

# Developing a Survey Application (Part 1)

By Bipin Joshi

LANGUAGES: VB.NET  
ASP.NET VERSIONS: 2.0

In today's competitive environment companies find it necessary to get feedback on their products and services. Companies also need to know the market trends, customer likings and dislikings. Often companies capture this vital information by conducting surveys about their products and services. A flexible web based survey system plays a major role in such scenarios. In this series of articles we are going to develop a survey application using ASP.NET 2.0, VS.NET 2005 and SQL Server 2005. The Part 1 discusses the functional requirements, database schema and creating survey definitions.

## ***Functional Requirements***

A survey application captures various pieces of information from the end user and needs to be flexible and easy to use. A survey consists of questions and feedback or answers to those questions. The answers can be objective type or in the form of comments. The application that we are going to develop will satisfy the following functional requirements:

- The administrator should be able to create a survey definition in the system
- Each survey consists of a set of questions. There should be a web pages that allows for addition/modification/deletion of questions and their possible choices
- The answer to a question can be single choice, multi choice or free text
- Depending on the answer type the end user should be presented with radio buttons, checkboxes or multi line textboxes at the time of taking the survey
- Each survey has certain participants. There should be a web page for managing participants and their email address.
- Once a survey is ready with its questions and participants the administrator should be able to send an invitation to all the participants via email. The invitation will consists of a URL where in the user can take the survey
- Once the end user goes to the survey URL he should see all the questions from the survey along with their choices
- Once the user submits the survey the answers should be saved for generating reports
- The administrator should be able to see the survey statistics for every objective type (single choice or multi choice) question
- The administrative pages should be protected from unauthorized users

## ***ASP.NET Feature Utilization***

We will develop this application using ASP.NET 2.0, ADO.NET 2.0 and SQL Server 2005 Express Edition. Specifically we will use the following features:

- SQL Data Source control
- Data bound controls - GridView, DetailsView and DataList
- Membership features for securing administrative pages
- TreeView control for navigation
- Sending emails
- ADO.NET (SQL Server Data Provider) objects to execute queries wherever required

## ***Creating Web Application***

To begin, create a new ASP.NET Web site using Visual Studio .NET or Visual Web Developer. Figure 1 shows the new “New Web Site” dialog box of Visual Studio .NET.

