Admin Data:  
When you run app.py first time, the app will auto-create a default admin:

* Email: admin@example.com
* Password: Admin@1234
* Username: system.admin.sou.min

To add and activate virtual environment:  
python -m venv .venv

.venv\Scripts\activate

To run the app:  
$env:FLASK\_APP = "app.py"

flask run

Orginal app.py:

from flask import Flask, render\_template, request, redirect, url\_for, flash, session

import config

from utils.data\_loader import ensure\_data\_files, load\_users

from utils.auth import create\_user, authenticate, find\_user\_by\_email, generate\_reset\_token, verify\_reset\_token, reset\_password, requires\_role, unlock\_account

from utils.viz import load\_minerals\_json, generate\_mineral\_chart, generate\_overview\_chart

from datetime import timedelta

import os

app = Flask(\_\_name\_\_)

app.secret\_key = config.SECRET\_KEY

app.permanent\_session\_lifetime = config.PERMANENT\_SESSION\_LIFETIME

# ensure data files exist

ensure\_data\_files()

# Create initial admin if none exists (convenience for first run)

def create\_initial\_admin():

    users = load\_users().get("users", [])

    if not any(u["role"] == "Administrator" for u in users):

        # WARNING: in production, require env-admin or manual setup

        admin = create\_user("System", "Admin", "admin@example.com", "SouthAfrica", "MINN", "Administrator", "Admin@1234")

        print("Created default admin user:")

        print(" username:", admin["username"])

        print(" email: admin@example.com")

        print(" password: Admin@1234")

create\_initial\_admin()

# ----- Routes -----

@app.route("/")

def home():

    if "username" in session:

        role = session.get("role")

        if role == "Administrator":

            return redirect(url\_for("dashboard\_admin"))

        elif role == "Investor":

            return redirect(url\_for("dashboard\_investor"))

        else:

            return redirect(url\_for("dashboard\_researcher"))

    return redirect(url\_for("login"))

@app.route("/login", methods=["GET", "POST"])

def login():

    if request.method == "POST":

        username = request.form.get("username", "").strip()

        password = request.form.get("password", "")

        success, message = authenticate(username, password, remote\_ip=request.remote\_addr)

        if success:

            flash("Login successful.", "success")

            return redirect(url\_for("home"))

        else:

            flash(message, "danger")

    return render\_template("login.html")

@app.route("/logout")

def logout():

    session.clear()

    flash("Logged out.", "info")

    return redirect(url\_for("login"))

@app.route("/signup", methods=["GET", "POST"])

def signup():

    """

    Signup endpoint. You may restrict signup in production.

    Administrators should be created only by admin accounts ideally.

    """

    if request.method == "POST":

        first\_name = request.form.get("first\_name", "").strip()

        last\_name = request.form.get("last\_name", "").strip()

        email = request.form.get("email", "").strip()

        country = request.form.get("country", "").strip()

        org = request.form.get("organization", "").strip()

        role = request.form.get("role", "Researcher")

        password = request.form.get("password", "")

        confirm = request.form.get("confirm", "")

        # basic validations

        if not (first\_name and last\_name and email and country and password):

            flash("Please fill in all required fields.", "danger")

            return render\_template("signup.html")

        if password != confirm:

            flash("Passwords do not match.", "danger")

            return render\_template("signup.html")

        if find\_user\_by\_email(email):

            flash("An account with that email already exists.", "danger")

            return render\_template("signup.html")

        # prevent creating Administrator via public signup

        if role == "Administrator":

            flash("Administrator accounts must be created by an existing admin.", "danger")

            return render\_template("signup.html")

        new\_user = create\_user(first\_name, last\_name, email, country, org, role, password)

        flash(f"Account created. Your username is {new\_user['username']}. Please login.", "success")

        return redirect(url\_for("login"))

    return render\_template("signup.html")

# ----- Dashboard routes per role -----

@app.route("/dashboard/admin")

@requires\_role("Administrator")

def dashboard\_admin():

    users = load\_users().get("users", [])

    role\_counts = {"Administrator": 0, "Investor": 0, "Researcher": 0}

    for u in users:

        role = u.get("role")

        if role in role\_counts:

            role\_counts[role] += 1

    return render\_template("dashboard\_admin.html",

                           users=users,

                           role\_counts=role\_counts,

                           form=None)

@app.route("/dashboard/investor")

@requires\_role("Investor")

def dashboard\_investor():

    return render\_template("dashboard\_investor.html")

from utils.viz import load\_minerals\_json

@app.route("/dashboard/researcher")

@requires\_role("Researcher")

def dashboard\_researcher():

    minerals = load\_minerals\_json()

    return render\_template("dashboard\_researcher.html", minerals=minerals)

