BioSketch - Development Milestones

Date: 11 December 2025 **Total Price:** $3,990 USD **Total Timeline:** 4-8 weeks

---

Milestone 1: Database & Authentication - $995 USD - 1-2 weeks

- Database (seed script, RLS policy fixes)
- Authentication & Email Verification

Milestone 2: Canvas Editor - $2,995 USD - 3-6 weeks

- Canvas Editor (tool fixes, advanced features, export improvements)

Milestone 1: Database & Authentication

Price: $995 USD **Timeline:** 1-2 weeks

---

1. Database

Current State

- Migrations contain hardcoded seed data that fails on reset
- ~130 RLS policies exist across tables with good foundation but gaps
- No seed script exists for populating initial data

Tasks

Seed Script:

- Remove seed data from migrations (`20251018034147`, `20251018033349`, `20251018213609`)
- Create `scripts/seed.ts` using service role key
- Add `npm run seed` script to package.json
- Seed icon categories (12 categories: chemistry, biology, physics, medical, cells, molecules, lab, anatomy, plants, animals, symbols, other)
- Seed sample icons for each category (core set for testing)
- Seed blog categories (3-5 categories with name, slug, color, icon)
- Seed blog tags (common tags for filtering)
- Seed sample blog posts (2-3 published posts with TipTap JSON content, SEO metadata, author)
- Seed testimonials (sample approved testimonials)
- Seed admin user role assignment
- Seed site settings (feature flags, rate limits)

RLS Policy Fixes:

- Add missing INSERT policy for `profiles` table (currently relies only on trigger)
- Fix `icon_categories` SELECT policy - currently blocking authenticated users with "permission denied for schema public" error
- Fix `icons` SELECT policy - ensure both authenticated and unauthenticated users can read

- Allow unauthenticated users to view `icons` and `icon_categories` (for public landing page showcase)
- Verify no conflicts between overlapping `canvas_projects` SELECT policies (5 policies exist: owner, public, approved, admin, collaborator)
- Ensure `login_attempts` policies work correctly with edge functions using service role

Tests

Unit:

- After seeding, database contains 12 icon categories
- After seeding, each icon category has at least one sample icon
- After seeding, blog categories and tags exist
- After seeding, admin user role is assigned
- Unauthenticated user can read icons and categories
- Authenticated user can read icons and categories
- Authenticated user can only view own projects + public projects
- User cannot access another user's private project
- Admin can access all resources
- Collaborator can access shared projects
- Authenticated user can INSERT their own profile via trigger (profile creation on signup)
- Authenticated user can SELECT their own profile
- Authenticated user can UPDATE their own profile
- User cannot SELECT another user's private profile data
- User cannot UPDATE another user's profile
- Unauthenticated user cannot INSERT profiles directly
- Login attempts policies allow edge functions using service role to insert records

Manual:

- Seed script creates all required data types without throwing errors
- Seed script is idempotent (can run multiple times safely)

---

2. Authentication & Email Verification

Current State

- Supabase Auth is configured (email templates may already be configured in Supabase Dashboard - unknown without project access)
- Login attempts tracked in `login_attempts` table
- Profile creation happens on user signup via trigger
- Existing UX for email verification:
  - Sign-up shows toast: "Account created! Please check your email to verify."
  - Password reset shows toast: "Check your email - We've sent you a password reset link"
  - Auth page handles `recovery` type in URL hash to show password update form
- No custom Edge Function for branded email delivery
- No dedicated verify email page - users stay on auth page after signup

Tasks

- Set up Resend account and obtain API key for branded email delivery
- Create Supabase Edge Function `send-email` to handle auth emails via Resend
- Configure Edge Function in `supabase/config.toml`
- Implement branded email templates for `signup` and `recovery`
- Create email template preview page to display all branded email templates (verification, password reset, etc.)
- Create user-facing verify email page at `/auth/verify-email`
- Update sign-up flow to redirect to verify email page after submission
- Implement sign-in handling for unverified emails:
  - Catch "email not confirmed" error
  - Auto-resend verification email via `supabase.auth.resend()`
  - Redirect to verify email page with toast notification
- Enable Send Email hook in Supabase Dashboard (Authentication > Hooks)

Tests

E2E:

Signup & Email Verification:

- User can sign up with email and password

- Sign-up redirects to `/auth/verify-email` after form submission
- Verify email page displays "Check your email" heading with appropriate messaging
- Verify email page shows "Return to Sign In" link
- "Return to Sign In" link navigates back to auth page

Unverified Email Handling:

- Sign-in attempt with unverified email detects "Email not confirmed" error
- Unverified email triggers automatic verification email resend
- After resend, user is redirected to `/auth/verify-email`
- Verify email page displays confirmation that email was resent
- User can click "Return to Sign In" link to navigate back to auth page

