

Lab 3

UI/UX Design - Polling Website

Team 7 - Broken Engineers:

Matthieu Stogsdill, Makenna Maguire, Tim Evans, Austin Wittenburg

1. Introduction

In this lab the primary goal was to create a website that contains all the necessary pieces for a well-structured user-interface (UI) and user-experience (UX) design. The task of this lab was to create a Doodle poll which would allow users to vote for a designated time slot for meetings and other tasks where meetings are needed. The Doodle Poll interface we have been using all year, has not met the user standards needed for the class, so it was our task to create an interface that would be similar to the Doodle poll interface. This would allow for the user to have an intuitive and efficient time working with the webpage. This means the flow of the design would be easy and natural for the user to understand and it also was simplistic. Another key feature for the user experience is the attractiveness of the webpage. While these features exemplify the user experience, the site also needs to be informative and robust. The interface provides the user with the essential information and it also makes sure the user can navigate any errors that arise in the process of making their site.

[Home](#) [Sign In](#) [Poll](#)

Home Page

Broken Engineers Doodle Poll

Looking for your next poll look no further!



Figure 1: Home Page for the web page: [index.js](#)

2. Design Documentation

The software design of this process consisted of multiple JavaScript scripts that were started using the Gatsby framework. The Gatsby starter provided a strong base for us to go off of. This provided the layout.js, 404.js, and all the Gatsby configure files. This gave us a head start and a good introduction to JavaScript and HTML that we needed. Following this we started to create four different pages that would carry the information needed for the poll to work properly. These pages included: a homepage, admin page: create, home and modify in that folder, poll page, and a login page. These pages hold all the pieces of our webpage which make it the success that it is. The four pages all have their own unique part and as seen in Figures 1-4 the pages are shown.

Index.js UI/UX Considerations: First comes the index.js page which we turned into our Homepage. On this page, shown in Figure 1, it is the home base for the whole site. The file is very simple to allow for easy user interaction and it includes simple HTML code that lays out the website. The Layout includes links to two other pages and also has the title and brief message for the site. The page also has a small poll logo to show the users what the site is about. The admin is able to clearly see the sign in page at the top along with access to the poll page which is where the user will be able to go to sign up for the poll time.

Page not found

Sorry 😞 we couldn't find what you were looking for.

Try creating a page in `src/pages/`.

[Go home.](#)

Figure 2: 404.js

404.js UI/UX Considerations: The 404.js was a file included with Gatsby and it is one of the most necessary files on the site. This file makes sure that when the user tries to open a page that does not exist or can not be loaded, the 404.js page will be redirected to the user and then the user is prompted to go home. This page is a vital part to any website's functionality.

[Home](#)[Sign In](#)[Poll](#)Username: Password: **Figure 3: login.js**

Login.js UI/UX Considerations: The login page is the next key page and this is where the admin can click sign up for the website in order to create polls for the user. The page is also written in simple HTML code and includes the Layout features and then a few other libraries were downloaded to make the page look more aesthetically pleasing. From the styled-components library a nicer button was added to this page. This page is equipped with the username and password holders for the admin. If the username or password of the admin is not correct, then an alert will pop up on the screen saying 'Incorrect username/password try again'.

[Home](#)

Step 1: Poll Information

Title: Location:

Notes and Comments:

Select your Time Zone:

Step 2: Date and Times

« < December 2021 > »						
MON	TUE	WED	THU	FRI	SAT	SUN
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Start time: End time:

Number of Blocks:

Length of Session:

Restrict votes per slot

Restrict votes per participant

Figure 4: create.js

Step 3: Invite Participants

Invite:

Step 4: Set a Deadline

Deadline:

