Register no : 927623BAD085

Name : Priyadharshini M

Email : [priyadharshini1375@gmail.com](mailto:priyadharshini1375@gmail.com)

Ph.no : 9025504654

Sites.google.com

Github : https://github.com/PriyadharshiniAD085

Portfolio

Leet code : https://leetcode.com/u/priyavibu/

**Online Voting System**

Product Name : Vote Sphere

Contribution : Full project

**Table Name:**

* User
* Candidate
* Vote

**Keywords:**

Relation - user

Domain – E-governance

Attribute – aadhaar\_id, name, date of birth, email, password

Constrain – Null constrain, Primary Key, Foreign Key, Composite Key

Tuple – one row in a table, Table - a set of tuples

**Normalization:**

Integrity , redendency

Types:

1NF , 2NF, 3NF, 4NF, 5NF

User (Master table)

Candidate

Vote (Transaction table)

1NF – User, There is a column name - user\_id , which is primary key.

2NF – No partial dependency – User.

3NF – No transitive partial dependency.

Eg : Student info – student id , name, job id, job name, native place id, native place.

1NF

Student id can be the primary key . Hence 1NF

2NF

Student role table – student id , job id

Student table – student id, name, native id, native place

Job table – job id, job name

3NF

Student role table – student id, job id

Student table – student id, name, native id

Job table – job id, job name

Native table – native id, native name

**Create :**

CREATE TABLE Users (

aadhaar\_no VARCHAR(12) PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

date\_of\_birth DATE NOT NULL,

is\_verified BOOLEAN DEFAULT 1,

role ENUM('voter', 'admin') DEFAULT 'voter'

);

**Insert :**

INSERT INTO Users (aadhaar\_no, name, email, password\_hash, date\_of\_birth, is\_verified, role)

VALUES

('234567890123', 'Ravi Kumar', 'ravi.kumar@gmail.com', '$2y$10$exampleHash1', '2000-04-15', 1, 'voter'),

('456789012345', 'Sandeep Singh', 'sandeep.singh@gmail.com', '$2y$10$exampleHash3', '1995-02-10', 1, 'admin');

**Delete :**

DELETE FROM Users WHERE aadhaar\_no = '234567890123';

**Joins :**

SELECT

u.name AS voter\_name,

u.aadhaar\_no,

c.name AS candidate\_name,

c.party

FROM

Votes v

JOIN Users u ON v.aadhaar\_no = u.aadhaar\_no

JOIN Candidates c ON v.candidate\_id = c.candidate\_id;

**DTD :**

CDATA – Character Data

PCDATA – Parsed Character Data

**XML :**

<login timeout = “10”></login>

**DTD :**

<!ELEMENT login EMPTY>

<!ATTLIST login timeout CDATA ”0”>

**FRONTEND:**

* HTML
* CSS
* tailwind css
* REACT JS
* FLASK

**BACKEND:**

* MySQL
* ORACLE
* JSON
* MANGODB (noSQL)

**LANGUAGE:**

* PYTHON
* PHP

**FRAMEWORK:**

* XAMPP

**APP DEVELOPER:**

* FRONTEND – android, flutter
* BACKEND – SQL lite, firebase

**FULL STACK AI ENGINEER:**

* TANGO EYE

|  |  |  |
| --- | --- | --- |
| **MODULE NAME** | **SUB MODULE NAME** | **DESCRIPTION** |
| User Module | User Registration & Login,  Voting Process | This module verifies the user’s identity and eligibility through Aadhaar verification and age checks. It allows users to create an account and securely log in for voting. |
| Admin Module | Admin Session,  Election Management,  Voting Monitoring & Tracking,  Results & Reporting | This module provides secure session management for administrators. It allows to control, and monitor elections, manage candidates, and track voter participation. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SUB MODULE NAME** | **FORM NAME** | **DESCRIPTION** | **TABLE NAME** | **TABLE DESCRIPTION** |
| |  | | --- | | Voting Panel | | |  | | --- | | vote.php | | |  | | --- | | Enables eligible users to cast their vote for a candidate. | | |  | | --- | | Votes | | Stores user votes along with election and candidate IDs. |
| |  | | --- | | View Results | | |  | | --- | | public\_results.php | | |  | | --- | | Displays published election results to all users. | | |  | | --- | | Votes, Results |  |  | | --- | |  | | |  | | --- | | Votes stores actual votes, Results stores calculated totals. | |