



STUDENT ID: AN/2021/0020

AINWIK INFOTECH

Development | Training | Placement

An ISO 9001 : 2015 Company

The council of research and training AINWIK Infotech certifies that

VIDHI VATS

has been awarded a certificate of proficiency in
PYTHON PROGRAMMING

for having successfully completed the program with
A grade

The program was conducted during the period

15/06/2021 to 25/07/2021

Program Co-ordinator



AINWIK INFOTECH LLP

(COMPLETE IT SOLUTIONS)

Issue Date : 03/08/2021

To Whom So Ever It May Concern

This is to certify that project entitled **Election Polling Meter** , design on python technology is an original work carried out by **Master of Computer Application (MCA) Student Ms Vidhi Vats** ,Enrollment Id **AN/2021/0020** of **Krishna Institute of Engineering and Technology Ghaziabad (U P)** , from **15/06/2021** to **25/07/2021** at **Ainwik Infotech Greater Noida (U P)**

The matter embodies in this project is a genuine work done by the student and has not been submitted to the university or any other university/institute for the fulfillment of the requirement of any course of study

Hope you will find the details in order



Address:- 802,8th Floor, Om Tower
Alpha-1,Commercial Belt,
Greater Noida (U.P.) 201308

Tel:-01204082305
www.ainwikinfotech.com

Ph:- +91 8467090016
+91 8178104315

ELECTION POLLING METER

**A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

by

Vidhi Vats - 1900290149108

Under the Supervision of

Ms. Neelam Rawat

KIET Group of Institution, Ghaziabad



to the

FACULTY OF COMPUTER APPLICATIONS

DR. APJ ABDUL KALAM TECHNICAL

UNIVERSITY LUCKNOW

(Formerly Uttar Pradesh Technical University, Lucknow)

JULY, 2021

CERTIFICATE

Certified that **Vidhi Vats (University Roll No 1900290149108)** have carried out the project work having “**Election Polling Meter**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Ms. Neelam Rawat

Signature of External Examiner

Associate Professor

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Dr. Ajay Srivastava

Professor & Head, Department of Computer Applications

KIET Group of Institutions, Ghaziabad

ABSTRACT

This project involves the design of online voting system which is a python flask-based web application made for a secure implementation of Indian elections made online.

The Polling Meter has been designed to provide security to the polls by insulating the data of user within a time stamp in an offline database. The data can only be accessed by Flask application. The web application has been developed free of any vulnerability. So, it is secured from any cyber attacks making it almost impossible for attackers to hack into the user information and access it.

As soon the user opens the site, the login page appears. Aadhaar No. is required to send an OTP to the respective mobile number linked to it. The application fetches the data from login screen by using Jinja template that makes the application free of any XSS attacks.

As the user requests for the OTP, the engine generates an OTP and sends it to the user's contact number. The OTP is then hashed into some certain hashing format and inserted into the database. Once it has been hashed and inserted in the database, a time count of 2 minutes starts after which the OTP will be automatically deleted from the database.

The expertise of this system is being developed to make it available only to the Government, as it requires the database of Aadhar card to select the citizen's eligibility for voting and use their data to perform the polling. Hence, we'll make sure that all the manipulations and calculation done by the engine is 100% accurate.

ACKNOWLEDGMENT

I am very grateful to my Project guide Mr. Sulabh Aggarwal for his invaluable guidance and assistance, without which the accomplishment of the task would have never been possible. We also thank her for giving this opportunity to explore into the real world and realize the interrelation of master of computer application without which a society can never progress. In our present project we have chosen the topic “**Election Polling Meter**” which aims to make voting in India digital as well as secure. We believe in this technology which is aimed at helping the mankind. In the end I will also like to express my thankfulness to my teachers.

Date:

Vidhi Vats (1900290149108)

TABLE OF CONTENTS

Certificate	ii
Abstract	iii
Acknowledgement	iv
Table of content	v-vii
List of Figures	vii
CHAPTER 1 Literature Review	1-4
1.1 Executable HTML	1
1.2 Python programming	1-2
1.3. Computing with python	2
1.4 HTML5	2
1.5 HTML, CSS and Java script	2
1.6 Java script for Data science	3
1.7 SQLITE3	3
1.8 Software engineering	3
1.9 E-voting system	4
CHAPTER 2 Introduction	5-10
2.1 Polling meter	5
2.1.1 Web application	6
2.1.2 Components of web application	6-7
2.2 Old methods of voting	8
2.3 Disadvantages of present voting system	8
2.4 why polling meter	8
2.5 Problem definition	9
2.6 Objective	9-10
2.7 Advantages of polling system	10

CHAPTER 3 Requirement specification	11-17
3.1 Hardware Requirement	11
3.2 Software Requirement	11
3.2.1 Languages used	11
3.2.1.1 Python flask	11-12
3.2.2 Javascript	12
3.2.3 HTML	13
3.2.4 CSS	13-14
3.3 Technologies used	14
3.3.1 Blockchain	14-15
3.2.2 SQLite	15
3.4 Modules used	15
3.4.1 Flask	15
3.4.2 SQLAlchemy	15-16
3.4.3 Flask cors	16-17
3.4.4 Way2 SMSAPI	17
CHAPTER 4 Working	18-24
4.1 Login page	18
4.1.1 OTP	19
4.1.2 Generating OTP	19-21
4.2 Polling page	21-22
4.2.1 Voting process	22
4.2.2 Security parameters	23
4.3 Thank you page	23-24

CHAPTER 5 Design	25-31
5.1 Polling through flowchart	25
5.2 System Design	25
5.2.1 Use case	25-26
5.2.1.1Voters role use case	27
5.2.1.2 Admin role use case	27
5.2.1.3 Voting process use case	28
5.2.2 Sequence diagram	28
5.2.2.1 Sequence diagram of reg.	29
5.2.2.2. Sequence diagram of voting	30
5.2.3 ER model	30-31
CHAPTER 6 Coding	32-43
6.1 Frontened	32
6.1.1 Login page	33
6.1.2 Polling page	34-35
6.1.3 Thank you page	36-37
6.2 Backened	38
6.2.1 Some important codes	38-43
CHAPTER 7 Testing	44-47
7.1 Introduction	44
7.2 Types of testing	44-47
CHAPTER 8 Conclusion and future scope	49-45
8.1 Conclusion	49
8.2 Future scope	49-50
CHAPTER 9 References	51-52

List of Figures

Fig 1 Polling meter	5
Fig 2 Why polling meter	9
Fig 3 Objective	10
Fig 4. Login page	19
Fig 5 Generate OTP	21
Fig 6 Polling Page	22
Fig 7 Voting process page	23
Fig 8 Thank you page	24
Fig 9 Flowchart	27
Fig 10 Voters role use case	28
Fig 11 Admin role use case	28
Fig 12 Voting process use case	29
Fig 13 Sequence Diagram of registration	30
Fig 14 Sequence Diagram of voting process	31
Fig 15 ER model	32

CHAPTER 1

LITERATURE REVIEW

1.1 Executable HTML

Batalas[1] Executable HTML is a collection of custom HTML elements that offer document authors the ability to write and execute procedural logic in HTML5. We discuss the motivation of this approach, which lies in the composition of formal Ambulatory Assessment protocols for clinical psychology research, and provide an overview of the software's inner workings.

Khan, V. J[2] This approach is enabled by the execution semantics of xUML and is based on automatic translation from xUML to S/R, the input language of the COSPAN model checker (R.H. Hardin et al., 1996). Model transformations are applied to reduce the state space of the resulting S/R model that is to be verified by COSPAN. An xUML level logic for specifying properties to be checked is defined.

1.2 Python Programming

Van Rossum[3] Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms

Zelle[4] The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <http://www.python.org/>, and can be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

1.3 Computing with Python

Herman[5] The discovery of music recording was a transformative event in human history. Here was an invention that almost everyone instantly desired. Recorded music can be enjoyed by people who cannot read or write, who cannot themselves play an instrument or sing. It is hard to look back on the first crude recording technologies with their scratchy noises and wobbly playback and understand how people could be satisfied. An amazing accumulation of clever technology has improved the quality of recording and playback to a remarkable extent. Even in the early days, engineers worked hard to reduce noise and ensure good fidelity in reproducing original sound.

1.4 HTML5

Lawson[5] HTML5 is the latest and most enhanced version of HTML. Technically, HTML is not a programming language, but rather a markup language.

1.5 HTML,CSS, and JavaScript

Fajfar[6] HTML provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and JavaScript. CSS is used to control *presentation, formatting, and layout*. JavaScript is used to control the *behavior* of different elements.

1.6 JavaScript for Data Science

Gans, M[7] JavaScript is well-known for its abilities in web-development, but how would those abilities carry over into an application in Data Science? First and foremost, we should consider that JavaScript is not a statistical programming language. The language was pretty much created for web-development, which the language excels at. This means that JavaScript from a typical Statistician's point of view might be a bit hard to grasp.

1.7 SQLite

Nemetz[8] An increasing number of programs like browsers or smartphone apps are using SQLite3 databases to store application data. In many cases, such data is of high value during a forensic investigation. Therefore, various tools have been developed that claim to support rigorous forensic analysis of SQLite database files, claims that are not supported by appropriate evidence. We present a standardized corpus of SQLite files that can be used to evaluate and benchmark analysis methods and tools.

1.8 Software Engineering

Al-Sarayreh[9] Software engineering, a fairly recent engineering discipline, is still evolving without a wide consensus on a body of fundamental principles as in traditional engineering fields with their own long-established principles originating from physics, chemistry and mathematics. This paper reports on a systematic mapping study (SMS) that identified 30 papers and books from 1969 to January 2020, each proposing their own sets of software engineering principles (SEP). Within these studies a total of 592 SEP were proposed, these studies were reviewed and classified on the basis of four mapping questions examining publication trends, use of explicit criteria for the proposed SEP, whether authors clearly described a methodology to come up with the proposed SEP, and the applicability of SEP throughout the development process.

1.9 E-Voting System

Pawlak[10] There are many existing voting solutions which have different benefits and issues. The most significant ones are lack of transparency and auditability. Recently developed blockchain technology may be a solution to these issues. The paper describes the use of intelligent agents and multi-agent system concept for Auditable Blockchain Voting System (ABVS), which integrates e-voting process with blockchain technology into one supervised non-remote internet voting system which is end-to-end verifiable.

CHAPTER 2

INTRODUCTION

2.1 POLLING METER

Polling Meter is a python flask-based web app, it is made for secured implementation of Indian elections made online. We provide security to the polls by insulating the data of user within a time stamp in an offline database. This database can only be accessed by Flask application.