Figure 5: create.js

Create.js UI/UX Considerations: The ‘admin/create’ route navigates to the page where an admin can create a poll. User experience considerations for this page include ease of use and informative layouts. The calendar display allows the user to just click on a day instead of having to enter the date. The user can also drag their mouse across multiple dates for multi-day selection. This is superior to the alternative of manually entering each date because it saves the user time and lessens the chance for input error. A similar approach is used with deadline selection. We used the react-datetime package to allow the user to select the deadline with a calendar GUI along with an hour/minute selection GUI. This saves the user time since they don’t have to manually enter the date. Another user experience consideration is with the calculation of the number of sessions and number of blocks the admin wants. If the admin wants to calculate the number of blocks, they simply select the start time, end time, and then type in how many blocks they want. The length of sessions will automatically be calculated and displayed to the user in real time. If they want to calculate the length of the sessions, they only have to select start time, end time, and then type in the length of sessions they want and the number of blocks will be displayed in real time. The results are displayed by clearing the text field of the variable to be calculated and displaying the result as soon as the user types in a number. This is superior to the alternative of having to press enter or a submit button because it reduces the number of steps for the user which increases simplicity and ease of use. The user can easily save their work by clicking the save button, or publish the poll with the publish button. This provides for a good user experience because they can save their work and continue later.

[Home](#)

Admin Home Page

To get started, create a poll --> [CREATE A POLL](#)

These are your polls!

- Nmak [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- time slots test [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- new title [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- timeslottest8 [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- Timeslottest5 [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- newtitle2 [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- timeslottest4 [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- hei [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)
- Dropdown test [\(modify\)](#) [\(remind\)](#) [\(publish\)](#)

[LOGOUT](#)

[LOAD POLLS](#)

Figure 6: Admin homepage to show polls and how to modify polls

Modify.js UI/UX Considerations: The ‘admin/modify’ route contains the page that the admin is directed to when they want to change a poll. This is a very easy thing to do for the admin. They can just click the ‘render’ button to load in the existing poll information, change any field they want, and then click ‘save’ or ‘publish’ to save the changes back to the database. The modify page looks almost identical to the create page, besides the render button, so it has all of the user experience considerations of the create page.

[Home](#) [Sign In](#) [Poll](#)

Poll Details

December 2021						
MON	TUE	WED	THU	FRI	SAT	SUN
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Identifier:

- Title: ""
- Location:
- Notes & Comments:
- Session Time: minutes
- Time Zone:
- Deadline:
- Poll ID:
- TimeSlots: -

Figure 7 : poll.js

Poll.js UI/UX Considerations: The poll page was created to show the created poll the user has access to. On this page the user will be able to get the title, location, notes, session times, the time zone, and deadline. Once the poll is loaded the time slots will show up and the user will be able to select which time slot they would like to sign up for. They accomplish this by putting their name in the identifier selection. Upon submission the user will be notified that they have signed up for the poll and it will show which slot they selected and the name of the poll. The calendar shows the days the poll is active for. Overall the poll page was created to be user friendly. All the data is in one place and the user will see all the time slots and days the slots are available.

Home.js UI/UX Considerations: The ‘admin/home’ route contains the page that the admin is directed to when they sign into their account. On this page as seen in figure 6, the admin has the ability to load the created polls with the load polls button at the bottom. Then once the polls are loaded, the admin has the ability to modify, remind, or publish the created poll. When the modify button is pressed the admin is directed to a page with the created page and the information is able to be updated by the admin on this page. The remind button when pressed will remind the users and an alert will pop up that says ‘Invited Users have been reminded of the poll’. The publish button when pressed will publish that specific poll. The polls are organized by title and then followed by the three buttons to make it easy for the admin to accomplish all necessary tasks as the admin of the poll. The admin also can create a poll by pushing the ‘create a poll’ button. When they push the button it will bring them to the create page which is seen in figures 4 and 5. Lastly the user has the ability to log out of their account. The admin homepage is very user friendly as it only has the necessary buttons on the page that the admin would need to create a poll. Overall this creates an easy and accessible page for admins trying to make a poll.

