# NAUTICAL ARCHAEOLOGY DIGITAL LIBRARY

An Undergraduate Research Scholars Thesis

by

MUHAMMAD NAUMAN

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Richard Furuta

May 2021

Major: Computer Science

# RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Muhammad Nauman, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

# TABLE OF CONTENTS

# ABSTRACT

Nautical Archaeology Digital Library

Muhammad Nauman
Department of Computer Science & Engineering
Texas A&M University

Research Faculty Advisor: Richard Furuta
Department of Computer Science & Engineering
Texas A&M University

A problem that exists in the Nautical Archaeology Digital Library at the Texas A&M University is that there is no easy way to organize a database of images incorporating good filtering, tagging, and large cropping capabilities. One goal of this research is to study and evaluate different web-based systems that will allow a community of scholars to upload and search images. It will allow the person uploading to assign metadata to the image/painting and also allow them to crop certain parts (the original will be preserved) of images to highlight and assign more metadata to the subpart(s). This metadata includes terms to categorize the ship or image, which enables filtering/searching capabilities. Furthermore, the system allows the admins to password-protect specific images, if needed. This system will also preserve images, with data, and provide a central place for images. Studying how this system can be implemented and how nautical archaeologists categorize ships will be very beneficial to the Nautical Archaeology community. This research is significant because it will help in building a central database for conservation of nautical archaeology evidence and items in one central database with metadata assigned to it. To provide long-term stability, this thesis also examines current cloud-based

mechanism for hosting the site. This thesis also examines how data about ships with coordinates can be mapped to an easily accessible and editable map.

# ACKNOWLEDGEMENTS

## Contributors

I would like to thank my faculty advisor, Dr. Richard Furuta, for his guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to the Nautical Archaeology team at Texas A&M Universiry for their encouragement and support throughout the course of this research. I would like to especially thank Kyle Pence, Austin Griffin, and Professor Felipe Castro, for their guidance and help in this research.

All other work conducted for the thesis was completed by the student independently.

## Funding Sources

This research received no funding and no funding was needed. However, the NADL project as a whole is based upon work supported by the National Science Foundation Nautical Archaeology Digital Library, under Grant No. IIS-0534314 (2006-2009), and a Texas A&M T3 Grant Dead Ships and Live Culture in Coastal Communities (2018-2020). It has also drawn from the work developed within a series of other grants, such as ForSEA discovery, iMARECULTURE, GROPLAN, and A Dive in History.

# NOMENCLATURE

| | |
|---|---|
| NADL | Nautical Archaeology Digital Library |
| CS | Computer Science |
| TAMU | Texas A&M University |
| AWS | Amazon Web Services |
| GB | Gigabyte |

# 1. INTRODUCTION

The Nautical Archaeology Digital Library is a tremendous effort created by joint contributions of several departments at the Texas A&M University. The main purpose of this archaeological project is to share the knowledge acquired with both the domain experts and the general public. A part of the vision of NADL is to “efficiently catalog, store, and manage artifacts and ship remains along with associated information from underwater archeological excavations” [4]. The project described in this thesis will investigate how to implement that functionality in the NADL website. The research reported here will also allow the images and paintings to be cropped and information associated to the subparts.

Evidence recovered from excavations has to be processed and stored to make sure it is preserved and conserved [2]. Most of the evidence can be photographed and stored in a database. This thesis will study how that can be done and what implementation will be the best for the database web-system. This helps in categorizing, storing, and preserving images and paintings of nautical archaeology objects.

The images web system was started by Austin Griffin and then worked on by Kyle Pence. Currently, I am working on it. The goal is to have a database of images with good cropping and filtering capabilities. This database has an About page explaining the functionality and importance of the database. It also has a Database tab where registered users will be able to upload images, assign metadata to the images, and crop and highlight specific parts of the images if needed. Then it has a Search tab where users can search and filter images. A stretch goal would be to implement some kind of a map in the Map tab, so we can have a visual representation of the data. There are also be Register and Login tabs so there aren’t any unnecessary images in the database and that users can be maintained. This database will also have a Contact tab for visitors to submit questions, feedback, and comments to us. The prototype is shown in Figure 1.1 below.

