# ONLINE VOTING SYSTEM

*A Socially relevant Project Report (19A05507) submitted to*

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITYANANTAPUR,**

*In partial fulfillment of the requirements for the award of the degree of*

## BACHELOR OF TECHNOLOGY

**In**

## COMPUTER SCIENCE AND ENGINEERING

*Submitted by*

## K. PAVAN KUMAR REDDY (19HR1A0559) J. JAYA PRAKASH (19HR1A0547)

**D. SHRAVANI (19HR1A0536) G. ROSHINI (19HR1A0546)**

*Under the esteemed guidance of*

**Mr.J.Naveen Kumar,M.E, Assistant Professor,**

 **CSE Department**

#### Department of Computer Science and Engineering

#### MOTHER THERESA INSTITUTE OF ENGINEERING AND TECHNOLOGY

**Melumoi (Post), Palamaner-517408.**

**Approved by AICTE, New Delhi and Affiliated to JNTUA, Anantapuramu-515002**

**NAAC Accredited and An ISO 9001:2015 Certified Institution**

**2021-2022**

**MOTHER THERESA INSTITUTE OF ENGINEERING & TECHNOLOGY**

**AN ISO 9001:2015 CERTIFIED INSTITUTION**

**(Approved by AICTE, New Delhi and Affiliated to J.N.T.U.A., Anantapuramu) Melumoi (Post), Palamaner-517 408, Chitoor (Dist), A.P.**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Certificate

*This is to certify that the Project Report entitled*

**ONLINE VOTING STSTEM**

*Is the bonafide work done and*

*Submitted by*

## K. PAVAN KUMAR REDDY (19HR1A0559) J. JAYA PRAKASH (19HR1A0547)

**D. SHRAVANI (19HR1A0536) G. ROSHINI (19HR1A0546)**

*In the Department of Computer Science and Engineering, Mother Theresa Institute of Engineering & Technology, Palamaner affiliated to J.N.T.U.A., Anantapuramu in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering during 2021-2022.*

**Submitted on:**

**Internal Guide HOD**

Mr.J.Naveen Kumar , M.E, Dr. U. Kumaran , M.E, Ph.D.,

Assistant Professor, Associate Professor&HOD,

Dept. of CSE Dept. of CSE

**Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

We are extremely thankful to our beloved Secretary & Correspondent **Sri. M. Ravindra Babu** and Chairman **Sri. M. R. Sunil,** who took keen interest and encouraged us in every effort throughout this course.

We owe our gratitude to **Dr. M. Lakshmikantha Reddy**, Principal, MOTHER THERESA INSTITUTE OF ENGINEERING & TECHNOLOGY for permitting us to use the facilities available to accomplish the project successfully.

We express our heartfelt thanks to Dr. **U.Kumaran**, M.E, Ph. D, Associate professor and Head of Department of Computer Science and Engineering, for his/her kind attention and valuable guidance to us throughout this course.

We are thankful to our Project Coordinator **Mrs. N.V.Neeraja**, Assitant professor of Computer Science and Engineering for his/her valuable support and guidance throughout the project work.

We are extremely thankful to our Project Guide **Mr.J.Naveen Kumar,** M.E,Assistant professor of Computer Science and Engineering who took keen interest and encouraged us in every effort throughout this project.

We also thank all the teaching and non-teaching staff of Department Name Department for their cooperation.

**K . PAVAN KUMAR REDDY (19HR1A0559)**

**J . JAYA PRAKASH (19HR1A0547)**

**D . SHRAVANI (19HR1A0536)**

**G . ROSHINI (19HR1A0546)**

**ABSTRACT**

Voting is a widely spread and democratic way of making decisions. For centuries, So many countries have been using the popular paper-based voting system, which does not provide the desirable blend of accessibility and efficiency. Missing ballot papers, invalid votes and miscount are some of the challenges associated with the paper-based voting system. Electronic voting has been attracting a lot of attention and research for the past few years all over the world, for it has some remarkable advantages over traditional paper-based voting. This research proposes the use of mobile phones to facilitate communication and rapid access to information and their diffusion has reached a larger proportion of the population in a short period of time. When such a device is available why not use it for a time saving, cost effective, secured method of casting a vote. Mobile phone voting has the capability to augment the participation rate and the quality of voting. In this paper, we designed and developed a mobile phone voting system, which allows users to spontaneously vote timeously using cellular network service providers. We developed the system using HTML ,CSS for front end , MySQL Server, PHP for back end and Java Script for both front end and back end. The project is mainly aimed at providing a secured and user friendly Online Voting System. This system deals with the design and development of a web-based voting system using voter’s details in order to provide a high performance and high security to the voting systems. The voting system is managed in a simpler way as all the users must log in by username and password and clicking on his/her favourable candidates to cast the vote. It provides enough security which reduces the dummy votes.

**Keywords:** HTML ,CSS, PHP

**TABLE OF CONTENTS:**

**CHAPTERS : Page no**

|  |  |
| --- | --- |
| **1.INTRODUCTION** | **1 - 5** |
| 1.1 INTRODUCTION OF THE PROJECT | 1 - 1 |
| 1.2 DETAILED DESCRIPTION OF THE PROJECT | 1 - 2 |
| 1.3 PROJECT OBJECTIVE AND SCOPE | 2 - 5 |
| **2. LITERATURE SURVEY** | **6 - 7** |
| 2.1 SUPPORTING MATERIAL TO BASE PAPER | 6 - 6 |
| 2.2 DETAILED DESCRIPTION OF THE PAPER | 6 - 7 |
| **3. SYSTEM ANALYSIS** | **8 - 27** |
| 3.1 EXISTING SYSTEM | 8 - 9 |
| 3.2 PROPOSED SYSTEM | 9 - 10 |
| 3.3 MODULE DESCRIPTION | 10 - 27 |
| **4. SYSTEM DESIGN AND IMPLEMETATION** | **28 - 70** |
| 4.1 SYSTEM REQUIREMENTS | 28 - 28 |
| 4.2 DESIGN USING UML DIAGRAMS | 28 - 53 |
| 4.3 PROJECT IMPLEMENTATION | 53 - 70 |
| **5. TESTING** | **71 - 76** |
| 5.1 DESCRIPTION OF THE TESTING TOOLS | 71 - 74 |
| 5.2 OTHER TESTING METHODOLOGIES | 74 - 76 |
| **6. RESULTS** | **77 - 82** |
| **7. CONCLUSION** | **83 – 85** |
| 7.1 DETAILED CONCLUSION | 83 - 84 |
| 7.2 FUTURE SCOPE OF THE PROJECT | 84 - 85 |

##### **CHAPTER-1**

**INTRODUCTION**

**1.1.Introdution**

An online voting system is a software platform that allows groups to securely conduct votes and elections. High-quality online voting systems balance ballot security, accessibility, and the overall requirements of an organization’s voting event.

At their core, online voting systems protect the integrity of your vote by preventing voters from being able to vote multiple times. As a digital platform, they eliminate the need to gather in-person, cast votes using paper, or by any other means (e.g. email, insecure survey software). You may hear an online voting system being referred to as an online election system, an online e voting system, or electronic voting. These all make reference to the same thing: a secure voting tool that allows your group to collect input from your group and closely scrutinize the results in real time. Online voting tools and online election voting systems help you make important decisions by gathering the input of your group in a way that’s systematic and verifiable.

**1.2 Detailed Description of the Project**

### **Cost Savings and Efficiency**

### The cost savings and efficiencies you’ll gain are unparalleled to any other method of voting. Groups switching to web-based online voting systems from more expensive and less efficient voting technologies like voting machines, paper ballots, and in-person meetings will reap these benefits without increasing risk.

### **Voter Accessibility**

Needing to fly halfway around the world to vote at your organization’s annual meeting is an example of a vote with low accessibility. On the other hand, tapping a link on your mobile device that securely logs you into the online voting system website is an example of a vote or election with high accessibility.

**Auditability and Verifiability**

With an online voting system, you can easily showcase election results to eliminate concern. Sharing all administrator activity during your election to prove no one went in altered the results is just one of the many trust-building tactics you’ll be able to use in light of a vote challenge.

**Security, Confidence, and Trust in Your Election Results**

The confidence in your voting and election results is by far the most valuable aspect that online voting systems will offer to your group. The fallout of a vote being perceived as unfair is expensive, time-consuming, and wrecks havoc on the hard-earned trust you’ve built among group members. From this perspective, an online voting system offers unparalleled election security.

**1.2.1 DETAILED DESCRIPTION OF PROJECT REQUIREMENTS**

**SRS:**

SRS(Software Requirement Specification) is a document that completely describes what the proposed should do, without describing how the software does it.

**Software Quality Attributes:**

1. **Maintainbility –** Since it is directly associated with the database, so there is very little maintainability problem with this application.
2. **Easy to learn –** Since there are less number of forms, this application is very easy to learn with user-friendly screens.
3. **Flexibility -** This application is very much flexible for future enhancements.

**1.3 Project Objective and Scope**

**Objective Of This Project:**

The objectives of this project are:

**a**. To build an online system this would enable voters to cast their votes on chosen candidates.

**b.** Create a secure authentication facility to check validate users logging into the voting system.

**c.** Create a database to be used to stored votes, and user information on the system.

**d.** Study and implement a security method to be used to ensure that votes being cast in the system will not be compromised and any outside attack.

**e.** Create tools for the administrator to add, delete and update details of voters, candidates and sub administrators on the system.

**f.** Enable administrators to generate reports on the vote results.

**g.** Prevent voters from voting more than once for their choose candidates.

**Scope of Project :**:

The Scope the project is to increase the number of voters as individuals will find it easier and more convenient to vote, especially those who unable to vote their votes at that moment. To reduce the time spend making long queues at the polling stations during voting. It shall also enable the voters to vote from any part of the globe as explained this is an online application available on the internet. The voters shall ample time to decide when and whom to vote for.

Online Voting is a web-based voting system that will help you manage your elections easily and securely. This voting system can be used for casting votes during the elections held in colleges, etc. In this system the voter do not have to go to the polling booth to cast their vote. They can use their personal computer to cast their votes. There is a database which is maintained in which all the name of the voters with their complete information is stored. The System Administrator registers the voters by simply filling a registration form toregister the voters. After registration, the voter is assigned a secret voter ID with which he/she can use to login to the system and cast his/her vote. If invalid/wrong details are submitted, then the person is not registered to vote. After the user successfully registers themselves, a link is sent on their respective E-mail IDs. The link is a key for the activation of the account of the user. The account is activated only after the user clicks on that link. The site will be activated only on the day of voting. Once the user logs in, they will be provided with a One Time Password (OTP) which has to be entered by the user before casting his/her vote. The password will be destroyed after casting of their respective vote. A receipt of the vote will be sent to the user on their respective E-mail IDs.

The advantage ofonline voting is that the voters have the choice of voting at their own free

time and there is reduced congestion. It also minimizes on errors of vote counting. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes. At the end of any political term, millions of voters are called upon to cast their votes for their next political representatives. Unfortunately, many eligible voters will not be able to reach a polling station during the election. But if citizens could vote from anywhere, and at any time, then these hurdles wouldn’t exist.

This is exactly where online voting comes in. It gives voters the possibility to easily and comfortably fill out and cast a ballot using their own personal computer or smartphone, allowing them to vote from virtually any location with Internet access.

**How does it work?**

For a voter, online voting is comprised of three main steps:

**Registration**

To register to vote, voters are added to an electoral roll of eligible voters, as with any election. They are then provided with credentials, such as a username and password, to access the voting system.

**Voting**

Voters can cast a ballot by opening the election website from an Internet browser, identifying themselves (with their username and password, for example), and selecting their voting choices. Then all they need to do is click on the “cast” button.

**Verification**

Verification is the last step, and it is usually optional, enabling voters to verify that their ballot was correctly registered by the voting system.

**Background**

ONLINE VOTING SYSTEM is a voting system by which any Voter can use his\her voting rights from any where in India. ONLINE VOTING SYSTEM contains-:

* Voter’s information in database.
* Voter’s Names with ID.
* Voter’s vote in a database.
* Calculation of total number of votes

Various operational works that are done in the system are:-

* Recording information of the Voter in Voter database.
* Checking of information filled by voter.
* Discard the false information.
* Each information is sent to ELECTION COMMISSION OF INDIA.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 Supporting material to base paper:**

The required literature survey and information is collected from different sources for developing the project.

Project Methodology:

The most recent developments are focused on active learning and semi supervised learning approaches.

The literature requires in the existing methodology, in this assess the use of deep neural networks

**References:**

[1]. A.S. Belenky and R.C. Larson, “To Queue or not to Queue?,” OR/MS 27, October 2013, pp [2]. R. Krimmer (ed.), Electronic Voting, Proceedings of the 2nd International Workshop, Gesellschaft für Informatik, Bonn, Köllen Druck+Verlag GmbH, Bonn, October 2013. [3]. “An Electronic Polling Service to Support Public Awareness Using Web Technologies”, Christos Bouras, Nikolaos Katris, Vassilis Triantafillou.

[4]. “E-voting on Android System” paper (International Journal of Emerging Technology and Advanced Engineering) prepared by : Kirti Autade, Pallavi Ghadge, Sarika Kale ,Co-authors- Prof. N. J. Kulkarni, Prof. S. S. Mujgond, February 2012. [5]. “Electronic Voting,” Encyclopedia of Computers and Computer History, prepared by Lorrie Faith Cranor and edited by Raul Rojas, published by Fitzroy Dearborn, 2001.

[6]. “Voting – What is, What Could be,” Caltech/MIT Voting Technology Project (VTP) Report, July 2001.

[7]. “A Modular Voting Architecture (“Frogs”),” Shuki Bruck, David Jefferson, and Ronald L. Rivest, August.

[8]. “Comments of Professor Ronald L. Rivest”, Caltech/MIT VTP Press Conference, July 16, 2001. [9]. “Testimony given before the U.S. House Committee on Administration”, Ronald L. Rivest, May 24, 2001.

[10]. “Electronic Voting,” Ronald L. Rivest, Technical Report, Laboratory for Computer Science, Massachusetts Institute of Technology.

**2.2 Detailed Description of the Paper:**

Online voting refers to the use of computers or computerized voting equipment to cast ballots

in an election. Sometimes this term is used more specially to refer to voting that takes place over the internet. Online voting system can be used to register voters, tally ballots, and record votes.

The sample survey , conducted with a sample size of over four lakh people by location based social network platform Public App, also found that over 80 percent of respondents trust the current voting process in the country.

“ Voting as a civic duty is a crucial contribution to social development by the citizens of the nation. As many as 86 percent of respondents were in agreement when asked if voting should be made mandatory in the country,” the survey stated.

There are many challenges in conducting election surveys that measure voter preferences correctly and when the results of these surveys are used to make seat predictions the margin of error can be large.

**CHAPTER-3**

**SYSTEM ANALYSIS**

**3.1 EXISTING STSTEM**

In the present system there is no such application level system provisions in the country to carry out the voting and procedure as a whole. Also in the present status, there is no such application in use for automated system for voting according to the voting structure existing in the country. All the step procedures are carried out by the authorized authorities according to the jobs assigned by the ECI. The fact is all the procedure are carried out manually, starting from the registration process to result publishing.

The government to do this process manually wastes a lot of time and money. Thus the present system proves itself to be an inefficient one. The existing system is not web based. The user or person must want to go the polling station for casting their votes.

The existing of voting is running manually. The Voter has to visit the booths to vote a Candidate so there is a wastage of time. The voter has to manually register into the voter list. Also vote counting has to be done manually. Voter must be present in his/her constituency to give their vote. There are electronic voting machines used which Takes more cost. Some of the existing systems are:

**🡪**Paper-Based voting

**🡪**Direct recording electronic voting machine

So it would be uploaded as image matching system. The image comparison system matches the two images if two images are same then the voter got as present. Some time the image matching not exactly matched

In paper based voting counting mistakes may occur due to manual counting and the counting of the votes will takes time to calculate. Most of the people are working somewhere from their constituencies but they are voters for that constituency. They have to travel such distance to cast their votes at their constituency. Due to paper voting the percentage of people voting their votes are nearly 70% because of they are unable to cast their vote at that time. Electronic Voting System is cost effective for the government and their may transport problem during the shifting of EVM’s from one place to another.

