**Identify the software project, create business case, arrive at a problem statement**

**DATE:**

**EX NO: 1**

**INTRODUCTION:**

EVote is an election system that facilitates voters to record their secure and secret ballot electronically. It has a friendly user interface and enables voters to cast their votes in few simple steps. We ensure the authenticity of the voters and the votes cast by them along with non-traceability of the casted vote.

EVote’s robust architecture has persistently manifested to be one of the most reliable, comprehensible and economical electronic voting solution.

**PROBLEMS:**

Basically, offline election is not only about voting, it’s also about counting. If voting takes some time, then counting will take much more time.

It becomes very difficult to manage record of all voters manually which results in heap of files. So, there is always a risk of loss of information due to various reasons.

Also, there is imbalance between the manpower availability and magnitude of the voting work which results in inaccuracy and delay in declaration of result.

**PROPOSAL:**

The online voting system is consist of the data and information:

* The database of the Voter's information and details
* Calculation of total votes
* Checking information by the voter
* Remove wrong information

**1.2 ONE PAGE BUSINESS CASE TEMPLATE: THE PROJECT:**

The Project is developed for the threat free and user oriented Online Voting System. The election can be conducted in two ways the paper ballot election and the automated ballot elections. The automated ballot elections are called the electronic voting. The online voting system is highly developed and the online polling system can be replaced by accurately and directly voting online and immediate results. The online voting system is done by the internet so it can be called the Internet Voting.

**THE HISTORY:**

The manual system does not provide secured registration and profile management of the Applicants properly. In manual system, all details are taken in the form of documents which can be misplaced anytime. Every individual had to stand in queue, which is more time consuming and involves more manpower.

**CONSTRAINS:**

The Applicants require a computer to submit their information. Prior knowledge of computers and English language should be known.

Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring. The Applicant must be careful while submitting the information.

**APPROACH:**

We approach this solution from the technical point of view. So, we are keen in maintaining an easy way to execute interface for the applicant as well as the back-end maintenance. The development of this project will require front end and back-end programming knowledge. Thus, it is important to equip with the same.

**BENEFITS:**

* Security is provided.
* Less time consumption.
* Improves the efficiency.
* Reduces the complexities.
* Elimination of human error.

**Stakeholder and user description, identify the appropriate process models, comparative study with agile model**

**DATE:**

**EX NO: 2**

# 2.1 IDENTIFYING STAKEHOLDERS:

1. **User:**

The persons who will use the project’s service are known as users.

In this project, users will be

* + Voters
  + Admin

Each of these types of users has different the system so each of them has their own requirements. The voters can only use the application to view their profile, result, and club president. This means that the user is expected to be Internet literate. The user interface will be as intuitive as possible.

# Sponsor:

Sponsor is the person or the organization that provides financial support for the project. As of now, this project is self-sponsored. Once we upgrade, we may require financial interference of a management.

# Program Manager:

A Program Manager articulates a program's strategy and objectives and assesses how it will impact.

# Project Management Office(PMO):

A project management office (PMO) is a team or department that sets and maintains standards for project management throughout an organization.

# Project Manager:

Project Managers (PMs) are responsible for planning, organizing, and directing the completion of specific projects for an organization while ensuring these projects are on time, on budget, and within scope.

# Project Team:

A project team is comprised of the project manager, project management team and the other members who carry out work.

# BENEFITS:

1. **Economic Feasibility:**

This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor. All hardware and software cost has to be borne by the organization. Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system.

# Technical Feasibility:

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS) and checked if everything was possible using different type of frontend and backend platforms.

# Operational Feasibility:

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

# IDENTIFY THE APPROPRIATE PROCESS MODELS: Process Model:

In our software, we are following prototyping Model (Evolutionary) as initially the requirements are not clear. The requirements are added in the prototype as and when they are understood. Software reviews are applied at various points during software engineering and serve to uncover the errors and defects that can be removed.

# COMPARISION BETWEEN WATERFALL AND AGILE MODEL:

**Why agile model is better than waterfall model?**

* + - The Agile Model is based on iterative development and hence it divides the entire project into smaller parts which reduces the risk factor which is not the case in waterfall model.
    - The Waterfall model cannot accept the changes in requirements but in agile model it is easy to change the system requirements.
    - In agile model, the entire project is divided into smaller parts which helps to minimize the project risk and to reduce the overall project delivery time requirements.
    - In waterfall model since risk factor is high, it is not suitable for complex projects.
    - In waterfall model the testing is done in later stage it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare, which is not the case in agile model.
    - In waterfall model, it follows a sequential approach whereas in agile model it explains the process in order of incremental approach.
    - In agile it performs the testing concurrently with software development whereas in waterfall model the testing comes after the build phase only.
    - In agile model the distance between the customer and developer is in short whereas in waterfall model it is long.
    - In agile there can be done any change in the project but in waterfall model there is no changes throughout the project work.

**Identify the Requirements, System Requirements, Functional Requirements, Non-functional Requirements**

**DATE:**

**EX NO: 3**

# REQUIREMENTS:

Requirements are defined during the early stages of the system development as a specification of what should be implemented. A collection of requirements is a requirements document. They may be user level facility description, detailed specification of system behaviour, general system property, a specific constraint on the system or information on how to carry on computation. The three types of requirements are explained below

# SYSTEM REQUIREMENTS:

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash.

* + - 1. **Hardware requirements:** Processor: Intel Pentium 4.0 Ram: 2GB

Hard Disk: 500GB

* + - 1. **Software requirements:** Operating System: Windows. Front-End: HTML, PHP. Database: MYSQL.

Model Design: Rational Rose

# 3.1.2 FUNCTIONAL REQUIREMENTS:

1. **Subject Information Management:**

The system will maintain information about various subjects being offered during different semesters of the course. The following information would be maintained for each subject. Subject code, Subject

Name, Subject Type (Core / Elective /Lab1 / Lab2 / Mini Project) Semester, Credits.

# Student Information Management:

System will maintain information about various students enrolled in the different courses in different years. The following information would be maintained for each student: Student enrolment No, Student Name, Year of enrolment. The system will allow creation/modification/deletion of new/ existing students and also have the ability to list all the students enrolled in a particular year.

# Student’s Subject Information Management:

The system will maintain information about different Elective subjects of different enrolment years in different semesters of students. The following information would be maintained: Student enrolment no, Semester, Student’s choices for a particular semester.

# Marks Information Management:

The system will maintain information about marks obtained by various students of different enrolment years in different semesters. The following information would be maintained: Student enrolment no, semester, subject code, internal marks, external marks, total marks and credits.

# Mark Sheet Generation:

The system will generate mark-sheet for every student in different semesters.

# Report Generation:

* + Student List Reports.
  + For each year a report will be generated containing the list of students enrolled in that batch year.
  + Student Subject List Report.
  + For each batch year a report will be generated containing the list of students.
  + Elective subject in the semester.
  + Semester-wise mark lists.
  + Rank-wise List Report.

# User Account Management:

The system will maintain information about various users who will be able to access the system. The following information would be maintained. Username, User ID, Password and Role.

# NON-FUNCTIONAL REQUIREMENTS:

* + - * Readability
      * Availability
      * Maintainability
      * Security
      * User Friendly
      * Performance
      * Efficiency
      * Privacy

**Prepare project plan based on scope, find job roles and responsibilities, calculate project effort based on resources**

**DATE:**

**EX NO: 4**

# 4.1 PROJECT PLAN:

1. **PROJECT NAME:**

“Online Voting Website”

# PROJECT MEMBERS:

Our project consists of three members:

* + Kaniskh. V - RA2011026020093
  + Vineesh. K - RA2011026020120
  + Praveen Raj. A – RA2011026020124

# MODULES:

**Voter:** Online Voting System Project in Python.

**Admin:** Online Voting System Project in Python.

# Admin Features:

* + Admin Dashboard
  + Admin can add/update Candidate
  + Admin can add/update Voters List
  + Admin can change own password
  + Admin can declare result of a student
  + Admin can register new voter and also edit info of the voter

# Voter Features:

* + Voter can view their result

Here, Student can check their results by entering Roll id. Admin can create & manage Classes, subjects. Add & Manage students and Declare Results. This project is done in PHP. It’s easy to operate and understand by users. The design is pretty simple, and user won’t find it difficult to understand, use and navigate.

# SCHEDULING:

|  |  |  |
| --- | --- | --- |
| **Task** | **Start Date** | **End Date** |
| **Business Case Development** | **4th April 2022** | **14th April 2022** |
| **Identifying Stakeholders and Required Modules** | **20th April 2022** | **27th April 2022** |
| **Identifying Requirements** | **30th April 2022** | **6th May 2022** |
| **Setting Cost Estimates And budget** | **10th May 2022** | **17th May 2022** |
| **UML Diagram** | **20th May 2022** | **25th May 2022** |
| **Coding** | **27th May 2022** | **8th June 2022** |
| **Final Revision** | **10th June 2022** | **13th June 2022** |

**PROJECT COST:**

# Development of the project:

|  |  |
| --- | --- |
| **Resource Requirement** | **Cost** |
| **Computer with core i5 10th gen processor,atleast16GbofRAM, running on Windows 10** | **Rs. 55,000/-** |
| **Code** | **Open Source** |
| **Printing** | **Rs. 600/-** |

**Other cost:**

|  |  |
| --- | --- |
| **Employee Salary** | **---** |
| **Maintenance Cost** | **Rs. 2500/ per year** |

# 4.1.2 IDENTIFYING JOB ROLES AND RESPONSIBILITIES:

|  |  |
| --- | --- |
| **Members** | **Roles and Responsibilities** |
| **KANISHK V (RA2011026020093)** | * **TEAM LEADER:** Responsibility of coordination of the team, checking for errors, updating for the current status of the project, guiding the team. * **DEVELOPER:** Responsibility of coding, compiling, anddebuggingof themodules. * **DESIGNER:** Identifying areas for modification in existing programs and developing the modification. |
| **VINEESH K (RA2011026020120)** | * **WEB DEVELOPER:** A web developer is someone who takes a web design, whichhas been created by either a client or a design team and turns into website. * **SOFTWARE ARCHITECT:** Design of UML diagrams and other blueprints. * **MANUAL TESTER**: Responsibility of   testing theproject at asmaller level and reportingthe errors.   * **DESIGNER:** Identifying areas for modification in existing programs and developing the modification. |
| **PRAVEEN RAJ A (RA2011026020124)** | * **TESTER:** Responsibility of testing the project at avast level andvariety of ways. * **MANUAL TESTER:** Responsibility of testing the project at a smaller level and reporting the errors. * **ANALYST:** His job is to gather the requirementsfromtheclient’sendand convert it into technicaldocumentation. |

**4.3.1 PROJECT EFFORT BASED ON RESOURCES:**

COCOMO2 (Constructive Cost Model 2) is an algorithmic cost estimation technique proposed by Boehm, which works in a bottom-up manner.

