Blog Section ("Field Notes")

Last Updated: January 13, 2026 **Related Docs:** WORK-SECTION.md | ROUTING.md | MASTER-OVERVIEW.md

---

Table of Contents

Overview

The Blog Section is a **minimalist publishing system** with a two-column layout: intro panel (left) and post cards (right).

Design Philosophy

"Field Notes": A personal archive of thoughts spanning philosophy, statistics, psychology, and the intersection of code and genetics.

Key Features

- **Two-Column Layout**: Intro panel + cards column
- **Featured Post**: First post displayed large at top
- **Placeholder Cards**: "Coming Soon" for future posts
- **Component-Based Posts**: Each post is a React component (not markdown)
- **Dynamic Routing**: `/blog/:slug` routes to individual post pages
- **Metadata Separation**: Post metadata in separate files for organization
- **Extensible**: Easy to add new posts via registry system

Component Architecture

File Structure

```
src/
├── components/
│   └── blog/
│       ├── BlogSection.jsx           # Main section (44 lines)
│       ├── BlogItem.jsx              # Individual card (23 lines)
│       ├── BlogPostPage.jsx          # Full post page (38 lines)
│       └── PlaceholderCard.jsx       # "Coming Soon" card (11 lines)
├── posts/
│   ├── index.js                      # Post registry (9 lines)
│   ├── load.js                       # Post loader (6 lines)
│   ├── ai-biotech-genetics.jsx       # Post content component
│   ├── code-to-kinase.jsx            # (Commented out)
│   ├── masters-origin.jsx            # (Commented out)
│   ├── scroll-systems.jsx            # (Commented out)
│   └── meta/
│       ├── ai-biotech-genetics.js    # Post metadata
│       ├── code-to-kinase.js
│       ├── masters-origin.js
│       └── scroll-systems.js
└── Styles/
    ├── Blog.css                      # Section styles
    └── BlogPost.css                  # Individual post page styles
```

Component Hierarchy

```
<BlogSection>
└── <div className="blog-layout">
    ├── <div className="blog-intro-panel">
    │   ├── Title: "Field Notes"
    │   ├── Subtitle
    │   └── Description
    └── <div className="blog-cards-column">
        ├── <BlogItem featured={true} />      # First post (large)
        └── <div className="publication-shelf">
            ├── <PlaceholderCard label="Coming Soon" /> × 4
            └── (Future posts will go here)
```

Blog Post System

Post Data Structure

Each Post Object:

```
{
  Component: AiBiotechGenetics,          // React component (JSX content)
  slug: 'ai-biotech-genetics',           // URL-safe identifier
  link: '/blog/ai-biotech-genetics',     // Full route path
  title: "How AI Has Affected Biotechnology and Genetics",
  tags: ["AI", "Biotechnology", "Genetics"],
  excerpt: "Tracing the rise of machine learning and LLMs in biotech and genomics.",
  image: "/Posts Images/Artificial_Intelligence_Meets_Genetics.jpg",
  size: "large"                          // Optional: "large", "medium", "small"
}
```

File Organization Pattern

For Each Post:

1. **Content File**: `/src/posts/ai-biotech-genetics.jsx`
   - Default export: React component with post content
   - Full HTML/JSX structure (headings, paragraphs, lists)
2. **Metadata File**: `/src/posts/meta/ai-biotech-genetics.js`
   - Named export: `metadata` object
   - Title, tags, excerpt, image, size
3. **Registry Entry**: `/src/posts/index.js`
   - Import content component
   - Import metadata
   - Combine into post object
   - Add to posts array

Why Separate Files? - **Organization**: Content separated from metadata - **Reusability**: Metadata can be imported independently - **Maintainability**: Easier to update titles/excerpts without touching content - **Performance**: Could lazy load content while showing metadata

---

BlogSection Layout

Component File

File: `/src/components/blog/BlogSection.jsx` (44 lines)