The web app has been developed free of any application vulnerability. So, it tricks the attackers and makes it almost impossible for them to get accessed to the user information.

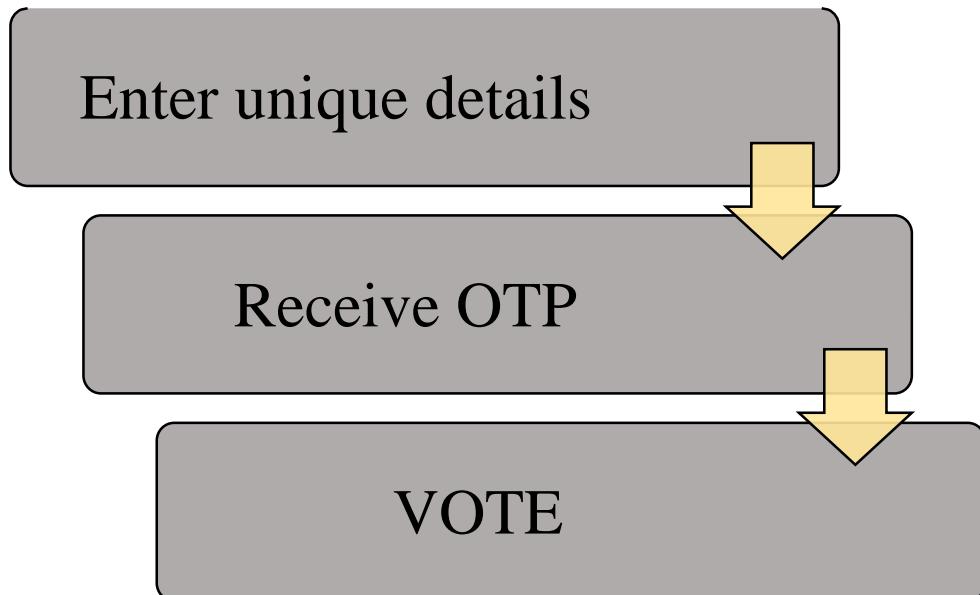


Figure 1Polling meter

Polling Meter has 3 major components:

Polling Meter interface.

Application Backend.

Database

2.1.1 Web Application

Our Polling Meter is a web application. So here are some basics and descriptive information about Web Application.

Talking in terms of computing, a web application or a web app can be termed as a client–server computer program where the client, including the user interface and client-side logic, runs in a web browser.

2.1.2 Components of Web Applications

Any web application, big or small, contains these major components:

View Layer - When you consider an MVC application, the View layer component gives an interface to the application. Regardless if it is for users with a browser or for another application using Web services. View layer is the bridge for getting the data in and out of the application.

It does not have business logic, like calculating interest for a banking application or storing items in a shopping cart for an online catalogue.

It also does not contain any code for existing data to or retrieving data from a data source. Business logic is managed by the Model layer. View layer is more focused on the interface.

Business Layer - It is also known as Business Logic or Domain Logic or Application Layer. The function of the business layer is to accept user requests from the browser, processes them, and determine the routes through which the data will be accessed.

The workflows by which the data and requests travel through the back end lay encoded in a business layer.

Data Access Layer- This layer is built to keep the code you use to pull data from your data store like database, flat files, or web services separate from business logic and presentation code.

So even if you have to change data stores, you don't end up rewriting the whole thing. There are many ORM frameworks that are blending the DAL with other layers which makes development easy during web application development services.

Error handling, security, logging - When you build a web application, people generally tend to focus on the end-goal, building and testing only for situations when things go right. Alas! Things rarely go right all the time in the real world.

This is where error handling is a vital part of any application's user experience. And, if it is done well, it can leave your users feeling informed and properly considered.

2.2 Old Methods of Voting

Paper-based voting: The voter gets a blank ballot and use a pen or a marker to indicate he want to vote for which candidate.

Lever voting machine: Lever machine is peculiar equipment, and each lever is assigned for a corresponding candidate. The voter pulls the lever to poll for his favorite candidate.

Direct recording electronic voting machine: This type, which is abbreviated to DRE, integrates with keyboard; touch screen, or buttons for the voter press to poll.

Punch card: The voter uses metallic hole-punch to punch a hole on the blank ballot

Optical voting machine: After each voter fills a circle correspond to their favourite candidate on the blank ballot, this machine selects the darkest mark on each ballot for the vote then computes the total result.

2.3 Disadvantages of Present Voting System

- The machines are not able to recognize the eligibility of a candidate, so the corrupted officers may misguide the people.
- This may increase the count of voting.
- After voting, if any technical problem or damage occurs, it may lead to re-election.
- During transportation of the machine, the person in charge can change the status of the machine and even may destroy.
- The system is not a cost effective. Since we need security, it requires safe place for counting.
- The voting takes place where the machine is located.

2.4 Why Polling Meter?

The idea of Polling Meter came from the concept of Digital India. If all the systems are being digitised then why not the Indian Elections?

It would save a lot of money and make it way easy for the citizen as well as Election Commission. For example, imagine a place or riot or a pandemic during election days. People fear walking to the EVMs. With Polling Meter in command, they can vote from their respective home. not able to go to your home for election? We have got it all sorted. making the votes more secure and easy.

Here are some reasons why people do not vote in India.

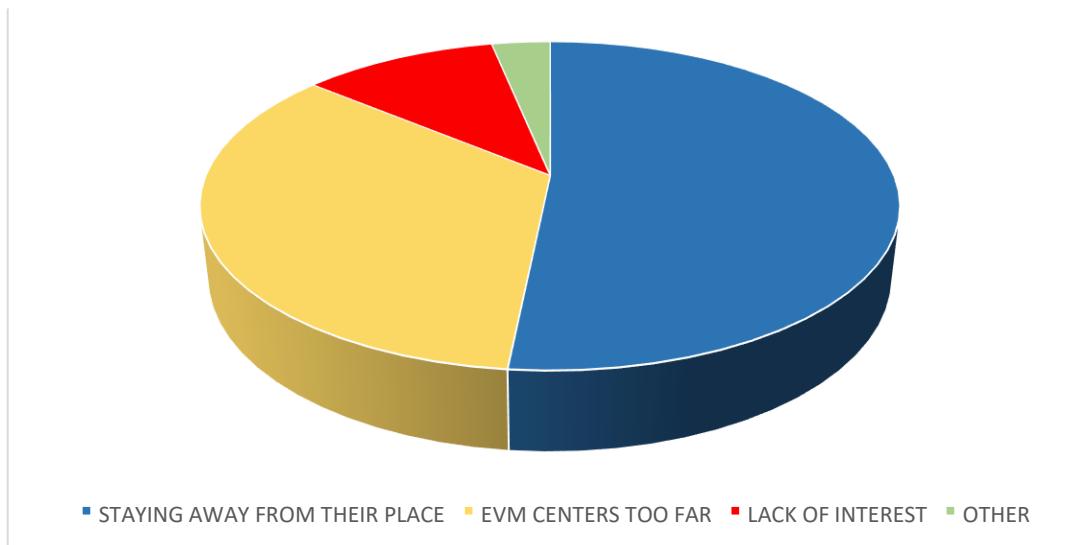


Figure 2 Why polling meter

2.5 Problem Definition

Online voting system will be increasingly replacing the traditional EVM system. These systems can make the voting process more convenient and may therefore lead to improved turnout. Online recording and counting of votes could be faster, more accurate and less labour intensive.

The goal of the Online voting as a product is to automate the voting process, help in solving fraud problems, decreasing the voting time, and process of counting.

2.6 Objective

The aim of Polling Meter is to provide a secure system for the citizen to vote through an online platform which is much more easier and time saving than going to the EVMs.

The drastic decline in the voters is a point of concern. It seems like the promotional activities and advertisements by the Election Commission to make voting look “cool” by people and celebrating proudly showing their inked finger haven’t really helped. It’s time we go digital.

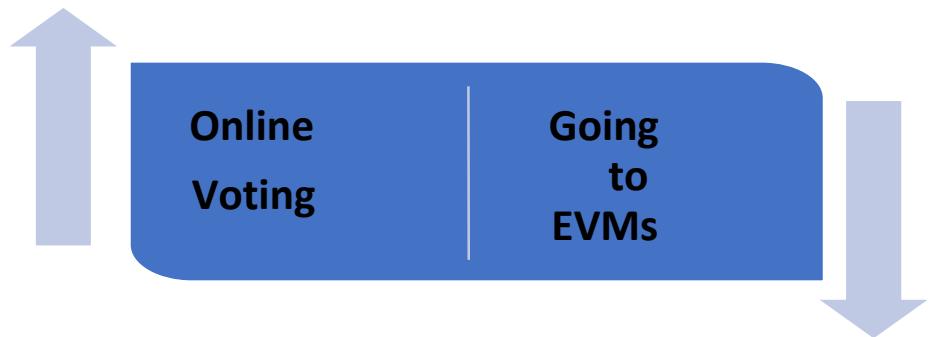


Figure 3 Objective

2.7 Advantages of Polling Meter

- Online voting would save a lot of money and would encourage the citizen to vote as it would be way much easier than before.
- People fear walking to the EVM. With Polling meter in command, they can vote from home.
- Many people are away from home during election day. It has all got covered.
- Making the votes more secure and easier.
- Keep a record of actually who and how many voted.

CHAPTER 3

Requirements Specifications

3.1 HARDWARE REQUIREMENTS

RAM:	4GB
Operating system	Windows

3.2 SOFTWARE REQUIREMENTS

3.2.1 LANGUAGES USED

The various programming and scripting languages used to process the concept are: Python Flask, JavaScript, HTML and CSS.

3.2.1.1 Python Flask

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Flask supports extensions that can add application features as if they were implemented in Flask itself.

Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.



3.2.2 JavaScript

JavaScript (JS) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document.

JavaScript (JS) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. Here JavaScript has been used to bake the website dynamic so no user faces any problem no matter what device he uses.



3.2.3 HTML

HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

Hyper Text: Hyper Text simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hyper Text is a way to link two or more web pages (HTML documents) with each other.

Markup language: A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.



