Culture Shock

by

David Herier
Jimmy Tran
Harmanjot Singh

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/Cybersecurity

David Herier	11/28/2021
David Herier	Date
Jimmy Tran	11/28/2021
Jimmy Tran	Date
Harmanjot Singh	11/28/2021
Harmanjot Singh	Date
Vineela Kunapareddi	Date

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

April 2022

Table of Contents

Abstract

There are studies that show a loss of cultural information for immigrants over time, as well as a disconnect between generations of immigrants. Because each generation of immigrant further dissociates from their cultural roots, the information is lost. The purpose of Culture Shock is to allow users to preserve and share cultural information through a web application, as well as form a community around their heritage. The project is a web application built using ASP.NET Core with components such as JavaScript, HTML, and Bootstrap. Data is stored using a SQL database and is retrieved and inserted with Dapper ORM. It contains a wiki component as well as a social media component. The result is a web application that allows users to upload their knowledge to be preserved, as well as interact with other users to further enrich their knowledge. This means that important cultural information will be preserved and shared.

Introduction

Project Summary:

Culture Shock is a web application that blends a social media application and content management system to create a platform oriented towards cultural information. The platform will allow users to submit information about their respective cultures such as ceremonies, food, events, and other aspects of culture. This platform will be like a combination of Facebook and Wikipedia.

Problem Statement:

There is no central platform for second-generation and third-generation immigrants to learn about and share their respective cultures. If the problem is not solved, many minorities in the US may find a disconnect from their heritage, if there was an effortless way to stay connected with it more people would stay connected.

In the life of second-generation immigrants what they learn growing up in the country that they were born in can be quite different than what their parents learned. Many second-generation immigrants can feel disconnected from their parents' culture and sometimes resent it as Deanne stated "The problem [of the second-generation immigrant] was solved by escape he wanted to forget everything" (Jones, Puloka). This can lead to a total disconnect for third-generation immigrants. As many of those second-generation immigrants want to learn more or the third-generation ones in general want to delve into their heritage, it would be nice for them to have a fun easy place to turn to, being this website.

Currently, there is no centralized platform for something like this. There are existing platforms such as Facebook groups or Wikipedia where users can go to meet others with a similar heritage and share information, but none of it is centralized. Culture Shock aims to implement such features in a culture forward platform.

Solution:

The solution to the problem is a platform where users can share information about their respective cultures as well as browse information pertaining to other cultures. The web application will have a wiki that will address the loss of information problem outlined above. Users will be able to share important knowledge that they have about their culture. The platform will also include a social media where users can post culturally significant things like pictures of food or areas they visit, or they may share text posts. There will be integrations that allow others to explore that information, such as Wikipedia, which will allow users to dive deeper if any information is missed.

Project Source:

The inspiration for this project came from when Harman took a visit to a cultural museum in Hawaii. It was discovered that there was little that was known about many different other cultures. The team came together from friendships and from work experience.

List of illustrations

Discussion

Project Objectives/Goals:

The solution will allow users with vast knowledge about their culture to contribute that information and ensure that the information is not lost over time. It will also allow others to discover parts of their heritage that they would not have discovered otherwise.

Main Goals:

- The wiki and content management system will provide a platform for users to share information about their respective culture. This tackles the problem of cultural knowledge dwindling over generations. This application will allow users to submit pages containing information about various aspects of their culture. This ensures that the information will stand the test of time and not be lost over generations.
    - Information will be organized in several layers, first starting with regions, then countries, then aspects such as food and rituals, and finally at the lowest level will be individual articles.
    - Users logging on will be able to browse different articles to gain information about cultural aspects.
    - Users will be able to submit articles on a certain aspect of culture such as food. The user will be able to choose which country it is for.

- The social media portion will provide a hub for users to share culturally significant media. This provides a way for users to discover things about other cultures, as well as share things about their own culture. The application provides a social platform where users can be exposed to the culture of other users.
    - Users can expect a dashboard like that of Facebook's. They will be able to add friends and view a feed of posts made by their friends.
    - Users will be able to upload media such as text posts and photos. Since the platform is culture focused, this allows users to share things such as important current events or pictures of culturally significant areas.