Rendered Structure

```
<section id="blog" className="screen-section blog-wrapper">
  <div className="blog-layout">
    {/* Left column: Intro panel */}
    <div className="blog-intro-panel">
      <h2 className="blog-intro-title">Field Notes</h2>
      <p className="blog-intro-subtitle">
        Reflections on genetics, technology, and the human experience.
      </p>
      <p className="blog-intro-description">
        A space for my thoughts spanning philosophy, statistics, and psychology.
        Here, I explore how code and genetics illuminate our understanding of
        ourselves and the world. This archive is just beginning—expect more soon.
      </p>
    </div>

    {/* Right column: All cards container */}
    <div className="blog-cards-column">
      {/* Featured article - top of right column */}
      {blogPosts.length > 0 && (
        <BlogItem {...blogPosts[0]} featured={true} />
      )}

      {/* Placeholder cards shelf - below featured article */}
      <div className="publication-shelf">
        <PlaceholderCard label="Coming Soon" />
        <PlaceholderCard label="Coming Soon" />
        <PlaceholderCard label="Coming Soon" />
        <PlaceholderCard label="Coming Soon" />
      </div>
    </div>
  </div>
</section>
```

Layout Strategy

Two-Column Grid (CSS expected):

```css
.blog-layout {
  display: grid;
  grid-template-columns: 1fr 2fr; /* 33% intro, 67% cards */
  gap: 4rem;
}

@media (max-width: 768px) {
  .blog-layout {
    grid-template-columns: 1fr; /* Stack on mobile */
  }
}
```

Visual Layout:

```
┌──────────────────┬────────────────────────────────────┐
│                  │                                    │
│  Field Notes     │  [Featured Blog Card - Large]      │
│                  │                                    │
│  Subtitle        │                                    │
│                  │  ┌────────┬────────┬────────┬────────┐│
│  Description     │  │Coming  │Coming  │Coming  │Coming  ││
│  ...             │  │ Soon   │ Soon   │ Soon   │ Soon   ││
│                  │  └────────┴────────┴────────┴────────┘│
└──────────────────┴────────────────────────────────────┘
```

Featured Post

First Post Always Featured:

```jsx
{blogPosts.length > 0 && (
  <BlogItem {...blogPosts[0]} featured={true} />
)}
```

Spread Operator (`{...blogPosts[0]}`): - Passes all post properties as individual props - Equivalent to: `javascript <BlogItem Component={AiBiotechGenetics} slug="ai-biotech-genetics" link="/blog/ai-biotech-genetics" title="How AI Has Affected Biotechnology and Genetics" tags={["AI", "Biotechnology", "Genetics"]} excerpt="Tracing the rise of machine learning..." image="/Posts Images/Artificial_Intelligence_Meets_Genetics.jpg" size="large" featured={true} />`

BlogItem Cards

Component File

File: `/src/components/blog/BlogItem.jsx` (23 lines)

Props Interface

```
<BlogItem
  title="How AI Has Affected Biotechnology and Genetics"
  excerpt="Tracing the rise of machine learning and LLMs..."
  tags={["AI", "Biotechnology", "Genetics"]}
  link="/blog/ai-biotech-genetics"
  image="/Posts Images/Artificial_Intelligence_Meets_Genetics.jpg"
  size="large"                 // Optional: "large", "medium", "small"
  featured={true}              // Optional: true for featured styling
/>
```

Rendered Structure

```
<a className="blog-card large featured" href="/blog/ai-biotech-genetics">
  <img
    src="/Posts Images/Artificial_Intelligence_Meets_Genetics.jpg"
    alt="How AI Has Affected Biotechnology and Genetics"
    className="card-image"
  />
  <div className="blog-card-overlay" />
  <div className="blog-card-content">
    <h3>How AI Has Affected Biotechnology and Genetics</h3>
    <p className="excerpt">
      Tracing the rise of machine learning and LLMs in biotech and genomics.
    </p>
    <div className="tags">
      <span className="tag">AI</span>
      <span className="tag">Biotechnology</span>
      <span className="tag">Genetics</span>
    </div>
  </div>
</a>
```