Right to vote is the important and basic right of every Indian. Election to elect candidates takes place in every 5 years. Voters have to go to polling booth to cast their votes .My project is an android application for online voting system. By using this application they can cast their votes securely and from their home itself. Here Aadhar Identity number is used for the unique identification of users and one time password used for security

 If the vote is not clear or if the seal presents in two places of the same ballot then the vote is considered to be not valid. Then the E- voting comes into existence. But sometimes this E- voting is not secure. And for some reasons some people did not come to polling places to cast their votes.

**3.2 PROPOSED SYSTEM**

In this proposed system android application is created so that people can cast their vote from their mobile phones. People can download the app file which will be available in the internet. People can use their Aadhar identity number to show their unique identity. Votes will be secure and can be counted easily from database.

The online voting system will manage the Voter’s information by which voter can login and use his voting rights. There is a Admin in which the admin user can manage all the voter’s data. At the time of registration the voter will be asked for this :

🡪First Name

🡪Last Name

🡪Gender

🡪Age

🡪Email-id

🡪Profile photo etc.,

At the time of requesting vote, voter will be asked to enter his/her credentials. Then the voter can give vote from one of the candidate from the list. The voter can only once. If the voter will again login into the page the previous candidate page will not open and it shows only the person who you have already voted. Every information collected by the admin will be stored in the respective database. This system will increase the voting percentage in voting system. If security is applied then it may reduce false votes.

“ ONLINE VOTING SYSTEM” is an online voting technique. It is based on the other online services like "ONLINE RESERVATION SYSTEM”. In this system people who have citizenship of INDIA and whose age is above 18 years of any sex can give his/her vote

online without going to any polling booth. There is a DATABASE which is maintained by the ELECTION COMMISION OF INDIA in which all the names of voter with complete information is stored. All the entries is checked by the DATABASE which has already all information about the voter, by using that ID and PASSWORD he/she can use his/her vote. If conditions are wrong then that entry will be discarded

Advantages of this Project:

🡪Accessible and easy to use

🡪 Secure

🡪Accurate

🡪Increased Turnout & Engagement

🡪Reduced costs

🡪Audit trail

🡪Saves you time and Money

🡪Votes are protected

🡪Transparency and Fairness

**3.3 MODULE DESCRIPTION**

3.3.1 Main Objectives/ Functionalities/Activities of the Modules:

We proposed a new model based on Digital based ,which provides an efficient approach for voting system. The whole process consists of four phases: Registration at Admin, data set formation for each user in Database, login and voting. These three phases are explained in the following subsections.

**MODULE DESIGN**

The proposed system consists of two modules,

* Administrator Module
* User Module

**USER (*VOTER*) MODULE**

User interface consists of a login name and unique password using which he/she can login into the Online Voting System. This will be supplied by the administrator to the user. Once the user has logged in, he has the privilege to view the names of the candidates listed by the administrator, view the results after the termination date of the election. The user module constitutes only one sub module:

**Authentication & Voting**

* Each voter is provided with unique username and password manually by the administrator. The voter uses the username and password for login and exercise the fundamental right of voting. if incorrect username and password entered, the access to is denied to the user. And also voter is allowed to vote only once. This is the security feature provided against external access of the system.
* After login the voter enters the voter home page ,which provides the links :

**Candidate List**

This facilitates the voter to view the candidate names, the constituency name, their symbol and their party name.

**Vote**

This provides the voter with a list of candidate with in his/her constituency along with selection option (radio button) to select the preferred candidate from the list. If the voting date is before termination date, the vote goes valid else goes invalid.

**View Results**

This provides graphical and user friendly representation of the votes obtained by each candidate. It includes the percentage of the votes obtained by each candidate. But the result can be viewed only after the termination date of the election.

**Logout**

This provides an option for the voter to quit the session ,while in the voter home page.

**ADMINISTRATOR MODULE**

Administrator interface consists of a login name and unique password using which admin can login into the Online Voting System. Administrator has the main control of the system. By logging into the page it can perform the following tasks.

**Add Candidate**

Here the admin can add the list of candidates in the election. It includes candidates name, address. gender, party, party symbol etc. The candidates will be added to the list only after completing the procedures.

**Add Voter**

Here the voters can be added to the database. The voters have also the privilege to check the voters list from the homepage. The voters details includes name, address, gender, age, constituency, image etc.

**Add Election**

Here the election to be conducted is selected. To add an election the constituency should be selected and termination date of election should be specified.

**Add Constituency**

The constituency that is going to conduct election should be selected.

**Voters List**

Here we can view the voters list. Each constituency will be having separate voters list.

**Candidate List**

The list of candidates participating in the election can be seen. It includes the candidates name, party name and party symbol

**The sub-modules of administrator are:**

* Voting Structure**,**
* Voters Registration
* Candidate Registration
* Counting & Categorization of Results

**Voting Structure**

Here the eligible voters who are permitted to login to the system can utilize the right to vote. Each voter can register a single vote to a candidate’s favour in his/her constituency. The security measures taken within the system prevents them from exercising their votes again i.e. the second vote by the same user goes invalid. The starting and ending dates of the election are specified by the administrator. The user must have an identity card and he must be in voters list.

**Voters Registration**

The registration procedure of all the eligible voters .This registration process is done by the administrator. According to voters database each voter is provided with a unique identification codes which includes username and password.

The details of the voters include username, password, name, address, gender, constituency, image etc.With the voter registration, thus producing the voter list with the given information of the voters. The voter list can be viewed by anyone accessing the webpage. The admin can view the voter list with in his homepage.

**Candidate Registration**

The registration of the candidates in each constituency is done by the administrator. The details of the candidate includes name, address, gender ,his/her constituency, party and image. With the candidate registration, thus producing the candidate list with the given information of the candidates. The candidate list can be viewed by admin and the vote within their respective homepages. According to candidates database (manual) each details of the candidates are stored in database controlled by the admin including candidates details.

**Counting & Categorization of Results**

When the voter votes, the number of votes obtained by the selected candidate is incremented by 1.The result is published only after the voting process is over. It is accessible from the next day after the termination date. Here we depict the result in the graphical representation according to the percentage of vote obtained by the candidate. Result can be viewed by everyone who visits into the site without any authentication problem**.** A link to view the result is kept in the index page and both admin and voter can view the result in their respective homepages. When the user clicks the “RESULT” link, before the termination date of the election,  
 “Result not Published yet” Message will be displayed. The result comes with their party symbol on the top of the graph representing the percentage of vote obtained by each candidate.

In this online voting system we used XAMPP software for the database of our project:

**Installing XAMPP:**

#### Step 1: Download

XAMPP is a release made available by the non-profit project Apache Friends. Versions with PHP 5.5, 5.6, or 7 are available for download on the [Apache Friends](http://www.apachefriends.org/de/download.html) website

#### Step 2: Run .exe file

Once the software bundle has been downloaded, you can start the installation by double clicking on the file with the ending [.exe](https://www.ionos.com/digitalguide/server/know-how/exe-file/).

#### Step 3: Deactivate any antivirus software

Since an active antivirus program can negatively affect the installation process, it’s recommended to temporarily pause any antivirus software until all XAMPP components have successfully been installed.



Before installing XAMPP, it is advisable to disable the anti-virus program temporarily

**Step 4: Deactivate UAC**

User Account Control (UAC) can interfere with the XAMPP installation because it limits writing access to the C: drive, so we recommend you deactivate this too for the duration of the installation process.



**Step 5: Start the setup wizard**

After you’ve opened the .exe file (after deactivating your antivirus program(s) and taken note of the User Account Control, the start screen of the XAMPP setup wizard should appear automatically. Click on ‘Next’ to configure the installation settings.



**Step 6: Choose software components**

Under ‘Select Components’, you have the option to exclude individual components of the XAMPP software bundle from the installation. But for a full local test server, we recommend you install using the standard setup and all available components. After making your choice, click ‘Next’.



**Step 7: Choose the installation directory**

In this next step, you have the chance to choose where you’d like the XAMPP software packet to be installed. If you opt for the standard setup, then a folder with the name XAMPP will be created under C:\ for you. After you’ve chosen a location, click ‘Next’

.

**Step 8: Start the installation process**

Once all the aforementioned preferences have been decided, click to start the installation. The setup wizard will unpack and install the selected components and save them to the designated directory. This process can take several minutes in total. You can follow the progress of this installation by keeping an eye on the green loading bar in the middle of the screen.



#### Step 9: Windows Firewall blocking

Your Firewall may interrupt the installation process to block the some components of the XAMPP. Use the corresponding check box to enable communication between the Apache server and your private network or work network. Remember that making your XAMPP server available for public networks isn’t recommended.

#### Step 10: Complete installation

Once all the components are unpacked and installed, you can close the setup wizard by clicking on ‘Finish’. Click to tick the corresponding check box and open the XAMPP Control Panel once the installation process is finished.



## The XAMPP Control Panel

Controls for the individual components of your test server can be reached through the XAMPP Control Panel. **The clear user interface**logs all actions and allows you to start or stop individual modules with a single. The XAMPP Control Panel also offers you various other buttons, including:

* **Config:** allows you to configure the XAMPP as well as the individual components
* [Netstat](https://www.ionos.com/digitalguide/server/tools/introduction-to-netstat/)**:** shows all running processes on the local computer
* **Shell:** opens a UNIX shell
* **Explorer:** opens the XAMPP folder in Windows Explorer
* **Services:** shows all services currently running in the background
* **Help:**offers links to user forums
* **Quit:**closes the XAMPP Control Panel



### **Starting modules**

Individual modules can be started or stopped on the XAMPP Control Panel through the corresponding buttons under ‘Actions’. You can see which modules have been started because their names are highlighted green under the ‘Module’ title.

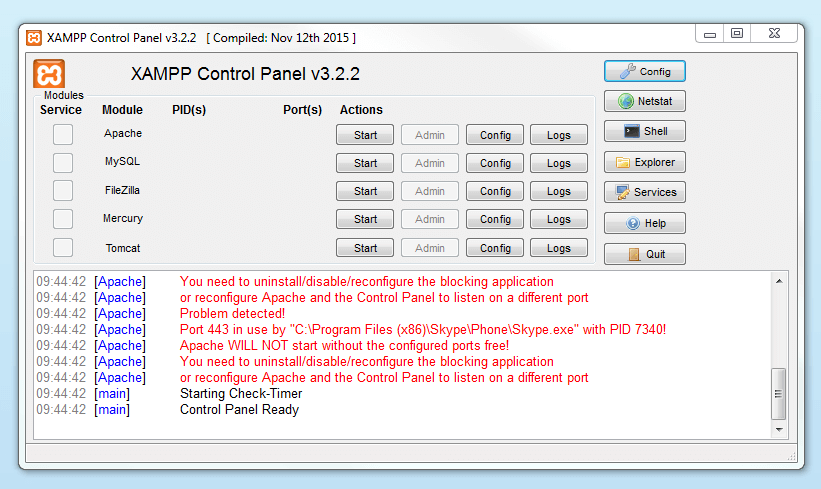


If a module can’t be started as a result of an error, you’ll be informed of this straight away in red font. A **detailed error report** can help you identify the cause of the issue.

### **Setting up XAMPP**

### A common source of error connected with Apache is **blocked ports**. If you’re using the standard setup, then XAMPP will assign the web server to main port 80 and the SSL port 443. The latter of these particularly is often blocked by other programs. In the example above, it’s likely that the Tomcat port is being blocked, meaning the web server can’t be started. There are three ways to solve this issue:

* **Change the conflicting port:**Let’s assume for the sake of example that the instant messenger program Skype is blocking SSL port 443 (this is a common problem). One way to deal with this issue is to change Skype’s port settings. To do this, open the program and navigate via ‘Actions’, ‘Options’, and ‘Advanced’, until you reach the ‘Connections’ menu. You should find a box checked to allow Skype access to ports 80 and 443. Deselect this checkbox now.
* **Change the XAMPP module port settings**: Click the Config button for the module in question and open the files httpd.conf and httpd-ssl.conf. Replace port number 80 in httpd.conf and port number 443 in httpd-ssl.conf with any free ports, before saving the file data. Now click on the general Config button on the right-hand side and select ‘Services and Ports Settings’. Customize the ports for the module server to reflect the changes in the conf files.
* **End the conflicting program:**The simplest way to avoid port conflicts in the short term is to end the conflicting program (Skype in this case). If you restart Skype after your XAMPP module servers are already running, it will select a different port and your issue will be resolved.



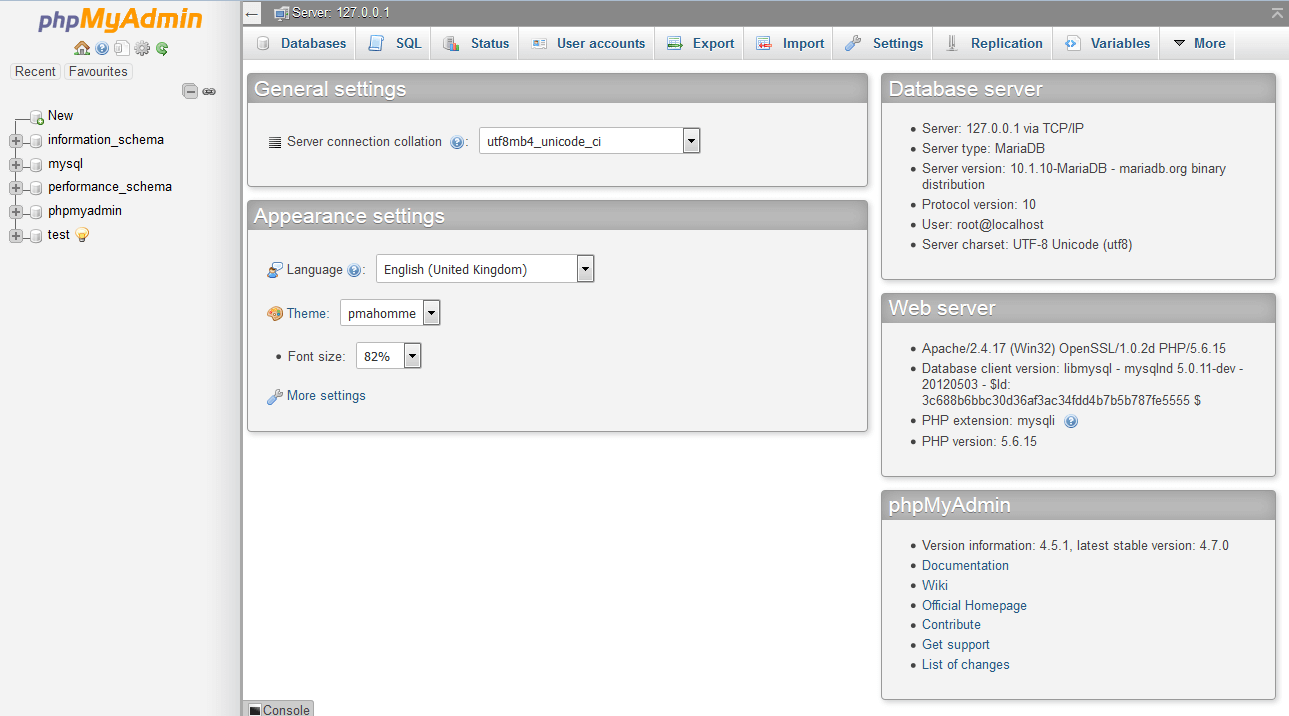
**Module administration**

You have an ‘Admin’ option located on the Control Panel for every module in your XAMPP.

* Click on the Admin button of your Apache server to go to the web address of your web server. The Control Panel will now start in your standard browser, and you’ll be led to the **dashboard of your XAMPP’s local host**. The dashboard features numerous links to websites for useful information as well as the open source project [BitNami](https://bitnami.com/stack/xampp" \t "_blank" \o "Link to homepage for BitNami), which offers you many different applications for your XAMPP, like WordPress or other content management systems. Alternatively, you can reach the dashboard through localhost/dashboard/.



* You can use the Admin button of your database module to open **phpMyAdmin**. Here, you can manage the databases of your web projects that you’re testing on your XAMPP. Alternatively, you can reach the administration section of your MySQL database via [localhost](https://www.ionos.com/digitalguide/server/know-how/localhost/)*/phpmyadmin/*.