It is designed to provide some mathematical equations to estimate software projects.

These mathematical equations are based on historical data and use project size in the form of KLOC.

The COCOMO model uses a multivariable size estimation model for effort estimation.

OBJECT POINT = ∑^3 i=1 ∑^3 i=j cij\*wij

|  |  |  |  |
| --- | --- | --- | --- |
|  | **SIMPLE** | **MEDIUM** | **COMPLEX** |
| **SCREENS** | 1 | 3 | 2 |
| **REPORT** | 1 | 3 | 1 |
| **3GL** | 0 | 0 | 1 |

(1\*1+3\*2+2\*3) +(1\*2+3\*5+1\*8) +(1\*10) = 13+25+10 = 48

NOP = Object Point\*(1-% reuse/100) = 48\* (1-0) = 48 EFFORTS = NOP/ PROD = 48/13 = 3.7

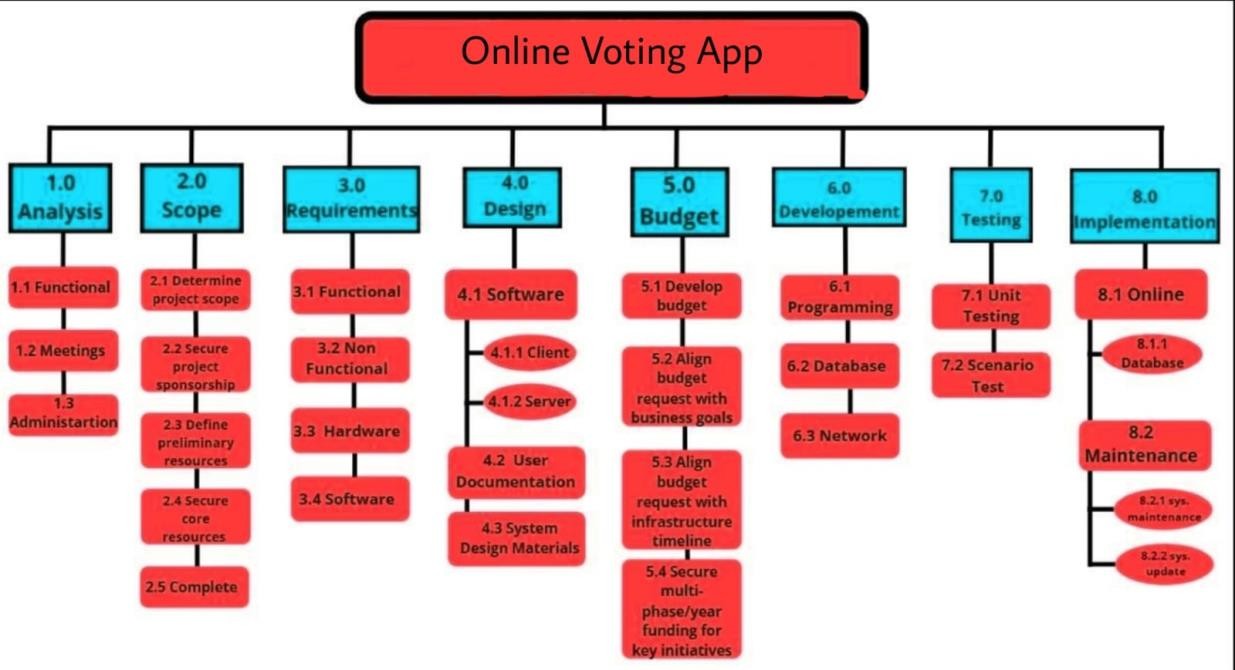
**Prepare the work breakdown structure based on timelines, Risk identification plan**

**EX NO: 5**

**DATE:**

# 5.1.1 WORK BREAKDOWN STRUCTURE:

Breaking work into smaller tasks is a common productivity technique used to make the work more manageable and approachable. For projects, the Work Breakdown Structure (WBS) is the tool that utilizes this technique and is one of the most important project management documents. It singlehandedly integrates scope, cost and schedule baselines ensuring that project plans are in alignment.



# RISK MANAGEMENT:

**DESCRIPTION:**

In the modern world, risk management refers to the practice of identifying potential risks in advance by analysing them and taking precautionary steps to curb the risk.

* + - * Risk management is the identification, evaluation, and prioritization of risks, controlling the probability or impact of unfortunate events.
      * When all risks have been identified, they will then be evaluated to determine their probability of occurrence.
      * Plans will be made to avoid each risk, to track each risk to determine if it is more or less likely to occur, and to plan for those risks should they occur.
      * The quicker the risks can be identified and avoided, the smaller the chances of having to face those particular risks consequence.

# RISKS TO BE HANDLED:

* + - * Computer crash
      * Late delivery of results
      * End users Resist System
      * Low website speed
      * Improper internet connection
      * Maintaining Database.

# MANAGING RISKS:

* + - * Performing periodic maintenance of the server.
      * Using of Captcha and other security protection things to protect from bot attack.
      * The bugs must be removed, and the code must pass as many test cases as possible.

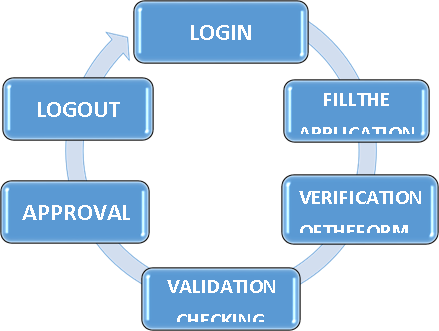
**Design a System Architecture, Use Case Diagram, ER Diagram, DFD Diagram, Class Diagram, Collaboration Diagram**

**EX NO: 6**

**DATE:**

# SYSTEM ARCHITECTURE:

Here we have used the basic software front end design model in order to represent the system architecture of our software model.



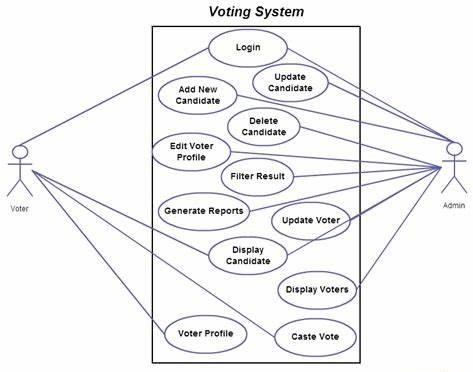
The above is a simple form of system design diagram which uses front end design. This shows a loop of functions that need to be executed when this project is implemented. This is a chain of operations through which this project is implemented.

# MODELING USECASE DIAGRAM AND SCENARIOS:

* + 1. **USE CASE DIAGRAM DESCRIPTION:**
       - There are 2 actors:

1. Admin
2. Voter
   * + - The login and logout, update profile, changing password use cases have relations with admin and both teacher and the student.
       - The Manage users and fill application, manage student, manage class use cases have relations with Admin.
       - The manage subject, manage exam, manage result use cases have relation with the Admin.
       - The Search roll number, view results, download results use cases have relation with student.

# 6.2.2 USE CASE DIAGRAM:



* 1. **MODELING OF ER DIAGRAM:**

# ER DIAGRAM DESCRIPTION:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education, and research.

# USES OF ER DIAGRAM:

* + - * Database design
      * Database troubleshooting
      * Business information systems
      * Business process re-engineering (BPR)
      * Education
      * Research

# COMPONENTS OF ER DIAGRAM:

ER Diagrams are composed of entities, relationships (Cardinality) and attributes. They also depict cardinality, which defines relationships in terms of numbers.

# ENTITY:

A definable thing—such as a person, object, concept or event— that can have data stored in it.

# ATTRIBUTES:

A property or characteristic of an entity.

# KEYS:

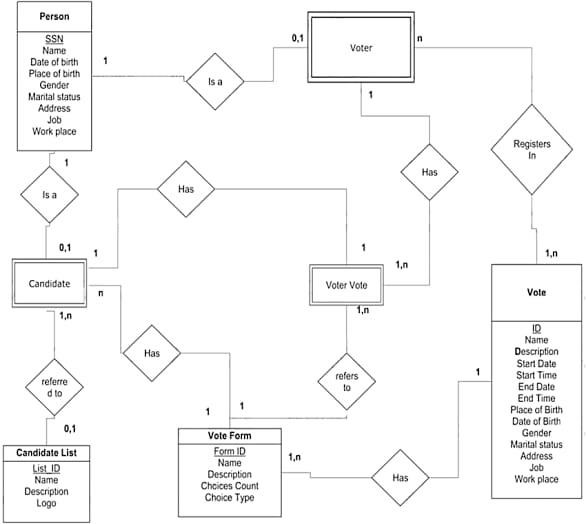
* + PRIMARY KEY(PK): It is unique, cannot be repeated and never null.
  + FOREIGN KEY(FK): It is not unique and can be repeated.

# CARDINALITY:

Defines the numerical attributes of the relationship between two entities.

