**ONLINE ELECTION SYSTEM SYSTEM**

**ABSTRACT**

Online Election System would have Candidate registration, document verification, auto-generated User ID and pass for candidate and Voters. Admin Login which will be handled by Election Commission .Candidate Login which will be handled By Candidate, Voters will get Unique ID and Password, Using which they can vote for a Candidate only once per Election. The project is beneficial for Election Commission, Voters as the can get to know the candidate background and choose wisely, and even for Candidate. The software system allows the Candidate to login in to their profiles and upload all their details including their previous milestone onto the system.

The admin can check each Candidate details and verify the documents, only after verifying Candidate’s ID and Password will be generated, and can remove faulty accounts. The software system allows Voters to view a list of Candidates in their area. The admin has overall rights over the system and can moderate and delete any details not pertaining to Election Rules.

The online election system is highly developed and the online polling system can be replaced by accurately and directly voting online and immediate results. The online voting system is done by the internet so it can be called the Internet Voting.

**INTRODUCTION**

The main objective of this system is to computerize the entire operation such as information maintenance and the voter voting process. The system is developed as a website for the user compatibility. The user can use this application and complete their need through online. This is an web application voter can voting their vote in this application. Admin can create voter and candidates, candidates who are stand their post then the admin easily found the category wise candidates. Finally the result will be announced by admin The project is aimed to develop by **JAVA** as Front end and **MS SQL SERVER** as Back end. The back end is used to store the information in this system.

**SYSTEM SPECIFICATION**

**HARDWARE SPECFICATION:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 180 GB.
* Floppy Drive : 1.44 Mb.
* Ram : 8 GB.

**SOFTWARE SPECIFICATION:**

* Operating system : Windows 7,8.
* Front End : JAVA and XML
* Back End : SQL Server 2005

**FRONT END: JAVA**

**JAVA**

**Java** is a general-purpose [computer-programming language](https://en.wikipedia.org/wiki/Programming_language" \o "Programming language) that is [concurrent](https://en.wikipedia.org/wiki/Concurrent_computing" \o "Concurrent computing), [class-based](https://en.wikipedia.org/wiki/Class-based_programming" \o "Class-based programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming), and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "[write once, run anywhere](https://en.wikipedia.org/wiki/Write_once,_run_anywhere" \o "Write once, run anywhere)" (WORA),meaning that [compiled](https://en.wikipedia.org/wiki/Compiler" \o "Compiler) Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode" \o "Java bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine" \o "Java virtual machine) (JVM) regardless of [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture" \o "Computer architecture). As of 2016, Java is one of the most [popular programming languages in use](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity" \o "Measuring programming language popularity), particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling" \o "James Gosling), a Canadian, at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) (which has since been [acquired by Oracle Corporation](https://en.wikipedia.org/wiki/Sun_acquisition_by_Oracle" \o "Sun acquisition by Oracle)) and released in 1995 as a core component of Sun Microsystems' [Java platform](https://en.wikipedia.org/wiki/Java_(software_platform)" \o "Java (software platform)). The language derives much of its original features from [SmallTalk](https://en.wikipedia.org/wiki/SmallTalk" \o "SmallTalk), with a [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)" \o "Syntax (programming languages)) similar to [C](https://en.wikipedia.org/wiki/C_(programming_language)" \o "C (programming language)) and [C++](https://en.wikipedia.org/wiki/C++" \o "C++), but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language" \o "Low-level programming language) facilities than either of them.

* JAVA provides services to allow the creation, deployment, and execution of Web Applications and Web Services
* Like ASP, JAVA is a server-side technology
* Web Applications are built using Web Forms. JAVA comes with built-in Web Forms controls, which are responsible for generating the user interface. They mirror typical HTML widgets like text boxes or buttons. If these controls do not fit your needs, you are free to create your own user controls.
* Web Forms are designed to make building web-based applications as easy as building Visual Basic applications

**JAVA Architecture**

JAVA is based on the fundamental architecture of Framework. Visual studio provide a uniform way to combine the various features of this Architecture.

Architecture is explained from bottom to top in the following:

At the bottom of the Architecture is Common Language Runtime .NET Framework common language runtime resides on top of the operating system services. The common language runtime loads and executes code that targets the runtime. This code is therefore called managed code. The runtime gives you, for example, the ability for cross-language integration.

This is evident from its two major design principles:

1. Disconnected Datasets—In ADO.NET, almost all data manipulation is done outside the context of an open database connection.

2. Effortless Data Exchange with XML—Datasets can converse in the universal data format of the Web, namely XML.

The 4th layer of the framework consists of the Windows application model and, in parallel, the Web application model. The Web application model-in the slide presented as JAVA-includes Web Forms and Web Services.

JAVA comes with built-in Web Forms controls, which are responsible for generating the user interface. They mirror typical HTML widgets like text boxes or buttons. If these controls do not fit your needs, you are free to create your own user controls.

Web Services brings you a model to bind different applications over the Internet. This model is based on existing infrastructure and applications and is therefore Standard-based, simple, and adaptable.

Web Services are software solutions delivered via Internet to any device. Today, that means Web browsers on computers, for the most part, but the device-agnostic design of .NET will eliminate this limitation.

The CLR and the .NET Frameworks in general, however, are designed in such a way that code written in one language can not only seamlessly be used by another language. Hence JAVA can be programmed in any of the .NET compatible language whether it is VB.NET, C#, Managed C++ or JScript

### Java JVM and Byte code

Main articles: [Java (software platform)](https://en.wikipedia.org/wiki/Java_(software_platform)" \o "Java (software platform)) and [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine" \o "Java virtual machine)

One design goal of Java is portability, which means that programs written for the Java platform must run similarly on any combination of hardware and operating system with adequate run time support. This is achieved by compiling the Java language code to an intermediate representation called [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode" \o "Java bytecode), instead of directly to architecture-specific [machine code](https://en.wikipedia.org/wiki/Machine_code" \o "Machine code). Java byte code instructions are analogous to machine code, but they are intended to be executed by a [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine" \o "Virtual machine) (VM) written specifically for the host hardware. [End users](https://en.wikipedia.org/wiki/End_user" \o "End user) commonly use a [Java Runtime Environment](https://en.wikipedia.org/wiki/Java_virtual_machine" \o "Java virtual machine) (JRE) installed on their own machine for standalone Java applications, or in a web browser for Java [applets](https://en.wikipedia.org/wiki/Applet" \o "Applet).

Standard libraries provide a generic way to access host-specific features such as graphics, [threading](https://en.wikipedia.org/wiki/Thread_(computing)" \o "Thread (computing)), and [networking](https://en.wikipedia.org/wiki/Computer_network" \o "Computer network).

