**Online Voting System**

**Overview**

This is an **Online Voting System** developed using **Flask (Python Web Framework)** and hosted on **Render.com**. The system allows voters to cast their votes securely and provides an **admin panel** to manage voters, candidates, and view results.

**Features**

✅ **Voter Login** - Secure authentication for voters.  
✅ **Voting System** - Voters can cast a vote only once.  
✅ **Admin Panel** - Manage candidates & voters.  
✅ **Live Results** - View election results instantly.  
✅ **Delete Candidates & Voters** - Admin can remove entries if needed.  
✅ **Auto-Hosted on Render** - Accessible from anywhere online.

**Installation & Setup**

**1. Clone the Repository**

git clone https://github.com/your-username/your-repo.git

cd your-repo

**2. Install Dependencies**

pip install -r requirements.txt

**3. Run the Application Locally**

python app.py

* Open your browser and go to [**http://127.0.0.1:5000/**](http://127.0.0.1:5000/).

**4. Deploy on Render**

1. Push the code to **GitHub**.
2. Go to **Render.com** → Create **New Web Service**.
3. Connect to your **GitHub repository**.
4. Set:
   * **Build Command** → pip install -r requirements.txt
   * **Start Command** → gunicorn app:app
5. Click **Deploy** 🚀

**How to Use**

**For Voters:**

1. Open the website link.
2. Click on **"Voter Login"**.
3. Enter your **Voter ID** and **Password**.
4. If login is successful, select a **candidate** and **submit your vote**.
5. You will see a confirmation message.

**For Admin:**

1. Open the website link.
2. Click on **"Admin Login"**.
3. Enter \*\*Username: \*\* and \*\*Password: \*\*.
4. In the **Admin Dashboard**, you can:
   * **Add Candidates**
   * **Remove Candidates**
   * **Add Voters**
   * **Remove Voters**
   * **View Results**
5. Click **Logout** when done.

**Libraries Used & Why**

1. **Flask** - A lightweight web framework for building the application.
2. **Flask-SQLAlchemy** - ORM (Object Relational Mapper) for handling the database easily.
3. **Gunicorn** - A production-ready WSGI server used to deploy the app.

**Why We Used These Libraries?**

* **Flask** was chosen because it is simple, flexible, and perfect for small applications like this voting system.
* **Flask-SQLAlchemy** simplifies database handling, making it easier to manage voters and candidates.
* **Gunicorn** is required for deployment on **Render.com**, as it efficiently handles multiple requests.

**Admin Login**

* **Username:** admin
* **Password:** admin123

**Tech Stack**

* **Backend:** Flask (Python)
* **Database:** SQLite
* **Frontend:** HTML, CSS (Bootstrap optional)
* **Deployment:** Render.com

**Future Enhancements**

🔹 Improve UI using **Bootstrap**.  
🔹 Add **Email-based Voter Registration**.  
🔹 Implement **Two-Factor Authentication**.  
🔹 Store votes securely using **PostgreSQL** instead of SQLite.

**License**

This project is **open-source** under the MIT License.

**Contribution**

Feel free to fork the repo and submit pull requests to improve the system!

**Developed By:** Your Name 🎉