OhioVoter.org

by

Jerry Bloom and Mat Collins

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Jerry Bloom — Date: 4/12/2017

Mat Collins — Date: 4-12-2017

Professor Russell McMahon — Date: Apr. 12, 2017

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2017

TABLE OF CONTENTS

ABSTRACT

OhioVoter.org provides a central location where Ohio voters can quickly and easily gather important facts about the upcoming election. Currently, voters have to visit multiple websites and navigate through various, sometimes confusing, pages in the hopes of finding all the relevant ballot information. Using the advantages of ASP.NET MVC with C# combined with the reliability and scalability of Amazon AWS cloud service, Ohio voters will have a stable environment ensuring uninterrupted access to candidate platforms and current legislation topics. Voters will also have the opportunity to setup a personalized account allowing them to tailor the information displayed. For those voters who do not want to create an account, they can still signup to receive emailed notifications about upcoming important dates. Voters can now enter the voting booth with the confidence they are well prepared as they cast their vote on Election Day.

1. PROBLEM STATEMENT

1.1 Introduction

Ohio is one of the most populated states in the United States and arguably the most important state when it comes to judging the political climate for the country. The voters of Ohio have proven their accuracy of selecting the winning presidential candidate for many years. This consistency has elevated the status of Ohio to be known nationally as a must win state for all presidential candidates. Our goal is to provide the Ohio voting community with easy access to all the necessary information so they can be well educated before heading out to vote on Election Day.

1.2 Project Description

We will be creating a new voter information website called OhioVoter.org targeting the Ohio voting community. This website will provide a central location where Ohio voters can easily view and learn about everything that will be on their ballot for the upcoming election. Voters will also have the ability to create an account so they can customize the information displayed. Candidates will have the ability to update or correct the information displayed on the website. This will add an extra level of accuracy to the information presented to the voters.

1.3 Problem

The Ohio voting community currently does not have an easy way to view and learn about all the possible candidates and ballot measures they can vote for on Election Day. Some websites claim to be a one-stop-shop for all this information. However, with twenty-three possible candidates

running for president on the November 2017 Ohio ballot, we were unable to find one website that provided an easy way to learn about all of them. We had a similar problem with locating information about all the ballot measures. Although there are websites that provide a list of all of them, we were unable to find one website that provided all the details about every ballot measure.

1.4 User Profile

The target users for the Ohio Voter Information Website are the voters, candidates, and systems administrators planning to participate in the upcoming election process. All users should be familiar with browsing websites. The voters will be able to view and compare information on candidates, local issues, and current political news. The candidates will be able to view information about their platforms and submit new or updated facts to our system administrators. The system administrators will be able to modify information displayed on the website and manage the web servers through cloud services. See Table 1. User Profile below for more information.

Application:
OhioVoter.org -- Ohio Voter Information Website
Potential Users
Voters looking up information about the candidates and issues that will be on the ballot. Candidates wanting to update information displayed about their campaign for the upcoming election. System administrators wanting to updated website features, information in the database, or network infrastructure.

Software and Interface Experience Users will be familiar with setting up an online account, logging into their account, and updating information.
Experience with Similar Applications Voters and candidates will have general experience using hyperlinks to navigate websites and interacting with common objects found on forms. System administrators will have a high level of experience with website development and implementing cloud services.
Task Experience Voters and candidates will have a basic knowledge for using a computer or smart phone to access and interact with websites through a web browser. System administrators will be able to update information displayed on the website and maintain the operation of the cloud services.
Frequency of Use The website will see an increase in traffic from users during each election season leading up to Election Day.
Key Interface Design Requirements That the Profile Suggests Voters and candidates have different requirements that will influence the design for their different interfaces. The design of the website is tailored to the requirements of the voter. A separate interface for the candidates will provide a similar design making it easy for updating their campaign information.

Table 1. User Profile

1.5 Use Case

Voters will use our website to learn about the candidates and issues that will be on the ballot for the upcoming election. Candidates will use our website to make sure the information about their

campaign is correct. Both will use our website to create personal accounts and stay current on other political news. The IT department will access and update various components of the website. View Figure 1. Use Case Diagram for more information.

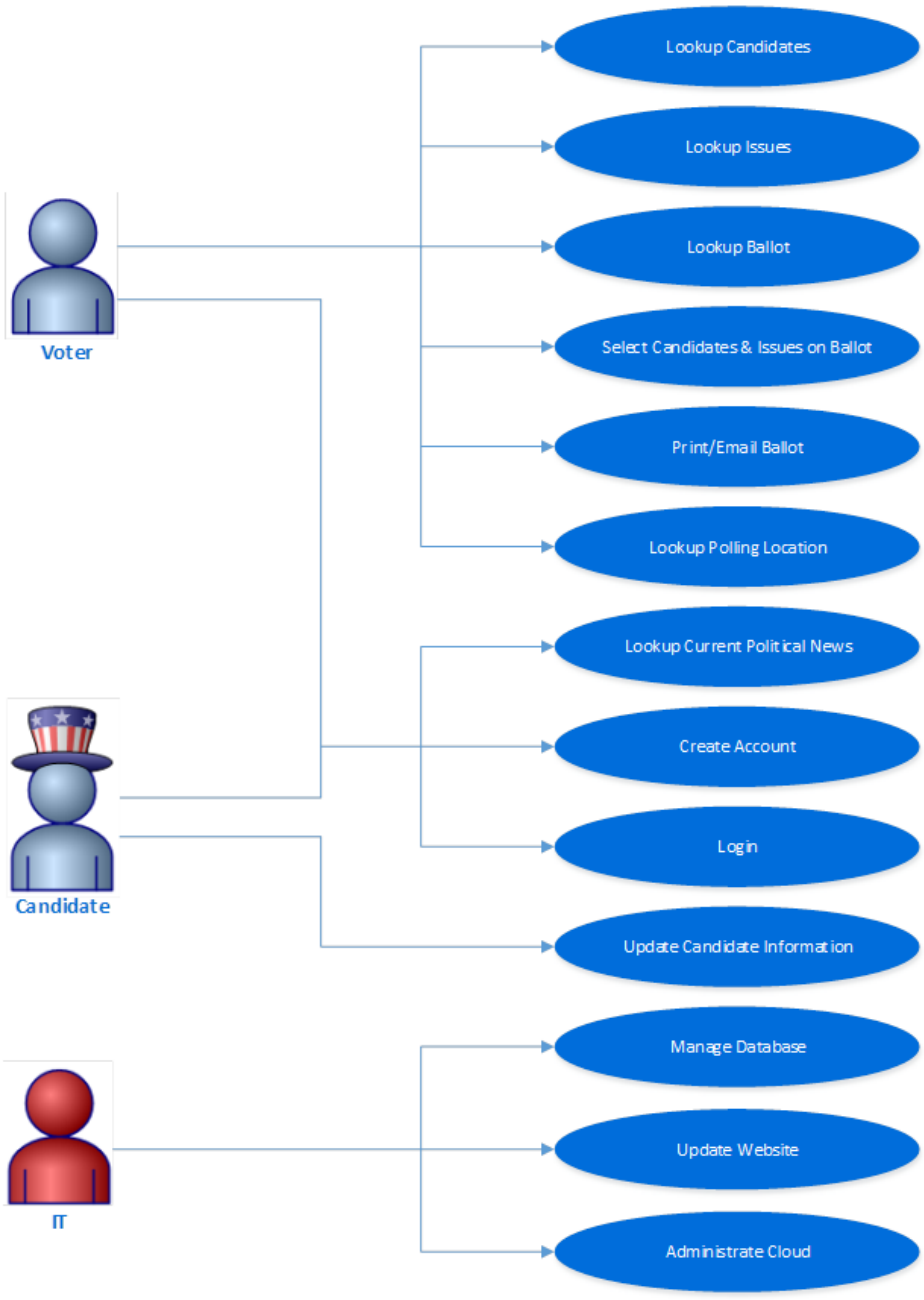


Figure 1. Use Case Diagram

2. PROJECT MANAGEMENT

2.1 Budget

Table 2 presents the budget for the project. The total cost for this project is $36,744, but we are donating this at no cost to the Ohio voting community. With this is being a free service, we are not expecting to have a return on investment.

