

Increasing Municipal Transparency through Council-Monitoring Applications

Mckervin Ceme

A.B. Computer Science, 2016

mceme@princeton.edu

Alan Kaplan, PhD.

Faculty Advisor

ak18@cs.princeton.edu

I pledge my honor that I have not violated the Honor Code during this paper. Mckervin Ceme.

Abstract

Knowledge of municipal government decisions amongst the average citizen is not thoroughly understood by many individuals. Furthermore, even for those individuals who feel particularly inclined to get involved in local government find getting information about recent legislative activity done by city council and other governing bodies can be difficult and cumbersome to access. As a result, this project aims to facilitate the process of finding legislative actions performed by a governing body. In particular, an older, open-source application called CouncilMatic was updated to provide a richer user interface for individuals to be able to search and browse their local municipality for recent legislation. This project is written for individual towns, and the premier test case is the township of Princeton, New Jersey. The goal of this project was to create a richer front-end experience for the front end user, as well as provide an easy-to-use backend service for local governments to host for their citizens, in an attempt to increase transparency in municipal government between average citizens and city officials.

1. Introduction

Political ignorance, which is the lack of knowledge of recent and even past political events, is a widespread problem in America. This issue ranges from national government – the amount of people who did not know that the Affordable Care Act was a law was a staggering 44% of Americans, according to the Cato institute [8] – to the lack of knowledge about city council agendas. This lack

of knowledge only further serves in abetting government mistrust at all levels of government. In a Pew Research Poll released in 2014, approximately 40% of Americans trust their local government [6]. In order to tackle this problem, the federal government has funded numerous projects that work to increase the sheer amount of policy data available on the Internet in the hope that larger access to data could lead to increased trust and transparency in government. While projects like <http://www.data.gov> have helped developers with more APIs that can access federal data, many of these APIs are unavailable for much smaller municipalities. Increasing the amount of transparency in local governments is a much larger problem to address. Having a stratified and varied number of municipalities, each with different approaches to dialogue between the average citizen and local government makes the challenge of increasing transparency and trust in government more daunting. Consider Figure 1, which shows a PDF file of recent legislation in a local municipality. Not only

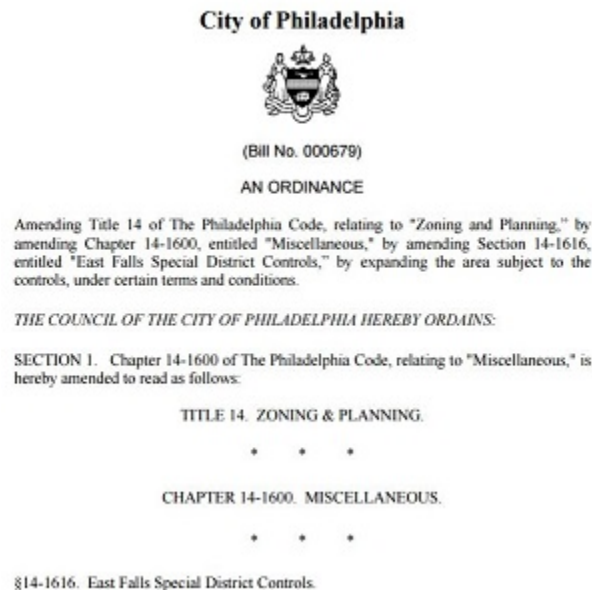


Figure 1: A part of a bill.

does the format present challenges for the average reader to comprehend (and the format varies by municipality), often times files such as these are buried behind multiple clicks on a town's website. Even when towns do provide access to information, the data itself is behind websites that are not managed as well as private web addresses. Unfortunately, this can stifle transparency efforts, as the inability to easily find information can only lead to more confusion about a town's policy efforts, as

opposed to fostering political participation.

Nevertheless, in order to try and tackle this particular problem, one can utilize open-source platforms to develop a general framework for local governments to use that provides a reliable, easy-to-use interface for the average citizen as well as a simple administration page for the local governing body, so that they can upload legislative minutes and decisions without requiring an extensive amount of preparation and maintenance for municipalities. This is the idea behind this project. By creating an application that can sift through legislative data and present it in a clean format, the presentational aspects of municipal data (among these aspects include summaries of policies, legislative minutes, how votes were cast in local elections, which district a policy applies to, and the like) increases, allowing for a more robust dialogue amongst concerned citizens and city council officials.

