e-vote: digital democracy

A Project Report for Industrial Internship Submitted by:

Ritesh Ghosh

Sreya Das

Sampriti Chowdhury

Biplab Pal

Suman Bhattacharya

In the partial fulfillment for the award of the degree of

# BTech

In the

**Department of Computer Science and Engineering**

##### Of

**Heritage Institute of Technology, Kolkata**



at

**Ardent Computech Pvt. Ltd.**

**Module**



# CERTIFICATE FROM SUPERVISOR

This is to certify that **“Ritesh Ghosh, Sreya Das, Biplab Pal, Suman Bhattacharya and Sampriti Chowdhury** of **2021-22”** have successfully completed the project titled "eVote: Digital Democracy" under my supervision during the period from “14-06-2024” to “14-07-2024” which is in partial fulfillment of requirements for the award of the **B-Tech** degree and submitted to the Department of “**Computer Science and Engineering**” of **“Heritage Institute of Technology, Kolkata”**.



**Signature of the Supervisor**

**Starting date:** 14/06/2024

**Ending date :** 14/07/2024

**Name of the Project Supervisor:**

# ACKNOWLEDGEMENT

The achievement that is associated with the successful completion of any task would be incomplete without mentioning the names of those people whose endless cooperation made it possible. Their constant guidance and encouragement made all our efforts successful.

We take this opportunity to express our deep gratitude towards our project mentor/external guide **[Mr. Subhojit Santra]**and internal guide **[Mr. Srimanta Santra]**for giving such valuable suggestions, guidance and encouragement during the development of this project work.

Last but not the least we are grateful to all the faculty members of **Ardent Computech Pvt.Ltd.**fortheirsupport.

# DECLARATION

This is to certify that the dissertation / project proposal entitled “**eVote: Digital Democracy**” is done by us, is an authentic work carried out for the partial fulfillment of the requirements for the award of the certificate of Bachelor of Technology under the guidance of Mr. Subhajit Santra. The matter embodied in this project work has not been submitted earlier for award of any certificate to the best of our knowledge and belief.

**Name of the Students:**

* Ritesh Ghosh
* Sreya Das
* Biplab Pal
* Sampriti Chowdhury
* Suman Bhattacharya

**Signature of students:**

**I.**

**II. III. IV.**

**V.**

# APPROVAL SHEET

This project report entitled **‘eVote: Digital Democracy’** by Ritesh Ghosh, Sreya Das, Biplab Pal, Sampriti Chowdhury, Suman Bhattacharya is approved for the degree of Bachelor of Technology in Computer Science and Engineering.

(Signature of Examiners)

Signature of Internal Guide

Signature of Principle

(Basab Chaudhuri)

Signature of Industrial Guide (Mr. Subhojit Santra)

Date: Place:

# CERTIFICATE BY GUIDE

This is to certify that this project entitled “**eVote: Digital Democracy** “submitted in partial fulfillment of the certificate of Bachelor of Technology in Computer Science and Engineering through Ardent Computech Pvt. Ltd., done by the

Group Members:

* Ritesh Ghosh
* Sreya Das
* Biplab Pal
* Sampriti Chowdhury
* Suman Bhattacharya

is an authentic work carried out under my guidance & best of our knowledge and belief.

**Signature of the students Signature of the Guide**

Date: Date:

# CERTIFICATE OF APPROVAL

This is to certify that this proposal of Minor project, entitled **“eVote: Digital Democracy”** is a record of bona-fide work, carried out by: **1. Ritesh Ghosh - 12621001098, 2. Sreya Das - 12621001144, 3. Biplab Pal - 12621001045, 4. Suman Bhattacharya - 12621001149, 5. Sampriti Chowdhury - 12621001108** under my super vision and guidance through the Ardent Computech Pvt. Ltd. In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the Heritage Institute of Technology as per regulations of the **Ardent®**. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for Bachelor of Technology.

Guide/Supervisor

##### Mr. Subhojit Santra

Subject Matter Expert & Technical Head (PHP) Ardent Computech Pvt Ltd(AnISO9001:2008Certified)

Module#132, Ground Floor, SDF Building, GP Block, Sector V, Bidhannagar, Kolkata,West Bengal 700091

**External Examiner(s) Head of the Department**

(Department of Computer Science and Engineering) Heritage Institute of Technology, Kolkata

# PROJECT RESPONSIBILITY FORM

|  |  |  |
| --- | --- | --- |
| **SL.NO.** | **MEMBER OF PROJECT** | **RESPONSIBILITY** |
| **1**. | RITESH GHOSH | BACKEND CODING |
| **2.** | SREYA DAS | WEBPAGE DESIGNING & DOCUMENTATION |
| **3.** | SUMAN BHATTACHARYA | POWERPOINT PRESENTATION |
| **4.** | BIPLAB PAL | POWERPOINT PRESENTATION |
| **5.** | SAMPRITI CHOWDHURY | DOCUMENTATION & DATA COLLECTION |



|  |  |  |
| --- | --- | --- |
|  | TABLE OF CONTENTS |  |
| **Sl. No.** | **Name of the Topic** | **PageNo.** |
| 1. | Company Profile | 14 |
| 2. | Introduction   * Objective * Scope | 15  16  17 |
| 3. | System analysis   * Identification of need * Feasibility study * Workflow * Study of the system * Input and Output * Software requirement specification * Software engineering paradigm applied | 18  19  20  21-25  26  27  28-29  30-32 |
| 4. | System Design   * Data-Flow Diagram * Entity Relationship Diagram | 33  34-39  41-44 |
| 5. | Output Screen   * User-interface Design * Snapshots | 54  55  56-64 |
| 6 | Implementation and Testing   * Introduction * Objectives of testing * Test Cases * White-Box Testing * Black-Box Testing * Output Testing * Goal of Testing * Integration Test Report | 65  65  66-67  67  68  68  70  72  75 |

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Name of the Topic** | Page No. |
| 7. | Gantt Chart   * Project schedule * Gantt Chart | 76  77  78 |
| 8. | System Security Measures   * Database security measures * System security measures * Limitations | 79  80  80  81 |
| 9. | Conclusion | 82 |
| 9 | Future Scope & Further Enhancements | 83 |
| 10 | Bibliography/References | 84 |

# ARDENT COMPUTECH PVT. LTD.

Ardent Computech Private Limited is an ISO 9001-2008 certified Software Development Company in India. It has been operating independently since 2003. It was recently merged with ARDENT TECHNOLOGIES.

Ardent Technologies

ARDENT TECHNOLOGIES is a Company successfully providing its services currently in UK, USA, Canada and India. The core line of activity at ARDENT TECHNOLOGIES is to develop customized application software covering the entire responsibility of performing the initial system study, design, development, implementation and training. It also deals with consultancy services and Electronic Security systems. Its primary clientele includes educational institutes, entertainment industries, resorts, theme parks, service industry, telecom operators, media and other business houses working in various capacities.

Ardent Collaborations

ARDENT COLLABORATIONS, the Research Training and Development Department of ARDENT COMPUTECH PVT LTD is a professional training Company offering IT enabled services & industrial trainings for B-Tech, MCA, BCA, MSc and MBA fresher’s and experienced developers/programmers in various platforms. Summer Training / Winter Training / Industrial training will be provided for the students of B.TECH, M.TECH, MBA and MCA only. Deserving candidates may be awarded stipends, scholarships and other benefits, depending on their performance and recommendations of the mentors.

Associations

ArdentisanISO9001:2008company.

It is affiliated to National Council of Vocational Training (NCVT), Directorate General of Employment & Training (DGET), Ministry of Labor & Employment, and Government of India.

## ABSTRACT

The word ‘vote’ means to choose from a list, to elect or to determine. The main goal of voting

(in a scenario involving citizens of a given country) is to come up with leaders of the people’s

choice.

With rapid growth in technologies the old voting methods can change to advanced voting

methods. Online voting software is a modern solution that can efficiently and securely facilitate

the voting process for various groups and organizations. The use of such software eliminates

the need for physical polling stations, as voters can cast their ballots from anywhere with

an internet connection. The benefits of using online voting software are many; it increases

accessibility, saves time and resources, ensures accuracy and transparency, and supports a

more democratic decision-making process. Eligibility verification and accurate voter

information are essential components of a successful online voting platform. While several

countries have already implemented online voting software, this approach still faces challenges

and limitations that must be addressed before universal adoption.

Key features within the scope include user-friendly interfaces for voters, secure mechanisms

for vote submission, and automated vote counting and result generation. Additionally, the system

provides administrative tools for election management, such as configuring election

parameters, monitoring voter activity, and generating reports. By integrating advanced security

protocols, transparency measures, and accessibility options, the system aims to streamline the

electoral process, reduce operational costs, and enhance voter engagement. This broad scope

ensures that the system can accommodate both small-scale and large-scale elections while

maintaining the highest standards of efficiency and integrity.

## INTRODUCTION

An online election system is a web-based digital platform designed to facilitate the process of voting and managing elections electronically over the internet. It aims to provide a secure, efficient, and transparent method for conducting elections, ensuring that every eligible voter can participate easily and their votes are accurately counted. It allows voters to register, authenticate, cast their ballots, and view results securely and efficiently from any location with internet access.