* + One to one
  + Many to one
  + One and only
  + Zero to one
  + One or many
  + Zero or many

# ER DIAGRAM:



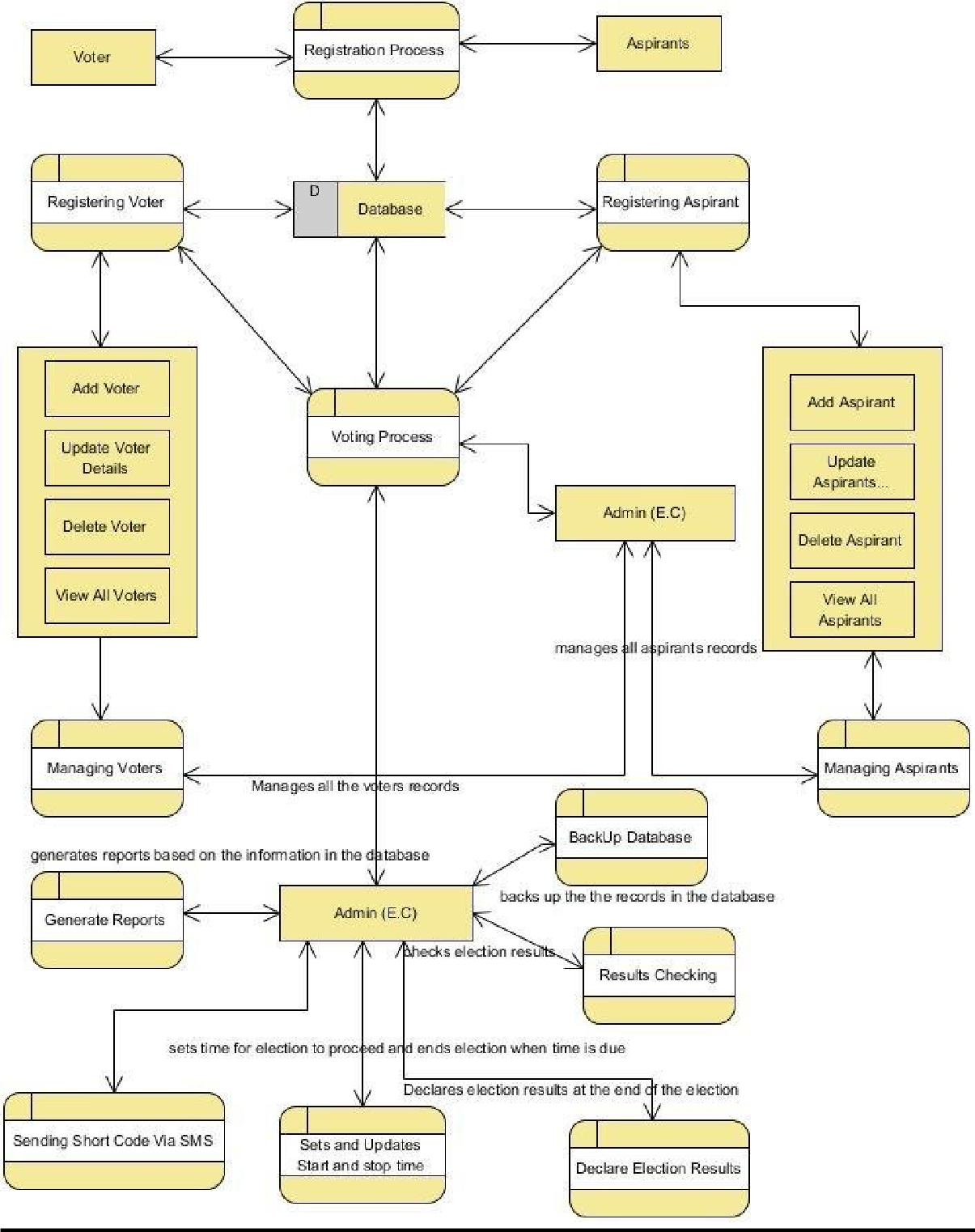
* 1. **MODELING OF DATA FLOW DIAGRAM:**

# DATA FLOW DIAGRAM DESCRIPTION:

Result Management System Data flow diagram is often used as a preliminary step to create an overview of the Result Management without going into great detail, which can later be elaborated it normally consists of overall application dataflow and processes of the Result Management process. It contains all of the user flow and their entities such all the flow of Student, Exam, Class, Subject, Result, Teacher, Semester. All of the below diagrams has been used for the visualization of data processing and structured design of the Result Management process and working flow.

# Main entities and output of First Level DFD (1st Level DFD):

* + - * Processing Student records and generate report of all Student
      * Processing login records and generate report of login credentials
      * Processing Class records and generate report of all Class
      * Processing Subject records and generate report of all Subject
      * Processing Result records and generate report of all Result
      * Processing Teacher records and generate report of all Teacher
      * Processing Semester, records and generate report of all Semester



# MODELING OF CLASS DIAGRAM:

* + 1. **CLASS DIAGRAM DESCRIPTION:**

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structures of a particular system by modelling its classes, attributes, operations and relationships b/w objects.

# COMPONENTS OF CLASS DIAGRAM:

The standard class diagram is composed of three sections:

# UPPER SECTION:

Contains the name of the class. This section is always required, whether you are talking about the classifier or an object.

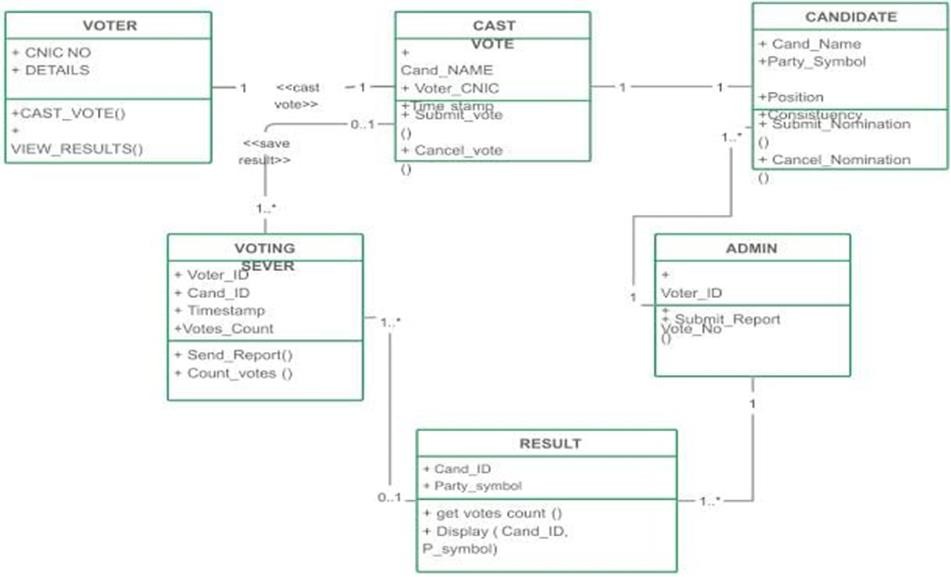
# MIDDLE SECTION:

Contains the attributes of the class. Use this section to describe the qualities of the class. This is only required when describing a specific instance of a class.

# BOTTOM SECTION:

Includes class operations (methods). Displayed in list format, each operation takes up its own line. The operations describe how a class interacts with data.

# CLASS DIAGRAM:



* 1. **MODELING OF COLLABORATION DIAGRAM:**
     1. **COLLABORATION DIAGRAM DESCRIPTION:** Communication diagrams, formerly known as collaboration diagrams, are almost identical to sequence diagrams in UML, but they focus more on the relationships of objects—how they associate and connect through messages in a sequence rather than interactions.

# COMPONENTS OF COMMUNICATION DIAGRAM:

1. **OBJECTS:**

Objects can be classed as either a supplier or a client. Suppliers call the function that supplies the message. Clients send the message to the supplier, who receives it. It is represented by rounded rectangle.

# ACTORS:

Stick figure represents the actor. It is the instances that invokes the interaction. Each actor has a specific name and a role.

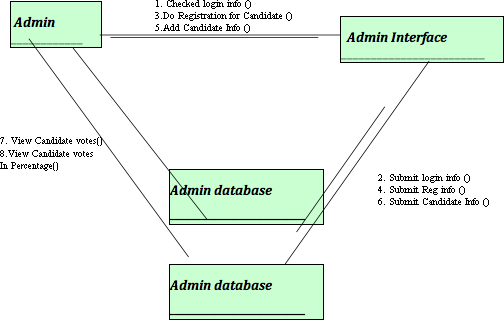
# LINKS:

A straight line connecting two objects indicates a relationship between them. The two objects are able to send messages to each other.

# MESSAGES:

Typically, messages will have a number and description next to them. The number determines the order in which messages should be read.

# COLLABORATION DIAGRAM:



**State and Sequence Diagram, Deployment Diagram, Sample Frontend Design**

**EX NO: 7**

**DATE:**

* 1. **STATECHART DIAGRAM:**

# STATECHART DIAGRAM DESCRIPTION:

State diagram describes the behaviour of a single object in response to a series of events in a system. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.

# COMPONENTS ARE:

* + - * **Initial State:**

A filled circle followed by an arrow represents the student’s login (object’s) initial state.

# States:

States in state chart diagram represent situations during the life of an Object. You can easily illustrate a state in Smart Draw by using a rectangle with rounded corners.

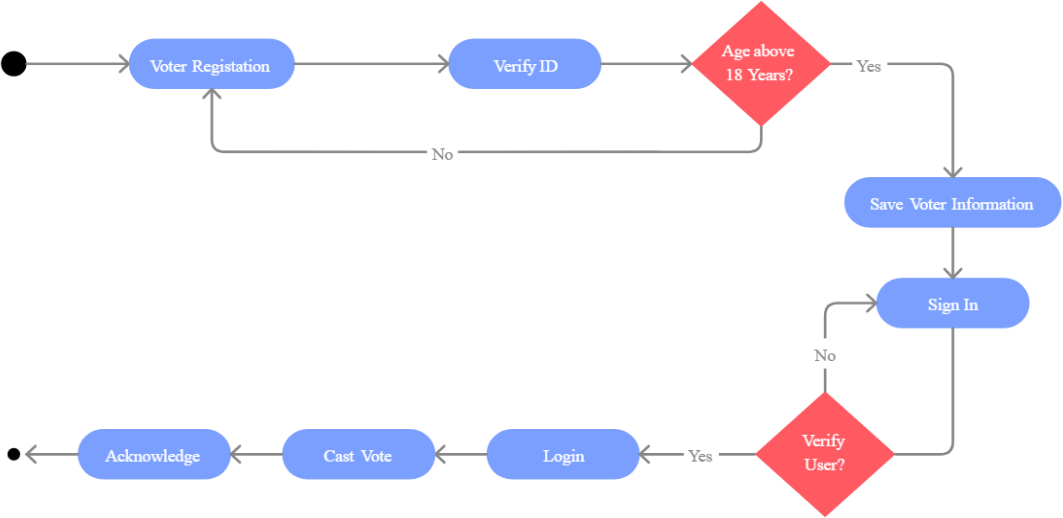
# Transition:

A solid arrow represents the path between different states of an object of Exam result management system.

# Final State:

An arrow pointing to a filled circle nested inside another circle represents the (object's) result.

# STATECHART DIAGRAM:



* 1. **SEQUENCE DIAGRAM:**

# SEQUENCE DIAGRAM DESCRIPTION:

Sequence diagram are a popular dynamic modelling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

# COMPONENTS IN SEQUENCE DIAGRAM:

1. **ACTOR:**

Stick figure represents the actor. Shows entities that interact the external objects of the system.

# OBJECTS:

Rectangular boxes represent the object, demonstrates how an object will behave in the context of the system.

