MAHATMA EDUCATION SOCIETY'S

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE
(Autonomous)

NEW PANVEL

PROJECT REPORT ON

"Secure Voting System"

IN PARTIAL FULFILLMENT OF

BACHELOR OF INFORMATION TECHNOLOGY

SEMESTER VI – 2024-25

PROJECT GUIDE

Name : Prof. Shrutika Ma'am

SUBMITTED BY: MEHTAB SINGH RANDHAWA

ROLL NO: 9524

Introduction

In today's digital era, elections play a crucial role in upholding democracy, and ensuring a secure and transparent voting process is of utmost importance. Traditional voting methods often suffer from issues such as voter fraud, ballot tampering, and inefficient counting processes. To address these concerns, the **Secure Voting System** has been developed to provide a reliable, encrypted, and fraud-resistant platform for conducting elections.

This project leverages cryptographic techniques such as **Substitution Cipher, Rail Fence Cipher, and AES Encryption** to protect voter credentials, voting choices, candidate details, and final election results. By implementing advanced security mechanisms, the system guarantees voter confidentiality, prevents unauthorized access, and ensures the integrity of the voting process.

The **Secure Voting System** provides a user-friendly interface for voter registration, authentication, and participation in elections. It incorporates secure backend operations using **Java, JSP, and SQL databases** to store and process encrypted data efficiently. With robust encryption techniques, this system significantly reduces the risks associated with digital voting and enhances trust in the electoral process.

This report presents a comprehensive overview of the **Secure Voting System**, including its design, implementation, security features, and testing methodologies. The project aims to contribute to the modernization of the voting process by offering a **secure, transparent, and scalable** solution for electronic elections.

index.html:

```html
<!-- index.html (Home Page) -->
<!DOCTYPE html>
<html>
<head>
    <title>Secure Voting System</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            padding: 50px;
            background:
url('https://media.kasperskydaily.com/wp-content/uploads/sites/92/2020/10/16044143/M
187_Digital-voting-header.png') no-repeat center center fixed;
            background-size: cover;
            color: white;
        }
        .nav {
            margin-top: 20px;
        }
        .nav a {
            margin: 10px;
            padding: 12px 25px;
            background: #4CAF50;
            color: white;
            text-decoration: none;
            border-radius: 5px;
            font-weight: bold;
            font-size: 16px;
        }
        .nav a:hover {
            background: #45a049;
        }
        .content {
            background: rgba(0, 0, 0, 0.7);
            padding: 30px;
            border-radius: 15px;
            display: inline-block;
            max-width: 800px;
        }
```

```
    h1 {
       font-size: 36px;
       margin-bottom: 20px;
    }
    p, ul {
       font-size: 18px;
       text-align: left;
       margin: 15px 0;
       line-height: 1.6;
    }
    ul {
       padding-left: 20px;
    }
    ul li {
       margin-bottom: 10px;
    }
  </style>
</head>
<body>
  <div class="content">
    <h1>Welcome to Secure Voting System</h1>
    <p><b>Ensuring secure and transparent elections is essential in today's digital
world.</b></p>
    <p>Our Secure Voting System provides a reliable and encrypted platform to
conduct elections without the fear of fraud or tampering.</p>

    <h2>Why Choose Our Secure Voting System?</h2>
    <ul>
      <li><b>Ensures voter confidentiality</b> using advanced encryption
techniques.</li>
      <li><b>Prevents vote tampering</b> through secure data handling.</li>
      <li><b>Guarantees data integrity</b> using cryptographic methods.</li>
      <li><b>Provides transparency and trust</b> in the voting process.</li>
    </ul>

    <h2>How Do We Keep Your Data Safe?</h2>
    <ul>
      <li><b>Voter credentials</b> are encrypted using <b>Substitution
Cipher</b>.</li>
```

```
        <li><b>Voting choices</b> are protected with <b>Rail Fence Cipher
Technique</b>.</li>
        <li><b>Candidate names</b> are securely encrypted and stored using <b>AES
Algorithm</b>.</li>
        <li><b>Final voting results</b> are decrypted using the <b>AES
Algorithm</b>.</li>
    </ul>

    <div class="nav">
        <a href="register.jsp">Register</a>
        <a href="login.html">Login</a>
    </div>
  </div>
</body>
</html>
```


Welcome to Secure Voting System

Ensuring secure and transparent elections is essential in today's digital world.

