.NET Bootcamp Summer 16 Class Project

The goal of this project is to create a multi-page, data driven web site using HTML, CSS, JavaScript, SQL Server, C#, and ASP.NET. You will present this project during the last week of class. In the interim, you will have periodic check-ins with the instructor to ensure satisfactory progress toward completion.

Description

You will create an intranet site to implement a purchase request system. This system will allow company employees to submit requests to purchase specific items from specific vendors with approval from management as determined through business rules. The status of each request should be readily visible and allow all users to see where things stand with each request that is in process.

Technical Requirements

The project should include the use of all major technologies presented in the course, including HTML, CSS, JavaScript, T-SQL, C#, ASP.NET or MVC and server side data access. If possible, that data should be exposed via a web service that can be consumed from other applications and sites. Additional requirements are:

1. Use of jQuery to find and manipulate HTML DOM elements
2. Use of a jQuery based visual element - jQuery UI and datatables.net may be good places to start
3. Use of stored procedures for queries should be the standard
4. Consistent look and feel across pages should be custom CSS, bootstrap or master pages
5. A visual studio web site or project that can be zipped up and submitted for evaluation
6. The solution should contain both client side and server side code
7. Screen mockups should exist for at least two pages of the site

Pages

The site will need to contain a minimum of the following pages. Any additional pages desired may also be added to the site. Each page in the site should have a consistent look, feel and navigation. This should include a background image, gradient or color as well as a logo.

1. Home/Default – This page is the landing page in the application. This page should contain global navigation to readily reach other pages in the site and any valuable external resources. This page should have three horizontal sections within a single column. The top should contain the menu/navigation which should at least include links to all pages noted in this document. There should be a footer with disclaimers on the bottom of this page that includes copyright and contact information. In the center,

there should be multiple large icons that allow users to quickly interact with the core functionality of the site. This includes creating a new request, viewing the status of requests and management approval.

2. New Request data entry form – This page should present three sections of content to the user but submit as a single form. Once submitted, a unique purchase request id and a submitted date should be assigned by the system. A success message should be displayed to the user. This page should have the ability to cancel the submission of the request.
    a. Section 1 – Request data. This should allow the entry of a short description, longer justification, the date by which the item is needed, the name of the requestor (which should be stored and automatically presented on subsequent trips to the page), the mode of delivery (pickup or via mail), the phone number and email of the requestor, and a checkbox to indicate whether supporting documents are attached. The one-line description is required, along with a valid phone number and email. A visual cue should be presented to the user when they have left a required field blank or have entered bad data and should go away when fixed.
    b. Section 2 – Vendor Selection. The submitter must select from a list of vendors. Some vendors in the system are pre-approved and others are not. There should be some indication as to which is which. Only one vendor may be selected per request. There should be some static text on this page letting submitters know that it is preferred that they work with pre-approved vendors first.
    c. Section 3 – Item details. The submitter should have the option to select multiple items on each purchase request. As each item is added to the request, a running total should be displayed on the page. This total should be submitted back with the request. The submitter should have the option to either select an item from a catalog that belongs to the selected vendor, or enter details for a brand new item. Only one of these should be possible. If they choose to select from a catalog, they should not be able to manually enter an item in the request. In either case, the item details should include a name, part number, quantity, unit (like package, inch, bucket, etc.), and unit price. Name, quantity and price are required.
3. Request Review page – This page should show a table or grid with all submitted purchase requests. Each request should reveal a status of either submitted, approved or rejected. Any rejected items should appear in red. This grid should be displayed in a box with a white background, with a shadow in the center of the page. If possible, this table should be sortable by submitted date, requested date and name of the item.
4. Vendor Page – This page should have a table or grid that displays the details for all vendors. This should include a 10 digit alphanumeric vendor ID, the vendor's name, a street address, city, state with only two characters (i.e., "KY"), a phone number, email and an indicator as to whether the vendor has been pre-approved. The user should be able to search this table by vendor name or by state and only see the vendors that match the search (one or many). There should be a hyperlink for each vendor that links to their catalog, if they have one.
5. Catalog Items Page – This page should show all items in the catalog on a per vendor basis in a table or grid format. This page should have two columns in the middle section. The right should display the table or grid as described above. The left should display a static list of the top 5 vendors (based on whatever requirements you choose). If possible, the user should be able to click on the item and open a page that shows just that item's details. If possible, a photo of the item (if available) should appear on this page along with the other details.
6. Manager Approval Page – a manager should have the ability to view each pending request individually and either approve or reject it. This page should not allow them to edit the details of the request, but

only see the data and mark it as either approved or rejected and save the item back to the database. This page should only ever show managers requests that need their attention, so it should automatically filter requests to show only those that are pending/submitted and only show requests that are over 50 dollars. Anything under that amount should be automatically approved.

7. If possible, a page should exist to allow the submitter or manager to search for an item on Google and have as a means of comparative shopping and information gathering. This page should display search results or redirect to Google's site. This does not have to integrate programmatically with the rest of the site, but should have the same look and feel.

Data Elements

The following tables should exist in the final solution: Purchase Requests, Request Items, Vendors, Catalogs. Fields for these tables should be collected from the page descriptions above. Any additional tables deemed necessary may be created as well. Each table should have a unique primary key. There will be a minimum set of relationships between Purchase Requests and Vendors, Purchase Requests and Request Items, Vendors and Catalogs. Other relationships will also be needed but may be subject to your interpretation.

General Requirements and Business Rules

1. Even though this is more of an intranet site, it should still have proper description and search engine optimization. Please add the necessary meta tags with keywords and description that tell about each page.
2. Any request over $50 should have manager approval. If the total request is under $50, it is automatically approved and the manager should not have to view it on their page.

Suggested Steps

Lab Day 1 – Create the Visual Studio web site; Develop a backlog of items considering each new topic a sprint, create a mockup for the home page and start creating the home page in HTML

Lab Day 2 – Make UI decisions for your web site, create a CSS stylesheet that all pages can use, apply the style to your home page, start creating the other required pages in the site and build them out as much as possible