Project Delivery Timeline for Pet Medical Record System

Month 1: Project Setup and Core Infrastructure

- **Sprint 1**:
  - Set up development environment: Node.js/Express/TypeScript, PostgreSQL, AWS S3, CloudWatch, Twilio.
  - Configure CI/CD pipeline (e.g., GitHub Actions).
  - Define database schema: `Owners`, `Pets`, `Clinics`, `Vets`, `Vet_Clinics`, `roles_permissions` (TRD 5.1).
  - Implement basic authentication: phone + OTP for owners, email + password for vets/receptionists.
  - Deliverable: Project repository, initial database schema, login API (`POST /auth/login`).
- **Sprint 2**:
  - Implement JWT authentication (`jsonwebtoken`) and RBAC middleware for Express.
  - Set up `POST /otp/generate`, `POST /otp/validate` with Twilio SMS integration.
  - Initialize React Native mobile app (Tailwind CSS) and React web UI (CDN-hosted, Tailwind CSS).
  - Deliverable: JWT-protected APIs, OTP flow, basic mobile/web UI skeletons.

Month 2: Pet Profile Management

- **Sprint 3**:
  - Implement `POST /owners`, `POST /pets` APIs for profile creation (TRD 3.1).
  - Support JSONB fields (`photos`, `emergency_contacts`, `insurance_info`) in PostgreSQL.
  - Develop mobile app UI (React Native) for pet profile creation.
  - Deliverable: Owner/pet profile creation, mobile UI for profile input.
- **Sprint 4**:
  - Implement `PATCH /pets/{pet_id}` for profile updates and `POST /pets/{pet_id}/share` for family member sharing (OTP-validated).
  - Set up `Family_Members` table with permissions (Full, Read-Only).
  - Add web UI (React) for profile management.
  - Deliverable: Profile update/sharing APIs, web/mobile UI for profile management.

Month 3: Medical Record Management

- **Sprint 5**:
  - Implement `POST /pets/{pet_id}/records` for immutable record creation (TRD 3.2).
  - Set up `Medical_Records` table with yearly partitioning by `visit_date`.
  - Integrate S3 for test results/images (URLs in `test_results`).
  - Deliverable: Medical record creation API, S3 integration.

- **Sprint 6**:
  - Implement `GET /pets/{pet_id}/records` with pagination/date filters and `GET /pets/{pet_id}/records/search` with PostgreSQL `tsvector`.
  - Develop mobile app timeline view for medical history.
  - Deliverable: Record retrieval/search APIs, timeline UI in mobile app.

Month 4: Access Control

- **Sprint 7**:
  - Implement `POST /pet_clinic_access` for vet assignment with monthly partitioning (TRD 3.3).
  - Set up `Trusted_Clinics` table and `POST /trusted_clinics` API.
  - Enforce RBAC for vet access (assigned pets only, trusted clinics).
  - Deliverable: Vet assignment and trusted clinic APIs.
- **Sprint 8**:
  - Develop web/mobile UI for OTP validation and trusted clinic management.
  - Implement cron job (Node.js `node-cron`) to expire `pet_clinic_access` records after 1 month.
  - Deliverable: Access control UI, expiration logic.

Month 5: Notifications and Appointment Scheduling

- **Sprint 9**:
  - Implement `POST /notifications/sms` for OTP, Pet ID, and visit summaries (TRD 3.4).
  - Develop `POST /appointments` and `GET /clinics/{clinic_id}/queue` APIs (TRD 3.5).
  - Add mobile/web UI for appointment scheduling and queue viewing.
  - Deliverable: Notification system, appointment APIs, scheduling UI.
- **Sprint 10**:
  - Set up `Audit_Logs` table and CloudWatch logging for API requests (TRD 4.4).
  - Implement basic edge case handling: forgotten Pet ID lookup (`GET /pets/lookup`) with phone + pet name (TRD 7).
  - Add UI for pet selection (multiple pets).
  - Deliverable: Audit logging, basic edge case APIs/UI.

Month 6: Testing and Demo Preparation

- **Sprint 11**:
  - Conduct integration testing for APIs (Node.js/Express) and UI (React/React Native).
  - Optimize database with indexes on `pet_id`, `visit_date` (TRD 4.1).
  - Perform load testing for 10K pet lookups, 100K record retrievals.
  - Deliverable: Stable, optimized system.
- **Sprint 12**:
  - Prepare client demo: first vet visit (profile creation, OTP, record addition) and existing user visiting new vet (OTP, vet assignment, record access).
  - Add minimal UI polish: error messages, progress indicators (TRD 4.5).
  - Create demo script and documentation.
  - Deliverable: MVP ready for client demo, demo script.

Demo Scope (Month 6)

The client demo will showcase:

- **First Vet Visit**: Owner creates pet profile (mobile app), receives Pet ID via SMS, vet adds medical record, owner gets visit summary.
- **Existing User, New Vet**: Owner shares Pet ID, receptionist validates OTP (web UI), vet accesses history and adds record (mobile/web).
- **UI**: Mobile app (React Native) with profile creation, timeline view, appointment scheduling; web UI (React) for receptionist/vet tasks.
- **Access Control**: Vet access limited to assigned pets, trusted clinics bypass OTP.

Success Criteria

- **Month 6 Demo**: Client sees both user journeys (first vet visit, existing user new vet) with stable APIs and UI (mobile/web).
- **Performance**: Pet lookup <2s, medical history load <3s, OTP delivery <30s (TRD 4.1).
- **Stability**: 99.9% uptime, no critical bugs in demo (TRD 4.3).
- **User Feedback**: Successful OTP validation, seamless vet access, high owner app usability.