ONLINE VOTING SYSTEM

AN INTERNSHIP PROJECT REPORT

Submitted in partial fulfillment of requirements to NEXT24TECH

For the award of the

Internship Certificate

By

Thangirala Sai Kumar Reddy

JULY - 2024

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without proper suggestions, guidance, and environment. The combination of these three factors acts as a backbone for our Internship Project **"Online Voting System".**

I would like to express my gratitude to the Management of **NEXT24Tech** for believing in me and assigning me this task.

I am greatly indebted to the **Project Assigner,** for providing detailed information and key features that need to be included in the project

I would like to express my special thanks of gratitude to **NEXT24TECH** for the continuous remainder of this project through the mail.

Thangirala Sai Kumar Reddy

List of Figures

Fig 1: Online Voting System Flowchart

Fig 2: Online Voting System ER Diagram

Fig 3: Login Page

Fig 4: Dashboard of Online Voting System

Table of Contents

1. Project: Online Voting System in PHP and MySQL

1.1. Creating an Online Voting System in PHP

1.2. Online Voting System

1.3. Requirements

1.4. Features of Online Voting System

1.5. Creating a Voting System in PHP

1.6. Database used for Voting System

1.7. Most Common Voting System

1.8. Implement Polling in PHP

1.9. Creating an Authentication System in PHP

1.10. Main Objective of the E-Voting System

1.11. Online Voting System Flowchart

1.12. Online Voting System ER Diagram

1.13. Relationships

1.14. Description of Online Voting System

1.14.1. Create the Login and Authentication System

1.14.2. Election/Contest Creation Module

1.14.3. Voting Module

1.14.4. Vote Counting Module

1.14.5. Results Display Module

1.15. Testing, Deployment, and Maintenance

1.15.1. Testing Online Voting System

1.15.2. Deployment

1.15.3. Maintenance

1.16. Screenshots

1.16.1. Planning and Designing Online Voting System

1.16.2. Determining the Features and Functionality of Online Voting System
1.16.3. Designing the User Interface of Online Voting System
1.16.4. Developing Online Voting System
1.16.4.1. Creating the Login and Authentication System
1.16.4.2. Creating the Election/Contest Creation Module
1.16.4.3. Creating the Vote Counting Module
1.16.4.4. Creating the Results Display Module
1.16.5. Testing, Deployment, and Maintenance of Online Voting System
1.16.5.1. Testing the Online Voting System in PHP and MySQL
1.16.5.2. Deploying the Online Voting System in PHP and MySQL
1.17. Maintaining the Online Voting System
1.18. How to Run This Online Voting System in PHP and MySQL
1.19. Conclusion

1. Project: Online Voting System in PHP and MySQL

An **online voting system** is a web-based application that allows users to cast their votes through the internet. This type of system is widely used in various fields such as politics, education, and business, as it enables efficient and convenient voting processes. Following are the steps for creating an online voting system in PHP, one of the most popular programming languages for web development.

1.1. Creating an Online Voting System in PHP

Online Voting Management System in PHP is an election system allowing voters to record a secret ballot and tabulate it electronically. Votes are stored so they can be re-counted should the need arise.

Online Voting Systems can speed up election results and lower the cost of conducting an election by significantly reducing the number of people required to operate a polling place and tabulate results. However, a primary concern with e-voting is how to store votes so they can be recounted if required.

1.2. Online Voting System

An **online voting system** is a software application that enables users to cast their votes through the internet. It provides an efficient and convenient way of voting, allowing users to participate from anywhere and at any time. Online voting systems are widely used in various fields, such as political elections, school and university elections, business and corporate decision-making, and surveys.

