SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

DEPT. Of Computer Science Engineering

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Team Number	04
Title of Project	Online Voting Management System
Name of the candidate	Shravya Sharan
Team Members	Prachet Balaji, Karthik Menon
Register Number	RA1911026010055
Date of Experiment	07-06-2021

Staff Signature with date

Aim:

To identify, design and present a Software Project.

Team Members:

Sr No	Register No	Name	Role
1	RA1911026010055	Shravya Sharan	Leader
2	RA1911026010054	Karthik Menon	Member
3	RA1911026010053	Prachet Balaji	Member

Project Title:

Online Voting Management System

Project Description:

Online Voting Management System is an unprecedented project premised on a collaborative approach that ensures greater citizen input through partnerships within the academic community, public interest organizations, and with policy makers, in the pursuit of establishing a voting systems development model that is collaborative and transparent - and which is founded on sound data. This project was developed in response to the growing voting system needs and in recognition of future regulatory changes and pending legal requirements our current systems are unable to meet.

Online Voting Management System seeks to effectively utilize Information and Communication Technology (ICT) initiatives with the goal of implementing a new and enhanced voting system that can be used as a tool for advancing democracy, building trust in electoral management, adding credibility to election results and increasing the overall efficiency of the electoral process

Table of Contents

1. Executive Summary

Online voting is often seen as a tool for advancing democracy, building trust in electoral management, adding credibility to election results and increasing the overall efficiency of the electoral process.

Technology upgrades in elections are always challenging projects that require careful deliberation and planning. Introducing online voting is probably the most difficult upgrade as this technology touches the core of the entire electoral process—the casting and counting of the votes. Online voting greatly reduces direct human control and influence in this process. This provides an opportunity for solving some old electoral problems, but also introduces a whole range of new concerns.

This report identifies and discusses the need for systematically producing a complete set of requirements specification for Online Voting Management Systems that unifies the requirements imposed by the existing legal framework, the functionality reflected by the conventional voting procedures, and the required security attributes that the system should exhibit. This report elucidates the requirements and its process while also validating and enhancing these requirements focusing, also, on non-functional ones with the expectation to incorporate the outcome of these activities in the system design and development phases.

The requirement of rigorous testing and their associated documentation during the software development life cycle arises to identify defects and reduce flaws in the component or system thereby increasing the overall quality of the system. There can also be a requirement to perform software testing to comply with legal requirements or industry-specific standards. These standards and rules can specify what kind of techniques should we use for product development.

This document also describes the plan for testing the Online Voting Management System. This Test Plan document identifies existing project information and the software that should be tested and describe the testing strategies to be employed as well as identify the required resources and list the deliverable elements of the test activities. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

This document provides the background and discusses typical features provided by online voting solutions and the various technical options associated with it, and also provides an overview of the strengths and weaknesses of this technology.

2. Problem Statement

Online voting is often seen as a tool for making the electoral process more efficient and for increasing trust in its management. Properly implemented, online voting solutions can increase the security of the ballot, speed up the processing of results and make voting easier.

The present voting system has proved inefficient because: -

1. The voters' registration process is slow.
2. The manual collation of results takes time and gives room for result manipulation.
3. The inaccessible nature of election venues which includes the long distances to be covered by voters to their registered location increases voters' apathy towards the election processes.
4. The issues of ballot box snatching, damage and other election violence.

The aforementioned issues associated with the traditional ballot paper voting, defiles the purpose of voting in election process as a formal process of expressing individual opinions for or against some motion. Online voting greatly reduces direct human control and influence in this process. This provides an opportunity for solving old electoral problems.

Successful implementation of the software will yield the following benefits for the organization: -

1. Generation of revenue through development of software and execution of voting process including other support system.
2. Establish strong brand equity for the organisation.
3. Project diversification.
4. Ease in vote collection process.

3. Requirements

3.1. High Level Requirements

The key requirements to fully address the business need are: -

1. Government support- To be able to present the software as a substitute for manual voting system, it is imperative for the software to be recognised and accepted by the Government as well as for getting access to voters' data.
2. Skilled manpower- From the creation till successful execution, skilled manpower is required to assist in proper implementation and use of software.
3. Capital – To purchase other requirements.
4. Logistic Requirement-Proper network and technical availability.

3.2. Functional Requirements

Requirement (#)	Requirement Description	Department	Business User	Priority
E1FR1	As a voter, I should receive a message declaring the success/failure of the vote casted.	IT Department (Developer)	Voters	Secondary
E1FR2	As a voter, I should receive appropriate error message in case of system malfunction.	IT Department (Developer)	Voters	Secondary
E2FR1	As an election organiser, I can create, view and modify election procedure.	IT Department (Developer)	Election Commission of India, Others	Primary
E3FR1	As system administrator, I can transfer information of system functions over public network while maintaining their privacy.	IT Department (Developer, Security)	Organization	Primary
E2FR3	As an election organiser, I can import, insert, view and modify electors for one or more election procedures.	IT Department (Developer)	Election Commission of India, Others	Primary
E2FR5	As an election organiser, I can notify the system about candidate parties for an election.	IT Department (Developer)	Election Commission of India, Others	Primary
E4FR1	As candidates or election organisers, I can insert, modify and delete candidate's data for a specific election region.	IT Department (Developer)	Candidates, Others	Primary
E4FR2	As candidate, I can provide information about candidate parties.	IT Department (Developer)	Candidates	Secondary
E1FR3	As voter, I can view all available candidates and cast my vote.	IT Department (Developer)	Voters	Primary
E3FR2	As system administrator or election organiser, I can verify the result integrity.	IT Department (Developer, Security)	Organization	Primary
E3FR3	As system administrator, I can view all internal system operation without sacrificing voter confidentiality.	IT Department (Security)	Organization	Primary

3.3. Non-Functional Requirements

Requirement (#)	Category of NFR	Requirement Specification	Department
NFRP1	Performance	Reasonably short response time (3 seconds)	IT Department
NFRP3	Performance	Capability of recovering from system crashes and continuing the voting process.	IT Department (Support)
NFRC1	Confidentiality	Security features (OTP) for voter anonymity.	IT Department (Security)
NFRE1	Compliance	Process supported by the system should adhere to the laws specified by the government.	Department of Compliance
NFRU1	Usability	The system should provide an easy-to-use interface which is easily navigable.	IT Department (Developer)
NFRY1	Security	Availability of password authentication and encrypted transactions.	IT Department (Security)
NFRY3	Security	The system should possess strategies to counter attempts of hacking or unauthorised access.	IT Department (Security)
NFRY5	Security	Access to administrator to shut down the server and close all connections immediately in case of any security lapse.	IT Department (Developer and Security)
NFRT1	Traceability	Storage of votes being polled in a backup server at rapid frequency	IT Department (Hardware)
NFRF2	Flexibility	Features for addition or subtraction of required data.	IT Department (Developer)
NFRR1	Reliability	The system should be robust and have a high degree of fault tolerance.	IT Department (Developer)
NFRR2	Reliability	The system should impose a successful strategy to avoid multi voting.	IT Department (Developer)
NFRR3	Reliability	Application should be able to accurately count the polled votes and display the result.	IT Department (Developer)
NFRI1	Integrity	The administrators must be authenticated before being granted access to the system.	IT Department (Developer and Security)
NFRI2	Integrity	The system should be logically and physically secure to protect the database.	IT Department (Developer)

3.4. Infrastructure Requirements

Requirement (#)	Requirement Specification	Department	Business User
IR1	MYSQL DBMS	IT Department	Organization
IR2	Biometric Setup	IT Department	Organization
IR3	PHP, HTML and CSS	IT Department	Organization
IR4	Cloudflare DDoS Mitigation Equipment	IT Department	Organization
IR5	Web Browser	IT Department	User and Organization

1. MYSQL DBMS - It allows combination, extraction, manipulation and organization of data in the voters' database. It is platform independent and therefore can be implemented and used across several such as Windows, Linux server and is compatible with various hardware mainframes. It is fast in performance, stable and provides business value at a low cost.

2. Biometric Setup – For additional security during voting procedures, an option of verification via biometrics is used wherein voter can verify their identity with their finger print and other allotted credentials.

3. PHP, HTML and CSS coding- The voting client and election server will be written in PHP and CSS while the site will be designed using HTML.

4. Cloudflare DDoS Mitigation Equipment - The DDoS protection in Cloudflare is multifaceted in order to mitigate the many possible attack vectors. Cloudflare's network runs 10% of the Internet, creating an advantage in analysing data from attack traffic around the globe.

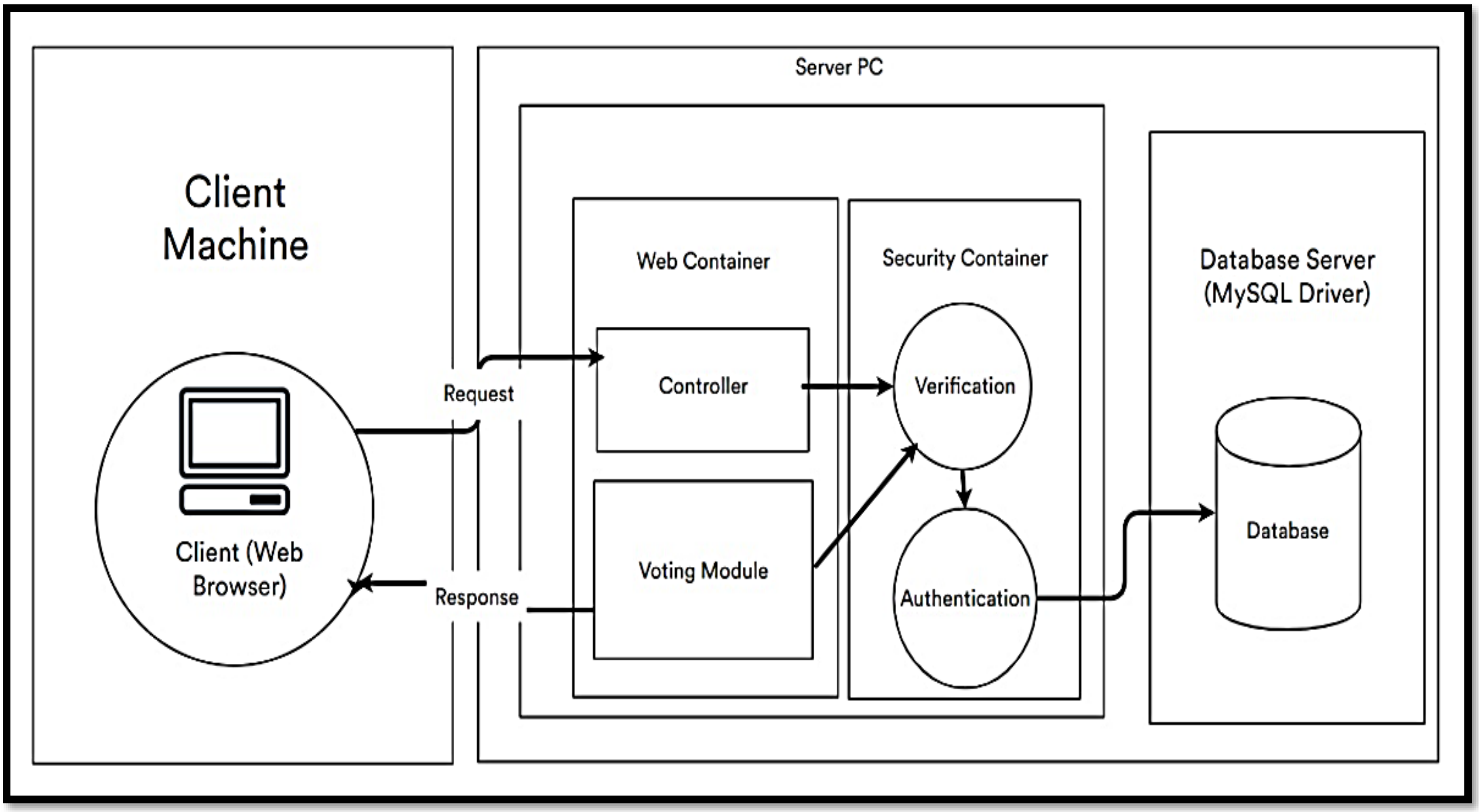
5. Web browsers: Mozilla Firefox, Google Chrome, Opera, Internet Explorer are required to run the application.

