

Digital Voting System With Dual Factor Security

Taylor Watson
Graduate Student, Computer Science
Vanderbilt University
Nashville, TN, USA

Abstract—My term project idea for this class is a digital voting system. Currently, voters have to go to the polling booth in person which takes time and can decrease the effective representation of the population as people choose not to or are unable to take the time out of their day to wait in long lines. Digital voting could help to solve this growing problem and advancements in cybersecurity could allow for such a system to exist without widespread fraud. The scope of this project is to develop a prototype for such a system. This prototype will be delivered in the form of a local voting system monitored by an Admin account. This system will utilize TCP protocol to allow multiple voters to log into the system and cast their votes. At login, each voter will be prompted to scan a QR code with their Google Authenticator app that will give them their one-time password (OTP) unique to each login attempt.

I. ARCHITECTURE

This prototype is made up of the following components:

A. Main

This is where the main Tk logic is held for launching the application.

B. Admin

This is where all logic related to the admin (or features that require admin privileges) can be found. It houses the Tk logic for the login screen, admin homepage, voter registration, and results page.

C. Voter

This is where all logic related to the voter can be found. It houses the Tk logic for the voter login and cast vote screens. Additionally, it contains the functions needed to interface with the server and record a user's vote.

D. Security

This is where all logic related to voter security can be found. It houses the logic for verifying a user's identity via two-factor authentication, as well as ensuring that the voter has not voted previously.

E. Server

This is where all logic related to the server can be found. This script was made using socket programming and TCP.

II. TOOLS USED

This prototype was developed using python as its primary language. One of the benefits of python is an extensive library of tools that can be used for each project. This section will discuss some of the tools used in this project.

A. Python Tkinter

Tkinter is the primary way to create Graphical User interfaces (GUIs) in python and is included in all standard distributions. This Python framework provides an interface to the Tk toolkit and works as a thin object-oriented layer on top of Tk. The Tk toolkit is a cross-platform collection of graphical control elements for building application interfaces.

B. The Python One-Time Password Library

PyOTP is a Python library for generating and verifying one-time passwords. It can be used to implement two-factor or multi-factor authentication methods in web applications and in other systems that require users to log in. PyOTP implements server-side support for both of these standards. Client-side support can be enabled by sending authentication codes to users over SMS or email or by instructing users to use Google Authenticator or another compatible app. Users can set up auth tokens in their apps easily by using their phone camera to scan QR codes provided by PyOTP.

C. Google Authenticator

Google Authenticator is a mobile security application based on two-factor authentication that helps to verify user identities before granting them access to websites and services.

Two-factor authentication makes it less likely that an intruder can disguise themselves as authorized users. Authentication factors are categories of credentials used to verify that someone is who they say they are. There are three categories: things you know (knowledge) such as a password or PIN, things you have (possession) such as a badge or smartphone, and things you are (inherence) such as a biometric like fingerprints or voice recognition.

Google Authenticator works for any site or service that has enabled two-factor authentication. Like most web-based two-factor authentication applications, the system combines knowledge and possession features. To access websites or web-based services, the user types in their normal username and password as well as a one-time passcode (OTP) that was delivered to their device. That combination verifies that the same person attempting to log in is in possession of the device to which the Google Authenticator app was downloaded. While passwords may be easy to crack or otherwise steal it is unlikely that a hacker will also have access to the user's physical device.

The Authenticator app is based on the time-based one-time password (TOTP) system. The TOTP algorithm generates a

six-digit passcode that factors in the current time of day to ensure that each passcode is unique.

III. CAPABILITIES

When launching the application you'll be faced with the initial home screen seen in Fig 1.

