.CatID;

            SqlParameter prmKeyword = new SqlParameter("@Keyword", 
SqlDbType.VarChar, 255);
            prmKeyword.Value = article.Keyword;

            SqlParameter prmSummary = new SqlParameter("@Summary", 
SqlDbType.VarChar, 500);
            prmSummary.Value = article.Summary;

            return DataAccess.Execute("spInsertArticle", prmUserID, prmTitle, 
prmContent, prmAuthor, prmCatID, prmKeyword, prmSummary);

        }

        /// <summary>
        /// Update a users article
        /// </summary>
        /// <returns></returns>
        public int UpdateArticle(article article)
        {
            SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 
4);
            prmUserID.Value = article.UID;

            SqlParameter prmID = new SqlParameter("@AID", SqlDbType.Int, 4);
            prmID.Value = article.ID;

            SqlParameter prmTitle = new SqlParameter("@Title", SqlDbType.VarChar, 
200);
            prmTitle.Value = article.Title;

            SqlParameter prmContent = new SqlParameter("@Content", 
SqlDbType.VarChar, 8000);
            prmContent.Value = article.Content;

            SqlParameter prmCatID = new SqlParameter("@CAT_ID", SqlDbType.Int, 4);
            prmCatID.Value = article.CatID;
            SqlParameter prmKeyword = new SqlParameter("@Keyword",
SqlDbType.VarChar, 100);
            prmKeyword.Value = article.Keyword;

            SqlParameter prmSummary = new SqlParameter("@Summary",
SqlDbType.VarChar, 500);
            prmSummary.Value = article.Summary;

            return DataAccess.Execute("spUpdateArticle", prmUserID, prmID,
prmTitle, prmContent, prmCatID, prmKeyword, prmSummary);
        }

        /// <summary>
        /// Finalize Insert article
        /// </summary>
        /// <returns></returns>
        public int FinalizeAddArticle(int ID)
        {
            SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmID.Value = ID;

            return DataAccess.Execute("FinalizeArticleSubmission", prmID);
        }

        /// <summary>
        /// Returns last submitted article ID
        /// </summary>
        /// <returns></returns>
        public IDataReader GetLastArticleID
        {
            get { return DataAccess.GetFromReader("GetLastArticleID"); }
        }

        /// <summary>
        /// Admin Recipe Manager Delete Recipe
        /// </summary>
        /// <returns></returns>
        public int AdminDeleteArticle(article article)
        {
            SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmID.Value = article.ID;

            return DataAccess.Execute("ArticleDelete", prmID);
        }

        /// <summary>
        /// Returns article search result
        /// </summary>
        public IDataReader GetArticleSearchResult(string Search, int CatId, int
OrderBy, int SortBy, int PageIndex, int PageSize)
        {
            SqlParameter prmSearch = new SqlParameter("@Search",
SqlDbType.VarChar, 20);
            prmSearch.Value = Search;

            SqlParameter prmCatId = new SqlParameter("@CATID", SqlDbType.Int, 4);
            prmCatId.Value = CatId;
            SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int,
4);
            prmOrderBy.Value = OrderBy;

            SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int,
4);
            prmSortBy.Value = SortBy;

            SqlParameter prmPageIndex = new SqlParameter("@PageIndex",
SqlDbType.Int, 4);
            prmPageIndex.Value = PageIndex;

            SqlParameter prmPageSize = new SqlParameter("@PageSize",
SqlDbType.Int, 4);
            prmPageSize.Value = PageSize;

            return DataAccess.GetFromReader("GetArticleSearchResult", prmSearch,
prmCatId, prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
        }

        /// <summary>
        /// Returns all article submitted by an author/user.
        /// </summary>
        public IDataReader GetAllArticleByAuthor(string Author, int OrderBy, int
SortBy, int PageIndex, int PageSize)
        {
            SqlParameter prmAuthor = new SqlParameter("@FindByAuthor",
SqlDbType.VarChar, 20);
            prmAuthor.Value = Author;

            SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int,
4);
            prmOrderBy.Value = OrderBy;

            SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int,
4);
            prmSortBy.Value = SortBy;

            SqlParameter prmPageIndex = new SqlParameter("@PageIndex",
SqlDbType.Int, 4);
            prmPageIndex.Value = PageIndex;

            SqlParameter prmPageSize = new SqlParameter("@PageSize",
SqlDbType.Int, 4);
            prmPageSize.Value = PageSize;

            return DataAccess.GetFromReader("spSelectAllArticleByUser", prmAuthor,
prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
        }