# ACTIVATION BOXES:

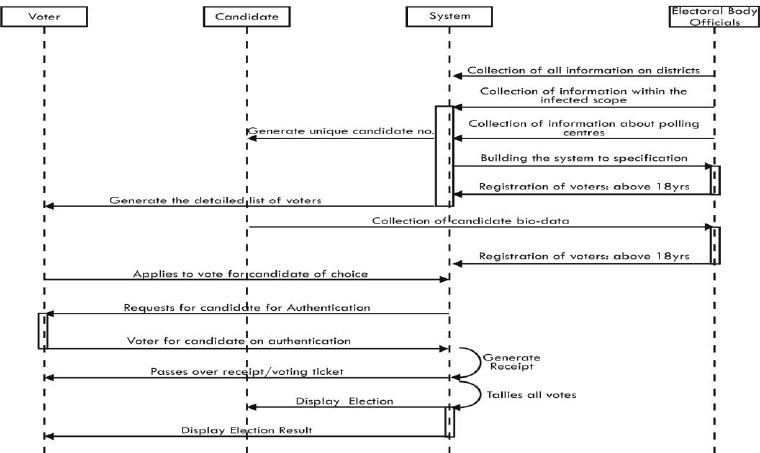
Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.

# MESSAGE SYMBOLS:

We use the following arrows and message symbols to show how information is transmitted between objects. These symbols may reflect the start and execution of an operation or the sending and reception of a signal.

* + **SYNCHRONOUS MESSAGE:** Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply.
  + **ASYNCHRONOUS MESSAGE:** Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram.
  + **REPLY MESSAGE:** Represented by a dashed line with a lined arrowhead, these messages are replies to calls.
  + **DELETE MESSAGE:** Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object.

# SEQUENCE DIAGRAM:



* 1. **DEPLOYMENT DIAGRAM:**

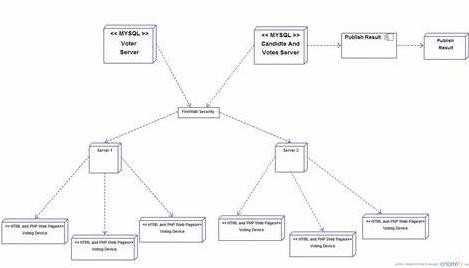
# DEPLOYMENT DIAGRAM DESCRIPTION:

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

# DEPLOYMENT DIAGRAM:



**7.4. SAMPLE FRONTEND DESIGN:**

The figures 7.4.1 show the page for voters to view available polls. And possesses a button that permits the voters to view the available polls.

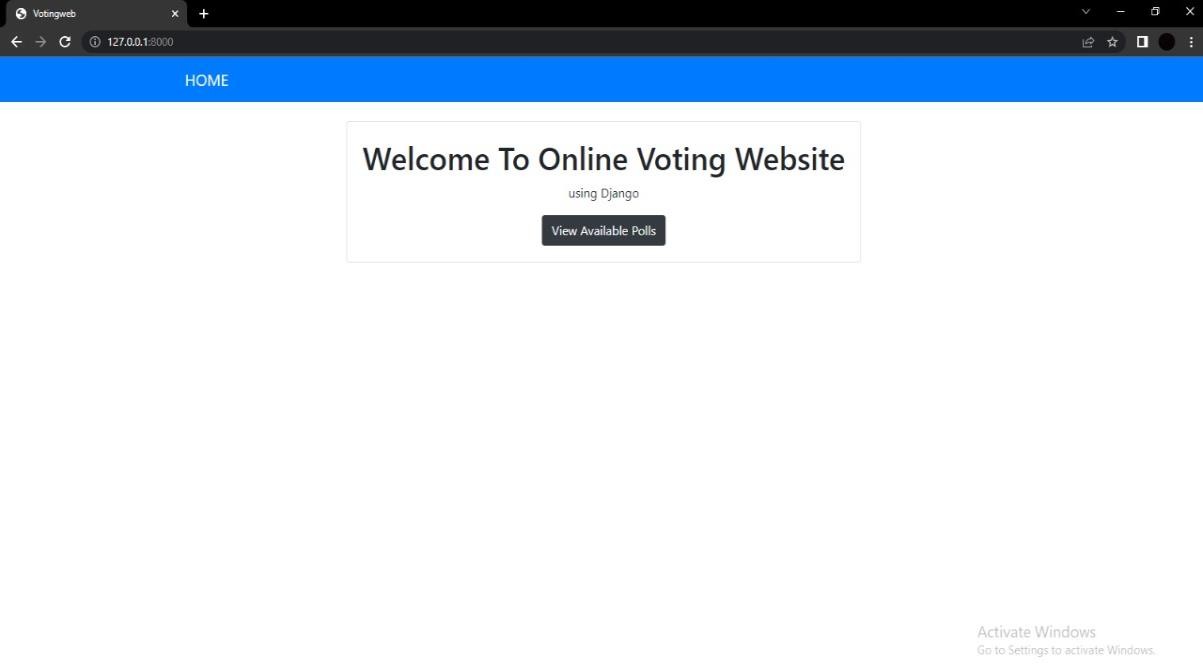


Fig 7.4.1

|  |  |
| --- | --- |
| **EX NO: 8**    **DATE:** | **Module Description, Module Implementation (phase 1) Using Agile** |

**8.1. MODULE DESCRIPTION:**

Following are the main Module of this Online Student Result Management system.

Login Module - At Admin Side

Classes Management Module - At Admin Side

Student Management Module - AT Admin Side

Exam Management Module - At Admin Side

User Management Module - At Admin Side  
Profile Management Module - At Admin Side

Result Management Module - At Admin & Student Side

**Features of Student Result Management System:**

Voters can get result by Search Result by entering their Voter ID number.

This is Multiuser System, one user can create number of Sub user.

Admin can see analytics data of total number of votess publish, total vote data, total vote data and total candidate data.

Admin can Add Edit and Delete Candidate data.

Admin can enable and disable the status of Candidate under this System.

Admin can Add, Edit and Delete Voters data.

Admin user can Enable and Disable the status of Voter.

Admin user can Add, Edit and Delete Voting data.

Admin user can create new user and he or she can also edit user data also.

Admin can disable the login of sub user and he or she can also enable the login of sub user also.

Admin and Sub User can edit their profile details.

Sub user can add edit and delete result data and Admin can see all user result data in single page.

Sub user and Admin can disable

**8.2. MODULE IMPLEMENTATION USING AGILE:**

The fig.8.2.1 shows the dashboard to check candidate and user details by the admin. any voter result for publish on internet and they can also enable it also.

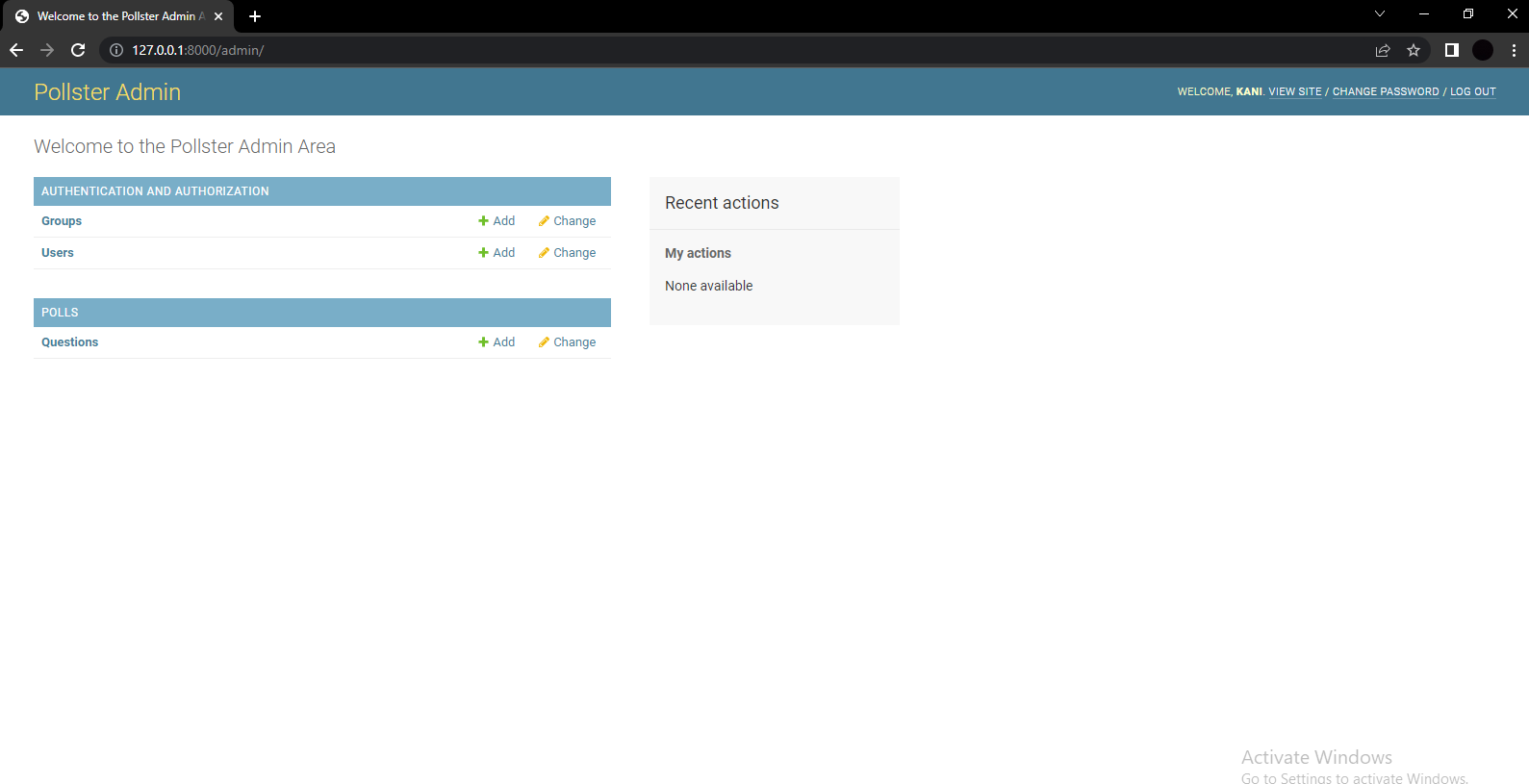


Fig 8.2.1

**Module Implementation, Scrum Master to Induce New requirements in Agile Development**

**EX NO: 9 DATE:**

The Fig.9.1 shows the landing page for our voting website. Click on the button ‘View Available Pools’. You can find the Fig.9.2 appear on screen after clicking the button. The pool available in the page is decided by the admin.

