# Cedrus - Audit & Compliance Platform

Cedrus is a Django-based platform for managing audits, compliance, and risk. It serves as a small, open-source alternative to commercial certification body portals (like DNV, LRQA, TÜV, Bureau Veritas).

## Tech Stack

- **Backend**: Django 5 (monolith architecture)
- **Database**: SQLite (development)
- **Templates**: Django templates (no React/SPA)
- **Frontend**: HTML + CSS with Bootstrap 5
- **JavaScript**: Vanilla JS only where needed
- **Authentication**: Django User + Group + Permissions

## Project Structure

- `accounts/` - User profiles, roles, permissions, organization membership
- `core/` - Organizations, sites, standards, certifications (the foundation)
- `audits/` - Audits, findings, recommendations, appendices

## Setup Instructions

### 1. Create and Activate Virtual Environment

```
# Create virtual environment
python3 -m venv venv

# Activate virtual environment
# On macOS/Linux:
source venv/bin/activate
# On Windows:
venv\Scripts\activate
```

### 2. Install Dependencies

```
pip install -r requirements.txt
```

### 3. Run Migrations

```
python manage.py makemigrations
python manage.py migrate
```

### 4. Create Superuser

```
python manage.py createsuperuser
```

### 5. Seed Initial Data (Optional)

```
python manage.py seed_data
```

This will create:

- User groups (cb_admin, lead_auditor, auditor, client_admin, client_user)
- Sample users for each role
- Sample organization, site, standard, and certification
- Sample audit in "draft" status

### 6. Run Development Server

```
python manage.py runserver
```

Visit `http://127.0.0.1:8000/` in your browser.

## User Roles

The platform uses Django Groups for role-based access:

- **cb_admin**: Certification Body administrators - can manage organizations, certifications, and all audits
- **lead_auditor**: Lead auditors - can manage assigned audits and add findings
- **auditor**: Regular auditors - can view and contribute to assigned audits
- **client_admin**: Client organization administrators - can view their organization's audits and respond to nonconformities
- **client_user**: Regular client users - can view their organization's audits and respond to nonconformities

## Default Login Credentials (After Seeding)

After running the seed script, you can log in with:

- **CB Admin**: `cbadmin` / `password123`
- **Lead Auditor**: `auditor1` / `password123`
- **Client Admin**: `clientadmin` / `password123`

(All passwords are `password123` - change them in production!)

## Key Features

### For CB Admins

- Manage organizations, sites, standards, and certifications
- Create audits and assign lead auditors
- View all audits and make final decisions
- Edit audit recommendations

### For Auditors

- View assigned audits
- Edit audit details (if lead auditor)
- Add findings (nonconformities, observations, opportunities for improvement)
- Update audit status through workflow

### For Clients

- View audits for their organization
- Respond to nonconformities (root cause, correction, corrective action)
- Upload evidence files
- View audit reports

## Development Notes

- The codebase favors clarity over cleverness
- Models are well-documented with docstrings
- Views use class-based views where appropriate
- Templates use Bootstrap 5 for styling
- All models are registered in Django admin with sensible configurations

## Future Enhancements

Planned additions (not yet implemented):

- `risks/` app for risk management
- `internal_audits/` app for internal audit tracking
- Enhanced file upload and management
- Email notifications
- PDF report generation
- API endpoints

## License

[Add your license here]

## Contributing

[Add contribution guidelines here]