



**Shivaji University, Kolhapur**

**A PROJECT REPORT ON**

**Bigg Boss Marathi Voting System**

**SUBMITTED FOR THE PARTIAL FULFILLMENT OF THE DEGREE**

**Bachelor of Computer Application**

**Part-III, Semester - V**

**By**

**Mr. Sushant Bhanudas Shekhar**

**Miss. Dipali Kumar Gaikwad**

**Miss. Aishwarya Bharat Shinde**

**Under The Guidance of**

**Dr. Kabir G.Kharade**



**Department of Computer Science,  
Shivaji University, Kolhapur**

**Year 2024-25**

## Certificate

This is to certify that, **Sushant Bhanudas Shekhar, Dipali Kumar Gaikwad and Aishwarya Bharat Shinde** have satisfactorily completed the project entitled as "**Bigg Boss Marathi Voting System**" in the partial fulfillment of BCA Part III SEM V during the academic year 2024-2025.

Place: Shivaji University, Kolhapur

Date:

ProjectGuide

Dr.KabirG.Kharade

Examiner

Head

Dr.K.S.Oza

## **ACKNOWLEDGEMENT**

Every project is always a scheduled, guided & coordinated team effort aimed at achieving common minimum goals. This minimum goal cannot be achieved without the guidance of guide.

It is with immense pleasure that we present our report to our project guide Dr. Kabir Kharade, Assistant Professor, Department of Computer Science, Shivaji University, Kolhapur. We find no words to describe his efforts and total confidence in our potential to see this project to completion. He has always been a source of inspiration and a tower of support boosting our moral beyond imagination. We would like to express our gratitude to our Head of the Department **Dr.K.S.Oza** for her continuing Support and encouragement. We sincerely express our gratitude to our parents for their blessing for making this project successful.

Finally, we are thankful to all the faculty members of Department of Computer and all our friends who have helped us to realize our efforts.

Thanking all of them, again.

Date:

Place: Kolhapur

**Mr. Sushant Bhanudas Shekhar**

**Miss. Dipali Kumar Gaikwad**

**Miss. Aishwarya Bharat Shinde**

## **DECLARATION**

We hereby declare that the project report entitled “**Bigg Boss Marathi Voting System**” have not formed earlier the basis for the award of any degree of this or any other university examining body.

Further, we declare we have not violated any of the previous under copyright act.

Place: Kolhapur

Date:..../..../2024

**Mr. Sushant Bhanudas Shekhar**

**Miss. Dipali Kumar Gaikwad**

**Miss. Aishwarya Bharat Shinde**

## Index

Sr.No	Contents	PageNo.
1.	<p>Introduction</p> <ul style="list-style-type: none"><li>• Overview of the Bigg Boss Marathi Voting System</li><li>• Purpose and Goals</li><li>• Advantage of Bigg Boss Marathi Voting System</li><li>• Limitations Bigg Boss Marathi Voting System</li></ul>	
2.	<p>System Analysis</p> <ul style="list-style-type: none"><li>• Define the Purpose and Scope</li><li>• Identify Stakeholders</li><li>• Gather Requirements</li><li>• System Architecture</li><li>• User Journey Mapping</li><li>• Technology Stack</li></ul>	
3.	<p>System Design</p> <ul style="list-style-type: none"><li>• Database Design</li><li>• Input Design</li><li>• Output Design</li></ul>	
4.	<p>System Diagram</p> <ul style="list-style-type: none"><li>• Data Flow Diagram</li><li>• Entity Relationship Diagram</li><li>• Uml</li></ul>	
5.	<p>Code</p> <ul style="list-style-type: none"><li>• Home</li><li>• Index</li><li>• Login</li><li>• Logout</li><li>• Profile</li><li>• Candidates</li><li>• Vote Success</li><li>• Flask Python for Backend</li></ul>	

	<ul style="list-style-type: none"> <li>• CSS</li> </ul>	
6.	<p>Used platforms</p> <ul style="list-style-type: none"> <li>• Existing System</li> <li>• FrontEnd– HTML, CSS, JavaScript</li> <li>• BackEnd–Flask in Python, MySQL</li> <li>• System Requirements</li> </ul> <ol style="list-style-type: none"> <li>1. Hardware</li> <li>2. Software</li> </ol>	
7.	<p>Conclusion &amp; Suggestion</p> <ul style="list-style-type: none"> <li>• Future Enhancement</li> <li>• Silent Feature</li> <li>• Suggestion</li> </ul>	
8.	Bibliography	

## **INTRODUCTION**

### ***Introduction***

The *Bigg Boss Marathi Voting System* is an online platform designed to facilitate the voting process for the popular reality show *Bigg Boss Marathi*. This system provides a transparent, user-friendly, and efficient way for viewers to cast their votes for their favorite contestants. Voting is a critical part of the show, as it determines the contestants' fate, deciding who stays and who gets evicted based on the audience's preferences.

The project idea revolves around creating a web-based application that simplifies the voting process by allowing users to log in, view contestant details, and cast votes securely. Using technologies such as HTML, CSS, Bootstrap, Flask for backend integration, and MySQL for data management, the system ensures a seamless and interactive experience for voters. The platform offers authentication to prevent fraudulent voting practices and ensures that each user can vote only once per voting cycle.

The voting system aims to mimic the high-stakes atmosphere of *Bigg Boss*, a show known for its dramatic evictions, emotional moments, and fierce competition among housemates. By engaging viewers and making them an active part of the decision-making process, this system reflects the interactive nature of the reality show.

### ***Overview of the Bigg Boss Marathi Voting System***

The *Bigg Boss Marathi Voting System* is a comprehensive web-based application developed to handle real-time voting for the reality show *Bigg Boss Marathi*. It allows users to register, log in, view contestant profiles, and vote for their favorite housemates. The system uses Flask as its backend framework, MySQL for database management, and a combination of HTML, CSS, and Bootstrap for front-end development. The project leverages a responsive design,

ensuring that users can access the platform from various devices, whether on desktop or mobile.

The voting system is secure, requiring users to sign up and verify their identity before casting votes. After logging in, they are presented with a list of contestants, along with their current standings, and can cast a vote for their preferred contestant. The system then securely stores these votes in the database and ensures that a user can only vote once per round, preventing multiple votes by the same individual.

This system is designed to handle large amounts of traffic, as *Bigg Boss Marathi* enjoys a massive viewer base. It includes features such as real-time vote count updates, contestant profiles with images and descriptions, and voting history tracking. With its streamlined interface, the system ensures that the voting process is both engaging and easy for the audience, making it a vital tool for showrunners to manage viewer engagement.

## **Purpose and Goals**

The primary purpose of the *Bigg Boss Marathi Voting System* is to create a transparent and reliable platform for audiences to participate in the voting process for *Bigg Boss Marathi*. This system ensures that viewers have an active role in determining which contestants continue their journey on the show, thus promoting fairness and audience engagement. The goal is to design a system that handles high volumes of data efficiently and provides a seamless experience to users while maintaining the integrity of the voting process.

The project seeks to create a voting system that is user-friendly, secure, and capable of preventing fraud by limiting votes per user session. It also aims to enhance the interactivity of the show by offering real-time vote updates and encouraging active viewer participation. In the long term, the goal is to create a scalable platform that can be adapted for

future seasons and other reality shows with minimal adjustments. Additionally, the system aspires to offer features like integration with social media platforms to enhance engagement and create a community-driven voting experience.