4. Design

4.1. Architecture Diagram

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system. The given diagram is a basic representation of the system.

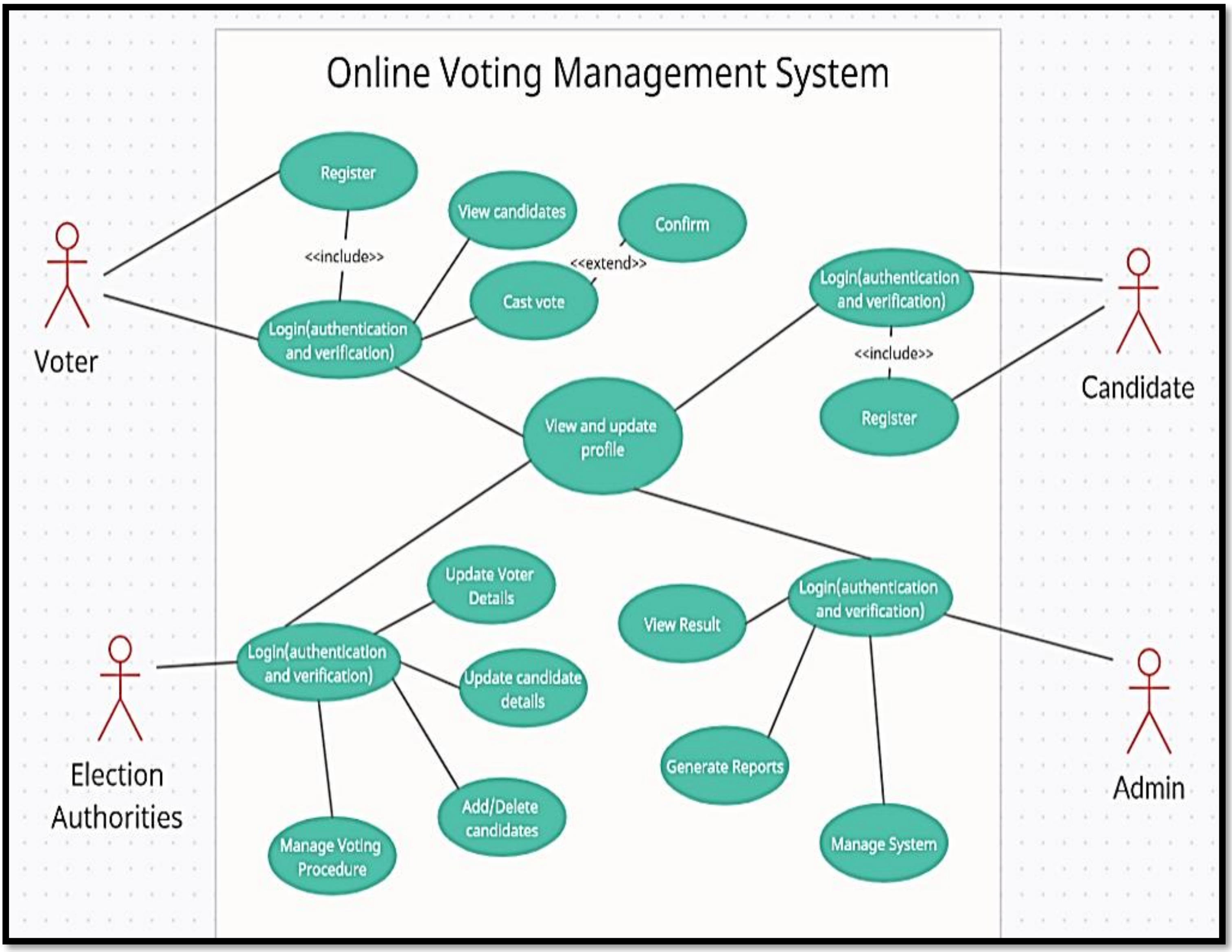
1. The client machine interacts with the user and based on their role allocates system functionalities to them. The user can access the system after successful registration and login.
2. The server PC contains web container, security container and database server which are key aspect of the online voting management system.
3. The web container holds processing data for user functionalities and is responsible for executing the processes involved within the system.
4. The security container prevents unauthorised access of database and only allows execution of process after successful authentication and verification.
5. The controller has access to the back-end functions and can modify system database after successful authentication.
6. The voting module incorporates all processes involved in the voting process and access the database for exchange of information.
7. The database stores all the requisite data.



4.2. *Use Case Diagram*

The Use Case Diagram is a graphic depiction of the interaction among the elements of Online Voting Management System. It represents the methodology used in system analysis to identify and organize system requirements of Online Voting Management System. In the use case diagram, the modules are represented as actions.

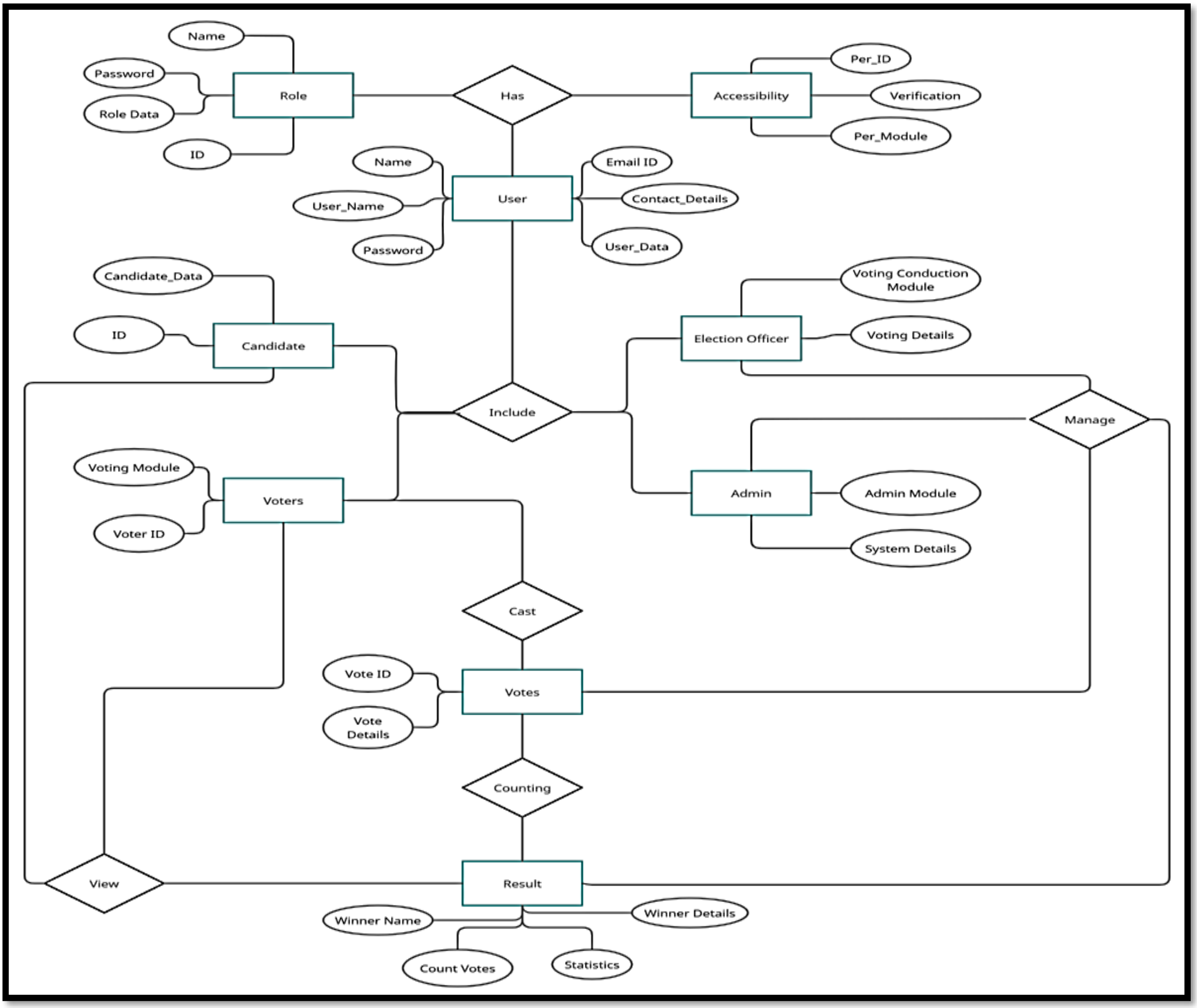
1. The Voter can cast their vote for the desired candidate after successful registration and login. Their vote is accepted if they pass the verification process.
2. The Candidate can contest for elections after successful login and registration. They can view and update their profile upon verification.
3. The Election Authorities can add and update voter and candidate details and manage the entire voting process including duration and mode of voting.
4. The Admin is mainly concerned with the system and can view its functionalities.



4.3. *ER Diagram*

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.

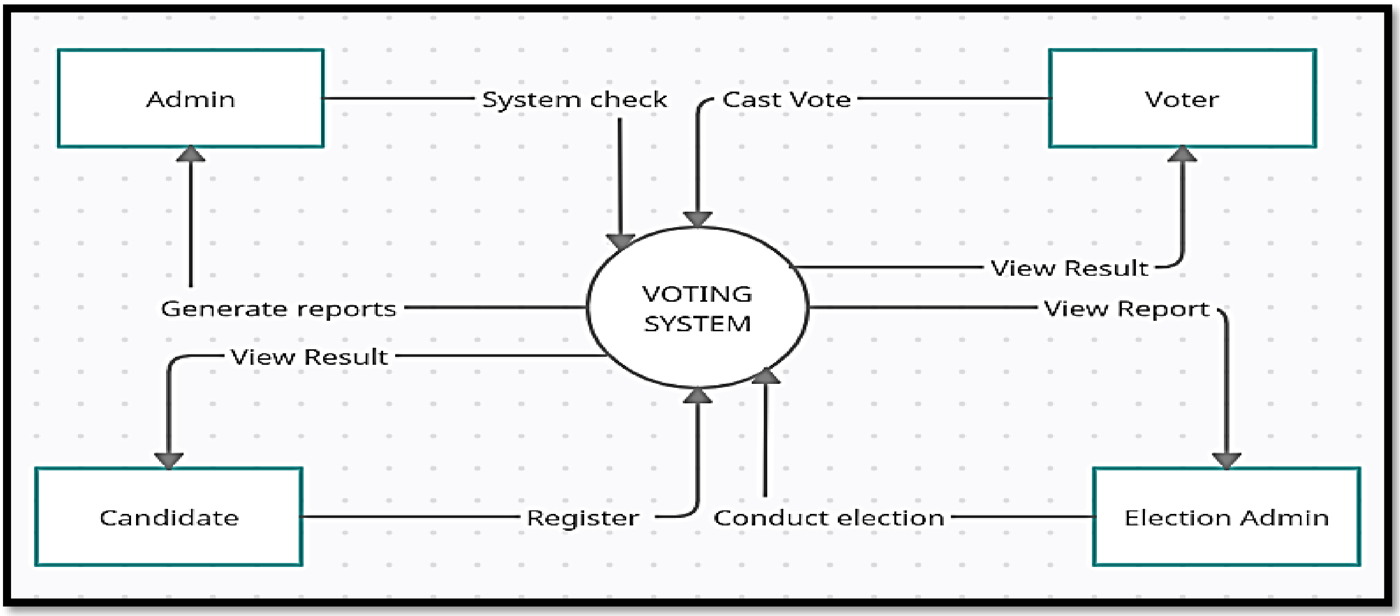
1. The User database is comprised of role and accessibility based on which the user can access the system.
2. The User database is further split into Candidate, Voters, Election Officer and Admin and each have their own data and functionalities.
3. The Votes database manages the voting process and stores the votes cast by voters and can be manages by Election Officer and Admin.
4. The Result database computes the result and displays the result along with reports and statistics.



