Institiúid Teicneolaíochta Cheatharlach

INSTITUTE of TECHNOLOGY

CARLOW

At the Heart of South Leinster

Online Voting System

Research Document

Name: Mark Kelly

Student Number: C00198041

Course: Bachelor of Science (Honours) Software Development

Supervisor: Dr. Lei Shi

Date: 18th April 2018

Abstract

The aim of this document is to provide a detailed account of the research that was conducted, in advance of developing an online voting system for the student's union at Institute of Technology Carlow.

This document will record in detail, all the technologies that were researched and considered for this project. It will provide information on both frontend and backend technologies researched. It will also document the findings of similar applications available on the market.

A conclusion will then be drawn as to what the most suitable technologies to use for this project.

Table of Contents

1. Introduction

> *"It is enough that the people know there was an election. The people who cast the votes decide nothing. The people who count the votes decide everything." -- Joseph Stalin* [1]

The problem with current electronic voting systems is that they all suffer from a critical design flaw; they are all designed to be centralised, which means a single organization has control over everything. This includes control over the code base, database, algorithms for counting votes and any outputs the system may produce. This means that, with a current electronic voting system, after a voter votes on a ballot, the voter must put their trust in the organization to record and count their vote as the voter intended. Not being able to provide an independent output that can be publicly verified by everyone casts doubt into the mind of the voter. How can the voter trust such a centralized system? A lack of trust can potentially limit voter participation.

In order to create our electronic voting system, certain standard requirements have to be met: [2]

- **Security**
  - A "secure" voting machine means one that cannot be tampered with or manipulated in any way.
- **Functionality**
  - Correctly registering and recording all votes cast.
- **Privacy**
  - Voters have a right to a secret ballot and to cast their vote in private.
- **Usability**
  - A voting machine should be as intuitive to use as possible and contain clear instructions regarding how to vote.
- **Accessibility**
  - All people, including those with visual, physical or cognitive disabilities, must have the opportunity to independently cast their votes.

Our proposed electronic voting system will aim to provide these core requirements while exploring the possibility of using blockchain technology to log votes in a decentralised manner. Using blockchain technology (The technology behind Bitcoin) could provide transparency, resulting in less need for trust to be put into a centralized organisation that counts their votes behind closed doors. The blockchain could allow the voters to independently verify the results, removing the need to place trust in the organisers of the election. Once a vote has been confirmed in the blockchain and has further blocks built upon it, this vote for a candidate becomes immutable. This means that the entire outcome of an election will be stored indefinitely, and can be accessed at any time in the future.

This research document contains the detailed review of the research that was carried out in relation to this project. The research for this project is detailed under the following headings:

- **Background**
    - This section will discuss the background behind the project. It will describe what the system is supposed to do and where it fits in.
- **Related Systems**
    - This section will survey current systems available on the market. It will look at the pros and cons of these systems and how well they deliver the end result.
- **Technologies**
    - This section will look at what technologies can be used for This project. This is a broad section and will examine IDE's, databases, frameworks and programming languages. This section will research both frontend and backend technologies.
- **Conclusion**
    - The last section will review all the technologies that were researched and make a decision as to what the most suitable ones would be for developing this project.

2. Background

The ITC Student's Union want to allow students to vote online in the student elections. The student's union are looking to develop an online voting platform for IT Carlow students to use to vote in IT Carlow Students Union elections and referenda.

The student's union wish to keep the traditional ballot paper on campus on the day of voting as it creates a great buzz on the campus during that day, but unfortunately, there is no opportunity for students who are not on campus on that day, for example students who are out on placement, etc. The student's union love the marketing opportunity for the union provided by the traditional elections, so for this reason a hybrid system is required.

The student's union would like the system to require students to register their intention to vote online in advance of the election (so they can remove them from the central register used in polling stations), and they then have to log in to the system during polling hours to cast their vote. When students have registered to vote online and log in during polling hours (they get an automated reminder email on the morning of the election), and can then log onto the system to cast their vote.

The student's union would also like to link the system to the student's blackboard account. This will require us to discuss this option further with computer services.

The system must provide the following functionality:

- **Register for Voting**
    - Allow students to register online using their student ID
- **Cast Votes**

    - Students can place their vote and have it recorded correctly
- **Count Votes**
    - Give the correct results of the election
- **Display Results Graphically**
    - Display charts showing data visually
- **Full Security**
    - Fully test the system against cyber attacks
    - Show that an individual user's votes cannot be tracked while also showing traceability of the result.
- **Management UI**
    - An online based secure management console for setting up elections

3. Related Systems

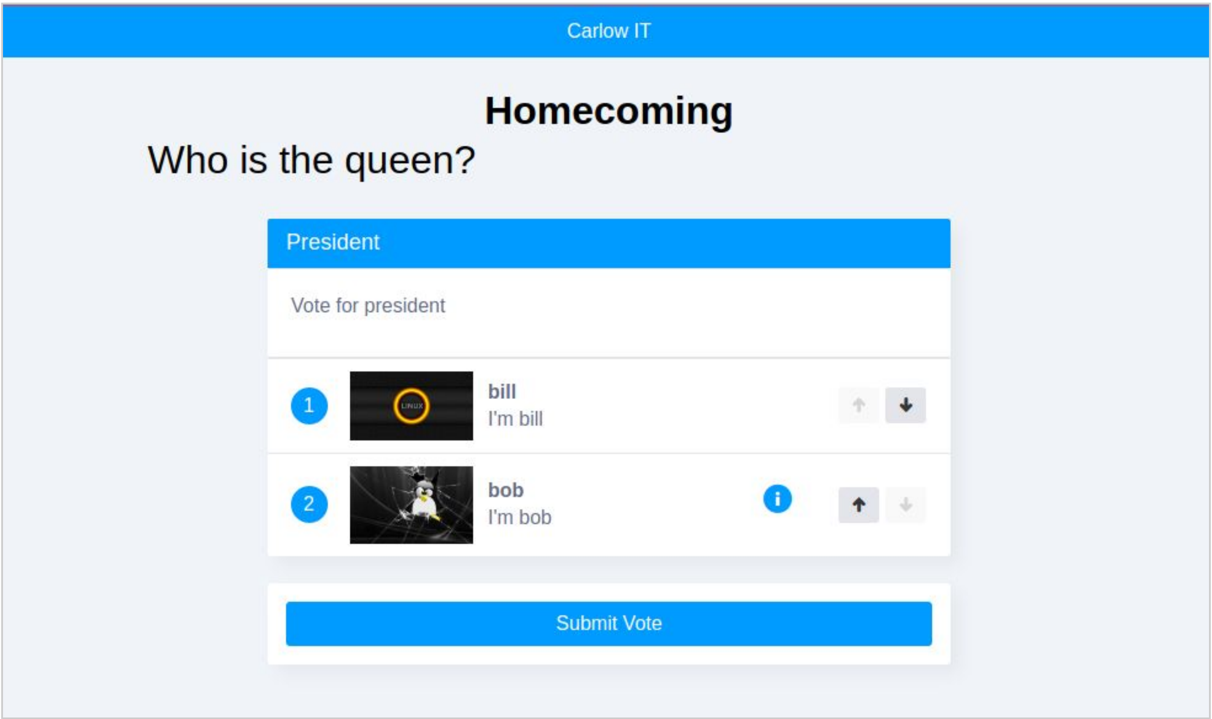
This chapter will look at some similar existing electronic voting systems available on the web. We will look at some of the nice and not so nice features of these solutions.