3.2.4 CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.



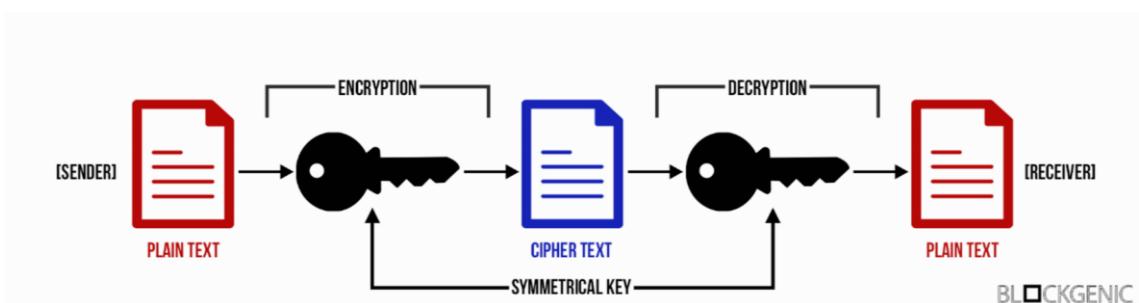
3.3 TECHNOLOGIES USED

The technologies used in the process for checking up with the security of data and other credentials are: Adobe Illustrator, Blockchain and Cryptography. These technologies are the backbone of the web application.

3.3.1 Blockchain

Blockchain is a growing list of records, called *blocks*, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree).

By design, a blockchain is resistant to modification of the data. It is "an open, distributed ledger that can record transactions between two parties efficiently and in a verifiable and permanent way".



Blockchain is an innovative application model that integrates distributed data storage, peer-to-peer transmission, consensus mechanisms, digital encryption technology and other computer technologies. It is decentralized, secure, and Information disclosure. In the blockchain, digital encryption technology has a core position. The security of user information and transaction data is a necessary condition for the promotion of blockchain.

3.3.2 SQLite:

The web application is running sqlite3 database which is a file system-based database, so the database is stored locally on the server.

It is an in-process library that implements a self-contained, server less, zero configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

3.4 MODULES USED

The main modules used here are mainly: Flask (used for backend purpose of web application), SQLAlchemy (client or database), Flask Cors (to stop XXS attack) and Way2sms API (for generating and receiving of OTP).

3.4.1 Flask

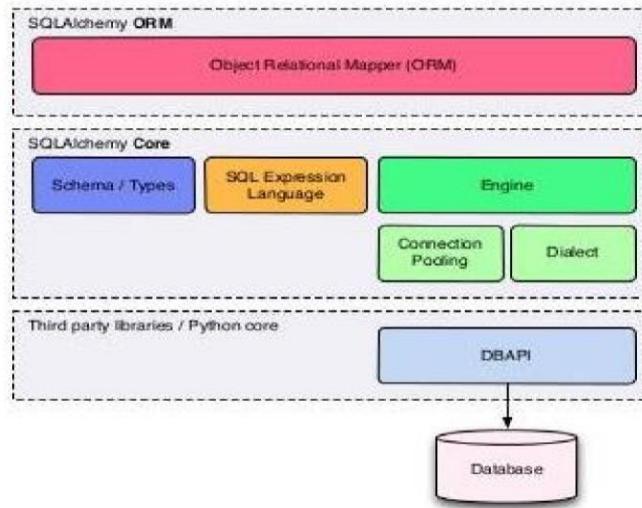
Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web based calendar application or a commercial website.

3.4.2 SQLAlchemy

SQLAlchemy is a popular SQL toolkit and Object Relational Mapper. It is written in Python and gives full power and flexibility of SQL to an application developer.

SQLAlchemy Overview

SQLAlchemy consists of the **Core** and the **ORM**



It is an open source and cross-platform software released under MIT license. SQLAlchemy is famous for its object-relational mapper (ORM), using which classes can be mapped to the database, thereby allowing the object model and database schema to develop in a cleanly decoupled way from the beginning. ORM (Object Relational Mapping) is a programming technique for converting data between incompatible type systems in object-oriented programming languages.

Usually, the type system used in an Object Oriented (OO) language like Python contains non-scalar types. These cannot be expressed as primitive types such as integers and strings. Hence, the OO programmer has to convert objects in scalar data to interact with backend database. However, data types in most of the database products such as Oracle, MySQL, etc., are primary. In an ORM system, each class maps to a table in the underlying database. Instead of writing tedious database interfacing code yourself, an ORM takes care of these issues for you while you can focus on programming the logics of the system.

3.4.3 Flask Cors

Flask-CORS. A Flask extension for handling Cross Origin Resource Sharing (CORS), making cross-origin AJAX possible. This package has a simple philosophy, when you want to enable CORS, you wish to enable it for all use cases on a domain. This means no mucking around with different allowed headers, methods, etc.

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served.^[1] A web page may freely embed cross-origin images, stylesheets, scripts, iframes, and videos.^[2] Certain "cross-domain" requests, notably Ajax requests, are forbidden by default by the same-origin security policy.

CORS defines a way in which a browser and server can interact to determine whether it is safe to allow the cross-origin request.^[3] It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests.

3.4.4 Way2sms API

Way2SMS helps users to send SMS to any mobile in the country. Sending messages from Way2SMS is free, easier, and personal; above all messages are delivered in just few seconds. The OTP will automatically be generated by this API and will send it to the respective voter as per the Aadhaar information.

Later the OTP gets automatically deleted and becomes invalid after 2 minutes



CHAPTER 4

WORKING

4.1 LOGIN PAGE

A login page is a web page or an entry page to a website that requires user identification and authentication, regularly performed by entering a username and password combination.

Logins may provide access to an entire site or part of a website. Logging in not only provides site access for the user, but also allows the website to track user actions and behaviour. Logging off a webpage or site may be manual by the user or they can occur automatically when certain conditions (such as closing the page, turning off the computer, a long-time delay, etc.) occur. First the user enters its Aadhaar ID on the login page. The application fetches the data from login screen by using JINJA template that makes the application free of any XSS attacks.

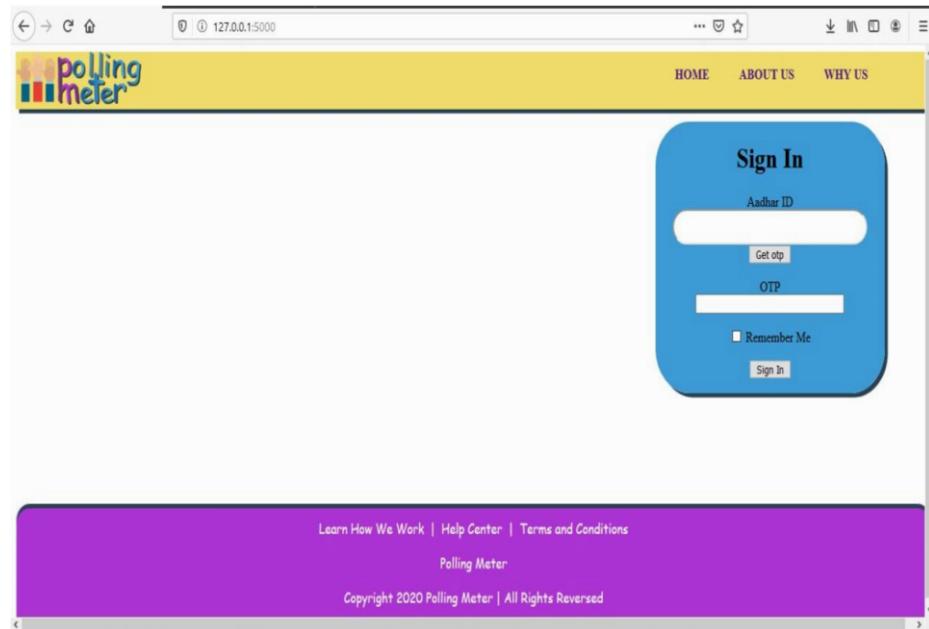


Figure 4Login page

4.1.1 OTP (One Time Password)

A one-time password (OTP), also known as one-time pin or dynamic password, is a password that is valid for only one login session or transaction, on a computer system or other digital device.

OTPs avoid a number of shortcomings that are associated with traditional (static) password-based authentication; a number of implementations also incorporate two-factor authentication by ensuring that the one-time password requires access to something a person has (such as a small keyring fob device with the OTP calculator built into it, or a smartcard or specific cell phone) as well as something a person knows (such as a PIN).

The most important advantage that is addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attacks. This means that a potential intruder who manages to record an OTP that was already used to log into a service or to conduct a transaction will not be able to abuse it, since it will no longer be valid.

A second major advantage is that a user who uses the same (or similar) password for multiple systems, is not made vulnerable on all of them, if the password for one of these is gained by an attacker.

A number of OTP systems also aim to ensure that a session cannot easily be intercepted or impersonated without knowledge of unpredictable data created during the previous session, thus reducing the attack surface further. OTPs have been discussed as a possible replacement for, as well as enhancer to, traditional passwords. On the downside, OTPs are difficult for human beings to memorize. Therefore, they require additional technology to work.

4.1.2 Generating OTP

OTP generation algorithms typically make use of pseudo randomness or randomness, making prediction of successor OTPs by an attacker difficult, and also cryptographic hash functions, which can be used to derive a value but are hard to reverse and therefore difficult for an attacker to obtain the data that was used for the hash.

A common technology used for the delivery of OTPs is text messaging. because text messaging is a ubiquitous communication channel, being directly available in nearly all mobile handsets and, through text-to-speech conversion, to any mobile or landline telephone, text messaging has a great potential to reach all consumers with a low total cost to implement.

One-time Password (OTP) SMS can be used to provide an additional layer of security for your clients. Unlike static passwords, OTPs are not susceptible to replay attacks. Here after entering the Aadhaar ID, the user clicks on the ‘Get OTP’ option which generates and sends the OTP to the registered number.

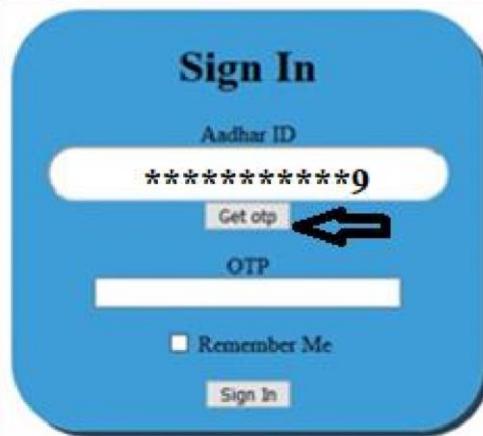


Figure 5 Generate OTP

When the user clicks on get OTP, the engine generates an OTP and sends it to the user's contact number. The OTP is then hashed into some certain hashing format and inserted into the database.

The database doesn't use any engine of its own and hence is only connected to the flask app by SQLITE3 engine. The database file is protected from being crawled, making it almost invincible. Once the OTP has been hashed and inserted in the database, a time count of 2 minutes starts after which the OTP will be automatically deleted from the database. The user gets an OTP through an online API platform (Way2sms) on their phone. Hence, an attacker is restricted to only 2 minutes to perform any SQL injection attack over database to retrieve the hash, but as this application is SQL injection free, it is not possible to attack on the database through it.

