Modeling The Effects of Importance and Availability on User Wait Times

While Utilizing Various Loading Screen Types

Pavel S Frolikov
Frolikov@ucdavis.edu
University of California Davis
Davis, California

Matthew Scates
msscates@ucdavis.edu
University of California Davis
Davis, California

Josh Pike
Japike@ucdavis.edu
University of California Davis
Davis, California

ABSTRACT

We conducted a study examining the effects that user perceived importance has on user willingness to wait for a website to load. We utilized four loading screen types: a countdown, spinning gears, a progress bar, and a blank page, in order to examine the interplay between a websites importance and these loading screen types and how this interplay affects a users wait time. We used the CAS login page for the University of California, Davis as a high importance website, along with an About Me page and Blog to serve as low importance websites. We found that higher perceived importance of a website does correlate with users being willing to wait longer amounts of time and that an increased amount of information provided to a user about the load time was the most effective means of getting the users to wait for this high importance website to load.

KEYWORDS

Availability, Importance, Computer Security, Security Design

1 INTRODUCTION

The availability of a website is a major factor in the acquisition of new users and the retention of regular users. How users react to this availability and the interplay between the importance of a website and its availability is vital to understand as this plays a significant role in user satisfaction.

Websites that run slow tend to have less users visiting the site since when the page loads up, users often think the site is broken and will become less engaged. A users perceived importance of a website also may have a major factor in determining how long said user is willing to wait for a website to load. We believe that the higher the perceived importance of a website the longer a user will be willing to wait for it to load. Additionally, we believe that the amount of information presented to the user during the loading of a website has a large part in affecting how long a user will wait for a page to load. We wish to see if and how a users wait time is affected by the combination of various loading types and the importance of a website to see if there are certain loading screen characteristics that may benefit websites or web-pages suffering from low availability.

Here, we developed a site that purposefully created a pseudo-loading screen to test how long users wait until they click off the page. We utilized four loading screens: a countdown, spinning gears, a progress bar, and a blank page. Users were given a choice of three websites to click on which would redirect them to said website, but be presented with the pseudo-loading screen to serve an artificial load time. We used the CAS login page for the University of California, Davis with UC Davis students as the participants in order to create a website of high perceived importance. We also created an About Me page and Blog to serve as websites of low perceived importance. If a user visits our page, we will log their information, including how long they waited and if they gave up. Based on this data, we will create an availability model that shows the impact of a slow website where the data is not readily available to users who are authorized to access it.

2 BACKGROUND

Prior to this work, we had some experience in web design through a python package known as web2py, a framework much like py4web, except it uses an older version of python (Python 2), where as py4web uses Python 3. We also have experience in building data models.

2.1 Prior Research

We have looked into a few resources that are relevant to data availability. We have a couple resources that goes over the basics of availability. We have also found a couple of related works. One such related work talks about a study by Gerdes et al. that found that "40 percent of people abandon a website that takes more than three seconds to load" [5]. This is essentially a basis of our experiment since we are also using loading times to see how long people are willing to wait for the web page to load. A study in 2003 conducted by Jess Hohenstein et al. [8] found that having a loading screen for a website gives a perceived performance of a website for users, and found that different types of loading screens affects how well users think the website performs. Specifically, this study compared interactive loading screens with progress bars and static animation, and found that an interactive loading screen gives the shortest perceived time while static animation has the longest, with progress bars in the middle. For our data availability model, we will include different types of loading screens because if the loading screen types were to affect perceived performance, it could also affect data availability.

2.2 CIA

The Confidentiality-Integrity-Availability (CIA) model is the basis of creating policy in computer security.

2.2.1 Confidentiality. Confidentiality means that data has some privacy, and that no one can access the data unless they are authorized to do so. It also means that those who need access to the data should have access to that data.

2.2.2 *Integrity.* Integrity means the data remains accurate and trustworthy because it can only be changed by authorized users. The data is also prevented from being tampered with by unauthorized users. Integrity can be protected through signatures, certificates, encryption. Integrity can be verified by preventing the data from being replicated so that unauthorized users can't replicate the data and distort it.

2.2.3 *Availability.* Availability, the aspect we are focusing on in this paper, is the insurance that the data for users and applications is readily available whenever it needs or intends to be used. If there is a disaster, or an internet outage, there should be a back up location or a disaster plan so that authorized users can still obtain them. Data availability is determined by factors such as implementation of handling data, the network, the security system, and the policies that surround the data. It is usually measured by ratio of total uptime over total time, as well as the current state of the data.

