

## CSCI 244 Web Development: Design Report – Emma’s Acres Alpaca Farm

Access using the following link:

[www.emmasacresalpacas.com](http://www.emmasacresalpacas.com)

*(Please note the link will not work with https currently, this will change soon as we continue to work with our clients)*

*If this link does not work, please use the following link:*

*Note that sometimes the page needs to be refreshed several times*

<https://emmas-acres-alpacas.herokuapp.com/>

*Nina Carlson, Isaac Schatia, Jonah Scudere-Weiss*

### Team Information:

Team Name: *The Alpaca Updaters*

Nina Carlson - [ncarlson@clarku.edu](mailto:ncarlson@clarku.edu)

Isaac Schatia – [ischatia@clarku.edu](mailto:ischatia@clarku.edu)

Jonah Scudere-Weiss - [jscudereweiss@clarku.edu](mailto:jscudereweiss@clarku.edu)

### Abstract:

We created a brand-new website for Emma’s Acres Alpaca Farm, an alpaca farm located not too far from Clark University’s main campus. While the farm had an existing website, it was created about 15 years ago. It was extremely outdated and hard to use. Our website is much more functional, user friendly, and aesthetically pleasing, while still catering to the requests of our clients. Our clients specified several aspects of the old website that they wanted to sustain in the new website, such as a calendar page, an alpaca facts page, and a picture of Emma the Alpaca somewhere on the website. We were able to include all these requests, while making a much more beautiful website for the farm.

### Introduction:

Emma’s Acres Alpacas is an alpaca farm located in Oxford, MA. The farm has been around since 2002, and is owned by Roger, Collette, and Amanda Gage. They are known for their knitted products, all made from alpaca fur. While the farm does have a website, it is extremely outdated (made approximately 15 years ago) and not necessarily aesthetically pleasing. Essentially, we worked with the Gage family for several months to create a newly updated website for Emma’s Acres Alpacas from scratch. The website is much more user friendly and more visually appealing. With a more appealing website, we hope that the farm will draw in more visitors. Any people who are interested in visiting the alpaca farm will benefit greatly from the website. The Gage family is warm and welcoming, and they love to greet visitors.

### Analysis:

Please refer to Appendix B for design materials and affinity diagrams. Please note that finding videos for our topic proved to be difficult, since we obtained all our necessary information about alpaca farms directly from our clients. Professor Niu approved this as a form of information gathering in place of videos.

## Design:

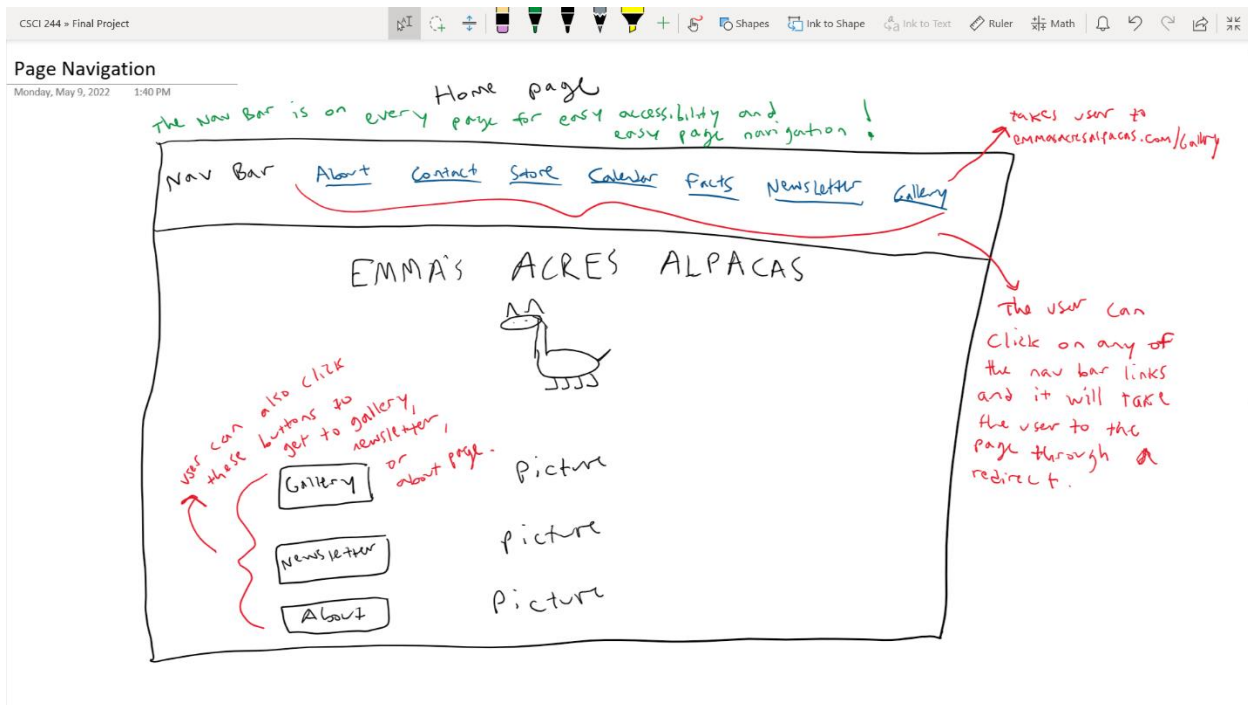
Please refer to Appendix B for the Wireframe design from Balsamiq. Note that this design was made before receiving feedback from the professor. Some initial designs may not match the finished website product.

## Implementation:

The website for Emma's Acres Alpaca farm is a full-stack, fully functional website that is publicly hosted using a domain name that our clients already purchased and were using for their previous website. A main challenge we faced was the deployment of the website and attaching our finished product to their domain name.

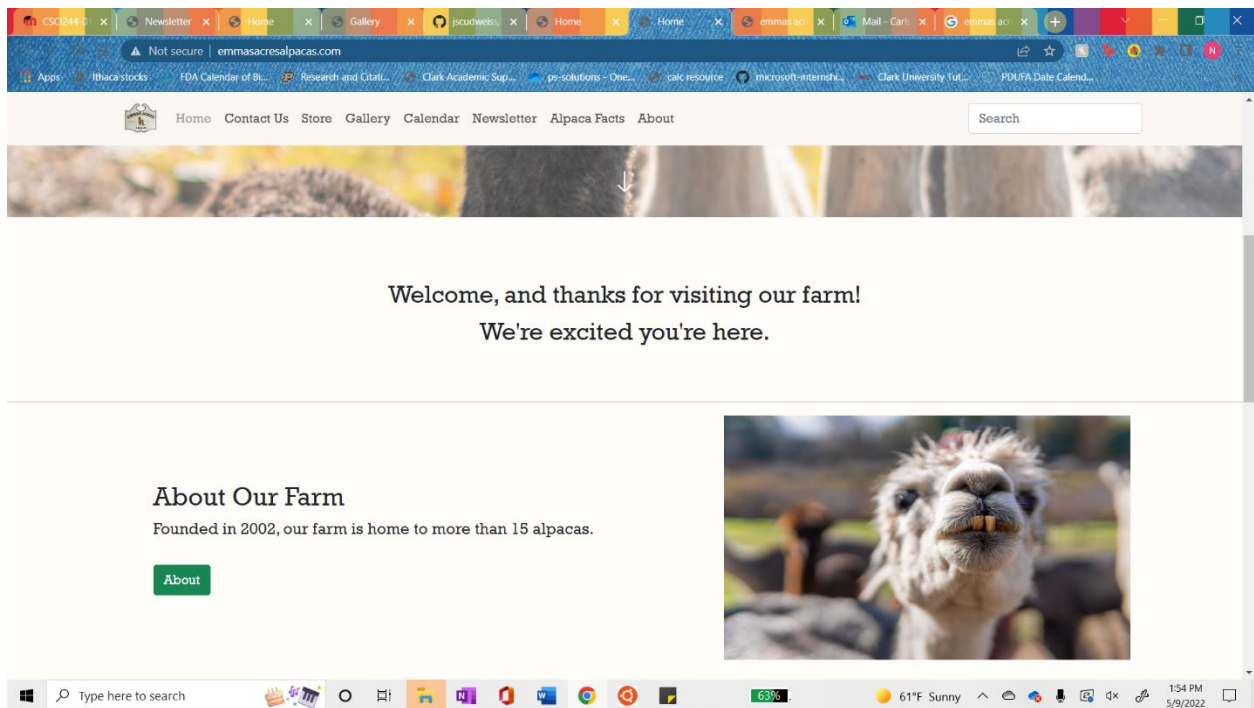
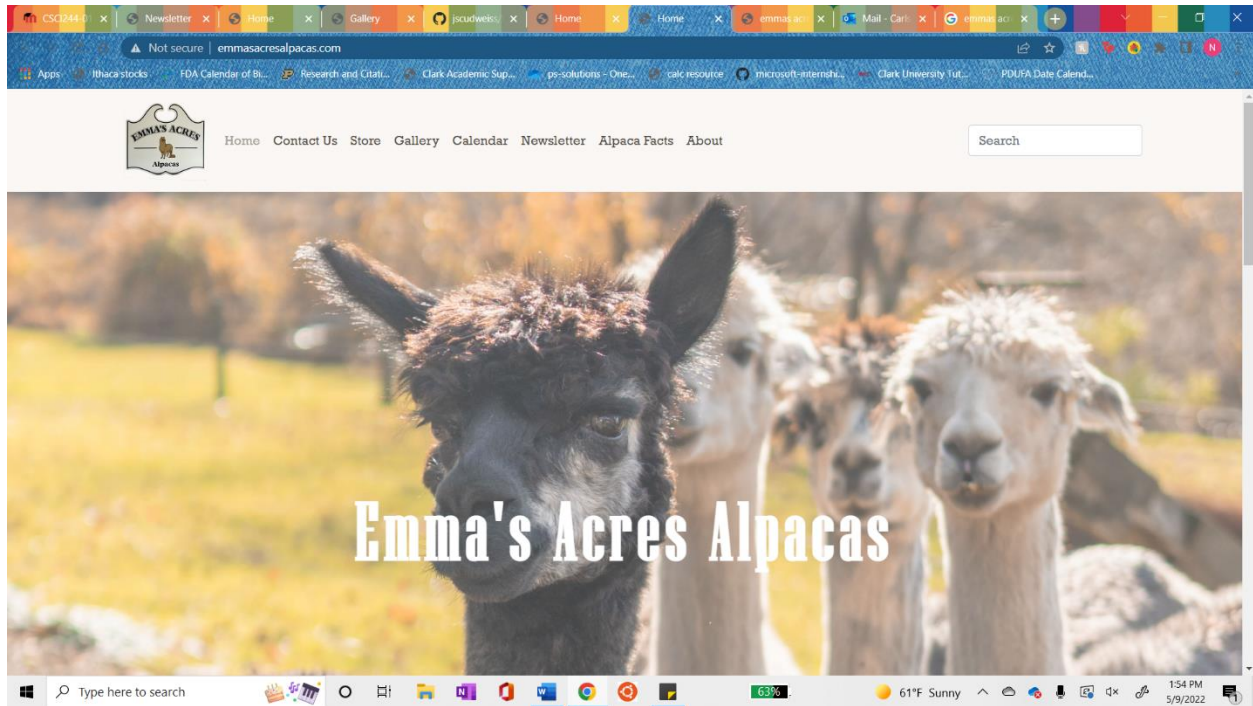
## Front End Design:

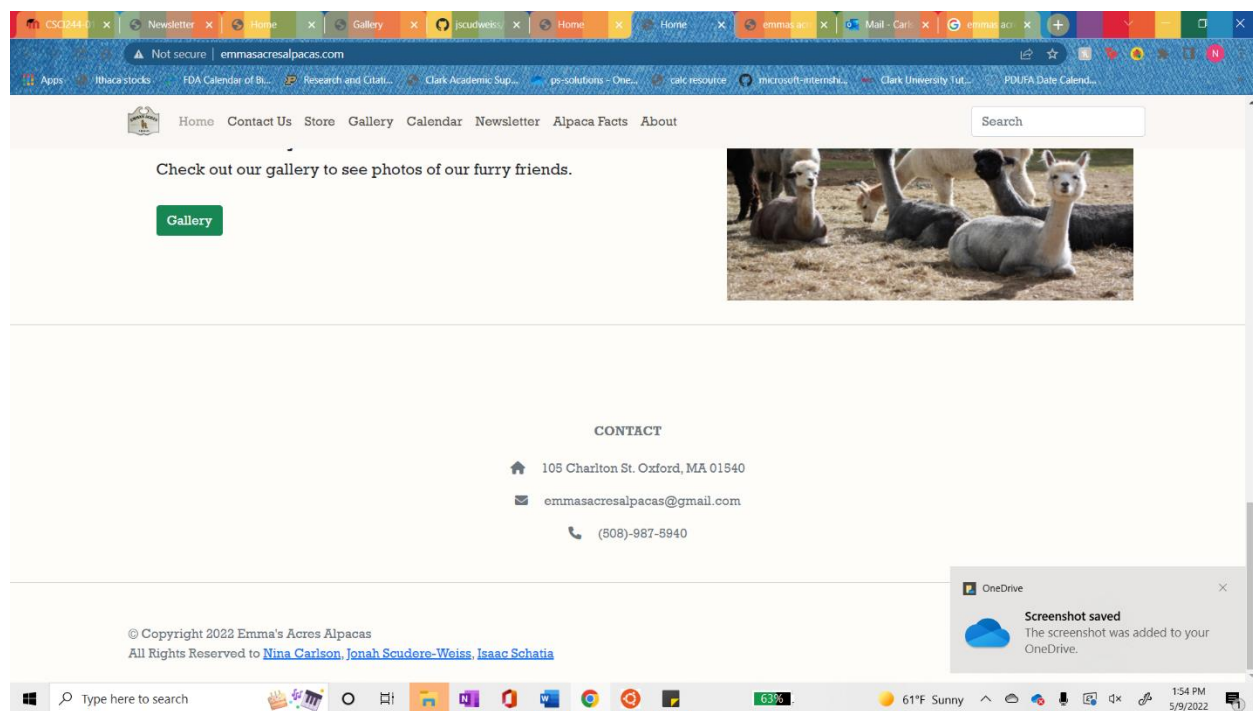
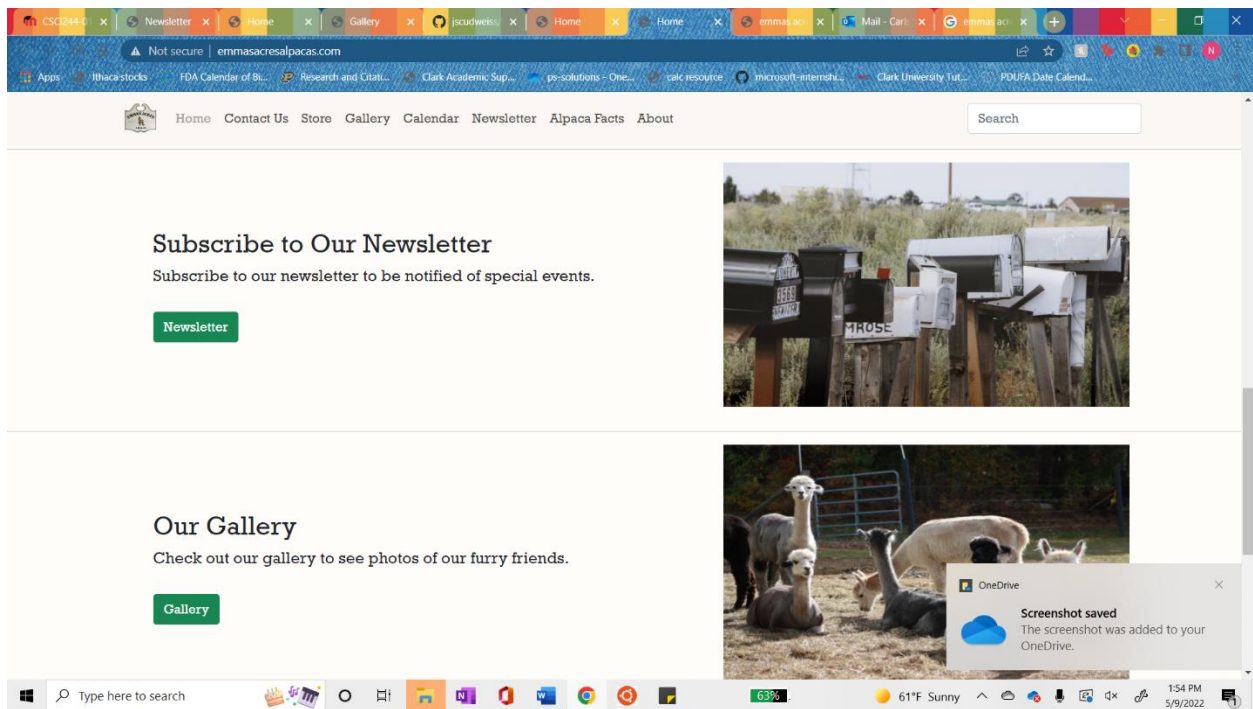
Diagram of how pages navigate to each other