4.2 THE POLLING PAGE

Once the user inputs the OTP, it is provided access to the polling page. The moment user logins, the OTP is deleted by engine automatically.

In the Polling Page, the user gets to select their preferred choice of the party he wants to vote. Once the option has been selected, the vote is stored and it redirects the user to the thank you page. The user is restricted to refresh the page for security reasons. The admin section of application is restricted to the server itself. So, it reduces the chances of threats.

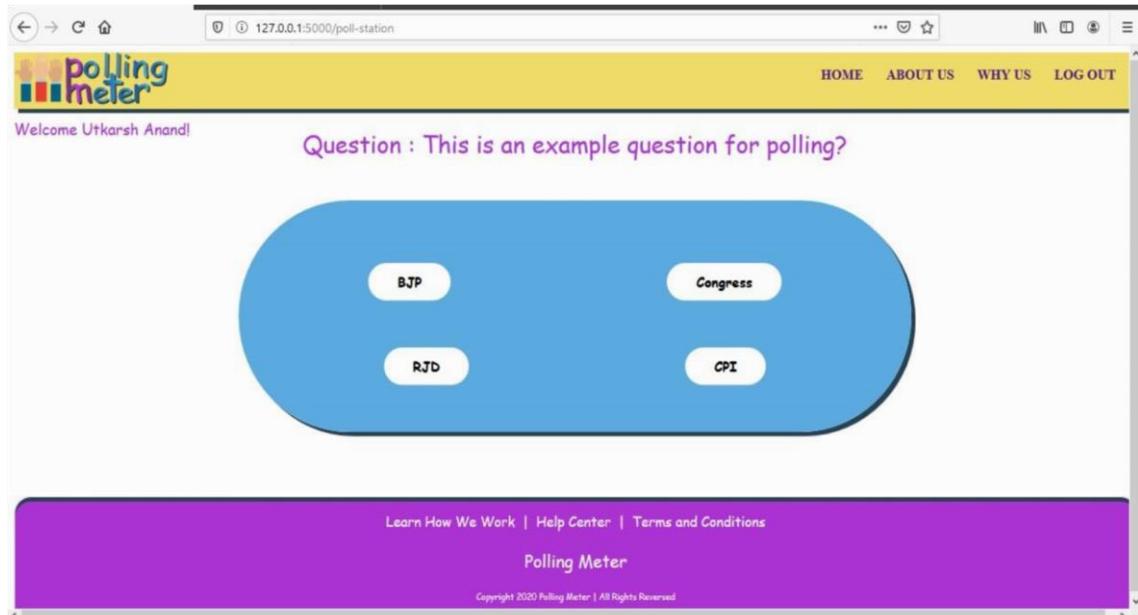


Figure 6 Polling page

After polling, the user cannot visit the polling page again by using the same identity, to save the poll from biasing. The user is redirected to a “Thank page” instead. The users can then logout of the application.

4.2.1 Voting Process

After login process is completed, the user is directed to the Polling Page. This page comes along with welcome note with the user Aadhaar name at the top of the site.

This page contains various options which is flexible for the user to vote. For example, the option present here are: BJP (Bhartiya Janata Party), Congress (Indian National Congress), RJD (Rastriya Janata Dal) and CPI (Communist Party of India). There may be many more options to vote as per the region the user belongs. The user selects out of the choice given. The vote is registered and the candidate is then directed to the Thank You page. The information is stored in encrypted form and there is no XSS attack or unfair means of voting. A person gets to vote only once and the records are stored. This gives a record of people voting and also analyse the data of voters according to the polls.

Question : This is an example question for polling?

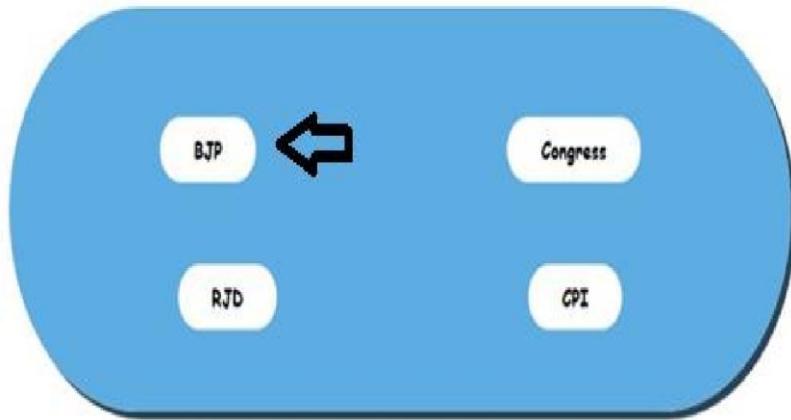


Figure 7 Voting Process

4.2.2 Security Parameters

- The admin section of application is restricted to the server itself. So, no one else can connect to it remotely. This is another security measure taken to dodge from attacks.
- The expertise of this system is being developed to make it available only to the government, as it requires the database of Aadhar card to select the citizens eligibility for voting and use their data to perform the polling.
- The system uses OTPs for the user logins, and will be tested again and again in different conditions and will be assured by our team to be free of any bugs or disorders.
- Hence, we'll make sure that all the manipulations and calculation done by the engine is 100% accurate.

4.3 THANK YOU PAGE

After the voting is done, the user is directed to the ‘THANK YOU’ page. And can not return back to the polling page.

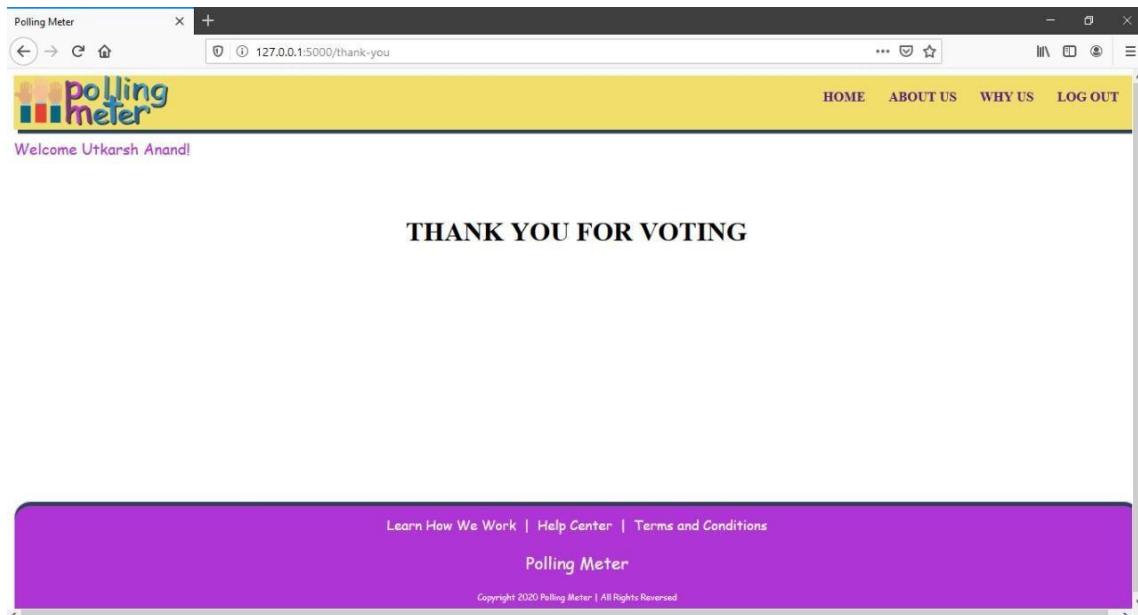


Figure 8Thankyou page

Now, the user can log out.

CHAPTER 5

Design

5.1 Polling through Flow Charts

The analysis of the online voting process can be easily understood by this simple flow chart. A flow chart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task .flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

5.2 System Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

System design is divided into two types:

Logical Design: The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. In the context of systems design, modelling can undertake the following forms, including Data flow diagrams Entity Life Histories, Entity Relationship Diagrams.

Physical Design: The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is inputted into a system, how it is verified/authenticated, how it is processed, and how it is displayed as output.

5.2.1 Use Case

A use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modelling Language (UML) as an actor) and a system to achieve a goal.

A use case is a methodology used in system analysis to identify, clarify and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. The below Use Case Diagram shows a voter, the admin and the judge and their respective roles. The diagram contains:

- Voter's Role
- Admin's Role

➤ Voting Process

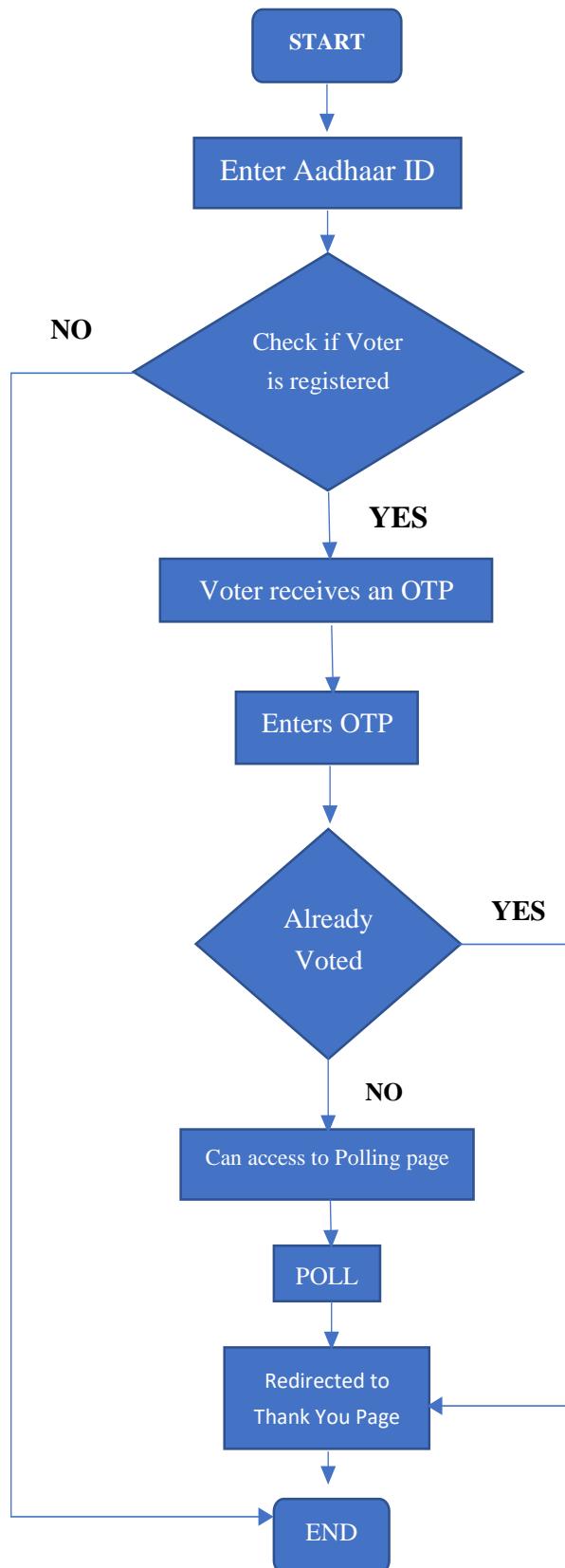


Figure 9 Flowchart

5.2.1.1 Voter's Role

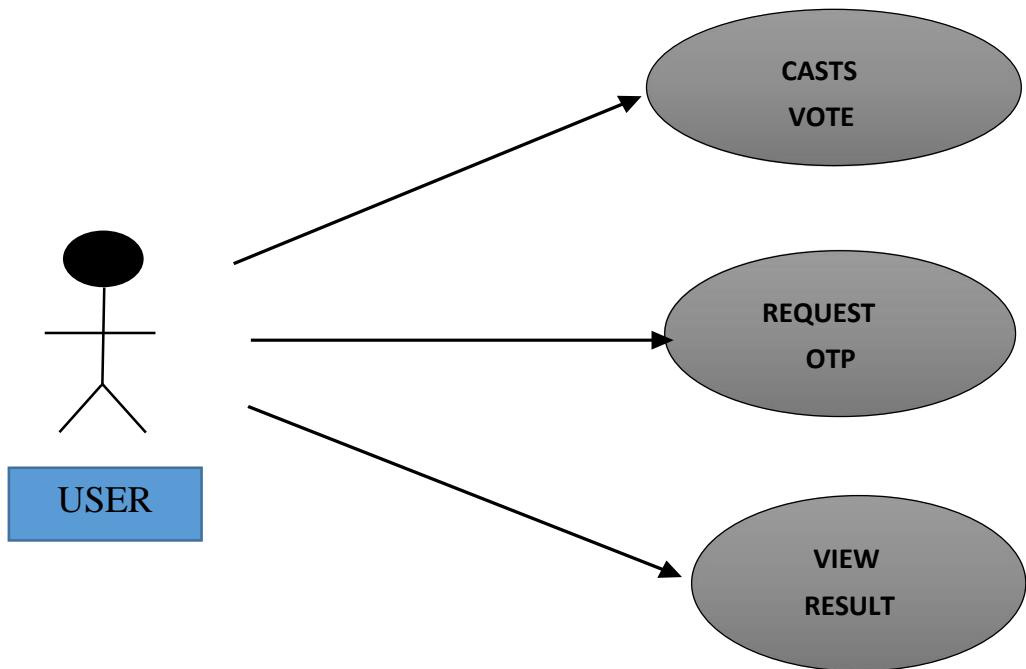


Figure 10Voters Role

5.2.1.2 Admin's Role

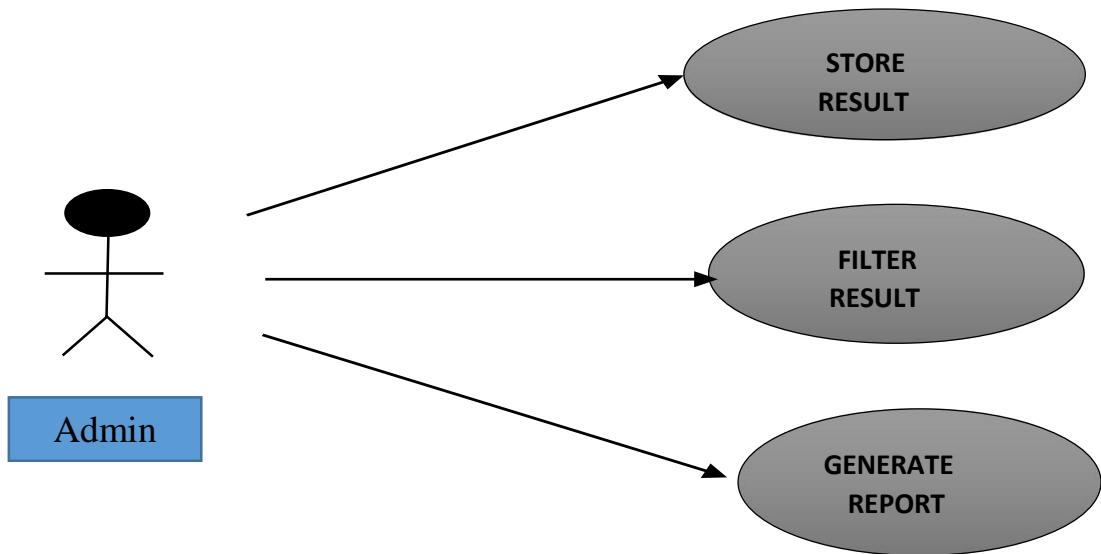


Figure 11Admin Role

5.2.1.3 Voting Process

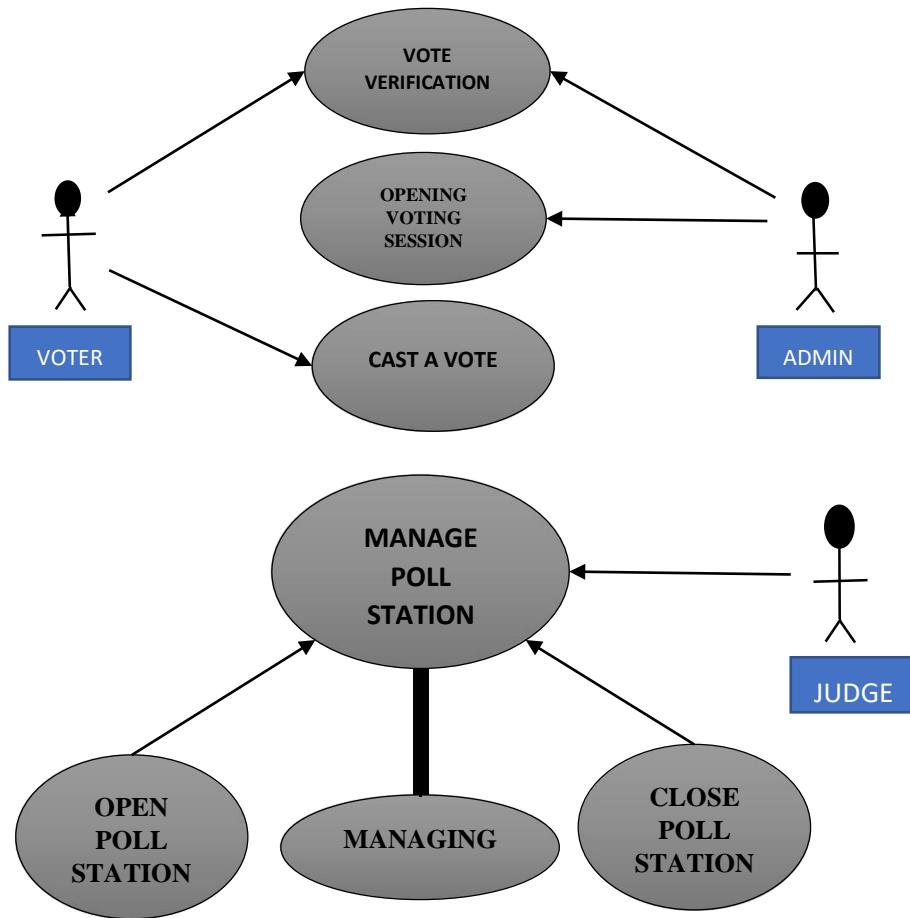


Figure 12 Voting Process

5.2.2 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

5.2.2.1 Sequence diagram of Registration

Like most of the system in this world the security consideration is very important. Since the Aadhaar ID is confidential, we use the feature to avoid the misuse of our Application OTP.

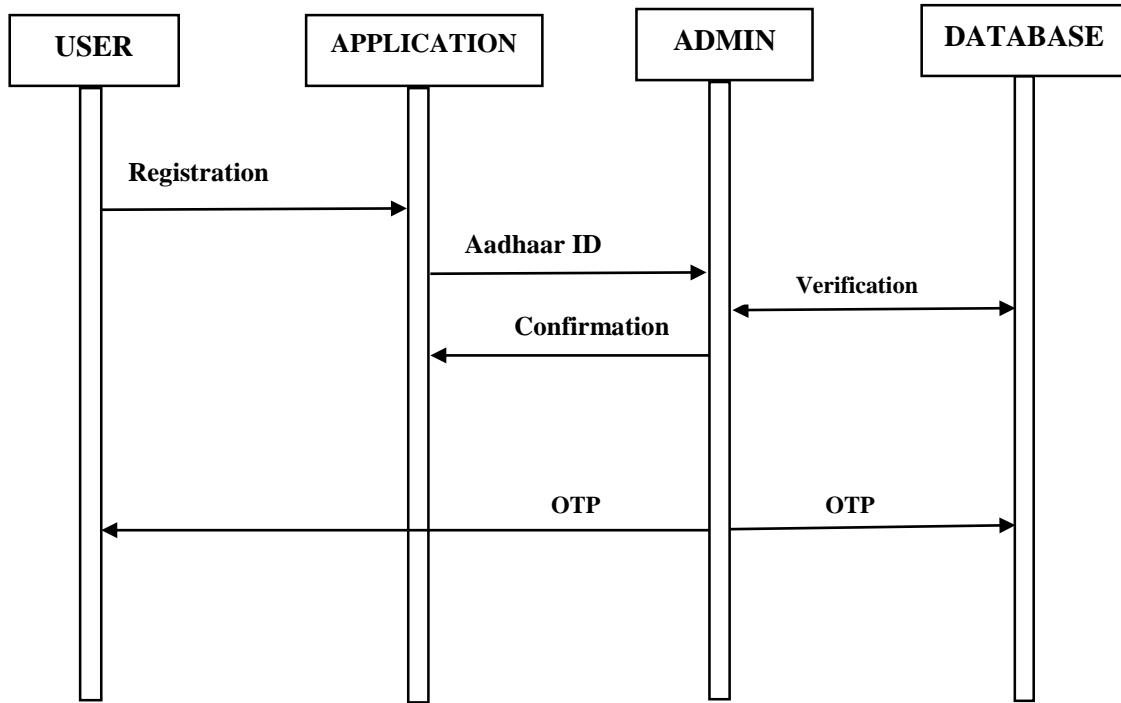


Figure 13 Sequence Diagram of registration

Class Roles: User, Application, Admin and Database

Activation Box

Message: Registration, Aadhaar ID, Confirmation, OTP, Verification and OTP.