1.3. Requirements

Name of the Project: Online Voting System

Language **:** PHP

Databases used **:** MySQL

Design used : HTML, JavaScript, Ajax, JQuery, Bootstrap

Browser used : IE8, Google Chrome, Opera Mozilla

Software used : WAMP/XAMPP/LAMP/MAMP

1.4. Features of Online Voting System

The **Online Voting System** comes with a range of features to ensure smooth voting experience:

- **Vote Preview:** Users can preview their votes before submission.
- **Multiple Votes:** Support for multiple voting options.
- **Result Tally via Horizontal Barchart:** Visual representation of voting results
- **Print Voting Result in PDF:** Option to print voting results for documentation.
- **Changeable Order of Positions:** Flexibility in displaying positions on the ballot.
- **CRUD Operations for Voters, Candidates, and Positions:** Easy management of voter, candidate, and position data.

1.5. Creating a Voting System in PHP

To create a voting system in PHP, a good understanding of the language's fundamentals and web development concepts is essential. Here are the basic steps involved in creating a voting system:

- **Define the requirements:** Define the purpose of the online voting system, the type of voting method to use, and the security measures needed to implement.
    - The first step in creating an online voting system is designing the schema. The database schema should include tables storing

information about voters, candidates, and the voting process. Here are some of the key tables that should be included in the database:

- Users: This table should store information about voters, such as their name, email address, and login credentials.
- Candidates: This table should store information about the candidates running for office, such as their name, party affiliation, and photo.
- Votes: This table should store information about the votes cast by users, including the user ID, candidate ID, and the date and time of the vote.

- **Design the database:** Design the database schema and create the necessary tables to store user information, voting data, and other relevant information.
- **Develop the front end**: Create a user-friendly interface for the online voting system, including the ballot and voting options.
    - The next step is to develop the user interface for the online voting system. The user interface should be user-friendly and easy to navigate, allowing voters to easily find information about the candidates and cast their votes. Here are some of the key features that should be included in the user interface:
    - Candidate profiles: Each candidate should have a profile page that includes information about their background, platform, and stance on key issues.
    - Voting page: The voting page should allow users to cast their vote for their preferred candidate.
    - Results page: The results page should display the current vote count for each candidate and update in real-time as votes are cast.
- **Develop the back-end**: Develop the back-end of the online voting system using PHP, including the logic for vote counting, result calculation, and data validation.

- **Test and deploy**. Test the online voting system thoroughly and deploy it on a web server.

1.6. Database used for Voting System

The choice of database for a voting system depends on several factors such as the size of the system, the number of users, and the complexity of the data. However, some of the most popular databases for voting systems are MySQL, PostgreSQL, and Oracle. MySQL is a popular open-source database that is widely used in web applications. It is easy to use, scalable and provides good performance. PostgreSQL is a powerful open-source database that provides advanced features such as transaction support, table partitioning, and data replication. Oracle is a commercial database that provides high performance, scalability, and reliability for large-scale voting systems.

1.7. Most Common Voting System

The most common voting system used in political elections is the plurality voting system, also known as the first-past-the-post system. In this system, the candidate with the most votes wins the election. However, there are several other voting systems used in different contexts such as ranked-choice voting, approval voting, and proportional representation.

1.8. Implement Polling in PHP

To implement polling in PHP, develop a back-end script that collects poll data from users, stores it in a database, and calculates the poll results. Here are the basic steps involved in implementing polling in PHP:

- **Create the poll form:** Create a form that displays the poll question and the options to the user.
- **Process the poll data:** Process the poll data submitted by the user and store it in a database.

- **Calculate the poll results:** Calculate the poll results based on the data stored in the database and display the results to the user.

1.9. Creating an Authentication System in PHP

To create an authentication system in PHP, develop a back-end script that authenticates the user's credentials and provides access to the voting system. Here are the basic steps involved in creating an authentication system in PHP:

- **Design the database:** Design the database schema and create the necessary tables to store user information, login credentials, and other relevant information.
- **Develop the login form**. Develop a user-friendly login form that prompts the user to enter their login credentials.
- **Validate the login credentials**. Validate the user's login credentials against the information stored in the database.
- **Set up user sessions**. Set up user sessions to maintain the user's login state and provide access to the voting system.