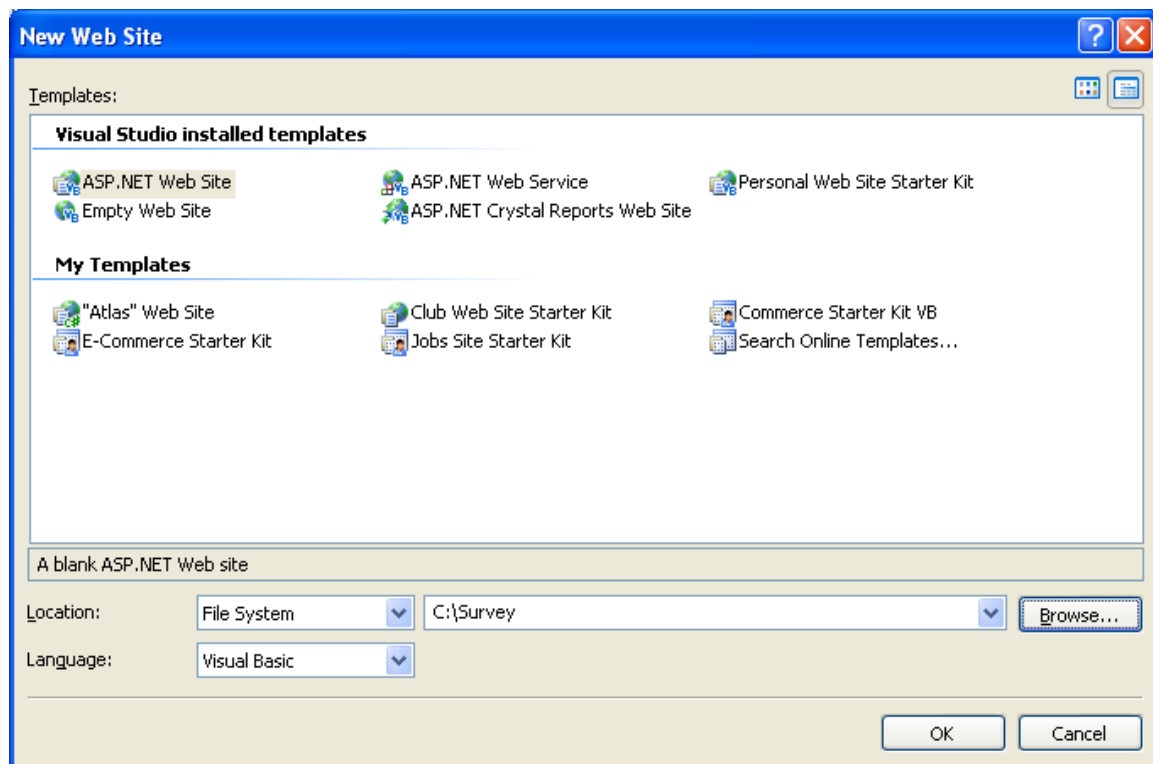


Figure 1: Creating a new Web site in VS.NET

Make sure to choose language as Visual Basic and give the name as Survey. Once the web site is created, create a folder called Admin inside the web site root folder. All the administrative pages will be stored in this folder. The other pages accessible to all the users will be stored directly under the root folder.

## Database Schema

All the data of the application will be stored in a SQL Server 2005 database. In order to add a new database to the web site right click on the web site, select “Add New Items...” and then select SQL Database from the dialog. Give the database name as Database1.mdf. Once you have database in your web site you can then create tables. Figure 2 shows the schema of various tables and Figure 3 lists them along with their purpose.

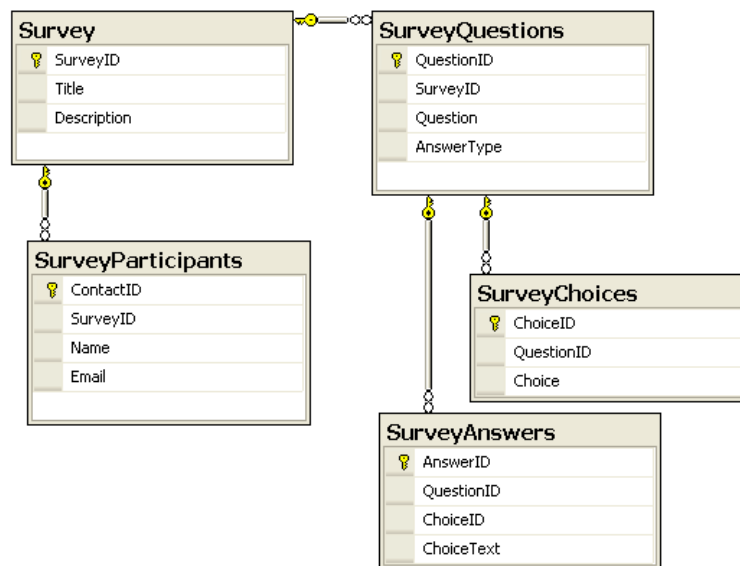


Figure 2: Database schema

Table Name	Description
Survey	This table stores the survey definition in the form of SurveyID, Name and Description.
SurveyQuestions	This table contains list of questions a survey. It also specifies the answer type – Single choice (S), Multi choice (M) or Text (T) for each question.
SurveyChoices	This table contains possible choices for single choice and multiple choice questions.
SurveyAnswers	Once the user takes a survey the answers to single choice, multi choice questions as well as free text answers are stored in this table
SurveyParticipants	This table contains a list of participants for a survey along with their name and email address.

Figure 3: Table description

### ***Web Forms and Master Pages***

In all we need to develop seven web forms and one master page. Figure 4 lists all of them with purpose of each.

<b>Web Form/Master Page</b>	<b>Folder</b>	<b>Description</b>
ManageSurvey.aspx	Admin	This web form allows the administrator to create, edit and delete survey definition. It also has a facility to send invitation to all the participants of a survey.
ManageQuestions.aspx	Admin	This web form is used to add, edit and delete questions belonging to a survey. Here, you also specify the answer type for each question.
ManageChoices.aspx	Admin	This web form allows the administrator to add, edit and delete choices for each question. Note that choices are applicable only for single and multiple choice questions and not for questions with free text answers.
ManageParticipants.aspx	Admin	This web form allows the administrator to add, edit or delete participants of a survey. The name and email of each participant is entered via this web form. The information gathered via this web form is used while sending survey invitations.
SurveyStats.aspx	Admin	The administrator would be interested to know the collective result of the surveys. This web forms displays such statistics. Only single choice and multi choice type of questions are considered for this purpose.
Login.aspx	Root	The administrative pages are accessible only to the administrator. This page allows the administrator to login to the system.
Survey.aspx	Root	This web form takes a survey ID as a query string parameter and displays questions for that survey to the end user. Depending on the

		answer type of a question it renders RadioButtons, CheckBoxes or TextBoxes. The URL to this web form is sent to the participants as a part of the email invitation.
AdminMasterPage.master	Admin	This master page represents the master page of all the administrative web forms. It presents a TreeView with links to all the administrative web forms.

