3-2018

Virtual Election Booth

Manohar Adepu
St. Cloud State University, madepu@stcloudstate.edu

Follow this and additional works at: http://repository.stcloudstate.edu/msia_etds

Virtual Election Booth

by

Manohar Adepu

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of Requirements

for the Degree of

Master of Science

in Information Assurance

March 2018

Starred Paper Committee:
Susantha Herath, Chairperson
Dien D. Phan
Balasubramanian Kasi

Abstract

The electronic voting system provides various advantages for voting and saves significant resources in the process of voting, collecting and counting of ballots. This implementation will provide a secure way for people to vote online, which also eliminates the hassle of physically being present at designated election locations. However, for all the above savings electronic voting systems must provide security in each stage of the process to avoid any compromise of the authenticity of the results. This project aims to provide one such secure framework by implementing a secure online election voting protocol which has the ability to maintain privacy. Also a goal was to use this platform for finding the statistics of who voted and who did not vote. In Short, this is secure, cost saving and easy to use one step online voting system. The project aims to improve features of the current voting system which are currently lacking. The phone or a personal workstation is a better and more convenient option for a user to vote with. Ways to improve the voting system interface should, therefore, be designed and developed in order to make voting more convenient in a secured way.

Table of Contents

List of Figures

6

Chapter I: Introduction

With a huge growth in the Internet and related technology, it would be convenient and significantly more secure to have web-based voting rather than the customary in-person voting process. Nonetheless, this web voting plan cannot be used if it does not maintain the privacy and individual protections of the voters. This can be accomplished with the assistance of today's outstanding technology around computer networks and cryptographic techniques. Nowadays a considerable number of issues and tricks and security violations have been seen around financial transactions. Accordingly, we need to consider a lot of security issues while actualizing the voting system on the web. A protected and reliable decision is the most critical component to a popularity-vote based nation to effectively thrive. Subsequently, a considerable measure of care ought to be practised while actualizing the web voting system. This suggests that the voting system requires a greater amount of security in contrast with the other financial exchanges transacted over the web. There exists a considerable amount challenge innate to actualize the web voting. There are many complicated issues to consider, and there are several voting protocols trying to address these inherent problems in the structure and requirements of the online voting system.

Problem Statement

Many countries have a long history of elections, but are still far from finding a perfect voting process. For 2014 midterm election in United States, just 36 percent (Bult, Durkin, & McShane, 2016) of eligible voters showed up at the polls, which was the worst rate in 72 years. US voters encounter problems like long lines (FESSLER, 2016), and were turned away for showing the wrong ID (The Opinion Pages, 2014) or their registration information might have been erroneously scrubbed from voter rolls (Bult, Durkin, & McShane, 2016).

In the present voting system, there are high chances of fraud, duplication of votes and forgery. The technological advancements can be used to build a better, fast and more secured virtual election booth which eliminates the tedious task of conducting elections physically and can also save resources.

Both the present and past voting systems do not implement secured voting mechanisms like security protocols and use of the world-wide web. Instead they use a lot of natural and human resources to conduct elections and also come with a high risk of fraud, many different voting systems have been introduced but were deemed not secure enough as each (in-person voting systems) had its own problems.

Therefore, the concept of CTF – central tabulation functionality and CLA – central legislative agency implementing with secured cryptographic mechanisms and protocols will help in developing a better virtual election system.

Nature and Significance of the Problem

It is the right of every individual to vote in a democratic country like the United States but the statistics prove that more than 50% of the population does not vote. Therefore, unless people choose the right candidate policies cannot be turned into reality. By taking advantage of the latest advancements in information technology, cryptography and the world wide web voting can be made a virtual process which plays a major role in eliminating the problems in the current voting system and will also allow people to vote at their fingertips.

Objective of the Study

The objective of the study is to develop an effective and secure virtual voting system. Electronic voting (e-voting) is used to solve the problems of traditional paper-based voting and can lower costs. Several security requirements should be met, such as verifiability,

unforgeability, identifiability, vote uniqueness, tally correctness, individual verifiability – unique identification for every voter.

Study Questions/Hypotheses

1)   What are the entities in the virtual election system?

2)   How to provide a secured voting system?

3)   Which technologies should be used to develop the secured virtual election booth?

Definition of Terms

The Virtual Election booth provides security by implementing secured authorization and authentication mechanisms to vote. There are three levels of authorized users that can access virtual election booth, they are CLA, CTF, and Voter. Below are the tasks that each entity is capable of or authorized to perform.

CLA (Central Legitimization Agency). The task of the CLA is to certify the voters. The CLA assigns validation numbers to the voters and maintains a list of validation numbers and voters to avoid any voter from voting twice.

CTF (Central Tabulating Agency). The task of the CTF is to count the votes. CTF receives the votes and the validation number from voters and records the vote for a particular election candidate and updates the validation number as "voted" to prevent the same voter from voting twice.

Voter. The person who will be using the electronic voting system to cast his vote. (List them in the order you presented them above)

Summary

The virtual voting system provides various advantages for voting and saves significant resources in the process of voting, collecting and counting of ballots. The virtual voting system

will provide a secure system for people to vote online which is convenient to the voters by eliminating the hassle of physically being present to vote. This project aims at providing one such secure framework by implementing a secure online election voting protocol which has the ability to maintain privacy. Also a goal is to use this Platform for finding the statistics of who voted and who did not vote. In short, this is a secure, cost saving and one step virtual voting system. The project aims to improve features of the current voting system which are lacking. The smart phone or a personal computer is a better and more convenient option for a user to vote. In the next chapters, we will look at ways to improve the voting system interface and therefore design and develop to achieve more efficient and secure voting system.

Chapter II: Literature Review

In this chapter, the background related to the problem of voting systems is described in depth and the literature that supports and encourages the building of a virtual election booth is discussed.

Background Related to the Problem

Voting systems have been evolved from counting hands in the initial days, to different systems that included punch cards, papers, mechanical lever, and optical-scan machines. From the recent examples of democratic elections around then globe especially India and United States, it has been shown that the winning margins could be less than the margin of error thus making elections a potential error-prone task. Therefore the use of e-voting systems has the capability to reduce or even remove unwanted errors that might be caused by humans, and also e-voting is more reliable and can also handle multiple modalities which help physically handicapped(such as voice assistance) to vote and increase scalability for huge elections. With physical voting machines, it is not always possible for people like soldiers or immigrants to vote. If a potential voter does not live in the respective geographic proximity to the physical voting machines, the use of e-voting is also an excellent mechanism that doesn't require geographical proximity of the voters.

Ensuring that voting system are secured and avoiding potential rigging is a complex task for the respective election departments or government agencies if the process is not e-voting. The virtual election system can provide assurance of securing sensitive information of the voters, make every voter sure that their vote has been taken into consideration, that no duplicate votes are recorded to ensure no rigging of votes has occurred, and we can also implement optional

modules which help in various ways like generating data to determine who voted and who did not.

Also, it is possible to take advantage of improvements in data analysis by retrieving the data from the virtual election booth and analyze the reasons and develop strategies to increase the voter count in the future.

Literature Related to the Problem

In 1983[2], Chaum proposed Blind Signature and several researchers developed and proposed their secure e-voting schemes by considering blind signature as the basis. Blind Signature implements secure e-voting schemes to assure security or protect sensitive information of the voters (Iuon-Chang Lina, 2003)

Janga and Chakchai have introduced a virtual election system where CLA and CTF are the two central facilities which help in accomplishing the objectives of a secure election. This scheme helps in overcoming the drawbacks of voting with blind signatures or with a single central facility in both cheating and privacy.

By considering the idea of implementing two central facilities and bringing my expertise to J2EE and the skills learned from Information Assurance program, the project has been designed which is described in the next chapters.

In 2017, Arthanayaka proposed an XML based security model for enhancing the integrity and privacy of e-voting systems, according to Arthanayaka the existing electronic voting systems are extremely complex and expensive to develop maintain and also to provide security attributes such as integrity and privacy and concludes that these systems cannot be implemented in specific platforms. He explains how to secure the integrity and privacy of the electronic voting system using xml.

The virtual election system is designed by considering Arthinayaka's idea of achieving integrity and privacy, this is achieved not by xml but designing the application in a more secured way and using an MD5 hash to generate a hash value of the sensitive information.

The secured online voting protocol should be capable of providing a secured voting process and also prevent cheating. Therefore, the requirements to achieve the both aforementioned requirements are as follows:

- Only authorized voters should be allowed to vote.

- Each voter should be able to vote only once.

- A vote should not be allowed to be duplicated.

- A vote cannot be changed after it has been counted.

- Every voter should be able to make sure that their vote has been taken into consideration.

- Everyone knows who voted and who did not.

Literature Related to the Methodology

The model view controller pattern is implemented in this project for better efficiency in developing the application and segregating the business code from the view code.

Three Layers of MVC Pattern.

Model. The model layer represents the data to the user, it defines the place where the application's data objects are stored. The model does not know anything about the view and the controllers. In this project, models are simple Java POJO – plain old java objects classes with setters and getters which allow setting the data entered by the users in forms which can be retrieved by using getters.

View. The views are generally Java servlet pages which contain the UI code. The view has the code that is displayed in the browser. The view is wired to the model and retrieves the data necessary to display it to the end user. Views can be used to develop dynamic pages. Front-end developers can code in the views while the backend developers code in the controllers which have the business logic.

Controller. The controller has the business logic code. In this project, we have controller classes which pass the required data to DAO – data access layer classes which do the work of verifying or saving the data to and from the database.

Spring Framework

The Spring Framework is one of the most powerful modular lightweight application development frameworks used for developing Enterprise Java application.

Benefits of the Spring Framework:

1. *Lightweight,* less size and more transparent.

2. *Inversion of Control,* loose coupling between classes can be achieved by utilizing the various features provided by IOC – inversion of control module.

3. *Spring Container* creates and manages the lifecycle of the objects.

4. *Aspect Oriented Programming* enables cohesive development

5. *MVC Framework* provides various annotations and configurations to ease development.

6. *Spring Security* helps in providing comprehensive security for an enterprise J2EE application.

7. *Transaction Management* provides generic abstraction layer.

   Spring Modules:
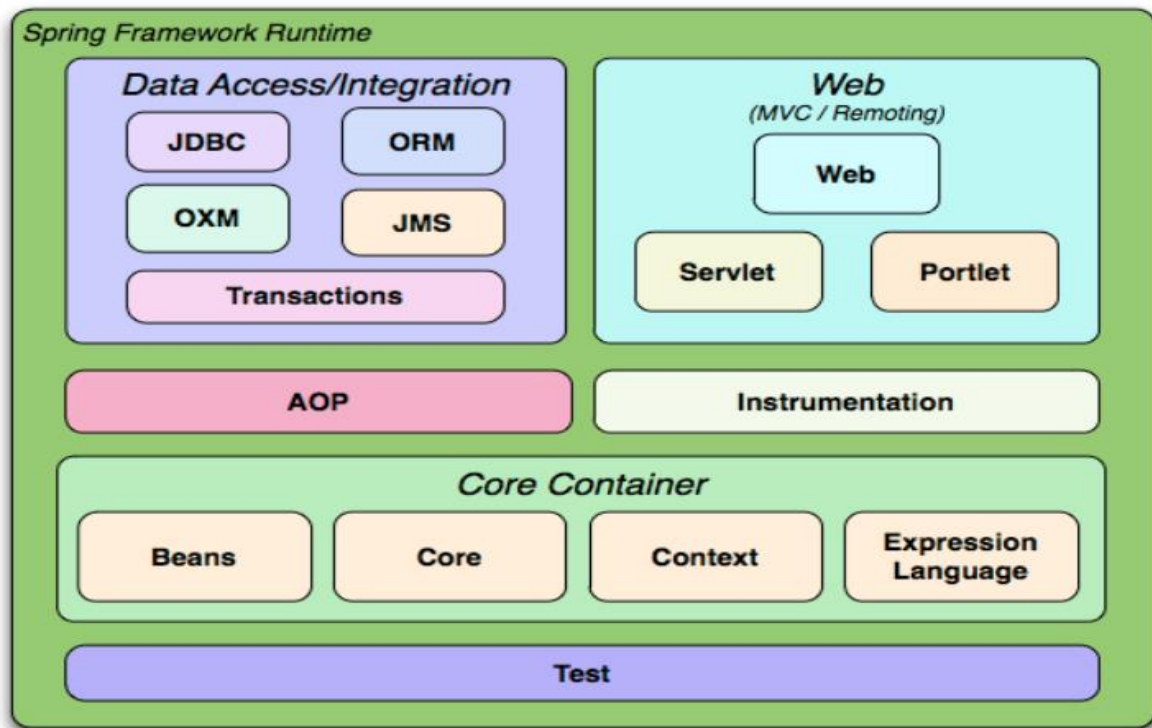
Figure 1. Spring Modules

Summary

The drawbacks of the existing voting systems are discussed and how the problems can be solved by implementing a virtual election booth, the researchers conducted by Janga and Chakchai are taken into consideration along with further enhancements by implementing MVC model with J2EE technologies, cryptography techniques, as well as authorization and authentication techniques to overcome the drawbacks of the existing systems will then be implemented.

Chapter III: Methodology

Because quick secured responses are critical, the J2EE web application strategy facilitates the availability of the voter's information and allows the voter to vote for a candidate by hashing the sensitive details like social security number using Secure Hash Algorithm-1 (MD5). Transparency in the voting process is ensured by making the Central Legitimizing Agency (CLA) unaware of which voter had exercised their vote until the Central Tabulating Facility(CTF) sends the voting status of the voter. Then, which candidate the vote has been exercised for, is only with the CTF until updated with the CLA. This is implemented by creating two different tables for CLA and CTF while designing the database.

Design of the Study

In the development process of the virtual election system prototype, principles from both waterfall and agile development are taken into consideration.

The starred paper is more of a prototype than a research model, when seen in the context of research this is more aligned to qualitative research as this is the implementation of proven security models and approaches.

The requirements of the project are unknown and may change likely, therefore principles from agile are used to handle the sudden change in requirements. In the waterfall model, requirements gathering and analysis, design implementation or coding, testing, deployment and maintenance are done after completion of each phase, therefore it is difficult to implement new ideas or techniques after a few phases have been completed. It also requires testing of the application because of the new changes which affect the timeline and thus there is a high chance of failure.

Although all the phases of the waterfall model were not implemented, the first phase of developing the project and documenting all the requirements was, however, there were changes in later stages. One of the drastic changes made to develop the application using spring framework instead of Struts2.0 for better security as spring provides a separate module named "Spring Security".

The concept of the sprint is taken from the agile methodology and each sprint is divided into 2 weeks and decisions are made around the tasks to be accomplished in the next sprint. The last thing that was done at the end of each spring was sprint retrospective, using this approach it is easy to identify things that the development process should accomplish:

- Start doing

- Stop doing

- Continue doing

Also, before each sprint, the features were divided and analyzed including the complexity and priority of each feature. And before starting the sprint which helps to finish the sprint and accomplish all the features within the timeline or to come to an understanding if the 2-week sprint was not sufficient enough to accomplish it.

Therefore, considering the techniques from both the waterfall and agile methods to develop the virtual election booth template helped in developing, testing and accomplishing it with all of the features embedded within the timeframe.

Phases

Phase 1: Users will register and the CLA will generate validation numbers.

Phase 2: CLA sends the list of validation numbers to CTF.

Phase 3: CTF counts the number of votes.

Phase 4: After all the votes are received the CTF publishes the outcome.

Security Support

The security in virtual election system is implemented by hashing the sensitive data like social security number using secure hash algorithm. By hashing the SSN at the stage of registration no authority will be able to know the private details of voters which are used for the authentication process such as SSN. This message digest class provides applications with the functionality of a message digest algorithm, message digest object then starts out initialized. Message digests are secure one-way hash functions that take an arbitrary-sized amount of data and provide an output of a fixed-length hash value. The hash is calculated using the message digest algorithm i.e., SHA-1 in this case.

The SHA-1 algorithm used in the virtual election system and was developed by the National Institute of Standards and Technology (NIST) and National Security Agency (NSA), implemented by creating message digest object. SHA-1 produces a 160-bit (20 bytes) message digest used for creating unforgeable digital signatures. This is slower than Message Digest 5(MD5) algorithm but is more resistant to brute force than the MD5 hashing algorithm and thus is more secure.SHA-1 is a one-way hashing function. It is easy to compute the SHA-1 message digest of a document, however it cannot go backwards and compose a document based off of a given message digest and it is not an encryption method. To ensure the security, privacy and the integrity of the voting system a hashed validation number is generated during the time of registration and save the hash of the validation number instead of the validation number itself. Therefore, in case of an attack on the database, the sensitive details will not get compromised and the data won't be misused by hackers. If any unauthorized users or hackers get a hold of the sensitive data like SSN or validation numbers, they could spoof or fabricate it by appearing to be

authorized users by knowing details such as SSN and validation number which are enough to vote which appears to be a complete failure of the Virtual voting system. To avoid this, the hash value of the sensitive information is generated and saved in the database rather than storing the plaintext equivalent of it.

Transparency in the voting process is ensured by making the Central Legitimizing Agency (CLA) unaware of which voter had exercised their vote until the Central Tabulating Facility(CTF) sends the voting status of the voter, and towards which candidate the vote had been exercised, is only with the CTF to CLA. This technique is implemented in the virtual election system by creating two different tables for CLA and CTF while designing the database. So that that the above-explained criteria can be successfully implemented. The results will be only released by the CTF. To take care that a voter has voted only once there must be proper communication between CLA and CTF. This process is implemented in a proper flow i.e. at the time of registration the CLA sends the hash of validation to CTF and the CTF gets a record of it. When the voter tries to exercise their vote by entering his validation number and SSN it gets authenticated and is redirected to the CTF. Where the voter votes for a candidate, this information is stored at the CTF and the voting status of the voter is changed at the CTF side and this status is communicated even to the CLA. CLA updates the voting status against that particular validation number and voter status is changed to "VOTED". If the same user tries to vote again, the CLA will not oblige the request and displays a warning message. The database is designed in a way that there are no redundancies in the data which could lead to issues.

The database design properties for an efficient database are commonly described as atomicity, consistency, isolation and durability (ACID) and have been implemented which results in a safer and more efficient system. Session timeouts have also been implemented. It

represents the event occurring when a voter does not perform any action on a website during an interval of some specified time period in the virtual election system, in this case a session time-out duration of 60seconds. The event, on the server side, changes the status of the user session to "invalid" (i.e. "not used anymore") and instruct the web server to delete all data contained in that session. This helps in maintaining the authenticity of the session and unauthorized use is prevented.

Chapter IV: The Project Design

Project Design
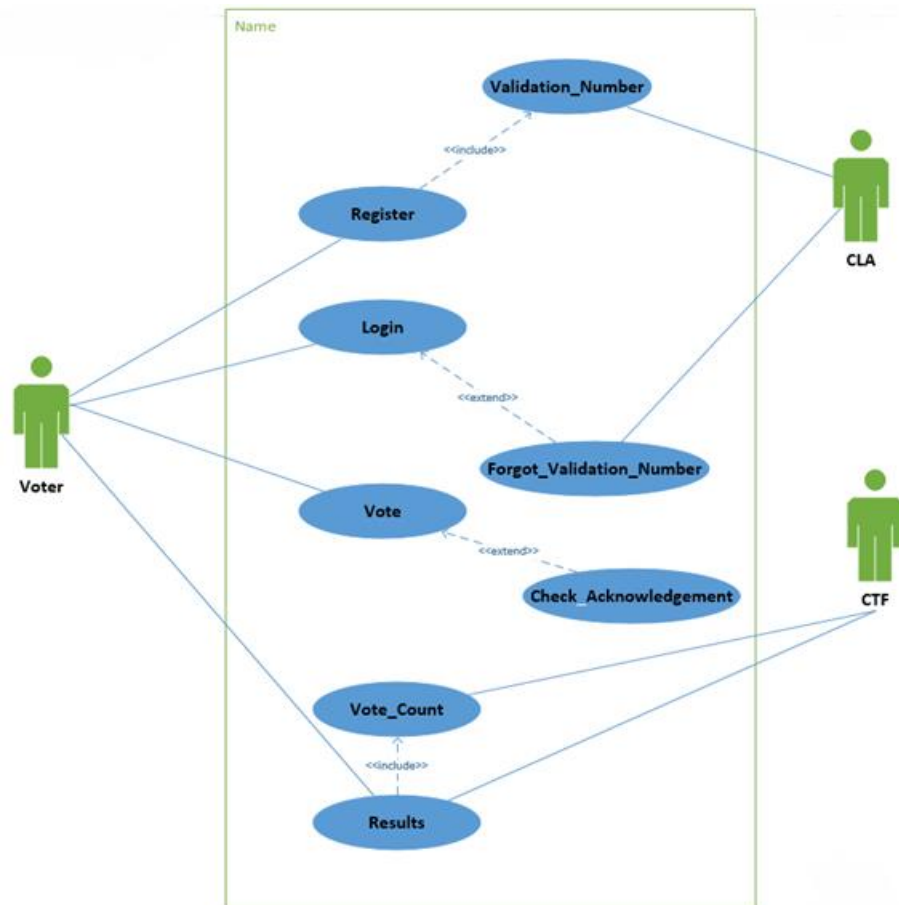
Figure 2. Project Design and Use Case Diagram

The virtual election booth has three entities: Voter, CLA and CTF.

Voter. The actions that can be performed by the voter are register, login, vote and view

results. The first step in this process is, the voter has to register in the registration page by

entering all the required fields. The entered data is then validated by the virtual election booth

and if the validation is successful they receive a unique validation number which is displayed in

a jQuery modal which should be saved and used to login into the virtual election booth.

After receiving the validation number, the voter can login into the virtual election booth and either cast their, vote, view results or view details about each party. Once the voter has performed the vote, the virtual election system performs some predefined validations which again verifies the user details, checks if the user has already cast their vote and if the admin has not yet stopped the election process.

If the user has already cast their vote and tries to vote again before the election process has finished, their vote will be overridden with the latest vote. The result every voter can cast their vote for only a single candidate and can vote only once. Also, there is a chance that the voter might lose or forget the validation number, in this situation the virtual election booth allows the voter to change the password which will we be replaced with the previous one in the database. Before the voter can change the password they will be asked to enter details in the "Forget Password" form which has a few security questions which they will be asked to answer at the first time of the registration in the virtual election system. After entering all the required fields in the "Forget Password" form all the details are validated and verified if the entered information is accurate and they decide to create a new password, and the validation is successful the CLA creates a new password and this password will override the old one and displayed in a jQuery modal to the voter which they can use to login to the virtual election system.

CLA. A Validation number is generated by the CLA at the time of registration of the voter, if the voter forgets the validation number the CLA generates a new validation number which overrides the previous validation number, therefore ensuring that the voter is assigned to only one validation number at any point in time.

The CLA does not know the validation number of any of the voter, the virtual election system is intelligently designed, it just shows the voter id which is a unique id different from the

validation number which is generated to keep the voter's information secret and therefore there is

no way to view the validation number. Each column in the CLA table has a CLA_ID, Voter_ID

and Vote_Status by which the CLA will be able to count how many votes have been performed

without knowing the details of the voter and also which party has received the vote.

   CTF. The CTF is the only authority which can see how many votes each party has

received and it is responsible to provide the results at the end of the election process.