Visual Layers

4 Overlapping Layers:

1. **Card Image** (`<img className="card-image" />`):
   - Background image covering full card
   - Expected CSS: `object-fit: cover`, `position: absolute`
2. **Card Overlay** (`<div className="blog-card-overlay" />`):
   - Dark gradient overlay for text readability
   - Expected CSS: `background: linear-gradient(to bottom, rgba(0,0,0,0), rgba(0,0,0,0.8))`
3. **Card Content** (`<div className="blog-card-content" />`):
   - White text on top of overlay
   - Title, excerpt, tags
   - Expected CSS: `position: relative`, `z-index: 2`
4. **Link Wrapper** (`<a className="blog-card" />`):
   - Entire card is clickable
   - Expected hover effect: scale, shadow

Size Classes

Dynamic Class Name:

```
const sizeClass = size ? ` ${size}` : "";
const featuredClass = featured ? " featured" : "";
return <a className={`blog-card${sizeClass}${featuredClass}`} href={link}>;
```

Examples: - `blog-card large featured` → Featured post - `blog-card medium` → Regular post - `blog-card` → Default size

Expected CSS:

```
.blog-card {
  /* Default size */
  width: 100%;
  height: 300px;
}

.blog-card.large {
  height: 500px; /* Taller featured card */
}

.blog-card.medium {
  height: 350px;
}
```

```
.blog-card.small {
  height: 250px;
}

.blog-card.featured {
  /* Special styling for featured post */
  border: 2px solid var(--accent-color);
  box-shadow: 0 10px 40px rgba(0, 0, 0, 0.3);
}
```

Tags Rendering

Array Mapping:

```
const tagList = Array.isArray(tags) ? tags : [];
return (
  <div className="tags">
    {tagList.map((tag, i) => (
      <span key={i} className="tag">
        {tag}
      </span>
    ))}
  </div>
);
```

Safety Check: `Array.isArray(tags)` prevents errors if tags is undefined/null.

BlogPostPage

Component File

File: `/src/components/blog/BlogPostPage.jsx` (38 lines)

Purpose

Full-Page Post View: Renders individual blog post content at `/blog/:slug`.

Route Integration

Dynamic Route (from App.jsx):

```
<Route path="/blog/:slug" element={
  <Layout>
    <BlogPostPage />
  </Layout>
} />
```

How It Works: 1. User clicks blog card → Navigate to `/blog/ai-biotech-genetics` 2. Router matches `/blog/:slug` route → Renders `<BlogPostPage />` 3. `useParams()` extracts slug → `{ slug: "ai-biotech-genetics" }` 4. `loadPost(slug)` finds matching post → Returns post object 5. Renders post component → `<AiBiotechGenetics />`

Component Logic

```
export default function BlogPostPage() {
  const { slug } = useParams();              // Extract slug from URL
  const [post, setPost] = useState(null);    // Post data state
  const [notFound, setNotFound] = useState(false); // 404 state

  useEffect(() => {
    loadPost(slug).then((data) => {
      if (data) {
        setPost(data);
        setNotFound(false);
      } else {
        setNotFound(true);
      }
    });
  }, [slug]);

  if (notFound) {
    return <h1>404 - Post Not Found</h1>;
  }

  if (!post) return <p>Loading...</p>;

  const { Component, title } = post;

  return (
    <article className="blog-post">
      <h1>{title}</h1>
      <Component />
    </article>
  );
}
```

State Flow

Three States:

1. **Loading** (`post === null`, `notFound === false`):

   ```
   <p>Loading...</p>
   ```

2. **Not Found** (`notFound === true`):

   ```
   <h1>404 - Post Not Found</h1>
   ```

3. **Loaded** (`post !== null`, `notFound === false`):

   ```
   <article className="blog-post">
     <h1>{title}</h1>
     <Component />
   </article>
   ```

Post Component Rendering

Dynamic Component Rendering:

```
const { Component, title } = post;
return (
  <article className="blog-post">
    <h1>{title}</h1>
    <Component />
  </article>
);
```

How It Works: - `Component` is a React component (e.g., `AiBiotechGenetics`) - `<Component />` renders it like `<AiBiotechGenetics />` - No props passed → Post content is self-contained

---

Placeholder System

PlaceholderCard Component

File: `/src/components/blog/PlaceholderCard.jsx` (11 lines)

Purpose

Visual Indicator: Shows where future blog posts will appear.