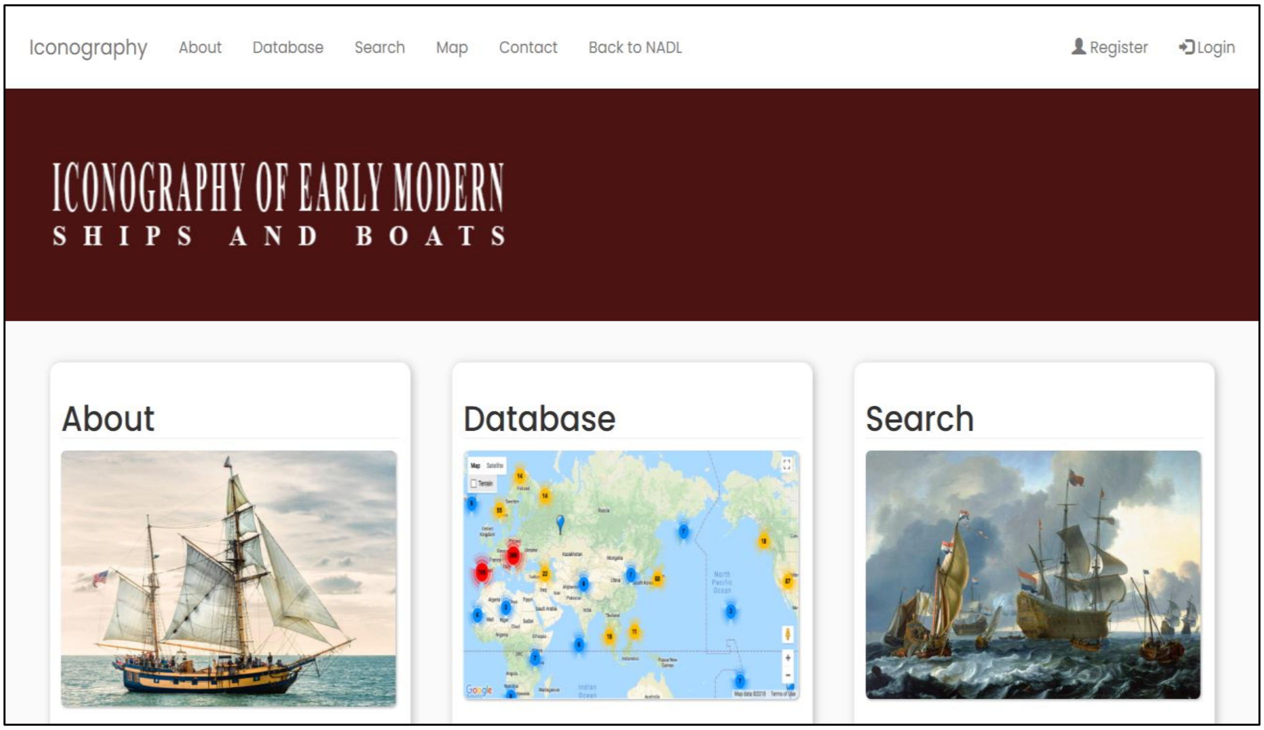


*Figure 1.1: Prototype of the image web system.*

## 1.1 Problems

A problem that exists in The Nautical Archaeology Digital Library at the Texas A&M University is that there is no database of images with good filtering, tagging, and cropping capabilities. To solve this problem, we take a look at different ways this can be implemented. The ideal implementation will allow the person uploading to assign metadata to the image/painting and also allow them to crop certain parts of the image to assign specific data to. The system will also have tags and keywords to ease filtering and searching. A map implementation for a visual representation for the data will also be helpful. Another issue with the NADL at TAMU is that the website is hosted on physical servers, which may be unreliable.

To solve this problem, we take a look at different cloud-based virtual servers that can be used to host the website and also make it easy to edit for future use. Since the NADL site is already established on physical servers, we would also need to make sure that the users, relationships between tables, and databases are also imported. Currently, we are in the process of moving the website from nadl.tamu.edu (physical TAMU servers) to shiplib.org (virtual server).