Database Design
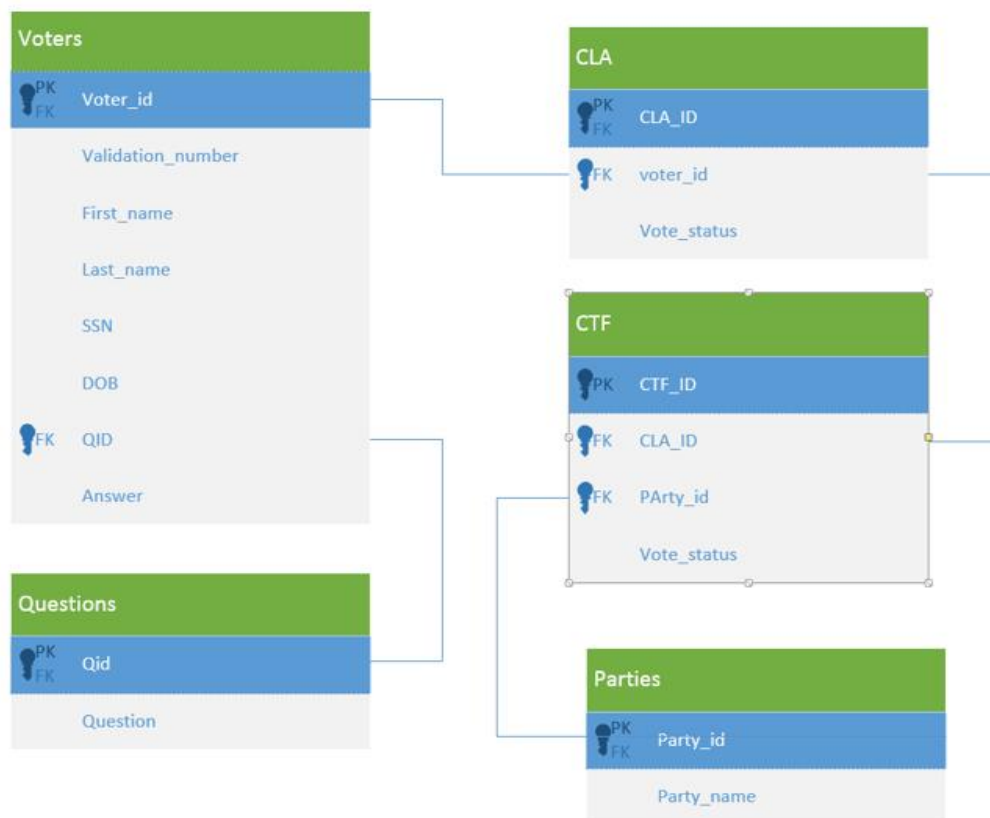
Figure 3. Database Design

   The project database has five tables as shown in Figure 3. The primary keys and foreign

keys can be seen in the tables as they are displayed with a key symbol and a PK and a FK

designation. The sensitive information like Voter_ID and social security numbers are hashed

which helps in ensuring security even if the database is hacked. The data in the database will be

of no use to the hacker because the information that is required to perform voting and other

sensitive information is hashed.

The database is designed in an intelligent way where there is no way to view the sensitive

information, the validation number of any voter cannot be viewed, and the sensitive information

is hashed, therefore even if the database is compromised there is no way to compromised the

data of the voters is not.

Sequence Diagrams
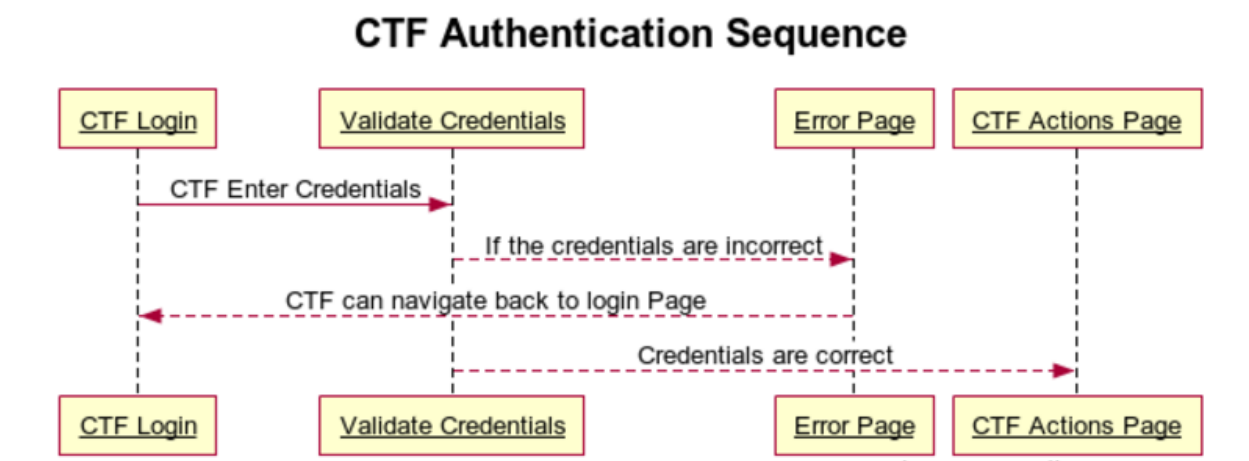
Figure 4. CTF Authentication Sequence Diagram

Even the CTF will be asked to log-in and the entered credentials are validated on the

backend and decide if they should be allowed to proceed further. Once the virtual election

system has been validated successfully and it allows the CTF to proceed.

After the CTF has logged in successfully and then logs out they will need to log in again

to perform any action, when they click on the back arrow in the browser they will be able to see

the actions stored in the cache. But when they tried to perform any action they will not be able to

perform them because the details of the session are already invalidated, therefore they will need

to log in again to perform any of the five actions.
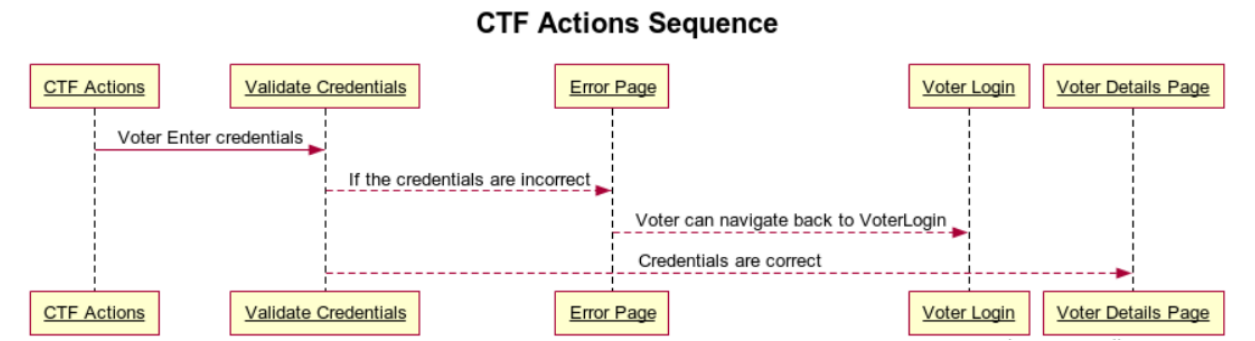
Figure 5. CTF Actions Sequence Diagram

If the details entered by the CTF in the login page are invalid, then they will be brought

to an error page notifying them that their credentials entered are invalid and provides a "Login"

button which helps them to navigate back to the login page and enter their credentials again.

CTF can perform any of five different actions as seen below:

1. Add Party/Candidates

    CTF can add a party and a candidate by clicking on the Add Party/candidates in the

    virtual election system UI. The virtual election system saves the details of the

    candidate and which party the candidate belongs to in the database. Therefore the

    voters can view which candidate belongs to which party and provides details specific

    to each party before the voter casts a vote.

2. Start Election Process

    It is the CTF which decides when the election process will start and can start the

    election process by logging in and clicking on the "Start Election Process" in the

    virtual election booth. A jQuery modal will then popup and ask for confirmation and

once the CTF confirms to start the election process the voters will then be allowed to cast their votes.

3. Stop Election Process

Once the election deadline has arrived it is the CTF which is authorized to stop the election process by clicking on the stop election process and confirming it. Once this action is performed by the CTF the user will no longer be able to vote and will be notified that the election process has been completed and no further voting will be allowed to vote.

4. Show Election Results

After the Election process, the CTF decides when to release the details and once the CTF has performed the action to show election results the voter can view the results by logging in.

5. Lock Results

As the elections are in progress or if CTF has decided to hide the election results it can lock the results by performing the Lock Results action.

Voter Authentication Sequence
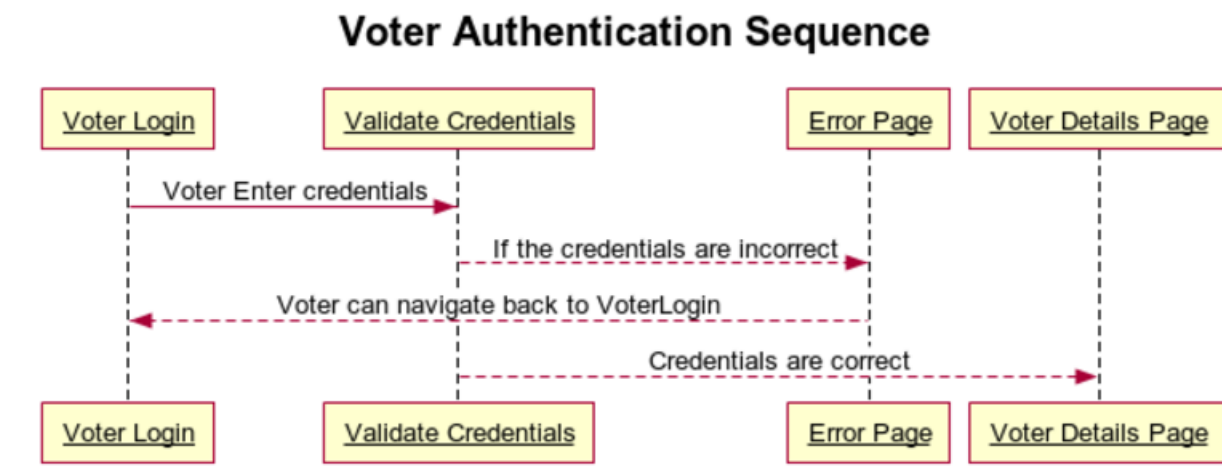
Figure 6. Voter Authentication Sequence

The voter first has to navigate to the virtual election system login page and enter the validation number and password, the validation number is saved as a hash value in the database, therefore the validation number entered by the voter is converted to a SHA-1 hash and is compared in the database and validated. If the entered credentials are correct, and if the validation is successful the voter will be able to login to the virtual election system or will be notified that the credentials are incorrect in a bootstrap warning element.

After a successful login the voter will be brought to a Home page where their details along with the party and candidate which they cast their vote for is displayed in a table, if the voter has not yet cast a vote, the Voted Party and Candidate Name columns are left empty. The action items are displayed on a black menu, the following are the actions the voter can perform:

1. Parties Information

    The voter can navigate to the Parties Information page by clicking on "Parties Information" in the black horizontal menu, initially the voter can see all the party names and if the voter is interested to view the candidates in a specific party they can click on the party name and the virtual election system displays the candidates in the specific party.

2. Vote

    The voter can navigate to the vote page by clicking on "Vote" in the black horizontal menu, this page displays a table which has two columns named "Party Name" and "Candidates". The "Party Name" column displays the name of the party and the "Candidates" column displays the candidates where each candidate is a radio

button. Once the voter has decided whom to cast their vote for, they can click on the

candidate and click on the "Vote" button.

Once the voter has clicked on the "Vote" button, the voter will receive a

notification in a jQuery popup with the message that they have voted, therefore the

voter can be assured that their vote has been taken into consideration.

After the vote has been taken into consideration, it means the information is saved

in the database and now if the voter is navigated back to the home page, the voted

party and candidate name columns will have the information that they has voted for.

However, the voter will be able to cast the vote only if the CTF has already

started the election process, the virtual election booth is intelligent enough to allow

the voter to view the page only if the CTF has approved to start the election process

and therefore if election process is not started by the CTF then the voter will receive a

notification stating that the voting process has not yet begun and therefore return later

to cast your vote.

3. Results

The voter can navigate to the Results page to view the results. The Results are

shown in two tables, one table shows the total votes each candidate has received and

the other table shows the total number of votes each party has received.

However, the results will only be displayed to the voter only if the CTF has

approved to show the results, therefore if the CTF has not approved then the voter

will receive a notification message in bootstrap warning element with the appropriate

message.

4. Analysis

   The voter can view the results in the form a pie chart on this page. The voter can

navigate to analysis page by clicking on the "Analysis" on the black horizontal menu.

Chapter V: Tools and Techniques

Tools

1. Eclipse IDE is used to develop the application which allows to code efficiently and allows to structure the project and use various plugins for the implementation of virtual election booth.

2. Apache Tomcat 7 servers are used to deploy the application.

3. Workbench, this a GUI for my SQL database to view the data in the database and query.

Techniques

Spring Framework is implemented to achieve model view controller pattern which helps in developing the application efficiently. The java servlet pages are the view part which has the hypertext markup language code and JavaScript.

The DAO classes are the model part which has the business logic implemented in these java classes and models are the regular POJO classes which have the getters and setters.

The MD5 Hashing algorithm is used to hash the sensitive details (social security number and validation number) to ensure security.

Unified modelling language to develop the project design diagrams which helps in the implementation stages.

The J2EE technologies and techniques (servlets, java server pages, collections, and design patterns) are used in developing the virtual election booth.

Hardware Requirements

Items	Minimum Value
Memory	4.0 GB
Processor Type	32/64 bit
Processor Speed	1Ghz
OS	Windows/Mac/Linux

Software Requirements

Technology	J2EE, JSP, jQuery, bootstrap, HTML, CSS
Database	My SQL
Server	Apache Tomcat
Framework	Spring
Operating System	Windows
IDE	Java Net beans

Chapter VI: Project UI

Screenshots
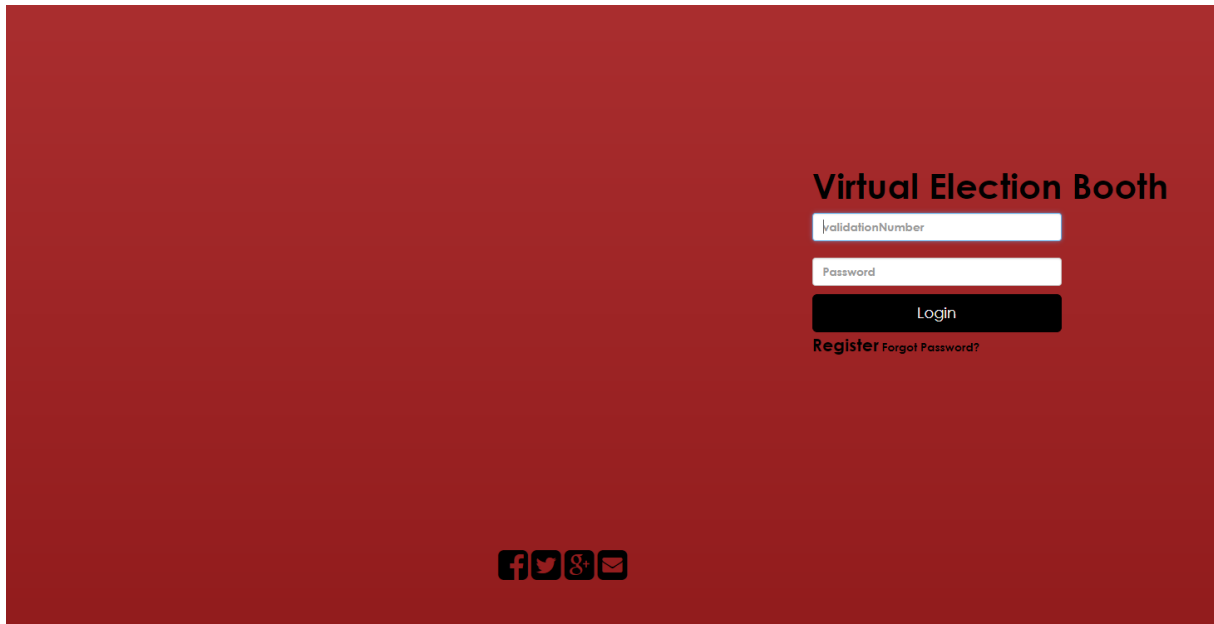
Figure 7. Voter Login Page

The voter has to navigate to the virtual election system login page as shown in the above figure

and enter the validation number and the password, the entered details are validated and the

virtual election system is intelligently designed to display the appropriate message or

successfully navigate to the voter home page which displays a table consisting of his/her details.

The validation number entered by the voter is converted to MD5 hash and compared with

the hash value saved in the database at the time of the voter registration and validates the

validation number generated by the virtual election system and the password created by the voter

at the time of registration.
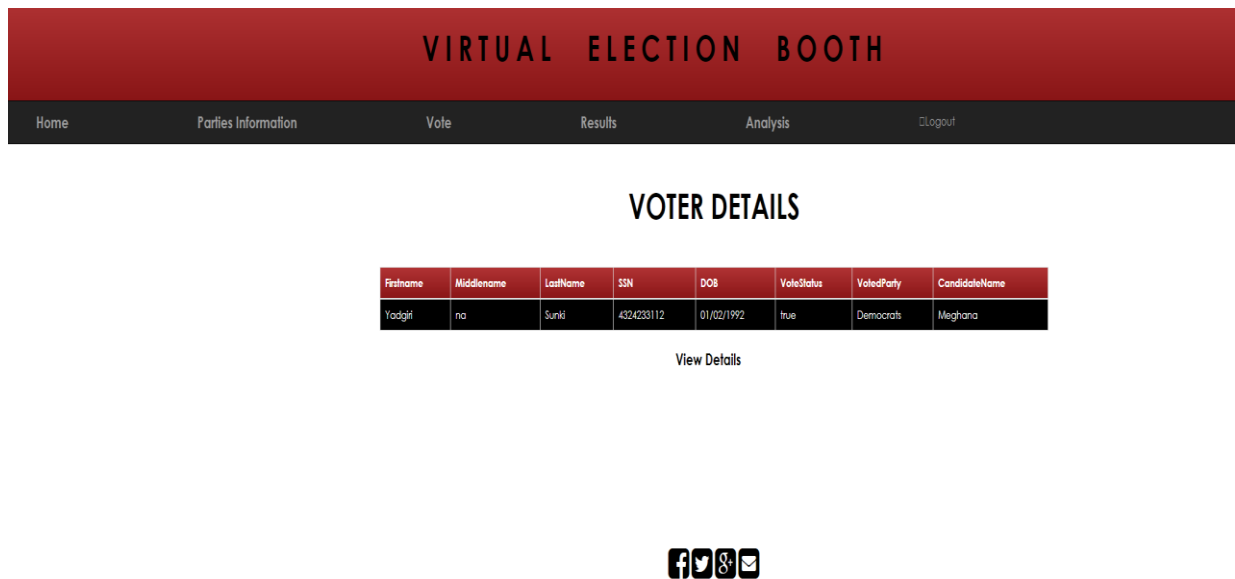
Figure 8. Voter Home Page

The page displayed above in Figure 8 is the voter home page which is displayed by the virtual election system after successfully validating the credentials entered by the voter on the login page. The voter can navigate to the parties information page, vote page, results page, analysis page or log out from the virtual election booth.

After first entering the virtual election booth the voter will not be able to view their personal details on the table until they click on the voter details link. The voted party and candidate name columns are empty until the voter has cast their vote, once the voted has cast the vote the party name and candidate which they voted for will then be displayed, therefore the voter can be sure that their vote has been taken into consideration correctly.

When Voter tries to vote before the voting process has started:
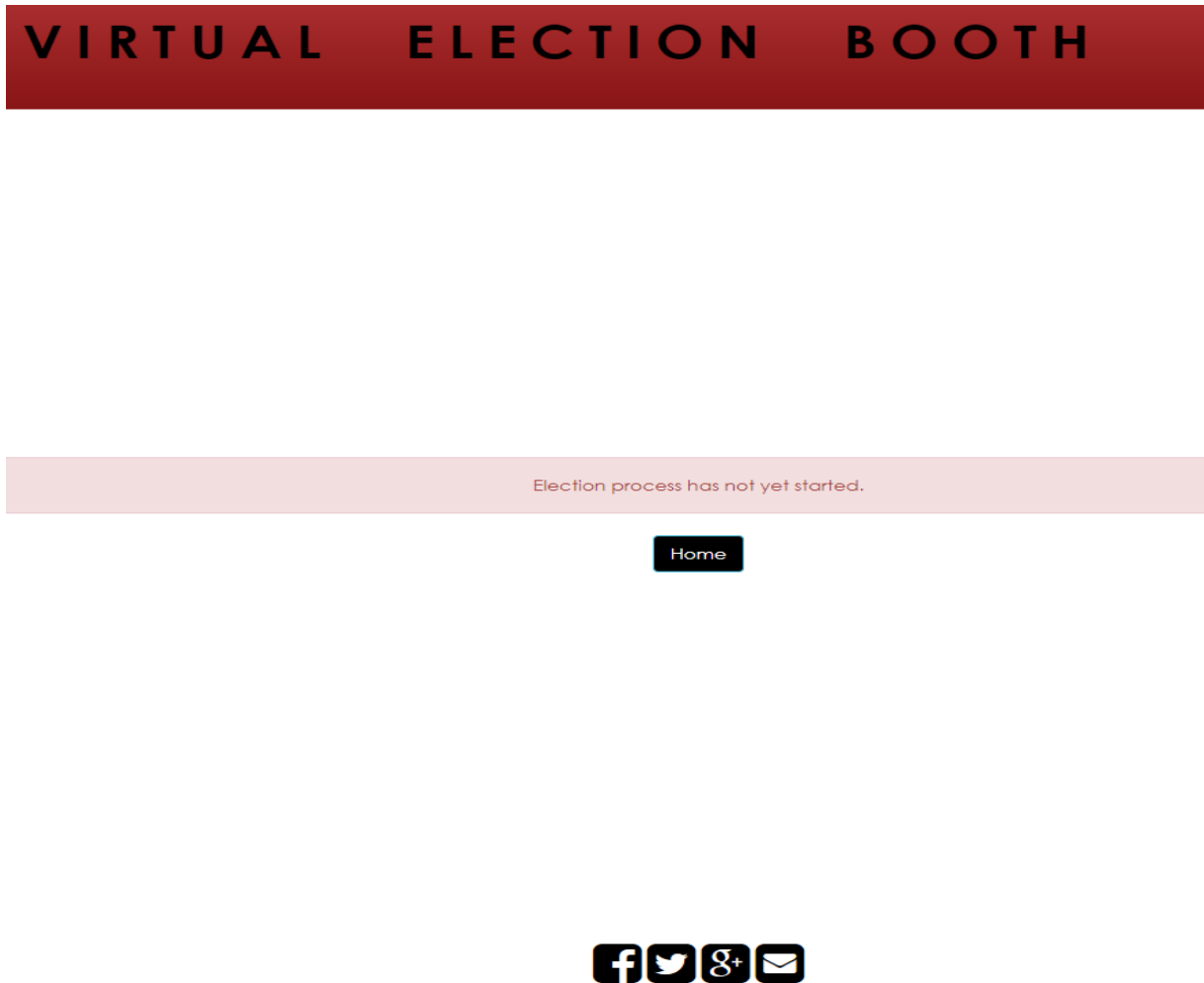
Figure 9. Election Process Not Started Error Message

The above-displayed bootstrap warning message is displayed when the voter tries to navigate to the vote page to cast his vote before the election process has approved to start by the CTF.

The virtual election system intelligently checks if the election process should be started by validating the database to verify if the CTF has approved to start the election process.

When Session is invalid:
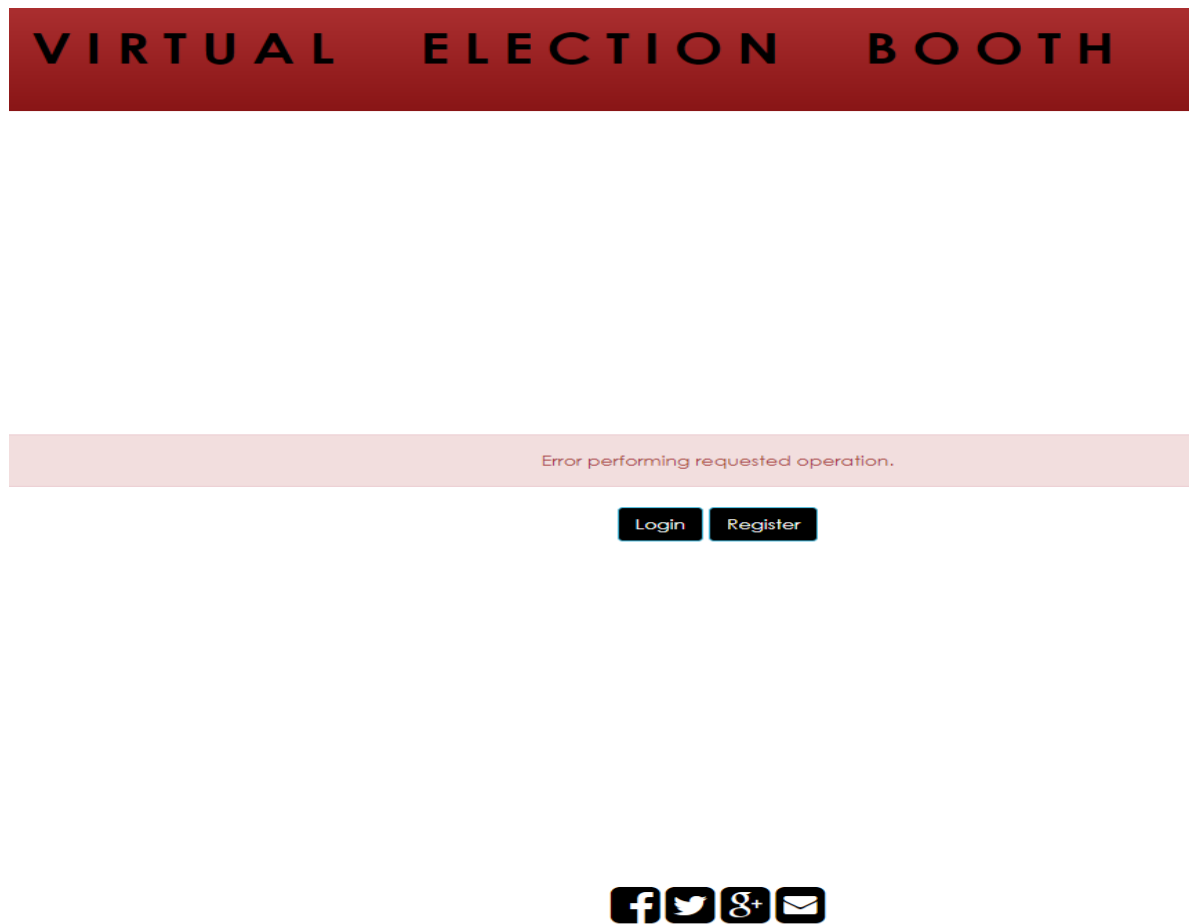
Figure 10. Session Invalid Error Message

In Figure10 the bootstrap error message "Error performing requested operation" is displayed when the user tries to perform any action after the session is invalidated. This usually occurs when the voter or CTF has logged out of the virtual election system and clicks on the back arrow in the browser to navigate to the previous page and then tries to perform any operation on that page. Once the voter or CTF has clicked on logout the session gets invalidated, i.e. the data in the session is deleted and the virtual election system will no longer be able to identify the user that is logged in or trying to perform the operation.