No.	Item	Unit (Each)	Unit Price (Dollars)	Line Item Total
Networking				
1	Labor	230	$75	$17,250
Software				
2	Labor	230	$75	$17,250
Supplies				
3	Amazon AWS Cloud Service	7	$33	$231
4	Domain Name (OhioVoter.org)	1	$13	$13
5	Cincinnati Bell Internet Service	7	$50	$350
6	Router	1	$200	$200
7	Uninterruptable Power Supply	1	$195	$195
8	VoterSmart.org API	1	$2,000	$2,000
9	Network Server	1	$763	$763
10	SQL Server 2014 Standard Edition	1	$850	$850
			Subtotal	$39,102
			Total to the Ohio Voting Community	$0

Table 2. Project Budget

2.2 Objectives/Deliverable

Table 3 presents a list of the major milestones along with their scheduled deadlines.

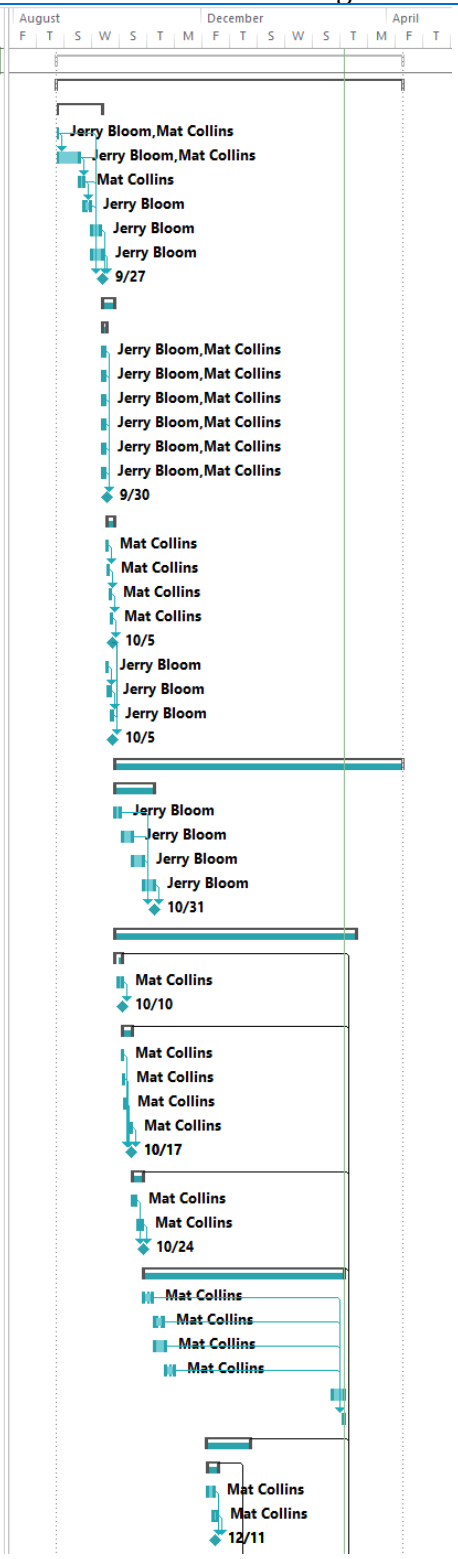
Major Project Milestones (Deliverables)			
Initiating	9/27/2016	Setup Servers	10/24/2016
Assemble Team	9/30/2016	Usability Testing	10/31/2016
Research Core Features and Design	10/5/2016	Setup Security	11/13/2016
		Presentation	11/28/2016
Signup for Cloud Services	10/10/2016	Build Website	3/28/2017
Setup Network	10/17/2016	System/Security Testing	4/11/2017

Table 3. Project Milestones and Due Dates

2.3 Project Schedule

During the fall semester we planned and built the network infrastructure using a hybrid approach for hosting a new website. We began the semester researching the core features for the network structure and website design options. Midway through the semester we got our servers up and running. We also created and tested prototypes for the website design. By the end of the semester we plan to start developing the website. During the spring semester we finished building our website and performed a variety of tests to ensure it is operating as planned before displaying our finished product at the IT Expo in April. See Figure 2 Gantt Chart for more information.