Our Secure Voting System provides a reliable and encrypted platform to conduct elections without the fear of fraud or tampering.

Why Choose Our Secure Voting System?

- **Ensures voter confidentiality** using advanced encryption techniques.
- **Prevents vote tampering** through secure data handling.
- **Guarantees data integrity** using cryptographic methods.
- **Provides transparency and trust** in the voting process.

How Do We Keep Your Data Safe?

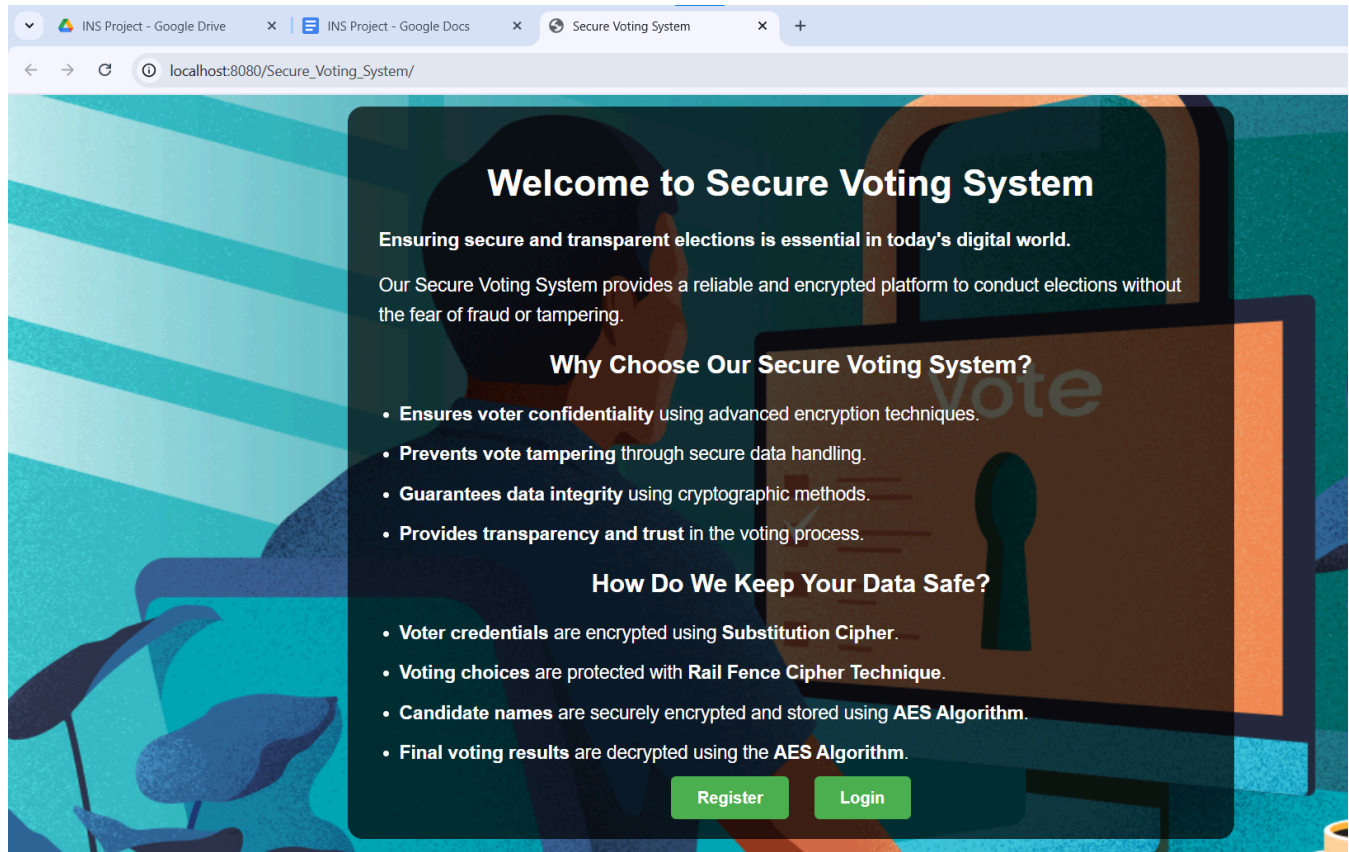- **Voter credentials** are encrypted using **Substitution Cipher**.
- **Voting choices** are protected with **Rail Fence Cipher Technique**.
- **Candidate names** are securely encrypted and stored using **AES Algorithm**.
- **Final voting results** are decrypted using the **AES Algorithm**.

