Week 1: AI Workflow Definition and Data Mapping

Foundational Tools

1.Bubble.io (UI&UX) (No-Code Workflows)
2.Supabase (External Database for tiggers)
3.OPENAI
4.GITHUB

Tasks:

- AI Workflow Design:
  - Define input data requirements for OpenAI API (e.g., product name, type, era, approximate value, detailed description, repair tools).
  - Map output data structure for integration with Supabase and Bubble.io.
- Integration Blueprint:
  - Plan Bubble.io workflows for sending inputs to OpenAI and storing outputs in Supabase.
  - Define error-handling processes for failed API calls or incomplete data.

Deliverables:

- Detailed AI workflow blueprint.
- Data input/output mapping for OpenAI API.

Week 2: Supabase Preparation and Initial AI Integration

Tasks:

- Supabase Setup:
  - Update database schema for storing AI predictions.
  - Add logging tables to track API usage for future optimization.
- API Connection in Bubble.io:
  - Establish API connector in Bubble.io for OpenAI.

- ○ Implement a test workflow for sending sample data and receiving predictions.

Deliverables:

- ● Supabase schema ready for AI data.
- ● Basic Bubble.io-to-OpenAI API connection functional.

Week 3: Core AI Functionality and Client Flow Integration

Tasks:

- ● Core Predictions:
  - ○ Develop workflows for generating basic AI predictions (e.g., product name and type).
  - ○ Test prediction accuracy and adjust prompts as needed.
- ● Client Flow Integration:
  - ○ Implement AI workflows into the upload and prediction flow.
  - ○ Create a splash screen for "AI processing" with progress feedback for users.

Deliverables:

- ● Functional AI predictions for client workflows.
- ● Enhanced user experience for AI integration.

Week 4: Tier Logic

Tasks:

- ● Advanced AI Predictions:
  - ○ Add support for era, approximate value, and detailed descriptions.
  - ○ Test and refine outputs to ensure data relevance.
- ● Tier Logic for AI Features:
  - ○ Restrict access to advanced AI features based on user tiers.
  - ○ Add tier upgrade prompts in workflows for locked features.

- Tier-based AI restrictions implemented.
- Payment Gateways is connected to STRIPE.
- Develop backend workflows to:
  - Process payments and upgrade/downgrade tiers.
  - Trigger emails/notifications for successful payments.
- Adjust wallet and sharing functionality.
- Update Supabase queries to respect tier restrictions.

Deliverables:

- Backend logic for tier validation in place.
- Tier-based feature restrictions functional.
- Wallet and sharing features aligned with payment tiers.

Week 5: Testing and Refinement

Tasks:

- AI Testing:
  - Test workflows with sample data across multiple tiers.
  - Validate API reliability and data accuracy under different usage scenarios.
- Bug Fixes and Refinements:
  - Address issues in prediction logic or UI flows.
  - Optimize Supabase queries for performance.

Deliverables:

- Stable and accurate AI workflows.
- Resolved bugs and optimized performance.

Week 6: Client Feedback

Tasks:

- Roll out to a limited group of users:

- ○ Collect feedback on AI predictions, user experience, and tier accessibility.
- Analyze feedback to prioritize enhancements:
  - ○ Identify gaps in prediction quality or workflow efficiency.

Deliverables:

- Comprehensive feedback report.
- Prioritized list of improvements.

Week 7: Final Iterations and Optimization

Tasks:

- Refinements:
  - ○ Implement feedback-driven adjustments to AI predictions and user flows.
  - ○ Finalize all workflows and resolve any remaining issues.
- Optimization:
  - ○ Ensure API calls and database queries are efficient and scalable.
  - ○ Conduct stress testing for high user volumes.

Deliverables:

- Polished AI and user workflows.
- Fully optimized platform for launch.

Week 8: Final Testing and Launch

Tasks:

- Conduct end-to-end testing of all workflows:
  - ○ AI predictions, tier restrictions, payment processing, and user flows.

Deliverables:

- Fully functional and user-tested MVP.

- Platform ready for public release.