**JAMIA MILLIA ISLAMIA**



**SOFTWARE PROJECT MANAGEMENT**

**Submitted by:-**

**SUMIT NAGPAL (20186254)**

**ACKNOWLEDGEMENT**

The simple act of saying ‘thank you’ is a demonstration of gratitude in response to an experience that was meaningful to us as students. We extend our thanks to our teacher **Prof. Mohammad Nazir** for guiding us throughout the project “**Online Election**” in every possible way with his invaluable advice, useful suggestions, and relevant ideas that facilitated the completion of our project. We feel honored and privileged to work under him.

**INTRODUCTION**

Online Election System would have Candidate registration, User ID and password for candidate and Voters. Admin Login which will be handled by Election Commission . Voters will get Unique ID and Password, Using which they can vote for a Candidate only once per Election.

The project is beneficial for Election Commission, Voters as the can get to know the candidate background and choose wisely.

To enable as many voters as possible to participate .

Online Voting can assist people who may not be able to attend a polling station in person.

For example: Because of physical disability, Absence from the locality or some other reason.

Here, we propose an online system that automatically allows user to register himself/herself, Vote for the Election, See Results by selecting an appropriate Election.

The main features of our system are:-

* Admin Login
* Voters Login
* Voters can view Candidate’s data
* Admin dashboard has overall functional rights
* Appropriate data processing and handling
* ID and Password for Candidate and Voters.
* Result Calculation module
* Election Creation module
* Voting conduction module

This system is easy to understand,easy to use and offers the simplicity of fast point and click service to the system.

**PROCESS MODEL**

**The Waterfall Model**

The waterfall model is a model which was developed for software development; that is to create software. It is called as such because the model develops systematically from one phase to other in a downward fashion, like a waterfall.

The most probable phases through which it progresses downwards are

 Definition Study/Analysis

 Basic Design

 Technical Design/Detailed Design

 Construction

 Testing

 Integration

 Management

 Maintenance.



Before the advent of this method, the software development in the computer companies suffered from a haphazard integrated software network like cluttered knitting. However with this method they hoped to bring clarity in their projects.

**Phases:**

As said earlier the waterfall model has been structured on multiple phases especially to help out the software construction companies to develop an organized system of construction. By following this method, the project will be divided into many stages thus easing out the whole process. For example you start with Phase I and according to this model, one only progresses to the next Phase once the previous one has been completed. This way one moves progressively to the final stage and once that point is reached, you cannot turn back; similar to the water in a waterfall.

**Brief Description of the Phases of Waterfall Model**

**Definition Study / Analysis**: During this phase research is being conducted which includes brainstorming about the software, what it is going to be and what purpose is it going to fulfill.

**Basic Design**: If the first phase gets successfully completed and a well thought out plan for the software development has been laid then the next step involves formulating the basic design of the software on paper.

**Technical Design / Detail Design:** After the basic design gets approved, then a more elaborated technical design can be planned. Here the functions of each of the part are decided and the engineering units are placed for example modules, programs etc.

**Construction / Implementation**: In this phase the source code of the programs is written.

**Testing**: At this phase, the whole design and its construction is put under a test to check its functionality. If there are any errors then they will surface at this point of the process

**Integration:** in the phase of Integration, the company puts it in use after the system has been successfully tested.

**Management and Maintenance:** Maintenance and management is needed to ensure that the system will continue to perform as desired.

Through the above mentioned steps it is clearly shown that the Waterfall model was meant to function in a systematic way that takes the production of the software from the basic step going downwards towards detailing just like a Waterfall which begins at the top of the cliff and goes downwards but not backwards.

**Advantages of the Waterfall Model**

 This model is simple and easy to understand and use.

 It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.

 In this model phases are processed and completed one at a time. Phases do not overlap.

 Waterfall model works well for smaller projects where requirements are very well understood.

**Disadvantages of the Waterfall Model**

 Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.

 No working software is produced until late during the life cycle.

 Not a good model for complex and object-oriented projects.

 Poor model for long and ongoing projects.

 Not suitable for the projects where requirements are at a moderate to high risk of changing.

**When to use the waterfall model:**

 This model is used only when the requirements are very well known, clear and fixed.

 Product definition is stable.

 Technology is understood.

 There are no ambiguous requirements

 Ample resources with required expertise are available freely

 The project is short.