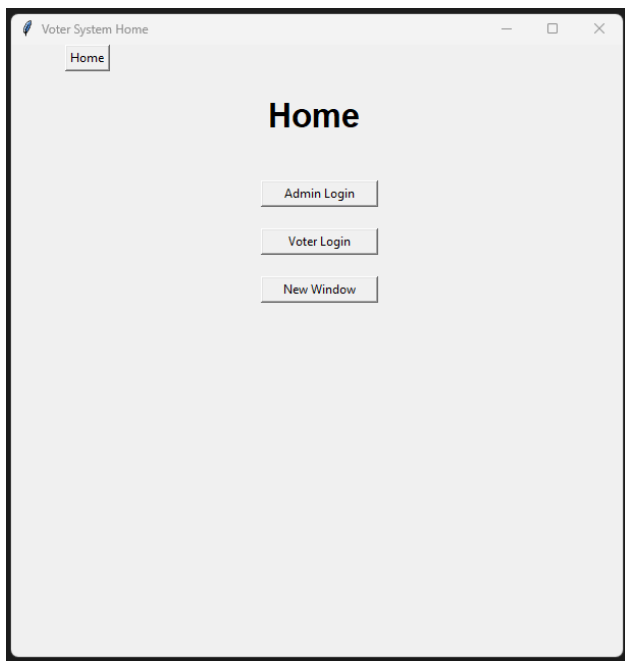


Fig. 1. Digital Voting System Home Page

In order to begin the voting process an admin will need to launch the server. This can be done by navigating to admin login, Fig 2, and clicking "start server" from the admin home page in Fig 3.