The use of universal byte code makes porting simple. However, the overhead of interpreting byte code into machine instructions made interpreted programs almost always run more slowly than native [executables](https://en.wikipedia.org/wiki/Executable" \o "Executable). [Just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation" \o "Just-in-time compilation) (JIT) compilers that compile byte-codes to machine code during run time were introduced from an early stage. Java itself is platform-independent and is adapted to the particular platform it is to run on by a [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine" \o "Java virtual machine) for it, which translates the [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode" \o "Java bytecode) into the platform's machine language.

#### Performance

Main article: [Java performance](https://en.wikipedia.org/wiki/Java_performance" \o "Java performance)

Programs written in Java have a reputation for being slower and requiring more memory than those written in C++. However, Java programs' execution speed improved significantly with the introduction of [just-in-time compilation](https://en.wikipedia.org/wiki/Just-in-time_compilation" \o "Just-in-time compilation) in 1997/1998 for [Java 1.1](https://en.wikipedia.org/wiki/Java_version_history" \o "Java version history), the addition of language features supporting better code analysis (such as inner classes, the String Builder class, optional assertions, etc.), and optimization in the Java virtual machine, such as [HotSpot](https://en.wikipedia.org/wiki/HotSpot" \o "HotSpot) becoming the default for Sun's JVM in 2000. With Java 1.5, the performance was improved with the addition of the java.util.concurrent package, including [lock free](https://en.wikipedia.org/wiki/Lock_free" \o "Lock free) implementations of the [ConcurrentMaps](https://en.wikipedia.org/wiki/Java_ConcurrentMap" \o "Java ConcurrentMap) and other multi-core collections, and it was improved further with Java 1.6.

### Non-JVM

Some platforms offer direct hardware support for Java; there are micro controllers that can run Java byte code in hardware instead of a software Java virtual machine, and some [ARM](https://en.wikipedia.org/wiki/ARM_architecture" \o "ARM architecture)-based processors could have hardware support for executing Java byte code through their [Jazelle](https://en.wikipedia.org/wiki/Jazelle" \o "Jazelle) option, though support has mostly been dropped in current implementations of ARM.

### Automatic memory management

Java uses an [automatic garbage collector](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)" \o "Garbage collection (computer science)) to manage memory in the [object lifecycle](https://en.wikipedia.org/wiki/Object_lifetime" \o "Object lifetime). The programmer determines when objects are created, and the Java run time is responsible for recovering the memory once objects are no longer in use. Once no references to an object remain, the [unreachable memory](https://en.wikipedia.org/wiki/Unreachable_memory" \o "Unreachable memory) becomes eligible to be freed automatically by the garbage collector. Something similar to a [memory leak](https://en.wikipedia.org/wiki/Memory_leak" \o "Memory leak) may still occur if a programmer's code holds a reference to an object that is no longer needed, typically when objects that are no longer needed are stored in containers that are still in use. If methods for a non-existent object are called, a "null pointer exception" is thrown.

One of the ideas behind Java's automatic memory management model is that programmers can be spared the burden of having to perform manual memory management. In some languages, memory for the creation of objects is implicitly allocated on the [stack](https://en.wikipedia.org/wiki/Stack_(abstract_data_type)" \o "Stack (abstract data type)) or explicitly allocated and deallocated from the [heap](https://en.wikipedia.org/wiki/Memory_management" \l "DYNAMIC" \o "Memory management). In the latter case, the responsibility of managing memory resides with the programmer. If the program does not deallocate an object, a [memory leak](https://en.wikipedia.org/wiki/Memory_leak" \o "Memory leak) occurs. If the program attempts to access or deallocate memory that has already been deallocated, the result is undefined and difficult to predict, and the program is likely to become unstable or crash. This can be partially remedied by the use of [smart pointers](https://en.wikipedia.org/wiki/Smart_pointer" \o "Smart pointer), but these add overhead and complexity. Note that garbage collection does not prevent "logical" memory leaks, *i.e.*, those where the memory is still referenced but never used.

Garbage collection may happen at any time. Ideally, it will occur when a program is idle. It is guaranteed to be triggered if there is insufficient free memory on the heap to allocate a new object; this can cause a program to stall momentarily. Explicit memory management is not possible in Java.

Java does not support C/C++ style [pointer arithmetic](https://en.wikipedia.org/wiki/Pointer_(computer_programming)" \o "Pointer (computer programming)), where object addresses can be arithmetically manipulated (e.g. by adding or subtracting an offset). This allows the garbage collector to relocate referenced objects and ensures type safety and security.

As in C++ and some other object-oriented languages, variables of Java's [primitive data types](https://en.wikipedia.org/wiki/Primitive_data_type" \o "Primitive data type) are either stored directly in fields (for objects) or on the [stack](https://en.wikipedia.org/wiki/Stack-based_memory_allocation" \o "Stack-based memory allocation) (for methods) rather than on the heap, as is commonly true for non-primitive data types (but see [escape analysis](https://en.wikipedia.org/wiki/Escape_analysis" \o "Escape analysis)). This was a conscious decision by Java's designers for performance reasons.

Java contains multiple types of garbage collectors. By default, HotSpot uses the [parallel scavenge garbage collector](https://en.wikipedia.org/w/index.php?title=Parallel_scavenge_garbage_collector&action=edit&redlink=1" \o "Parallel scavenge garbage collector (page does not exist)).[[57]](https://en.wikipedia.org/wiki/Java_(programming_language)" \l "cite_note-57) However, there are also several other garbage collectors that can be used to manage the heap. For 90% of applications in Java, the [Concurrent Mark-Sweep](https://en.wikipedia.org/wiki/Concurrent_mark_sweep_collector" \o "Concurrent mark sweep collector) (CMS) garbage collector is sufficient.[[58]](https://en.wikipedia.org/wiki/Java_(programming_language)" \l "cite_note-58) Oracle aims to replace CMS with the [Garbage-First collector](https://en.wikipedia.org/wiki/Garbage-First_collector" \o "Garbage-First collector) (G1).

Having solved the memory management problem does not relieve the programmer of the burden of handling properly other kind of resources, like network or database connections, file handles, etc., especially in the presence of exceptions. Paradoxically, the presence of a garbage collector has faded the necessity of having an explicit destructor method in the classes, thus rendering the management of these other resources more difficult!

**Web Services**

* Web services are small units of code built to handle a limited task.
* Web services uses XML based communicating protocols.
* Its operating systems are independent. And programming languages also independent.
* It’s connecting the people, system, and devices.
* Web services use the standard web protocols HTTP, XML, SOAP, WSDL, and UDDI.

**Benefits of Web Services**

* Easy to communicate between the application.
* Easy to reuse existing system.
* Rapid development.
* Easier to distribute information to more consumers.
* It can create new possibilities for many business because it provides an easy way to a large number of consumers.

**Important Features of JAVA**

**Robust database-driven Functionality**

JAVA is purely object-oriented makes it very powerful. JAVA allows developers to develop web applications that interact with the database.