**Software requirement specification**

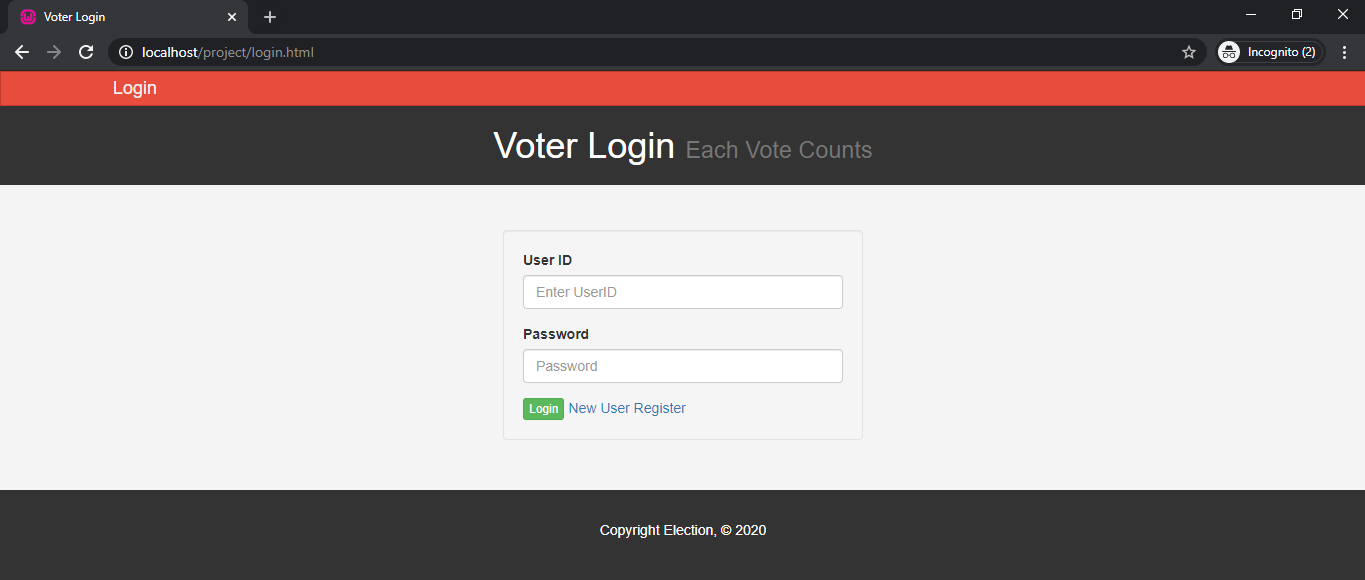
A software requirement specification (SRS) is a desciption of a software system to be developed . It lays out functional and non-functional requirements , and may include a set of use cases that describe user interactions that the software must provide.

**Purpose**

Requirement analysis is done in order to understand the problem , which is to be solved. That is very important activity for the development of database system. The requirements and the collection analysis phase produce both data requirements and functional requirements. The data requirements should be specified in as detailed and complete form as possible.

In parallel with specifying the data requirements, it is useful to specify the known functional requirements of the application. These consists of user-defined operations that will be applied to the database(retrievals and updates). The functional requirements are used as a source of application software design.

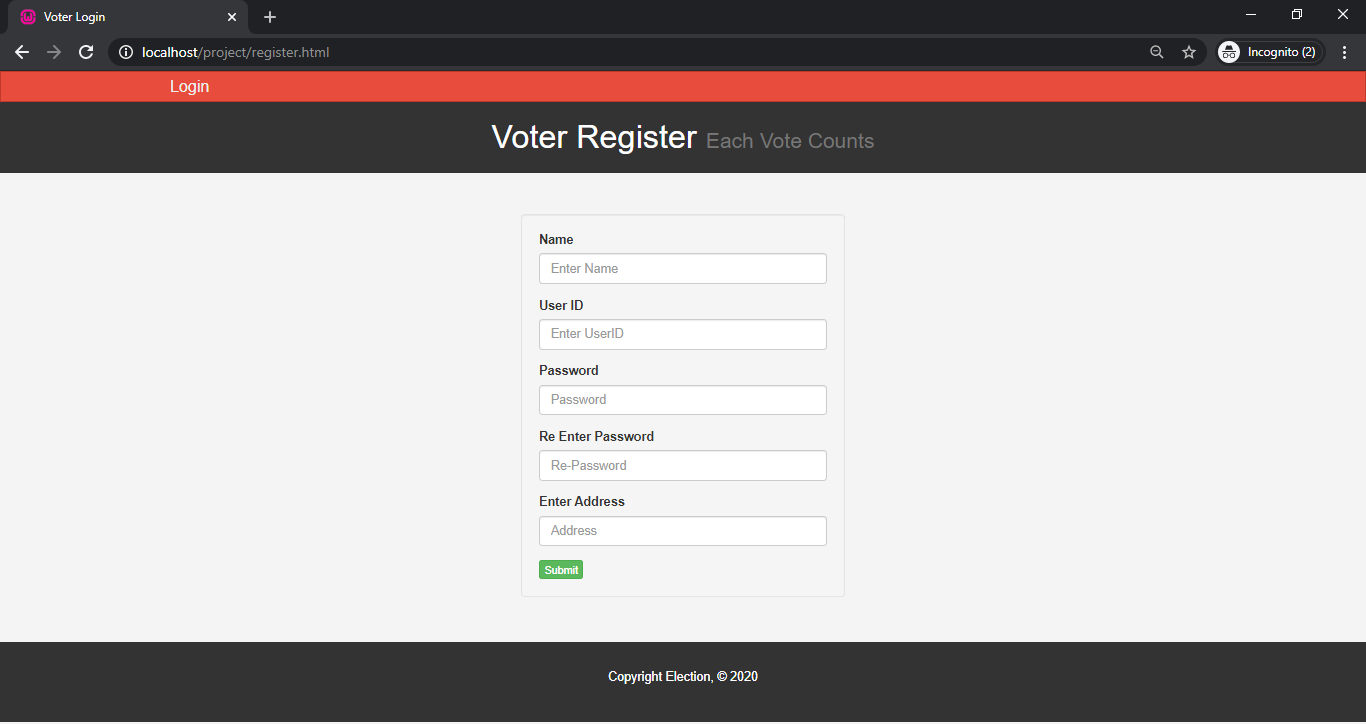
The main modules in this project include :- ADMIN and VOTER .