1.10. Main Objective of the E-Voting System

The main objective of an e-voting system is to provide an efficient and convenient way of voting, allowing users to participate from anywhere and at any time. E-voting systems also aim to improve the accuracy and integrity of the voting process by reducing errors, minimizing fraud, and increasing transparency. Additionally, e-voting systems can reduce the cost and time involved in traditional voting methods.

1.11. Online Voting System Flowchart

This flowchart provides a detailed depiction of the entire process flow within the online voting system, starting from user authentication to the creation of elections, casting votes, accessing results, and finally logging

out. Each step is clearly defined, indicating the flow of actions and decisions within the system.

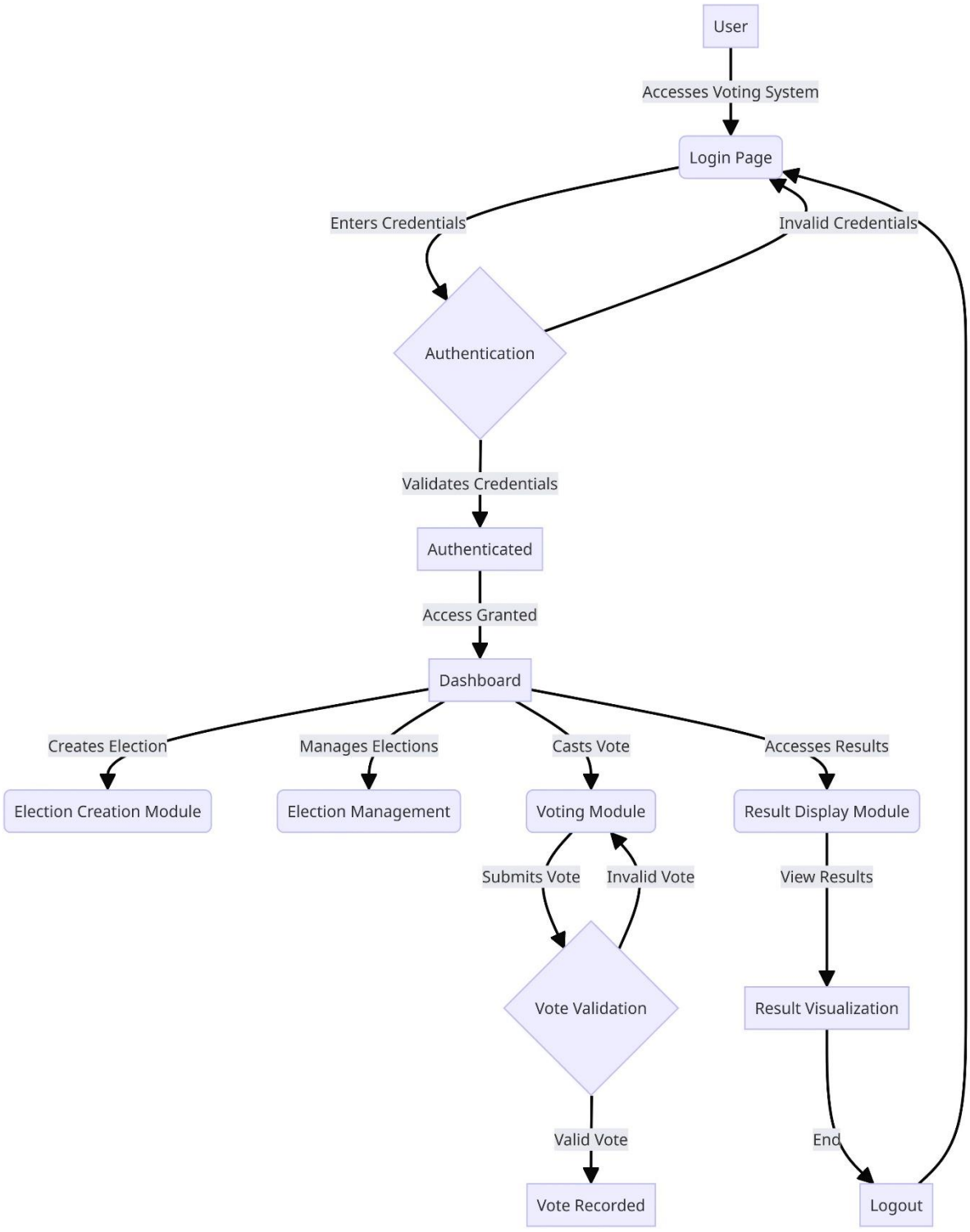

Fig 1: Online Voting System Flowchart

1.12. Online Voting System ER Diagram

Entities:

1. **User:**
    - Attributes:
        - **Id:** Primary Key (PK) representing the unique identifier for each user.
        - **username**: VARCHAR field storing the username of the user.
        - **password**: VARCHAR field storing the password of the user.
        - **role**: VARCHAR field representing the role of the user (e.g., ‘admin’ or ‘voter’).

2. **Election:**
    - Attributes:
        - **id**: Primary Key (PK) representing the unique identifier for each election.
        - **title**: VARCHAR field storing the title of the election.
        - **description**: TEXT field storing the description of the election.
        - **start_date**: DATETIME field indicating the start date and time of the election.
        - **end_date**: DATETIME field indicating the end date and time of the election.

