**Source Code**

**Reference** [**URL:-**](URL:-)Took reference from the following source code

Used camera and bar code scanner Technologies

[**https://github.com/vgz8b/GiftApp**](https://github.com/vgz8b/GiftApp)

**Photo Capture:-**

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="PhotoCapture.aspx.cs" “Inherits="challenge.PhotoCapture" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head id="Head1" runat="server">

<title></title>

<style type="text/css">

.button

{

border:1px solid #ffffff;

border-radius:3px 3px 3px 3px;

color:#FFFFFF;

background-color:#000000;

width:88px;

}

</style>

</head>

<body style="overflow:hidden; max-height:400px; max-width:350px; background-color:#000000; color:#FFFFFF;">

<form id="form1" runat="server">

<div>

<table><tr><td valign="top">

<!-- First, include the JPEGCam JavaScript Library -->

<script type="text/javascript" src="webcam.js"></script>

<script type="text/javascript">

webcam.set\_api\_url('../uploadimges.aspx');

webcam.set\_quality(90); // JPEG quality (1 - 100)

webcam.set\_shutter\_sound(true); // play shutter click sound

webcam.set\_hook('onComplete', 'my\_completion\_handler');

function do\_upload() {

// upload to server

document.getElementById('<%=upload\_results.ClientID%>').innerHTML = '<h1>Yükleniyor...</h1>';

webcam.upload();

}

function my\_completion\_handler(msg) {

// extract URL

if (msg.match(/(http\:\/\/\S+)/)) {

var image\_url = RegExp.$1;

// show JPEG image in page

document.getElementById('<%=upload\_results.ClientID%>').innerHTML =

'<h1>Resim Başarıyla Yüklendi!</h1>' +

'<img src="' + image\_url + '';

// reset camera for another shot

webcam.reset();

}

else alert("Error: " + msg);

}

</script>

<table border="0" cellpadding="0" cellspacing="5">

<tr>

<td valign="top">

<h3 id="tk\_pic" style="margin-left: 110px;">

Capture barcode of product</h3>

<div id="pic\_area">

<table id="Table2" runat="server">

<tr>

<td>

<script type="text/javascript" language="JavaScript">

document.write(webcam.get\_html(320, 240));

</script>

</td>

</tr>

<tr>

<td>

<%--<input type="button" value="Configure..." onclick="webcam.configure()" />--%>

&nbsp;&nbsp;

<input class="button" type="button" value="Çlick" onclick="webcam.freeze()" />

&nbsp;&nbsp;

<input class="button" type="button" value="Upload" onclick="do\_upload()" />

&nbsp;&nbsp;

<input class="button" type="button" value="Reset" onclick="webcam.reset()" />

</td>

</tr>

<tr>

<td>

</td>

</tr>

</table>

</div>

</td>

</tr>

</table>

</td><td width="50">

&nbsp;</td><td valign="top">

</td></tr></table>

<div id="upload\_results" runat="server" style="background-color:#eee;"></div>

</div>

</form>

</body>”

</html>

**Upload Images:-**

using System;

“using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.IO;

using ZXing;

using System.Drawing;

namespace challenge

{

public partial class uploadimges : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

System.Drawing.Image originalimg;

string strFile = DateTime.Now.ToString("dd\_MMM\_yymmss") + ".jpg";

FileStream log = new FileStream(Server.MapPath(strFile), FileMode.OpenOrCreate);

byte[] buffer = new byte[1024];

int c;

while ((c = Request.InputStream.Read(buffer, 0, buffer.Length)) > 0)

{

log.Write(buffer, 0, c);

}

originalimg = System.Drawing.Image.FromStream(log);

originalimg = originalimg.GetThumbnailImage(320, 240, new System.Drawing.Image.GetThumbnailImageAbort(ThumbnailCallback), IntPtr.Zero);

originalimg.Save(Server.MapPath("Images") + "\\" + strFile);

//Write jpg filename to be picked up by regex and displayed on flash html page.

log.Close();

originalimg.Dispose();

File.Delete(Server.MapPath(strFile));

Response.Write("../Images/" + strFile);

Session["ImgUrl"] = "/Images/" + strFile;