2. Problem

2.1. Motivation

This project tackles the problem of transparency in local politics. In particular, it addresses the issue of a lack of a platform that has both a clean user-interface for the end-user (the concerned citizen) and has a relatively straightforward installation process for the municipality. It does this by redesigning web interfaces using modern techniques and technologies, such as updated versions of HTML, CSS, and jQuery to provide an intuitive, minimalistic, clean user-interface. Furthermore, this application addresses the problem of installation by using core web framework technologies like Django [1], the RESTful framework, BeautifulSoup (a web-scraping API), and PostgreSQL, a relational database management system. When these technologies are combined, this application can be launched in many different towns and cities and can provide the ability to pull legislative data (such as bills, agendas, minutes, votes, and the like) and present the data in a format that is easy to read and understand for the average citizen. Transparency is a bedrock aspect of strong and just democracies. Therefore, utilizing web technologies to help increase transparency can lead to increased political participation as well as decrease levels of corruption, as well as provide scrutiny to specific bills that may disproportionately affect one group in a community versus another.

2.2. Approach

The major goal of this project is to increase transparency and communication channels between average citizens and town officials. To help achieve this goal, the CodeForAmerica [3] open source project, Councilmatic [7], was used as a foundation for this project. Using their GitHub page as source code, this project aims to re-configure CouncilMatic to be compatible with more modern web hosting platforms such as Heroku and Amazon Web Services. Councilmatic proved to be the proper starting point for this application for a number of reasons. Primarily, the original application was written using Django as a high-level Python-based web framework. The structure of Django applications, which include sub-folders for every tool used in a given project, is not only clean, but it facilitates the process of further customizing a given project. In addition, as new versions of Django are released, porting changes to the base code is not a very time-intensive task, because often times certain functions are merely deprecated, and as such only require nominal edits to the core application. Another reason that Councilmatic was chosen was because it was initially developed in America with local town councils in mind. In particular, Councilmatic allows a person to search for bills by councilmember, by their local district, and records legislative data such as minutes, agendas, and legislation. While the idea of using “minutes” in a municipality is not unique to the United States of America, having an application that can return data that more-or-less fits the American style of government proved to be more useful in the long run (as opposed to applications like TheyWorkForYou, which is based out of the United Kingdom and presents data in a parliamentary manner). Moreover, Councilmatic used a PostgreSQL database with a PostGIS extension. What this allows developers to do is “geo-tag” a piece of legislation, that is, to determine where a particular bill or policy will be applied. This is an important feature of council-monitoring applications, as citizens should be able to information about policy changes (new bike lanes on a certain street, as an example) based on the zones by which a certain bill is appropriated towards. This native support provides the functionality that a council-monitoring application should have: district-sorted legislative data and localization support for municipalities in the United States.

Once it was decided that Councilmatic was the base for this project, then the rest of the project

involved updating the GitHub so that it could work for more modern technologies (such as Twitter Bootstrap, newer versions of jQuery, etc.). Major feature enhancements include:

- Change settings to support PostGIS engine on the Heroku cloud platform.
- Added newest version of Bootstrap template framework to the source code.
- Add multiple supportive binaries that allows the application to go live on Heroku
- Add calendar to main dashboard view that shows past, present, and future agenda meetings.
- Keep recent legislation feed fixed on the screen in order to give citizens a quick view of bills that recently went to the legislative floor.

3. Hosting Application Instance

A major proponent of this project was ascertaining which web hosting platform to host the updated GitHub code on. Previously, the application was hosted on the dotCloud framework, which allows developers to configure a simple web stack and push their applications to the cloud. However, installing newer binary dependencies, which this application needed, proved difficult. Furthermore, the dotCloud infrastructure that the original application was originally built on has been altered significantly. In fact, the old dotCloud infrastructure has been re-written into an entirely new framework, dotCloud PaaS. As a result, many of the features of dotCloud that had been useful for deploying older Django-based applications have been redesigned. These changes have ranged from newer shell commands to push applications to a more protracted method to insert add-ons. Since the newer dotCloud framework had just been recently rolled out, launching an application on this platform would not be ideal for a municipal-based application. Not only does the application require the PostGIS extension for the PostgreSQL database, but other binary libraries are necessary. These include GEOS, GDAL, and PROJ4. And adding binary packages to the newer dotCloud platform can be difficult for municipal employees to install.

