Product Requirements Document (PRD) for Security Business Website
Built with React, TypeScript, Node.js, and Deployed to GitHub Pages

---

1. Project Overview
- **Objective**: Build a mobile-first security business website with modern tooling.
- **Tech Stack**:
  - Frontend: React + TypeScript + Tailwind CSS
  - Build Tool: Vite
  - Testing: Vitest + Testing Library
  - Deployment: GitHub Pages
- **Design Reference**: [Styling & Layout Guide](https://4d212b8c-2c6a-4849-86ac-402c15600ce4-00-whhbi2lg9v6b.riker.replit.dev)

---

2. Development Environment Setup
2.1 Prerequisites
- Node.js v18+ (for npm scripts and tooling)
- Git
- GitHub Account

2.2 Initialize Project
```bash
mkdir security-business-website
cd security-business-website
git init
npm init -y
```

---

3. Core Dependencies
3.1 Install Dependencies
```bash
# Frontend
npm install react react-dom react-router-dom @emailjs/browser axios framer-motion react-helmet react-hook-form @hookform/resolvers yup tailwindcss @headlessui/react react-icons

# TypeScript
npm install typescript @types/node @types/react @types/react-dom @types/react-router-dom @types/axios @types/react-helmet --save-dev

# Build & Tooling
```

```
npm install vite @vitejs/plugin-react prettier eslint eslint-config-prettier husky lint-staged --save-dev

# Testing
npm install @testing-library/react @testing-library/jest-dom @testing-library/user-event vitest --save-dev

# Deployment
npm install gh-pages --save-dev
```

---

4. Project Configuration
4.1 TypeScript Config
```json
// tsconfig.json
{
  "compilerOptions": {
    "target": "ES2020",
    "lib": ["ES2020", "DOM", "DOM.Iterable"],
    "module": "ESNext",
    "strict": true,
    "jsx": "react-jsx",
    "baseUrl": ".",
    "paths": { "@/*": ["src/*"] }
  },
  "include": ["src"]
}
```

4.2 Tailwind Config
```javascript
// tailwind.config.js
module.exports = {
  content: ["./src/**/*.{ts,tsx}"],
  theme: {
    screens: { sm: "640px", md: "768px", lg: "1024px" },
    colors: { primary: "#1a365d", secondary: "#2d3748" }
  },
  plugins: [require("@tailwindcss/aspect-ratio")]
};
```

4.3 Vite Config
```typescript
// vite.config.ts
```

```
import { defineConfig } from 'vite';
import react from '@vitejs/plugin-react';

export default defineConfig({
  plugins: [react()],
  base: "/security-business-website/" // Match GitHub repo name
});
```

---

5. Project Structure
```
security-business-website/
├── src/
│   ├── assets/        # Images, fonts, global CSS
│   ├── components/    # Reusable components
│   ├── pages/         # Page components
│   ├── services/      # API/EmailJS integration
│   ├── App.tsx        # Root component
│   └── main.tsx       # Entry point
├── public/            # Static assets
├── .github/workflows/   # CI/CD (optional)
├── .env.example       # Environment variables
├── package.json
└── tsconfig.json
```

---

6. Implementation Steps
6.1 Setup Mobile-First Meta Tags
```html
<!-- public/index.html -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="theme-color" content="#1a365d">
```

6.2 Create Core Components
Follow the component structure from the [design reference](https://4d212b8c-2c6a-4849-86ac-402c15600ce4-00-whhbi2lg9v6b.riker.replit.dev), including:
- Responsive navigation (`MobileNav.tsx`)
- Contact form with validation (`ContactForm.tsx`)
- Image gallery with lazy loading (`ImageGallery.tsx`)

6.3 Routing
```typescript
```

```
// App.tsx
import { BrowserRouter, Routes, Route } from 'react-router-dom';
import { HomePage, AboutPage, ServicesPage } from './pages';

const App = () => (
  <BrowserRouter>
    <Routes>
      <Route path="/" element={<HomePage />} />
      <Route path="/about" element={<AboutPage />} />
      <Route path="/services" element={<ServicesPage />} />
    </Routes>
  </BrowserRouter>
);
```

---

7. EmailJS Integration
7.1 Environment Variables
```env
# .env
VITE_EMAIL_SERVICE_ID=your_service_id
VITE_EMAIL_TEMPLATE_ID=your_template_id
VITE_EMAIL_PUBLIC_KEY=your_public_key
```

7.2 Email Service
```typescript
// src/services/email.ts
import emailjs from '@emailjs/browser';

export const sendEmail = async (params: EmailParams) => {
  await emailjs.send(
    import.meta.env.VITE_EMAIL_SERVICE_ID,
    import.meta.env.VITE_EMAIL_TEMPLATE_ID,
    params,
    import.meta.env.VITE_EMAIL_PUBLIC_KEY
  );
};
```

---

8. Testing
8.1 Example Test
```typescript
// src/components/__tests__/Button.test.tsx
```

```
import { render, screen } from '@testing-library/react';
import Button from '../common/Button';

test('renders button', () => {
  render(<Button variant="primary">Click Me</Button>);
  expect(screen.getByText(/click me/i)).toBeInTheDocument();
});
```

---

9. Deployment to GitHub Pages
9.1 Configure package.json
```json
{
  "scripts": {
    "dev": "vite",
    "build": "tsc && vite build",
    "predeploy": "npm run build",
    "deploy": "gh-pages -d dist"
  },
  "homepage": "https://[your-github-username].github.io/security-business-website"
}
```

9.2 Deployment Steps
1. Push code to GitHub repository.
2. Run:
```bash
npm run deploy
```
3. Enable GitHub Pages in repo settings (`Settings > Pages > Branch: gh-pages`).

---

10. Post-Deployment Checklist
- [ ] Test responsive design on mobile/desktop.
- [ ] Validate SEO meta tags using [Google Rich Results Test](https://search.google.com/test/rich-results).
- [ ] Ensure EmailJS integration works in production.
- [ ] Add analytics (e.g., Google Analytics).

---

11. Maintenance
- Use GitHub Issues for bug tracking.
- Enable Dependabot for dependency updates.

- Add CI/CD pipeline for automated testing/deployment.

This PRD covers all aspects of development, testing, and deployment. Follow the design reference for styling specifics and ensure all TypeScript types are strictly enforced.