5.2.3 Sequence Diagram of Voting Process

Prior to voting as shown, OTP is required.

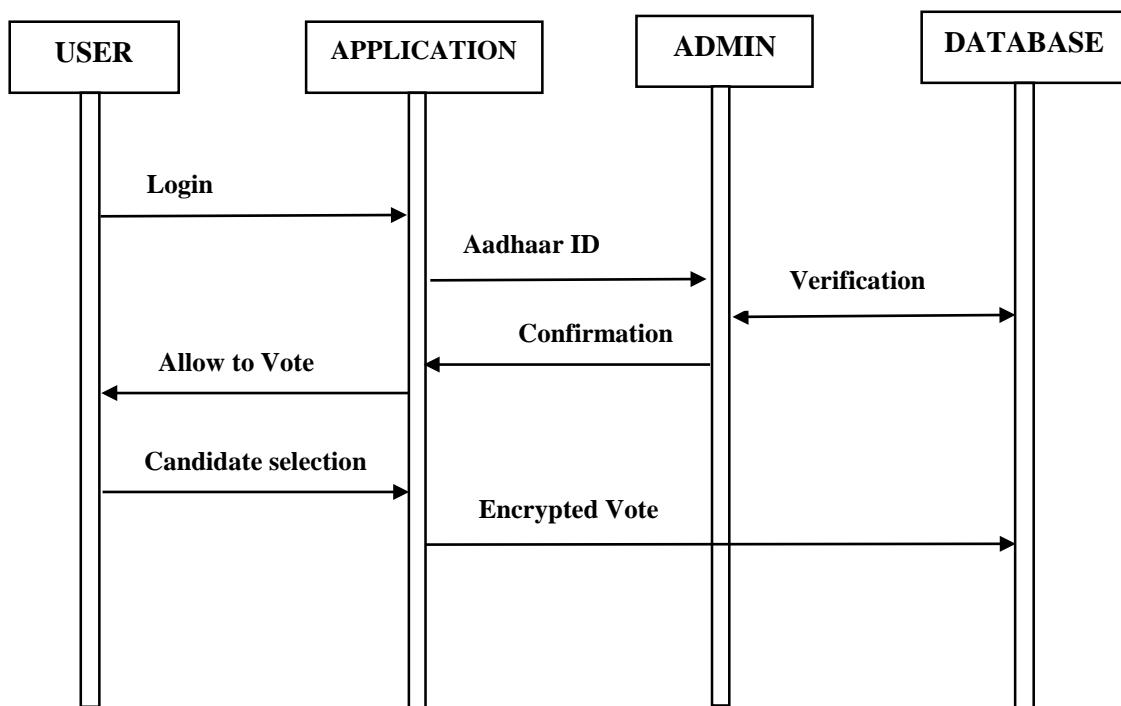


Figure 14 Sequence diagram of voting process

Class Roles: User, Application, Admin and Database

Activation Box

Message: Login, allow to vote, Candidate Selection, Aadhaar ID, Confirmation, Encrypted Vote and Verification.

5.2.4 ER Diagram

ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram.

ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship

The ER diagram of the Voting process is shown below.

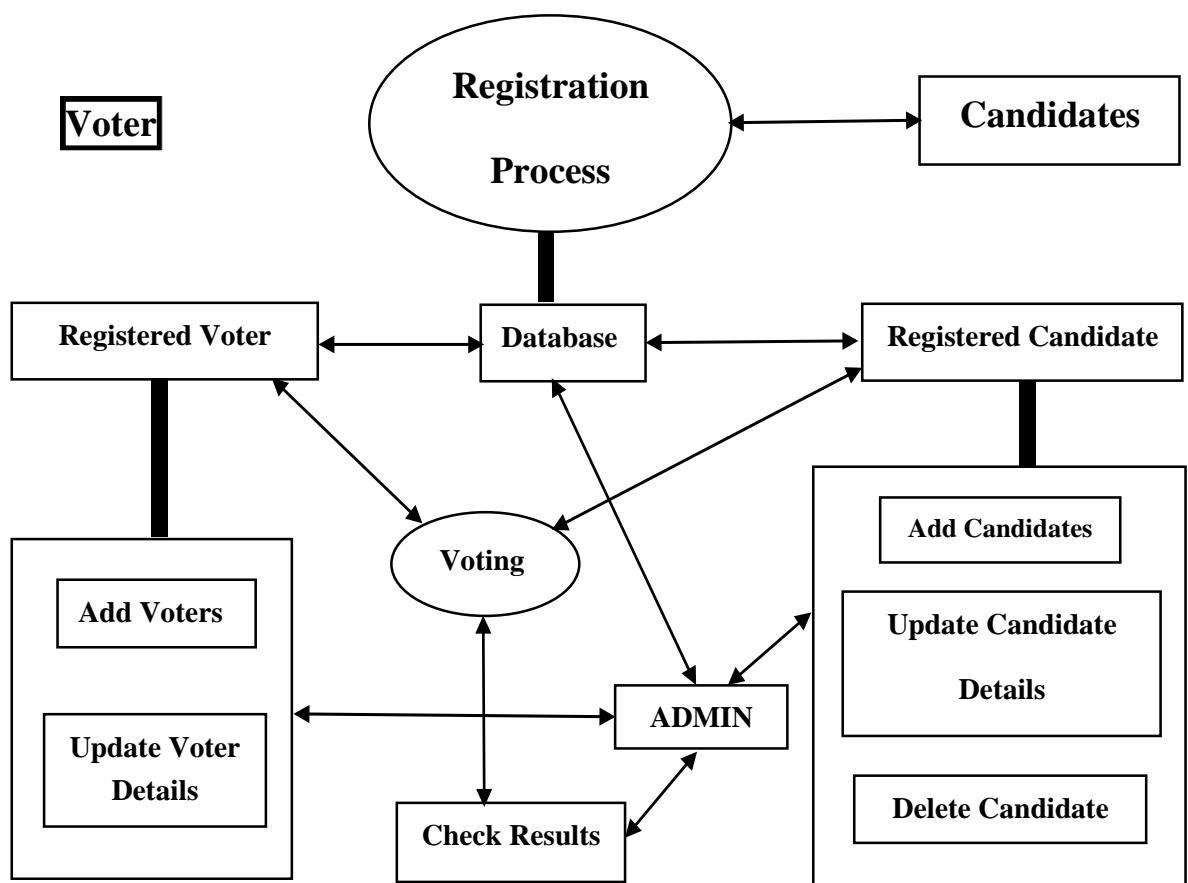


Figure 15ER Model

CHAPTER 6

CODING

6.1 Frontend

The front end of a website is the part that users interact with. Everything that you see when you're navigating around the Internet, from fonts and colours to dropdown menus and sliders, is a combo of HTML, CSS, and JavaScript being controlled by your computer's browser.'

Here HTML, CSS and JavaScript is being used to make the website dynamic and user-friendly. To ease the Voting process, the site has been kept simple so that every user is easily able to process with the system.



The following are the codes for the designing of the web application.frontend:

- Login Page
- Polling Page
- Thank You Page

6.1.1 Login Page



The screenshot shows a code editor window with the title bar "File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X". The toolbar below has various icons for file operations like Open, Save, Print, and others. The main area displays the "login.html" file content:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8      <h1>Sign In</h1>
9      <form action="" method="post" novalidate>
10         {{ form.hidden_tag() }}
11         <p>
12             {{ form.aadharid.label }}<br>
13             {{ form.aadharid(size=32) }}
14             {% for error in form.aadharid.errors %}
15                 <span style="color: red;">{{ error }}</span>
16             {% endfor %}
17         </p>
18         <p>
19             {{ form.otp.label }}<br>
20             {{ form.otp(size=32) }}
21             {% for error in form.otp.errors %}
22                 <span style="color: red;">{{ error }}</span>
23             {% endfor %}
24         </p>
25         <p> {% with messages = get_flashed_messages() %}
26             {% if messages %}
27                 {% for message in messages %}
28                     <span style = "color : red;">{{message}}</span>
29                 {% endfor %}
30                 {% endif %}
31                 {% endwith %}
32         </p>
33         <p>{{ form.remember_me() }} {{ form.remember_me.label }}</p>
34         <p>{{ form.login() }}</p>
35     </form>
36  </body>
37 </html>
```

Figure 16 Login page coding

6.1.2 Polling Page



The screenshot shows a code editor window with the title bar "polling page.html". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, ?, and X. The toolbar contains various icons for file operations like Open, Save, Print, and Find. The code editor displays the following HTML code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Polling Meter Login</title>
6      <link rel="stylesheet" type="text/css" href="/static/d.css">
7      <style = "stylesheet">
8          a{text-decoration:none;}
9      </style>
10     </head>
11     <body width="100%">
12         <div class="one">
13             <div style="position:absolute" height="65" rightmargin = "594">
14                 
15             </div>
16
17             <p ALIGN="RIGHT">
18                 <b><a href="/">HOME</a>
19                     &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
20                     <a href="/about">ABOUT US</a>
21                     &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
22                     <a href="/why-us">WHY US</a></b>
23             </div>
24             <div class="two">
25                 <div class="log">
26                     <center>
27                         <h1>Sign In</h1>
28                         <form action="" method="post" novalidate>
29                             {{ form.hidden_tag() }}
30                             <p>
31                                 {{ form.aadharid.label }}<br>
32                                 {{ form.aadharid(size=32, class='login') }}
33                                 <button type="button" onclick="window.location+= 'login/' +
34                                     document.getElementById('aadharid').value">Get otp</button>
35                                 {% for error in form.aadharid.errors %}
36                                     <span style="color: red;">{{ error }}</span>
37                                 {% endfor %}
```