Figure 4: List of web forms and master pages

### ***Creating the Master Page***

All the administrative pages use a master page called MasterPage.master. This master page provides the navigation tree and overall layout to the other pages. In order to create this master page, add a new master page in the Admin folder of the web site using “Add New Item...” dialog (Figure 5).

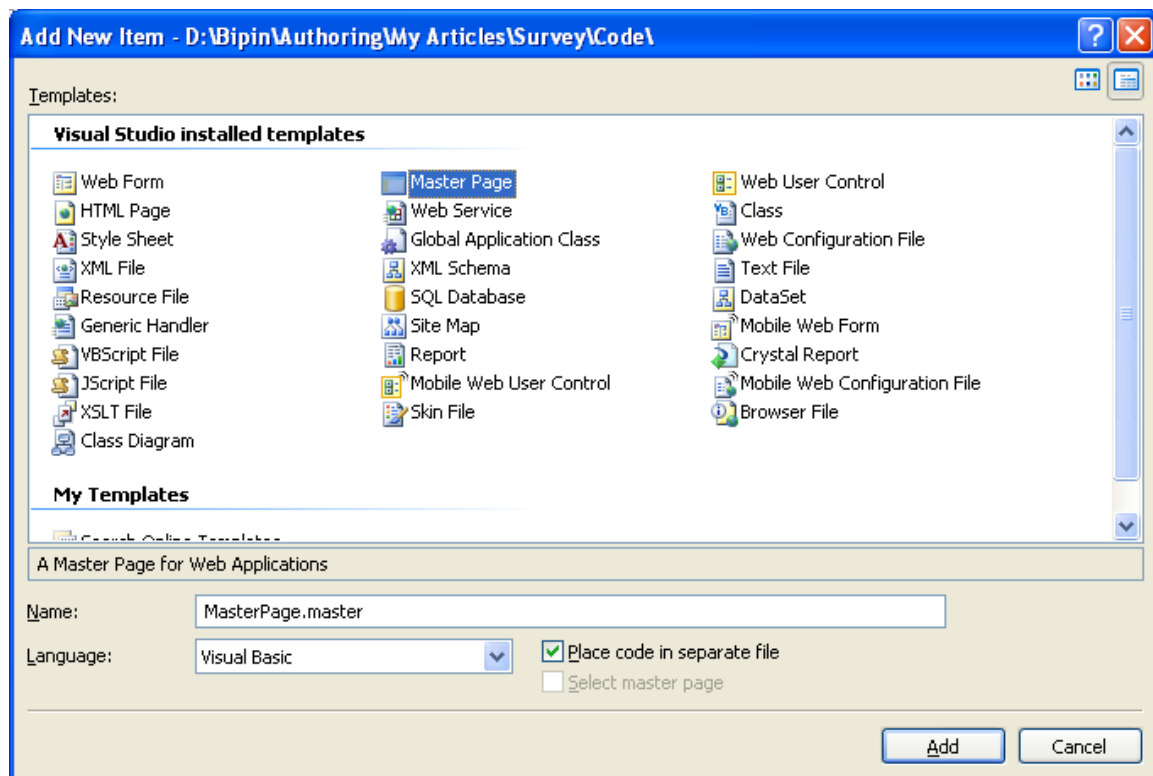


Figure 5: Adding a master page

Figure 6 shows the master page in design mode.