Home page URL: http://127.0.0.1:8000 Available poll: http://127.0.0.1:8000/polls

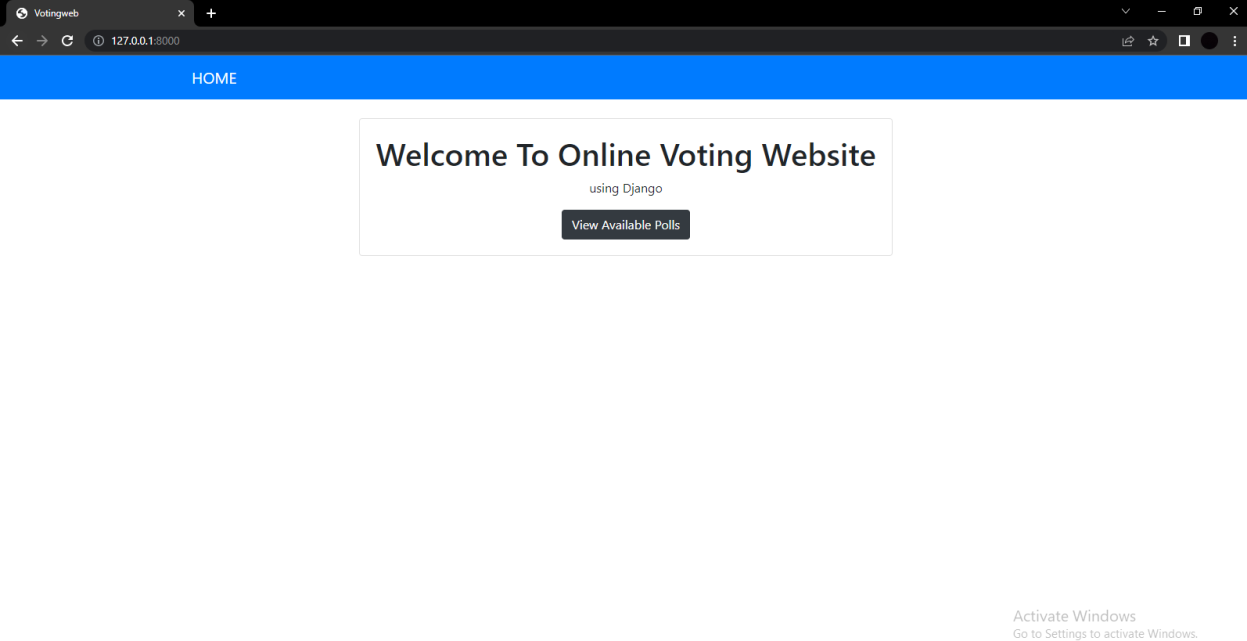


Fig:9.1 Landing page

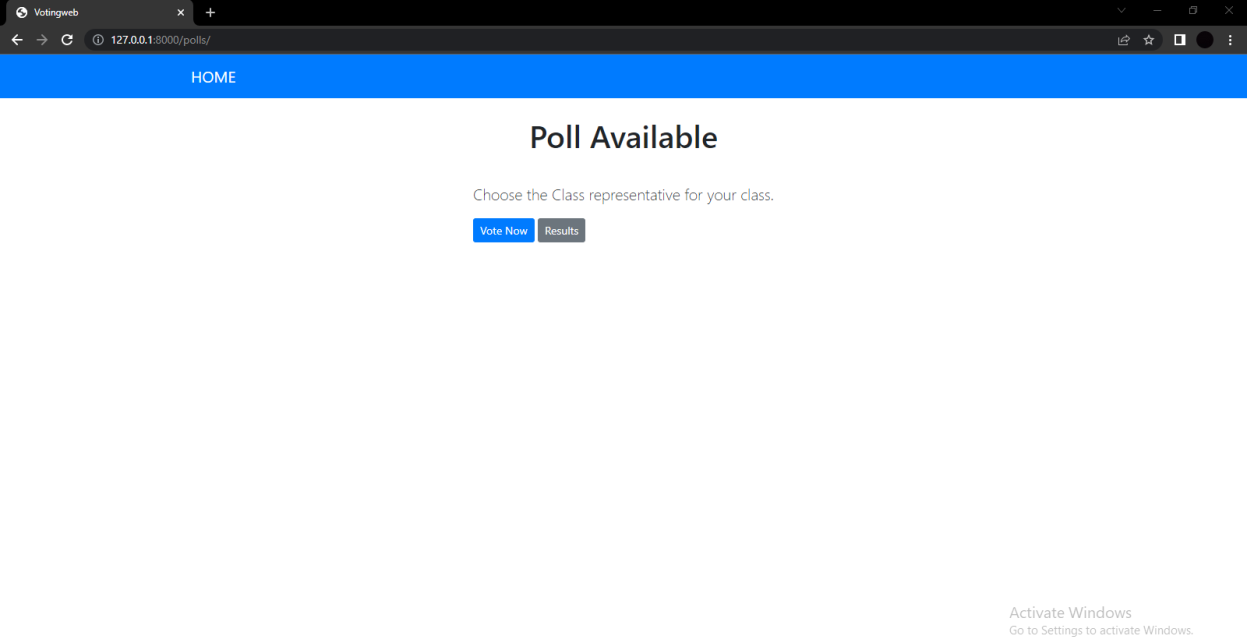


Fig:9.2 Page 2 after clicking the button

**Module Implementation (Phase 2), Scrum Master to Induce New Issues in Agile Development**

**EX NO: 10 DATE:**

In Fig:9.1 and Fig:9.2 we saw what the voters see. Now we will by going to admin page by the URL: http://127.0.0.1:8000/admin/ .The admin page looks like in the Fig:10.1.

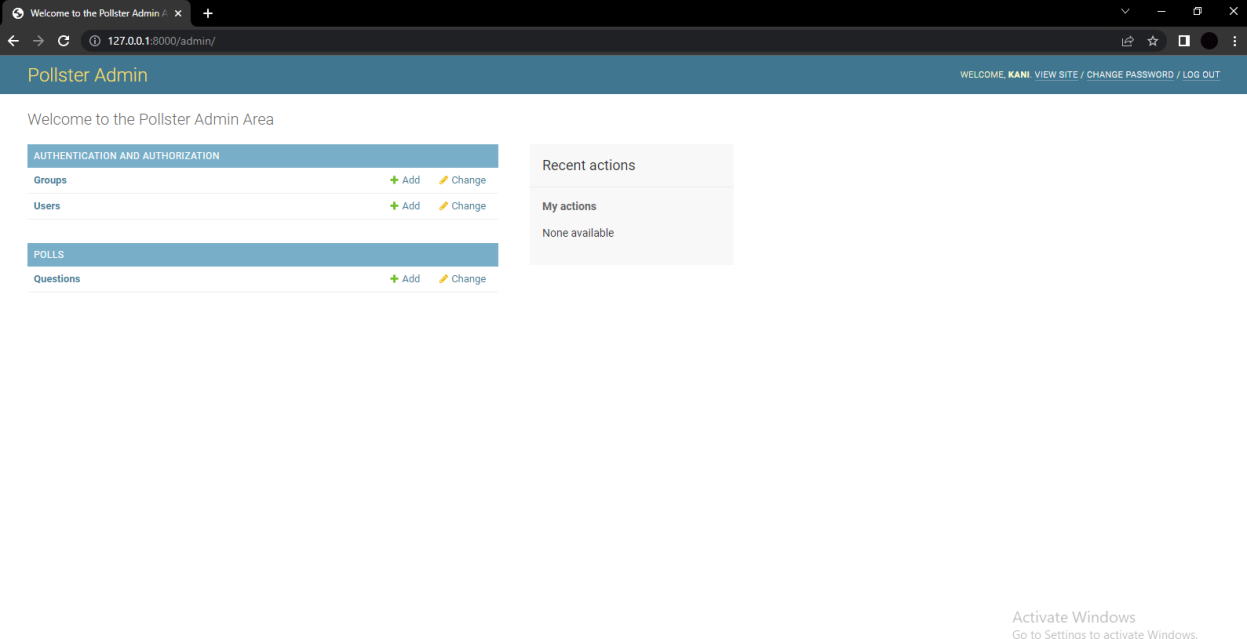


Fig:10.1

We go to POLLS and click on ‘Questions’. The Fig:10.2 depicts the page after clicking ‘Questions’. Now we create the polling question and the options to choose.

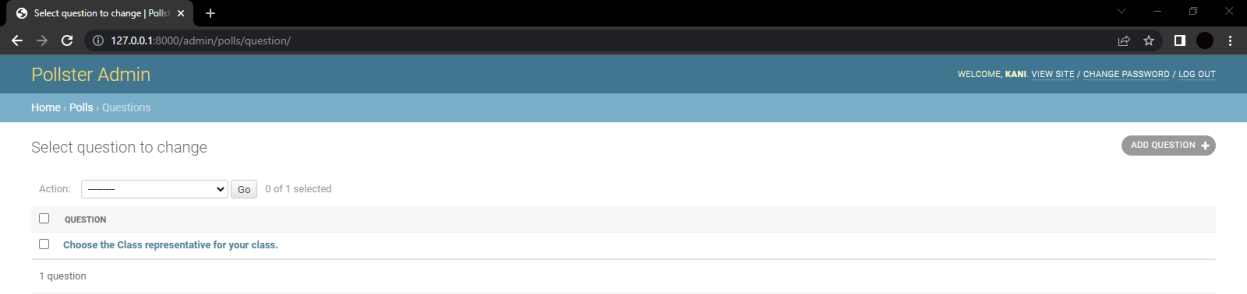


Fig:10.2

One the question tag you can fill the respected question and mention the choices. Enter the date and time. Click on the SAVE button after completing.

