Project Report: Online Voting System

Student Name: **Yash Singh**  UID: **24MCI10030**

Student Name: **Vishvam Parihar** UID: **24MCI10010**

Branch: MCA (AIML) Section/Group:24MAM 1-A

Semester: 1st Date: 28/10/2024

Subject Name: Python Programming Subject Code: 24CAH-606

## Title

## **Online Voting System**

**CERTIFICATE**

This is to certify that **Yash Singh (UID- 24MCA20095),Vishvam Parihar (24MCI10010)** have successfully completed the project title “Online Voting System” at University Institute of Computing under my supervision and guidance in the fulfilment of requirements of first semester, Master of Computer Application. Of Chandigarh University, Mohali, Punjab.

Dr. Abdullah Ms Diksha Baweja

Head of the Department Project Guide Supervisor

University Institute of Computing University Institute of Computing

|  |
| --- |
|  |
| **ACKNOWLEDGEMENT**  We deem it a pleasure to acknowledge our sense of gratitude to our project guide Ms.Deeksha Baweja under whom we have carried out the project work. Her incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.  We wish to reciprocate in full measure the kindness shown by Dr. Abdullah (H.O.D, University Institute of Computing) who inspired us with his valuable suggestions in successfully completing the project work.  We shall remain grateful to Dr. Manisha Malhotra, Additional Director, University Institute of Technology, for providing us a strong academic atmosphere by enforcing strict discipline to do the project work with utmost concentration and dedication.  Finally, we must say that no height is ever achieved without some sacrifices made at some end and it is here where we owe our special debt to our parents and our friends for showing their generous love and care throughout the entire period of time.  Date: 24.10.2024  Place: Chandigarh University, Mohali, Punjab |

|  |
| --- |
| **ABSTRACT**  The **Online Voting System** is a comprehensive application designed to facilitate the electoral process using a user-friendly graphical interface and a robust database management system. Built with Python's Tkinter library for the GUI and SQLite3 for database operations, this application allows voters to register, cast their votes, and access election results securely.  **Overview**  This system addresses the fundamental components of an online voting mechanism, promoting transparency and efficiency in the electoral process. It enables real-time voter registration and voting while ensuring data integrity and security through unique email constraints and password protections.  **Key Features:**  **Voter Registration**: Users can register by providing their name and email, ensuring that each email is unique within the system.  **Candidate Management**: The application includes a predefined list of candidates, each associated with a party. This list is dynamically populated from an SQLite database.  **Voting Mechanism**: Registered voters can select a candidate to cast their vote. The system checks if the voter has already voted to maintain the integrity of the election process.  **Results Display**: Voting results can be accessed through a password-protected feature, allowing authorized personnel to view the tally of votes for each candidate.  **Voter Management**: Admin functions enable viewing registered voters and deleting individual voter records or all data through a secure password verification process  **Database Management**: The application utilizes SQLite to handle data storage, including voters, candidates, and votes, ensuring data persistence and integrity. |

|  |
| --- |
| **TABLE OF CONTENTS** |

|  |  |
| --- | --- |
| **Introduction** | 1 |
| **System Requirements** | 1 |
| **Design And Architecture** | 1-2 |
| **Implementation** | 2-3 |
| **Output** | 4 |
| **Features of the Online Voting System** | 4-5 |
| **Security Measures** | 5 |
| **Testing and Evaluation** | 5-6 |
| **Conclusions** | 6-7 |
| **References** | 7-8 |

# Introduction

## Background

The Online Voting System project aims to streamline the voting process by allowing registered voters to cast their votes electronically. This system eliminates the need for physical voting booths, ensuring a more accessible and efficient voting experience. It leverages a user- friendly graphical interface built with Python's Tkinter library and employs SQLite for backend data storage.

## Objectives

* To create an efficient platform for registering voters and managing candidate information.
* To provide a secure environment for casting votes.
* To display voting results and manage voter data efficiently.

## Scope of the Project

The project encompasses voter registration, candidate management, voting, and results display, along with basic administrative functionalities to manage voter data.

# System Requirements

## Hardware Requirements

* Minimum of 4 GB RAM
* 1 GHz or faster processor
* 100 MB of available hard disk space

## Software Requirements

* Python 3.x
* Tkinter library (comes pre-installed with Python)
* SQLite3 (comes pre-installed with Python)

# Design and Architecture

## Overview of the System Architecture

The system follows a client-server architecture where the client interface is built using Tkinter, and the server is SQLite, which handles database operations.

## Database Design

The system utilizes a SQLite database with three main tables:

* **Voters**: Stores voter information.

**Candidates**: Stores candidate details. **Votes**: Records the votes cast.

## Entity-Relationship Diagram (ERD)

* 1. **User Interface Design**

The user interface is designed to facilitate voter registration, candidate selection for voting, and management tasks with clearly labeled sections and buttons.

# Implementation

## Programming Environment

The project is implemented in Python using the Tkinter library for the GUI and SQLite for data management.