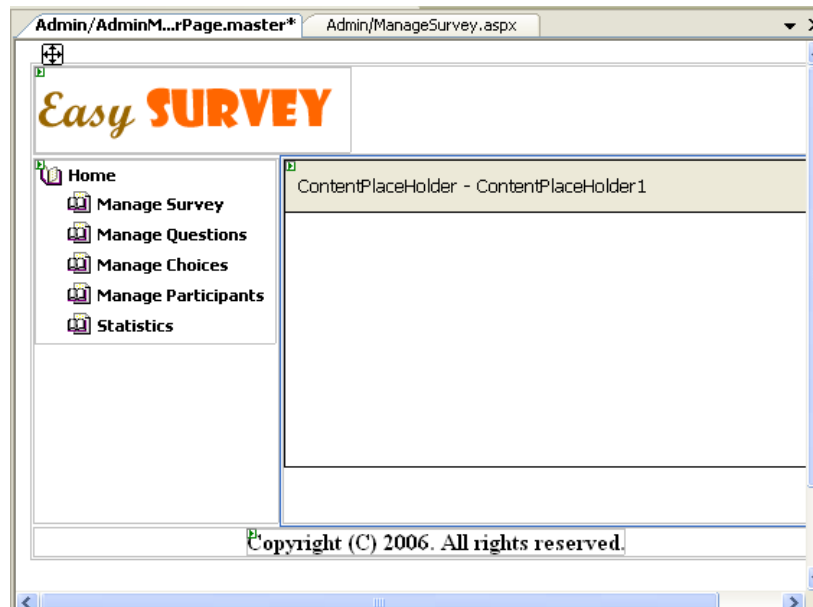


Figure 6: Master page in design mode

The master page consists of a TreeView control with nodes pointing to other administrative pages. To design the master page drag and drop a TreeView control on the master page. Choose “Edit Nodes...” from its smart tag to open TreeView Node Editor (Figure 7). Then add nodes to the TreeView so as to provide navigation structure for all the administrative pages. Set Text and NavigateUrl properties of various nodes as shown in Figure 8.

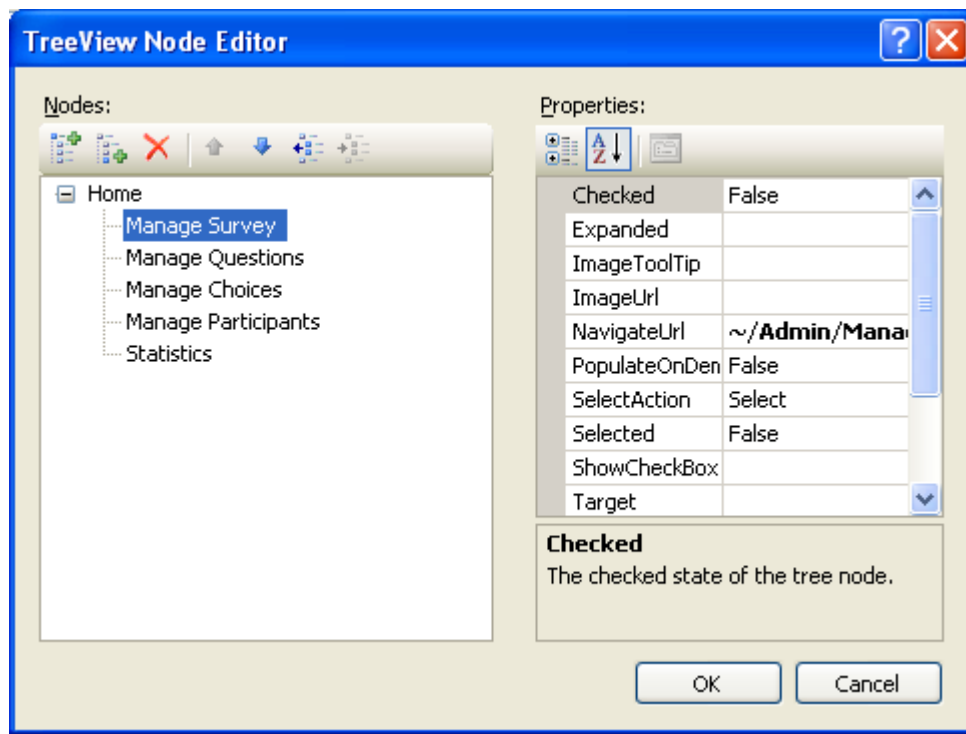


Figure 7: TreeView Node Editor

TreeView Node	Text	NavigateUrl
Manage Survey	Manage Survey	~/Admin/managesurvey.aspx
Manage Questions	Manage Questions	~/Admin/managequestions.aspx
Manage Choices	Manage Choices	~/Admin/managechoices.aspx
Manager Participants	Manager Participants	~/Admin/manageparticipants.aspx
Survey statistics	Statistics	~/Admin/surveystats.aspx