Reach Goals:

- Add group functionality allowing users to connect with other users in a more specific manner, like Facebook groups. Users will be able to post events, make announcements, and interact with other users within the group.
- A personal messaging client allowing users to directly communicate with one another.
- The web application may have an interactive globe as the "main menu" for the wiki. Instead of a list, users will be able to interact with a globe, and clicking on a country on the globe will pull up the wiki section for that country.

Project Scope:

- The wiki will allow users to upload, edit, and read pages pertaining to cultural information. These functionalities will be supported by respective C# code and SQL queries. The information will be stored in C# objects which will then be inserted into

SQL tables stored on a database. The pages will be organized first by culture, and then by aspects such as food, events, and ceremonies.

- A social media dashboard that serves as the social hub for the user. The dashboard will show posts made by those on the users' friends list and allow for commenting and liking. This dashboard will support text, photo, and video posts.
- Technical goals
    - The application will use Microsoft's ASP.NET framework to create a well-designed and practical web application. SQL will be utilized for storing and querying data.
    - Front end
        - The web application will have two pages, one that mimics a social media feed and one that mimics a wiki for content management.
        - The social media page will use bootstrap to create a layout that is aesthetically pleasing. It will be the central hub for social interaction between users, allowing for posting, commenting, and adding friends. Users will be able to upload media that they find culturally significant and will be able to see posts that are uploaded by the friends that have been added.
        - The wiki page will render informational pages in a minimalistic and practical manner. It will also allow users to upload or edit pages.
        - Web application will dynamically adjust components to account for varying devices and screen sizes.

- Back end
    - The back end will perform actions that the user performs on the front end, such as the insertion, editing, and viewing of data. The front-end HTML pages will pass user input data to the controllers, which will then pass the data into the business layer code to perform business layer logic. That data is then passed to the data layer which acts as the liaison between the code and the SQL database.
    - Effective code planning to minimize tech debt. Objects and classes will be designed to separate logic into the right directories and files.
    - The code should interact with SQL database to store objects correctly and allow for retrieval later. The SQL tables will mimic the C# objects to allow for consistency when performing data layer actions.

Quick Project Timeline:

Table 1 Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1.a	Create wiki HTML home page and controller	Sprint 1	10/7/21	10/13/21
1.b	Create social media HTML page and controller	Sprint 1	10/7/21	10/13/21
1.c	Create models pertaining to wiki	Sprint 1	10/7/21	10/13/21
1.d	Create SQL tables for wiki	Sprint 1	10/7/21	10/13/21
1.e	GET requests for country pages	Sprint 1	10/7/21	10/13/21
1.f	POST request for creating a page for a new country	Sprint 1	10/7/21	10/13/21

1.g	Backend endpoints connecting to database for data access methods	Sprint 1	10/7/21	10/13/21
1.h	Map out SQL tables and foreign key relationships between tables	Sprint 1	10/7/21	10/13/21
1.j	Map out object relationships to reflect SQL table	Sprint 1	10/7/21	10/13/21
2.a	Create country information HTML page	Sprint 2	10/14/21	10/20/21
2.b	Controller GET and POST actions for country information page	Sprint 2	10/14/21	10/20/21
2.c	Data access layer methods retrieve, insert, and update country information data	Sprint 2	10/14/21	10/20/21
2.d	Controller GET and POST actions for country aspect categories	Sprint 2	10/14/21	10/20/21
2.e	Data access methods for aspect categories	Sprint 2	10/14/21	10/20/21
2.f	OAuth and account logic	Spring 2	10/14/2021	10/20/21
3.a	HTML pages for individual aspect pages	Sprint 3	10/21/21	10/27/21
3.b	Controller GET and POST actions for aspect pages	Sprint 3	10/21/21	10/27/21
3.c	Data access methods for individual aspects	Sprint 3	10/21/21	10/27/21
4.a	HTML page for social media. Handles displaying and uploading text posts	Sprint 4	10/21/21	10/27/21
4.b	Controller GET and POST actions for text posts	Sprint 4	10/21/21	10/27/21
4.c	Tie OAuth logic to social media pages	Sprint 4	10/21/21	10/27/21

