Simple Voting System

Team 2

Members:

- Abhinand A S
- Shadin P K
- U Shivadha
- Diljith
- Akhil Das

Institution: NSTI Calicut

Date: 28-02-2025

Acknowledgment:

We would like to express our gratitude to our mentor Adwaith R S and our teammates for their unwavering support and guidance throughout the development of this project. We would also like to thank Institution for providing the resources and environment necessary to complete the project successfully.

Abstract:

This project presents a Simple Voting System developed using Python and MySQL for managing voter registration, vote casting, and tracking votes for candidates. The system allows eligible voters to register with their personal details and cast their votes for selected candidates. The project also ensures that each voter can only vote once and provides transparent results by displaying the total votes for each candidate. The database structure consists of three main tables: Voters, Candidates, and Votes, and the system is built on MySQL for data storage. This system is designed for small-scale elections or internal voting processes.

Table of Contents:

Introduction:

Overview of the Project:

The **Simple Voting System** is an interactive and secure platform that allows voters to register, cast votes, and track voting results. The system ensures that only eligible individuals (18 years or older) can register and vote, and each voter is allowed to vote only once.

Purpose and Importance:

This system aims to digitize voting processes, making them more efficient, secure, and transparent, with applications in small elections or organizational voting.

Problem Statement:

Traditional paper-based voting systems are prone to errors, fraud, and are difficult to manage. There is a need for a simple, secure, and transparent digital voting system.

Objectives:

- To create a secure registration process for voters.
- To allow voters to cast their votes for selected candidates.
- To ensure voters can vote only once.
- To display total votes for each candidate in real-time.

Background Study:

Existing systems have either lacked proper voter validation or allowed multiple voting by a single voter. Some existing solutions involve digital voting but are either complex or not user-friendly. This project fills these gaps by offering a simple, efficient, and secure voting system using Python and MySQL.

System Requirements:

Hardware Requirements

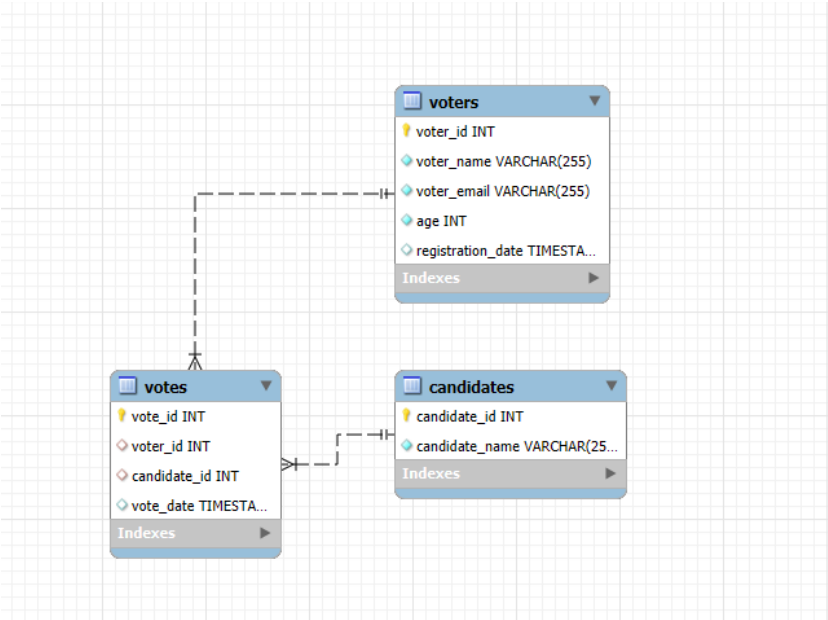
- A computer with internet access for installing MySQL and Python.
- A system with at least 2GB of RAM and 500MB of free disk space.

Software Requirements

- **Programming Language**: Python 3.x
- **Database**: MySQL 5.7 or higher
- **Libraries**: mysql.connector (for Python-MySQL interaction)
- **IDE/Tools**: Any Python IDE (e.g., PyCharm, VSCode)

System Design:

ER Diagram



Implementation Details:

Technologies and Frameworks Used

- **Python**: Used for the backend logic, including voter registration, voting functionality, and result display.
- **MySQL**: Used as the database to store voter details, candidate details, and vote data.

Modules and Functionalities

1. **Register Voter**: Handles voter registration, checks if the email is already registered, and ensures the voter is eligible.
2. **Cast Vote**: Allows registered voters to cast their vote for a selected candidate and ensures a voter votes only once.
3. **Display Votes**: Displays the total number of votes for each candidate.

Testing:

Test Cases and Results

- **Test Case 1**: Voter Registration (Valid Input)

```
import mysql.connector

def register_voter():
    conn = None
    cursor = None
    try:
        conn = mysql.connector.connect(host="localhost", user="root", password="root123", database="voting_system")
        cursor = conn.cursor()

        name = input("Enter your name: ")
        email = input("Enter your email: ")

        # Ask for the voter's age
        age = int(input("Enter your age: "))

        # Check if the email already exists
        cursor.execute("SELECT * FROM Voters WHERE voter_email = %s", (email,))
        if cursor.fetchone():
            print("Email already registered!")
            return
```

Result: Successfully registers the voter.

```
Welcome to the Voting System!
1. Register as a voter
2. Cast your vote
3. View total votes for each candidate
4. Exit
Please select an option (1-4): 1
Enter your name: shadi
Enter your email: shadi@gmail.com
Enter your age: 18
Registration successful!
```

