Gyan Ganga Institute of Technology and Sciences

Jabalpur, Madhya Pradesh

Major Project 2 Report

Cloud Native Voting Application

May 2022

Aaryan Soni	0206CS181001
Adarsh Singh Raghuwanshi	0206CS181011
Aman Jaisinghani	0206CS181017
Deepankar Dubey	0206CS181047
Dev Khare	0206CS181050
Dwij Jain	0206CS181056

Certificate

This is to certify that Aaryan Soni, Adarsh Singh Raghuwanshi, Aman Jaisinghani, Deepankar Dubey, Dev Khare and Dwij Jain, students of class CS-1 have worked on their project entitled “Cloud Native Voting Application” under my supervision. This project is approved for submission towards partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

Prof. Sapan Jain

Faculty, Computer Science and Engineering

Dr. Ashok Verma
Head of Department,
Computer Science and Engineering

Declaration

We hereby declare that the project entitled “Cloud Native Voting Application” submitted to Gyan Ganga Institute of Science and Technology, Jabalpur, is a record of original work done by our team under the guidance of Dr. Ashok Verma, Head of Department, Computer Science and Engineering.

This project is submitted in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. The results embodied in this thesis have not been submitted to any other University of Institute for the award of any other degree of any degree of diploma.

May 4, 2022

Jabalpur

Acknowledgement

We are grateful to Dr. Ashok Verma, Head of Department of Computer Science Engineering, Gyan Ganga Institute of Science and Technology, Jabalpur, for providing every help in accomplishing to its final stage the project on “Cloud Native Voting Application” within stipulated time.

The components of this software have been developed in consultation with my associate faculty and we have been greatly benefited by their thoughtful and constructive suggestions, for which we are further thankful to them.

Aaryan Soni
Adarsh Singh Raghuwanshi
Aman Jaisinghani
Deepankar Dubey
Dev Khare
Dwij Jain

May 4, 2022
Jabalpur

Table of Content

1. Introduction

1.1 Purpose

An online voting app is a Software Application that allows groups to securely conduct votes and elections.

High quality online voting applications provide security, accessibility, and an overall requirement to conduct the body in an efficient way.

1.2 Scope

At their core, online voting applications protect the integrity of your vote by preventing voters from being able to vote multiple times. As a digital platform, they eliminate the need to gather in-person, cast votes using paper, or by any other means (e.g. email, insecure survey software).

You may hear an online voting application being referred to as an online election system, an online e voting system, or electronic voting. These all refer to the same thing: a secure voting tool that allows your group to collect input from your group and closely scrutinise the results in real time.

1.3 Target Audience and Assumption

Anyone with some programming experience can understand this document.

The document is intended for developers, software architects, testers, project managers and documentation writers.

1.4 Description

Cloud Native Voting Application is be a desktop software application in order to provide all the required features mentioned in the document.

Product Features: This software package will allow organizations to design and run polls securely and privately on a dedicated server. It will be possible to customize the polls before every round of voting. The software will provide voters to register their votes on the different alternatives that are available and helps them view the results alongside it. The result data from the voters will be used to calculate statistics and displays the percentage share of all the alternates provided.

1.5 Team Structure and Organisation

Name	**Roles**	
Aaryan Soni	Backend	Networking
Adarsh Singh Raghuwanshi		Oracle Cloud
Dwij Jain		Containerization
Aman Jaisinghani	Web Dev	HTML, CSS, JS
Deepankar Dubey		Flask, Angular
Dev Khare	Documentation and Testing	

Team 1

The task of team 1 is to make the application cloud native.

Team 2

The assigned task of team 2 was to develop the website.

Team 3

Documentation and Testing: The responsibility of Dev Khare was to keep documenting the progress made by the team, handle version control, and continuously integrate progress made after testing code.

1.6 Duration

SNo.	Project Phase	Working Days
1	Business Requirements	7
2	Application Design	7
3	Development and Testing	20
4	Documentation	7
5	Deployment User Acceptance Testing	4
6	Project Management	5
	Total	50

2. System Overview

This section is an overview of the main features of the project. Requirements are structured by functionality.

2.1 Functional Requirements

- Internet
- List of candidates eligible for voting (candidate list).
- Web application - A modern browser (Chrome, Firefox, Opera).
- Eligible parties for which a vote can be casted.
- A national identification card (Voter Id, Aadhar Card).