	Task Name	Start	Finish
0	GanttChart_Spring_Updated	Mon 8/29/16	Tue 4/11/17
1	1 Project deliverables	Mon 8/29/16	Tue 4/11/17
2	1.1 Initiating	Mon 8/29/16	Tue 9/27/16
3	1.1.1 Determine project idea	Mon 8/29/16	Tue 8/30/16
4	1.1.2 Get project idea approved	Tue 8/30/16	Mon 9/12/16
5	1.1.3 Determine project scope	Mon 9/12/16	Thu 9/15/16
6	1.1.4 Determine major milestones	Fri 9/16/16	Mon 9/19/16
7	1.1.5 Locate an expert	Tue 9/20/16	Tue 9/27/16
8	1.1.6 Locate a community partner	Tue 9/20/16	Tue 9/27/16
9	1.1.7 Initiating complete	Tue 9/27/16	Tue 9/27/16
10	1.2 Planning	Tue 9/27/16	Wed 10/5/16
11	1.2.1 Assemble project team	Tue 9/27/16	Fri 9/30/16
12	1.2.1.1 Identify project manager	Tue 9/27/16	Fri 9/30/16
13	1.2.1.2 Identify website developer	Tue 9/27/16	Fri 9/30/16
14	1.2.1.3 Identify database manager	Tue 9/27/16	Fri 9/30/16
15	1.2.1.4 Identify server administrator	Tue 9/27/16	Fri 9/30/16
16	1.2.1.5 Identify network administrator	Tue 9/27/16	Fri 9/30/16
17	1.2.1.6 Identify test/QA lead	Tue 9/27/16	Fri 9/30/16
18	1.2.1.7 Assemble project team	Fri 9/30/16	Fri 9/30/16
19	1.2.2 Research core features and design	Fri 9/30/16	Wed 10/5/16
20	1.2.2.1 Research cloud service models	Fri 9/30/16	Fri 9/30/16
21	1.2.2.2 Research providers	Sat 10/1/16	Sat 10/1/16
22	1.2.2.3 Research operating systems	Sun 10/2/16	Sun 10/2/16
23	1.2.2.4 Research database options	Mon 10/3/16	Mon 10/3/16
24	1.2.2.5 Research network designs	Tue 10/4/16	Wed 10/5/16
25	1.2.2.6 Research software tools	Fri 9/30/16	Fri 9/30/16
26	1.2.2.7 Research website design	Sat 10/1/16	Mon 10/3/16
27	1.2.2.8 Research content for website	Mon 10/3/16	Tue 10/4/16
28	1.2.2.9 Research core features and design	Wed 10/5/16	Wed 10/5/16
29	1.3 Executing	Wed 10/5/16	Tue 4/11/17
30	1.3.1 Conduct usability testing	Wed 10/5/16	Mon 10/31/16
31	1.3.1.1 Create sitemap	Wed 10/5/16	Mon 10/10/16
32	1.3.1.2 Design prototypes	Mon 10/10/16	Mon 10/17/16
33	1.3.1.3 User evaluation testing of prototypes	Mon 10/17/16	Mon 10/24/16
34	1.3.1.4 Analyze results from user evaluations	Mon 10/24/16	Mon 10/31/16
35	1.3.1.5 Usability testing complete	Mon 10/31/16	Mon 10/31/16
36	1.3.2 Setup Cloud Environment	Wed 10/5/16	Sun 3/12/17
37	1.3.2.1 Sign up for Services	Wed 10/5/16	Mon 10/10/16
38	1.3.2.1.1 Sign up for Amazon Web Services	Fri 10/7/16	Mon 10/10/16
39	1.3.2.1.2 Sign up for services complete	Mon 10/10/16	Mon 10/10/16
40	1.3.2.2 Setup Cloud Network	Mon 10/10/16	Mon 10/17/16
41	1.3.2.2.1 Establish Elastic IP	Mon 10/10/16	Tue 10/11/16
42	1.3.2.2.2 Open Up Ports	Tue 10/11/16	Wed 10/12/16
43	1.3.2.2.3 Setup DNS	Wed 10/12/16	Thu 10/13/16
44	1.3.2.2.4 Install SSL Certificate	Fri 10/14/16	Mon 10/17/16
45	1.3.2.2.5 Setup cloud network complete	Mon 10/17/16	Mon 10/17/16
46	1.3.2.3 Setup Servers	Mon 10/17/16	Mon 10/24/16
47	1.3.2.3.1 Install Apache Web Service on AWS	Mon 10/17/16	Thu 10/20/16
48	1.3.2.3.2 Install SQL Server on local server	Thu 10/20/16	Mon 10/24/16
49	1.3.2.3.3 Setup servers complete	Mon 10/24/16	Mon 10/24/16
50	1.3.2.4 Setup Security	Mon 10/24/16	Sat 3/4/17
51	1.3.2.4.1 Setup User Account	Mon 10/24/16	Sun 10/30/16
52	1.3.2.4.2 Configure SQL Server	Mon 10/31/16	Sun 11/6/16
53	1.3.2.4.3 Network Both Servers	Tue 11/1/16	Mon 11/7/16
54	1.3.2.4.4 Setup Database	Mon 11/7/16	Sun 11/13/16
55	1.3.2.4.5 Setup VPN	Sat 2/25/17	Sat 3/4/17
56	1.3.2.4.6 Setup Security complete	Sat 3/4/17	Sat 3/4/17
57	1.3.2.5 Switch to Hypervisor Solution	Mon 12/5/16	Mon 1/2/17
58	1.3.2.5.1 Install UNRAID	Tue 12/6/16	Mon 12/12/16
59	1.3.2.5.1.1 Setup NAS pool	Tue 12/6/16	Fri 12/9/16
60	1.3.2.5.1.2 Add Second Parity Drive	Fri 12/9/16	Sun 12/11/16
61	1.3.2.5.1.3 Install UNRAID complete	Sun 12/11/16	Sun 12/11/16



GANTT CHART

62	**1.3.2.5.2 Setup Windows Server Virtual Machine**	**Tue 12/13/16**	**Mon 12/26/16**
63	1.3.2.5.2.1 Reinstall SQL Server	Tue 12/13/16	Thu 12/15/16
64	1.3.2.5.2.2 Change Default SQL Server Ports	Thu 12/15/16	Sat 12/17/16
65	1.3.2.5.2.3 Map SQL Server to AWS	Sun 12/18/16	Wed 12/21/16
66	1.3.2.5.2.4 Backup Virtual Machine to NAS Pool	Thu 12/22/16	Sun 12/25/16
67	1.3.2.5.2.5 Setup Windows Server VM complete	Sun 12/25/16	Sun 12/25/16
68	**1.3.2.5.3 Additional Hardware**	**Tue 12/27/16**	**Mon 1/2/17**
69	1.3.2.5.3.1 Upgrade UNRAID Server Memory	Tue 12/27/16	Fri 12/30/16
70	1.3.2.5.3.2 Setup Uninterruptable Power Supply	Fri 12/30/16	Sun 1/1/17
71	1.3.2.5.3.3 Additional Hardware installed	Sun 1/1/17	Sun 1/1/17
72	**1.3.2.5.4 Switch to Hypervisor Solution complete**	**Mon 1/2/17**	**Mon 1/2/17**
73	**1.3.2.6 Setup Webserver 2.0**	**Mon 1/16/17**	**Sun 2/5/17**
74	1.3.2.6.1 Setup Elastic IP	Mon 1/16/17	Thu 1/19/17
75	1.3.2.6.2 Setup Windows 2012 R2 Server from AWS	Fri 1/20/17	Sun 1/22/17
76	1.3.2.6.3 Install IIS	Mon 1/23/17	Thu 1/26/17
77	1.3.2.6.4 Install WebDeploy	Fri 1/27/17	Sat 1/28/17
78	1.3.2.6.5 Install letsencrypt SSL Certificate	Sun 1/29/17	Mon 1/30/17
79	1.3.2.6.6 Redirect OhioVoter.org Domain	Tue 1/31/17	Sun 2/5/17
80	1.3.2.6.7 Setup Webserver 2.0 complete	Sun 2/5/17	Sun 2/5/17
81	**1.3.2.7 Implement VPN**	**Mon 2/27/17**	**Mon 3/6/17**
82	1.3.2.7.1 Research VPN options	Mon 2/27/17	Wed 3/1/17
83	1.3.2.7.2 Install VPN	Mon 2/27/17	Fri 3/3/17
84	1.3.2.7.3 Test VPN	Mon 2/27/17	Sun 3/5/17
85	1.3.2.7.4 Troubleshoot VPN	Mon 2/27/17	Mon 3/6/17
86	**1.3.2.8 SQL Server Database Implementation**	**Mon 2/13/17**	**Sun 3/12/17**
87	1.3.2.8.1 Setup Database Tables	Mon 2/13/17	Sun 2/26/17
88	1.3.2.8.2 Setup Database Permissions	Mon 2/27/17	Sun 3/5/17
89	1.3.2.8.3 Test SQL Server Data Input	Mon 3/6/17	Sun 3/12/17
90	1.3.2.8.4 SQL Server Database Implementation complete	Sun 3/12/17	Sun 3/12/17
91	**1.3.2.9 Setup Cloud environment complete**	**Sun 3/12/17**	**Sun 3/12/17**
92	**1.3.3 Build website**	**Mon 10/31/16**	**Tue 3/28/17**
93	**1.3.3.1 Create pages**	**Mon 10/31/16**	**Mon 3/27/17**
94	1.3.3.1.1 Create home page	Mon 10/31/16	Mon 11/14/16
95	1.3.3.1.2 Create candidate lookup page	Mon 1/16/17	Mon 1/30/17
96	1.3.3.1.3 Create issue lookup page	Mon 1/30/17	Mon 2/13/17
97	1.3.3.1.4 Create ballot lookup page	Mon 2/13/17	Mon 2/27/17
98	1.3.3.1.5 Create voter login page	Mon 2/27/17	Mon 3/13/17
99	1.3.3.1.6 Create candidate login page	Mon 3/13/17	Mon 3/27/17
100	1.3.3.2 Upload pages to server	Mon 3/27/17	Tue 3/28/17
101	1.3.3.3 Build website complete	Tue 3/28/17	Tue 3/28/17
102	**1.3.4 Testing**	**Tue 3/28/17**	**Tue 4/11/17**
103	1.3.4.1 Test website functionality	Tue 3/28/17	Tue 4/4/17
104	1.3.4.2 Test website security	Tue 3/28/17	Tue 4/4/17
105	1.3.4.3 Fix issues found	Tue 4/4/17	Tue 4/11/17
106	1.3.4.4 Testing complete	Tue 4/11/17	Tue 4/11/17
107	**1.4 Monitoring and Controlling**	**Mon 8/29/16**	**Tue 4/11/17**
108	1.4.1 Weekly progress meetings	Sat 9/17/16	Sat 4/8/17
134	1.4.2 Work with expert	Mon 9/26/16	Tue 4/11/17
135	1.4.3 Work with community partner	Mon 9/26/16	Tue 4/11/17
136	**1.5 Closing**	**Mon 1/9/17**	**Tue 4/11/17**
137	**1.5.1 Prepare for IT Expo**	**Mon 1/9/17**	**Tue 4/11/17**
138	1.5.1.1 Equipement Requirements	Mon 1/9/17	Mon 1/23/17
139	1.5.1.2 Create Poster	Mon 1/23/17	Mon 3/20/17

