PROJECT PROPOSAL

ON

Online Voting System

Submitted By: -

SHUBHAM KUMAR

Student ID : - AF0414103

Requiremant :-

In today's digital age, the need for secure, efficient, and accessible voting methods has never been more critical. Traditional voting systems, while reliable, often face challenges such as logistical difficulties, long queues, and limited accessibility, particularly for those with physical disabilities or living in remote areas. The advancement of technology presents an opportunity to overcome these challenges through the development of an online voting system.

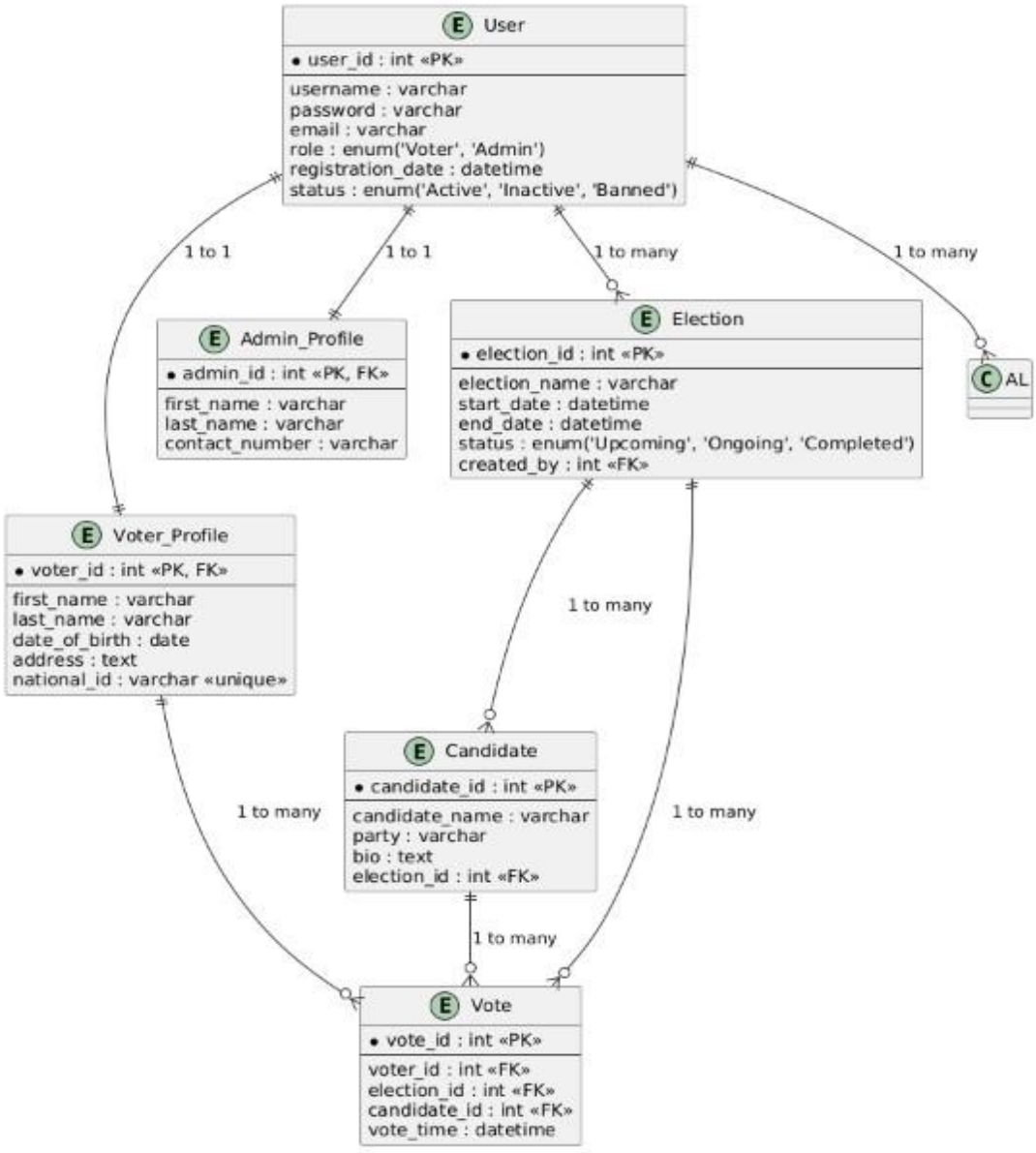
This project aims to create a robust Online Voting System using Java, offering a secure and user-friendly platform for conducting elections electronically. The system is designed to streamline the voting process, allowing eligible voters to cast their ballots from anywhere with internet access. It leverages modern web technologies and strong encryption methods to ensure that votes are confidential and tamper-proof.