Therefore, a simpler platform was necessary. In order to host this application, Heroku [2] was selected as the web hosting platform for this project since it supports applications that can take advantage of PostgreSQL databases. Like dotCloud, Heroku provides a back-end web platform

in which developers can upload their applications to. However, Heroku offers a few advantages over the dotCloud platform. Primarily, Heroku offers an intuitive way to add binary add-ons in the form of buildpacks. Buildpacks are open-source packages that can be added to Heroku instances that are already running with a straightforward command. These can range from binary solutions that would normally require tools like subversion and/or manual compilation to database add-ons, such as a geospatial addition like PostGIS. In addition, Heroku natively uses PostgreSQL to store information, which only minimizes the amount of work a town or local developer would need to do to get a zone-based council-monitoring application to function. Figure 2 describes the flowchart for the application, in particular how the source is housed on Heroku and where the data itself is housed. What makes this stack beneficial to outside developers is that it can pull data directly

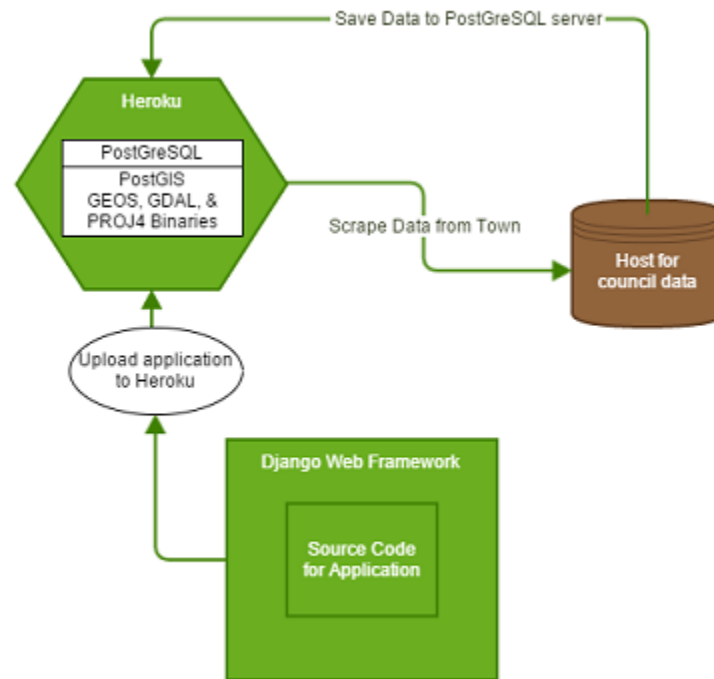


Figure 2: Stack for this application.

from municipalities' websites. Since the application does not replace systems that town councils use, edits can be made fairly frequently to the base code without altering outside data sources. Moreover, the developers at Heroku provide much more up-to-date systems documentation than other web-hosting platforms, again helping local developers get instances of their application live

much easier. The source code gets upload to Heroku, and from there, the application can scrape data from the town at regular intervals and update independently; no developer interference needed.

3.1. Legistar®

Many municipalities use a system called Legistar® by Granicus® that hosts all of their council data. Legistar is an automated workflow system that allows municipalities to simply submit agenda items on their local docket, and this vendor can generate agendas, minutes, and final action data and publish this information to town websites. Councilmatic was initially built to take advantage of this particular infrastructure. In particular, the web-scraping capabilities involve using data found on the Legistar system, and parsing items into data that can live on a PostgreSQL database.

4. Front-End User Interference

From the end-user perspective, a major goal of this project was to also provide a clean aesthetic, so that a user can readily find information that pertains to them. Therefore, the front-end design principles were to alter the templates for the front-end into a more modern, clean user-interface. This includes such things as bright fonts, large dialog boxes, and feeds that provide the most recent legislation added in a given number of days. This approach, often called Responsive design [2], emphasizes flat, minimalist web elements that can respond to a variety of different screen sizes. A major goal of this application is to provide a user-interface that can easily be ported over to phones and tablets, as well as traditional computers. In order to achieve these design principles, a newer version of Bootstrap was used. Specifically, version 3.3.4 was added to this project. By using the newer Bootstrap template, the design challenges of this project could be ameliorated. Bootstrap is a mobile-friendly, responsive HTML, CSS, and JavaScript web platform that provides developers with intuitive ways to design websites. The newer version takes advantage of “responsive design” principles such as flat buttons on sections, sleek color palettes, and optimized preprocessors for rapid page load. Figure 3 shows the initial template architecture that Councilmatic originally had. Figure 4, by contrast, utilizes the new design principles of modern websites. Information is readily visible right when the user opens the web page. The recent legislation ticker, located at the left side