## ***Advantages of Bigg Boss Marathi Voting System***

The *Bigg Boss Marathi Voting System* offers several advantages that make it a valuable addition to the show's ecosystem. First and foremost, it streamlines the voting process, allowing viewers to cast their votes conveniently from their own devices. The user-friendly interface ensures that even those with minimal technical skills can navigate the system with ease.

The system also enhances fairness by implementing security measures like authentication and vote restrictions to prevent fraudulent voting. This not only ensures that the voting process remains fair but also increases the credibility of the results, which is crucial for maintaining the integrity of the show. Additionally, the use of MySQL databases ensures that data is managed and stored securely, with all votes accounted for accurately.

Another advantage is that the system can handle high traffic volumes, which is essential for a show as popular as *Bigg Boss Marathi*. The scalability of the system allows it to accommodate a large number of users without lagging or crashing, ensuring a smooth voting experience. Furthermore, the platform is responsive, meaning it can be accessed from both desktop and mobile devices, making it accessible to a broader audience.

## ***Limitations of Bigg Boss Marathi Voting System***

Despite its many advantages, the *Bigg Boss Marathi Voting System* has certain limitations. One of the primary limitations is its reliance on internet connectivity. Since it is a web-based system, users without stable

internet access may face difficulties in participating in the voting process, which could potentially exclude a portion of the audience.

Another limitation is that the system is susceptible to technical glitches, such as server downtime or slow page loading, especially during peak voting times when a large number of users access the platform simultaneously. While the system is designed to handle high traffic, unexpected surges can still cause performance issues.

Additionally, the system restricts each user to a single vote per cycle, which may be seen as a limitation by some users who wish to support their favorite contestants more actively. Moreover, despite security measures, there is always a risk of cyber-attacks or hacking attempts, which could compromise the integrity of the voting process. Although authentication measures are in place, determined attackers could potentially exploit vulnerabilities in the system.

Lastly, the voting system may not fully reflect the preferences of the broader audience, as it is limited to those who have access to the internet and are tech-savvy enough to use the platform, potentially skewing the results in favor of certain demographics.

## System Analysis

### *Define the Purpose and Scope*

The purpose of the *Bigg Boss Marathi Voting System* is to provide a reliable, efficient, and user-friendly platform for audience members to cast their votes for contestants of the reality show *Bigg Boss Marathi*. This system aims to make the voting process seamless, transparent, and secure while handling the high volume of user traffic and votes that the show typically attracts. It

plays a crucial role in determining which contestants continue on the show, contributing to the dramatic and competitive nature of the program.

The scope of the project includes creating a fully functional voting platform where users can register, log in, view contestant profiles, and vote. The system ensures that each user can vote only once per cycle, thus promoting fairness and accuracy in the voting process. It also includes administrative capabilities, such as the ability to view live vote counts, manage contestant profiles, and update system features. Additionally, the system must handle large volumes of real-time data, ensure secure vote storage, and provide real-time updates to users.

## ***Identify Stakeholders***

The primary stakeholders in the *Bigg Boss Marathi Voting System* include:

- **Audience/Viewers:** These are the end-users who interact with the system by logging in, viewing contestants, and casting votes. Their experience should be smooth and straightforward to ensure maximum participation.
- **Show Organizers/Administrators:** They are responsible for managing the system and ensuring its smooth operation. They need access to backend functionalities like vote monitoring, contestant management, and system performance analysis.
- **Contestants:** While not directly interacting with the system, contestants rely on the voting results to determine their standing in the show. Their profiles are displayed to users for voting purposes.
- **System Developers:** They are responsible for maintaining and updating the system, ensuring security, scalability, and reliability throughout the show's run.

- **Show Sponsors/Advertisers:** They might be indirectly involved as stakeholders, benefiting from higher audience engagement driven by the voting system.

## **Gather Requirements**

The following requirements are essential for the successful development and operation of the *Bigg Boss Marathi Voting System*:

### **1. Functional Requirements:**

- User registration and login with email/password authentication.
- Profile viewing of contestants with pictures, details, and current standing.
- Secure voting mechanism where users can vote once per cycle.
- Real-time vote counting and display.
- Administration panel for monitoring votes and updating contestant profiles.

### **2. Non-Functional Requirements:**

- **Scalability:** The system must handle large numbers of concurrent users and votes.
- **Security:** User authentication, encryption of sensitive data (like passwords), and protection against hacking.
- **Performance:** The system should load quickly and handle real-time updates without delays.
- **Usability:** The user interface should be intuitive, responsive, and accessible across devices.
- **Data Integrity:** The voting data should be stored securely in the MySQL database with appropriate backups to avoid data loss.

## **System Architecture**

The *Bigg Boss Marathi Voting System* is designed using a multi-tier architecture, separating the frontend, backend, and database layers to ensure scalability, security, and maintainability.

1. **Frontend Layer:** The user interface is built using HTML, CSS, and JavaScript to create a visually appealing and interactive experience. The responsive design ensures that the platform works across various devices, from desktops to mobile phones.
2. **Backend Layer:** The backend is powered by Python's Flask framework, which manages the core logic of the application, including user authentication, vote processing, and data interaction with the MySQL database. Flask handles the API requests, ensuring that the system runs efficiently and securely.
3. **Database Layer:** MySQL is used to store all system data, including user credentials, voting records, and contestant details. The database ensures that data is stored in a structured and secure manner, with encryption mechanisms used to protect sensitive information such as passwords.

## **User Journey Mapping**

The user journey is a critical aspect of the system's design, mapping out the interactions users have with the platform from the moment they access the voting system.

1. **User Registration/Login:** The journey begins with a user landing on the home page, where they can either register as a new user or log in using their existing credentials. Once logged in, they are redirected to the main voting page.
2. **Viewing Contestants:** On the voting page, users are presented with a list of current contestants. Each contestant profile includes relevant details like their name, photo, and performance summaries. Users can browse through these profiles to make an informed voting decision.

3. **Voting:** Once a user selects a contestant, they can cast their vote. The system then verifies that the user has not already voted during the current cycle. After validation, the vote is submitted, and the user is shown a confirmation message.
4. **Post-Vote:** After voting, users can view real-time voting results or logout. The system ensures that users are limited to one vote per cycle.

## ***Technology Stack***

The technology stack used in the development of the *Bigg Boss Marathi Voting System* includes:

- **Frontend:**
  - **HTML:** For structuring the web pages.
  - **CSS:** For styling and making the pages visually appealing, including the use of Bootstrap for a responsive design.
  - **JavaScript:** For adding interactivity, such as dynamic content updates without page reloads.
- **Backend:**
  - **Flask (Python):** The core framework that powers the backend logic, handles HTTP requests, and connects the frontend with the database.
  - **MySQL:** For storing and managing the system's data, including user credentials, voting records, and contestant profiles.
  - **MySQL Connector:** To facilitate interaction between Flask and the MySQL database, allowing data to be retrieved, stored, and processed efficiently.
  - **Encryption Options:** Encryption is used to secure sensitive data like passwords. This ensures that user credentials are stored in a safe manner, safeguarding against data breaches and unauthorized access.

This combination of technologies ensures that the *Bigg Boss Marathi Voting System* is both scalable and secure, capable of handling high traffic while providing a seamless user experience.