Register   Login

register.jsp:

```jsp
<!-- register.jsp (Voter Registration Page) -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ page import="java.sql.*, java.util.*" %>
<!DOCTYPE html>
<html>
<head>
    <title>Voter Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            background:
url('https://www.codingnepalweb.com/wp-content/uploads/2023/01/registration-form-bac
kground.jpg') no-repeat center center fixed;
            background-size: cover;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .container {
            background: rgba(255, 255, 255, 0.9);
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 350px;
        }
        .container h2 {
            text-align: center;
            margin-bottom: 20px;
        }
        .input-group {
            margin-bottom: 15px;
        }
        .input-group label {
            display: block;
            font-weight: bold;
```

```
        }
        .input-group input {
           width: 100%;
           padding: 10px;
           border: 1px solid #ccc;
           border-radius: 5px;
        }
        .btn {
           width: 100%;
           padding: 10px;
           background: #4CAF50;
           color: white;
           border: none;
           border-radius: 5px;
           cursor: pointer;
        }
        .btn:hover {
           background: #45a049;
        }
    </style>
</head>
<body>
    <div class="container">
       <h2>Voter Registration</h2>
       <form action="register.jsp" method="post">
          <div class="input-group">
             <label for="name">Full Name</label>
             <input type="text" name="name" required>
          </div>
          <div class="input-group">
             <label for="email">Email</label>
             <input type="email" name="email" required>
          </div>
          <div class="input-group">
             <label for="password">Password</label>
             <input type="password" name="password" required>
          </div>
          <button type="submit" class="btn">Register</button>
       </form>
       <%
```

```
if ("POST".equalsIgnoreCase(request.getMethod())) {
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String password = request.getParameter("password");

    // Substitution Cipher (Caesar Cipher) for Password Encryption
    StringBuilder encryptedPassword = new StringBuilder();
    for (char c : password.toCharArray()) {
        encryptedPassword.append((char) (c + 3)); // Shift characters by 3
    }
    try {
        Connection conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/voters", "root", "password");
        String sql = "INSERT INTO voters (name, email, password) VALUES (?, ?,
?)";

        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setString(1, name);
        ps.setString(2, email);
        ps.setString(3, encryptedPassword.toString());
        ps.executeUpdate();
        out.println("<p style='color:green;'>Registration successful!</p>");
        ps.close();
        conn.close();
    } catch (Exception e) {
        out.println("<p style='color:red;'>Error: " + e.getMessage() + "</p>");
    }
}
%>
    </div>
</body>
</html>
```

Voter Registration

Full Name

Mehtab Singh Randhawa

Email

mehtabsingh@gmail.com

Password

•••••

Register

Voter Registration

Full Name

Email

Password

Register

Registration successful!

select * from ROOT.VOTERS ×

Page Size: 20 | Total Rows: 4 Page: 1 of 1

#	NAME	EMAIL	PASSWORD
1	Amandeep Kaur	aman123@gmail.com	dpdq456
2	Mehtab Singh Randhawa	mehtab@gmail.com	45678
3	Mehtab Singh Randhawa	singh@gmail.com	45678
4	Mehtab Singh Randhawa	mehtabsingh@gmail.com	45678

login.html:

```html
<!-- login.html (User Login Page) -->
<!DOCTYPE html>
<html>
<head>
   <title>User Login</title>
   <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
   <style>
      body {
         font-family: Arial, sans-serif;
         background:
url('https://www.codingnepalweb.com/wp-content/uploads/2023/01/registration-form-bac
kground.jpg') no-repeat center center fixed;
         background-size: cover;
         display: flex;
         justify-content: center;
         align-items: center;
         height: 100vh;
      }
      .container {
         background: rgba(255, 255, 255, 0.9);
         padding: 20px;
         border-radius: 10px;
         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
         width: 350px;
         text-align: center;
      }
      .container h2 {
         margin-bottom: 20px;
      }
      .input-group {
         margin-bottom: 15px;
      }
      .input-group label {
         display: block;
         font-weight: bold;
      }
      .input-group input {
         width: 100%;
```

```
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
        .btn {
            width: 100%;
            padding: 10px;
            background: #4CAF50;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        .btn:hover {
            background: #45a049;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>User Login</h2>
        <form action="login.jsp" method="post">
            <div class="input-group">
                <label for="email">Email</label>
                <input type="email" name="email" required>
            </div>
            <div class="input-group">
                <label for="password">Password</label>
                <input type="password" name="password" required>
            </div>
            <button type="submit" class="btn">Login</button>
        </form>
    </div>
</body>
</html>
```