This system enhances accessibility for all voters, streamlines the election process, and provides robust security measures to ensure the integrity and confidentiality of votes. By reducing the need for physical polling stations and paperwork, it also offers cost savings and environmental benefits, thereby increasing voter turnout, especially among those who may find it difficult to vote in person. While there are concerns regarding security and privacy, the benefits of an online voting system cannot be denied. In this context, the purpose and scope of the system are to ensure that every citizen can participate in the democratic process in a secure and hassle-free manner.

## OBJECTIVE

Online election systems aim to enhance the electoral process by providing a secure, transparent, and accessible platform for voting. By enabling voters to cast their ballots from anywhere with an internet connection, these systems increase participation and streamline election management. Also by maintaining robust security protocols, the system aims to build trust among voters and safeguard the validity of the election outcomes. The modernization from traditional paper-based voting to an online platform, seeks to address several key areas:

**1. To Create a Secure and Transparent Voting System:** Implement strong security measures and transparency features to ensure election integrity and build voter trust.

**2. To Increase Voter Participation by Making Voting More Accessible:** Enable online voting to remove barriers, making it easier for more people to participate in elections.

**3. To Streamline the Election Process and Reduce Associated Costs:** Automate and simplify election procedures to lower costs and enhance efficiency.

## PURPOSE AND SCOPE

The purpose of the online voting system is to provide a convenient platform for voters to

exercise their democratic right without hassles. The system seeks to eliminate the need for

standing in queues and using paper ballots, EVM machines that may be challenging to locate

a specific candidate. The scope of the system is vast, as it can be used for various elections,

ranging from local/state government to national assembly polls. Additionally, the benefits of

the system include an increase in voter turnout and enhanced accessibility for all. The

features of the system include a secure authentication and verification process using username

and PIN.

However, the potential challenges and concerns with the system include the possibility of

hacking or tampering with the votes, which can lead to false results. Implementing the system

would require collaboration between various government agencies, and there is a need

to create awareness among voters about the system's benefits. Overall, the purpose and scope

of the online voting system can revolutionize the way elections are conducted in India, and it is

essential to work towards its implementation in a secure and transparent manner. The system

supports various types of elections, such as governmental, organizational, and community-

based elections, and is designed to handle a diverse range of voting scenarios.

# ACTORS

The online election system primarily involves two main actors: voters and administrators.

Voters are the individuals who participate in the election by registering, logging in, and casting

their ballots through the online platform. Their primary interaction with the system includes

accessing their accounts, selecting candidates or options, and confirming their votes.

Administrators, on the other hand, manage and oversee the entire election process. Their

responsibilities encompass configuring election parameters, monitoring voter registration and

activity, handling technical support, and ensuring the smooth operation of the system.

Administrators also have access to tools for generating reports, auditing votes, and resolving

any issues that may arise, ensuring the integrity and efficiency of the electoral process.

# IDENTIFICATION OF NEED

System analysis is a crucial phase in the development of any system, involving the meticulous process of gathering and interpreting facts, diagnosing problems, and providing information to recommend system improvements. It is a problem-solving activity that requires intensive communication between system users and developers. System analysis or study is an important phase of any system development process. The system studies the minute test detail and gets analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system analysis process starts with a thorough examination of the current system, delving into minute details to understand its functioning. System analysts play the role of interrogators, deeply exploring how the present system operates. They view the system holistically, identifying inputs and tracing outputs back to the various processes within the organization.

The primary goals of system analysis are to become aware of the problem, identify relevant and decision-making variables, analyze and synthesize various factors, and determine an optimal or at least satisfactory solution or course of action. This involves a detailed study using various techniques such as interviews and questionnaires to collect data. The collected data is then scrutinized to understand the system's functionality, identifying any problem areas.

This understanding forms what is known as the existing system. The existing system is closely examined to pinpoint areas of difficulty. The system designer then acts as a problem solver, working to address the issues faced by the enterprise. Solutions are proposed and these proposals are weighed against the existing system. The best proposal is selected based on a thorough analytical comparison. The selected proposal is presented to the user for endorsement. If the user requests changes, the proposal is reviewed and modified accordingly. This iterative loop continues until the user is completely satisfied with the solution.

.

# FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose the organization for the amount of work.

A feasibility study is conducted to determine whether a project, upon completion, will fulfill the organization's objectives given the required amount of work, effort, and time investment. This study allows developers to anticipate the project's future and assess its usefulness. It evaluates a system proposal based on its practicality, impact on the organization, ability to meet user needs, and efficient resource utilization. Consequently, any new application typically undergoes a feasibility study before receiving approval for development.

The document provides the feasibility of the project that is being designed and lists various area that were considered very carefully during the feasibility study of this project such as Technical, Economic and operational feasibilities.

# WORKFLOW

This document plays a crucial role in the Software Development Life Cycle (SDLC) as it outlines the complete system requirements. It is intended for use by developers and serves as a fundamental reference during the testing phase. Any future changes to the requirements will need to go through a formal change approval process.

The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear- sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

As the earliest SDLC approach used for software development, the Waterfall Model

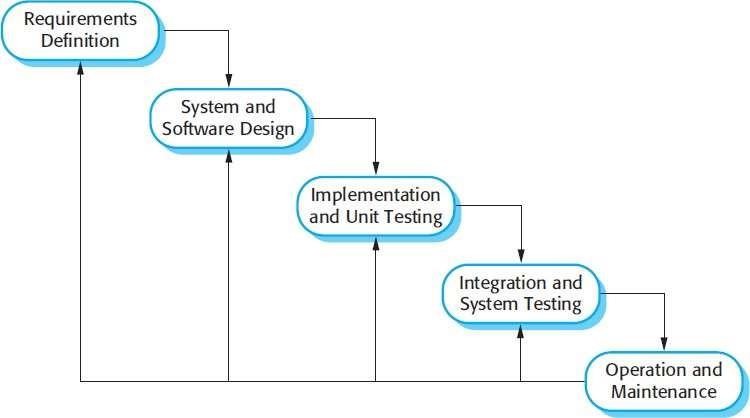
represents the development process in a linear sequential flow. This means that any

phase in the development process starts only after the previous phase is finished.

There is no overlapping of phases in the Waterfall model.

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

Following is a diagrammatic representation of different phases of waterfall model:



The sequential phases in Waterfall model are:

**Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

**System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

**Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

**Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

**Deployment of system:** Once the functional and nonfunctional testing is done, the product is deployed in the customer environment or released into the market.

**Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards(likeawaterfall)throughthephases.Thenextphaseisstartedonlyafterthedefinedset of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Waterfall Model Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* There are no ambiguous requirements.
* Ample resources with required expertise are available to support the product.
* The project is short.

The advantage of waterfall development is that it allows for departmentalization and control.

A schedule can be set with deadlines for each stage of development and a product can proceed

through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, trouble-shooting, and ends up at operation and maintenance. Each phase of development proceeds in

strict order.

## Module:

## STUDY OF THE SYSTEM

The modules used in this software are as follows:

* HOME: This page contains an overview of highlights for the entire system working

on with various features to guide users/voters.

* SUB-HOME: This page is redirected from home where user/admin can register or

login to vote or view profile.

* REGISTER: This module is for new users to register themselves.
* LOGIN: This module is for USER.
* ADMIN-LOGIN: This module is for ADMIN.
* ADMIN has the authority to Add candidates, Approve users, View results and Delete candidates.
* USER has the authority to register, login, vote and view their profile.
* LOGOUT: This module redirects back to sub-homepage.
* FEEDBACK: This module is for users to provide their valuable feedback.
* VOTING-POTRAL: This page displays user profile and the voting credentials, also assures foe one time voting.

# INPUT AND OUTPUT

The main inputs, outputs and the major functions are:

INPUT

User can register using username, password, aadhar-card number, voter-card number, mobile

number, email and image(optional) to login to voting page.

User can login using username, password, voter-card number and captcha to vote or view

profile once approved by admin.

User can vote their favorite candidate displayed and logout.

Admin can using username and password.

Admin can add or delete candidates to or from list.

Admin can approve valid users applicable to vote and also get the list of approved users.

OUTPUT

Admin receives notification for users waiting for approval and can approve only valid users.

Admin gets update of the results.

User cannot vote once already voted and gets alert message.

# SOFTWARE REQUIREMENT SPECIFICATIONS

Software Requirements Specification provides an overview of the entire project. It is a description of a software system to be developed, laying out functional and nonfunctional requirements. The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements, we need to have clear and thorough understanding of the project to be developed. This is prepared after the detailed communication with project team and the customer.

The developer is responsible for:-

* Developing the system, which meets the SRS and solving all the requirements of the

system.

* Demonstrating the system and installing the system at client’s location after acceptance testing is successful.
* Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
* Conducting any user training that might be needed for using the system.
* Maintain the system for a period of one year after installation.

#### HARDWARE REQUIREMENTS:

* Computer that has a 1.6 GHz or faster processor
* 4 GB (32 Bit) or 4 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
* HDD 20 GB Hard Disk Space and Above Hardware Requirements
* 5400 RPM hard-disk drive
* DVD-ROM Drive

#### SOFTWARE REQUIREMENTS:

* WINDOWS OS (Windows10, Windows11, windows XP)
* MySQL Server (5.6.12)
* XAMPP Server
* AdobeDreamweaverCS6

# SOFTWARE ENGINEERING PARADIGM APPLIED

Software paradigms refer to the methods and steps, which are taken while designing the software. There are many methods proposed and are in work today, but we need to see where in the software engineering these paradigms stand. These can be combined into various categories, though each of them is contained in one another.



