Hi,

Below are my answers to the queries,

* **Your resume with complete work history and salary history (in English),**

I am attaching my resume

|  |  |
| --- | --- |
| Companies | Salaries (Monthly in USD) |
| Net2Apps | 1,920.00+ |
| Algorithm Consulting | 2,400.00+ |
| Shams Group | 2,880.00+ |
| Contour Software | 3,600.00+ |

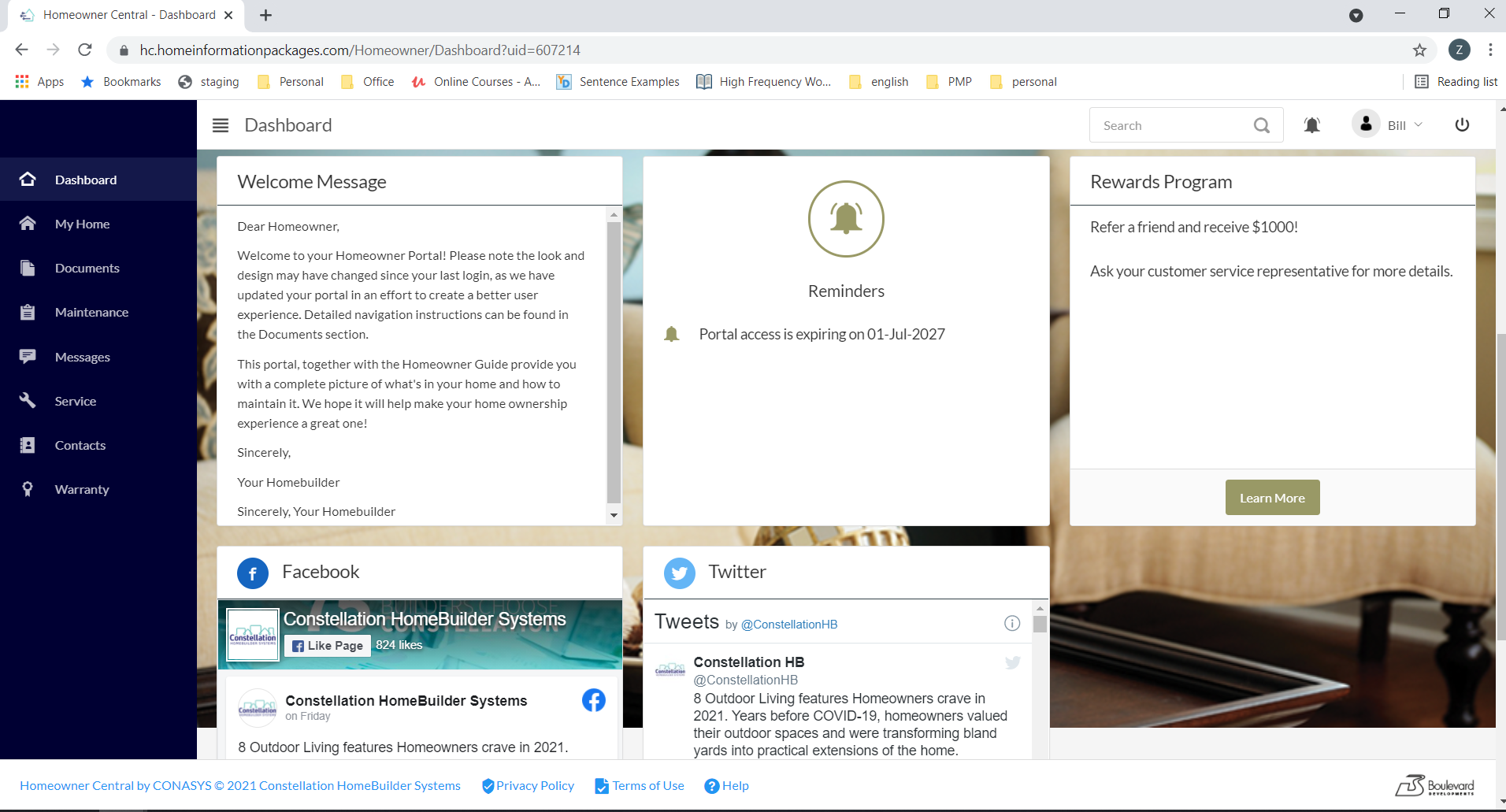
* **A list of programming languages and your level of proficiency in those languages.**

