# **Wazobia List Project Plan (Updated)**

*For Non-Developers with Intermediate Python/SQL and Basic Linux Skills*

## **Component Choices**

|  |  |  |
| --- | --- | --- |
| **Component** | **Choice** | **Why?** |
| **Frontend** | HTML/CSS + Bootstrap | Simple and fast to launch without JavaScript complexity. |
| **Backend** | Python (Flask + Flask-Login) | Leverages your Python skills; Flask-Login simplifies authentication compared to building from scratch. |
| **Database** | SQLite → PostgreSQL | Start free with SQLite; migrate to PostgreSQL when traffic grows (~500+ users) for better scalability. |
| **Hosting** | Render.com (Free Tier) | Free for the initial launch with auto-deploy from GitHub, keeping costs low. |
| **Domain** | WazobiaList.ng ($10/year) | Brandable, local (.ng) domain that strengthens your market identity. |

## **Revised 4-Week Launch Plan with Enhancements**

### **Week 1: Setup & Core Features (Minimal Viable Product)**

**Goal:** Launch a basic ad-posting site for Lagos with proper version control, basic security, and documentation.

**Day 1-2: Setup & Initial Documentation**

* **Domain & Hosting:**
  + Buy the domain from Namecheap.
  + Set up the Render.com free tier and connect it to your GitHub repository.
* **Development Environment:**
  + Initialize a Git repository in your project folder.
  + Create a README.md that outlines your project structure, installation steps, and future migration plans.
  + Set up a Python virtual environment and install Flask, Flask-Login, SQLAlchemy, and other dependencies.
* **Project Structure:**
  + Create your main file (app.py) and set up a basic SQLite database with a **Listings** table.
  + Establish a folder structure for potential tests (e.g., /tests).

**Day 3-4: Core Features Development**

* **Guest Posting:**
  + Implement ad posting without login (with basic input validation to reduce spam).
* **Ad Form & Listings:**
  + Create a simple ad posting form (fields: Title, Price, Location, Description).
  + Display posted ads on a homepage using Bootstrap cards.
* **Initial Automated Testing:**
  + Write basic unit tests for core functionalities (e.g., ad posting, database insertion) to catch critical issues early.

**Day 5-7: Polish, Basic Security & Deployment**

* **Enhancements:**
  + Add a WhatsApp contact button for quick inquiries (email masking can be deferred).
  + Consider integrating a simple CAPTCHA or input validation mechanism to reduce spam in guest posting.
* **Deployment:**
  + Deploy your application to Render.com.
  + Submit your site to Google Search Console.
* **Logging & Monitoring:**
  + Set up basic Flask logging to capture errors and key usage events.

**Week 1 Deliverable:**  
 👉 Live at **WazobiaList.ng** featuring guest ad posting with a Lagos focus.

### **Week 2: User Accounts & Security (Optional, Based on Week 1 Success)**

**Goal:** Enhance user experience and security after validating the basic concept.

* **User Authentication:**
  + Integrate Flask-Login to enable user registration, login, and session management.
  + Implement email verification for new accounts.
* **Enhanced Security:**
  + Add manual WhatsApp verification for "Trusted Sellers".
  + Introduce rate limiting on endpoints to prevent spam and brute-force attacks.
* **Documentation Update:**
  + Update your README.md with new authentication and security implementation details.

### **Week 3: Differentiators & Growth Initiatives**

**Goal:** Incorporate unique features to set your site apart from competitors like jiji.ng.

* **Localization & UI Enhancements:**
  + Implement a Nigerian Pidgin UI toggle to make the platform more locally appealing.
* **SEO & Social Proof:**
  + Add basic SEO optimizations (meta tags, sitemap.xml).
  + Prepare promotional posts for Nairaland and relevant Facebook groups.
* **Performance Enhancements:**
  + Review and improve input validations, and consider additional measures (like CAPTCHA) if spam becomes an issue.

**Key Message:**  
 “Post free Lagos ads—no fees, no signup!”

### **Week 4: Feedback, Monitoring & Scaling Preparation**

**Goal:** Prepare for growth by gathering feedback and setting up future-proof strategies.

* **User Feedback & Analytics:**
  + Integrate Google Analytics to monitor user behavior and traffic.
  + Add a feedback form or survey to collect user insights.
* **Backup & Documentation:**
  + Script daily backups for your SQLite database (e.g., using a simple cron job/script).
* **Performance Monitoring:**
  + Enhance logging for performance tracking and error monitoring.
* **Scaling Roadmap:**
  + Document plans for migrating from SQLite to PostgreSQL once user growth justifies the change.
  + Outline any future containerization (Docker) or additional caching strategies (Redis, etc.) in your documentation.