2.1.1 Hardware Requirements

PC with minimum Pentium Dual Core.
RAM is more than 2 GB.
Keyboard and mouse or any equivalent device.

2.1.2 Software Requirements

Google Chrome or any latest browser.
OS: Linux or Windows with WSL.
Latest version of Docker installed on server.

2.1.3 User Interface Requirements

User interface requirements are briefly mentioned below:

1. Content presentation
2. Easy Navigation
3. Simple interface
4. Responsive
5. Consistent UI elements
6. Default settings
7. User centric approach

2.1.4 Interactive Components

There are two sections of the website: vote and view result.
Voting: User can select a candidate to vote for.
View result: Result is displayed for the user.

2.1.5 Non-interactive Components

Non-interactive components of this project are icons, background colours, button shapes and sizes, and other static components required to make the application work smoothly.

2.2 Non-functional Requirements

This section describes the non-functional requirements of the project. The non-functional requirements are to secure against attackers, the application should be easy to use, automatic, universally available and for individuals.

2.2.1 Usability Requirements

The application has a simple interface which is intuitive to use for the user. Though it does not have the latest design guidelines, it still provides all the basic functionality for user interaction. The system developed is intended to be used by the general public and should not have a steep learning curve. Also, anyone around the globe can access the application with no barriers.

2.2.2 Security Requirements

Anonymous voting
Data transfer is secure

2.2.3 Performance Requirements

Vote should be casted in a timely manner.
Casting should be guaranteed.
The entire transaction takes place at once or doesn't happen at all.
The database must be consistent before and after the transaction.
Multiple transactions occur independently without interference.
The changes of a successful transaction occur even if the system failure occurs.

2.2.4 Reliability Requirements

Reliability is an attribute which specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions.

2.2.5 Project Methodology

Agile and Scrum software development approach was used to develop this project. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change. Scrum is one of the methodologies that follow Agile principles. It is a combination of iterative and incremental models. Agile focuses on a single task with daily collaboration and communication and focuses on fast delivery.

Agile software development model, when compared to traditional software development models is fast, flexible and cost effective. By clarifying an acceptance criterion, estimation and breakdown of user stories, tasks can be broken down that can fit in a spirit cycle.

In the spirit pre planning phases of this project, user stories were estimated and broken down into Themes, Epics, Stories and Tasks. A task was assigned to each team member. The tasks when completed were reviewed, tested and then used as a base for the next spirit cycle.

Agile software development approach enabled us to have small scale focuses with rapid development cycles and minimised resources allocated to testing.

2.3 Design Techniques

Cloud

"The cloud" refers to servers that are accessed over the Internet, and the software and databases that run on those servers. Cloud servers are located in data centres all over the world. By using cloud computing, users and companies do not have to manage physical servers themselves or run software applications on their own machines.

The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data centre, instead of locally on the user device. This is why a user can log in to their Instagram account on a new phone after their old phone breaks and still find their old account in place, with all their photos, videos, and conversation history.

Java

Java is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and that can be used to create applications or applets.

We have used the following properties of Java:

- Variable Declaration: The types of all variables must be declared. The primitive types are byte, short, int, long (8, 16, 32, and 64 bit integer variables, respectively), float and double (32 and 64-bit floating point variables), Boolean (true or false), and char. Boolean is a distinct type rather than just another way of using integers. Strings are not a primitive type, but are instances of the String class. Because they are so common, string literals may appear in quotes just as in other languages.
- Classes are effectively new programmer-defined types; each class defines data (fields) and methods to manipulate the data. Fields in the class are template for the instance variables that are created when objects are instantiated (created) from that class. A new set of instance variables is created each time that an object is instantiated from the class.

Redis

Redis is an open source (BSD licensed), in-memory data structure store used as a database, cache, message broker, and streaming engine. Redis provides data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

PostgreSQL

PostgreSQL, commonly pronounced “Post-GRES,” is an open source database that has a strong reputation for its reliability, flexibility, and support of open technical standards. Unlike other RDMBS (Relational Database Management Systems), PostgreSQL (link resides outside ibm.com) supports both non-relational and relational data types. This makes it one of the most compliant, stable, and mature relational databases available today.