JAVA (ADO stands for Activex Data Objects) is especially launched for JAVA for handling all types of database related queries.

**Faster Web Applications**

JAVA Web applications are faster due to two reasons: compiled code and caching. Earlier, the code was interpreted into machine language when website visitor used to view the page.

Caching is the storing of information in a non-physical media for speeding up the processing for the future use. Web pages that are commonly visited by end-users are cached for a predefined period of time to improve the performance of website.

**Memory Leak and Crash Protection**

Now developers need not bother about error like deadlocks and memory leaks because JAVA can automatically recover from these errors and your website always remains available to the end-users.

Memory leak is a situation in which a program component fails to release the memory when the memory is no longer needed it.

A deadlock is a situation where each out of the two or more processors is waiting for some other process in the group to finish first and thus none of the processors finishes ever.

**Easy Deployment**

Development of web applications in JAVA is quiet easy because the configuration information is of built-in type.

**Multiple development Language support**

Developers can write their code in different .NET languages for example Visual Basic, Visual C#, Visual J#, etc.

**Difference between ASP and JAVA  
ASP:**

1. ASP is Interpreted language based on scripting languages like Jscript or VBScript.  
  
2. ASP has Mixed HTML and coding logic.  
  
3. Limited development and debugging tools available.  
  
4 Limited OOPS support.  
  
5. Limited session and application state management.  
  
6. Poor Error handling system.  
  
7. No in-built support for XML.  
  
8. No fully distributed data source support.

**JAVA:**1. JAVA is supported by compiler and has compiled language support.  
  
2. Separate code and design logic possible.  
  
3. Variety of compilers and tools available including the Visual Studio.NET.  
  
4. Completely Object Oriented.  
  
5. Complete session and application state management.  
  
6. Full proof error handling possible.  
  
7. Full XML Support for easy data exchange.  
  
8. Fully distributed data source support.

**Reason for Selecting the JAVA**

* Web applications created with JAVA are easier to create, debug, and deploy because those tasks can all be performed within a single development environment Visual Studio .NET.
* Executable portions of a Web application compiled so they execute more quickly than interpreted scripts.
* On-the-fly updates of deployed Web applications without restarting the server.
* Access to the .NET Framework, which extends the Windows API.
* Use of the widely known Visual Basic programming language, which has been enhanced to fully support object-oriented programming.
* Introduction of the new Visual C# programming language, which provides a type-safe, object-oriented version of the C programming language.
* Automatic state management for controls on a Web page (called server controls) so that they behave much more like Windows controls.
* The ability to create new, customized server controls from existing controls.
* Built-in security through the Windows server or through other authentication/authorization methods.
* Integration with Microsoft ADO.NET to provide database access and database design tools from within Visual Studio .NET.
* Full support for Extensible Markup Language (XML), cascading style sheets (CSS), and other new and established Web standards.
* Built-in features for caching frequently requested Web pages on the server, localizing content for specific languages and cultures, and detecting browser capabilities.

**BACK END USED**

**Microsoft SQL Server used as back end tool.**

Microsoft SQL Server2005

Microsoft SQL Server is an application used to create computer databases for the Microsoft windows family of server operating system. It provides an environment used to generate databases that can access workstations, the web or other media such as a Personal Digital Assistant (PDA).

Microsoft SQL Server is a probably the most accessible and the most documentation enterprise database environment right now. The following is a list of the features provided in SQL Server 2005.

* User-defined functions
* Indexed views
* Distributed partitioned views
* Cascading RI constraints
* Multiple SQL Server instances
* XML support

User-Defined Functions

SQL Server has always provided the ability to store and execute SQL code routines via stored procedures. In addition, SQL Server has always supplied a number of built-in functions. Functions can be used almost anywhere. An expression can be specified in query.

SQL Server 2005introduces the long-awaited support for user-defined functions. User-defined functions can take zero or more input parameters and return a single value like the system-defined functions, or a table result.

Table valued functions can be used anywhere. Table or view expressions can be used in queries, and they can perform more complex logic than is allowed in a view.

Indexed Views

Views are often used to simplify complex queries, and they can contain joins and aggregate functions. However, in the past, queries against views were resolved to queries against the underlying base table, and any aggregates were recalculated each time you ran a query against the view.

Indexed views can improve performance for the following types of queries:

* Joins and aggregate that process many rows.
* Join and aggregation operations that performed frequently with in many queries.
* Decision support queries that rely on summarized, aggregated data that is infrequently updated.

Distributed partitioned views

SQL server 7.0 provided the ability to create partitioned views using the UNION ALL statement in a view definition. It was limited, however, in that all the tables had to reside within the same SQL Server where the view was defined. SQL Server 2005expands the ability to create partitioned views by allowing to horizontally partition tables across multiple SQL Servers.

The features helps to scale out one database server to multiple database servers, while making the data appear as if it comes from a single table on a single SQL Server. In addition, partitioned views are now able to be updated.

Cascading RI Constraints

In previous versions of SQL Server, referential integrity (RI) constraints were restrictive only. If an insert, update, or delete operation violated referential integrity, it was aborted with an error message.

SQL Server 2005 provides the ability to specify the action to take when a column referenced by a foreign key constrain is updated or deleted. You can still abort the update or delete if related foreign key records exist by specifying the NO ACTION option, or you can specify the new CASCADE option, which will cascade the update or delete operation to the related foreign key records.

Multiple SQL Server Instances

Previous versions of SQL Server supported the running of only a single instance of SQL Server at a time on a computer.

Running multiple instance or multiple versions of SQL Server required switching back and forth between the instances, requiring changes in the Windows registry. (The SQL Server Switch provides with 7.0 switches between 7.0 and 6.5 performed the registry changes for us.)

SQL Server 2005 provides support for running multiple instances of SQL Server on the same system. This allows you to simultaneously run one instance of SQL Server 6.5 or 7.0 along with one or more instances of SQL Server 2005.

Each SQL Server instance runs independently of the others and has its own set of system and user databases, security configuration and so on. Applications can connect to the different instance in the same way they connect to different SQL Servers on different machine.

XML Support

Extensible Markup Language has became a standard in web-related programming to describe the contents of a set of data and how the data should be output or displayed on a web page. XML, like HTML, is derived from the Standard Generalize Markup Language (SGML).

When linking a web application to SQL Server, a translation needs to take place from the result set returned from SQL Server to a format that can be understood and displayed web applications. Previously, this translation needed to be done in a client application.

SQL Server 2005 provides native support XML. These new features provide the ability to do the following:

Return query result sets directly in XML format.

* Retrieve data from an XML document as it were a SQL Server table.
* Access SQL Server through a URL using HTTP. Through Internet Information Services (IIS), you can define a virtual root that gives you HTTP access to the data and XML functionality of SQL Server 2000.