3.1 Election Runner

[3]

3.1.1 About

Election Runner is an online web application that allows you to build and launch secure cloud based elections for your school or organization. Election Runner has a clean, attractive and easy-to-use voter interface, as shown in, Fig.3.1. Election Runner allows you to create elections by adding election details like election name, dates and settings. You can also design the ballot by adding positions, candidates and questions with photos and bios, as shown in, Fig.3.2. Election Runner also provides a feature to allow the election administrator to import voters and ballots from an excel spreadsheet or CSV file [4].



Carlow IT

Homecoming

Who is the queen?

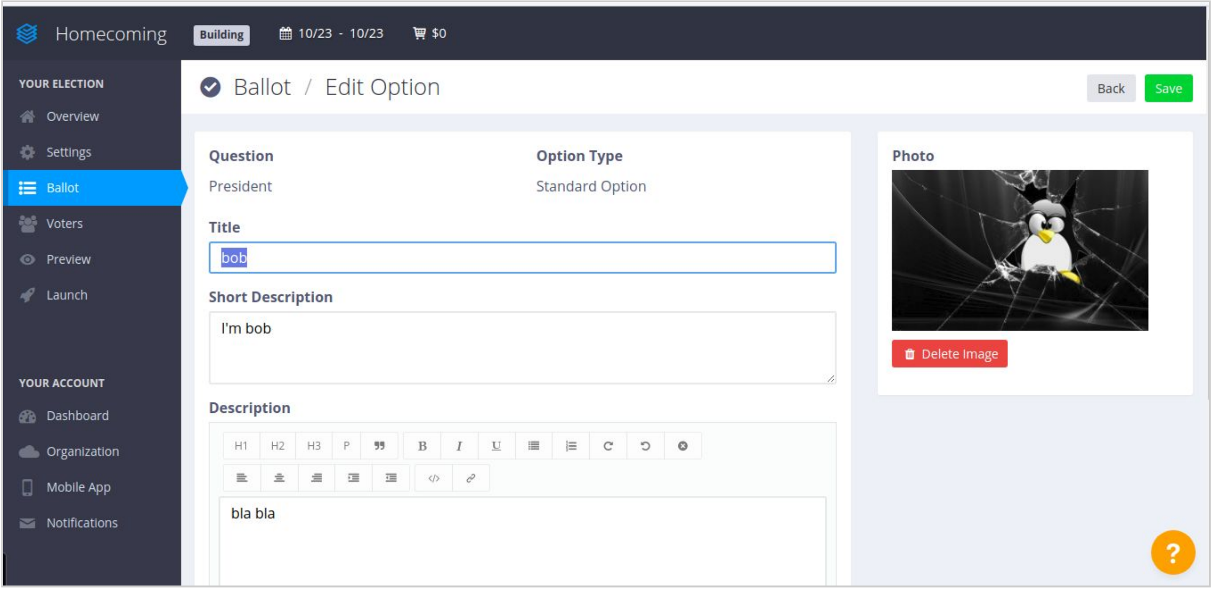
President

Vote for president

1 bill
I'm bill

2 bob
I'm bob

Fig. 3.1 voter interface [4]



Ballot / Edit Option

Question: President

Option Type: Standard Option

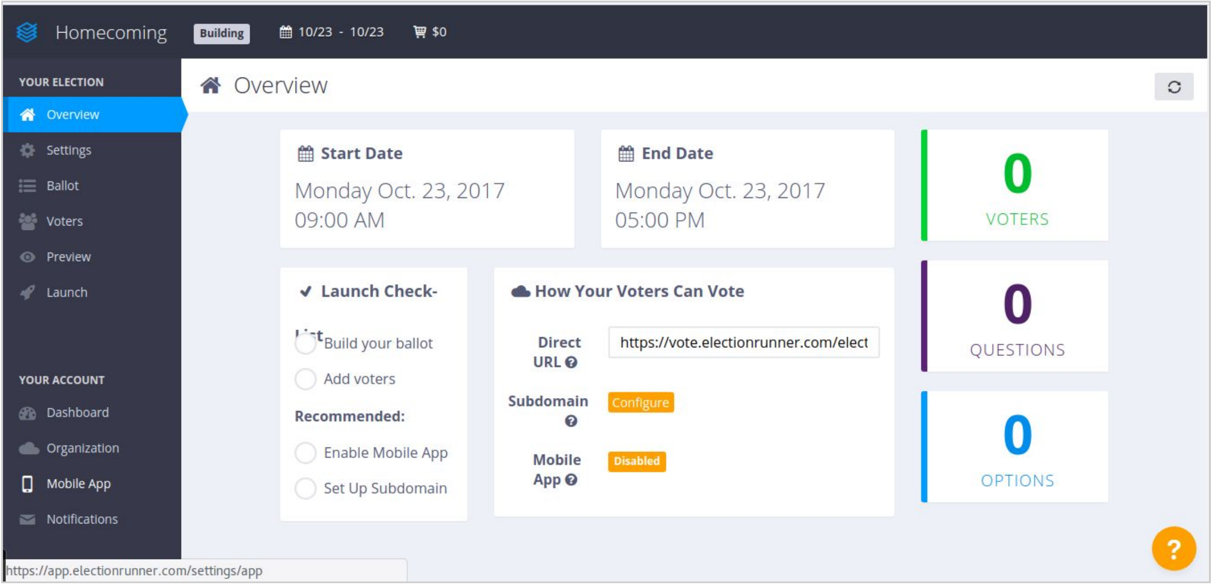
Title: bob

Short Description: I'm bob

Description: bla bla

Photo

Fig. 3.2 ballot screen [4]



Start Date
Monday Oct. 23, 2017
09:00 AM

End Date
Monday Oct. 23, 2017
05:00 PM

0 VOTERS

0 QUESTIONS

0 OPTIONS

Launch Check-List
○ Build your ballot
○ Add voters
Recommended:
○ Enable Mobile App
○ Set Up Subdomain

How Your Voters Can Vote
Direct URL: https://vote.electionrunner.com/elect
Subdomain: Configure
Mobile App: Disabled

https://app.electionrunner.com/settings/app

Fig. 3.3 overview screen [4]

3.1.2 Pros

Election Runner has a very nice user interface, as shown in, Fig.3.3. It's a relatively simple process to set up an election, add ballots and voters. My favorite part of the Election Runner application is the UI. There is also a cool preview screen to allow the election administrator to preview elections while you build them. There is also a nice feature to allow the voters to be imported on an Excel spreadsheet. Results of the elections are displayed graphically, providing election administrator with useful information at a glance.