Rendered Structure

```
<div className="placeholder-card">
  <div className="placeholder-content">
    <div className="placeholder-icon">□ </div>
    <p className="placeholder-label">Coming Soon</p>
  </div>
</div>
```

Props Interface

```
<PlaceholderCard label="Coming Soon" />
```

Default Label: `"Coming Soon"` if no label prop provided.

Visual Design (Expected CSS)

```
.placeholder-card {
  border: 2px dashed var(--border-color);
  background: rgba(100, 100, 100, 0.1);
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 250px;
  border-radius: 8px;
}

.placeholder-icon {
  font-size: 3rem;
  opacity: 0.5;
}

.placeholder-label {
  font-size: 1.125rem;
  color: var(--text-secondary);
  opacity: 0.7;
}
```

Publication Shelf

Grid of 4 Placeholders:

```
<div className="publication-shelf">
  <PlaceholderCard label="Coming Soon" />
  <PlaceholderCard label="Coming Soon" />
  <PlaceholderCard label="Coming Soon" />
  <PlaceholderCard label="Coming Soon" />
</div>
```

Expected CSS:

```
.publication-shelf {
  display: grid;
  grid-template-columns: repeat(4, 1fr); /* 4 columns */
  gap: 1.5rem;
}

@media (max-width: 1200px) {
  .publication-shelf {
    grid-template-columns: repeat(2, 1fr); /* 2 columns on tablet */
  }
}

@media (max-width: 768px) {
  .publication-shelf {
    grid-template-columns: 1fr; /* 1 column on mobile */
  }
}
```

Post Registration

index.js Registry

File: `/src/posts/index.js` (9 lines)

Current State:

```javascript
import AiBiotechGenetics from './ai-biotech-genetics.jsx';
import { metadata as aiBioMeta } from './meta/ai-biotech-genetics.js';

const posts = [
  {
    Component: AiBiotechGenetics,
    slug: 'ai-biotech-genetics',
    link: '/blog/ai-biotech-genetics',
    ...aiBioMeta
  },
];

export default posts;
```

Spread Operator (`...aiBioMeta`): - Unpacks metadata object into post object - Equivalent to: `javascript { Component: AiBiotechGenetics, slug: 'ai-biotech-genetics', link: '/blog/ai-biotech-genetics', title: "How AI Has Affected Biotechnology and Genetics", tags: ["AI", "Biotechnology", "Genetics"], excerpt: "Tracing the rise...", image: "/Posts Images/Artificial_Intelligence_Meets_Genetics.jpg", size: "large" }`

Adding New Posts

3-Step Process:

1. **Create Content File**: `/src/posts/my-new-post.jsx`

   ```javascript
   export default function MyNewPost() {
     return (
       <div>
         <h2>Section Title</h2>
         <p>Content here...</p>
       </div>
     );
   }
   ```

2. **Create Metadata File**: `/src/posts/meta/my-new-post.js`

   ```javascript
   export const metadata = {
     title: "My New Post Title",
     tags: ["Tag1", "Tag2"],
     excerpt: "Short description...",
     image: "/Posts Images/my-new-post.jpg",
     size: "medium"
   };
   ```

3. **Register in index.js**:

   ```
   import MyNewPost from './my-new-post.jsx';
   import { metadata as myNewPostMeta } from './meta/my-new-post.js';

   const posts = [
     {
       Component: AiBiotechGenetics,
       slug: 'ai-biotech-genetics',
       link: '/blog/ai-biotech-genetics',
       ...aiBioMeta
     },
     {
       Component: MyNewPost,
       slug: 'my-new-post',
       link: '/blog/my-new-post',
       ...myNewPostMeta
     },
   ];
   ```

Automatic Features: - First post becomes featured (large card) - New posts appear in BlogSection - Routing automatically works for `/blog/my-new-post`

Routing Integration

Route Definition

From `/src/App.jsx`:

```
<Route path="/blog/:slug" element={
  <Layout>
    <BlogPostPage />
  </Layout>
} />
```

Dynamic Parameter: `:slug` matches any URL segment.