By transitioning from paper-based to digital voting, the system aims to enhance voter turnout, reduce administrative costs, and minimize errors associated with manual vote counting. The Online Voting System is intended for use in various types of elections, including government elections, student body elections, and corporate board elections, providing a flexible and scalable solution that can be tailored to different needs.

Database Design:-

1. Entity-Relationship (ER) Diagram

The ER diagram visually represents the relationships between different entities in the system. Below are the key entities and their relationships.



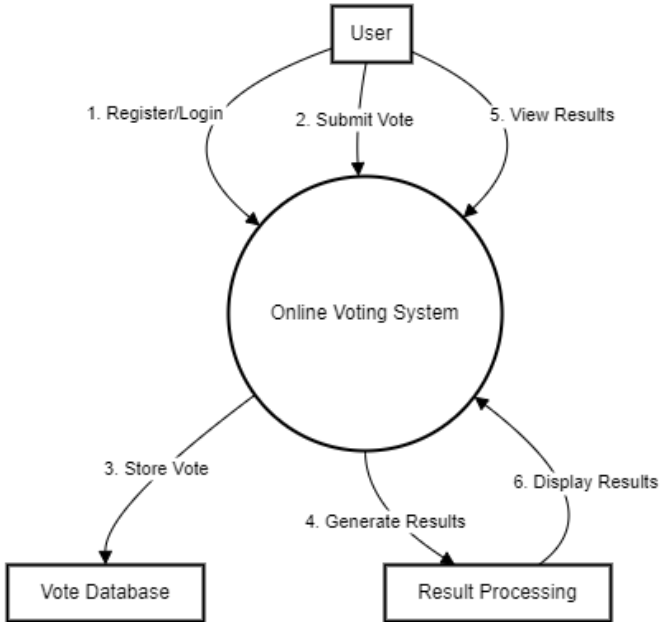
2. Key Entities and Attributes

1. **User**
    - **user_id (PK)**: Unique identifier for each user (voter or admin).
    - **username**: User's chosen username.
    - **password**: Hashed password for authentication.
    - **email**: User's email address.
    - **role**: Defines the role of the user (e.g., Voter, Admin).
    - **registration_date**: Date when the user registered.
    - **status**: Indicates if the user is active, inactive, or banned.
2. **Voter_Profile**
    - **voter_id (PK, FK)**: Unique identifier linked to the User table.
    - **first_name**: Voter's first name.
    - **last_name**: Voter's last name.
    - **date_of_birth**: Voter's date of birth.
    - **address**: Residential address of the voter.
    - **national_id**: Unique national identification number.
3. **Admin_Profile**
    - **admin_id (PK, FK)**: Unique identifier linked to the User table.
    - **first_name**: Admin's first name.
    - **last_name**: Admin's last name.
    - **contact_number**: Admin's contact number.
4. **Election**
    - **election_id (PK)**: Unique identifier for each election.
    - **election_name**: Name of the election (e.g., Presidential Election 2024).
    - **start_date**: Date and time when the election starts.
    - **end_date**: Date and time when the election ends.
5. **Candidate**
    - **candidate_id (PK)**: Unique identifier for each candidate.
    - **candidate_name**: Name of the candidate.
    - **party**: Political party or affiliation.
    - **election_id (FK)**: The election in which the candidate is participating.
    - **bio**: A brief biography of the candidate.
6. **Vote**
    - **vote_id (PK)**: Unique identifier for each vote.
    - **voter_id (FK)**: The voter who cast the vote.