3.1.3 Cons

Election Runner is not a hybrid system and does not allow for running concurrently with a traditional paper system. Although Election Runner claims their services are free, this is only for the first 20 voters. It costs $450 for 5000 voters [4]. Election Runner's system is centralized, which means voters have to put their trust in the designers of the system.

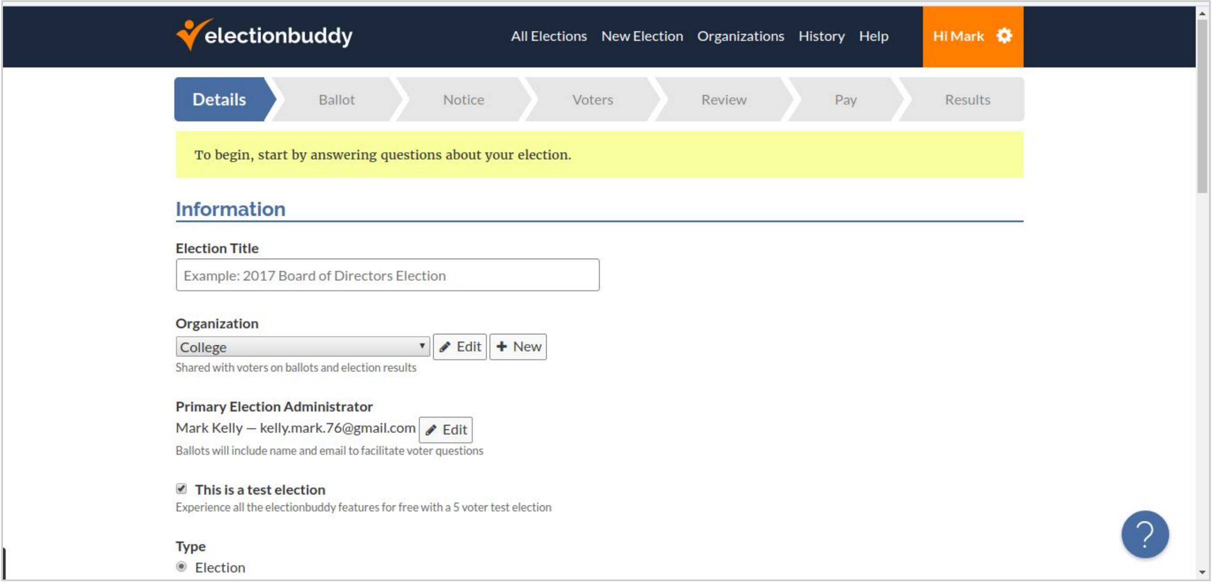
3.2 Election Buddy

electionbuddy

[5]

3.2.1 About

Election Buddy is also an online web application for easy, secure, anonymous online voting and elections. It's ideal for associations, schools, unions or anyone. You can notify voters through email, mail or SMS. It ensures elections are private and secure by using 256-bit encryption, the same security as major banks. The election administrator can receive alerts when email notices bounce so they can be resent [5].



To begin, start by answering questions about your election.

Information

Election Title

Example: 2017 Board of Directors Election

Organization

College

Shared with voters on ballots and election results

Primary Election Administrator

Mark Kelly — kelly.mark.76@gmail.com

Ballots will include name and email to facilitate voter questions

☑ This is a test election

Experience all the electionbuddy features for free with a 5 voter test election

Type

● Election

Fig. 3.4 create election [5]

3.2.2 Pros

One of the only things I liked about Election Buddy was that ensuring elections are secure by using 256-bit encryption. I also liked that you can receive alerts when email notices bounce so you can resend.

3.2.3 Cons

I really hated the UI/UX experience when using this application. I found it confusing to set up an election.

4. Technologies

This section will consider all the technologies available to take this project from paper to production. We will look at backend and frontend technologies, web frameworks, database solutions, deployment options, and blockchain technology.

4.1 Backend

Here we will look at backend languages, web frameworks and database technologies considered for this project. Not all of the following technologies will be used in this project.

4.1.1 Languages

For this project we will look at the two most popular server side languages, PHP and Python. PHP is commonly used as a server-side scripting language, while Python is appreciated for its dynamics, availability, and simplicity. Despite their almost identical popularity among developers, these two languages have many specifics and differences. The following sections will research which of them is better and decide which is more convenient for this project?

4.1.1.1 PHP

The PHP Hypertext Preprocessor (PHP) is a server-side scripting language that allows web developers to create dynamic content that interacts with databases. PHP is primarily for web development but also used as a general-purpose programming language. PHP code may be embedded into HTML markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code with the generated web page [6].

4.1.1.2 Python

Python is a widely used high-level programming language for general-purpose programming. It was created by Guido van Rossum and first released in 1991. Python is an interpreted language and has a design philosophy that emphasizes code readability. It uses whitespace indentation to define code blocks rather than curly brackets or keywords. Python's syntax allows programmers to express concepts in fewer lines of code than might be used in other languages such as C++ or Java. Python is an excellent language for web development and comes with many web frameworks like Django and Pyramid, including micro frameworks such as Flask and Bottle [7].

4.1.1.3 Conclusion

Both Python and PHP are open source with the latter dominating the server market. From my own experience on my second year project, I found the PHP syntax complex, messy and confusing. I detested how PHP integrated into HTML. This complexity leads to challenges in trying to maintain PHP code. PHP has a lot of flexibility, which can make working with it problematic. Python on the other hand is more rigid and famous for being more readable and easier to maintain. Python has a huge community with tons of documentation, tutorials and guides and is still evolving as a language. For my third year project I used Python, and I absolutely loved how it was very readable and easy to use. I found it much easier to maintain. I could easily pass my Python code to a team member and they could read and understand the code relatively quick and easy. I may be a little biased towards Python, but it will definitely be the server side language of choice for this project.

4.1.2 Frameworks

I done a lot of research on this area. I carried out a little research into PHP frameworks, just to test the water really, as I know at this stage that Python will be the server side language of choice for this project. I have some experience with the Flask framework and will be considering it for this project. It's impossible to research Python web frameworks without looking at Django. In this section we will look at one PHP web framework and compare and contrast two Python web frameworks. I did also have a quick look at Web2py, Bottle and Pyramid, but I did not invest too much time into them.

4.1.2.1 Laravel

Laravel is a relatively new PHP framework (it was released in 2011), according to Sitepoint's online survey in 2015, it is the most popular framework among developers, as shown in, Fig 4.1. Laravel has a huge community with an instant hosting and deployment platform. It's official website offers many screencast tutorials called Laracasts. Laravel has many features that make rapid application development possible. Laravel has its own lightweight templating engine called "Blade". It has an elegant syntax that facilitates tasks we frequently need to do, such as authentication, sessions, queueing, caching and RESTful routing. Laravel also includes a local development environment called Homestead that is a packaged Vagrant box [8].