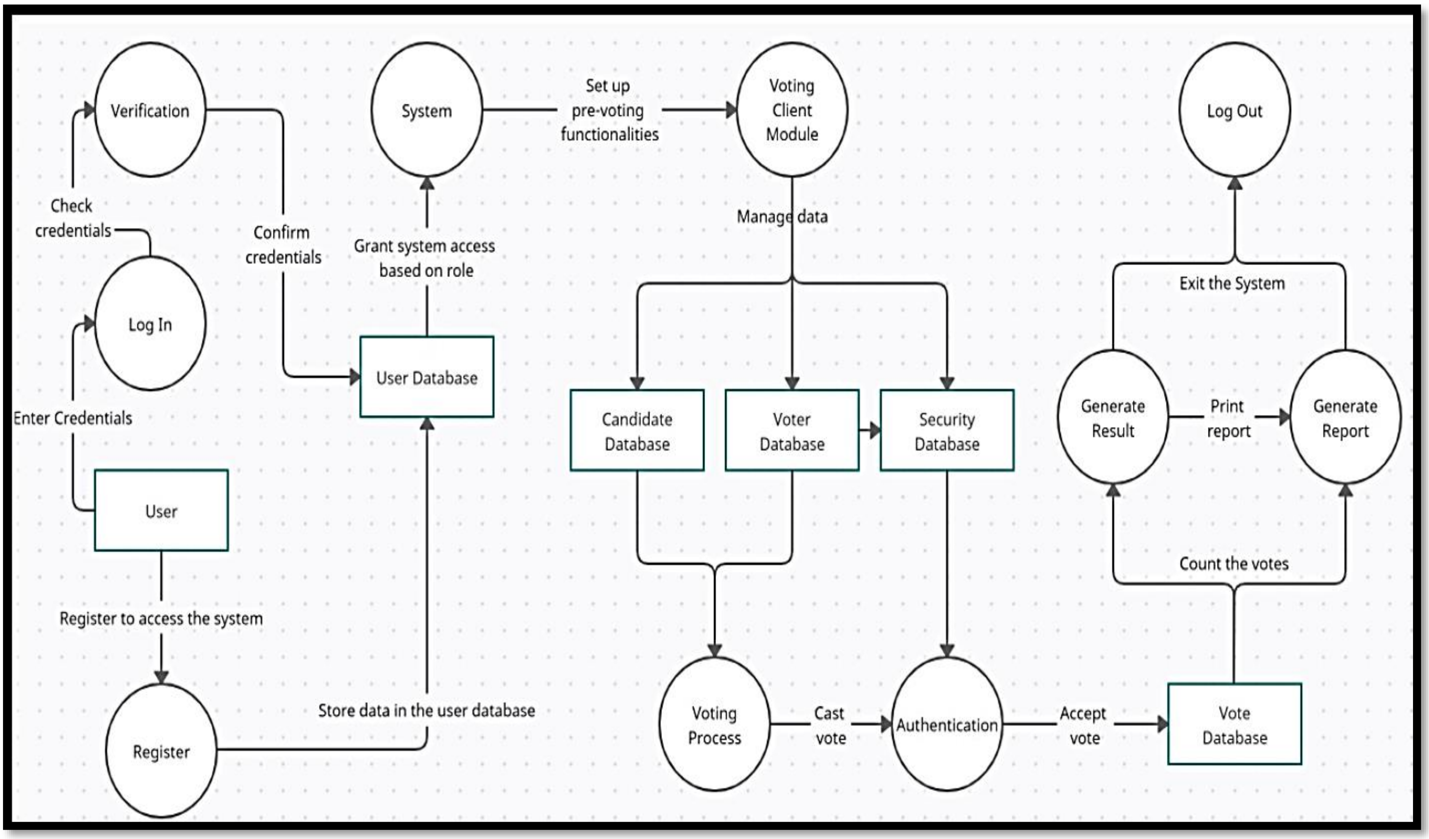
4.4. Data Flow Diagram

Data flow shows how the system is divided into sub-systems, each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Online Voting Management System as a whole.

4.4.1. DFD Level 0



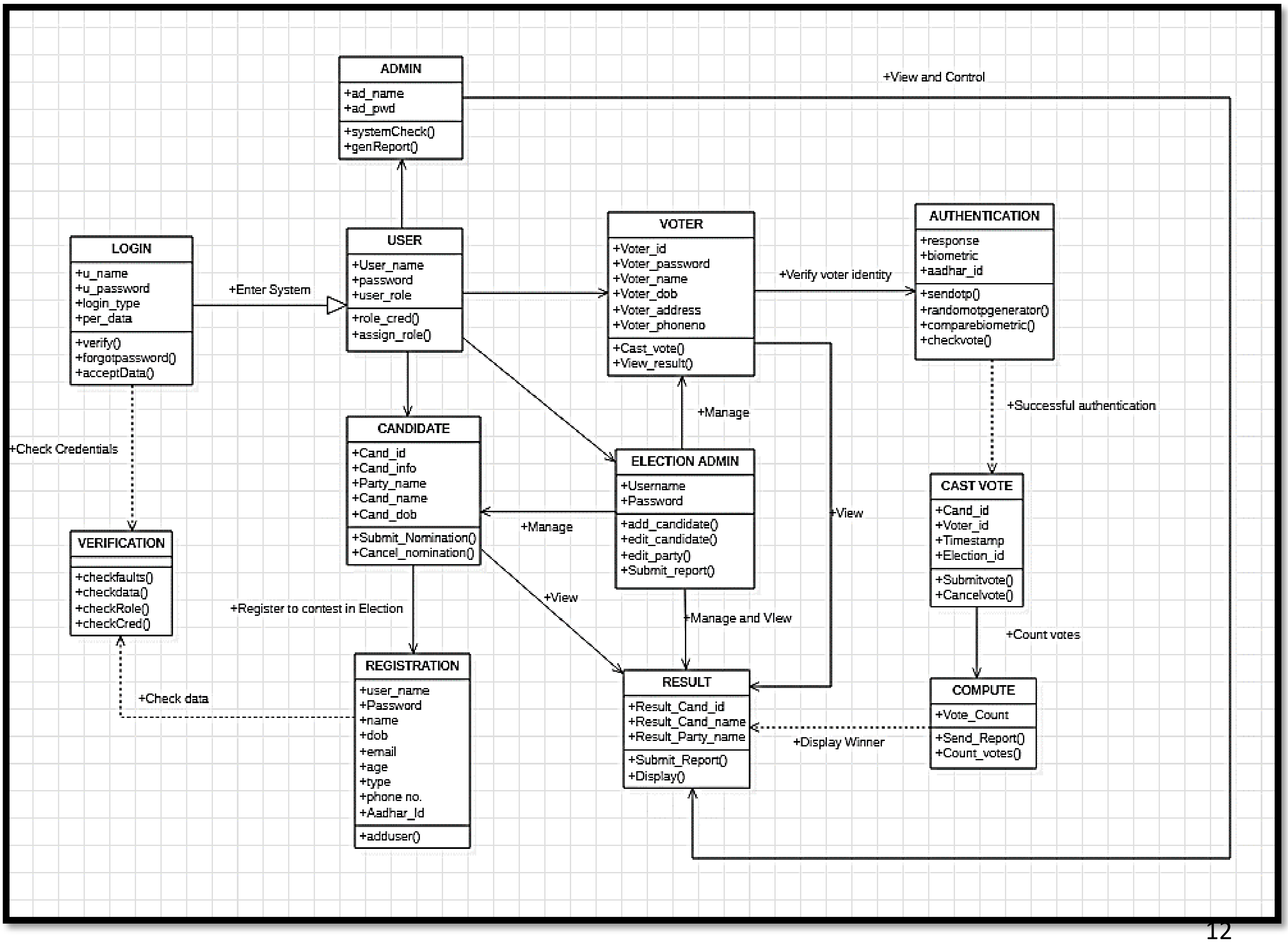
4.4.2. DFD Level 1



4.5. Class Diagram

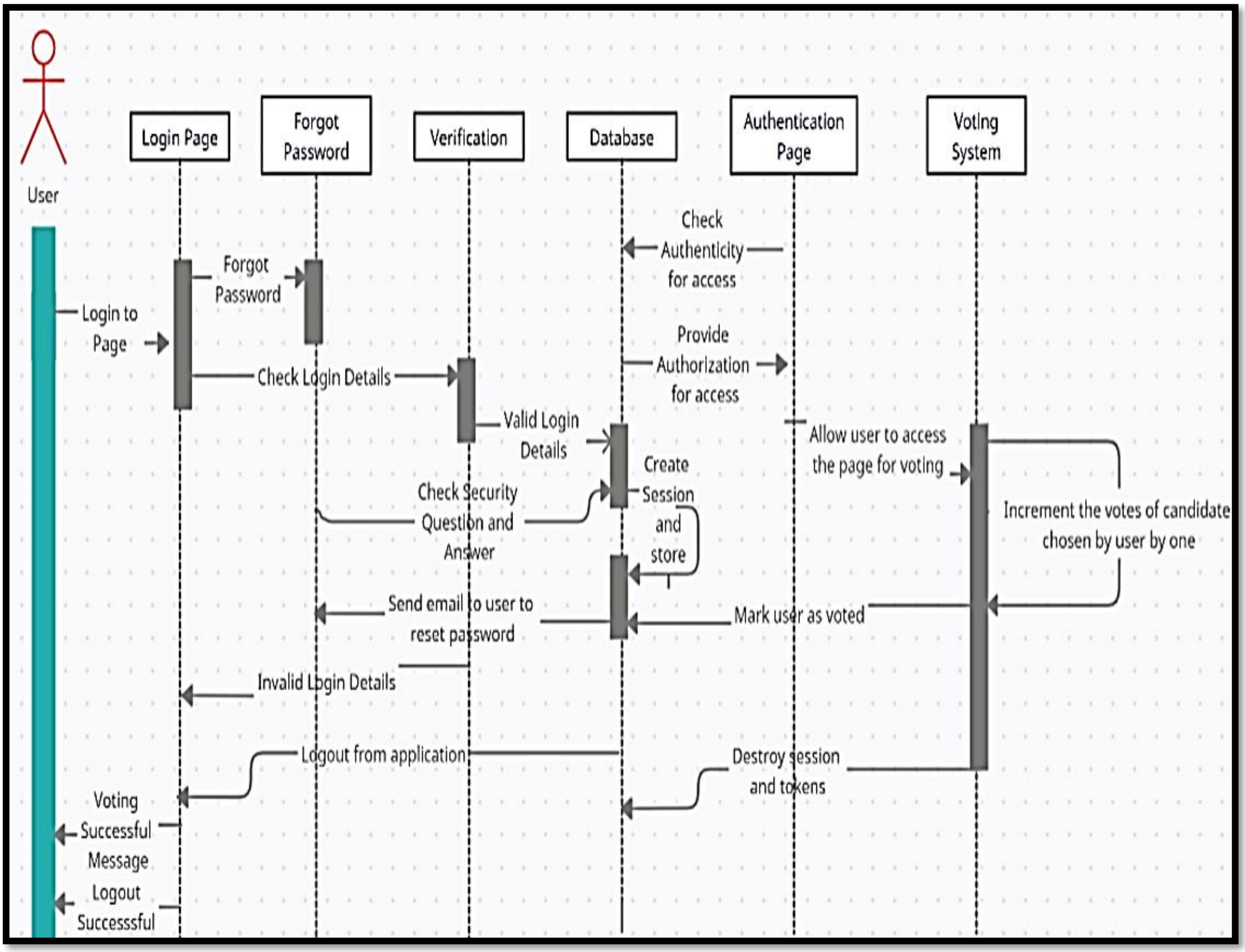
In software engineering, a class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

1. The Login Class allows user to login to the system after entering correct credentials.
2. The Verification Class verifies all data being entered into the system.
3. The User Class assigns role and directs user based on their role.
4. The Admin, Election Admin, Voter and Candidate Classes consist requisite data associated with the user and the operations they perform in the system.
5. The Registration Class allows Candidates to register for election.
6. The Authentication Class checks the voter identity using biometrics and approves them for voting.
7. The Cast Vote, Compute and Result Class collect, count and display the votes received and declare the winner.



