**ONLINE VOTING PLATFORM WEB APPLICATION**

**Description:**

The Online Voting application is a platform designed to host elections and provide a seamless voting experience for users. It utilizes a tech stack consisting of EJS (Embedded JavaScript) for the client-side rendering, Tailwind CSS for responsive web design, Node.js and Express for the server-side implementation, and PostgreSQL as the database.

The main features of the Online Voting platform include:

1. Election Hosting: The platform allows users to host elections, enabling them to create and manage different voting events.

2. Voter Management: Users can add and manage voters, ensuring that only authorized individuals can participate in the elections.

3. Live Result Preview: The platform provides a real-time result preview, allowing participants to view the progress and outcome of the election as the votes are being counted.

4. Responsive Design: The use of Tailwind CSS ensures that the website is fully responsive, providing an optimal user experience across different devices and screen sizes.

5. CSRF Protection: The platform incorporates CSRF (Cross-Site Request Forgery) tokens to prevent unauthorized access and protect against attacks.

6. Automated Result Visibility: Once the election ends, the results are automatically made visible to the voters, ensuring transparency and enabling participants to review the outcome.

Overall, the Online Voting application provides a comprehensive platform for hosting elections, managing voters, and visualizing real-time results. It aims to ensure a secure and user-friendly experience for both election organizers and participants.

**Technologies used:**

**Client**: EJS (Embedded JavaScript), Tailwind CSS

**Server:** Node.js , Express.js

**Database:** PostgreSQL

**Deploy Web Application using:** RenderAPI

**Why we use?**

**1. npm (Node Package Manager):**

npm is the default package manager for Node.js, a JavaScript runtime environment. It allows developers to easily manage and install third-party libraries and packages for their Node.js projects. npm provides a vast ecosystem of open-source modules and tools that can be easily integrated into Node.js applications. It simplifies dependency management, version control, and project structuring, making it an essential tool for Node.js developers.

**2. Node.js:**

Node.js is a runtime environment that allows developers to run JavaScript code on the server-side. It provides an event-driven, non-blocking I/O model that makes it lightweight and efficient for building scalable and high-performance network applications. Node.js is built on the V8 JavaScript engine and provides a rich set of APIs and libraries that enable developers to build web servers, command-line tools, real-time applications, and more. It has gained popularity for its ability to handle concurrent requests and its vast ecosystem of modules available through npm.

**3. Express.js:**

Express.js is a minimal and flexible web application framework for Node.js. It provides a robust set of features for building web applications and APIs. Express.js simplifies the process of handling HTTP requests, routing, middleware integration, and view rendering. It follows the "middleware" approach, allowing developers to plug in various middleware functions to handle specific tasks such as authentication, logging, and error handling. Express.js is widely used and known for its simplicity and scalability, making it a popular choice for building web applications using Node.js.

**4. PostgreSQL:**

PostgreSQL is an open-source, object-relational database management system (DBMS). It provides a robust and scalable solution for storing and retrieving structured data. PostgreSQL supports a wide range of data types, including JSON, and offers advanced features such as transactions, concurrency control, and multi-version concurrency control (MVCC). It provides a SQL-based query language for interacting with the database and supports various indexing techniques for efficient data retrieval. PostgreSQL is highly reliable, secure, and widely used in both small-scale and enterprise-level applications.

In summary, npm is a package manager for Node.js, which is a runtime environment for running JavaScript on the server-side. Express.js is a web application framework that simplifies building web applications with Node.js. PostgreSQL is a powerful and feature-rich open-source database management system. Together, these technologies form a robust stack for building modern web applications.

**How to execute Project?**

**Step-1:** Basically, To run the project we have to use command:

1.Use **node filename** Eg :node index,js

2.By using scripts :

Before to run project you should mention particular command in scripts:

**Scripts: {**

**start: ”node index.js”**

**}**

**Command: npm start**

**Step-2:** Before to run the above command make sure **To start the PostgreSQL services:**

**Command : service postgres start**

Now, The server will start running and you should see a message indicating the server is listening on a specific port number (mostly we preferred port 3000).

**Step-3:**Open the web browser and visit [**http://localhost:3000**](http://localhost:3000)(or the specific port you mentioned)to access the Online Voting Application.

**About RenderAPI and Process to Deploy:**

1.Basically, RenderAPI is a cloud Platform that enables the developer to deploy and manage their application using a simple API(Application Interface). And alos we automate the process of deployment.

2.This render providers various services like :

1.creating static site

2.Web Services

3.Private services

4.PostgresSQL etc.

**Steps to follow for deployment:**

**Step-1:** First, we have to create a render account in **render.com** if you haven’t account in render.

**Note:** It is better to **Sign Up** with GitHub account rather than Gmail account or personal mail. The advantage of using GitHub account is to **easily to connect** with web Application and **Commit changes in GitHub repository of particular application (Here ,Online Voting Application) will automatically replicate those change in render application)**

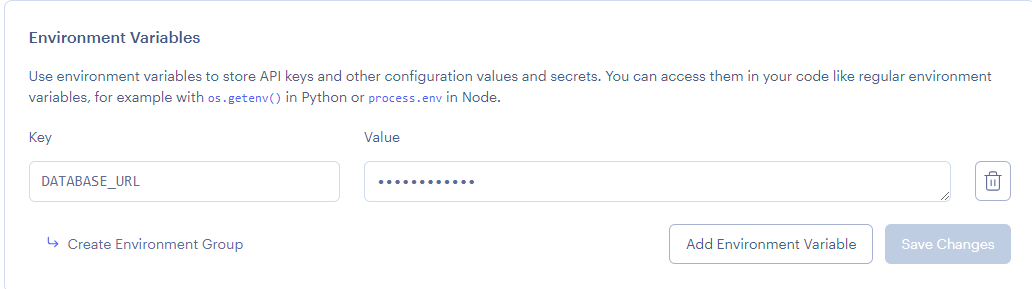
**Step-2**: Make sure ready with online voting application without any error in code. This helps to avoid getting errors in render Application while deployment Process.

**Note:** Those errors can see in log files panel in render application. If you get any error in render

**Step-3:** At First , Create the web services in render and do the setup which require for creation of web services

**Step-4**: Then Define the require configuration for application like specifying the Nodejs runtime ,and specifying the build command, build environment, setting up the environment variable.

**Snapshot of Environmental variable**

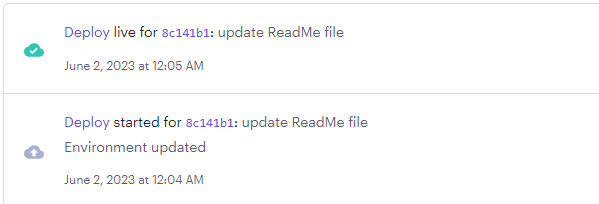


**Step-5:** At the same way, we have to create the PostgreSQL service to the application in renderAPI and do the setup for creation of database as mention in render.

**Step-6: whoohooo!** It’s time to deploy the application in renderAPI by click on **Deploy button** which is present in web Services

Now the Services are start deploying the application and after successful deployment .

**The Application is ready to host the web services:**

****

**Finally ,we get the Host or deployment Link for public access and we can share anyone to access our Application services.**

**--------THANK YOU---------**