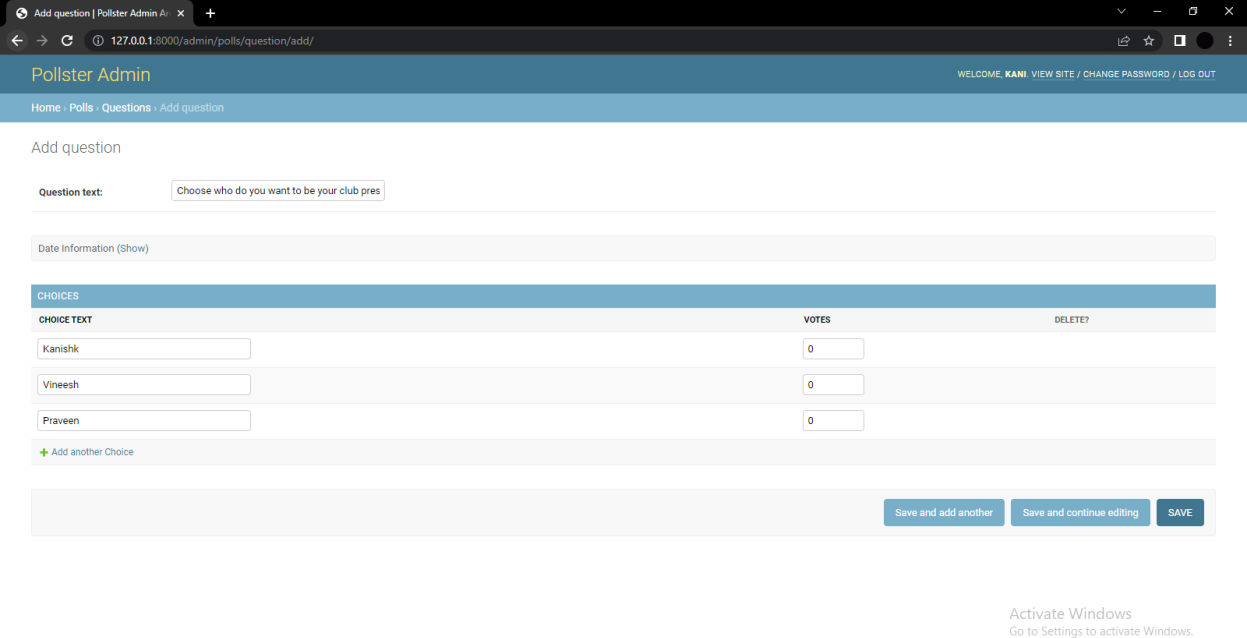


Fig:10.3



Fig:10.4

Fig:10.4 after clicking the SAVE button. You can find your question being there. Now go back to the URL: http://127.0.0.1:8000/polls/ , You will be able to find your question being displayed(Fig:10.5).

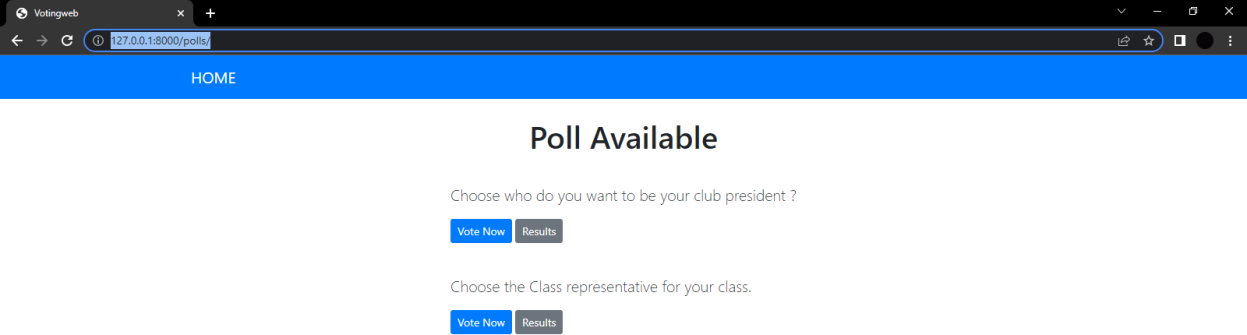


Fig:10.5

**Module Implementation (Phase 3) Scrum Master to Induce New requirements in Agile Development, Scrum Master to Induce New Issues in Agile Development, Code development**

**EX NO: 11 DATE:**

# MODULE IMPLEMENTATION (PHASE 3):

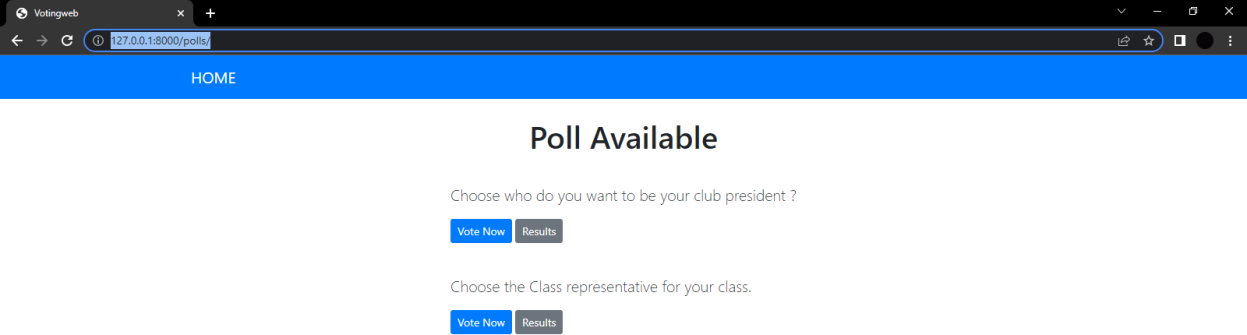


Fig:11.1 Poll available

In Fig:11.1 we are able to find two questions and two buttons i.e., Vote now and Results. Click on the button Vote Now.

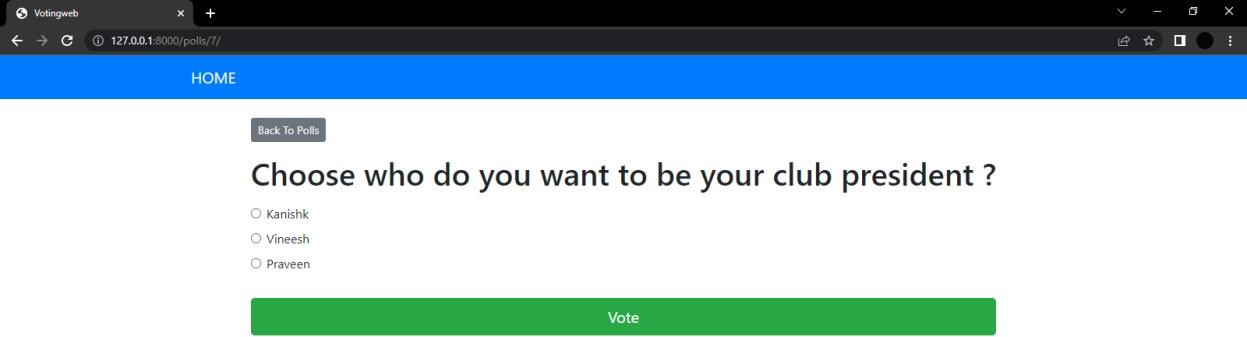


Fig:11.2 Casting vote

After clicking the button it gets directed to the respective webpage.Now choose who do you want to cast your vote.

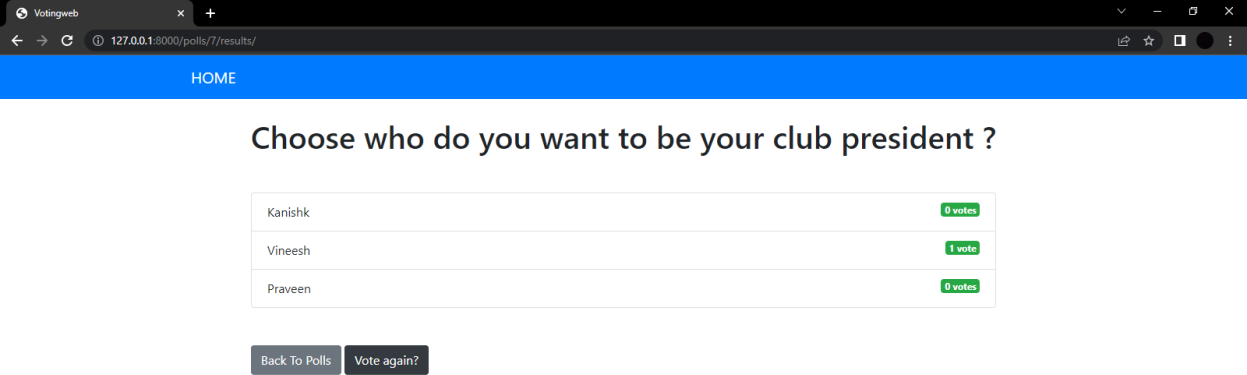


Fig:11.3 Results

In Fig:11.3 you are able to find the immediate result. Now you can cast your vote again or go back to the polls. Click ‘Go to Polls’.

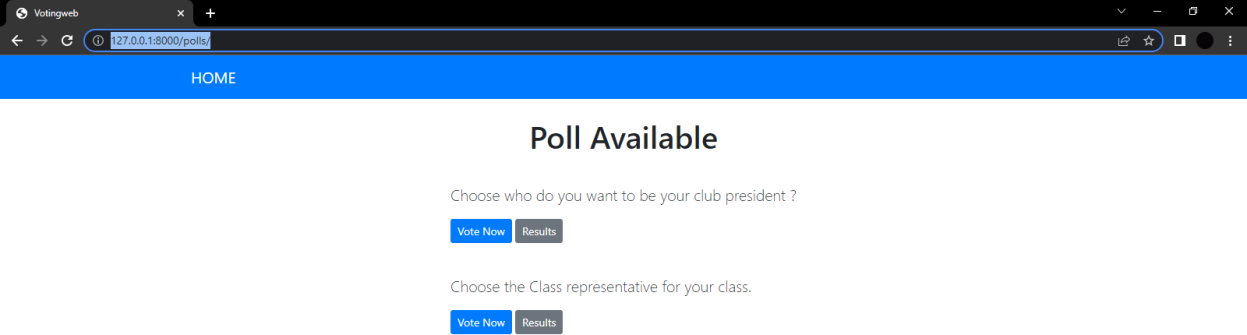


Fig:11.4 Poll available

You will be directed to this page again and now click on ‘results’ to view the result page.