Login & Session Management:

- User can sign in with valid credentials (verified email + correct password)
- Sign-in with invalid credentials displays error message or toast
- User session persists across page refresh (user remains authenticated)
- User can sign out successfully
- After sign out, session is cleared and user cannot access protected routes

Password Reset Flow:

- Forgot password form is accessible from auth page
- Forgot password form validates email format
- Submitting valid email shows confirmation message
- Forgot password form has "Back to Sign In" link
- Password reset confirmation page validates password requirements (minimum length, etc.)
- User can successfully set new password via reset link
- After password reset, user can sign in with new password

Unit:

Edge Function Tests:

- `send-email` edge function handles signup event with valid payload
- `send-email` edge function handles recovery event with valid payload
- `send-email` edge function returns success response for valid requests
- `send-email` edge function returns error for missing required fields
- `send-email` edge function correctly formats email templates with user data

Database Trigger Tests:

- Profile creation trigger fires after user signup via Supabase Auth
- Profile record is created with correct user_id matching auth.users.id
- Profile creation trigger sets default values for new profiles

Manual:

Email Delivery:

- Signup sends verification email via Resend (check email received in test inbox)
- Password reset sends recovery email via Resend (check email received in test inbox)
- Emails use branded templates with correct formatting
- Verification email contains clickable verification link
- Password reset email contains clickable reset link

Milestone 2: Canvas Editor

Price: $2,995 USD **Timeline:** 3-6 weeks

---

3. Canvas Editor

Current State

Core Features:

- Fabric.js-based canvas with extensive drawing tools (selection, text, shapes, lines, connectors)
- Properties panel for styling, layers panel for z-order management
- Grid system with snap-to-grid, undo/redo history, auto-save
- Projects stored in `canvas_projects` table with version history
- Project sharing via URL

Drawing Tools:

- Eraser tool implemented using Fabric.js PencilBrush with `globalCompositeOperation: "destination-out"` - eraser paths marked with `isEraserPath: true` and made non-selectable
- Issue: Erased mask does not move with the object - when a partially erased object is moved, the erased area stays in original position
- Custom rotation control in `FabricCanvas.tsx` with blue circular handle and hand/grab icon - cursor style set to `'grab'`
- Quadratic Bezier curves in `curvedLineTool.ts` with green dot control point (#10b981) and guide lines
- Issue: Green dot and guide lines persist on exported canvas - should be hidden during export
- Issue: Control points become disjointed from curved lines during manipulation
- Freehand drawing with PencilBrush and `decimate: 0.01` setting - no effective smoothing
- No ability to add/remove anchor points on existing lines, no point type conversion (smooth ↔ corner), no line jumps, no path simplification

Shape Opacity:

- Opacity controlled via slider in `StylePanel.tsx` using native Fabric.js `opacity` property
- Issue: Opacity reverts to baseline when shape is re-selected

Alignment & Distribution:

- Six alignment functions exist in `ArrangePanel.tsx` : alignLeft, alignCenter, alignRight, alignTop, alignMiddle, alignBottom
- Smart snap guides exist with 5px threshold (`AlignmentGuides.tsx`)
- No "distribute evenly" functionality for equal spacing
- No "match size" functionality for making objects same dimensions
- No visual spacing guides showing gap measurements

Image Handling:

- Basic image cropping exists in `CropTool.tsx` with rectangular and circular crop via Fabric.js `clipPath`
- Image eraser tool for pixel-level editing (`ImageEraserTool.tsx`)
- No arbitrary shape masking (custom polygon, rounded rectangle clipping)
- No color adjustment layers (brightness, contrast, saturation)
- No image filters (grayscale, sepia, scientific color maps)

Icon Library:

- Icon library in `IconLibrary.tsx` with category browsing and search
- SVG loading with on-demand category loading and pagination (20 icons per page)
- IndexedDB caching with 24-hour TTL (`iconCache.ts`)
- SVG sanitization removes scripts and normalizes viewBox
- Web worker processing for complexity analysis (`svgWorker.ts`)
- Issue: Some icons appear completely black on canvas and in icon library panel
- Issue: Icon loading can be slow for complex SVGs

Export:

- Export to PNG/JPEG/SVG with DPI and quality options (150, 300, 600)
- PDF export shows "coming soon" message - not implemented
- No CMYK color mode - all exports use RGB color space
- 26 paper size presets exist in `src/types/paperSizes.ts`
- No custom dimension input fields - users cannot enter exact canvas size in inches, cm, or px

Annotations:

- Panel label tool exists for A, B, C letter annotations (`PanelLabelTool.tsx`)