2.4 Tier Architecture

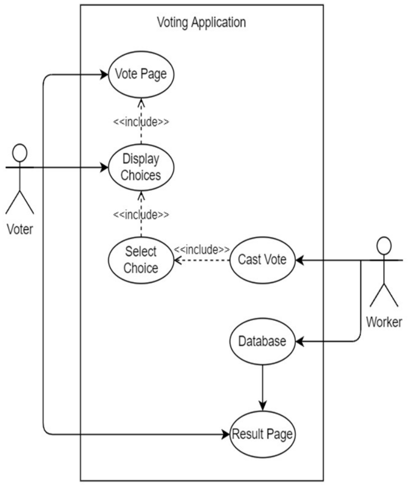
Presentation Layer - The web site or the voting application is called the presentation layer. The presentation layer is the most important layer simply because it's the one that everyone sees and uses. Even with a well-structured business and data layer, if the presentation layer is designed poorly, this gives the users a poor view of the system. Presentation layer is the form where we design using the controls like textbox, labels, command buttons etc.

Business Layer: Though a web site could talk to the data access layer directly, it usually goes through another layer called the business layer. This layer is a class which we use to write the function which works as a mediator to transfer the data from Application or presentation layer data layer. In the three-tier architecture we never let the data access layer to interact with the presentation layer.

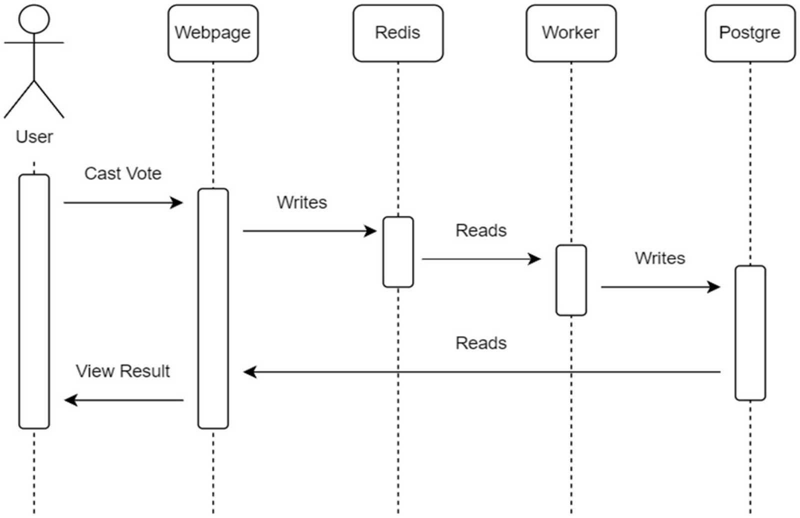
Data layer: The key component to most applications is the data. The data has to be served to the presentation layer somehow. The data layer is a separate component whose sole purpose is to serve up the data from the database and return it to the caller.