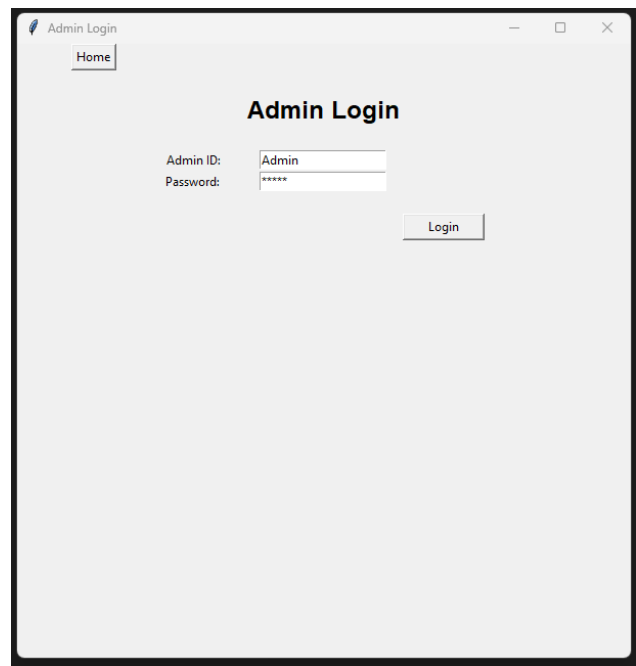


Fig. 2. Digital Voting System Admin Login Screen

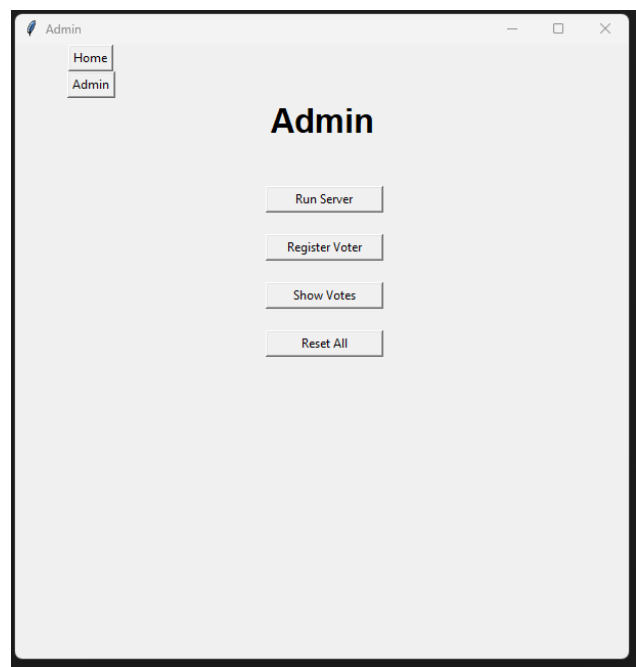
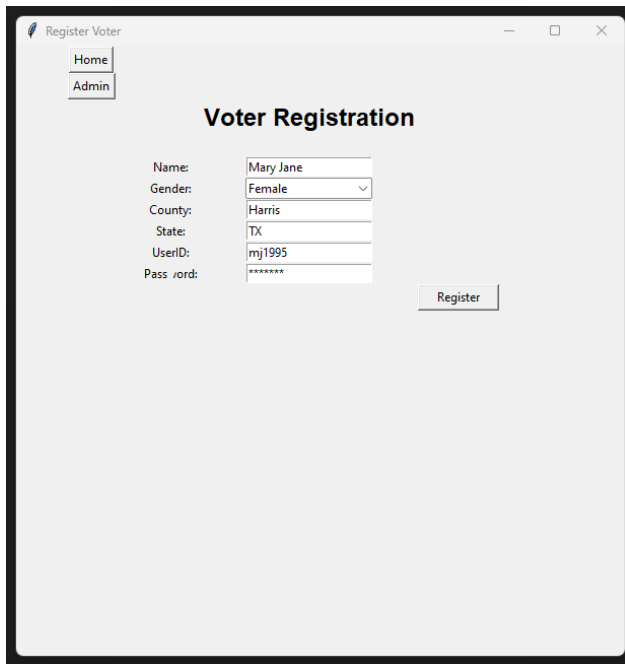


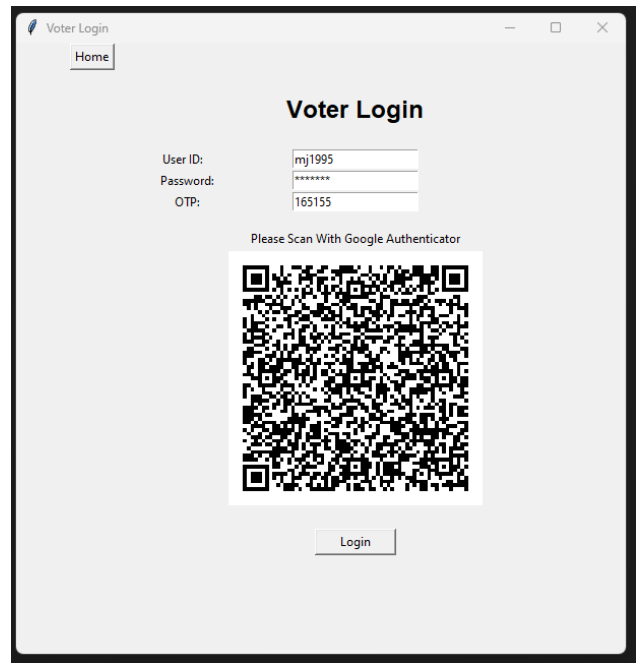
Fig. 3. Digital Voting System Admin Home Screen

Additionally, the admin is in charge of registering voters. Each voter will need to provide the following information: name, gender, county, state, userID, and password. The form for this can be seen in Fig 4. Once registered they will be given a voter registration ID that the system will refer to them as, Fig 5.