4.6. Sequence Diagram

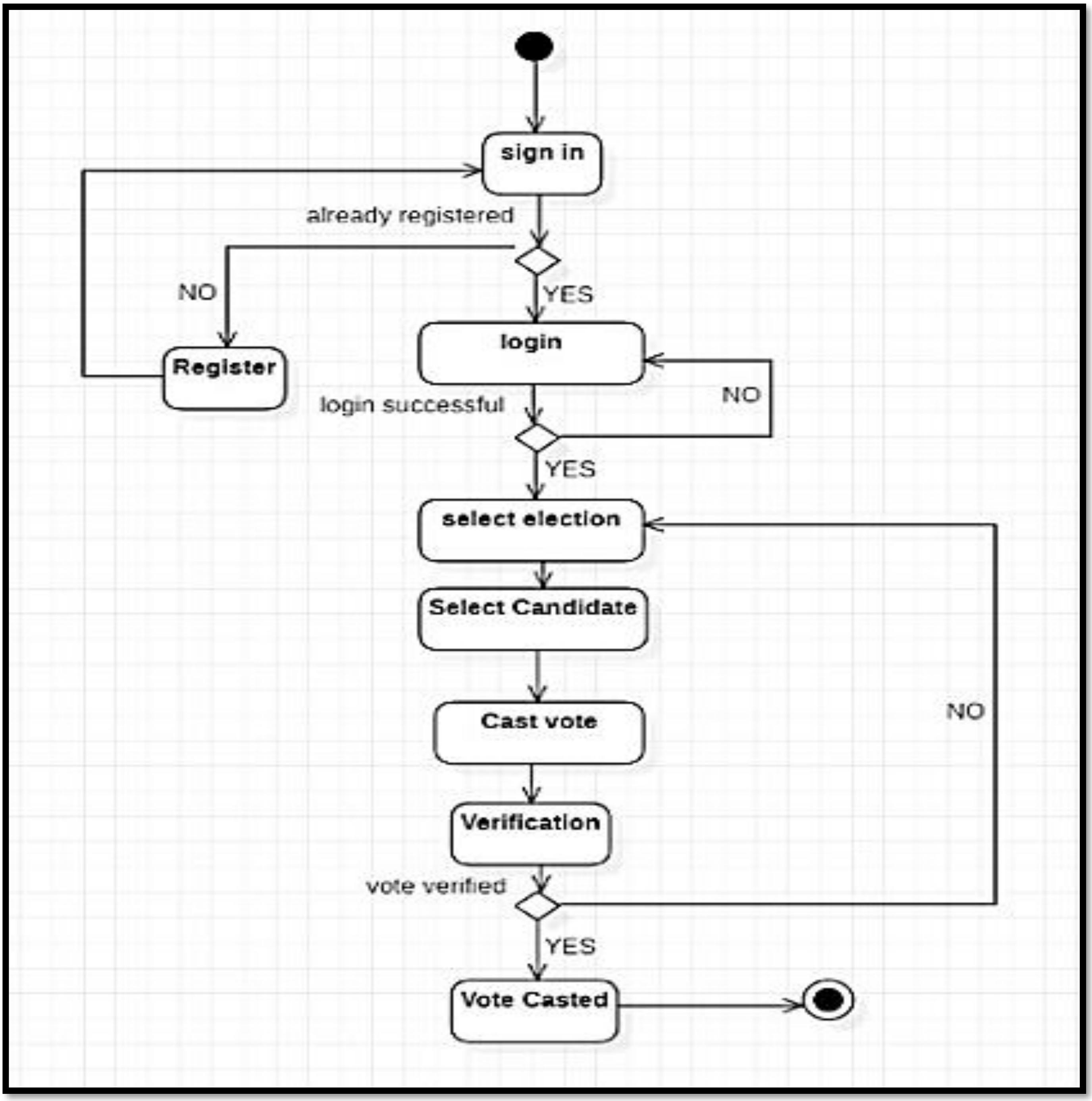
A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. The User enters their credentials which are checked with the database. If the credentials match, the user can enter the system and is directed to the voting process where their authenticity is checked. If the User is successfully authenticated, then they can process to casting their vote. The user is marked as voted within the database to prevent multi-voting and the candidate receiving the vote is incremented. Once the user has voted they can exit the voting system and log out.


User
Login Page
Forgot Password
Verification
Database
Authentication Page
Voting System
Check Authenticity for access
Forgot Password
Login to Page
Provide Authorization for access
Check Login Details
Valid Login Details
Allow user to access the page for voting
Create Session and store
Check Security Question and Answer
Increment the votes of candidate chosen by user by one
Send email to user to reset password
Mark user as voted
Invalid Login Details
Logout from application
Destroy session and tokens
Voting Successful Message
Logout Successsful

4.7. State Diagram

State chart diagram is used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagram describes the flow of control from one state to another state.

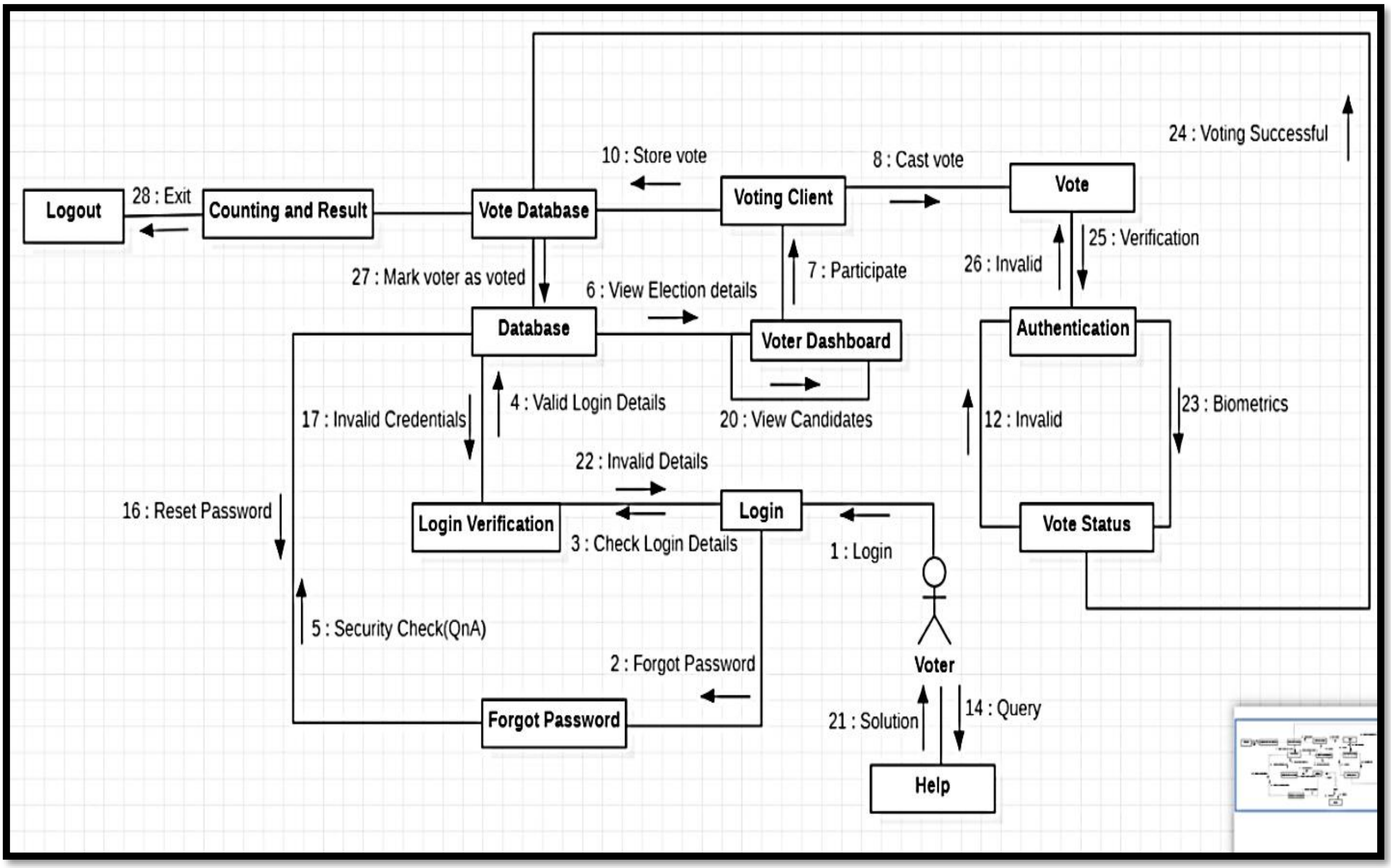
8. The user can access the system via sign in where they enter their credentials to login. In case they haven't registered, they can complete the registration process to generate credentials.
9. The voter the selects the election they want to participate in and cast their vote for the desired candidate.
10. If their identity and vote casted is verified then they can exit the system else they are redirected to election module.