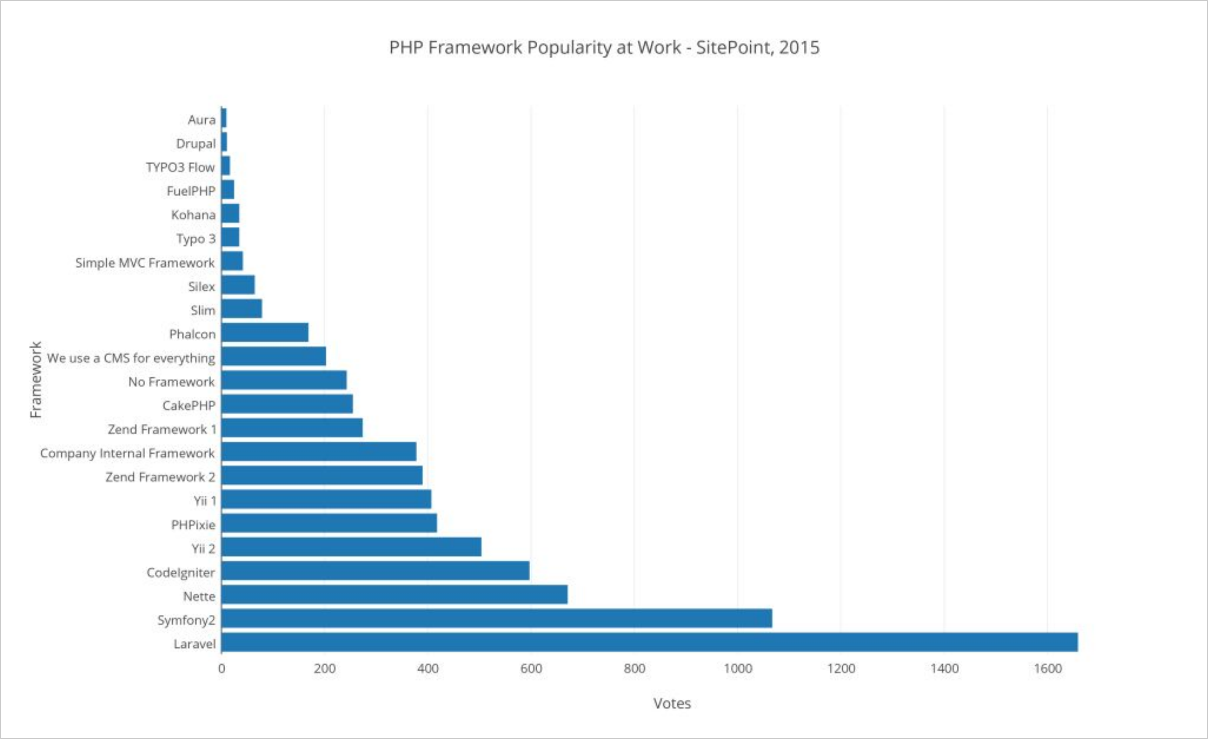


Fig. 4.1 PHP Framework Popularity 2015 [9]

4.1.2.2 Flask

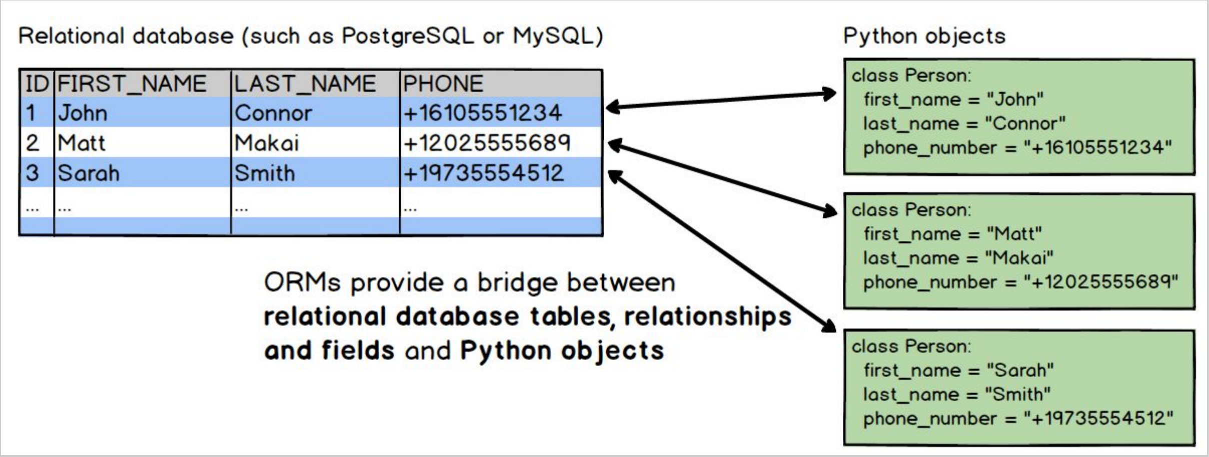
Flask is a micro web framework written in Python and based on the Werkzeug toolkit (A web service gateway interface utility library) and Jinja2 template engine. It is fully BSD licensed, extensively documented and comes with a built-in development server and debugger. Flask is called a micro framework because it does not require particular tools or libraries and apart from Werkzeug and Jinja2, has no dependencies. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask does support extensions that can add application features as if they were part of Flask itself. This allows for Flask to be light-weight by only adding the extensions that we need. This does mean we have to add and implement these extensions ourself [10].

Flask comes with another use full extension called Flask-Admin, which allows us to build an admin interface on top of our existing data models. It lets us create a user-friendly interface to manage our web service's data. Flask-Admin lets us build complicated interfaces by grouping individual views together in classes. Each web page at the frontend, represents a method on a class that has been added to the interface. When these view classes are tied to particular database models, they let us group together all of the usual Create, Read, Update, Delete (CRUD) view logic into a single, self-contained class for each of our models.

4.1.2.3 Django

Django is a high-level, free and open source Python Web framework. It encourages rapid development and clean, pragmatic design. It takes care of much of the problems of Web development, so we can focus on writing our app without needing to reinvent the wheel. Django provides a full-featured MVC Framework that not only comes with batteries included, it includes the whole kitchen sink. This makes Django a heavyweight framework. One of the most powerful parts of Django is the automatic admin interface. It reads metadata from our models to provide a quick, model-centric interface where trusted users can manage content on our site.

Django also comes with an object-relational mapper (ORM) library that automates the transfer of data stored in relational databases tables into objects that are more commonly used in application code. ORMs provide a high-level abstraction upon a relational database that allows us to write Python code instead of SQL to CRUD data and schemas in the database [11]. Fig.4.4. Shows a diagram of how an ORM works. Removing SQL statements from our application code can also prevent SQL injections, thus providing better security for our database.