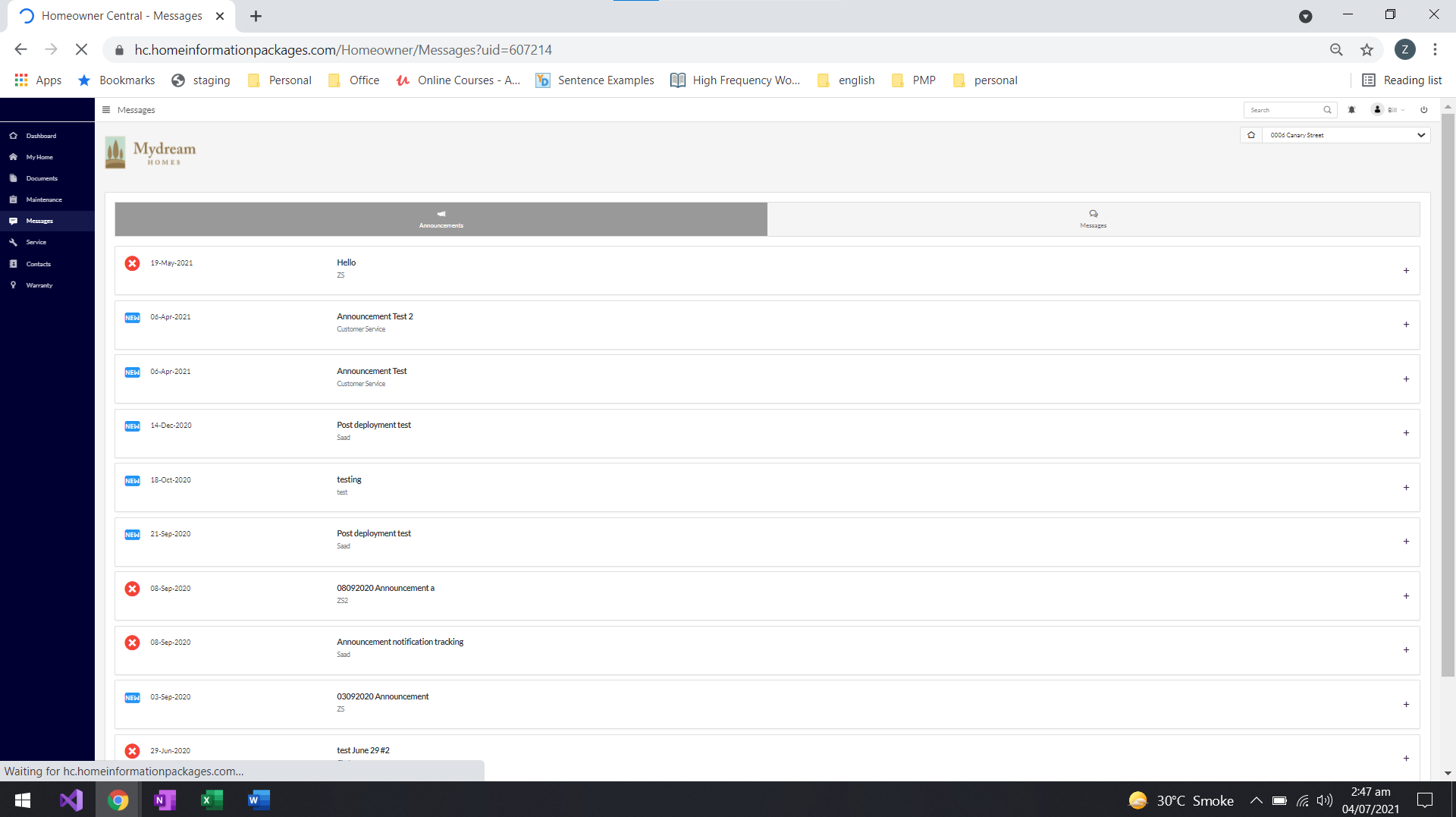
|  |  |
| --- | --- |
| Languages | Proficiency (1 – 10) |
| C# | 10 |
| SQL Server | 10 |
| JavaScript, TypeScript | 10 |
| JQuery | 10 |
| LINQ | 10 |
| HTML5 | 10 |
| CSS | 10 |
| MVC | 10 |
| JSON, Ajax, Bootstrap, WEB API’s, WCF, XML, XSLT, Entity Framework | 10 |
| NodeJS | 8 |
| MySQL, SQL Lite | 7 |
| Angular | 7 |
| VB.net | 7 |
| KnockOutJS, JQueryMobile | 7 |
| Azure | 7 |
| AWS | 5 |
| PHP, ColdFusion, Objective C, C++ | 5 |