3. Diagrams

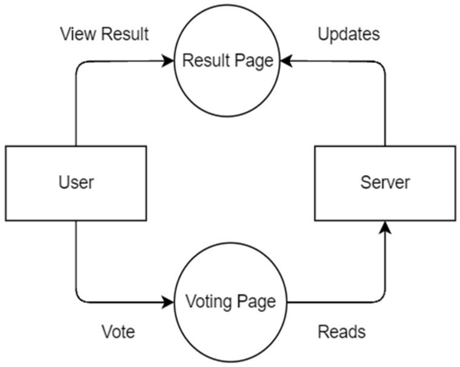
Use Case Diagram



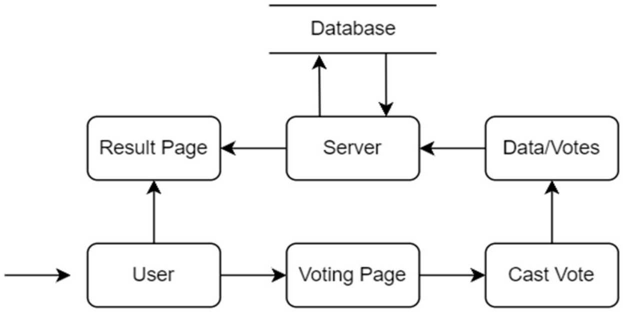
Sequence Diagram



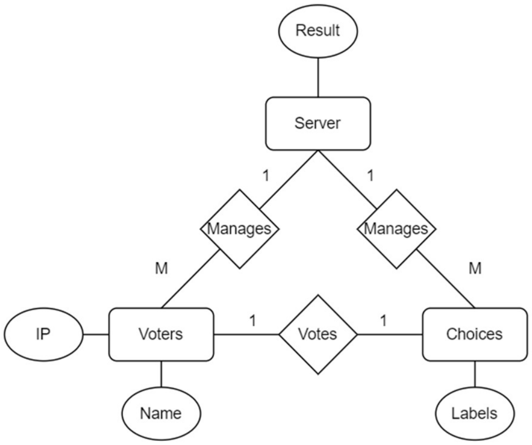
Data Flow Diagram Level 0



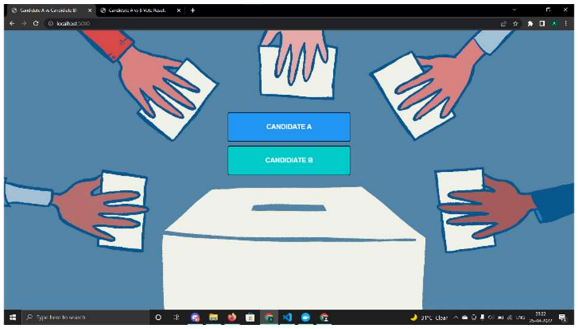
Data Flow Diagram Level 1



Entity Relationship

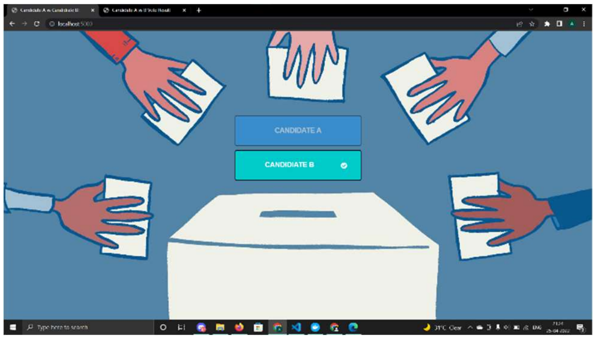


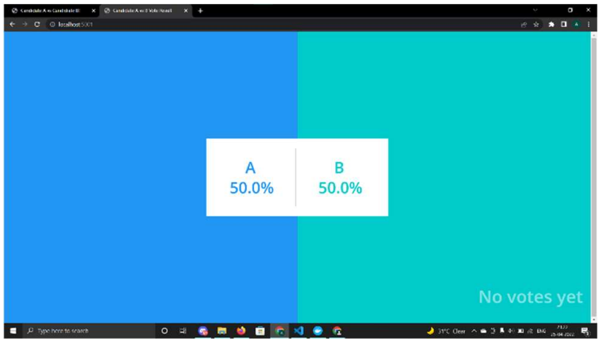
4.Screenshots



Voting Page

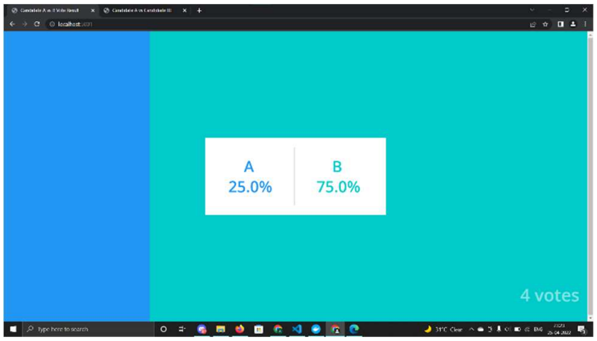
Vote Selected

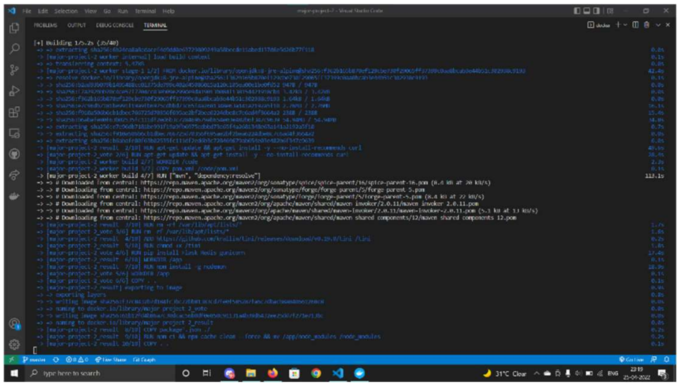




Result Page

Result Page after Votes



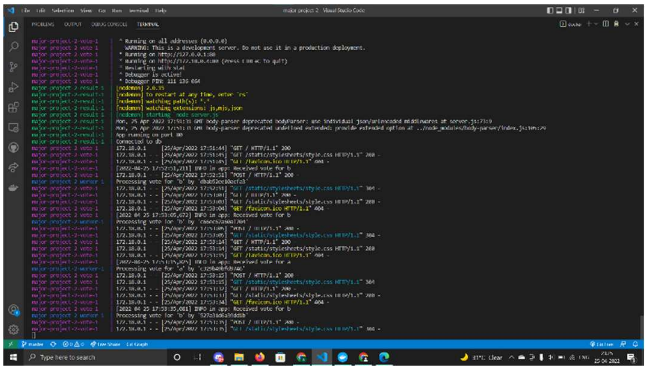


Buildtime Logs

File Edit Selection View Go Run Terminal Help
major-project-2 - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
docker

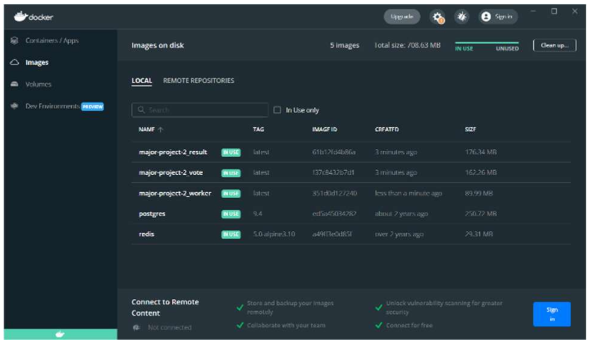
```
[+] Building 1/5.2s (35/40)
 => => extracting sha256:[illegible]                                                          0.8s
 => [major-project-2 worker internal] load build context                                      0.1s
 => => transferring context: 5.47kB                                                           0.0s
 => [major-project-2 worker stage-1 1/2] FROM docker.io/library/openjdk:8-jre-alpine@sha256:[illegible]   42.4s
 => => resolve docker.io/library/openjdk:8-jre-alpine@sha256:[illegible]                      0.1s
 => => sha256:[illegible] 947B / 947B                                                         0.0s
 => => sha256:[illegible] 1.47kB / 1.47kB                                                     0.0s
 => => sha256:f362b165b870ef129cbe730f29065ff37399c0aa8bcab3e44b51c302938c9193 1.64kB / 1.64kB 0.0s
 => => sha256:[illegible] 2.76MB / 2.76MB                                                     16.1s
 => => sha256:f910a50d6cb1dbec780725d78350f035ac2bf2bea0224dbc8c7c6ad4f5664a2 238B / 238B     15.4s
 => => sha256:[illegible] 54.9MB / 54.9MB                                                     34.8s
 => => extracting sha256:[illegible]                                                          0.7s
 => => extracting sha256:[illegible]                                                          0.0s
 => => extracting sha256:[illegible]                                                          6.8s
 => [major-project-2 result  2/10] RUN apt-get update && apt-get install -y --no-install-recommends curl   49.6s
 => [major-project-2_vote 2/6] RUN apt-get update && apt-get install -y --no-install-recommends curl       38.4s
 => [major-project-2 worker build 2/7] WORKDIR /code                                          2.3s
 => [major-project-2_worker build 3/7] COPY pom.xml /code/pom.xml                             0.1s