The features of the SQL Server 2005

It is a client-server architecture and not shared-file application as access.

* Symmetric Multiprocessing (SMP) supports up to 32 simultaneous processors.
* It can have database up to 1 terabyte (1024 GB) in size.
* It can handle up to 32,767 simultaneous user connections.
* It provides data integrity, data recovery and functionalities that are transparent to the developer.
* SQL Server optimizes network resources. Only the data requested by the client needs to be sent across the network.
* Security can be enforced at ones central location.

**Benefits of SQL**

* SQL has become the database language of choice because it is flexible, powerful and easy to learn.
* SQL is a non-procedural language.
* Process sets of records rather than just one at a time and provides automatic navigation to the data.
* SQL provides commands for a variety of tasks including: querying data, creating, updating and replacing and inserting, updating and deleting rows.
* All major relational database management systems support SQL thus one can transfer all the skills gained with SQL from one RDBMS to another.

**SYSTEM STUDY**

**EXISTING SYSTEM:**

The voter gets a blank ballot and use a pen or a marker to indicate he want to vote for which candidate. Hand-counted ballots is a time and labor consuming process, but it is easy to manufacture paper ballots and the ballots can be retained for verifying, this type is still the most common way to vote.

**DISADVANTAGES:**

* Chances of hacking candidate or voters account.
* Every voter does not have pc or net connection, so voters can even go in polling booth and vote online there.

**PROPOSED SYSTEM:**

In this system voter can voting their candidate via online. With the help of that system we can voting easily. User can’t voting more than one time. In this system helps to counting the vote very easy.

**ADVANTAGES OF THE PROPOSED SYSTEM:**

* Fast and easy way of conducting Election.
* Voters can view background of each Candidate.
* Candidate can present themselves against voters.
* Admin can verify the documents and details of Candidate.
* System Generated Unique ID and Password gives more Secure Logins

**SYSTEM DESIGN AND DEVELOPMENT**

**INPUT DESIGN**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:’

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES**

* Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
* It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
* When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
* will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**OUTPUT DESIGN**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**MODULES**

1. Voter Registration
2. Candidate Registration
3. Pooling vote

* Election result

**MODULES DESCRIPTION:**

**Voter Registration**

Voter can created by admin in this module. Once voter created then voter username and password will be generated to that user. Which will helpful to login to voter module.

**Candidate Registration**

Candidate can created by admin. Who are all stand the election the member has created in this module.

**Pooling vote**

Voter can pooling their vote in this module. Once voter pooling their vote can’t vote more than one time.

**Election Result**

Election result will be declared after the election will completed. Which is the main module for finding the winner.

**SYSTEM IMPLEMENTATION**

When the initial design was done for the system, the client was consulted for the acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about the working of the system. The aim of the system illustration was to identify any malfunction of the system.

After the management of the system was approved the system implemented in the concern, initially the system was run parallel with existing manual system. The system has been tested with live data and has proved to be error free and user friendly.

Implementation is the process of converting a new or revised system design into an operational one when the initial design was done by the system; a demonstration was given to the end user about the working system.

This process is uses to verify and identify any logical mess working of the system by feeding various combinations of test data. After the approval of the system by both end user and management the system was implemented.

System implementation is made up of many activities. The six major activities are as follows.

**1. CODING**

Coding is the process of whereby the physical design specifications created by the analysis team turned into working computer code by the programming team. A design code may be a tool which helps ensure that the aspiration for quality and quantity for customers and their requirements, particularly for large scale projects, sought by the water agency Design pattern are documented tried and tested solutions for recurring problems in a given context. So basically you have a problem context and the proposed solution for the same.

**2. INSTALLATION**

Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, and documentation and work procedures to those consistent with the new system.

**3. DOCUMENTATION**

Documentation is descriptive information that describes the use and operation of the system. The user guide is provided to the end user as the student and administrator. The documentation part contains the details as follows,

User requirement and water agency details administration has been made online. Any customer can request their water requirement details through online and also use of documentation, they can view the purpose of each purpose, The admin could verify the authentication of the users, users requirements and need to take delivery process, thus the documentation is made of full view of project thus it gives the guideline to study the project and how to execute also.

**USER TRAINING AND SUPPORT**

The software is installed at the deployment environment, the developer will give training to the end user of the regional transport officer and police admin officer in that software. The goal of an end user training program is to produce a motivated user who has the skills needed to apply what has been to apply what has been learned to perform the job related task. The following are the instruction which is specified the handling and un-handling events in the application,

* The authenticated user of admin and office workers only login in the application with authorized username and password.
* Don’t make user waste their time to come straight to the water agency or make a phone call.
* It can easily track through online by the user.
* Very user friendliness software

**INSTALLATION STEPS**

Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, and documentation and work procedures to those consistent with the new system. The following steps are used to install the application in the end users’ machine.

Step 1: A folder named ‘ONLINE ELECTION SYSTEM(main folder) is created in ‘D’ drive; but in any drive with any name the folder can created since the coding inside the application creates path dynamically in places wherever required.

Step 2: Two folders named ‘bin’ and ‘obj’ is created inside the main folder. Inside those two folders, a folder with name ‘Debug’ is created. All the files in the development machine’s Debug folder content are copied into this folder.

Step 3: The database folder is also copied to the main folder which contains the files ‘ONLINE ELECTION SYSTEM \_Data.MDF’ and ‘ONLINE ELECTION SYSTEM \_Log.LDF’.

Step 4: In Start->Programs->Microsoft SQL Server->Enterprise Manager Option is selected.

Step 5: The left side tree ‘Console Root’ is expanded until the local SQL Server instance node is open.

Step 6: Inside which the folder with name ‘Database’ is right clicked and All Tasks->Attach Database option is selected.

Step 7: The ellipsis button right side to the text box (for path) is clicked and the database main file ‘ONLINE ELECTION SYSTEM\_Data.MDF’ is selected. Then OK button is clicked.

Step 8: After the message display about the successful database attachment, the executable file inside the D:\ ONLINE ELECTION SYSTEM \bin\debug folder is double clicked and checked with one of the username and password input.

Step 9: The database is connected and so the main form appeared. This action confirms the application is running and as well as database connection is proper.

Step 10: The further options are worked out and checked so that all the records are appended in all the tables correctly.

**SYSTEM TESTING AND IMPLEMENTATION**

**SYSTEM TESTING**

System testing is the process of exercising software with the intent of finding and ultimately correcting errors. This fundamental philosophy does not change for web applications, because Web-based systems and application reside on a network and interoperate with many different operating system, browsers, hardware platforms, and communication protocols; the search for errors represents a significant challenge for web application.

The distributed nature of client\server environments, the performance issues associated with transaction processing, the potential presence of a number of different hardware platforms, the complexities of network communication, the need to serve multiple clients from a centralized database and the requirements imposed on the server all combine to make testing of client\server architectures.