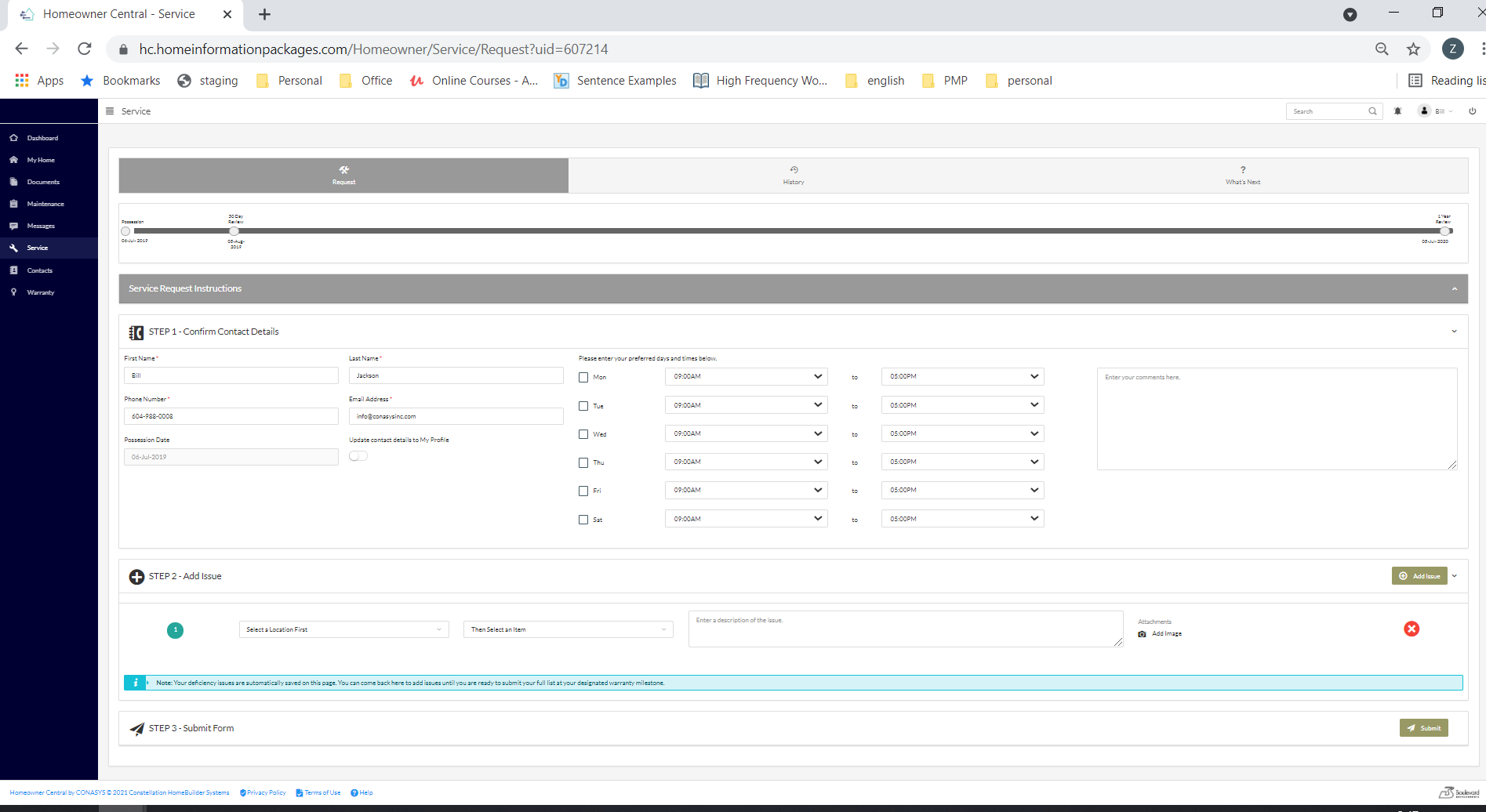
* Code snippets and screenshots of projects that you have done in the past?