Another question this research project tackles is what would be the best way to find the coordinates (i.e. latitude, longitude) of data given descriptive locations, or cities, countries, etc, and to evaluate how certain these coordinates are by picking a radius. The smaller the radius, the more certain and true the coordinates will be. Finally, the thesis evaluates what the best method would be to visualize and map this data. This data can be of ships that have been sunk, lost at sea, destroyed, etc. A sample of the data can be seen in Figure 1.1.1. "Approx Lat", "Approx Long", and "Radius" would need to be filled in.

| | A | V | W | X | Y | Z | AA | AB | AC | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Name 1 | Country | Area | Location 1 | Location 2 | | Approx Lat | Approx Long | Radius | Bib. ( |
| 3 | unknown | | | Near Aguada de S. Brás / Mossel Bay | | | | | | |
| 4 | *São Rafael* | | | Baixos de São Rafael | | | | | | |
| 5 | *unknown* | | | | | | | | | |
| 6 | unknown | | | | | | | | | |
| 7 | unknown | | | | | | | | | |
| 8 | unknown | | | | | | | | | |
| 9 | unknown | | | Coast of Melinde | | | | | | |
| 10 | *São Pantaleão?* | | | Parcel de Sofala | | | | | | |
| 11 | *Esmeralda* | | | Curia Muria Islands | | | | | | |
| 12 | *São Pedro* | | | Curia Muria Islands | | | | | | |
| 13 | *N Sra da Conçeição* | | | Baixos de São Lázaro | | | | | | |
| 14 | *Rainha* | | | Baixos de São Lázaro? | | | | | | |
| 15 | *Faial* | | | shallows of São Lázaro | | | | | | |
| 16 | unknown | South Africa | | close to Aguada de São Brás / Mossel Bay | | | | | | |
| 17 | unknown | | | close to Cabo das Correntes | | | | | | |
| 18 | *Bela* | | | coast of Guinee, latitude of Equador | | | | | | |
| 19 | *Santiago* | Portugal | | Barra of Lisbon | | | | | | |
| 20 | *São João* | | | Mozambique or Quiloa | | | | | | |
| 21 | *Santiago «Galega»* | | | leaving the barra of Quiloa | | | | | | |
| 22 | *São Vicente* | | | Coast of São Lorenço or Terra do Natal | | | | | | |
| 23 | *N Sra da Luz* | | | Pate | | | | | | |
| 24 | *Leitoa* | | | Mozambique | | | | | | |
| 25 | *São Romão?* | | | between Sofala and Mozambique | | | | | | |

Lists entered | Ship List | Ship List (2) | Archives | Bib.Refs. | Maps

Ready 150%

*Figure 1.1.1: Sample of the data to determine latitude, longitude, and radius based on country, area, location 1, location 2, etc.*

## 1.2 Importance/Significance

Seafaring and ships have been used for traveling, trading, exchanging of ideas, sharing ways of life, etc. for a long time [3]. Preservation of information and artifacts of ships and seafaring is very important when planning the recovery of material from a marine archaeological site [1]. Without conservation, important historic data will be lost, which will be a loss for future archaeologists who wish to study or reexamine the archaeological site [1]. Evolution of ships and advancements of science and technology can be seen and studied through the preservation and conservation of these artifacts [3]. This project will allow future generations to study and research multiple maritime and nautical topics in one central place. Visualizing the data of ships will allow researchers to see where the ships have been lost and perhaps extrapolate something from the visualization. Also, it will allow nautical archaeologists to decide what the best place to excavate for nautical artifacts would be.

# 2. METHODS

The method of conducting this research is to first study and read about different ways the problem can be solved. After that, we implement a small scale or part of the solution to see the ease and usability of the solution.

## 2.1 Methods to Improve the Storing and Uploading of Images

To research and evaluate different implementations and technology stacks for creating a web-based system to upload, store, sort, and filter Nautical Archaeology images, I did the following. First, I researched and read about what information and data is most important to categorize and store with the image. In other words, what metadata when attached to the image will allow users to easily search and access images, as well as give information and context about the image to learn more. Secondly, I looked at how hard would it be to implement a cropping mechanism for images so the users can crop specific parts of images and assign more data to it. Finally, I created a small scale implementation with the programming language or technology stack I am researching to see what problems we may run into. I then shared these results with my faculty advisor and/or the NADL team to receive feedback and areas of improvement.