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. System testing is the state of implementation that is aimed at assuring that the system works accurately and efficiently. Testing is the vital to the success of the system. System testing makes the logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

**The objective of testing is as follows:**

* + Testing is the process of executing a program with the intent of finding an error.
  + A successful test is that one of the cover of undiscovered error.

### TESTING ISSUES

* Client GUI considerations
* Target environment and platform diversity considerations
* Distributed database considerations
* Distributed processing considerations

**TESTING METHODOLOGIES**

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfills this objective a series of test step unit, integration. validation and system tests were planned and executed the test steps are:

**System Testing**

Testing is an important phase in project development. System testing makes a logical assumption that if all parts of the system are correct, and the goal will be achieved successfully. The software must meet the user specification and it must satisfy according to the needs of the users.

Testing is the process of executing a project within the intend of finding errors. A good test case is one that has a high probability of finding an undiscovered error.

**Unit Testing**

Unit testing focuses verification efforts on the smallest unit of software design of the module. This is also known as “module testing”. This testing is carried out during programming stage itself. In this testing step, each module is found to be working satisfactorily as regards to the expected output of the modules.

**In Project**, Each module such customer registration module, request module, employee details module, stock module, vehicle module and area detail modules are tested individually for example, Customer details module can contain the more forms to maintain the information so all forms could be tested like entered information store appropriately in database access page or not. If correctly accessed means the testing of registration module successfully completed. Likewise all modules are tested successfully.

**Integration Testing**

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

**In Project,** Integration of two modules can be tested together such as customer registration details and customer login module for verification purposes providing proper accessibility to users. The communication of Registration and Login module can test and executed successfully.

**Validation Testing**

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists:

**In this project,** Admin login details form Enter without username and password in textbox enter the submit button then Login failed message otherwise checks the both textbox value that is true means valid page displayed. Enter Password Displaying password character \*.if it displays the characters security is not availed so testing of software is failed.

**Output Testing**

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

**In project** All the forms are tested as it gives the necessary output to the user’s search such as view response details.

**4.3 SYSTEM IMPLEMENTATION**

System implementation is the stage of the project that the theoretical design is turned into a working system. If the implementation stage is not properly planned and controlled, it can cause error. Thus it can be considered to be the most crucial stage in achieving a successful new system and in giving the user confidence that the new system will work and be effective.

Normally this stage involves setting up a coordinating committee, which will act as a sounding board for ideas; complaints and problem. The first task is implementation planning; i.e., deciding on the methods and time scale to be adopted. Apart from planning two major task of preparing for implementation are, education takes place much earlier in the project; at the implementation stage the emphasis must be on training in new skills to give staff confidence they can use the system. Once staff has been trained, the system can be tested.

After the implementation phase is completed and the user staff is adjusted to the changes created by the candidate system, evaluation and maintenance is to bring the new system to standards. The activities of the implementation phase can be summarized as,

* + - * Implementation planning
      * Education planning
      * System planning

**IMPLEMENTATION PROCEDURES**

Implementation includes all the activities that take place to convert the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organization requirements. Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

**Implementation Procedures**

**Pilot Running:**

Processing the current data by only one user at a time called the pilot running process. When one user is accessing the data at one system, the system is sets to be engaged and connected in network. This process is useful only in system where more then one user is restricted.

**Parallel Running:**

Processing the current data by more then one user at a time simultaneously is said to be parallel running process. This same system can be viewed and accessed by more then one user at the time. Hence the implementation method used in the system is a pilot type of implementation.

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

The stage consists of,

* Testing the developed program with sample data.
* Detection’s and correction of error.
* Creating whether the system meets user requirements.
* Making necessary changes as desired by the user.
* Training user personnel.

**4.3.2 USER MANUAL**

**User Training**

User Training is designed to prepare the user for testing &consenting the system. .

They are

1) User Manual.

2) Help Screens.

3) Training Demonstration.

**1) User Manual:**

The summary of important functions about the system and software can be provided as a document to the user.

**2) Help Screens:**

This features now available in every software package, especially when it is used with a menu. The user selects the “Help” option from the menu. The system accesses the necessary description or information for user reference.

**3) Training Demonstration:**

Another User Training element is a Training Demonstration. Live demonstrations with personal contact are extremely effective for Training Users.

**SYSTEM MAINTENANCE**

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits***.***

* Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
* Users are not fully cognizant of the maintenance problem or its high cost.
* Few tools and techniques are available for maintenance.
* A good test plan is lacking.
* Standards, procedures, and guidelines are poorly defined and enforced.
* Programs are often maintained without care for structure and documentation.
* There are minimal standards for maintenance.
* Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

**Corrective Maintenance**

It means repairing, processing or performance failure or making changes because of previously uncovered problems or false assumptions. Task performed to identify, isolate, and rectify a fault so that the failed equipment, machine, or system can be restored to an operational condition within the tolerances or limits established for in-service operations.

Corrective maintenance can be subdivided into "immediate corrective maintenance" (in which work starts immediately after a failure) and "deferred corrective maintenance" (in which work is delayed in conformance to a given set of maintenance rules).

**Perfective Maintenance**

It means changes made to a system to add new features or to improve performance. Preventive maintenance is predetermined work performed to a schedule with the aim of preventing the wear and tear or sudden failure of equipment components. process or control equipment failure can have adverse results in both human and economic terms. In addition to down time and the costs involved to repair and/or replace equipment parts or components, there is the risk of injury to operators, and of acute exposures to chemical and/or physical agents.

Time-based or run-based Periodically inspecting, servicing, cleaning, or replacing parts to prevent sudden failure .On-line monitoring of equipment in order to use important/expensive parts to the limit of their serviceable life. Preventive maintenance involves changes made to a system to reduce the chance of future system failure.

An example of preventive maintenance might be to increase the number of records that a system can process far beyond what is currently needed or to generalize how a system sends report information to a printer so that so that the system can adapt to changes in printer technology.

**Preventive Maintenance**

Changes made to a system to avoid possible future problems Perfective maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. The objective of perfective maintenance is to improve response time, system efficiency, reliability, or maintainability.

  During system operation, changes in user activity or data pattern can cause a decline in efficiency, and perfective maintenance might be needed to restore performance. Usually, the perfective maintenance work is initiated by the IT department, while the corrective and adaptive maintenance work is normally requested by users.

**CONCLUSION AND FUTURE ENHANCEMENT**

**CONCLUSION:**

This Project has been designed to the admin login and create and voters the candidates and voters details are already collected by administrator. Then admin will verify and create candidates and voters. Voter can access this portal and voting their favourite candidate. Voter can’t vote more than one time in an same type of voting. Then the voting result will be announced by administrator.

