MALNAD COLEGE OF ENGINEERING

Under the auspices of M.T.E.S

(An autonomous institution under Visvesvaraya Technological University, Belgaum)
Hassan – 573201, Karnataka, India

Report on

Online Polling System

Course- Full Stack & Data Visualization

Submitted by

Submitted to,

Name	USN
Shrihari	4MC23IS0102
Ullas.M.	4MC23IS120
Ranjan.Y.	4MC23IS085
Suhas.B.R	4MC23IS111
Srujan	4MC23IS

Mr.Krishna Swaroop A

Assistant Professor

Dept. of ISE

Department of Information Science & Engineering

Malnad College of Engineering

Hassan– 573202

2025-26

CONTENT PAGE

1.Abstract

The Online Polling System is a comprehensive digital platform designed to automate and modernize voting procedures in educational, organizational, and institutional environments. The primary aim of this project is to replace conventional paper-based polling methods, which are often slow, inefficient, and prone to human error, with a web-based solution that enhances reliability and accessibility. Built using Python, Django, HTML, CSS, and MySQL, the system offers a secure platform where users can register, authenticate, and cast votes seamlessly. Administrators can create, edit, and manage multiple polls, each consisting of customizable options. Robust backend logic ensures that every user can cast only one vote per poll through strict validation mechanisms and database constraints. Real-time vote counting enables instant generation of live results, reducing delay and increasing transparency. The system effectively improves data accuracy, simplifies participation, and supports scalability, making it suitable for student elections, organizational surveys, opinion collection, and controlled digital voting environments. The outcome is a highly efficient, secure, and user-friendly polling system that demonstrates the potential of digital solutions to transform traditional operations.

2.Introduction

In an era where digital transformation is reshaping everyday processes, the need for efficient and trustworthy polling systems has become increasingly significant. Traditional polling methods rely heavily on manual operations—printing ballots, distributing paper slips, counting votes manually, and securely storing sensitive information. Such processes are time-consuming, resource-intensive, and susceptible to errors, duplicate voting, tampering, and delays in result processing.

Background of the Problem

Organizations, institutions, and colleges frequently conduct elections and surveys but often struggle with manual vote counting, breach of privacy, and logistical challenges. The absence of automated systems leads to inaccurate results, reduced participation, and lack of transparency.

Why This Domain Was Chosen

The domain of digital polling was chosen because:

- It represents a real-world need across multiple sectors.
- There is a lack of secure, accessible, and user-friendly voting technologies in many institutions.
- Digitization can solve the majority of the challenges faced by offline voting systems.
-

Real-World Scenario Description

Examples where digital polling is essential include:

- Student council elections
- Corporate employee feedback systems

- Club or association voting
- Community decision-making
- Online public opinion surveys

In all these cases, accuracy, speed, and security are critical.

Issues in the Existing System

- Manual counting errors
- Duplicate or unauthorized voting
- Time-consuming result compilation
- Difficulty ensuring confidentiality
- Measurable drop in voter participation due to physical presence requirement
-

How the Proposed Solution Helps

The Online Polling System resolves these issues by providing:

- Secure user authentication
- Automated vote validation
- Efficient poll management
- Real-time result display
- Reduced manpower and cost
- Increased accessibility by allowing users to vote remotely

The system therefore represents a modern, scalable alternative to outdated manual polling methods.

3.Objectives of the Project

- To build an online platform for managing polls, elections, surveys, and opinion collection with digital records.
- To simplify the process of creating, distributing, and participating in polls by providing a structured, user-friendly interface.
- To provide users with a secure environment where they can cast their votes with privacy and without any risk of manipulation.
- To digitize traditional voting processes by automating vote storage, validation, and result generation.
- To offer secure access to the system through role-based authentication for users and administrators.
- To improve operational efficiency and reduce human errors commonly found in manual vote-counting systems.
- To ensure one-vote-per-user through strict backend validation and database constraints.
- To maintain real-time synchronization of poll statistics for immediate result availability.
- To create a scalable platform capable of supporting multiple polls and large numbers of users simultaneously.

4.System Requirements

Software Requirements

Python

- Version: 3.8 or above
- Used for backend logic, managing server-side operations, and integrating REST functionalities.

Django Framework

- Version: 4.x
- Provides an MVC-based architecture, secure authentication system, URL routing, ORM-based database interactions, and inbuilt administrative interface.