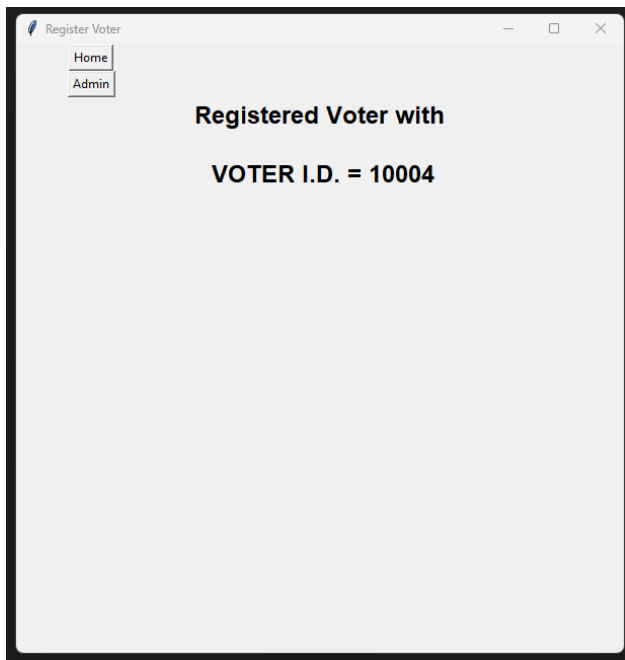
The screenshot shows a web browser window titled "Register Voter". It has a navigation bar with "Home" and "Admin" links. The main heading is "Voter Registration". Below it, there are input fields for "Name" (filled with "Mary Jane"), "Gender" (a dropdown menu showing "Female"), "County" (filled with "Harris"), "State" (filled with "TX"), "UserID" (filled with "mj1995"), and "Password" (filled with "*****"). A "Register" button is located at the bottom right of the form.

Fig. 4. Digital Voting System Voter Registration Screen



The screenshot shows a web browser window titled "Voter Login". It has a navigation bar with "Home" and "Admin" links. The main heading is "Voter Login". Below it, there are input fields for "User ID:" (filled with "mj1995"), "Password:" (filled with "*****"), and "OTP:" (filled with "165155"). Below these fields, there is a text prompt "Please Scan With Google Authenticator" and a large QR code. A "Login" button is located at the bottom right of the form.

Fig. 6. Digital Voting System Voter Login Screen

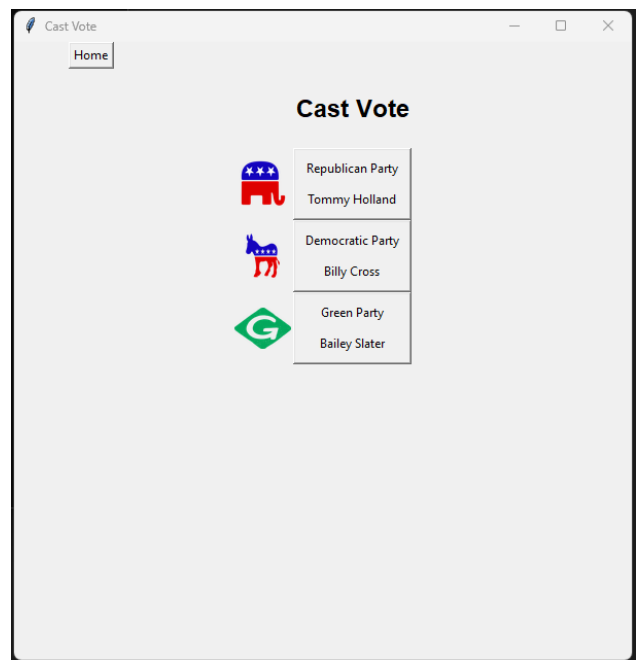


The screenshot shows a web browser window titled "Registered Voter". It has a navigation bar with "Home" and "Admin" links. The main heading is "Registered Voter with VOTER I.D. = 10004".

Fig. 5. Digital Voting System Voter Successfully Registered Screen

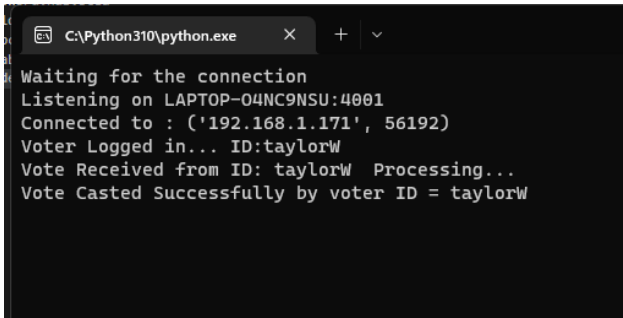
Once the server has been initialized registered voters will be able to log in and cast their votes. The login screen, as seen in Fig 6, will prompt the user with a QR code to be scanned with their Google Authenticator to receive an OTP to accompany their username and password.