Programming paradigm is a subset of Software design paradigm which is further a subset of Software development paradigm.

There are two levels of reliability. The first is meeting the right requirement. A carefully and through systems study is needed to satisfy this aspect of reliability. The second level of systems reliability involves the actual working delivered to the user. At this level, the systems reliability is interwoven with software engineering and development.

There are three approaches to reliability:

1. Error avoidance: Prevents errors from occurring in software.
2. Error detection and correction: In this approach errors are recognized whenever they are encountered and correcting the error by effect of error of the system does not fail.
3. Error tolerance: In this approach errors are recognized whenever they occur, but

enables the system to keep running through degraded perform or applying values that

instruct the system to continue process.

###### Maintenance:

The key to reducing need for maintenance, while working, if possible to do essential tasks.

1. More accurately defining user requirement during system development.
2. Assembling better systems documents.
3. Using some effective methods for designing, processing, and login and communicating information with project team members.
4. Making better use of existing tools and techniques.

# DATAFLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), 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 system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a dataflow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's dataflow diagrams to draw comparisons to implement a more efficient system. Dataflow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report.

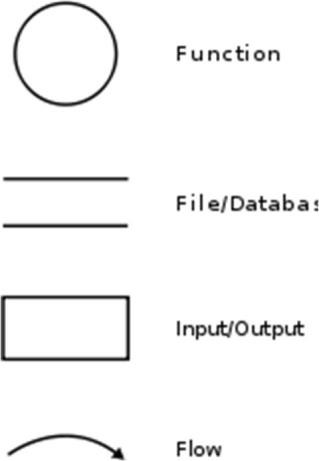
How any system developed can be determined through a dataflow diagram model?

In the course of developing a set of leveled data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to identify the transaction data in the data model.

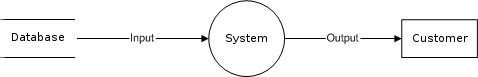
Dataflow diagrams can be used in both Analysis and Design phase of the SDLC.

There are different notations to draw dataflow diagrams, defining different visual

representations for processes, data stores, data flow, and external entities.

DFD NOTATION :

DFD EXAMPLE :



Steps to Construct DataFlow Diagram:-

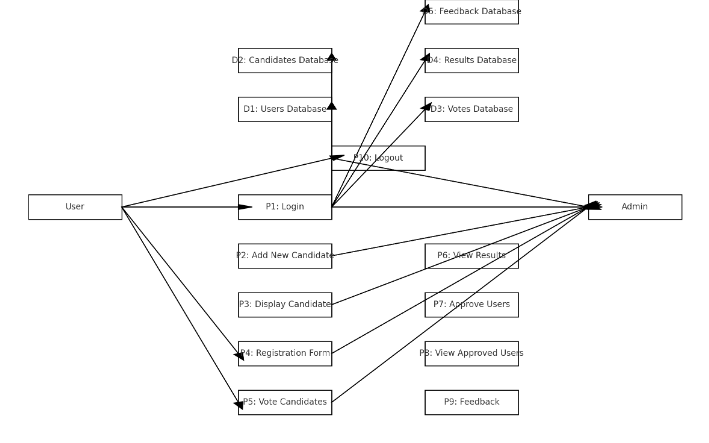
Four steps are generally used to construct a DFD:

* Process should be named and referred for easy reference. Each name should be representative of the reference.
* The destination of flow is from top to bottom and from left to right.
* When a process is distributed into lower-level details they are numbered.
* The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a DataFlow Diagram: -

* Arrows should not cross each other.
* Squares, Circles, Lines must bear a name.
* Decomposed dataflow squares and circles can have same names.
* Draw all dataflow around the diagram.

**LEVEL-1 DFD DIAGRAM**

****

# SEQUENCE DIAGRAM

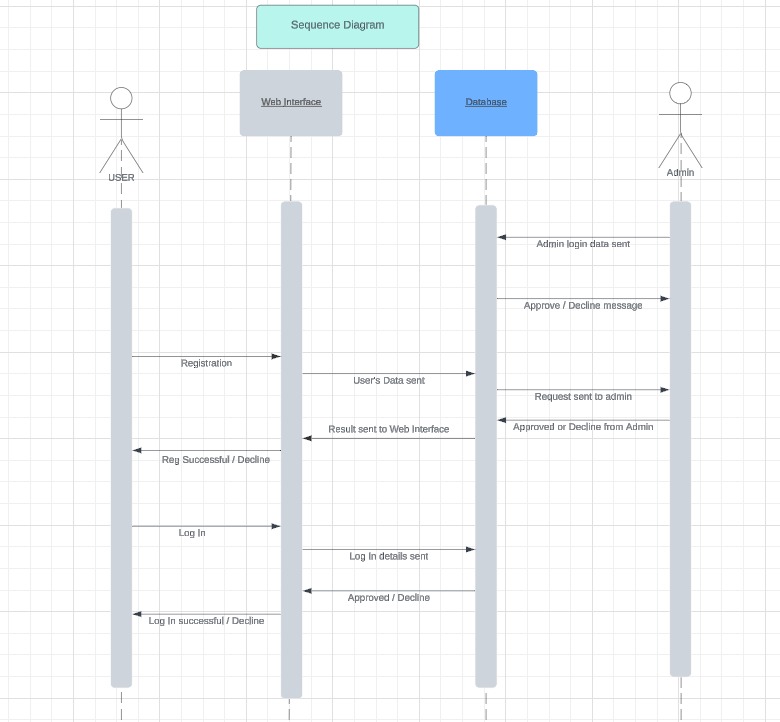
A **Sequence diagram** is an [interaction diagram](http://en.wikipedia.org/wiki/Interaction_diagram) that shows how processes operate with one another and what is their order. It is a construct of a [**Message Sequence Chart**](http://en.wikipedia.org/wiki/Message_Sequence_Chart). A sequence diagram 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. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

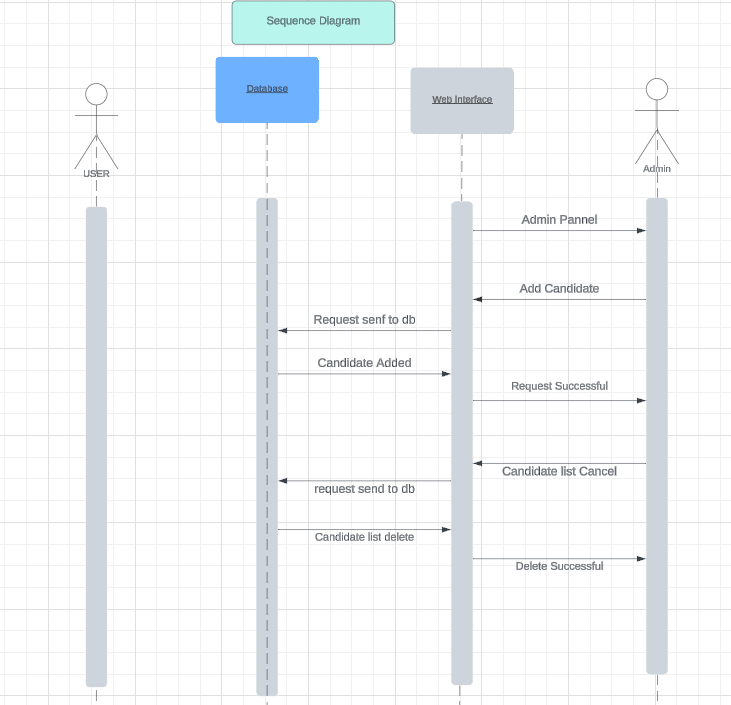
A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

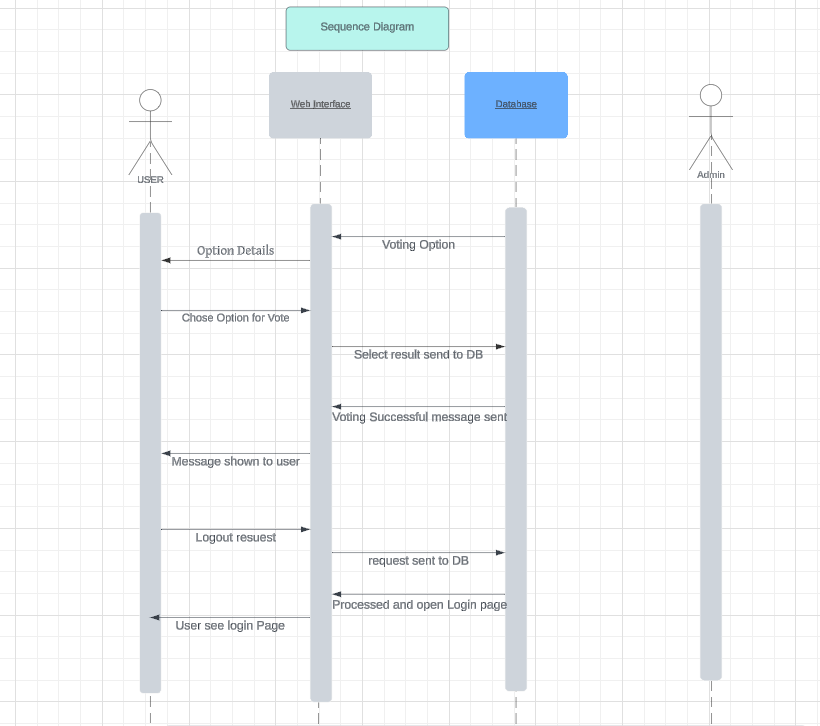
Sequence diagramis the most common kind of [interaction diagram](http://www.uml-diagrams.org/uml-25-diagrams.html#interaction-diagram), which focuses on the [message](http://www.uml-diagrams.org/sequence-diagrams.html#message) interchange between a number of [lifelines.](http://www.uml-diagrams.org/sequence-diagrams.html#lifeline)

Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

The following nodes and edges are typically drawn in a UML sequence diagram: [lifeline,](http://www.uml-diagrams.org/sequence-diagrams.html#lifeline) [execution-specification](http://www.uml-diagrams.org/sequence-diagrams.html#execution), [message](http://www.uml-diagrams.org/sequence-diagrams.html#message), fragment, interaction, [state invariant](http://www.uml-diagrams.org/sequence-diagrams.html#state-invariant), continuation, [destruction occurrence.](http://www.uml-diagrams.org/sequence-diagrams.html#destruction-occurrence-seq)







# ENTITY RELATIONSHIP DIAGRAM

In software engineering, an Entity-Relationship Model (ER model) is a data model used to describe the data or informational aspects of a business domain or its process requirements in an abstract manner. This abstraction facilitates the eventual implementation of the model in a database, such as a relational database. The primary components of ER models are entities (which represent objects or things) and the relationships that exist between these entities.

An Entity-Relationship Model (ER model) is the outcome of a systematic process used to describe and define a specific area of business data. It does not outline business processes; rather, it visualizes business data.

The data is represented as components (entities) that are interconnected by relationships, illustrating the dependencies and requirements among them. For example, a building may contain zero or more apartments, but each apartment can only belong to one building.

Entities can have various properties (attributes) that characterize them. Diagrams created

to graphically represent these entities, attributes, and relationships are called Entity-

Relationship Diagrams (ER diagrams).

An ER model is typically implemented as a database. In the case of a relational database,

which stores data in tables, each row in a table represents an instance of an entity. Some

data fields in these tables reference indexes in other tables; these references are the physical

implementation of the relationships.

The three-schema approach to software engineering involves developing three levels of

ER models:

#### Conceptual Data model:

The conceptual Entity-Relationship (ER) model typically defines master reference data entities commonly used across an organization. Developing an enterprise-wide conceptual ER model is valuable for documenting the organization's data architecture. This model can serve as the foundation for one or more logical data models. The primary purpose of the conceptual ER model is to establish structural metadata commonality for the master data entities across different logical ER models. It can also be used to create commonality relationships between ER models, providing a basis for data model integration..

#### Logical Data Model

The logical Entity-Relationship (ER) model contains more detail than the conceptual ER

model. In addition to master data entities, it defines operational and transactional data

entities. The specifics and attributes of each data entity are elaborated upon, and the

relationships between these entities are meticulously established. However, the logical ER

model is developed independently of the underlying technology into which it will be eventually

implemented.

#### Physical Data Model

One or more physical Entity-Relationship (ER) models may be developed from each logical ER

model. The physical ER model is designed to be instantiated as a database, and therefore, must

contain sufficient detail to produce a fully functional database. Each physical ER model is

technology-dependent, as each database management system has advanced, unique applicable characteristics.

The physical model is typically instantiated in the structural metadata of a database management

system (DBMS) as relational database objects, such as tables, unique key indexes, and constraints

like foreign key constraints or commonality constraints. The ER model is also used to design modifications to these relational database objects and to maintain the structural metadata of the

database.

The initial stage of information system design utilizes these models during requirements analysis

to describe the information needs or the types of information to be stored in a database. The data modeling technique can describe any ontology, providing an overview and classification of terms

and their relationships within a specific area of interest.

In the design of an information system based on a database, the conceptual data model is mapped

to a logical data model during the logical design phase. This logical model is subsequently mapped

to a physical model during the physical design phase. It is important to note that both of these

phases are sometimes collectively referred to as "physical design". The ER model is also crucial

in the broader context of database management systems.

#### ENTITY RELATIONSHIP DIAGRAM

#### 

# USECASE DIAGRAM

A use case diagram, in its simplest form, represents how users interact with a system by showing the relationships between users and the various use cases they participate in. It helps identify different types of users and the specific use cases they are involved with. Use case diagrams are often used alongside other diagrams to provide a more complete understanding of the system's functionality and user interactions.

Modeling a system requires more than just capturing its static behavior; dynamic behavior is equally crucial. In UML, there are five types of diagrams designed to model dynamic aspects of a system, and the use case diagram is one of these. Although use case diagrams themselves focus on the dynamic interactions, they involve internal and external factors known as actors.

Use case diagrams consist of actors, use cases, and their relationships. They are employed to model the functionality of a system or subsystem. Each use case diagram represents a specific functionality of the system, and to model the entire system comprehensively, multiple use case diagrams are typically used.

The purpose of a use case diagram is to capture the dynamic aspects of a system, but this

description is broad since other diagrams, like activity, sequence, collaboration, and state

chart diagrams, also serve similar purposes. What sets use case diagrams apart is their role in gathering system requirements, including both internal and external influences, primarily

focusing on design requirements. They are used to identify functionalities and actors during

system analysis. Once the initial requirements are gathered, use case diagrams are created to

present the external view of the system.

So, in brief, the purposes of use-case diagrams can be as follows:

* Used to gather requirements of a system.
* Used to get an outside view of a system.
* Identify external and internal factors influencing the system.
* Show the interaction among the requirements and actors.

How to draw Use-case Diagram?

Use-case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use-cases are the actors. Actors can be defined as something that interacts with the system.

The actors can be human user, some internal applications or may be some external applications. So in a brief when we are planning to draw a use case diagram we should have the following items identified.

* + Functionalities to be represented as an use-case
  + Actors
  + Relationships among the use-cases and actors

Use case diagrams are drawn to capture the functional requirements of a system. So after

identifying the above items we have to follow the following guidelines to draw an efficient

use-case diagram.

* The name of a use-case is very important. So the name should be chosen in such a

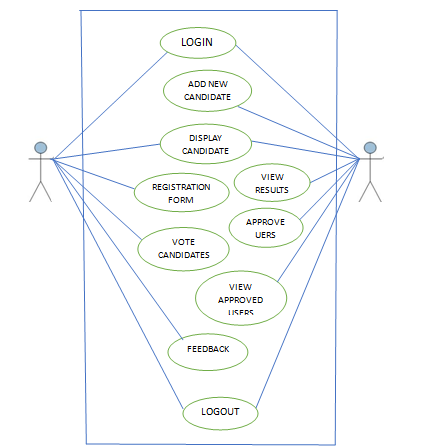
way so that it can identify the functionalities performed.

* Give a suitable name for actors.
* Show relationships and dependencies clearly in the diagram.
* Do not try to include all types of relationships. Because the main purpose of the

diagram is to identify requirements.

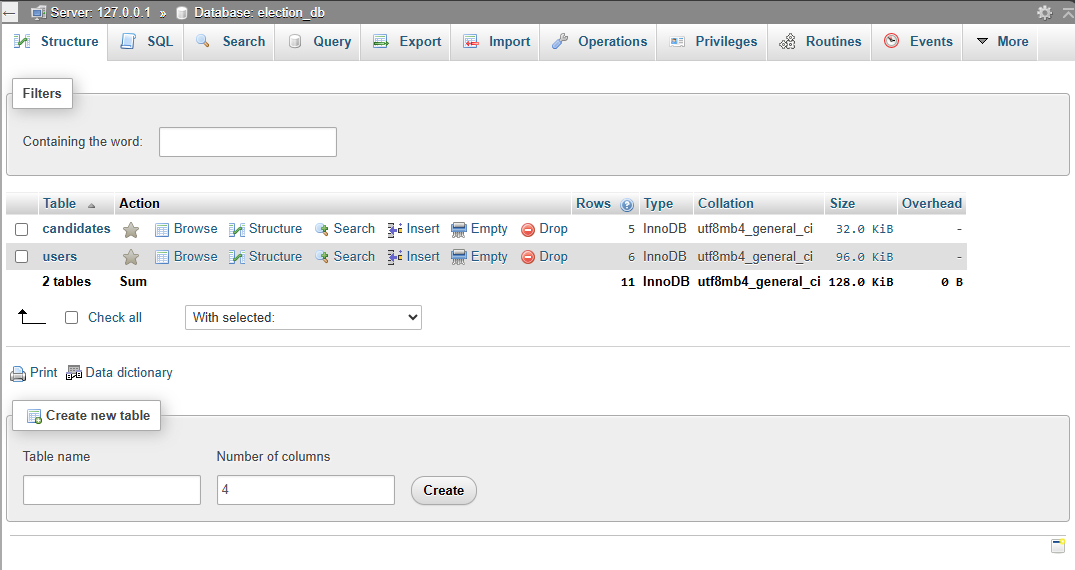
* Use note whenever required to clarify some important points.