5.a	Backend logic and models for social media friends	Sprint 5	11/5/21	11/11/21
5.b	Change social media feed to display text posts made by user's friends	Sprint 5	11/5/21	11/11/21
5.c	Add UI on social media page to show friends	Sprint 5	11/5/21	11/11/21
5.d	Add logic to handle liking of posts between friends	Sprint 5	11/5/21	11/11/21
6.a	Host production site on server	Sprint 6	11/12/21	11/18/21
6.b	Ensure no network issues or bugs	Sprint 6	11/12/21	11/18/21
6.c	Penetration testing and security audit of web application	Sprint 6	11/12/21	11/18/21
7	Develop test plan to ensure all scenarios are accounted for	Sprint 7	11/19/21	11/25/21
8	Testing	Sprint 8	11/26/21	12/2/21
9	Testing	Sprint 9	12/6/21	12/9/21

Technologies Used:

- C# ASP.NET MVC
    - ASP.NET MVC is a framework provided by Microsoft for web applications. It uses the Model-View-Controller design pattern that allows for separation of concerns in the code. The framework uses C# as its language for the backend. We are choosing to use C# ASP.NET MVC because it provides a robust framework for web applications, with a well laid out architecture for front end and backend. Both developers have a strong background using this framework
- Razor

- Razor is a feature of Microsoft's ASP.NET framework that allows for the injection of C# code right into HTML files. Traditionally, HTML files are strictly HTML, but the Razor feature allows for C# to be written in the files, allowing for dynamic page rendering based on parameters passed in from the backend, and for advanced routing techniques. We are choosing to use Razor because it is the default for ASP.NET web applications, and it will also allow for dynamic rendering on the front end.

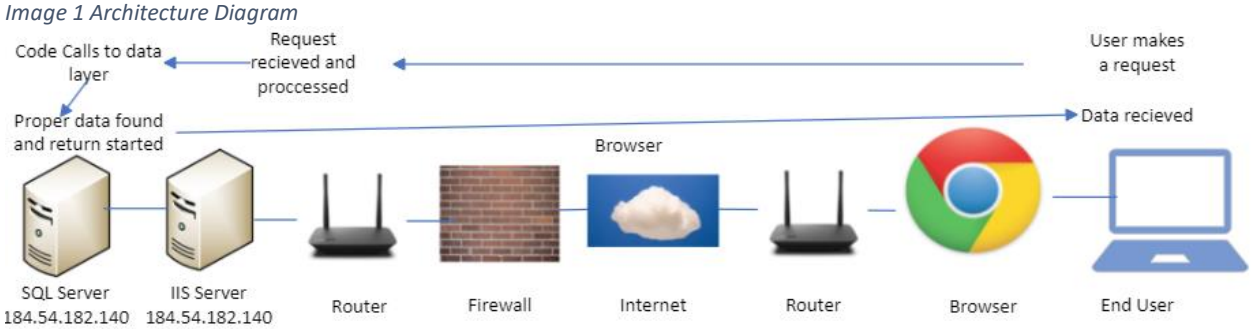
- Visual Studio
  - Visual studio will be used for the development of code of the project. Microsoft provides tools for the creation and templating of .NET projects. Also, since Microsoft is the creator of .NET, there is plenty of documentation for the framework.
- SQL/SQL Server Management Studio
  - SQL will be used for the databases and querying objects. SQL is a powerful language that allows for custom queries, which can be useful for when a user needs data to be displayed a certain way. It tends to be used in conjunction with .NET, and is frequently utilized for web applications
- JavaScript
  - JavaScript will be used to handle UI behaviors. The goal of the website is to be aesthetically pleasing and easy to use. JavaScript allows the modification of HTML to perform functions such has hiding or displaying elements of the page or dynamic modification of page elements.