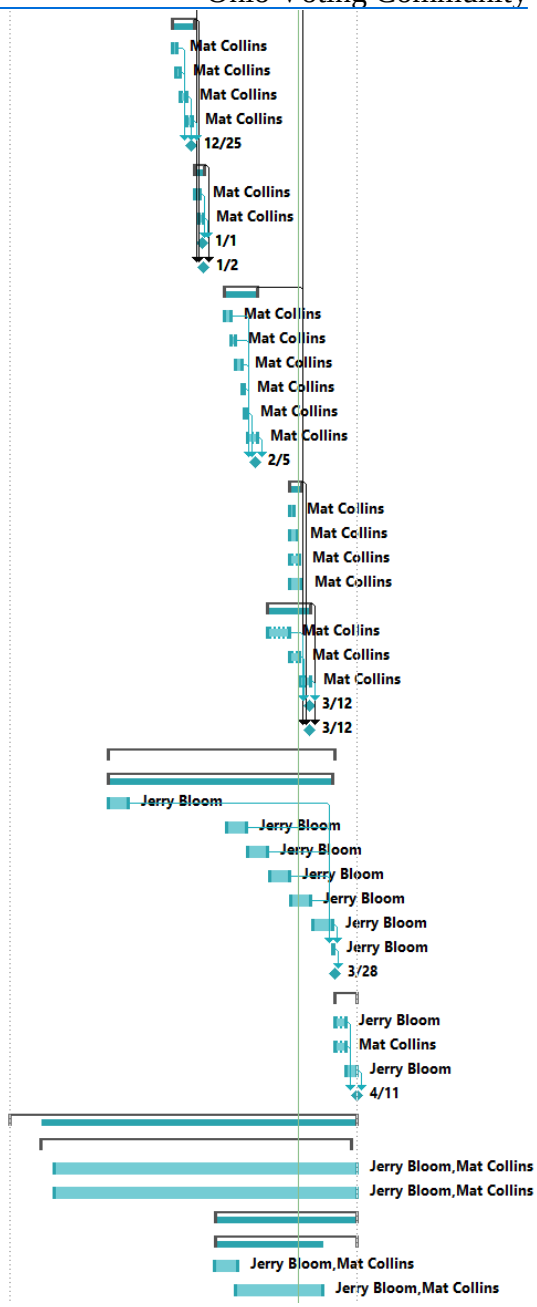


Figure 2. Gantt Chart

3. TECHNICAL ELEMENTS

3.1 Introduction

There are many the technical elements involved in developing OhioVoter.org. With a combination of hardware and software setup on Amazon Web Services and at home, we were able to create a stable and secure infrastructure for hosting our website.

3.2 Network

OhioVoter.org operates on a hybrid cloud environment. The website is hosted on a Windows Server 2012 virtual machine (VM) from Amazon Web Services. The website connects to a Microsoft SQL database that is stored on a virtual machine that is locally hosted a server. The local server connects to the internet via a static IP provided by a local service provider.

3.3 Cloud

OhioVoter.org is published on Amazon Web Services. These services can rapidly expand when needed at peak periods such as election season. Figure 3 shows the Amazon Web Services EC2 Virtual Machine Instances currently in use. Also, Amazon Web Services has additional services that can be implemented when you are ready to make changes.

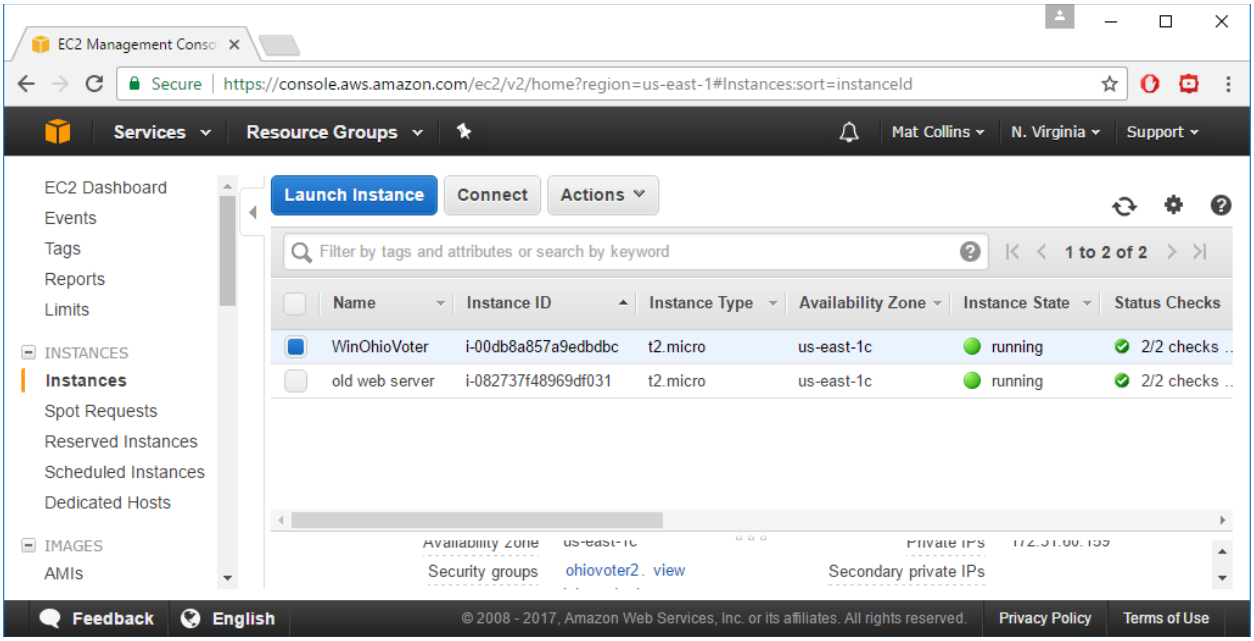


Figure 3. Amazon Web Services EC2 Virtual Machine Instances

3.4 Management and Security

The web server is managed remotely via the Remote Desktop Protocol (RDP). Figure 4 shows the tools that are available through this remote connection. The server allows the web developer to use Web Deploy, shown in Figure 5, to upload the webpages into the server. By using Web Deploy, we are able to improve security by providing only the necessary privileges so Visual Studio can publish updates to the server. The web server has an SSL certificate provided by Let'sEncrypt.org. The SQL box will only allow the elastic IP from the Amazon Web server to connect to the SQL with set port restrictions. The SQL box allows the server administrator to connect via VNC remote desktop shown in figure 6. The SQL box is stored locally to provide an additional security option for protecting the data. The SQL box will be using windows firewall with a limited number of open ports.