IDE

- Visual Studio Code / PyCharm
- Used for code development, debugging, and project structuring.

Database

- MySQL 8.x
- Stores user data, polls, options, votes, and results securely and efficiently.

Additional Dependencies

- Django ORM
- HTML5 and CSS3 for frontend
- Bootstrap (optional) for responsive layouts
- MySQL connector
- Django Session Framework for tracking user sessions

5.System Design

A typical diagram will include:

- User Interface
- Django Views
- Django Models
- Database (MySQL)
- Admin Panel

Explanation of Architecture

The Online Polling System is developed using Django's MTV (Model–Template–View) architecture:

1. Templates (Frontend Layer)

- HTML pages display polls, results, and forms.
- CSS enhances visual appeal and usability.
- Users interact directly with this layer.

2. Views (Application Layer)

- Acts as a controller that receives requests from the frontend.
- Performs data validation, checks authentication, and processes business logic.
- Communicates with Models to access or store data.

3. Models (Data Layer)

- Represents database tables through Python classes.
- Manages relationships between polls, options, and votes.
- Ensures data consistency through constraints.

Data Flow Explanation

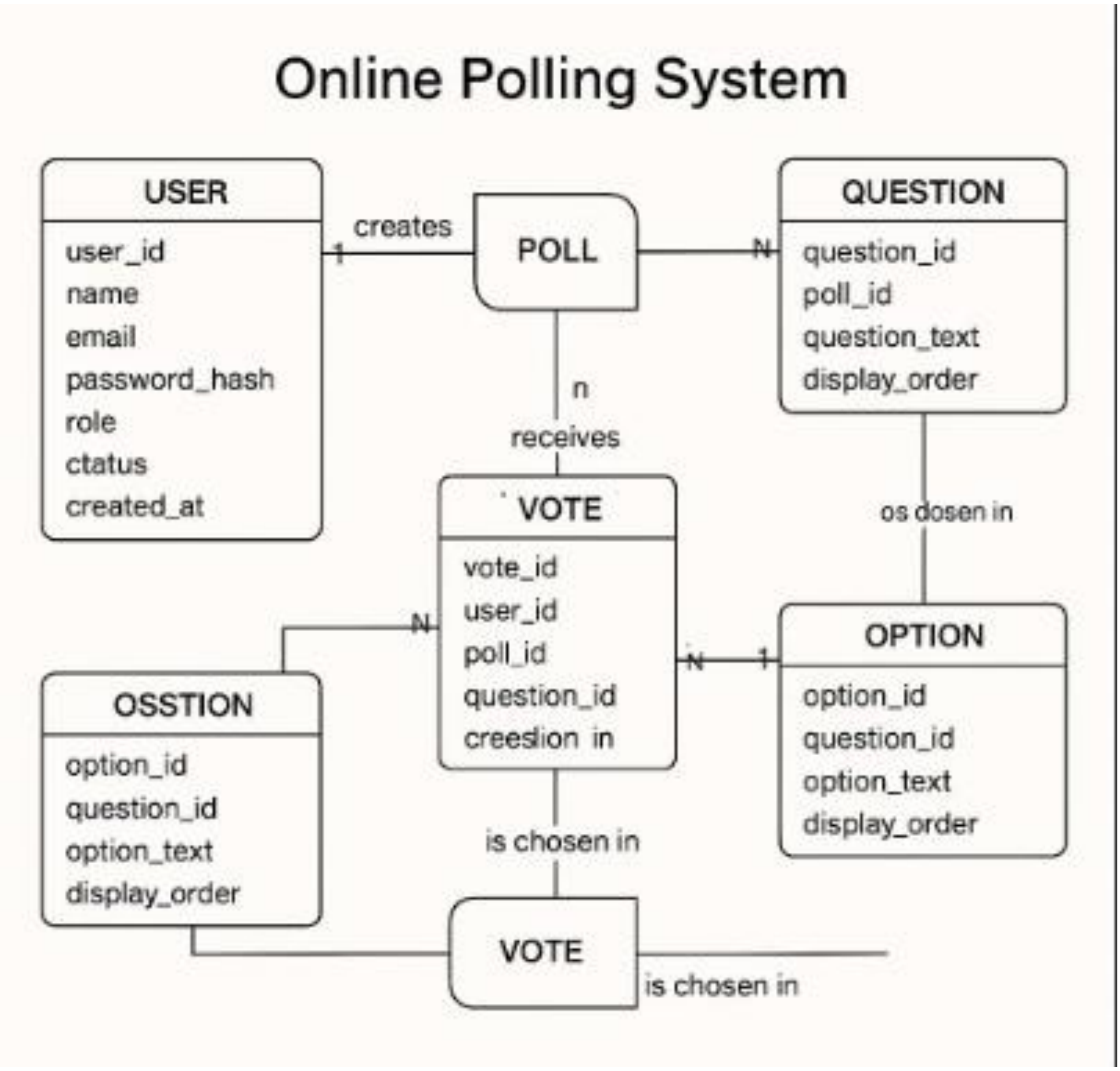
1. User sends request
2. URL Router maps request
3. View processes request and communicates with the model
4. Model interacts with database
5. Response is returned to template
6. Template displays data to user