 => [major-project-2 worker build 4/7] RUN ["mvn", "dependency:resolve"]                      113.1s
 => => # Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/16/spice-parent-16.pom (8.4 kB at 20 kB/s)
 => => # Downloading from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge parent/5/forge parent 5.pom
 => => # Downloaded from central: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/5/forge-parent-5.pom (8.4 kB at 22 kB/s)
 => => # Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven invoker/2.0.11/maven invoker 2.0.11.pom
 => => # Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven-invoker/2.0.11/maven-invoker-2.0.11.pom (5.1 kB at 13 kB/s)
 => => # Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/shared/maven shared components/12/maven shared components 12.pom
 => [major-project-2 result  3/10] RUN rm -rf /var/lib/apt/lists/*                            1.7s
 => [major project 2_vote 3/6] RUN rm -rf /var/lib/apt/lists/*                                1.8s
 => [major-project-2 result  4/10] ADD https://github.com/krallin/tini/releases/download/v0.19.0/tini /tini   0.2s
 => [major project 2_result  5/10] RUN chmod +x /tini                                         1.8s
 => [major-project-2 vote 4/6] RUN pip install flask Redis gunicorn                           17.4s
 => [major-project-2_result  6/10] WORKDIR /app                                               0.1s
 => [major-project-2 result  7/10] RUN npm install -g nodemon                                 18.9s
 => [major-project-2_vote 5/6] WORKDIR /app                                                   0.1s
 => [major project 2_vote 6/6] COPY . .                                                       0.1s
 => [major-project-2_result] exporting to image                                               0.9s
 => => exporting layers                                                                       0.8s
 => => writing image sha256:[illegible]                                                       0.0s
 => => naming to docker.io/library/major project 2_vote                                       0.0s
 => => writing image sha256:[illegible]                                                       0.0s
 => => naming to docker.io/library/major project 2_result                                     0.0s
 => [major-project-2 result  8/10] COPY package*.json ./                                      0.2s
 => [major project 2_result  9/10] RUN npm ci && npm cache clean --force && mv /app/node_modules /node_modules   0.2s
 => [major-project-2 result 10/10] COPY . .                                                   0.1s
[]
```