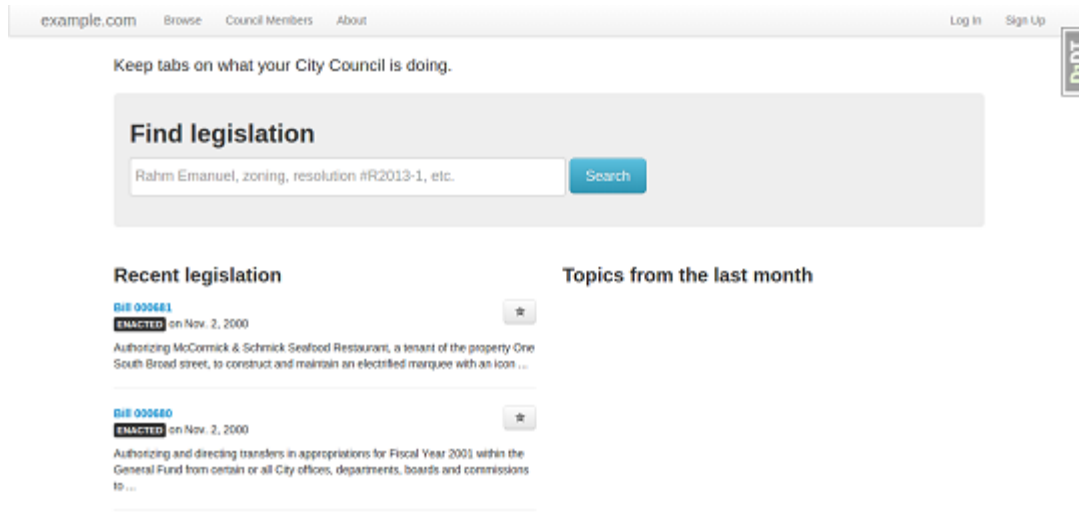


Figure 3: Original Template

of the page, is a fixed structure that provides the most up-to-date legislative decisions made by a town council. This feed is always visible, even as the user scrolls down the rest of the page. Quick



Figure 4: Newer Design Principles

access to legislative information is an important aspect in template design, as citizens should not need many clicks to get quick access to information; the details that a user may need should be available the moment the website is accessed. The truncated bill summaries for each piece of recent legislation enables an end-user to quickly understand what a certain bill entails. On the right of the recent legislation bar, in the main activity area of the page, lies the search function, which enables users to search for council decisions based on the councilperson, the district and/or keywords in a

given bill.

Under the search feature lies the calendar, which provides a clean interface for an end-user to browse legislation by month, starting with the current month. Not only does the calendar view offer an additional view for the legislative actions in a given time period, but it also allows the developer to scrape upcoming legislation as well, and place important dates here. This is an additional feature that can give citizens more information about ongoing council debates prior to final votes on approval. One of the major goals of updating this application was to give citizens the advanced knowledge of bills in addition to recent legislation. The calendar feature, built using the modal features of Twitter Bootstrap, achieves this goal and allows developers to scrape other aspects of a government website to fill in the calendar.

Compared to previous templates that Councilmatic used, the newer design emphasizes a flattened structure. On every page, the navigation bar rests at the top and is fixed even as users scroll down.

4.1. Challenges

One of the major design challenges associated with this project was to outline the steps by which a developer, whether a local developer or the municipality, could get an instance of the application live onto the Heroku web platform. This proved to be a difficult challenge to overcome, due to numerous reasons. Among these include outdated documentation for pushing the application to dotCloud. Previously, only two simple commands were necessary: `dotCloud create` and `dotCloud push`. However, much of the support for adding custom buildpacks has been recently re-written in the redesigned dotCloud PaaS cloud system. With this motivation, Heroku was chosen as the cloud services hosting company that would handle instances of the application.

This is not to say that there were not challenges associated with pushing the geospatial Django application to Heroku. In particular, installing GEOS, GDAL, and PROJ4 require more than just adding the custom build packs to a live Django application. The general outline for the necessary changes that needed to be made can be seen below:

- Change the default Procfile to support running the server from a particular sub-folder.

