**FRONTEND - REACTJS**

**REGIONAL ADMIN**

import React, { useState, useEffect } from "react";

import { Table } from "react-bootstrap";

import APISERVICE from "../Services/APISERVICE";

import { useCookies } from "react-cookie";

import { withRouter } from "react-router-dom";

function RegionalAdmin() {

  const [data, setData] = useState([]);

  const [refreshkey, setrefreshkey] = useState(0);

  const [token, setToken, removeToken] = useCookies(["mytoken"]);

  function updatedata(e) {

    const id1 = e.target.id;

    data.map(async (dt) => {

      if (dt.id == id1) {

        dt.R\_admin\_verified = true;

*// console.log(dt.R\_admin\_verified);*

       await APISERVICE.UpdatePassport(id1, dt, token["mytoken"]);

      }

    });

    setrefreshkey((olddata) => olddata + 1);

  }

  useEffect(() => {

    fetch("http://127.0.0.1:8000/api/Passport/", {

      method: "GET",

      headers: {

        "Content-Type": "application/json",

        Authorization: `Token 661622af4055ff002c8867bd3e9ba08b`,

      },

    })

      .then((resp) => resp.json())

      .then((resp) => setData(resp))

      .catch((error) => console.log(error));

  }, [refreshkey]);

  const logoutBtn = () => {

    removeToken(["mytoken"]);

  };

  return (

    <div>

      <div style={{ overflowX: "auto", display: "block", width: "100%" }}>

        <div

          className="title1"

          style={{

            display: "flex",

            justifyContent: "center",

            alignItems: "center",

            height: "70px",

            fontWeight: "bold",

            fontSize: "20px",

            color: "black",

            backgroundColor: "#00fff347",

          }}

        >

          REGIONAL ADMIN

        </div>

        <Table striped bordered hover>

          <thead

            className="thead-dark"

            style={{

              backgroundColor: "blue",

              color: "white",

              fontSize: "14px",

            }}

          >

            <tr>

              <th scope="col">Id</th>

              <th scope="col">Name</th>

              <th scope="col">FatherName</th>

              <th scope="col">DOB</th>

              <th scope="col">Address</th>

              <th scope="col">PermanentAddress</th>

              <th scope="col">PhoneNumber</th>

              <th scope="col">PanNo</th>

              <th scope="col">EmailId</th>

              <th scope="col">VERIFY</th>

            </tr>

          </thead>

          {data.map((passport) => {

            return passport.R\_admin\_verified === false ? (

*//<Table striped bordered hover>*

              <tbody id={passport.id}>

                <tr

                  style={{

                    backgroundColor: "#ff606082",

                    color: "white",

                    fontSize: "14px",

                    fontWeight: "bold",

                  }}

                >

                  <td>{passport.id}</td>

                  <td>{passport.Name}</td>

                  <td>{passport.FatherName}</td>

                  <td>{passport.DateOfBirth}</td>

                  <td>{passport.Address}</td>

                  <td>{passport.PermanentAddress}</td>

                  <td>{passport.PhoneNumber}</td>

                  <td>{passport.PanNo}</td>

                  <td>{passport.EmailId}</td>

                  <td>

                    <button

                      id={passport.id}

                      value={true}

                      type="submit"

                      className="btn btn-success"

                      onClick={(e) => updatedata(e)}

                    >

                      VERIFY

                    </button>

                  </td>

                </tr>

              </tbody>

            ) : null;

          })}

        </Table>

      </div>

    </div>

  );

}

export default withRouter(RegionalAdmin);

**POLICE**

function Police() {

  const [data, setData] = useState([]);

  const [refreshkey, setrefreshkey] = useState(0);

  const [token, setToken, removeToken] = useCookies(["mytoken"]);

  useEffect(() => {

    fetch("http://127.0.0.1:8000/api/Passport/", {

      method: "GET",

      headers: {

        "Content-Type": "application/json",

        Authorization: `Token 661622af4055ff002c8867bd3e9ba08b`,

      },

    })

      .then((resp) => resp.json())