## System Design

### *Database Design*

The *Bigg Boss Marathi Voting System* relies on a structured and secure database to manage user information, contestant profiles, and voting data. The database is designed using **MySQL** to ensure efficient handling of data and scalable performance. Below is an overview of the database structure:

- **Purpose of Database:** The database is designed to store essential information related to users, contestants, and votes. This data is crucial for ensuring the integrity and fairness of the voting process.
- **Tables:**
  1. **Users Table:**
    - This table stores user registration details such as:
      - user\_id (Primary Key)
      - name
      - email
      - password (encrypted using bcrypt)
    - Each user has a unique user\_id that acts as the primary identifier in the system.

```
mysql> describe voter;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int | NO | PRI | NULL | auto_increment |
| name | varchar(100) | NO | | NULL | |
| password | varchar(255) | YES | | NULL | |
| email | varchar(100) | YES | | NULL | |
| address | varchar(200) | YES | | NULL | |
+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

## 2. Votes Table:

- This table records votes cast by users:
  - vote\_id (Primary Key)
  - user\_id (Foreign Key linking to Users table)
  - contestant\_id (Foreign Key linking to Contestants table)
  - vote\_timestamp (Date and time of the vote)
- This setup ensures that each vote is unique and linked to both the user and the contestant.

```
mysql> describe votes;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int | NO | PRI | NULL | auto_increment |
| voter_id | int | YES | MUL | NULL | |
| candidate_name | varchar(100) | YES | | NULL | |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

- Relationships:** The Votes table establishes relationships between the Users and Contestants tables through foreign keys. This ensures that votes are tracked and tied to specific users and contestants, creating a transparent and accountable voting system. Input Design

## *Input Design*

The *Input Design* for the *Bigg Boss Marathi Voting System* describes how users interact with the system by inputting data through various forms. These forms capture important details for user registration, login, and voting processes. Below are the key input forms:

- **Home Page:**
  - The home page serves as the entry point, featuring a "Signup Here" button that directs new users to the registration form. This allows users to sign up before casting votes.
- It also provides clear navigation options for existing users to log in and new users to get started.



- **Signup Page:**
  - The signup form is designed to capture the following information from users:
    - **Voter Name:** A text field for the user's name.
    - **Password:** A secure password field, encrypted using bcrypt to ensure safe storage.

- **Email Address:** An email field validated to prevent duplicates or incorrect formats.
- **Address:** A text field to collect voter address, ensuring the system can verify regional eligibility.
- Input validation ensures all fields are completed correctly before submission.
- Upon submission, the form creates a new voter entry in the database, securely storing credentials for future logins.



- **Login Page:**
  - The login page consists of two fields:
    - **Voter Name:** A text field for the voter's registered name.
    - **Password:** A password field validated against the stored encrypted password.
  - Once authenticated, the system allows the voter to access the voting page or navigate to other features of the system.
  - Error messages are displayed if the user's credentials do not match the records.



- **Candidates List Page:**
  - After logging in, users are directed to a page that lists all current contestants. Each contestant's profile can be viewed via a "View Profile" button.
- Users select their favorite contestant by navigating to their individual profile.

**Candidates List**

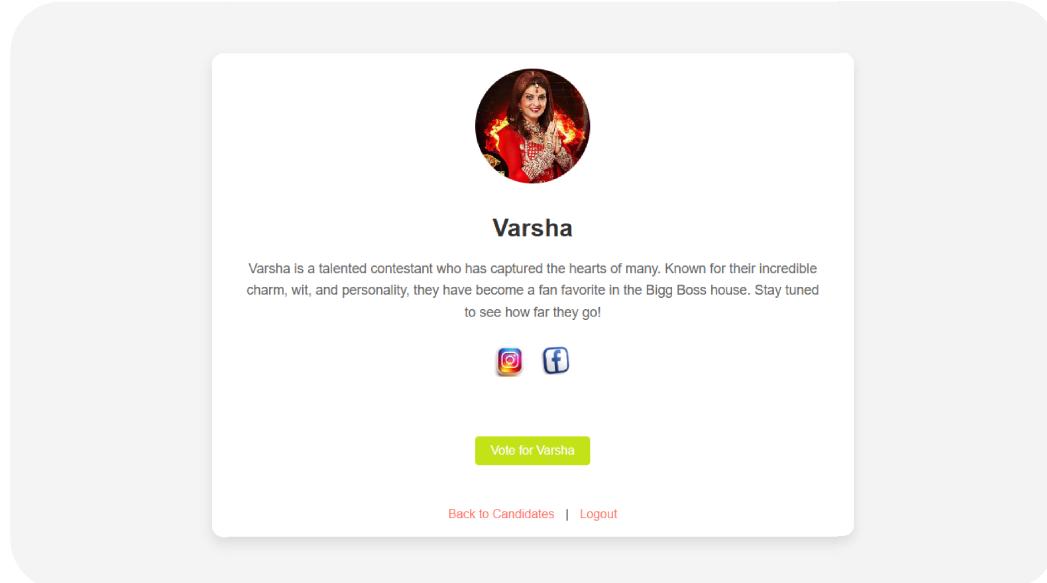
Login successful

 Ankita <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Aarya <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Abhijeet <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Arbaaz <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Dhananjanj <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Ghanshyam <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Irina <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Jantivi <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>
 Nikhil <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Nikki <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Pandarinath <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Suraj <a href="#" style="background-color: #ff0000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Vaibhav <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Varsha <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	 Yogita <a href="#" style="background-color: #008000; color: white; padding: 2px 10px; text-decoration: none; font-weight: bold;">View Profile</a>	

[Back to Home](#) | [Logout](#)

- **Vote Button on Candidate Profile:**
  - On the candidate's profile, users are presented with a **Vote** button.

- The form ensures that the user can vote only once per voting cycle, preventing multiple votes from the same account.
- The vote is submitted and added to the respective contestant's total vote count.