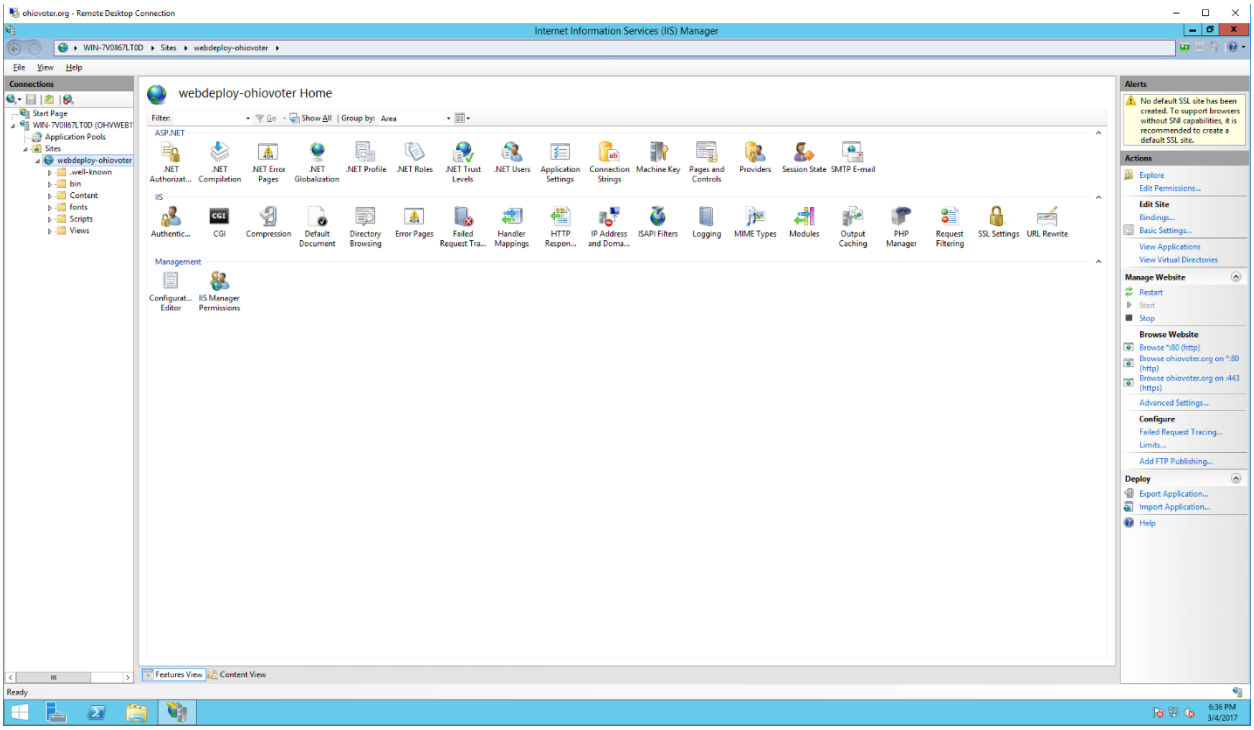


Figure 4. OhioVoter.org remote desktop connection

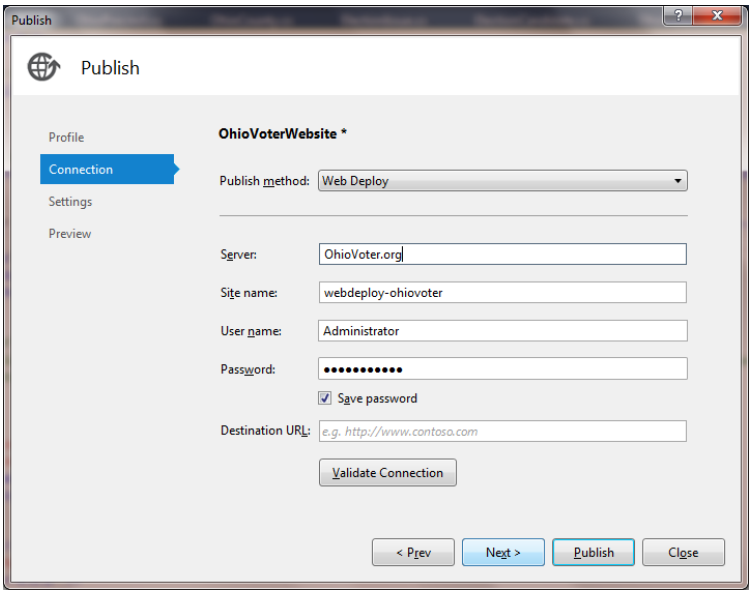


Figure 5. Web Deploy

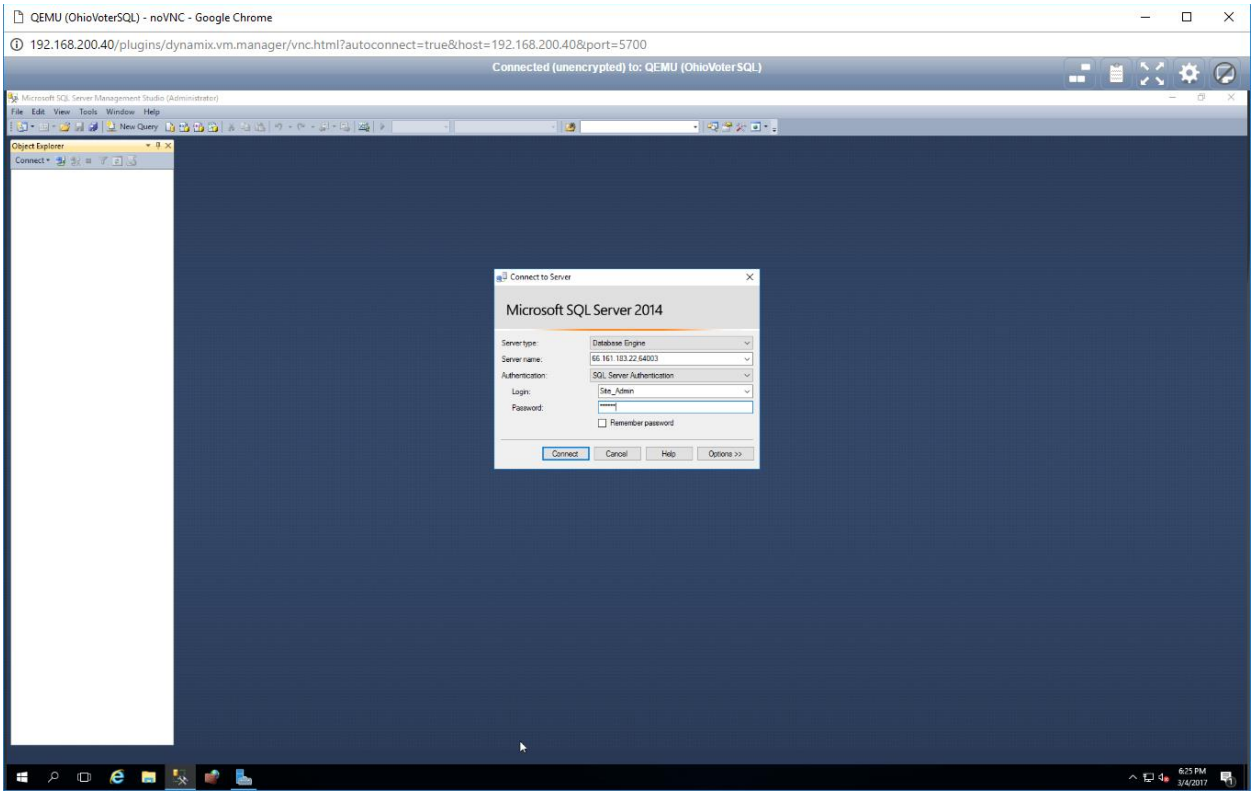


Figure 6. SQL Server Remote Connection

3.5 Software

The website was developed using the ASP.NET MVC framework with the C# programming language. Figure 7 shows the Visual Studio development environment we used for creating the webpages. Updates were pushed to a GitHub repository for maintaining source control. The website is stored on Windows Server 2012 R2 AWS instance with IIS and Web Deploy installed. The local server is powered by a Linux distro installed called unRAID. With unRAID, we have the capability to manage Network Attached Storage (NAS) pools, Docker, virtual machines, and a variety of plugins to improve the user interface. The local server has a virtual machine with Microsoft Server 2012R2 and Microsoft SQL Server 2014 Standard Edition with Service Pack 1 for managing the database. Figure 8 shows some of the virtual machines that can