Runtime Logs



File Edit Selection View Go Run Terminal Help
major project 2 - Visual Studio Code
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
docker

```
major-project-2-vote-1     |  * Running on all addresses (0.0.0.0)
major project 2 vote 1     |    WARNING: This is a development server. Do not use it in a production deployment.
major project 2-vote-1     |  * Running on http://127.0.0.1:80
major-project-2-vote-1     |  * Running on http://172.18.0.6:80 (Press CTRL+C to quit)
major-project-2-vote-1     |  * Restarting with stat
major-project-2-vote-1     |  * Debugger is active!
major project 2 vote 1     |  * Debugger PIN: 111 136 664
major project 2 result 1   | [nodemon] 2.0.15
major-project-2-result-1   | [nodemon] to restart at any time, enter `rs`
major-project-2-result-1   | [nodemon] watching path(s): *.*
major-project-2-result-1   | [nodemon] watching extensions: js,mjs,json
major project 2 result 1   | [nodemon] starting `node server.js`
major project 2 result 1   | Mon, 25 Apr 2022 17:51:31 GMT body-parser deprecated bodyParser: use individual json/urlencoded middlewares at server.js:73:9
major-project-2-result-1   | Mon, 25 Apr 2022 17:51:31 GMT body-parser deprecated undefined extended: provide extended option at ../node_modules/body-parser/index.js:105:29
major-project-2-result-1   | App running on port 80
major-project-2-result-1   | Connected to db
major project 2 vote 1     | 172.18.0.1 - - [25/Apr/2022 17:51:44] "GET / HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:51:45] "GET /static/stylesheets/style.css HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:51:45] "GET /favicon.ico HTTP/1.1" 404 -
major-project-2-vote-1     | [2022-04-25 17:52:51,211] INFO in app: Received vote for b
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:52:51] "POST / HTTP/1.1" 200 -
major project 2 worker 1   | Processing vote for 'b' by 'dbab52ee10ecfa3'
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:52:51] "GET /static/stylesheets/style.css HTTP/1.1" 304 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:03] "GET / HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:03] "GET /static/stylesheets/style.css HTTP/1.1" 200 -
major project 2 vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:04] "GET /favicon.ico HTTP/1.1" 404 -
major project 2 vote 1     | [2022 04 25 17:53:05,672] INFO in app: Received vote for b
major-project-2-worker-1   | Processing vote for 'b' by 'c66ecc67a60a1704'
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:05] "POST / HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:05] "GET /static/stylesheets/style.css HTTP/1.1" 304 -
major project 2 vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:14] "GET / HTTP/1.1" 200 -
major project 2 vote 1     | 172.18.0.1 - - [25/Apr/2022 17:53:14] "GET /static/stylesheets/style.css HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:15] "GET /favicon.ico HTTP/1.1" 404 -
major-project-2-vote-1     | [2022-04-25 17:53:15,325] INFO in app: Received vote for a
major-project-2-worker-1   | Processing vote for 'a' by 'c329b49bfd9746'
major project 2 vote 1     | 172.18.0.1 - - [25/Apr/2022 17:53:15] "POST / HTTP/1.1" 200 -
major project 2 vote 1     | 172.18.0.1 - - [25/Apr/2022 17:53:15] "GET /static/stylesheets/style.css HTTP/1.1" 304 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:32] "GET / HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:33] "GET /static/stylesheets/style.css HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:34] "GET /favicon.ico HTTP/1.1" 404 -
major project 2 vote 1     | [2022 04 25 17:53:35,081] INFO in app: Received vote for b
major project 2 worker 1   | Processing vote for 'b' by '527a31d6a39dd1b'
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:35] "POST / HTTP/1.1" 200 -
major-project-2-vote-1     | 172.18.0.1 - - [25/Apr/2022 17:53:35] "GET /static/stylesheets/style.css HTTP/1.1" 304 -
[]
```