****

**Voter MODULE :**

**1. Voter registration**

* Voter Registration with name, a User ID (for logging in later), password, reenter password, address.

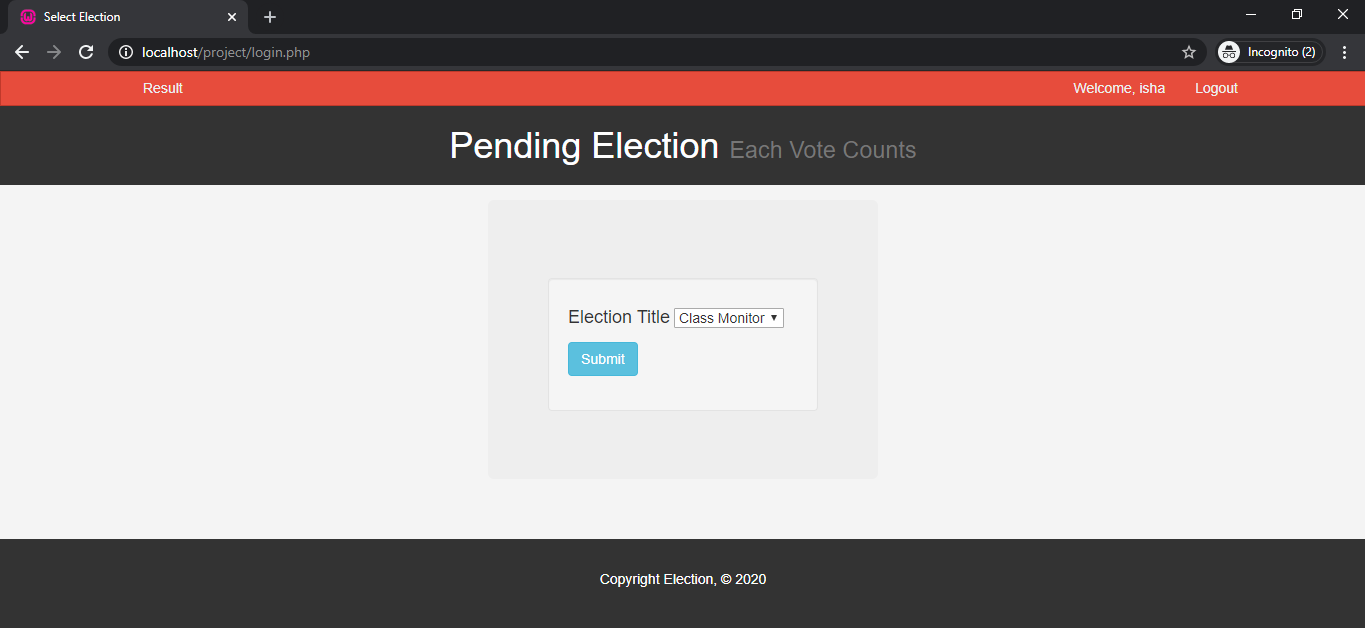


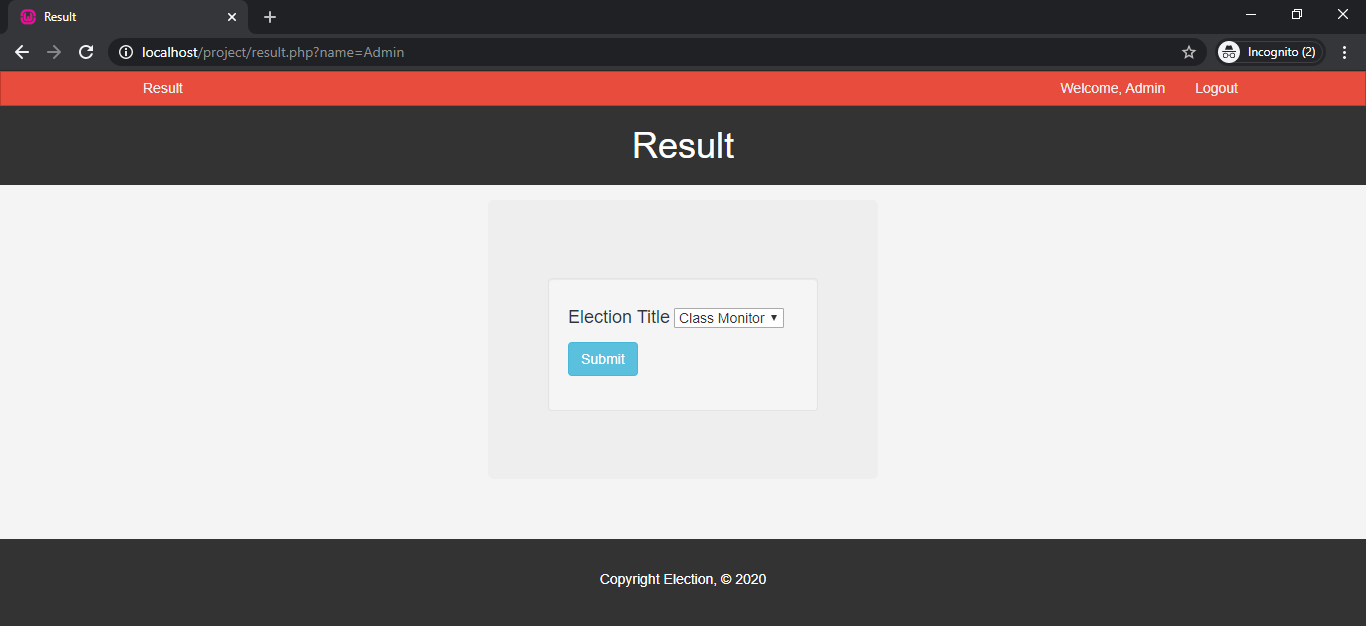
**2. Voter login**

* Voter can vote for election after login.