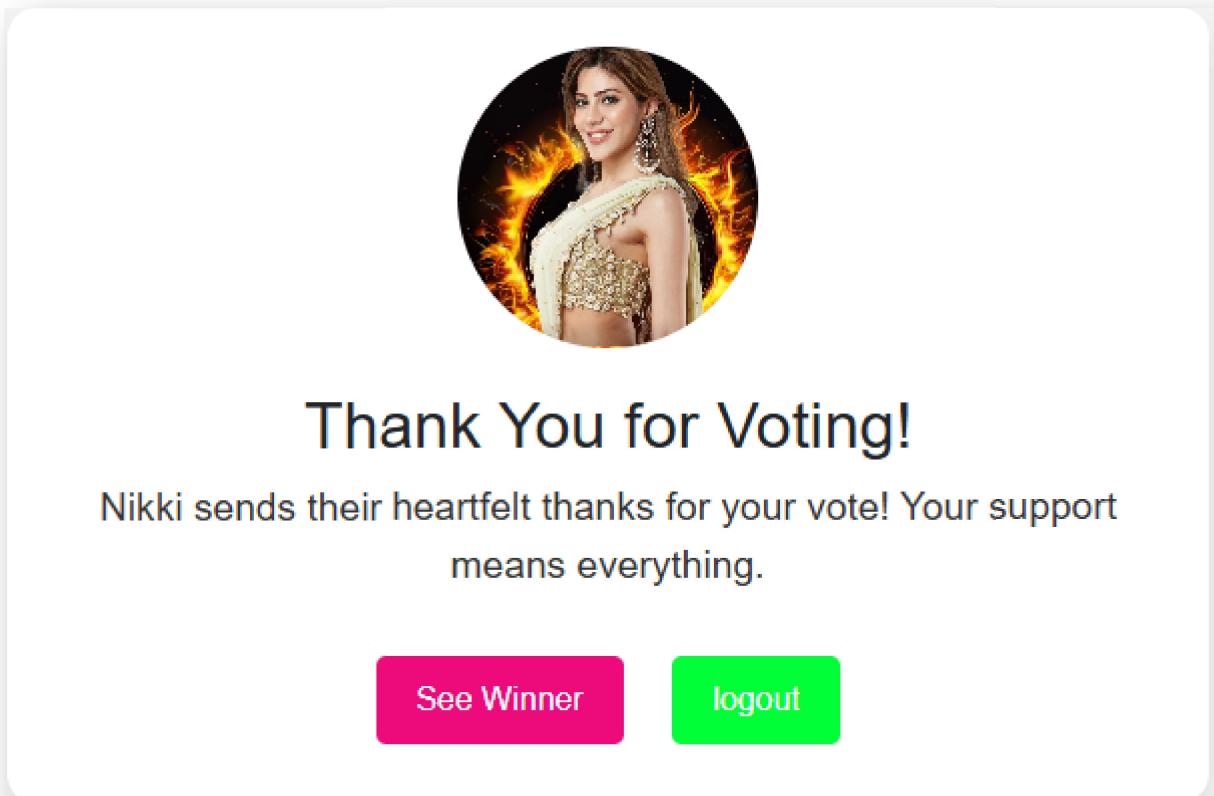


## *Output Design*

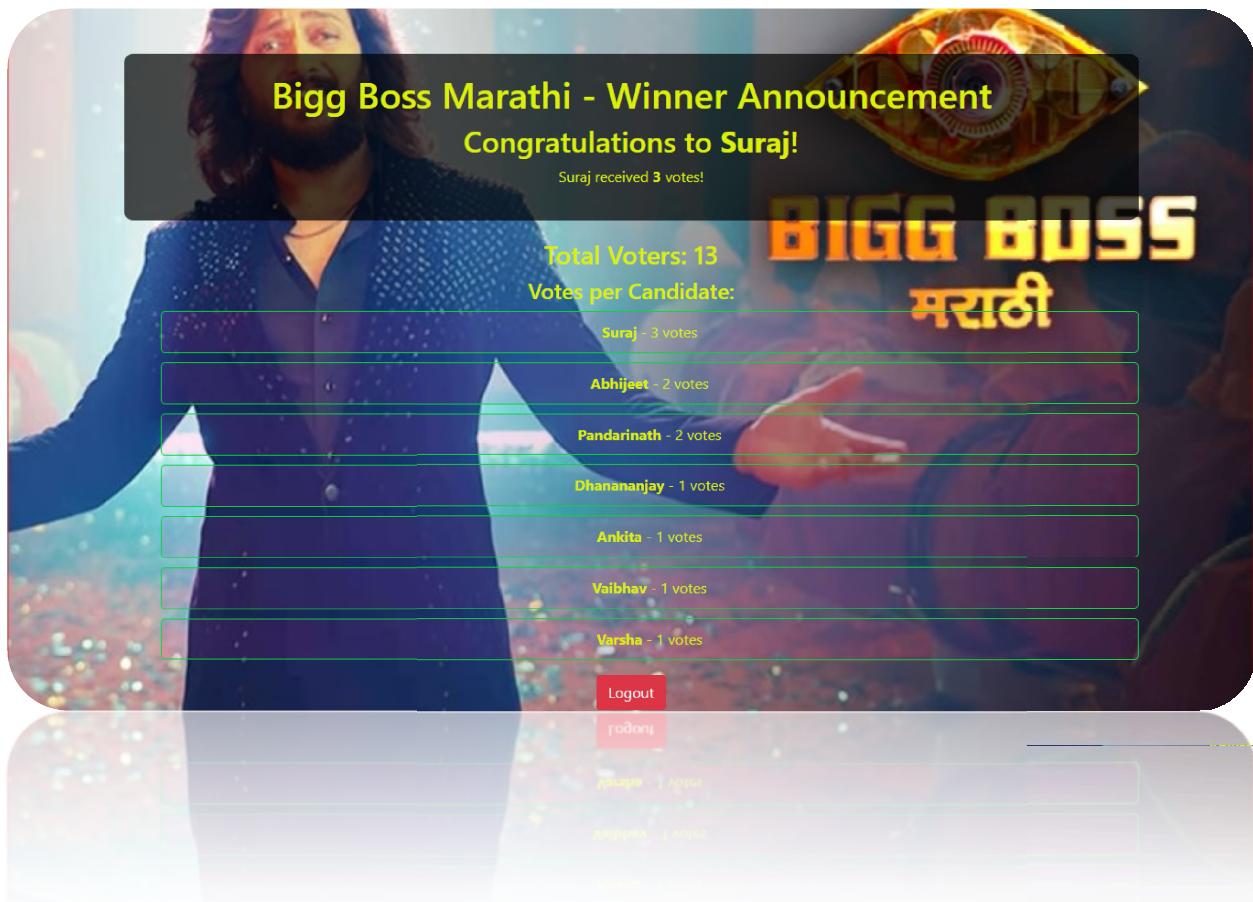
The *Output Design* focuses on how the system presents information back to the users, including confirmation messages and result data. Below are the primary outputs:

- **Candidates List:**
- The system outputs a list of candidates with each candidate's profile picture, name, and current vote status (if enabled). Users can navigate to individual profiles for more detailed information
- **Vote Confirmation:**
  - Once a vote is successfully cast, the system provides a confirmation screen, thanking the user for their vote. This ensures that users know their vote was received and counted.

- If the user attempts to vote again during the same cycle, a message is displayed informing them that they have already voted.
- **Thanks Card:**
  - After successfully voting, users are directed to a "Thank You" page or card, acknowledging their participation in the voting process.



- **Get Winner Data Page:**
  - This page displays the ongoing results of the voting. It shows the current vote count for each contestant and allows users and administrators to see the overall standing of each candidate.
  - Once the voting period is over, the winner of the contest is highlighted along with their total votes, providing transparent feedback to all users.