3. **Candidate:**
    - Attributes
        - **id**: Primary Key (PK) representing the unique identifier for each candidate.
        - **name**: VARCHAR field storing the name of the candidate.
        - **party_affiliation**: VARCHAR field storing the party affiliation of the candidate.
        - **photo_url**: VARCHAR field storing the URL of the candidate’s photo.
        - **election_id**: Foreign Key (FK) referencing the election to which the candidate belongs.

4. Vote:

- Attributes:
  - **id**: Primary Key (PK) representing the unique identifier for each vote.
  - **voter_id**: Foreign Key (FK) referencing the user who cast the vote.
  - **candidate_id**: Foreign Key (FK) referencing the candidate for whom the vote is cast.
  - **date_time**: DATETIME field indicating the date and time when the vote was cast.

1.13. Relationships

1. User-Vote:

- One-to-many relationship indicating that a user can submit multiple votes.
- Relationship Label: “Submits”

2. Use-Election:

- One-to-many relationship indicating that a user can manage multiple elections.
- Relationship Label: “Manages”

3. Election-Candidate:

- One-to-many relationship indicating that an election can have multiple candidates competing.
- Relationship Label: “Competes In”

4. Vote-User:

- One-to-many relationship indicating that a vote is cast by a single user.
- Relationship Label: “Cast By”

5. Vote-Candidate:

- One-to-many relationship indicating that a vote is cast for a single candidate.
- Relationship Label: “Cast For”

This ER diagram outlines the structure of the online voting system database, including the entities, their attributes, and the relationships between them.