be managed from the unRAID console. Additionally there is a Krusader Docker setup for managing files on the NAS pool.

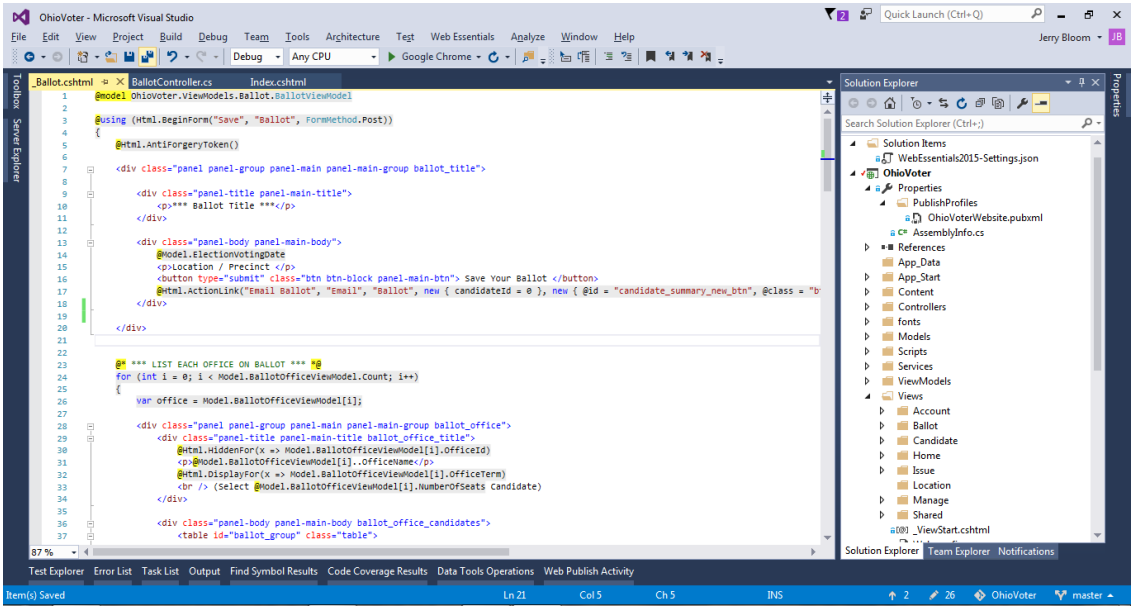


Figure 7. Visual Studio

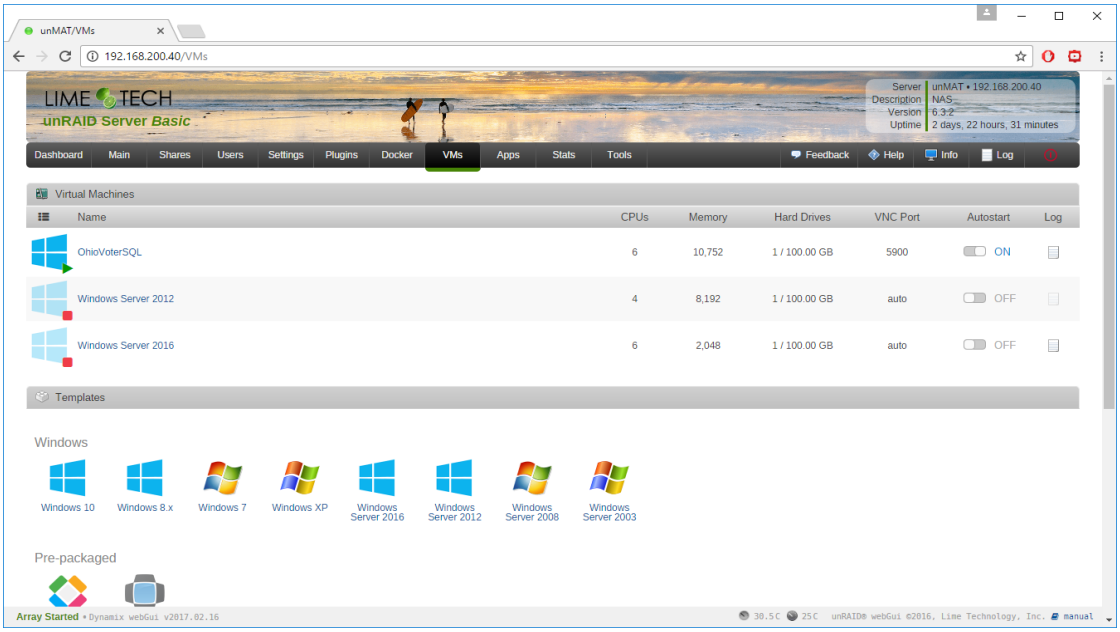


Figure 8. unRAID Console

3.6 Backup

The SQL virtual machine is backed up to a NAS pool that is hosted on the same machine. The NAS pool provides a way to verify drive integrity and view storage space. Figure 9 shows how this is set up. The Krusader Docker, shown in figure 10, allows us to manage files with a Graphical User Interface (GUI) and copy data from the virtual machine source to the NAS pool.