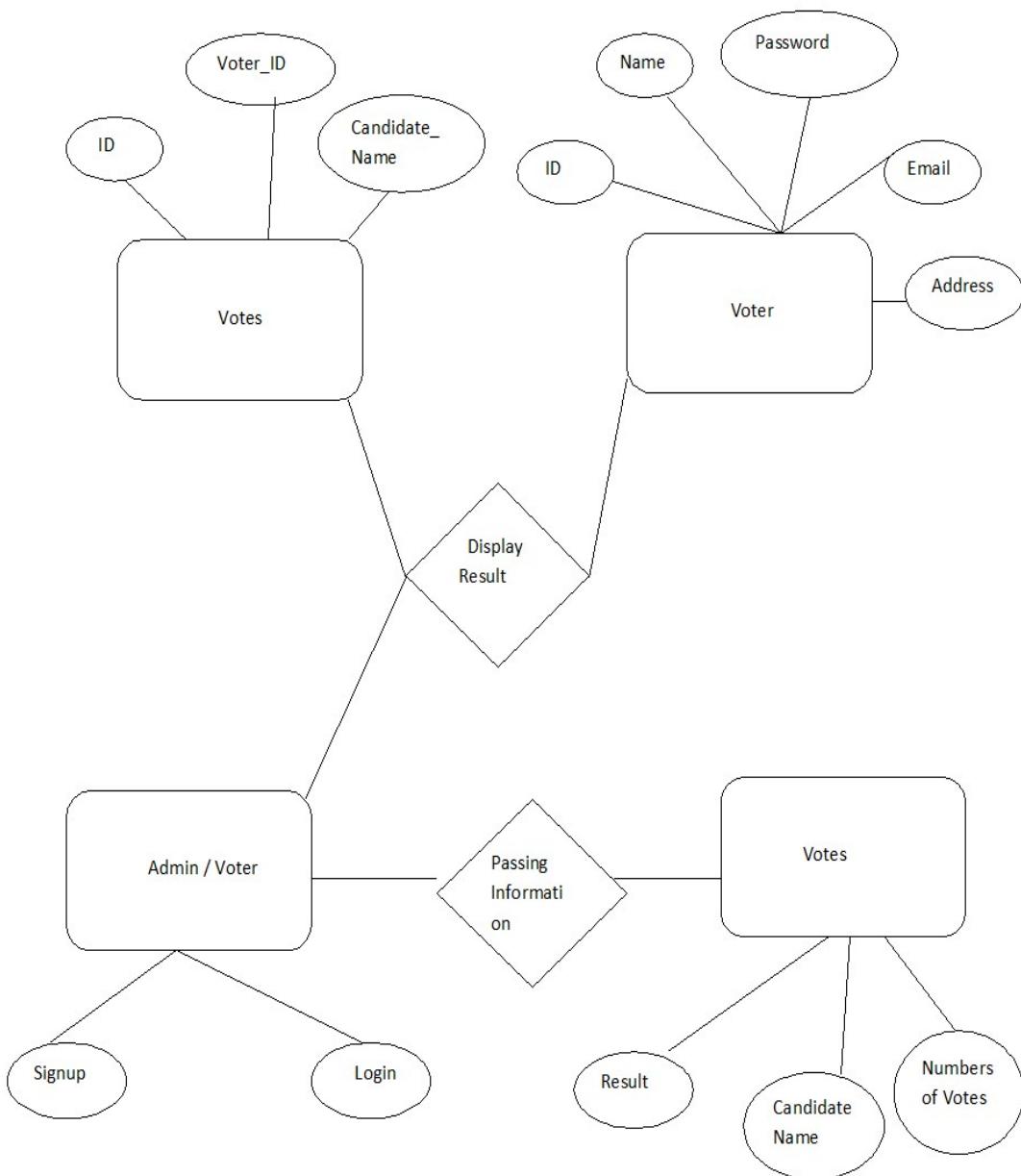


In summary, the *Input and Output Designs* are crucial for ensuring smooth interaction between users and the system, ensuring both input validation and clear, timely feedback at every step of the voting process.

## System Diagram

- **Data Flow Diagram**

- Entity Relationship Diagram



- UML

## CODE:

### app.py

```
from flask import Flask, render_template, request, redirect, url_for, session, flash, jsonify
from werkzeug.security import generate_password_hash, check_password_hash
import mysql.connector

# Initialize Flask app
app = Flask(__name__)
app.secret_key = 'your_secret_key'

# Import database configuration
from database_config import db_connection

# Candidate List
CANDIDATES = [
    "Ankita", "Aarya", "Abhijeet", "Arbaaz", "Dhanananjay", "Ghanshyam",
    "Irina", "Janhvi", "Nikhil", "Nikki", "Pandarinath", "Suraj",
    "Vaibhav", "Varsha", "Yogita"
]

@app.route('/')
def home():
    return render_template('home.html')

# Sign-up Route
@app.route('/index', methods=['GET', 'POST'])
def signup():
    if request.method == 'POST':
        voter_name = request.form['votername']
        password = request.form['password']
        email = request.form['email']
        address = request.form['address']

        # Hash the password before storing it
        hashed_password = generate_password_hash(password)

        cursor = db_connection.cursor()
        cursor.execute(
            "INSERT INTO voter (name, password, email, address) VALUES (%s, %s, %s, %s)",
            (voter_name, hashed_password, email, address)
        )
        db_connection.commit()
        cursor.close()

        flash("Sign-up successful! Please login.")
        return redirect(url_for('login'))

    return render_template('index.html')
```

```

# Login Route
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        voter_name = request.form['votername']
        password = request.form['password']

        cursor = db_connection.cursor()
        cursor.execute(
            "SELECT * FROM voter WHERE name=%s", (voter_name,))
    )
    voter = cursor.fetchone()
    cursor.close()

    if voter and check_password_hash(voter[2], password): # Verify password
        session['voter_id'] = voter[0]
        flash("Login successful!")
        return redirect(url_for('candidates'))
    else:
        flash("Invalid votername or password.") # Clear error message

    return render_template('login.html')

# List of Candidates
@app.route('/candidates')
def candidates():
    return render_template('candidates.html', candidates=CANDIDATES)

# Profile Route with Voting Logic
@app.route('/profile/<candidate_name>', methods=['GET', 'POST'])
def profile(candidate_name):
    if 'voter_id' not in session: # Ensure user is logged in
        flash("Please log in to vote.")
        return redirect(url_for('login'))

    voter_id = session['voter_id']

    # Check if the voter has already voted
    cursor = db_connection.cursor()
    cursor.execute("SELECT candidate_name FROM votes WHERE voter_id=%s", (voter_id,))
    vote = cursor.fetchone()
    cursor.close()

    if vote:
        flash(f"You have already voted for {vote[0]}." ) # Display a flash message
        return redirect(url_for('candidates')) # Redirect to candidates list with flash

    if request.method == 'POST':
        # Insert the vote if not already voted
        cursor = db_connection.cursor()
        cursor.execute(
            "INSERT INTO votes (voter_id, candidate_name) VALUES (%s, %s)",

```

```

        (voter_id, candidate_name)
    )
db_connection.commit()
cursor.close()

flash(f"You have voted for {candidate_name} successfully!")
return redirect(url_for('vote_success', candidate_name=candidate_name))

return render_template('profile.html', candidate_name=candidate_name)
@app.route('/vote_success/<candidate_name>')
def vote_success(candidate_name):
    return render_template('vote_success.html', candidate_name=candidate_name)

# Logout Route to Display Flash Message
@app.route('/logout')
def logout():
    session.clear() # Clear the session
    flash("You have been logged out.")
    return render_template('logout.html')

@app.route('/get_winner_data')
def get_winner_data():
    cursor = db_connection.cursor()

    # Fetch total number of voters
    cursor.execute("SELECT COUNT(*) FROM voter")
    total_voters = cursor.fetchone()[0]

    # Fetch the winner
    cursor.execute(
        "SELECT candidate_name, COUNT(*) AS votes FROM votes GROUP BY candidate_name ORDER BY
votes DESC LIMIT 1"
    )
    winner = cursor.fetchone()

    # Fetch all candidates and their votes
    cursor.execute(
        "SELECT candidate_name, COUNT(*) AS votes FROM votes GROUP BY candidate_name ORDER BY
votes DESC"
    )
    candidates = cursor.fetchall()

    result_data = {
        "total_voters": total_voters,
        "winner": {"name": winner[0], "votes": winner[1]},
        "candidates": [{"name": candidate[0], "votes": candidate[1]} for candidate in
candidates]
    }

    cursor.close()

    return render_template('get_winner_data.html', result_data=result_data)