login.jsp:

```
<!-- login.jsp (User Authentication with Substitution Cipher) -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
```

```jsp
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
  <title>Login Processing</title>
</head>
<body>
  <%
    String email = request.getParameter("email");
    String password = request.getParameter("password");
    boolean isValidUser = false;

    // Substitution Cipher (Caesar Cipher) for Password Encryption
    StringBuilder encryptedPassword = new StringBuilder();
    for (char c : password.toCharArray()) {
       encryptedPassword.append((char) (c + 3)); // Shift characters by 3
    }

    try {
       Connection conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/voters", "root", "password");
       String sql = "SELECT * FROM voters WHERE email=? AND password=?";
       PreparedStatement ps = conn.prepareStatement(sql);

       ps.setString(1, email);
       ps.setString(2, encryptedPassword.toString()); // Compare encrypted password
       ResultSet rs = ps.executeQuery();

       if (rs.next()) {
          isValidUser = true;
          session.setAttribute("user", email);
       }

       rs.close();
       ps.close();
       conn.close();
    } catch (Exception e) {
       out.println("<p style='color:red;'>Database Error: " + e.getMessage() + "</p>");
    }
```

```
        if (isValidUser) {
            response.sendRedirect("dashboard.jsp"); // Redirect to user dashboard
        } else {
            out.println("<p style='color:red;'>Invalid email or password. <a
href='login.html'>Try again</a></p>");
        }
    %>
</body>
</html>
```
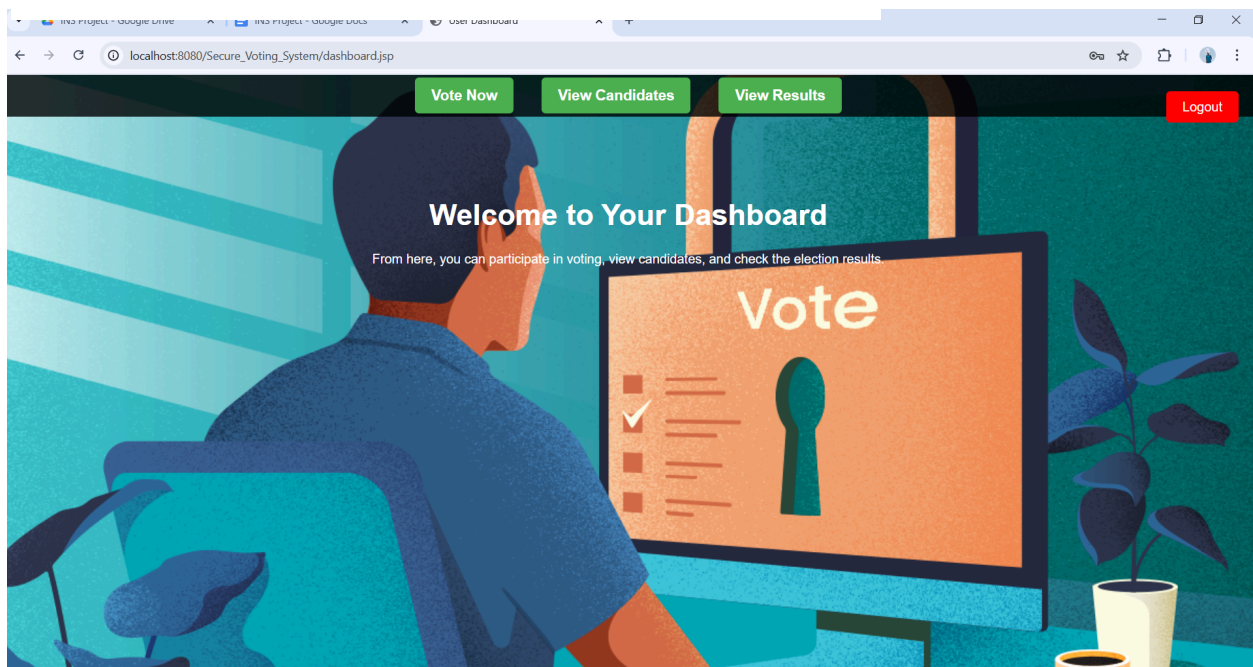
dashboard.jsp:

```jsp
<!-- dashboard.jsp (User Dashboard with Stylish Navigation) -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
    <title>User Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background:
url('https://media.kasperskydaily.com/wp-content/uploads/sites/92/2020/10/16044143/M
187_Digital-voting-header.png') no-repeat center center fixed;
            background-size: cover;
            color: white;
            margin: 0;
            padding: 0;
        }
        .navbar {
            background: rgba(0, 0, 0, 0.8);
            padding: 15px;
            text-align: center;
        }
        .navbar a {
            color: white;
            text-decoration: none;
            font-size: 18px;
            margin: 15px;
            padding: 12px 20px;
            background: #4CAF50;
            border-radius: 5px;
            font-weight: bold;
            transition: 0.3s;
        }
        .navbar a:hover {
            background: #45a049;
        }
        .container {
            text-align: center;
```

```
        margin-top: 100px;
      }
      .container h1 {
        font-size: 36px;
      }
      .logout {
        position: absolute;
        top: 20px;
        right: 20px;
        padding: 10px 20px;
        background: red;
        color: white;
        border: none;
        border-radius: 5px;
        font-size: 16px;
        cursor: pointer;
      }
    </style>
</head>
<body>
  <button class="logout" onclick="window.location.href='logout.jsp'">Logout</button>
  <div class="navbar">
    <a href="vote.jsp">Vote Now</a>
    <a href="candidates.jsp">View Candidates</a>
    <a href="results.jsp">View Results</a>
  </div>
  <div class="container">
    <h1>Welcome to Your Dashboard</h1>
    <p>From here, you can participate in voting, view candidates, and check the
election results.</p>
  </div>
</body>
</html>
```

