Case Study: Digital Dynamic Voting System (DDVS) for Local Elections

---

Business Scenario:
Electown's town council is on a mission to transition from its traditional voting mechanisms. As the number of residents grows and the digital age demands more, the vision for a transparent, secure, and user-friendly Digital Dynamic Voting System (DDVS) becomes paramount.

Objectives:

1. **Scalability**: Facilitate any number of candidates and voters.
2. **Digital Accessibility**: Seamless, secure, and transparent digital voting.
3. **Transparency**: Real-time statistics and insights into voting patterns.
4. **Security**: Authenticate voters and prevent voting fraud.

Functional Requirements:

1. **Candidate Management**:
    - Register candidates with a unique ID, full name, vote count, voter logo, and photo.
    - Authorized personnel can add, modify, or delete candidates.
    - Approved format and size for photos and logos.
2. **Voting Mechanism**:
    - Voters provide their unique Voter ID to vote.
    - Voter IDs are authenticated to prevent duplicate voting.
    - Live update of candidate vote counts.
    - Voting confirmation for voters.
3. **User Interface & Accessibility**:
    - Dynamic presentation of all candidates with their full names, photos, and logos.
    - An intuitive voting process.
    - Compatible across multiple devices.
    - Features tailored for the differently-abled.
4. **Reporting & Analytics**:
    - Live vote counts for every candidate.
    - Total votes cast.
    - Reports in various formats (CSV, PDF).
    - Visualizations showcasing voting patterns.
5. **Security & Authentication**:
    - Voter registration and authentication using unique Voter IDs.
    - Detect and deter fraudulent voting activities.
    - Data encryption.

---

6. Detailed Functional requirements.

FR ID	Category	Description (with examples)
FR1.1	Candidate Management	Register candidates with details. **Example**: Authorized user can add "John Doe" with a candidate ID "JD123", photo of him smiling, a "Peace" logo, and an initial vote count of 0.
FR1.2	Candidate Management	Update any details of an existing candidate. **Example**: Changing John Doe's photo to a newer one or updating his logo.
FR1.3	Candidate Management	Remove candidates. **Example**: If John Doe withdraws, his data is purged from the system.
FR1.4	Candidate Management	Define and control image specifications. **Example**: Uploaded images must be .JPEG or .PNG, with a max size of 5MB. The system offers tools to crop the image to 500x500 px.
FR1.5	Candidate Management	View all candidates. **Example**: An admin can see a list of all candidates with their details and photos in a dashboard.
FR2.1	Voter Authentication & Management	Register voters. **Example**: "Alice" registers, providing her personal details and receives a Voter ID "A7890".
FR2.2	Voter Authentication & Management	Voter login. **Example**: Alice logs in using her Voter ID "A7890" before voting.
FR2.3	Voter Authentication & Management	Access voting history. **Example**: Alice can see she voted for John Doe in the last election but can't alter this data.
FR3.1	Voting Mechanism	Voting interface displays. **Example**: Alice sees a list of candidates with their photos, logos, and a button to vote next to each.
FR3.2	Voting Mechanism	Restrict one vote per Voter ID. **Example**: Once Alice votes for a candidate, she can't cast another vote with her ID "A7890".
FR3.3	Voting Mechanism	Confirm voting action. **Example**: After voting, Alice sees a message, "Thank you for voting for John Doe!".
FR3.4	Voting Mechanism	End voting session. **Example**: Voting is open from 8 AM to 5 PM, post which the system doesn't accept votes.
FR4.1	User Interface & Accessibility	Device-friendly platform. **Example**: Alice can vote using her smartphone, tablet, or desktop seamlessly.
FR4.2	User Interface & Accessibility	Efficient search. **Example**: Alice types "John" into a search bar and quickly finds John Doe among the candidates.
FR4.3	User Interface & Accessibility	Multi-language option. **Example**: Alice can switch the platform language to Spanish if she prefers.
FR4.4	User Interface & Accessibility	Comply with guidelines like WCAG. **Example**: The site has voice-over capabilities for visually impaired users.
FR5.1	Reporting & Analytics	Live vote counts. **Example**: Admins can view a live count of votes each candidate has received.
FR5.2	Reporting & Analytics	Various report formats. **Example**: Admins can download a report of votes in a .CSV format.