#### USECASE DIAGRAM

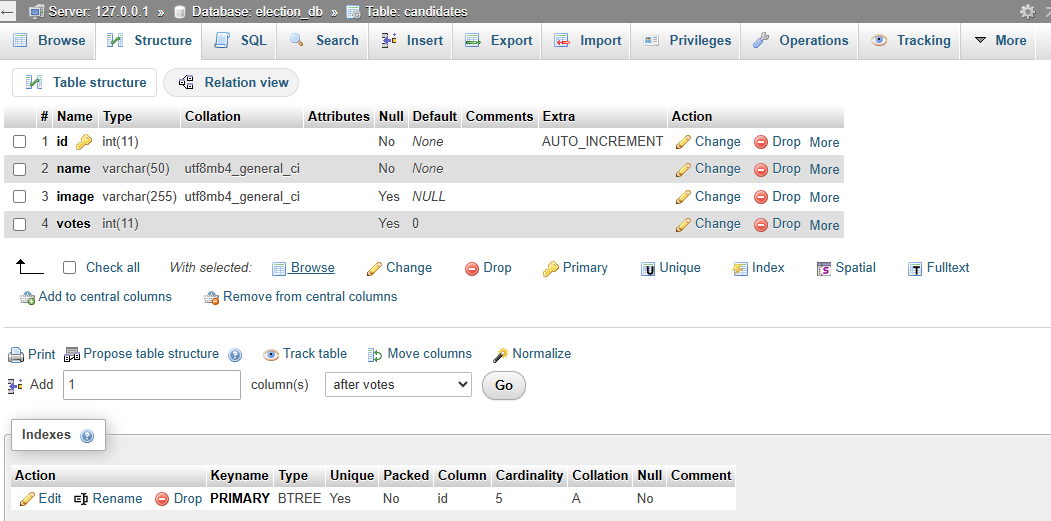
****

# DATA DICTIONARY

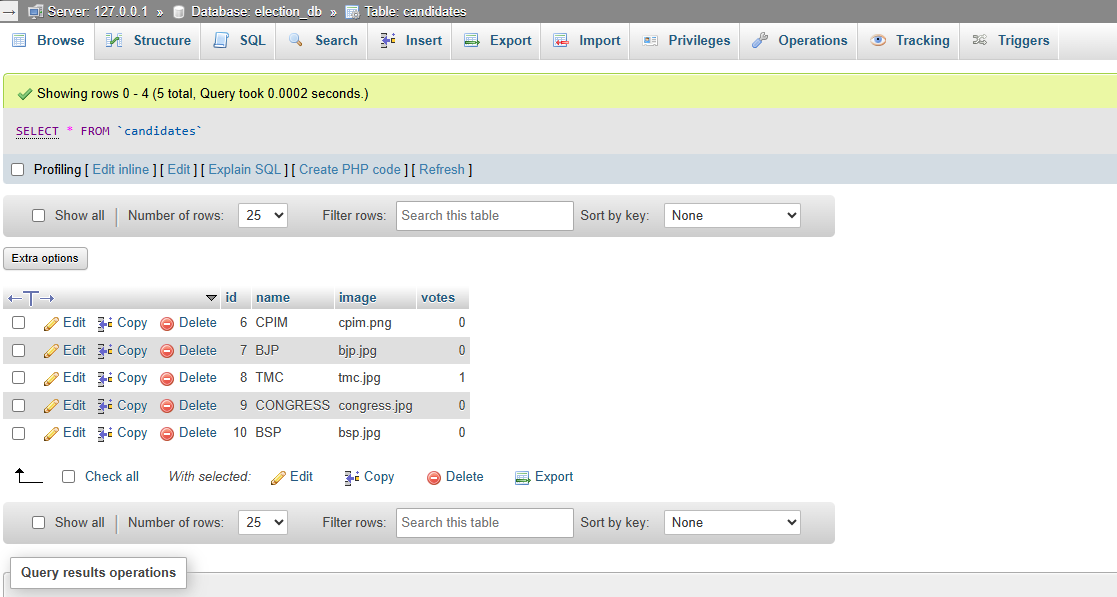
Database



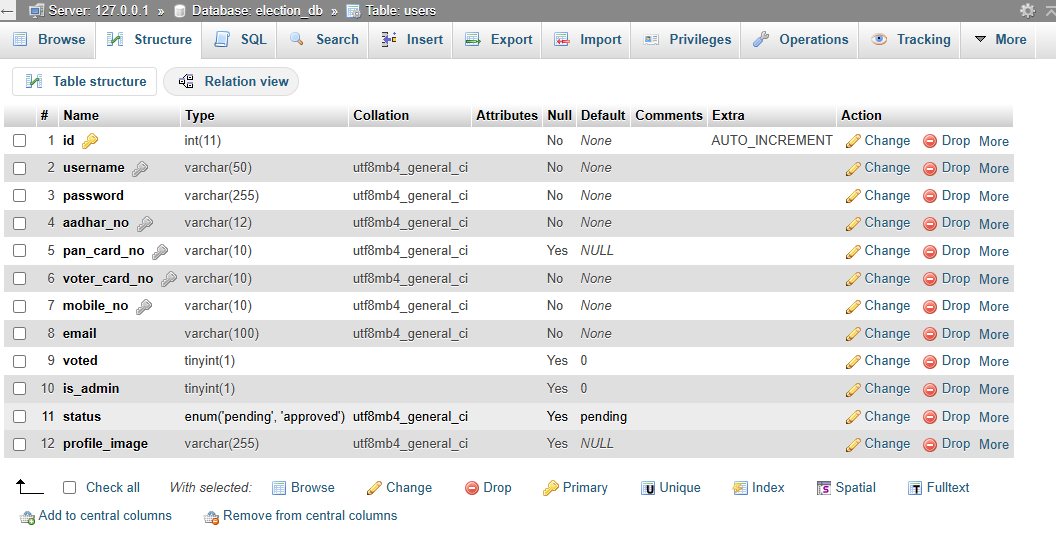
Candidates Table



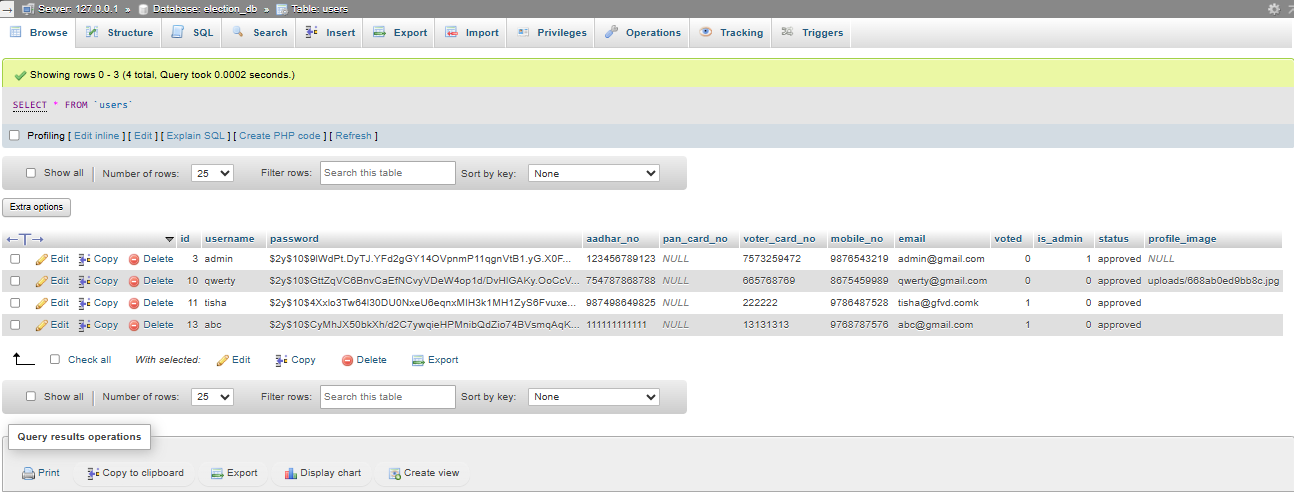
Candidates Table Data



User Table



**User Table Data**

****

# USER INTERFACE DESIGN

User Interface Design (UID) or User Interface Engineering focuses on designing interfaces for

machines and software, including computers, home appliances, mobile devices, and other

electronic systems, with the aim of maximizing user experience. The primary goal of user

interface design is to make interactions as simple and efficient as possible, ensuring that users

can accomplish their goals with ease (user-centered design).

Effective user interface design allows users to complete tasks seamlessly without

unnecessary distractions. It employs graphic design and typography to enhance usability and

aesthetic appeal, as the visual aspects of the design can influence user interactions and

overall satisfaction. The design process must balance technical functionality with visual elements,

such as mental models, to create systems that are both operational and adaptable to evolving user

needs.