Figure 8 : Properties of various TreeView nodes

Figure 9 shows the complete markup of the master page.

```
<%@ Master Language="VB"
CodeFile="AdminMasterPage.master.vb"
Inherits="Admin_AdminMasterPage" %>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table width="100%">
<tr>
<td colspan="2">
<asp:HyperLink ID="HyperLink1" runat="server"
ImageUrl="~/Images/logo.gif"></asp:HyperLink></td>
</tr>
<tr>
<td valign="top" width="10%">
<asp:TreeView ID="TreeView1" runat="server" Font-
Bold="True" ImageSet="WindowsHelp">
<ParentNodeStyle Font-Bold="False" />
<HoverNodeStyle Font-Underline="True" ForeColor="#6666AA"
/>
<SelectedNodeStyle BackColor="#B5B5B5" Font-
Underline="False" HorizontalPadding="0px"
VerticalPadding="0px" />
<Nodes>
<asp:TreeNode Text="Home" Value="Home">
<asp:TreeNode Text="Manage Survey" Value="Manage Survey"
NavigateUrl="~/Admin/ManageSurvey.aspx"></asp:TreeNode>
<asp:TreeNode Text="Manage Questions" Value="Manage
Questions"
NavigateUrl="~/Admin/ManageQuestions.aspx"></asp:TreeNode>
```

```

<asp:TreeNode Text="Manage Choices" Value="Manage Choices"
NavigateUrl="~/Admin/ManageChoices.aspx"></asp:TreeNode>
<asp:TreeNode Text="Manage Participants" Value="Manage
Participants"
NavigateUrl="~/Admin/ManageParticipants.aspx"></asp:TreeNod
e>
<asp:TreeNode NavigateUrl="~/Admin/SurveyStats.aspx"
Text="Statistics" Value="Statistics">
</asp:TreeNode>
</asp:TreeNode>
</Nodes>
<NodeStyle Font-Names="Tahoma" Font-Size="8pt"
ForeColor="Black" HorizontalPadding="5px"
NodeSpacing="0px" VerticalPadding="1px" />
</asp:TreeView>
</td>
<td width="90%">
<asp:contentplaceholder id="ContentPlaceHolder1"
runat="server">
</asp:contentplaceholder>
</td>
</tr>
<tr>
<td align="center" colspan="2">
<asp:Label ID="Label1" runat="server" Font-Bold="True"
Text="Copyright (C) 2006. All rights
reserved."></asp:Label></td>
</tr>
</table>
</div>
</form>
</body>
</html>

```

Figure 9: Markup of Master Page

## ***Managing Surveys***

The first step in a survey campaign is to create a survey. We will create a web form that allows us to add, edit and delete surveys. Proceed by adding a new web form in the Admin folder called Survey.aspx. The overall layout of the web form is shown in Figure 10.