3. Design Process and Experimentation

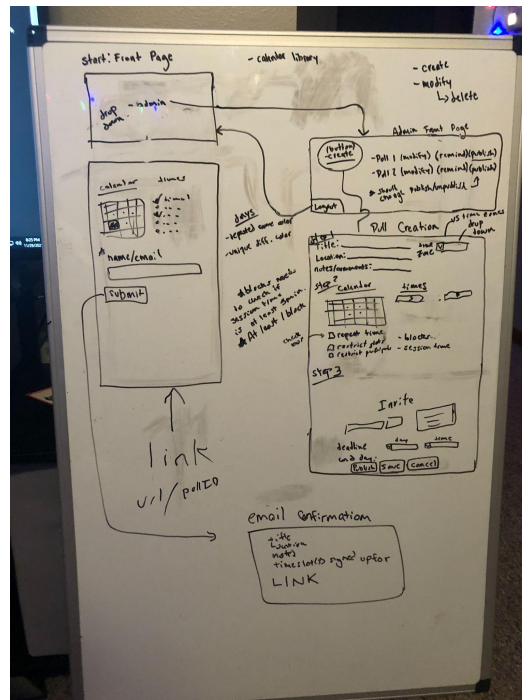


Figure 8: Polling Website Storyboard

In any UI design the friendliness and intuition of the layout is crucial in the success of this lab. In order to create an easy to use and intuitive UI we planned out each component of the website beforehand on a storyboard. The storyboard represents the skeleton and general flow of

our website including important aspects such as pages, buttons, features, text, etc. We first started with how many different pages we would need to complete the website starting with a home page and stemming off into admin login, poll viewing, and poll creation. This storyboard showcases these pages and also shows how they will be linked to each other. In the picture the first page is the login which then once filled in will take the admin to the admin homepage. On the admin homepage it allows the admin to create, modify, publish polls as well as remind users as well. The admin homepage is linked to the create poll page which allows the admin to publish a poll to the poll page. The create page is connected to the poll page which shows the users the poll and allows them to sign up for a time slot for that poll. The storyboard was a key piece to the success of the lab and getting the lab designed right away kept us organized throughout the whole lab.

We initially had our framework choices narrowed down to either Gatsby or Flask. Flask has good back-end support for web apps that would have made storing and retrieving data easy and Gatsby had a lot of great front-end support to make the creation of the user interface easy and quick to design. Considering that this lab was centered more around the front-end user experience we decided to go with Gatsby because we felt that it would be the best choice to deliver a more user friendly experience.

We originally wanted to send email notifications to our users using SendGrid but we decided that since we already have Twilio set up and working from other projects that it would be easier and save us more time by just modifying our design to send users SMS text messages containing the link to the polls instead. The twilio implementation turned out to be tricky because of the integration with Gatsby Cloud anyway though.

4. Test Report

Test Procedure	Expected Result	Pass /Fail	Value
Only Admin can create a poll.	Admin is only able to create a poll after logging in with username and password.	Pass	
Only Admin can modify a poll.	Admin is only able to modify a poll after logging in with username and password.	Pass	
Only Admin can delete a poll.	Admin is only able to delete a poll after logging in with username and password.	Fail	
Only regular users can put their names and make selections to an existing poll.	Regular users and invited users can enter their name and vote on a poll.	Pass	

Admin can login with hardcoded username and password.	Upon using admin username and password you are taken to the admin home page.	Pass	
Admin should get an incorrect login alert upon entering an incorrect username and/or password.	Upon entering an incorrect login an alert notifies the user that the login is incorrect.	Pass	“Incorrect Login”
Poll has a required field title.	Poll will give an alert if no title is entered.	Fail	
Poll should have <u>optional</u> fields for a location and notes/comments.	Poll shows the optional location and notes inputs on the poll.	Pass	
Poll allows for a selection of many dates and times.	Admin is able to select multiple dates from the calendar and add them to the poll.	Pass	
Slots are a minimum of 5 minutes.	Website will calculate the time per slot based on the slot input.	Pass	
There is a minimum of 1 time slot per poll.	Website will calculate the number of slots based on the session time.	Pass	
The poll defaults to no time zone.	Drop down menu requires the user to select one of the US time zones.	Pass	
Admin can restrict the number of votes per time slot.	Check box restricts the number of votes to 1 or infinite.	Pass	
Admin can restrict the number of votes per participant.	The participants are not allowed to vote more than once	Pass	
Admin can invite participants to poll.	Invited participants receive an invite link upon the admin adding their contact to the list.	Fail	
Admin can remind invited participants to vote in the poll.	Invited participants receive a reminder notification to vote in the poll.	Pass	
Admin can set a deadline for how long the poll is open and available.	Deadline is show on the poll and the poll is unpublished upon reaching the deadline	Pass	
The Publish button allows participants to see the poll.	Published polls are visible to regular users.	Pass	
The Modify button allows Admin to	Modify button takes you to the create	Pass	

edit any of the fields in the poll.	poll page to modify information		
The user can enter in an identifier (i.e.name or email) to sign up for the poll.	Users get a notification showing their identifier used to sign up for the poll after voting.	Pass	
Invited users can “vote” or “reserve” in a poll.		Fail	
Submit button on a poll allows users to confirm their “vote” or “reservation”.		Pass	
After the user submits their “vote” or “reservation” an alert will notify them with their information and details of the poll.	Users receive an alert showing all information about the poll they signed up for.	Pass	