- Bootstrap
    - Bootstrap is a CSS framework that provides many pre-rendered components of a web page, as well as rules handling changing window sizes and viewing on different devices. The framework was originally developed for Twitter to keep things consistent across the project. Bootstrap will be used for this project due to its focus on consistently rendering front end components and the availability of many pre-made assets.

Technical Architecture Diagram:

Image one represents the flow of data when the user uses the website and how the requests made when doing so flow from one end to the other.

Image 1 Architecture Diagram



User Personas:

Table 2.1 User Persona Table

User Persona: 1	
	Title The adventurous cook
	Name

	Magen Smith
	Age 38
	Gender Female
About	• Magen loves to travel and loves learning about different cultures through education, friends and traveling • Has a passion for cooking from a young age • Loves to socialize with people that share the same interest as her • Works for a local restaurant
Goals	• Wants to learn how to make food from all different parts of the world • Wants to share different recipes she knows with other people • wishes to open her own restaurant one day
Environment	• Born in France and moved to Italy at a young age and come to America for school as her parents our originally from America • Went to school for Culinary Arts and got a job as a Chef at a local restaurant

Table 2.2 User Persona Table

User Persona: 2	
	Title Lost Soul
	Name Satwinder Gill
	Age 20
	Gender Male

	
About	• Satwinder is very social and has made many friends at school, but it seems that he does not fit in with his peers of his Indian Community due to his lack of knowledge of the culture • Satwinder attends only some family functions because he does not fully grasp the culture.
Goals	• Wants to learn the different holidays and how they are celebrated • Wants to learn about various aspects of the culture that he does not understand • Satwinder also wants to socialize and make connections in his Indian community
Environment	• Born in United States, he is a 1st generation Indian American as his parents are originally from India • Grew up with his parents in their household who are westernized and speak English

Use Cases:

Table 2.3 Use Case Table

Use Case ID	Wiki_001
Use Case Name	Home Page for non-registered users
End Objective	This Page is the first impression on visitor so must be user friendly and fair representation of website, this page also is the wiki which holds all the information.
User/Actor	Non-registered Visitor
Trigger	Going to website if no user is logged in
Frequency of Use	Every Time

Preconditions	• Visitor can go to search engine and enter key words to result webpage • Navigate to website
Basic Flow	The website has a navigation bar on the top. After Navigation bar You have Wiki page. The wiki layout will be categorized by country. Each country will have different key words that categorize things within such as food, culture, weddings.
Alternate Flow	System determines user is already logged on and instead of going to Wiki tab it navigates to Social Tab

Use Case ID	Social_001
Use Case Name	Home Page for users
End Objective	This Page is the interaction page, and the social media side of the websites that will include news feed and other features from open source.
User/Actor	user
Trigger	Going to website while being logged in
Frequency of Use	Every Time
Preconditions	• User can go to search engine and enter key words to result webpage • Navigate to website
Basic Flow	The website has a navigation bar on the top for website. Then 2nd Navigation bar on the side for social media controls. Lastly, the page will have news feed from friends.
Alternate Flow	System Determines user is not logged on and instead of going to social tab, it navigates to Sign in tab.