- Open a cedar-10 stack on Heroku.
- Use the custom buildpack to install the necessary binary packages.
- Provision a PostGIS extension for the PostgreSQL database that Heroku uses.
- Make edits to the source code that enables the application to go live on Heroku.

While some of these steps did not require too much effort (only one line of code needed to be changed in the Procfile), others required more protracted steps. The first thing that was required was create the cedar-10 stack on Heroku. This is a specific web stack that Heroku provides that, for our purposes, allowed the binary packages to rest in a specific location on Heroku (cedar-14, in contrast, houses the binary applications in a different root folder, thus causing dependency issues when trying to use Geo-Spatial libraries like GDAL). Once PostGIS and the buildpack that housed the dependencies were installed, the more arduous step involved editing the source code to handle the database specifics of Heroku. This involved finding the library paths for both GEOS and GDAL as well as configuring packages such as BeautifulSoup4 and dj-database-url. The former is a necessary package for web scraping, while dj-database-url is a newer library that provides the 'engine' for this application (in this case, the 'engine' is PostGIS). These changes were necessary, as some of the code has since been deprecated as newer versions of Heroku and Django have been released.

5. Evaluation

The current stable version of this project can be seen in Figure 4, which is the main dashboard, while the right shows the view for a given bill. The dashboard page was designed using vibrant colors and a placeholder image of Raphael's School of Athens in order to create an atmosphere that encouraged civic participation. With this in mind, I pitched this project to Princeton, NJ town officials in order to get their feedback, in particular on what a municipality might want from an application of this sort. Many towns are actively involved in increasing transparency between its town council and its citizens. It is important to note what a municipality might want in an application of this sort. Projects like this one that work in the domain of council-monitoring can offer clean user-interfaces that municipalities can take advantage of when designing their own websites. Furthermore, some

of the specific architecture that a city uses may not be well known, and having that information would be beneficial to the developer, as he/she can then scrape data much more efficiently if the programmer can understand exactly how legislation is hosted by a municipality.

In talking to local officials and showing them the progress of the semester, I was able to learn quite a few things. Municipal workers often feel overburdened, and as such those who work in the Town Clerk's office tend to use systems that they are already familiar with when uploading data. This is an important insight. Initially, the goal of this project was to replace many of the intermediary systems that municipalities use. However, when discussing with Princeton officials, I discovered that many of the goals of this application can be achieved over time without having to force municipalities to change the manner in which they upload legislative data. In terms of the aesthetic of the project, local officials were pleased with the template design that I had implemented. In particular, the clerk's office liked the idea of a recent legislation side bar and a calendar to show upcoming legislative meetings. The clerk's office felt that immediacy of information was one of the most important features in transparency efforts. Often times, transparency requires not just open data, but ease of access, and in this respect the project emphasized that. Moreover, the aesthetic, being modern and using Bootstrap as a base, appealed to the city officials. It is a drastic improvement over the city's current website in terms of the design principles of the project. The sticky navigation bar, the flat design of HTML elements, and the Helvetica font all helped to convince city officials that a Bootstrap-based approach would be the best method to ensure ease-of-use for average citizens.

The Clerk's office also wanted an additional video feature in the project. The Princeton, NJ town council offers video of some council meetings in addition to the actual agendas themselves. Thus, they requested that on a bill page, a user be able to view the accompanying video that may or may not be available. In addition, the Clerk's Office explained how employees record the minutes and agendas. In Princeton, NJ, the clerk's office uses a product called MinuteTraQTM by IQM2 [5]. Similar to Legistar, this is a different application that provides templates for generating agenda PDFs, minutes PDFs, and links to video files as well. As this project is an ongoing partnership with the town, in the future I will be adding this feature in particular for the local township. The

clerk's office also liked the idea that this application could be built on top of their already existing uploading mechanisms. This is because, having gotten used to the IQM2 system, city employees felt that this project had merit in that it would allow them to continue to upload data in the manner that they saw fit, but furthermore they would not have to worry about making the data be presentable to the average citizen. This was the general sentiment upon viewing the project in its current state. In future revisions, I will be adding support for scraping the MinuteTraqTM system, so that the project can scrape both Legistar® towns and IQM2 cities.