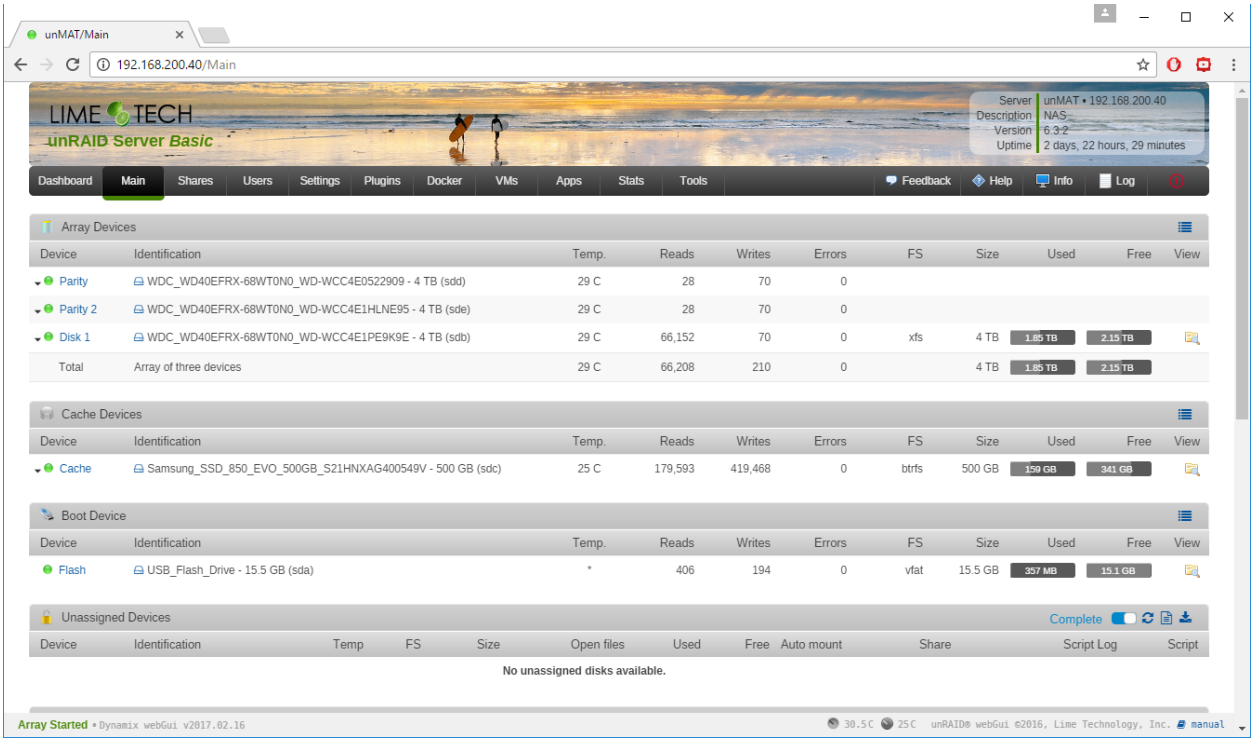


Figure 9. NAS Drive Pool Setup

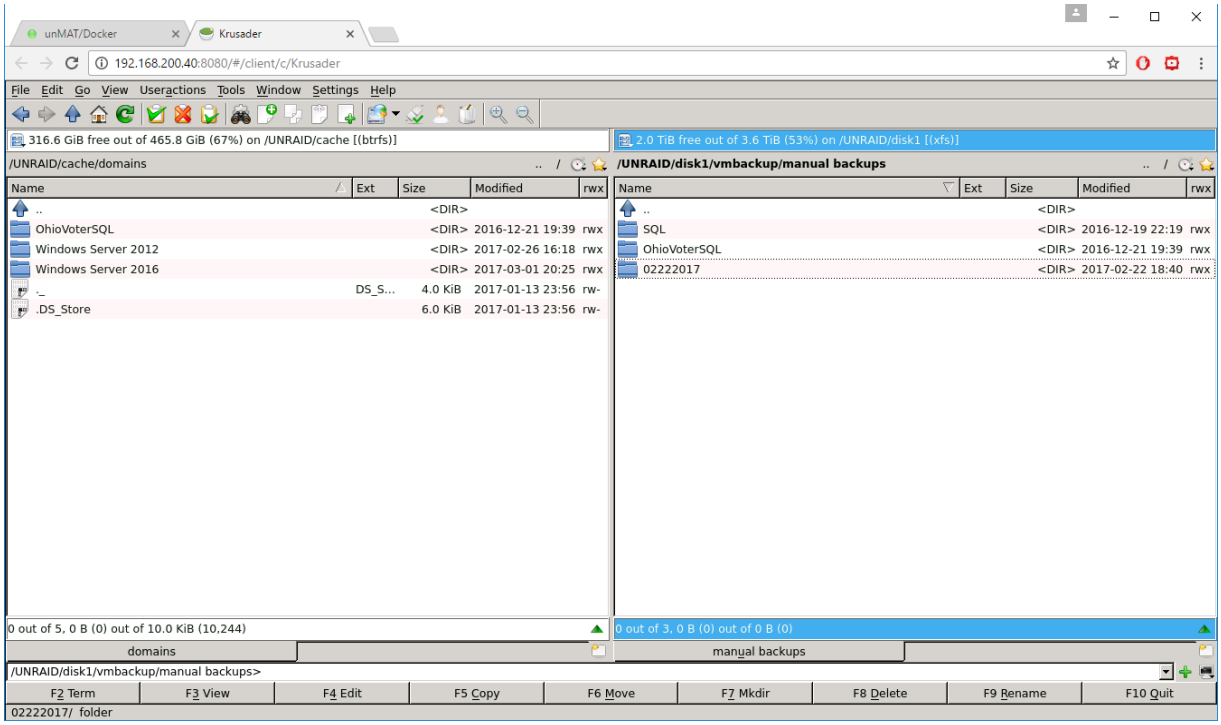


Figure 10. Krusader Docker

4. APPLICATION

4.1 Application Architecture

Figure 11. Physical Model Diagram illustrates the network infrastructure for our Cloud server and SQL server. The SQL server is managed locally to provide additional security over our database. The Microsoft SQL server is setup as a virtual machine that backs up to NAS pool with parity drives. Should one drive fail, the additional drive will keep the system running. The network attached storage is managed on the same server as the virtual machine. The unRAID server hosting the systems connects to a secure router that has ports open to the static IP address provided by a local ISP. The static IP connects the SQL server to the web server hosting the OhioVoter.org website. The web server is hosted on Amazon Web Services cloud service. The IT staff has the ability to connect with Remote Desktop into the server to update the server and the website when needed. The voters and candidates are able to access the OhioVoter.org website from their own computers.

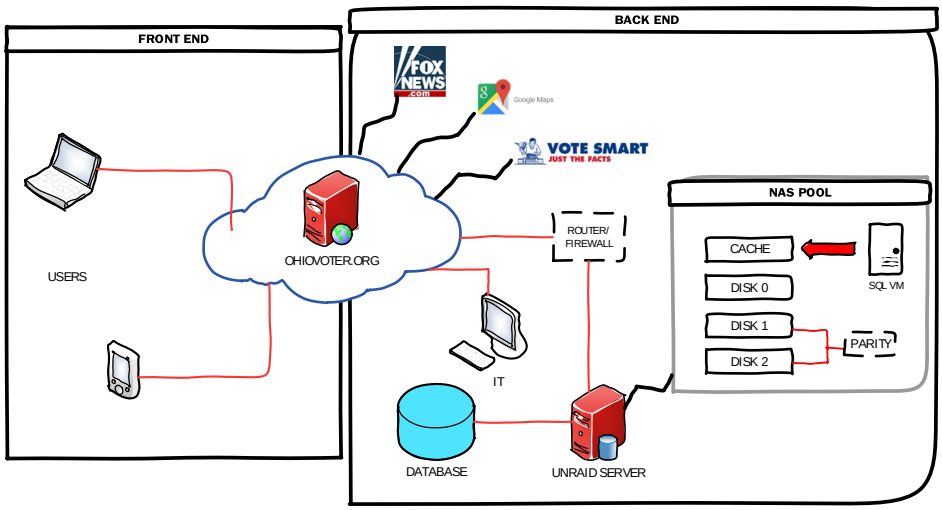


Figure 11. Physical Model Diagram

5. Testing Plan

5.1 Overview

This section explains our testing methodology for using OhioVoter.org on a variety of common devices ranging in size from the smaller smart phone up through the larger desktop computer. Specific tests will be structured according to the various requirements of each component of the application.

5.2 Scope

The scope of testing is to assess the functionality of OhioVoter.org website on phones, tablets, laptops, and desktop computers. The assessment will be standardized based on the requirements that have been established for the website.

5.3 Objective

Each component will be tested both individually and collectively to ensure they are working as expected. Developers will be responsible for performing tests and making adjustments on the components they are developing.