vote.jsp:

```jsp
<!-- vote.jsp (Voting Page) -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ page import="java.sql.*, java.util.*" %>
<!DOCTYPE html>
<html>
<head>
    <title>Vote Now</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            background:
url('https://www.codingnepalweb.com/wp-content/uploads/2023/01/registration-form-background.jpg') no-repeat center center fixed;
            background-size: cover;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .container {
            background: rgba(255, 255, 255, 0.9);
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 350px;
        }
        .container h2 {
            text-align: center;
            margin-bottom: 20px;
        }
        .input-group {
            margin-bottom: 15px;
        }
        .input-group label {
            display: block;
            font-weight: bold;
```

```
        }
        .btn {
            width: 100%;
            padding: 10px;
            background: #4CAF50;
            color: white;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        .btn:hover {
            background: #45a049;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Vote for Your Party</h2>
        <form action="vote.jsp" method="post">
            <div class="input-group">
                <label for="candidate">Select Candidate</label>
                <select name="candidate" required>
                    <option value="BhartiyaJanataParty">Bhartiya Janata Party</option>
                    <option value="IndianNationalCongress">Indian National
Congress</option>
                    <option value="AamAadmiParty">Aam Aadmi Party</option>
                </select>
            </div>
            <button type="submit" class="btn">Submit Vote</button>
        </form>
        <%
            if ("POST".equalsIgnoreCase(request.getMethod())) {
                String candidate = request.getParameter("candidate");

                // Rail Fence Cipher for Vote Encryption
                int rail = 3; // Number of rails
                StringBuilder[] fence = new StringBuilder[rail];
                for (int i = 0; i < rail; i++) {
                    fence[i] = new StringBuilder();
                }
```

```
            int dir = 1, row = 0;
            for (char c : candidate.toCharArray()) {
                fence[row].append(c);
                row += dir;
                if (row == rail - 1 || row == 0) {
                    dir = -dir;
                }
            }

            StringBuilder encryptedVote = new StringBuilder();
            for (StringBuilder sb : fence) {
                encryptedVote.append(sb);
            }

            try {
                Connection conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/voters", "root", "password");
                String sql = "INSERT INTO votes (candidate) VALUES (?)";
                PreparedStatement ps = conn.prepareStatement(sql);
                ps.setString(1, encryptedVote.toString());
                ps.executeUpdate();
                out.println("<p style='color:green;'>Vote submitted successfully!</p>");
                ps.close();
                conn.close();
            } catch (Exception e) {
                out.println("<p style='color:red;'>Error: " + e.getMessage() + "</p>");
            }
        }
    %>
  </div>
</body>
</html>
```

Vote for Your Party

Select Candidate

Bhartiya Janata Party ▾

Submit Vote

Vote for Your Party

Select Candidate

Bhartiya Janata Party ▾

Submit Vote

Vote submitted successfully!