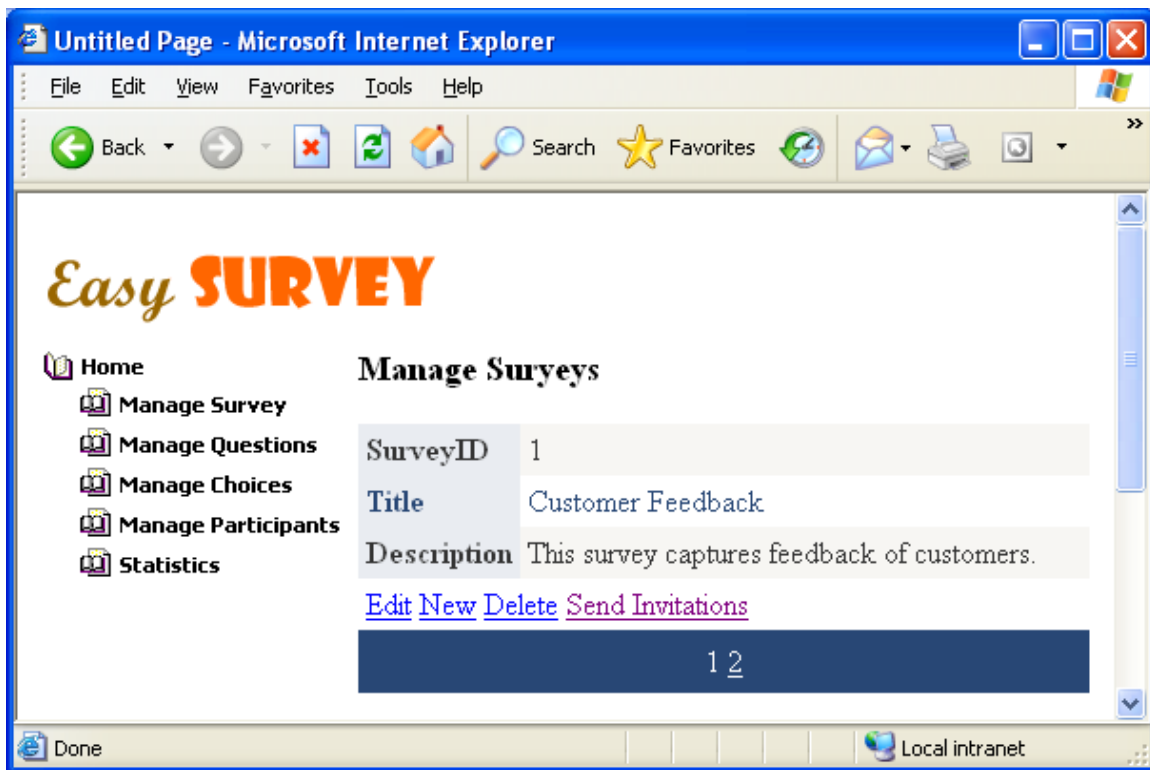


Figure 10: Managing survey definition

In order to design the web form, drag and drop an SQL data source control (SqlDataSource1) on the web form and configure it to select all the records from the Survey table (Figure 11).

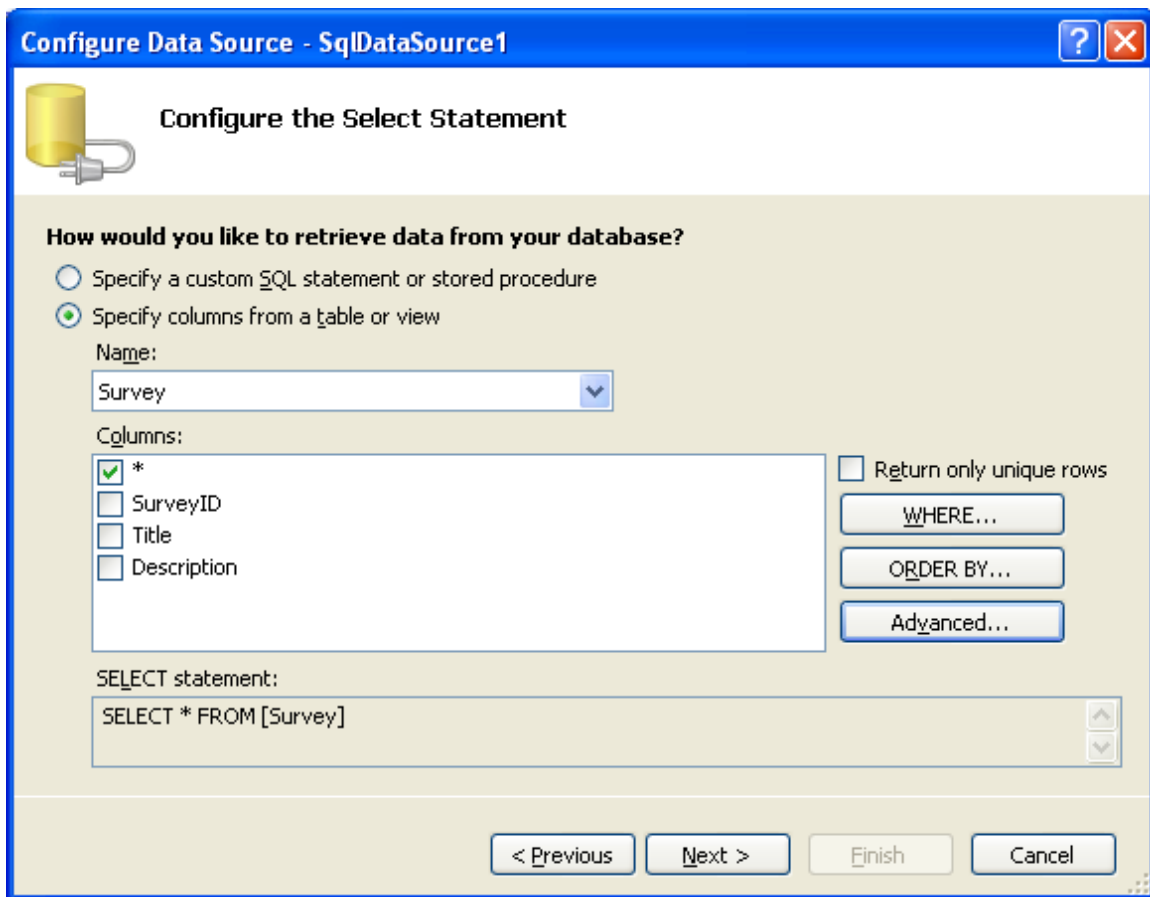


Figure 11: Configuring SQL data source

Then click on the “Advanced...” button and check “Generate INSERT, UPDATE and DELETE statements” checkbox (Figure 12).

