Portfolio Website Development Plan

Tech Stack: React + GitHub Pages/Netlify | **Time Estimate:** 2 Hours

1. Project Structure Diagram

```
flowchart TD
   A[Portfolio Website] --> B[Header]
   A --> C[Hero Section]
   A --> D[Projects Grid]
   A --> E[Blog Feed]
   A --> F[Contact]
   B --> G[Hamburger Menu Mobile]
   D --> H[Project Cards 9:16]
   E --> I[Markdown Blog System]
```

2. User Story Mapping

User Story	Implementation Task	Time
As a visitor, I want to:		
View key info quickly	Create Hero Section (Name, Role, 1-line Bio)	15m
See technical projects	Build 3x3 Grid of Project Cards (9:16 ratio)	25m
Read blog articles	Implement Markdown-based Blog System	20m
Access navigation easily	Create Responsive Hamburger Menu	15m
Contact via standard methods	Add Social Links & Email	10m

3. Component Architecture

```
 // Core Components
<Portfolio>
  <Header />       // With Hamburger (Mobile)
  <Hero />
  <Projects />     // 3x3 Grid
  <BlogFeed />     // 3 Latest Posts
  <Contact />
</Portfolio>

// Blog Components
<BlogLayout>
  <MarkdownRenderer />
  <Comments />      // Optional (Disqus/Giscus)
</BlogLayout>
```

4. Implementation Roadmap

Phase 1: Setup (30 mins)

```
 1. Create GitHub Repo: `username-portfolio`
2. Initialize React App: `npx create-react-app portfolio --template cra-template-pwa`
3. Install Dependencies:
   - `npm install react-markdown @emotion/styled @mui/icons-material`
4. Setup Netlify/GitHub Pages:
   - Connect repo to hosting provider
   - Configure build commands (`npm run build`)
```

Phase 2: Core Features (45 mins)

Component	Key Features	Code Snippet
Hamburger Menu	Mobile-first, Animated	

```
 const [isOpen, setIsOpen] = useState(false);
<MenuIcon onClick={() => setIsOpen(!isOpen)} />
{isOpen && <NavLinks />}
```

| *Project Cards* | 9:16 Ratio, Hover Effects |

```
 .card {
  aspect-ratio: 9/16;
  transition: transform 0.3s;
}
.card:hover {
  transform: scale(1.03);
}
```

| *Blog System* | Markdown Processing |

```
 import ReactMarkdown from 'react-markdown';
<ReactMarkdown>{postContent}</ReactMarkdown>
```

Phase 3: Deployment (30 mins)

```
 1. Optimize Build:
    - `npm run build`
    - Compress images (Squoosh.app)
2. Deploy:
    - GitHub Pages: `gh-pages` branch setup
    - Netlify: Drag-and-drop build folder
3. Verify:
    - Test mobile responsiveness
    - Check blog post rendering
```

---

5. Content Structure

```
src/
├── content/
│   ├── projects.json       # Project data
│   └── posts/              # Blog markdowns
│       └── 2024-02-20.md
├── components/
│   ├── Card.jsx            # Project cards
│   └── HamburgerMenu.jsx
```

Sample Project Data (projects.json):

```
[
  {
    "title": "AI Chat System",
    "image": "ai-project.jpg",
    "aspectRatio": "9/16",
    "github": "link",
    "preview": "live-demo.com"
  }
]
```

---

6. Visual Guidelines

Black & White Theme Specs

```
 /* theme.js */
export const theme = {
  colors: {
    primary: '#000000',
    secondary: '#FFFFFF',
    accent: '#1A1A1A',
    text: '#333333'
  },
  breakpoints: ['768px'] // Mobile-first
};
```

Grid Layout Example

```
 // 3x3 Responsive Grid
<Grid container spacing={2}>
  {projects.map((project) => (
    <Grid item xs={12} sm={6} md={4}>
      <ProjectCard {...project} />
    </Grid>
  ))}
</Grid>
```

---

7. Deployment Checklist

✅ Functional Hamburger Menu (Test on <768px)
✅ All Project Cards Link Correctly
✅ Blog Posts Render Markdown Properly
✅ CNAME Setup (If Using Custom Domain)

✅ SEO Meta Tags Added
✅ 404 Page Implemented

---

8. Time Optimization Tips

1. Use **AI Code Generation** for repetitive components (Buttons, Cards)
2. Pre-load markdown content in `/public` folder
3. Use Netlify's **Deploy Preview** for instant QA
4. Copy-paste ready-made React snippets from open sources

Final Live URL Structure:
`https://yourusername.netlify.app` or `https://username.github.io/portfolio`

Need any component code samples or want me to create specific implementation templates? Share your preferred styling approach (**CSS-in-JS vs traditional**) and I'll provide exact code blueprints!