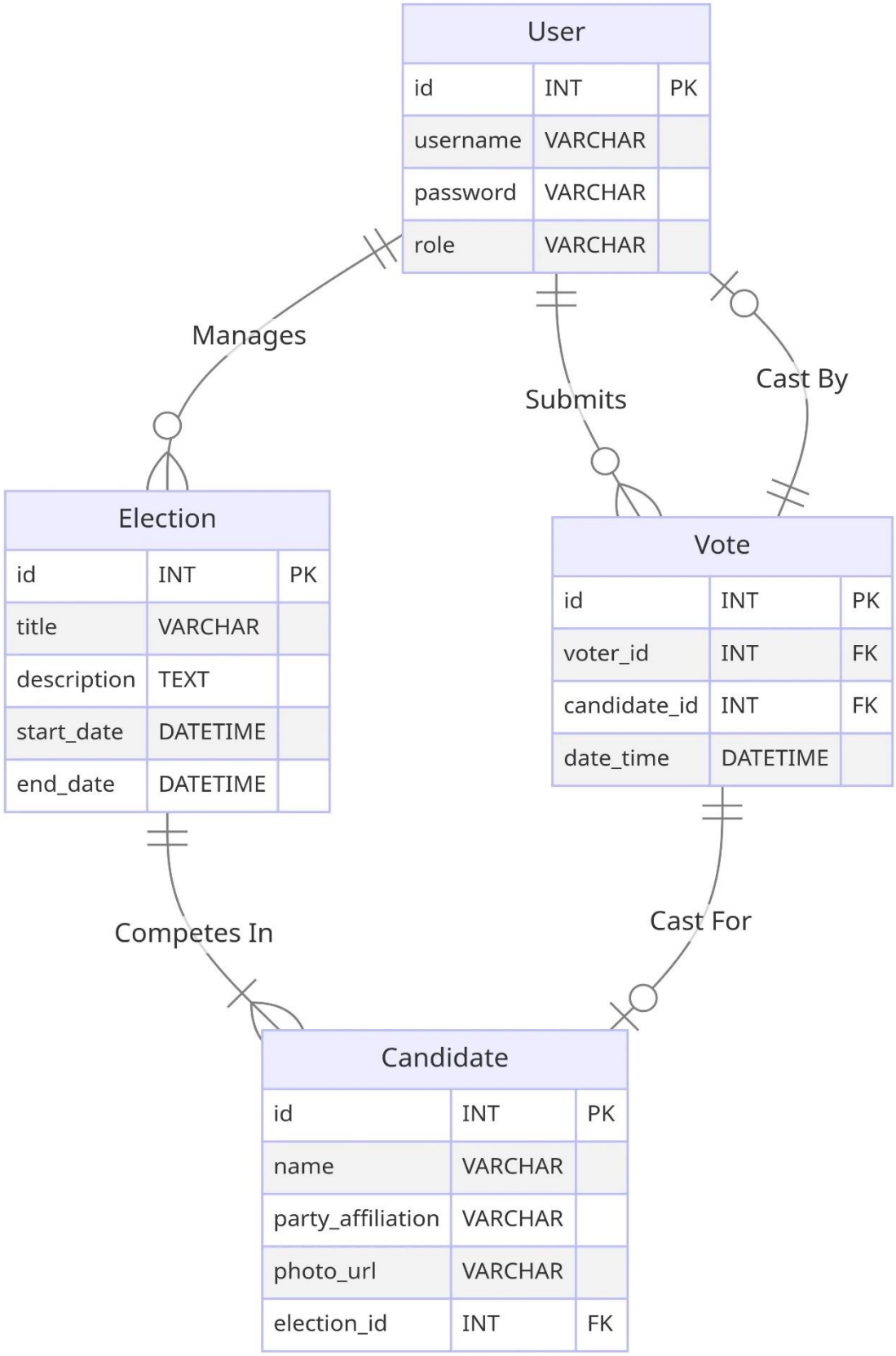

Fig 2: Online Voting System ER Diagram

1.14. Description of Online Voting System

1.13.1. Create the Login and Authentication System

Start by implementing a robust login and authentication system to secure access to the voting platform. Validate user credentials, manage sessions, and incorporate encryption for enhanced security.

1.13.2. Election/Contest Creation Module

Design a module to create and manage elections or contests. Allow administrators to define election parameters such as title, description, candidates, and voting duration.

1.13.3. Voting Module

Develop the core voting functionality, enabling users to cast their votes online. Present voters with clear options, validate votes and ensure data integrity throughout the process.

1.13.4. Vote Counting Module

Implement a module to tally votes and calculate results accurately. Apply appropriate algorithms to handle different voting methods and generate comprehensive reports.

1.13.5. Results Display Module

Create a module to showcase election results to users in an understandable format. Utilize tables, graphs, or charts to visualize data and provide insights into voting outcomes.

1.15. Testing, Deployment, and Maintenance

1.14.1. Testing Online Voting System

Thoroughly test the voting system to identify and rectify any bugs or

vulnerabilities. Conduct usability testing with real users to gather feedback for improvement.