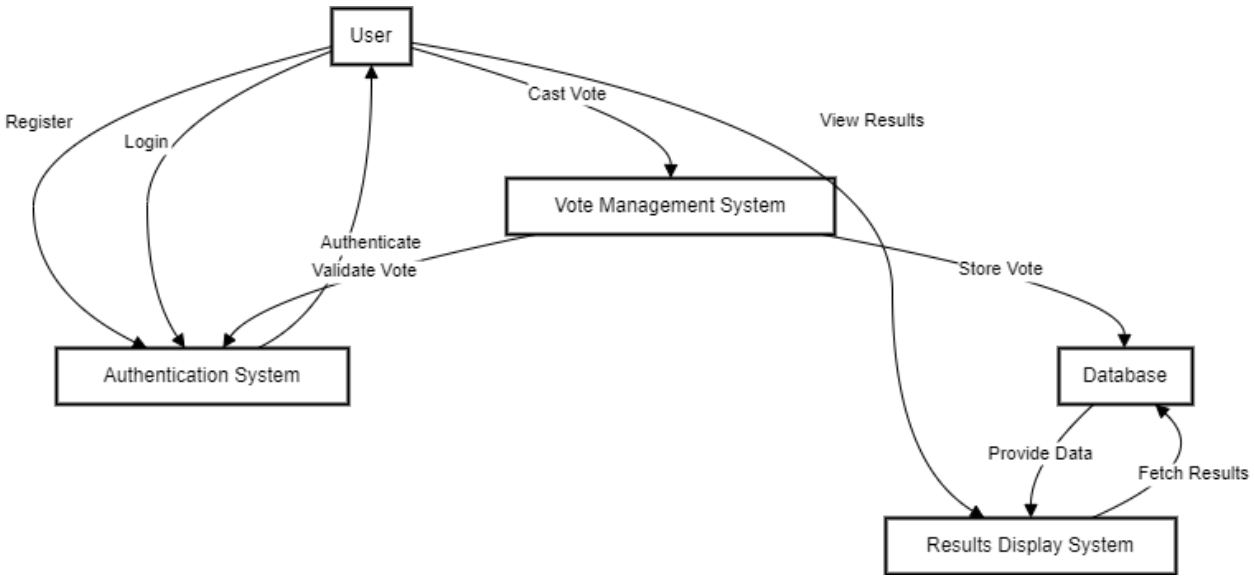
- **election_id (FK):** The election in which the vote was cast.

Data-Flow Diagram :-

0 Level DFD :-



1 Level DFD :-



Modules Description :-

The Online Voting System is divided into several key modules, each responsible for a specific aspect of the system's functionality. These modules work together to provide a seamless, secure, and user-friendly voting experience.

1. User Management Module

Description: This module handles the registration, authentication, and management of users, including voters and administrators. It ensures that only authorized users can access the system and participate in elections.

Key Features:

- **User Registration:** Allows new users to sign up by providing necessary details such as name, email, password, and role (Voter or Admin).
- **User Authentication:** Manages login functionality, including secure password handling and optional multi-factor authentication (MFA).
- **Profile Management:** Enables users to update their personal information, such as contact details and password.
- **Role Management:** Differentiates between various roles (e.g., Voter, Admin) and grants appropriate access levels.
- **Account Recovery:** Provides functionality for password reset and account recovery.

2. Election Management Module

Description: This module is responsible for the creation, management, and monitoring of elections by administrators. It allows admins to define election parameters and oversee the election process.

Key Features:

- **Election Creation:** Enables admins to set up new elections by defining election name, description, start and end dates, and status (Upcoming, Ongoing, Completed).

- **Candidate Management:** Allows admins to add, update, or remove candidates for each election, including details such as name, party affiliation, and biography.
- **Voter Eligibility:** Lets admins define voter eligibility criteria and manage the list of eligible voters.
- **Real-Time Monitoring:** Provides tools for admins to monitor election progress, including voter turnout and status updates.
- **Election Closure:** Allows admins to close elections, at which point voting is disabled, and results can be processed.