---

select * from ROOT.VOTES ×

Page Size: 20

#	CANDIDATE
1	IatansninainlogesdNoCr
2	IatansninainlogesdNoCr
3	AaPyaAdiatmmr
4	BtJtrhriaaaaataynPy
5	IatansninainlogesdNoCr
6	AaPyaAdiatmmr
7	BtJtrhriaaaaataynPy
8	BtJtrhriaaaaataynPy

candidates.jsp:

```
<!-- candidates.jsp (Political Parties & Candidates List with AES Encryption &
Decryption) -->
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ page import="java.sql.*, java.util.*, javax.crypto.Cipher,
javax.crypto.spec.SecretKeySpec, java.util.Base64" %>

<%!
    private String secretKey = "ThisIsASecretKey"; // 16 bytes key for AES-128

    // AES Encryption Method
```

```java
    private String encryptAES(String strToEncrypt, String secret) throws Exception {
        SecretKeySpec secretKeySpec = new SecretKeySpec(secret.getBytes(), "AES");
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);
        return
Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));
    }

    // AES Decryption Method
    private String decryptAES(String strToDecrypt, String secret) throws Exception {
        SecretKeySpec secretKeySpec = new SecretKeySpec(secret.getBytes(), "AES");
        Cipher cipher = Cipher.getInstance("AES");
        cipher.init(Cipher.DECRYPT_MODE, secretKeySpec);
        return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
    }
%>

<!DOCTYPE html>
<html>
<head>
    <title>List of Political Parties & Candidates</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            background:
url('https://www.codingnepalweb.com/wp-content/uploads/2023/01/registration-form-bac
kground.jpg') no-repeat center center fixed;
            background-size: cover;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .container {
            background: rgba(255, 255, 255, 0.9);
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
            width: 600px;
            text-align: center;
        }
        .container h2 {
            margin-bottom: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: left;
        }
        th {
            background: #4CAF50;
            color: white;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>List of Political Parties & Candidates</h2>
        <table>
            <tr>
                <th>Constituency</th>
                <th>AAP Candidate</th>
                <th>BJP Candidate</th>
                <th>Congress Candidate</th>
            </tr>
            <%
                try {
                    Connection conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/voters", "root", "password");
                    Statement stmt = conn.createStatement();
                    // Insert encrypted constituencies and candidates if not already present
                    String[][] candidates = {
                        {"NARELA", "Sharad Chauhan", "Raj Karan Khatri", "Aruna Kumari"},
```

```java
        {"BURARI", "Sanjeev Jha", "Shailendra Kumar (JDU)", "Mangesh
Tyagi"},

        {"TIMARPUR", "Surender Pal Singh", "Surya Prakash Khatri", "Lokendra
Chauhdary"},

        {"ADARSH NAGAR", "Mukesh Goel", "Raj Kumar Bhatia", "Shivank
Singhal"},

        {"BADLI", "Ajesh Yadav", "Deepak Chaudhary", "Devendra Yadav"},
        {"WAZIRPUR", "Rajesh Gupta", "Poonam Sharma", "Ragini Nayak"},
        {"CHANDNI CHOWK", "Punardeep Singh", "Satish Jain", "Mudit
Agarwal"},

        {"TILAK NAGAR", "Jarnail Singh", "Shweta Saini", "PS Bawa"},
        {"NEW DELHI", "Arvind Kejriwal", "Parvesh Verma", "Sandeep Dikshit"},
        {"KONDLI", "Kuldeep Kumar", "Priyanka Gautam", "Akshay Kumar"}
    };
    for (String[] candidate : candidates) {
        String encryptedConstituency = encryptAES(candidate[0], secretKey);
        String encryptedAAP = encryptAES(candidate[1], secretKey);
        String encryptedBJP = encryptAES(candidate[2], secretKey);
        String encryptedCongress = encryptAES(candidate[3], secretKey);
        stmt.executeUpdate("INSERT INTO candidates (constituency, aap, bjp,
congress) SELECT '" + encryptedConstituency + "', '" + encryptedAAP + "', '" +
encryptedBJP + "', '" + encryptedCongress + "' FROM SYSIBM.SYSDUMMY1 WHERE
NOT EXISTS (SELECT 1 FROM candidates WHERE constituency='" +
encryptedConstituency + "') FETCH FIRST 1 ROW ONLY");
            }

        // Retrieve and decrypt constituency and candidate names
        ResultSet rs = stmt.executeQuery("SELECT constituency, aap, bjp,
congress FROM candidates");
        while (rs.next()) {
        String decryptedConstituency = decryptAES(rs.getString("constituency"),
secretKey);
        String decryptedAAP = decryptAES(rs.getString("aap"), secretKey);
        String decryptedBJP = decryptAES(rs.getString("bjp"), secretKey);
        String decryptedCongress = decryptAES(rs.getString("congress"),
secretKey);
        %>
        <tr>
          <td><%= decryptedConstituency %></td>
          <td><%= decryptedAAP %></td>
```

```
                <td><%= decryptedBJP %></td>
                <td><%= decryptedCongress %></td>
            </tr>
            <%
        }
        rs.close();
        stmt.close();
        conn.close();
    } catch (Exception e) {
        out.println("<p style='color:red;'>Error: " + e.getMessage() + "</p>");
    }
    %>
    </table>
    </div>
</body>
</html>
```