## Screenshots of website and explanation for each page

### Homepage:

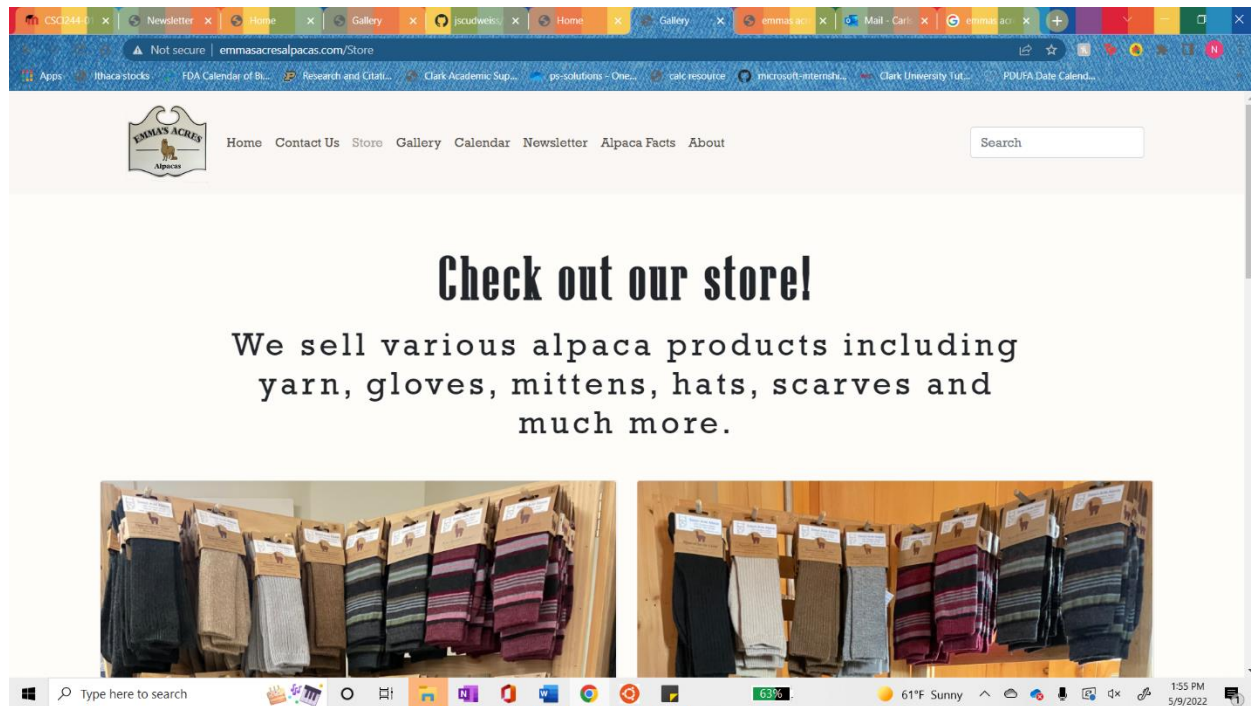
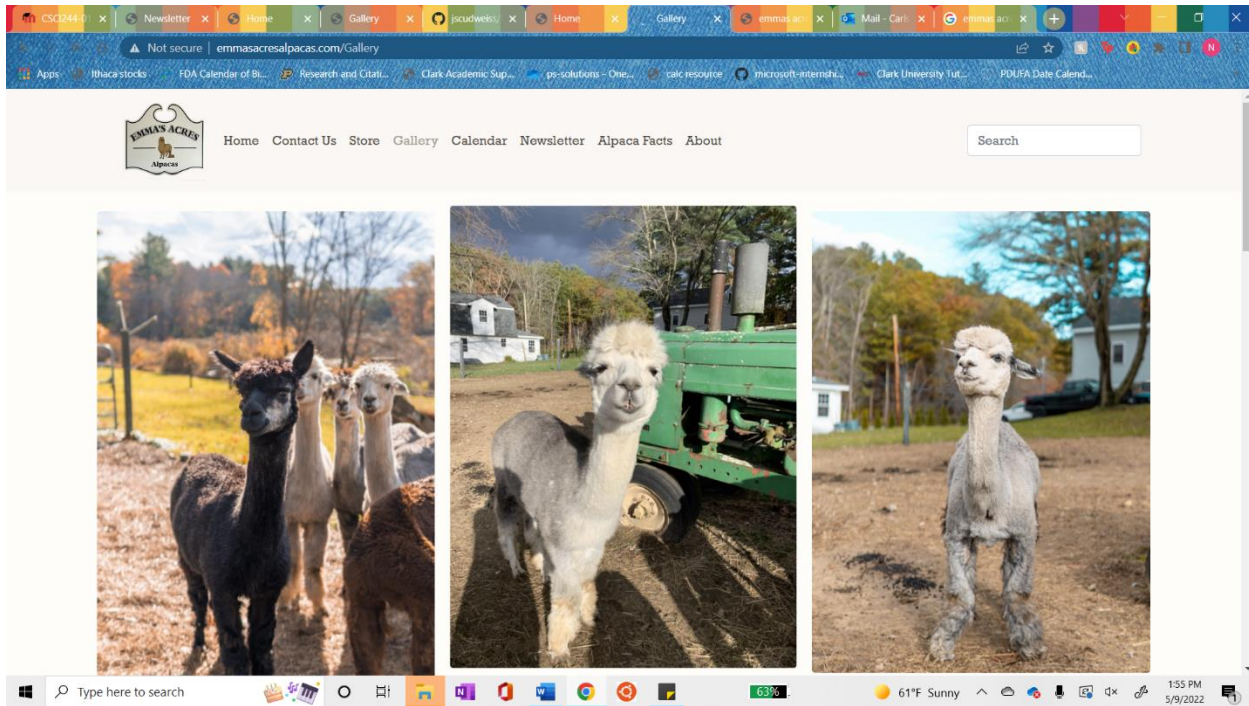




These first four screenshots are of the home page. The home page is the first page that visitors will see. They can scroll down and click the buttons to go to different pages or click the links in the nav bar. They can also use the search bar in the nav bar to search key words and go to specific pages based on key words. If they scroll down further, they find contact information.



**Gallery:** The gallery contains pictures of the farm so that users can see before they visit. If the user hovers their cursor over each picture, it will pop out.



**Store:** The farm is known for knitted products from alpaca fibers. The buttons on the store page under each image link directly the farm's store on square market. Users can shop on the square market for products by clicking on any of the links.

**Contact Page:** The contact page allows users to type in their contact information and send a message to the owners of the farm, inquiring about visits, products for sale, etc. The information goes into a database for our clients to access and keep track of so that they can answer messages from prospective visitors.

Not secure | emmasacresalpacos.com/Contact

Home Contact Us Store Gallery Calendar Newsletter Alpaca Facts About

## Contact Us!

Visits are always welcome! Just give us a quick call before heading over.

Name:

Email:

Phone Number:

Enter your message:

Roger & Collette Gage  
Amanda Gage  
105 Charlton St.  
Oxford, MA 01840  
E-mail: emmasacresalpacos@gmail.com  
Phone: (508)-987-5940  
Find us on Facebook!

Type here to search

61°F Sunny 1:55 PM 5/9/2022

Not secure | emmasacresalpacos.com/Contact

Home Contact Us Store Gallery Calendar Newsletter Alpaca Facts About

Name

Email:

Phone Number:

Enter your message:

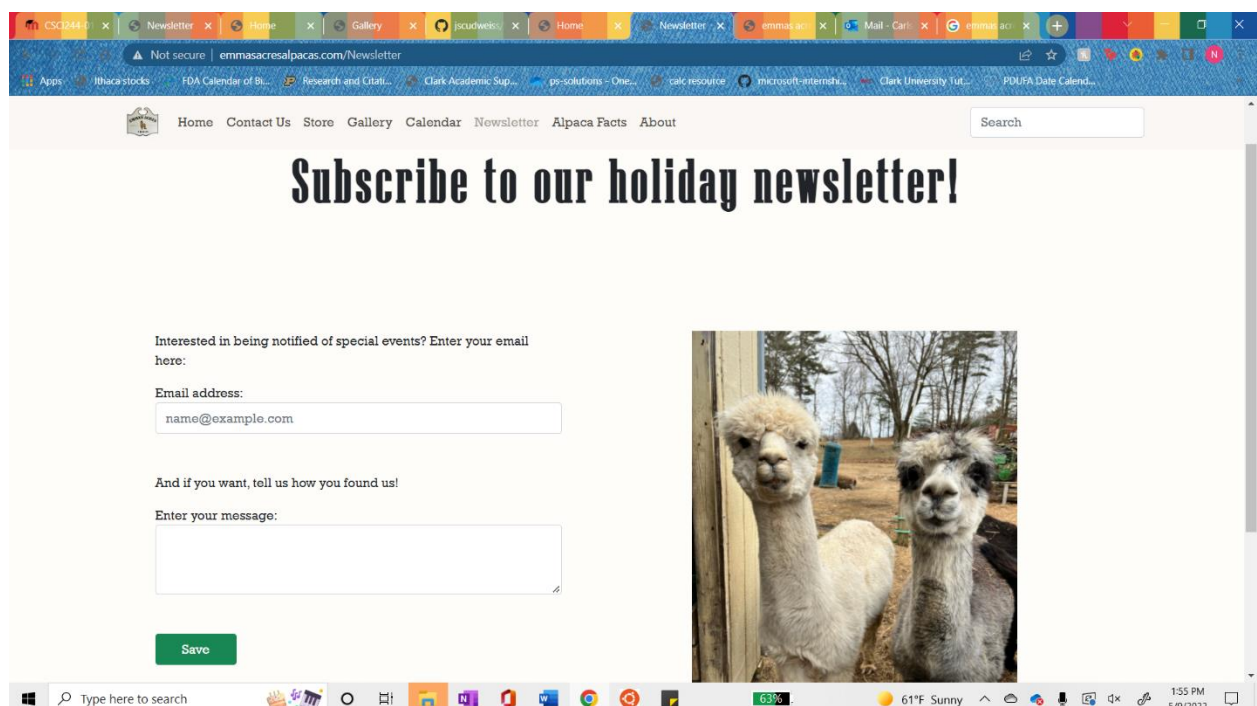
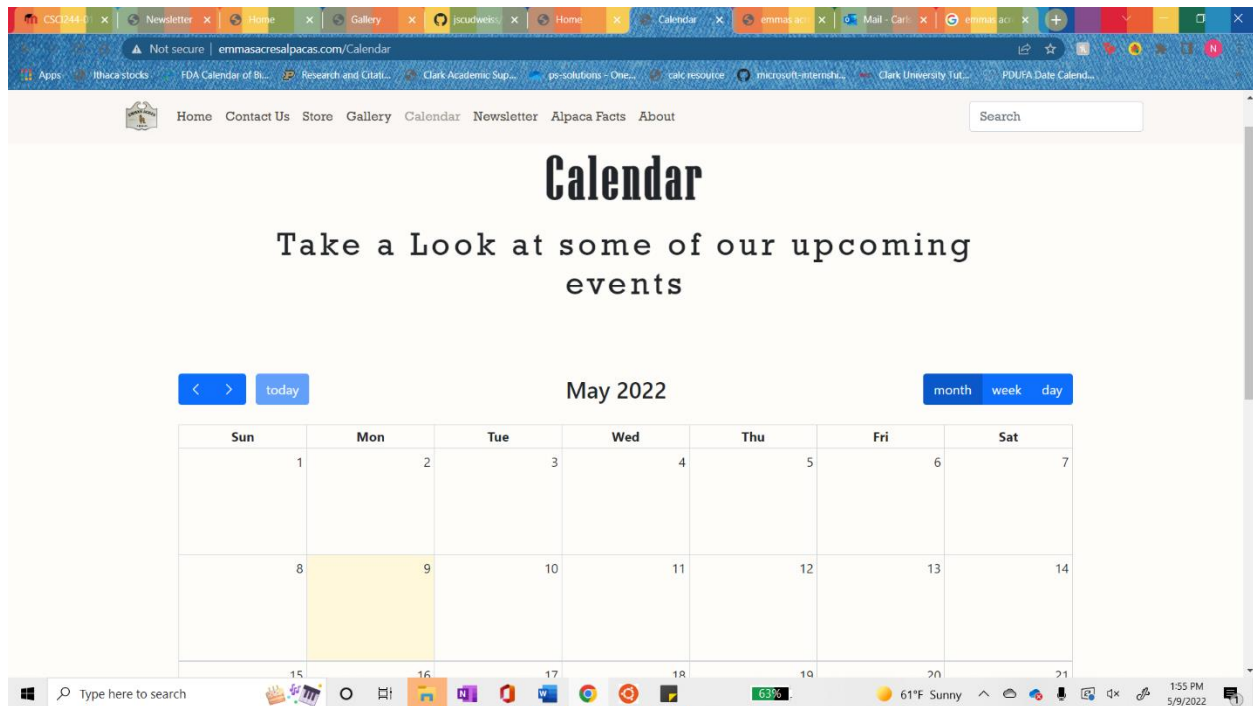
Roger & Collette Gage  
Amanda Gage  
105 Charlton St.  
Oxford, MA 01840  
E-mail: emmasacresalpacos@gmail.com  
Phone: (508)-987-5940  
Find us on Facebook!

© Copyright 2022 Emma's Acres Alpacas  
All Rights Reserved to [Nina Carlson](#), [Jonah Scudero-Weiss](#), [Isaac Schatz](#)

Type here to search

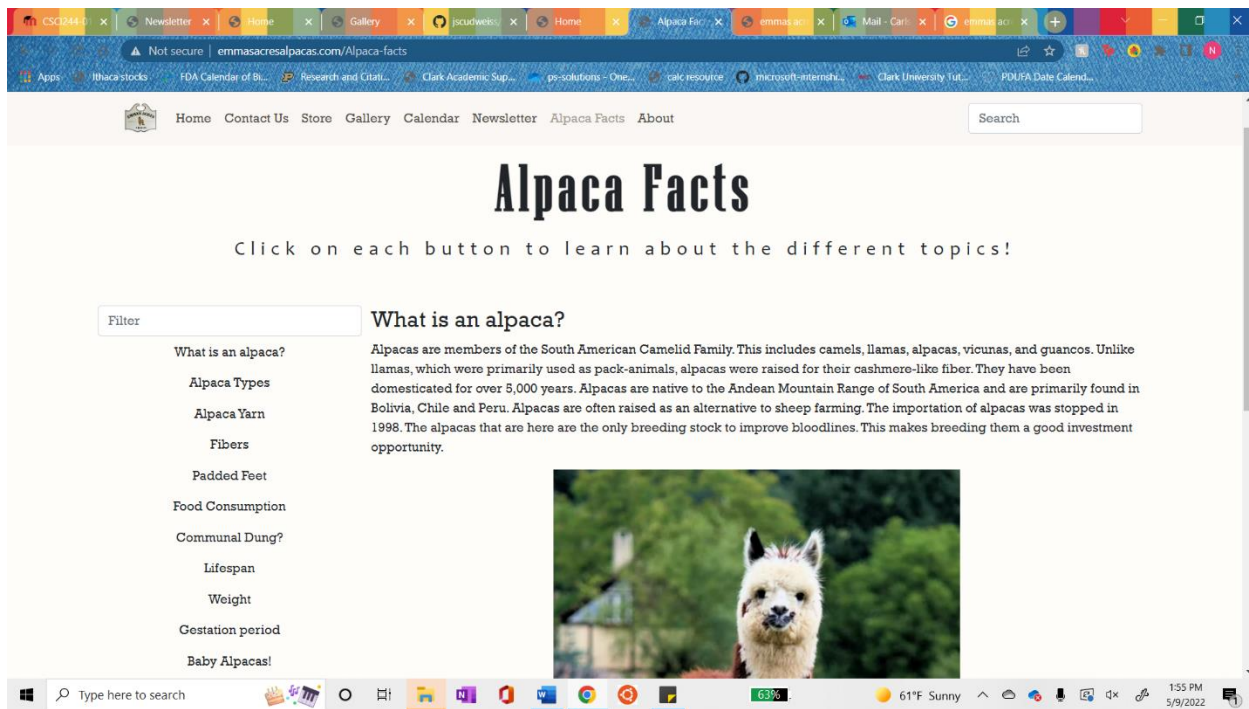
61°F Sunny 1:55 PM 5/9/2022

**Calendar Page:** the calendar page simply contains a calendar that is linked to the farm's already existing google calendar. When they add upcoming events to the calendar, it will automatically update on the website as well. Visitors can look at the calendar and plan their visits around upcoming events at the farm such as sheerings, fairs, and more.



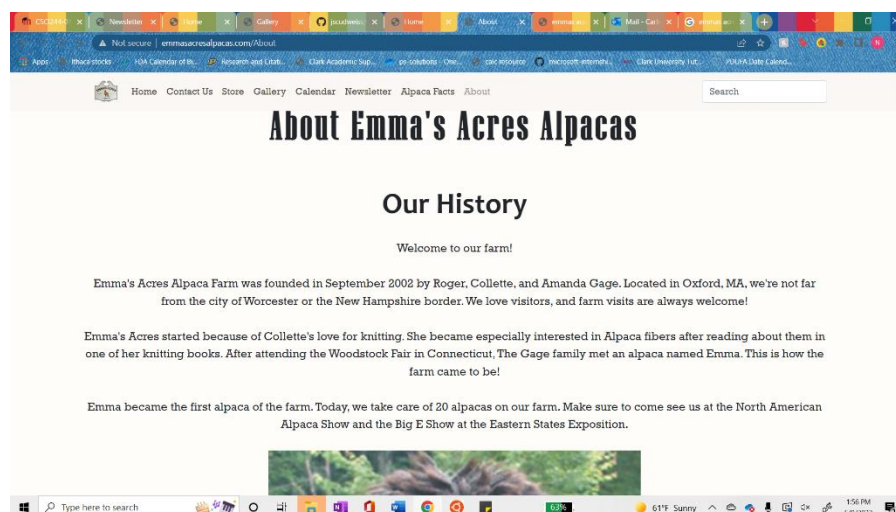


**Newsletter Page:** visitors of the website can subscribe to the farm's holiday newsletter and be informed of products on sale for the holidays by typing in their contact information via the form and submitting it. The data will be saved in a database for our clients to access so that they know who to send the newsletter to around the holiday season.



**Alpaca Facts Page:** Using a jQuery list and a JSON file, users can click on the topics and a fact and picture will pop up about alpacas. This is a way for visitors to learn more about alpacas and hopefully to draw their interest into the farm. Users can also type into the filter bar what they want to know and only the corresponding facts will pop up for them to read.

**About page:** the about page simply contains text that visitors can read to learn more about the farm, and it also contains a picture of Emma, the first alpaca that the farm ever had.





## Backend Design:

### jQuery:

1. Alpaca-facts: pulls from a JSON file and searches them with a text filter. It then loads the data from the JSON into buttons and pulls images as well. Finally, when the button is clicked it loads that data into the center of the page with jQuery. The filter automatically loads in URL data from the search bar searches.
2. Search-bar results: takes the list of search results from the search bar and loads them into a series of links which are under the search bar. Updates every time a letter is typed. If the letters are not contained within any of the page names but results are still returning, it displays the search query after the page names. If the list is empty an element saying no results for query pops up.