## Testing your XAMPP installation

To check whether your test server is installed and configured correctly, you have the option to create a **PHP test page**, store them on your XAMPP’s local host, and retrieve them via the web browser.

* Open the XAMPP directory through the ‘Explorer’ button in the Control Panel and choose the folder htdocs (C:\xampp\htdocs for standard installations). This directory will store file data collected for web pages that you test on your XAMPP server. The htdocs folder should already contain data to help configuration of the web server. But you should store your own projects in a new folder (like ‘Test Folder’ for example).
* You can create a new PHP page easily by using the following content in your editor and storing it as test.php in your ‘ test’ folder (C:\xampp\htdocs\test):

<html>

<head>

<title>PHP-Test</title>

</head>

<body>

<?php echo '<p>Hello World</p>'; ?>

</body>

</html>

* The last step now is to open your web browser and load your PHP page via localhost/test/test.php. If your browser window displays the words ‘Hello World’, then you’ve successfully installed and configured your XAMPP

**Database Tables:**

**Admin Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data Type** | **Description** |
|  | **Login id** | **Varchar** | **Login id for Admin.(Primary key)** |
|  | **Password** | **Varchar** | **Password for Login** |
|  | **Name** | **Varchar** | **Name of the Administrator** |

**Voter information Data Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data Type** | **Description** |
|  | **Login id** | **Varchar** | **Login id for Voter(Primary key)** |
|  | **Password** | **Varchar** | **Password for Login** |
|  | **Name** | **Varchar** | **Name of the voter** |
|  | **Father’s/Husband name** | **Varchar** | **Voter’s father or**  **husband name** |
|  | **House no.** | **Varchar** | **House no. of voter** |
|  | **Address** | **Varchar** | **Address of voter** |
|  | **City** | **Varchar** | **City of voter** |
|  | **Mobile** | **Varchar** | **Mobile No of voter** |
|  | **E-Mail** | **Varchar** | **E-Mail of voter** |

**Represents Constituency**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **TYPE** | **KEY** |
| **Constituency id** | **int(11)** | **Primary** |
| **Location id** | **int(11)** | **Foreign** |
| **Constituency name** | **varchar(20)** |  |
| **Candidate number** | **int(11)** |  |
| **Voter number** | **int(11)** |  |

**Represents Election details**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **TYPE** | **KEY** |
| **Election details id** | **Int(11)** | **Primary** |
| **Voter id** | **int(11)** | **Foreign** |
| **Election master id** | **int(11)** | **Foreign** |
| **Status** | **varchar(20)** |  |

**Represents Election master**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **TYPE** | **KEY** |
| **Election master id** | **int(11)** | **Primary** |
| **Election name** | **varchar(30)** |  |
| **Constituency id** | **int(11)** | **Foreign** |
| **Termination date** | **Date** |  |

**Represents Location**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **TYPE** | **KEY** |
| **Location id** | **int(11)** | **Primary** |
| **Location name** | **varchar(40)** |  |
| **Voter no** | **int(11)** |  |

**Represents Login**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **TYPE** | **KEY** |
| **Login id** | **int(11)** | **Primary** |
| **Username** | **varchar(20)** |  |
| **Password** | **varchar(20)** |  |
| **Role** | **varchar(20)** |  |
| **Status** | **varchar(20)** |  |

**Represents Result**

|  |  |  |
| --- | --- | --- |
| **FIELD** | **TYPE** | **KEY** |
| **Result id** | **int(11)** | **Primary** |
| **Candidate id** | **int(11)** | **Foreign** |
| **Count** | **int(11)** |  |
| **Term date** | **Date** |  |

**MYSQL**

MySQL provides our small, medium and large enterprise customers with affordable, open access to their web data warehouses. MySQL allows us to offer our System Administrator low cost, low maintenance database solution for applications without sacrificing power, performance or scalability.

Benefits of MySQL are as follows:

* Easy to maintain & upgrade, does not have a slew of administrative tasks to put up.
* Its table format does not vary between releases
* It has cleanly separated table handler modules and can mix access to different type of tables.
* It seems to be developed iteratively, and the features are very stable when they ship them.

It is a relational database. Over the past several years, this relational database management systems have become the most widely accepted way to manage data.

It offers benefits such as:

* Easy to access data
* Flexibility in data modeling
* Reduced data storage and redundancy
* Independence of physical storage and logical data design
* A high level data manipulation language

.

**CHAPTER-4**

**SYSTEM DESIGN AND IMPLEMENTATION**

**4.1 System Requirements**

**4.1.1 Software Requirements:**

Software System Configuration:-

Operating system : Windows 7 and above Ultimate

Coding Language : HTML, CSS, Javascript, PHP, Mysql

**4.1.2 Hardware Requirements :**

Hardware System Configuration:-

* Processor - Pentium – 4 ,2.4 GHz
* RAM - 4 GB (min)
* Hard Disk - 40 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA
* Floppy drive - 1.44mb

**4.2 Design using UML diagrams :**

**4.2.1 Object Oriented Methodology:**

Object oriented methodology is a system development approach encouraging and facilitating re-use of software components. With this methodology, a computer can be developed on a component basis which enables the effective reuse of existing components and facilitates the sharing of its components by other systems. It employs international standard unified modelling language (UML) from the object management group (OMG) .

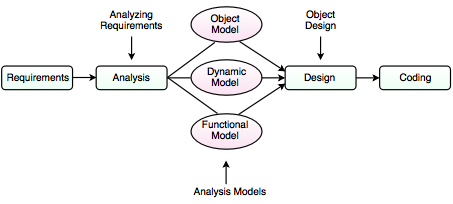
Using this methodology , a system can be developed on a component basis , which enables the effective reuse of existing components, it facilitates the sharing of its other system components. Objects oriented methodology asks the analyst to determine what the objects of the system are?, what responsibilities and relationships an object has to with the other objects? And how they behave over time?

**There are three types of object oriented methodologies:**

* Object Modelling Technique(OMT)
* Object Process Methodology(OPM)
* Rational Unified Process(RUP)

**Object Modelling Technique(OMT):**

It was one of the first object oriented methodologies and was introduced by RumBaugh in 1991. OMT uses that are combined in a way that are combined in a way that is analogous to the older structured methodologies.



**Analysis:**

The main goal of the analysis is to build models of the world. The requirements of the users, developers and managers provide the information needed to develop the initial problem statements.

**OMT MODELS**

**I. Object Model**

* It depicts the object classes and their relationships as a class diagram, which represents the static structure of the system.
* It observes all the objects as static and does not pay any attention to their dynamic nature.

**II. Dynamic Model**

* It captures the behaviour of the system over time and the flow control and events in the Event-Trace Diagrams and State Transition Diagrams.
* It portrays the changes occurring in the states of various objects with the events that might occur in the system.

**III. Functional Model**

* It describes the data transformations of the system.
* It describes the flow of data and the changes that occur to the data throughout the system.

**Design**

* It specifies all of the details needed to describe how the system will be implemented.
* In this phase, the details of the system analysis and system design are implemented.
* The objects identified in the system design phase are designed.

**Object Process Methodology (OPM):**

It is also called as second generation methodology. It was first introduced in 1995. It had only one diagram that is the object process diagram which is used for modelling the structure, function and behaviour of the system. It has a strong emphasis on modelling but has a weaker emphasis on process. It consists of three main process

**I. Initiating:** It determines high level requirements, the scope of the system and the resources that will be required.  
**II. Developing:** It involves the detailed analysis, design and implementation of the system.

**III. Deploying:** It introduces the system to the user and subsequent maintenance of the system.

**Rational Unified Process (RUP):**

It was developed in Rational Corporation in 1998. It consists of four phases which can be broken down into iterations.  
I. Inception  
II. Elaboration  
III. Construction  
IV. Transition

Each iteration consists of nine work areas called disciplines. A discipline depends on the phase in which the iteration is taking place. For each discipline, RUP defines a set of artefacts (work products), activities (work undertaken on the artefacts) and roles (the responsibilities of the members of the development team)

**4.2.2 Objectives of object oriented methodologies:**

* To encourage greater re-use.
* To produce a more detailed specification of system constraints.
* To have fewer problems with validation (Are we building the right product?).

**Benefits of object oriented methodologies:**

* It represents the problem domain, because it is easier to produce and understand designs.
* It allows changes more easily.
* It provides nice structure for thinking abstracting and leads to modular design.
* Simplicity
* Reusability
* Increased quality
* Maintainable
* Scalable
* Modularity

**Objectives**

After going through this unit, you should be able to:

* Find the importance of OO approach.
* Define the basic concepts of OO approach.
* Differentiate between object and procedure-oriented approaches.
* Know about various OO languages. Object Oriented Technology and Java
* Describe the applications of OOP, and
* Understand the benefits of OO approach

**Evolution of OO Methodology**

The earliest computers were programmed in machine language using 0 and 1. The mechanical switches were used to load programs. Then, to provide convenience to the programmer, assembly language was introduced where programmers use pneumonic for various instructions to write programs. But it was a tedious job to remember so many pneumonic codes for various instructions. Other major problem with the assembly languages is that they are machine architecture dependent. To overcome the difficulties of Assembly language, high-level languages came into existence. Programmers could write a series of English-like instructions that a compiler or interpreter could translate into the binary language of computers directly. These languages are simple in design and easy to use because programs at that time were relatively simple tasks like any arithmetic calculations. As a result, programs were pretty short, limited to about a few hundred line of source code. As the capacity and capability of computers increased, so did the scope to develop more complex computer programs. However, these languages suffered the limitations of reusability, flow control (only goto statements), difficulty due to global variables, understanding and maintainability of long programs.

**Structured Programming**

When the program becomes larger, a single list of instructions becomes unwieldy. It is difficult for a programmer to comprehend a large program unless it is broken down into smaller units. For this reason languages used the concept of functions (or subroutines, procedures, subprogram) to make programs more comprehensible. A program is divided into functions or subroutines where each function has a clearly defined purpose and a defined interface to the other functions in the program. Further, a number of functions are grouped together into larger entity called a module, but the principle remains the same, i.e. a grouping of components that carryout specific tasks. Dividing a program into functions and modules is one of the major characteristics of structured programming. By dividing the whole program using functions, a structured program minimizes the chance that one function will affect another. Structured programming helps the programmer to write an error free code and maintain control over each function. This makes the development and maintenance of the code faster and efficient. Structured programming remained the leading approach for almost two decades. With the emergence of new applications of computers the demand for software arose with many new features such as GUI (Graphical user interface). The complexity of such programs increased multi-fold and this approach started showing new problems. The problems arose due to the fundamental principle of this paradigm. The whole emphasis is on doing things. Functions do some activity, maybe a complex one, but the emphasis is still on doing. Data are given a lower status. For example in banking application, more emphasis is given to the function which collects the correct data in a desired format or the function which processes it by doing some summation, manipulation etc. or a function which displays it in the desired format or creates a report. But you will also agree that the important part is the data itself. The major drawback with structured programming are its primary components, i.e., functions and data structures. But unfortunately functions and data structures do not model the real world very well. Basically to model a real world situation data should be given more importance. Therefore, a new approach emerges with which we can express solutions in terms of real world entities and give due importance to data.

**Object Oriented Programming**

The world and its applications are not organized as functions and values separate from one another. The problem solvers do not think about the world in this manner. They always deal with their problems by concentrating on the objects, their characteristics and behavior.

The world is Object Oriented, and Object Oriented programming expresses programs in the ways that model how people perceive the world. It shows different real world objects around us which we often use for performing different functions. This shows that problem solving using the objects oriented approach is very close to our real life problem solving techniques.

The basic difference in Object Oriented programming (OOP) is that the program is organized around the data being operated upon rather than the operations performed. The basic idea behind OOP is to combine both, data and its functions that operate on the data into a single unit called object. Now in our next section, we will learn about the basic concepts used extensively in the Object Oriented approach.

**Basic Concepts of OO Approach**

Object Oriented methods are favored because many experts agree that Object Oriented techniques are more disciplined than conventional structured techniques. (Martin and Odell 1992)

The main components of Object Oriented technology are ‘objects and classes’, ‘data abstraction and encapsulation’, ‘inheritance’ and ‘polymorphism’. It is very important for you to understand these concepts. Further, in this unit you can find the details of these concepts.

**Objects**

Let’s start with “Object”. The first thing that we should do in the Object Oriented approach is to start thinking in terms of Objects. The problem to be solved is divided into objects. Start analyzing the problem in terms of objects and the nature of communication between them. Program object should be chosen such that they match closely with real-world objects. Let’s start creating objects using real-life things, for example, the dog. You can create an object representing a dog, It would have data like How hungry is it? How happy is it? Where is it? Now think what are the different functions you can perform on a dog, like eat, bark, run and dig. Similarly, the following can be treated as objects in different programming problems:

• Customers and accounts in a banking system

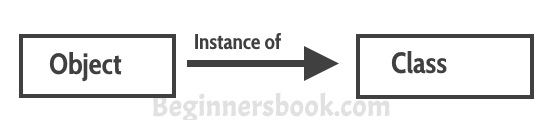
• Salesman, products, customers in a sales tracking system

• Data structures like linked lists, stacks, etc.

• Hardware devices like magnetic tape drive, keyboard, printer etc.

• GUI elements like windows, menus, events, etc. in any window-based application.

Each object contains data and the functions that operate on the data. Objects can interact without having to know details of each other’s data or functions. It is sufficient to know the type of message accepted and the type of response returned by the object. For example, in the banking system, customer object may send a message named as check balance to the account object to get the response, i.e. bank balance. An Object Oriented system can be considered as network of cooperating objects which interact by sending messages to each other. Let us see the below figure, how objects interact by sending messages to one another.



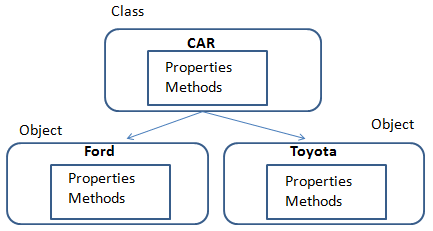
**Classes**

Objects of the similar type can be grouped together to form a class. Can you tell to which class dog belongs? Yes, of course, it belongs to the animal class. Now, let us concentrate on the creation of objects. This can be easily answered if we look at the way of creating any variable in common programming languages. Almost all computer languages have built-in data types, for example integer, character, real, Boolean , etc. One can declare as many variables of any built-in type as needed in any problem solution. In the similar way one can define many objects of the same class. You can take a class as a type created by a programmer.

A class serves as a plan or template. The programmer has to specify the entire set of data and functions for various operations on the data for an object as a user-defined type in the form of a class. In other words, the programmer defines the object format and behavior by defining a class. The compiler of that language does not know about this user-defined data type. The programmer has to define the data and functionality associated with it by designing a class.

Finally, defining the class doesn’t create an object just as the existence of a built-in type integer doesn’t create any variable.

A class is thus a collection of objects of similar type. For example, in a collection of potatoes each individual potato is an object and belongs to the class potato. Similarly, each individual car running on the road is an object, Collectively these cars are known as cars.



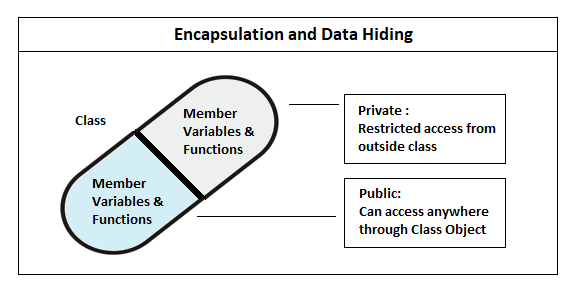
**Data abstraction and encapsulation**

The wrapping up of data and functions into a single unit is known as encapsulation. This is one of the strong features of the object oriented approach. The data is not directly accessible to the outside world and only the functions, which are wrapped in the class, can access it. Functions are accessible to the outside world. These functions provide the interface to access data. If one wants to modify the data of an object, s/he should know exactly what functions are available to interact with it. This insulation of the data from direct access by the program is known as data hiding.