        /// <summary>
        /// Returns all article commented by user.
        /// </summary>
        public IDataReader GetAllArticleCommentedByUser(string Author, int
OrderBy, int SortBy, int PageIndex, int PageSize)
        {
            SqlParameter prmAuthor = new SqlParameter("@FindByAuthor",
SqlDbType.VarChar, 20);
            prmAuthor.Value = Author;

            SqlParameter prmOrderBy = new SqlParameter("@OrderBy", SqlDbType.Int,
4);
            prmOrderBy.Value = OrderBy;

            SqlParameter prmSortBy = new SqlParameter("@SortBy", SqlDbType.Int,
4);
            prmSortBy.Value = SortBy;

            SqlParameter prmPageIndex = new SqlParameter("@PageIndex",
SqlDbType.Int, 4);
            prmPageIndex.Value = PageIndex;

            SqlParameter prmPageSize = new SqlParameter("@PageSize",
SqlDbType.Int, 4);
            prmPageSize.Value = PageSize;

            return DataAccess.GetFromReader("spSelectGetUserArticleCommentByUser",
prmAuthor, prmOrderBy, prmSortBy, prmPageIndex, prmPageSize);
        }

        #endregion
        /// <summary>
        /// Returns article detail.
        /// </summary>
        /// <returns></returns>
        public IDataReader GetArticleDetail(int ID, int Show)
        {
            SqlParameter prmID = new SqlParameter ("@AID", SqlDbType.Int, 4);
            prmID.Value = ID;

            SqlParameter prmShow = new SqlParameter ("@Show", SqlDbType.Int, 4);
            prmShow.Value = Show;

            return DataAccess.GetFromReader("GetArticleDetails", prmID, prmShow);
        }
        /// <summary>
        /// Returns an article comments
        /// </summary>
        public IDataReader GetArticleComments(int ID)
        {
            SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmID.Value = ID;

            return DataAccess.GetFromReader("spSelectCommentArticle", prmID);
        }
        //#region Article Comments - Add, select, delete and update
        /// <summary>
        /// Insert article comments
        /// </summary>
        /// <returns></returns>
        public int InsertArticleComment(Comment comment)
        {
            SqlParameter prmCommentID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmCommentID.Value = comment.ID;

            SqlParameter prmAuthor = new SqlParameter("@Author", 
SqlDbType.VarChar, 50);
            prmAuthor.Value = comment.Author;

            SqlParameter prmEmail = new SqlParameter("@Email", SqlDbType.VarChar, 
50);
            prmEmail.Value = comment.Email;

            SqlParameter prmComment = new SqlParameter("@Comments", 
SqlDbType.VarChar, 350);
            prmComment.Value = comment.Comments;

            SqlParameter prmUserID = new SqlParameter("@UserID", SqlDbType.Int, 
4);
            prmUserID.Value = comment.UID;

            return DataAccess.Execute("spInsertArticleComment", prmCommentID, 
prmAuthor, prmEmail, prmComment, prmUserID);
        }

        /// <summary>
        /// Update article comment
        /// </summary>
        /// <returns></returns>
        public int UpdateArticleComment(Comment comment)
        {
            SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmID.Value = comment.ID;

            SqlParameter prmComment = new SqlParameter("@Comment", 
SqlDbType.VarChar, 350);
            prmComment.Value = comment.Comments;

            return DataAccess.Execute("UpdateArticleComments", prmID, prmComment);
        }

        /// <summary>
        /// Delete individual article comment
        /// </summary>
        /// <returns></returns>
        public int DeleteArticleComment(Comment comment)
        {
            SqlParameter prmID = new SqlParameter("@ID", SqlDbType.Int, 4);
            prmID.Value = comment.ID;

            SqlParameter prmAID = new SqlParameter("@AID", SqlDbType.Int, 4);
            prmAID.Value = comment.RECID;

            return DataAccess.Execute("AdminDeleteArticleComments", prmID, 
prmAID);
        }

        /// <summary>
        /// Delete multiple/batch article comment
        /// </summary>
        /// <returns></returns>
        public int DeleteMultipleArticleComment(string CsvID, string CsvItemID)
        {
            SqlParameter prmCsvID = new SqlParameter("@CsvID", SqlDbType.VarChar, 
1000);
            prmCsvID.Value = CsvID;

            SqlParameter prmCsvItemID = new SqlParameter("@CsvItemID", 
SqlDbType.VarChar, 1000);
            prmCsvItemID.Value = CsvItemID;

            return DataAccess.Execute("AdminDeleteArticleComments", prmCsvID, 
prmCsvItemID);
        }
    }

}
```