6. Related Work

There has been open-source work in the field of increasing transparency between municipalities and their citizens. An open-source application in the United Kingdom is called TheyWorkForYou, an open-source application by mySociety. This is a UK Parliamentary monitoring application that provides the recent voting history of all members of Parliament (MP) and members of the House of Lords. The work here is a rigorous study of the manner in which federal officials in the United Kingdom voted on a range of topics and the bills associated with those topics. Although this application operates more-so on a federal government level as opposed to municipal government, the application was useful in research, as I looked into ways of showing how council members participated in policy decisions. There has also been open-source work that has tackled the transparency issue from the legislator's perspective. In particular, one organization called DemocracyOS [4] has an innovative idea, which is to create a platform by which a city council member can directly interact with their constituency through a comments section located at the bottom of bills that the councilperson authors. The idea is that by having a discussion section at the bottom of a proposed bill, citizens can aid in the design process of a bill. This includes voicing their concerns through dialogue at the bottom of the bill page as well as offering alternative strategies to approach a specific issue. Their approach is interesting in that it focuses primarily on the dialogue by using the format of a news article with a comments section at the bottom, in the hope that the councilperson hosting the application (each individual councilperson individually hosts

a DemocracyOS framework) engages in dialogue. While this organization created customizable frameworks that individual council members could follow, the goal of this project was to create a general framework that a city can use, so that incoming and outgoing council members do not need to maintain records themselves of past and present legislation. One of the largest companies that work in this space is a startup called FiscalNote. FiscalNote provides an immense amount of data for all state legislatures as well as the federal government itself. It combines machine learning with large amounts of data to forecast policy outcomes based on a number of factors, including party affiliation, key legislators, and other local factors that shape the likelihood of a bill passing or failing on the floor of the legislature. Although the user-interface for this application is clean, the application was designed with policy and legal professionals in mind as opposed to the average concerned citizen. The average citizen may not need the data analytics for a particular bill; they may simply want to know exactly what it is. Furthermore, while FiscalNote used open data, the actual application itself is not free. Thus, this inherently limits the amount of people able to use that platform to those who are willing to pay. Ultimately, this defeats the purpose of this particular project; the overall goal is to help inform average citizens about the local politics in their town or city. Another open-source application is Councilmatic, an application developed by the non-profit Code for America [3]. This application provides subscription services for citizens, enabling them to follow legislation that they deem is important to them. Furthermore, the application also provides information about current city councilmen and councilwomen, including the district they served and bills they have sponsored/co-sponsored. This project has many features that tackle the problem of increasing transparency in local government, however much of the open-source code is out of date, and as such the user-interface design suffers. The application does not have many of the design principles of more up-to-date websites. Still, the work provided a good base platform to build off of, as it contains template HTML and CSS files that could be edited to make the platform appear more user-friendly, and *inviting*. The design of the landing page (Figure 5), while minimalist, does not entice end-users in the same manner in which other Bootstrap-based websites do: a lack of images and vibrant colors make the page seem not as inviting as Figure 4, which has a flatter design and

colors that pop-out.

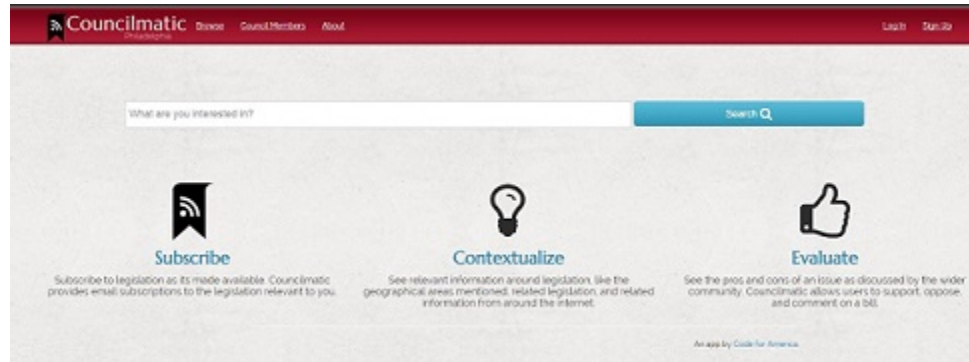


Figure 5: Simple Landing Page

7. Conclusion

7.1. Areas for Improvement

Nevertheless, there are still areas in which this application can be improved. The most important addition to a project of this scale would be to set up a general scraping framework that can, without too much developer input, allow websites that are not on the Legistar system to be able to be scraped. By increasing the amount of systems that the application can scrape, the amount of cities that will be able to implement this type of application will be augmented. This will require a larger re-write of the core functionality of the application. Furthermore, it may be difficult for local developers to uncover what systems that municipalities use to handle their legislative data, if any at all. Some town websites may not host council agendas, bills, and minutes. And even the ones that do provide open access to their legislative data often have non-intuitive user-interfaces that do not provide ease-of-access to legislative data. Therefore, a major improvement aspect to this field would be to provide an easy-to-implement web scraper that would only require minor changes to the source code between cities with distinct legislative-hosting platforms.

