COMPASS
by Nisria.inc

Compass by Nisria.Inc — Technical Documentation

Audience: Software Engineers
Goal: Translate Compass's vision into a scalable, secure, and modular digital system

1. System Overview

Compass is a centralized platform designed to streamline data, documents, and workflows for Nisria's community programs and grant processes. It supports internal operations, funding initiatives, and long-term organizational memory.

2. System Architecture Summary

- **Frontend:** React.js with Tailwind CSS
- **Backend:** RESTful API using Django REST Framework
- **Database:** PostgreSQL (relational)
- **Storage:** AWS S3 or Firebase for secure document storage, Cloudinary for Videos and Images
- **Authentication:** Token-based (JWT) + Role-Based Access Control (RBAC)

- **Notifications:** Email & in-app alerts using a scheduler (e.g., Celery)
- **Hosting:** AWS, Heroku, or DigitalOcean with Git-based CI/CD

3. Core Modules & Functional Flow

3.1. Program Management

Each community program (Education, Rescue, Microfund, etc.) will have a **dedicated dashboard** showing performance and funding metrics.

Main Features:

- Track enrollment, participants, and progress
- Store and edit data like start dates, budgets, and participant names
- Enable filtering and timeline-based views

Example Endpoints:

- GET /api/programs – List all programs
- GET /api/programs/{id} – Get program details
- POST /api/programs – Create a new program
- PUT /api/programs/{id} – Update a program
- DELETE /api/programs/{id} – Remove a program (admin-only)

3.2. Grants & Funding Management

The grant module handles all funding-related workflows and archives, making applications faster and more organized.

3.2.1. Grant Archive

- Store bank statements, certifications, pitch decks, reports, and templates
- Tag files by program, year, and status

3.2.2. Grant Workflow

- Match opportunities to relevant programs
- Use status labels: **Pending**, **Applied**, **Approved $ Denied**
- Auto-sort pending grants by closest deadline
- Notifications for due applications

Example Endpoints:

- GET /api/grants – List all grants
- POST /api/grants – Add new opportunity
- PATCH /api/grants/{id}/status – Update status (e.g., to "Applied")
- GET /api/documents/grants – List or search grant-related documents
- POST /api/documents/grants/upload – Upload a file (with metadata)

3.3. Document Management

- Centralized file storage categorized by program and document type
- Metadata: type, owner, upload date, linked programs
- Auto-archive logic for files older than X months
- Optional: Soft delete with undo function (like trash bin)

Endpoints:

- GET /api/documents – Browse/search documents

- POST /api/documents/upload – Upload file
- DELETE /api/documents/{id} – Soft-delete a file
- POST /api/documents/{id}/restore – Restore a deleted file

3.4. User & Admin Roles

Each team member has a defined role. The system enforces access permissions based on these roles:

Role	Access Level
Super Admin	Full control: user & document management
Admin	Edit/view all programs & upload documents
Management Leads	Oversee grants & data accuracy
Grant Officers	Upload docs, manage grant statuses

Endpoints:

- GET /api/users – View user list (super admin)
- POST /api/users – Add user with assigned role
- PUT /api/users/{id}/role – Change access level
- GET /api/profile – Logged-in user's permissions

4. Notifications & Alerts

- Email or in-app alerts for upcoming grant deadlines
- System reminders for document expiry or missing uploads

Trigger Examples:

- Every morning, check grants due within 7 days → Notify assigned officers
- Alert admins if a required program field is missing data

5. Analytics & Dashboards

- Display summaries like:
  - Total students enrolled
  - Grants applied this year
  - Monthly budget allocation per program
- Interactive charts & tables (filter by year, program, or status)

Endpoints:

- GET /api/analytics/overview
- GET /api/analytics/programs/{id}
- GET /api/analytics/grants

6. Security & Access Control

- Token-based login (JWT)
- Role-based permissions for endpoints
- Encrypted file uploads
- Audit logs for actions like uploads, edits, deletions

Security endpoints:

- POST /api/login
- POST /api/logout
- POST /api/password-reset
- GET /api/logs/user-activity

7. System Maintenance

- Daily backups of database and storage
- Admin-only dashboard for system logs and backups
- Automatic archiving of unused documents

8. Scalability Considerations

Phase 1: MVP

- All dashboards
- Manual document upload
- Manual grant input

Phase 2: Automation

- Grant matching using keywords and program needs
- Integration with grant opportunity databases
- Offline sync for field workers

Compass will grow to be an overall solution to management within NGO`s