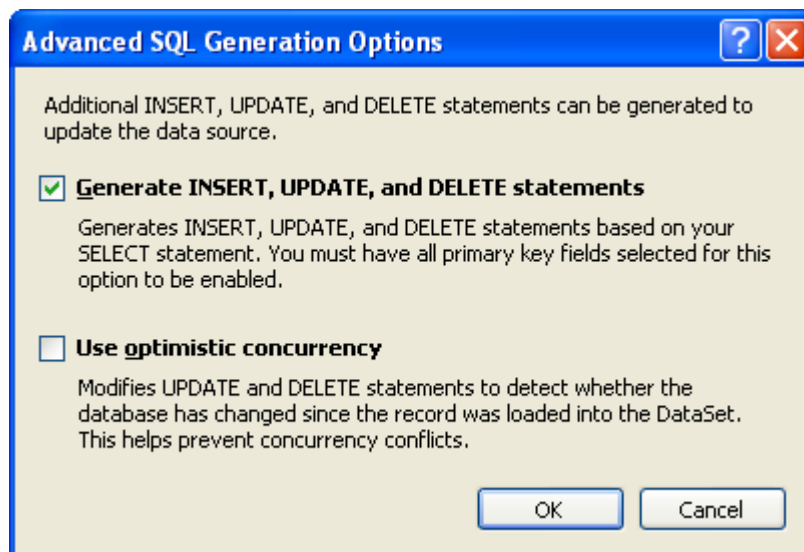


Figure 12: Advanced SQL generation options

Now, drag and drop a DetailsView control on the web form and set its DataSourceID property to SqlDataSource1. From the smart tag of the DetailsView enable paging, inserting, editing and deleting. Note that Figure 10 shows a button titled “Send Invitations” along with Edit, New and Delete buttons. This button is used to send invitations to all the participants of current survey. In order to add this button open the Fields dialog of DetailsView, select the Command Field and click on “Convert this field into a TemplateField” button. This will convert the command field into a template field and we can add the “Send Invitations” button there. Then right click on the DetailsView and select “Edit Template” and then select Field[3]. This will open the template designer as shown in Figure 13.

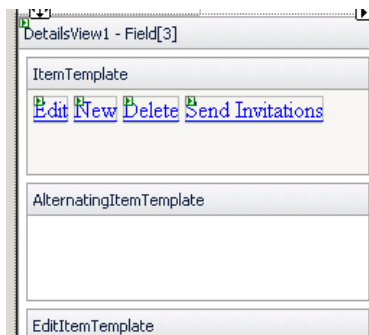


Figure 13: Adding Send Invitations button

Drag and drop a LinkButton in the ItemTemplate beside the Delete button and set its Text property to “Send Invitations”. Also, set its CommandName property to “Invite”. This way we will be able to identify it in our code. Finally drag and drop a Label control below the DetailsView and set its EnableViewState property to False and Text property to an empty string. This label will be used to display a success message after sending the invitations.

Figure 14 shows the complete markup of ManageSurvey.aspx.

```
<%@ Page Language="VB"
MasterPageFile="~/Admin/AdminMasterPage.master"
AutoEventWireup="false" CodeFile="ManageSurvey.aspx.vb"
Inherits="Admin_CreateSurvey" title="Untitled Page" %>
<asp:Content ID="Content1"
ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
<asp:Label ID="Label2" runat="server" Text="<h3>Manage
Surveys</h3>"></asp:Label><asp:DetailsView
ID="DetailsView1" runat="server" AllowPaging="True"
AutoGenerateRows="False"
CellPadding="4" DataKeyNames="SurveyID"
DataSourceID="SqlDataSource1" ForeColor="#333333"
GridLines="None" Height="50px" Width="100%">
<FooterStyle BackColor="#5D7B9D" Font-Bold="True"
ForeColor="White" />
```

```

<CommandRowStyle BackColor="#E2DED6" Font-Bold="True" />
<EditRowStyle BackColor="#999999" />
<RowStyle BackColor="#F7F6F3" ForeColor="#333333" />
<PagerStyle BackColor="#284775" ForeColor="White"
HorizontalAlign="Center" />
<Fields>
<asp:BoundField DataField="SurveyID" HeaderText="SurveyID"
InsertVisible="False"
ReadOnly="True" SortExpression="SurveyID" />
<asp:BoundField DataField="Title" HeaderText="Title"
SortExpression="Title" />
<asp:TemplateField HeaderText="Description"
SortExpression="Description">
<EditItemTemplate>
<asp:TextBox ID="TextBox1" runat="server" Text='<%#
Bind("Description") %>' TextMode="MultiLine"></asp:TextBox>
</EditItemTemplate>
<InsertItemTemplate>
<asp:TextBox ID="TextBox1" runat="server" Text='<%#
Bind("Description") %>' TextMode="MultiLine"></asp:TextBox>
</InsertItemTemplate>
<ItemTemplate>
<asp:Label ID="Label1" runat="server" Text='<%#
Bind("Description") %>'></asp:Label>
</ItemTemplate>
</asp:TemplateField>
<asp:TemplateField ShowHeader="False">
<EditItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server"
CausesValidation="True" CommandName="Update"
Text="Update"></asp:LinkButton>
<asp:LinkButton ID="LinkButton2" runat="server"
CausesValidation="False" CommandName="Cancel"
Text="Cancel"></asp:LinkButton>
</EditItemTemplate>
<InsertItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server"
CausesValidation="True" CommandName="Insert"
Text="Insert"></asp:LinkButton>
<asp:LinkButton ID="LinkButton2" runat="server"
CausesValidation="False" CommandName="Cancel"
Text="Cancel"></asp:LinkButton>
</InsertItemTemplate>
<ItemTemplate>
<asp:LinkButton ID="LinkButton1" runat="server"
CausesValidation="False" CommandName="Edit"
Text="Edit"></asp:LinkButton>