Our experiment is based upon the data availability where rather then have the data completely unavailable, we have our data not readily available, which is simulated by our pseudo-loading screen.

2.3 py4web

PY4WEB is a web framework for rapid development of efficient database driven web applications. It is an evolution of the popular web2py framework but much faster and slicker. This is perfect for our development, since web2py (what we have used previously) had Database limitations that would have prevented us from being able to accurately gather user data for our model.

2.4 Vue.js

Vue.js is the front end JavaScript framework that we settled on for our project. Designed for single page web-applications, it is perfect for our use case. Having a lightweight framework was important, because we needed to ensure our base website had the lowest possible loading times. As seen in Figure 1, we managed to get a perfect PageSpeed Insights score, which measures performance on both mobile and desktop devices. Vue also enabled our "About Me" web page, which is essentially a single user post feed.

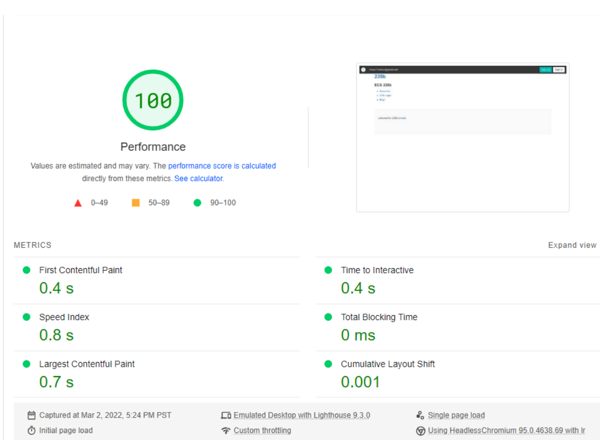


Figure 1: Web Page load times

Importance	Web Page
High	UC Davis Central Authentication Service (CAS) login
Low	Blog Post about Bonsai Trees About Me Page

Table 1: Web Pages mapped to importance

2.5 PythonAnywhere

PythonAnywhere is an online IDE and web-hosting service, specifically made for popular WSGI-based web hosting (such as py4web and other server-based python interfaces like bottle, Flask, and Django). The streamlined workflow provided to us by PythonAnywhere allowed for rapid (re)deployment and development.

2.6 Unforeseen Consequences

During the development of the website, and logging we ran into a few, but large, consequences with our technologies chosen.

2.6.1 *web2py.* Initially, instead of py4web for our web framework, we wanted to use web2py, which is a slightly simpler web framework that is the precursor to py4web. All of us have had experience with web2py, so we decided on it. Unfortunately, web2py does not scale very well. The database file does not allow for concurrent accesses, so when multiple users were having their data logged, there would be a race condition and only one of the users will have their data logged. This was resolved by pivoting to py4web.

2.6.2 *busy waits.* An issue we faced is the busy wait the user experiences after selecting the page they want. A busy wait is a type of waiting that chews up process time as the webpage does a wait, and during this event, no other event gets processes, including the timer that times how long users wait. It is vital that this wait is accurate and allows the user to leave whenever they want. This happened because the wait was not being handled asynchronously and due to the nature of vue.js, this implementation of the loading screen did not work. We decided to use a redirect page that the user will be directed to to serve their randomized wait time (and loading screen), and then will be directed to the desired page. This way the user will have the ability to go back to the previous page if they want and interact with the page in general.

2.6.3 *gathering user data.* . We initially thought that gathering user data for the model would be a trivial task, as we had access to a mail listing with over 1500 active users. Turns out about 6 percent of users actually clicked our link, and 20 percent of those people didn't interact with our loading pages.

3 METHODOLOGY AND IMPLEMENTATION

Our project contains three main web pages: the main UC Davis CAS login page, an about me page, and a blog post. We picked these pages because they show two different classes on importance to the user as seen in table 1. When clicking the respective link to each web page, the user will be greeted with one of four loading screens that will load the web page in 0 - 40 seconds exclusive.