Interface design spans a wide range of projects, from computer systems to automobiles and

commercial aircraft. While these projects share fundamental human interaction principles, they

also require specialized skills and knowledge. Consequently, designers often specialize in specific

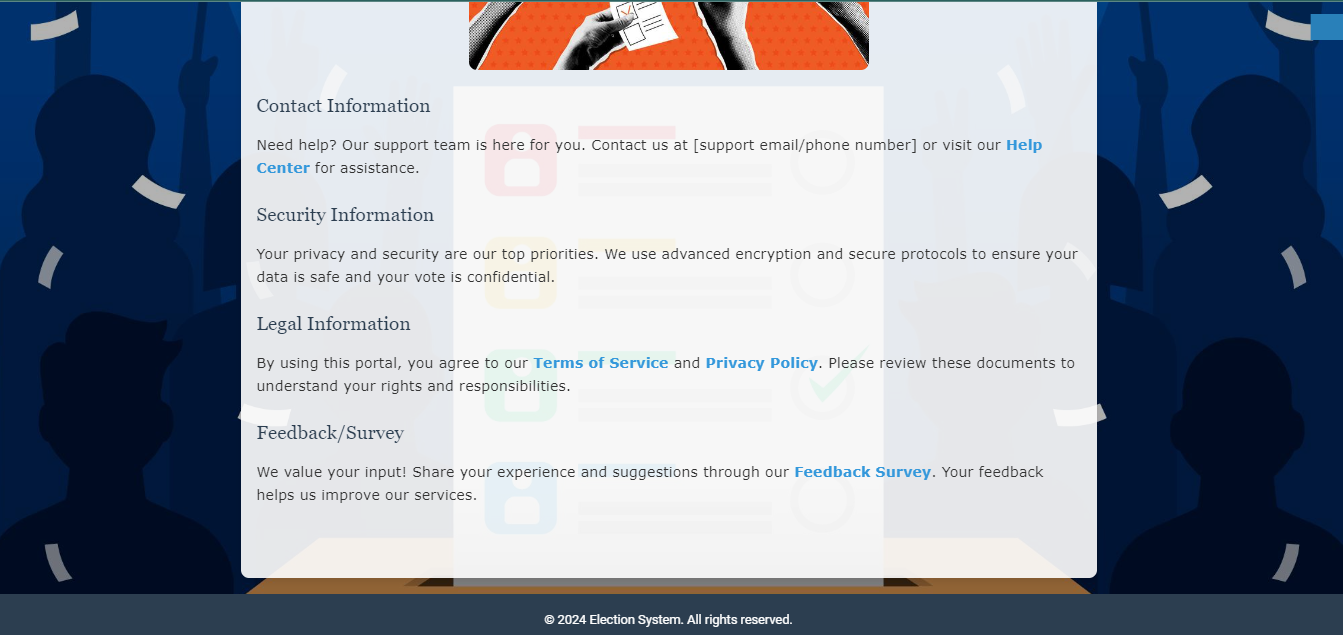
areas, such as software design, user research, web design, or industrial design, based on their

expertise.