Relational database (such as PostgreSQL or MySQL)
Python objects
ORMs provide a bridge between relational database tables, relationships and fields and Python objects

ID	FIRST_NAME	LAST_NAME	PHONE
1	John	Connor	+16105551234
2	Matt	Makai	+12025555689
3	Sarah	Smith	+19735554512
...	...	...	...

```
class Person:
  first_name = "John"
  last_name = "Connor"
  phone_number = "+16105551234"
```

```
class Person:
  first_name = "Matt"
  last_name = "Makai"
  phone_number = "+12025555689"
```

```
class Person:
  first_name = "Sarah"
  last_name = "Smith"
  phone_number = "+19735554512"
```

Fig. 4.4 Object relational mapper example [12]

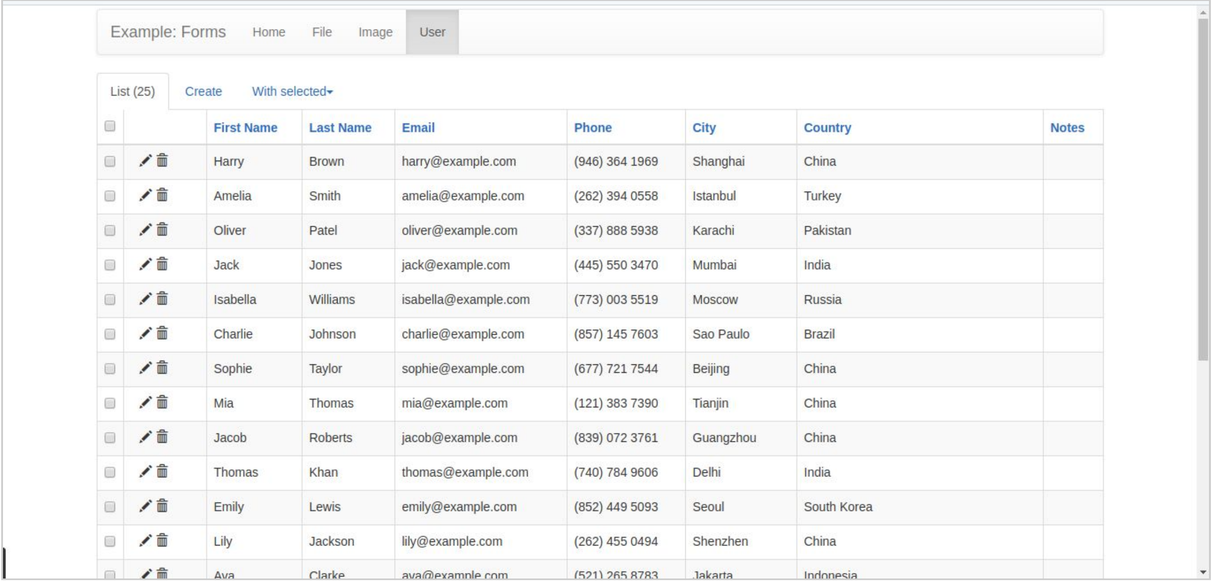
4.1.2.4 Conclusion

Django and Flask are both frameworks for creating web applications in Python. The Flask microframework has a bare-minimum approach and allows the developer to build up the application by adding on extensions to provide different functionality. Flask lets us build web services with very little overhead. It allows us to implement our project in a way that suits our particular application. But before long, we find ourself adding extensions like Flask-Login for authentication, Flask-Mail for sending emails and Flask-SQLAlchemy, a database abstraction layer and object-relational mapper (ORM).

Django on the other hand, follows a "batteries included" philosophy and gives you a lot more out of the box. So where Flask provides simplicity, flexibility and more control, Django provides an all-inclusive experience: you get an admin panel, database interfaces, an ORM, and directory structure for your apps and projects out of the box. Django is a heavier framework than Flask. When learning web programming, it can be harder to figure out which pieces are responsible for what functionality, and what you need to change to get the results that you want.

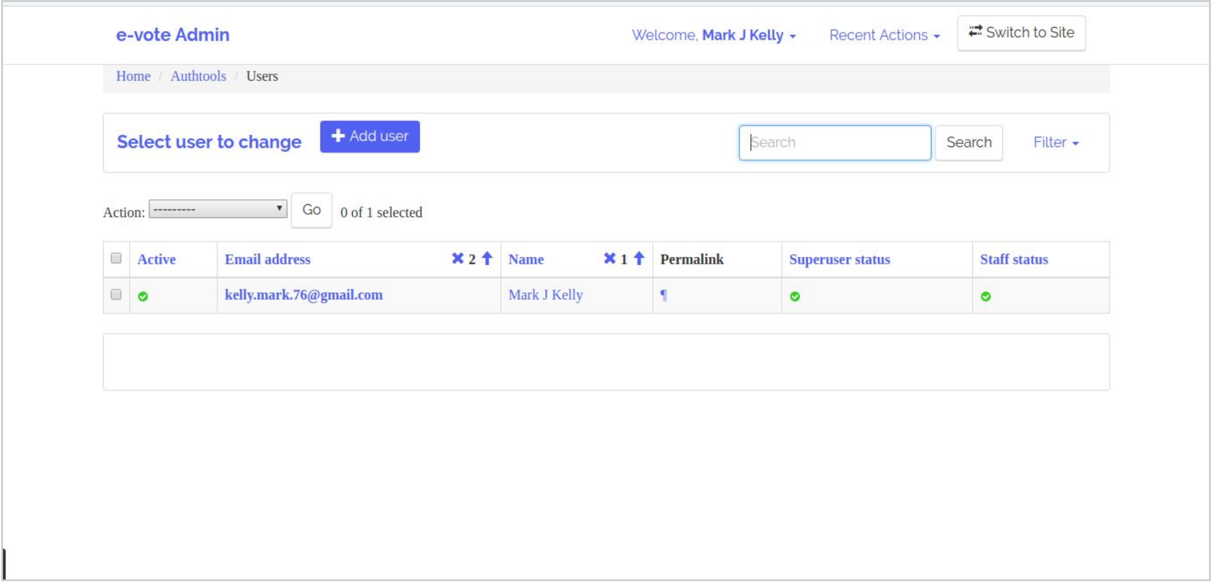
A clean, elegant URL scheme is very important in a web application. Django lets you design URLs however you want, with no framework limitations. In order to design URLs for our web app, we need to create a Python module called a URLconf. This module will allow us to map between our URL patterns (These are simple regular expressions) and our Python functions or views. Django’s URLs can be tricky to work with as a new Django developer, but it is well worth learning as it allows us to provide clean usable URLs for our web application [13].