- Zoom callout tool exists for magnification indicators (`ZoomCalloutTool.tsx`)
- No numbered callout system (①②③) or leader lines with text labels

Tasks

Tool Fixes:

- *Eraser:* Convert eraser strokes to object-local clip paths instead of global canvas paths; ensure erased mask moves with the object when repositioned; maintain erased state through save/load cycles
- *Rotation Handle:* Update rotation control cursor to show grabbing hand or rotation icon during active rotation; change cursor from `grab` to `grabbing` when user is actively dragging
- *Curved Lines:* Ensure green dot control points and guide lines are excluded from export; add control points to export exclusion list in `exportAsPng`, `exportAsJpg`, `exportAsSvg` functions; fix control point synchronization to stay attached to curve endpoints during manipulation
- *Shape Opacity:* Debug opacity state persistence in selection handler; ensure opacity value is correctly read from object when selected; prevent opacity slider from resetting to default on object selection

Advanced Path Editing:

- Add ability to click on path to add new anchor point
- Add ability to select and delete existing anchor points
- Add smooth ↔ corner anchor toggle (smooth points: control handles aligned; corner points: independent handles)
- Detect when lines cross other lines on canvas and automatically create arc/jump at intersection points (configurable: arc, gap, bridge)
- Add "simplify path" button to reduce anchor points while maintaining overall shape with tolerance slider
- Improve freehand to smooth path conversion with curve fitting algorithm and smoothing intensity control

Smart Distribution & Spacing:

- Implement distribute horizontally/vertically - equal spacing between selected objects
- Implement match width/height - make selected objects same dimensions as reference
- Add spacing input field (e.g., "space 20px apart") to ArrangePanel
- Show real-time dimension labels displaying distances between objects during drag
- Implement smart spacing suggestions based on nearby object gaps
- Display visual spacing guides showing gap measurements when moving objects

Image Masking & Filters:

- Implement shape masking - clip images to circles, rounded rectangles, custom shapes (non-destructive)
- Add brightness, contrast, saturation adjustment sliders (-100 to +100) using Fabric.js image filters
- Implement grayscale, sepia filters
- Add scientific color maps for microscopy (false color, heat map, etc.)

Icon Library Improvements:

- Investigate SVG parsing issues causing black icon rendering; check for missing fill attributes and unresolved currentColor references
- Add fill color normalization during SVG processing
- Implement progressive SVG loading with placeholder and loading skeleton
- Implement lazy loading for off-screen icons in library panel
- Add SVG minification during upload/import
- Increase cache TTL for frequently used icons; implement preloading for commonly used categories

PDF Export & Custom Canvas Size:

- Integrate PDF generation library (e.g., jsPDF with svg2pdf.js or pdfkit)
- Convert canvas SVG output to PDF with vector graphics preservation
- Implement CMYK color mode option for print publications
- Add embedded fonts option with fallback to outlined text
- Add custom dimension input fields to canvas settings panel with unit selection (inches, cm, px)

Numbered Annotations & Leader Lines:

- Implement numbered callout system with auto-incrementing circled numbers (①②③)
- Add leader line tool - lines with text labels that point to specific areas
- Create annotation presets - arrows with text, brackets with labels
- Build legend generator to auto-generate figure legends from annotations on canvas

Tests

E2E - Core Functionality:

- User can create, select, and modify objects on canvas
- User can use undo/redo to navigate history
- Grid snapping works when enabled

- Multi-object selection and grouping works
- Layer visibility toggle hides/shows objects
- Layer lock prevents object modification
- Layer reordering changes z-index correctly

E2E - Icon Library Improvements:

- Icon library loads and displays icons within reasonable time
- User can upload and delete custom assets

E2E - Project Operations:

- User can create and save a new project
- User can open and edit existing project
- User can delete a project
- User can view and rollback to previous version
- Changes auto-save to database
- Project thumbnail generates correctly
- Project metadata (keywords, citations) saves and displays

E2E - Export & Sharing:

- User can export canvas as PNG, JPEG, and SVG formats
- User can select export DPI options (150, 300, 600)
- JPEG quality slider affects output file size
- User can share project via URL and shared URL loads correctly

E2E - PDF Export & Custom Canvas Size:

- User can export canvas as PDF file
- CMYK color mode option is available in export dialog
- User can enter custom canvas width and height values
- User can select dimension units (inches, cm, px)
- Canvas resizes to match entered custom dimensions
- Custom canvas size persists after save and reload

E2E - Numbered Annotations & Leader Lines:

- User can add numbered callout that displays ① on canvas
- Adding second callout auto-increments to ②
- User can create leader line with text label
- User can apply annotation preset (arrow with text, bracket with label)
- Legend generator creates text listing all canvas annotations

Unit Tests - Algorithms & Utilities:

- Line intersection detection returns correct crossing points
- Jump arc path calculation produces valid SVG path string
- Path simplification algorithm reduces point count within tolerance threshold
- `distributeHorizontally()` calculates correct x positions for equal spacing
- `distributeVertically()` calculates correct y positions for equal spacing
- `matchWidth()` and `matchHeight()` calculate correct dimensions
- Brightness filter generates correct Fabric.js filter matrix
- Contrast filter generates correct Fabric.js filter matrix
- Saturation filter generates correct Fabric.js filter matrix
- SVG sanitizer adds default fill attribute when missing
- SVG sanitizer replaces currentColor references with explicit color values
- SVG complexity score is calculated correctly
- SVG minification reduces file size
- RGB to CMYK color conversion produces valid CMYK values
- Unit conversion between inches/cm/px calculates correctly
- Curve fitting algorithm produces smooth path from freehand points

Manual Tests - Visual & UX Verification:

Tool Fixes:

- User can erase part of an object on canvas
- Moving erased object maintains the erased mask in correct position (verifies object-local clip paths)
- Erased state persists after save and reload
- Exported PNG does not contain green control points or guide lines
- Exported SVG does not contain green control points or guide lines
- Curved line control points remain attached during manipulation
- Set shape opacity to 50%, deselect, reselect - opacity slider shows 50%
- Set shape opacity, save project, reload - opacity value preserved
- Rotation handle cursor changes appropriately (grab → grabbing during drag)
- Curved line control points move smoothly during manipulation
- Guide lines update in real-time during curve editing

Advanced Path Editing:

- User can modify path by adding and deleting anchor points
- User can toggle anchor point between smooth and corner types
- Crossing lines display visual jump indicator (arc/gap/bridge)
- Path simplification reduces anchor point count

- Freehand drawn lines are converted to smooth curves
- Freehand strokes visually smooth to professional curves
- Smoothing intensity control affects smoothness result

Smart Distribution & Spacing:

- Distribute horizontally creates equal spacing between 3+ objects
- Distribute vertically creates equal spacing between 3+ objects
- Match width makes selected objects same width
- Match height makes selected objects same height
- Spacing input field accepts value and applies exact spacing
- Real-time dimension labels appear between objects during drag
- Visual spacing guides display correct gap measurements
- Smart spacing suggestions appear when nearby gaps are similar

Image Masking & Filters:

- User can apply shape mask (circle or rounded rectangle) to image
- Mask can be edited after application
- Original image is preserved when mask is removed (non-destructive)
- Brightness, contrast, and saturation adjustments are applied to images
- Image adjustments persist after save and reload
- Grayscale and sepia filters are applied to images
- Scientific color map filter is applied to images

Icon Library:

- User can browse and search icons
- User can drag icon onto canvas
- Icons display with correct colors (not black) in library and on canvas
- Icon library visual appearance - all icons show with correct colors
- Icon library performance - search returns results quickly (<500ms)
- Icon library performance - scrolling is smooth without janking
- Progressive loading displays placeholders while icons load

Export:

- User can export with transparent background
- Selection-only export includes only selected objects
- PDF export contains vector graphics (text remains selectable in PDF)
- Export quality visual verification (PDF, PNG at different DPIs)
- CMYK export color accuracy compared to RGB

Notes

Optional refactor: Hover tilt effects could be replaced with subtler alternatives (scale, shadow, border) to improve click accuracy and accessibility.

Required API Keys:

The following API keys are required for Milestone 1. Keys can be provided for development, but production keys are required for deployment.

- **Resend API Key** - For branded email delivery (signup verification, password reset)

Codebase Refactoring:

Significant refactoring opportunities exist in the codebase, including splitting large components, eliminating code duplication, improving type safety, and adding performance optimizations. However, this refactoring cannot be performed until comprehensive app-wide test coverage is established to ensure that refactoring doesn't break existing functionality.

Milestone Scope & Support:

Each milestone describes the tasks to be completed and the tests that verify that the functionality is performing as expected. Improvements or additions beyond the specified scope can be completed in additional milestones.

Ongoing messaging support and bug fixing is provided for all unaltered code and related database infrastructure within the scope of completed milestones.

Development & Testing Process:

Development and testing will be conducted using an independently owned and managed Supabase project. All functionality will be validated against this development environment prior to migration. Upon milestone completion, database schema, edge functions, and related configurations will be replicated to the client's production Supabase project - an invitation to the production Supabase project is required to complete this process. Preview deployments will be hosted independently via Vercel for client review at scheduled progress updates.

Testing is designed to verify that specified test cases pass and does not guarantee that the application is free of defects or will meet all user expectations. Test coverage will be as comprehensive as reasonably achievable within the defined scope.