1.14.2. Deployment

Deploy the system on a reliable web hosting service, ensuring compatibility with PHP, MySQL, and other required technologies. Secure data transmission with SSL encryption for user privacy.

1.14.3. Maintenance

Regularly update the system to incorporate new features and address security concerns. Monitor system performance, fix bugs promptly, and provide ongoing support to users.

1.16. Screenshots

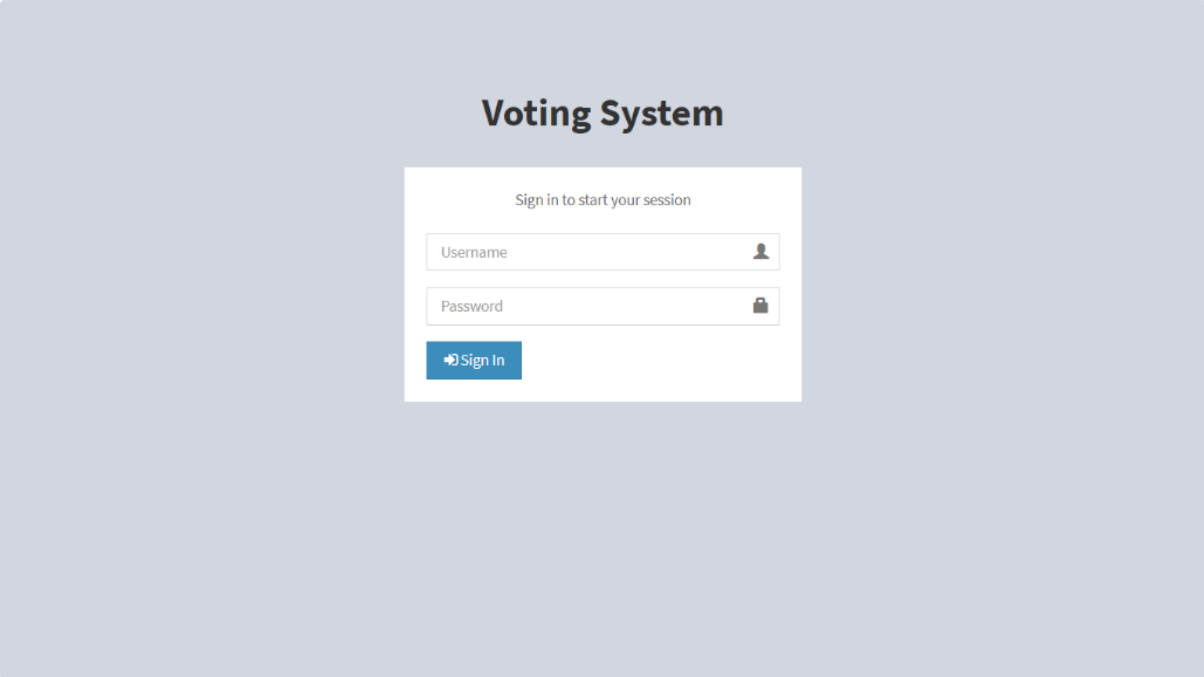

Fig 3: Login Page

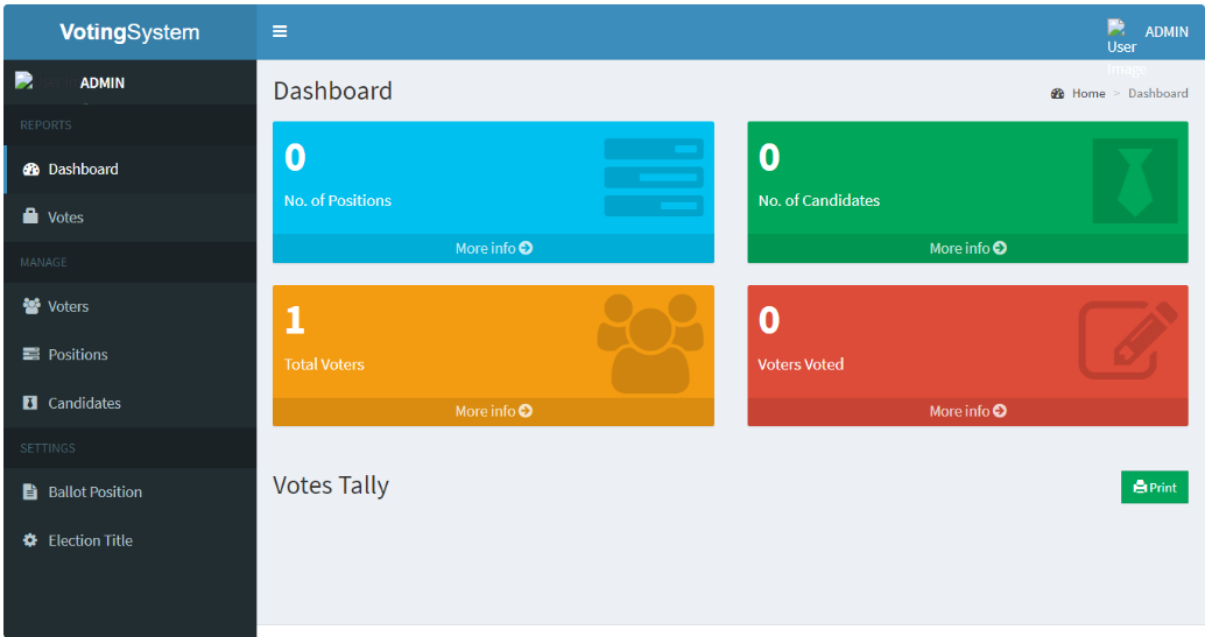

Fig 4: Dashboard of Online Voting System

1.16.1. Planning and Designing Online Voting System

In this section, we'll discuss the planning and designing stage of the online voting system. This includes determining the features and functionality of the system and designing its user interface.

1.16.2. Determining the Features and Functionality of Online Voting System

Before coding the online voting system, plan and conceptualize its features and functionality. Below are some questions that help get started:

- What types of elections or contests can be created?
- What are the specifications and requirements of each election or contest?
- How many candidates or entries are allowed?
- What types of voting systems are supported?
- How can voters cast and modify their votes online?
- What types of reports and analytics are needed?
- How will the system handle errors, invalid votes, and fraud attempts?

1.16.3. Designing the User Interface of Online Voting System

After determining the features and functionality of the online voting system, the next step is to design its user interface. The user interface is the system's visual and functional representation, and it's crucial to make it user-friendly and accessible.

Here are some tips to consider when designing the system's user interface:

- Keep it simple and easy to navigate
- Use clear and concise labels and instructions
- Make it responsive and accessible through different devices and platforms
- Use contrasting colors and visual hierarchies to highlight important elements