localhost:8080/Secure_Voting_System/candidates.jsp

List of Political Parties & Candidates

Constituency	AAP Candidate	BJP Candidate	Congress Candidate
NARELA	Sharad Chauhan	Raj Karan Khatri	Aruna Kumari
BURARI	Sanjeev Jha	Shailendra Kumar (JDU)	Mangesh Tyagi
TIMARPUR	Surender Pal Singh	Surya Prakash Khatri	Lokendra Chauhdary
ADARSH NAGAR	Mukesh Goel	Raj Kumar Bhatia	Shivank Singhal
BADLI	Ajesh Yadav	Deepak Chaudhary	Devendra Yadav
WAZIRPUR	Rajesh Gupta	Poonam Sharma	Ragini Nayak
CHANDNI CHOWK	Punardeep Singh	Satish Jain	Mudit Agarwal
TILAK NAGAR	Jarnail Singh	Shweta Saini	PS Bawa
NEW DELHI	Arvind Kejriwal	Parvesh Verma	Sandeep Dikshit
KONDLI	Kuldeep Kumar	Priyanka Gautam	Akshay Kumar

select * from ROOT.CANDID... ×

Page Size: 20 Total Rows: 10 Page: 1 of 1 Matching Rows:

#	CONSTITUENCY	AAP	BJP	CONGRESS
1	oNVYHed+xlLkB6AEGR6CgA==	7DZsV+12sgC1fX1SdoqVgA==	QAcu/I3bpApeCRsgcO2dSvRRYGjkU77b/7tz3twFxRc=	f7A04I597JRO+ahfCUwLLA==
2	zBLBRK5vggGTKlTMH9WOmg==	V6LtXAsnT9gEjlyvhLLy/A==	1ISoiiWFUfOqnBid+/bHUQp+GY0Lyx+WCeef9y1fDNA=	9MfCyKUyh0HVCzY0VHJNGw==
3	EMYmKNQ3RpH6blGllRfbMQ==	N0Qh5yr9xwKQjXWvazGOfY51yI/FNJwXuSaoSZrjVSc=	XJz8jmV/RGGg0cXjduXRNOyNrWztjDzqTIfxXR05amM=	UqEtwmHJ2Rbku0A9Ezyj3kNGaTbvzKEtTApB++daeG8=
4	zrVAsS79C2puBMJyxRo44g==	p9B23DxGOwZGapyQZXQwjQ==	tBTvlCDUipPn9rTowzfrTvRRYGjkU77b/7tz3twFxRc=	XJVe0h5v+PieiD8hVkteyg==
5	pP9YuSHIgmVqUWJl5RMEgQ==	J5eeqf3pxulDGSUzvqMlKA==	CQ0zFrRolxra+7fNCfpF7/RRYGjkU77b/7tz3twFxRc=	HgPyF4wmeMIZ/z6Eld+Uhg==
6	R4T6MagpUNVwVrXjYacXHQ==	dSeMETdoc9ezmV5TmefJZA==	JnPVZTM2ymorEyF9gNXj1w==	2bpkLx0VsSQqfd51uILq9A==
7	E/iIlLp4QPugcWVAE4a4nQ==	MSrHBvhwfYNoQPav0K71Iw==	eAuZZS0YLLuvN4nfVtAOxg==	DZkEtGCwT1QiZXxbt0LjDQ==
8	a79XvuPMYK92mD/ftDRkww==	B7Guy8udVyMLwKJGXMzkbQ==	D1kHqFGpGc5TD4VE7FI+Wg==	qytep1UH//sp4Jdl4lK+uA==
9	38VpiZzzc5IqLjZxDo5sTQ==	fzKiEsV1LOrOEp4KAdoE+Q==	rrYseDFhxWe2O4tWLMyPFw==	LlpSruZ6Dgb0GPOIyritrA==
10	J7NGVaEQf1G6fE/czYVYfA==	jEamsaFS28wARyNmBUWwHQ==	h5nGQMlRLX25amFf7dsA6A==	VxGk3Y4XOHFMfXtMWKpRrA==