      .then((resp) => setData(resp))

      .catch((error) => console.log(error));

  }, [refreshkey]);

  function updatedata(e) {

    const id1 = e.target.id;

    data.map((dt) => {

      if (dt.id == id1) {

        dt.Police\_verified = true;

        APISERVICE.UpdatePassport(id1, dt, token["mytoken"]);

        setrefreshkey(olddata => olddata+1);

      }

    });

  }

**PASSPORT ADMIN**

return (

    <div>

    <div

        className="title1"

        style={{

          display: "flex",

          justifyContent: "center",

          alignItems: "center",

          height: "70px",

          fontWeight: "bold",

          fontSize: "20px",

          color: "black",

          backgroundColor: "rgba(15, 75, 255, 0.62)",

        }}

      >

        PASSPORT ADMIN

      </div>

      <Table striped bordered hover>

        <thead className="thead-dark"

        style={{

          backgroundColor: "rgba(134, 17, 144, 0.51)",

          color: "white",

          fontSize: "14px",

        }}>

          <tr>

            <th scope="col">Id</th>

            <th scope="col">Name</th>

            <th scope="col">FatherName</th>

            <th scope="col">DOB</th>

            <th scope="col">Address</th>

            <th scope="col">PermanentAddress</th>

            <th scope="col">PhoneNumber</th>

            <th scope="col">PanNo</th>

            <th scope="col">EmailId</th>

            <th scope="col">RegionalAdmin\_Verified</th>

            <th scope="col">Police\_Verified</th>

            <th scope="col">VERIFY</th>

          </tr>

        </thead>

      {data.map((passport) =>{

        return (

          passport.Police\_verified && !passport.Passport\_admin\_verified ?

*//<Table striped bordered hover>*

              <tbody id={passport.id}>

                 {console.log(passport.Police\_verified.toString())}

                <tr style={{

                  backgroundColor: ")",

                  color: "white",

                  fontSize: "14px",

                  fontWeight:"bold"

                }}>

                  <td>{passport.id}</td>

                  <td>{passport.Name}</td>

                  <td>{passport.FatherName}</td>

                  <td>{passport.DateOfBirth}</td>

                  <td>{passport.Address}</td>

                  <td>{passport.PermanentAddress}</td>

                  <td>{passport.PhoneNumber}</td>

                  <td>{passport.PanNo}</td>

                  <td>{passport.EmailId}</td>

                  <td>{passport.R\_admin\_verified.toString()}</td>

                  <td>{passport.Police\_verified.toString()}</td>

                  <td>

                  <button id = {passport.id} value = {true} type="submit"  className="btn btn-success" onClick={e => updatedata(e)}

                    >VERIFY</button>

                  </td>

                </tr>

              </tbody>

           : null

             )}

             )}</Table>

    </div>

  );

}

export default withRouter(PassportAdmin);

**LOGIN**

return (

    <div>

      <div className="login1">

        <div className="container py-5 h-100">

          <div className="row d-flex justify-content-center align-items-center h-100">

            <div className="col-xl-10">

              <div className="card rounded-3 text-black">

                <div className="row g-0">

                  <div className="col-lg-6">

                    <div className="card-body p-md-5 mx-md-4">

                      <div className="text-center">

                        <img

                          src={PASSPORT\_IMAGE}

                          style={{ width: "185px" }}

                          alt="logo"

                        />

                        <h4 className="mt-1 mb-5 pb-1">PAS LOGIN</h4>

                      </div>

                      <form>

                        <div className="form-outline mb-4">

                          <label

                            className="form-label"

                            forname="form2Example11"

                          >

                            Username

                          </label>

                          <input

                            type="text"

                            id="form2Example11"

                            className="form-control"

                            placeholder="Username"

                            value={username}

                            onChange={(e) => setUsername(e.target.value)}

                          />

                        </div>

                        <div className="form-outline mb-4">

                          <label

                            className="form-label"

                            forname="form2Example22"

                          >

                            Password

                          </label>

                          <input

                            type="password"

                            id="form2Example22"

                            className="form-control"