Database Interaction

- Django ORM translates Python code into SQL queries.
- Ensures secure data handling through abstraction.
- Prevents SQL injection and maintains relational consistency.

6.Database Design

ER Diagram / Schema Diagram



Stores all registered user accounts.

- **Attributes:** user_id (PK), name, email, password, role, created_at
- One user can vote in multiple polls, but only once per poll.

Poll Table

Stores poll-related information.

- **Attributes:** poll_id (PK), question, description, start_date, end_date, created_by
- One poll contains multiple answer options.

Options Table

Stores the various choices for each poll.

- **Attributes:** option_id (PK), poll_id (FK), option_text
- Poll–Option relationship is one-to-many.

Vote Table

Stores each individual vote.

- **Attributes:** vote_id (PK), poll_id (FK), option_id (FK), user_id (FK), timestamp
- Unique constraint: user_id + poll_id (to avoid duplicate voting)

Admin Table

Stores admin login credentials and permissions.

7.Implementation

Models Overview

- **User Model:** Manages registration and login.
- **Poll Model:** Manages poll creation, duration, and description.
- **Option Model:** Handles poll options.
- **Vote Model:** Implements vote storage with validation.

URL Routing Overview

Important routes include:

- /login
- /register
- /polls
- /poll/<id>/vote
- /poll/<id>/results
- /admin/create_poll

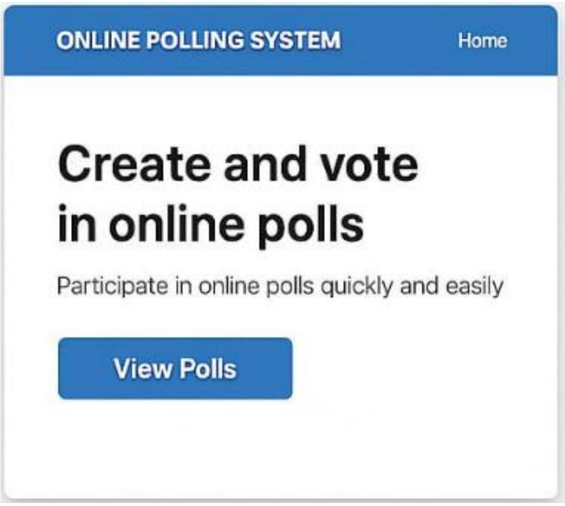
Important Functionalities

- Secure login and session management
- Admin can create and schedule polls
- Users can view polls and cast votes
- Preventing duplicate votes using backend logic
- Automatic result generation

Special Logic

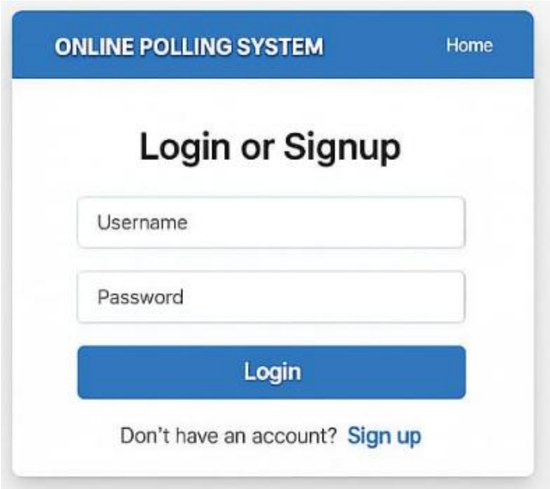
- Restricting one vote per user through unique constraints
- Auto-sync result updates using real-time counts
- Poll expiry based on end date