4.8. Collaboration Diagram

The collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

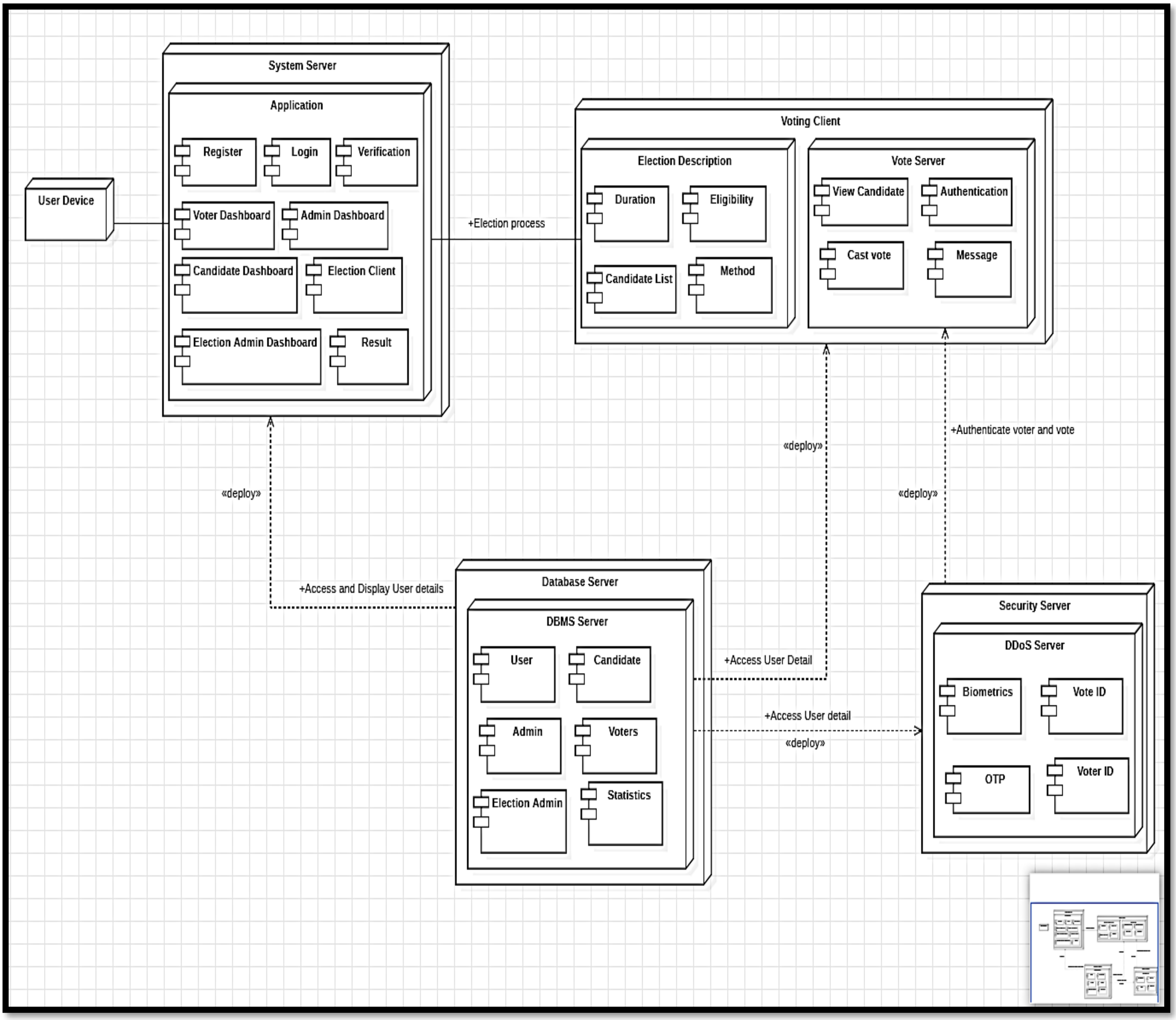
5. The Voter enters their credentials to access the system. The entered credentials are verified with the database and if they are valid then the voter is directed to the dashboard.
6. The Voter has option of forgot password in case they are unable to recall their credentials, after which they are directed to answer pre-determined security questions to generate new credentials.
7. Within voting dashboard, the voter can view election details and candidates and based on this choose which voting process to participate in.
8. The voting client displays the candidates and allows voter to cast their vote. The voter and their vote are authenticated using security measures and if successful the vote is stored in the database and the voter is marked as voted while the candidate receives the vote.
9. The stored votes are tallied and result is displayed. The voter can logout from the system.



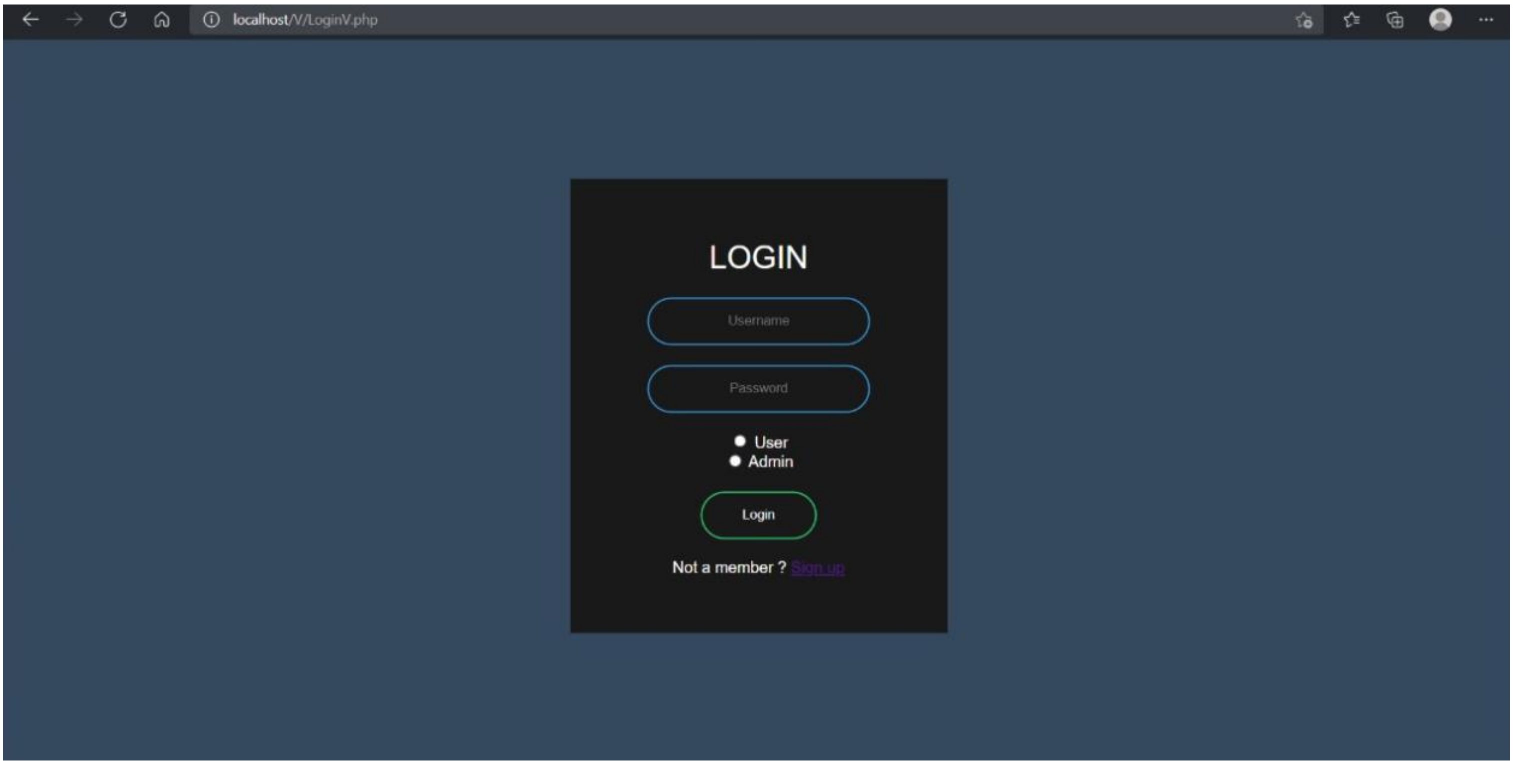
4.9. Deployment Diagram

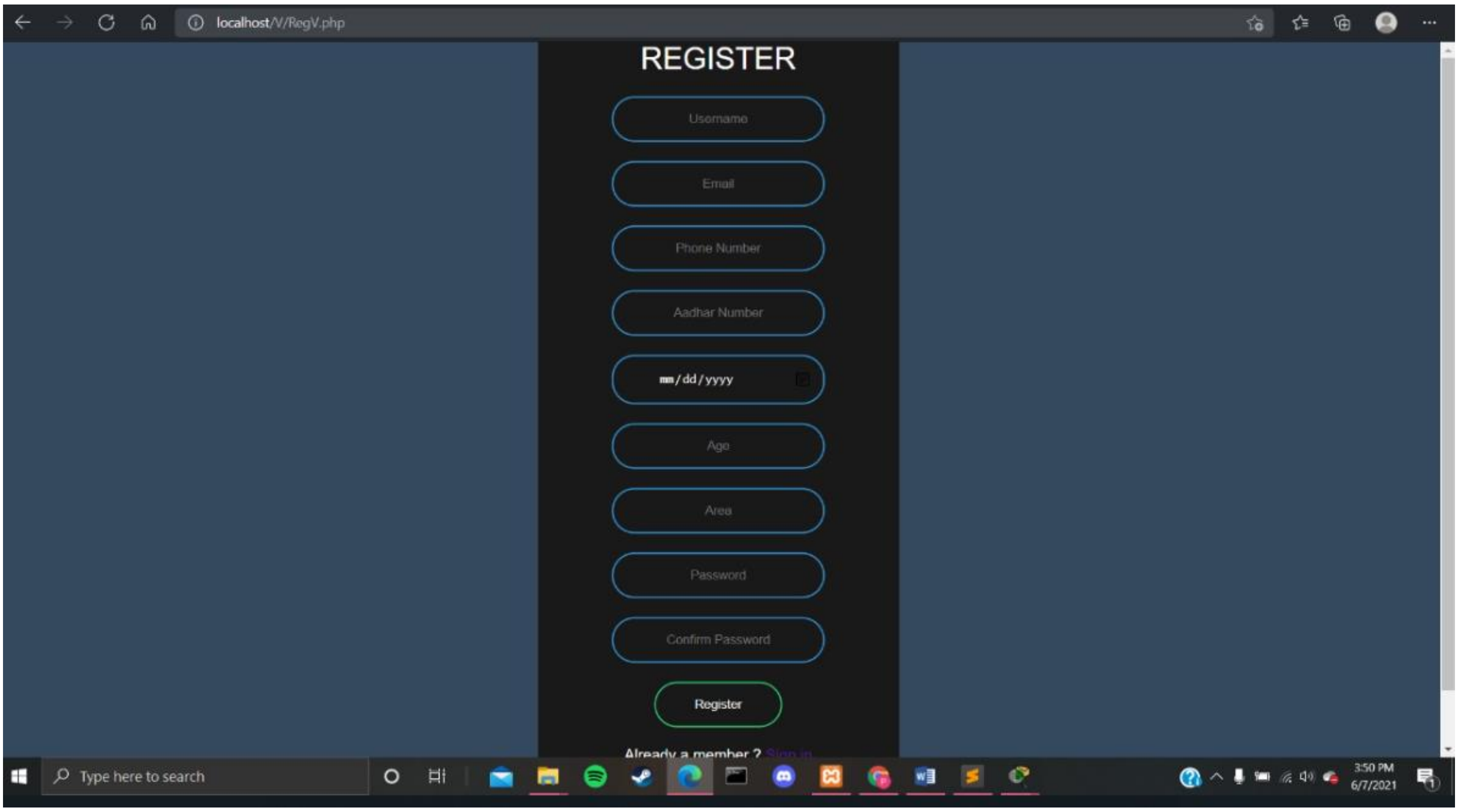
A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modelling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