Table 2.4 Use Case Table

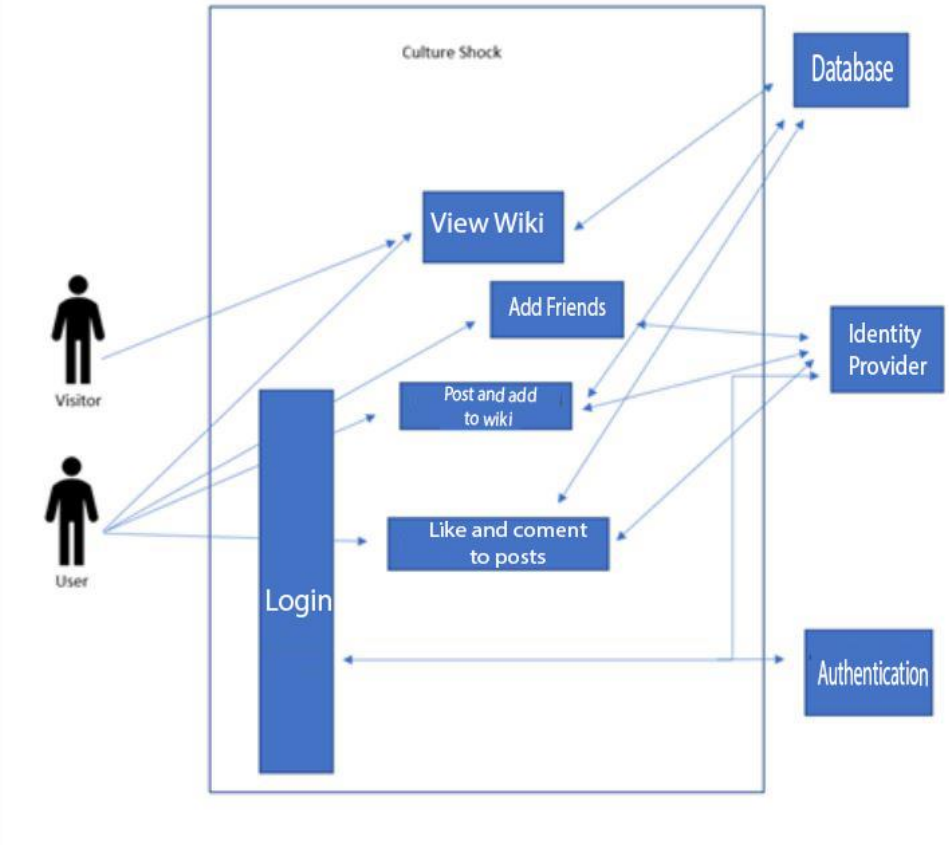
Use Case ID	Navigation_001
Use Case Name	Navigation bar for users.
End Objective	This will help navigate the website.
User/Actor	user / visitor
Trigger	From any webpage, click any of the four options on the main Navigation bar.
Frequency of Use	Every Time you need to change page.

Preconditions	Must be on website.
Basic Flow	This navigation bar will have two bold tabs in the center and 2 smaller tabs to the right The two bold tabs are: Wiki, Social. The two smaller tabs to right are “Sign in” and “More” tab. >> More reveals “Help” and “About”
Alternate Flow	If already signed on, then the “sign in” tab will change from sign in to log off, but the rest of navigation bar will stay the same.

Use Case Diagram:

We have two different kinds of users, one that is a visitor and one that has an account. The Visitor can only view the Wiki data. The user can log on and be authenticated. Once authenticated the user can add posts, comment, add friends, and like other posts.

Image 2 Use Case Diagram



Testing Plan:

Overview:

This testing plan will encompass all that goes into how this project is being tested. An overview of how and why tests are being run, along with the scope of the test is shown along with what the result of this testing should be. The quality assurance role will be creating a log of what was tested and the results of it, then reviewing that log in a retrospective, which will then be used for continuous improvement of the project.

Methodology:

A proactive approach will be taken, where each function of the website is tested as early as possible to find and fix defects before the page is completely created. This allows the project to be built faster without having to retroactively fix issues.

Scope:

Use Cases:

The use cases can be broken down into two sections:

Social Media:

The main use case here is to interact with people about their culture. That interaction includes adding friends, making posts, and interacting with those posts.

Wiki:

The main use case here is to learn about different cultures and the small amount of interaction that can be done with each post.

Features:

The two key features of the website will be the social media side and the wiki side. With the social media side, users will be able to interact with people using posts, and under those will be reactions and comments. The wiki side will also include comments and reactions but be much better for learning due to direct user interaction.

Objectives:

The testing strategy is iterative and should is built upon what was worked on in the previous week. Every use case will need to be tested and accounted for to ensure every bug is spotted and fixed before the IT expo.