Homepage

# SNAPSHOTS

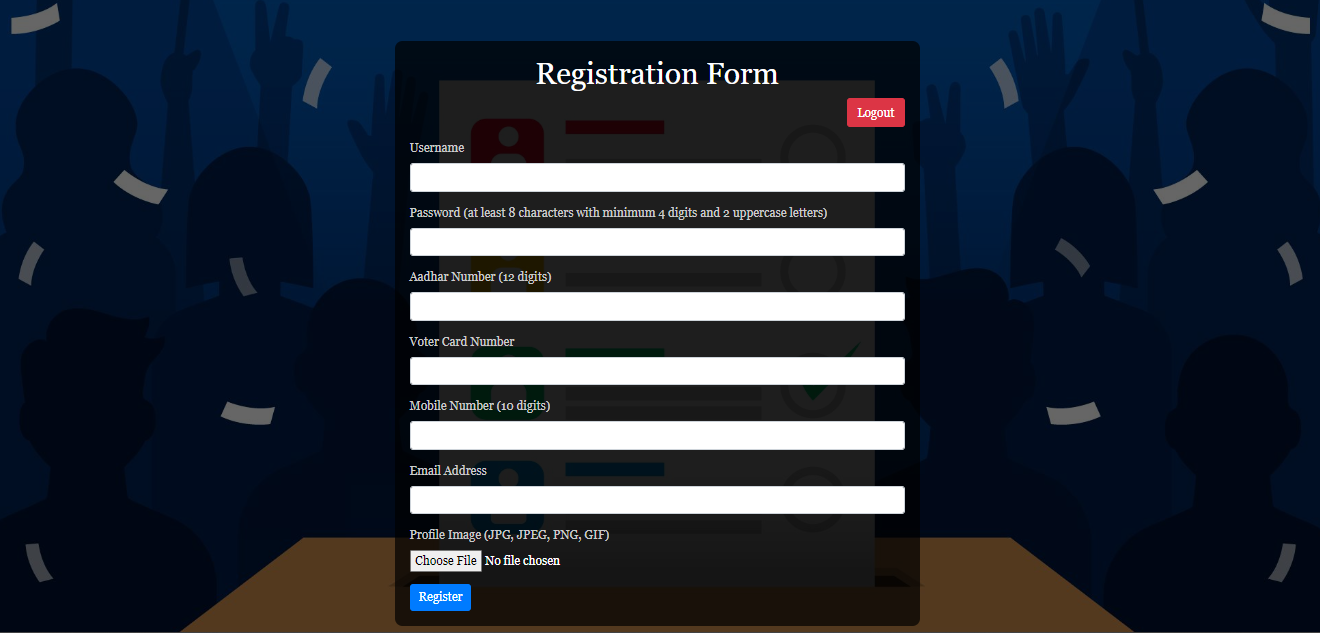




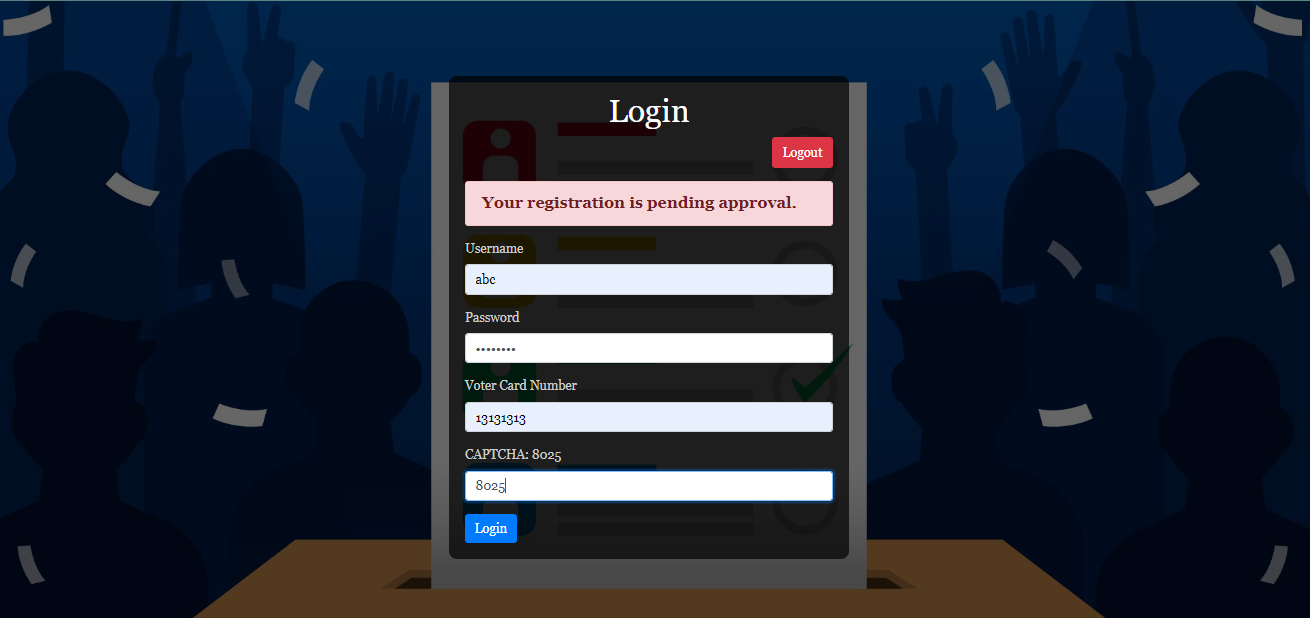
Sub-Homepage



User Registration Page



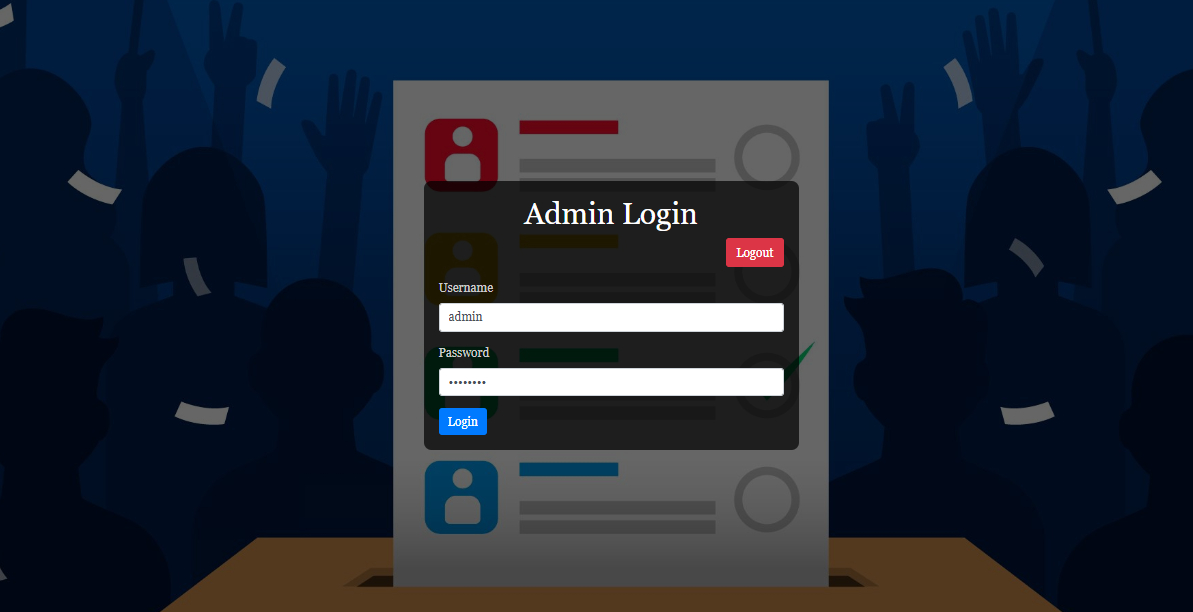
User Login Page



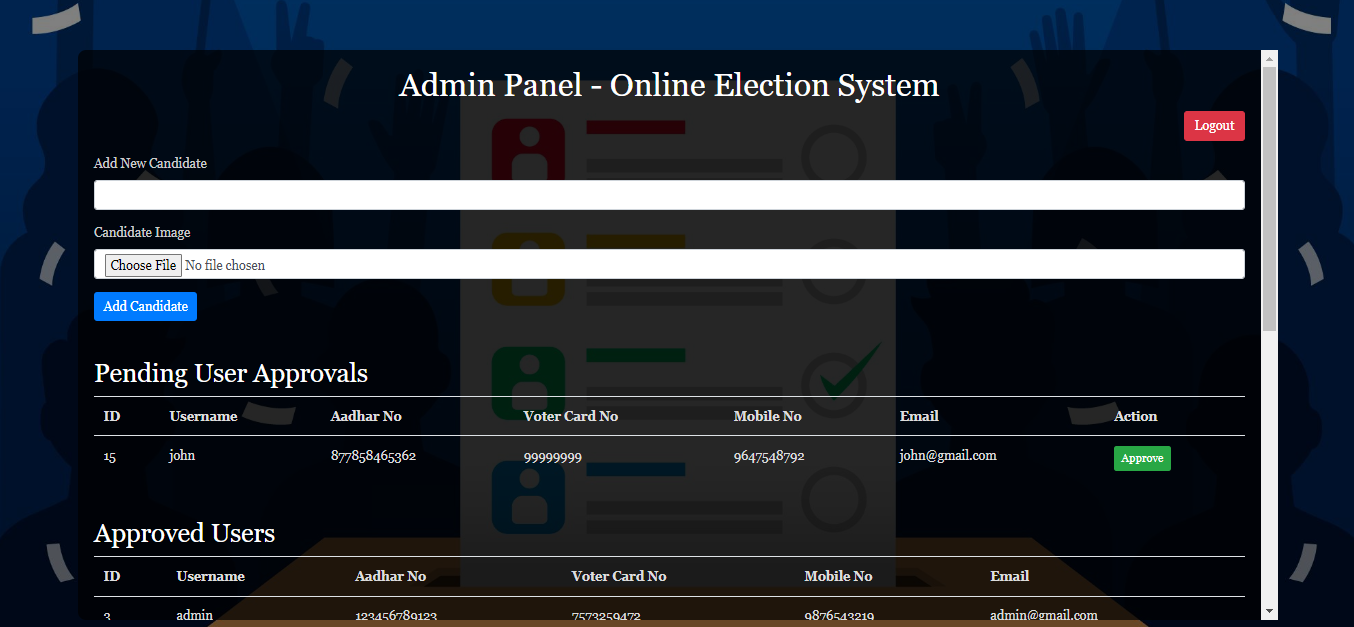
Voting Page

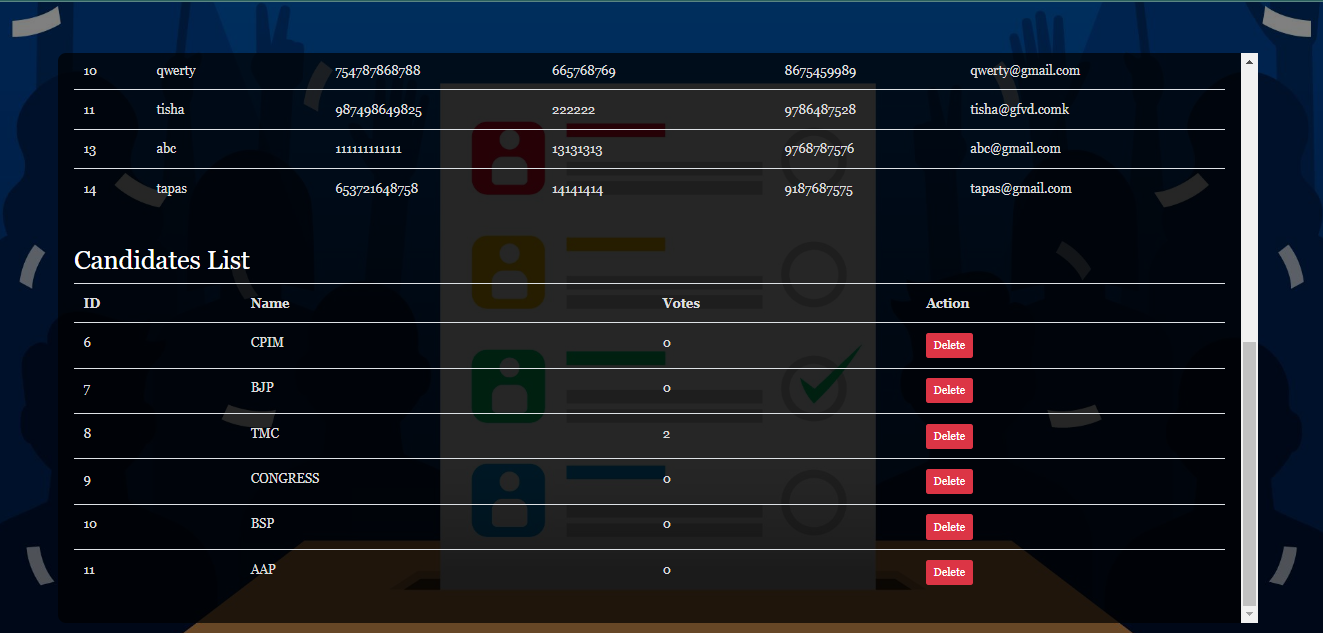


Admin Login Page

****

Admin Panel





# IMPLEMENTATION AND TESTING

A software system test plan is a document that describes the objectives, scope, approach and focus of software testing effort. The process of preparing a test plan is a usual way to think the efforts needed to validate the acceptability of a software product. The complete document will help people outside the test group understand the "WHY" and "HOW" product validation. It should be through enough to be useful but not so through that no one outside the test group will read it.

**INTRODUCTION**

Testing is the process of executing a system to identify errors and ensure its integrity. By detecting deviations from the design and uncovering errors, testing enhances the system's reliability. It focuses on identifying error-prone areas to prevent issues before they occur. Additionally, testing adds value to the product by verifying that it meets user requirements and performs as expected.

The primary purpose of testing is to identify errors and pinpoint areas prone to errors within a system. Testing must be comprehensive and carefully planned, as a system that is only partially tested is as risky as one that is not tested at all. The cost of an untested or inadequately tested system can be substantial.

The implementation phase is crucial and involves user training and system testing to ensure the successful deployment of the system. During this phase, users test the system, and adjustments are made based on their feedback to meet their needs effectively.

**OBJECTIVES OF TESTING**

The objective of our test plan is to identify and report as many bugs as possible to enhance the

integrity of our program. While exhaustive testing is not feasible, we will employ a wide range

of tests to achieve this goal. Our user interface is designed to be user-friendly, allowing for easy

manipulation of the tree structure. Although the application is intended primarily as a

demonstration tool, we aim to ensure that it operates efficiently and remains user-friendly across

various platforms that can be made to run on with minimal impact on the overall system

performance.

# PROCESS OVERVIEW

The following represents the overall flow of the testing process:

* + 1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
    2. Identify which particular test(s) will be used to test each module.
    3. Reviewthetestdataandtestcasestoensurethattheunithasbeenthoroughlyverifiedand that the test data and test cases are adequate to verify proper operation of the unit.
    4. Identify the expected results for each test.
    5. Document the test case configuration, test data, and expected results. Perform the test(s).
    6. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
    7. Successful unit testing is required before the unit is eligible for component integration/system testing.
    8. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
    9. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

# TEST CASES

A test case is a document that describes an input, action, or event and expected response, to determine if a feature of an application is working correctly. A test case should contain particular elements such as test case identifier, test condition, input data requirement expected results. The process of developing test cases can help find problems in the requirement or design of an application, since it requires completely thinking through the operation of the application.

