Introduction

Online voting system is developed to make the voting system digitized and easier. By using this website, voter verification can be done by answering a security question. Then after verification, voters can cast vote and see the current election results.

Project features

The main features of this websites are

- Voter verification by answering a security question
- User registration and login
- Edit user profile
- Election info and candidate names can be found in homepage
- User can see
- Voters can cast vote
- All the registered user can view election result

Platforms

We have used some very specific software in our project. They are-

- Html, Materialize CSS, Google Material Design for designing
- Java, JSP, Servlet for logical implementation
- MySQL for database.

Class Lists

The main Controller Classes in our project -

HomePageController	By using this class, the information about election time, candidate name and their profile and results are shown. The data is fetched from the database.
UserController	User registration, login,edit user profile,vote count are managed from this class
VoterVerifyController	This class manages the voter verification when a user enters this website and fill up their NID and name. If the verification is done successfully, user can see the website’s homepge.

Public Member Functions

String	**getServletInfo** ()

Protected Member Functions

void	**processRequest** (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException, ClassNotFoundException, SQLException, FileUploadException, Exception
void	**doGet** (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
void	**doPost** (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

Member Function Documentation

doGet()

Handles the HTTP GET method.

Parameters

request	servlet request
response	servlet response

Exceptions

ServletException	if a servlet-specific error occurs
IOException	if an I/O error occurs

doPost()

Handles the HTTP POST method.

Parameters

request	servlet request
response	servlet response

Exceptions

ServletException	if a servlet-specific error occurs
IOException	if an I/O error occurs

getServletInfo()

Returns a short description of the servlet.

Returns

A String containing servlet description

processRequest()

Processes requests for both HTTP GET and POST methods.

Parameters

request	servlet request
response	servlet response

Exceptions

ServletException	if a servlet-specific error occurs
IOException	if an I/O error occurs
java.lang.ClassNotFoundException	
java.sql.SQLException	
org.apache.commons.fileupload.FileUploadException	

Use case diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system. It is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

Here is the use case diagram we used in our project.

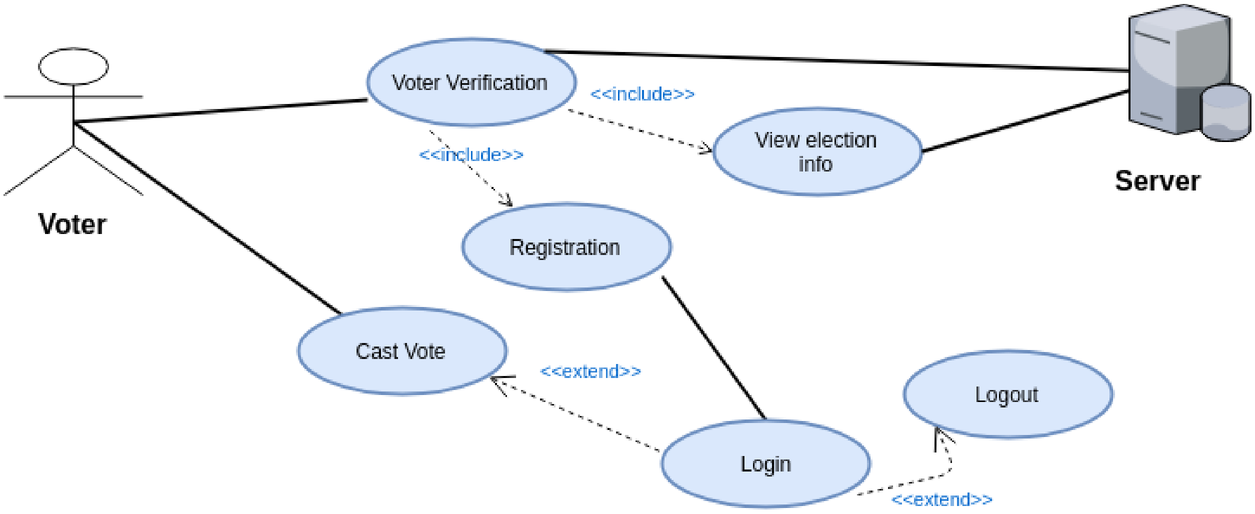

Fig 1: Use case diagram (Online Voting System)

Use case description

1. **Use case : Voter Verification**

   Actor: Voter

   Pre-condition: Verify security question

   Primary path:

   i) Enter voter NID

ii) Enter name

Alternate path:

i) Wrong NID

ii) Wrong name

2. **Use case: Cast Vote**

Actor: Voter

Pre-condition: Login

Primary path:

i) Choose the candidate

ii) Vote for him/her

Alternate path:

i) Vote time is ended

3. **Use case: Registration**

Actor: Voter

Pre-condition: Voter verification

Primary path:

i) Fill up username,email,phone number,password

Alternate path:

i) User name should be unique

4. **Use case: View Election Info**

Actor: Voter

Pre-condition: Voter Verification

Primary path:

i) Election start and end time

ii) Candidate info

iii) Voting result

Alternate path:

i) Not verified

ii) Server down

Entity relationship diagram (ERD)

