WalsoftAI-Genealogy: MVP Implementation Blueprint

Project Name: WalsoftAI-Genealogy
Author: Philip Tambiti Leo Walekhwa
Date: June 2025
Version: v1.0 (Unified, Updated)

---

1. Purpose

This document consolidates the core features, architecture, and updated technical decisions for the MVP (Minimum Viable Product) of WalsoftAI-Genealogy, a Django-based ancestry platform focused on preserving Luhya (Bakhabi) cultural heritage with integrated AI storytelling.

It replaces and integrates the content of:

- `doc_001.pdf` — Cultural and technical project overview
- `doc_002_achitecture.pdf` — Architecture and model layout

---

2. MVP Goals

- Enable lineage recording through modular apps
- Support polygamy, remarriage, and dual-parent tracking
- Generate AI-assisted cultural narratives
- Provide a role-based admin dashboard for secure collaboration

---

3. Core Features (MVP Scope)

Feature	Description
Person Management	Add/edit individuals, photos, aliases, origins, and relationships
Marriage & Relationships	Handle polygamy, widow remarriage, stepchildren, and dual parent links
Event Tracking	Birth, death, migration, education, marriage, etc.
Narrative Generation	Use local AI models to generate and store biographies in `Person.story`
Admin Dashboard	Superuser and contributor access, model editing
Version History	Track all changes via `django-simple-history`

---

4. Technical Foundations

4.1 Django App Structure

- `users/` – CustomUser with roles
- `people/` – Core Person model
- `relationships/` – Parent-child and marriage links
- `events/` – Life milestones
- `narratives/` – AI story generation & editing
- `documents/` – Oral records, photos, PDFs
- `admin_tools/` – Merge and data integrity

4.2 Environment Setup

- `.env` used for local credentials
- Three-tier settings: `base.py`, `dev.py`, `prod.py`
- `DJANGO_ENV` toggles environment

---

5. Models & Structure

5.1 UUIDs as Primary Keys

All core models use:

```
id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False)
```

5.2 Person Model Enhancements

- `aliases`: JSONField for alternative names
- `cultural_notes`: JSONField for clan, idioms, totems, etc.
- `story`: TextField for AI-generated or user-edited narrative
- `photo`: ImageField
- `father`, `mother`: Self-referencing FKs
- Version tracking: `HistoricalRecords()`

5.3 Narrative Integration

- Local model: `llama-cpp-python` with Phi-3 or TinyLLaMA

- Prompts use `jinja2`
- Output stored in `Person.story`
- Narrative templates reside in: `narratives/prompts/`

---

6. Infrastructure Decisions

Component	Stack/Tool	Reason
DB Engine	PostgreSQL 17.5	JSONField support, reliability
AI Runtime	llama-cpp-python	Offline narrative generation
Frontend Admin	Django Admin	Simplicity for internal use
Dev Environment	Windows 11 + VSCode	Familiar development tools
Deployment OS	Ubuntu 24.04	Stability, server-grade support

---

7. Post-MVP Ready Enhancements (Already Structured for)

- Role-based dashboards: `Admin`, `Elder`, `Researcher`
- Config-driven AI tuning using `clans.json`, `idioms.json`, etc.
- Async task queue (Celery + Redis)
- Deduplication and merge logic
- Graph traversal engine for ancestry trees

---

8. Git & Versioning

- GitHub repository: `https://github.com/WALEKHWAPHILIP/genealogy`
- `.gitignore` excludes `.env`, venv, `__pycache__`, migrations, and media/static files
- Use `main` branch for stable, `dev` for features

---

9. Final Note

This blueprint reflects both the cultural authenticity and technical foresight required for a modern genealogy platform rooted in Luhya lineage. By starting clean and integrating key architectural patterns early, the system is fully positioned to scale securely and meaningfully.

Prepared by:
Walekhwa Tambiti Leo Philip
AI Systems Architect & Cultural Technologist