```

```
if __name__ == '__main__':
    app.run(debug=True)
```

## HOME:

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bigg Boss Home</title>

    <!-- Link to external CSS for additional styling -->
    <link rel="stylesheet" href="/static/css/styles.css">

<style>
    /* Set the background image and style for the entire page */
    body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        height: 100vh;
        background-image: url('/static/images/Big-Boss.jpg');
        background-size: cover;
        background-position: center;
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        color: gold;
    }

    /* Apply a semi-transparent overlay to darken the background for readability */
    .overlay {
        background-color: gold(0, 0, 0, 0.6); /* Black with 60% opacity */
        position: absolute;
        top: 0;
        left: 0;
        right: 0;
        bottom: 0;
        z-index: 1;
    }

    /* Container for the content to ensure it's centered */
    .content {
```

```
        position: relative;
        z-index: 2;
        padding: 20px;
    }

    /* Header title styling */
    h2 {
        font-size: 3rem;
        margin-bottom: 20px;
    }

    p {
        font-size: 1.2rem;
        margin-bottom: 30px;
    }

    /* Navigation link styling */
    .nav-links a {
        text-decoration: none;
        font-size: 1.2rem;
        padding: 10px 20px;
        margin: 5px;
        background-color: #ff6f61;
        color: white;
        border-radius: 5px;
        transition: background-color 0.3s;
    }

    .nav-links a:hover {
        background-color: #e55a4e;
    }

    /* Responsive adjustments for smaller screens */
    @media (max-width: 600px) {
        h2 {
            font-size: 2rem;
        }

        p {
            font-size: 1rem;
        }

        .nav-links a {
            display: block;
            width: 80%;
            margin: 10px auto;
        }
    }

```

</style>

</head>

<body>

<!-- Overlay to darken the background image -->

```

<div class="overlay"></div>

<!-- Flash messages if any --&gt;
{% with messages = get_flashed_messages() %}
    {% if messages %}
        &lt;ul&gt;
            {% for message in messages %}
                &lt;li style="color: red; z-index: 2;"&gt;{{ message }}&lt;/li&gt;  <!-- Customize style as
needed --&gt;
            {% endfor %}
        &lt;/ul&gt;
    {% endif %}
    {% endwith %}

<!-- Main Content --&gt;
&lt;div class="content"&gt;
    <!-- Welcome Message --&gt;
    &lt;h2&gt;Welcome to Bigg Boss Voting&lt;/h2&gt;
    &lt;p&gt;Vote for your favorite contestant and make them the winner of the Bigg Boss
season!&lt;/p&gt;

    <!-- Navigation Links --&gt;
    &lt;div class="nav-links"&gt;
        &lt;a href="/index"&gt;Sign Up to Vote&lt;/a&gt;
    &lt;/div&gt;
&lt;/div&gt;

&lt;/body&gt;
&lt;/html&gt;
</pre>

```

## INDEX (Signup)

```

<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="/static/css/styles.css">
    <title>Sign Up</title>
    <style>
        body {
            background-image: url('static/images/Boss.jpg'); /* Replace with your image path
*/
            background-size: cover; /* Makes the image cover the entire page */
            background-position: center; /* Centers the image */
            background-repeat: no-repeat; /* Prevents the image from repeating */
            height: 100vh; /* Full viewport height */
            margin: 30;
        }
    </style>
</head>

```

```

<body>
    {% with messages = get_flashed_messages() %}
        {% if messages %}
            <ul>
                {% for message in messages %}
                    <li style="color: red;">{{ message }}</li>  
                {% endfor %}
            </ul>
        {% endif %}
    {% endwith %}
    <h2>Sign Up</h2>
    <form method="POST">
        <input type="text" name="votername" placeholder="Voter Name" required><br>
        <input type="password" name="password" placeholder="Password" required><br>
        <input type="email" name="email" placeholder="Email Address" required><br>
        <input type="text" name="address" placeholder="Address" required><br>
        <button type="submit">Sign Up</button>
    </form>
    <b>Already have an account? <a href="/login">Login here</a>.</b>
</body>
</html>

```

## Login:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>

    <!-- Link to external CSS for additional styling -->
    <link rel="stylesheet" href="/static/css/styles.css">

<style>
    /* Set the background image and style for the entire page */
    body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        height: 100vh;
        background-image: url('/static/images/Big-Boss.jpg');
        background-size: cover;
        background-position: center;
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        color: white;
    }

```

```
}

/* Apply a semi-transparent overlay to darken the background for readability */
.overlay {
    background-color: rgba(0, 0, 0, 0.6); /* Black with 60% opacity */
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    z-index: 1;
}

/* Container for the content to ensure it's centered */
.content {
    position: relative;
    z-index: 2;
    padding: 20px;
}

h2 {
    font-size: 2.5rem;
    margin-bottom: 20px;
}

/* Styling the login form */
form {
    display: inline-block;
    background-color: rgba(255, 255, 255, 0.8); /* White background with some transparency */
    padding: 20px;
    border-radius: 10px;
}

form input {
    margin: 10px 0;
    padding: 10px;
    font-size: 1rem;
    border-radius: 5px;
    border: 1px solid #ccc;
    width: 100%;
}

form button {
    padding: 10px 20px;
    font-size: 1rem;
    border-radius: 5px;
    background-color: #ff6f61;
    color: white;
    border: none;
    cursor: pointer;
    transition: background-color 0.3s;
}
```

```

        }

    form button:hover {
        background-color: #e55a4e;
    }

    /* Link styling */
    p a {
        color: #ff6f61;
        text-decoration: none;
    }

    p a:hover {
        text-decoration: underline;
    }

    /* Responsive adjustments for smaller screens */
    @media (max-width: 600px) {
        h2 {
            font-size: 2rem;
        }

        form {
            width: 80%;
        }
    }

```

</style>

</head>

<body>

<!-- Overlay to darken the background image --&gt;</p>

</div>

<!-- Main Content --&gt;</p>

<h2>Login</h2>

<!-- Flash messages if any --&gt;</p>

{%- with messages = get\_flashed\_messages() %}

{%- if messages %}

{%- for message in messages %}

<li style="color: red;">{{ message }}</li> <!-- Customize style as needed --&gt;</p>

{%- endfor %}

</ul>

{%- endif %}

{%- endwith %}

<!-- Login Form --&gt;</p>

<input type="text" name="votername" placeholder="Voter Name" required><br>

<input type="password" name="password" placeholder="Password" required><br>

<button type="submit">Login</button>

34 | Page

```
</form>

<p>Don't have an account? <a href="/">Sign up here</a>.</p>
</div>

</body>
</html>
```

## CANDIDATES:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Candidates</title>

    <!-- Link to external CSS for additional styling -->
    <link rel="stylesheet" href="/static/css/styles.css">

    <style>
        /* Overall page styling */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
            text-align: center;
            color: #333;
        }

        /* Heading style */
        h2 {
            font-size: 2.5rem;
            margin-bottom: 20px;
            color: #ff6f61;
        }

