Sprint 0 (Weeks 1-2)

- **Deliverables**: Prototype, user workflow document, refined backlog.
- **Focus**: Learning tools, setting up the development environment, and building a simple prototype.

Sprint 1 (Weeks 3-4)

- **Deliverables**: MVP enhancements (basic CRUD functionality, DNS Twist integration).
- **Focus**:
  - Add/remove domains in the frontend.
  - Backend processes multiple domains using DNS Twist.
  - Basic database integration.

Sprint 2 (Weeks 5-6)

- **Deliverables**: Full-stack integration.
- **Focus**:
  - Frontend connects to the backend.
  - Results are displayed dynamically.
  - Store fuzzing results in the database.

Sprint 3 (Weeks 7-8)

- **Deliverables**: Advanced features and optimizations.
- **Focus**:
  - Schedule recurring fuzz tests (e.g., daily).
  - Implement Pub/Sub messaging for results.
  - Docker Compose setup for multi-container deployment.

Sprint 4 (Weeks 9-10)

- **Deliverables**: Final product and presentation.
- **Focus**:
  - Add stretch features (e.g., user authentication, search/filtering).
  - Comprehensive testing and debugging.
  - Prepare for the final demo.

Sprint Zero Outline:

Objectives for Sprint Zero (First Two Weeks)

- Deliverables: **Minimum Viable Product (Prototype)** and **User Workflow Document**.

Tasks

1. **Understand Requirements (1 Day)**
   - Clarify the expectations for the prototype and the user workflow document.
   - Define what constitutes the "minimum viable product" (MVP). For example:
     - A frontend interface to add/remove domains.
     - A backend function that runs DNS Twist on one domain.
     - A database storing a basic domain list.
2. **Create a User Workflow Document (2 Days)**
   - **Definition**: A user workflow document outlines how users interact with the application step-by-step.
     - Example:
       - **Step 1**: User adds a domain via the web interface.
       - **Step 2**: The backend schedules fuzz testing for the domain.
       - **Step 3**: Results are stored in the database and displayed to the user.
   - Tool: Use diagrams or flowcharts (Lucidchart, draw.io).
3. **Build the Prototype (8 Days)**
   - **Frontend**:
     - A simple interface with a form to input a domain name.
     - Display a placeholder for fuzz testing results.
   - **Backend**:
     - Use DNS Twist to process a single hardcoded domain.
     - Output basic results (e.g., console logs or placeholder JSON).
   - **Database**:
     - Create a basic schema for storing domain names and fuzz testing results.
   - **Docker**:
     - Package the prototype backend as a container.

4. **Product Backlog Refinement (2 Days)**
    - Break down features into user stories and tasks for Sprint One.
    - Example backlog items:
        - Add/remove domains functionality.
        - Schedule recurring fuzz testing.
        - Implement a basic results page.
5. **Deliverables**
    - A working prototype that demonstrates the MVP.
    - A user workflow document.
    - A refined backlog for Sprint One.