results.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ page import="java.sql.*, java.util.*" %>
<%@ page import="util.RailFenceCipher" %>
<!DOCTYPE html>
<html>
<head>
    <title>Voting Results</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            background:
url('https://www.codingnepalweb.com/wp-content/uploads/2023/01/registration-form-background.jpg') no-repeat center center fixed;
            background-size: cover;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .container {
            background: rgba(255, 255, 255, 0.9);
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            width: 450px;
        }
        .container h2 {
            text-align: center;
            margin-bottom: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
        }
        th, td {
            border: 1px solid #ddd;
```

```
            padding: 8px;
            text-align: center;
        }
        th {
            background-color: #4CAF50;
            color: white;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Voting Results</h2>
        <table>
            <tr>
                <th>Party Name</th>
                <th>Votes</th>
            </tr>
            <%
                Map<String, Integer> voteCount = new HashMap<String, Integer>();

                try {
                    Connection conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/voters", "root", "password");
                    Statement stmt = conn.createStatement();
                    ResultSet rs = stmt.executeQuery("SELECT candidate FROM votes");

                    while (rs.next()) {
                        String encryptedCandidate = rs.getString("candidate");
                        String decryptedCandidate =
RailFenceCipher.decrypt(encryptedCandidate, 3);

                        if (voteCount.containsKey(decryptedCandidate)) {
                            voteCount.put(decryptedCandidate,
voteCount.get(decryptedCandidate) + 1);
                        } else {
                            voteCount.put(decryptedCandidate, 1);
                        }
                    }

                    rs.close();
```

```
                stmt.close();
                conn.close();
            } catch (Exception e) {
                out.println("<p style='color:red;'>Error: " + e.getMessage() + "</p>");
            }

            for (Map.Entry<String, Integer> entry : voteCount.entrySet()) {
        %>
        <tr>
            <td><%= entry.getKey() %></td>
            <td><%= entry.getValue() %></td>
        </tr>
        <% } %>
    </table>
  </div>
</body>
</html>
```

Voting Results

Party Name	Votes
BhartiyaJanataParty	3
IndianNationalCongress	3
AamAadmiParty	2

select * from ROOT.VOTES ×

#	CANDIDATE
1	IatansninainlogesdNoCr
2	IatansninainlogesdNoCr
3	AaPyaAdiatmmr
4	BtJtrhriaaaaataynPy
5	IatansninainlogesdNoCr
6	AaPyaAdiatmmr
7	BtJtrhriaaaaataynPy
8	BtJtrhriaaaaataynPy

Conclusion

The **Secure Voting System** successfully demonstrates the implementation of a secure and efficient electronic voting platform by integrating advanced cryptographic methods. Through the use of **Substitution Cipher, Rail Fence Cipher, and AES Encryption**, the system ensures data security, prevents vote manipulation, and maintains voter anonymity.

This project has effectively addressed common challenges in digital voting, such as unauthorized access, vote tampering, and data breaches. By encrypting voter credentials, securing vote submissions, and using strong authentication mechanisms, the system enhances trust in the electoral process.

Additionally, the project provides an intuitive **graphical user interface (GUI)** that allows users to **register, log in, cast votes, and view results** seamlessly. The backend implementation ensures efficient data processing, secure storage, and accurate vote tallying.

In conclusion, the **Secure Voting System** offers a **reliable, scalable, and future-ready** solution for conducting elections electronically. While this project provides a strong foundation for secure e-voting, future enhancements could include **biometric authentication, blockchain-based vote recording, and real-time monitoring** to further strengthen the integrity and transparency of the voting process. This system serves as a **step forward in the evolution of digital democracy**, paving the way for **safer and more accessible** electoral participation.