Docker Images

Images on disk 5 images Total size: 708.63 MB IN USE UNUSED

□ In Use only

NAME ↑		TAG	IMAGE ID	CREATED	SIZE
major-project-2_result	IN USE	latest	61b17fd4b86a	3 minutes ago	176.34 MB
major-project-2_vote	IN USE	latest	f37c8432b7d1	3 minutes ago	162.26 MB
major-project-2_worker	IN USE	latest	351d0d127240	less than a minute ago	89.99 MB
postgres	IN USE	9.4	ed5a45034282	about 2 years ago	250.72 MB
redis	IN USE	5.0-alpine3.10	a49f3e0d85f	over 2 years ago	29.31 MB

Connect to Remote Content

Not connected

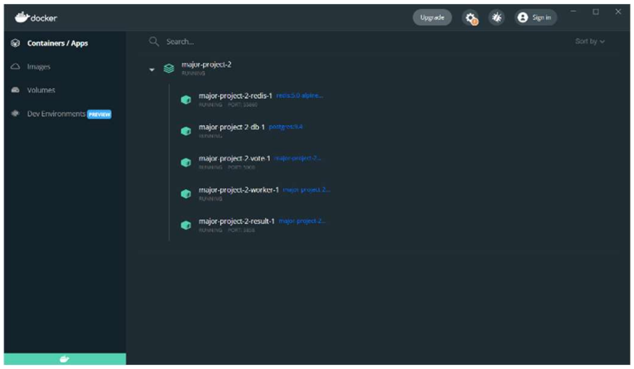
✓ Store and backup your images remotely

✓ Collaborate with your team

✓ Unlock vulnerability scanning for greater security

✓ Connect for free

Docker Containers



- major-project-2 RUNNING
  - major-project-2-redis-1 redis:5.0-alpine... RUNNING PORT: 55000
  - major project 2 db 1 postgres:9.4 RUNNING
  - major project 2 vote 1 major-project-2... RUNNING PORT: 5000
  - major-project-2-worker-1 major project 2... RUNNING
  - major-project-2-result-1 major project 2... RUNNING PORT: 5858

4.1 Code Snippet

Frontend

```
<body>
    <div id="content-container">
      <div id="content-container-center">
        <form id="choice" name='form' method="POST" action="/">
          <button id="a" type="submit" name="vote" class="a"
value="a">{{option_a}}</button>
          <button id="b" type="submit" name="vote" class="b"
value="b">{{option_b}}</button>
        </form>
      </div>
    </div>
    <script src="http://code.jquery.com/jquery-latest.min.js"
type="text/javascript"></script>
    <script src="//cdnjs.cloudflare.com/ajax/libs/jquery-
cookie/1.4.1/jquery.cookie.js"></script>
 </body>
```

Backend

```
package worker;

import redis.clients.jedis.Jedis;
import redis.clients.jedis.exceptions.JedisConnectionException;
import java.sql.*;
import org.json.JSONObject;

class Worker {
  public static void main(String[] args) {
    try {
      Jedis redis = connectToRedis("redis");
      Connection dbConn = connectToDB("db");

      System.err.println("Watching vote queue");

      while (true) {
        String voteJSON = redis.blpop(0, "votes").get(1);
        JSONObject voteData = new JSONObject(voteJSON);
        String voterID = voteData.getString("voter_id");
        String vote = voteData.getString("vote");

        System.err.printf("Processing vote for '%s' by '%s'\n", vote,
voterID);
        updateVote(dbConn, voterID, vote);
      }
    } catch (SQLException e) {
      e.printStackTrace();
      System.exit(1);
    }
  }

  static void updateVote(Connection dbConn, String voterID, String vote)
throws SQLException
  { ... }

  static Jedis connectToRedis(String host)
  { ... }

  static Connection connectToDB(String host) throws SQLException
  { ... }

  static void sleep(long duration)
  { ... }
}
```

5 Software Testing

Testing is the process of executing a program with the aim of finding errors. We perform testing of our system before deployment for efficient functioning of our system.