Test Logs and Procedures:

Testing is done by using the deployed website. Each use case will be broken up and tested individually. When a bug is identified, it is added to a table to keep track of all of them. Once the bug is fixed it can be reviewed again and the status can change.

QA / Bug Testing Wiki:

Table 3.1

Issue #	Name	Description	Result
1	Navigation display	Navigation of page should be displayed when in a particular post. For example "Search By country > India > Pizza"	Fail
2	Country navigation via the map	When navigating to the country via the map it opens a new tab, which should be the same tab.	Fail
3	Text spillover on mobile	(Noticeable on mobile) when opening any wiki note with a long title, no frame is created and text runs over the edge and extends the page horizontally, while the navigation bar and everything else stays the same size.	Fail
4	New Wiki	When a new Wiki is done being created within a Country, you should not go back to the	Fail

		overview of all countries but stay within the county that the new wiki is created or end up on the page where you can view the new wiki.	
5	Wiki Title Limit	No limit on letter count for a title when creating a new wiki. The title should have a limit, so it is to the point.	Fail
6	Unknown URL	When an unknown URL is entered it should produce something like "Page not found" instead of http://cultureshock.ddns.net/Wiki/CountryPage?countryID=790.	Fail
7	Wiki Edit	No way to remove/edit a wiki post once added, need either a button to report a post, delete a post, or edit a post.	Fail
8	Country Name	The name is different on the map when hovered over vs when a country page is opened for countries such as Laos, South Korea, and a few others.	Fail
9	Adding Wiki Pictures	When a photo is uploaded it is only saved as the cover image for the post	Fail

		and cannot be seen within the wiki post.	
10	Extending map	We could extend the map to the whole screen after the navigation bar for a better user experience.	Fail
11	Country Names Flags	Country names are not displaying correctly and also some flags are not showing up. “Eswatini (fmr. "Swaziland")”	Fail
12	Populate Countries	Something we need to look into is populating countries with information to allow first-time users of websites to become regulars.	Fail
13	Label Ocean	We could add each ocean as its own page as well.	Fail

QA / Bug Testing Social:

Table 3.2.1

Issue #	Name	Assigned to	Description	Result
1	Unverified Email	Not Assigned	Email is not verified when creating an account.	Fail

Security Testing:

Table 3.2.2

Test #	Name	Results	Description
1	Resolve IP	184.54.182.140	The IP address was resolved using website
2	Ping	Average = 250ms	100% packet received
3	Ports	4 Open Ports	80,443,1433,3389

4	SSL Certificate	Valid	Issued Date - Jan 31, 2022 Expiry Date - Apr 1, 2022
5	Blacklist check	Not blacklisted by anyone	Checked by many different lists
6	Availability Check	Available from all countries tested	Tested website from many countries from each continent; all can visit website
7	Load test	Withstood load	Tested with 50 website visits every 15 seconds for 5 Minutes
8	Backup	Created	Backup is created however we only have 1 server, so no failover is in place currently

UAT:

Table 3.3

Client	Test	Feedback
1	Wiki Side by map	The map is very cool and fun to interact with. I love how you can select an area and it zooms into the map. I also like how alternatively you can scroll to zoom in.
2	Wiki Side by map	The map is still only one color, and within a box. I would love to interact with it a little bit more. Instead of keeping it in the box, you should let it take over the screen after the navigation bar. Navigation Bar MAP map MAP map MAP map MAP map MAP
3	Wiki Side by map	This seems like a fun website. You guys did an excellent job. The only suggestion for improvement is adding the ocean to add seafood or something along the lines.
4	Wiki search by country	This is a cool concept however you guys do not have any information in most countries. You guys should populate these countries with information to help users come on board. Overall looks good.
5	Wiki search by country	I think something you can improve on is making the pictures of the flag clearer and adding pictures to missing countries.

6	Wiki search by country	Some of the names of the countries are messed up such as “Eswatini (fmr. "Swaziland")” and also some countries do not have flags.