Abstraction refers to the act of representing essential features without including the background details to distinguish objects/ functions from other objects/functions. In case of structured programming, functional abstraction was provided by telling, which task is performed by function and hiding how that task is performed. A step further, in the Object Oriented approach, classes use the concept of data abstraction. With data abstraction, data structures can be used without having to be concerned about the exact details of

implementation. As in case of built-in data types like integer, floating point, etc. The

programmer only knows about the various operations which can be performed on these data types, but how these operations are carried out by the hardware or software is hidden from the programmer. Similarly in Object Oriented approach, classes act as abstract data types. Classes are defined as a set of attributes and functions to operate on these attributes. They encapsulate all the essential properties of the objects that are to be created.



**Inheritance**

Inheritance is the process by which objects of one class acquire the properties of objects of another class in the hierarchy. For example, the scooter is a type of the class two-wheelers, which is again a type of (or kind of) the class motor vehicles. As shown in the Figure 5 the principle behind it is that the derived class shares common characteristics with the class from which it is derived.

New classes can be built from the existing classes. It means that we can add additional features to an existing class without modifying it. The new class is referred as derived class or sub class and the original class is known as base class or super class. Therefore, the concept of inheritance provides the idea of reusability. This inheritance mechanism allows the programmer to reuse a class that is made almost, but not exactly, similar to the required one by adding a few more features to it.

Feature A and Feature B of the base class are inherited in all the three derived classes. Also, each derived class has added its own features according to the requirement. Therefore, new classes use the concept of reusability and extend their functionality.

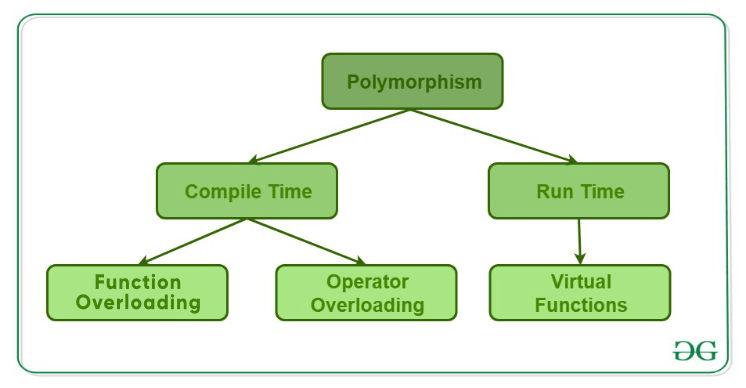


**Polymorphism**

Polymorphism means the ability to take more than one form of the same property. For example, consider an addition operation. It shows a different behavior in different types of data. For two numbers, it will generate a sum. The numbers may integers or float. Thus the addition for integers is different from the addition to floats.

An example is shown in Figure 6, where single function name, i.e. draw can be used to draw different shapes. The name is the same in all the classes but the functionality differs. This is known as function overriding, which is a type of polymorphism. We will discuss it in detail in our next unit.

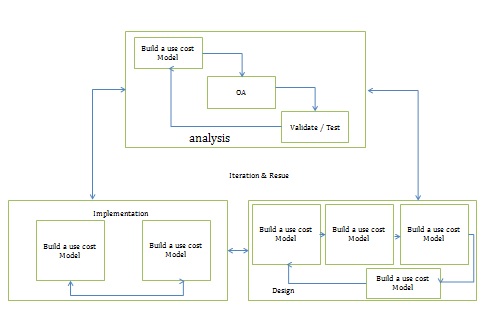
In our example, we also used a function “area” which was inherited by all the three derived classes, i.e. triangle, circle and rectangle. But in the cases of the circle and the triangle, we override the function area because the data types and number of parameters varies.



**Object oriented approach:**

The major factor, which leads to the development of this new approach i.e, Object Oriented approach is to resolve many problems encountered earlier in the procedural approach.

In this approach, we decompose a problem into a number of entities called objects and then build data and functions around these entities. The notion of “Object” comes into the picture. ‘A collection of data and its operations is referred to as an object’. Data is a vital element in the program development. Data is local to an object. This is encapsulated within an object and is not accessible directly from outside the object. These objects know how to interact with another object through the interface (a set of operations). The organization of data and functions in Object Oriented programs is



**The salient features of Object Oriented programming are:**

• More emphasis is on data rather than procedure.

• Programs are modularized into entities called objects.

• Data structures methods characterize the objects of the problem.

• Since the data is not global, there is no question of any operations other than those defined within the object, accessing the data. Therefore, there is no scope of accidental modification of data.

• It is easier to maintain programs. The manner in which an object implements its operations is internal to it. Therefore, any change within the object would not affect external objects. Therefore, systems built using objects are resilient to change.

• Object reusability, which can save many human hours of effort, is possible. An application

developer can use objects like ‘array’, ‘list’, ‘windows’, ‘menus’, ‘event’ and many other components, which were developed by other programmers, in her program and thus reduce program development time.

• It employs bottom-up approach in program design.

**Benefits of OOP:**

OOP offers several benefits to both the program developer and the user. The new technology provides greater programmer productivity, better quality of software and lesser maintenance cost. The major benefits are:

• Ease in division of job: Since it is possible to map objects of the problem domain to those objects in the program, the work can be easily partitioned based on objects.

• Reduce complexity: Software complexity can be easily managed.

• Provide extensibility: Object Oriented systems can be easily upgraded from small to large system.

• Eliminate redundancy: Through inheritance we can eliminate redundant code and extend the use of existing classes.

• Saves development time and increases productivity: Instead of writing code from scratch, solutions can be built by using standard working modules.

• Allows building secure programs: Data hiding principle helps programmer to build secure programs that cannot be accessed by code in other parts of the program.

• Allows designing simpler interfaces: Message passing techniques between objects allows making simpler interface descriptions with external systems.

**Software Architecture with UML Technology:**

What makes software so complex and so difficult to grasp is the fact that the number of information loaded onto a single person is vastly exceeding the capabilities of the human mind. We are not able to handle thousands of pieces of information at any given time as the human short-term memory is quite limited in that respect. The 7+2 rule is a well-known example. This rule states that the human mind can usually only handle 7 distinct things (plus or minus 2) at the same time. Given that restriction, one obvious question that arise is how have we managed even the simplest piece of technology that consists of more (much more) than 9 pieces. The answer to that is probably the concept of modelling. If we encounter too much information ,we consciously and/or unconsciously group this information in some way which makes it easier for us to recall it later on.

The field of Software Engineering uses this human capability of abstraction to create "theories, methods, and tools which are needed to develop software [1]." Those theories, methods, and tools in turn use further abstractions until we have pieces which are small enough for us to comprehend without any further abstraction. Moreover, this need for abstraction is clear in where it emphasizes: "it is not the number of details, as such, that contributes to complexity, but the number of details of which we have to be aware off at the same time." This is the very reason systems engineers have been modelling systems since the inception of the discipline. Specifically, the reasons for modelling systems include among others: definition of the work to be done (requirements), allocation of responsibilities, documentation of the completed system, and communication with the various disciplines who will create, operate, and maintain the system.

Unified Modelling Language has recently been used by Systems Engineers to model systems [12-14]. Many papers and articles have been written on various aspects of systems modelling with UML, most especially Use Cases, but little attention has been focused to develop the system and software architecture of embedded real-time systems. Papers that address development and modelling of system and software architecture do so with a "one size fies all philosophy, by mandating a prescriptive structure that does not address the true needs a many practical system's development. The following of such approaches also may some time result into a badly architectured system which can be a serious handicap to the development of a system.

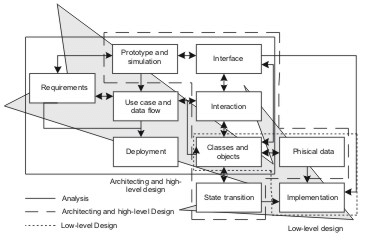
A. Basics of Model In the engineering of software, 'model' is used in many different contexts. But in essence, a model is an abstraction. It abstracts away all kinds of details that are not relevant. The quality of the model is judged from the purpose of the model. There are different model like "system model', 'process model', 'mathematical model", "platform models", "meta model [15] etc used in software engineering for different purpose. A system model is a presentation of particular aspects of (software) systems and their application context that is defined by predefined description techniques based on text, graphics and tables.

**A .Architectural Views and Styles in UML:**

Architecting in UML is represented by different views of UML, which is shown in Fig. 2. It shows the analysis, architecture, and design of a software system, implementation and testing, which often are part of the major development stages (from the developers point of view). The arrows depict the dependencies of the views onto information in other views. Architecture is the appropriate level of abstraction at which rules of a compositional style (i.e., an architectural style) can be exploited and should be elaborated. Doing so results in a set of heuristics that if followed will guarantee a resulting system with certain desirable properties. In many literature UML Views are used to mainly represent the following architecture of a system.

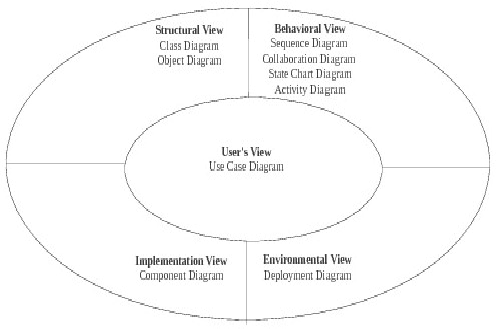
Function Architecture in which the system is structured into a family of sub functions also called services or features.

Logical Component Architecture in which the system is decomposed into sets of components forming architecture.



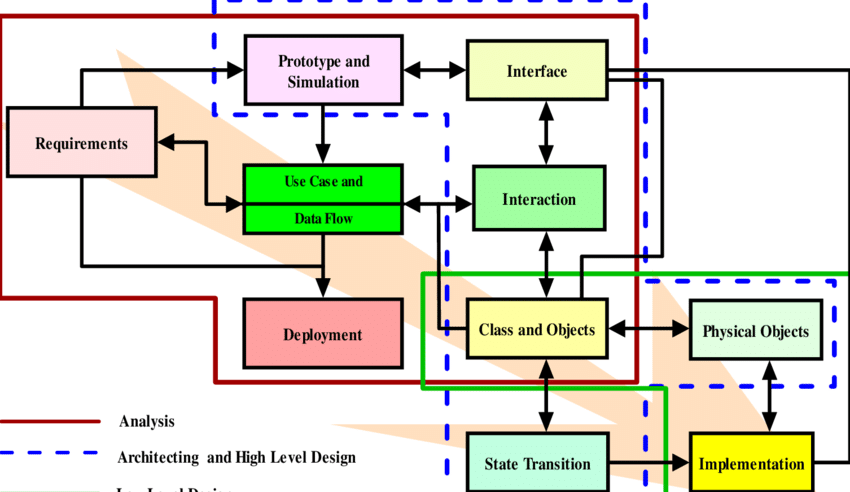
Code architecture: Each state machine is realized by families of modules or classes which form code architecture.

Deployment architecture that maps the components onto a network or hardware systems connected by communication links or bus



**B. Approaches to Architectural Modelling:**

Using UML The suitability of UML as an ADL has been conceived by many as it provides a common platform and notation from architecture through design to implementation. Like any ADLs, as UML also satisfies the requirement of an ADL with the advantage that it supports more views of representation of the system from different perspectives, this paper considers its suitable extension for architectural modeling. The basic promise of software architecture research is that better software systems can result from modeling their important aspects during, and especially early in the development. Choosing which aspects to model and how to evaluate them are two decisions that frame software architecture research. The four-layer metamodeling architecture of UML suggests three possible approaches for modeling software architectures in UML: Using UML as it is, constrain the UML meta model using UML's built in extension mechanisms, and augment the UML meta model to directly support the needed architectural concepts. Each approach has certain potential advantages and disadvantages for forward and reverse engineering, discussed below



**ARCHITECTURAL METHODS**

The modal driven development process incorporates modelling in the development process at suitable stages that plays a defined role for a given modelling technique as shown. For instance the development of the architecture of a system can be obtained by the use of different modelling techniques for the architecture. The different milestone for architecting the system can be achieved by Function Architecture, Logical Component Architecture, Code architecture etc which give the essential structures of a software system in its different phases using architecting methods as described in the following subsection.

Artifacts Driven In this method of architectural design, the starting point is the textual requirements that are used to identify the artifacts types in the process and procedure to the objectives. The identified artifacts are grouped into subsystems to represent the architectural components and subsequently, the relation between these subsystems are defines elaborate the architecture and architectural boundaries. Although the methodology is relatively simpler, however, obstacles often arise as textual requirements are imprecise and less useful as a source

of architectural abstraction.

Domain Driven Most software systems can be classified according to the business area and the kind of tasks they support, eg, airline reservation systems, medical record systems, portfolio management systems, order processing systems, inventory management systems, etc. Similarly, we can also classify parts of software systems according to their functionality, e.g. database systems synchronization packages, work flow systems, GUI libraries, numerical code libraries, etc. We refer to areas organized around classes of systems or parts of systems as domains. Obviously, specific systems or components within a domain share many characteristics since they also share many requirements. Therefore, an organization, which has built a number of systems or components in a particular domain, can take advantage of the acquired knowledge when building subsequent systems or components in the same domain. By capturing the acquired domain knowledge in the form of reusable assets and by reusing these assets in the development of new products, the organization will be able to deliver the new products in a shorter time and at a lower cost. Domain Engineering collecting, organizing, and storing past experience in building systems or parts of systems in a particular domain in the form of reusable assets (i.e. reusable work products), as well as providing an adequate means for reusing these assets ( i.e. retrieval, qualification, dissemination, adaptation, assembly, etc.) when building new systems. The purpose of Domain Analysis is to select and define the domain of focus and collect relevant domain information and integrate it into a coherent domain model.

Pattern Driven Experienced developers find when they approach a new problem to solve, that the situation usually has something in common with a solution they have already either created or seen. The problems are not identical and the identical solution will rarely solve the new problems, but the problems are still similar, so a similar solution will probably work. The "similar solution" generalized and formalized, is called a design pattern. Creating design patterns is a problem of abstracting the similarities of the two problems and the solution so that the generic aspects of the original solution can be applied to the new problem at hand. Design patterns codify design expertise and have been used in an intuitive way as long as design has been a recognized activity. The intentional use of design patterns allows us to think about our designs using a vastly richer vocabulary and to this think critically about the trade offs we make during the design process resulting in better optimized, more robust systems. A design pattern is "a generalized solution to a commonly occurring problem." To be a pattern, the problem must recur often enough to be usefully generalizable. The solution must also be general enough to be applied in a wide set of application domains. If it only applies to a single application

domain, then it is probably an analysis pattern. An analysis pattern is similar to a design pattern but applies to a specific application domain such as finance or aerospace. Analysis patterns define ways for organizing problem-specific object analysis models within a single application domain. For some examples of domain-specific analysis patterns.

Use Case Driven The use case driven approach has been the widely accepted architecting technique used by the OO development community to define the business and application functionality from user point of view; this is why the technique is so popular. In this method. the architecture shows what the user wants to do, and then design and build an application to help the user do this. Although, there are number of variants of the same concept, however, the main theme to derive an architectural representation by gathering the business requirements in enough details such that the risk to the project is minimized remains the same

UML is not a programming vernacular yet rather instruments can be utilized to make code in different tongues utilizing UML graphs. UML has an incite relationship with question composed examination and outline, UML expect a fundamental part in portraying trade viewpoints of a structure

**Use case Diagram:**

The use case graph is for demonstrating the direct of the structure. This chart contains the course of action of use cases, performing pros and their relationship. This chart might be utilized to address the static perspective of the structure. The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart ) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows

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

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.

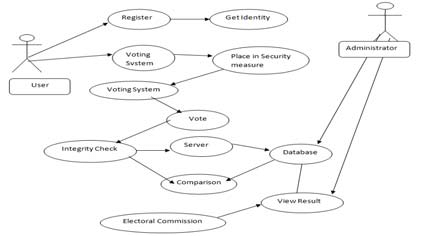
We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

* Functionalities to be represented as use case
* Relationships among the use cases and actors.
* Actors

In the below use case diagram . use cases are used to register, voting system, Get identity, Place in security measure, Database, View results, Server, Integrity check, comparison, Election commission and the actors are the User, Administrator who involved in the below use case diagram

**Use case diagram for online voting system:**



**Class Diagram:**

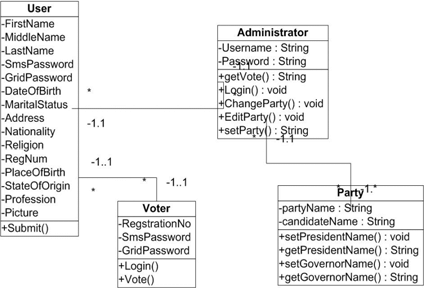
The class graph is the most normally pulled in layout UML. It addresses the static course of action perspective of the structure. It solidifies the strategy of classes, interfaces, joint attempts and their affiliations.

