**Digilink IT Solutions**

Project Documentation

Digilink IT Solutions Website Documentation

2. Core Features

1. Introduction

A comprehensive guide to the architecture, setup, and deployment of the Digilink IT Solutions professional website. Last Updated: January 2025.

4. Technology Stack

3. Application Workflow

5. Project Structure

6. Setup & Configuration

# Introduction

Welcome to the official documentation for the **Digilink IT Solutions** website project. This document provides a deep dive into every aspect of the project, from its core features and technology stack to detailed setup instructions and deployment workflows.

10. Monitoring & Maintenance

9. Design System

8. Interactive Features

7. Deployment Guide

The project is a professional, AI-themed website designed to serve as the primary digital presence for Digilink IT Solutions. It features a modern, responsive design, dynamic animations, and a fully integrated backend system for handling client communications. The core of its functionality lies in a robust contact form that captures user submissions, stores them securely in a cloud database, and triggers real-time email notifications.

12. Support

11. Security

**Company Slogan:** "Connecting You to the Digital World"

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

This documentation is intended for developers, system administrators, and project stakeholders who need to understand, maintain, or extend the project's capabilities.

# Core Features

The website is built with a rich set of features designed to provide a professional user experience and a streamlined administrative workflow.

**Professional AI-Themed Design:** A visually appealing interface featuring neural network animations and a modern aesthetic that aligns with the IT solutions theme.

**Fully Responsive Layout:** The user interface is meticulously crafted to provide an optimal viewing and interaction experience across a wide range of devices, from desktops to mobile phones.

**Integrated Contact Form:** A user-friendly contact form with real-time, client-side validation to guide users, and robust server-side validation for data integrity.

**AWS DynamoDB Integration:** All contact form submissions are securely and reliably stored in an AWS DynamoDB table, providing a persistent record of all inquiries.

**Real-Time Email Notifications:** Utilizes the Resend API to send instant email notifications upon successful form submission, ensuring prompt follow-up.

**Smooth Scroll Navigation:** A seamless single-page navigation experience, allowing users to jump between sections of the homepage effortlessly.

**WhatsApp & Email Integration:** Direct contact methods are integrated, including a clickable phone number that opens WhatsApp and an email link that opens the user's default mail client.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

**Optimized Performance:** Leverages Next.js features like Image Optimization, lazy loading, and Tailwind CSS purging to ensure fast load times and a smooth user experience.

**Secure by Design:** Implements security best practices, including the use of environment variables for secrets, server-side validation, and enforced HTTPS in production.

# Application Workflow

Understanding the data flow is crucial for debugging and extending the application. The primary workflow revolves around the contact form submission process. This section details the step-by-step journey of a user's message from their browser to the database and administrator's inbox.

## Contact Form Submission Flow

The following diagram and steps illustrate the integration between the frontend, backend API, database, and email service.

##### User Interaction (Client-Side)

A user fills out and submits the contact form on the Next.js frontend.

↓

##### API Request

The frontend sends a POST request with the form data to the backend API endpoint:

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

/api/contact/route.ts .

↓

##### Server-Side Validation

The API endpoint receives the data and performs rigorous server-side validation to ensure all fields are correct and sanitized.

↓

##### Database Storage (AWS DynamoDB)

Using IAM credentials, the API connects to AWS and stores the validated submission data in the digilink-contact-submissions

DynamoDB table. A unique ID is generated for each entry.

↓

##### Email Notification (Resend API)

The API then calls the Resend service using the RESEND\_API\_KEY to send a detailed email notification to the designated administrator address ( [apmkhwanazi@gmail.com](mailto:apmkhwanazi@gmail.com) ).

↓

##### Response to Client

The API sends a success response back to the frontend, which then displays a confirmation message to the user.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide

#### Error Handling and Fallbacks

The system is designed to be resilient. If the email notification via Resend fails for any reason (e.g., API downtime), the process does not halt. The submission is still saved to AWS DynamoDB, ensuring no data is lost. The API will log the email failure, which can be monitored, but the primary goal of data capture is always achieved.

# Technology Stack

The project is built on a modern, robust, and scalable technology stack, primarily leveraging the JavaScript/TypeScript ecosystem and AWS cloud services.

1. Interactive Features
2. Design System
3. Monitoring & Maintenance
4. Security
5. Support

**Category**

Framework

Language

**Technology**

**Next.js 14**

**TypeScript**

**Description**

The core React framework, utilizing the App Router for modern routing and server components.

Provides static typing for enhanced code quality, maintainability, and developer experience.

Styling

**Tailwind CSS v4**

A utility-first CSS framework for rapid and

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

**Category**

UI

Components

Database

Email Service

Hosting

**Technology**

**shadcn/ui**

**AWS**

**DynamoDB**

**Resend**

**AWS**

**Amplify + Route53**

**Description**

consistent UI development.

A collection of beautifully designed, accessible, and reusable components.

A fully managed, serverless NoSQL database for storing contact form submissions.

A modern email API for developers, used to send

transactional email notifications.