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. It describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

In software engineering an ER model is commonly formed to represent things that a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

The three main components of ERD are:

- ➢ Entity
- ➢ Relationship
- ➢ Attribute

Entity Set Names

- Voter
- User
- Candidate
- Result
- Election

Entity Sets & Their Attributes

Entity: Voter

➢ Attributes:

• user_id	: INT (PRIMARY KEY)
• user_nid	: INT(PRIMARY KEY)
• user_name	: VARCHAR
• user_securityQ	:VARCHAR
• user_securityA	:VARCHAR

Entity: User

- Attributes:

user_id	: ID (PRIMARY KEY)
user_name	: VARCHAR (PRIMARY KEY)
user_emai	: VARCHAR (PRIMARY KEY)
user_phoneNo	: VARCHAR
user_password	: VARCHAR

Entity: Candidate

- Attributes:

c_id	: INT (PRIMARY KEY)
c_name	: VARCHAR
c_education	: VARCHAR
c_agenda	: VARCHAR
c_experience	: VARCHAR
image	: BLOB
election_name	:VARCHAR (FOREIGN KEY)

Entity: Result

- Attributes:

result_id	: INT (PRIMARY KEY)
election_name	: VARCHAR (FOREIGN KEY)
c_id	: VARCHAR (FOREIGN KEY)
c_name	: VARCHAR (FOREIGN KEY)
vote	: INT

Entity: Election

- Attributes:

election_id	: INT (PRIMARY KEY)
election_name	: VARCHAR
election_start	: DATETIME
election_end	: DATETIME

ER Diagram

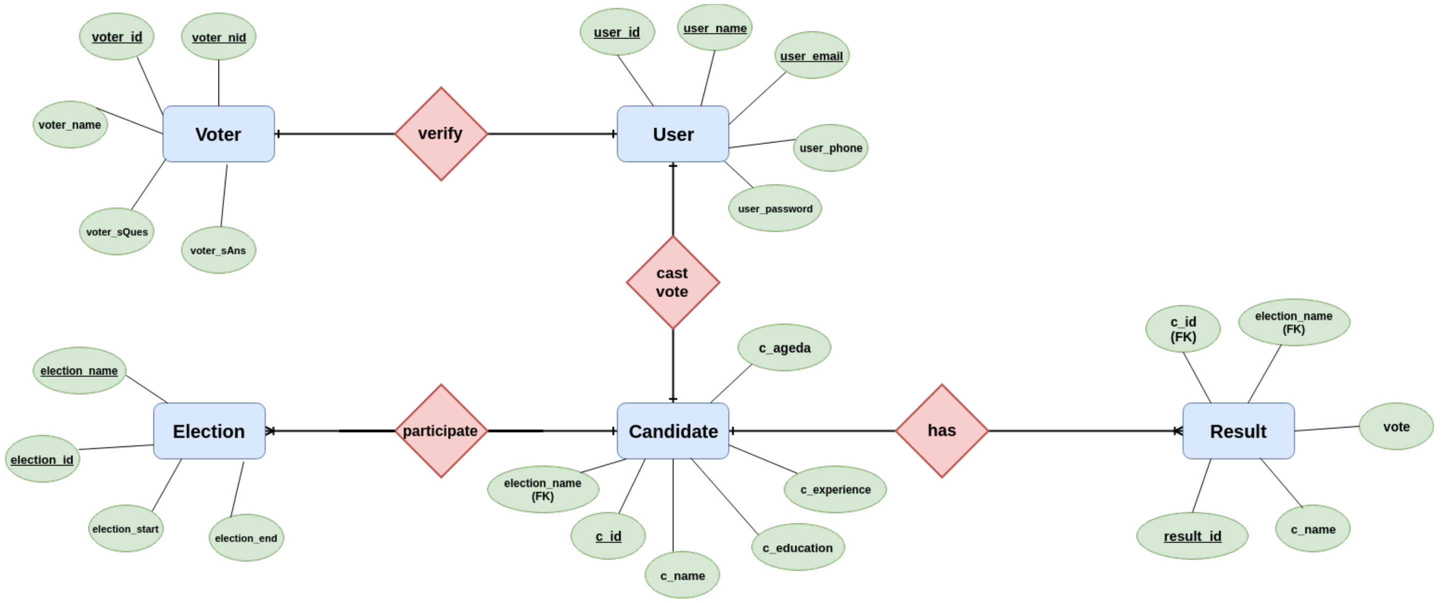

Fig 2: Entity Relationship Diagram (Online Voting System)

Future Plan and Conclusion

- Applying blockchain for encrypting the vote
- Make the website more robust and user friendly

The website is simple and has an interactive UI design to make the voting system much easier.