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

* Analysis and design of the static view of an application
* Describe responsibilities of a system.
* Base for component and deployment diagrams.
* Forward and reverse engineering



In the above class diagram, the relationship that is dependence between each one of the classes is sketched out, Additionally, even the operations performed in each and every class is similarly appeared...

**Sequence Diagram:**

This is a cooperation design which tends to the time requesting of messages. It includes set of parts and the messages sent and gotten by the instance of parts. This chart is utilized to address the dynamic perspective of the structure.

A succession outline indicates question communications masterminded in time arrangement. in the above graph, there are five articles cooperating with each other. Each protest has a vertical dashed line which speaks to the presence of a question over some undefined time frame. This graph has additionally a tall, thin rectangle which is called center of control that demonstrates the timeframe amid which a protest is playing out an activity, either specifically or through a subordinate system.

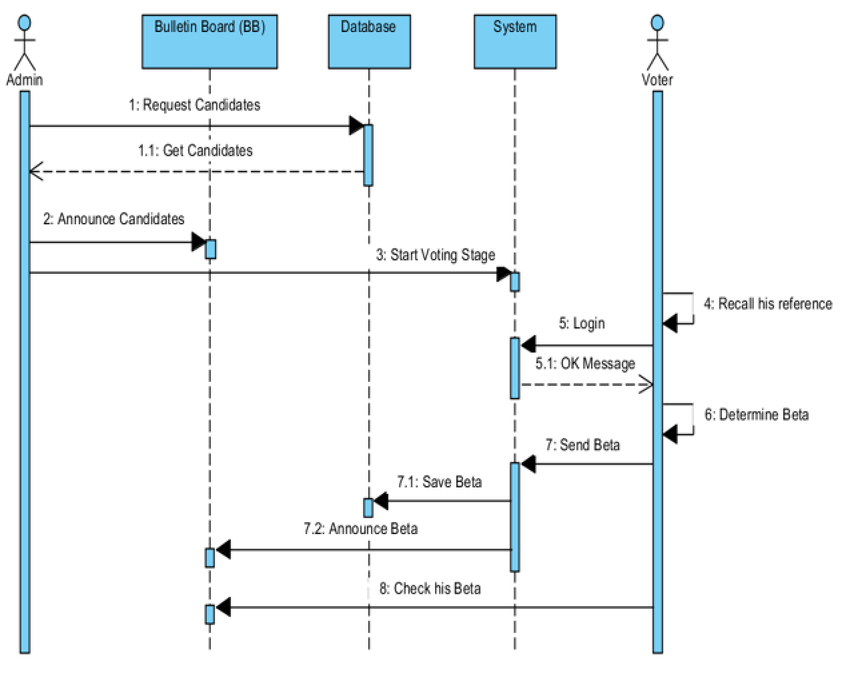
The following scenarios are ideal for using a sequence diagram:

**Usage scenario:** A usage scenario is a diagram of how your system could potentially be used. It's a great way to make sure that you have worked through the logic of every usage scenario for the system.

**Method logic:** Just as you might use a UML sequence diagram to explore the logic of a use case, you can use it to explore the logic of any function, procedure, or complex process.

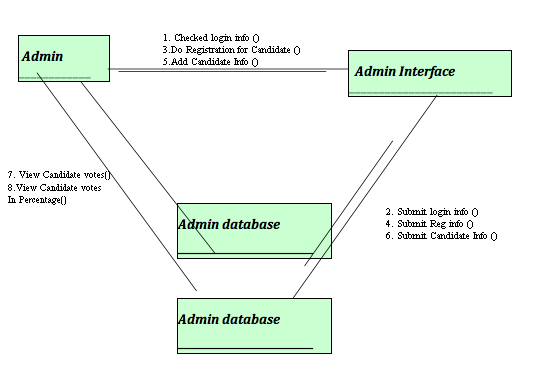
**Service logic:** If you consider a service to be a high-level method used by different clients, a sequence diagram is an ideal way to map that out.

**Sequence diagram Visio** - Any sequence diagram that you create with Visio can also be uploaded into Lucid chart. Lucid chart supports .vsd and .vdx file import and is a great Microsoft Visio alternative. Almost all of the images you see in the UML section of this site were generated using Lucid chart.



In the above sequence diagram messages passed from one object to other objects which starts from the object Admin announces the voting day on that the admin send a request a message for the candidates to login to start their vote. After that voter will login with his username and password then voter enters into candidates page at there voter will make their voter for their candidate in this list of candidates. Once voter submit the information of their voting information is stored in the database.

**Collaboration Diagram:**



This is a support format, which tends to the principal relationship of articles that send and get messages. It incorporates set of parts, connectors that interface the parts and the messages sent and get by those parts. This graph is utilized to address the dynamic perspective of the framework.

The joint effort outline contains articles, way and arrangement number. In the above graph, there are five questions specifically customer, client, framework, Python and server. These items are connected to each other utilizing a way. A succession number show the time request

**State chart Diagram:**

The state graph contains the game-plan of states, occasions and exercises. This graph is noteworthy for tending to the lead of the interface, class and made effort. The key centralization of state outline is to show the occasion sort out lead of the request. The state follows diagram the dynamic perspective of the framework.

A state outline graph contains two components called states and progress. States speak to

circumstances amid the life of a question. We can without much of a stretch outline a state in Smart Draw by utilizing a rectangle with adjusted corners. Change is a strong bolt speaks to the way between various conditions of a question. Name the change with the occasion that activated it and the activity those outcomes from it. State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

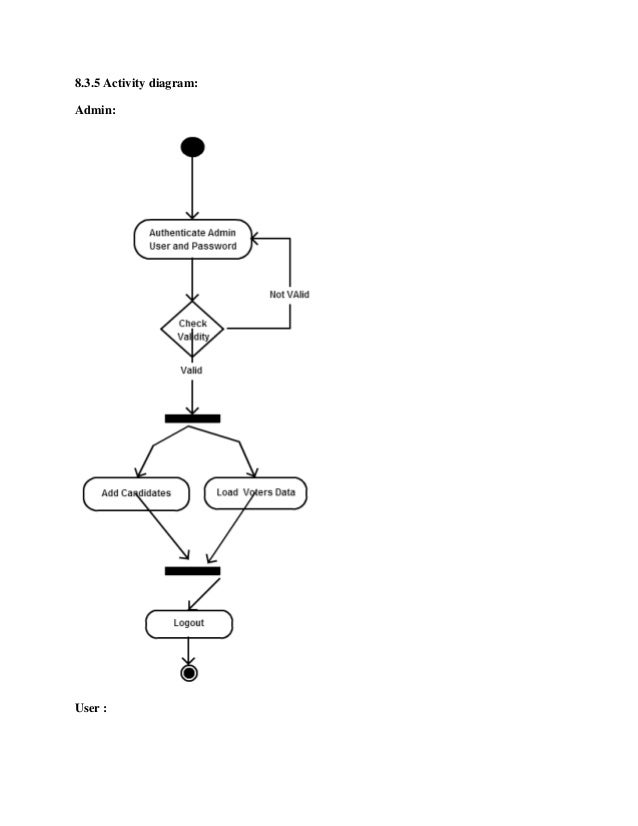
State chart diagram describes the flow of control from one state to another state. States are defined as a condition in w of State chart object exists when some event is triggered. The most important purpose creation to termination.

Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams –

* To model the dynamic aspect of a system.
* To model the life time of a reactive system
* To describe different states of an object during its life time.
* Define a state machine to model the states of an object.

**Statechart Diagram:**



**Component Diagram :**

The imperative portion of part format is segment. This diagram demonstrates within parts, connectors and ports that understand the piece. Precisely when section is instantiated, duplicates of inside parts are besides instantiated.

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.

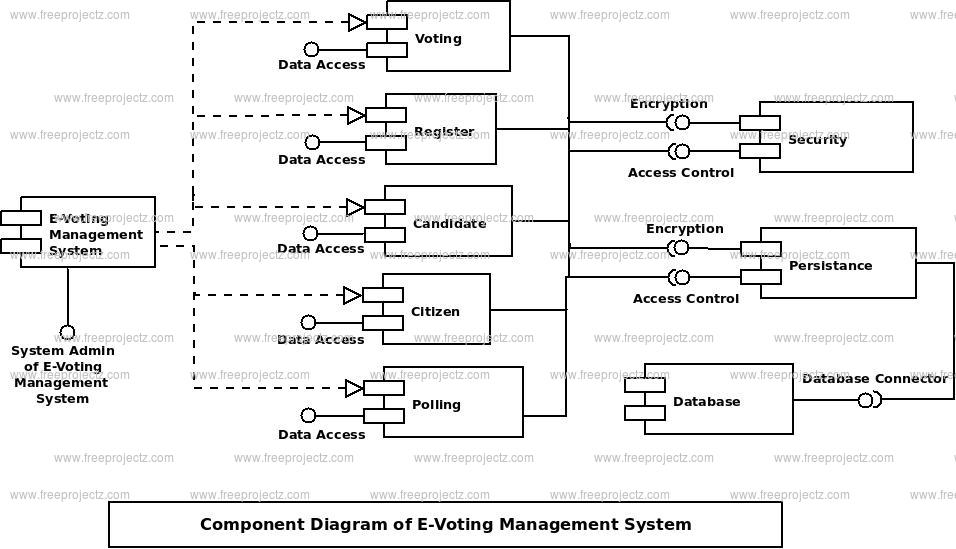
Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.

A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

* Visualize the components of a system.
* Construct executables by using forward and reverse engineering.
* Describe the organization and relationships of the components.



A part outline is spoken to utilizing segment. A part is a physical building piece of the framework. It is spoken to as a rectangle with tab. Part outline portrays the inward handling of the venture. The information sent to the Python where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

**Deployment Diagram:**

The fundamental fragment in game-plan layout is a middle point. The strategy of focus focuses

and their relationship with other is tended to utilizing sending plot. The sending outline is identified with the area diagram, that is one focus purpose obviously of activity format frequently includes no short of what one sections. This outline is in like way critical for tending to the static perspective of the framework.

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

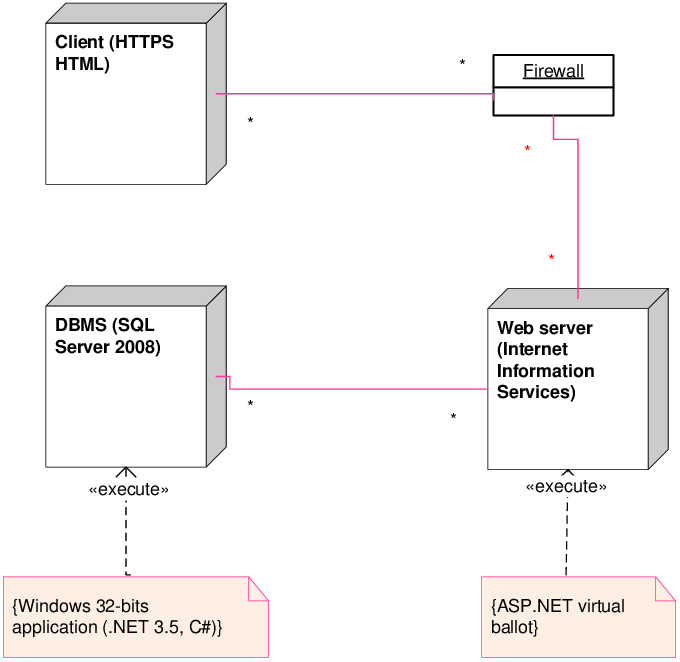
UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

Most of the UMI. diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as –

* Visualize the hardware topology of a system.
* Describe the hardware components used to deploy software components.
* Describe the runtime processing nodes.

From the below Deployment diagram we can see there is a client and web server and DBMS server which are protected by the firewall. We can execute the our project with windows 32 and other higher version. For user interface we are using HTML, CSS, Javascript and for back end we are using PHP and Mysql for information storage which we are collecting from the user’s and their related information



**4.3 PROJECT IMPLEMANTATION**

**4.3.1 Introduction to implementation languages in this project**

**HTML:**

HTML is the standard markup language for creating Web pages.

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* The <!DOCTYPE html> declaration defines that this document is an HTML5 document
* The <html> element is the root element of an HTML page
* The <head> element contains meta information about the HTML page
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The <h1> element defines a large heading
* The <p> element defines a paragraph

**CSS**

CSS is the language we use to style a Web page.

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files
* HTML was NEVER intended to contain tags for formatting a web page!
* HTML was created to describe the content of a web page, like:
* <h1>This is a heading</h1>
* <p>This is a paragraph.</p>
* When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process

**Javascript**

**JavaScript** is a lightweight, cross-platform, and interpreted scripting language. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for [**Client-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [**Server-side**](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments. JavaScript contains a standard library of objects, like [**Array**](https://www.geeksforgeeks.org/arrays-in-javascript/), [**Date**](https://www.geeksforgeeks.org/javascript-date-objects/), and [**Math**](https://www.geeksforgeeks.org/javascript-math-object/), and a core set of language elements like **operators**, **control structures**, and **statements**.

* **Client-side:** It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as **mouse clicks**, **form input**, and **page navigation**. Useful libraries for the client-side are [**AngularJS**](https://www.geeksforgeeks.org/introduction-to-angularjs/), [**ReactJS**](https://www.geeksforgeeks.org/react-js-introduction-working/), **VueJS** and so many others.
* **Server-side:** It supplies objects relevant to running JavaScript on a server. Like if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is [**node.js**](https://www.geeksforgeeks.org/introduction-to-nodejs/).

JavaScript can be added to your HTML file in [two ways](https://www.geeksforgeeks.org/where-to-put-javascript-in-an-html-document/):

* **Internal JS:** We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.
* **External JS:** We can write JavaScript code in other file having an extension .js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

**4.3.2 Code Generation and Validation For Online Voting System**

To create the home page of the screen

<?php include 'includes/session.php'; ?>

<?php include 'includes/header.php'; ?>

<body class="hold-transition skin-blue layout-top-nav">

<div class="wrapper">

<?php include 'includes/navbar.php'; ?>

<div class="content-wrapper">

<div class="container">

<!-- Main content -->

<section class="content">

<?php

$parse = parse\_ini\_file('admin/config.ini', FALSE, INI\_SCANNER\_RAW);

$title = $parse['election\_title'];

?>

<h1 class="page-header text-center title"><b>

<?php echo strtoupper($title); ?></b></h1>

<div class="row">

<div class="col-sm-10 col-sm-offset-1">

<?php

if(isset($\_SESSION['error'])){

?>

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert" aria

hidden="true">&times;</button>

<ul>

<?php

foreach($\_SESSION['error'] as $error){

echo "<li>".$error."</li>";

?>

</ul>

</div>

<?php

unset($\_SESSION['error']);

}

if(isset($\_SESSION['success'])){

echo "

<div class='alert alert-success alert-dismissible'>

<button type='button' class='close' data-dismiss='alert' aria-hidden='true'>&times;</button>

<h4>

<i class='icon fa fa-check'></i>

Success!</h4>

".$\_SESSION['success']."