URL Matching Examples

URL	Matches?	Extracted slug
`/blog/ai-biotech-genetics`	Yes	`"ai-biotech-genetics"`
`/blog/code-to-kinase`	Yes	`"code-to-kinase"`
`/blog/123-test-post`	Yes	`"123-test-post"`
`/blog`	No	N/A (no slug)
`/blog/`	No	Empty slug

Post Loading

File: `/src/posts/load.js` (6 lines)

```
import posts from './index.js';

export async function loadPost(slug) {
  return posts.find((p) => p.slug === slug) || null;
}
```

Why Async? - **Future-Proofing**: Could lazy load post components in future - **Consistent API**: Returns promise (consistent with fetch patterns) - **Currently Synchronous**: No actual async operation, but maintains promise interface

Find Logic: - Searches posts array for matching slug - Returns post object if found - Returns `null` if not found (triggers 404 in BlogPostPage)

Content Authoring

Post Content Structure

Example: `/src/posts/ai-biotech-genetics.jsx`

```
// Metadata is defined in a separate module to satisfy ESLint rules.

export default function AiBiotechGenetics() {
  return (
    <div>
      <h2>I. Introduction: A Double Helix Meets an Algorithm</h2>
      <p>
        The convergence of Artificial Intelligence (AI) with the life
        sciences—specifically biotechnology, genetics, and bioinformatics—has
        catalyzed a paradigm shift in how we decode, manipulate, and engineer
        biological systems...
      </p>

      <h2>II. From Statistical Genomics to Early ML (2000s – 2012)</h2>
      <h3>Early AI in Life Sciences: Weak Learners, Strong Potential</h3>
      <p>
        The early 2000s saw the rise of machine learning in bioinformatics,
        driven by the explosion of biological data from projects like the Human
        Genome Project (completed in 2003)...
      </p>

      <ul>
        <li>Gene expression profiling (microarray data)</li>
        <li>Protein sequence classification</li>
        <li>SNP (Single Nucleotide Polymorphism) association mapping</li>
        <li>Basic protein structure prediction</li>
      </ul>

      {/* More sections... */}
    </div>
  );
}
```

Supported HTML Elements

Full JSX/HTML Support: - **Headings**: `<h2>`, `<h3>`, `<h4>` - **Paragraphs**: `<p>` - **Lists**: `<ul>`, `<ol>`, `<li>` - **Links**: `<a href="...">` - **Emphasis**: `<strong>`, `<em>`, `<code>` - **Blockquotes**: `<blockquote>` - **Images**: `<img src="..." alt="..." />` - **Custom Components**: Can import and use React components

Styling

Global Styles (from `/src/Styles/BlogPost.css`):

```
.blog-post {
  max-width: 800px;
  margin: 0 auto;
  padding: 4rem 2rem;
  color: var(--text-primary);
}

.blog-post h1 {
  font-size: 3rem;
  margin-bottom: 2rem;
  color: var(--heading-color);
}

.blog-post h2 {
  font-size: 2rem;
  margin-top: 3rem;
  margin-bottom: 1rem;
  color: var(--heading-color);
}

.blog-post p {
  font-size: 1.125rem;
  line-height: 1.8;
  margin-bottom: 1.5rem;
}

.blog-post ul, .blog-post ol {
  margin-left: 2rem;
  margin-bottom: 1.5rem;
}

.blog-post li {
  margin-bottom: 0.5rem;
}
```

Mobile Responsiveness

Layout Breakpoints

768px Breakpoint:

```
/* Desktop (> 768px) */
.blog-layout {
  display: grid;
  grid-template-columns: 1fr 2fr; /* Intro panel + Cards */
}

/* Mobile (≤ 768px) */
@media (max-width: 768px) {
  .blog-layout {
    grid-template-columns: 1fr; /* Stack vertically */
  }
}
```

Visual Change:

Desktop:

```
┌──────────────────┬───────────────────────────┐
│  Intro Panel     │  Featured Card            │
│                  │  Placeholder Cards        │
└──────────────────┴───────────────────────────┘
```

Mobile:

```
┌──────────────────────────┐
│  Intro Panel             │
├──────────────────────────┤
│  Featured Card           │
├──────────────────────────┤
│  Placeholder Card 1      │
├──────────────────────────┤
│  Placeholder Card 2      │
├──────────────────────────┤
│  ...                     │
└──────────────────────────┘
```