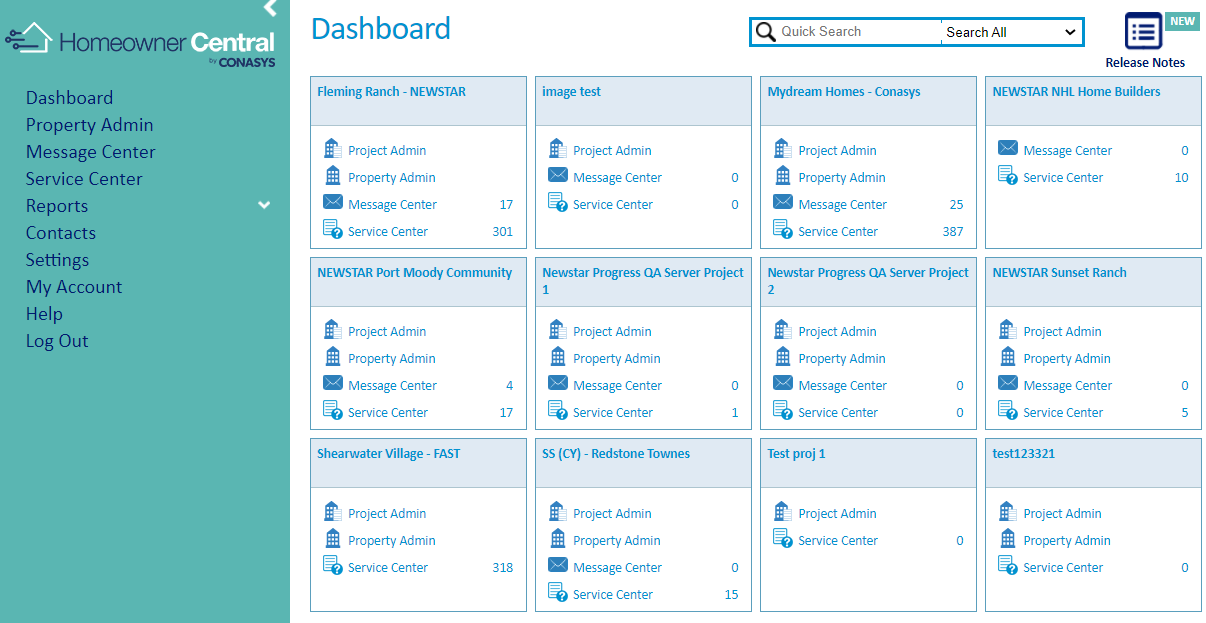
Here is the list of projects been developed, enhanced, and provided support in my job career. Since, most of them are client sensitive, so I can’t share more details about these.

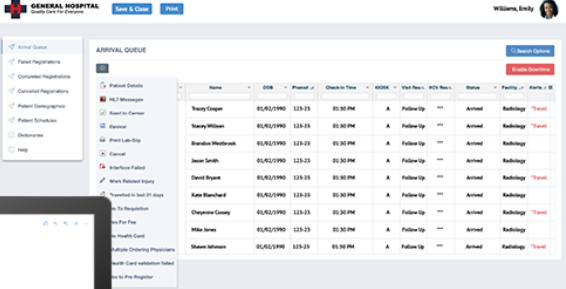
1. warranty management System ([www.homeinformationpackages.com](http://www.homeinformationpackages.com)).
2. Survey Application for buyers (<http://wosvoc.com>),
3. customer connect (<http://Customer-connect.ca>)
4. customer insight (<http://customerinsight.ca/>)
5. 4Tanslation ([www.4translation.com](http://www.4translation.com))
6. Physician Office Solutions
7. Patient Kiosk
8. ezWayFinder
9. EzDictate
10. E-Learning.
11. Inventory Management
12. Import Bond
13. GatePass
14. Finance Module
15. Procurement Module
16. Fippex
17. Net2Invoice
18. Net2Survey