</div> ";

unset($\_SESSION['success']);

}

?>

<div class="alert alert-danger alert-dismissible" id="alert" style="display:none;">

<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

<span class="message"></span>

</div>

<?php

$sql = "SELECT \* FROM votes WHERE voters\_id = '".$voter['id']."'";

$vquery = $conn->query($sql);

if($vquery->num\_rows > 0)

{

?>

<div class="text-center">

<h3>You have already voted for this election.</h3>

<a href="#view" data-toggle="modal" class="btn btn-flat btn-primary btn-lg">View Ballot</a>

</div>

<?php

}

else{

?>

<!-- Voting Ballot -->

<form method="POST" id="ballotForm" action="submit\_ballot.php">

<?php

include 'includes/slugify.php';

$candidate = '';

$sql = "SELECT \* FROM positions ORDER BY priority ASC";

$query = $conn->query($sql);

while($row = $query->fetch\_assoc()){

$sql = "SELECT \* FROM candidates WHERE position\_id='".$row['id']."'";

$cquery = $conn->query($sql);

while($crow = $cquery->fetch\_assoc()){

$slug = slugify($row['description']);

$checked = '';

if(isset($\_SESSION['post'][$slug])){

$value = $\_SESSION['post'][$slug]; if(is\_array($value)){ foreach($value as $val){

if($val == $crow['id']){ $checked = 'checked';

}

}

}

else{

if($value == $crow['id']){

$checked = 'checked';

}

}

}

$input = ($row['max\_vote'] > 1) ? '<input type="checkbox" class="flat-red '.$slug.'" name="'.$slug."[]".'" value="'.$crow['id'].'" '.$checked.'>' : '<input type="radio" class="flat-red '.$slug.'" name="'.slugify($row['description']).'" value="'.$crow['id'].'" '.$checked.'>';

$image = (!empty($crow['photo'])) ? 'images/'.$crow['photo'] : 'images/profile.jpg';

$candidate .= '

<li>

'.$input.'<button type="button" class="btn btn-primary btn-sm btn-flat clist platform" data-platform="'.$crow['platform'].'" data-fullname="'.$crow['firstname'].' '.$crow['lastname'].'">

<i class="fa fa-search"></i> Platform</button>

<img src="'.$image.'" height="100px" width="100px" class="clist">

<span class="cname clist">'.$crow['firstname'].' '.$crow['lastname'].'</span>

</li>

';

$instruct = ($row['max\_vote'] > 1) ? 'You may select up to '.$row['max\_vote'].' candidates' : 'Select only one candidate';

echo '

<div class="row"> <div class="col-xs-12">

<div class="box box-solid" id="'.$row['id'].'">

<div class="box-header with-border">

<h3 class="box-title"><b>'.$row['description'].'</b></h3>

</div>

<div class="box-body">

<p>'.$instruct.'

<span class="pull-right">

<button type="button" class="btn btn-success btn-sm btn-flat reset" data-desc="'.slugify($row['description']).'">

<i class="fa fa-refresh"></i> Reset</button>

</span>

</p>

<div id="candidate\_list">

<ul>

'.$candidate.'

</ul>

</div>

</div>

</div>

</div>

</div>

';

$candidate = '';

}

?>

<div class="text-center">

<button type="button" class="btn btn-success btn-flat" id="preview"><i class="fa fa-file-text"></i> Preview</button>

<button type="submit" class="btn btn-primary btn-flat" name="vote"><i class="fa fa-check-square-o"></i> Submit</button>

</div>

</form>

<!-- End Voting Ballot -->

<?php

}

?>

</div>

</div>

</section>

</div>

</div>

<?php include 'includes/footer.php'; ?>

<?php include 'includes/ballot\_modal.php'; ?>

</div>

<?php include 'includes/scripts.php'; ?>

<script>

$(function(){

$('.content').iCheck({

checkboxClass: 'icheckbox\_flat-green',

radioClass: 'iradio\_flat-green'

});

$(document).on('click', '.reset', function(e){

e.preventDefault();

var desc = $(this).data('desc');

$('.'+desc).iCheck('uncheck');

});

$(document).on('click', '.platform', function(e){

e.preventDefault();

$('#platform').modal('show');

var platform = $(this).data('platform');

var fullname = $(this).data('fullname');

$('.candidate').html(fullname);

$('#plat\_view').html(platform);

});

$('#preview').click(function(e){

e.preventDefault();

var form = $('#ballotForm').serialize();

if(form == ''){

$('.message').html('You must vote atleast one candidate');

$('#alert').show();

}

else{

$.ajax({

type: 'POST',

url: 'preview.php',

data: form,

dataType: 'json',

success: function(response){

if(response.error){

var errmsg = '';

var messages = response.message;

for (i in messages) {

errmsg += messages[i];

}

$('.message').html(errmsg);

$('#alert').show();

}

else{

$('#preview\_modal').modal('show');

$('#preview\_body').html(response.list);

}

}

});

}

});

});

</script>

</body>

</html>

**To connect the database :**

-- phpMyAdmin SQL Dump

-- version 4.7.9

-- https://www.phpmyadmin.net/

-- Host: 127.0.0.1

-- Generation Time: May 18, 2018 at 07:34 AM

-- Server version: 10.1.31-MariaDB

-- PHP Version: 7.1.15

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time\_zone = "+00:00";

CREATE TABLE `admin` (

`id` int(11) NOT NULL,

`username` varchar(50) NOT NULL,

`password` varchar(60) NOT NULL,

`firstname` varchar(50) NOT NULL,

`lastname` varchar(50) NOT NULL,

`photo` varchar(150) NOT NULL,

`created\_on` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `admin` (`id`, `username`, `password`, `firstname`, `lastname`, `photo`, `created\_on`) VALUES

(1, 'nurhodelta', '$2y$10$fLK8s7ZDnM.1lE7XMP.J6OuPbQ.DPUVKBo7rENnQY7gYq0xAzsKJy', 'Neovic', 'Devierte', 'facebook-profile-image.jpeg', '2018-04-02');

CREATE TABLE `candidates` (

`id` int(11) NOT NULL,

`position\_id` int(11) NOT NULL,

`firstname` varchar(30) NOT NULL,

`lastname` varchar(30) NOT NULL,

`photo` varchar(150) NOT NULL,

`platform` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `positions` (

`id` int(11) NOT NULL,

`description` varchar(50) NOT NULL,

`max\_vote` int(11) NOT NULL,

`priority` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `voters` (

`id` int(11) NOT NULL,

`voters\_id` varchar(15) NOT NULL,

`password` varchar(60) NOT NULL,

`firstname` varchar(30) NOT NULL,

`lastname` varchar(30) NOT NULL,

`photo` varchar(150) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `votes` (

`id` int(11) NOT NULL,

`voters\_id` int(11) NOT NULL,

`candidate\_id` int(11) NOT NULL,

`position\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indexes for dumped tables

-- Indexes for table `admin`

ALTER TABLE `admin`

ADD PRIMARY KEY (`id`);

-- Indexes for table `candidates`

ALTER TABLE `candidates`

ADD PRIMARY KEY (`id`);

-- Indexes for table `positions`

ALTER TABLE `positions`

ADD PRIMARY KEY (`id`);

-- Indexes for table `voters`

ALTER TABLE `voters`

ADD PRIMARY KEY (`id`);

-- Indexes for table `votes`

ALTER TABLE `votes`

ADD PRIMARY KEY (`id`);

-- AUTO\_INCREMENT for dumped tables

-- AUTO\_INCREMENT for table `admin`

ALTER TABLE `admin`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

-- AUTO\_INCREMENT for table `candidates`

ALTER TABLE `candidates`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=18;

-- AUTO\_INCREMENT for table `positions`

ALTER TABLE `positions`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

-- AUTO\_INCREMENT for table `voters`

ALTER TABLE `voters`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;

-- AUTO\_INCREMENT for table `votes`

ALTER TABLE `votes`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=81;

COMMIT;

**To create login page**

<?php

session\_start();

include 'includes/conn.php';

if(isset($\_POST['login'])){

$voter = $\_POST['voter'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM voters WHERE voters\_id = '$voter'";

$query = $conn->query($sql);

if($query->num\_rows < 1){

$\_SESSION['error'] = 'Cannot find voter with the ID';

}

else{

$row = $query->fetch\_assoc();

if(password\_verify($password, $row['password'])){

$\_SESSION['voter'] = $row['id'];

}

else{

$\_SESSION['error'] = 'Incorrect password';

}

}

}

else{

$\_SESSION['error'] = 'Input voter credentials first';

}

header('location: index.php');

?>

**CHAPTER-5**

**TESTING**

**5.1 Description of the testing tools:**

It is otherwise called Functional testing. A product testing strategy whereby the inward workings of the thing being tried are not known by the analyzer. For instance, in a discovery test on a product outline the analyzer just knows the information sources and what the normal results ought to be and not how the program touches base at those yields. The analyzer does not ever inspect the programming code and does not require any further learning of the program other than its determinations. In this system some experiments are produced as information conditions that completely execute every single practical prerequisite for the program. This testing has been utilizations to discover mistakes in the accompanying

classifications:

Incorrect or missing capacities

Interface blunders

Errors in information structure or outside database get to

Performance blunders

Initialization and end blunders.

In this testing just the yield is checked for rightness.

**White Box Testing**

It is otherwise called Glass box, Structural, Clear box and Open box testing. A product testing procedure whereby express learning of the inner workings of the thing being tried are utilized to choose the test information. Not at all like discovery has testing, white box testing utilized particular learning of programming code to inspect yields. The test is precise just if check whether the program veers from its expected objective. White box testing does not represent blunders caused by oversight, and all obvious code should likewise be discernable. For an entire programming examination, both white box and discovery tests are required.

In this the experiments are produced on the rationale of every module by drawing stream diagrams of that module and sensible choices are tried on every one of the cases. It has been

utilizations to produce the experiments in the accompanying cases:

Guarantee that every single freeway have been Executed.

Execute every single intelligent choice on their actual and false Sides.

**Integration Testing**

Coordination testing guarantees that product and subsystems cooperate an entirety. It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime reconciliation is happening, lead tests to reveal blunders related with interfaces. Its Objective is to take unit tried modules and assemble a program structure in view of the recommended outline

Two Approaches of Integration Testing

Non-incremental Integration Testing

Incremental Integration Testing

**System Testing**

Framework testing includes in-house testing the whole framework before conveyance to the client. Its point is to fulfill the client the framework meets all necessities customer's determinations. This testing assesses working of framework from client perspective, with the assistance particular report. It doesn't require any inward learning structure of code

It contains utilitarian and non-useful zones of utilization/item. Framework Testing known as super arrangement of wide range of testing as all the significant of testing are shrouded in it. spite of the fact that attention on sorts of testing may differ the premise of association procedures, course of events and necessities. Framework Testing the genuine testing where you test an item all in all and a module/highlight

**Acceptance Testing**

Acknowledgment testing, a testing method performed to decide if the product framework has met the prerequisite particulars. The principle motivation behind this test is to assess the framework's consistence with the business necessities and check in the event that it is has met the required criteria for conveyance to end clients. It is a pre-conveyance testing in which whole framework is tried at customer's site on genuine information to discover blunders. The acknowledgment test bodies of evidence are executed against the test information or utilizing an acknowledgment test content and afterward the outcomes are contrasted and the normal ones.

The acknowledgment test exercises are completed in stages. Right off the bat, the essential tests are executed, and if the test outcomes are palatable then the execution of more intricate situations are done.

**Test Approach**

A Test approach is the test system usage of a venture, characterizes how testing would be done.

The decision of test methodologies or test technique is a standout amongst the most intense factor in the achievement of the test exertion and the precision of the test designs and gauges

Testing should be possible in two ways

Bottom up approach

Top down approach

**Bottom up Approach**

Testing can be performed beginning from littlest and most reduced level modules and continuing each one in turn. In this approach testing is directed from sub module to primary module, if the fundamental module is not built up a transitory program called DRIVERS is utilized to recreate the principle module. At the point when base level modules are tried consideration swings to those on the following level that utilization the lower level ones they are tried exclusively and afterward connected with the already inspected bring down level modules.

**Top down approach**

In this approach testing is directed from fundamental module to sub module. in the event that the sub module is not built up an impermanent program called STUB is utilized for mimic the sub module. This sort of testing begins from upper level modules. Since the nitty gritty exercises more often than not performed in the lower level schedules are not given stubs are composed. A stub is a module shell called by upper level module and that when achieved legitimately will restore a message to the calling module demonstrating that appropriate association happened.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

***Test strategy and approach***

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**5.2 Other Testing Methodologies:**

**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 user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

**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 the specified format Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways-one is on screen and another in printed format.

**Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

**Preparation of Test Data :**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have they entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, suming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package "Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

A technique for programming testing coordinates the outline of programming experiments into an all around arranged arrangement of steps that outcome in fruitful improvement of the product. The procedure gives a guide that portrays the means to be taken, when, and how much exertion, time, and assets will be required. The procedure joins test arranging. experiment configuration, test execution, and test outcome gathering and assessment. The procedure gives direction to the specialist and an arrangement of points of reference for the chief. Due to time weights, advance must be quantifiable and issues must surface as ahead of schedule as would be prudent.

Keeping in mind the end goal to ensure that the framework does not have blunders, the distinctive levels of testing techniques that are connected at varying periods of programming improvements.

**Unit Testing**

Unit Testing is done on singular modules as they are finished and turned out to be executable. It is restricted just to the planner's prerequisites. It centers testing around the capacity or programming module. It Concentrates on the interior preparing rationale and information structures. It is rearranged when a module is composed with high union.

Reduces the quantity of experiments

Allows mistakes to be all the more effectively anticipated and revealed

**Black Box Integration Testing**

Coordination testing guarantees that product and subsystems cooperate an entirety. It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime reconciliation is happening, lead tests to reveal blunders related with interfaces In Objective is to take unit tried modules and assemble a program structure in view of the commended outline

Two Approaches of Integration Testing

Non-rental Integration Testing

Incremental Integration Testing

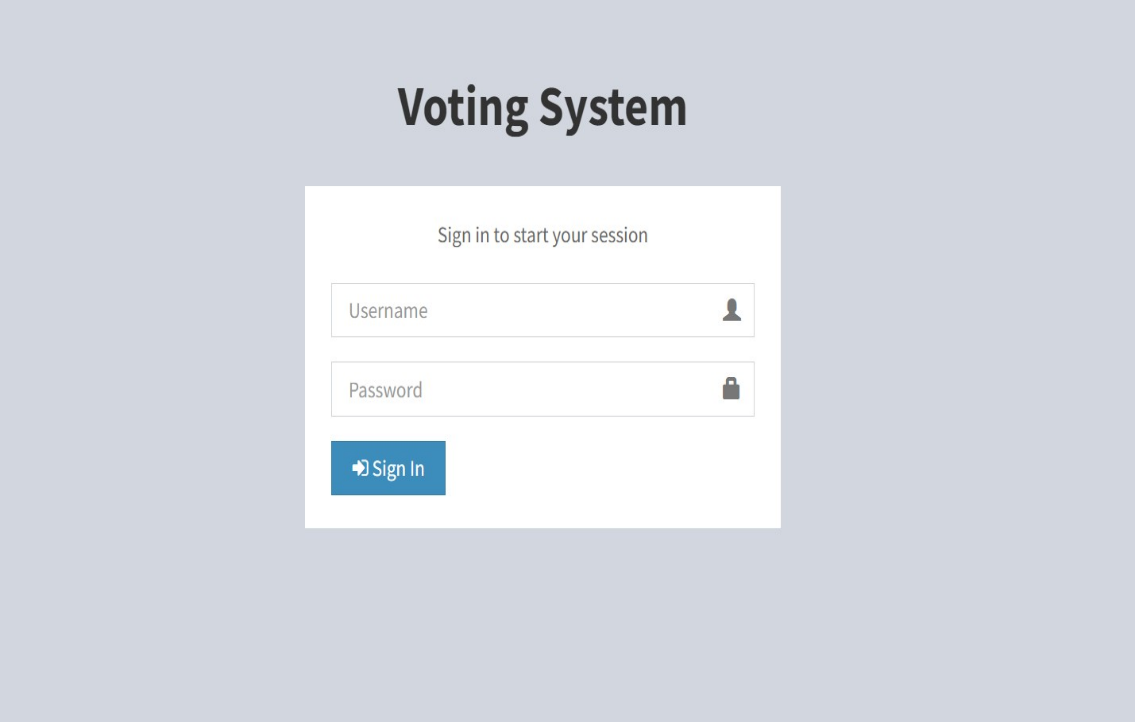
**CHAPTER-6**

**RESULTS**

**6.1 Brief Control flow of the execution screenshots:**

To implement this project we have designed 7 modules at the admin side and 4 modules at the voter side