* Voter login with user name and password.

**3. Pending Election**

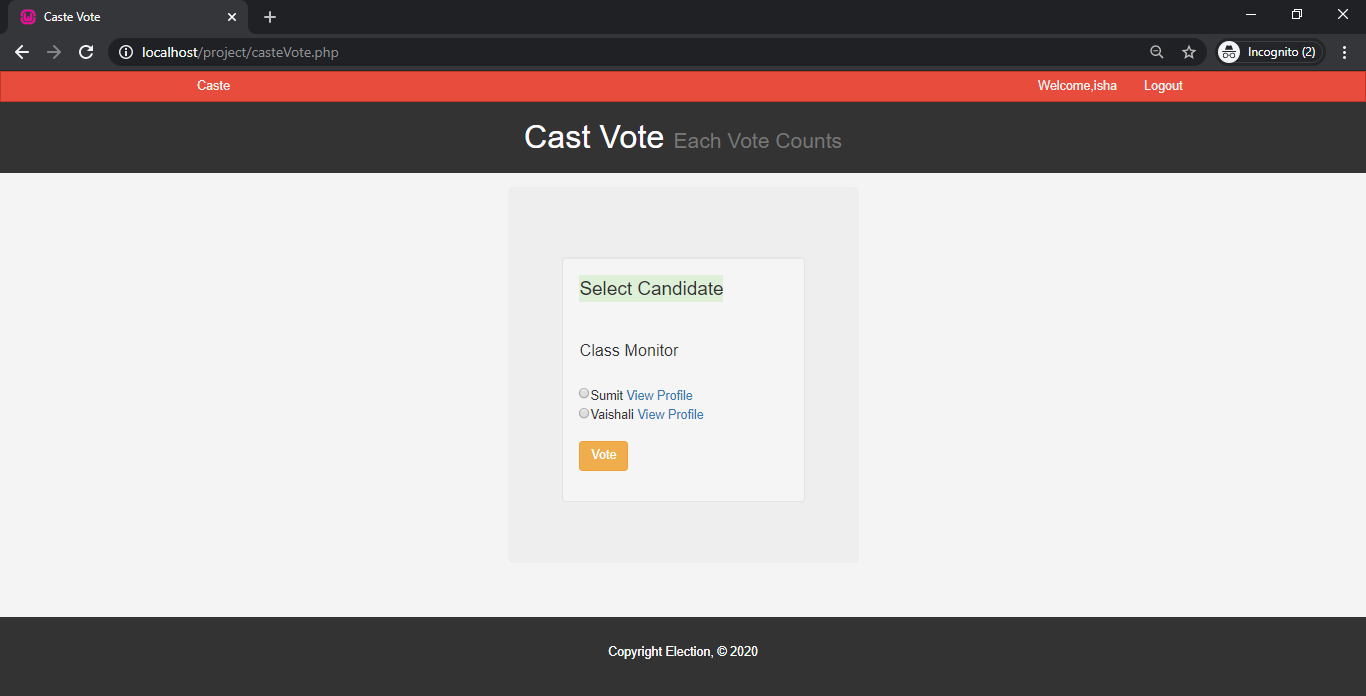
* After login, Voter can see the Pending Election from the dropdown.
* User can select the Election and go to cast vote.
* User can also check the result of the Election.
* User can logout from account.