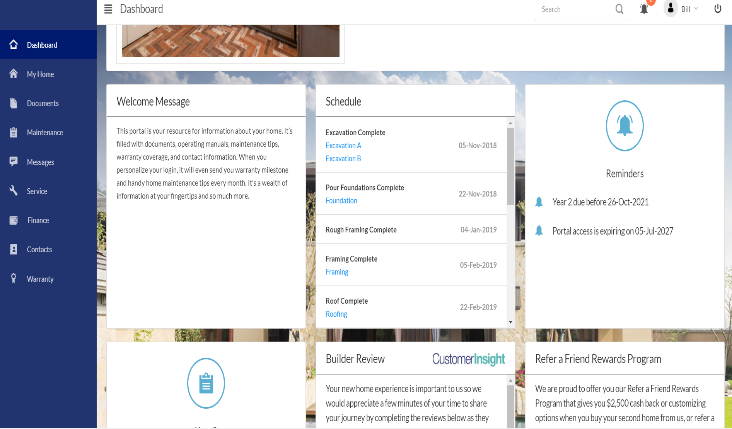


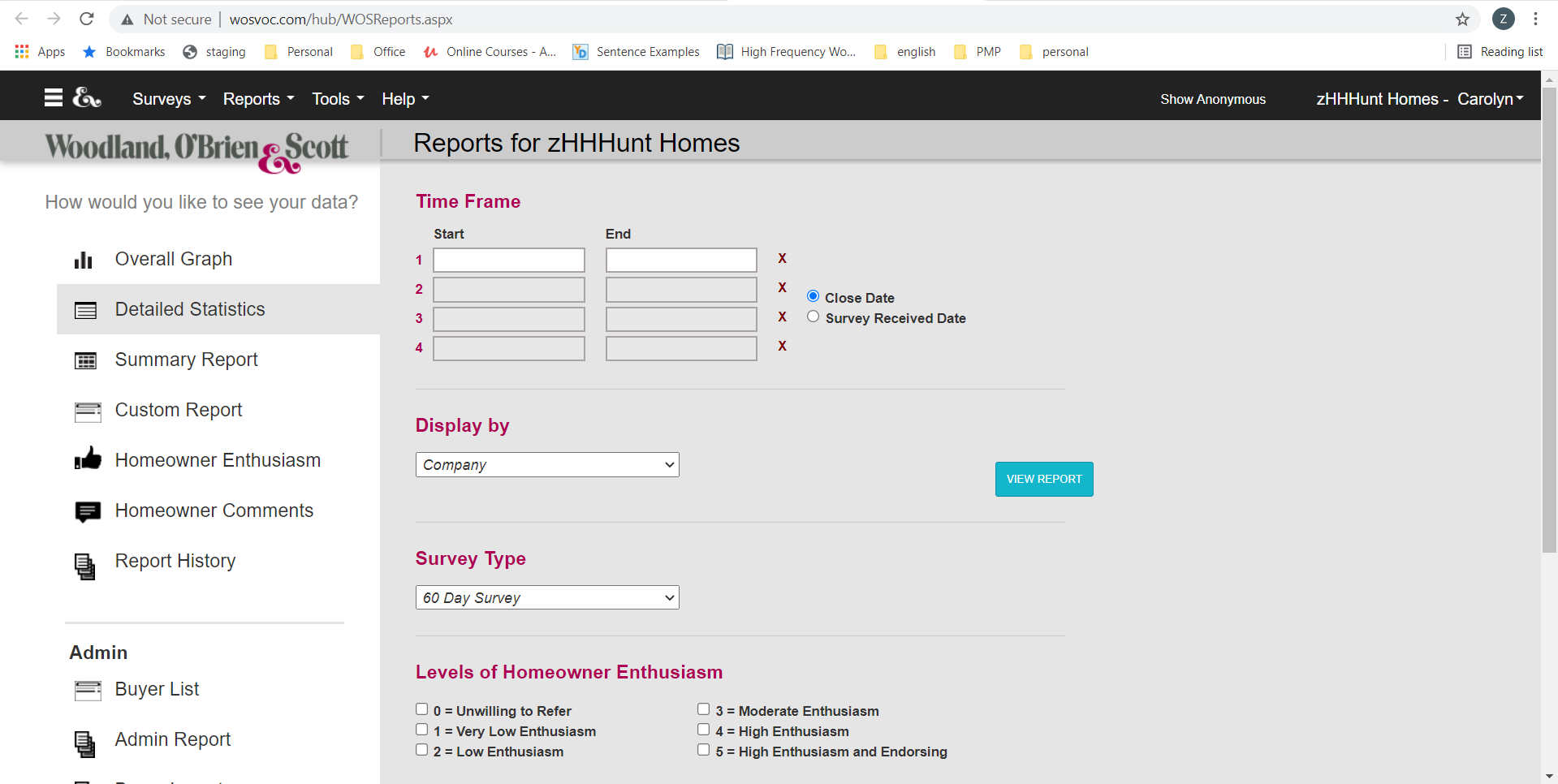


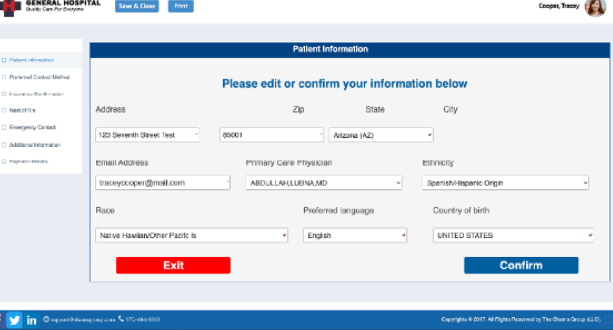
















C# Code Snippets

using HomeOwnerMVC.Properties;