3.1 Web Pages

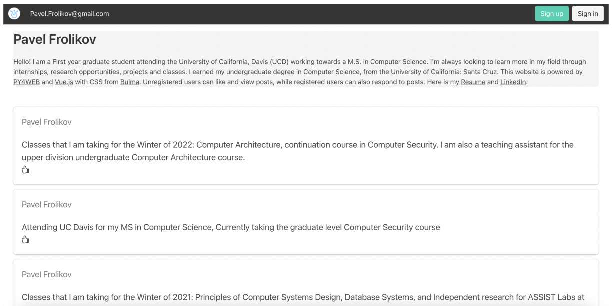


Figure 2: About Me Page

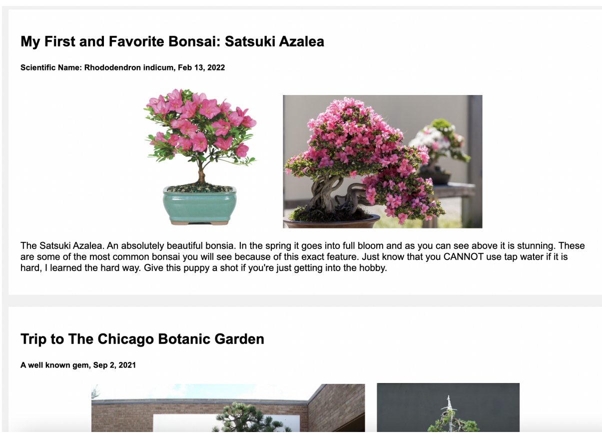


Figure 3: Blog Post

3.1.1 about me. The About Me page, as seen in figure 2 is a low importance simple post feed. Users are able to like and dislike the posts on the page. Registered users are able to respond to the main posts.

3.1.2 Bonsai Blogger. The Blog web page figure 3, similar to the About Me page, is also a low importance page. Consisting of a few articles about bonsai trees.

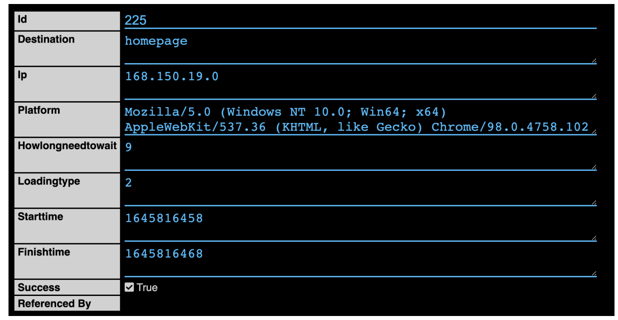
3.1.3 CAS. The UC Davis Central Authentication System is a high importance Web Page for the majority of the student and faculty body. Initially we had recreated the CAS page ourselves, but quickly realized that our web page looked like a phishing page, so we had our redirect page simply redirect to the actual CAS login page.

3.2 Loading Screen Types

Table 2 contains a brief description of all of the loading screens implemented as well as a visualization of each one.

3.2.1 Gears. The Gears are an in-determinate loading indicator, which means that the user knows that the site is loading at all, but does not know how long they need to wait.

3.2.2 Countdown Timer. The countdown timer, on the other hand is a text based determinate loading indicator, as the user knows exactly how long they need to wait. The timer's text goes down every second until the redirect occurs.



Id	225
Destination	homepage
Ip	168.150.19.0
Platform	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.102
Howlongneedtowait	9
Loadingtype	2
Starttime	1645816458
Finishtime	1645816468
Success	True
Referenced By	

Figure 4: Example of py4web log Database Entry

3.2.3 Progress Bar. The progress bar, similar to the countdown timer is also a determinate loading indicator, but is animated. The user knows roughly how long they need to wait as they see the progress bar consistently incrementing, but they do not know exactly how long they will need to wait.

3.2.4 Blank Screen. The final loading screen that we implemented is just a blank page. This is most definitely in-determinate as there is no information given to the user at all.

3.3 Logging

The majority of our users come from the UC Davis student body. As seen we in figure 4, we log the following data whenever a user successfully waits out their randomly generated wait time, or if they navigate away from the redirect page.

- Incident ID
- Destination Page
- IP Address
- Platform (Browser, OS, Mobile)
- Time needed to wait
- Loading Screen Type
- Start time (UNIX Time)
- Finish time (UNIX Time)
- Successfully redirected

We ended up gathering data from over 100 unique IP addresses, for a total of 252 data points.