# ----- Password reset -----

@app.route("/reset-request", methods=["GET", "POST"])

def reset\_request():

    if request.method == "POST":

        email = request.form.get("email", "").strip()

        user = find\_user\_by\_email(email)

        if not user:

            flash("If an account exists with that email, a link will be sent.", "info")

            return redirect(url\_for("login"))

        token = generate\_reset\_token(email)

        # In production: send token via email. For this course: show link on screen (simulated)

        reset\_link = url\_for("reset\_password", token=token, \_external=True)

        flash("Password reset link generated (for demo it's shown below). In production send by email.", "info")

        return render\_template("reset\_request.html", reset\_link=reset\_link)

    return render\_template("reset\_request.html")

@app.route("/reset/<token>", methods=["GET", "POST"])

def reset\_password\_route(token):

    email = verify\_reset\_token(token)

    if not email:

        flash("Invalid or expired token.", "danger")

        return redirect(url\_for("reset\_request"))

    if request.method == "POST":

        pw = request.form.get("password", "")

        confirm = request.form.get("confirm", "")

        if pw != confirm:

            flash("Passwords do not match.", "danger")

            return render\_template("reset\_password.html", token=token)

        # call the imported util correctly now

        reset\_password(email, pw)

        flash("Password has been reset. Please login.", "success")

        return redirect(url\_for("login"))

    return render\_template("reset\_password.html", token=token)

# ----- Admin unlock account action -----

@app.route("/admin/unlock/<user\_id>", methods=["POST"])

@requires\_role("Administrator")

def admin\_unlock(user\_id):

    ok = unlock\_account(user\_id)

    if ok:

        flash("Account unlocked.", "success")

    else:

        flash("Could not find user.", "danger")

    return redirect(url\_for("dashboard\_admin"))

@app.route("/minerals")

def minerals\_dashboard():

    """Display list of minerals with overview chart."""

    minerals = load\_minerals\_json()

    chart\_html = generate\_overview\_chart()

    return render\_template("minerals\_dashboard.html", minerals=minerals, chart\_html=chart\_html)

@app.route("/minerals/<mineral\_name>")

def mineral\_detail\_chart(mineral\_name):

    """Display individual mineral chart."""

    chart\_html = generate\_mineral\_chart(mineral\_name)

    return render\_template("minerals\_dashboard.html", chart\_html=chart\_html, single=True, mineral\_name=mineral\_name)

# Run

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

For admin html

{% extends "base.html" %}

{% block content %}

<h3 class="text-primary mb-4">Administrator Dashboard</h3>

<!-- ===== Summary Cards ===== -->

<div class="row text-center">

  <!-- Total Users -->

  <div class="col-md-4 mb-3">

    <div class="card shadow-sm border-0 bg-light">

      <div class="card-body">

        <h6>Total Users</h6>

        <h3 class="fw-bold text-primary">{{ users | length }}</h3>

      </div>

    </div>

  </div>

  <!-- Administrators -->

  <div class="col-md-4 mb-3">

    <div class="card shadow-sm border-0 bg-light">

      <div class="card-body">

        <h6>Administrators</h6>

        <h3 class="fw-bold text-success">

          {{ role\_counts.Administrator }}

        </h3>

      </div>

    </div>

  </div>

  <!-- Locked Accounts -->

  <div class="col-md-4 mb-3">

    <div class="card shadow-sm border-0 bg-light">

      <div class="card-body">

        <h6>Locked Accounts</h6>

        <h3 class="fw-bold text-danger">

          {{ users | selectattr('locked\_until', 'ne', None) | list | length }}

        </h3>

      </div>

    </div>

  </div>

</div>

<!-- ===== Role Distribution Chart ===== -->

<div id="rolesChart" class="my-4"></div>

<!-- Store Jinja data as JSON so VS Code sees valid JS -->

<script id="role-data" type="application/json">

  {{ role\_counts | tojson }}

</script>

<script>

  document.addEventListener("DOMContentLoaded", function () {

    // Retrieve and parse role data JSON rendered by Jinja

    const roleData = JSON.parse(

      document.getElementById("role-data").textContent

    );

    const data = [{

      values: [

        roleData.Administrator,

        roleData.Investor,

        roleData.Researcher

      ],

      labels: ['Administrator', 'Investor', 'Researcher'],

      type: 'pie',

      marker: { colors: ['#007bff', '#28a745', '#ffc107'] }

    }];

    const layout = {

      title: 'User Role Distribution',

      paper\_bgcolor: 'rgba(0,0,0,0)',

      plot\_bgcolor: 'rgba(0,0,0,0)',

      font: { family: 'Arial, sans-serif', size: 14 }

    };

    if (typeof Plotly !== "undefined") {

      Plotly.newPlot("rolesChart", data, layout, { responsive: true });

    } else {

      console.error("Plotly is not loaded. Please include plotly.js in base.html");

    }

  });

</script>

<!-- ===== User Management Table ===== -->

<h5 class="mt-5">User Management</h5>

<div class="table-responsive">

  <table class="table table-striped table-sm align-middle">

    <caption class="text-muted">

      List of all registered users and their account status.