One of the most powerful parts of Django is the automatic admin interface. Django provides a production ready user interface to allow us to create, read, update and delete (CRUD) our models in our database. Flask also comes with an admin interface for interacting with our models, but Django’s admin interface is miles ahead of Flask’s admin interface. As part of my research, I created two simple apps, a Flask app and a Django app. I wanted to get a feel for both admin interfaces. I found Flask’s admin interface very good, but Django’s admin interface is production ready and it felt really polished in comparison to Flask’s. Fig.4.2 and Fig.4.3 shows a comparison between the Flask and Django admin interfaces.



□		First Name	Last Name	Email	Phone	City	Country	Notes
□	✎ 🗑	Harry	Brown	harry@example.com	(946) 364 1969	Shanghai	China	
□	✎ 🗑	Amelia	Smith	amelia@example.com	(262) 394 0558	Istanbul	Turkey	
□	✎ 🗑	Oliver	Patel	oliver@example.com	(337) 888 5938	Karachi	Pakistan	
□	✎ 🗑	Jack	Jones	jack@example.com	(445) 550 3470	Mumbai	India	
□	✎ 🗑	Isabella	Williams	isabella@example.com	(773) 003 5519	Moscow	Russia	
□	✎ 🗑	Charlie	Johnson	charlie@example.com	(857) 145 7603	Sao Paulo	Brazil	
□	✎ 🗑	Sophie	Taylor	sophie@example.com	(677) 721 7544	Beijing	China	
□	✎ 🗑	Mia	Thomas	mia@example.com	(121) 383 7390	Tianjin	China	
□	✎ 🗑	Jacob	Roberts	jacob@example.com	(839) 072 3761	Guangzhou	China	
□	✎ 🗑	Thomas	Khan	thomas@example.com	(740) 784 9606	Delhi	India	
□	✎ 🗑	Emily	Lewis	emily@example.com	(852) 449 5093	Seoul	South Korea	
□	✎ 🗑	Lily	Jackson	lily@example.com	(262) 455 0494	Shenzhen	China	
□	✎ 🗑	Ava	Clarke	ava@example.com	(521) 265 8783	Jakarta	Indonesia	

Fig. 4.2 Flask-Admin example



Action: --------- 0 of 1 selected

□	Active	Email address ✖ 2 ↑	Name ✖ 1 ↑	Permalink	Superuser status	Staff status
□	✓	kelly.mark.76@gmail.com	Mark J Kelly	¶	✓	✓

Fig. 4.3 Django-Admin example

I do have experience with the Flask framework as I used it on my 3rd year project. I have not yet used the Django framework, so there will be a learning curve involved. But taking everything into consideration, I believe the time invested into learning Django will be beneficial to both myself and this project, as I am looking forward to getting some experience with Django. So it’s decided, the Django framework will be the framework of choice for this project.

4.1.3 Database Technology

MySQL is a popular open source database, it enables the cost-effective delivery of reliable, high-performance and scalable Web-based and embedded database applications. It is an integrated transaction safe, ACID-compliant database with full commit, rollback, crash recovery, and row-level locking capabilities. MySQL delivers the ease of use, scalability, and high performance, as well as a full suit of database drivers and visual tools to help developers and DBAs build and manage their business-critical MySQL applications. MySQL is developed, distributed, and supported by Oracle [14]. MySQL is ideal for both small and large applications, its very fast, reliable, and easy to use, and there is a free version too. MySQL will be the database technology of choice for this project.

4.1.4 Email Client

SendGrid is a cloud-based SMTP provider that allows you to send email without having to maintain email servers. SendGrid manages all of the technical details, from scaling the infrastructure to ISP outreach and reputation monitoring to whitelist services and real time analytics [15].

I have experience with SendGrid as I used it in my third year project, and I know that SendGrid allows you to send email using HTTP(S) requests, and their API endpoint is on Pythonanywhere whitelist.

4.2 Frontend

4.2.1 HTML, CSS and JavaScript

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. Along with Cascading Style Sheets (CSS) and JavaScript, it forms a trio of foundation technologies for the World Wide Web [16]. These three technologies will be used as the foundation for the front end web interface for this project.

4.2.2 Bootstrap

Bootstrap is an open source toolkit for developing with HTML, CSS, and JavaScript. It allows you to quickly prototype your ideas or build your app with Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery [17].

With most students using smartphones to access web pages on the internet, it is essential that we include smaller resolution devices such as smartphones or tablets when creating our web interface for voters. We will use Bootstrap to ensure a responsive design to accommodate all our users on any screen resolution.

4.3 Boilerplate

In software development, a boilerplate is a unit of writing that can be reused over and over without change. By extension, the idea is sometimes applied to reusable programming as in "boilerplate code." A boilerplate can be compared to a certain kind of template, which can be thought of as a fill-in-the-blanks boilerplate.

It's a good idea to start a project with some kind of boilerplate as most web applications have similar core functionality. AWESOME DJANGO [18] is a curated list of awesome Django apps, projects and resources, and comes with links to multiple boilerplates and tons of other resources that will help you kickstart your project. I cannot recommend AWESOME DJANGO enough!, It’s absolutely packed with Django resources to help you develop your project with speed.

The boilerplate of choice for this project, is Django Edge v2 [19]. Edge is cutting-edge Django project skeleton. It has loads of features to help you get your project up and running with speed. It uses Bootstrap to give a modern feel. Below is some of the features that Django Edge v2 comes bundled with;

- Python 3.4 (or higher) support - enjoy cool Python 3 idioms
- Django 2.0 support - e.g. no south dependency, Jinja2 support
- Pipenv - use the official Python packaging tool
- Sign-ups and Log-ins - Basic user registration, log-ins, forgot password etc. styled with crispy forms

- User Profiles - extendible user profile with great defaults like profile picture
- Bootstrap bundled - Themed Bootstrap 3 based home page and admin out of the box
- Clean start - Based on Django 1.8 project structure
- Secrets Secure - Picks SECRET_KEY from environment as a best practice
- Minimal dependencies - Only essential packages listed in requirements.txt not all recommended ones

4.4 Django-elect

django-elect is a simple voting app intended for small-scale elections within academic organizations. It relies on Django's authentication system for verifying voter identity [20]. Django-elect has the following features;

- Highly customizable and easy-to-use admin interface.
- Supports both preferential (using the Borda count method) and plurality ballots with optional secrecy.
- Supports write-in candidates.
- Customizable biographies page with biographical information on each candidate.
- Statistics page with up-to-date election results.
- Can generate Excel spreadsheets with complete election data.

With a bit of surgery to the code, django-elect will be a great starting point for my online voting system.

4.5 Blockchain Technology

What is Blockchain?