FR ID	Category	Description (with examples)
FR5.3	Reporting & Analytics	Understand voting trends. **Example**: Graphs show a spike in voting at lunchtime.
FR5.4	Reporting & Analytics	Log all actions. **Example**: A log entry is created when Alice votes, capturing time and choice.
FR6.1	Security & Authentication	Data protection. **Example**: Alice's details are encrypted and cannot be read if intercepted.
FR6.2	Security & Authentication	Security layers. **Example**: Alice must solve a CAPTCHA before voting to prove she isn't a bot.
FR6.3	Security & Authentication	Efficient session handling. **Example**: After 15 minutes of inactivity, Alice's session expires and she needs to re-login.
FR6.4	Security & Authentication	Detailed activity tracking. **Example**: All of Alice's interactions with the system, from login to voting, are logged with timestamps.
FR7.1	Notifications & Alerts	Confirmation post-vote. **Example**: Alice receives an email confirming her vote for John Doe.
FR7.2	Notifications & Alerts	Monitor irregularities. **Example**: Admins receive an alert if there's a sudden surge in votes for a candidate, indicating potential fraud.
FR7.3	Notifications & Alerts	Voting reminders. **Example**: Alice receives an SMS reminder to vote if she hasn't by 4 PM.
FR8.1	Help & Support	Address common issues. **Example**: An FAQ section answers "How do I register?" or "What if I can't login?".
FR8.2	Help & Support	Support channels. **Example**: Alice can chat with a support agent if she faces issues during voting.
FR8.3	Help & Support	Collect feedback. **Example**: After voting, Alice can rate her experience or provide suggestions.

Non-Functional Requirements:

1. **Performance**: Fast response times with capacity for numerous concurrent users.
2. **Reliability**: Consistent uptime during voting periods and regular data backups.
3. **Security**: Adherence to cybersecurity best practices and regulations.
4. **Scalability**: Infrastructure that can accommodate growth in voters and candidates.
5. **Usability**: Intuitive design with regular integration of feedback.

Technical Stack:

- **Frontend**: Angular (latest version).
- **State Management**: NgRx.
- **Backend**: Node.js, Express.js.
- **Database**: PostgreSQL or MongoDB.
- **Image Storage**: Cloud solutions like Amazon S3 or Google Cloud Storage.
- **UI Testing**: Tools like Jasmine and Protractor.

UI Testing:
Assuring the reliability and usability of the system demands rigorous UI testing:

1. **Unit Tests**:
    - Examine individual components/functions.
    - **Example**: Verify that the candidate component displays the name, logo, and photo accurately.
2. **Integration Tests**:
    - Gauge how different application parts cooperate.
    - **Example**: Confirm that casting a vote updates the vote tally and registers the Voter ID.
3. **End-to-End (E2E) Tests**:
    - Assess the application as a whole.
    - **Example**: Ensure the entire process, from Voter ID entry to vote confirmation, operates without hitches.
4. **Accessibility Testing**:
    - Verify system use by differently-abled individuals.
    - **Example**: Ensure compatibility with screen readers and sufficient visual contrast.
5. **Image & Logo Display Tests**:
    - Ensure all candidate images and logos display correctly without distortion.
    - Test image and logo upload functionalities for different file formats and sizes.

Stakeholders:

1. **Voters**: Participate using their unique Voter IDs.
2. **Election Commission**: Supervise the process and access analytics.
3. **Candidates**: Compete and view their real-time vote tally.
4. **IT Support Team**: Maintain smooth operation and address issues.

Challenges & Considerations:

1. **Digital Divide**: All residents may not be digitally inclined.
2. **Training**: Transition from traditional to digital methods may require guidance.
3. **Security**: Safeguarding Voter IDs and preserving the sanctity of each vote.

Conclusion:
The Digital Dynamic Voting System (DDVS) embodies the future for Electown, combining advanced technology with a user-first approach. With a focus on transparency, security, and usability, DDVS is poised to revolutionize the local electoral process. Proper implementation, bolstered by meticulous testing, ensures that the DDVS will stand as a beacon of progress in the digital age.