                            placeholder="Password"

                            value={password}

                            onChange={(e) => setPassword(e.target.value)}

                          />

                        </div>

                        <div className="text-center pt-1 mb-5 pb-1">

                          <button

                            className="btn btn-primary btn-block fa-lg gradient-custom-2 mb-3"

                            type="button"

                            onClick={handleLogin}

                          >

                            Log in

                          </button>

                        </div>

                        <div className="d-flex align-items-center justify-content-center pb-4">

                          <p className="mb-0 me-2">Don't have an account?</p>

                          <button

                            type="button"

                            className="btn btn-outline-danger"

                            onClick={handleRegister}

                          >

                            REGISTER

                          </button>

                        </div>

                      </form>

                    </div>

                  </div>

                  <div className="col-lg-6 d-flex align-items-center gradient-custom-2">

                    <div className="text-white px-3 py-4 p-md-5 mx-md-4">

                      <h3 className="mb-4 text">PASSPORT AUTOMATION SYSTEM</h3>

                      <h6 className="mb-4 text1">-By ROCKSTARS Group</h6>

                      <p className="small mb-0">

                        Automation is no longer just a problem for those working

                        in manufacturing. Physical labor was replaced by robots;

                        mental labor is going to be replaced by AI and software.

                      </p>

                    </div>

                  </div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

      <ParticlesBg type="cobweb" num={200} bg={true} />

    </div>

  );

}

export default Login1;

**BACKEND – DJANGO**

**VIEWS.PY**

from django.shortcuts import render,HttpResponse,get\_object\_or\_404

from .models import PassportDatabase

from .serializers import PassportSerializer,UserSerializer

from django.http import JsonResponse

from rest\_framework import status,generics,mixins,viewsets

from rest\_framework.parsers import JSONParser

from rest\_framework.decorators import api\_view,APIView,permission\_classes

from rest\_framework.response import Response

from rest\_framework.authentication import TokenAuthentication

from rest\_framework.permissions import IsAuthenticated,IsAuthenticatedOrReadOnly,SAFE\_METHODS,BasePermission

from django.contrib.auth.models import User

class PassportViewSet(viewsets.ModelViewSet):

    queryset = PassportDatabase.objects.all()

    serializer\_class = PassportSerializer

class UserViewSet(viewsets.ModelViewSet):

    queryset = User.objects.all()

    serializer\_class = UserSerializer

**MODELS.PY**

from django.db import models

import datetime

*# Create your models here.*

class PassportDatabase(models.Model):

    Name                    = models.CharField(max\_length=50)

    FatherName              = models.CharField(max\_length=50)

    DateOfBirth             = models.DateField()

    Address                 = models.CharField(max\_length=100)

    PermanentAddress        = models.CharField(max\_length=100)

    PhoneNumber             = models.IntegerField()

    PanNo                   = models.IntegerField()

    EmailId                 = models.EmailField(max\_length=30)

    ApplicationNo           = models.IntegerField(unique=True)

    Username                = models.CharField(max\_length=50)

    Password                = models.CharField(max\_length=15)

    R\_admin\_verified        = models.BooleanField(default=False)

    Police\_verified         = models.BooleanField(default=False)

    Passport\_admin\_verified = models.BooleanField(default=False)

    def \_\_str\_\_(self):

        return *self*.Name

    class Meta:

        ordering = ('ApplicationNo',)

**URLS.PY**

from django.contrib import admin

from django.urls import path,include

from .views import PassportViewSet,UserViewSet

from rest\_framework.routers import DefaultRouter

router = DefaultRouter()

router.register('Passport',PassportViewSet, basename='Passport')

router.register('users',UserViewSet)

urlpatterns = [

    path('api/',include(router.urls)),

]