3. Voting Module

Description: The core functionality of the system, this module manages the voting process itself, ensuring that votes are securely cast, recorded, and stored.

Key Features:

- **Ballot Presentation:** Displays the list of candidates to the voter, allowing them to make their selection.
- **Vote Casting:** Ensures that votes are securely recorded, and voters can only vote once per election.
- **Vote Confirmation:** Provides voters with a confirmation message or receipt after their vote has been successfully recorded.
- **Encryption:** Implements encryption techniques to ensure that votes are securely transmitted and stored, preventing tampering or unauthorized access.

4. Results and Reporting Module

Description: This module handles the processing and presentation of election results, offering tools for both real-time reporting and post-election analysis.

Key Features:

- **Vote Counting:** Automatically tallies votes once an election is closed, ensuring accuracy and integrity in the results.
- **Results Display:** Provides visual and statistical representations of election outcomes, including vote totals for each candidate.

- **Audit Trails:** Logs all actions related to the election process, providing transparency and accountability.
- **Reports Generation:** Allows administrators to generate detailed reports on voter turnout, election results, and other relevant metrics.

5. Security Management Module

Description: This module is dedicated to ensuring the overall security of the system, protecting it from external threats and unauthorized access.

Key Features:

- **Encryption Management:** Handles the encryption of sensitive data, including voter credentials and ballots.
- **Access Control:** Implements role-based access control (RBAC) to restrict access to sensitive functionalities based on user roles.
- **Incident Response:** Monitors for potential security breaches and provides mechanisms for administrators to respond to incidents.

6. Notification Module

Description: This module manages the communication between the system and its users, ensuring that all participants are informed about key events and updates.

Key Features:

- **Email Notifications:** Sends automated emails to users for various events, such as registration confirmation, election start, and voting reminders.
- **SMS Alerts:** Provides optional SMS alerts for important notifications, such as election closure or password recovery.
- **In-App Notifications:** Displays real-time notifications within the system interface for logged-in users.

HARDWARE/SOFTWARE REQUIREMENT

Hardware Requirement

Server-Side Hardware:

- Processor :- Intel Xeon or equivalent (Quad-core or higher)
- RAM :- 4GB or higher (8 GB recommended for larger deployments)
- Storage :- SSD with 250 GB or higher
- Network :- High-speed internet connection with a minimum of 1 Gbps

Client-Side Hardware:

- Processor :- Intel Core i3 or equivalent (or higher)
- RAM :- 4 GB or higher
- Storage :- 100 GB hard disk space or higher
- Display :- 1024x768 resolution minimum
- Network :- Internet connection with a minimum of 5 Mbps
- Input Devices :- Keyboard, mouse, or touch-enabled devices

Server-Side Software:

Operating System:

Linux-based OS (e.g., Ubuntu Server, CentOS)

Alternatively, Windows Server OS can be used.

Web Server:

Apache Tomcat or any other compatible servlet container

Database Management System (DBMS):

MySQL, PostgreSQL, or any other RDBMS compatible with Java

Java Development Kit (JDK):

JDK 8 or higher (preferably the latest stable version)

Client-Side Hardware:

Windows 10 or later.

macOS 10.14 (Mojave) or later.

Linux distributions (e.g., Ubuntu 18.04 LTS or later).

Google Chrome (latest version).

Testing :-

1. Unit Testing:

- Objective: To test individual components or modules of the application (e.g., user authentication, vote submission, etc.) in isolation.

2. Integration Testing:

- **Objective:** To test the interaction between different modules of the system and ensure they work together as expected.

3. System Testing:

- **Objective:** To test the entire system as a whole, ensuring that all components function together in a real-world scenario.

4. Security Testing:

- **Objective:** To identify vulnerabilities and ensure the application is secure against potential attacks.

5. Performance Testing:

- **Objective:** To ensure that the system can handle expected and peak loads without performance degradation.

6. User Acceptance Testing (UAT):

- **Objective:** To ensure the system meets the requirements and expectations of the end-users.

7. Regression Testing:

- **Objective:** To ensure that new updates or changes do not introduce new bugs into the existing system.