**FUTURE ENHANCEMENT**

This application is developed by using JAVA and MS SQL SERVER as back end. In future this system may be developed by android or any other technology which is peak in that time. According to the user requirement every new technology or any techniques are developed. In future the security will be enhanced with high level. A New technology is implemented with this system to get better performance. The future system will be developed with different features according to the users need. The future enhancement includes a more advance features and secures the data which will be incorporate all the methods, which are convenient for the organization to give the better performance.

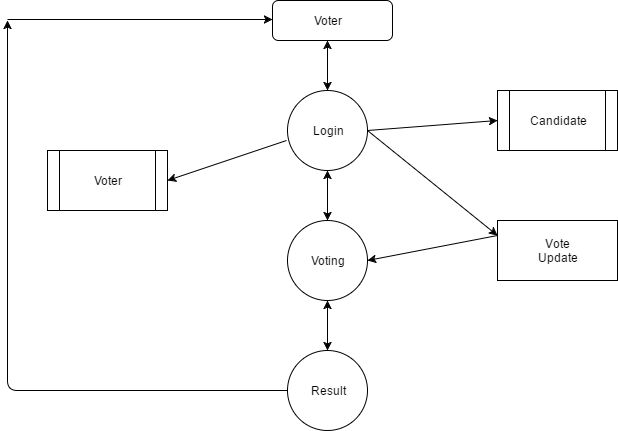
**APPENDIX**

**DATA FLOW DIAGRAM**

LEVEL 0:

****

LEVEL 1:

****

**TABLE DESIGN**

**Table Name : Admin Table**

**Primary key : ID**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| Id | Int | Admin user ID |
| Username | Varchar(15) | Admin Username |
| Password | Varchar(15) | Admin password |

**Table Name : Voter Table**

**Primary key : voter ID**

**Foreign key : created\_by\_id**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| VoterId | Integer | Voter ID |
| Created\_by\_id | Integer | Administrator ID |
| Name | Varchar(15) | Name of the voter |
| Gender | Varchar(6) | Gender of voter |
| Age | Integer | Age of voter |
| City | Varchar(15) | City of voter |
| Security | Text | Security Question |
| Status | Boolean | Status of voter(he/she can vote or not) |

**Table Name : Candidate Table**

**Primary Key : candidate\_id**

**Foreign Key : created\_by\_id**

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
| Canidate\_id | Integer | Candidate ID |
| Created\_by\_id | Integer | Administrator ID |
| Name | Varchar(15) | Name of the voter |
| Sex | Varchar(6) | Gender of voter |
| Age | Integer | Age of voter |
| City | Varchar(15) | City of voter |
| Count | Integer | Count the no of votes |

SAMPLE CODING

<html>

<head>

<link rel="stylesheet" type="text/css" href="/stylesheets/bootstrap.css">

<link rel="stylesheet" type="text/css" href="/stylesheets/login.css">

<link rel="stylesheet" type="text/css" href="/stylesheets/sweetalert.css">

<script src="/javascripts/jquery.js"></script>

<script src="/javascripts/bootstrap.js"></script>

<script src="/javascripts/sweetalert.js"></script>

<script src="/javascripts/src/login.js"></script>

</head>

<body>

<div class="container">

<div class="row">

<div class="col-sm-6 col-md-4 col-md-offset-4">

<h1 class="text-center login-title">Sign in to continue Pooling Vote</h1>

<div class="account-wall">

<img class="profile-img" src="https://lh5.googleusercontent.com/-b0-k99FZlyE/AAAAAAAAAAI/AAAAAAAAAAA/eu7opA4byxI/photo.jpg?sz=120"

alt="">

<input type="text" name="username" class="form-control" placeholder="Username" required autofocus>

<input type="password" name="password" class="form-control" placeholder="Password" required>

<button class="btn btn-lg btn-primary btn-block" id="login()" onclick="login()">

Sign in</button>

</div>

</div>

</div>

</div>

</body>

</html>

<html>

<head>

<link rel="stylesheet" type="text/css" href="/stylesheets/bootstrap.css">

<link rel="stylesheet" type="text/css" href="/stylesheets/admin.css">

<link rel="stylesheet" type="text/css" href="/stylesheets/sweetalert.css">

<script src="/javascripts/jquery.js"></script>

<script src="/javascripts/bootstrap.js"></script>

<script src="/javascripts/sweetalert.js"></script>

<script src="/javascripts/chart.js"></script>

<!--<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>-->

<script src="/javascripts/src/admin.js"></script>

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-12">

<!-- Nav tabs -->

<div class="card">

<ul class="nav nav-tabs" role="tablist">

<li role="presentation" class="active"><a href="#profile" aria-controls="profile" role="tab" data-toggle="tab">Profile</a></li>

<li role="presentation"><a href="#candidate" aria-controls="candidate" role="tab" data-toggle="tab">Candidate</a></li>

<li role="presentation"><a href="#voter" aria-controls="voter" role="tab" data-toggle="tab">Voter</a></li>

<li role="presentation"><a href="#report" aria-controls="report" role="tab" data-toggle="tab">Report</a></li>

<li role="presentation"><a onclick="window.location.href='/'" style="cursor:pointer" aria-controls="report" role="tab" data-toggle="tab">Logout</a></li>

</ul>

<!-- Tab panes -->

<div class="tab-content">

<div role="tabpanel" class="tab-pane active" id="profile">

<div class="row">

<div class="col-xs-12 col-sm-6 col-md-6">

<div class="well well-sm">

<div class="row">

<div class="col-sm-6 col-md-4">

<img id="profileimg" src="http://placehold.it/380x500" alt="" class="img-rounded img-responsive" />

</div>

<div class="col-sm-6 col-md-8">

<h4 id="proname">Gokul</h4>

<small><cite id="promobile" title="San Francisco, USA">7418227833<i class="glyphicon glyphicon-map-marker">

</i></cite></small>

<p id="proemail">

<i class="glyphicon glyphicon-envelope"></i>Female

</p>

<p">

<i class="glyphicon glyphicon-globe"></i>Admin</a>

</p>

<p id="proaddress">

<i class="glyphicon glyphicon-gift"></i>18-08-2017

</p>

<!-- Split button -->

</div>

</div>

</div>

</div>

</div>

</div>

<div role="tabpanel" class="tab-pane" id="candidate">

<div class="col-sm-6">

<div class="form-group col-sm-8">

<label for="inputsm" >Firstname</label>

<input name="canfirstname" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm" >Lastname</label>

<input name="canlastname" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Stand On</label>

<select class="form-control" id="canoption">

<option value="1">Chairman</option>

<option value="2">Vice Chairman</option>

<option value="3">College Secretary</option>

<option value="4">Finance Secretary</option>

<option value="5">Treasurer</option>

<option value="6">Sports Secretary</option>

</select>

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Mobile</label>

<input name="canmobile" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Email</label>

<input name="canemail" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Address</label>

<input name="canaddress" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<center><button onclick="addCandidate()" type="button" class="btn btn-warning">Submit</button></center>