using HomeOwnerMVC.Areas.HomeOwner.Models.ViewModels;

using HomeOwnerMVC.Filters;

using HomeOwnerMVC.HelperRepository.PageHelpers;

using HomeOwnerMVC.Models.Interfaces;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Conasys.HIP.Util.Logging;

using WRObjectModel;

namespace HomeOwnerMVC.Areas.HomeOwner.Controllers

{

    [ExceptionFilter]

    public class ServiceController : BaseHOController

    {

        private ServiceHelper sh;

        private LoggingService \_logger;

        public LoggingService LoggerService

        {

            get { return \_logger ?? (\_logger = LoggingService.GetLoggingService(GetType())); }

        }

        public ServiceController()

        {

            sh = new ServiceHelper(userProfile, hip);

            ViewBag.isServiceProceduresEnabled = hip.CurrentHIP.ServiceProceduresEnabled;

            ViewBag.isServiceRecordEnabled = hip.CurrentHIP.ServiceRecordEnabled;

            ViewBag.isServiceRequestEnabled = hip.CurrentHIP.ServiceRequestEnabled;

            ViewData["MultiServiceRequestId"] = sh.srm.MultiServiceRequestId;

            ViewData["ServiceRequestImageSize"] = Settings.Default.ServiceRequestImageSizeMBs;

        }

        #region Protected

        [ActionName("Request")]

        public ViewResult ServiceRequests()

        {

            int unitId = hip.CurrentHIP.HomeInformationPackageID;

            int userId = userProfile.UserID;

            ServiceRequestModel serviceRequestModel = new ServiceRequestModel();

            serviceRequestModel.isDefaultUser = sh.checkIfIsDefaultUser();

            serviceRequestModel.lstMultiServiceRequest = sh.GetMultiServiceRequestItems().ToList();

            serviceRequestModel.profile = sh.SetProfile();

            serviceRequestModel.serviceRequestMessage = sh.srm.serviceRequestMessage;

            serviceRequestModel.availability = sh.GetAvailability();

            serviceRequestModel.ISAvailabilityOfServiceEnabled = sh.srm.ISAvailabilityOfServiceEnabled;

            serviceRequestModel.srConfiguration = sh.ShowSRByConfiguration();

            serviceRequestModel.ERPType = sh.GetERPType();

            ViewBag.UId = hip.CurrentHIP.HomeInformationPackageID;

            ViewBag.IsHomeOwnerPortalRenewed = hip.IsHomeOwnerPortalRenewed;

            return View("ServiceRequests", serviceRequestModel);

        }

        public PartialViewResult AddNewItem()

        {

            var result = sh.AddNewItem();

            return PartialView("/Areas/HomeOwner/Views/Shared/EditorTemplates/DeficiencyEntryViewModel.cshtml", result);

        }

        public PartialViewResult RefreshGrid()

        {

            var data = sh.GetMultiServiceRequestItems().ToList();

            return PartialView("DeleteMultiServiceRequestItem", data);

        }

        public PartialViewResult DeleteMultiServiceRequestItem(int multiServiceRequestItemId)

        {

            var result = sh.DeleteMultiServiceRequestItem(multiServiceRequestItemId);

            if (result)

            {

                var data = sh.GetMultiServiceRequestItems().ToList();

                return PartialView("DeleteMultiServiceRequestItem", data);

            }

            else

            {

                throw new Exception("Unable to delete this item");

            }

        }

        public JsonResult UpdateMultiServiceRequest(int multiServiceRequestItemId, string locationId, string locationName, string productId, string productName, string description)

        {

            var result = sh.UpdateMultiServiceRequest(multiServiceRequestItemId, locationId, locationName, productId, productName, description);

            return Json(result, JsonRequestBehavior.AllowGet);

        }

        [HttpPost]

        public PartialViewResult AddMultipleServiceRequestIssueImages(string httpPostedFileBaseString, int multiServiceRequestItemId, string fileName, string contentLength)

        {

            var result = sh.AddServiceRequestIssueImages(httpPostedFileBaseString, multiServiceRequestItemId, fileName, contentLength);

            if (result)

            {

                var data = sh.GetMultiServiceRequestItems().ToList();

                return PartialView("DeleteMultiServiceRequestItem", data);

            }

            else

            {

                throw new Exception("Unable to add this image");

            }

        }

        public PartialViewResult RemoveServiceRequestIssueImages(string reviewItemDocId)

        {

            var result = sh.RemoveServiceRequestIssueImages(reviewItemDocId);

            if (result)