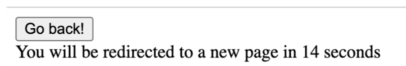
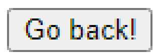
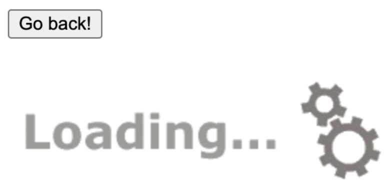
4 MODEL AND RESULTS

With the data we gathered we sought to expand upon the works of Gerdes et al. [5] and Jess Hohenstein et al. [8] by coming to conclusions on the effect of importance on how long a user is willing to wait and if the amount of information shown to the user by a loading screen can increase the amount of time a user will wait for websites of varying importance. We hypothesize that the more important a website is perceived to be by a user the longer they will be willing to wait and that the more information is given to the user on a loading screen the longer they will be willing to wait.

4.1 Modeling Effect of Importance

Without an accurate labeling of a website's perceived importance by a user, it would be impossible to understand the effects importance

Table 2: Loading Screens

Loading Screen Type	Visuals	Description
Countdown Timer		(1) Countdown timer appears and decrements every second until page redirect. (2) determinate (3) text based
Progress Bar		(1) Progress bar that increments every second until page redirect. (2) determinate (3) animated
Blank Screen		(1) Blank white screen with "Go back!" button. (2) in-determinate (3) neither text based nor animated
Loading Gears		(1) Animated gif with cogs spinning until page redirect. (2) in-determinate (3) animated

has on anything. Therefore, we specifically designed our three websites/web-pages to cater to UC Davis students, taking advantage of their common background and experiences to create a consistent hierarchy of importance. This common background led us to use the CAS login page as the website of highest importance. This alone does not make the CAS login page the most important however. When we sent our study in a few large UC Davis Discord servers we made sure to explicitly include the statement, "We are just making sure that you can login thru CAS after clicking on the link, but feel free to check out the other two pages linked." so that the users who did click on the link would perceive the CAS login page as being the most important of the three. The other two pages, the blog and About Me, were relegated to being low importance as the users who decided to participate in the study would most likely not find these other two seemingly meaningless pages important whatsoever. With this importance hierarchy in place (which can be seen in Table 1) the data we gathered would be able to be used reliably to model the effect of importance on a user's wait time. Through examination of the average percent of the required wait time users waited for across various required wait durations and cross comparing this between the three websites we would be able to determine the value importance has on a user's willingness to wait, as we should see that the most important web-page (the CAS login) has the highest average percent wait time across all or most required wait lengths.

Additionally, the reason we randomized the required wait time for users when they accessed a page is because we believed that if we had a single static loading time (say 40 seconds) like some previous studies had, user's would assume that the links we provided did not work after their second or third attempt. While users clicking off after a certain amount of attempts is not inherently bad (in fact it is essential to getting reliable and accurate data) we assumed that because we were using students as subjects and sent our links in Discord servers with in total no more than 2,000 to 2,500 members (likely less than half of which actually check the server regularly) we would only get a small percent of those total members interacting with our site. So, by giving users the possibility to successfully load before they would possibly click off, they would realize that our website did indeed load successfully and perhaps that this lack in availability was because this was a student made website, causing them to actually wait for the websites they deemed important. We do not believe that this choice affects our determination of importance level because after all, our websites must have been deemed important enough to interact with in the first place if a user chose to interact with it giving each website an equal base importance. This, in conjunction with our phrasing of the message mentioned previously, would add on additional heightened importance to the CAS page.

4.2 Modeling Effect of Loading Screen Types

Modeling the effect different loading screen types had on a user's wait time was more straightforward than modeling importance as the loading screen types do not depend on the perceptions of a user. We utilized four different screen types from most information presented to the user to least information presented to the user:

a countdown, a progress bar, spinning gears, and nothing; all of which can be seen in Table 2. We hoped to see that users would be most likely to wait the entirety of the required "load" time when they are presented with accurate information on how long it will take, the countdown and less-so progress bar, less likely to wait for the entirety of the "load" time when they were presented with something animated to know the website hadn't crashed or ran into an error, the spinning gears, and rarely would wait the entirety of the "load" time when given no information at all. Like when modeling importance previously, by observing the average percent of the required wait ("load") time users waited for across varied required wait times, we would be able to determine to what extent the various loading screens benefit or hurt a user's willingness to wait longer amounts of time.