Table 1: Test Report Table for Polling Website

Above is **Table 1** an organized listing of all the main requirements and functionality of the website. While this project requires that these are satisfied the most important aspect is the user experience which cannot be simply summarized in a table of pass/fail constraints. As stated before the success of any UI/UX design is the structure and intuitive experience you get from using the interface. In addition, since this was an entirely software based project that required only coding, a lot of testing and verification of working parts came in the form of debugging, an essential part of any coding project. This being said, the above table does not reflect the general process of debugging and testing needed to arrive at the constraints listed in the table passing their expected results.

5. Project Retrospective

a. Project Outcome

Overall the lab turned out well despite changing a few things from the original plan. The research stage of the lab took longer than expected. This was mainly because of the change in the requirements of the lab. However, after all the research we stuck with Gatsby. Our research portion took most of the time and left us not a lot of time for the software piece. Even with not a lot of time the lab was completed well, spending numerous hours in the last week coding and designing the layout of the lab. By splitting up the work this lab was completed to the best of our abilities and was a key success in the lab. People were working to

accomplish different pieces of the requirements and when questions arose we worked them out together.

b. Changes for Future Labs

For future labs, we believe starting the coding part of the project earlier would have been more efficient rather than spending all the time researching the frameworks. The other piece was looking into the databases for the backend part of the lab. By the time we got to this part of the lab, we did not have a definite plan on which database to use and it took us a lot longer to figure out this part of the lab. If we spent more time looking into the databases that work better with Gatsby because we struggled with integration of the database with the Gatsby framework.

