AUTOMATED TESTING

We used automated testing for unit and functional testing. The testing included a set of test scripts that were used as part of the automated deployment process.

TEST SCRIPTS

```
var myCode = require('../wwwroot/js/Mapfunctions.js');
var assert = require('assert');
var jsdom = require('mocha-jsdom');

//Test data
//Provider name returned in the search result is used to pass/fail tests.
var tc1_data = 'HERNANDO HEADSTART';
var tc5_data = 'ALMA GARCIA';
var tc6_data = '11TH STREET CHILD DEVELOPMENT CENTER';
var tc7_data = 'KARINA MARIE ALVAREZ';
var tc8_data = 'ALLIANCE AFTER SCHOOL CARE AT BONHAM';
var tc9_data = '1.2..ME...CHILDCARE';

describe('Apptest', function(){
    jsdom();

    before(function () {
        $ = require('jquery');
    });

    //Set timeoutvar
    var timeoutvar = 15000;
    //Test Condition 1
    describe('Run Automated Unit Tests for MSRFP-3717', function(){
        it('Test Condition 1. Should return search results with a zip code(38632) only', function(){
            this.timeout(timeoutvar);
            // Call the exported function from the module
            var arr  = [];
                            arr = myCode.LoadMapsByFilters(38632,10,'-1','-1',null,null);
            assert.equal(arr[0][0], tc1_data);
        });
```

```
    it('Test condition 2. Should NOT return search results with a zip code(38632) ,rating(1-Rated) and provider type(6-
Relative In-Home)filter combination', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                    arr = myCode.LoadMapsByFilters(38632,10,'1','6',null,null);
        assert.equal(arr.length, 0);
    });


    it('Test condition 3. Should NOT return search result with invalid address as special character (@) ', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                    arr = myCode.LoadMapsByFilters('@',10,'-1','-1',null,null);
        assert.equal(arr.length, 0);
    });


    it('Test Condition 4. Should return the expected number of records set by the search filter with zip code(38632) and
Number of records(15).', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                    arr = myCode.LoadMapsByFilters('38632',15,'-1','-1',null,null);
        assert.equal(arr.length, 15);
    });


    it('Test Condition 5. Should filter records set by the search filter with zip code(38632) and Provider Type(Group
Home).', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                    arr = myCode.LoadMapsByFilters('38632',10,'-1','3',null,null);
        assert.equal(arr[0][0], tc5_data);
    });


    it('Test Condition 6. Should filter records set by the search filter with zip code(38632) and Provider Type(Group
Home) and Special Needs(YES)', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                    arr = myCode.LoadMapsByFilters('38632',10,'-1','3',null,'YES');
        assert.equal(arr[0][0], tc6_data);
    });
```

```
    it('Test Condition 7. Should filter records set by the search filter with zip code(38632) and Provider Type(Non-
Relative Out of Home) and Availability(YES)', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                        arr = myCode.LoadMapsByFilters('38632',10,'-1','7','YES',null);
        assert.equal(arr[0][0], tc7_data);
    });


    it('Test Condition 8. Should filter records set by the search filter with zip code(38632) and Provider Type(Slot
Contractor) and Rating(Rated)', function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                        arr = myCode.LoadMapsByFilters('38632',10,'1','2',null,null);
        assert.equal(arr[0][0], tc8_data);
    });


    it('Test Condition 9. Should be able to search by a specific address(2391 MEMPHIS STREET HERNANDO 38632)',
function(){
        this.timeout(timeoutvar);
        // Call the exported function from the module
        var arr  = [];
                        arr = myCode.LoadMapsByFilters('2391 MEMPHIS STREET HERNANDO 38632',10,'-1','-
1',null,null);
        assert.equal(arr[0][0], tc9_data);
    });
  });

});
```

```
\ dotnet build
Project MSRFP-3717 (.NETCoreApp,Version=v1.0) will be compiled because project is not safe for incremental compilation. Use --build-profile flag for more information.
Compiling MSRFP-3717 for .NETCoreApp,Version=v1.0
[16:39:27] Using gulpfile C:\Shibu\Cambria\Git\MSRFP-3717\gulpfile.js
[16:39:27] Starting 'run-tests'...
  Apptest
    Run Automated Unit Tests for MSRFP-3717
      ΓêÜ Test Condition 1. Should return search results with a zip code(38632) only (3029ms)
      ΓêÜ Test condition 2. Should NOT return search results with a zip code(38632) ,rating(1-Rated) and provider type(6-Relative In-Home)filter combination (2884ms)
      ΓêÜ Test condition 3. Should NOT return search result with invalid address as special character (@)  (2744ms)
      ΓêÜ Test Condition 4. Should return the expected number of records set by the search filter with zip code(38632) and Number of records(15). (3049ms)
      ΓêÜ Test Condition 5. Should filter records set by the search filter with zip code(38632) and Provider Type(Group Home). (3470ms)
      ΓêÜ Test Condition 6. Should filter records set by the search filter with zip code(38632) and Provider Type(Group Home) and Special Needs(YES) (2853ms)
      ΓêÜ Test Condition 7. Should filter records set by the search filter with zip code(38632) and Provider Type(Non-Relative Out of Home) and Availability(YES) (2818ms)
      ΓêÜ Test Condition 8. Should filter records set by the search filter with zip code(38632) and Provider Type(Slot Conytractor) and Rating(Rated) (3064ms)
      ΓêÜ Test Condition 9. Should be able to search by a specific address(2391 MEMPHIS STREET HERNANDO 38632) (2922ms)
  9 passing (28s)
[16:39:55] Finished 'run-tests' after 28 s
Bundling with configuration from C:\Shibu\Cambria\Git\MSRFP-3717\bundleconfig.json
Processing wwwroot/css/site.min.css
Unexpected token, found '@import'
Unexpected token, found 'url(http://fonts.googleapis.com/css?family=Roboto:400,300)'
Unexpected token, found '@import'
Unexpected token, found 'url(http://netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css)'
Processing wwwroot/js/site.min.js
  Bundled
  Minified

Compilation succeeded.
    0 Warning(s)
    0 Error(s)

Time elapsed 00:00:33.4613317
 (The compilation time can be improved. Run "dotnet build --build-profile" for more information)
```