4.3 Results: Importance

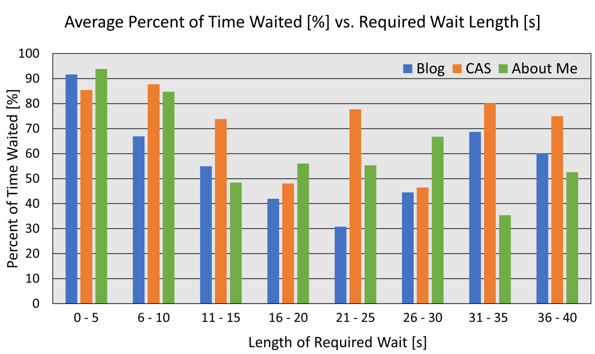


Figure 5: Average percent of the required wait time users stayed for compared to the required wait/load time separated into 5 second wide bins for each web-page

Bin [s]	Blog	CAS	About Me
0 - 5	91.67%	85.45%	93.86%
6 - 10	66.96%	87.76%	84.72%
11 - 15	55.00%	73.81%	48.44%
16 - 20	41.98%	48.07%	56.07%
21 - 25	30.78%	77.78%	55.35%
26 - 30	44.50%	46.45%	66.71%
31 - 35	68.72%	80.00%	35.39%
36 - 40	60.23%	75.00%	52.60%

Table 3: The average percent of the required wait time a user waited for compared to the required wait/load time separated into 5 second wide bins for each web-page

From the 252 data points we received, 38 of them were unusable as they represented users who clicked off too fast for their times to be recorded. From these now 214 data points we observed the average percent wait times for the three pages across required wait time bins separated into five second intervals which you can see in Figure 5. From this graph we see that most users tended to wait most of the "load" duration when the required wait was between 0 through 5 seconds, which should be the case as [5] Gerdes et al. found a similar result. While the CAS page did not have the highest average percent in this bin standing at 85.45% with the Blog and About Me pages standing at 91.67% and 93.86% respectively (seen in Table 3) these results do not go against our preconceptions as they are all relatively high which we should be seeing from the study stated previously. We then see a steady decrease in wait percentages until the 21 through 25 bin, where this downward trend ceases to be consistent for the remainder of the bins. This downward trend is attributed to the average person not wanting to wait longer than 3 to 5 seconds, thus pushing the percentages down for higher required wait times. Along this downward trend, as well as after it, the CAS page dominates the other pages having, on average, 14% and 10% higher wait percentages across each individual 5-second bin for the Blog and About Me respectively. This result demonstrates that users were more likely to wait for the high importance CAS page than the lower importance Blog or About Me, especially for the longer of the required wait times. We attribute this correlation to how we phrased our study when sending it to students, pulling their focus to the CAS page thereby forcing them to hold the CAS page as the highest importance of the three web-pages. The CAS page's perceived high importance then enabled the users to be more willing to wait longer durations.

4.4 Results: Loading Screen Type

	Countdown	Gears	Bar	Nothing
Blog	68.57%	72.25%	64.82%	31.73%
CAS	83.77%	72.09%	69.68%	44.35%
About Me	70.37%	84.47%	73.01%	37.15%
All Pages	73.41%	76.63%	70.02%	36.58%

Table 4: The average percent of the required wait time users waited for across all data points for a each page and all pages put together

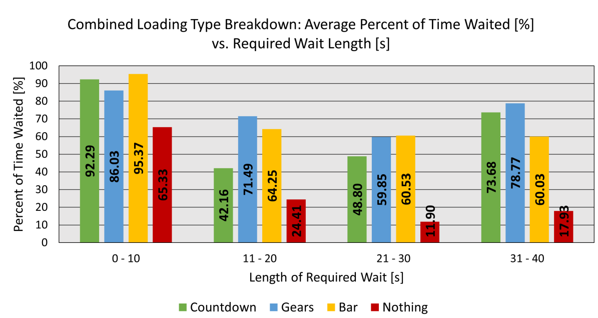


Figure 6: This shows the waiting time percentage across all of our web-pages.

