# Project Plan

## 1. Project Overview

**Project Name:** Envora – BAS Project & Design Automation Platform

**Owner:** Enviromatic Systems (HQ – DFW)

**Primary Stakeholders:** Ops (PM List users), Design Engineering, Programming, Branch Ops, Ownership

**Vision:**

Replace Excel/Access/VBA patchwork with a cloud-native, project-centric BAS platform that:

- Is the **single source of truth** for projects, equipment, points, and status
- **Generates complete submittal packages** at the click of a button
- **Integrates natively with Visio** via a desktop bridge, not VBA
- Can scale from engineering-only to **company-wide adoption**

**High-Level Deliverables:**

1. Web platform (ASP.NET + Blazor + Azure SQL)
2. Data model for projects, equipment, points, templates, BOM, deliverables
3. Visio desktop bridge (Windows Service) using COM interop
4. Submittal generation engine (PDF assembly)
5. Migration of PM List + core engineering data into platform
6. Documentation, onboarding, and rollout plan

---

## 2. Phasing & Timeline

Assume **~12–16 weeks** total, with you in Cursor and 1–2 internal reviewers.

## Phase 0 – Discovery & Design (1–2 weeks)

**Goals:**

- Lock architecture, scope, and success criteria
- Derive concrete requirements from current files/workflows

**Key Tasks:**

1. **Requirements Workshops**
    - Stakeholder interviews: PMs, engineers, programmers, branch ops
    - Clarify:
        - What must the platform do to replace PM List
        - What must be automated in submittals v1
        - What can be left for v2
2. **Workflow Definition**
    - Map your 12-step HVAC Controls Design Process to platform states:
        - Intake → Design → Review → Submittal → Install → As-built
    - Define transitions and who owns each step
3. **Scope Cut for v1**
    - In scope:
        - Project hub (PM List replacement slice)
        - Equipment + point templates
        - Basic BOM + schedules
        - Visio export for 1–2 diagram types (e.g., control panel & floorplan)
        - Submittal PDF assembly
    - Out of scope (later):
        - Service module
        - Deep estimating
        - Full CRM integration
4. **Architecture & Data Model Finalization**

- Lock stack: ASP.NET Core + Blazor Server + EF Core + Azure SQL + Blob
- Lock integration approach: web app + desktop bridge
- Finalize core tables:
  - Projects, Equipment, Points, EquipmentTemplates
  - Deliverables, VisioMetadata, BOM, Documents, Users/Teams

**Deliverables:**

- Architecture document (what you already sketched, cleaned up)
- Data model diagrams (ERD)
- Finalized Phase 1 feature list

---

# Phase 1 – Core Platform & Data Model (3–4 weeks)

**Goals:**

- Replace PM List for engineering-facing needs
- Stand up the real database + CRUD UI

**Key Tasks:**

1. **Infrastructure Setup**
   - Azure SQL Database
   - Azure Blob Storage
   - Azure App Service (or containers)
   - CI/CD pipeline (GitHub Actions or Azure DevOps)
2. **Core Data Model Implementation**
   - EF Core entities + migrations for:
     - `Projects`
     - `Equipment`, `EquipmentTypes`
     - `Points`
     - `EquipmentTemplates`

    - `Documents`, `Deliverables`
    - `Users`, `Teams`, assignments
3. **Project Hub UI**
    - Project list (replaces PM List **view** for engineers)
    - Project detail page:
        - Overview (job #, customer, dates, contract value)
        - Assignments (AE, PM, PROG, installer, MEP, graphics)
        - Status (design%, programming%, install%)
4. **Equipment Management**
    - Equipment list per project
    - Equipment edit form (tag, type, description, specs blob)
    - Basic search/filter
5. **Point List Management (v1)**
    - Auto-create points from `EquipmentTemplate` on add
    - Points grid per equipment (editable list)
    - Types: input/output/network, units, range, tag
6. **Equipment Template Management (seed only in v1)**
    - Hard-code/seed a few templates:
        - RTU, AHU, VAV, pump, chiller, generic controller
7. **Document Management (v1)**
    - Attach documents to project:
        - Specs, calc sheets, existing Visio, etc.
    - Store in Blob, track in `Documents` table

**Deliverables:**

- Running web app that:
    - Shows real projects

- Lets you enter equipment and see auto-generated points
- Has initial templates in code/seed data

---

## Phase 2 – Data-Driven Schedules & Reports (2–3 weeks)

**Goals:**

- Generate all non-Visio parts of submittals from DB
- No Excel required

**Key Tasks:**

1. **Equipment Schedules**
   - Report generator for:
     - Equipment-by-type
     - Key spec fields shown by type (RTU, VAV, etc.)
   - Render as:
     - HTML (for on-screen)
     - PDF (for submittals)
2. **Point List Reports**
   - Unified point list per project
   - Sort order: equipment type → equipment tag → point tag
   - Export to PDF (and CSV/Excel optional)
3. **BOM Generator**
   - Basic table structure for BOM:
     - Items, quantity, unit cost, extended cost, category
   - Simple parts catalog seed (from your existing design tool)
   - Equipment → BOM items mapping (via template)
4. **Valve & Damper Schedules (v1)**

- Create `Valve` and `Damper` structures:
  - Link to equipment and specs
- Generate tabular schedules from DB
- Basic sizing logic stubbed (can refine later from existing Excel formulas)

5. **PDF Generation Pipeline**
   - Choose library (iTextSharp or Aspose)
   - Implement:
     - Cover page generation (project info)
     - TOC
     - Section assembly for:
       - Equipment schedules
       - Point lists
       - BOM
       - Valve/damper schedules