        /* Candidate grid styling */
        ul {
            display: grid;
            grid-template-columns: repeat(auto-fill, minmax(200px, 1fr)); /* Responsive grid
        */
            gap: 20px;
            padding: 0;
            list-style-type: none;
            justify-items: center;
        }
    </style>

```

```
/* Candidate list item styling */
ul li {
    background-color: white;
    border-radius: 10px;
    padding: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s ease, box-shadow 0.3s ease;
    text-align: center;
    width: 75%;
    max-width: 250px;
}

ul li:hover {
    transform: translateY(-10px);
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
}

/* Image styling */
ul li img {
    width: 100%;
    border-radius: 50%;
    margin-bottom: 15px;
    transition: transform 0.3s ease;
}

ul li img:hover {
    transform: scale(1.1);
}

/* Candidate name styling */
ul li p {
    font-size: 1.2rem;
    color: #555;
    margin: 10px 0;
}

/* Profile link button styling */
ul li a {
    text-decoration: none;
    font-size: 1rem;
    color: white;
    background-color: #93f40a;
    padding: 10px 20px;
    border-radius: 5px;
    display: inline-block;
    transition: background-color 0.3s;
}

ul li a:hover {
    background-color: #e55a4e;
}
```

```

/* Navigation links */
.nav-links {
    margin-top: 20px;
}

.nav-links a {
    text-decoration: none;
    font-size: 1rem;
    color: #ff6f61;
    margin: 0 10px;
    transition: color 0.3s;
}

.nav-links a:hover {
    color: #e55a4e;
}

/* Responsive styling for mobile */
@media (max-width: 600px) {
    h2 {
        font-size: 2rem;
    }

    ul {
        grid-template-columns: repeat(auto-fill, minmax(150px, 1fr));
    }

    ul li {
        max-width: 200px;
    }

    ul li img {
        width: 80%;
    }
}

```

</style>

</head>

<body>

**Candidates List**

!-- Flash Messages -->

{% with messages = get\_flashed\_messages() %}

{% if messages %}

<ul>

{%- for message in messages %}

<li style="color: red;">{{ message }}</li>

{%- endfor %}

</ul>

{%- endif %}

{%- endwith %}

!-- Candidates List -->

```

<ul>
    {% for candidate in candidates %}
        <li>
            
            <p>{{ candidate }}</p>
            <a href="/profile/{{ candidate }}">View Profile</a>
        </li>
    {% endfor %}
</ul>

<!-- Navigation Links --&gt;
&lt;div class="nav-links"&gt;
    &lt;a href="/"&gt;Back to Home&lt;/a&gt; | &lt;a href="/logout"&gt;Logout&lt;/a&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

## PROFILE:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/static/css/styles.css">
    <title>{{ candidate_name }}'s Profile</title>

    <style>
        /* Basic body styling */
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
            margin: 0;
        }

        /* Profile card styling */
        .profile-card {
            background-color: white;
            width: 90%;
            max-width: 800px;
            border-radius: 15px;
            padding: 20px;
            box-shadow: 0 8px 16px rgba(0, 0, 0, 0.1);
            text-align: center;
        }
    </style>

```

```
        position: relative;
        display: flex;
        flex-direction: column;
        align-items: center;
    }

    /* Profile image styling */
    .profile-img {
        width: 150px;
        height: 150px;
        border-radius: 50%;
        object-fit: cover;
        margin-bottom: 20px;
    }

    /* Candidate name styling */
    h2 {
        font-size: 2rem;
        color: #333;
        margin-bottom: 10px;
    }

    /* Info paragraph styling */
    .candidate-info {
        font-size: 1.1rem;
        color: #666;
        margin: 10px 0;
        padding: 0 20px;
        line-height: 1.6;
    }

    /* Social media icon container */
    .social-media {
        display: flex;
        justify-content: center;
        margin: 20px 0;
    }

    /* Social media icon styling */
    .social-media a {
        margin: 0 10px;
    }

    .social-media img {
        width: 40px;
        transition: transform 0.3s;
    }

    .social-media img:hover {
        transform: scale(1.1);
    }
```

```

/* Vote button styling */
.vote-button {
    padding: 10px 20px;
    background-color: #c3e217;
    color: white;
    border: none;
    border-radius: 5px;
    font-size: 1rem;
    cursor: pointer;
    transition: background-color 0.3s;
    margin-top: 20px;
}

.vote-button:hover {
    background-color: #e55a4e;
}

/* Navigation links */
.nav-links {
    margin-top: 20px;
}

.nav-links a {
    text-decoration: none;
    font-size: 1rem;
    color: #ff6f61;
    margin: 0 10px;
    transition: color 0.3s;
}

.nav-links a:hover {
    color: #e55a4e;
}

```

</style>

</head>

<body>

<div class="profile-card">

<!-- Candidate Profile Image -->



<!-- Candidate Name -->

<h2>{{ candidate\_name }}</h2>

<!-- Candidate Info -->

<p class="candidate-info">

{{ candidate\_name }} is a talented contestant who has captured the hearts of many.  
 Known for their incredible charm, wit, and personality, they have become a fan favorite in the  
 Bigg Boss house. Stay tuned to see how far they go!
 </p>

<!-- Social Media Links with icons -->

```

<div class="social-media">
    <a href="https://instagram.com/{{ candidate_name | lower }}" target="_blank">
        
    </a>
    <a href="https://facebook.com/{{ candidate_name | lower }}" target="_blank">
        
    </a>
</div>

<!-- Vote Button -->
{%
    if session.get('voter_id') %}
    <form method="POST">
        <button type="submit" class="vote-button">Vote for {{ candidate_name }}</button>
    </form>
{%
    else %}
    <p>Please log in to vote.</p>
{%
    endif %}

<!-- Navigation Links -->
<div class="nav-links">
    <a href="/candidates">Back to Candidates</a> | <a href="/logout">Logout</a>
</div>
</div>
</body>
</html>

```

## SUCCESS VOTE CARD:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Vote Success</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">
    <style>
        body {
            background-color: #f4f4f4;
            font-family: 'Arial', sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }
        .thank-you-card {