Response.Redirect("PhotoCapture.aspx");

}”

public bool ThumbnailCallback() { return false; }

}

}

**Stores:-**

using System;

using System.Collections.Generic;

“using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Xml;

using System.Xml.XPath;

using System.Net;

namespace challenge

{

public partial class stores : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void SearchBtn\_Click(object sender, EventArgs e)

{

rslt.Text = "";

String data = Srch.Text.ToString();

String StoreRequest = CreateRequest(data);

xmldocStoreResponse = MakeRequest(StoreRequest);

ProcessResponse(StoreResponse);

}

public string CreateRequest(string queryString)

{

string UrlRequest = "http://www.supermarketapi.com/api.asmx/ReturnStoresByName?APIKEY=0295cbc332&StoreName=" + queryString;

return (UrlRequest);

}

public XmlDoc MakeRequest(string requestUrl)

{

try

{

HttpReq request = WebRequest.Create(requestUrl) as HttpWebRequest;

HttpResp response = request.GetResponse() as HttpWebResponse;

xmldocxmlDoc = new XmlDocument();

xmlDoc.Load(response.GetResponseStream());

return (xmlDoc);

}

catch (Exception e)

{

rslt.Text = e.Message;

return null;

}

}

public void ProcessResponse(XmlDocument locationsResponse)

{

XmlNamespaceManager nsmgr = new XmlNamespaceManager(locationsResponse.NameTable);

nsmgr.AddNamespace("product", "http://www.SupermarketAPI.com");

//Get formatted addresses: Option 1

//Get all locations in the response and then extract the formatted address for each location

XmlNodeList locationElements = locationsResponse.SelectNodes("//product:Store", nsmgr);

foreach (XmlNode StoreName in locationElements)

{

for (int i = 0; i < StoreName.ChildNodes.Count; i++)

{

rslt.Text = rslt.Text + StoreName.ChildNodes[i].Name + ": " + StoreName.ChildNodes[i].InnerText + "<br/>";

}

rslt.Text = rslt.Text + "<hr/>";”

}

}

}

}

**Products:-**

using System;

using System.Collections.Generic;

“using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Xml;

using System.Xml.XPath;

using System.Net;

namespace challenge

{

public partial class supermarkets : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void SearchBtn\_Click(object sender, EventArgs e)

{

rslt.Text = "";

String data = Srch.Text.ToString();

String StoreRequest = CreateRequest(data);

xmldocStoreResponse = MakeRequest(StoreRequest);

ProcessResponse(StoreResponse);

}

public string CreateRequest(string queryString)

{

string UrlRequest = "http://www.SupermarketAPI.com/api.asmx/COMMERCIAL\_SearchByProductName?APIKEY=0295cbc332&ItemName=" + queryString;

return (UrlRequest);

}

publicxmldocMakeRequest(string requestUrl)

{

try

{

HttpWebReq req = WebRequest.Create(requestUrl) as HttpWebReq;

HttpWebRes res = request.GetResponse() as HttpWebResp;

xmldocxmlDoc = new XmlDocument();

xmlDoc.Load(response.GetResponseStream());

return (xmlDoc);

}

catch (Exception e)

{

rslt.Text = e.Message;

return null;

}

}

public void ProcessResponse(XmlDocument locationsResponse)

{

XmlNamespaceManager nsmgr = new XmlNamespaceManager(locationsResponse.NameTable);”

“ nsmgr.AddNamespace("product", "http://www.SupermarketAPI.com");

//Get formatted addresses: Option 1

//Get all locations in the response and then extract the formatted address for each location

XmlNodeList locationElements = locationsResponse.SelectNodes("//product:Product\_Commercial", nsmgr);

foreach (XmlNode Itemname in locationElements)

{

for (int i = 0; i < Itemname.ChildNodes.Count; i++)

{

if (i == 4)

{

rslt.Text = rslt.Text + "<img src='" + Itemname.ChildNodes[i].InnerText + "'/><br/>";

}

else

{

rslt.Text = rslt.Text + Itemname.ChildNodes[i].Name + ": " + Itemname.ChildNodes[i].InnerText + "<br/>";

}

}

rslt.Text = rslt.Text + "<hr/>";”

}

}

}

}