**Deliverables:**

- "Generate Data Submittal" button:
  - Produces a PDF with everything except drawings
  - No Visio dependency yet

---

# Phase 3 – Visio Desktop Bridge (3–4 weeks)

**Goals:**

- Replace VBA with a C# Windows Service running on engineer PCs
- Prove export path: DB → Visio → PDF → platform

**Key Tasks:**

1. **Bridge Service Skeleton**
   - Windows Service project:

    - Register with platform (`bridgeId`, user, machine)
    - Poll `/api/bridge/jobs`
    - Report completion/errors

2. **Visio Interop Setup**
    - Reference Visio COM
    - Create helper class to:
        - Start Visio (or attach to existing)
        - Open a template
        - Iterate pages/shapes
        - Read/write custom properties
        - Export PDF

3. **Job Infrastructure in Platform**
    - `Jobs` table: `JobId`, `Type`, `Status`, `Payload`, `Result`
    - API endpoints for:
        - Queue job
        - Get pending jobs
        - Mark complete/fail

4. **First Visio Use Case**
    - Pick a single, high-value diagram type, e.g.:
        - RTU control panel layout
    - Create a clean Visio template with:
        - Shape naming conventions (shape name or Prop.Tag ← equipment tag)
        - Custom properties stubbed for equipment fields you care about
    - Implement export job:
        - Input: projectId, equipmentIds, templateUrl

    - Bridge:
        - Download template
        - Fill shapes from equipment data
        - Export PDF and upload

5. **Platform Integration**
    - In project “Drawings” tab:
        - Show last generated PDF
        - Button “Regenerate RTU Panel Drawing”
    - When generating submittal:
        - Queue RTU diagram job
        - Wait (with timeout)
        - Embed resulting PDF section

**Deliverables:**

- Bridge installed on 1–2 engineer machines
- End-to-end demo:
    - Create project + equipment
    - Click “Generate RTU Drawing”
    - PDF produced via Visio bridge and stored in platform

---

# Phase 4 – Full Submittal Builder (2–3 weeks)

**Goals:**

- One button generates full submittal: cover → schedules → BOM → drawings → sequences

**Key Tasks:**

1. **Deliverables & Workflow Model**
    - Implement `Deliverables` table:

    - Type: “Design Package”, “Submittal”, “As-built”
    - Status: Draft → Internal Review → External → Approved
  - Tie deliverables to:
    - Generated PDFs
    - Revision history (Rev A/B/C)
2. **SubmittalBuilder Service**
  - Orchestrate:
    - Data queries
    - Visio jobs
    - PDF section generation
    - PDF assembly
  - Error handling and rebuilding on failure
3. **Sequences of Operation Integration (v1)**
  - Reference only:
    - Attach spec PDFs/sequences to project
    - Include in submittal as final section
4. **Hyperlinks & TOC**
  - Inject bookmark structure into PDF:
    - Sections → pages
  - Optional: cross-links from BOM/points to equipment pages
5. **Review & Approval UI**
  - Internal review:
    - Engineer → PM → Programmer
  - External:
    - Track when sent/approved (metadata only v1)

**Deliverables:**

- “Generate Full Submittal” button that:
  - Queues Visio jobs
  - Builds all data sections
  - Merges into one bookmarked PDF
  - Stores it and exposes a download link

---

## Phase 5 – Data Migration & Rollout (2–4 weeks, can overlap late dev)

**Goals:**

- Get REAL data in
- Onboard initial users
- Retire PM List for engineering

**Key Tasks:**

1. **PM List Migration**
   - Build one-time import pipeline:
     - Map PM List columns to `Projects` + assignments + key fields
   - Validate against current ops workflows
2. **Design Tool Data Migration (v1)**
   - For 2–3 active projects:
     - Move equipment & critical points into platform
   - Keep Excel design tool as reference for formulas, not source of truth
3. **Pilot Rollout**
   - Start with engineering group only:
     - 2–4 engineers + 1 PM + you
   - Use platform for:
     - New projects

    - Selected in-flight projects
4. **Training**
    - Short internal “field manual”:
        - How to create project
        - How to add equipment
        - How to generate submittals
        - How the Visio bridge works
5. **Feedback & Iteration**
    - Weekly feedback cycle
    - Fix pain points, missing fields, UI issues

**Deliverables:**

- Several real projects fully tracked in Envora
- PM List no longer used by engineering for those pilot projects
- Validated submittals generated by platform and accepted by customers/engineers

---

# Phase 6 – Hardening & Expansion (ongoing)

**Goals:**

- Make it robust enough for full company adoption
- Extend beyond engineering

**Key Tasks:**

- Performance and scaling (many projects/points)
- Role-based access control (sales, ops, engineering, programming, accounting)
- Enhanced estimating module
- Integrations (SharePoint, CRM, accounting if desired)
- Branch multi-tenancy (if relevant)

---

## 3. Roles & Ownership

- **Product Owner:** You (defines requirements, accepts features)
- **Technical Lead:** Also you, with Cursor as multiplier
- **Engineering Users:** 2–4 engineers for pilot feedback
- **Ops Rep:** 1 PM representing PM List interests
- **Executive Sponsor:** Your boss (approves scope, budget, time)

---

## 4. Success Criteria

You'll know v1 is successful when:

1. Engineering can run **new projects** without touching Excel or Access.
2. You can **generate a complete submittal package** from the platform that:
   - Engineers trust
   - Customers accept
   - Takes minutes, not days
3. The **PM List is no longer the daily source of truth** for pilot projects.
4. Engineers say: "This is better than the old design tool + macros" and keep using it.