            {

                var data = sh.GetMultiServiceRequestItems().ToList();

                return PartialView("DeleteMultiServiceRequestItem", data);

            }

            else

            {

                throw new Exception("Unable to delete this image");

            }

        }

        public PartialViewResult GetAvailability()

        {

            var result = sh.GetAvailability();

            return PartialView("GetAvailability", result);

        }

        [HttpPost]

        [ValidateInput(false)]

        [ActionName("Request")]

        public ViewResult ServiceRequests(ServiceRequestModel srm)

        {

            try

            {

                if (Request.Form["chkAddEditProfile"] != null)

                {

                    srm.addEditProfile = (Request.Form["chkAddEditProfile"].ToString() == "on") ? true : false;

                }

                srm.serviceRequestMessage = sh.srm.serviceRequestMessage;

                srm.ISAvailabilityOfServiceEnabled = sh.srm.ISAvailabilityOfServiceEnabled;

                srm.lstMultiServiceRequest = sh.GetMultiServiceRequestItems().ToList();

                srm.srConfiguration = sh.ShowSRByConfiguration();

                if (hip.CurrentHIP.PossessionDate.HasValue)

                    srm.profile.possessionDate = hip.CurrentHIP.PossessionDate.Value;

                var lstAvailableTimeValues = sh.GetProjectAvailabilityTime();

                srm.ERPType = sh.GetERPType();

                if (srm.availability != null)

                {

                    foreach (var av in srm.availability)

                    {

                        av.AvailableTimeValues = lstAvailableTimeValues;

                    }

                }

                ViewBag.IsHomeOwnerPortalRenewed = hip.IsHomeOwnerPortalRenewed;

                ViewBag.UId = hip.CurrentHIP.HomeInformationPackageID;

                if (ModelState.IsValid)

                {

                    var result = sh.SubmitServiceRequest(srm);

                    srm.srSubmissionResult = result;

                    return View("ServiceRequests", srm);

                }

                else

                {

                    return View("ServiceRequests", srm);

                }

            }

            catch (Exception ex)

            {

                LoggerService.Debug(ex.StackTrace);

                srm.srSubmissionResult = new SRSubmissionResult { IsResult = true, Result = ex.Message };

                return View("ServiceRequests", srm);

            }

        }

        #endregion

        [ChildActionOnly]

        public PartialViewResult TimeLine()

        {

            var timelineData = sh.GetTimeLineData();

            return PartialView(timelineData);

        }

        public JsonResult GetDownloadDeficienciesStatus()

        {

            WarrantyDataContext db = DataContextFactory.GetWarrantyDataContext();

            var srStatus = db.ServiceRequestStatus.ToList().OrderBy(x => x.DisplayOrder).Select(a => new

            {

                a.ServiceRequestStatusID,

                a.Name

            });

            return Json(srStatus, JsonRequestBehavior.AllowGet);

        }

        public void DownloadDeficienciesReport()

        {

            GenerateReportHelper generateReportHelper = new GenerateReportHelper(userProfile, hip);

            generateReportHelper.GenerateReport();

        }

    }

}

Typescript, JQuery ode Snippets

namespace HomeOwnerMVC.HO {

    export class BaseScript {

        public isPostBack: boolean;

        public Page\_Load() {

            this.IconOnMobileFooter();

        }

        public blockUI(obj): void {

            $(obj).block({

                message: '<i class="icon-spinner4 spinner"></i>',

                overlayCSS: {

                    backgroundColor: '#fff',

                    opacity: 0.8,

                    cursor: 'wait'

                },

                css: {

                    border: 0,

                    padding: 0,

                    backgroundColor: 'transparent'

                }

            });

        }

        public unblockUI(obj): void {

            $(obj).unblock({

                message: '<i class="icon-spinner4 spinner"></i>',

                overlayCSS: {

                    backgroundColor: '#fff',

                    opacity: 0.8,

                    cursor: 'wait'

                },

                css: {

                    border: 0,

                    padding: 0,

                    backgroundColor: 'transparent'

                }

            });

        }

        public populateDdl(ddl, data): void {

            data = JSON.parse(data);

            var additionalClass = "";

            var oldValue = "";

            if (ddl.length > 1) {

                additionalClass = ddl.replace("#", ".");

                oldValue = $(ddl).text().trim();

            }

            else {

                additionalClass = ddl.find("id");

                oldValue = $(ddl).text().trim();

                additionalClass = "." + additionalClass;

            }

            $('.selectpicker' + additionalClass).empty();