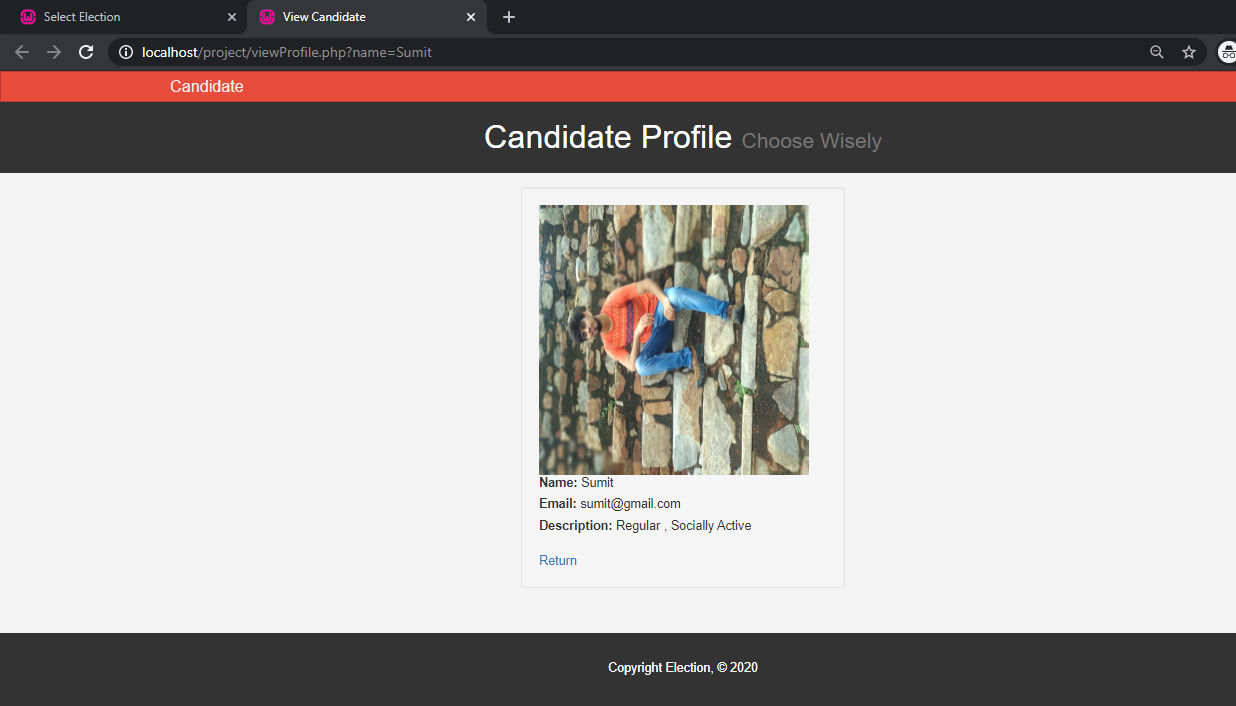




**4. Cast Vote**

* After choosing election, Voter can see the registered Candidates.
* User can select the Candidate and cast vote.
* User can also check the candidate profile.
* User can logout from account.