The results from the loading screen type were much more prominent then compared to the results of importance. In Table 4, we see that the users who visited our site are more likely to give up

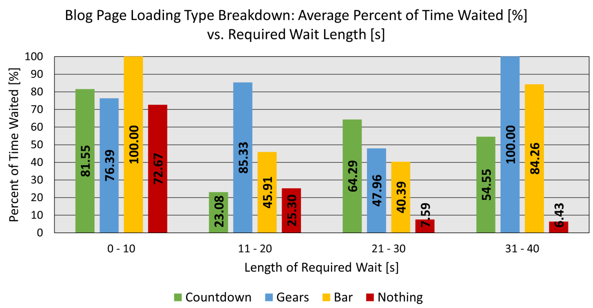


Figure 7: The length of time people are willing to wait for the blog page varies based off the average length of required waiting time and the loading screen type.

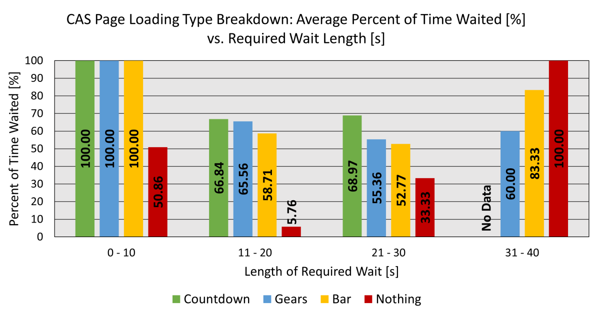


Figure 8: This shows how long people are willing to wait for the CAS page to load up based on the loading screen type.

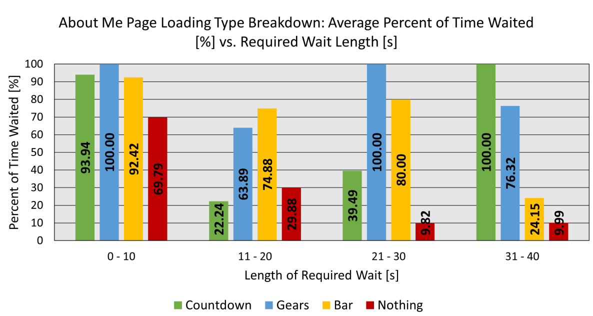


Figure 9: This shows how long people are willing to wait for the "About Me" page to load.

on waiting for the page to load if the page is blank, with a average percent wait for all web-pages at 36.58%, versus having any amount of load information provided to the user, with the lowest average percent wait for all web-pages at 70.02% for the progress bar. This trend remains consistent for all the web pages in our site, which we expect to happen because users who see the blank screen might have thought the site was broken, or have no idea when the page will finish loading.

While the blank loading screen has the lowest percentage of users who finished waiting, the progress bar loading screen performed worse than the spinning gears, which contradicts Hohenstein et al's study. We found this surprising because the progress bar would give a better, more accurate idea on when the loading will finish than the spinning gears. In addition, the spinning gears also performed better than the countdown across all pages except the CAS page which goes against our hypothesis that the more load information is provided to a user, the longer they will be willing to wait. We believe this is the case because when load times that are longer than what a user is willing to wait are presented to the user they will click off sooner than if they hadn't have known the wait time at all. This is somewhat supported by Figures 6, 7, 8, and 9 where the countdown's average percent wait falls after the 0 to 10 second bin. We also see that in all figures other than the CAS page (Figure 8), the countdown's average percent wait actually grows from the 11 to 20 second bin through the 31 to 40 second bin. This is quite an intriguing result as users appear to be more likely to wait for the countdown load type as the wait time increases. We are unsure the exact reason for this as it could be a result of users who attempt to refresh hoping to get a lower wait time getting frustrated with not getting lower wait times and accepting to wait for a longer amount of time; however, more testing is needed to confirm this.

Taking a look at the individual pages, we see the CAS page has the highest percentage of users waiting through the entire blank screen sequence, most likely due to its highest perceived importance. The CAS page also had the highest wait percentage for the countdown leading us to believe that the amount of load information provided to a user does cause increased willingness to wait for more important websites.

4.5 Future Work

Overall, we believe that while these results are promising, more data is required to draw any definitive conclusions. Originally we hoped to use our data to produce a scatter plot and a regression model of the percent waited versus the required wait times, but due to the low number of data points this plot failed to show any conclusions. We hoped to get a data point for every required time possible, however this did not happen which further made us unable to see any trends when using a scatter plot. We also want to bring attention to the trend of wait percentages for the countdown loading screen increasing as the required wait increased. If this trend does hold up to further scrutiny, it could be very useful to retain users when the availability of a website is known to be low.