Figure 17 Polling page coding

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
polling page.html
35      |  {% for error in form.aadhardid.errors %}|
36      |  <span style="color: red;">{{ error }}</span>
37      |  {% endfor %}
38      |  </p>
39      |  <p>
40      |  {{ form.otp.label }}<br>
41      |  {{ form.otp(size=32) }}
42      |  {% for error in form.otp.errors %}|
43      |  <span style="color: red;">{{ error }}</span>
44      |  {% endfor %}
45      |  </p>
46      |  <p> {{ with messages = get_flashed_messages() }}|
47      |  {% if messages %}|
48      |  {% for message in messages %}|
49      |  <span style = "color : red;">{{message}}</span>
50      |  {% endfor %}
51      |  {% endif %}
52      |  {% endwith %}
53      |  </p>
54      |  <p>{{ form.remember_me() }} {{ form.remember_me.label }}</p>
55      |  <p>{{ form.login() }}</p>
56      |  </form>
57
58      |  </center>
59      |  </div>
60      |  </div>
61  <div class="three">
62    <center>
63      <p ALIGN="center">
64        <a style="color:white" href="/learn">Learn How We Work</a>
65        &nbsp|&nbsp;
66        <a style="color:white" href="/help">Help Center</a>
67        &nbsp|&nbsp;
68        <a style="color:white" href="/policies">Terms and Conditions</a>
69      </p>
70      <p style="font-size:20">Polling Meter</p>
71      <p style="font-size:10">Copyright 2018 Polling Meter | All Rights Reserved
72      </p>
73    </center>
74  </div>
75  </body>
76  </html>
```

6.1.3 Thank You Page

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? X
[File Icons] [thank you.html]
poling page.html thank you.html

1 <html>
2 <head>
3   <link rel="stylesheet" type="text/css" href="/static/d.css">
4   <title>Polling Meter</title>
5   <style = "stylesheet">
6     a{text-decoration:none;}
7   </style>
8 </head>
9
10 <body width="100%">
11   <div class="one">
12     <div style="position:absolute" height="65" rightmargin = "594">
13       
14     </div>
15
16     <p ALIGN="RIGHT">
17       <b><a href="/">HOME</a>
18         &nbsp; &nbsp; &nbsp;
19       <a href="/about">ABOUT US</a>
20         &nbsp; &nbsp; &nbsp;
21         <a href="/why-us">WHY US</a></b>
22         &nbsp; &nbsp; &nbsp;
23       <b><a href="/logout">LOG OUT</a></b>
24         &nbsp; &nbsp; &nbsp;
25     </p>
26   </div>
27   <div class="two">
28     <div style="position:absolute">
29       <p style="font-size:18; color:#AF34D5; font-family: 'comic sans MS'" >
30         Welcome {{current_user.Name}}!
31       </p>
32     </div>
33     <center>
34       &nbsp;
35       <br/>
```

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
X
polling page.html thank you.html
31     </div>
32     <center>
33         &nbsp;
34         <br/>
35         &nbsp;
36         <br/>
37         &nbsp;
38         <br/>
39         &nbsp;
40         <br/>
41         &nbsp;
42         <br/>
43         <H1 style="size=100px">THANK YOU FOR VOTING</H1>
44     </center>
45 </div>
46 </div>
47 <div class="three">
48     <center>
49         <p ALIGN="center">
50             <a style="color:white" href="/learn">Learn How We Work</a>
51             &nbsp;|&nbsp;
52             <a style="color:white" href="/help">Help Center</a>
53             &nbsp;|&nbsp;
54             <a style="color:white" href="/policies">Terms and Conditions</a>
55         </p>
56         <p style="font-size:20">Polling Meter</p>
57         <p style="font-size:10">Copyright 2018 Polling Meter | All Rights Reserved
58     </center>
59 </div>
60 </body>
61 </html>
```

Figure 18 Thank you page coding

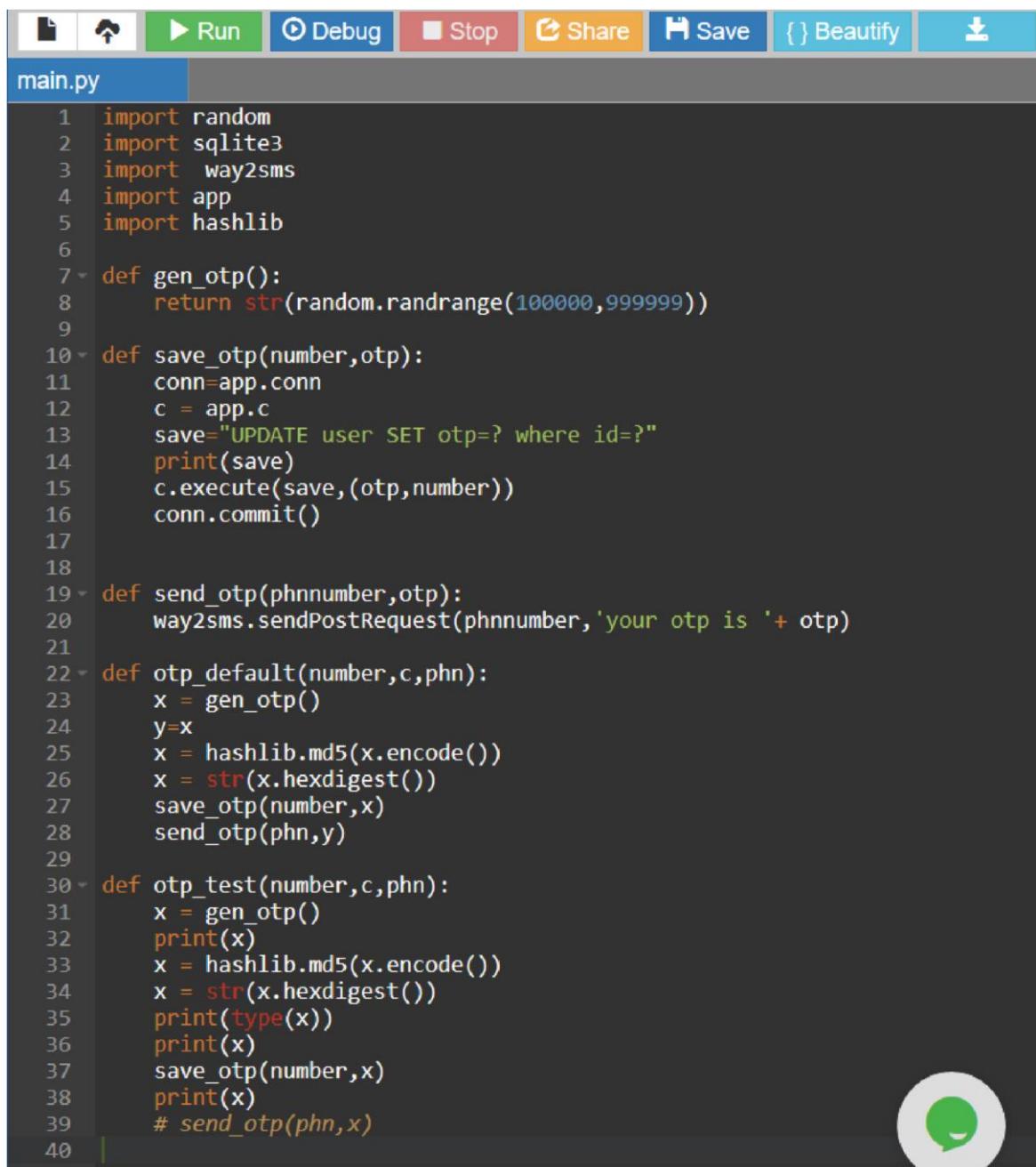
6.2 Backend

So, what makes the front end of a website possible? Where is all that data stored? This is where the back end comes in. The back end of a website consists of a server, an application, and a database. A back-end developer builds and maintains the technology that powers those components which, together, enable the user-facing side of the website to even exist in the first place. Here Flask is been. It is a micro web framework written in Python.

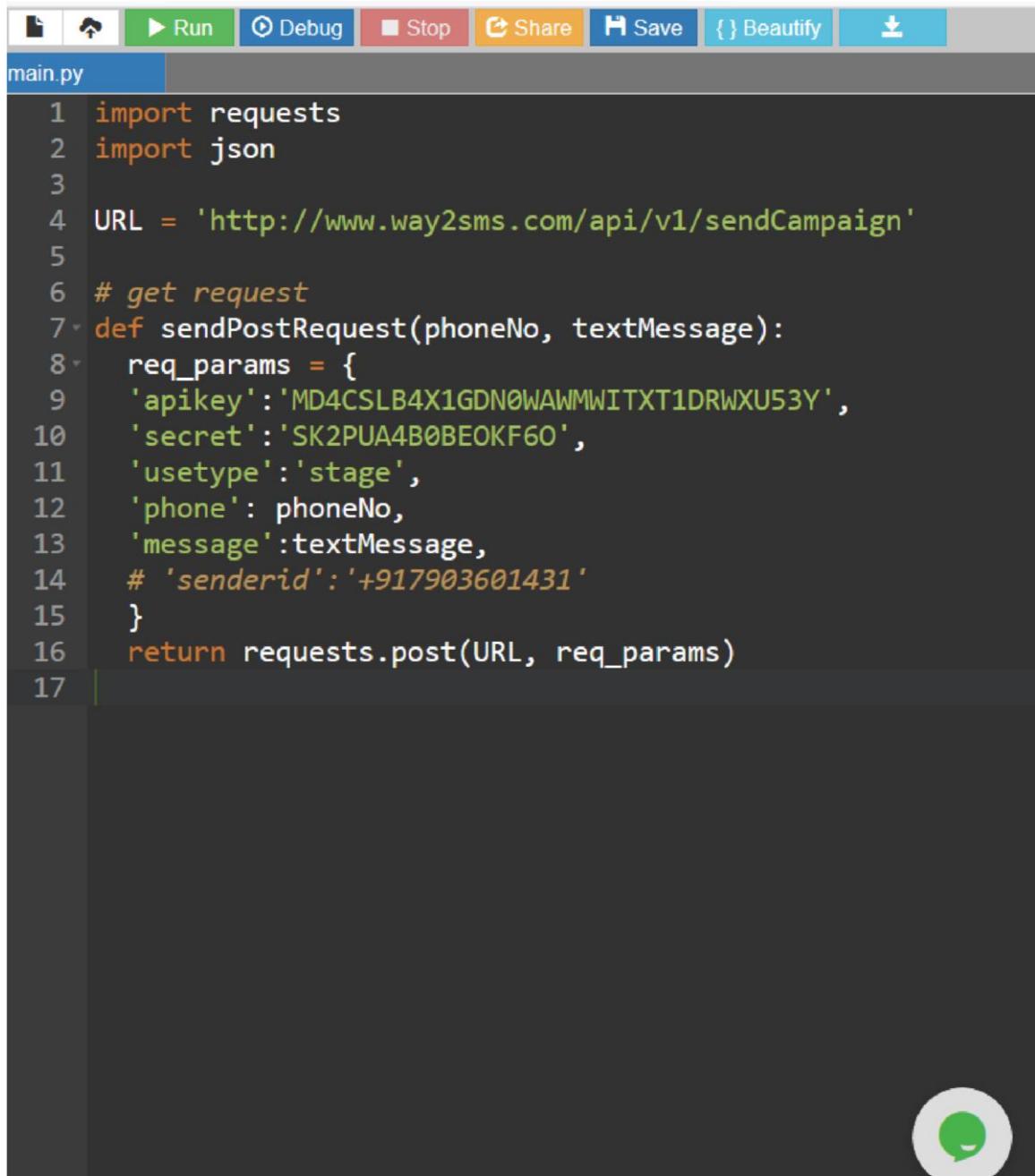
Some common backend languages are Ruby, PHP, Java. Net, and Python. These programming languages often run on frameworks that simplify the web development process. Rails, for example, is a framework written in Ruby. for its approachable syntax and widespread server-side use makes Python a core programming language for back-end development.backend:

- OTP Code Page
- Way2sms Code Page
- Database Code Page
- Form Code Page

6.2.1 Some important codes