Provides a complete CI/CD and hosting solution, with DNS management via Route53.

Authentication

**AWS IAM**

Manages secure access between the application backend and AWS services like DynamoDB.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

# Project Structure

The project follows the standard Next.js App Router directory structure. This organization separates concerns and makes the codebase easy to navigate.

.

├── app/

│ ├── api/contact/route.ts # Contact form

│ ├── globals.css # Global style

│ ├── layout.tsx # Root layout

│ └── page.tsx # Main homepag

├── components/

│ ├── header.tsx # Navigation b

│ ├── hero-section.tsx # The main her

│ ├── services-section.tsx # Component to

│ ├── about-section.tsx # Component fo

│ ├── contact-section.tsx # The contact

│ └── footer.tsx # Footer with

├── lib/

│ └── aws-config.ts # AWS SDK and

├── public/

│ └── images/

│ └── fulllogo.png # Company logo

└── README.md # Project docu

# Setup and Configuration

This section provides a comprehensive guide to setting up the project for local development and configuring the necessary cloud services.

## Prerequisites

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

Before you begin, ensure you have the following installed and configured:

**Node.js:** Version 18.x or later.

**npm or yarn:** Package manager for Node.js.

**AWS Account:** An active Amazon Web Services account.

**Resend Account:** A free or paid account at [resend.com](https://resend.com/).

**Git:** For version control.

## AWS Services Setup

The application relies on AWS for database storage and secure access. Follow these steps to configure the required services.

#### Step 1: Create a DynamoDB Table

This table will store all contact form submissions.

* + 1. Navigate to the **DynamoDB** console in your AWS account.
    2. Click **Create table**.
    3. Use the following configuration:

**Table Name:** digilink-contact-submissions

**Partition Key:** id (Type: String)

**Table settings:** Use default settings.

* + 1. Click **Create table**. Make sure to create the table in your desired region (e.g., us-east-1 ).

#### Step 2: Create an IAM User for Programmatic Access

This user will grant the application permissions to access the DynamoDB table.

1. Navigate to the **IAM** console in your AWS account.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support
13. Go to **Users** and click **Create user**.
14. Enter a **User name** (e.g., digilink-app-user ).
15. Select **Provide user access to the AWS Management Console - optional** and choose **I want to create an IAM user**.
16. On the permissions page, select **Attach policies directly**.
17. Search for and attach the following policies: AmazonDynamoDBFullAccess (For broader permissions during development. For production, it's better to create a custom policy with more granular permissions for the specific table).
18. Complete the user creation process.
19. On the final screen, go to the **Security credentials**

tab for the new user and click **Create access key**.

1. Choose **Application running outside AWS** as the use case.
2. Create the key and securely copy the **Access key ID** and **Secret access key**. You will need these for your environment variables.

**Important:** The Secret Access Key is only shown once. Store it in a secure location immediately. Do not commit it to your Git repository.

## Email Service (Resend) Setup

To enable email notifications, you need to configure the Resend API.

* + 1. Sign up for an account at [resend.com](https://resend.com/).
    2. Navigate to the **API Keys** section in your dashboard.
    3. Click **Create API Key**.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support
    * 1. Give the key a name (e.g., Digilink Website Key ) and set permissions to **Full access** for sending emails.
      2. Click **Create** and copy the generated API key. This will be your RESEND\_API\_KEY .

## Local Development Environment

With the cloud services configured, you can now set up the project on your local machine.

#### Step 1: Clone the Repository

git clone <your-repository-url> cd <repository-directory>

#### Step 2: Install Dependencies

npm install

#### Step 3: Create Environment Variables File

Create a file named .env.local in the root of the project and add the credentials you obtained in the previous steps.

# AWS DynamoDB Configuration AWS\_REGION=us-east-1 AWS\_ACCESS\_KEY\_ID=your\_access\_key\_from\_iam

AWS\_SECRET\_ACCESS\_KEY=your\_secret\_key\_from\_iam AWS\_DYNAMODB\_TABLE\_NAME=digilink-contact-submis

# Email Notifications RESEND\_API\_KEY=re\_your\_resend\_api\_key\_from\_rese

Replace the placeholder values with your actual keys and region.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

#### Step 4: Run the Development Server

npm run dev

Open your browser and navigate to [http://localhost:3000](http://localhost:3000/). The website should now be running locally.

#### Step 5: Testing the Contact Form

1. Navigate to the contact section on your local site.
2. Fill out and submit the form.
3. **Check the terminal** where npm run dev is running. You should see debug logs from the API endpoint.
4. **Verify email notification:** Check the inbox of

[apmkhwanazi@gmail.com](mailto:apmkhwanazi@gmail.com) for a new message.

1. **Check DynamoDB:** Go to your AWS DynamoDB console, select the digilink-contact-submissions table, and click on **Explore table items** to verify that the new data has been stored.

# Deployment Guide

The project is configured for easy and automated deployment using **AWS Amplify**.

## AWS Amplify Deployment

Amplify provides a Git-based workflow that automatically builds and deploys your application on every push to a specified branch.