TESTING STEPS:

Unit Testing:

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

Integration Testing:

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

Validation:

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

The function or performance characteristics confirm to specification and are accepted. A

deviation from specification is uncovered and a deficiency list is created. Proposed system

under consideration has been tested by using validation test and found to be working

satisfactory.

|  |  |  |  |
| --- | --- | --- | --- |
| Tested By: | | Ritesh Ghosh | |
| Test Type | | Unit Testing | |
| Test Case Number | | 1 | |
| Test Case Name | | Admin Identification | |
| Test Case Description | | The admin should enter his/ her accurate username and password so that he/she can be able to access the admin panel. The test case will check the application for the same since the admin can only login with a particular username and password set before. | |
| Item(s) to be tested | | | |
| 1 | Verification of the username and password with the record in the database. | | |
| Specifications | | | |
| Input | | | Expected Output/Result |
| Correct Username and password Incorrect username or Password | | | Successful login Failure Message |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tested By | | Ritesh Ghosh | | |
| Test Type | | Unit Testing | | |
| Test Case Name | | 2 | | |
| Test Case Name | | User Approval | | |
| Test Case Description | | Once a user registers in the site, it has to wait for the admin’s approval to login to the voting portal. If the admin after verifying the user details thinks him/her valid and approves, then the user can login successfully login. | | |
| Item(s) to be tested | | | | |
| 1 | Approval by admin for successful user login | | | |
| Specifications | | | | |
| Input | | | | Expected Output/Result |
| Admin approves  Admin does not approve | | | Successful user login  Pending approval message | |  |

###### WHITE-BOX TESTING:

In white-box testing, the UI is by-passed. Inputs and outputs are tested directly at the code level and the results are compared against specifications. This form of testing ignores the function of the program under test and will focus only on its code and the structure of that code. Test case designers shall generate cases that not only cause each condition to take on all possible values at least once, but that cause each such condition to be executed at least once. To ensure this happens, we will be applying Branch Testing. Because the functionality of the program is relatively simple, this method will be feasible to apply.

Each function of the binary tree repository is executed independently; therefore, a program flow for each function has been derived from the code.

**BLACK-BOX TESTING:**

Black-box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Equivalence Partitioning and Boundary Value Analysis testing on our application.

SYSTEM TESTING:

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity. But in our case we will focus only on function validation and performance. And in both cases we will use the black-box method of testing.

**OUTPUT TESTING:**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

.

USER ACCEPTANCE TESTING:

User acceptance of a system is the key factor for the success of any system. The system

under consideration is tested for the user acceptance by constantly keeping in touch with

the prospective system users at the time of developing and making changes whenever

required.

This is done in regard to the following points:

* + 1. **Input Screen Design**
    2. **Output Screen Design**
    3. **Format of reports and other outputs**

#### GOAL OF TESTING:

"Program testing can be used to slow the presence of bug, but never to slow their absence." If the results delivered by the system are different from the excepted ones then the system is incorrect and these bugs should be fixed.

# INTEGRATION TEST REPORTS

Software testing is always used in association with verification and validation. In the testing phase of this project our aim is to find the answer to following two questions.

* + - Whether the software matches with the specification (i.e. process base) to verify the product.
    - Whether this software in one client what wants (i.e. product base) to validate the product.

Unit testing and integration testing has been carried out to find the answer to above questions. In unit testing each individual module was test to find any unexpected behavior if exists. Later all the module was integrated and flat file was generated.

#### FUNCTIONAL TESTING:

These are the points concerned during the stress test:

**Nominal Input:** Character is inputted in the place of digits and the system has to flash the message "Data error".

**Boundary Value Analysis**: Exhaustive test cases have been designed to create an output report

that produces the maximum (and minimum) allowable number of table entries.

## Testing Method Used:

We have adopted a testing method which is a mix of both (structural) and black box (functional) testing. For modules we have adopted white box testing. Then we integrated the module into sub- systems and further into the system. Thus we adopted black box testing for checking the

correctness of the system.

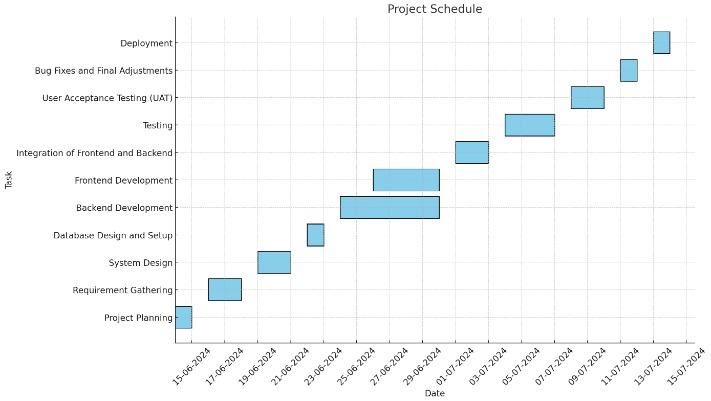
**Requirements Validated and Verified:**

* + The data is getting entered properly into database.
  + The screens are being loaded correctly.
  + The various functions specified are being performed completely.

# PROJECT SCHEDULE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Task** | **Start Date** | **End Date** | **Duration (days)** | **Status** |
| **1** | **Project Planning** | 14-06-2024 | 15-06-2024 | 2 | **Done** |
| **2** | **Requirement Gathering** | 16-06-2024 | 18-06-2024 | 3 | **Done** |
| **3** | **System Design** | 19-06-2024 | 21-06-2024 | 3 | **Done** |
| **4** | **Database Design and Setup** | 22-06-2024 | 23-06-2024 | 2 | **Done** |
| **5** | **Backend Development** | 24-06-2024 | 30-06-2024 | 7 | **Done** |
| **6** | **Frontend Development** | 26-06-2024 | 30-06-2024 | 5 | **Done** |
| **7** | **Integration of Frontend and Backend** | 01-07-2024 | 03-07-2024 | 3 | **Done** |
| **8** | **Testing** | 04-07-2024 | 07-07-2024 | 4 | **Done** |
| **9** | **User Acceptance Testing (UAT)** | 08-07-2024 | 10-07-2024 | 3 | **Done** |
| **10** | **Bug Fixes and Final Adjustments** | 11-07-2024 | 12-07-2024 | 2 | **Done** |
| **11** | **Deployment** | 13-07-2024 | 14-07-2024 | 2 | **Done** |

**GANTT CHART**



# DATABASE SECURITY

System security measure is meant to be provided to make your system reliable and secured from unauthorized user may create threats to the system. So, you should follow some security measures. We have used security levels in database level at system level. The system should ensure high availability, especially during the critical voting period. Redundancy and failover mechanisms

should be in place to minimize downtime. Regular backups and a robust disaster recovery plan

should be implemented to protect against data loss. It should implement data integrity checks to

prevent and detect any unauthorized changes to voter data or votes.

## SYSTEM SECURITY

If we talk about the system security in our proposed system we have implemented with the help of

maintain the session throughout the system’s use. Once a user has logged out then he/she will not

be able to perform any task before signing back again.

A high level of authentic login is given to the system so this is a very tedious task to enter without authorization and authentication. The system must employ strong encryption protocols to protect

data both in transit and at rest. It should also include security measures like firewalls, intrusion

detection systems, and regular security audits.

## LIMITATIONS

While online voting has the potential to increase voter turnout and reduce costs, there are

several challenges and concerns that must be addressed. One of the biggest concerns is the

security and privacy of the voting process. It is crucial that the authentication and verification

methods are secure and so that no outside interference can corrupt the results. Additionally,

there is a concern that online voting could disenfranchise certain groups who may not have

easy access to the internet or who may not be tech-savvy. Another challenge is the potential for

technical difficulties or glitches during the voting process.

It is important that the system is reliable and can handle a high volume of users without crashing.

Finally, there is a concern that online voting could lead to a lack of transparency in the election

process.

.

## CONCLUSION

In conclusion, Online Voting System is a highly innovative and technological solution to many

of the challenges faced in traditional voting systems. It not only simplifies voting process but

also saves time and resources. With secure authentication and verification measures in place,

the system offers transparency and accountability in the electoral process. However, the

implementation of the online voting system still raises concerns regarding security, privacy,

and accessibility. It is important to address these issues before making the system available to

the public. The online voting system has tremendous potential to revolutionize the electoral

process and ensure greater participation and representation for all.

The way forward is to focus on developing a comprehensive security infrastructure and

addressing concerns to make the system accessible to all. By doing so, we can create a more

inclusive and democratic electoral process for the future.

## FUTURE SCOPE

The future scope of the online election system is marked by exciting opportunities for

enhancement and innovation. As technology continues to advance, the system is expected to

incorporate more sophisticated security measures, such as biometric authentication and

block chain technology, to further safeguard the integrity of the voting process. Artificial

intelligence and machine learning will likely play a crucial role in real-time fraud detection,

voter behavior analysis, and optimization of election logistics, making the system both smarter

and more efficient. Accessibility features will be expanded to support a wider range of languages

and disabilities, ensuring that the system is inclusive and user-friendly for all voters. Additionally,

there will be a focus on refining the user experience with more intuitive interfaces and

integrating emerging technologies like voice assistants and augmented reality. The system will

also aim for global adaptability, supporting diverse legal and regulatory requirements for various

types of elections across different countries.

Future developments may include enhanced analytics tools for in-depth reporting and insights into election trends and system performance. Furthermore, integration with mobile platforms and

Internet of Things (IoT) devices will provide voters with more flexible access options and real-

time updates. These advancements will collectively contribute to a more secure, accessible, and

efficient online election system, aligning with the evolving needs of modern democracies and organizations.

# BIBLIOGRAPHY

<https://projectworlds.in/free-projects/php-projects/online-voting-system-project-in-php-2>

<https://www.browserstack.com/guide/build-a-website-using-html-css>

<https://www.w3schools.com/php/php_mysql_intro.asp>

https://www.sourcecodester.com/php/4808/voting-system-php.html

https://youtu.be/E9bRLp0bgm4?si=MXY49t6kqb50pPDj