            $.each(data, function (key, value) {

                if (oldValue == value.Name) {

                    $('.selectpicker' + additionalClass).append($('<option selected="selected"></option>').attr('value', value.Id).text(value.Name));

                }

                else {

                    $('.selectpicker' + additionalClass).append($('<option></option>').attr('value', value.Id).text(value.Name));

                }

            });

            serviceRequestScript.UnBlockUI();

            $(ddl).attr('loaded', 'true');

            $('.selectpicker' + additionalClass).selectpicker('refresh');

        }

        public showAlertOnLoad(msg): void {

            alert(msg);

        }

        public ShowLoaderOnHTTP(): void {

            $('.overlay-loader').removeClass('d-none');

        }

        public HideLoaderOnHTTP(): void {

            $('.overlay-loader').addClass('d-none');

        }

        public IconOnMobileFooter(): void {

            $('.icon-on-footer').click(function () {

                if ($('.icon-on-footer').attr('aria-expanded') == "true") {

                    $('.icon-on-footer i').addClass('icon-circle-up2').removeClass('icon-circle-down2');;

                }

                else {

                    $('.icon-on-footer i').addClass('icon-circle-down2').removeClass('icon-circle-up2');

                }

            });

        }

        public OpenInCurrentTab(fileUrl): void {

            if (fileUrl != '')

                window.open(fileUrl, '\_self');

        }

        public OpenInNewTab(fileUrl): void {

            if (fileUrl != '')

                window.open(fileUrl, '\_blank');

        }

        public setCookie(name, value, days) {

        var expires = "";

        if (days) {

            var date = new Date();

            date.setTime(date.getTime() + (days \* 24 \* 60 \* 60 \* 1000));

            expires = "; expires=" + date.toUTCString();

        }

        document.cookie = name + "=" + (value || "") + expires + "; path=/";

    }

        public getCookie(name) {

        var nameEQ = name + "=";

        var ca = document.cookie.split(';');

        for (var i = 0; i < ca.length; i++) {

            var c = ca[i];

            while (c.charAt(0) == ' ') c = c.substring(1, c.length);

            if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length, c.length);

        }

        return null;

    }

        public eraseCookie(name) {

        document.cookie = name + '=; Max-Age=-99999999;';

        }

        private GoToRenewalPage(unitId) {

            window.location.href = '/Homeowner/RenewPortal?uid=' + unitId;

        }

        public CheckURLforMobile(DesktopURL, MobileURL) {

            var isMobile = baseScript.mobileAndTabletCheck();

            if (isMobile && MobileURL != "")

                baseScript.OpenInCurrentTab(MobileURL);

            else if (isMobile)

                baseScript.OpenInCurrentTab(DesktopURL);

            else

                baseScript.OpenInNewTab(DesktopURL);

        }

}

}

var baseScript: HomeOwnerMVC.HO.BaseScript;

var Sys: any;

if (baseScript == undefined)

    baseScript = new HomeOwnerMVC.HO.BaseScript();

$(document).ready(function () {

    baseScript.Page\_Load();

});

SQL Server Code Snippets

CREATE proc [dbo].[GetConfiguredAPIForCompany]

@CompanyId int

as

select a.APIID,a.APIDetailID,a.CompanyID,a.APIName,a.APIURL,a.APIToken, a.APIKey,a.APIModule,a.AllowSubmit,a.ConfiguredAPIModuleId

from

(

select ca.APIID,cad.APIDetailID,ca.CompanyID,ca.APIName,isnull(cad.APIURL,'') APIURL,isnull(cad.APIToken,'') APIToken, isnull(cad.APIKey,'') APIKey,

isnull(cad.APIModule,'') APIModule,cad.AllowSubmit,cad.ConfiguredAPIModuleId from ConfiguredAPI ca

inner join [ConfiguredAPIDetails] cad

on ca.APIID = cad.APIID

left join ConfiguredAPIModule cam

on cam.ConfiguredAPIModuleId = cad.ConfiguredAPIModuleId

)a

where a.CompanyID  = @CompanyId

Voice Message for below paragraph,

“If you have been a programmer for longer than 2 or 3 years, you have probably been slowed down by messy code. The degree of the slowdown can be significant. Over the span of a year or two, teams that were moving very fast at the beginning of a project can find themselves moving at a snail’s pace. every change they make to the code breaks two or three other parts of the code. no change is trivial. Every addition or modification to the system requires that the tangles, twists, and knots be understood so that more tangles twist and knots can be added. Over time the mess becomes so big and so deep and so tall they cannot clean it up. There is no way at all. “

<https://voca.ro/15X5ZPNg9hZ9>