This is designed to develop a secured virtual election booth. There are chances that the voter or CTF has logged out of the application but the data is stored in the browser cache and the next person who uses the computer will be able to navigate back to it, but now because the session has been invalidated, once the user is logged out there no way to perform any action.

When Voter tries to view results (locked):
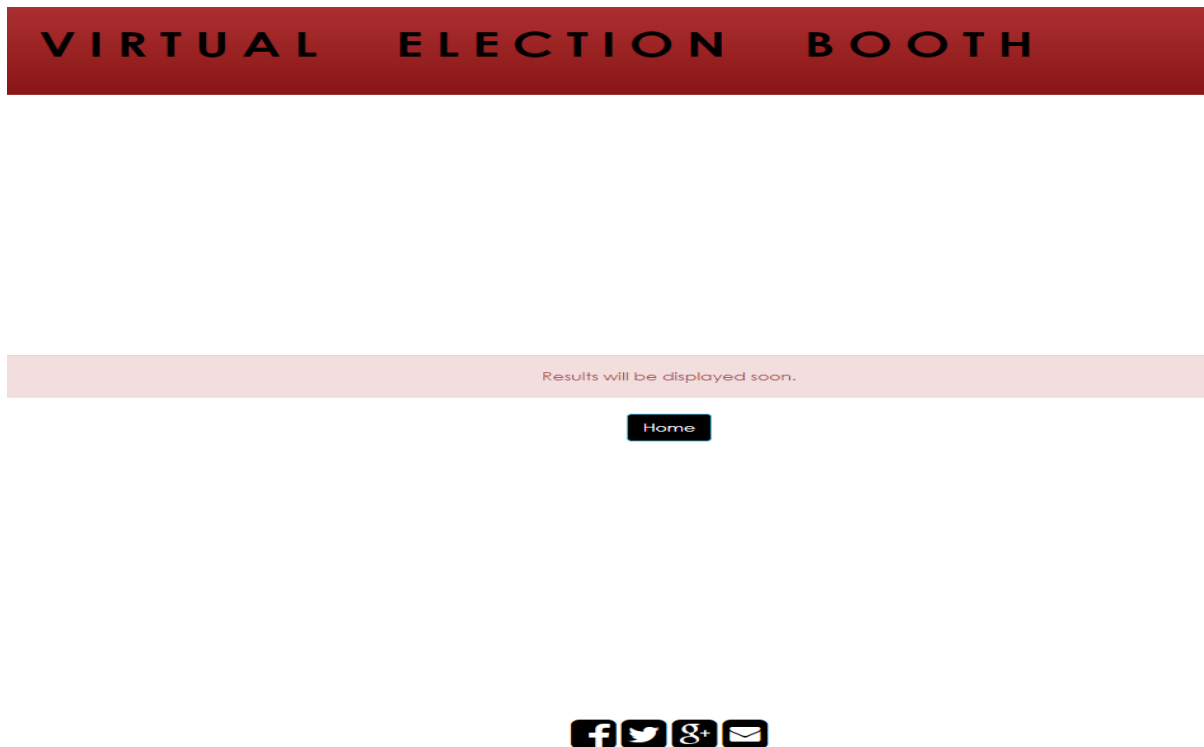
Figure 11. Results Locked Error Message

Figure 11 shows the message "Results will be displayed soon" which will be displayed when the voter tries to view the results before the CTF has approved that the voters can view the results.

The virtual election booth will validate in the database and validates if the CTF has approved to show the results to the voters, if yes then- the results are displayed in two different

tables, one table showing the total count of votes for each party and the other table showing the total vote count for each candidate.

Figure 12. Voter Registration Page

The first step for any voter is to register by entering the details in the registration page which can be seen in Figure 12, the entered details will be validated with the data in the database and decide if the user is authentic and generates a unique validation number for each voter.

The voter is asked to enter a security question of their choice and answer, the security question, and the answer is then saved in the database for the specific user and this information is used by the virtual election to generate a new validation number.

There is a chance that the voter might forget thieir password, but there is no need to worry, the voter can navigate to the password change page and answer the security question which they entered at the time of registration. The virtual election system then validates the answer to the security question and if it is correct it allows the voter to create a new password which will override the previous password.
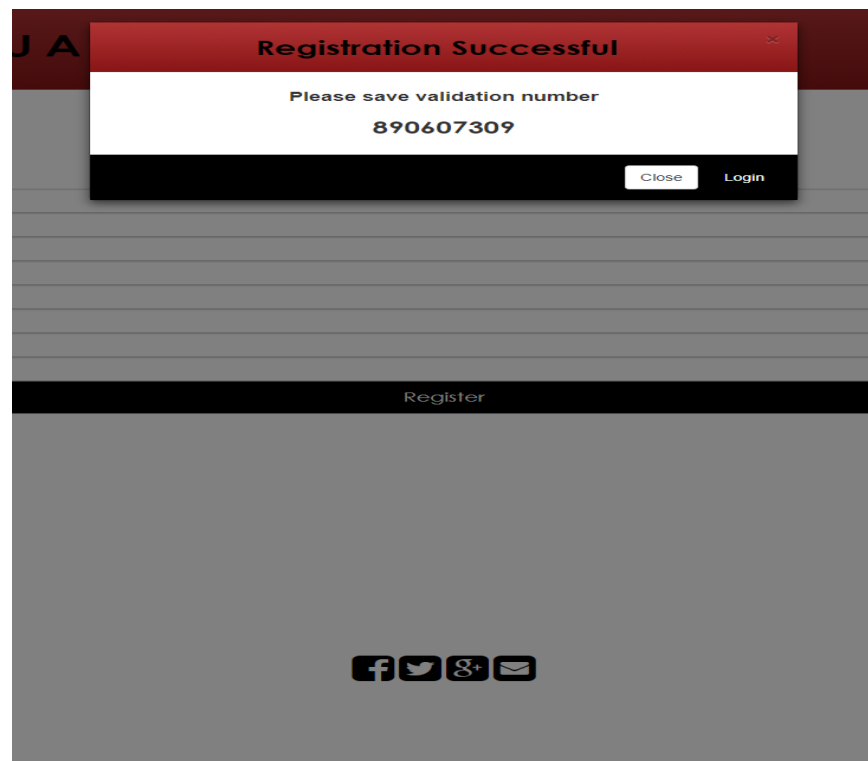
Figure 13. Registration Succesful

The details entered on the registration page are validate and the virtual election system generates a unique validation number if the information entered is correct, as seen in Figure13. The voter can use this validation number along with the password they created on the registration page to login to the virtual election booth and cast their vote, or view results or view party information.
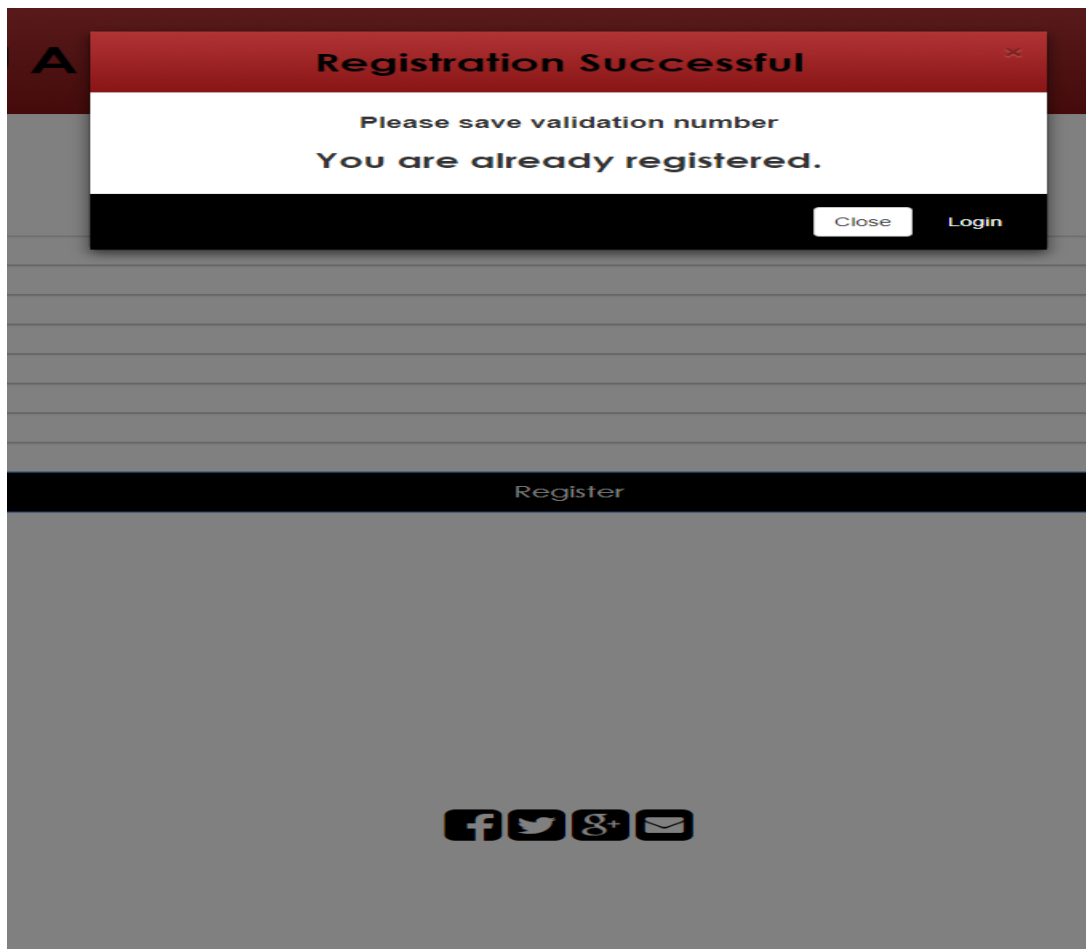
Figure 14. Registration Failure

If the user who has already registered tries to register again, the virtual election system is intelligent enough to identify that they have already registered in the virtual election system and does not allow to register again and displays the message."You are already registered" as seen in Figure 14.
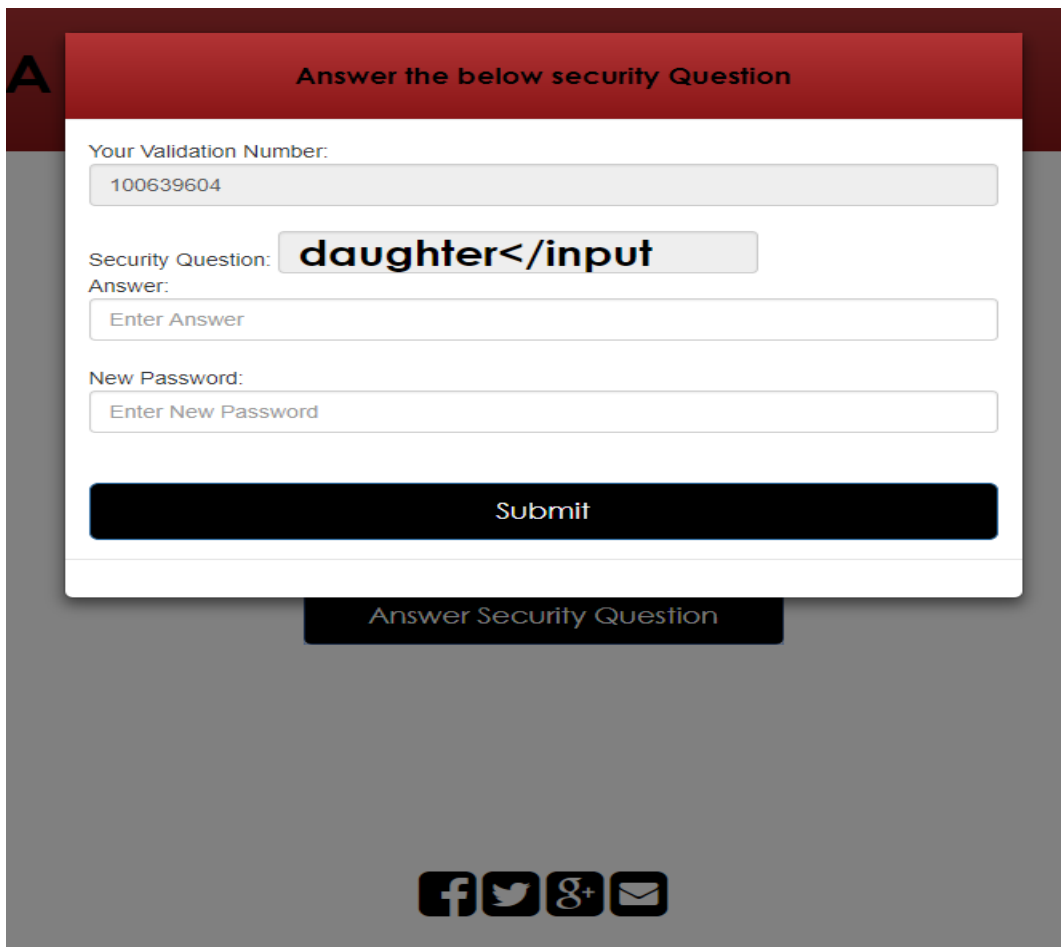
Figure 15. Forgot Password Page

The user can navigate to the forgot password page seen in Figure 15 from the login page

if they have forgotten their password. Once they are on the forgot password page they are asked

to enter their validation number and if the validation number is already in the database then they

will see a bootstrap modal displayed where they are asked to enter the answer for the security

question and create anew password as seen in Figure 16.

Figure 16. Security Question Modal

In Figure 16, the voter has to enter the answer to the security question and the new password. The virtual election system validates if the answer for the security question is correct and if the validation is successful the new password will replace the old password and the voter will receive a notification saying that the password was successfully changed and they will be able to login to virtual election booth with the newly created password.
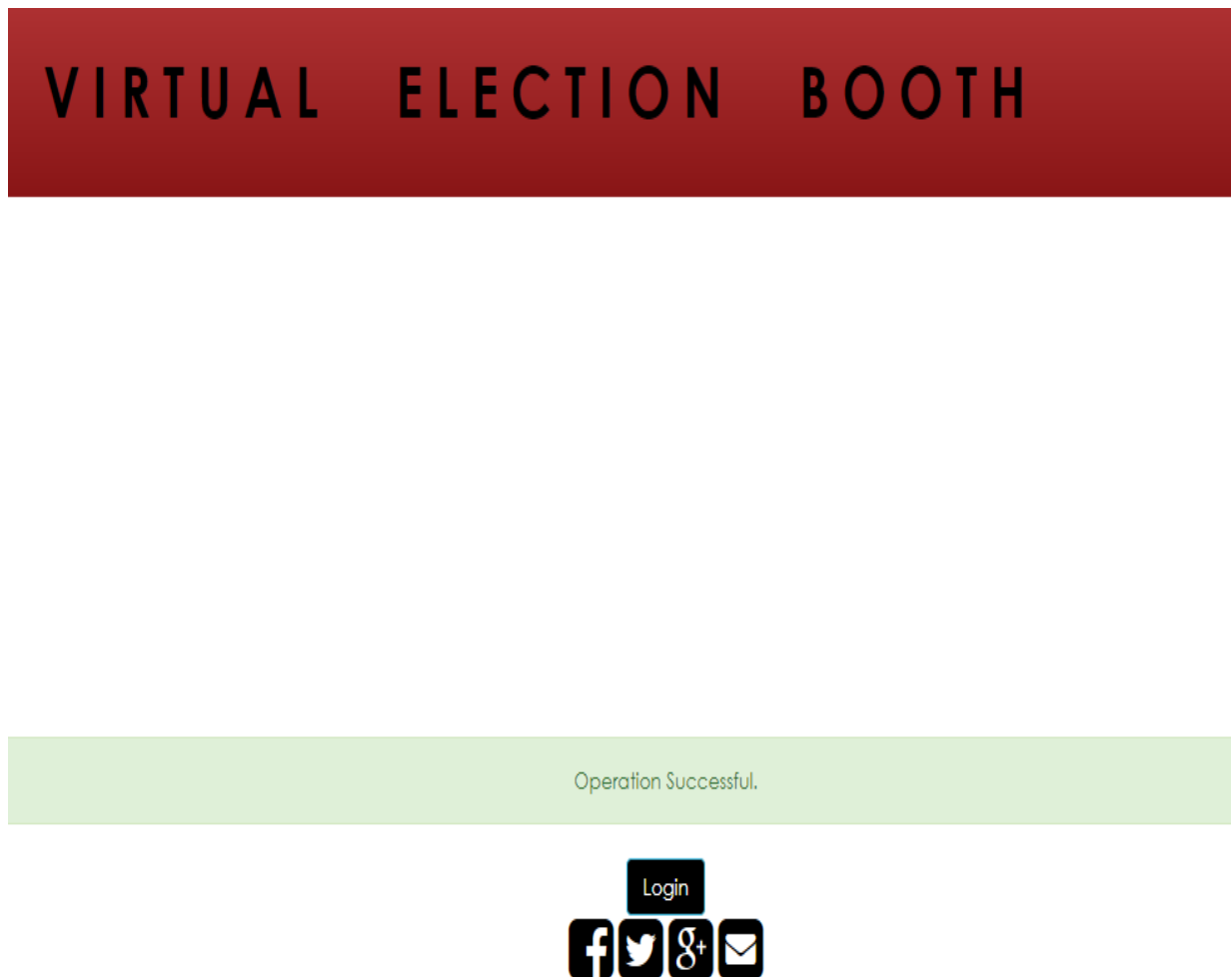
Figure 17. Password Changed Success Message

Once the answer to the security question is validated and the validation is successful the virtual election booth notifies the voter with the bootstrap success element displayed in Figure 17 with the appropriate message so that the user can be sure that their password has been updated.
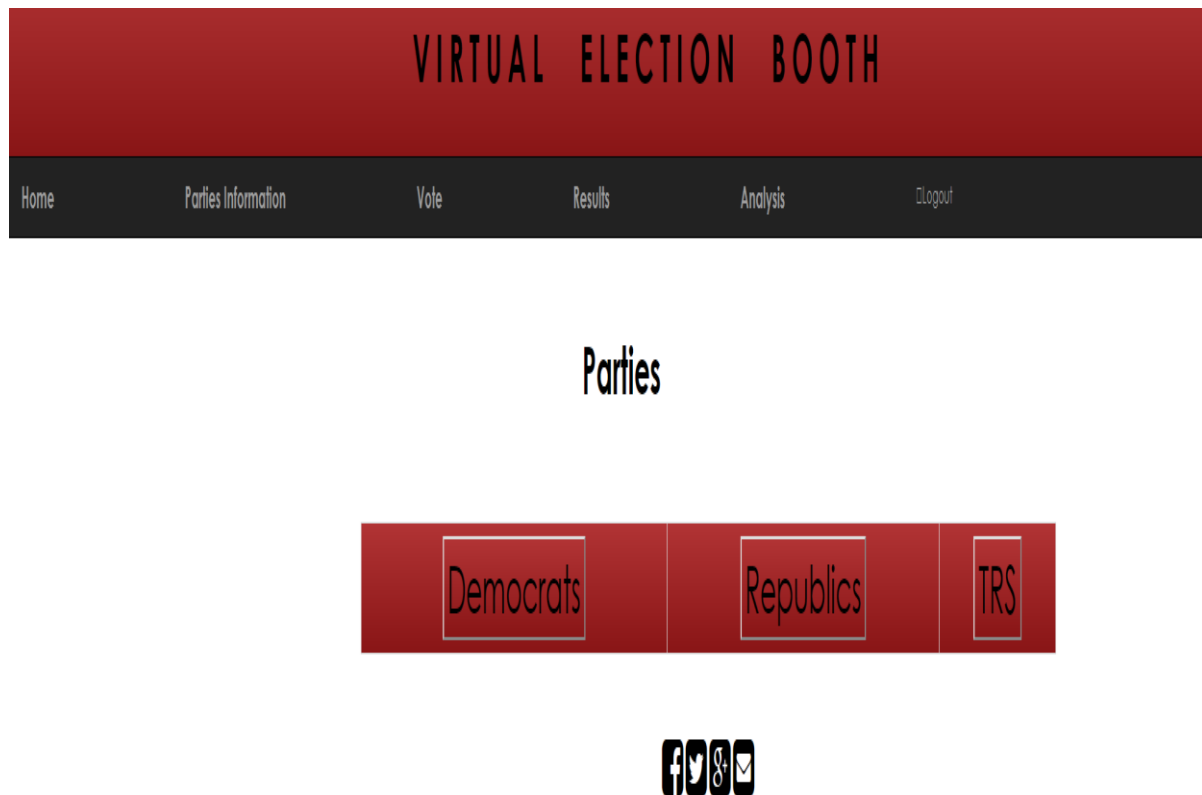
Figure 18. Parties Information Page

The voter can navigate to the parties page by clicking on the black horizontal menu, and the parties page will be displayed as shown in Figure 18.

Once this page is displayed the user can click on the party which they are interested in, once they clicks on the party a new table will be displayed dynamically with the help of Ajax on the same page without the need to refresh the page.

Figure 19. Candidates Information

As seen in Figure 19 the voter can view all the candidates in each party by clicking on the party name.