    </caption>

    <thead class="table-primary">

      <tr>

        <th scope="col">User ID</th>

        <th scope="col">Username</th>

        <th scope="col">Email</th>

        <th scope="col">Role</th>

        <th scope="col">Locked</th>

        <th scope="col">Action</th>

      </tr>

    </thead>

    <tbody>

      {% for u in users %}

      <tr>

        <td>{{ u.user\_id }}</td>

        <td>{{ u.username }}</td>

        <td>{{ u.email }}</td>

        <td>{{ u.role }}</td>

        <td>{{ u.locked\_until or "No" }}</td>

        <td>

          {% if u.locked\_until %}

          <form method="post" action="{{ url\_for('admin\_unlock', user\_id=u.user\_id) }}">

            {% if form and form.csrf\_token %}

              {{ form.csrf\_token }}

            {% endif %}

            <button class="btn btn-sm btn-warning">Unlock</button>

          </form>

          {% else %}

          <small class="text-muted">Active</small>

          {% endif %}

        </td>

      </tr>

      {% endfor %}

    </tbody>

  </table>

</div>

{% endblock %}

For investor:  
  
{% extends "base.html" %}

{% block content %}

<h3 class="text-success mb-4">Investor Dashboard</h3>

<div class="row mb-4 text-center">

  <div class="col-md-4 mb-3">

    <div class="card shadow-sm border-0 bg-light">

      <div class="card-body">

        <h6>Total Countries Tracked</h6>

        <h3 class="fw-bold text-success">9</h3>

      </div>

    </div>

  </div>

  <div class="col-md-4 mb-3">

    <div class="card shadow-sm border-0 bg-light">

      <div class="card-body">

        <h6>Top Producing Mineral</h6>

        <h3 class="fw-bold text-primary">Cobalt (DRC)</h3>

      </div>

    </div>

  </div>

  <div class="col-md-4 mb-3">

    <div class="card shadow-sm border-0 bg-light">

      <div class="card-body">

        <h6>Avg GDP (2024)</h6>

        <h3 class="fw-bold text-warning">$72 Bn</h3>

      </div>

    </div>

  </div>

</div>

<div class="card shadow-sm mb-4">

  <div class="card-body">

    <h5 class="card-title">Mineral Production Overview</h5>

    <div id="inv\_chart"></div>

  </div>

</div>

<script>

  const data = [{

    x: ['Cobalt (DRC)', 'Lithium (Zimbabwe)', 'Graphite (Mozambique)', 'Copper (Zambia)', 'Manganese (SA)'],

    y: [220000, 12000, 170000, 720000, 7400000],

    type: 'bar',

    marker: {color: ['#0d6efd','#6610f2','#20c997','#198754','#fd7e14']}

  }];

  const layout = {title: 'Top Mineral Production (2024)', xaxis:{title:'Mineral'}, yaxis:{title:'Tonnes'}};

  Plotly.newPlot('inv\_chart', data, layout);

</script>

{% endblock %}

For base html:

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>MINN — African Critical Minerals App</title>

  <meta name="viewport" content="width=device-width,initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-primary">

  <div class="container-fluid">

    <a class="navbar-brand" href="{{ url\_for('home') }}">

      <strong>MINN</strong> Critical Minerals

    </a>

    <div class="collapse navbar-collapse">

      <ul class="navbar-nav ms-auto">

        {% if session.username %}

          <li class="nav-item"><a class="nav-link" href="#">{{ session.username }}</a></li>

          <li class="nav-item"><a class="nav-link" href="{{ url\_for('logout') }}">Logout</a></li>

        {% else %}

          <li class="nav-item"><a class="nav-link" href="{{ url\_for('login') }}">Login</a></li>

          <li class="nav-item"><a class="nav-link" href="{{ url\_for('signup') }}">Sign up</a></li>

        {% endif %}

      </ul>

    </div>

  </div>

</nav>

<div class="container mt-4">

  {% with messages = get\_flashed\_messages(with\_categories=true) %}

    {% if messages %}

      {% for cat, msg in messages %}

        <div class="alert alert-{{ cat }} alert-dismissible fade show" role="alert">

          {{ msg }}

          <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

        </div>

      {% endfor %}

    {% endif %}

  {% endwith %}

  {% block content %}{% endblock %}

</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

<script src="{{ url\_for('static', filename='js/password\_strength.js') }}"></script>