### MongoDB:

1. Search-Function-Database: used the tool cheerio to scan through the HTML files for all elements with the “searchable” class. When it finds this class it reads all of the text into the MongoDB with the page name as top. There is also a third field for extra details, but that is just for the alpaca-facts.
2. Search: when a search is done, the server sends a search request to the MongoDB for the word currently typed. It then returns all elements with the word or phrase in them, gets a score for the types of elements returned for list ordering (100x for page title, 5x for page\_data, 1x for extra\_info), and merges and adds the elements together based on page name to return a list of results including both page names and the data on the page.
3. Contact and Newsletter Page: takes in the data from the form and checks the format using a schema. If there are no problems a new contact/newsletter item is created and added to the corresponding database. For the contact page the data is a name (string), an email (string), a phone number (number), and message (string). For the newsletter there is an email (string) and an optional message (string).

### Cheerio:

1. Scans through all the HTML files

## Team Peer Evaluation:

**Nina (36.5%):** responsible for front end implementation, including design of each page, all bootstrap functionalities, HTML and CSS code, etc. Also responsible for writing up the final design report, communicating with clients, and conducting interviews.

**Jonah (36.5%):** responsible for backend implementation including the search bar, and the database storing data which runs it., the nav bar, the mobile aspects of the alpaca page, the alpaca page jQuery list, and deployment of the website. Also figured out and installed the calendar via Fullcalendar.

**Isaac (27%):** Responsible for newsletter and contact databases and making both those pages mobile aspects.

Nina, Jonah, and Isaac worked well as a team. Everybody did their share of the work, and everybody worked above and beyond to ensure that the website is fit for use by our clients. Everyone communicated well with each other, showed up for all team meetings, and completed

their parts of the project in a timely manner. Overall, we enjoyed working together as a team, and look forward to working with each other again in the future.

### **Design Report Questions (must be answered during interview by client)**

**1. *Why do you need to design (redesign) your website?***

The current website for Emma's Acres Alpaca Farm is significantly outdated in terms of both aesthetics and functionality. The website was created ~15 years ago and has not been updated since.

**2. *What goals is your current website not meeting?***

During the interview, our clients expressed concerns about the calendar function. The calendar currently does not work, since it does not show any upcoming events. The website also has a link to the farm's store on Square Space. Our clients said that the product inventory on the Square Space site does not update each time a purchase is completed. This becomes a large issue for the farm because products continue to be listed on the website even if they have already been sold and are no longer available. Furthermore, our clients requested a function in which they can contact their customers in case of an issue with the order. Lastly, our clients want us to work on a function in which their holiday newsletter will be sent to all their constituents around the same time every year.

**3. *What goals do you want to accomplish with the new website?***

Our team would like to create a brand-new version of the Emma's Acres Alpaca Farm website. We want to build a website that is aesthetically pleasing, user friendly, and can carry out all the functions specified by our clients. We hope to create an optimized way to keep track of store inventory (perhaps by using a database), create a holiday newsletter blast, add events to the calendar on the website, and create a backend function that can allow our clients to keep track of website traffic (i.e., how many visitors the site gets per day).

**4. *How is your website useful to visitors?***

The Emma's Acres Alpaca Farm website contains information about the farm's hours of operations, products for sale, and care instructions for alpacas. Visitors will be able to quickly find this information to plan visits, buy alpaca products, and even prepare to care for their own alpacas.

**5. *What kind of website does your brand need?***

During the interview, our clients emphasized that the website should be "user-friendly and appealing". Our clients want many pictures of the farm and the alpacas advertised on the site to draw visitors in.

**6. *Who do you want to target?***

Our clients strive to target visitors living in the local Oxford, MA area. Also, our clients hope to target organizers of bigger alpaca grooming competitions, crafts fairs, etc.

**7. *How do you evaluate success?***

Our clients would consider this to be a successful project if the website is able to draw in more visitors to the farm itself. They also would consider it successful if their online store became easier to use and manage. Lastly, our clients emphasized that they want us to have fun building their website!

**8. *What functions do you want your website to portray?***

Our clients specified the following functionalities during the interview:

- Homepage
- About Page
- Contact Us page

- Visit store page linked to optimized square space page to keep track of inventory
- Alpaca facts/caring for alpaca page
- Gallery of pictures
- Newsletter page

**9. *What is the scope of this project?***

The project will last about 2 months, in total. The interview was conducted on 3/19, and the design report with initial wireframes, affinity diagrams, and sketches will be completed by 4/1. A prototype of the website will be presented to our clients on 4/29, and the final product will be ready for professional use no later than 5/9/22.

**10. *What features are must-have vs nice to have?***

The website must have a home page, about the farm page, contact page, visit store page, gallery page, and alpaca care page. We will also work on a way to track visits to the website, but it is not imperative that this function is included.

**11. *How much traffic are you anticipating?***

Emma's Acres Alpacas is relatively small and not well known outside of the local Massachusetts area. We are hoping to increase the amount of traffic to the website in the coming months by creating a more appealing and functional website.

**12. *How can we avoid failure?***

We will work to avoid failure by meticulously planning out our time and strictly adhering to our plan of action. We will check in often with our clients to ensure that their needs are being met and that they are satisfied. We will reach out to Professor Niu with any technical questions or obstacles that we may encounter throughout the course of the project.

**1 Paragraph for Each Function**

*Home Page*

The home page will be the first page seen by visitors of the website. At the top of the home page, our clients specified that a large picture of Emma, the alpaca after whom the farm was named, should be included. Our clients described Emma as “the monarch of the farm” and want her picture to be prominently displayed. The home page will also include links to the other pages on the website. The home page should be visually appealing and easy to navigate, in order to encourage visitors to visit other parts of the website. The home page will include a navigation bar in the top right corner that will allow users to search for keywords on the website.

*About Page*

The about page will contain some text concerning the history of the farm, as well as some information about the owners. Our clients have provided us with background information that they would like included. This page will also contain a few pictures of some of the alpacas, as well as some pictures of the farm itself.

*Contact Us page*

The contact us page will implement a backend function that allows user to type in their contact information and send a message to the owners of the farm about scheduling a visit, asking questions, etc. These submissions will be stored in some sort of database.

*Visit Our Store / Square Market*

The visit our store page will contain pictures of some of the alpaca products that the farm sells. It will also link directly to the farm’s Square Market store.

*Alpaca Facts & Alpaca Care Page*



The alpaca facts & care page will contain information about how to care for alpacas, as well as some interesting facts about alpacas that may not be well known. Our clients specifically requested that there be a page that lists alpaca facts, so we are sure to include this in the building of the website.

#### *Gallery of Pictures*

There will be a page of the website called the gallery that will solely contain photos of the farm, the alpacas, the merchandise, etc. We have enlisted Ethan Doyle ([edoyle@clarku.edu](mailto:edoyle@clarku.edu)), a studio art student at Clark University, to photograph the farm. These pictures will be posted on the gallery page for visitors to look at.

#### *Newsletter*

The owners of the farm have a Holiday Season newsletter that they send out to visitors of the farm each year to advertise alpaca products as holiday gifts. We will create a form that allows visitors to subscribe to the newsletter, and we will keep track of these subscriptions by implementing a backend function using a database.

#### *Calendar of Events*

There will be a page that shows a calendar of events. Visitors can scroll through the calendar to see events that are coming up in the coming weeks, months, etc.

## Appendix A: Key Code Snippets:

### Alpaca Facts – JQUERY

```
function get_fact_List(fact, idx) {
    return `
    <div class="row">
      <button type="button" class="btn fact-btn" value="${fact.Info}" id="Button${idx}" name="${fact.url}">${fact.Title}</button>
    </div>
  `;
}

function loadFacts(data, filterVar) {
    console.log(data);
    $('#factNameList').empty();
    const lFilter = filterVar.toLowerCase();
    const fData = data.filter((fact) => {
        return (fact.Title.toLowerCase().includes(lFilter) || fact.Info.toLowerCase().includes(lFilter));
    });
    let len = fData.length;
    fData.forEach((alpFact) => {
        $('#factNameList').append(() => {
            len--;
            return get_fact_List(alpFact, len);
        });
    });
    $(".fact-btn").click(function () {
        let factText = $(this).attr('value');
        let factTitle = $(this).text();
        let factUrl = $(this).attr("name");
        console.log(factText);
        $('#factDetail').text(factText);
        $('#factTitle').text(factTitle);
        $('#factPic').attr('src', factUrl);
        console.log(window.innerWidth);
        if(window.matchMedia( query: 'only screen and (min-device-width: 769px)' ) || window.matchMedia( query: '(min-width: 769px)' )){
            location.href = '#split';
        }
    });
}
```

**Reads in a series of facts, saves them to a button. Using the name, value and text of the button to store all relevant information to the fact. Then retrieves that info with jQuery when clicked, additionally filters th list of buttons by a filter textbox.**

## Search bar, view results, jQuery -

```
function searchAndSet() {  
    let sk = $("#search").val().toLowerCase();  
    $.get("/get_search_results", {  
        search_key: sk,  
    }).done((data) => {  
        $('#searchResults').empty();  
        searchData = data.data;  
        switch (sk) {  
            case "":  
                searchData.sort( compareFn: (a :T, b :T) => {  
                    return a._id.localeCompare(b._id) || (b.count - a.count);  
                })  
                break;  
            default:  
                searchData.sort( compareFn: (a :T, b :T) => {  
                    return (b.count - a.count) || a._id.localeCompare(b._id);  
                })  
        }  
        console.log(searchData)  
        switch (searchData.length) {  
            case 0:  
                searchData = [{_id: "NA", count: 0}]  
                break;  
        }  
    })  
}
```

```
searchData.forEach((input) => {  
    let shownText = "";  
    let link = "";  
    switch (input._id) {  
        case "NA":  
            shownText = "NO RESULTS"  
            link = "#"  
            break;  
        case "contactus":  
            shownText = "CONTACT US"  
            link = "contact"  
            break;  
        default:  
            shownText = input._id.toUpperCase()  
            link = input._id  
    }  
    let includes = !(shownText.includes(sk.toUpperCase()))  
    switch (includes) {  
        case true:  
            shownText = shownText + " : " + sk;  
            link = link + "?search=" + sk;  
            break;  
    }  
    $('#searchResults').append(`  
    <a href="/${link}">${shownText}</a>  
    `);  
})  
}
```



Gets the results of the search function, sorts them by score, and displays them in order using jQuery. Has additional cases for if the search key that was searched is not contained in the list item name, and for an empty list.