**SERIALIZERS.PY**

from rest\_framework import serializers

from .models import PassportDatabase

from django.contrib.auth.models import User

from rest\_framework.authtoken.views import Token

*#MODEL SERIALZIER*

class PassportSerializer(serializers.ModelSerializer):

    class Meta:

        model = PassportDatabase

        fields =['id','Name','FatherName','DateOfBirth','Address','PermanentAddress','PhoneNumber','PanNo','EmailId','ApplicationNo','Username','Password','R\_admin\_verified','Police\_verified','Passport\_admin\_verified']

class UserSerializer(serializers.ModelSerializer):

    class Meta:

        model = User

        fields = ['id','username','password']

        extra\_kwargs = {'password':{

            'write\_only':True,

            'required':True }}

    def create(self,validated\_data):

        user = User.objects.create\_user(\*\*validated\_data)

        Token.objects.create(user=user)

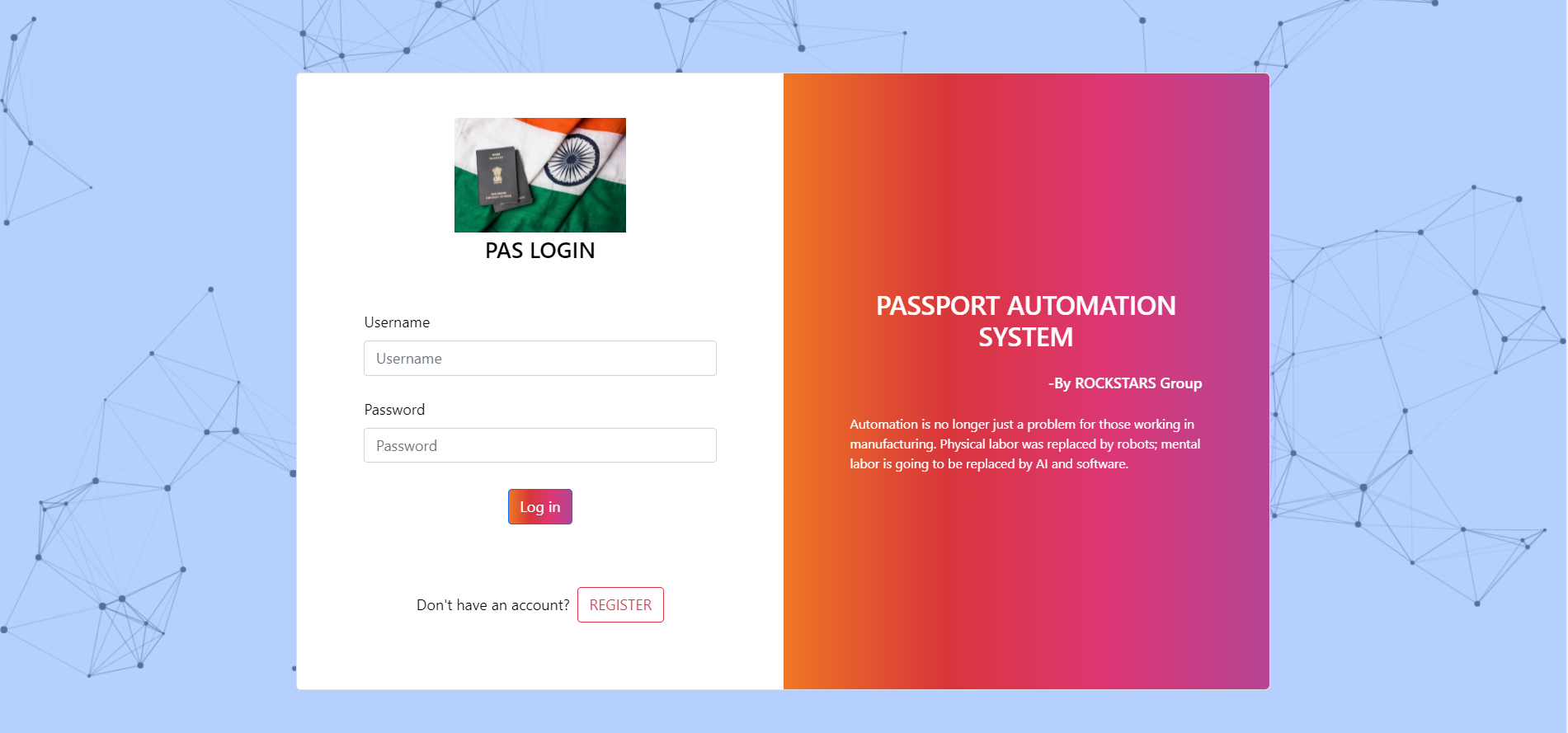
        return user

**SCREENSHOTS OF PROJECT**

**HOMEPAGE**



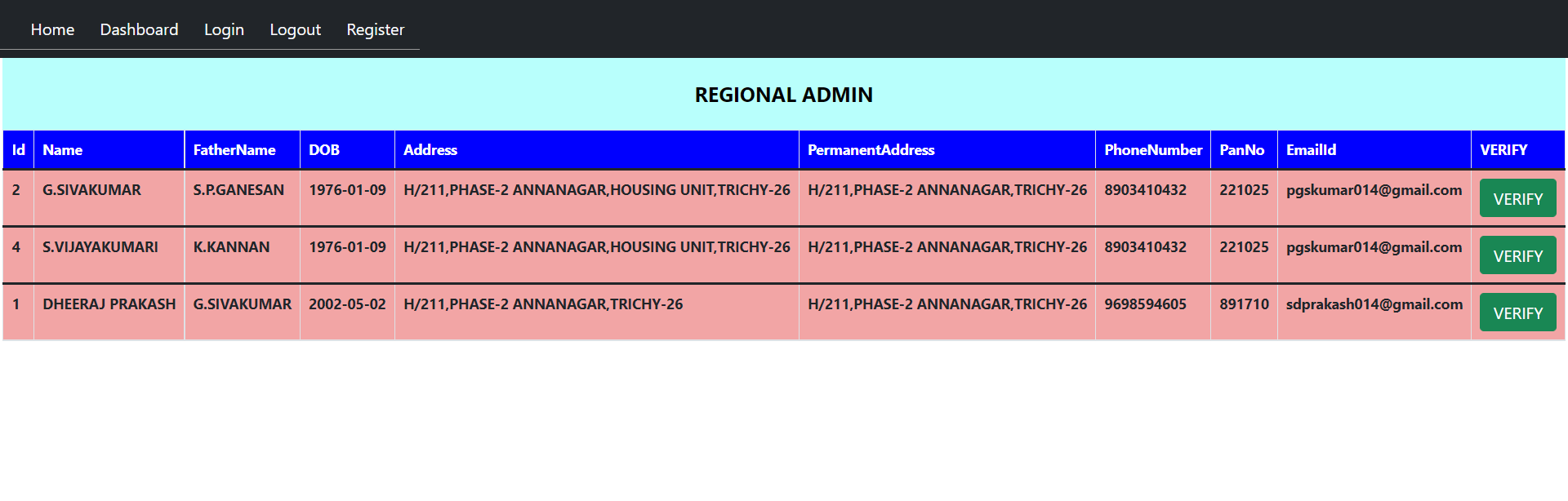
**LOGIN PAGE**



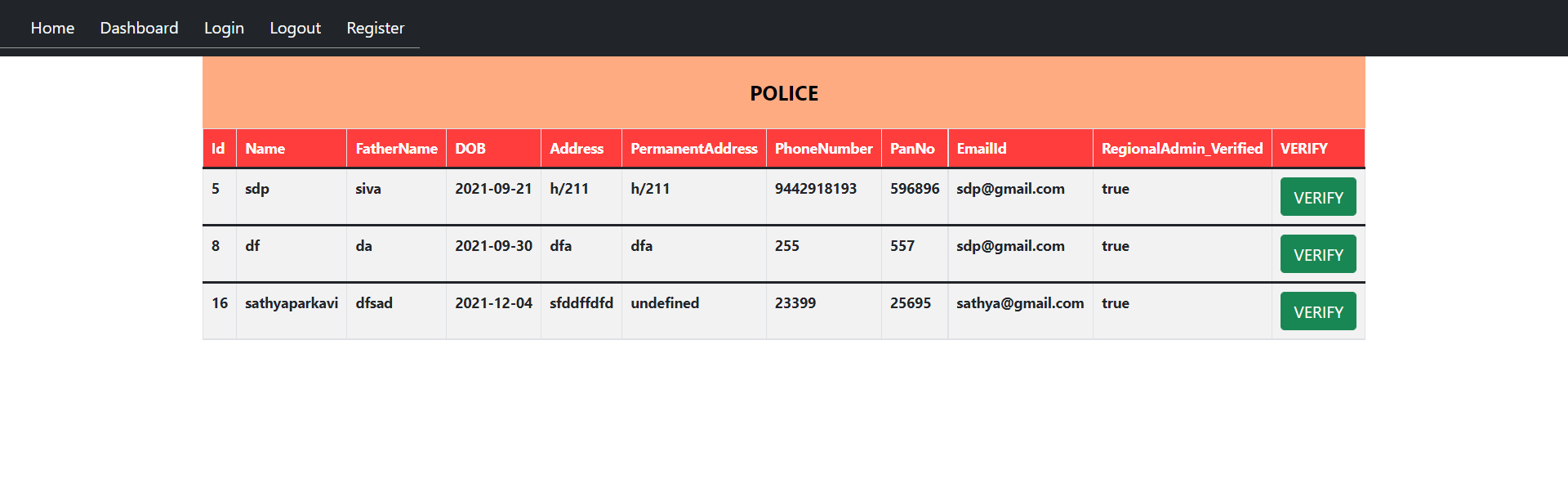
**REGISTER PAGE**



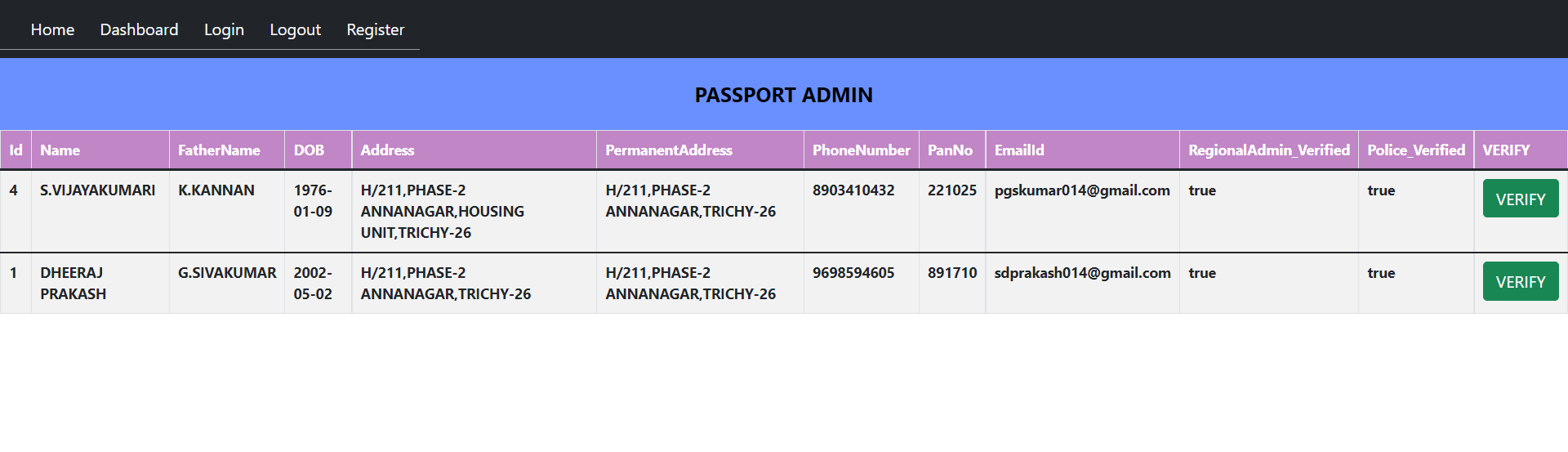
**REGIONAL ADMIN**



**POLICE**



**PASSPORT ADMIN**



**ISSUING AUTHORITY**