</div>

</div>

<div class="col-sm-6">

<input class="glyphicon glyphicon-search" type="text" id="myInput" onkeyup="myFunction()" placeholder="Search for names.." title="Type in a name">

<ul id="myUL">

</ul>

</div>

</div>

<div role="tabpanel" class="tab-pane" id="voter">

<div class="col-sm-6">

<div class="form-group col-sm-8">

<label for="inputsm" >Firstname</label>

<input name="votfirstname" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm" >Lastname</label>

<input name="votlastname" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Mobile</label>

<input name="votmobile" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Email</label>

<input name="votemail" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<label for="inputsm">Address</label>

<input name="votaddress" class="form-control input-sm" id="inputsm" type="text">

</div>

<div class="form-group col-sm-8">

<center><button onclick="addVoter()" type="button" class="btn btn-warning">Submit</button></center>

</div>

</div>

<div class="col-sm-6">

<input class="glyphicon glyphicon-search" type="text" id="myInput" onkeyup="filterVoter()" placeholder="Search for names.." title="Type in a name">

<ul id="filtervote">

</ul>

</div>

</div>

<div role="tabpanel" class="tab-pane" id="report">

<div class ="col-sm-6">

<h1 style="color:BLUE">Chairman</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>Count</th>

</tr>

</thead>

<tbody id="chairman">

</tbody>

</table>

</div>

<div class ="col-sm-6">

<h1 style="color:BLUE">Vice Chairman</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>Count</th>

</tr>

</thead>

<tbody id="vicechairman">

</tbody>

</table>

</div>

<div class ="col-sm-6">

<h1 style="color:BLUE">College Secretary</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>Count</th>

</tr>

</thead>

<tbody id="collegesec">

</tbody>

</table></div>

<div class ="col-sm-6">

<h1 style="color:BLUE">Finance Secretary</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>Count</th>

</tr>

</thead>

<tbody id="financesec">

</tbody>

</table></div>

<div class ="col-sm-6">

<h1 style="color:BLUE">Treasurer</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>Count</th>

</tr>

</thead>

<tbody id="treasurer">

</tbody>

</table></div>

<div class ="col-sm-6">

<h1 style="color:BLUE">Sports Secretary</h1>

<table class="table table-bordered">

<thead>

<tr>

<th>Rank</th>

<th>Name</th>

<th>Count</th>

</tr>

</thead>

<tbody id="sportssec">

</tbody>

</table></div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Modal -->

<div id="myModal" class="modal fade" role="dialog">

<div class="modal-dialog">

<!-- Modal content-->

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal">&times;</button>

<h4 class="modal-title"> Deatail</h4>

</div>

<div class="modal-body">

<div class="row">

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

<div class="well well-sm">

<div class="row">

<div class="col-sm-6 col-md-4">

<img id="modalimg" src="http://placehold.it/380x500" alt="" class="img-rounded img-responsive" />

</div>

<div class="col-sm-6 col-md-8">

<h4 id="modname"></h4>

<small><cite id="modmobile" title="San Francisco, USA"></cite></small>

<p>

<i class="glyphicon glyphicon-globe"></i><a id="modpost"></a>

<br />

<i class="glyphicon glyphicon-envelope"></i><a id="modemail"></a>

<br />

<i class="glyphicon glyphicon-gift"></i><a id="modaddress"></a></p>

<!-- Split button -->

</div>

</div>

</div>

</div>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

</body>

</html>

var candidateList;

var voterList;

var final\_id;

$(document).ready(function(){

var url\_string = window.location.href;

var url = new URL(url\_string);

final\_id = url.searchParams.get("id");

$("#profileimg").attr('src',"/images/user/"+final\_id+".jpg");

ViewCandidateList();

setProfile(final\_id);

ViewVoterList();

setReport();

});

function setReport(){

var chair='';

var vicechair='';

var collegesec='';

var financesec='';

var treasurer='';

var sportssec='';

$.ajax({

type:"GET",

url:"/api/votreport",

success: function(datas) {

var ch=0,vc=0,cs=0,fs=0,ts=0,ss=0;

for(var i=0;i<datas.length;i++){

var data = datas[i];

if(data.postid==1){

ch++;

chair=chair+`<tr><td>`+ch+`</td>

<td>`+data.firstname+`</td>

<td>`+data.count+`</td></tr>`;

}

if(data.postid==2){

vc++;

vicechair=vicechair+`<tr><td>`+vc+`</td>

<td>`+data.firstname+`</td>

<td>`+data.count+`</td></tr>`;

}

if(data.postid==3){

cs++;

collegesec=collegesec+`<tr><td>`+cs+`</td>

<td>`+data.firstname+`</td>

<td>`+data.count+`</td></tr>`;

}

if(data.postid==4){

fs++;

financesec=financesec+`<tr><td>`+fs+`</td>

<td>`+data.firstname+`</td>

<td>`+data.count+`</td></tr>`;

}

if(data.postid==5){

ts++;

treasurer=treasurer+`<tr><td>`+ts+`</td>

<td>`+data.firstname+`</td>

<td>`+data.count+`</td></tr>`;

}

if(data.postid==6){

ss++;

sportssec=sportssec+`<tr><td>`+ss+`</td>

<td>`+data.firstname+`</td>

<td>`+data.count+`</td></tr>`;

}

}

$("#chairman").html(chair);

$("#vicechairman").html(vicechair);

$("#collegesec").html(collegesec);

$("#financesec").html(financesec);

$("#treasurer").html(treasurer);

$("#sportssec").html(sportssec);

},

});

}

function setProfile(id){

$.ajax({

type:"POST",

url:"/api/getprofile",

data:{id:id},

success: function(datas) {

var data = datas[0];

$("#proname").text(data['firstname']+" "+data['lastname']);

$("#promobile").text(data['mobile']);

$("#proemail").text(data['email']);

$("#proaddress").text(data['address']);

},

});

}

function addCandidate(){

swal({

title: "Are you sure?",

text: "You want to add Candidate!",

type: "warning",

showCancelButton: true,

confirmButtonClass: "btn-danger",

confirmButtonText: "Yes, Sumbit it!",

cancelButtonText: "No, cancel!",

closeOnConfirm: false,

closeOnCancel: false

},

function(isConfirm) {

if (isConfirm) {

var firstname=$("input[name=canfirstname]").val().trim();

var lastname=$("input[name=canlastname]").val().trim();

var mobile=$("input[name=canmobile]").val().trim();

var email=$("input[name=canemail]").val().trim();

var address=$("input[name=canaddress]").val().trim();

var post=$('#canoption').val().trim();

if(firstname.length==0 || lastname.length==0 || mobile.length==0 || email.length==0 || address.length==0 || post.length==0){

swal( "Required Fields", 'Should Enter all fields!', 'warning');