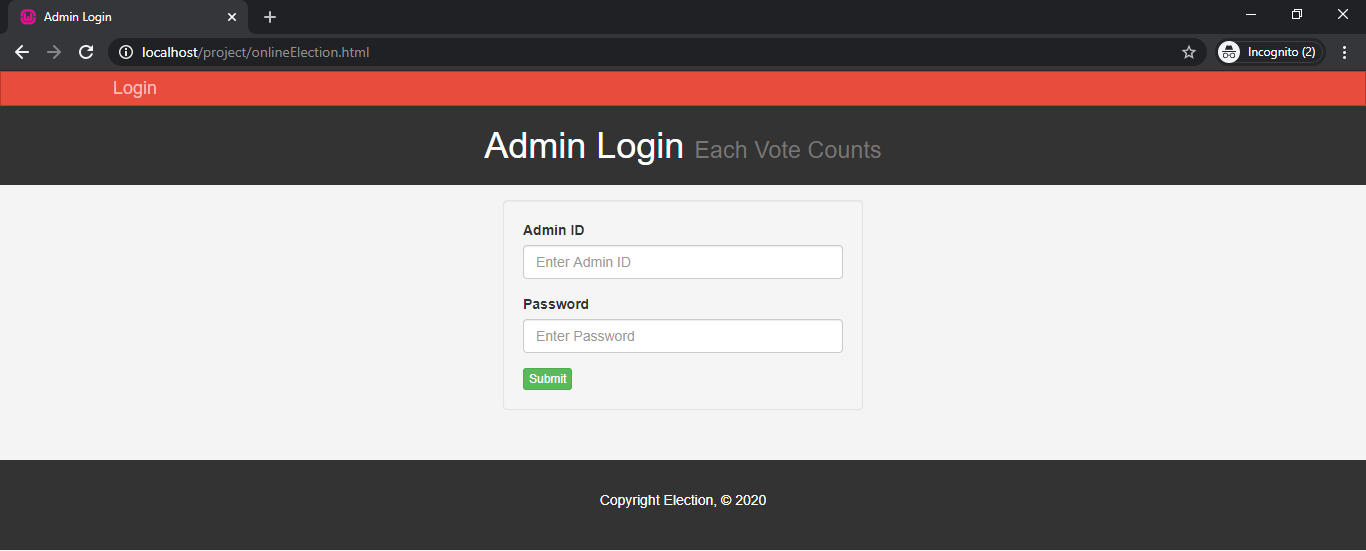




**ADMIN CONTROL :**

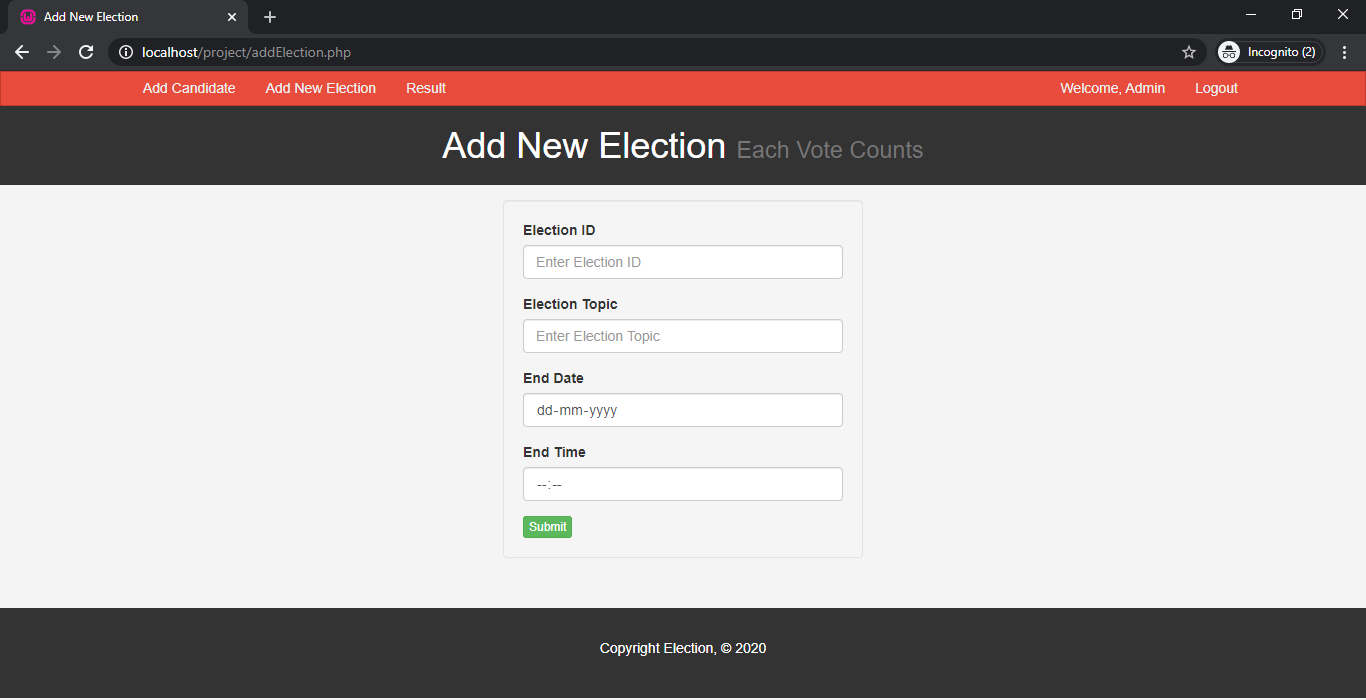
**1. Admin User Login**

* Admin user login with admin username and password.



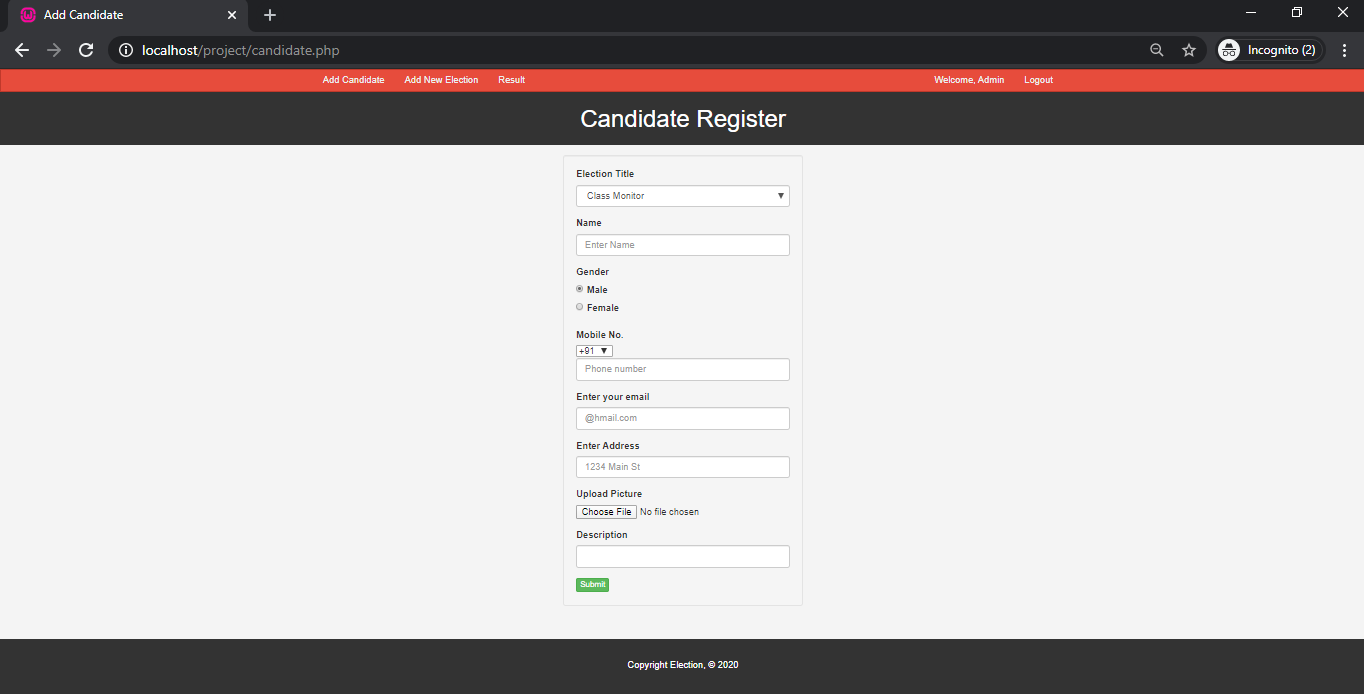
**2. Admin Action**

* Admin can add Election and Candidate.



**3. Add Election**

* Admin can add Election Details by entering ElectionID,Election Topic, End Date, End Time.
* Admin can corresponding candidates too.



**4.Candidate Register**

* Admin can add Candidates corresponding to Election .
* Admin can select Election Title and add candidate details including name, Gender, Mobile Number, Email, Address, Picture and Description
* Admin can also check Results

**FUNCTIONAL REQUIREMENTS**

* The date of the election will be generated via a query according to the data selected by the admin.
* A way in which users(both voter and admin) can login to the system to perform different operation.
* A way in which the voter can vote by “point and click” method.
* The system can verify the data before submission.