```

```

<asp:LinkButton ID="LinkButton2" runat="server"
CausesValidation="False" CommandName="New"
Text="New"></asp:LinkButton>
<asp:LinkButton ID="LinkButton3" runat="server"
CausesValidation="False" CommandName="Delete"
Text="Delete"></asp:LinkButton>
<asp:LinkButton ID="LinkButton4" runat="server"
CommandName="Invite">Send Invitations</asp:LinkButton>
</ItemTemplate>
</asp:TemplateField>
</Fields>
<FieldHeaderStyle BackColor="#E9ECF1" Font-Bold="True"
Width="20%" />
<HeaderStyle BackColor="#5D7B9D" Font-Bold="True"
ForeColor="White" />
<AlternatingRowStyle BackColor="White" ForeColor="#284775"
/>
</asp:DetailsView>
<asp:Label ID="Label3" runat="server"
EnableViewState="False" Font-Bold="True"
ForeColor="Red"></asp:Label><br />

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%= $ConnectionStrings:ConnectionString
%>"
DeleteCommand="DELETE FROM [Survey] WHERE [SurveyID] =
@SurveyID" InsertCommand="INSERT INTO [Survey] ([Title],
[Description]) VALUES (@Title, @Description)"
SelectCommand="SELECT * FROM [Survey]"
UpdateCommand="UPDATE [Survey] SET [Title] = @Title,
[Description] = @Description WHERE [SurveyID] = @SurveyID">
<DeleteParameters>
<asp:Parameter Name="SurveyID" Type="Int32" />
</DeleteParameters>
<UpdateParameters>
<asp:Parameter Name="Title" Type="String" />
<asp:Parameter Name="Description" Type="String" />
<asp:Parameter Name="SurveyID" Type="Int32" />
</UpdateParameters>
<InsertParameters>
<asp:Parameter Name="Title" Type="String" />
<asp:Parameter Name="Description" Type="String" />
</InsertParameters>
</asp:SqlDataSource>
</asp:Content>

```

Figure 14: Markup of ManageSurvey.aspx

## **Conclusion**

In the Part 1 of this three part series we gathered the functional requirements for the survey application and designed the database schema for the same. We also began developing administrative web forms by creating a master page and a web form for managing survey definitions. In Part 2 of this series we are going to develop web forms for managing survey questions and their choices. We will also develop a web form for managing survey participants.

*Bipin Joshi is the founder and owner of BinaryIntellect Consulting ([www.binaryintellect.com](http://www.binaryintellect.com)) where he conducts professional training programs on .NET technologies. He is the author and publisher of Developer's Guide to ASP.NET 2.0 and co-author of three WROX press books on .NET 1.x. He runs two community web sites dedicated to .NET articles and code samples - [www.dotnetbips.com](http://www.dotnetbips.com), and [www.binaryintellect.net](http://www.binaryintellect.net). He is a Microsoft MVP, member of ASPInsiders, MCAD and MCT. When away from computers he spends time practicing Yoga and meditation. You can contact him via his blog at [www.bipinjoshi.com](http://www.bipinjoshi.com) where he writes about life, Yoga and spirituality*