}else{

$.ajax({

type:"POST",

url:"/api/addCandidate",

data:{firstname:firstname,lastname:lastname,mobile:mobile,email:email,address:address,post:post},

success: function(datas) {

swal("Success!", "Your Record has been deleted.", "success");

$("input[name=canfirstname]").val("");

$("input[name=canlastname]").val("");

$("input[name=canmobile]").val("");

$("input[name=canemail]").val("");

$("input[name=canaddress]").val("");

ViewCandidateList();

},

});

}

} else {

swal("Cancelled", "Your record is note Added :)", "error");

}

});

}

function ViewCandidateList(){

//

$.ajax({

type:"POST",

url:"/api/viewCandidates",

success: function(datas) {

var html='';

candidateList=datas;

for(var i=0;i<datas.length;i++){

var data = datas[i];

html =html+ `<li><img onclick="deletecandidate('`+i+`')" style="float:right" src="/images/delete.png"></img><a onclick="modal('`+i+`','candidate')" style="cursor:pointer" data-toggle="modal">`+data['firstname']+` `+data['lastname']+`</a></li>`;

}

$("#myUL").html(html);

},

});

}

function modal(pos,path){

var postname='';

if(candidateList[pos]['post']==1){

postname="Chairman";

}else if(candidateList[pos]['post']==2){

postname="Vice Chairman";

}

else if(candidateList[pos]['post']==3){

postname="College Secretary";

}

else if(candidateList[pos]['post']==4){

postname="Finance Secretary";

}

else if(candidateList[pos]['post']==5){

postname="Treasurer";

}

else if(candidateList[pos]['post']==6){

postname="Sports Secretary";

}

$("#modalimg").attr('src',"/images/"+path+"/"+candidateList[pos]['userid']+".jpg");

$("#modname").text(candidateList[pos]['firstname']+' '+ candidateList[pos]['lastname']);

$("#modmobile").text(candidateList[pos]['mobile']);

$("#modpost").text(postname);

$("#modemail").text(candidateList[pos]['email']);

$("#modaddress").text(candidateList[pos]['address']);

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

}

function deletecandidate(pos){

swal({

title: "Are you sure?",

text: "You will not be able to recover Candidate detail!",

type: "warning",

showCancelButton: true,

confirmButtonClass: "btn-danger",

confirmButtonText: "Yes, delete it!",

cancelButtonText: "No, cancel plx!",

closeOnConfirm: false,

closeOnCancel: false

},

function(isConfirm) {

if (isConfirm) {

$.ajax({

type:"POST",

data:{id:candidateList[pos]['userid']},

url:"/api/deleteCandidates",

success: function(datas) {

swal("Deleted!", "Your imaginary file has been deleted.", "success");

ViewCandidateList();

},

});

} else {

swal("Cancelled", "Your record is safe :)", "error");

}

});

}

function addVoter(){

swal({

title: "Are you sure?",

text: "You want to add Voter!",

type: "warning",

showCancelButton: true,

confirmButtonClass: "btn-danger",

confirmButtonText: "Yes, Sumbit it!",

cancelButtonText: "No, cancel!",

closeOnConfirm: false,

closeOnCancel: false

},

function(isConfirm) {

if (isConfirm) {

var firstname=$("input[name=votfirstname]").val().trim();

var lastname=$("input[name=votlastname]").val().trim();

var mobile=$("input[name=votmobile]").val().trim();

var email=$("input[name=votemail]").val().trim();

var address=$("input[name=votaddress]").val().trim();

if(firstname.length==0 || lastname.length==0 || mobile.length==0 || email.length==0 || address.length==0 ){

swal( "Required Fields", 'Should Enter all fields!', 'warning');

}else{

$.ajax({

type:"POST",

url:"/api/addVoter",

data:{firstname:firstname,lastname:lastname,mobile:mobile,email:email,address:address},

success: function(datas) {

swal("Success!", "Your Record has been Register.", "success");

$("input[name=votfirstname]").val("");

$("input[name=votlastname]").val("");

$("input[name=votmobile]").val("");

$("input[name=votemail]").val("");

$("input[name=votaddress]").val("");

ViewVoterList();

},

});

}

} else {

swal("Cancelled", "Your record is note Added :)", "error");

}

});

}

function ViewVoterList(){

//

$.ajax({

type:"POST",

url:"/api/viewVoter",

success: function(datas) {

var html='';

voterList=datas;

for(var i=0;i<datas.length;i++){

var data = datas[i];

html =html+ `<li><img onclick="deletevoter('`+i+`')" style="float:right" src="/images/delete.png"></img><a onclick="votmodal('`+i+`','user')" style="cursor:pointer" data-toggle="modal">`+data['firstname']+` `+data['lastname']+`</a></li>`;

}

$("#filtervote").html(html);

},

});

}

function deletevoter(pos){

swal({

title: "Are you sure?",

text: "You will not be able to recover Voter detail!",

type: "warning",

showCancelButton: true,

confirmButtonClass: "btn-danger",

confirmButtonText: "Yes, delete it!",

cancelButtonText: "No, cancel plx!",

closeOnConfirm: false,

closeOnCancel: false

},

function(isConfirm) {

if (isConfirm) {

$.ajax({

type:"POST",

data:{id:voterList[pos]['userid']},

url:"/api/deleteVoter",

success: function(datas) {

swal("Deleted!", "Your imaginary file has been deleted.", "success");

ViewVoterList();

},

});

} else {

swal("Cancelled", "Your record is safe :)", "error");

}

});

}

function votmodal(pos,path){

var postname='';

$("#modalimg").attr('src',"/images/"+path+"/"+candidateList[pos]['userid']+".jpg");

$("#modname").text(voterList[pos]['firstname']+' '+ voterList[pos]['lastname']);

$("#modmobile").text(voterList[pos]['mobile']);

$("#modpost").text("Voter");

$("#modemail").text(voterList[pos]['email']);

$("#modaddress").text(voterList[pos]['address']);

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

}

function filterVoter() {

var input, filter, ul, li, a, i;

input = document.getElementById("myInput");

filter = input.value.toUpperCase();

ul = document.getElementById("filtervote");

li = ul.getElementsByTagName("li");

for (i = 0; i < li.length; i++) {

a = li[i].getElementsByTagName("a")[0];

if (a.innerHTML.toUpperCase().indexOf(filter) > -1) {

li[i].style.display = "";

} else {

li[i].style.display = "none";

}

}

}

function myFunction() {

var input, filter, ul, li, a, i;

input = document.getElementById("myInput");

filter = input.value.toUpperCase();

ul = document.getElementById("myUL");

li = ul.getElementsByTagName("li");

for (i = 0; i < li.length; i++) {

a = li[i].getElementsByTagName("a")[0];

if (a.innerHTML.toUpperCase().indexOf(filter) > -1) {

li[i].style.display = "";

} else {

li[i].style.display = "none";

}

}

}