Screenshots



Create and vote in online polls

Participate in online polls quickly and easily

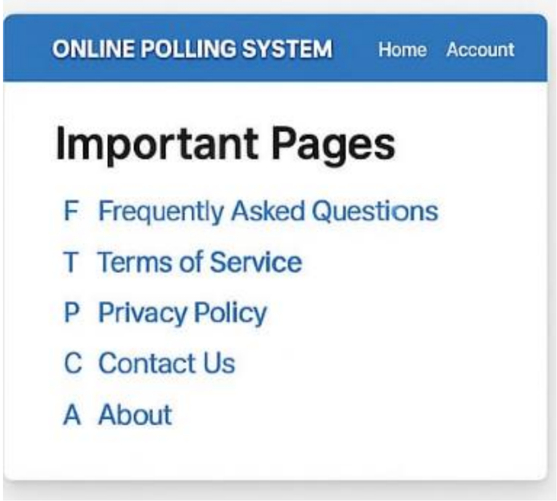


Login or Signup

Username

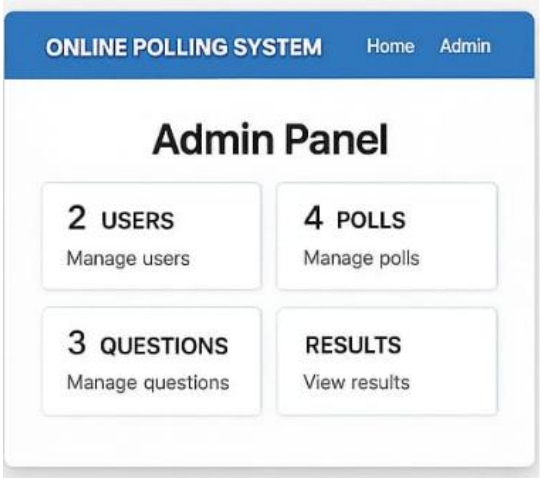
Password

Don't have an account? Sign up



Important Pages

- F Frequently Asked Questions
- T Terms of Service
- P Privacy Policy
- C Contact Us
- A About



Admin Panel

2 USERS Manage users	4 POLLS Manage polls
3 QUESTIONS Manage questions	RESULTS View results

8.Testing

Test Case Categories

1. **Form Validation Testing**
   - Empty field prevention
   - Email validation
   - Password mismatch errors
2. **Login/Authentication Testing**
   - Valid user login
   - Block incorrect credentials
   - Logout functionality
3. **CRUD Operations Testing**
   - Creating, editing, and deleting polls
   - Managing options
4. **Voting Functionality Testing**
   - Single vote restriction
   - Poll visibility only within active dates
5. **Error Handling Testing**
   - 404 page
   - Invalid poll access
   - Unauthorized vote attempts
6. **Database Testing**
   - Foreign key mappings
   - Vote duplication handling
   - Data retrieval accuracy

9.Results

The Online Polling System successfully demonstrates a complete digital voting platform that automates registration, poll creation, voting, and result generation. It ensures accuracy through database-level validation and improves transparency through real-time result display. Users can participate conveniently without physical presence, and administrators gain full control over poll management. Overall, the system resolves major challenges of manual polling and provides a secure, scalable web-based solution suitable for institutions, organizations, and public use.

10. Conclusion

The Online Polling System highlights the ability of modern web technologies to transform traditional polling operations. By integrating Django's secure framework and MySQL's reliable database structure, the system delivers an efficient, user-friendly, and accurate digital voting experience. It ensures fairness, enhances user participation, and eliminates delays caused by manual counting. The project demonstrates strong adaptability and forms the foundation for advanced polling technologies such as mobile platforms, secure blockchain-based voting, and AI-driven analytics.

11.References

- Django Official Documentation – https://docs.djangoproject.com/
- Python Official Documentation – https://docs.python.org/
- MySQL Developer Guide
- Stack Overflow Community Discussions
- Research papers on online voting and survey systems
- Web development tutorials and learning resources