Since town websites are not innately homogeneous, then another improvement would be to incrementally support different types of legislative systems from many different providers. Examples of this would include open data from the federal government, direct hosting of PDF files, and

third-party vendors such as IQM2. In the future, a person should be able to enter in their zip code or town name, and be able to pull up the appropriate council-monitoring instance for the location that they choose. This would require a much larger and more robust infrastructure to house that much data (one would have to store the recent legislation for an extraordinary amount of cities across each state, let alone all of America). Nevertheless, it is an endeavor that would provide interpretable information for the maximum amount of individuals.

Other improvements that could be made to the council-monitoring application would be a way to view the video files of certain agenda meetings. Some municipalities provide this information to their citizens in order to increase transparency. One could imagine a user clicking on an agenda for a particular day, and the user would see a video playing. At the same time, the transcript of the paper would be beneath the video, so that a user could easily watch the video with a quick reference of the agenda only a few pixels away. A more aesthetic improvement would be to add a slideshow of a given community, and set this as the background for the instance. Many modern websites have a video reel composed of small clips that ultimately serve to make the web page more aesthetically pleasing to the average person. Furthermore, when browsing through bills in particular, one can be able to browse bills/resolutions by district, so that a citizen can see the recent legislation that may affect a certain area.

7.2. Council Monitoring Limitations

It is important to note, however, that while council-monitoring applications can provide easy and simple mechanisms that any end-user can use, these applications cannot make up for the amount of citizens that either do not have access to a computer/mobile device with website support or simply do not use electronic means as a form of dialogue between themselves and their town council, for a variety of reasons (lack of understanding, health, and the like). For example, currently in the city of Princeton, Princeton Public Works has to create leaflets that outline policy changes to those residents who are not necessarily as tech-savvy as their younger cohorts. While it is important to note that an application such as this cannot solve all of the transparency and information exchange

problems, it can help raise political intrigue amongst individuals who may be peripherally interested in their town council.

Another feature that I initially wanted to add was to have election-season data available in the project. During an election year/six months before an election, special aspects of the website would be unlocked. These would display the nearest polling stations to your location, who the candidates are for a particular election, and how/when should an average citizen get registered to vote. These type of features almost supersede the general idea of a ‘council-monitoring’ application, as many more scripts would have had to have been written. While that kind of feature may be beyond the scope of the application, it would have been an interesting data source to add. In addition, another feature that I had originally planned to implement was a Twitter feed that users could follow and see what was new in terms of the Princeton Fire House and the local police department. Before I narrowed down my application to focus on city hall, my dream was to design application that would completely remove the need to go to a municipality’s website. That would include information about a whole host of items, tax information, etc. Interestingly enough, when the project idea was first pitched to municipal employees, one of the stipulations that the municipality wanted was the existence to links that redirect a user back to the government website. Therefore, I was forced to narrow down the project, lest it become a simple proxy to the local governments’ website. Furthermore, while having a lot of information is important for transparency purposes, in this case it may have worked to the detriment of the average citizen. If an individual just wants to quickly know what is happening around them, they do not need a plethora of dialogs and buttons that link elsewhere – just a simple, clean, minimalist design that has the direct information necessary.