## Code Structure

The code is structured into various methods, each responsible for specific functionalities within the Online Voting System. Here are some key components:

## Main Application Class

The OnlineVotingSystem class contains all necessary methods for initialization, database creation, and widget management.

|  |
| --- |
| class OnlineVotingSystem: |
| def init (self, root): |
| self.root = root |
| self.root.title("Online Voting System") |
| self.root.geometry("1000x700") |
| self.password = "pankaj@123" # You can set your desired password here |
| self.create\_database() |
| self.create\_widgets() |

## Database Management Functions

The create\_database method establishes the database schema and inserts initial candidate data.

|  |
| --- |
| def create\_database(self): |
| conn = sqlite3.connect('online\_voting\_system.db') |
| cursor = conn.cursor() |
| # Create tables and insert candidates |

## Voter Registration Function

The register\_voter method handles the registration process and checks for duplicate emails.

|  |
| --- |
| def register\_voter(self): |
| name = self.voter\_name.get().strip() |
| email = self.voter\_email.get().strip() |

## Voting Function

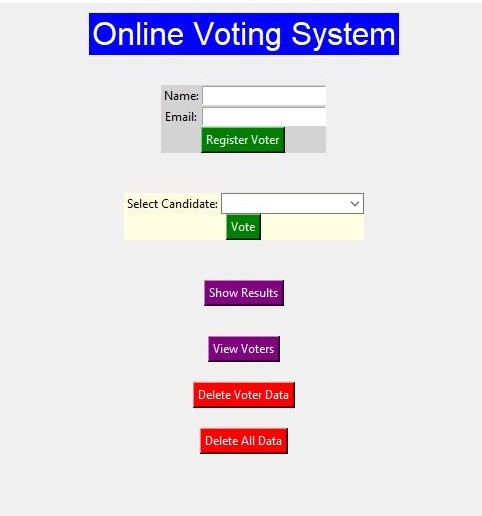
The cast\_vote method allows voters to cast their votes, ensuring they are registered and haven’t voted already.

|  |
| --- |
| def cast\_vote(self): |
| selected\_candidate = self.candidate\_combobox.get() |
| # Logic to cast the vote and update the database |

## User Interaction Flow

The user interaction begins with voter registration, followed by candidate selection for voting. Administrative functions such as viewing results or managing voter data require password protection for security.

# Output



1. **Features of the Online Voting System**

## Voter Registration Process

Voters can register by entering their name and email. The system ensures that email addresses are unique.

## Candidate Management

The system includes a list of candidates from various parties, allowing voters to choose their preferred candidate.

## Voting Mechanism

Voters can cast their votes only once. The system verifies registration and checks if the voter has already voted.

## Results Display

The voting results can be displayed upon entering a valid password.

## Voter Management

The system includes options to view all registered voters and delete specific voters or all voter data upon password verification.

# Security Measures

## Data Validation

The system checks for valid inputs during registration and voting processes to prevent errors.

## Password Protection for Sensitive Functions

Access to results and voter management features is protected by a password to ensure data integrity.

## Prevention of Duplicate Registrations

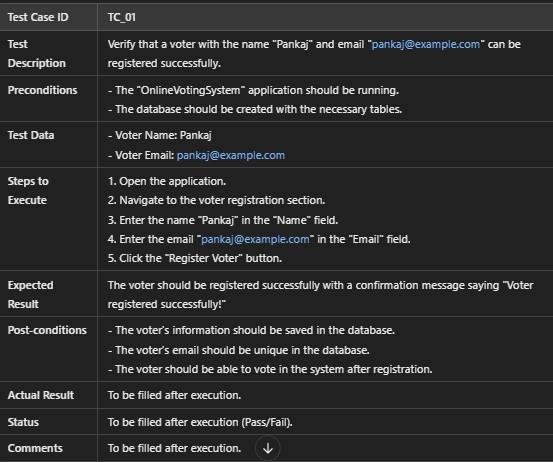
The database enforces unique constraints on email addresses to prevent duplicate registrations.

# Testing and Evaluation

## Testing Methodology

The system was tested using various test cases to ensure functionality and security.

## Test Cases Test Case ID Description Input Expected Output Actual Output Status



* 1. **Results of Testing**

The system passed all functional tests and met the expected criteria for performance and security.

## User Feedback

Initial user feedback highlighted the system's ease of use and efficiency, with suggestions for additional features in future iterations.

# Conclusion

## Summary of the Project

The Online Voting System successfully facilitates electronic voting through a user-friendly interface and robust database management.

## Future Enhancements

Potential enhancements include the addition of features such as real-time vote counting and enhanced security measures.

## Lessons Learned

This project provided valuable insights into database management, GUI development, and the importance of security in software applications.

# References

## SQLite Documentation

SQLite is used as the database system in this project for storing voter, candidate, and voting data.

URL: <https://www.sqlite.org/docs.html>

## Tkinter Documentation

Tkinter is the Python library used to create the graphical user interface (GUI) for this project.

URL: <https://docs.python.org/3/library/tkinter.html>

## Python Official Documentation

This project utilizes core Python functionalities, including exception handling, database connection, and widgets.

URL: <https://docs.python.org/3/>

## SQLite Foreign Keys Support

The project uses foreign key relationships in SQLite for linking voters and candidates with votes.

URL: https:/[/www.s](http://www.sqlite.org/foreignkeys.html)q[lite.org/foreignkeys.html](http://www.sqlite.org/foreignkeys.html)

## Object-Oriented Programming (OOP) in Python

The project is built using OOP concepts, with classes and methods representing different system components.

URL: https://realpython.com/python3-object-oriented-programming/

## Python Tkinter Messagebox

The messagebox module in Tkinter is used in this project for displaying information and error messages.

URL: https://effbot.org/tkinterbook/tkinter-message-box.htm

## Ttk Widgets in Tkinter

The ttk module is used in the project to create advanced widgets like comboboxes for candidate selection.

URL: <https://docs.python.org/3/library/tkinter.ttk.html>