**SOFTWARE AND HARDWARE REQUIREMENTS**

* Intel core i3 2nd generation is used as processor because it is faster and provide reliable and stable working environment.
* A RAM size of 1gb is used as it will provide fast reading and writing capabilities.

**NON-FUNCTIONAL REQUIREMENTS**

* The non-functional requirements specify the qualitative attributes such as user-friendliness and performance of the system that are critical for the increased user-acceptance of the application.

**FEASIBILITY STUDY**

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work , effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability which is the impact on the organization, ability to meet their user needs and effective use of recourses. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development. The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as technical economic and operational feasibilities . The following are its features:

**Technical Feasibility:**

 The system must be evaluated from the technical point of view first. The assessment of this feasibility shall be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation goes on to suggest the type of equipment, required method of developing the system and of running the system once it has been designed. The project shall be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology.

**Economic Feasibility**

 The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

 Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the system is economically possible for development.

**Behavioral Feasibility**

 The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

**DESIGN ENGINEERING**

Software architecture alludes to “the overall structure of the software and the ways in which that structure provides conceptual integrity for a system”. In its simplest form, architecture is the structure or organization of program components (modules), the manner in which these components interact, and the structure of data that are used by the components. In a broader sense, however, components can be generalized to represent major system elements and their interactions. One goal of software design is to derive an architectural rendering of a system. This rendering serves as a framework from which more detailed design activities are conducted. A set of architectural patterns enables a software engineer to solve common design problems.

**FLOWCHART**

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows.

Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields. They help the people to understand a process, and perhaps also find flaws, bottlenecks, and other less obvious features within it.

**Online Election System**

Caste Vote

Check Result

Pending Election

Voter Registration

Add Candidate

Add Election

N0 YES

YES

**VERIFICATION**

Check

Validity

Access to limited resources

Check if admin?

Check Password Correct or not

Enter password

Input ID

NO YES

Access to all resources

YES NO

NO YES

**CYCLOMATIC COMPLEXITY**

Cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of a program. When used in the context of the basis path testing method, the value computed for cyclomatic complexity defines the number of independent paths in the basis set of a program and provides you with an upper bound for the number of tests that must be conducted to ensure that all statements have been executed at least once.

Cyclomatic complexity has a foundation in graph theory and provides you withan extremely useful software metric. Complexity is computed in one of three ways:

1.The number of regions of the flow graph corresponds to the cyclomatic complexity.

2. Cyclomatic complexity V(G) for a flow graph G is defined as

**V(G) = E – N + 2**

where E is the number of flow graph edges and N is the number of flow graph nodes.

3. Cyclomatic complexity V(G) for a flow graph G is also defined as

**V(G) = P + 1**

where P is the number of predicate nodes contained in the flow graph .

**FLOWGRAPH**

**Online Election**

INDEPENDENT PATHS :**V(G) = E – N + 2**

**1->2->3->4->5->6->8 = 9 – 8 + 2**

**1->2->3->4->5->7->8 = 3**

**1->2->3->4->5->6->4….**

**VERIFICATION**

INDEPENDENT PATHS :**V(G) = E – N + 2**

**1->2->3->4->5->6->8->9 = 11 – 9 + 2**

**1->2->3->4->5->6->7->9 = 4**

**1->2->3->9**

**1->2->3->4->5->2 . . . .**

**TESTING**

**TEST CASES**

A test case is a list of the conditions or issues of what the tester want to test in a software. Test case helps to come up with test data. A test case has an input description, Test sequence and an expected behavior. The characteristics of a test case are that there is a known input and an expected output, which is worked out before the test. The known input should test a pre-condition and the expected output should test a post-condition.

Test cases include a description of the functionality to be tested taken from either the requirements or use cases, and the preparation required to ensure that the test can be conducted.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Test Cases** | **Expected Result** | **Test Result** |
| 1. | Click on Login button with valid username and password | System allow voter to login | Successful |
| 2. | Click on Login button with invalid username and password | Message “Username/Password Not Found” | Successful |
| 3. | Click on register button | System allow to go on register page | Successful |
| 4. | Test whether Voter is able to see Pending Election | System shows pending Elections | Successful |
| 5. | Click on View Candidate Profile | System allow to see selected candidate profile | Successful |
| 6. | Click on Caste Vote | System allow to vote to selected candidate | Successful |
| 7. | On completion of casting vote, voter not allowed to vote again | System should display “You don't have any Pending Election to vote for” | Successful |
| 8. | Click on Result | System should display Winner corresponding to Election | Successful |