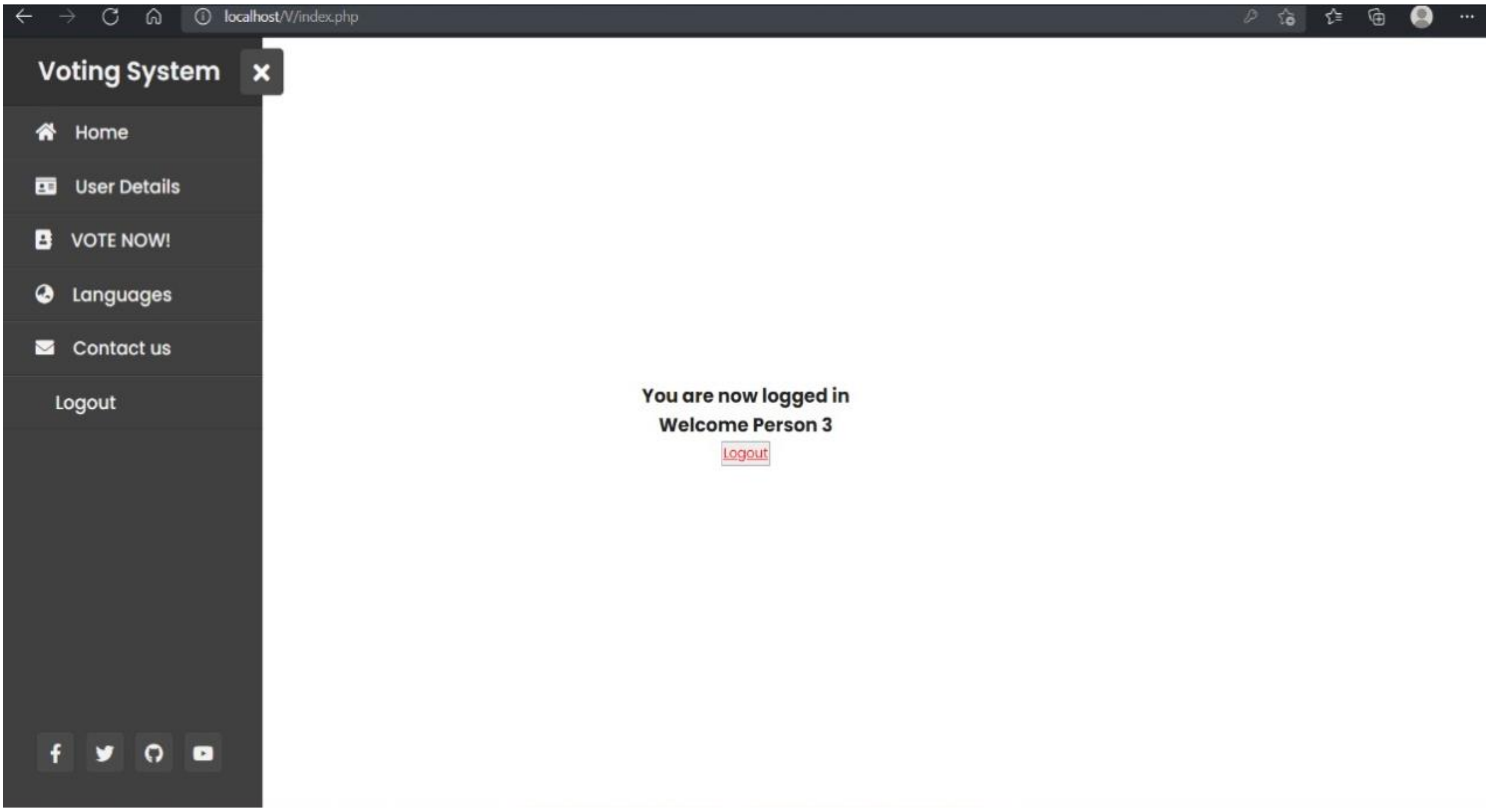
5. The User device access the system server and can register, login, view their dashboard based on their role and view result.
6. The voting client executes the voting process by accepting data from database and verifying vote and voter identity using security server.



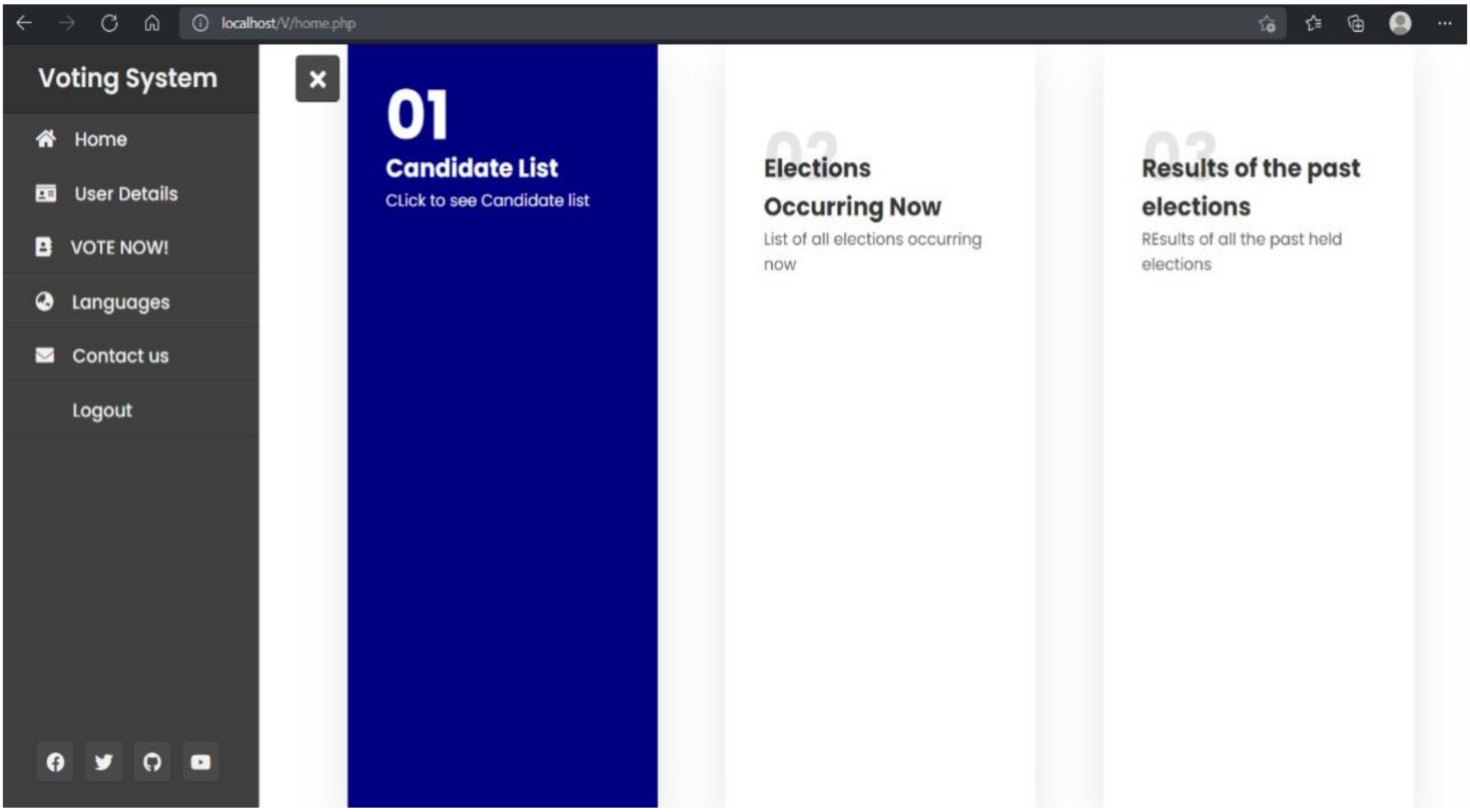
5. Demo


localhost/V/LoginV.php
LOGIN
Username
Password
● User
● Admin
Login
Not a member ? Sign up


localhost/V/RegV.php
REGISTER
Username
Email
Phone Number
Aadhar Number
mm/dd/yyyy
Age
Area
Password
Confirm Password
Register
Already a member ? Sign in
Type here to search
3:50 PM
6/7/2021



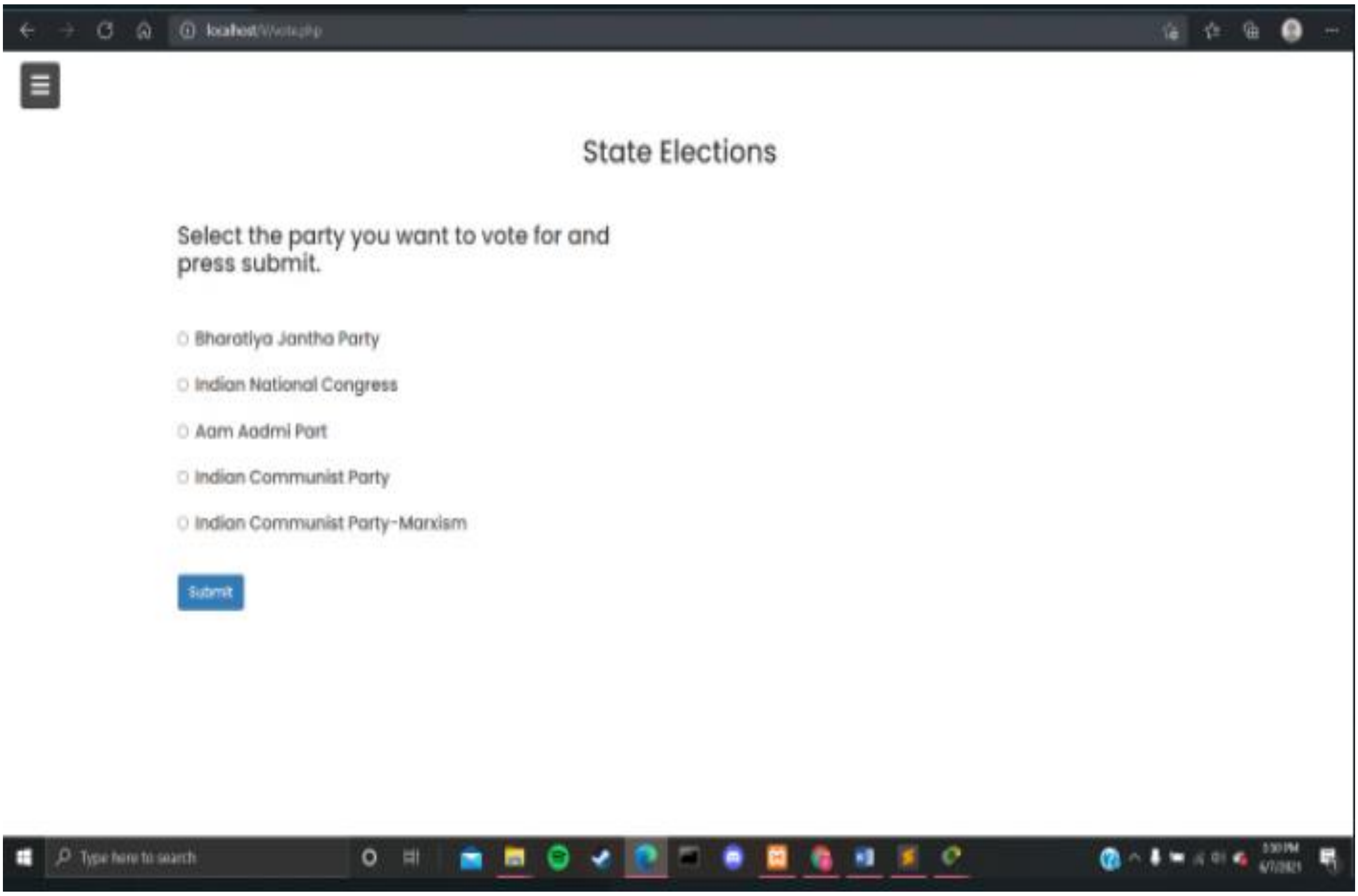
You are now logged in
Welcome Person 3
Logout



01
Candidate List
CLick to see Candidate list

02
Elections Occurring Now
List of all elections occurring now

03
Results of the past elections
REsults of all the past held elections



State Elections

Select the party you want to vote for and press submit.