VIRTUAL ELECTION BOOTH

Vote | Results | Analysis | Logout

Desktop Search Engine Market Share, Dec-2012

● Hillary Clinton
● Donald Trump

Donald Trump 8.16%

Hillary Clinton 83.24%

Trial Version
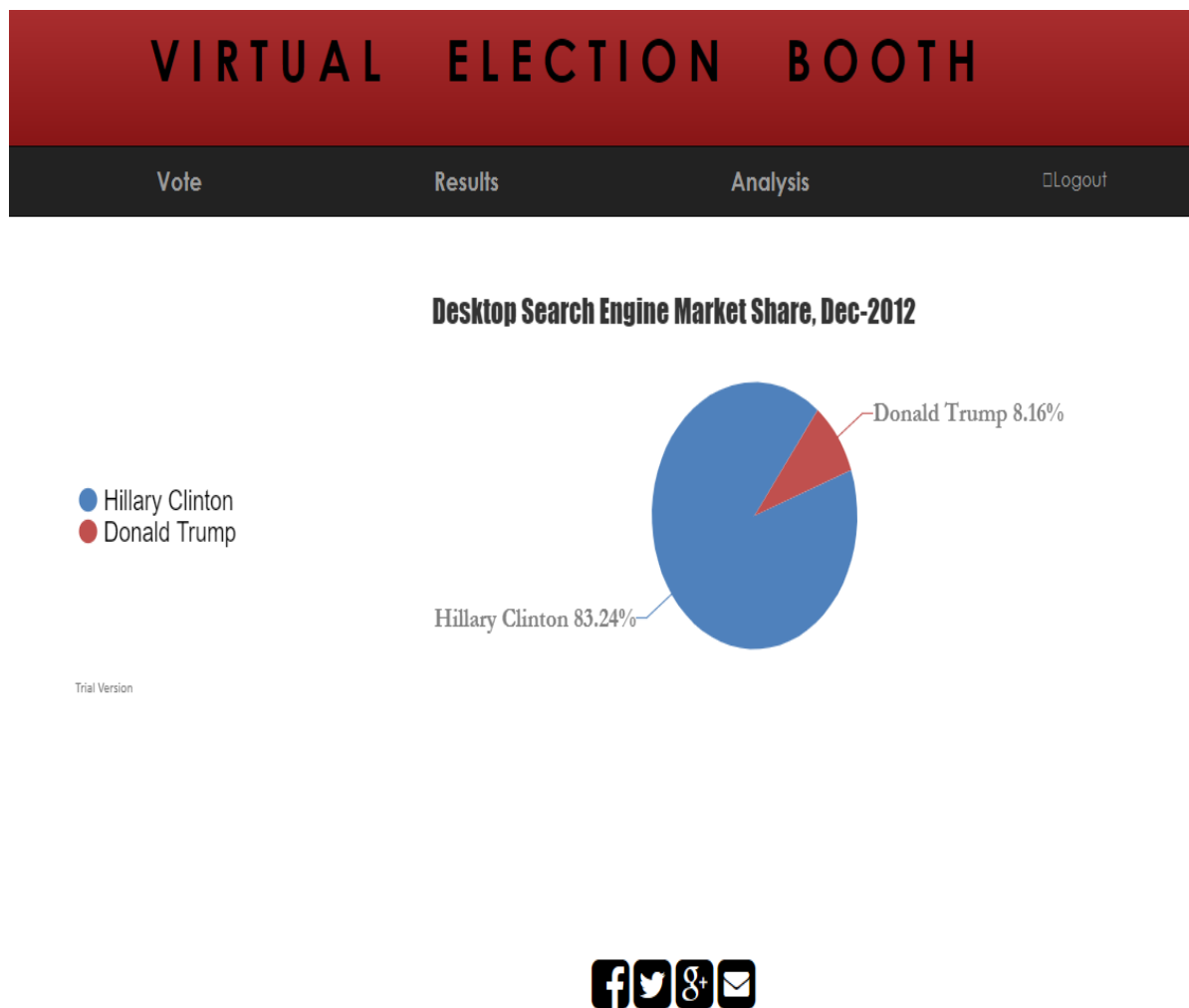
Figure 20. Vote Analysis Page

As seen in Figure 20 the voter can navigate to the analysis page to view the results in the form of a pie chart.

VIRTUAL ELECTION BOOTH

Login as Admin

admin

•••••

Login

Figure 21. CTF Login Page

Similar to the voter, the CTF is asked to enter credentials before they can log in to perform any action. The credentials entered by the CTF are validated with the information in the database and come to a conclusion if they should be authorized to proceed further.
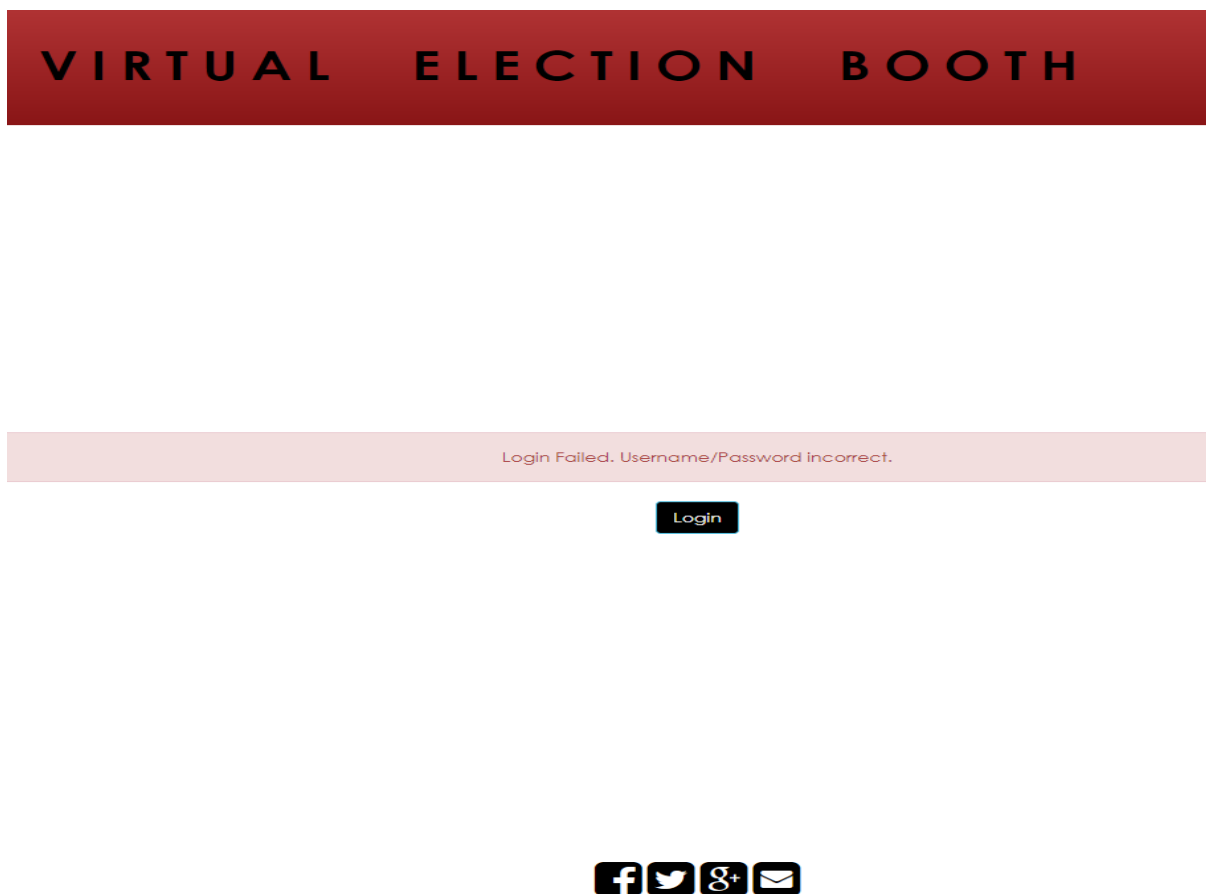
Figure 22. CTF Login Failure Error Message

If the credentials entered by the CTF are invalid then they will be notified by the

bootstrap error element with the message "Login Failed. Username/Password incorrect" and not

allowed to proceed further and therefore cannot perform any CTF action.
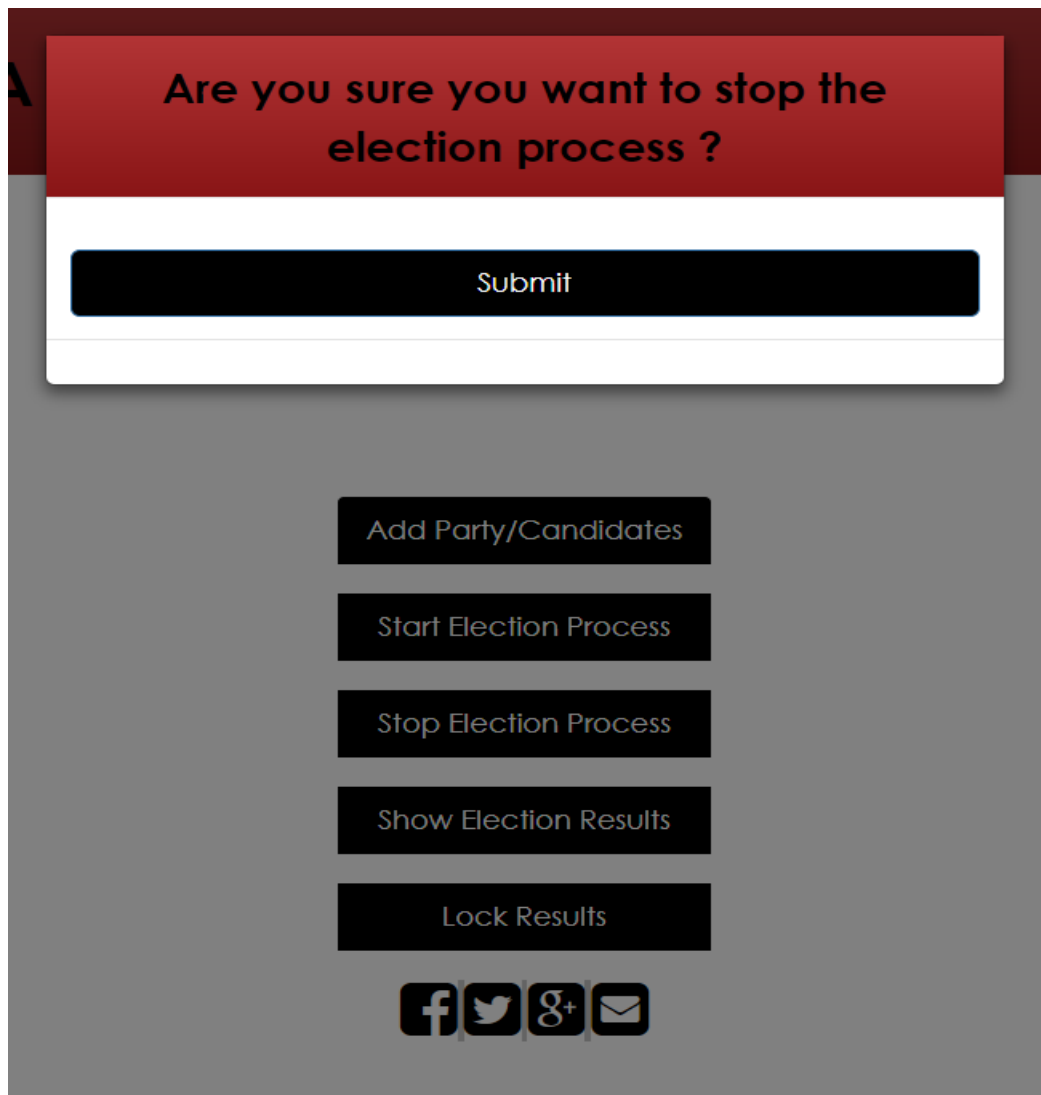
Figure 23. CTF Stop Election Process Modal

Once the CTF is logged in successfully they can stop the election process by clicking on the black link "Stop Election Process" as seen in Figure 23 and will see a confirmation bootstrap modal where they can submit and it will change the database and therefore the election process will be stopped and no voter can cast their vote.
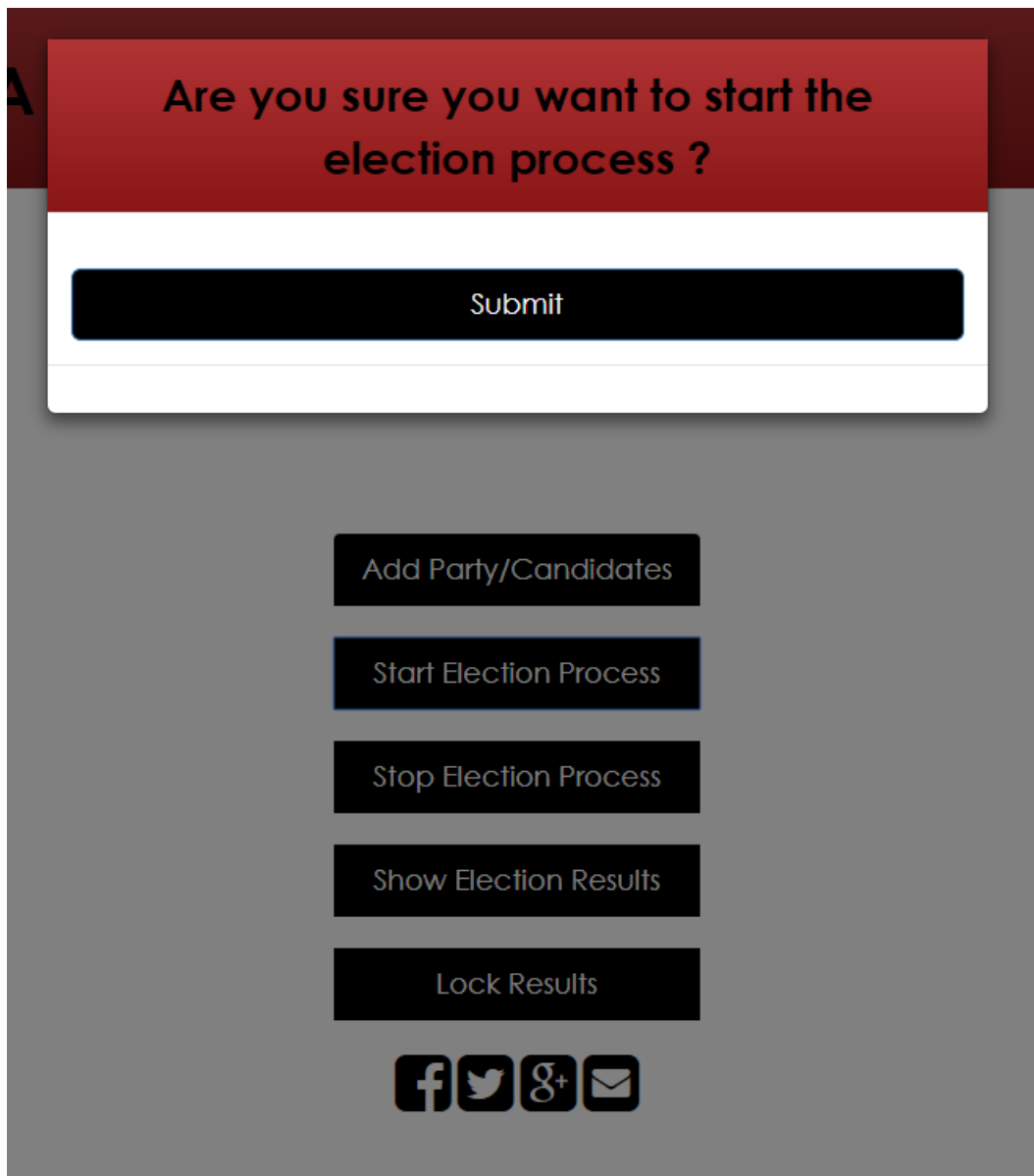
Figure 24. CTF Start Election Process Modal

Similar to stop election process action, the CTF can start the election process, once this action is performed by the CTF the voters can cast their vote, as seen in Figure 24.
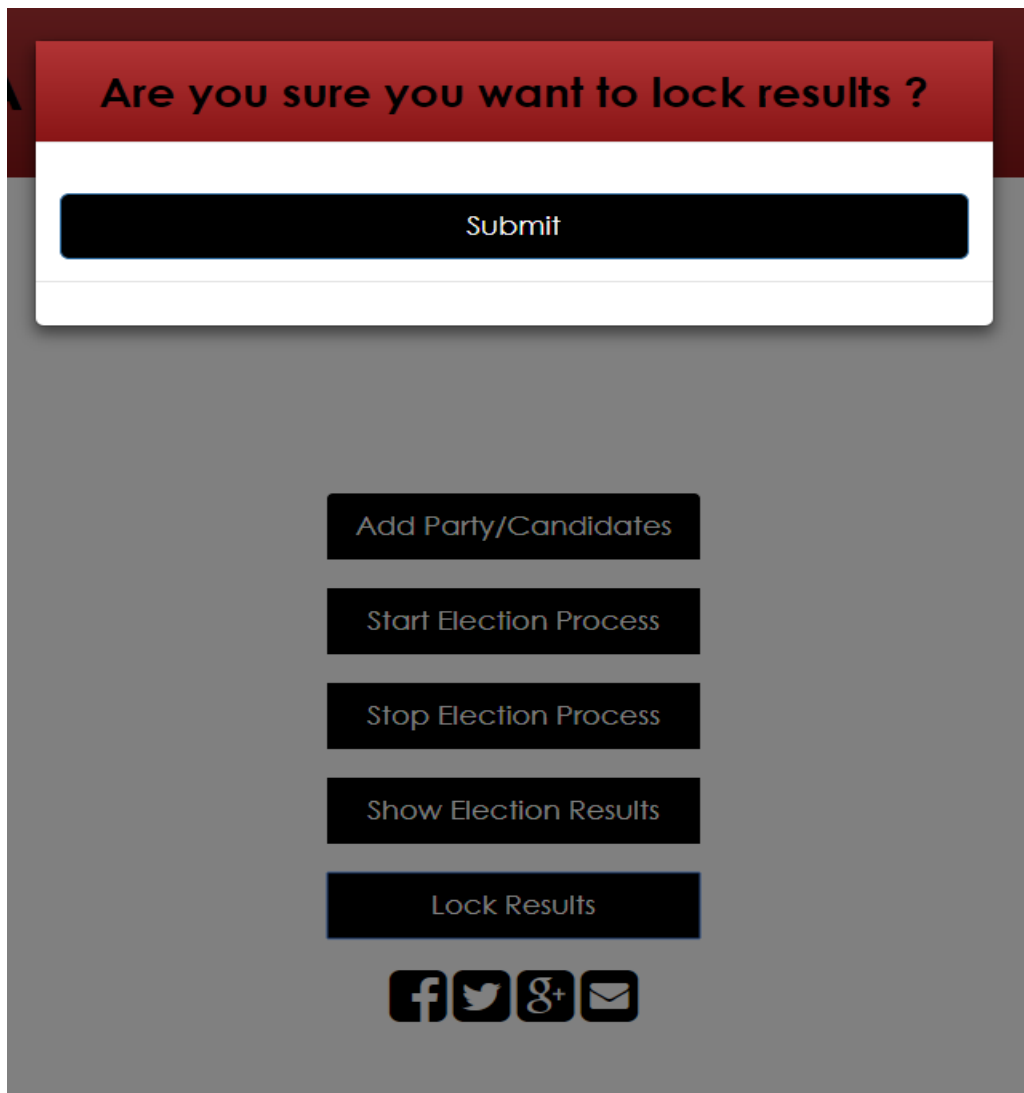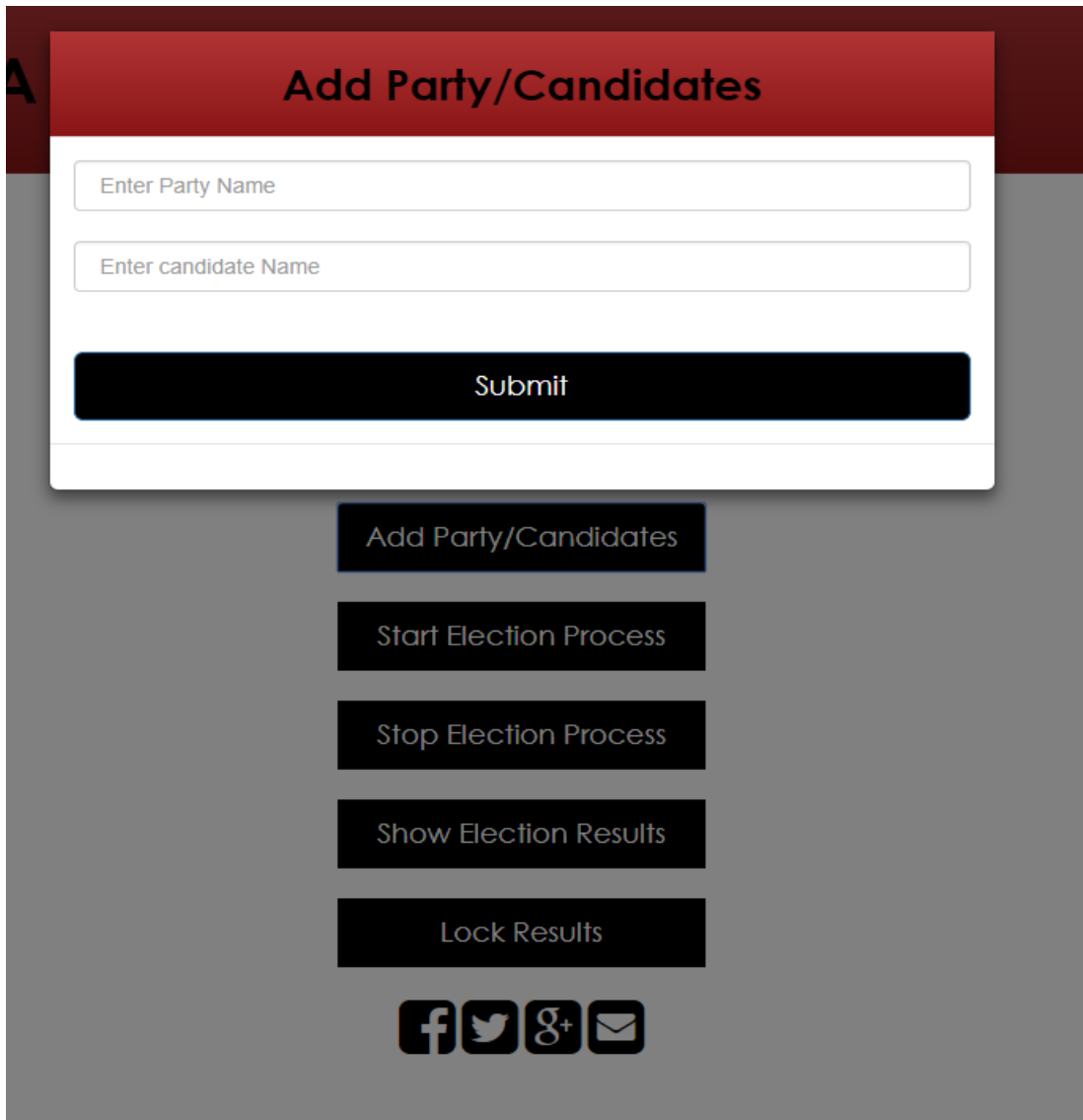
Figure 25. CTF Lock Results Modal

The voters cannot view the results if the CTF has locked the results as shown in Figure 25.

Figure 26. CTF Add Party and Candidate Modal

It is the CTF's responsibility to add parties and candidates, the data entered here is securely saved in the database. Once the party and candidate information is entered by the CTF, the voter will be able to see the information of the parties information page and also on the vote page where the voter can cast their vote.

Project Structure
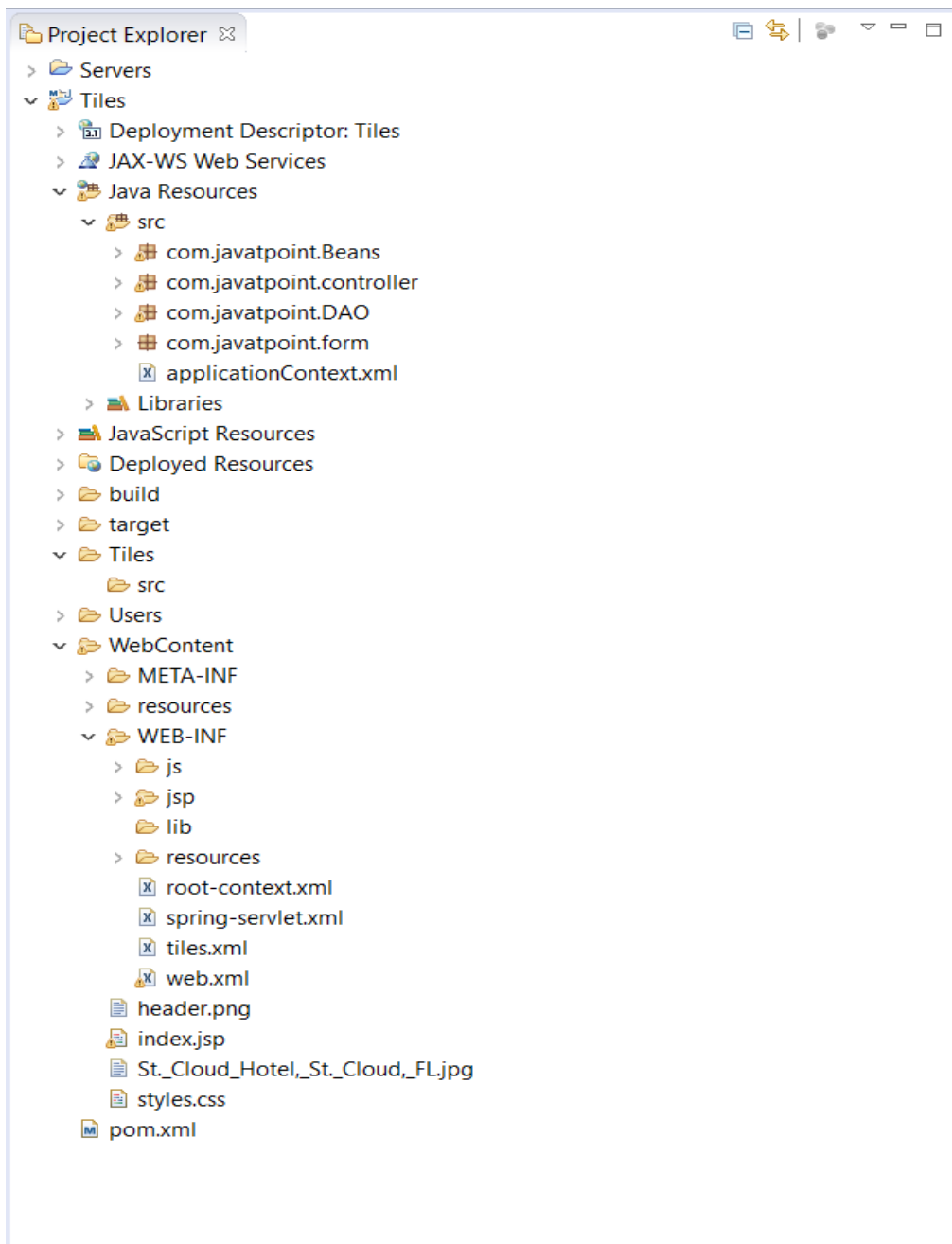
Figure 27. Project Structure

Figure 27 displays the project structure of virtual election booth, this is a screenshot from

the Eclipse IDE. The java classes reside in "src" folder which again contains 4 packages inside it.

