PardonMe: Phased Development Plan and Design Specification

Project Overview

PardonMe is a nonprofit web-based application designed to assist individuals with past felony convictions in preparing a compelling pardon application, starting with the State of Wisconsin. The application will guide users through a clear, step-by-step form flow, and ultimately generate a completed PDF suitable for mailing. In a second phase, AI tools will be integrated to improve writing, enhance clarity, and simplify the process.

Two-Phase Development Strategy

Phase 1: Core Application Workflow (No AI)

The goal of this phase is to build a stable, complete, and responsive web application that mirrors the structure and requirements of the Wisconsin pardon application.

Scope

- Collect all required application data across seven sections
- Use session management to maintain user state
- Provide upload support for necessary documents
- Generate a well-formatted PDF version of the application

Tech Stack (Phase 1)

- Backend: Python + Flask
- Frontend: HTML + Jinja2 Templates + Bootstrap 5
- PDF Generation: pdfkit or reportlab
- Session Management: Flask sessions with secure cookies
- Local Uploads: uploads/ directory (no permanent storage)
- UI: Mobile-first responsive layout

📓 User-Facing Workflow

1. Personal Information

2. Criminal History (including repeatable entries and file uploads)

3. Grounds for Pardon (free-text user responses only)

4. Personal Growth (education, employment, community)

5. Attachments (recommendations, evidence)

6. Review and Consent (summary of all inputs)

7. Download PDF

☑ MVP Completion Criteria

- Fully working multi-section form

- Each section saves cleanly to session or database

- PDF output replicates Wisconsin's official format

- Simple input validation in place

- Secure, mobile-friendly interface

Phase 2: AI-Powered Enhancements

Once the MVP is functional and tested, we will add AI tools that improve usability, clarity, and output quality.

🤖 AI Features to Integrate

| Section | AI Enhancement

| 1. Narrative | GPT-powered draft generator from user input

| 2. Personal Growth | AI-generated summary paragraph

| 3. Attachments | Auto-drafted recommendation letters

| 4. Review | AI provides suggestions: "expand this answer"

🚀 Tech Stack (AI Layer)

- AI Integration: OpenAI API or Groq (LLM interface)

- Prompt Tuning: Dynamic, section-specific prompts based on form values

- Security: Use environment variables to protect keys

- User Control: AI suggestions must be optional and editable

🔍 Prompt Logic Examples

- "Take this bullet list of jobs and generate a cohesive growth narrative."

- "Based on the user's explanation of their offense, generate a formal but remorseful summary."

- "Given a relationship type (e.g., pastor, employer), write a sample letter of recommendation.

☑️ AI Completion Criteria

- Each AI feature is triggered by a button or toggle

- Outputs appear in editable textareas (not locked text)

- User can regenerate or ignore suggestions

- No sensitive data is stored or logged

📊 Estimated Timeline (Weekend Dev Pace)

| Task Group | Hours | Target |

| Phase 1: Full core application| ~60–70 hrs| 10 weekends |

| Phase 2: AI Layer Integration | ~25–30 hrs| 4–5 weekends |

🔧 Tools to Use

- GitHub (version control)

- VS Code (IDE)

- Chrome DevTools (mobile/responsive testing)

- Postman (API testing)

- Bootstrap 5 (styling)

- Python-dotenv (environment variable management)

🚀 Deployment Plan

- Platform: Render (preferred), Heroku, or Railway

- Static Files: Hosted in /static/

- Persistent Storage (Phase 2+): S3 or Firebase (TBD)

- Domain: pardonme.org

💼 Final Goal

Build a clean, easy-to-use, secure tool that gives people the dignity and clarity to tell their story, complete their application, and reclaim their future.