7.3. Conclusion

This council-monitoring application is a functioning program that provides quick access to important information in a clean, easy-to-access manner. Using Bootstrap, I was able to change many of the outdated templates into templates that have a more modern, sleek, feel to them. This ultimately helped to reduce the amount of clicks necessary to find a bill, from the 4-5 clicks, as outlined in

Figure 6, to as little as 1 for bills recently discussed on the council floor. The reduction in the

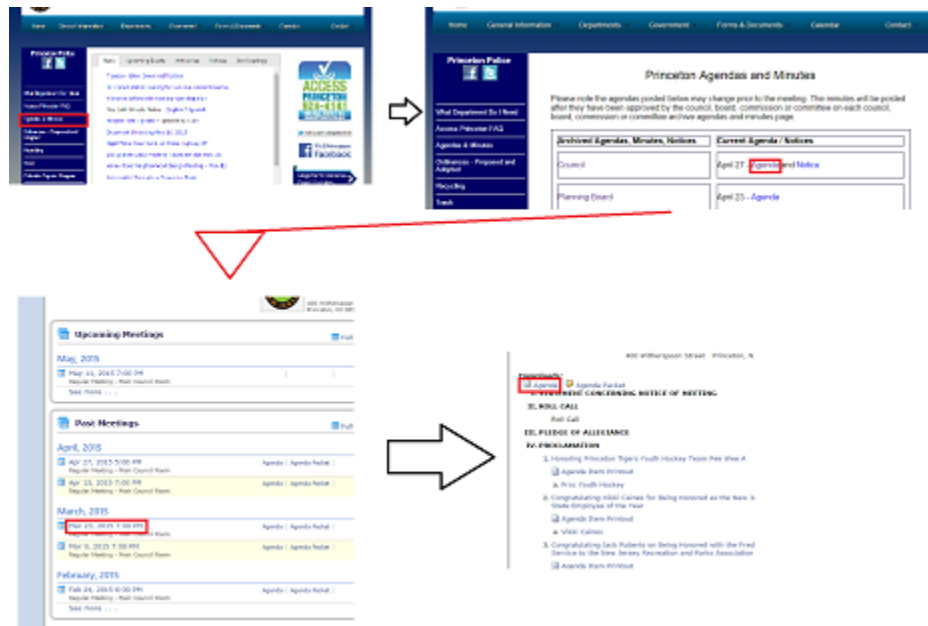


Figure 6: Simple Landing Page

number of clicks leads to quicker access to information, thereby giving the end user more time to parse the data and get a stronger grasp on the debates in the legislature. The advantage that Bootstrap offered in the design was that it scales fairly easily to mobile devices. As a result, the experience that a user has on their laptop or desktop is very similar to the experience they will have on their mobile device.

Although the application was not initially designed to run on Heroku, by changing the source code to enable Heroku integration, the project not only gets all of the benefits of a cloud web-scaling application, but in addition the up-to-date documentation that Heroku provides allows the older instances of the application to be quickly updated, if and when certain functions get deprecated. These are in the form of migration guides, which are useful debugging tools. Moreover, the architecture of Heroku allows the developer to sync the database and make migrations manually so that a developer can populate a PostgreSQL database. This is of the form `heroku run <python>`. While Heroku does have certain features that are only available for paid users, the scaling ability as well as the ability to run individual python scripts from the Terminal prove that Heroku was the best cloud service to choose from.

This project is not static, as many more changes to the source code will be made in order to expand the list of eligible cities that use this application. Primarily, development will continue to support IQM2-based townships such as Princeton, NJ. In addition, more town-specific support will be integrated into the base of the application (i.e. options for video, audio, and general agenda pages) in order to better satisfy the needs of the town as well as the needs of the average end-user.

References

- [1] *Django Software Foundation*, 2005. [Online]. Available: <https://www.djangoproject.com/>
- [2] *Heroku | Cloud Application Platform*, 2007. [Online]. Available: <https://www.heroku.com/>
- [3] *Code for America*, 2009. [Online]. Available: <http://www.codeforamerica.org/>
- [4] *DemocracyOS - Better Decisions, Together*, 2012. [Online]. Available: <http://www.democracyos.org/>
- [5] *IQM2 - Intelligent Meeting Management*, 2015. [Online]. Available: <http://www.iqm2.com/>
- [6] C. Cillizza, *There are a remarkably small number of people who trust the government*, 2015. Available: <http://www.washingtonpost.com/blogs/the-fix/wp/2015/04/21/the-remarkably-small-number-of-people-who-trust-the-government/>
- [7] M. Poe, *A subscription service for city council legislative information, started in Philadelphia.* <http://councilmatic.org>, 2013. Available: <https://github.com/codeforamerica/councilmatic>
- [8] I. Somin, *Democracy and Political Ignorance*, 2009. Available: <http://www.cato-unbound.org/2013/10/11/ilya-somin/democracy-political-ignorance>