- Test it with real users to get feedback and improve its usability

1.16.4. Developing Online Voting System

In this section, we'll discuss the core functionality of the online voting system. This includes creating the login and authentication system, election/contest creation module, voting module, vote counting module, and results display module.

1.16.4.1. Creating the Login and Authentication System

The login and authentication system is responsible for verifying the identity of the user and granting them access to specific features or resources. Here are the steps to create one in PHP:

- Create a login form that can accept the username and password of the user.
- Validate the user's credentials by checking them against the database records.

- Store the user's login session and permission levels in a session variable.
- Implement a logout feature that can terminate the user's login session.
- Secure the login and authentication system by using encryption, password hashing, and other security measures.

1.16.4.2. Creating the Election/Contest Creation Module

The election/contest creation module is responsible for creating and managing elections or contests. Here are the steps to create one in PHP:

- Create a form that can collect the necessary information for creating an election/contest, such as the title, description, start and end dates, and candidates or entries.
- Validate the information and store it in the database.
- Display the list of existing elections/contests and allow the user to edit or delete them.
- H4: Creating the Voting Module
- The voting module is responsible for allowing voters to cast and modify their votes online. Here are the steps to create one in PHP:
- Create a form that can display the election/contest details and the candidates or entries.
- Allow the user to select one or more candidates or entries and submit their vote.
- Validate the vote and store it in the database.
- Allow the user to modify their vote before the deadline.