> *“The blockchain is an incorruptible digital ledger of economic transactions*
> *that can be programmed to record not just financial transactions but*
> *virtually everything of value.”*
> -- Don & Alex Tapscott, authors Blockchain Revolution (2016) [21]

Currently, most people use a trusted middleman such as a bank to make a transaction. But blockchain allows consumers and suppliers to connect directly, removing the need for a middleman. Blockchain is peer to peer software technology that protects the integrity of a digital piece of information. The important thing here is it’s not centralised, as shown in, Fig.4.2, which means there is no single point of failure. Blockchain is a way to store and record transactions. It’s very much like a traditional database, but the blocks are immutable and linked together cryptographically, in order to make sure they are tamper proof. Each block contains a timestamp, data and a hashed link to the previous block, thus making it extremely hard to tamper with. It was invented in 2008 to create a digital currency, Bitcoin [22]. Blockchain technology can also be used for other digital currencies, digital contracts, online voting systems and many other applications [23].

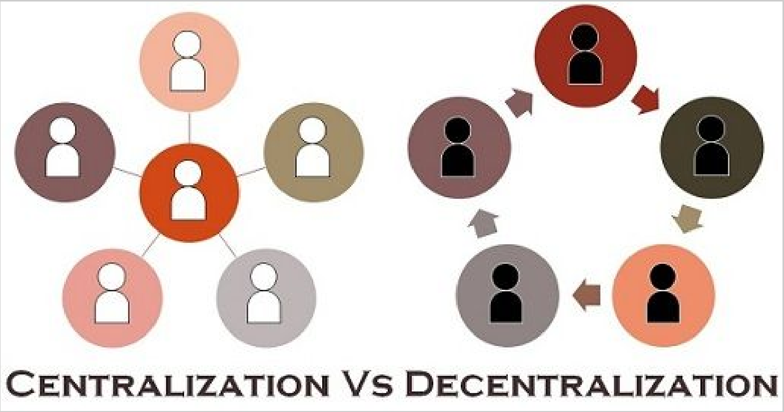


Fig. 4.2 Centralised Vs Decentralised [24]

How does Blockchain work?

Blockchain stores information permanently across a network of personal computers. This not only decentralises the information, but distributes it too. The more peers on the network, the harder it is to take down or corrupt the network. The many people who run the system use their own personal computers to hold all the records submitted by others. The records are known as blocks. Each block contains a timestamp and a link to the previous block, forming a chronological chain. You can view and add information to the block, but you can't change the information that is already there, making the block immutable. Blockchain is the underpinning technology used in Bitcoin.The blockchain stores the details of each transaction of the digital currency, and the technology prevents the same Bitcoin being spent more than once. Fig.4.3. Below, shows how a Blockchain works [25].

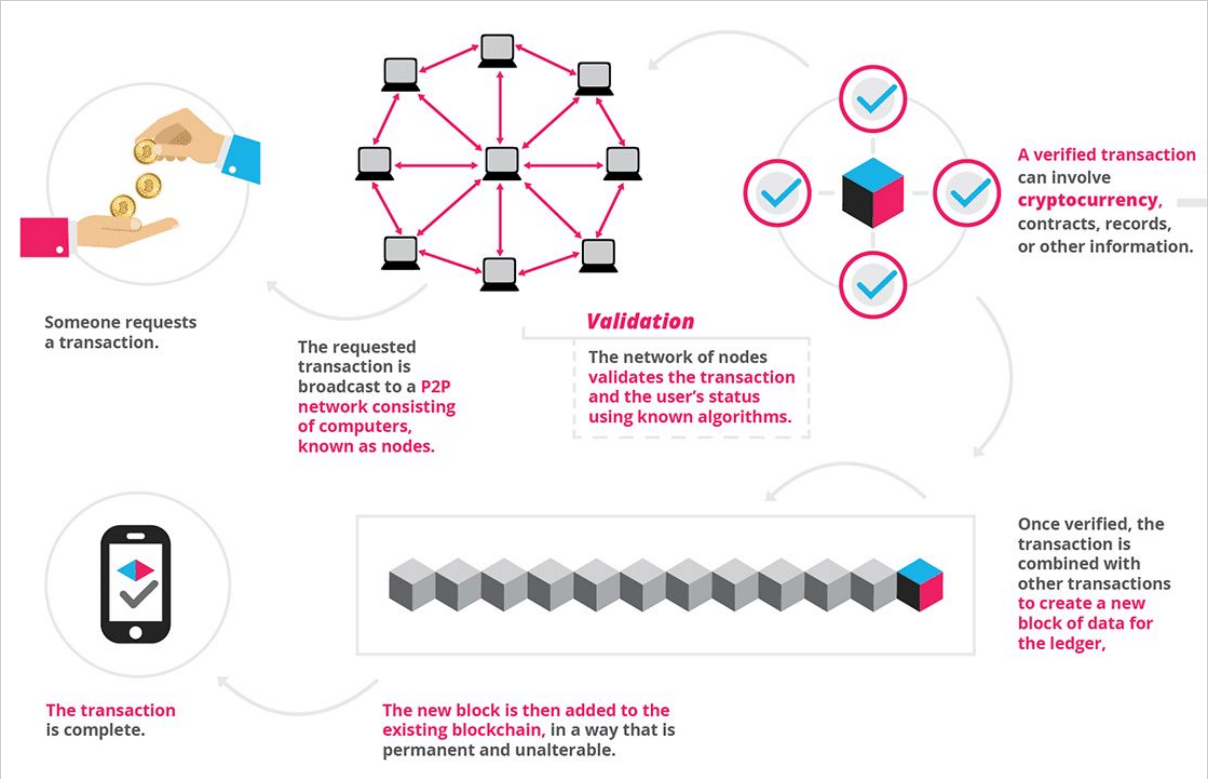

Someone requests a transaction.
The requested transaction is broadcast to a P2P network consisting of computers, known as nodes.
Validation
The network of nodes validates the transaction and the user's status using known algorithms.
A verified transaction can involve cryptocurrency, contracts, records, or other information.
Once verified, the transaction is combined with other transactions to create a new block of data for the ledger,
The new block is then added to the existing blockchain, in a way that is permanent and unalterable.
The transaction is complete.

Fig.4.3 How a Blockchain works [25]

Why use Blockchain for online voting?
Since the dawn of democracy, elections throughout the world have been plagued with accusations of illegitimacy. As many democratic societies and organizations across the world are adopting technology to improve the efficiency of their election process. It has been discovered that certain types of technology can be extremely vulnerable, which may have the potential to unfairly influence the outcome of elections.

Using blockchain technology it is possible to gain transparency into our elections, without compromising voter privacy, and have a way to mathematically prove that the elections results are accurate. Also, at the voter's request, there would even be a way to allow a voter to cast their vote online in an election and follow their vote into the ballot box to ensure that their vote was safely and securely stored without being changed or altered in any way [26].

5. Deployment

This section will consider deployment options and make a conclusion on which deployment option will be the best.

5.1 PythonAnywhere

PythonAnywhere is an online integrated development environment (IDE) and web hosting service based on the Python programming language. It provides in-browser access to server-based Python and Bash command-line interfaces, along with a code editor with syntax highlighting. Program files can be transferred to and from the service using the user's browser. Web applications hosted by the service can be written using any WSGI-based application framework [27].