Beans package consists of all java models which contains the getters and the setters, the

controller has all the spring controller classes which are responsible to display the appropriate page depending on the business logic and the URL and finally the DAO package consists of all the java classes which have the database connectivity and business logic which involves the connection to the database.

WEB-INF folder consists of javascript and java server pages which are responsible to display the data on the browser, tiles.xml is used to organize the website in header, footer and body sections to simplify the development process and spring-servlet.xml is the heart of the spring framework which takes care of the lifecycle of each and every java object and helps in dependency injection.

.

Chapter VII: Threat Scenarios

In this chapter, threat scenarios in the virtual election system are described and also it explains how the mentioned threat is handled by the virtual election system.

Threat 1: Only authorized voters can vote

The virtual election system allows a voter or CTF to log in only after successful validation of the entered credentials. Each voter will receive a unique validation number at the time of registration which is to save secretly by the voter and use this validation number along with the password created at the time of registration to login to the virtual election system.

Threat 2: Voter may try to vote multiple times

The Virtual election system is designed and developed intelligently in a way that it allows any voter to vote exactly only once and only to one candidate. However, the voter can perform his vote multiple times until the voting process is stopped by the CTF but it will override the previous vote, therefore resulting only in one vote.

Threat 3: The voter's vote might be disclosed

The voter's vote is not disclosed to anybody except to him/her, even the CTF/CLA does not know to whom any voter has voted. The CLA can just know if the voter has voted or not, CTF can know the total vote count for each candidate and each party has received but cannot know which voter has voted for whom.

Threat 4: Vote might be duplicated

There is no chance of duplication of votes by any voter, although the voter can vote multiple times if his/her mind changes but the previous vote will be overridden therefore resulting in only one vote without any duplication.

Threat 5: Votes might be duplicated in the final tabulation

CTF is responsible for the final tabulation, there no chance of duplication even at the time of final tabulation because even the CTF will not know which voter has voted for whom.

Threat 6: Voters vote might not be acknowledged

Voter will be acknowledged to whom he voted for in the home page. The table in the home page gets updated dynamically even if the voter has voted for a different candidate.

CIA Implementation

Confidentiality, Integrity and Availability are three main security goals of the virtual election booth which addresses the following three goals:

1. Protect the confidentiality of the voters

2. Preserve the integrity of the voters and the casted vote details

3. Promote the availability of the virtual election system to vote from a remote location

Confidentiality. To protect the confidentiality of the voters is one of the three security goals of the virtual election system. Various techniques have been implemented to provide confidentiality to the voters.

In the first step, when the voter enters the details in the registration page and clicks on submit, the virtual election system validates the information entered by the voter and generates an alphanumeric validation number which is unique for each voter. This validation number is used by the voters to login into the system.

The generated validation number and all the sensitive information of the voter are hashed using the MD5 hash algorithm.Therefore, the data cannot be compromised from the database as the sensitive information is hashed.

Even the CTF and CLA cannot know the details of the voter, CLA can just know the if the voter

has voted but cannot know who the specific voter has voted for.

    Integrity. All the necessary precautions have been taken to protect the integrity of the

data. At any point of time, the virtual election system takes care to protect the integrity of the

data. When the voter tries to vote the data is saved either after the voter has cast is vote

completely or the system will remain the same before the vote has been cast.

    Availability. It is the CTF who is authorized to start the elections, stop the elections, show

the results or lock the results. Once the CTF has authorized to start the elections the virtual

election system is available to the voters 24*7 until the CTF again stops the election process.

Outcomes

    The outcomes of the Virtual election system were achieved with the help latest

technological advancements in developing a J2EE application and also implementing security

and cryptographic algorithms.

    The outcomes of the virtual election system are as follows:

1. The Virtual Election system is intelligently designed and therefore allows only

    authorized voters to vote.

    a. The voter should first register before actually being able to enter the virtual

        election system, the system validates the information entered by the voter to

        verify if it is correct and then creates a unique validation number or notify an

        appropriate warning or error message.

    b. The voter can vote multiple times before the election process deadline has

        reached, but each time the voter has voted it overrides the previous vote,

therefore, resulting in one vote for one voter and does not allow duplicate votes.

2. The Voter can vote multiple times before the voting process has stopped by CTF.

    a. The voter can vote multiple times before the election process has been stopped by the CTF, but this does not mean that the voter can vote for multiple candidates or duplicate votes. Each time the voter votes after his first attempt the previous vote gets overridden and virtual election system does not allow duplicacy.

3. The voter's vote is secured and will not be disclosed to anyone except the person who voted.

    a. Even the CTF or CLA does not know which voter has voted for whom, this is designed keeping in mind to reduce the fraud chance to the maximum.

    b. The CLA can know the total vote count for each party and each candidate but cannot know which voter has voted for which candidate or which party.

    c. The CTF can only know if the voter has voted or not but cannot know which party or candidate the voter has cast his vote for.

4. The status of the voter's vote can be known by logging in by the voter on the home page.

5. The Analysis page in the virtual election system shows the vote count in the form a pie chart.

6. In case the voter has forgotten his password he can change his password by correctly answering the security questions, the newly generated password will override the old password.

7. The validation number that represents each voter is hashed using an MD5 hash algorithm and saved in the database which keeps the voter's data secure.

Conclusion

Conduction election in a fast and secured manner along with eliminating the wastage of natural resources and reducing economic burden achieved by the virtual election booth. The concept of developing a virtual election booth which can be accessed on the internet at the convenience of the voter encourages a number of voters to vote and therefore helps in the progress of the institution, organization or country where the virtual election booth is used. The Parties and candidates can be added by CTF in seconds which is dynamically added to the database and displayed to the voters, similarly, CTF can also start, stop, show or lock the results with a button click which is secured. The Virtual election system developed with latest technologies implementing the concept of CTF and CLA along with hashing algorithms and secured database eliminates the problems that were faced in old and present election systems.

Limitations

In this prototype the virtual election system was not performance tested, it is not known about the amount of load it can handle. Load testing, stress testing, spike testing, endurance testing, scalability testing, volume testing has not been performed and a load balancer could be implemented to handle huge amount traffic.

Also, the application should be deployed in better infrastructure environment consisting of multiple servers and cloud technologies in a place like dockers and also microservice patterns are not implemented which will increase the robustness of the application.

Future Work

The virtual election system developed does not use any web services, cloud technologies or dockers, the electronic system can be developed using latest advancements in the technology and develop a more stable application even at the time of high traffic and also for an easy man of the maintenance application.

References

Arthanayaka, A. (2017). XML based security model for enhancing Integrity and Privacy of the electronic voting system.SCSU Thesis.

Bitar, N., Gringeri, S., & Xia, T. J. (2013). Technologies and protocols for data centre and cloud networking. *Communications Magazine, IEEE, 51*(9), 24-31.

Boneh, D., & Hamburg, M. (2008). Generalized identity based and broadcast encryption schemes. In J. Peiprzyk (Ed.), *Advances in Cryptology-ASIACRYPT 2008* (Vol. 5350). Berlin Heidelberg, Germany: Springer

Cui, B., Liu, Z., & Wang, L. (2015). Key-aggregate searchable encryption (KASE) for group data sharing via cloud storage. *Computers, IEEE Transactions On, PP*(99), 1-1.

Dutta, R., & Annappa, B. (2014). Protection of data in an unsecured public cloud environment with open, vulnerable networks using threshold-based secret sharing. *Network Protocols and Algorithms, 6*(1), 58-75.

Krishna, N. S. R., & Prasd, Y. V. (2015). A novel system for scalable data sharing in cloud storage using key-aggregate. *IJSEAT, 3*(9), 461-465.

Lewko, A., & Waters, B. (2011). Decentralizing attribute-based encryption. In *Advances in Cryptology–EUROCRYPT 2011* (pp. 568-588). Berlin Heidelberg, Germany: Springer.

Nurmi, D., Wolski, C., Grzegorczyk, G., Obertelli, S., Soman, L., . . . Zagordonov, D. (2009). *The eucalyptus open-source cloud-computing syste*m. Proceedings of 2009 ACM/IEEE International Conference on Grid Computing (pp. 124-131).

Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., & Jahanian, F. (2008). *Virtualized in-cloud security services for mobile devices*. Proceedings of the First Workshop on Virtualization in Mobile Computing (pp. 31-35).

Okuhara, M., Shiozaki, T., & Suzuki, T. (2010). Security architecture for cloud computing. Fujitsu Sci. *Tech. J, 46*(4), 397-402.

Ramgovind, S., Eloff, M. M., & Smith E. (2010). *The management of security in cloud computing*. Retrieved from http://uir.unisa.ac.za/bitstream/handle/10500/3883/ramgovind.pdf?sequence=1

Reddy, M. S. K., & Yadav, T. S. (2015). Public-key patient-controlled encryption for flexible data sharing in cloud storage. *International Journal of Innovative Technologies, 3*(1), 0005-0007.

Ruj, S., Nayak, A., & Stojmenovic, I. (2011, November). Dacc: Distributed access control in clouds. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on* (pp. 91-98). IEEE.

Schoo, P., Fusenig, V., Souza, V., Melo, M., Murray, P., Debar, H., . . . Zeghlache, D. (2011). *Challenges for cloud networking security*. Paper presented at the International Conference on Mobile Networks and Management (pp. 298-313). Berlin Heidelberg, Germany: Springer.

Wu, T. Y., Zhou, C., Wang, E. K., Pan, J. S., & Chen, C. M. (2014). Towards time-bound hierarchical key management in cloud computing. In *Intelligent Data Analysis and Its Applications*, Volume I (pp. 31-38). New York, NY: Springer International Publishing.

Zhu, Z., Jiang, Z., & Jiang, R. (2013). *The attack on Mona: Secure multi-owner data sharing for dynamic groups in the cloud*. Proceedings of the 2013 International Conference on Information Science and Cloud Computing Companion (pp. 514-519).

Appendix

BEAN Classes:

AddParty.java

```java
package com.javatpoint.Beans;

public class AddParty {
        String Party;
        String CandidateName;
        public String getParty() {
                return Party;
        }
        public void setParty(String party) {
                Party = party;
        }
        public String getCandidateName() {
                return CandidateName;
        }
        public void setCandidateName(String candidateName) {
                CandidateName = candidateName;
        }
        @Override
        public String toString() {
                return "AddParty [Party=" + Party + ", CandidateName=" + CandidateName + "]";
        }

}
```

AdminLoginBean.java

```java
package com.javatpoint.Beans;

import org.springframework.web.context.annotation.SessionScope;
l
@SessionScope
public class AdminLoginBean {

        String username;
        String password;

        public String getUsername() {
                return username;
        }
        public void setUsername(String username) {
                this.username = username;
        }
        public String getPassword() {
                return password;
        }
        public void setPassword(String password) {
                this.password = password;
        }
```

```java
        @Override
        public String toString() {
                return "AdminLoginBean [username=" + username + ", password=" + password + "]";
        }
}
```

CandidateNamesBean.java

```java
package com.javatpoint.Beans;

import java.util.ArrayList;

public class CandidateNames {

        ArrayList<String> candidateNames;

        public ArrayList<String> getCandidateNames() {
                return candidateNames;
        }

        public void setCandidateNames(ArrayList<String> candidateNames) {
                this.candidateNames = candidateNames;
        }

        @Override
        public String toString() {
                return "candidateNames [candidateNames=" + candidateNames + "]";
        }

}
```

ForgotPasswordBean.java

```java
package com.javatpoint.Beans;

public class ForgotPasswordBean {
        String validationNumber;
        String securityQuestion;
        String answer;
        String newPassword;

        public String getNewPassword() {
                return newPassword;
        }
        public void setNewPassword(String newPassword) {
                this.newPassword = newPassword;
        }
        public String getValidationNumber() {
                return validationNumber;
        }
        public void setValidationNumber(String validationNumber) {
                this.validationNumber = validationNumber;
        }
```

```java
        public String getSecurityQuestion() {
                return securityQuestion;
        }
        public void setSecurityQuestion(String securityQuestion) {
                this.securityQuestion = securityQuestion;
        }
        public String getAnswer() {
                return answer;
        }
        public void setAnswer(String answer) {
                this.answer = answer;
        }
        @Override
        public String toString() {
                return "ForgotPasswordBean [validationNumber=" + validationNumber + ", securityQuestion="
+ securityQuestion
                                        + ", answer=" + answer + ", newPassword=" + newPassword + "]";
        }

}
```

LoginBean.java

```java
package com.javatpoint.Beans;

public class LoginBean {
        private String validationNumber;
        private String password;

        public String getValidationNumber() {
                return validationNumber;
        }
        public void setValidationNumber(String validationNumber) {
                this.validationNumber = validationNumber;
        }
        public String getPassword() {
                return password;
        }
        public void setPassword(String password) {
                this.password = password;
        }

}
```

MySQLBean.java

```java
package com.javatpoint.Beans;

public class MySQL {
        String Url;
        String userName;
        String passowrd;
        String className;
```

```java
        public String getUrl() {
                return Url;
        }

        public void setUrl(String url) {
                Url = url;
        }

        public String getUserName() {
                return userName;
        }

        public void setUserName(String userName) {
                this.userName = userName;
        }

        public String getPassowrd() {
                return passowrd;
        }

        public void setPassowrd(String passowrd) {
                this.passowrd = passowrd;
        }

        public String getClassName() {
                return className;
        }

        public void setClassName(String className) {
                this.className = className;
        }

}
```

PartyAndCandidateBean.java

```java
package com.javatpoint.Beans;

import java.util.List;

public class PartyAndCandidateBean {
        String Party;
        String Candidate;
        public String getParty() {
                return Party;
        }
        public void setParty(String party) {
                Party = party;
        }
        public String getCandidate() {
                return Candidate;
        }
        public void setCandidate(String candidate) {
                Candidate = candidate;
```

```
            }

    }




RegisterBean.java
package com.javatpoint.Beans;

public class RegisterBean {
        public String firstname;
        public String middlename;
        public String lastname;
        public String dob;
        public String ssn;
        public String question;
        public String answer;
        public String password;

        public String getPassword() {
                return password;
        }

        public void setPassword(String password) {
                this.password = password;
        }

        public String getFirstname() {
                return firstname;
        }

        public void setFirstname(String firstname) {
                this.firstname = firstname;
        }

        public String getMiddlename() {
                return middlename;
        }

        public void setMiddlename(String middlename) {
                this.middlename = middlename;
        }

        public String getLastname() {
                return lastname;
        }

        public void setLastname(String lastname) {
                this.lastname = lastname;
        }

        public String getDob() {
                return dob;
        }
```

```java
        public void setDob(String dob) {
                this.dob = dob;
        }

        public String getSsn() {
                return ssn;
        }

        public void setSsn(String ssn) {
                this.ssn = ssn;
        }

        @Override
        public String toString() {
                return "RegisterBean [firstname=" + firstname + ", middlename=" + middlename + ", lastname="
+ lastname
                                        + ", dob=" + dob + ", ssn=" + ssn + ", question=" + question + ", answer=" +
answer + ", password="
                                        + password + "]";
        }

        public String getQuestion() {
                return question;
        }

        public void setQuestion(String question) {
                this.question = question;
        }

        public String getAnswer() {
                return answer;
        }

        public void setAnswer(String answer) {
                this.answer = answer;
        }

}
```

ResultsBean.java

package com.javatpoint.Beans;

```java
public class Results {
        String party_name;
        String candidate_name;
        int totalVotes;

        public String getParty_name() {
                return party_name;
        }
        public void setParty_name(String party_name) {
                this.party_name = party_name;
        }
```

```java
        public String getCandidate_name() {
                return candidate_name;
        }
        public void setCandidate_name(String candidate_name) {
                this.candidate_name = candidate_name;
        }
        public int getTotalVotes() {
                return totalVotes;
        }
        public void setTotalVotes(int totalVotes) {
                this.totalVotes = totalVotes;
        }

        @Override
        public String toString() {
                return "Results [party_name=" + party_name + ", candidate_name=" + candidate_name + ",
totalVotes=" + totalVotes
                                                + "]";
        }
}
```

User.java

```java
package com.javatpoint.Beans;

public class User {
        String voter_id;
        String first_name;
        String middle_name;
        String last_name;
        String ssn;
        String dob;

        String vote_status;
        String votedparty;
        String candidateName;

        public String getCandidateName() {
                return candidateName;
        }
        public void setCandidateName(String candidateName) {
                this.candidateName = candidateName;
        }
        public String getVoter_id() {
                return voter_id;
        }
        public void setVoter_id(String voter_id) {
                this.voter_id = voter_id;
        }
        public String getFirst_name() {
                return first_name;
        }
        public void setFirst_name(String first_name) {
```

```java
                this.first_name = first_name;
        }
        public String getMiddle_name() {
                return middle_name;
        }
        public void setMiddle_name(String middle_name) {
                this.middle_name = middle_name;
        }
        public String getLast_name() {
                return last_name;
        }
        public void setLast_name(String last_name) {
                this.last_name = last_name;
        }
        public String getSsn() {
                return ssn;
        }
        public void setSsn(String ssn) {
                this.ssn = ssn;
        }
        public String getDob() {
                return dob;
        }
        public void setDob(String dob) {
                this.dob = dob;
        }
        public String getVote_status() {
                return vote_status;
        }
        @Override
        public String toString() {
                return "User [voter_id=" + voter_id + ", first_name=" + first_name + ", middle_name=" +
middle_name
                                + ", last_name=" + last_name + ", ssn=" + ssn + ", dob=" + dob + ",
vote_status=" + vote_status
                                + ", votedparty=" + votedparty + "]";
        }
        public void setVote_status(String vote_status) {
                this.vote_status = vote_status;
        }
        public String getVotedparty() {
                return votedparty;
        }
        public void setVotedparty(String votedparty) {
                this.votedparty = votedparty;
        }


}
```

Validation.java

package com.javatpoint.Beans;

```java
public class validation {
        String status;
        Object reuslt;
        public String getStatus() {
                return status;
        }
        public void setStatus(String status) {
                this.status = status;
        }
        public Object getReuslt() {
                return reuslt;
        }
        public void setReuslt(Object reuslt) {
                this.reuslt = reuslt;
        }
}
```

VotedParty.java

package com.javatpoint.Beans;

```java
public class VotedParty {
        String partyAndCandidateName;
        String validationNumber;
        public String getPartyAndCandidateName() {
                return partyAndCandidateName;
        }
        public void setPartyAndCandidateName(String partyAndCandidateName) {
                this.partyAndCandidateName = partyAndCandidateName;
        }
        public String getValidationNumber() {
                return validationNumber;
        }
        public void setValidationNumber(String validationNumber) {
                this.validationNumber = validationNumber;
        }
        @Override
        public String toString() {
                return    "VotedParty    [partyAndCandidateName="    +    partyAndCandidateName    +    ",
validationNumber=" + validationNumber
                                        + "]";
        }
}
```

CONTROLLERS

CTFActionController.java

package com.javatpoint.controller;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

```java
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;

import com.javatpoint.Beans.AddParty;
import com.javatpoint.Beans.AdminLoginBean;
import com.javatpoint.DAO.AddPartyDAO;
import com.javatpoint.DAO.AdminLoginDAO;
import com.javatpoint.DAO.LockResultsDAO;
import com.javatpoint.DAO.ShowResultsDAO;
import com.javatpoint.DAO.StartElectionsDAO;
import com.javatpoint.DAO.StopElectionsDAO;

@Controller
@Scope("session")
public class AdminActionsController {
        @Autowired
        HttpSession session;

        @RequestMapping(value = "/admin")
        public ModelAndView adminAccess() {
                return new ModelAndView("admin");
        }
        @RequestMapping(value = "/adminLogin", method = RequestMethod.POST)
        public ModelAndView adminActions(HttpServletRequest request, HttpServletResponse response,
                        @ModelAttribute("adminloginBean") AdminLoginBean adminLoginBean) {
                System.out.println("in admin login");
                ModelAndView mv = null;
            session.setAttribute("User","admin");
                        AdminLoginDAO adminLoginDAO = new AdminLoginDAO();
                        String result = adminLoginDAO.adminLoginVerification(adminLoginBean);
                        mv= new ModelAndView(result);
                return mv;
        }
        @RequestMapping(value = "/addParty", method=RequestMethod.POST)
        public ModelAndView addParty(HttpServletRequest request, HttpServletResponse response,
                        @ModelAttribute("addParty") AddParty addParty) {
                String userName = (String) session.getAttribute("User");
                ModelAndView mv = null;
                if(null== userName) {
                        mv = new ModelAndView("adminError");
                }else {
                        AddPartyDAO addPartyDAO = new AddPartyDAO();
                        String result = addPartyDAO.addPartyAndCandidate(addParty);
                        mv = new ModelAndView(result, "message", "Success!");
                }
                return mv;
        }
```

```java
@RequestMapping(value = "/startElection", method=RequestMethod.POST)
public String startElection(HttpServletRequest request, HttpServletResponse response) {
    String userName = (String) session.getAttribute("User");
    String result="adminError";
    System.out.println("Session val: "+ userName);
    if(null!= userName){
        StartElectionsDAO StartElections = new StartElectionsDAO();
        result = StartElections.startElectionProcess();
    }
    System.out.println("Result:"+result);
    return result;
}

@RequestMapping(value = "/stopElection", method=RequestMethod.POST)
public String stopElection(HttpServletRequest request, HttpServletResponse response) {
    String userName = (String) session.getAttribute("User");
    String result="adminError";
    System.out.println("Session val: "+ userName);
    if(null!= userName){
        StopElectionsDAO stopElectionsDAO = new StopElectionsDAO();
        result = stopElectionsDAO.stopElectionProcess();
    }
    System.out.println("Result:"+result);
    return result;
}

@RequestMapping(value = "/showResults", method=RequestMethod.POST)
public String showResults(HttpServletRequest request, HttpServletResponse response) {
    String userName = (String) session.getAttribute("User");
    String result="adminError";
    if(null!= userName){
        ShowResultsDAO showResultsDAO = new ShowResultsDAO();
        result = showResultsDAO.stopElectionProcess();
    }
    return result;
}

@RequestMapping(value = "/lockResults", method=RequestMethod.POST)
public String lockResults(HttpServletRequest request, HttpServletResponse response) {
    String userName = (String) session.getAttribute("User");
    String result="adminError";
    if(null!= userName){
        LockResultsDAO lockResultsDAO = new LockResultsDAO();
        result = lockResultsDAO.lockResults();
    }
    System.out.println("Result:"+result);
    return result;
}
@RequestMapping(value="/logoutAdmin")
public ModelAndView logoutVoter(){
    session.invalidate();
    session.setAttribute("Error", "Session has ended.  Please login.");
    return new ModelAndView("admin");
}
}
```