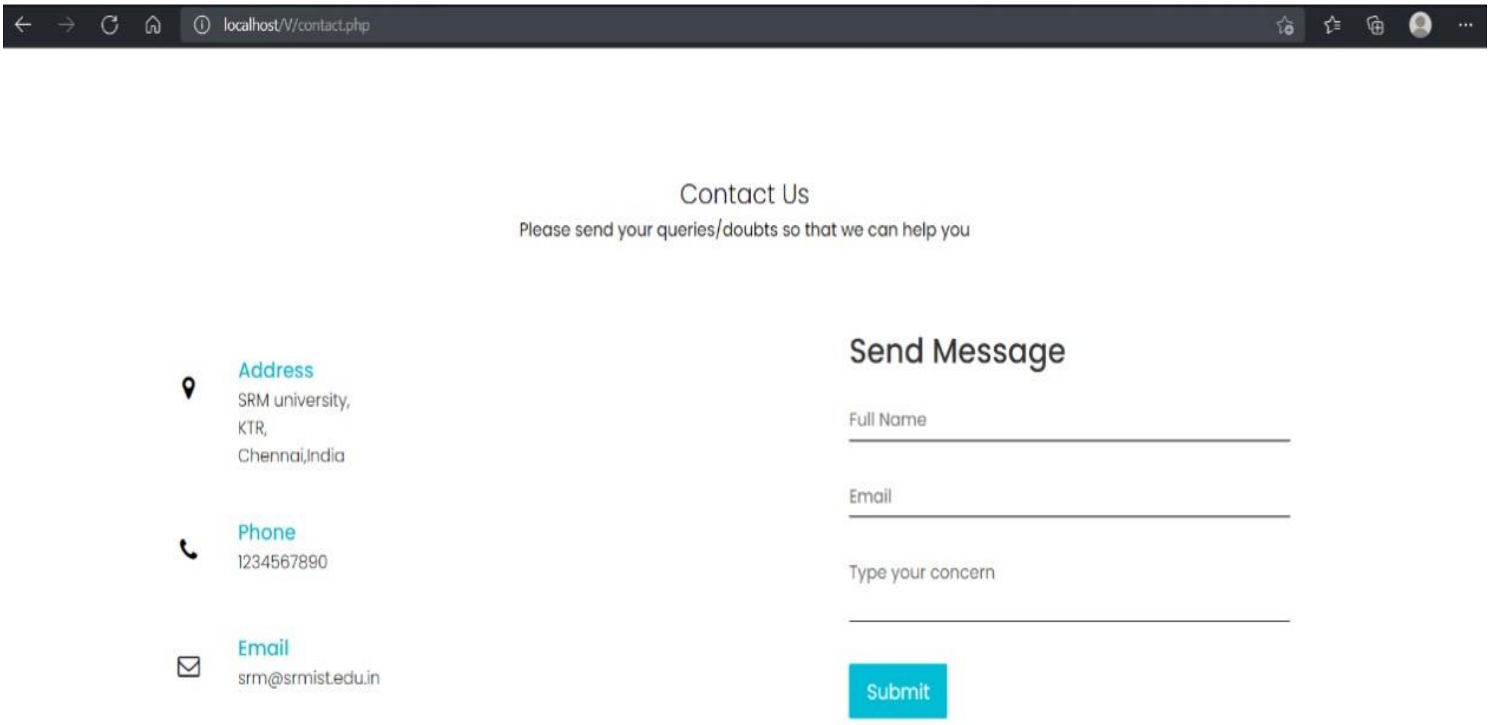
○ Bharatiya Jantha Party

○ Indian National Congress

○ Aam Aadmi Part

○ Indian Communist Party

○ Indian Communist Party-Marxism



Contact Us

Please send your queries/doubts so that we can help you

Address
SRM university,
KTR,
Chennai,India

Phone
1234567890

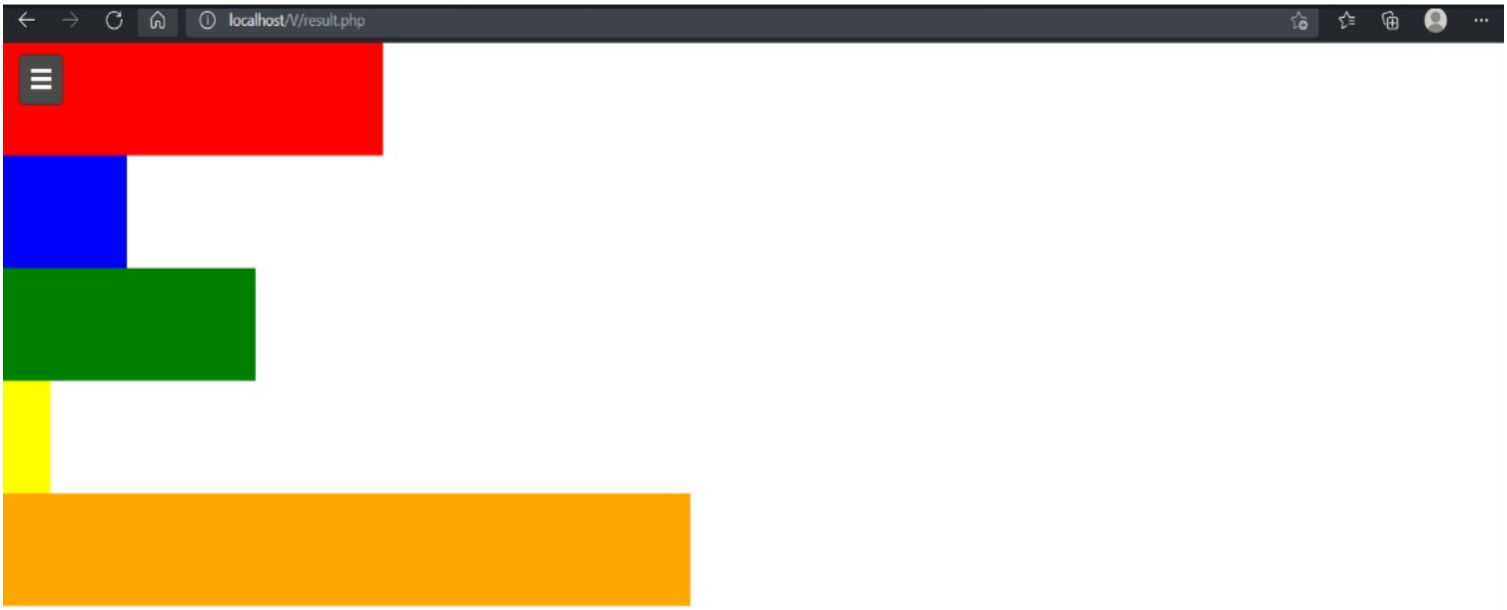
Email
srm@srmist.edu.in

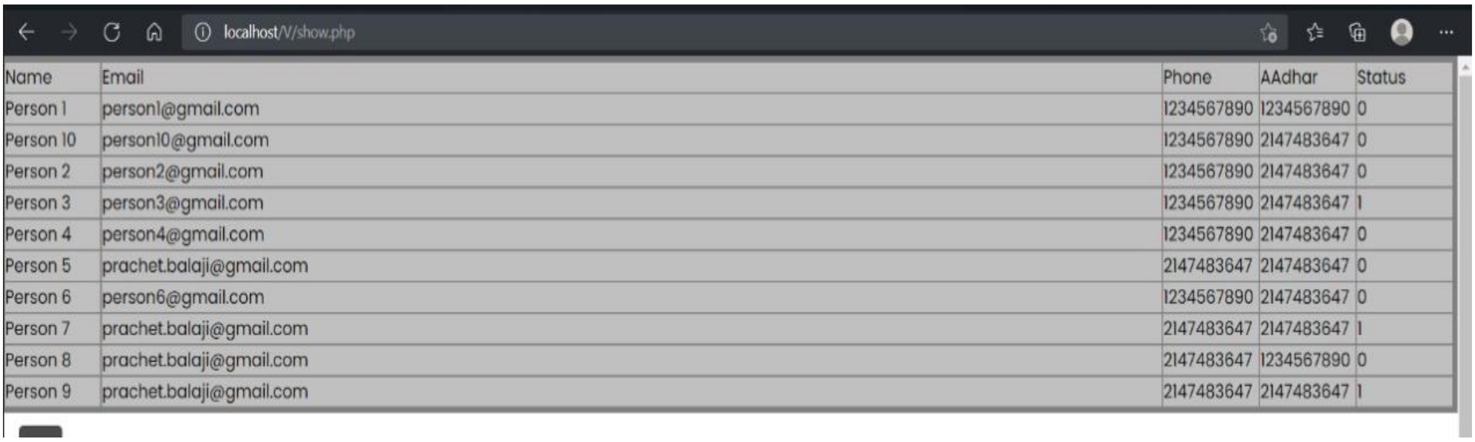
Send Message

Full Name

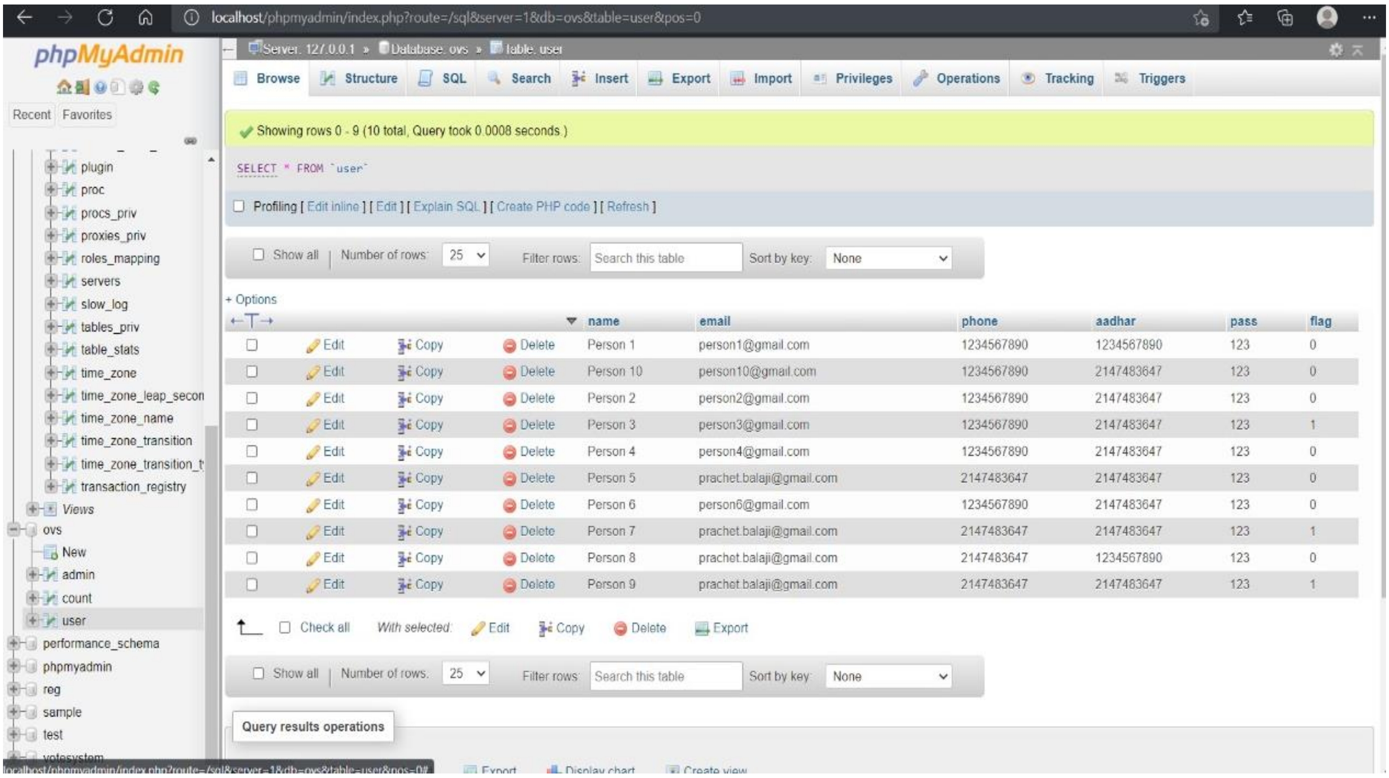
Email

Type your concern





Name	Email	Phone	AAdhar	Status
Person 1	person1@gmail.com	1234567890	1234567890	0
Person 10	person10@gmail.com	1234567890	2147483647	0
Person 2	person2@gmail.com	1234567890	2147483647	0
Person 3	person3@gmail.com	1234567890	2147483647	1
Person 4	person4@gmail.com	1234567890	2147483647	0
Person 5	prachet.balaji@gmail.com	2147483647	2147483647	0
Person 6	person6@gmail.com	1234567890	2147483647	0
Person 7	prachet.balaji@gmail.com	2147483647	2147483647	1
Person 8	prachet.balaji@gmail.com	2147483647	1234567890	0
Person 9	prachet.balaji@gmail.com	2147483647	2147483647	1