- **Test Case 2**: Voter Registration (Age below 18)

```
    # Check if the voter is 18 or older
    if age < 18:
        print("Sorry, you must be at least 18 years old to register.")
        return

    # Insert the voter into the database with the age provided
    cursor.execute("INSERT INTO Voters (voter_name, voter_email, age) VALUES (%s, %s, %s)", (name, email, age))
    conn.commit()
    print("Registration successful!")

except mysql.connector.Error as err:
    print(f"Error: {err}")
finally:
    if cursor:
        cursor.close()
    if conn:
        conn.close()
```

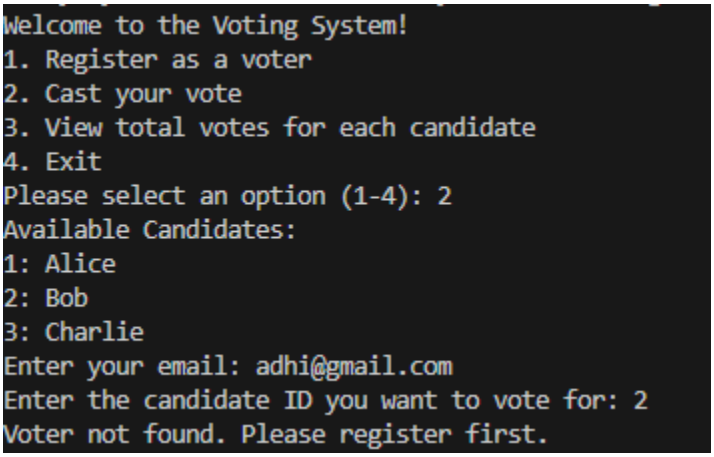
Result: Displays an error message, "You must be at least 18 years old to register."

```
Welcome to the Voting System!
1. Register as a voter
2. Cast your vote
3. View total votes for each candidate
4. Exit
Please select an option (1-4): 1
Enter your name: adhi
Enter your email: adhi@gmail.com
Enter your age: 17
Sorry, you must be at least 18 years old to register.
```

- **Test Case 3**: Cast Vote (Valid Voter)

```python
def cast_vote():
    conn = None
    cursor = None
    try:
        conn = mysql.connector.connect(host="localhost", user="root", password="root123", database="voting_system")
        cursor = conn.cursor()

        # Fetch available candidates
        cursor.execute("SELECT * FROM Candidates")
        candidates = cursor.fetchall()
        print("Available Candidates:")
        for candidate in candidates:
            print(f"{candidate[0]}: {candidate[1]}")

        # Get the voter's email and the chosen candidate's ID
        email = input("Enter your email: ")
        candidate_id = int(input("Enter the candidate ID you want to vote for: "))

        # Get voter_id using email
        cursor.execute("SELECT voter_id FROM Voters WHERE voter_email = %s", (email,))
        voter = cursor.fetchone()
        if not voter:
            print("Voter not found. Please register first.")
            return
        voter_id = voter[0]

        # Check if the voter has already voted
        cursor.execute("SELECT * FROM Votes WHERE voter_id = %s", (voter_id,))
        if cursor.fetchone():
            print("You have already voted!")
        else:
            cursor.execute("INSERT INTO Votes (voter_id, candidate_id) VALUES (%s, %s)", (voter_id, candidate_id))
            conn.commit()
            print("Your vote has been cast successfully!")

    except mysql.connector.Error as err:
        print(f"Error: {err}")
    finally:
        if cursor:
            cursor.close()
        if conn:
            conn.close()
```

Result: Successfully casts the vote for the chosen candidate.



- **Test Case 4**: Display Votes

```python
def display_votes():
    conn = None
    cursor = None
    try:
        conn = mysql.connector.connect(host="localhost", user="root", password="root123", database="voting_system")
        cursor = conn.cursor()
        cursor.execute("""
            SELECT c.candidate_name, COUNT(v.vote_id) AS total_votes
            FROM Candidates c
            LEFT JOIN Votes v ON c.candidate_id = v.candidate_id
            GROUP BY c.candidate_name
        """)
        results = cursor.fetchall()
        print("Total votes for each candidate:")
        for result in results:
            print(f"{result[0]}: {result[1]} votes")

    except mysql.connector.Error as err:
        print(f"Error: {err}")
    finally:
        if cursor:
            cursor.close()
        if conn:
            conn.close()
def main_menu():
    print("Welcome to the Voting System!")
    print("1. Register as a voter")
    print("2. Cast your vote")
    print("3. View total votes for each candidate")
    print("4. Exit")
```

Result: Displays the total votes for each candidate.

```
Welcome to the Voting System!
1. Register as a voter
2. Cast your vote
3. View total votes for each candidate
4. Exit
Please select an option (1-4): 3
Total votes for each candidate:
Alice: 0 votes
Bob: 0 votes
Charlie: 0 votes
```

Results and Discussion:

Key Findings:

- The system is able to register voters, record votes, and display results accurately.
- The system effectively prevents multiple voting attempts.

Performance Analysis:

- The system performs efficiently even with a larger number of voters and candidates, as it uses optimized SQL queries.

Conclusion:

This **Simple Voting System** is a robust and efficient solution for managing voter registration, casting votes, and displaying results. The system is secure and prevents multiple votes, making it ideal for small-scale elections or organizational voting processes. Future enhancements could include adding user.

authentication for more security and expanding the system to handle larger-scale elections.