VoterController.java

```java
package com.javatpoint.controller;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.Base64;
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.servlet.ModelAndView;

import com.google.gson.Gson;
import com.javatpoint.Beans.AddParty;
import com.javatpoint.Beans.AdminLoginBean;
import com.javatpoint.Beans.ForgotPasswordBean;
import com.javatpoint.Beans.LoginBean;
import com.javatpoint.Beans.RegisterBean;
import com.javatpoint.Beans.Results;
import com.javatpoint.Beans.User;
import com.javatpoint.Beans.VotedParty;
import com.javatpoint.DAO.AddPartyDAO;
import com.javatpoint.DAO.AdminLoginDAO;
import com.javatpoint.DAO.GetPartiesDao;
import com.javatpoint.DAO.GetSecurityQuestionDAO;
import com.javatpoint.DAO.LoginVerificationDAO;
import com.javatpoint.DAO.RegisterDAO;
import com.javatpoint.DAO.ShowResultsDAO;
import com.javatpoint.DAO.votePerformedDAO;
import com.javatpoint.form.Contact;

@Controller
@Scope("session")
public class HelloWorldController {

	@Autowired
	HttpSession session;
```

```java
String user;

@RequestMapping("/hello")
public ModelAndView helloWorld() {
        String userName = (String) session.getAttribute("user");
        ModelAndView mv;
        if(session.getAttribute("user")!=null) {
                LoginVerificationDAO lvDAO = new LoginVerificationDAO();
                User user1 = new User();
                user1 = lvDAO.getUserDetails(session.getAttribute("user").toString());
                mv =new ModelAndView("hello",  "user1", user1);
        }else {
                mv = new ModelAndView("voterError");
        }
        return mv;
}
@RequestMapping("/vote")
public ModelAndView vote() {
        ModelAndView mv =new ModelAndView();
        GetPartiesDao getPartiesDao = new GetPartiesDao();
        if(session.getAttribute("user")!=null) {
                if(getPartiesDao.getStatus().equals("stop")){
                        mv.setViewName("electionProcessError");
                }else{
                        HashMap<String, ArrayList<String>> parties = getPartiesDao.getParties();
                        mv.addObject("parties",parties);
                        mv.setViewName("vote");
                }
        }
        else {
                mv = new ModelAndView("voterError");
        }
        return mv;
}
@RequestMapping("/parties")
public ModelAndView parties() {
        ModelAndView mv = new ModelAndView("voterError");
        if(session.getAttribute("user")!=null) {
                GetPartiesDao gDao = new GetPartiesDao();
                ArrayList<String> list = gDao.getPartiesInfo();
                mv = new ModelAndView("parties", "list", list);
        }
        return mv;
}
@RequestMapping("/results")
public ModelAndView results() {
        HashMap<Integer,Results> res = new HashMap<>();
        ModelAndView mv = new ModelAndView("voterError");
        ShowResultsDAO showresultsDao = new ShowResultsDAO();

        if(session.getAttribute("user")!=null) {
                if(showresultsDao.getStatus().equals("stop")){
                        mv.setViewName("ResultsError");
                }
```

```java
                else{
                        res = showresultsDao.getResults();
                        for (Entry<Integer, Results> entry : res.entrySet()) {
                           System.out.println(entry.getKey()+" : "+entry.getValue());
                        }

                        HashMap<String,Integer> TotalVotesEachParty = new HashMap<>();
                        TotalVotesEachParty = showresultsDao.getTotalVotesEachParty();
                        for (Entry<String, Integer> entry : TotalVotesEachParty.entrySet()) {
                           System.out.println(entry.getKey()+" : "+entry.getValue());
                        }

                        mv.addObject("res",res);

        mv.addObject("TotalVotesForEachParty",showresultsDao.getTotalVotesEachParty());
                        mv.setViewName("results");
                }
            }
            return mv;
        }
        @RequestMapping("/analysis")
        public ModelAndView analysis() {
                ModelAndView mv = new ModelAndView("voterError");
                if(session.getAttribute("user")!=null) {
                        mv = new ModelAndView("analysis");
                }
                return mv;
        }
        @RequestMapping(value = "/loginPage", method = RequestMethod.POST)
        public ModelAndView loginForm(HttpServletRequest request, HttpServletResponse response,
                        @ModelAttribute("loginBean")        LoginBean        loginBean)        throws
NoSuchAlgorithmException {
                ModelAndView mv;
                LoginVerificationDAO lvDAO = new LoginVerificationDAO();
                user = lvDAO.verifyLogin(loginBean);
                System.out.println("in /loginpage");
                if(session.getAttribute("user")!=null) {
                        mv = new ModelAndView("voterError");

                }else {
                        MessageDigest digest = MessageDigest.getInstance("SHA-256");
                        byte[]                                        hash                        =
digest.digest(loginBean.getValidationNumber().getBytes(StandardCharsets.UTF_8));
                        String encoded = Base64.getEncoder().encodeToString(hash);
                        session.setAttribute("user", encoded);
                        mv = new ModelAndView("hello");
                }
                return mv;
        }

        @RequestMapping(value = "/index")
        public ModelAndView index(HttpServletRequest request, HttpServletResponse response) {
                return new ModelAndView("index");
        }
```

```java
@RequestMapping(value = "/register")
public ModelAndView registration(HttpServletRequest request, HttpServletResponse response,
                @ModelAttribute("loginBean") LoginBean loginBean) {
        System.out.println(loginBean.getValidationNumber());
        return new ModelAndView("register", "command", new Contact());
}


@RequestMapping(value = "/forgotPassword")
public ModelAndView forgotPassword(HttpServletRequest request, HttpServletResponse response,
                @ModelAttribute("loginBean") ForgotPasswordBean forgotPasswordBean) {
        return new ModelAndView("forgotPassword");
}


@RequestMapping(value = "/getSecurityQuestion")
@ResponseBody
public String getSecurityQuestion(HttpServletRequest request, HttpServletResponse response,
                @ModelAttribute("loginBean") ForgotPasswordBean forgotPasswordBean) throws
NoSuchAlgorithmException {
        GetSecurityQuestionDAO getSecurityQuestionDAO = new GetSecurityQuestionDAO();
        String result =
getSecurityQuestionDAO.getQuestion(forgotPasswordBean.getValidationNumber());
        return result;
}


@RequestMapping(value="/party/{partyName}")
@ResponseBody
public String getCandidatesForParty(@PathVariable("partyName") String partyName){
        GetPartiesDao dao = new GetPartiesDao();
        ArrayList<String> candidates = dao.getCandidates(partyName);
        Gson gson = new Gson();
        String json = gson.toJson(candidates);
        return json;
}


@RequestMapping("/contact")
public ModelAndView showContacts() {
        return new ModelAndView("contact", "command", new Contact());
}


@RequestMapping(value = "/registerForm")
@ResponseBody
public String registrationForm(HttpServletRequest request, HttpServletResponse response,
                @ModelAttribute("registrationBean") RegisterBean registerBean) throws
NoSuchAlgorithmException {
        RegisterDAO rdao1 = new RegisterDAO();
        String result =rdao1.registerVoter(registerBean);
        return result;
}


@RequestMapping(value = "/updatePassword")
public String updatePassword(HttpServletRequest request, HttpServletResponse
response,@ModelAttribute("forgotpassword") ForgotPasswordBean forgotPasswordBean) throws
NoSuchAlgorithmException{
        GetSecurityQuestionDAO updatePwd = new GetSecurityQuestionDAO();
```

```java
                String result = updatePwd.overridePassword(forgotPasswordBean);
                return result;
        }

        @RequestMapping(value="/logoutVoter")
        public ModelAndView logoutVoter(){
                session.invalidate();
                session.removeAttribute("user");
                System.out.println("in logout");
                return new ModelAndView("index");
        }

        @RequestMapping(value = "/votePerformed")
        public     ModelAndView    performVote(HttpServletRequest    request,  HttpServletResponse  response,
@ModelAttribute("VotedParty") VotedParty votedParty){
                ModelAndView mv ;
                votePerformedDAO vpd = new votePerformedDAO();
                votedParty.setValidationNumber(session.getAttribute("user").toString());
                System.out.println("PartyName-CandidateName:"+votedParty.getPartyAndCandidateName());
                String result = vpd.performVote(votedParty);

                mv = new ModelAndView("voterError");
                return mv;
        }
}
```

Data Access Layers

AddPartyDAO.java

package com.javatpoint.DAO;

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.AddParty;
import com.javatpoint.Beans.MySQL;



public class AddPartyDAO {
        @SuppressWarnings("finally")
        public String addPartyAndCandidate(AddParty addParty) {
                String result = "adminActions";
                int partyId;
                Boolean partyname = true;
```

```java
                        try {
                    ApplicationContext context =
                                new         ClassPathXmlApplicationContext(new             String[]
{"applicationContext.xml"});
                                MySQL mySQL = (MySQL)context.getBean("mySQL");
                                Class.forName(mySQL.getClassName()).newInstance();
                                Connection         conn         =         DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                                Statement sta = conn.createStatement();
                                Statement sta1 = conn.createStatement();
                                Statement sta2 = conn.createStatement();
                                Statement sta3 = conn.createStatement();
                                Statement sta4 = conn.createStatement();
                                Statement sta5 = conn.createStatement();

                                String Sql = "select * from parties";
                                ResultSet rs = sta.executeQuery(Sql);

                                while (rs.next()) {
//If the party is already present in database we only add the candidate name in candidates table.
                                if(rs.getString("party_name").equals(addParty.getParty())){
                                        int num=0;
                                        partyId = rs.getInt("party_id");
//inserting candidate with party id as foreign key in candidates table.

                                        String  sql  =  "select  candidate_name  from  candidates  where
party_id="+partyId;

                                        ResultSet rs2 = sta4.executeQuery(sql);
                                        int i=1;

                                        String CandidateName= "NotPresent";

                                        while(rs2.next()){

        if(rs2.getString("candidate_name").equals(addParty.getCandidateName())) {
                                                        CandidateName = "present";
                                                }
                                                i++;
                                        }
                                        if(CandidateName=="NotPresent"){
                                                String   InsertCandidate   =   "Insert   into   candidates
(party_id,candidate_name) values("+rs.getInt("party_id")+",'"+addParty.getCandidateName()+"')";
                                                sta2.executeUpdate(InsertCandidate);
                                                //result = "voterError";
                                                }
                                        partyname=false;
                                }
                                }

                        if(partyname) {
                                String      sql      =      "Insert      into      Parties      (party_id,party_name)
values(NULL,'"+addParty.getParty()+"')";
                                sta1.executeUpdate(sql);
                                if(addParty.getCandidateName()!=null){
                                        String sql_partyId = "select * from parties";
```

```
                                    rs = sta.executeQuery(sql_partyId);
                                    while(rs.next()){
                                            if(rs.getString("party_name").equals(addParty.getParty())){
                                                    String InsertCandidateName = "insert into candidates
(party_id, candidate_name) values("+rs.getString("party_id")+",'"+addParty.getCandidateName()+")";
                                                    sta3.executeUpdate(InsertCandidateName);
                                            }
                                    }
                            }
                    }  catch  (InstantiationException  |  IllegalAccessException  |  ClassNotFoundException  |
SQLException e) {
                            result="adminActionsError";
                    }
            finally {
                    return result;
            }
        }
}
```

AdminLoginDAO.java

```
package com.javatpoint.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.AdminLoginBean;
import com.javatpoint.Beans.MySQL;

public class AdminLoginDAO {
        public String adminLoginVerification(AdminLoginBean adminLoginBean) {
                String result = "adminError";
                try {
                        System.out.println("in admin login dao");
                        ApplicationContext context =
                                new                ClassPathXmlApplicationContext(new                String[]
{"applicationContext.xml"});
                        System.out.println("wait");
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn        =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String Sql = "select * from admins";
                        ResultSet rs = sta.executeQuery(Sql);
```

```java
                while (rs.next()) {
                        System.out.println("username"+ rs.getString("username"));
                        System.out.println("password: "+ rs.getString("password"));
                        Boolean                    one                    =
rs.getString("username").equals(adminLoginBean.getUsername());
                        System.out.println("username: "+ one);
                        Boolean                    two                    =
rs.getString("password").equals(adminLoginBean.getPassword());
                        System.out.println("password:                    "+
rs.getString("password").equals(adminLoginBean.getPassword()));
                        if(rs.getString("username").equals(adminLoginBean.getUsername())    &&
rs.getString("password").equals(adminLoginBean.getPassword()))
                        {
                                result="adminActions";
                        }
                }

        } catch (InstantiationException | IllegalAccessException | ClassNotFoundException |
SQLException e) {
                e.printStackTrace();
        }
        return result;
    }
}
```

GetPartiesDAO.java

```java
package com.javatpoint.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.HashMap;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.MySQL;

public class GetPartiesDao {
 public HashMap<String, ArrayList<String>> getParties() {
        ArrayList<String> p = new ArrayList<String>();
        HashMap<String, ArrayList<String>> map = new HashMap<String, ArrayList<String>>();
        try{
                ApplicationContext context =
                        new ClassPathXmlApplicationContext(new String[] {"applicationContext.xml"});
                MySQL mySQL = (MySQL)context.getBean("mySQL");
                Class.forName(mySQL.getClassName()).newInstance();
                Connection conn = DriverManager.getConnection(mySQL.getUrl(), mySQL.getUserName(),
mySQL.getPassowrd());
```

```java
                Statement sta = conn.createStatement();
                ResultSet rs = sta.executeQuery("select * from parties");
                while(rs.next()) {
                        p.add(rs.getString("party_name"));
                }



                for (String partyName : p) {
                        System.out.println("party name: "+partyName);
                        ArrayList<String> candidateNames = null;
                        String getParty_id = "select * from parties where party_name='"+partyName+"'";
                        System.out.println("getParty_id"+ getParty_id);
                        Statement sta1 = conn.createStatement();
                        ResultSet rs1 = sta1.executeQuery(getParty_id);
                        while(rs1.next()){
                                int pID = rs1.getInt("party_id");
                                System.out.println("PartyID: "+pID);
                                Statement sta2 = conn.createStatement();
                                String getCandidates = "select candidate_name from candidates where
party_id="+rs1.getInt("party_id");
                                System.out.println("getCandidates:" +getCandidates);
                                ResultSet rs2 = sta2.executeQuery(getCandidates);
                                candidateNames = new ArrayList<>();
                                while(rs2.next()) {
                                        System.out.println("candidateName:
"+rs2.getString("candidate_name") );
                                        candidateNames.add(rs2.getString("candidate_name"));
                                }
                                rs2.close();
                                sta2.close();
                        }
                        map.put(partyName, candidateNames);
                }
        }catch(InstantiationException | IllegalAccessException | ClassNotFoundException | SQLException e){
                e.printStackTrace();
        }
        return map;
        }

 public ArrayList<String> getPartiesInfo(){
        ArrayList<String> partyNames = new ArrayList<>();
        try{
                ApplicationContext context =
                        new             ClassPathXmlApplicationContext(new          String[]
{"applicationContext.xml"});
                MySQL mySQL = (MySQL)context.getBean("mySQL");
                Class.forName(mySQL.getClassName()).newInstance();
                Connection       conn      =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                Statement sta = conn.createStatement();
                ResultSet rs = sta.executeQuery("select * from parties");
                while(rs.next()) {
                        partyNames.add(rs.getString("party_name"));
                }
```

```java
                }
                        catch(InstantiationException  |  IllegalAccessException  |  ClassNotFoundException  |
SQLException e){
                                e.printStackTrace();
                        }
        return partyNames;
        }

 public ArrayList<String> getCandidates(String partyName){
        int party_id = 0;
        ArrayList<String> candidates = new ArrayList<>();
        System.out.println("Partyname from getCandidates"+partyName);
        try{
                ApplicationContext context =
                        new ClassPathXmlApplicationContext(new String[] {"applicationContext.xml"});
                MySQL mySQL = (MySQL)context.getBean("mySQL");
                Class.forName(mySQL.getClassName()).newInstance();
                Connection  conn  =  DriverManager.getConnection(mySQL.getUrl(),  mySQL.getUserName(),
mySQL.getPassowrd());
                Statement sta = conn.createStatement();
                String getPartyId = "select * from parties where party_name='"+partyName+"'";
                ResultSet rs = sta.executeQuery(getPartyId);
                while(rs.next()){
                        party_id = rs.getInt("party_id");
                }
                System.out.println("Party_ID:"+ party_id);
                Statement sta1 = conn.createStatement();
                String getCandidates = "select * from candidates where party_id="+party_id;
                ResultSet rs1 = sta1.executeQuery(getCandidates);
                while(rs1.next()){
                        System.out.println(rs1.getString("candidate_name"));
                        candidates.add(rs1.getString("candidate_name"));
                }
        }catch(InstantiationException  |  IllegalAccessException  |  ClassNotFoundException  |  SQLException
e){e.printStackTrace();}

        System.out.println("Before return: "+candidates.toString());
        return candidates;

 }

public String getStatus() {
        String result = "stop";
        try{
                ApplicationContext context =
                        new ClassPathXmlApplicationContext(new String[] {"applicationContext.xml"});
                MySQL mySQL = (MySQL)context.getBean("mySQL");
                Class.forName(mySQL.getClassName()).newInstance();
                Connection  conn  =  DriverManager.getConnection(mySQL.getUrl(),  mySQL.getUserName(),
mySQL.getPassowrd());
                Statement sta = conn.createStatement();
                String query = "select * from admins";
                ResultSet rs= sta.executeQuery(query);
                while(rs.next()){
                        if(rs.getString("process").equals("start")) {
```

```
                                        result="start";
                        }
                }
        }catch(Exception e ){e.printStackTrace();}
        return result;
}
}
```

GetSecurityQuestionsDAO.java

```java
package com.javatpoint.DAO;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Base64;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.ForgotPasswordBean;
import com.javatpoint.Beans.MySQL;

public class GetSecurityQuestionDAO {
        String result="failure";
        public String getQuestion(String ValidationNumber) throws NoSuchAlgorithmException{
                try {
                        MessageDigest digest = MessageDigest.getInstance("SHA-256");
                        byte[] hash = digest.digest(ValidationNumber.getBytes(StandardCharsets.UTF_8));
                        String encoded = Base64.getEncoder().encodeToString(hash);

                        ApplicationContext context = new ClassPathXmlApplicationContext(new String[] {
"applicationContext.xml" });
                        MySQL mySQL = (MySQL) context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn        =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();

                        System.out.println(ValidationNumber);

                        String getQ = "select * from voters where validation_number = '"+encoded+"'";
                        ResultSet rs = sta.executeQuery(getQ);
                        while(rs.next()){
                                result = rs.getString("qid");
                        }
                }
```

```java
            catch(InstantiationException | IllegalAccessException | ClassNotFoundException | SQLException
e){
                result="failure";
            }
            return result;
        }
    public String overridePassword(ForgotPasswordBean forgotPasswordBean) throws
NoSuchAlgorithmException {
            result = "voterError";
            MessageDigest digest = MessageDigest.getInstance("SHA-256");
            byte[] hash =
digest.digest(forgotPasswordBean.getValidationNumber().getBytes(StandardCharsets.UTF_8));
            String encoded = Base64.getEncoder().encodeToString(hash);

            try {
                ApplicationContext context = new ClassPathXmlApplicationContext(new String[] {
"applicationContext.xml" });
                MySQL mySQL = (MySQL) context.getBean("mySQL");
                Class.forName(mySQL.getClassName()).newInstance();
                Connection conn = DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                Statement sta = conn.createStatement();

                String verifyPwd = "select * from voters where validation_number='"+encoded+"'";
                System.out.println("verifyPwd: "+verifyPwd);
                ResultSet rs = sta.executeQuery(verifyPwd);
                Statement sta1 = conn.createStatement();
                while(rs.next()){
                    System.out.println("rs.getString: "+rs.getString("validation_number"));
                    if(rs.getString("validation_number").equals(encoded)){
                        String sqlUpdate ="update voters set
password='"+forgotPasswordBean.getNewPassword()+"' where validation_number='"+encoded+"'";
                        System.out.println("sqlUpdate: "+sqlUpdate);
                        sta1.executeUpdate(sqlUpdate);
                        result="suxes";
                    }
                }
            }
            catch(InstantiationException | IllegalAccessException | ClassNotFoundException | SQLException
e){
                e.printStackTrace();
                result="voterError";
            }
            return result;
        }
}
```

LockResultsDAO.java

package com.javatpoint.DAO;

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
```

```java
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.MySQL;

public class LockResultsDAO {

        @SuppressWarnings("finally")
        public String lockResults() {
                String result="";
                try {
                        System.out.println("In Start Elections DAO");
                        ApplicationContext context =
                                new               ClassPathXmlApplicationContext(new               String[]
{"applicationContext.xml"});
                        System.out.println("wait");
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn       =          DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String Sql = "update admins set results='stop' where username='admin'";
                        sta.executeUpdate(Sql);
                        result = "adminActions";
                } catch (Exception e) {
                        result = "adminActionsError";
                }
                finally {
                        return result;
                }
        }

}
```

LoginVerificationDAO.java

```java
package com.javatpoint.DAO;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Base64;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.LoginBean;
import com.javatpoint.Beans.MySQL;
import com.javatpoint.Beans.User;
```

```java
public class LoginVerificationDAO {

        /*@Autowired
        MySQL mySQL;*/

        public String verifyLogin(LoginBean loginDetails) {
                String result="failure";
                try {
                        MessageDigest digest = MessageDigest.getInstance("SHA-256");
                        byte[]                                    hash                                    =
digest.digest(loginDetails.getValidationNumber().getBytes(StandardCharsets.UTF_8));
                        String encoded = Base64.getEncoder().encodeToString(hash);

                        ApplicationContext context =
                                new           ClassPathXmlApplicationContext(new              String[]
{"applicationContext.xml"});
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection       conn      =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String Sql = "select * from voters";
                        ResultSet rs = sta.executeQuery(Sql);
                        System.out.println("before while!");
                        while(rs.next()){
                                if(rs.getString("validation_number").equals(encoded)                  &&
rs.getString("password").equals(loginDetails.getPassword())){
                                        System.out.println("inside while:"+rs.getString("first_name"));
                                        result = rs.getString("validation_number");
                                }
                        }
                } catch (Exception e) {
                        return result;
                }
                        return result;
                }

        public User getUserDetails(String validationNumber){
                User userDetails = new User();
                try{

                        int voter_id = 0;
                        int cla_id=0;
                        ApplicationContext context =
                                new           ClassPathXmlApplicationContext(new              String[]
{"applicationContext.xml"});
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection       conn      =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();

                        String sql = "select * from voters where validation_number = '"+validationNumber+"'";
                        System.out.println("sql"+ sql);
```

```java
                ResultSet rs = sta.executeQuery(sql);
                while(rs.next()) {
                        voter_id = rs.getInt("voter_id");
                        userDetails.setFirst_name(rs.getString("first_name"));
                        userDetails.setMiddle_name(rs.getString("middle_name"));
                        userDetails.setLast_name(rs.getString("last_name"));
                        userDetails.setSsn(rs.getString("ssn"));
                        userDetails.setDob(rs.getString("dob"));
                }

                String sql1 = "select * from cla where voter_id ="+voter_id;
                System.out.println("sql1"+ sql1);
                Statement sta1 = conn.createStatement();
                ResultSet rs1 = sta1.executeQuery(sql1);
                while(rs1.next()) {
                        userDetails.setVote_status(rs1.getString("vote_status"));
                        if(rs1.getString("vote_status").equals("true")){
                                Statement sta2 = conn.createStatement();
                                String sql2 = "select * from ctf where cla_id="+rs1.getInt("cla_id");
                                System.out.println("sql2"+ sql2);
                                ResultSet rs2 = sta2.executeQuery(sql2);
                                while(rs2.next()) {
                                        userDetails.setVotedparty(rs2.getString("party_name"));

        userDetails.setCandidateName(rs2.getString("candidate_name"));
                                }
                        }
                }
        }catch(Exception e){
                e.printStackTrace();
                return userDetails;
        }
        System.out.println("Candidate Name: "+userDetails.getCandidateName());
        return userDetails;
    }
}
```