5.2 Heroku

Heroku is a cloud platform as a service (PaaS) supporting several programming languages that is used as a web application deployment model. Heroku, supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it lets the developer build, run and scale applications in a similar manner across all the languages [28].

5.3 Conclusion

I have deployed a few apps to Heroku, I love how it integrates with GitHub, allowing you to easily deploy web apps. But for this project we shall deploy to pythonanywhere, as it has a nice feature that allows you to set up Scheduled tasks, which is a script that will run every day at a time of your choosing. It can to do stuff like scraping websites, or checking that your server is running. This feature will be very handy for us to check the database for inactive accounts, and also for checking if there is an election on that day, so reminder emails can be sent to registered voters.

6. Conclusion

For this project we shall create a web app, written in Python and based on the Django framework. Bootstrap will be used to create a responsive web design. MySQL will be used for the database. SendGrid will be used for sending emails. The web app will be deployed to pythonanywhere. The hope is to try to store the results of the elections on the Blockchain when I have the web app running, time permitting.

7. Bibliography

[1] BrainyQuote. 2017. It is enough that the people know there was an election. The people who cast the votes decide nothing. The people who count the votes decide everything. - Joseph Stalin - BrainyQuote. [ONLINE] Available at: https://www.brainyquote.com/quotes/quotes/j/josephstal109571.html. [Accessed 31 October 2017].

[2] Ncslorg. 2017. Ncslorg. [ONLINE] Available at: http://www.ncsl.org/research/elections-and-campaigns/voting-system-standards-testing-and-certification.aspx. [Accessed 1 November 2017].

[3] twitter.com. 2017. No page title. [ONLINE] Available at: https://twitter.com/electionrunner. [Accessed 22 October 2017].

[4] Build a Secure Online Election for Free | Election Runner. 2017. Build a Secure Online Election for Free | Election Runner. [ONLINE] Available at: https://electionrunner.com/. [Accessed 25 October 2017].

[5] electionbuddy - powerful and secure online election software . 2017. electionbuddy - powerful and secure online election software . [ONLINE] Available at: https://electionbuddy.com/. [Accessed 25 October 2017].

[6] PHP: What is PHP? - Manual . 2017. PHP: What is PHP? - Manual . [ONLINE] Available at: http://php.net/manual/en/intro-whatis.php. [Accessed 01 November 2017].

[7] Python.org. 2017. Welcome to Python.org. [ONLINE] Available at: https://www.python.org/. [Accessed 01 November 2017].

[8] Taylor Otwell. 2017. Laravel - The PHP Framework For Web Artisans. [ONLINE] Available at: https://laravel.com/. [Accessed 01 November 2017].

[9] SitePoint. 2017. The Best PHP Framework for 2015: SitePoint Survey Results — SitePoint. [ONLINE] Available at: https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/. [Accessed 01 November 2017].

[10] Welcome | Flask (A Python Microframework). 2017. Welcome | Flask (A Python Microframework). [ONLINE] Available at: http://flask.pocoo.org/. [Accessed 02 November 2017].

[11] The Web framework for perfectionists with deadlines | Django. 2017. The Web framework for perfectionists with deadlines | Django. [ONLINE] Available at: https://www.djangoproject.com/. [Accessed 03 November 2017].

[12] Matt Makai. 2017. Object-relational Mappers (ORMs) - Full Stack Python. [ONLINE] Available at: https://www.fullstackpython.com/object-relational-mappers-orms.html. [Accessed 04 November 2017].

[13] URL dispatcher | Django documentation | Django. 2017. URL dispatcher | Django documentation | Django. [ONLINE] Available at: https://docs.djangoproject.com/en/1.11/topics/http/urls/. [Accessed 04 November 2017].

[14] MySQL. 2018. MySQL. [ONLINE] Available at: https://www.mysql.com/. [Accessed 02 April 2018].

[15] Sendgrid User Guide - Sendgrid Documentation | Sendgrid. Sendgrid.com. N.p., 2017.[ONLINE] Available at: https://sendgrid.com/docs/User_Guide/index.html [Accessed 10 November 2017].

[16] HTML - Wikipedia. 2018. HTML - Wikipedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/HTML. [Accessed 01 April 2018].

[17] Mark Otto, Jacob Thornton, and Bootstrap contributors. 2018. Bootstrap · The most popular HTML, CSS, and JS library in the world.. [ONLINE] Available at: https://getbootstrap.com/. [Accessed 10 April 2018].

[18] Awesome Django. 2017. Awesome Django. [ONLINE] Available at: http://awesome-django.com/. [Accessed 10 November 2017].

[19] Django Edge Documentation. 2017. Django Edge Documentation. [ONLINE] Available at: https://django-edge.readthedocs.io/en/latest/. [Accessed 10 November 2017].

[20] Bitbucket. 2017. hssonline / django-elect — Bitbucket . [ONLINE] Available at: https://bitbucket.org/hssonline/django-elect. [Accessed 11 November 2017].

[21] Blockgeeks. 2017. What is Blockchain Technology? A Step-by-Step Guide For Beginners. [ONLINE] Available at: https://blockgeeks.com/guides/what-is-blockchain-technology/. [Accessed 02 November 2017].

[22] Blockchain - Wikipedia. 2017. Blockchain - Wikipedia. [ONLINE] Available at: https://en.wikipedia.org/wiki/Blockchain. [Accessed 02 November 2017].

[23] Decentralized Blockchain Technology and the Rise of Lex Cryptographia by Aaron Wright, Primavera De Filippi :: SSRN. 2017. Decentralized Blockchain Technology and the Rise of Lex Cryptographia by Aaron Wright, Primavera De Filippi :: SSRN. [ONLINE] Available at: http://bit.ly/2hgosA6. [Accessed 02 November 2017].

[24] Key Differences. 2017. Difference Between Centralization and Decentralization (with Comparison Chart) - Key Differences. [ONLINE] Available at: http://keydifferences.com/difference-between-centralization-and-decentralization.html. [Accessed 02 November 2017].

[25] Blockgeeks. 2017. What is Blockchain Technology? A Step-by-Step Guide For Beginners. [ONLINE] Available at: https://blockgeeks.com/guides/what-is-blockchain-technology/. [Accessed 02 November 2017].

[26] The Balance. 2018. How the Blockchain Can Change How We Vote. [ONLINE] Available at: https://www.thebalance.com/how-the-blockchain-will-change-how-we-vote-4012008. [Accessed 10 March 2018].

[27] PythonAnywhere LLP. 2018. Host, run, and code Python in the cloud: PythonAnywhere. [ONLINE] Available at: https://www.pythonanywhere.com/. [Accessed 03 April 2018].

[28] Cloud Application Platform | Heroku. 2018. Cloud Application Platform | Heroku. [ONLINE] Available at: https://www.heroku.com/. [Accessed 03 April 2018].