Once the voter's identity has been verified, they will be prompted with the voting screen in Fig 7. The server will also keep track of logins and cast votes seen in Fig 8.



The screenshot shows a web browser window titled "Cast Vote". It has a navigation bar with "Home" and "Admin" links. The main heading is "Cast Vote". Below it, there are three party options, each with a logo and a name: "Republican Party" with a red elephant logo and "Tommy Holland", "Democratic Party" with a blue donkey logo and "Billy Cross", and "Green Party" with a green circle logo and "Bailey Slater".

Fig. 7. Digital Voting System Cast Vote Screen



```
C:\Python310\python.exe x + v
Waiting for the connection
Listening on LAPTOP-04NC9NSU:4001
Connected to : ('192.168.1.171', 56192)
Voter Logged in... ID:taylorW
Vote Received from ID: taylorW Processing...
Vote Casted Successfully by voter ID = taylorW
```

Fig. 8. Digital Voting System Server

Throughout the election process, admins will be able to check the results as seen in Fig 9. The election process will end when the server is closed. Voters are unable to log into the system once this is done.

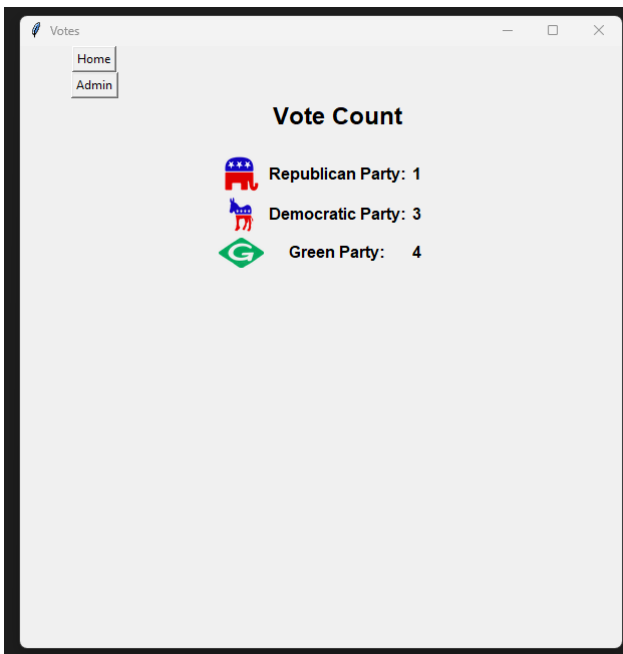


Fig. 9. Digital Voting System Current Results Page

IV. FUTURE WORK

To ensure that I was able to complete the prototype within the allotted time I had to limit the scope of my project. If I were to continue working on this there are a number of features I would like to explore. The first of these, and the most obvious, would be to improve the appearance of the tool. I believe that the voting system would benefit greatly from a new UX/UI. Additionally, I would like to work on making the voting tool a web application. This would allow for the system to be used more widely. I would also like to explore using a more sophisticated data storage system like SQL for both the voter's and candidate's information.

V. ACKNOWLEDGMENT

This work was conducted as part of a graduate-level course on Computer Networks.

REFERENCES

- [1] PyOTP - The Python One-Time Password Library, <https://pyotp.readthedocs.io/en/stable/>.
- [2] Get verification codes with Google Authenticator, <https://support.google.com/accounts/answer/1066447?hl=en&co=GENIE.Platform%3DAndroid>.
- [3] Python interface to Tcl/Tk, <https://docs.python.org/3/library/tkinter.html>.