```
main.py
1 import random
2 import sqlite3
3 import way2sms
4 import app
5 import hashlib
6
7 def gen_otp():
8     return str(random.randrange(100000,999999))
9
10 def save_otp(number,otp):
11     conn=app.conn
12     c = app.c
13     save="UPDATE user SET otp=? where id=?"
14     print(save)
15     c.execute(save,(otp,number))
16     conn.commit()
17
18
19 def send_otp(phnnumber,otp):
20     way2sms.sendPostRequest(phnnumber,'your otp is '+otp)
21
22 def otp_default(number,c,phn):
23     x = gen_otp()
24     y=x
25     x = hashlib.md5(x.encode())
26     x = str(x.hexdigest())
27     save_otp(number,x)
28     send_otp(phn,y)
29
30 def otp_test(number,c,phn):
31     x = gen_otp()
32     print(x)
33     x = hashlib.md5(x.encode())
34     x = str(x.hexdigest())
35     print(type(x))
36     print(x)
37     save_otp(number,x)
38     print(x)
39     # send_otp(phn,x)
40
```



The screenshot shows a Python code editor interface with the following details:

- Toolbar:** Includes icons for Run, Debug, Stop, Share, Save, and Beautify.
- File:** The file is named "main.py".
- Code Content:** The code is as follows:

```
1 import requests
2 import json
3
4 URL = 'http://www.way2sms.com/api/v1/sendCampaign'
5
6 # get request
7 def sendPostRequest(phoneNo, textMessage):
8     req_params = {
9         'apikey': 'MD4CSLB4X1GDN0WAWMWITXT1DRWXU53Y',
10        'secret': 'SK2PUA4B0BEOKF60',
11        'usetype': 'stage',
12        'phone': phoneNo,
13        'message': textMessage,
14        # 'senderid': '+917903601431'
15    }
16    return requests.post(URL, req_params)
17
```

Figure 19Main.py

The screenshot shows a Python IDE interface with a dark theme. The top bar includes standard icons for file operations (New, Open, Save) and tool buttons for Run, Debug, Stop, Share, Save, and Beautify. The main window displays a file named 'main.py' containing the following Python code:

```
1 import sqlite3
2 from config import *
3 from flask_login import current_user
4 import time
5
6 db="poll_database.db"
7
8 def connect():
9     conn=sqlite3.connect(db,check_same_thread=False)
10    return conn
11
12 def create_poll_table(var,c):
13     code = "CREATE TABLE IF NOT EXISTS "+var+
14     "(name VARCHAR(50) NOT NULL, count INTEGER NOT NULL DEFAULT 0)"
15     #print(code)
16     c.execute(code)
17
18 def create_option(c,option,var):
19     code="INSERT INTO "+var+"(name) VALUES (?)"
20     #print(code)
21     c.execute(code,(option))
22
23 def update_vote(conn,c,option,var):
24     conn=sqlite3.connect("poll_database.db")
25     c=conn.cursor()
26     code = "UPDATE "+var+" SET count=count+1 WHERE name=?"
27     print(code)
28     c.execute(code,(option,))
29     save = "UPDATE user SET voted=1 WHERE id = ?"
30     print(current_user.id)
31     c.execute(save,(current_user.id,))
32     conn.commit()
33
34 def clear_otp(conn, c, table, x):
35     conn=sqlite3.connect("poll_database.db")
36     c.execute("DELETE FROM "+table+" WHERE otp=?")
```



```
main.py
38     c.execute(code, (X,))
39     conn.commit()
40
41 def func1(conn,c,value):
42     time.sleep(120)
43     clear_otp(conn, c, "user", value)
44
45
46 def find_user(uid):
47     c=connect().cursor()
48
49     c.execute("SELECT otp FROM user WHERE id=(?)",(uid,))
50     a=c.fetchone()
51     if (a==None):
52         return 0
53     else:
54         return a[0]
55
56 def user_login(c,uid):
57     c.execute("SELECT * FROM user WHERE id=(?)",(uid,))
58     return c.fetchone()
59
60 def main():
61     db="poll_database.db"
62     conn=connect(db)
63     c=conn.cursor()
64
65     var=input("ENTER POLL NAME : ")
66     create_poll_table(var,c)
67     option=input("ENTER OPTIONS USING COMMAS : ")
68     option=option.split(",")
69     for i in range(0,len(option)):
70         #print(option[i])
71         create_option(c,option[i],var)
72     print("Connection Established")
73     conn.commit()
```

The screenshot shows a code editor window with a dark theme. At the top, there is a toolbar with icons for Run, Debug, Stop, Share, Save, and Beautify. Below the toolbar, the file tab 'main.py' is selected. The code in the editor is:

```
1 from flask_wtf import FlaskForm
2 from wtforms import StringField, PasswordField, BooleanField, SubmitField
3 from wtforms.validators import DataRequired
4
5 class LoginForm(FlaskForm):
6     aadharid = StringField('Aadhar ID', validators=[DataRequired()])
7     otp = PasswordField('OTP', validators=[DataRequired()])
8     remember_me = BooleanField('Remember Me')
9     login = SubmitField('Sign In')
10
```

CHAPTER 7

TESTING

6.1 INTRODUCTION

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user's dissatisfaction and downfall of reputation.

"Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors." The first step in System testing is to develop the plan that all aspect of system .Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the program fail.

6.2 Types of Testing

➤ Black Box (Functional) Testing:

Testing against specification of system or components. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes, e.g. positive numbers vs. negative numbers. Programs normally behave the same way

for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested. Internal System design is not considered in this type of testing. Tests are based on requirements and functionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external database access.
- Performance errors.
- Initialization and termination errors.

➤ **White Box (Structural) Testing:**

Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test cases need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has no control statements, there will be up to 2^n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow

graph derived testing:

- Data complexity could not take into account.
- We cannot test all paths in combination.
- It is really only possible at unit and module testing stages because beyond that complexity is too high.

This testing is based on knowledge of the internal logic of an application's code. Also known as a Glass Box Testing .Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

➤ **Unit Testing:**

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

➤ **Incremental Integration Testing:**

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

➤ **Integration Testing:**

Testing of integration modules to verify combined functionality after integration .Modules are typically code modules, individual applications, client and server and distributed systems.

➤ **Functional Testing:**

This type of testing ignores the internal parts and focus on the output is as per requirement or not .Black box type testing geared to functionality requirements of an application.

➤ **System Testing:**

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

This Online Voting system will manage the Voter's information by which voter can login and use his voting rights. The system will incorporate all features of voting system. It provides the tools for maintaining voter's vote to every party and it count total no. of votes of every party. There is a database which is maintained by the Election Commission of India in which all the names of voter with complete information is stored.

The expertise of this system is being developed to make it available only to the Government, as it requires the database of Aadhar card to select the citizens eligibility for voting and use their data to perform the polling. The system uses OTPs for the user logins, and will be tested again and again in different conditions and will be assured by our team to be free of any bugs or disorders. Hence, we'll make sure that all the manipulations and calculation done by the engine is 100% accurate.

8.2 FUTURE SCOPE

The Online Voting System platform can be made more secure by using the following methods

- Fingerprinting
- Cornea Detection

The OTP used by the user to vote is provided by the administrator. In the future the user can be given the privilege of security of personal data. So, it helps to increase the security of the system.

The other methods that can be used are cornea detection. But here the problem is that it decreases the scope of the platform because these systems need some electronic components to implement. So, it will avoid the user's privilege to cast the votes at their fingertips. But it can guarantee that fake voting will be impossible.

CHAPTER 9

REFERENCES

1. Batalas, N., Khan, V. J., & Markopoulos, P. (2021). ExecutableHTML. *SoftwareX*, 14, 100691.
2. Van Rossum, G. (2003). *An introduction to Python* (p. 115). F. L. Drake (Ed.). Bristol: Network Theory Ltd..
3. Herman, T. (2019). *A Functional Start to Computing with Python*. Chapman and Hall/CRC.
4. Lawson, B., & Sharp, R. (2011). *Introducing html5*. New Riders.
5. Fajfar, I. (2019). *Start Programming Using HTML, CSS, and JavaScript*. Chapman and Hall/CRC.
6. Gans, M., Hodges, T., & Wilson, G. (2020). *JavaScript for Data Science*. Chapman and Hall/CRC.
7. Nemetz, S., Schmitt, S., & Freiling, F. (2018). A standardized corpus for SQLite database forensics. *Digital Investigation*, 24, S121-S130.
8. Yamamoto, M., & Koizumi, H. (2013). An experimental evaluation using sqlite for real-time stream processing. *Journal of International Council on Electrical Engineering*, 3(1), 68-72.
9. Al-Sarayreh, K. T., Meridji, K., & Abran, A. (2020). Software engineering principles: A systematic mapping study and a quantitative literature review. *Engineering Science and Technology, an International Journal*.

10. Garousi, V., Rainer, A., Lauvås Jr, P., & Arcuri, A. (2020). Software-testing education: A systematic literature mapping. *Journal of Systems and Software*, 165, 110570.
11. Baugh, J., & Liu, S. (2016, May). A general characterization of the Hardy Cross method as sequential and multiprocess algorithms. In *Structures* (Vol. 6, pp. 170-181). Elsevier.
12. Reisinger, E., Genthner, L., Kerssemakers, J., Kensche, P., Borufka, S., Jugold, A., ... & Eils, J. (2017). OTP: An automatized system for managing and processing NGS data. *Journal of biotechnology*, 261, 53-62.
13. Gilreath, C. L. (2019). *Computerized literature searching: Research strategies and databases*. Routledge.
14. Oliveira, A., Kohwalter, T., Kalinowski, M., Murta, L., & Braganholo, V. (2020). XChange: a semantic diff approach for XML documents. *Information Systems*, 94, 101610.
15. Pawlak, M., Poniszewska-Marańda, A., & Kryvinska, N. (2018). Towards the intelligent agents for blockchain e-voting system. *Procedia Computer Science*, 141, 239-246.