As mentioned in the introduction, the ideal web system will have 5 different tabs: About, Database, Search, Map, and Contact. Some optional steps for future improvement would be to check how hard it would be to create a map display of the images and data uploaded. Some kind of Registration and user database will also be helpful to track which images were uploaded by which user. Another useful feature to add would be an export button to export the data from the site into an excel sheet or some other format. This can be useful for researchers to edit and analyze the data themselves.

## 2.2 Methods to Evaluate and Research Virtual Servers for NADL

To research and evaluate where we can move the NADL site from the physical servers at TAMU, I followed these steps. First, I researched and read about the virtual server or hosting service. Second, I saw if it is compatible with WordPress. WordPress is an easy to use website editor to create webpages, articles, blogs, etc. with very little programming. Since almost all of the pages of the NADL website are created, edited, and updated using WordPress features, this step was important. Furthermore, there are many essential WordPress plugins that are used in the NADL site. Third, I checked if we can host custom pages made with HTML, PHP, etc. on the same servers. This is also important because the image uploading web system is implemented in this way. Finally, I estimated and found out approximately how much it would cost. The pricing can be estimated either per GB or for the whole of NADL site.

After these initial steps, the next step was to check how the already established relationships between the tables in the database could be imported over. These tables and relationships include the users table, the plugins table, and the relationship between different sections and headings of the content. This step was important because it would be cumbersome to recreate the users table and give all the users the privileges they had before.

We also needed to check what backup capabilities the server had and would these capabilities cost more or not. One of the main problems with the physical servers at TAMU is that they are unreliable and sometimes go down with no warning. This can be troublesome since someone needs to physically go and check why the servers are down.

## 2.3 Methods to Geocode and Map Data

To find coordinates based on the given data, a program was written using the Google Maps API and the MapQuest API. Despite using these APIs, most of the coordinates could not be found due to the location being descriptive or in a foreign language. So, some of the coordinates were filled in manually using Google, Google Maps, etc. It would be a good idea to harmonize or standardize the location column so coordinates could be filled in automatically. To visualize the coordinates on a map, different APIs, programming languages, and online services were tried.

# 3. RESEARCH

Many different systems, servers, technology stacks, and programming languages were researched and evaluated for both the images web-system and virtual servers. The results are summarized below.

## 3.1 Images Web-System Research

I researched, evaluated, and implement on a small scale many different ways the images web-system can be implemented. Some of the options were great for one aspect of the site, but lacking in some other aspect.

### *3.1.1 Python and Django/Flask*

Python can be used to make this image system. There are a lot of libraries and documentation available online to do this. However, the cropping mechanism from the user side only works if it is used in a desktop app, and not on the website. Since we are trying to implement images web-system as part of the NADL website, this is a big obstacle. The geographical map to display where the images are around the world can be implemented using different libraries, such as pandas or geopandas.

### *3.1.2 WordPress*

It is really easy to make forms and webpages in WordPress. However, these webpages are in a blog format and are created using drag and drop. This is fine for the general NADL site, but for the images web-system, which needs cropping mechanism, this is not fine. Furthermore, there is no way to let the users upload the image they desire since they cannot crop it. Since WordPress already have a user system built-in, it may be useful. So overall, WordPress would not be a good stack to implement the images web-system in, especially since it would be very hard to implement the cropping capabilities.

### *3.1.3 NicePage*

NicePage is an application that you can use to drag and drop components of a webpage in. After that, NicePage generates the HTML and CSS code automatically. This code can be placed in a webpage. However, NicePage also has the same problem of not having a block to crop images based on user preference. So even though NicePage can be used to generate a part of the web-system, it cannot be used to generate the whole system.