### Calendar – FullCalendar

```
document.addEventListener( type: 'DOMContentLoaded', listener: function() {  
  var calendarEl = document.getElementById( elementId: 'calendar');  
  
  var calendar = new FullCalendar.Calendar(calendarEl, {  
    googleCalendarApiKey: 'AIzaSyAX8G6BQ7251lgDb1OKQjdSUKq3JbvT9gA',  
    themeSystem: 'bootstrap5',  
    initialView: 'dayGridMonth',  
    headerToolbar: {  
      left: 'prev,next today',  
      center: 'title',  
      right: 'dayGridMonth,timeGridWeek,timeGridDay'  
    },  
    events: {  
      googleCalendarId: '08g6o144ot5l6e2niu01td1dmo@group.calendar.google.com'  
    }  
  });  
  
  calendar.render();  
});
```

Generates a calendar using the full calendar API and links it to the clients google calendar.

Mongoose/mongo-DB commands –

## Search:

```
app.get("/get_search_results", (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  const sk = req.query.search_key;
  Page.aggregate( pipeline: [
    {
      $facet: {
        name_match: [
          {$match: {page_name: {$regex: sk}}},
          {$group: {_id: "$page_name", count: {$sum: 100}}}
        ],
        data_match: [
          {$match: {page_data: {$regex: sk}}},
          {$group: {_id: "$page_name", count: {$sum: 5}}}
        ],
        extra_match: [
          {$match: {page_info_extra: {$regex: sk}}},
          {$group: {_id: "$page_name", count: {$sum: 1}}}
        ],
      },
    },
    {
      $project: {
        all: {
          $concatArrays: ["$name_match", "$data_match", "$extra_match"]
        }
      }
    },
    {
      $unwind: "$all"
    },
    {
      $group: {
        "_id": "$all._id",
        "count": {$sum: "$all.count"}
      }
    }
  ])
})
```

```
],
options: function (err, data) {
  if (err) {
    console.log("err" + err);
    res.send( body: {
      "message": "error: " + err,
      "data": data
    })
  } else {
    //console.log(data);
    res.send( body: {
      "message": "success",
      "data": data
    })
  }
}
).collation( options: {locale: 'en', strength: 1})
})
;
```

Searches the pages database which stores the data from different webpages, then returns the results with scored based on which result type contained the input, finally merges and adds those scores based on which page it came from originally. Returns this result as an array.

## Newsletter

```
app.post( path: "/save_newsletter", handlers: (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  Newsletter.create( doc: {
    email: req.body.email,
    message: req.body.message
  }, callback: function (err : CallbackError , newsletter : Document<unknown, any, unknown> & Require_id<unknown> ) {
    if (err) {
      return console.error(err);
    } else {
      console.log("newsletter saved to database");
      res.redirect( url: "/ThankYou.html");
    }
  });
});
```

Stores newsletter emails for the clients newsletter from the newsletter form.

## Contact

```
app.post( path: "/save_contact", handlers: (req : Request<P, ResBody, ReqBody, ReqQuery, Locals> , res : Response<ResBody, Locals> ) => {
  Contact.create(
    doc: {
      name: req.body.name,
      email: req.body.email,
      phone_number: req.body.phone_number,
      message: req.body.message
    },
    callback: function (err : CallbackError , contact : Document<unknown, any, unknown> & Require_id<unknown> ) {
      if (err) {
        console.log(err["message"]);
        res.redirect( url: "/ContactUs.html" +
          "?error_message="+err["message"] +
          "&name="+req.body.name +
          "&email="+req.body.email +
          "&phone_number="+req.body.phone_number +
          "&message="+req.body.message);
      } else {
        console.log("contact saved to database");
        res.redirect( url: "/ThankYou.html");
      }
    }
  );
});
```

Stores messages and contact information for the clients customers from the contact form.



## Search Database\_INITIALIZER: MongoDB, Cheerio

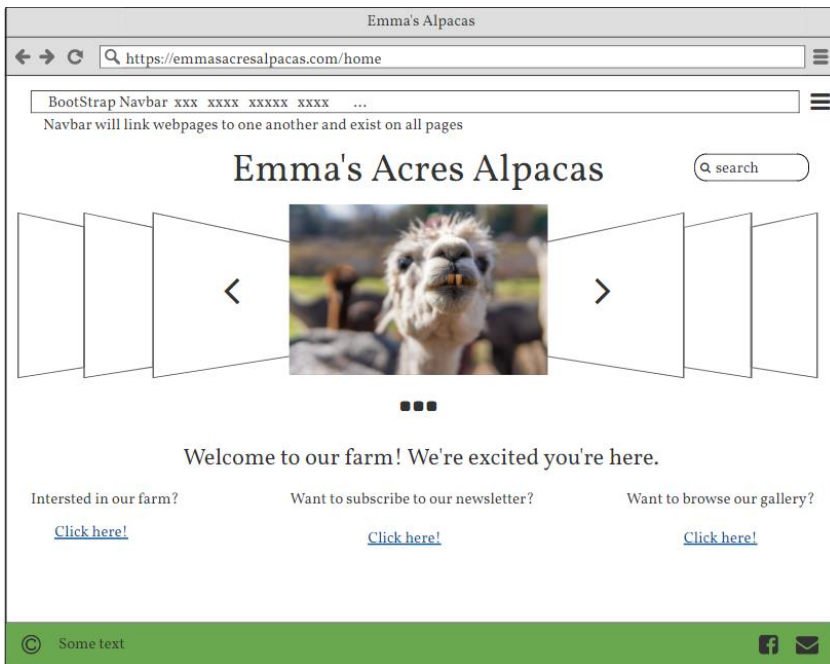
```
files.forEach(function (file :string ) {
  fs.readFile( path: 'public/' + file, options: 'utf-8', callback: function (err :ErrorException| null , contents :string ) {
    const pageList = [];
    let $ = cheerio.load(contents);
    const nameLoaded = {
      'page_name': file.slice(0, -5).toLowerCase(),
      'page_data': file.slice(0, -5).toLowerCase()
    }
    pageList.push(nameLoaded);
    $('searchable').each( function (i :number, elm) {
      const pagedat = {
        'page_name': file.slice(0, -5).toLowerCase(),
        'page_data': $(this).text().toLowerCase()
      }
      pageList.push(pagedat);
    })
    Page.insertMany(pageList, options: {}, callback: (err :CallbackError ) => {
      if (err) {
        console.log(err);
      } else {
        console.log(file + " data saved");
      }
    });
  });
});
```

```
fs.readFile( path: __dirname + "/public/data/Alpaca-Facts.json", options: 'utf-8', callback: function (err :ErrorException| null , data :string ) {
  if (err) {
    console.log(err)
  } else {
    let factdata = JSON.parse(data)
    const len = factdata.length
    let factList = [];
    for (let cur = 0; cur < len; cur++) {
      const fdata = {
        'page_name': "alpaca-facts",
        'page_data': factdata[cur].Title.toLowerCase(),
        'page_info_extra': factdata[cur].Info.toLowerCase()
      }
      factList.push(fdata)
      if (cur === (len - 1)) {
        Page.insertMany(factList, options: {}, callback: (err :CallbackError ) => {
          if (err) {
            console.log(err);
          } else {
            console.log("fact data saved");
          }
        });
      }
    }
  }
})
```

