Since the advent of the internet, few political entities have relied on online ballots for elections and referenda. You may want to consider why this is the case, and why this hasn’t changed yet. The answer to the second question is this assignment. In line with our philosophy of encouraging you to vote if possible, you will now create the backend API (application programming interface) for a secure online voting system.

This project is due on November 3, 2020 at 12am, which is Election Day in the United States. Starting early is the key to testing your project to perfection. Because the deadline is very close to the beginning of voting on that day, we will not accept any late submissions.

1 Your Assignment

You are to create the backend for an online voting system that includes voter registration, vote selection, and vote tallying. Your voting system should support multiple methods of voting. See the next section for information on different election types.

Unlike previous assignments, you are free to implement this assignment however you wish. There is no starter code provided for the backend, nor are there any interfaces. However, you must clearly document your API and link it to our basic frontend. (You can also create your own frontend for karma.)

You should also create security features for your system that prevent various security breaches from occurring. This includes preventing fraudulent/unlawful voter registration, repeat voting, and system penetration. (Read up on penetration testing, and feel free to request a classmate to pentest your implementation as long as they don’t see your code.)

2 Details

There are many different details to take into account when creating this system.

Voter registration. Your program should allow for same-day voter registration. There is no reason this should not be allowed in an online voter registration system.

Ballot creation. Before any voting can occur, ballots must be created. The election administrator should be able to log in to the ballot creator and add as many kinds of ballots as they wish, and as many different races onto each ballot as they wish. For instance, one precinct’s ballot may contain different races from another precinct’s. They should also be able to choose the electoral system they need to use. Examples of electoral systems are first-past-the-post (most elections in the United States use this) and single transferrable vote. You do not need to consider whether the government follows the presidential system or the Westminster system, as it is not your job to decide that. You will need to implement first-past-the-post and single transferrable vote at the very least; any other systems you may wish to implement can be done so for karma, and are available by searching the internet using your Web Crawler.

Vote selection. This is where the actual voting happens. Depending on the electoral system selected by the election administrator, which precinct they are in/which ballot they should receive, etc., your voter should see a specific screen with specific races and the ability to choose either one option (in the case of first-past-the-post) or rank options (in the case of single transferrable vote). More ways to vote should be accommodated depending on what kind of other election systems you might have.

Grading for this assignment. To pass this assignment, your flawed implementation must be good enough to be used in an actual election. Only the best implementation will be used by the Texas Secretary of State. Therefore, everyone else will fail this assignment. Remember, we learn by failing.

Also, if the flawed implementation ends up failing on Election Day and causes a redo election to be ran, then that implementation’s creator will also fail this assignment, and possibly face jail time.
3  Karma

As mentioned above, there are several opportunities for karma.

One way to earn karma is to create your own frontend. Be sure to justify the changes in functionality that it has between our frontend, or else if you create a worse frontend, we will be sure to deduct points/give you negative karma for not appreciating our already great system.

Another way to earn karma is to implement an election system that is not first-past-the-post or single transferrable vote. This can be one used by a political entity, created by political scientists, one you’ve created, or anything else. (Do not attempt to create a North Korean-style election system, as rubber stamp parliaments are as useless as bogosort and only speaks volumes about your lack of work ethic. Not heeding this warning will result in negative karma.)

4  Submission

Since your project’s submission will be comprised of several files, all files will need to be contained in a ZIP file called prog10.zip. Your submission is due on **November 3, 2020 at 12am** via email to the State of Texas Secretary of State’s Office of Electoral Projects: oep@sos.texas.gov.

As always, you will need to write a report and turn it in with your submission. Follow the format provided on the previous assignments, and make adjustments for this assignment as you believe are appropriate. Your report must also include documentation for your API. If your documentation is not clear enough for your TA to read, they reserve the right to fail you on the assignment completely. (Not that you would pass anyway, since only one person will pass this assignment.)

5  Acknowledgements

This assignment was created by Vladimir Putin of Mother Russia in conjunction with Mark Zuckerberg of Facebook and Donald Trump. Special thanks to the Texas Secretary of State for agreeing to pilot this system.