Testing Review:

Something learned during testing is that before creating the whole website we should render the webpage and show these renders to do UAT before starting. Then every week we should meet to have a retrospective on what has been tested following the guidelines. We need to identify the issue and get a plan for fixing it before the end of the meeting. The fix will fall onto who made that particular part of the project so we can fix it in a timely manner. Once fixed we need to test it again before it passes.

Change Management Plan:

If a change is required, which includes addition, modification, or removal of anything that could have a direct or indirect effect on our project please follow the following steps.

1. Any team member can make a change order, which is documenting and creating a list of what changes are occurring and how the project scope will be affected with pros and cons listed on the documentation. This change order must also list a criticality level for prioritization 1-5, 1 being most critical.
2. Once the change order is created, you need all three team members to sign off on the change order, giving consent for the changes.
3. Lastly, this must be reviewed/approved by the stakeholders like the advisors.

4. Once reviewed and approved, we can implement it in the order of criticality level.

Budget:

Table 4 Project Budget

	Rate Per/Hr	Work Effort (Hours)	1 X Costs	Ongoing Annual		
				Rate Per/Hr	Work Effort (Hours)	1 X Support Cost
Labor - IT	20	800	$16,000.00	20	200	$4,000.00
Labor - External	0	0	$-	0	0	$-
Software - External			$144			$144
Hardware - External						
Misc.						
TOTAL			$16,144.00			$4,144.00

Problems Encountered and Analysis of Problems Solved:

Throughout the development process, there were a couple of issues that arose, and some of them were novel issues that we had not experienced before. One of the issues was how to host the website and where to host it. Most of the group's experience was with working with existing projects, so deploying and hosting a web application was something that we did not have experience with. This issue was resolved by conducting research on setting up and deploying a web service on a home server. To do this, we researched port forwarding and hosted the website on David's home computer, which he set up to be used as a server.

After we figured out where to host our web application, there were a couple of issues that we ran into regarding IIS settings and getting a hostname. When we tried to deploy the web application, IIS kept throwing configuration errors. This problem was solved by speaking with

the advisor for some direction and conducting more research. We ended up resolving the issue through trial and error. As for getting a hostname, we went through a couple of free services before settling on one that gave us the hostname we wanted.

Those were the infrastructure issues that arose. There were a couple of minor issues regarding the code and database as well. One minor issue was getting images to work with SQL databases. An issue that arose was how to store images in the SQL tables. Since our application allows users to upload images, it was important to make sure that those images were stored properly in the database for retrieving later. To solve this issue, we looked online for guidance, and we ended up converting the image into a string using the code, and then storing that in the database since that is an accepted data type in SQL.

One thing that was not necessarily an issue but something the group had to decide on was the structure of the code. As developers, we wanted to follow best practices and practice separation of concerns. What we ended up doing was drawing knowledge that we had from work experience modeling the structure of our project after our enterprise development experience. The code project has a clear separation of business logic, data access logic, and presentation logic.

Conclusion

Lessons Learned:

There were a few good lessons that we learned throughout the semester, like patience, development time, and how to audit reports. When it comes to patience it is good to have, and helps with resilience, if you can figure it out but keep trying you can get it done. On development, there is a lot more that goes into it than just the actual coding and developing.

Other things to consider are budgeting, timeline, and keeping the end-user in mind. With report auditing, it is important to keep a high-level relationship between the end-user and the product in mind. Relating the product to the end-user keeps the information in the report relevant.

Abilities/skills developed or enhanced:

- SQL skills have been developed a lot.
- Full stack skills have been enhanced.
- CSHTML skills have increased.
- Learned the skill to host a website from a desktop.
- Testing skills have been developed.
- Teamwork and communication

References

Puloka, D. S. (n.d.). *Cultural Identity and Heritage Languages*. Cultural Identity and heritage languages. Retrieved September 20, 2021, from https://www.mckendree.edu/academics/scholars/issue15/puloka.htm.