</body>

</html>

Integrared base.html:

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>MINN — African Critical Minerals App</title>

<meta name="viewport" content="width=device-width,initial-scale=1">

<!-- Bootstrap & custom styles -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

<!-- Plotly for interactive charts -->

<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg navbar-dark bg-primary shadow-sm">

<div class="container-fluid">

<a class="navbar-brand fw-bold" href="{{ url\_for('home') }}">

<strong>MINN</strong> Critical Minerals

</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"

aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav ms-auto">

{% if session.username %}

<li class="nav-item">

<a class="nav-link" href="#">

👤 {{ session.username }} ({{ session.role }})

</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('logout') }}">🚪 Logout</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('login') }}">Login</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{{ url\_for('signup') }}">Sign up</a>

</li>

{% endif %}

</ul>

</div>

</div>

</nav>

<!-- Main container -->

<div class="container my-4">

{% with messages = get\_flashed\_messages(with\_categories=true) %}

{% if messages %}

{% for cat, msg in messages %}

<div class="alert alert-{{ cat }} alert-dismissible fade show" role="alert">

{{ msg }}

<button type="button" class="btn-close" data-bs-dismiss="alert"></button>

</div>

{% endfor %}

{% endif %}

{% endwith %}

{% block content %}{% endblock %}

</div>

<!-- Scripts -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

<script src="{{ url\_for('static', filename='js/password\_strength.js') }}"></script>

</body>

</html>

Current Base.html:

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>MINN — African Critical Minerals App</title>

  <meta name="viewport" content="width=device-width,initial-scale=1">

  <!-- Bootstrap & custom styles -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

  <!-- Plotly for interactive charts -->

  <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg navbar-dark bg-primary shadow-sm">

  <div class="container-fluid">

    <a class="navbar-brand fw-bold" href="{{ url\_for('home') }}">

      <strong>MINN</strong> Critical Minerals

    </a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"

      aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNav">

      <ul class="navbar-nav ms-auto">

        {% if session.username %}

          <li class="nav-item">

            <a class="nav-link" href="#">

              👤 {{ session.username }} ({{ session.role }})

            </a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="{{ url\_for('logout') }}">🚪 Logout</a>

          </li>

        {% else %}

          <li class="nav-item">

            <a class="nav-link" href="{{ url\_for('login') }}">Login</a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="{{ url\_for('signup') }}">Sign up</a>

          </li>

        {% endif %}

      </ul>

    </div>

  </div>

</nav>

<!-- Main container -->

<div class="container my-4">

  {% with messages = get\_flashed\_messages(with\_categories=true) %}

    {% if messages %}

      {% for cat, msg in messages %}

        <div class="alert alert-{{ cat }} alert-dismissible fade show" role="alert">

          {{ msg }}

          <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

        </div>

      {% endfor %}

    {% endif %}

  {% endwith %}

  {% block content %}{% endblock %}

</div>

<!-- Scripts -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

<script src="{{ url\_for('static', filename='js/password\_strength.js') }}"></script>

<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

</body>

</html>

…………………………

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>MINN — African Critical Minerals App</title>

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap & custom styles -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <link rel="stylesheet" href="{{ url\_for('static', filename='css/styles.css') }}">

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg navbar-dark bg-primary shadow-sm">

  <div class="container-fluid">

    <a class="navbar-brand fw-bold" href="{{ url\_for('home') }}">

      <strong>MINN</strong> Critical Minerals

    </a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"

      aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNav">

      <ul class="navbar-nav ms-auto">

        {% if session.get('username') %}

          <li class="nav-item">

            <a class="nav-link" href="#">

              👤 {{ session.get('username') }} ({{ session.get('role') }})

            </a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="{{ url\_for('logout') }}">🚪 Logout</a>

          </li>

        {% else %}

          <li class="nav-item">

            <a class="nav-link" href="{{ url\_for('login') }}">Login</a>

          </li>

          <li class="nav-item">

            <a class="nav-link" href="{{ url\_for('signup') }}">Sign up</a>

          </li>

        {% endif %}

      </ul>

    </div>

  </div>

</nav>

<!-- Main container -->

<div class="container my-4">

  {% with messages = get\_flashed\_messages(with\_categories=true) %}

    {% if messages %}

      {% for cat, msg in messages %}

        <div class="alert alert-{{ cat }} alert-dismissible fade show" role="alert">

          {{ msg }}

          <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

        </div>

      {% endfor %}

    {% endif %}

  {% endwith %}

  {% block content %}{% endblock %}

</div>

<!-- Scripts -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

<script src="{{ url\_for('static', filename='js/password\_strength.js') }}"></script>

</body>

</html>