RegisterDAO.java

```java
package com.javatpoint.DAO;

import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Base64;
import java.util.Random;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
```

```java
import com.javatpoint.Beans.MySQL;
import com.javatpoint.Beans.RegisterBean;

public class RegisterDAO {
        int number;
        String str;
        public String registerVoter(RegisterBean registerBean) throws NoSuchAlgorithmException {
                try {
                        ApplicationContext context = new ClassPathXmlApplicationContext(new String[] {
"applicationContext.xml" });
                        MySQL mySQL = (MySQL) context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection conn = DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        Statement sta1 = conn.createStatement();
                        System.out.println("From RegisterDAO: " + registerBean.toString());

                        String sql1 ="select * from voters where ssn ='"+registerBean.getSsn()+"'";
                        ResultSet rSet = sta1.executeQuery(sql1);
                        if(rSet.next()){
                                str = "You are already registered.";
                        }
                        else {
                                //Verify the details entered by the user
                                //Generate random number
                                Random r = new Random();
                                number = r.nextInt(999999999);

                                String numberStr = Integer.toString(number);
                                MessageDigest digest = MessageDigest.getInstance("SHA-256");
                                byte[] hash = digest.digest(numberStr.getBytes(StandardCharsets.UTF_8));
                                String encoded = Base64.getEncoder().encodeToString(hash);
                                System.out.println("encoded validation number: "+encoded);

                                //If the details entered by user are authentic then save the entered details in the
voters table
                                String Sql = "INSERT INTO voters ( voter_id,
validation_number,first_name,middle_name,last_name,ssn,dob,qid,answer,password ) VALUES (
NULL,'"+encoded+"','"+
registerBean.getFirstname()+"','"+registerBean.getMiddlename()+"','"+registerBean.getLastname()+"','"+registerBea
n.getSsn()+"','"+registerBean.getDob()+"','"+registerBean.getQuestion()+"','"+registerBean.getAnswer()+"','"+regist
erBean.getPassword()+"')";
                                sta.executeUpdate(Sql);
                                str = Integer.toString(number);


                                //insert voter_id and vote_status=false in cla table
                                int voter_id = 0;
                                String vote_status = "false";
                                String getVoterId = "select * from voters where validation_number
='"+encoded+"'";

                                Statement sta2 = conn.createStatement();
                                ResultSet rs1 = sta2.executeQuery(getVoterId);
```

```java
                                  while(rs1.next()) {
                                          voter_id = rs1.getInt("voter_id");
                                  }
                                  String claInsertQuery    =    "Insert    into    cla    (cla_id,    voter_id)
values(NULL,"+voter_id+")";
                                  sta2.executeUpdate(claInsertQuery);

                                  //get cla_id, party_name,vote_status in ctf
                                  int cla_id = 0;
                                  String vote_status1;
                                  String getclaid = "select * from cla where voter_id ="+voter_id;
                                  Statement sta3 = conn.createStatement();
                                  ResultSet rs2 = sta3.executeQuery(getclaid);
                                  while(rs2.next()) {
                                          cla_id = rs2.getInt("cla_id");
                                  }

                                  Statement sta4 = conn.createStatement();
                                  String insertIntoCtf= "insert into ctf (ctf_id, cla_id) values(NULL,"+cla_id+")";
                                  sta4.executeUpdate(insertIntoCtf);
                          }
                  } catch (InstantiationException  |  IllegalAccessException  |  ClassNotFoundException  |
SQLException e) {
                          e.printStackTrace();
                  }

                  return str;
          }
}
```

ShowResultsDAO.java

```java
package com.javatpoint.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.HashMap;
import java.util.Map.Entry;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.MySQL;
import com.javatpoint.Beans.Results;

public class ShowResultsDAO {

        @SuppressWarnings("finally")
        public String stopElectionProcess() {
                String result="";
                try {
```

```java
                    System.out.println("In Start Elections DAO");
                    ApplicationContext context =
                                new        ClassPathXmlApplicationContext(new          String[]
{"applicationContext.xml"});
                    System.out.println("wait");
                    MySQL mySQL = (MySQL)context.getBean("mySQL");
                    Class.forName(mySQL.getClassName()).newInstance();
                    Connection      conn      =       DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                    Statement sta = conn.createStatement();
                    String Sql = "update admins set results='start' where username='admin'";
                    sta.executeUpdate(Sql);
                    result = "adminActions";
            } catch (Exception e) {
                    result = "adminActionsError";
            }
            finally {
                    return result;
            }
        }

        public HashMap<Integer, Results> getResults() {
                HashMap<Integer,Results> results = new HashMap<Integer,Results>();
                try{
                    ApplicationContext context =
                                new        ClassPathXmlApplicationContext(new          String[]
{"applicationContext.xml"});
                    System.out.println("wait");
                    MySQL mySQL = (MySQL)context.getBean("mySQL");
                    Class.forName(mySQL.getClassName()).newInstance();
                    Connection      conn      =       DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                    Statement sta = conn.createStatement();

                    String query = "select party_name,candidate_name,count(candidate_name) as totalVotes
from ctf where party_name in(select distinct party_name from ctf) group by candidate_name";
                    ResultSet res = sta.executeQuery(query);


                    int i=1;
                    while(res.next()) {
                            Results resBean = new Results();
                            resBean.setParty_name(res.getString("party_name"));
                            resBean.setCandidate_name(res.getString("candidate_name"));
                            resBean.setTotalVotes(res.getInt("totalVotes"));
                            results.put(i, resBean);
                            i++;
                    }

                    for (Entry<Integer, Results> entry : results.entrySet()) {
                       System.out.println(entry.getKey()+" : "+entry.getValue());
                    }

                }catch(Exception e){}
                return results;
```

```java
        }

        public HashMap<String, Integer> getTotalVotesEachParty() {
                HashMap<String,Integer> TotalVotesEachParty = new HashMap<>();
                try{
                        ApplicationContext context =
                                new               ClassPathXmlApplicationContext(new                String[]
{"applicationContext.xml"});
                        System.out.println("wait");
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn        =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String sql = "select party_name,count(*) as TotalVotes from ctf group by party_name;";
                        ResultSet rs = sta.executeQuery(sql);

                        while(rs.next()) {
                                TotalVotesEachParty.put(rs.getString("party_name"), rs.getInt("TotalVotes"));
                        }
                        for (Entry<String, Integer> entry : TotalVotesEachParty.entrySet()) {
                           System.out.println(entry.getKey()+" : "+entry.getValue());
                        }
                }catch(Exception e){
                        e.printStackTrace();
                }
                return TotalVotesEachParty;
        }

        public String getStatus() {
                String result="stop";
                try{
                        ApplicationContext context =
                                new               ClassPathXmlApplicationContext(new                String[]
{"applicationContext.xml"});
                        System.out.println("wait");
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn        =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String sql = "select * from admins";
                        ResultSet rs = sta.executeQuery(sql);
                        while(rs.next()){
                                if(rs.getString("results").equals("start")){
                                        result="start";
                                }
                        }
                }catch(Exception e){e.printStackTrace();}
                return result;
        }
}
```

StartElectionsDAO.java

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.MySQL;

public class StartElectionsDAO {
        String result="adminActionsError";
        @SuppressWarnings("finally")
        public String startElectionProcess(){
                try {
                        System.out.println("In Start Elections DAO");
                        ApplicationContext context =
                                new              ClassPathXmlApplicationContext(new              String[]
{"applicationContext.xml"});
                        System.out.println("wait");
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection       conn      =       DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String Sql = "update admins set process='start' where username='admin'";
                        sta.executeUpdate(Sql);
                        result = "adminActions";
                } catch (Exception e) {
                        result = "adminActionsError";
                }
                finally {
                        return result;
                }
        }
}
```

StopElectionsDAO.java

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.MySQL;
```

```java
public class StopElectionsDAO {

        @SuppressWarnings("finally")
        public String stopElectionProcess() {
                String result = "";
                try {
                        System.out.println("In Start Elections DAO");
                        ApplicationContext context =
                                new                ClassPathXmlApplicationContext(new                String[]
{"applicationContext.xml"});
                        System.out.println("wait");
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn        =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();
                        String Sql = "update admins set process='stop' where username='admin'";
                        sta.executeUpdate(Sql);
                        result = "adminActions";
                } catch (Exception e) {
                        result = "adminActionsError";
                }
                finally {
                        return result;
                }
        }

}
```

VoterPerformedDAO.java

```java
package com.javatpoint.DAO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.javatpoint.Beans.MySQL;
import com.javatpoint.Beans.VotedParty;

public class votePerformedDAO {
        String result = "success";
        public String performVote(VotedParty votedParty) {
                try{

                        ApplicationContext context =
                                new                ClassPathXmlApplicationContext(new                String[]
{"applicationContext.xml"});
```

```java
                        MySQL mySQL = (MySQL)context.getBean("mySQL");
                        Class.forName(mySQL.getClassName()).newInstance();
                        Connection        conn        =        DriverManager.getConnection(mySQL.getUrl(),
mySQL.getUserName(), mySQL.getPassowrd());
                        Statement sta = conn.createStatement();

                        String[] parts = votedParty.getPartyAndCandidateName().split("-");
                        String PartyName = parts[1];
                        String CandidateName = parts[0];

                        System.out.println("PartyName:"+PartyName+"CandidateName:"+CandidateName);
                        System.out.println("wait");


                        //get voter id
                        int voter_id=0;
                        String validationNumber = votedParty.getValidationNumber();
                        String    getVoterId    =    "select    *    from    voters    where    validation_number    =
'"+validationNumber+"'";
                        ResultSet rs = sta.executeQuery(getVoterId);
                        while(rs.next()) {
                                voter_id = rs.getInt("voter_id");
                        }

                        //Update Cla Table
                        String updateCla = "Update cla set vote_status='true' where voter_id = "+voter_id;
                        Statement sta1 = conn.createStatement();
                        sta1.executeUpdate(updateCla);

                        //get cla_is for voted memeber
                        int cla_id =0;
                        String claid = "select * from cla where voter_id = "+voter_id;
                        Statement sta2 = conn.createStatement();
                        ResultSet rs2 = sta2.executeQuery(claid);
                        while(rs2.next()) {
                                cla_id = rs2.getInt("cla_id");

                        }
                        //Update Ctf table
                        String stat = "true";
                        String              updateCtf            =            "Update            ctf            set
party_name='"+PartyName+"',candidate_name='"+CandidateName+"',          vote_status='"+stat+"'          where
cla_id="+cla_id;
                        System.out.println("QUERY: "+updateCtf);
                        Statement sta3 = conn.createStatement();
                        sta3.executeUpdate(updateCtf);


                }catch(InstantiationException    |    IllegalAccessException    |    ClassNotFoundException    |
SQLException e){e.printStackTrace();}
                return "success";
        }
}
```

ApplicationContext.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>
        <bean id="mySQL" class="com.javatpoint.Beans.MySQL">
                        <property name="Url" value="jdbc:mysql://localhost:3306/veb" />
                        <property name="userName" value="root" />
                        <property name="passowrd" value="manohara" />
                        <property name="className" value="com.mysql.jdbc.Driver" />
        </bean>
</beans>
```

JSP Files:

admin.jsp

```jsp
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Admin</title>
<style type="text/css" >
.form-signin {
        width: 280px;
        left:100px
        display: inline;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
#registerForm_header {
        font-size: 2em;
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
        clear: both;
    display: inline-block;
    overflow: hidden;
    white-space: nowrap;
}
#adminLoginBtn{
        background-color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
</style>
</head>
<body>
</br></br></br></br></br></br></br></br></br>
```

```html
<div id="reg" class="container" >
 <center><h2 id="registerForm_header">Login as Admin</h2></center></br>
                <center><form class="form-signin" id=rForm action = "adminLogin" method="post">
                        <input type="text" class="form-control" name="username"
                                placeholder="User Name" required="" autofocus="" id="username"/>
                        <input type="password" class="form-control" name="password"
                                placeholder="Password" required="" autofocus="" id="password"/>
                                </br>
                        <button      id="adminLoginBtn"      class="btn      btn-lg      btn-primary      btn-block"
type="submit">Login</button>
                </form></center>
                </div>
                <!-- </br></br></br></br></br></br></br></br></br></br></br></br></br></br> -->
</body>
</html>
```

AdminActionsError.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
        <style type="text/css">
                #btn_adminError {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                        background-color:black;
                }
                #alert {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                }
        </style>
</head>
<body>
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <div id="alert" class="alert alert-danger" role="alert">
                <center>Error performing requested operation.</center>
        </div>
        <div id="btn_adminError" class="container">
                <center><a href="admin" class="btn btn-info" role="button" aria-pressed="true">Admin
Actions</a>
        </div>

</body></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</html>
```

adminError.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"></link>
        <style type="text/css">
                #btn_adminError {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                        background-color:black;
                }
                #alert {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                }
        </style>
</head>
<body>
</br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <div id="alert" class="alert alert-danger" role="alert">
                <center>Login Failed. Username/Password incorrect.</center>
        </div>
        <div class="container">
                <center><a id="btn_adminError" href="admin" class="btn btn-info" role="button" aria-
pressed="true">Login</a>
                </center>
        </div>

</body></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</html>
```

analysis.jsp

```
c<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Spring Tiles Contact Form</title>
<script type="text/javascript">
window.onload = function () {
        var chart = new CanvasJS.Chart("chartContainer",
        {
                title:{
                        text: "Desktop Search Engine Market Share, Dec-2012"
                },
```

```
            animationEnabled: true,
                    legend:{
                            verticalAlign: "center",
                            horizontalAlign: "left",
                            fontSize: 20,
                            fontFamily: "Helvetica"
                    },
                    theme: "theme2",
                    data: [
                    {
                            type: "pie",
                            indexLabelFontFamily: "Garamond",
                            indexLabelFontSize: 20,
                            indexLabel: "{label} {y}%",
                            startAngle:-20,
                            showInLegend: true,
                            toolTipContent:"{legendText} {y}%",
                            dataPoints: [
                                    {  y: 83.24, legendText:"Hillary Clinton", label: "Hillary Clinton"  },
                                    {  y: 8.16, legendText:"Donald Trump", label: "Donald Trump"  },
                            ]
                    }
                    ]
            });
            chart.render();
}
</script>
<script type="text/javascript" src="https://canvasjs.com/assets/script/canvasjs.min.js"></script>
</head>
<style type="text/css" >
 .bar{
                background-color: #f2f2f1;
                width:50%;
                height: 100px;
                margin-bottom:0px;
        }
        .bar-inner{
                background-color: #2ecc71;
                width:90%;
                height:100%;
        }

</style>
<body>
        <div class="container">
                <div id="chartContainer" style="height: 300px; width: 100%;"></div>
        </div>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
        </br></br></br></br></br></br></br></br></br>
        </body>
</body>
</html>
```

Contact.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Spring Tiles Contact Form</title>
<style type="text/css">
        body {
                background-image: url("\WebContent\header.png");
        }
</style>
</head>
<body>
        <div class="container">
        <h2>In contact.jsp</h2>
        </div>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
</html>
```

electionProcessError.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
        <style type="text/css">
                #btn_adminError {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                        background-color:black;
                }
                #alert {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                }
        </style>
</head>
<body>
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <div id="alert" class="alert alert-danger" role="alert">
                <center>Election process has not yet started.</center>
        </div>
        <div class="container">
                <center><center><a id="btn_adminError" href="hello" class="btn btn-info" role="button" aria-
pressed="true">Home</a>
```

```
            </div>

</body></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
            <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
            <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</html>
```

Footer.jsp

```
<head>
<link rel="stylesheet"
            href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<style type="text/css">
.login {
            width: 300px;
            height: 100px;
            position: absolute;
            top: 0;
            bottom: 0;
            left: 0;
            right: 0;
            margin: auto;
            left:900px;
            top: -300px;

            display: inline;
}
.leftDiv {
            width: 300px;
            height: 100px;
            position: absolute;
            top: 0;
            bottom: 0;
            left: 0;
            right: 0;
            margin: auto;
            left:-300px;
            top: 300px;
}
#login{
            background-color:black;
            border-color:black;
}
#reglink {
            text-align: center;
            font-weight: bold;
            color: teal;
}
a{
            color: black;
}

h2 {
            font-size: 3em;
            color:black;
```

```css
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
        clear: both;
    display: inline-block;
    overflow: hidden;
    white-space: nowrap;
}
#reglink{
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-size: 1.5em;
}
h4 {
        text-align: center;
        font-weight: 600;
}


</style>
</head>
<body>
    <div>
     <div class="text-center center-block">
        <a  href="https://www.facebook.com/manoharadepu45"><i id="social-fb"  class="fa  fa-facebook-square fa-3x social"></i></a>
            <a  href="https://twitter.com/adepumano"><i  id="social-tw"  class="fa  fa-twitter-square  fa-3x social"></i></a>
            <a  href="https://plus.google.com/116467010772685963266"><i  id="social-gp"  class="fa  fa-google-plus-square fa-3x social"></i></a>
            <a  href="mailto:mano.adepu@gmail.com"><i id="social-em"  class="fa  fa-envelope-square  fa-3x social"></i></a>
            </div>
        <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
```

forgotPassword.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<style type="text/css">
#securityQuestion {
        width: 300px;
        right: 500px;
        font-family:"Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-size: 2em;
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
```

```css
        clear: both;
    display: inline-block;
    overflow: hidden;
    white-space: nowrap;
}

input {
        display: inline;
}
#modalForm{
}

#formHeader {
        font-weight: bold;
}
.btn-group-lg>.btn, .btn-lg {
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
#modal-submit{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        background-color:black;
}

.modal-title {
        font-size: 2em;
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
.modal-header{
        background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
}

</style>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>

</head>
<body>
</br></br></br></br></br></br></br></br></br>
<center>
        <div class="forgotPassword">
                <form class="form-forgotPassword" id="securityQuestion" >
                        <h3 id="formHeader">Enter Validation Number</h3></br>
                        <input      type="text"        class="form-control"        name="validationNumber"
id="validationNumber"
                                        placeholder="validationNumber" required="" autofocus="" /></br>
                        <button  class="btn  btn-lg  btn-primary  btn-block"  id="login"  type="submit">Answer
Security Question</button>
                </form>
                </div></center>

                <div class="modal fade" id="myModal" role="dialog">
                <div class="modal-dialog">
                        <!-- Modal content-->
                        <div class="modal-content">
```

```html
                                    <div class="modal-header">
                                            <h4 class="modal-title" id = "actionModalHeading"></h4>
                                    </div>
                                    <div id="modalBody" class="modal-body">
                                            <form              id="modalForm"              class="form-signin"
action="updatePassword"></form>
                                    </div>
                                    <div class="modal-footer">
                                    </div>
                            </div>
                    </div>
                    <p></p>
            </div>
                    <div class="modal fade" id="myModalAfterUpdate" role="dialog">
                    <div class="modal-dialog">
                            <!-- Modal content-->
                            <div class="modal-content">
                                    <div class="modal-header">
                                            <h4 class="modal-title" id = "actionModalHeading"></h4>
                                    </div>
                                    <div id="modalBody" class="modal-body">
                                    <h3 id="result"></h3>
                                    </div>
                                    <div class="modal-footer">
                                    </div>
                            </div>
                    </div>
                    <p></p>
            </div>

                    <script type="text/javascript">
                    $(document).ready(function() {
                            $('#securityQuestion').submit(function(e) {
                                    e.preventDefault();
                                    console.log('in javascript');
                                    var formData = $('#securityQuestion').serialize();
                                    console.log(formData);
                                    $.ajax({
                                            data : formData,
                                            type : 'GET',
                                            url : 'getSecurityQuestion',
                                            success : function(result) {
                                                    //if result = failure, show error message modal
                                                    if(result=='failure'){
                                                            console.log("in failure f u");
                                                            $(".modal-header").append("<h2>Incorrect
Validation Number<h2>");
                                                            $("#modalBody").append("<h4>Please enter correct
validation number.<h4>");
                                                            $(".modal-footer").append("<button     type='button'
class='btn btn-default' data-dismiss='modal'>Close</button>");
                                                            $('#myModal').modal('show');
                                                    }
                                                    else {
                                                            console.log("in else");
```

```
                                                  $("#actionModalHeading").append("<h4>Answer the
below security Question</h4>");

                                                  var validation = $("#validationNumber").val();
                                                  var            securityQuestion            =
$("#validationNumber").val();

                                                  $("#modalForm").append("Your            Validation
Number:<input        type='text'        readonly='readonly'        class='form-control'        id='validationNumber'
name='validationNumber' value = "+validation+"></input></br>");

                                                  $("#modalForm").append("Security Question: <input
type='text'  readonly='readonly'  class='form-control'  id='securityQuestion'  name='securityQuestion'  value  =
"+result+"</input></br>");

                                                  $("#modalForm").append("Answer:            <input
type='text' class='form-control' id='answer' name='answer' placeholder='Enter Answer'></input></br>");
                                                  $("#modalForm").append("New  Password:  <input
type='text'        class='form-control'        id='newPassword'        name='newPassword'        placeholder='Enter     New
Password'></input></br>");

                                                  $("#modalForm").append('</br><input
type="submit" id="modal-submit" class="btn btn-lg btn-primary btn-block" type="submit" align="right""/>');

                                                  $('#myModal').modal('show');
                                                  }
                                      //else show security question modal
                              }
                        });
                });

        });
        </script>


</body></br></br></br></br></br></br></br></br></br>
</html>
```

header.jsp

```html
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<style type="text/css">
#pageHeader{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
        font-size: 3em;
        text-align: center;
        font-weight: 700;
        word-spacing: 30px;
        letter-spacing: 10px;
        text-align: center;
        display: inline;
}
#logout1{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
#mainheader {
```

```
padding-right:200px;
background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
font-weight: bold;
}
/* #nav {
          text-decoration: none;
                    font-size: 24px;
          font-style: normal;
          font-weight: normal;
          text-transform: normal;
          letter-spacing: normal;
          line-height: 1.3em;
          color: #596365;
} */

#logout:hover{
background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
}

#nav {
align:center;
          font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
          text-decoration: none;
          padding-left:100px;
          padding-right:120px;
          font-size: 19px;
          font-style: normal;
          font-weight: bold;
          text-transform: normal;
          letter-spacing: normal;
          line-height: 0.8    em;
}
#logout {
          font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
          font-size: 15px;
          font-style: normal;
          font-weight: bold;
          text-transform: normal;
          letter-spacing: normal;
          line-height: 0.8    em;
          padding-left:100px;
          padding-right:120px;
          font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
}
</style>
</head>
<body >
          <div id="mainheader">
                    </br></br>
                    <center><h2 id="pageHeader">VIRTUAL ELECTION BOOTH</h2></center>
                    </br></br>
          </div>
          <nav class="navbar navbar-inverse">
  <div class="container-fluid">
```

```html
    <ul class="nav navbar-nav">
      <li><a id="nav" href="hello.html">Home</a></li>
                <!-- <li><a id="nav" href="contact.html">Profile</a></li> -->
                <li><a id="nav" href="parties.html">Parties Information</a></li>
                <li><a id="nav" href="vote.html">Vote</a></li>
                <li><a id="nav" href="results.html">Results</a></li>
                <li><a id="nav" href="analysis.html">Analysis</a></li>
                <li><a id = "logout" href="logoutVoter">
        <span id ="logout1" class="glyphicon glyphicon-log-out">Logout</span>
      </a></li>
    </ul>
  </div>

</nav>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
</html>
```

hello.jsp

```html
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Voter Details</title>
<style type="text/css">
        h2 {
        font-size: 3em;
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
        clear: both;
    display: inline-block;
    overflow: hidden;
    white-space: nowrap;
}
table{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        color:white
}
tbody {
        background:black;
        font-size: 1em;
}
thead{
        text-align:center
        font-size: 1em;
        font-weight: bold;
        background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
}

</style>
</head>
```

```
<body>
        <div class="container">
 <center><h2>VOTER DETAILS</h1></center></b></br></br>
 <table class="table table-bordered ">
  <thead id="tableHeader">
   <tr>
     <th>Firstname</th>
     <th>Middlename</th>
     <th>LastName</th>
     <th>SSN</th>
     <th>DOB</th>
     <th>VoteStatus</th>
     <th>VotedParty</th>
     <th>CandidateName</th>
   </tr>
  </thead>
  <tbody>
   <tr>
     <td>${user1.first_name}</td>
     <td>${user1.middle_name}</td>
     <td>${user1.last_name}</td>
     <td>${user1.ssn}</td>
     <td>${user1.dob}</td>
     <td>${user1.vote_status}</td>
     <td>${user1.votedparty}</td>
     <td>${user1.candidateName}</td>
   </tr>

  </tbody>
 </table>
 <center><a id="nav" href="hello.html">View Details</a></center>
</div>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
</br></br></br></br></br></br></br></br></br></br>
</html>
```

layout.jsp

```
<%@ taglib uri="http://tiles.apache.org/tags-tiles" prefix="tiles"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title><tiles:insertAttribute name="title" ignore="true" /></title>
</head>
<body>
        <div>
                <tiles:insertAttribute name="header" />
        </div>
        <%-- <div style="float: left; padding: 10px; width: 15%;">
                <tiles:insertAttribute name="menu" />
```

```
</div> --%>
<div
        style="float: left; padding: 10px; width: 100%; border-left: 1px solid pink;">
        <tiles:insertAttribute name="body" />
</div>
<div style="clear: both">
        <tiles:insertAttribute name="footer" />
</div>

</body>
</html>
```