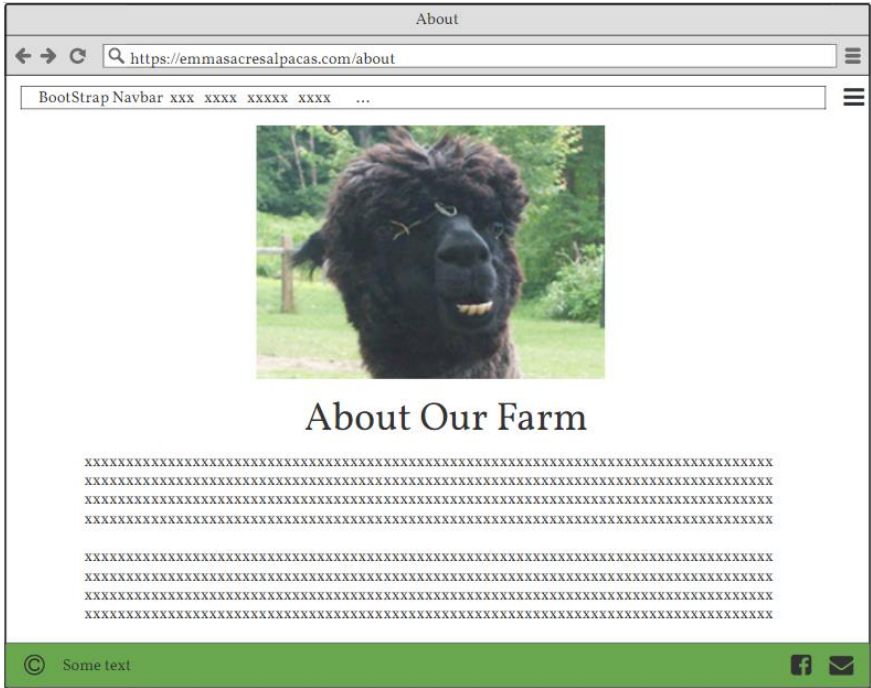
Scans through all html pages of the website using cheerio, when it finds an html with the an element of class “searchable” It stores that text into the pages DB for searching. Then adds all the alpacafacts in the same database, with the long paragraphs of facts being stored in the extra section.



## Wireframes

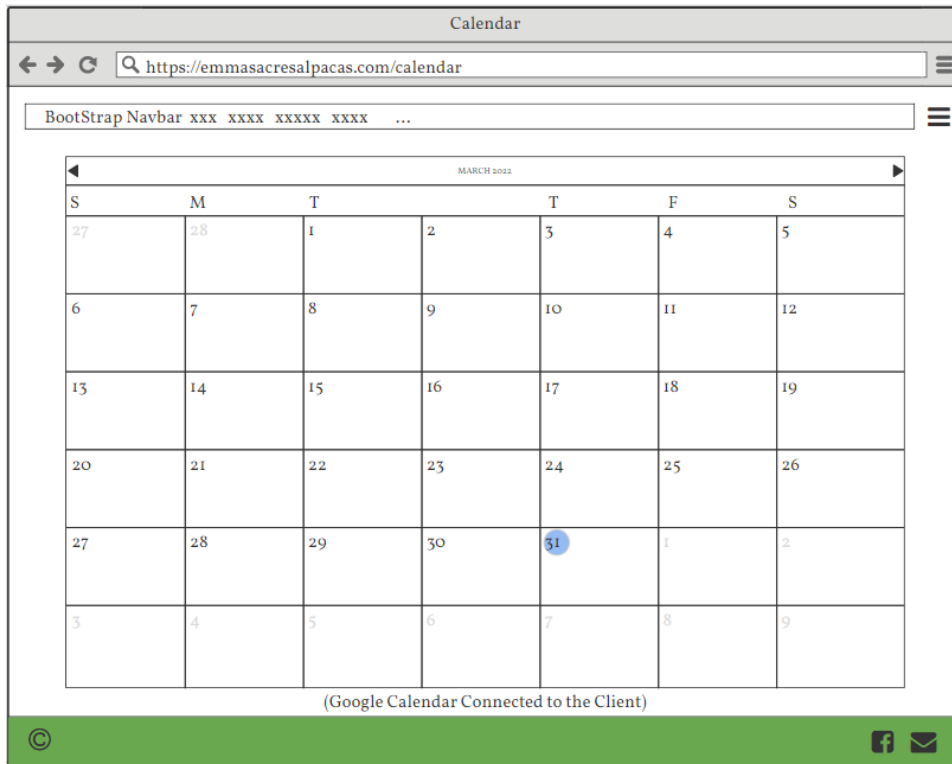


This is the wireframe for the homepage. As you can see, our design changed drastically from the initial wireframe to the final product. We decided to use a single picture with an opacity characteristic and overlay the title of the farm on top of it, so that visitors would see the entire picture without scrolling. That was a flaw in the initial design here. We also turned the links into buttons, and added pictures to each. Lastly, for the footer, we credited ourselves as well as the website developers.

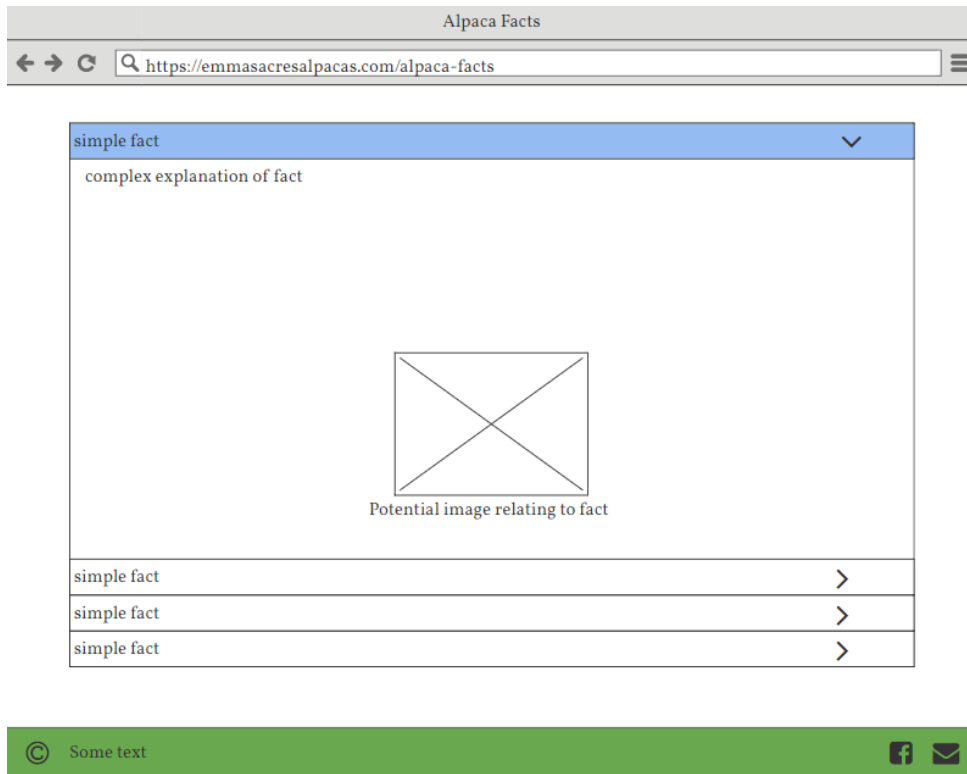


The about page wireframe mostly matches our design for the actual about page on the website. The only change is that the picture of emma is below the description of the farm instead of above, to once again avoid the visitor only seeing half a picture without scrolling.