```

```
background-color: #fff;
border-radius: 15px;
padding: 20px;
box-shadow: 0 8px 16px rgba(0, 0, 0, 0.1);
max-width: 600px;
text-align: center;
margin: 20px;
}

.thank-you-card img {
    border-radius: 50%;
    width: 150px;
    height: 150px;
    object-fit: cover;
    margin-bottom: 20px;
}

.thank-you-message {
    font-size: 1.2rem;
    color: #333;
    margin-bottom: 15px;
}

.redirect-message {
    font-size: 1rem;
    color: #666;
    margin-top: 15px;
}

.logout-button {
    background-color: #4ef307;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
    font-size: 1rem;
}

.logout-button:hover {
    background-color: #e55a4e;
}

/* Success Icon */
.success-icon {
    color: #28a745;
    font-size: 3rem;
    margin-bottom: 20px;
    animation: pop 0.6s ease-in-out;
}
```

```

        @keyframes pop {
            0% {
                transform: scale(0);
            }
            100% {
                transform: scale(1);
            }
        }
    </style>
</head>
<body>
    <div class="thank-you-card">
        <!-- Success Icon -->
        <div class="success-icon">
            <i class="bi bi-check-circle-fill"></i>
        </div>

        <!-- Candidate Image -->
        

        <!-- Thank You Message -->
        <h2>Thank You for Voting!</h2>
        <p class="thank-you-message">
            {{ candidate_name }} sends their heartfelt thanks for your vote! Your support
            means everything.
        </p>

        <!-- Redirect Message -->
        <p class="redirect-message">
            You are now being redirected to the logout page.
        </p>

        <!-- Logout Button -->
        <form method="GET" action="{{ url_for('logout') }}">
            <button type="submit" class="logout-button">Logout</button>
        </form>
    </div>

    <!-- Bootstrap JS (optional) -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## WINNER:

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Bigg Boss Marathi - Winner</title>
<!-- Bootstrap CSS -->
<link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
    body {
        background-image: url('/static/images/bigg_boss_marathi_logo.jpg');
        background-size: cover;
        background-position: center;
        color: rgb(220, 239, 11);
        text-align: center;
    }
    .winner-card {
        background-color: rgba(0, 0, 0, 0.7);
        padding: 20px;
        border-radius: 10px;
        margin-top: 50px;
        color: gold(124, 22, 233);
    }
    .candidate-list {
        margin-top: 20px;
    }
    .candidate-list li {
        list-style: none;
        padding: 10px;
        background-color: rgba(237, 20, 168, 0.1);
        border: 1px solid rgb(15, 211, 64);
        border-radius: 5px;
        margin-bottom: 10px;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="winner-card">
            <h1>Bigg Boss Marathi - Winner Announcement</h1>
            <h2>Congratulations to <strong>{{ result_data.winner.name }}</strong>!</h2>
            <p>{{ result_data.winner.name }} received <strong>{{ result_data.winner.votes }}</strong> votes!</p>
        </div>

        <div class="candidate-list">
            <h3>Total Voters: {{ result_data.total_voters }}</h3>
            <h4>Votes per Candidate:</h4>
            <ul>
                {% for candidate in result_data.candidates %}
                    <li>
                        <strong>{{ candidate.name }}</strong> - {{ candidate.votes }} votes
                    </li>
                {% endfor %}
            </ul>
        </div>
    </div>
</body>

```

```

        </ul>
    </div>

    <a href="/logout" class="btn btn-danger">Logout</a>
</div>

<!-- Bootstrap JS and dependencies --&gt;
&lt;script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"&gt;&lt;/script&gt;
&lt;script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

## GENERAL CSS FILE CODE:

```

/* General Styles */
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
    margin: 0;
    padding: 0;
    text-align: center;
}

h2 {
    color: #333;
    margin-top: 20px;
}

/* Form Styles */
form {
    margin: 20px auto;
    width: 300px;
    padding: 15px;
    background-color: gold(0, 0, 0, 0.6);
    box-shadow: 0px 0px 10px gold (0, 0, 0, 0.1);
    border-radius: 8px;
}

input[type="text"], input[type="password"], input[type="email"] {
    width: 90%;
    padding: 10px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 4px;
}

button {
    background-color: #ff5733;
}

```

```
        color: white;
        border: none;
        padding: 10px 20px;
        cursor: pointer;
        border-radius: 5px;
    }

button:hover {
    background-color: #c73c22;
}

/* Navigation Links */
a {
    color: #ff5733;
    text-decoration: none;
    margin: 10px;
}

a:hover {
    text-decoration: underline;
}

/* Candidate List Styles */
ul {
    list-style-type: none;
    padding: 0;
}

ul li {
    display: inline-block;
    margin: 15px;
    text-align: center;
}

ul li img {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    object-fit: cover;
    border: 2px solid #333;
    margin-bottom: 10px;
}

/* Profile Page */
.profile-container {
    max-width: 600px;
    margin: 20px auto;
    background-color: #fff;
    padding: 20px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
}
```

```
.profile-container img {
    width: 200px;
    height: 200px;
    border-radius: 50%;
    margin: 20px 0;
    object-fit: cover;
}

ul.social-links {
    list-style-type: none;
    padding: 0;
}

ul.social-links li {
    display: inline-block;
    margin: 0 10px;
}

.social-links a {
    text-decoration: none;
    color: #3b5998;
    font-size: 1.2em;
}

/* Logout Page */
.logout-container {
    margin-top: 50px;
    font-size: 1.2em;
}

/* Responsive Design */
@media (max-width: 600px) {
    ul li {
        display: block;
        margin: 15px auto;
    }

    form {
        width: 90%;
    }
}
```

## **Used Platforms**

- **Existing System:** The current system of voting for *Bigg Boss Marathi* is primarily done through official websites or mobile apps, where users register and cast their votes for contestants. However, our system brings this experience to a more streamlined and user-friendly web-based interface, ensuring easy registration, login, and secure voting options.
- **FrontEnd – HTML, CSS, JavaScript:** The frontend is designed using **HTML, CSS, and JavaScript** to create an interactive and responsive user interface. This ensures a seamless experience for users across different devices. HTML provides the structure of the website, CSS ensures its visual appeal, and JavaScript adds dynamic functionality, such as form validation and interactive elements.
- **BackEnd – Flask in Python, MySQL:**
  - The backend is powered by **Flask**, a lightweight Python framework, which handles the API logic, data processing, and communication between the frontend and the database.
  - **MySQL** is used to store data securely, including user credentials, contestant information, and voting details. The system also employs MySQL connectors to link the backend logic with the database.
  - **Encryption** tools like **bcrypt** are used for securely storing user passwords.

## **System Requirements**

1. **Hardware:**
  - Processor: Minimum Intel i3 or AMD equivalent.
  - RAM: 4GB or more.
  - Hard Disk: 20GB free space.
  - Internet Connection: Required for system operation.

## 2. Software:

- Operating System: Windows, macOS, or Linux.
  - Web Browser: Chrome, Firefox, or any modern browser.
  - Python 3.x (Flask framework).
  - MySQL database.
  - Text editor/IDE (e.g., Visual Studio Code, PyCharm).
- 

## Conclusion & Suggestions

### *Future Enhancement*

- **Mobile App Integration:** Future versions could include a mobile app that offers real-time notifications for voting cycles and contestant updates.
- **Advanced Security Features:** Implementing multi-factor authentication (MFA) for user login to enhance security.
- **Enhanced User Analytics:** Providing admins with more detailed analytics, such as voting trends over time.

### *Silent Features*

- **User-Friendly Interface:** The system is easy to navigate with a clean, intuitive layout for all users.
- **Secure Voting Process:** The use of encrypted passwords and vote validation ensures the security and fairness of the voting process.
- **Real-Time Voting Results:** Users and administrators can view live voting data, ensuring transparency and engagement.

### *Suggestions*

- **Expand Voting Limits:** Consider allowing users to vote more than once in specific scenarios, such as during "Double Voting" periods.
- **Include Social Media Sharing:** Adding social media sharing options can increase user engagement by letting them share their vote on platforms like Facebook and Twitter.

- **Improved Candidate Profiles:** Enhancing contestant profiles with videos, social media links, and more detailed descriptions to engage voters.
- 

## Bibliography

1. Flask Documentation: <https://flask.palletsprojects.com/>
2. MySQL Documentation: <https://dev.mysql.com/doc/>
3. HTML, CSS, and JavaScript Tutorials: <https://www.w3schools.com/>
4. Bcrypt Encryption: <https://pypi.org/project/bcrypt/>
5. Python Official Documentation: <https://docs.python.org/>