### *3.1.4 HTML, CSS, and JavaScript*

Since Python does not have support for cropping images in a browser, I tried using JavaScript. HTML and CSS was used to make the form, while JavaScript was used to let the user upload and crop images. This combination of languages worked well. However, it is a challenge to save the images and information submitted using these languages.

### *3.1.5 HTML, CSS, JavaScript, PHP, and SQL*

As mentioned in section 3.1.4, HTML, CSS, and JavaScript work well together, but that system needs an uploading/saving mechanism. To implement the uploading/saving part, PHP and SQL were used. A challenge with this implementation is that when the images and data are stored, it is a bit difficult to access them again. Since we would also need to implement a filtering/searching mechanism, this can be challenging.

### *3.1.6 Angular, HTML, CSS, JavaScript, PHP, and SQL*

To solve the challenge of accessing the saved images and metadata, the angular framework can be used. We can create custom APIs to access the different characteristics of the images and data to filter and search all the images, which would use PHP. So the combination of

Angular, HTML, CSS, JavaScript, PHP, and SQL would be best to implement the images web system.

## 3.2 Virtual Servers Research

Some different virtual servers and hosting services were researched and evaluated. I tried different services out and I summarize my findings below.

### *3.2.1 AWS LightSail*

AWS Lightsail is an easy-to-use virtual server that is user friendly and very affordable. Since it is one of the biggest virtual server hosting services, there is a lot of documentation available to learn and use it. It is easy to create a WordPress instance on the Lightsail server since it’s built in. Furthermore, custom web pages can be added using secure remote connections, such as FileZilla, or by simply using the command line. Since WordPress has plugins, the All-in-One WP Migration plugin can be used to move the site to Lightsail. AWS Lightsail also has backup/snapshot capabilities, so the instance can be reverted back to a previous point, if needed. Users will not know the difference since the site will still be WordPress. Also, Lightsail supports development and sandbox environments which can be helpful.

### *3.2.2 DreamHost*

DreamHost is another option for a virtual server to host WordPress. It is also affordable and easy to use. The WordPress site can be moved using the same All-in-One WP Migration plugin or DreamHost Automated Migration plugin. A free domain is also included if you sign up for 1 or 3 year plan. It is harder to add custom web pages to this instance. There is also an option to add custom email addresses, which may be helpful in the future. DreamHost is more expensive than AWS Lightsail, but still pretty affordable, especially if an annual or three year plan is purchased.

#### *3.2.3 Google Cloud*

Google Cloud is another option for a virtual server. It is much harder to use and not as affordable. Google Cloud services are usually more expensive. One nice thing about it is that it has a visual representation of folders of the whole system so it is easier to visualize. We can again use the same All-in-One WP Migration plugin to migrate the website. It may be hard to create or add custom web pages to this instance. Google Cloud may be a good choice if other Google Cloud products would be needed in the future.

#### *3.2.4 GoDaddy*

GoDaddy is also an option for the virtual server. It is more expensive than DreamHost and AWS Lightsail, but less expensive than Google Cloud. It comes with a free domain, free business email, and free SSL certificate. We can migrate using the same All-in-One WP Migration plugin. GoDaddy also has backup capabilities with 1-click restore. Since GoDaddy also does website hosting, it would be useful if we need to access cPanel or SQL databases. CPanel is a web hosting software to view the website structure graphically.

### **3.3 Research to Geocode and Map Data**

To find the coordinates of the given data, programs were written using the Google Maps API and the MapQuest API. These programs were written in HTML and Python. This resulted in some of the data being filled in. The rest of the data was filled in manually using Google and Google Maps. To visualize this data, I tried to create a webpage using HTML, JavaScript, and CSS. Some of the challenges I ran into was converting the Excel file to JSON objects, trying to find a good and free API to map the data in, trying to create an import function, etc. Since creating a webpage was challenging, some other service, like Maptive or Google Maps would be better. Maptive is not free, while Google Maps is free. Furthermore, Google Maps can be shared and embedded.