Publication Shelf Breakpoints

```css
/* Desktop (> 1200px) */
.publication-shelf {
  grid-template-columns: repeat(4, 1fr); /* 4 columns */
}

/* Tablet (769px - 1200px) */
@media (max-width: 1200px) {
  .publication-shelf {
    grid-template-columns: repeat(2, 1fr); /* 2 columns */
  }
}

/* Mobile (≤ 768px) */
@media (max-width: 768px) {
  .publication-shelf {
    grid-template-columns: 1fr; /* 1 column */
  }
}
```

Future Enhancements

Potential Additions

1. **MDX Support**
   - **Current**: Posts are JSX components (React code)
   - **Enhancement**: MDX files (markdown with JSX components)
   - **Benefit**: Easier authoring for non-technical writers
   - **Implementation**: Install `@mdx-js/react`, update Vite config
2. **Post Filtering by Tags**
   - **Current**: All posts visible
   - **Enhancement**: Click tag → Filter posts
   - **Implementation**: State for active tag + filter logic
3. **Search Functionality**
   - **Current**: No search
   - **Enhancement**: Search bar filtering by title/excerpt/content
   - **Implementation**: Fuse.js or simple string matching
4. **Pagination**
   - **Current**: All posts on one page
   - **Enhancement**: "Load More" or page numbers
   - **Implementation**: Slice posts array, state for current page
5. **Reading Time Estimate**
   - **Current**: No time estimate
   - **Enhancement**: "5 min read" badge on cards
   - **Implementation**: Calculate word count / 200 WPM
6. **Post Dates**
   - **Current**: No dates shown
   - **Enhancement**: Published date + last updated
   - **Implementation**: Add `date` field to metadata
7. **RSS Feed**
   - **Current**: No RSS
   - **Enhancement**: Programmatically generate RSS XML
   - **Implementation**: Build-time script generating `rss.xml`
8. **Social Sharing**
   - **Current**: No share buttons
   - **Enhancement**: Twitter, LinkedIn, copy link buttons

- **Implementation**: Share API or direct links

9. **Comments System**
   - **Current**: No comments
   - **Enhancement**: Disqus, Commento, or custom system
   - **Implementation**: Embed third-party widget or build custom
10. **Related Posts**
    - **Current**: No recommendations
    - **Enhancement**: "Related Articles" based on tags
    - **Implementation**: Tag similarity algorithm

Related Documentation

- WORK-SECTION.md - Previous section (interactive lab)
- CONTACT-SECTION.md *(coming soon)* - Next section (contact form)
- ROUTING.md - Blog routing details
- ARCHITECTURE.md - Component hierarchy
- MASTER-OVERVIEW.md - Full portfolio overview

Quick Reference

Key Files

File	Lines	Purpose
`BlogSection.jsx`	44	Main section with intro + cards
`BlogItem.jsx`	23	Individual blog card
`BlogPostPage.jsx`	38	Full post page (dynamic route)
`PlaceholderCard.jsx`	11	"Coming Soon" card
`index.js`	9	Post registry
`load.js`	6	Post loader by slug

Post Metadata Schema

```
{
  title: string,        // Post title
  tags: string[],       // Array of tag strings
  excerpt: string,      // Short description (1-2 sentences)
  image: string,        // Path to featured image
  size: string          // Optional: "large", "medium", "small"
}
```

Post Object Schema

```
{
  Component: React.Component,  // Content component
  slug: string,                // URL-safe identifier
  link: string,                // Full route path
  ...metadata                  // Spread metadata fields
}
```

Adding a New Post Checklist

- ☐ Create content file: `/src/posts/my-post.jsx`
- ☐ Export default component with post content
- ☐ Create metadata file: `/src/posts/meta/my-post.js`
- ☐ Export `metadata` object with title, tags, excerpt, image
- ☐ Import both in `/src/posts/index.js`
- ☐ Add to `posts` array with slug and link
- ☐ Test at `/blog/my-post`

This blog system prioritizes simplicity and extensibility, allowing easy addition of posts while maintaining clean component architecture.