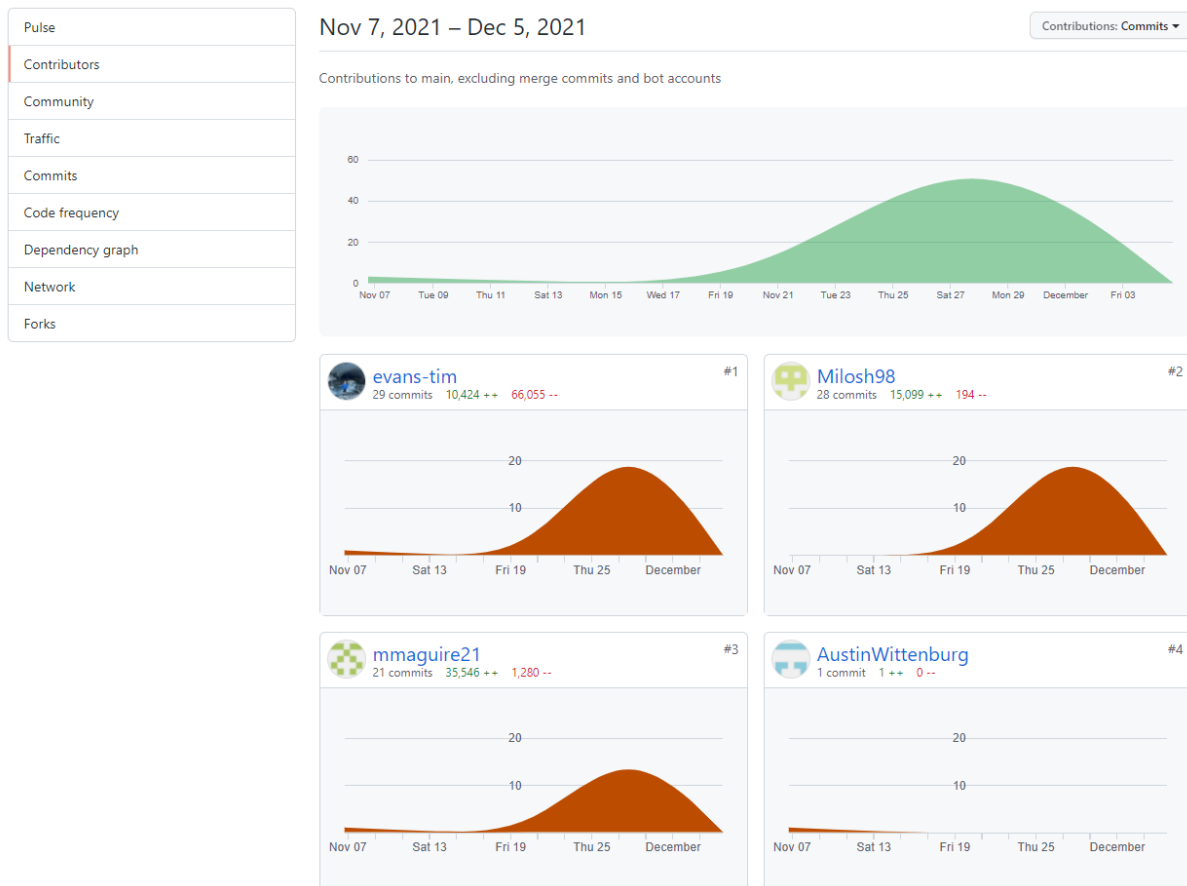
c. Roles and Responsibilities

On this project each member of our group was given a specific webpage to work on based on their understanding and skill with coding and web development. In order to efficiently complete this project it's reasonable to assign a group member with more experience to a part of the project like the poll creation page which includes numerous aspects of web development. Matt worked on the admin log-in page which required the creation of textboxes for the user to enter in their username and password and a submit button. In addition, Matt helped with some of the aesthetic design of the website making colorful buttons and making sure everything moved with the page and scaled correctly. Makenna was in charge of setting up the Gatsby framework and connected Gatsby to Github. After this Makenna completed the index page which was the homepage of the doodle poll. Also, Makenna started with the poll page on the design portion, however, with the help from Tim and Austin we were able to complete this page effectively. Makenna was in charge of making sure the pages aesthetically looked good and all the pages looked the same. Tim set up the database on Google Firestore and connected it to the data layer of gatsby. He also focused on the functionality and UI/UX of the poll creation and modification page with layout help from Matt. Tim also worked on some aspects of poll.js like splitting up time slots and displaying the slots to the user. Austin worked on the admin home page and handled the transfer of information between different pages such as the specific information for each poll.

While this project was entirely software based it is possible that this project can be done remotely split between each member of our group. During the research

and planning portion we mostly conducted our meetings over a voice call and discussed possible methods for completing the website. Once the time came for implementation and building of the website we decided meeting in person would be beneficial to the success of our website as we could storyboard physically in person around a white board to plan out visually what our website would look like. Also everyone being in the same room provided a sense of responsibility and accountability that everyone was working on their part. When one of us had a question or idea it was easy to simply show your computer screen and the group could provide feedback immediately or even show another member how to implement a certain aspect.

The GitHub contributions are attached below. There are two pictures as the first one is the amount of commits added, however, GitHub had an error in loading Austin's commits in the statistics page so the second picture then shows that Austin committed 20 times and contributed to a significant amount of the lab.





d. Project Management

Throughout this lab, a waterfall driven methodology was used to complete the lab. The lab was completed again in phases in order to make sure every requirement was passed and the user experience and user interface was good. The lab requirements were specific and there were a lot of them as well so to make sure none were forgotten, working in phases helped accomplish this. The phases involved adding pieces to the webpage first and getting the little design choices and aesthetic first. Then as the page started to develop we made sure we connected the pages and linked the pages correctly. After all the pages were linked together, the database was set up and then information was able to be collected. After this phase the lab was completed and a final phase was completed to make sure the design was what we wanted and met all the requirements for the lab.

e. Gantt Chart

Throughout the lab, it was attempted to follow the Gantt chart as best as possible, however, with the professors switching the ability to use a different framework, more time was spent researching and planning the project. At first Gatsby was used and the framework was set up, however, different programs were also considered like Flask, Ruby on Rails, and Node.js. Overall after the research stage, the rest of the Gantt charge was followed as best as could be, but the software piece was completed mainly in the last week which was due to the fact that researching the framework took longer than expected.