menu.jsp

```
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
</head>

<body>
<ul class="nav nav-pills nav-stacked">
                <li role="presentation" ><a href="hello.html">Home</a></li>
                <li role="presentation"><a href="contact.html">Profile</a></li>
                <li role="presentation"><a href="parties.html">Parties Information</a></li>
                <li role="presentation"><a href="vote.html">Vote</a></li>
                <li role="presentation"><a href="results.html">Results</a></li>
                <li role="presentation"><a href="analysis.html">Analysis</a></li>
</ul>
</body>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

parties.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Spring Tiles Contact Form</title>
<style type="text/css">
        body {
                background-image: url("\WebContent\header.png");
        }
        a:hover {
        font-weight:bold;
                color:black;
        }
        a:active {
        font-weight:bold;
                color:black;
        }
```

```css
        a:focused {
        font-weight:bold;
                color:black;
        }
        a:visited {
        font-weight:bold;
                color:black;
        }
        button{
        background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
        }

        table{
        border-radius: 45px;
        font-size: 3em;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        color:black;
}

        tbody {
                background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
                font-size: 1em;
        }
        .first{
         padding: 10px;
   height:150px;
        }

        #partyNames td:hover {
   color:black;
}

</style>
</head>
<body>
        <div class="container">
  <div class="row text-center">
   <div class="col-sm-9 first">
        <h2 class="header">Parties</h2></div>
  <table class="table table-bordered" id="partyNames">
   <tbody>
    <tr>
        <c:forEach items="${list}" var="pname">
        <td><button class = "PartyName" id="${pname}" value="${pname}" type="button" class="btn btn-primary btn-lg">${pname} </button></td>
                </c:forEach>
    </tr>
   </tbody>
  </table>
   </div>
  </div>
  <div class="row">
   <div class="col-sm-6">

   </div>
```

```html
    <div class="container">
            <table class="table table-bordered" id="candidateName">
            <!-- Generate candidates of the party selected -->
            </table>
    </div>
  </div>
</div>

<script>
$("button").click(function() {
        var partyName = $(this).val();
        $.ajax({
                type : 'GET',
                accept : "application/json",
                url : 'party/'+partyName,
                success : function(result) {
                        console.log("Candidate Names: "+result);
                        $("#candidateName").html("");
                        $("#candidateName").append("<th>Candidate Names</th>");
                        var parseData = JSON.parse(result);
                        for(i=0;i<parseData.length;i++){
                                showTable(parseData[i]);
                        }
                }
        });
});

function showTable(item){
        console.log(item);

        $("#candidateName").append("<tr><td>"+item+"</td></tr>")

}
</script>

        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
<!--
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br
></br></br></br></br></br></br></br></br></br></br></br></br></br>
 -->
 </html>
```

register.jsp

```jsp
<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Spring Tiles Contact Form</title>
<style type="text/css">
#modal-submit {
```

```css
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
                AppleGothic, sans-serif;
}

.modal-footer {
        background-color: black;
}

.modal-title {
        font-size: 2em;
        color: black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
                AppleGothic, sans-serif;
}

.modal-header {
        background: linear-gradient(to bottom, #b23435 0%, #891516 100%);
}

#registerForm {
        width: 300px;
        right: 500px;
        background-color: aqua;
}

#mheader {
        background-color: linear-gradient(to bottom, #b23435 0%, #891516 100%);
}

#registerForm_header {
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
                AppleGothic, sans-serif;
        font-size: 2em;
        color: black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
                AppleGothic, sans-serif;
        font-weight: bold;
        clear: both;
        display: inline-block;
        overflow: hidden;
        white-space: nowrap;
}

#testmodalpara {
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
                AppleGothic, sans-serif;
        font-weight: bold;
}

#registerBtn {
        background-color: black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
                AppleGothic, sans-serif;
}
```

```
body {
        font-family: font-family : "Avant Garde", Avantgarde, "Century Gothic",
                CenturyGothic, AppleGothic, sans-serif;
}
</style>
</head>
<body>
        </br>
        </br>
        </br>
        </br>
        <div id="reg" class="container">
                <center>
                        <h3 id="registerForm_header">
                                REGISTER
                                </h2>
                </center>
                <center>
                        <form class="form-signin" id=rForm>
                                <input type="text" class="form-control" name="firstname"
                                        placeholder="First Name" required="" autofocus="" id="firstName" />
                                <input type="text" class="form-control" name="middlename"
                                        placeholder="Middle Name" required="" autofocus="" /> <input
                                        type="text" class="form-control" name="lastname"
                                        placeholder="Last Name" required="" autofocus="" /> <input
                                        type="text" class="form-control" name="dob" placeholder="DOB"
                                        required="" autofocus="" /> <input type="text"
                                        class="form-control" name="ssn"
                                        placeholder="Social Security Number" required="" autofocus="" />
<input
                                        type="text" class="form-control" name="question"
                                        placeholder="Enter a Security Question" required="" autofocus="" />
                                <input type="text" class="form-control" name="answer"
                                        placeholder="Enter Security Question's Answer" required=""
                                        autofocus="" /> <input type="password" class="form-control"
                                        name="password" placeholder="enter password" required=""
                                        autofocus="" />
                                <button class="btn btn-lg btn-primary btn-block" id="registerBtn"
                                        type="submit">Register</button>
                        </form>
                </center>
                <div class="modal fade" id="myModal" role="dialog">
                        <div class="modal-dialog">

                                <!-- Modal content-->
                                <div class="modal-content">
                                        <div id="mheader" class="modal-header">
                                                <button type="button" class="close" data-
dismiss="modal">&times;</button>
                                                <h4 class="modal-title">Registration Successful</h4>
                                        </div>
                                        <div id="modalBody" class="modal-body">
                                                <center>
                                                        <h4>Please save validation number</h4>
```

```
                                                <h3
id=testmodalpara>${model.registerBean.firstname}</h3>
                                        </center>
                                </div>
                                <div class="modal-footer">
                                        <button    type="button"    class="btn    btn-default"    data-
dismiss="modal">Close</button>

                                        <a type="button" id="login" href="index" class="btn btn-info"
                                                role="button" aria-pressed="true">Login</a>
                                </div>
                        </div>
                </div>
                <p></p>
        </div>

        </div>
        <script
                src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script
                src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

        <script type="text/javascript">
                $(document).ready(function() {
                        $('#rForm').submit(function(e) {
                                e.preventDefault();
                                console.log('in javascript');
                                var formData = $('#rForm').serialize();
                                console.log(formData);
                                $.ajax({
                                        data : formData,
                                        type : 'GET',
                                        url : 'registerForm',
                                        success : function(result) {
                                                console.log(result);
                                                $("#testmodalpara").html("<p>" + result + "</p>")
                                                $('#myModal').modal('show');
                                        }
                                });
                        });
                });
        </script>
        </br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
</body>
</html>
```

results.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Spring Tiles Contact Form</title>
<style type="text/css">
```

```css
table{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        color:white
}
tbody {
        background:black;
        font-size: 1em;

}
td{
        background: black;

}
td:hover{
        background: linear-gradient(to bottom, #b23435 0%,#891516 100%);

}
thead{
        text-align:center
        font-size: 1em;
        font-weight: bold;
        background:linear-gradient(to bottom, #b23435 0%,#891516 100%);

</style>
```
```jsp
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
</head>
<body>
        <div class="container">
        <center><h2>Vote Count for each candidate</h2></center>
        <table class="table table-bordered ">
                <thead id="tableHeader">
          <tr>
           <th>Party</th>
           <th>Candidate</th>
           <th>Total Votes</th>
          </tr>
          <c:forEach var='item' items='${res}'>
                <tr>
                        <td>${item.value.getParty_name()}</td>
                        <td>${item.value.getCandidate_name()}</td>
                        <td>${item.value.getTotalVotes()}</td>
                </tr>
          </c:forEach>
          </thead>
        </table>
        <center><h2 class="text-center">Vote Count for Each Party</h2></center>
        <table class="table table-bordered">
                <thead id="tableHeader">
                        <tr>
                                <th>Party</th>
                    <th>Total Votes</th>
                        </tr>
                </thead>
                 <c:forEach var='item' items='${TotalVotesForEachParty}'>
                <tr>
                        <td><c:out value="${item.key}"/></td>
                        <td><c:out value="${item.value}"/></td>
```

```
            </tr>
            </c:forEach>
        </table>
        <object width="400" height="400" data="WebContent/WEB-INF/resources/Husky.png"></object>
        </div>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
</html>
```

ResultsError.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
        <style type="text/css">
                #btn_adminError {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                        background-color:black;
                }
                #alert {
                        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic,
AppleGothic, sans-serif;
                }
        </style>
</head>
<body>
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <div id="alert" class="alert alert-danger" role="alert">
                <center>Results will be displayed soon.</center>
        </div>
        <div class="container">
                <center><center><a id="btn_adminError" href="hello" class="btn btn-info" role="button" aria-
pressed="true">Home</a>
        </div>

</body></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</html>
```

Success.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
        <style type="text/css">
                #btn_adminError {
                        font-family:    "Avant    Garde",    Avantgarde,    "Century    Gothic",    CenturyGothic,
AppleGothic, sans-serif;
                        background-color:black;
                }
                #alert {
                        font-family:    "Avant    Garde",    Avantgarde,    "Century    Gothic",    CenturyGothic,
AppleGothic, sans-serif;
                }
        </style>
</head>
<body>
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <div id="alert" class="alert alert-success" role="alert">
                <center>Operation Successful.</center>
        </div>
        <div class="container">
                <center><center><a id="btn_adminError" href="index" class="btn btn-info" role="button" aria-
pressed="true">Login</a>
        </div>

</body><!-- </br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br> -->
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</html>
```

Vote.jsp

```
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<title>Spring Tiles Contact Form</title>
<style type="text/css">
body {
        background-image: url("\WebContent\header.png");
}

#votePerformed {
        max-width: 400px;
        font-size: 1.5em;
        font-weight: bold;
        margin: 35px auto 0 auto;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}

#votePerformedBtn{
        background-color:black;
```

```css
        border-color:black;
}
h2 {
        text-align:center;
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
        clear: both;
   display: inline-block;
   overflow: hidden;
   white-space: nowrap;
}
table{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        color:white
}
tbody {
        background:black;
        font-size: 1em;
}
td:hover{
        background: linear-gradient(to bottom, #b23435 0%,#891516 100%);;
}
label:hover{
        font-size: 1.5em;
}
thead{
        text-align:center
        font-size: 1em;
        font-weight: bold;
        background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
#radiobtnLaebl{
        width: 50px;
        position: relative;
        margin: 0 0 10px 0;
        font-size: 16px
}

</style>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
</head>
<body>
        <div class="container">
                <form class="vote" id="votePerformed" >
                <h2 class="text-center">Vote for Betterment</h2>
                <table class="table table-bordered ">
                 <thead id="tableHeader">
            <tr>
             <th>PartyName</th>
             <th>Candidates</th>
            </tr>
            </thead>
                <c:forEach var='item' items='${parties}'>
                   <div class="radio">
```

```html
                <tr>
                 <label id="radiobtnLaebl"><td><c:out value="${item.key}"/></td></label>
                 <td><c:forEach items='${item.value}' var="pname">
                 <label><input id = "partyAndCandidateName" type="radio" name= "partyAndCandidateName"
value="${pname}-${item.key}"><c:out value="${pname}"/></label></br>
                 </c:forEach></td>
                </tr>
                </div>
                </c:forEach>
        </table>
                </br></br>
                <button       class="btn       btn-lg       btn-primary       btn-block"      id="votePerformedBtn"
type="submit">Vote</button>
        </div>
        </form>

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

                                <!-- Modal content-->
                                <div class="modal-content">
                                        <div id="mheader" class="modal-header">
                                                <button        type="button"         class="close"        data-
dismiss="modal">&times;</button>
                                                <h4 class="modal-title">Vote Performed</h4>
                                        </div>
                                        <div id="modalBody" class="modal-body">
                                                <p   id=testmodalpara>Your   vote   has   been   taken   into
consideration!</p>
                                        </div>
                                        <div class="modal-footer">
                                                <button    type="button"    class="btn    btn-default"    data-
dismiss="modal">Close</button>
                                        </div>
                                </div>
                        </div>
                        <p></p>
                </div>

        <script type="text/javascript">
        $(document).ready(function() {
                $('#votePerformed').submit(function(e) {
                        e.preventDefault();
                        console.log('in javascript');
                        var formData = $('#votePerformed').serialize();
                        console.log("DATA:"+formData);
                        $.ajax({
                                data : formData,
                                type : 'POST',
                                url : 'votePerformed',
                                success : function(result) {
                                        $('#myModal').modal('show');
                                }
                        });
                });
```

```
        });
        </script>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body><!-- </br></br></br></br></br></br></br></br></br></br></br></br> -->
</html>
```

VoteError.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE     html      PUBLIC        "-//W3C//DTD      HTML       4.01       Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
        <style type="text/css">
                #btn_adminError {
                        font-family:  "Avant  Garde",  Avantgarde,  "Century  Gothic",  CenturyGothic,
AppleGothic, sans-serif;
                        background-color:black;
                }
                #alert {
                        font-family:  "Avant  Garde",  Avantgarde,  "Century  Gothic",  CenturyGothic,
AppleGothic, sans-serif;
                }
        </style>
</head>
<body>
</br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <div id="alert" class="alert alert-danger" role="alert">
                <center>Error performing requested operation.</center>
        </div>
        <div class="container">
                <center><center><a id="btn_adminError" href="index" class="btn btn-info" role="button" aria-
pressed="true">Login</a>
                <a    id="btn_adminError"    href="register"    class="btn    btn-info"    role="button"    aria-
pressed="true">Register</a></center>
        </div>

</body></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br></br>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</html>
```

spring-servlet.jsp

```
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:p="http://www.springframework.org/schema/p"
        xmlns:context="http://www.springframework.org/schema/context"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
```

```
http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.0.xsd">

    <context:component-scan base-package="com.javatpoint.*" />
    <bean id="viewResolver"
            class="org.springframework.web.servlet.view.UrlBasedViewResolver">
            <property name="viewClass">
                    <value>
                                org.springframework.web.servlet.view.tiles3.TilesView
                    </value>
            </property>
    </bean>
    <bean id="tilesConfigurer"
            class="org.springframework.web.servlet.view.tiles3.TilesConfigurer">
            <property name="definitions">
                    <list>
                                <value>/WEB-INF/tiles.xml</value>
                    </list>
            </property>
    </bean>
</beans>
```

Tiles.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE tiles-definitions PUBLIC
    "-//Apache Software Foundation//DTD Tiles Configuration 3.0//EN"
    "http://tiles.apache.org/dtds/tiles-config_3_0.dtd">
<tiles-definitions>
        <definition name="base.definition" template="/WEB-INF/jsp/layout.jsp">
                <put-attribute name="title" value="" />
                <put-attribute name="header" value="/WEB-INF/jsp/header.jsp" />
                <put-attribute name="body" value="" />
                <put-attribute name="footer" value="/WEB-INF/jsp/footer.jsp" />
        </definition>
        <definition name="base.register" template="/WEB-INF/jsp/register_layout.jsp">
                <put-attribute name="title" value="" />
                <put-attribute name="header" value="/WEB-INF/jsp/headerWithoutMenu.jsp" />
                <put-attribute name="body" value="" />
                <put-attribute name="footer" value="/WEB-INF/jsp/footer.jsp" />
        </definition>
        <definition name="base.NoHeaderMenu" template="/WEB-INF/jsp/register_layout.jsp">
                <put-attribute name="title" value="" />
                <put-attribute name="header" value="/WEB-INF/jsp/headerWithoutMenu.jsp" />
                <put-attribute name="body" value="" />
                <put-attribute name="footer" value="/WEB-INF/jsp/footer.jsp" />
        </definition>
        <definition name="contact" extends="base.definition">
                <put-attribute name="title" value="Contact Manager" />
                <put-attribute name="body" value="/WEB-INF/jsp/contact.jsp" />
        </definition>

        <definition name="hello" extends="base.definition">
                <put-attribute name="title" value="Hello Spring MVC" />
```

```
        <put-attribute name="body" value="/WEB-INF/jsp/hello.jsp" />
</definition>
<definition name="vote" extends="base.definition">
        <put-attribute name="title" value="Hello Spring MVC" />
        <put-attribute name="body" value="/WEB-INF/jsp/vote.jsp" />
</definition>
<definition name="parties" extends="base.definition">
        <put-attribute name="title" value="Hello Spring MVC" />
        <put-attribute name="body" value="/WEB-INF/jsp/parties.jsp" />
</definition>
<definition name="results" extends="base.definition">
        <put-attribute name="title" value="Hello Spring MVC" />
        <put-attribute name="body" value="/WEB-INF/jsp/results.jsp" />
</definition>
<definition name="analysis" extends="base.definition">
        <put-attribute name="title" value="Hello Spring MVC" />
        <put-attribute name="body" value="/WEB-INF/jsp/analysis.jsp" />
</definition>
<definition name="adminActions" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="Welcome Admin" />
        <put-attribute name="body" value="/WEB-INF/jsp/adminActions.jsp" />
</definition>

<definition name="adminActionsError" extends="base.register">
        <put-attribute name="title" value="Welcome Admin" />
        <put-attribute name="body" value="/WEB-INF/jsp/adminActionsError.jsp" />
</definition>

<definition name="adminError" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="Admin Login Error" />
        <put-attribute name="body" value="/WEB-INF/jsp/adminError.jsp" />
</definition>

<definition name="index" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="VEB" />
        <put-attribute name="body" value="/WEB-INF/jsp/index.jsp" />
</definition>

<definition name="voterError" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="Admin Login Error" />
        <put-attribute name="body" value="/WEB-INF/jsp/voterError.jsp" />
</definition>
<definition name="ResultsError" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="Results Error" />
        <put-attribute name="body" value="/WEB-INF/jsp/ResultsError.jsp" />
</definition>

<definition name="electionProcessError" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="Results Error" />
        <put-attribute name="body" value="/WEB-INF/jsp/electionProcessError.jsp" />
</definition>

<definition name="suxes" extends="base.NoHeaderMenu">
        <put-attribute name="title" value="Operation Successful" />
        <put-attribute name="body" value="/WEB-INF/jsp/success.jsp" />
```

```
        </definition>

        <definition name="register" extends="base.register">
                <put-attribute name="title" value="Register" />
                <put-attribute name="body" value="/WEB-INF/jsp/register.jsp" />
        </definition>

        <definition name="forgotPassword" extends="base.register">
                <put-attribute name="title" value="ForgotPassword" />
                <put-attribute name="body" value="/WEB-INF/jsp/forgotPassword.jsp" />
        </definition>

        <definition name="admin" extends="base.NoHeaderMenu">
                <put-attribute name="title" value="Admin" />
                <put-attribute name="body" value="/WEB-INF/jsp/admin.jsp" />
        </definition>
</tiles-definitions>
```

index.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<head>
<link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
<style type="text/css">
.login {
        width: 300px;
        height: 100px;
        position: absolute;
        top: 0;
        bottom: 0;
        left: 0;
        right: 0;
        margin: auto;
        left:900px;
        top: -300px;

        display: inline;
}
.leftDiv {
        width: 300px;
        height: 100px;
        position: absolute;
        top: 0;
        bottom: 0;
        left: 0;
        right: 0;
        margin: auto;
        left:-300px;
        top: 300px;
}
#login{
        background-color:black;
        border-color:black;
}
```

```css
#reglink {
        text-align: center;
        font-weight: bold;
        color: teal;
}
a{
        color: black;
}

h2 {
        font-size: 3em;
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-weight: bold;
        clear: both;
   display: inline-block;
   overflow: hidden;
   white-space: nowrap;
}
#reglink{
        color:black;
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
        font-size: 1.5em;
}
h4 {
        text-align: center;
        font-weight: 600;
}

body {
                font-weight: bold;
        background:linear-gradient(to bottom, #b23435 0%,#891516 100%);
}
.form-signin{
        font-family: "Avant Garde", Avantgarde, "Century Gothic", CenturyGothic, AppleGothic, sans-serif;
}
</style>
</head>
<body>
<img src="<c:url value='WebContent/WEB-INF/resources/huskies.png'/>"/>
<img src="<c:url value='/WEB-INF/resources/huskies.png'/>"/>
<img src="<c:url value='/resources/huskies.png'/>"/>
<div class="leftDiv"></div>
<img src="<c:url value='/resources/huskies.png'/>"/>
        <div class="login">
                <form class="form-signin" method="post" action="loginPage" >
                        <h2>Virtual Election Booth</h2>
                        <input type="text" class="form-control" name="validationNumber"
                                placeholder="validationNumber" required="" autofocus="" /><br> <input
                                type="password" class="form-control" name="password"
                                placeholder="Password" required="" /> <label class="checkbox">
                        <button        class="btn        btn-lg        btn-primary        btn-block"        id="login"
type="submit">Login</button>
                        <a href="register.html" action="register" role="button"  id="reglink">Register</a>
```

```
                          <a       href="forgotPassword.html"       action="forgotPassword"       role="button"
id="forgotPassword">Forgot Password?</a>
                </label>
                </form>
        </div>
        <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br></br></b
r>
        <div>
    <div class="text-center center-block">
        <a href="https://www.facebook.com/bootsnipp"><i id="social-fb" class="fa fa-facebook-square fa-3x
social"></i></a>
                <a   href="https://twitter.com/bootsnipp"><i   id="social-tw"   class="fa  fa-twitter-square  fa-3x
social"></i></a>
                <a href="https://plus.google.com/+Bootsnipp-page"><i id="social-gp" class="fa  fa-google-plus-
square fa-3x social"></i></a>
                <a href="mailto:bootsnipp@gmail.com"><i id="social-em" class="fa  fa-envelope-square  fa-3x
social"></i></a>
                </div>
        <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.min.css" rel="stylesheet">
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</body>
```

Web.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/root-context.xml</param-value>
  </context-param>
  <servlet>
                <servlet-name>spring</servlet-name>
                <servlet-class>
                        org.springframework.web.servlet.DispatcherServlet
                </servlet-class>
                <load-on-startup>1</load-on-startup>
        </servlet>

        <servlet-mapping>
                <servlet-name>spring</servlet-name>
                <url-pattern>/</url-pattern>
        </servlet-mapping>
  <listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
</web-app>
```

Pom.xml

```xml
<project    xmlns="http://maven.apache.org/POM/4.0.0"    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```xml
<modelVersion>4.0.0</modelVersion>
<groupId>Tiles</groupId>
<artifactId>Tiles</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<dependencies>
        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-context</artifactId>
                <version>4.3.7.RELEASE</version>
        </dependency>
        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-aop</artifactId>
                <version>4.3.7.RELEASE</version>
        </dependency>
        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-webmvc</artifactId>
                <version>4.3.7.RELEASE</version>
        </dependency>
        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-web</artifactId>
                <version>4.3.7.RELEASE</version>
        </dependency>
        <dependency>
                <groupId>javax.servlet</groupId>
                <artifactId>jstl</artifactId>
                <version>1.2</version>
        </dependency>
        <dependency>
                <groupId>javax.servlet</groupId>
                <artifactId>javax.servlet-api</artifactId>
                <version>3.0.1</version>
                <scope>provided</scope>
        </dependency>
        <dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>2.0.4</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>2.0.4</version>
</dependency>

                <dependency>
                <groupId>org.webjars</groupId>
                <artifactId>bootstrap</artifactId>
                <version>3.3.6</version>
        </dependency>
        <dependency>
                <groupId>org.webjars</groupId>
```

```xml
                    <artifactId>jquery</artifactId>
                    <version>1.9.1</version>
        </dependency>
        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-jsp</artifactId>
                    <version>3.0.0</version>
        </dependency>

        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-core</artifactId>
                    <version>3.0.0</version>
        </dependency>

        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-api</artifactId>
                    <version>3.0.0</version>
        </dependency>

        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-servlet</artifactId>
                    <version>3.0.0</version>
        </dependency>

        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-template</artifactId>
                    <version>3.0.0</version>
        </dependency>
        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-jsp</artifactId>
                    <version>3.0.0</version>
        </dependency>

        <dependency>
                    <groupId>org.apache.tiles</groupId>
                    <artifactId>tiles-core</artifactId>
                    <version>3.0.0</version>
        </dependency>

        <dependency>
                    <groupId>org.slf4j</groupId>
                    <artifactId>slf4j-log4j12</artifactId>
                    <version>1.5.6</version>
        </dependency>
        <dependency>
                    <groupId>com.microsoft.sqlserver</groupId>
                    <artifactId>sqljdbc4</artifactId>
                    <version>4.0</version>
        </dependency>
        <dependency>
```

```xml
            <groupId>com.google.code.gson</groupId>
            <artifactId>gson</artifactId>
            <version>1.4</version>
        </dependency>
        <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>5.1.6</version>
</dependency>
        </dependencies>
        <properties>
                <maven.compiler.source>1.8</maven.compiler.source>
                <maven.compiler.target>1.8</maven.compiler.target>
        </properties>
        <build>
                <sourceDirectory>src</sourceDirectory>
                <plugins>
                        <plugin>
                                <artifactId>maven-compiler-plugin</artifactId>
                                <version>3.3</version>
                                <configuration>
                                        <source>1.8</source>
                                        <target>1.8</target>
                                </configuration>
                        </plugin>
                        <plugin>
                                <artifactId>maven-war-plugin</artifactId>
                                <version>2.6</version>
                                <configuration>
                                        <warSourceDirectory>WebContent</warSourceDirectory>
                                        <failOnMissingWebXml>false</failOnMissingWebXml>
                                </configuration>
                        </plugin>
                </plugins>
        </build>
</project>
```