**Slide 1: Title Slide**

* **Project Title:** Simple Voting System
* **Students:**
* Thaliyakula Likitha (24KB1A05KW)
* Thoppani Tejasri (24KB1A05LF)
* Tupakula Sarika (24KB1A05LN)
* Sulluru Tejasri (24KB1A05KD)
* **Guide:** Padavala Suneetha
* **Institution:** N.B.K.R Institute of Science and Technology, Department of CSE

**Slide 2: Abstract**

* Computerized system to facilitate fair and efficient elections
* Secure, single-vote-per-user, and error-free results
* Ideal for small-scale organizations

**Slide 3: Introduction**

* Traditional methods are time-consuming and error-prone
* Duplicate voting and delays common
* Need for reliable and user-friendly e-voting

**Slide 4: Objectives**

* Digitize voting for accuracy and speed
* Ensure one vote per user
* Provide easy admin and user interface
* Enable real-time result viewing

**Slide 5: Literature Survey**

* Rise of e-voting in academic and organizational use
* Paper-based vs console-based voting comparison
* Benefits: Transparency, accuracy, security

**Slide 6: Requirements**

* **Functional:**
  + Voter Authentication
  + Vote Casting
  + Candidate Management
  + Result Viewing
* **Non-functional:**
  + Usability
  + Reliability
  + Security

**Slide 7: Design – Architecture**

* Client-server model
* Console UI
* Backend logic for validation and result tally
* File-based data storage

**Slide 8: Design – Components**

* **Voter Module**: Login, vote once
* **Admin Module**: Manage candidates and results
* **Storage**: File system for persistent data

**Slide 9: Data Flow Diagram**

* **Voter Flow**: Login → Vote → Record → Update Result
* **Admin Flow**: Login → Manage Candidates → View Results

**Slide 10: User Interface**

* **Main Menu**: Voter login, Admin login, Exit
* **Voter Menu**: Cast vote, Logout
* **Admin Menu**: Add/Remove candidate, View results, Logout

**Slide 11: Flowcharts**

* **Login Process**: Credential validation → Access menus
* **Voting**: Select candidate → Record vote
* **Admin**: Manage candidates and view results

**Slide 12: Coding (Overview)**

* Language: C
* Structures: Candidate, Voter
* Functions: addCandidate(), addVoter(), displayCandidates(), displayVoters()

**Slide 13: Sample Code Snippet**

c

void addCandidate(int id, const char\* name) {

candidates[candidateCount].candidateId = id;

strcpy(candidates[candidateCount].name, name);

candidates[candidateCount].votes = 0;

candidateCount++;

}

**Slide 14: Testing**

* **Black Box**: Voter casts vote → Vote recorded
* **White Box**: Input validation for candidate/voter IDs

**Slide 15: Output Screens**

* Login Screen
* Voting Interface
* Result Display (console mockups/screenshots)

**Slide 16: Conclusion & Future Scope**

* Functional system for small-scale elections
* **Future Enhancements**:
  + GUI Interface
  + Database Integration
  + Biometric/OTP-based Authentication
  + Web-based Access

**Slide 17: References**

* C Programming by Dennis Ritchie
* GeeksforGeeks
* Tutorialspoint
* Instructor’s Notes