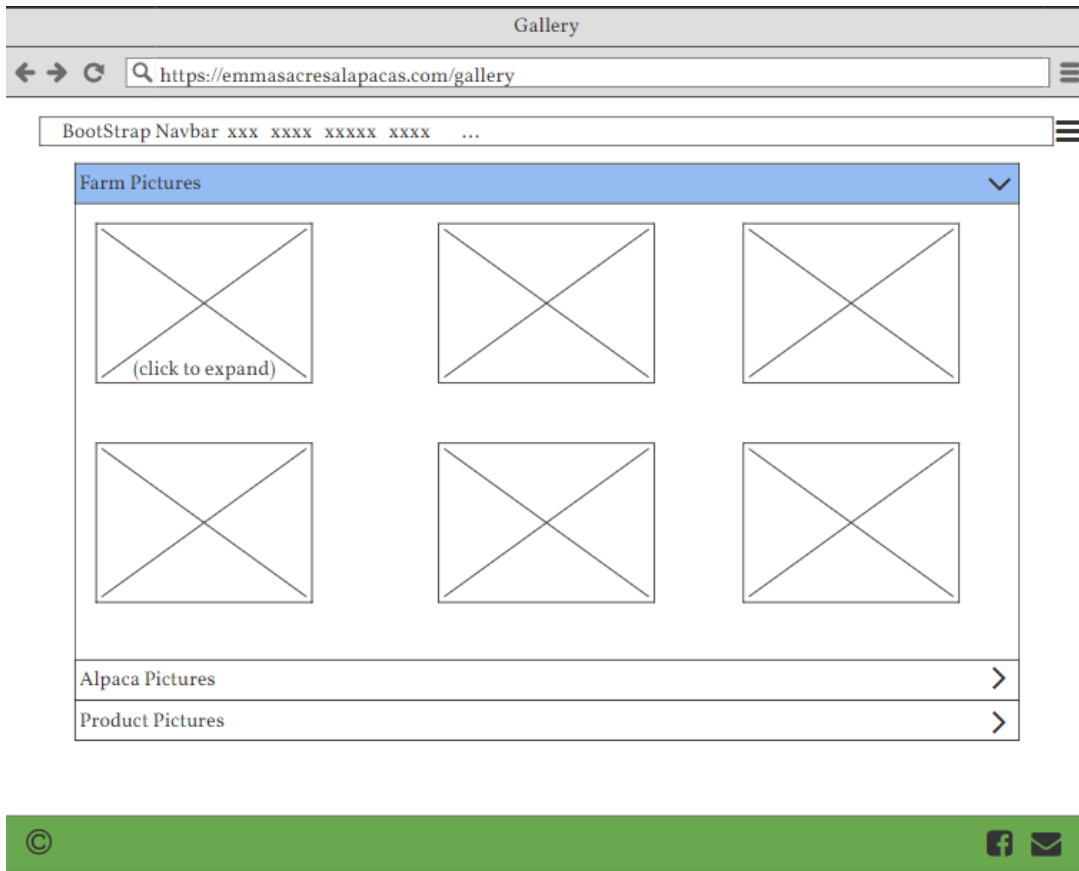




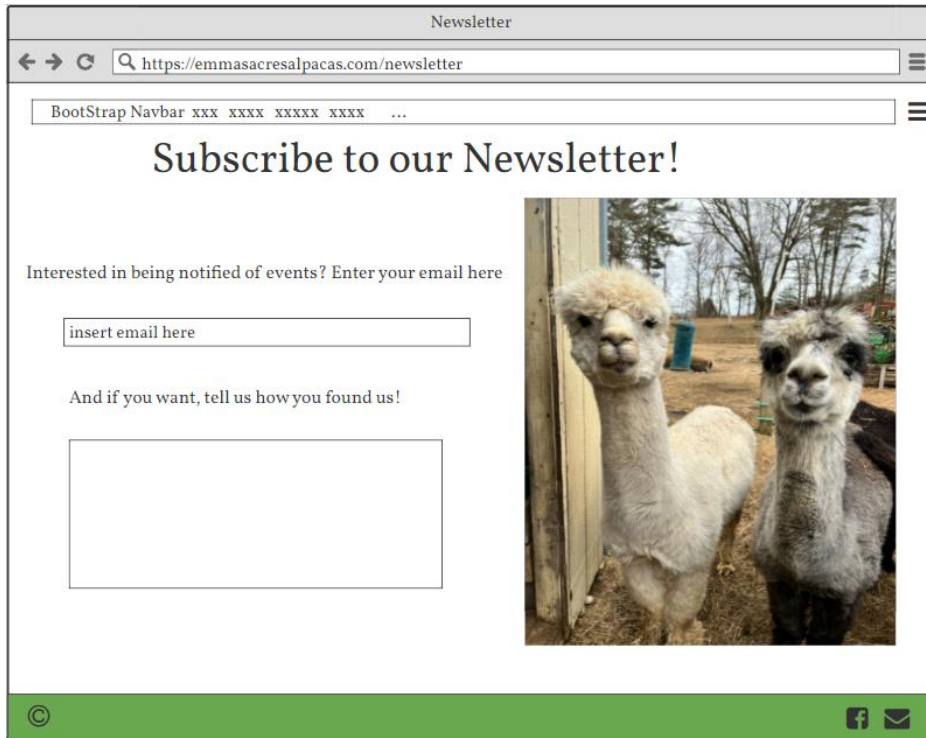
The calendar page wireframe mostly matches the actual implemented design. The only difference is that the actual website has a title on this page. The calendar on the website can also be organized by month, day, or year.



The alpaca facts page is implemented a little differently than what is shown in this wireframe. Instead of a drop down list, we have a list with a scroll bar on the side, and when each element of the list is clicked, it brings up the corresponding fact and picture from a JSON file using a jQuery list. There is also a filter function that allows users to type in the fact they want to see, rather than having to scroll to find what they are looking for.

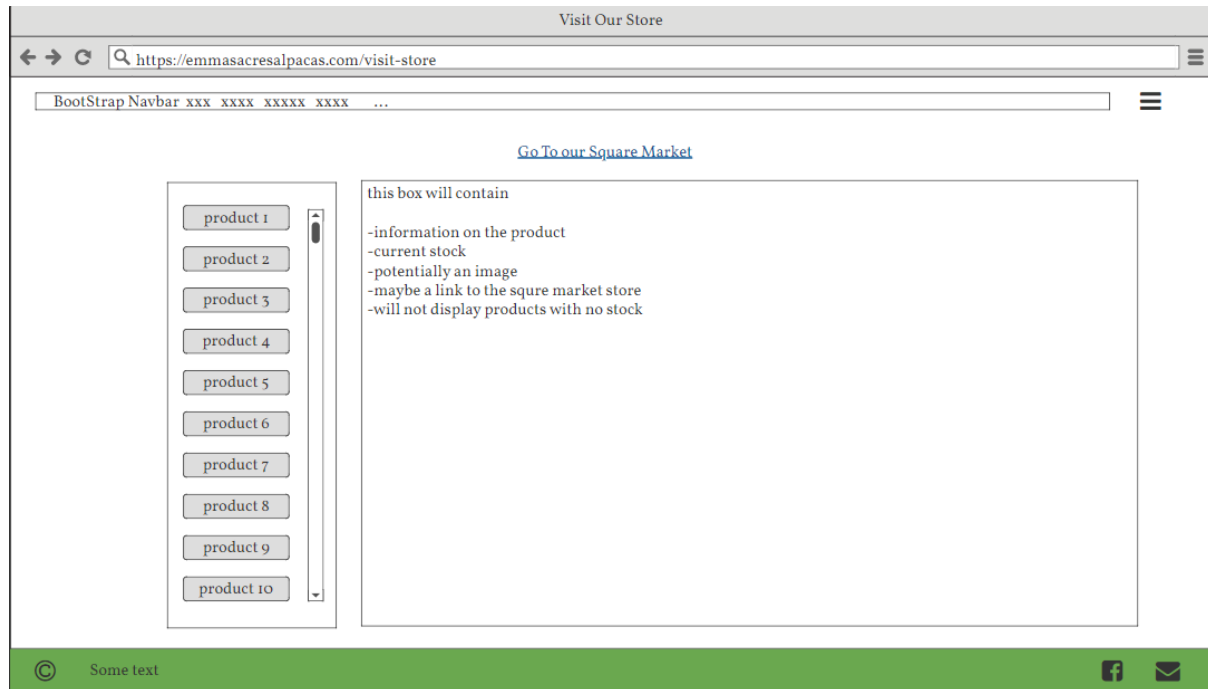


The gallery page has a completely different design on the implemented website than shown in this wireframe. After receiving feedback from Professor Niu, we changed the design entirely. Now, the gallery page is simply rows and columns of pictures that pop out if the user places their cursor on top of each picture. There is no more drop down list.

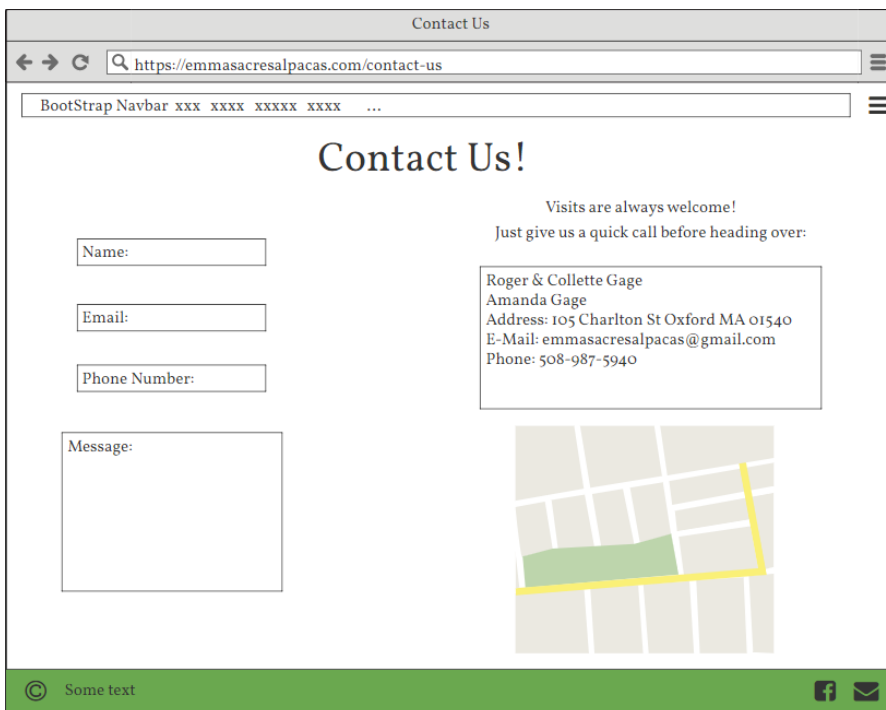


The newsletter page wireframe is exactly the same as the actual implementation on the website. Users can insert their email and an optional message that gets sent to a database to be accessed by our clients using MongoDB.





The design on this wireframe for the store page ended up being the layout for the alpaca facts page instead. The square market required handling very sensitive credit card information, which our team did not feel comfortable doing. So, instead of handling the store on back end, we simply included pictures of products and buttons that link to the square market website.



The contact us page wireframe is exactly the same as the actual implemented page on the website. Visitors can input their phone number, name and email, and a message that gets sent to a database to be accessed by our clients. There is also a google map frame on the page and when it is clicked, it takes the user directly to google maps.