Types of Testing:

1 Functional Testing
It ensures that the application is functioning correctly. This type of testing focuses on the main purpose and flow of the application, ensuring that all its features are responsible and meet specifications

2 Usability Testing
Usability testing is a technique used in user-centred interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system. It is more concerned with the design intuitiveness of the product and tested with users who have no prior exposure to it. Such testing is paramount to the success of an end product as a fully functioning application that creates confusion amongst its users will not last for long.

3 Compatibility Testing
Compatibility testing is a part of non-functional testing conducted on application software to ensure the application's compatibility with different computing environments.

Web Interface Testing

Module	Scenario	Expected Result	Result
User	Visits URL	Website loads	Success
Vote Select	User selects candidate	Data entered	Success
Vote Cast	User votes	Vote is cast	Success
Result View	User views result	Result is displayed	Success

Application Testing

Module	Scenario	Expected Result	Result
Load Testing	High traffic	Website handles load	Success
Cache Database	Data is fed	Details are recorded	Success
Worker	Fetches data from cache database	Data is fetched	Success
Persistent Database	Data is fed by Worker	Data is recorded	Success

6 Business Model

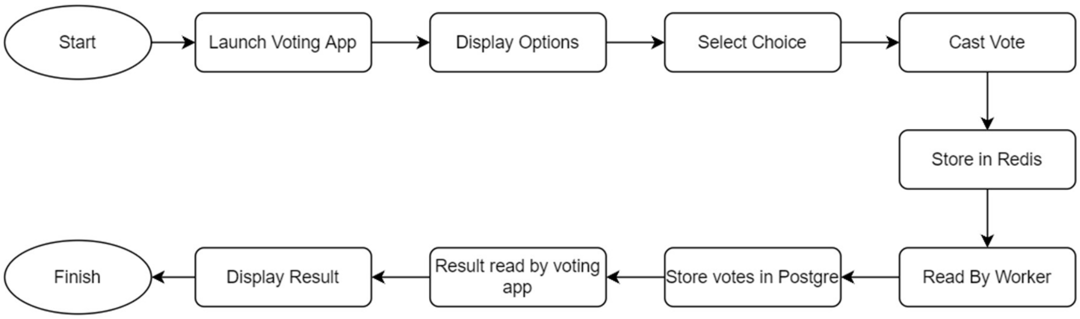
Business Requirement

Business Requirement focuses on the business perspective as it holds the details of the business solution for a project. It defines what the organization wants or needs to be able to do once the project is completed and also describes the changes in capabilities that will result from the project.

Business Process Modelling Notation (BPMN) is a flow chart method that models the steps of a planned business process from end to end.

Its purpose is to model ways to improve efficiency, account for new circumstances or gain competitive advantage. The method has been undergoing a standardisation push in the past few years and is now often called by a slightly different name: **Business Process Model and Notation**, still using the BPMN acronym. It differs from Unified Modelling Language (UML) used in software design.

Business Process Diagram



7 Future Prospects

The Online Voting System platform can be made more secure by using the following methods:

1. Password Changing
2. Fingerprinting
3. Cornea Detection

The password used by the user to vote is provided by the administrator. In the future the user can be given the privilege of changing the password. So it helps to increase the security of the system.

The other two methods that can be used are cornea detection and fingerprinting. But here the problem is that it decreases the scope of the platform because these systems need some electronic components to implement. So it will avoid the users privilege to cast the votes at their fingertips. But it can guarantee that fake voting will be impossible.

8 Conclusion

Using an online voting tool will generate confidence in the results of your votes and elections, lower your voting-related costs, and streamline the election process for both you and your voters.

Cost Savings and Efficiency

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

Voter Accessibility

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

High accessibility generates greater turnout rates among your group.

Auditability and Verifiability

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

Security, Confidence, and Trust in Your Election Results

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

9 References

1 AWS

https://docs.aws.amazon.com/general/latest/gr/Welcome.html

https://docs.aws.amazon.com/ec2/index.html

2 Docker

https://docs.docker.com/reference/

3 PostgreSQL

https://www.postgresql.org/docs/current/

4 Redis
https://redis.io/commands