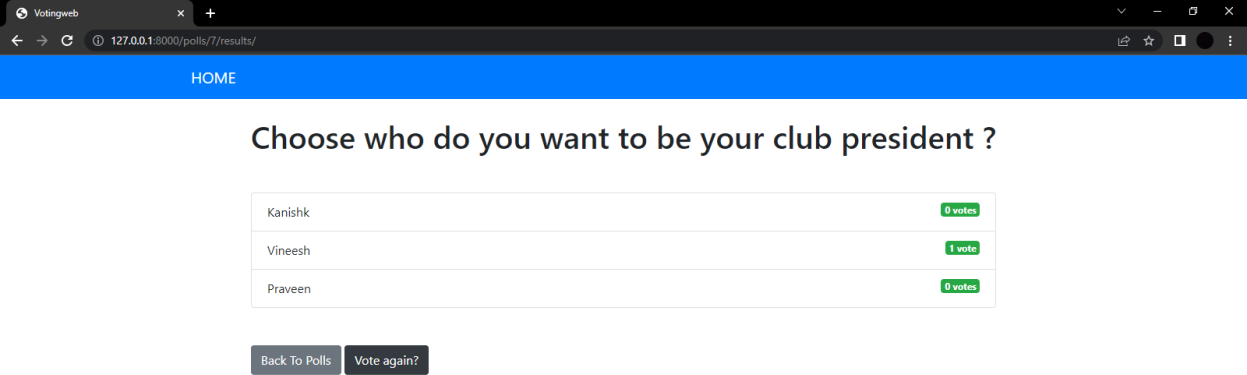
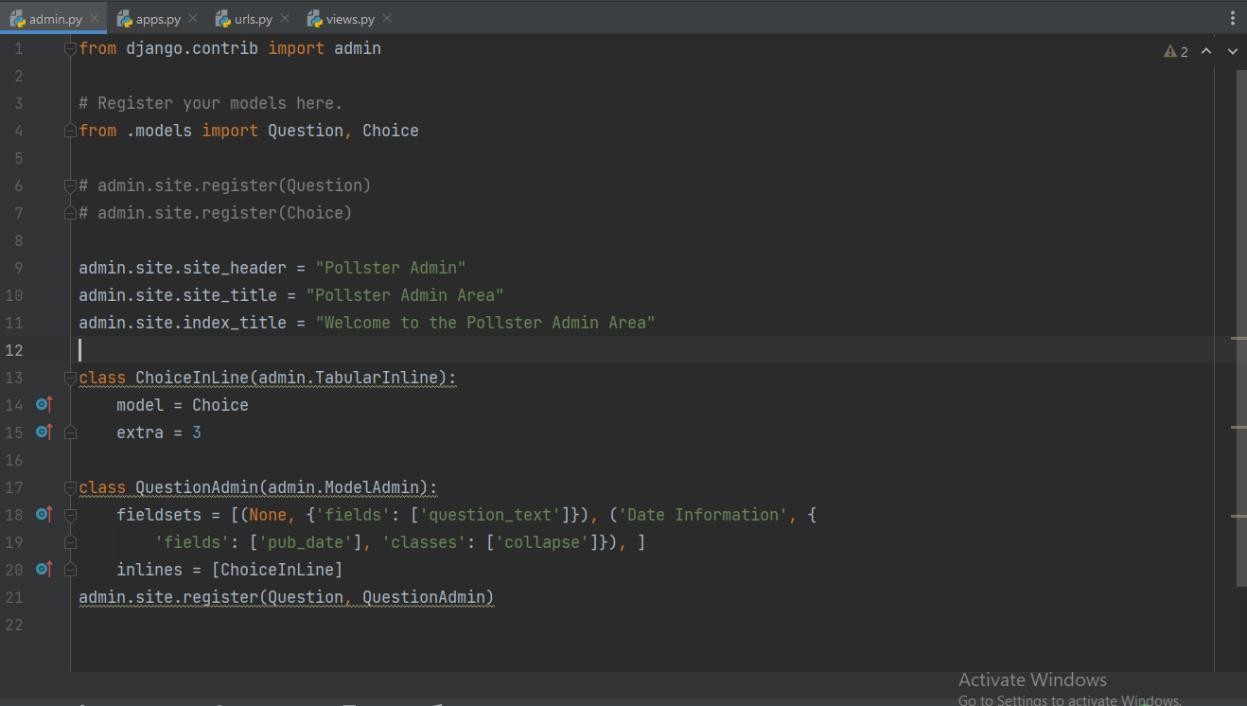
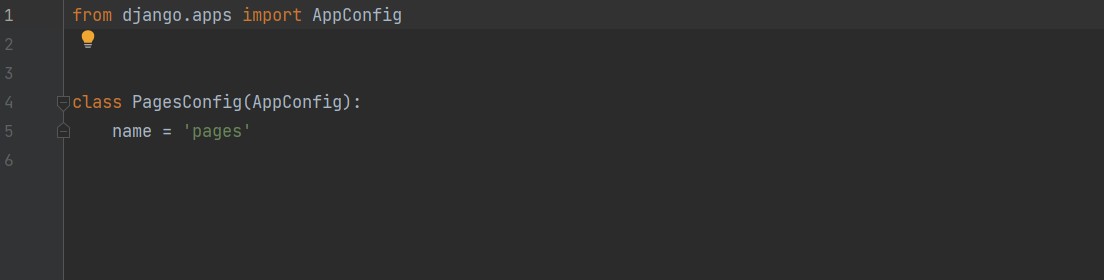


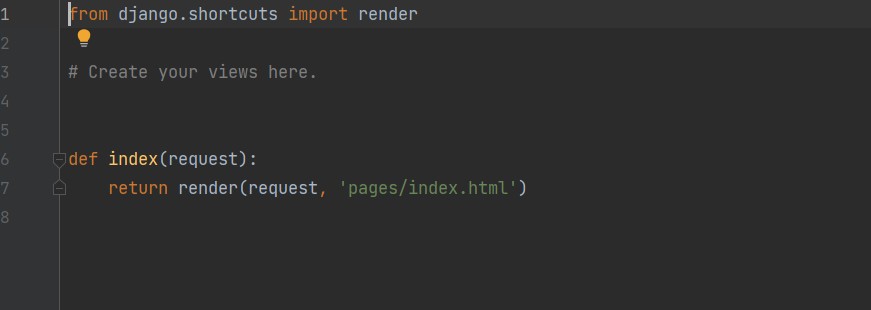
Fig:11.5 Results

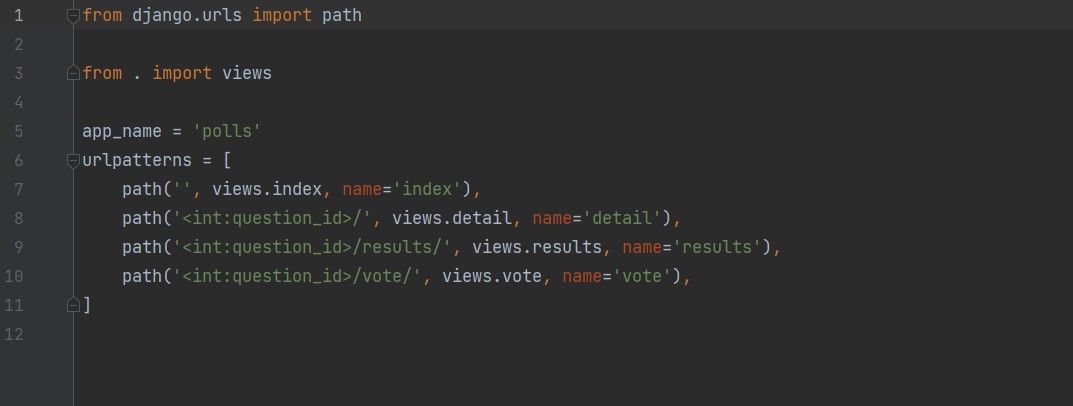
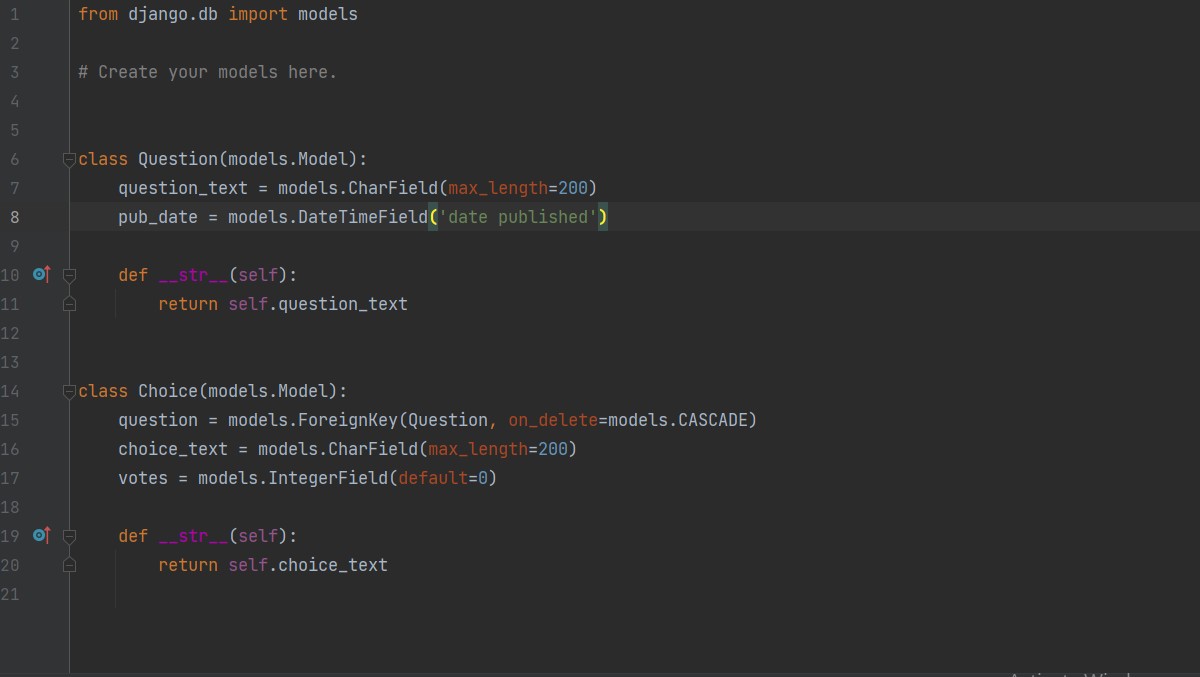
# CODE DEVELOPMENT:

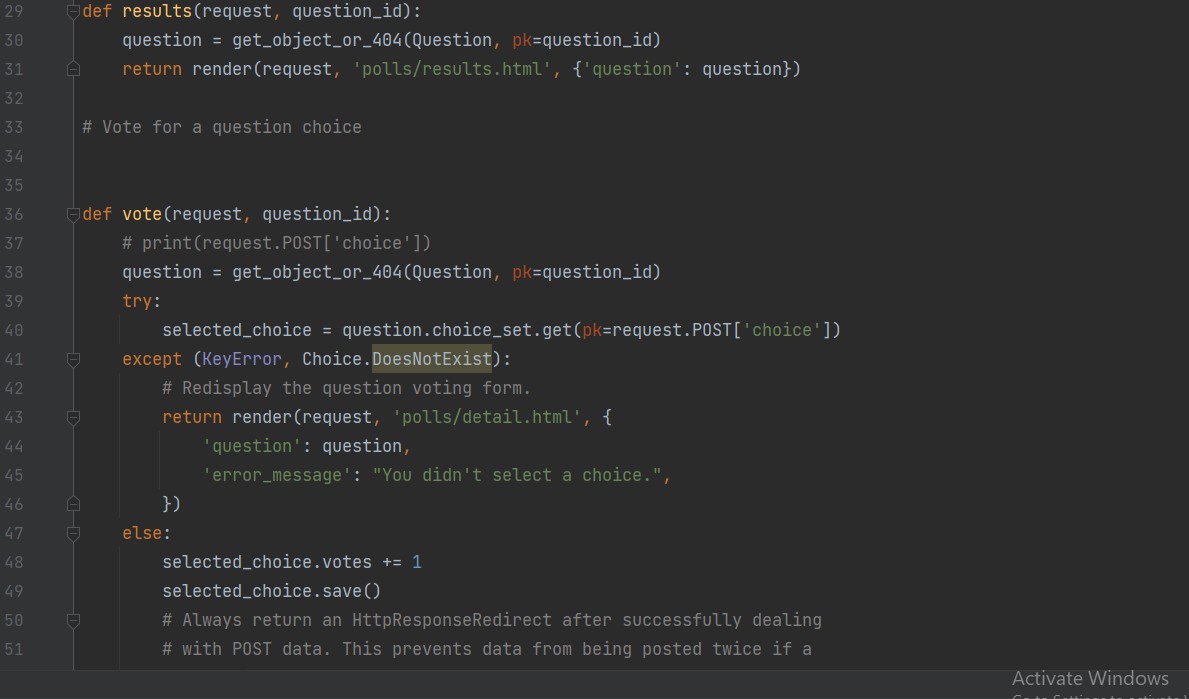
We have implemented this project using Python Django web framework, HTML and CSS. Also, use case diagram is presented to show the different categories of the system users and the various functionalities associated the different system user.

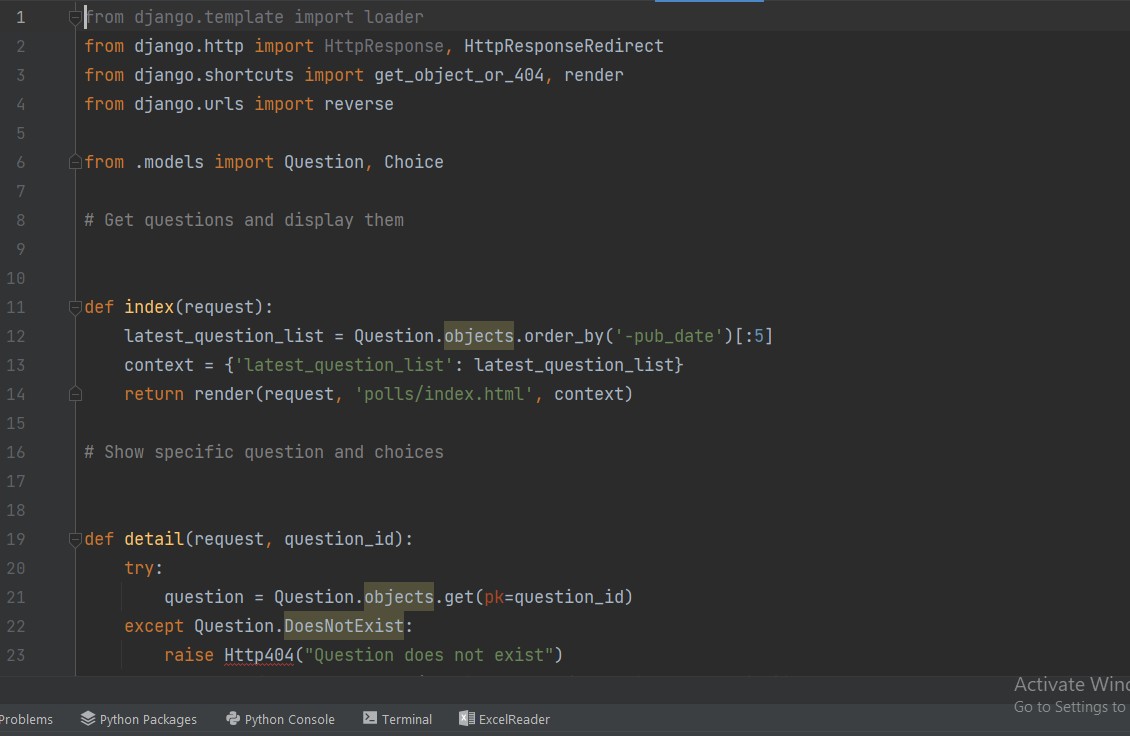


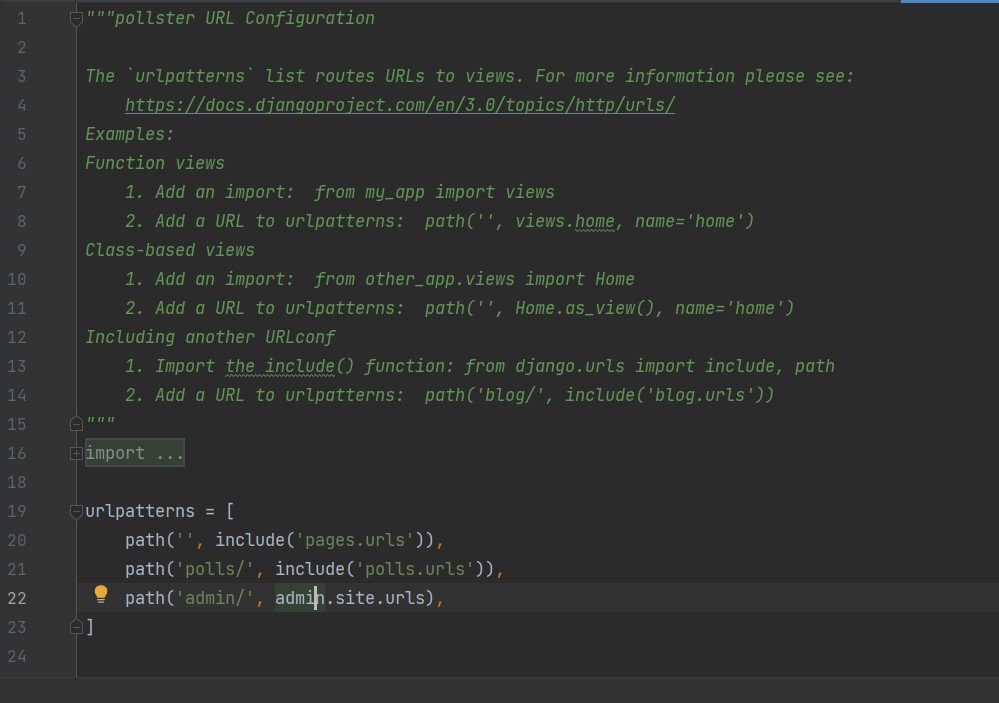












**Master Test Plan, Test Case Design (Phase 1)**

**EX NO: 12 DATE:**

# TEST MASTERPLAN:

|  |  |
| --- | --- |
| **TESTING OBJECTIVE** | **TESTING OBJECTIVE FOCUSING ON PERFORMANCE ISSUES** |
| Test Items | Login system, Registration system,  Voting/result system |
| Features to be tested | Login verification, Registration feature, Voting/result feature. |
| Features not to be tested | Database Connectivity |
| Approach | Method – Manual Testing |
| Required Hardware/Software | A PC with 8 GB RAM, Internet  Connectivity |
| Risks | Instability of the product |
| Testers & Schedule | Tester: PRAVEEN RAJ, Scheduling Information: 31st May 2022, 3:00 PM |
| Estimate | Rs400/- (Excluding Tax and other  charges) |

* 1. **TEST CASE DESIGN:**

# Testing:

* The process of executing a system with the intent of finding an error.
* Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
* Quality is defined as justification of the requirements
* Defect is nothing but deviation from the requirements.
* Testing --- The presence of bugs
* Testing can demonstrate the presence of bugs, but not their absence
* Debugging and Testing are not the same thing!
* Testing is a systematic attempt to break a program or the AUT
* Debugging is the art or method of uncovering why the script /program did not execute properly.

# Testing Methodologies:

* Black box Testing: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.
* White box Testing: is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually, The Developers are involved in white box testing.
* Gray Box Testing: is the process in which the combination of black box and white box techniques are used.

# Positive Test Case:

* The positive flow of the functionality must be considered
* Valid inputs must be used for testing.
* Must have the positive perception to verify whether the requirements are justified.

# Negative Test Case:

* Must have negative perception.
* Invalid inputs must be used for test.

**Manual Testing**

**EX NO: 13 DATE:**

# 13.1 MANUAL TESTING:

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST AREA** | **INPUT** | **TEST DESCRIPTION** | **OUTPUT/RESULT** |
| Login Module | Login | Permits the user to enter the  application | Tested |
| Poll available | Click the available poll | Allows the user to enter and cast his  vote | Tested |
| Voting result | Total votes | Final votes | Tested |

**User Manual, Analysis of Costing, Effort and Resources**

**EX NO: 14 DATE:**

* 1. **USER MANUAL :**

# Introduction:

The online voting website allows users to create polls and able to cast their vote. In the end we are able to find the result of the winner immediately.

* + 1. Getting Started:

Download and install file and check whether the browser are updated and supported to run the file.

* + 1. Quick Start: User:
       1. Click on the website and click on the button ‘View available polls’, by clicking it delivers to the ‘Available polls page’.
       2. Click on which one poll you want to participate and click on the button
       3. ‘Vote now’ to directs to webpage where you can cast your vote.

Choose your respective candidate and click ‘Vote’. You have successfully caste your vote.

* + - 1. Click on the ‘Result’ button to check the results

# 14.1.3. Troubleshooting:

File mismatch error or improper website.

# ANALYSIS OF COSTING, EFFORT AND RESOURCES: DEVELOPMENT OF PROJECT:

|  |  |
| --- | --- |
| **RESOURCE REQUIREMENTS** | **COST** |
| Computer with core i3, 6th Gen , at least 4gbRAM running on windows 10 or greater | Rs.50000/- |
| Code | Open source |
| Printing | Rs.1000 |

**SERVER-END:**

|  |  |
| --- | --- |
| **RESOURCE REQUIREMENTS** | **COST** |
| Http web services | Rs.1200/- |
| UPS | Rs.2500/- |

# OTHER COSTS:

|  |  |
| --- | --- |
| Employee salary | - |
| Maintenance cost | Rs.800/- |