5 CONCLUSION

We found that the perceived importance a user places on a website does affect the willingness said user has to wait longer amounts of time. Even for wait times that exceeded 30 seconds users tended to wait for the more important web-page for at least 70% of the required wait time whereas for the less important web pages their percent of wait times fell steadily, on average, as the required wait increased. We believe these results indicate that the importance of a website will allow for users to wait extended amounts of time and that this trend will be further proven if more data is collected.

Additionally, we hypothesised that the more loading information was presented to a user the longer they would be willing to wait. While generally we found this to be true, when users were presented

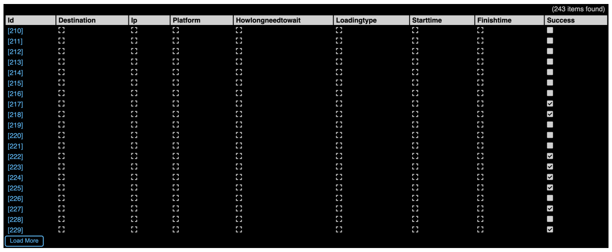


Figure 10: Example of py4web log Database

with no information they didn't wait nearly as long as other load types, this was not the case when comparing a countdown, spinning gears, and a progress bar. When looking at all web-pages combined, the spinning wheel correlated with higher percentages of required wait time across the majority of required wait times going against our hypothesis. We did find, however, that for the CAS page the countdown was the most effective loading type and did cause users to wait for longer amounts of time leading us to believe that for high importance websites our hypothesis holds true.

ACKNOWLEDGMENTS

Special thanks to Doctor Bishop at the University of California, Davis for supporting our study and helping us along the way.

A INSTRUCTIONS FOR RUNNING (MACOS)

(1) cd into working directory
(2) pip install py4web==1.20200510.1
(3) py4web setup apps (y,y,y,y)
(4) py4web set-password (password will be for dashboard)
(5) unzip 235BProj.zip and place 235BProj folder into (workingDir)/apps
(6) py4web run apps
(7) You will be redirected to http://127.0.0.1:8000
(8) To access the logs (figure 10), go to http://127.0.0.1:8000/_dashboard/dbadmin?app=235BProj&dbname=db&tablename=logTable

B INSTRUCTIONS FOR RUNNING (LINUX)

identical to MacOS

C INSTRUCTIONS FOR RUNNING (WINDOWS)

(1) Download py4web binaries https://github.com/nicozanf/py4web-pyinstaller/archive/refs/tags/v1.20200510.1.zip
(2) unzip "py4web win 1.20200510.1.zip" http://web2py.com/init/default/download
(3) Go to that folder and run 'py4web-start set-password'
(4) unzip 235BProj.zip and place 235BProj folder into (workingDir)/apps
(5) then 'py4web-start run' from the Command Prompt / Terminal

D IMPORTANT LINKS FOR RUNNING

(1) Important Links
(2) py4web dashboard http://127.0.0.1:PORT/_dashboard
(3) home page http://127.0.0.1:PORT/235BProj
(4) where to see logs http://127.0.0.1:PORT/_dashboard/dbadmin?app=235BProj&dbname=db&tablename=logTable

E PROJECT LINKS

http://www.web2py.com/
https://www.py4web.com
https://ucdavis.edu/
http://nob.cs.ucdavis.edu/classes/ecs235a-2021-04/
https://github.com/y76/ecs235b/

REFERENCES

[1] (Matt, Josh, Pavel)
(Matt, Josh, Pavel)

[2] Overleaf Guide to LaTeX
https://www.overleaf.com/learn/latex/Tutorials

[3] Wikibooks LaTeX Guide
https://en.wikibooks.org/wiki/LaTeX

[4] Hao, Yanbin: Understanding loading animations — types and applications
https://uxdesign.cc/e41ba914b847

[5] John H. Gerdes and Betsy Bender Stringam: First Impressions in the Digital Age: A Comparison of Internet Load Times Between Desktop and Mobile Platforms
https://tinyurl.com/yc3td7ts

[6] Betsy Stringam and John H. Gerdes: Service gap in hotel website load performance
https://www.emerald.com/insight/content/doi/10.1108/IHR-09-2018-0012/full/html

[7] Janakiraman et al. Automated system design for availability
https://tinyurl.com/2c8eb2mc

[8] Hohenstein et al. Shorter wait times: the effects of various loading screens on perceived performance
https://tinyurl.com/peuc2rvz