✓ Showing rows 0 - 9 (10 total, Query took 0.0008 seconds.)

```
SELECT * FROM `user`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

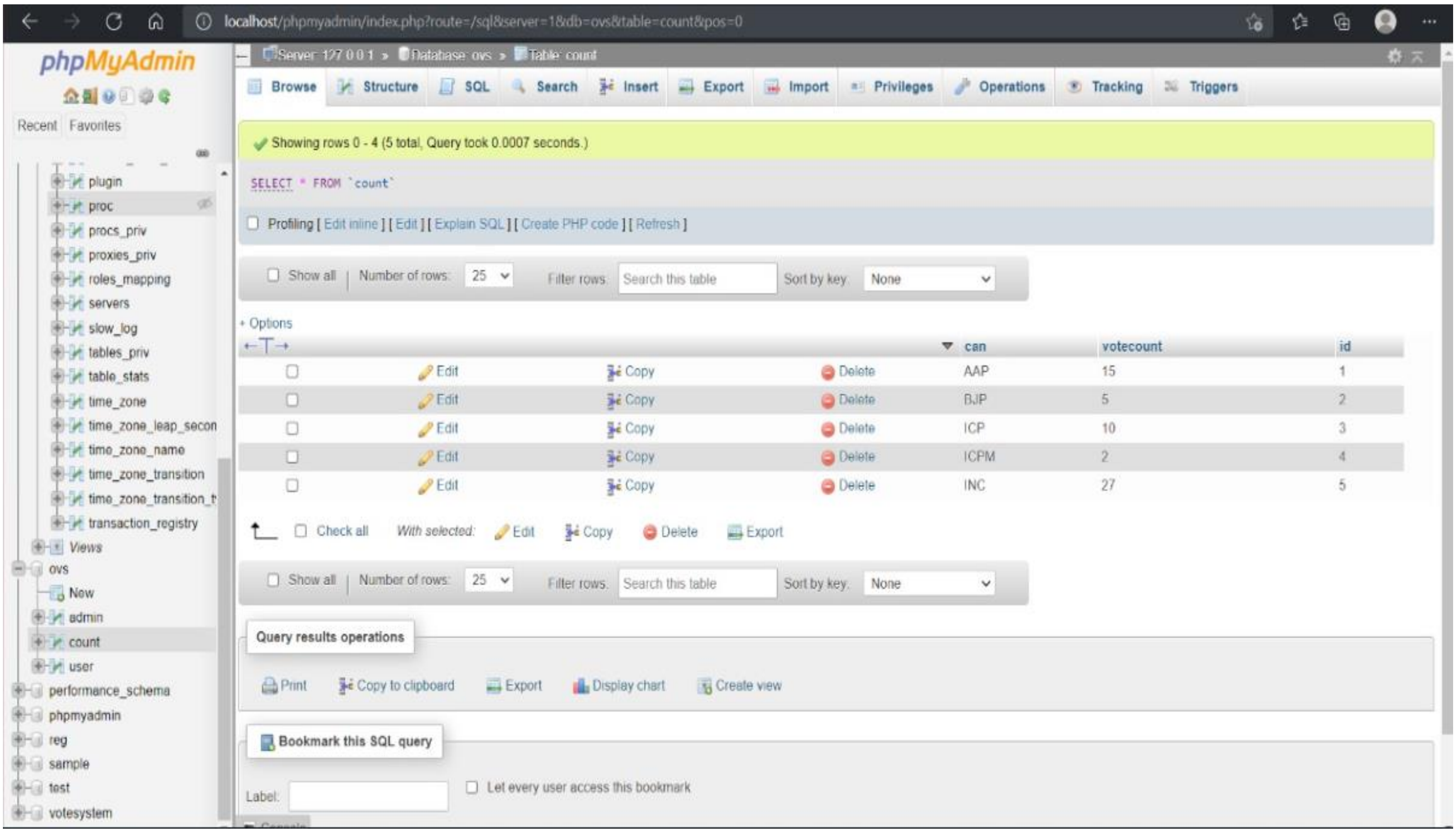
				name	email	phone	aadhar	pass	flag
□	Edit	Copy	Delete	Person 1	person1@gmail.com	1234567890	1234567890	123	0
□	Edit	Copy	Delete	Person 10	person10@gmail.com	1234567890	2147483647	123	0
□	Edit	Copy	Delete	Person 2	person2@gmail.com	1234567890	2147483647	123	0
□	Edit	Copy	Delete	Person 3	person3@gmail.com	1234567890	2147483647	123	1
□	Edit	Copy	Delete	Person 4	person4@gmail.com	1234567890	2147483647	123	0
□	Edit	Copy	Delete	Person 5	prachet.balaji@gmail.com	2147483647	2147483647	123	0
□	Edit	Copy	Delete	Person 6	person6@gmail.com	1234567890	2147483647	123	0
□	Edit	Copy	Delete	Person 7	prachet.balaji@gmail.com	2147483647	2147483647	123	1
□	Edit	Copy	Delete	Person 8	prachet.balaji@gmail.com	2147483647	1234567890	123	0
□	Edit	Copy	Delete	Person 9	prachet.balaji@gmail.com	2147483647	2147483647	123	1

□ Check all *With selected:* Edit Copy Delete Export

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

localhost/phpmyadmin/index.php?route=/sql&server=1&db=ovs&table=user&pos=0#
Export
Display chart
Create view


localhost/phpmyadmin/index.php?route=/sql&server=1&db=ovs&table=count&pos=0

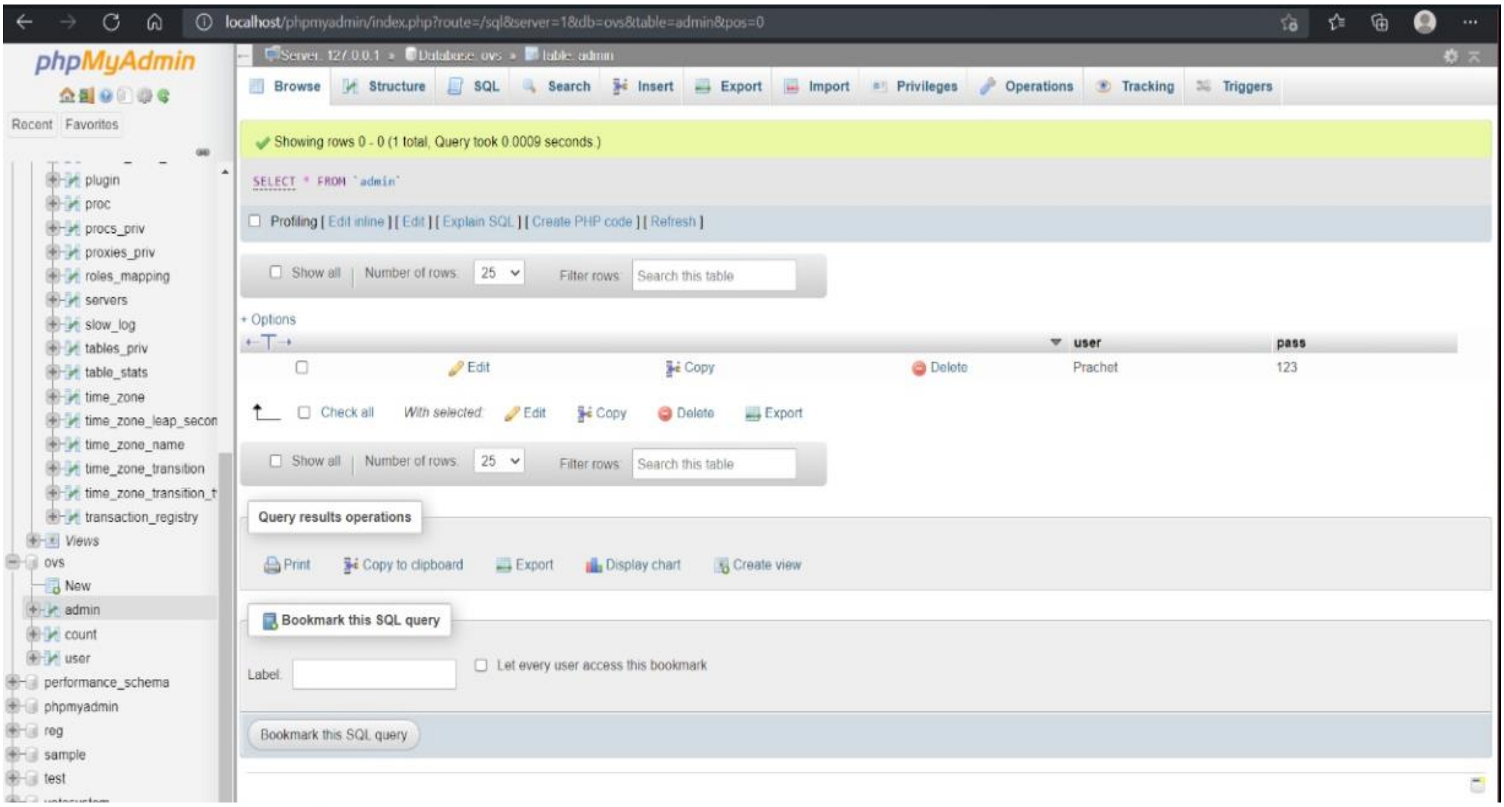
Server: 127.0.0.1 » Database: ovs » Table: count

✓ Showing rows 0 - 4 (5 total, Query took 0.0007 seconds.)

```
SELECT * FROM `count`
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

				can	votecount	id
□	Edit	Copy	Delete	AAP	15	1
□	Edit	Copy	Delete	BJP	5	2
□	Edit	Copy	Delete	ICP	10	3
□	Edit	Copy	Delete	ICPM	2	4
□	Edit	Copy	Delete	INC	27	5


localhost/phpmyadmin/index.php?route=/sql&server=1&db=ovs&table=admin&pos=0

Server: 127.0.0.1 » Database: ovs » Table: admin

✓ Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT * FROM `admin`
```

Number of rows: 25 | Filter rows: Search this table

				user	pass
□	Edit	Copy	Delete	Prachet	123

6. Test Case

6.1. Functional Test Case

1. Login and Registration Module

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Status
#T1	Verify User Login	Accept the user login if username and password match	1.User clicks on login. 2.He enters the username and password. 3.Clicks on the login button.	User should be redirected to the index page where his dashboard is shown	Successful
#T2	Verify User Registration	User is prompted to enter details to register. All Details must be entered and both the passwords should match.	1.User clicks on register 2.He enters all the required details 3.He clicks on register	If details are entered correctly then the data is stored in the database and user is redirected to login page	Successful

2. Voting Module

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Status
#T1	Voting process	User is prompted to vote for one of the options given.	1.User clicks on Vote now in the navigation bar 2.He clicks on the option he wants to vote for 3.Clicks on Vote button	If Valid user, the vote is counted into the database.	Successful
#T2	Verify User	If User has already voted, he will not be allowed to vote again	1.User has already voted 2.User clicks on Vote now in the navigation bar 3.He clicks on the option he wants to vote for 4.Clicks on Vote button	If the user has already voted then he is prompted with the message "Already Voted"	Successful

3. Admin Module

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Status
#T1	Admin Login	Admin enters his username and password and selects the admin checkbox	1.Admin clicks on login 2.Admin enters his login details. 3.Admin clicks on the admin checkbox 4.Admin clicks on login	If Valid admin, he is redirected to the admin side.	Successful
#T2	Admin view results.	Admin can view the live results of the ongoing elections	1.Admin clicks on the results tab in the navigation bar.	Admin is shown the results in different formats (tables graphs)	Successful
#T3	Admin can add or remove candidates	Admin can remove or add candidates	1.Admin clicks on the add or remove candidates 2.He is redirected to database where he can make the necessary changes	Admin can add or remove candidates form the database and make changes to the form	Successful

6.2. Non-Functional Test Case

Test Case #	Test Case	Domain	Status
#N1	Application load time should not be more than 5 secs up to 1000 users accessing it simultaneously	System Testing	Successful
#N2	Software should be installable on all versions of Windows and Mac	Acceptance Testing	Successful
#N3	Large number of people can login into a software.	System and Integration Testing	Successful

Result:

Thus, the software project was designed and documented successfully.