Project:

WebApp to conduct Elections

Languages and Frameworks used:

- python 3
- Flask
- HTML5 & CSS
- sqlite3(database)

Visual Walkthrough of my project:

https://youtu.be/R9_H8i2uNOg

Experience while working on it:

This idea came to me thinking about how my first experience of elections would be. Standing in line, waiting, was the first thing that came to my mind. So, I decided to work on this. Later, I researched and found that some countries already do conduct e-voting. I made this project in flask as I was pretty confident about it (having made my python project in it). Though, this time I was able to make it better, efficient and with more functionality.I started from scratch by first organising my ideas on a paper. This made it easy for me to plan the routes as I was able to visualise my project. Initially, it was just a raw webpage with a vote counter after login. But then after talking to others and asking them what they would like a website for elections to have I put in more pages, options for voters and login, and information about the candidates as well.

Scope Of Using The Website For Nationwide Elections:

This website can be used for conducting sac – elections in our college. Currently, countries such as China, Estonia and Uganda conduct e-voting. Keeping the advantages aside, protecting the website from hacking is a major barrier for evolution into e-voting.

Implementation Of Idea And Procedure

As mentioned ahead in the “errors faced” section I have two different logins for my project.
A candidate has the following login details:

USERNAME	PASSWORD	KEY
Chloe Marshall	chloe	1234
Olivia Ambrosia	olivia	4321
James Howard	james	6789
Luke Chambers	luke	9876

For registering a voter the required **secretid** is **1011.**
If a test case is required for testing the one-vote-only case try:
username : IMT2018004 password: IMT2018004

Method for vote counting:

1. There is a separate database maintained for counting the votes. When a new voter logins for voting, the option selected by the user is used to call the table corresponding to that option in database. For checking the value I used the value tag for a radio-button of HTML. A button in recognised by its value as either “one”, “two” ,”three” or “four”. The routes module then checks the radiobutton clicked and correspondingly calls the table from votes.db. It then updates the value by doing a “+1” to the value initially stored.

2. How to ensure that a specific user logins and votes only once? So when a person submits the option another databse is called (voters.db). When the person votes, the table named votedid gets a new updated value “1011”. Then when the user clicks on the button “vote” a check is made in the database for existance of this value. If true, the user is redirected to another page.

Method for viewing different pages for candidate and voter:

1. Since I dont have separate login manager for the user I had to figure out a different method for separating actions for candidates and voters. In my __init__.py you will notice that there is a global.db database. This database is the reason why I am able to display different pages to candidate and voter and how I am able to prevent a voter from accessing certain tabs. And the initialization for it is in the run.py.

Now, in my routes module I am changing the value of the table named key to different values depending upon who logins. So under the candidatelogin route the value of the key is made equal to userid and in voterlogin key is made is made equal to 0. Now, for example in viewaboutcandidate the voter is denied entrance by checking the value of world.key. If it is 0 I am redirecting the person to the home page with a flash. The same thing has also been for votes. If a candiate tries to click on vote he is shown a message different from voter and not allowed to vote. (These things can be seen in my walkthrough).

Making graph:

For doing that, I used matplotlib. I am using the info from the votes database to create a pie chart. The piechart automatically gives percentage share for eaach coordinate passed. So, to calculate the exact number of votes per person I am using the function make_autopct() to give me that.

Other features:

For candidates:

A candidate can view how his/her information will be shown to the voters. Also, he/she can update that info (which includes a little about themselves or election propaganda, and links to their accounts on facebook and github). Though they don't have the facility to vote, they can see the stats of the elections on the result page.

For voters:

Other than the function to vote they can view the info about the candidates(in the about page, not the info tab) and the result of the elections.

Errors faced:

1. There are two different categories of users trying to login. But in my login manager in models.py I am returning user with id relative to User data and not Candidate data. Because of this, I tried adding more functionality to my webpage such as a different tab when a candidate logins but am unable to do that.
So, currently a problem will arise when the number of candidates becomes more than the number of voters because the database of Users won't have enough ids to match the database of candidates.

I tried to tackle the problem in two ways:
First: I tried creating a global variable which stores the value of the user(object) when logged in. But since the script runs everytime, the variable takes the default value previously given. So that couldn’t solve it.
Second: Now, in the above method the issue was that the value during logging in wasn’t being stored. So I thought of connecting it to another database. But a class object can’t be stored so that the attributes be utilised(we can store it as a string but then the attributes can’t be accessed).
So, to solve that I created different columns for different attributes. That way the problem was solved. Now, my webapp has different pages for different type of logged in user.

Sources:

- w3schools for css
- matplotlib documentation
- reddit for a discussion on making graph