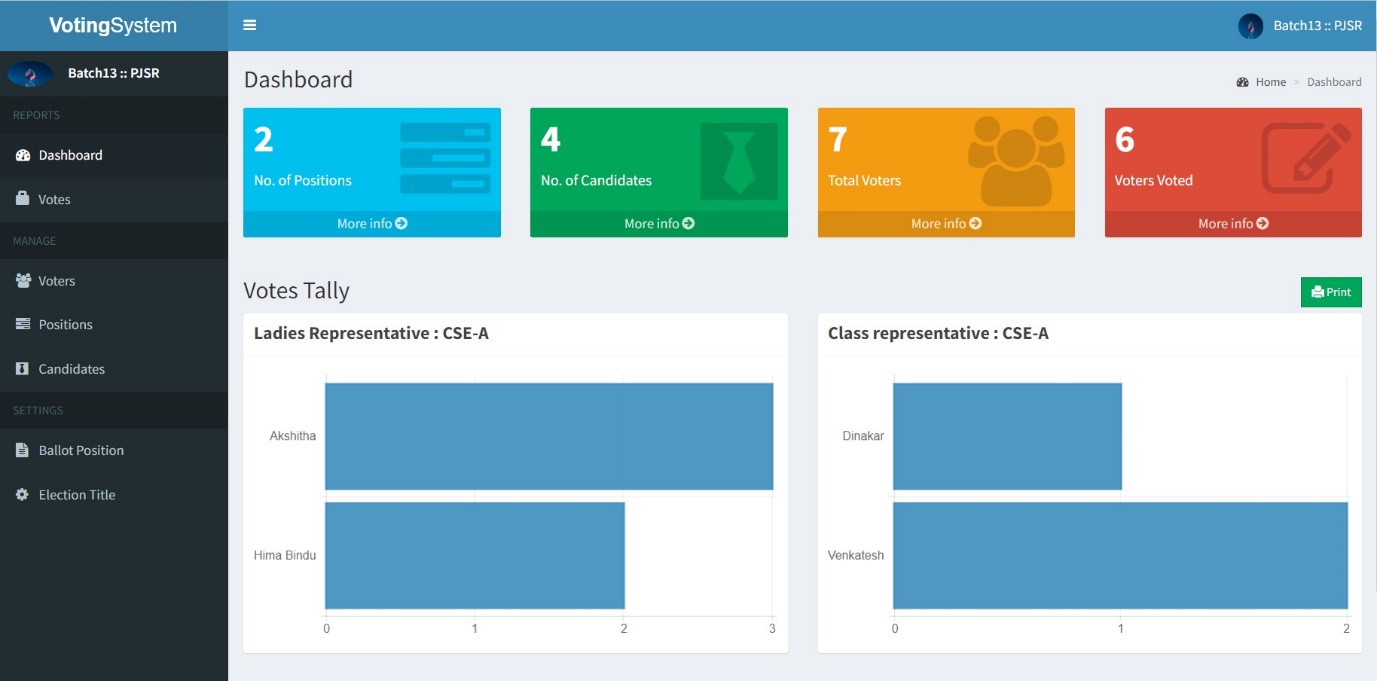
1. **Login page for admin image:**



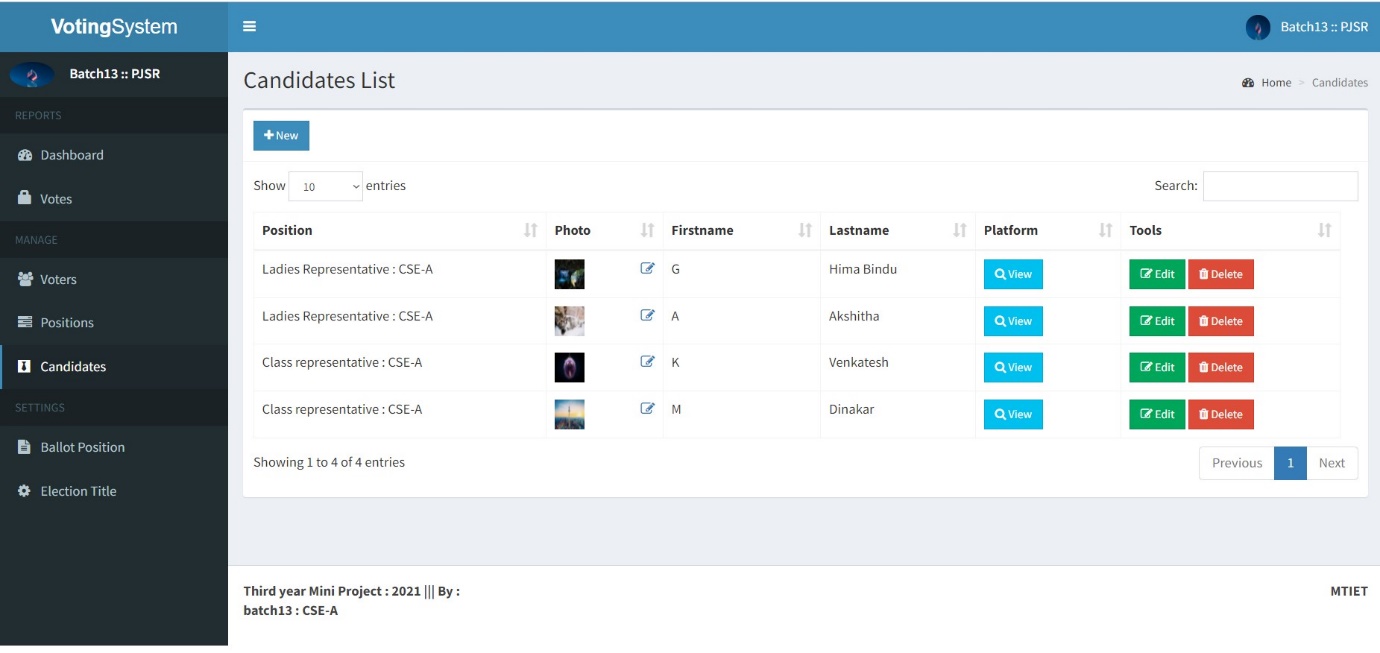
In the above picture shows the login page of the admin to start the session by entering the Username and password of the admin

1. **Admin Template:**

In the Admin template we see there are different options which can choose from these options to conduct the election. First it shows the dashboard of the Admin in the dashboard you see all information like No of Positions means for how many position we are conducting the election. From the above picture we are electing two positions and we can see No of Candidates are present in this elections for a certain position. And we can see total no of voters in that particular election and after that we can see total no of voters voted their votes for their respective candidates. These all information we can see in the Admin’s Dashboard.

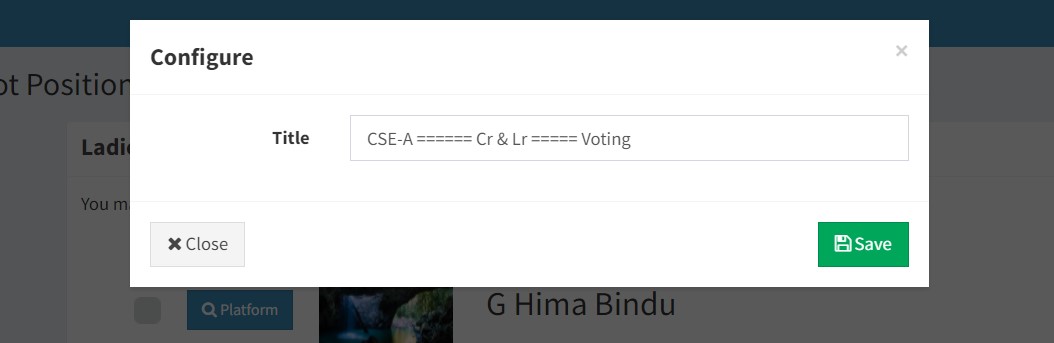


**3.Candidates Page:**



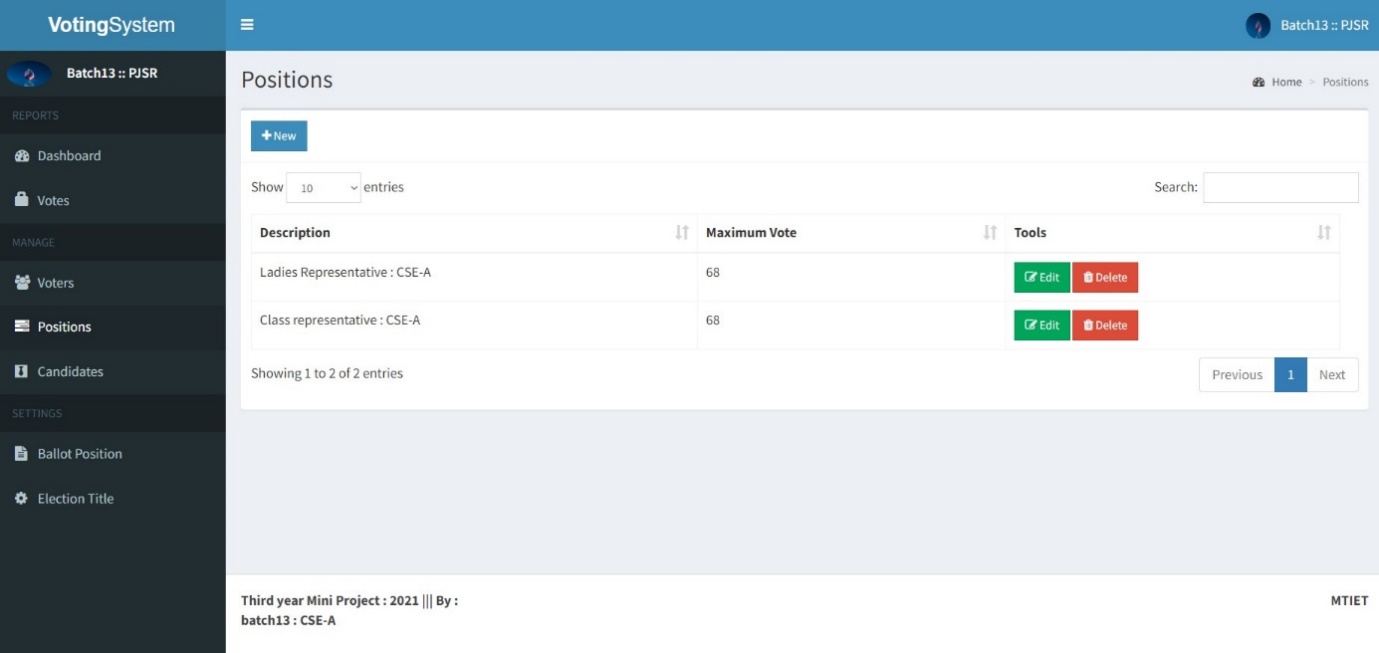
In the above picture shows the candidates page in the admin login. In this Admin can add the candidates for their respective position . we can give the Firstname and Lastname of the respective candidates and we give symbol for each user

**4.Eletion Title page image:**



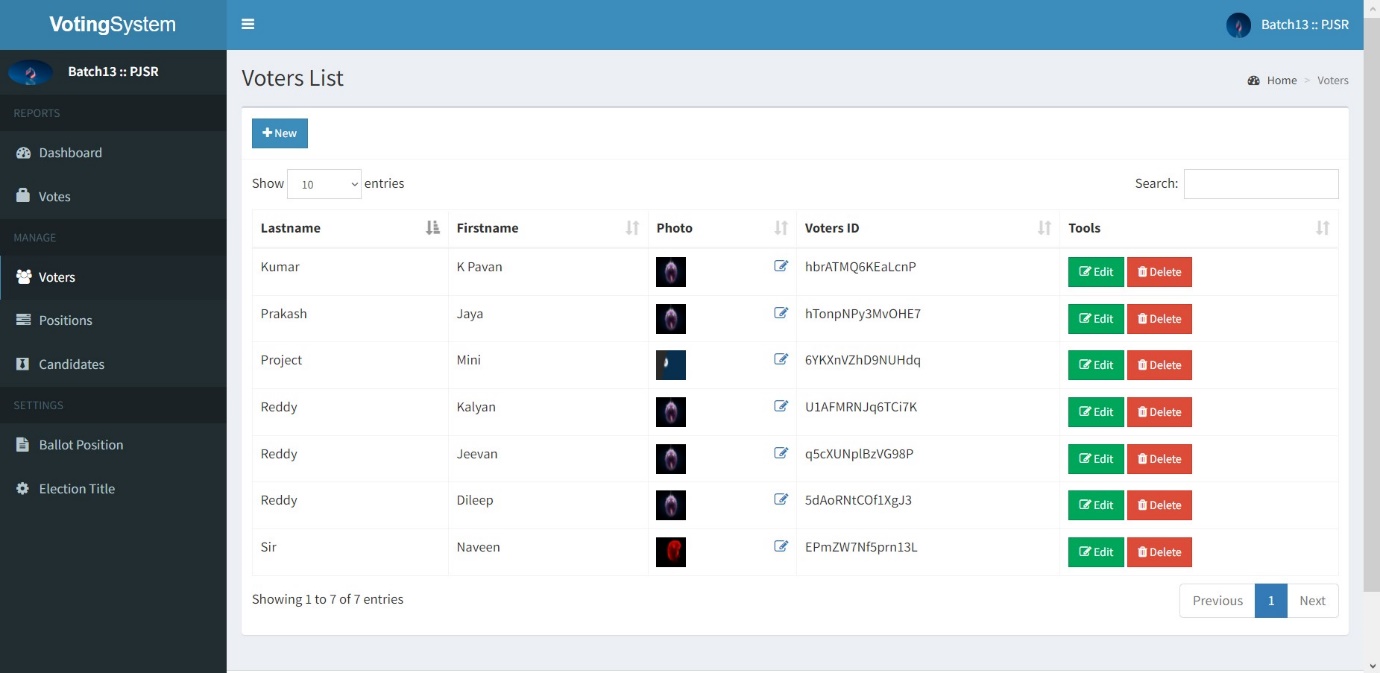
In the above picture represents the title page in the admin login. In this page we can write the title of the elections .

**5. Position page Image:**



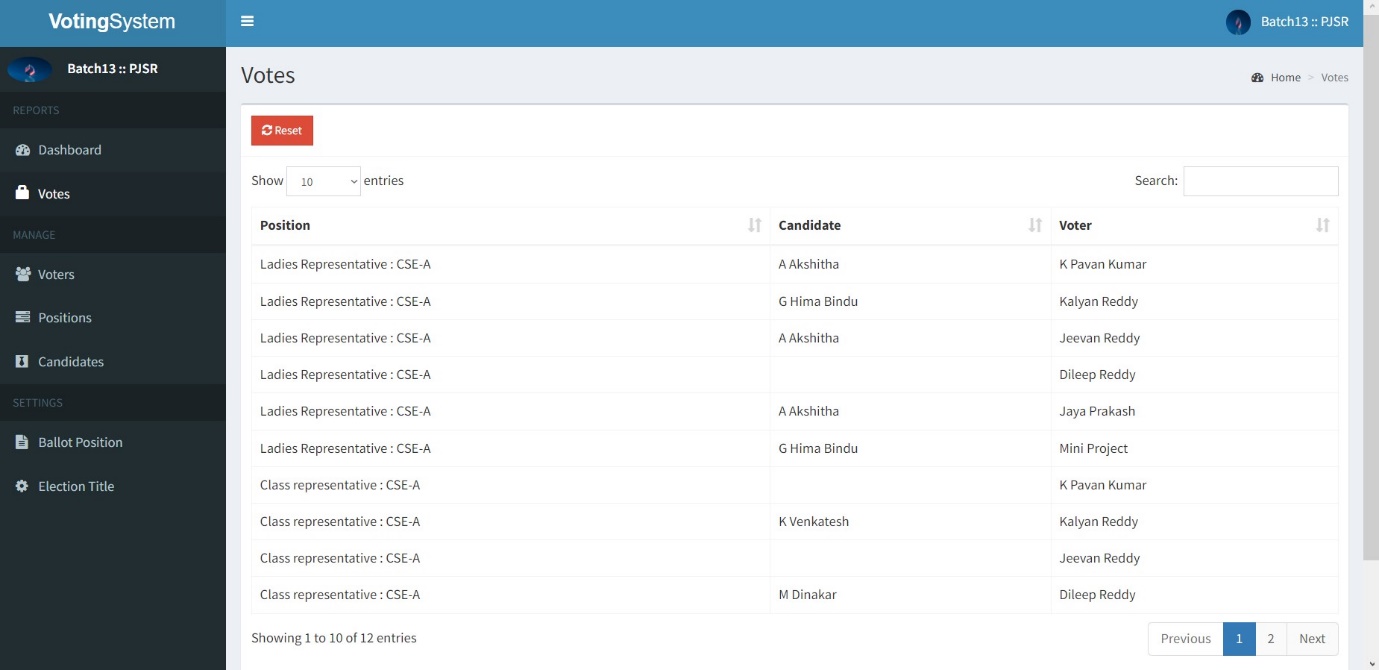
In the above picture shows the position page of the candidate page in admin login page. In this page we can decide the positions of the nominated candidates for their respective position like MLA, MP, Sarpanch, Wardmember etc..