# 4. RESULTS

## 4.1 Results of Images Web-System Research

After conducting the necessary research, the combination of Angular, HTML, CSS, JavaScript, PHP, and SQL was determined to be best to implement the images web system. HTML would be used to create the base architecture and base website. HTML would also be used to create the form. CSS would be used to stylize the website and to create a theme. Since JavaScript is very compatible with HTML, JavaScript would be used to create the cropping mechanism. Since WordPress is structurally built using PHP, PHP would be used to add custom APIs if needed and also if we need to connect this Web-System back to the NADL site. PHP would also be used to implement some of the registration and login capabilities. SQL would be used to create the database and to store the data and images entered by the users. Finally, the Angular framework would be used to connect all of the different tabs together. The Angular framework would also be used to import in different themes and styling components.

## 4.2 Results of Virtual Servers Research

After trying out different virtual server hosting services, Amazon Web Services Lightsail would be ideal to host the NADL site on. There are many reasons Lightsail would be ideal to migrate the site to. Lightsail supports WordPress instances and makes it very easy to import existing WordPress sites. Lightsail has the Linux command line built in which is very similar to the TAMU physical servers command line interface. It is also pretty easy to connect to a LightSail instance externally. There is also extensive documentation available to aid in troubleshooting problems, if they arise. It is worthwhile to note that the Lightsail instance is running Debian 10. The physical servers are running CentOS. Both Debian 10 and CentOS are distributions of Linux. A disadvantage with CentOS though is that it doesn't seem to be getting much recent support.

LightSail is also one of the most affordable virtual server hosting services. Lightsail also has backup capabilities with the snapshot feature. It also supports adding custom webpages using the command line or FileZilla or similar services. Lightsail also supports development and sandbox environments which can be helpful if new features need to be tested.

### 4.3 Geocoding and Mapping Data Results

After conducting the necessary research, I determined it would be best to fill in the coordinates of the given data manually to make sure there are no discrepancies. Furthermore, there is no easy way to determine how accurate the coordinates given by different APIs are. This might change if the data is more standardized.

To visualize and map data, Google Maps would be the best service to map data in. Custom maps can be built in Google Maps by clicking on the hamburger (three vertical lines) menu on maps.google.com and choosing “Your Places.” Then, click on Maps on the top bar and then Create Map on bottom. Google Maps offer multitude of features. The Google Maps would be constantly updated by Google. Also, it is incredibly easy to import in excel files and visualize the data. The map of the given data can be shared using a link and it can also be embedded in webpages or WordPress sites. Furthermore, individual markers can be edited or deleted to update the map.

# 5. CONCLUSION

## 5.1 Conclusion of Images Web-System

In conclusion, for the images web-system, using a combination of Angular, HTML, CSS, JavaScript, PHP, and SQL would be best to implement the images web system with cropping, filtering, and searching capabilities, as well as adding registration and mapping capabilities. This web system is being currently developed.

## 5.2 Conclusion of Virtual Servers Research

For the virtual servers, AWS Lightsail would be the best service to host the NADL site on due to it user-friendliness and affordability. It offers a multitude of features including WordPress instance, Linux command line, SSH and SFTP capabilities, backup feature, extensive documentation, etc. We are currently in the process of migrating the NADL site to AW Lightsail instance on ShipLib.org.

## 5.3 Conclusion of Geocoding and Mapping Data Research

For the geocoding of data, it would be best to manually enter the coordinates to make sure they are correct. Google Maps and Google at large can be used to find the coordinates. It would be ideal to standardize the data. This can be done by having city, then location, so these columns can be read automatically by APIs. It would also be a good idea to have all of the data in English.

For the mapping and visualizing of data, Google Maps would be the best service. There are many reasons for it. It is easy to import in Excel files, it can be worked on by multiple people, it can be shared, it can be embedded, etc. A sample of how the data would look in Google Maps can be seen in Figure 5.3.