5.4 Criteria (Entry/Exit)

- Entry Criteria:
    - Website completion
    - Self-assessment complete
    - Staging environment is prepared

- Exit Criteria:
    - Complete assessment
    - Website errors are documented and resolved

5.5 Logging Test and Reporting

- Create List?
- Case/Requirements/Description-Expected Result/Actual Result/Pass-Fail/Notes
- Create Weekly Plan/Schedule?

5.6 System Testing

OhioVoter.org will be assessed as a fully functional website with each feature being verified. The assessment will greatly improve our chances of catching issues/bugs with the functionality of the website. Testing will be used as a tool to complete the service of OhioVoter.org.

5.7 Testing Procedures

Table 4 displays the list of testing scenarios that each component must pass before the application is deployed. These tests focus on network connectivity, user interface, and functionality. Documentation will be maintained as each test is performed with the details of the results. Tests that fail will be addressed at our weekly meeting. A plan will be formed to correct the issue based on its severity.

5.8 Pass/Fail Conditions

It is mandatory that the website completes a full assessment. If any functionality is lacking, then the tester shall log this in the log book.

5.9 Risks

The potential issues can invalidate the assessment:

- Device failure
- Web connectivity issues
- Availability of database server
- Delay of scheduled meetings

Date	Case	Requirements	Description / Expected Results	Pass/Fail	Notes
4/10/17	1	**Stability**	Runs on SmartPhone device properly	Pass	
4/10/17			Runs on Tablet device properly	Pass	
4/10/17			Runs on Laptop device properly	Pass	
4/10/17			Runs on DeskTop device properly	Pass	
4/10/17	2	**Login Authentication**	User login screen is displayed properly	Pass	
4/10/17			User login is successful	Pass	
4/10/17			User session is maintained until logout	Pass	
4/10/17	3	**Sidebar Display**	User location update form is displayed properly	Pass	
4/10/17			User location update is successful	Pass	
4/10/17			User location update displays address properly	Pass	
4/10/17			User location is maintained until leaving site	Pass	
4/10/17			Polling location is displayed properly	Pass	
4/10/17			Google Civic API is integrated into project	Pass	
4/10/17			Google Maps API is integrated into project	Pass	
4/10/17			Interactive map is displayed when user clicks "View Map"	Pass	
4/10/17			State location is displayed properly	Pass	
4/10/17			County location is displayed properly	Pass	
4/10/17			County location is maintained until leaving site	Pass	
4/10/17	4	**Home Display**	Election dates are pulled from database	Pass	
4/10/17			Upcoming election dates are displayed properly	Pass	
4/10/17			Polling results are pulled from database	Pass	
4/10/17			OhioVoter polling results are displayed properly	Pass	
4/10/17			CNBC Political RSS feeds integrated into project	Pass	
4/10/17			Individual feed properly displays in a new tab when selected	Pass	
4/10/17	5	**Candidate Lookup Display**	Candidate information pulled from database	Pass	
4/10/17			Votesmart.org API is integrated into project	Pass	
4/10/17			Google Civic API is integrated into project	Pass	
4/10/17			Candidate list is displayed properly	Pass	
4/10/17			Selecting a candidate from list is successful	Pass	
4/10/17			Selected candidate information is displayed properly	Pass	
4/10/17			Compare candidate list displays other candidates properly	Pass	
4/10/17			Selecting a candidate to compare from list is successful	Pass	
4/10/17			Compare candidate information is displayed properly	Pass	
4/10/17			Removing selected candidate to compare is successful	Pass	
4/10/17	6	**Issue Display**	Issue information pulled from database	Pass	
4/10/17			List of issues are displayed properly based on location in sidebar	Pass	
4/10/17			County list is displayed properly	Pass	
4/10/17			Selecting a county from list is successful	Pass	
4/10/17			List of issues are displayed properly based on selected county	Pass	
4/10/17			Selecting an issue from list is successful	Pass	
4/10/17			Issue information is displayed properly	Pass	
4/10/17			Selecting full issue link when available is successful	Pass	
4/10/17			Additional full text information is displayed properly	Pass	
4/10/17	7	**Ballot Display**	Ballot information pulled from database	Pass	
4/10/17			Votesmart.org API is integrated into project	Pass	
4/10/17			Google Civic API is integrated into project	Pass	
4/10/17			Offices are listed properly based on location in sidebar	Pass	
4/10/17			Candidates are properly displayed when office is selected	Pass	
4/10/17			Write-in candidates are properly displayed when selected	Pass	
4/10/17			Office information is properly displayed when selected	Pass	
4/10/17			Selecting a candidate to lookup is successful	Pass	
4/10/17			Candidate lookup page opens successfully	Pass	
4/10/17			Selecting candidate(s) to vote for is successful	Pass	
4/10/17			Ballot information is successful sent to user	Pass	
4/10/17			Ballot information is saved successfully for logged in users	Pass	
4/10/17			Ballot information successfully captured for polling results	Pass	
4/10/17	8	**Web Servers**	Log into Amazon Web Service	Pass	
4/10/17			Verify that EC2 Instance Running	Pass	
4/10/17			Log into Web server - SSH	Pass	
4/10/17			Check database connectivity	Pass	
4/10/17	9	**Database Servers**	Verify Virtual Machine backup - Krusader Docker	Pass	
4/10/17			Log into Virtual Machine	Pass	
4/10/17			Check MSSQL Logs	Pass	
4/10/17			Check for Microsoft updates	Pass	
4/10/17			Check uninterruptable power supply	Pass	

Table 4. Testing Scenarios

6. CONCLUSION

6.1 Fall Semester 2016

In the 2016 fall semester, we tested concepts for our website through user evaluation and began creating our home page for OhioVoter.org. We finished setting up our cloud services through Amazon AWS that will be hosting our website. Our SQL server was also set up onsite. By the end of the fall semester communication was established between the servers on our hybrid network.

6.2 Spring Semester 2017

In the 2017 spring semester, we developed and implemented various components necessary for deploying OhioVoter.org. This included setting up a revised web server, storing our data on our local SQL server, and implementing a VPN solution. After developing and publishing our website using Web Deploy, we ran a series tests to ensure it was working correctly before attending the IT Expo in April.

6.3 Lessons Learned

At the beginning of this project we spoke a variety of people from the Hamilton County Board of Elections and League of Women Voters, but were unsuccessful finding a subject matter expert willing to partner with us. As a result, we had to do a lot of research to learn about the voting process. A lesson we learned here was we were unaware of how much time would be required to learn about such a complex topic and piecing all the components together.

Another challenge we encountered was hosting a website developed in ASP.NET on an Apache web server. By implementing a Windows Server with Web Deploy, we were able to simplify the deployment of our website from Visual Studio while also improving the security of our application.

6.4 Next Steps

The goal of our project was to create a platform where voters can easily access a comprehensive list of ballot information. This project was created by focusing on the voters of Hamilton County to prove our concept. The next steps for our project will be to include information for all the counties of Ohio. This may require updating some of the information displayed to be more in line with the information the Ohio voting community finds helpful.

A second goal for our website was to provide candidates with the ability to verify and update their information if needed. We ran out of time to implement this portion of the project. This will need to be completed during the next steps of the project.

BIBLIOGRAPHY

Coffey, Daniel, John Green, David Cohen, and Stephen Brooks, "Buckeye Battleground: Ohio, Campaigns, and Elections in the Twenty-First Century," University Press Publications (2011), accessed October 10, 2016, http://ideaexchange.uakron.edu/uapress_publications/152.

Kocoshis, Marie, 2016, Interview by Jerry Bloom and Mat Collins, League of Women Voters, Cincinnati, Ohio, September 21.