**6. Voter’s page image:**



In the above diagram shows the voter’s list page from the admin’s login. In this page we can add a new voter for the election. Here we will collect the Firstname, Lastname.profile photo, date of birth, gender etc.. By this details Admin create the voter id which id unique of all other users, It is sixteen digit random generated with alphabets and numbers after that Admin will give one default password for the creation of voter id,

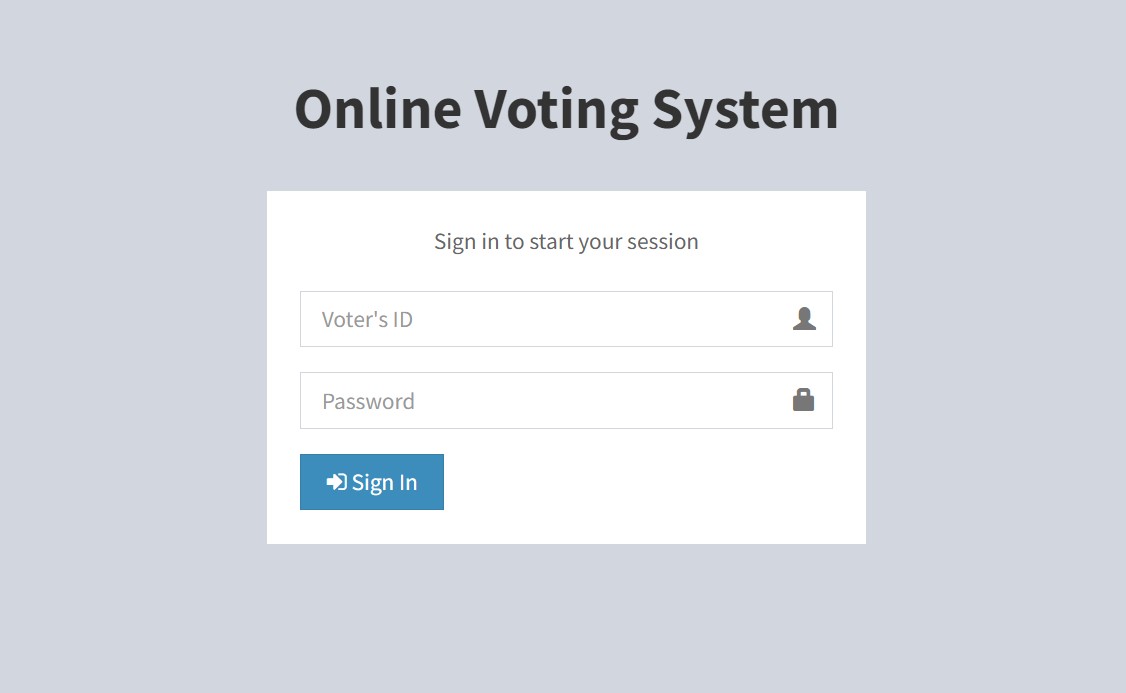
**7. Voters activity list page image:**



From the above diagram it is a voters activity page from the admin login page. In this page we can see the results of the elections

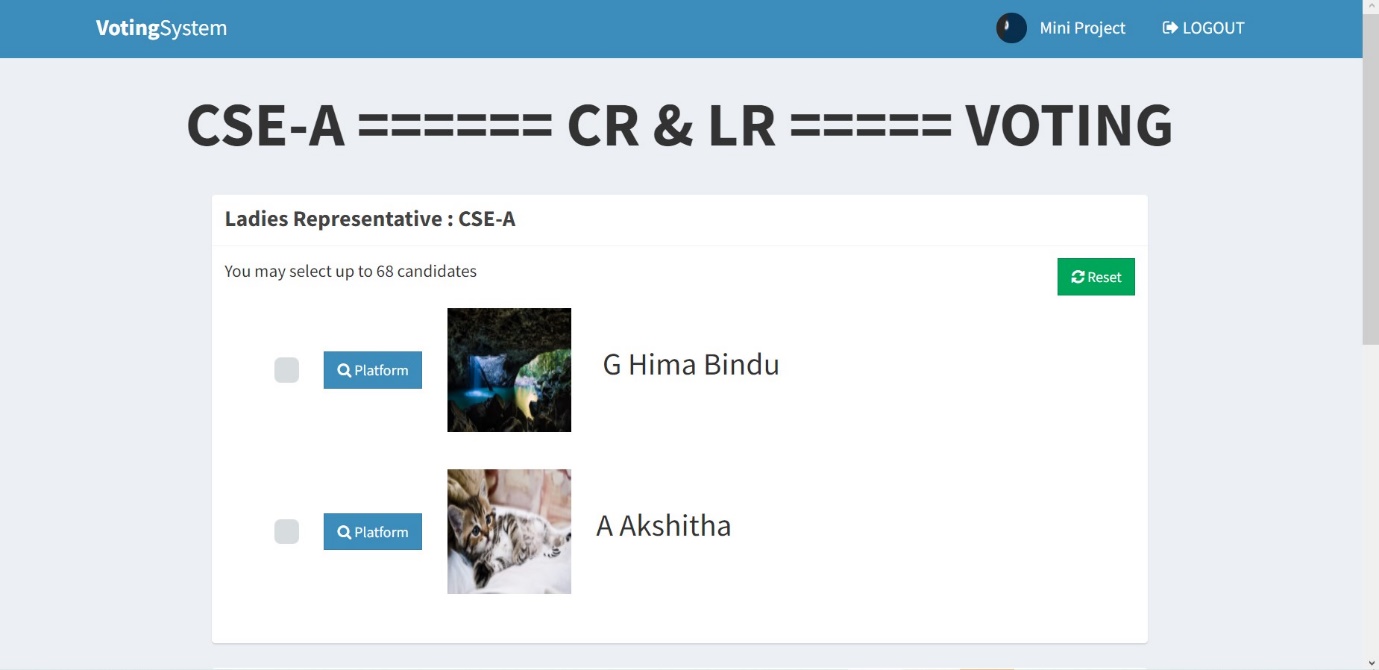
**Output Results:**

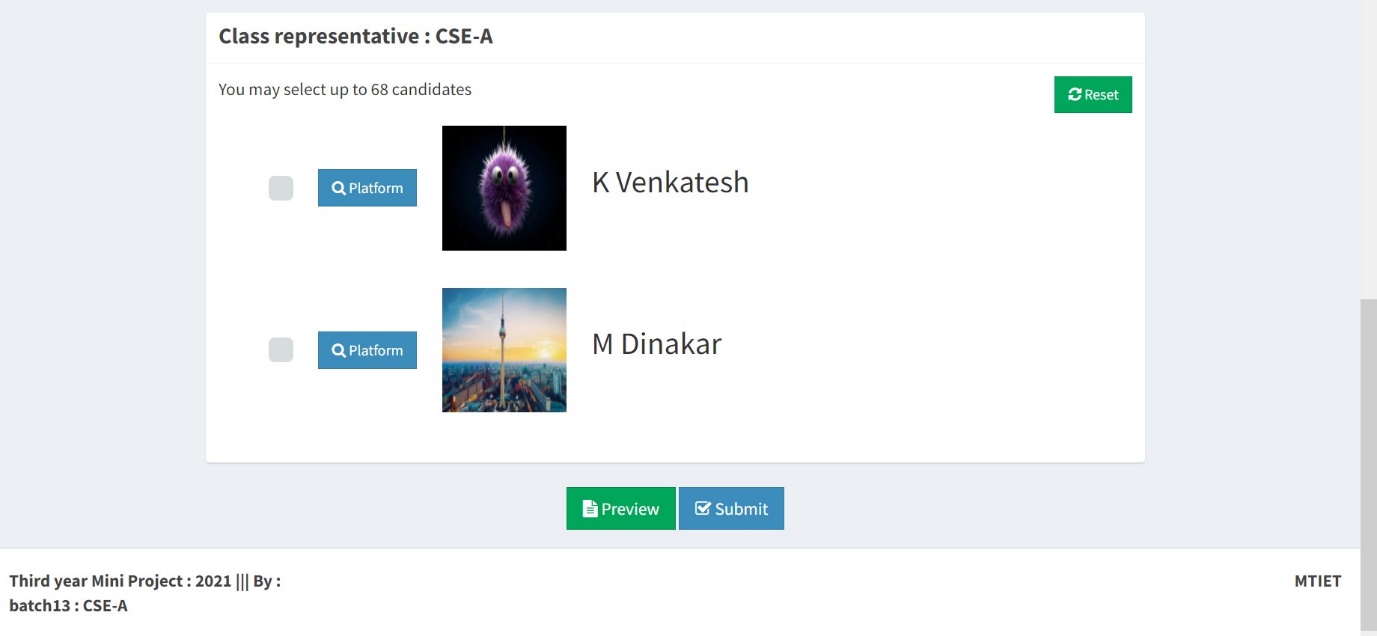
**Voter’s login page image:**



From the above diagram we see that it is a login page for voter. By using the voter’s ID and Password a voter can sign in to voting page.

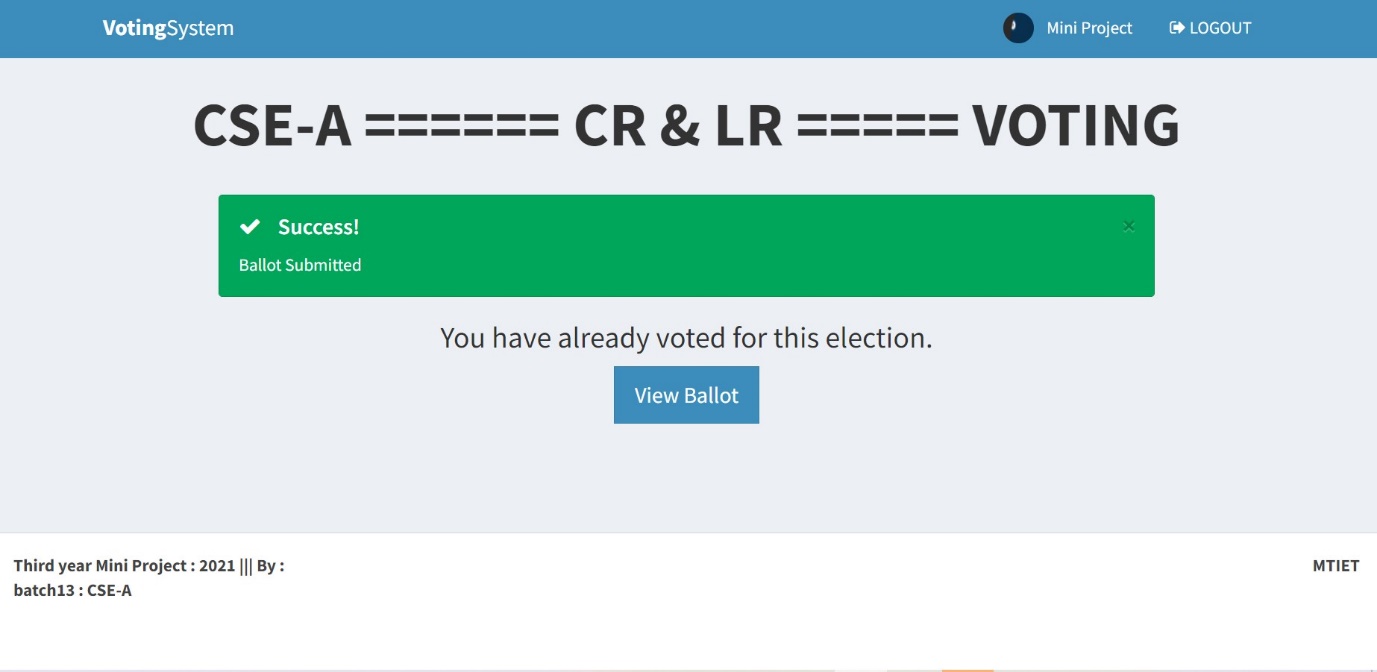
**Voting page image:**





From the above two pictures we can see the voter’s login page . In this voter can select their favourite candidate for each and every position. Before submitting voter can once again check their results with preview button. After that voter will submit their votes,

**Submit Template:**



In the above picture it shows that after submitting the votes by the voter this page will be displayed

**CHAPTER-7**

**CONCLUSIONS**

**7.1 Detailed conclusions:**

By doing this project we were able to bring a new system for online national voting for our country. With the advent of technology and Internet in our day to day life, we were able to offer advanced voting system to voters both in the country and outside through our Online voting system.

**Top 4 Reasons to Move to the Online Voting Platform:**

1. Efficient and Cost Effective

The system offers significant cost benefits over paper elections in a vote to vote comparison. It saves an organization the cost of creating, printing and postage, since everything can be handled electronically. Online elections reduce the use of paper and the amount of work for both the organization, as well as voters.

2. Intelligent

The Online Voting Platform offers intelligent ballots, smart checklist features, vote tallying, tabulation and reporting. These functions are automatic and do not need to be assigned to personnel in-house. Additionally, it allows administrators to create rules on ballots so that voters cannot cast invalid votes, nor do they need to be checked while counting.

3. Save Time and Resources

Online voting system drastically reduces the time required to set up and conduct elections. There is no excess time required to create paper ballots, set up paper ballots for printing, wait for the printing to be completed, or wait for ballots to be filled out and returned.

4. Easy and convenient

The Online Voting Platform offers the easiest and most convenient method for administrators and voters alike. For administrators, the process of setting up a ballot and conducting an election is simple and manageable.

The other advantages that the system offers are:

* Efficient data storage.
* Accuracy, real-time response and user friendliness.
* Intelligent Management

 External voting may be difficult both because of lack of transparency in the process and because of lack of resources to be able to collect first-hand information from a wide geographical area. Observation may therefore in some cases be limited to the overall assessment of the conditions for external voting, or to parts of the processes.

The political conditions for external voting should always be assessed. If the very fact that external voting is permitted is controversial, it is likely that general trust in its implementation will be low.

External voting will always be less transparent than in-country personal voting in polling stations. Voting outside controlled environments will be less easy to observe than voting in controlled areas, and e-voting will produce fewer audit trails (such as ballot papers). Therefore general confidence in the EMB and the election administrators is the first criterion for observers to assess when observing external voting.

Should such confidence not be in place, it is difficult to create it by observing the elections. Even if part of the process can be checked, it is difficult to ensure that the process cannot be manipulated by insiders. However, if there is general trust in the intentions of the election administrators, some parts of the process are possible to check and can be observed. They include:

* the registers of external electors;
* the validation of the voters;
* the content of the ballot material used for postal votes, and the manner in which the return of voting material is checked for correctness and against impersonation;
* the way an e-voting system is procured and validated by the EMB;
* the available audit trails; and
* security measures taken against attacks from outside and against technical failure in the case of electronic voting.

**7.2 Future Scope of the project:**

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

* + - * + Password Changing
        + Fingerprinting
        + Cornea Detection

The password used by the user to vote is provided by the administrator. In the future the user can be given the privilege of changing the password. So it helps to increase the security of the system. The other two methods that can be used are cornea detection and fingerprinting. But here the problem is that it decreases the scope of the platform because these systems need some electronic components to implement. So it will avoid the users privilege to cast the votes at their fingertips. But it can guarantee that fake voting will be impossible. **The OVS system can be used for different elections.**

In this project we had done election fora single loksabha constituency. But this same system can be used in future for conducting different elections like local body, legislative assembly, loksabha elections for many constituencies and wards. For eg: In Kerala there are more than 10000 panchayat wards. The whole election can be conducted using this single platform. The only requirement is that we need to create the whole voters database.

**Results update through SMS.**

In future we can add an SMS query also. ie we wil get the result updates at the time of counting. To receive the SMS we need to register with our mobile number in the site.