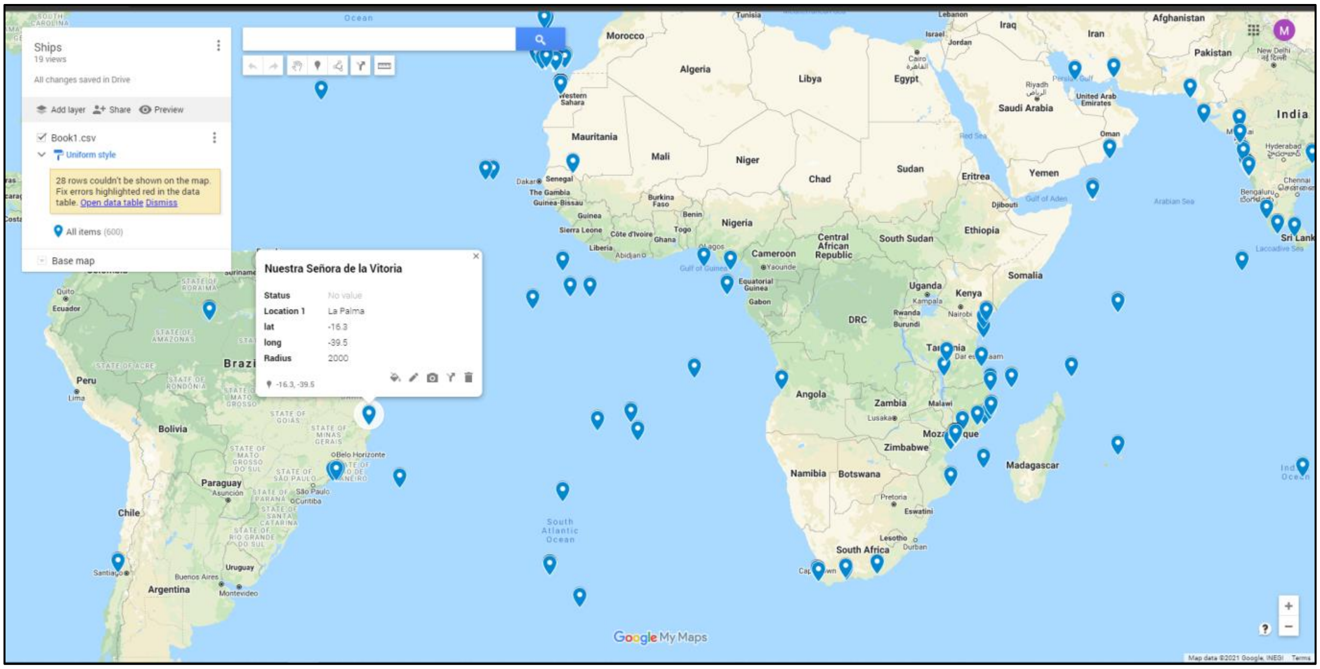


*Figure 5.3: Visualization of Data in Google Maps*

# 6. FUTURE WORK

The future work for the images database web-system would include completing the web-system and then continuously improve it. The web-system would require maintenance often to make sure everything is working correctly. The maintenance would also check if all APIs are still active and working correctly. There is also potential to add more features and functionalities to the web-system.

The future work for the cloud-based virtual servers would be to upgrade the AWS Lightsail plan, if needed. The AWS LightSail would also require some maintenance. There are a lot of functionalities in LightSail that can be studied and investigated for future work.

The future work for geocoding and data mapping would be to keep on adding more data to the map. This addition of more data will allow us to see where the shipwrecks are concentrated. The data should be cleaned and standardized in the future so it is easier to use and analyze the data. The Google Maps should be embedded into the necessary WordPress sub sites as needed.

# REFERENCES

[1] Donny L. Hamilton. (1998). Methods of conserving underwater archaeological material culture. Conservation files: ANTH 605, 34.

[2] Carlos Monroy, Richard Furuta and Filipe Castro. (2012). A Digital Library Perspective: The Synthesis and Storage of Maritime Archaeological Data to Assist in Ship Reconstruction. DOI: 10.1093/oxfordhb/9780195375176.013.0015.

[3] Carlos Monroy, Richard Furuta and Filipe Castro. (2008). Design of a computer-based frame to store, manage, and divulge information from underwater archaeological excavations: the Pepper wreck case. Edge of Empire: p.275.

[4] Carlos Monroy, Nicholas Parks, Richard Furuta and Filipe Castro. (2006). The Nautical Archaeology Digital Library. 4172, 544-547. DOI: 10.1007/11863878_64.