1.16.4.3. Creating the Vote Counting Module

The vote counting module is responsible for tallying the votes and computing the results. Here are the steps to create one in PHP:

- Retrieve the votes from the database and organize them according to the voting system used.

- Apply the vote-counting algorithm and compute the results.
- Store the results in the database and display them to the users.

1.16.4.4. Creating the Results Display Module

The results display module is responsible for showing the election/contest results to the users. Here are the steps to create one in PHP:

- Retrieve the results from the database and display them in a table or graph.
- Allow the user to filter and sort the results according to different criteria.
- Provide a summary and analysis of the results using charts or other visual aids.

1.16.5. Testing, Deployment, and Maintenance of Online Voting System

The testing, deployment, and maintenance of the online voting system includes testing it for bugs, errors, and vulnerabilities, deploying it on the web, and maintaining it for updates and improvements.

1.16.5.1.Testing the Online Voting System in PHP and MySQL

Testing the online voting system is crucial to ensure its security, functionality, and usability. Here are some tips to consider when testing the system:

- Use automated testing tools to check for bugs and vulnerabilities.
- Test the online voting system on different devices, platforms, and browsers.
- Involve real users to test the system's usability and user experience.
- Assess the system's security using penetration testing and vulnerability scanning.

1.16.5.2. Deploying the Online Voting System in PHP and MySQL

Deploying the online voting system on the web requires the following steps:

- Choose a reliable web hosting service that can support PHP, MySQL, and other web technologies.
- Register a domain name that suits the online voting system and is easy to remember.
- Obtain an SSL certificate to secure the system's data transmission and communication.
- Configure the server's settings and database to ensure optimal performance and security.

1.17. Maintaining the Online Voting System

Maintaining the online voting system involves keeping it up-to-date, fixing bugs and errors, and improving its features and functionality. Here are some tips to consider:

- Regularly update the system to the latest version of PHP, MySQL, and other web technologies.
- Fix bugs and errors promptly to ensure optimal performance and security.
- Implement user feedback and suggestions to improve the system's usability and user experience.
- Conduct security audits and risk assessments to identify potential threats and vulnerabilities.

1.18. How to Run this Online Voting System in PHP and MySQL

Above all, to run this project, it is essential to have installed a virtual server i.e. XAMPP on the PC.

Follow the following steps after Starting Apache and MySQL in XAMPP:

- Firstly, Extract the file
- After that, Copy the main project folder
- Then Paste it in xampp/htdocs/

Further, Now Connecting Database

- So, for now, Open a browser and go to the URL "http://localhost/phpmyadmin/"
- After that, Click on the Databases tab
- So, Create a database named "votingsystem" and then click on the import tab
- Certainly, Click on the browse file and select the "votingsystem.sql" file which is inside the "DB" folder
- Meanwhile, click on the Go button.

After Creating Database**,**

- Moreover, Open a browser and go to the URL "http://localhost/votingsystem"

****LOGIN DETAILS****

Admin

User name: Vote

Password: vote

1.19. Conclusion

Creating an online voting system in PHP and MySQL offers numerous benefits for organizations and institutions seeking to streamline their election processes. By following this comprehensive guide, it is easy to develop a secure, accessible, and efficient voting platform tailored to the specific requirements. Empower the stakeholders with the ability to participate in elections conveniently and securely—start building the online voting system today!

Creating an online voting system in PHP requires technical skills, planning, and dedication. But with the right tools and knowledge, it is easy to develop a secure, accessible, and reliable system that's suitable for different types of elections and contests. Whether a government agency, an organization, or a

business, an online voting system can help streamline election processes and increase voter turnout.