* + 1. **Connect Repository:** In the AWS Amplify console, choose to host a new web app and connect your GitHub (or other Git provider) repository.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support
    * 1. **Configure Build Settings:** Amplify will typically auto-detect the Next.js framework and provide default build settings. Confirm these are correct. The build command should be npm run build and the base directory for artifacts is usually .next .
      2. **Add Environment Variables:** This is a critical step. In the Amplify console, navigate to **App Settings > Environment Variables**. Add all the same variables from your .env.local file:

AWS\_REGION AWS\_ACCESS\_KEY\_ID AWS\_SECRET\_ACCESS\_KEY AWS\_DYNAMODB\_TABLE\_NAME RESEND\_API\_KEY

* + 1. **Deploy:** Save and deploy the application. Amplify will

provision the necessary resources, build the project, and deploy it to a global CDN (Amazon CloudFront).

After the first deployment, any subsequent push to your connected branch (e.g., main ) will trigger a new build and deployment automatically.

## Domain and DNS Setup

**Domain Management:** The primary domain is managed through **AWS Route53**.

**Custom Domain in Amplify:** Within the Amplify console's **Domain management** section, you can add your custom domain from Route53.

**SSL/TLS:** Amplify automatically provisions and manages a free SSL/TLS certificate for your custom domain, ensuring all traffic is served over HTTPS.

**CDN:** All assets are automatically distributed through Amazon CloudFront, providing low-latency access for users worldwide.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

# Interactive Features

The user experience is enhanced by several interactive elements designed for intuitive navigation and easy communication.

## Navigation

**"Discover Our Services" Button:** Located in the hero section, this button smoothly scrolls the user down to the services section of the page.

**"Get Free Consultation" Button:** This primary call-to-action button scrolls the user directly to the contact form for immediate engagement.

## Contact Methods

**Email Link:** The email address [info@digilinkict.co.za](mailto:info@digilinkict.co.za) is a clickable mailto: link that opens the user's default email client with the address pre-filled.

**Phone/WhatsApp Link:** The phone number +27 (0)

11 234 5678 is configured to open WhatsApp, facilitating instant messaging for mobile users.

**Contact Form:** The primary asynchronous contact method that integrates with the backend system as detailed in the Application Workflow section.

# Design System

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

A consistent and professional design system is used throughout the website to maintain brand identity and visual coherence.

## Colors

**Primary:** A professional blue ( hsl(210 100% 50%) ) used for key interactive elements and branding.

**Background:** A warm, off-white ( oklch(98% 0.02 45) ) for a clean and modern feel.

**Accent:** A neural blue ( hsl(220 100% 60%) ) used in animations and highlights.

**Text:** High-contrast grays to ensure readability and accessibility.

## Typography

**Headings: Geist Sans** in multiple weights for a strong and modern typographic hierarchy.

**Body Text: Geist Sans** (regular weight) for clear and legible content.

**Code: Geist Mono** for all code snippets and technical text.

## Animations & Effects

**Neural Network Patterns:** Subtle, animated background patterns that reinforce the AI theme.

**Floating Circuit Elements:** Dynamic visual elements that add depth and interest.

**Smooth Hover Transitions:** All interactive elements have smooth transitions on hover and focus states.

**Glow Effects:** Subtle glow effects on buttons and links to draw user attention.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

# Monitoring and Maintenance

The application includes several mechanisms for monitoring its health and performance.

## Form Submissions

**Primary Storage:** All submissions are reliably stored in the **AWS DynamoDB** table. This should be the single source of truth for all inquiries.

**Notifications:** Real-time email notifications via **Resend** provide immediate alerts. The Resend dashboard can be used to monitor email delivery status.

**Logging:** During local development, detailed logs are printed to the console. In production on AWS Amplify, logs can be accessed through the Amplify console or Amazon CloudWatch for debugging.

**Validation:** Both client-side and server-side validation are in place to ensure data quality and prevent bad submissions.

## Performance

**Image Optimization:** Next.js's built-in <Image> component is used to automatically optimize and serve images in modern formats like WebP.

**CSS Purging:** Tailwind CSS automatically removes unused styles in the production build, resulting in a minimal CSS file size.

**Component Lazy Loading:** Sections or components that are not immediately visible can be lazy-loaded to improve initial page load time.

### Digilink IT Solutions

Project Documentation

1. Introduction
2. Core Features
3. Application Workflow
4. Technology Stack
5. Project Structure
6. Setup & Configuration
7. Deployment Guide
8. Interactive Features
9. Design System
10. Monitoring & Maintenance
11. Security
12. Support

**Responsive Images:** The application serves appropriately sized images based on the user's device viewport.

# Security

Security is a top priority. The following measures are implemented to protect the application and its data.

**Environment Variables:** All sensitive data, such as API keys and AWS credentials, are stored in environment variables and are not hard-coded or committed to version control.

**Server-Side Validation:** All data submitted through the contact form is re-validated on the server to prevent malicious or malformed data from entering the system.

**AWS IAM Roles:** Access to AWS services is controlled by a dedicated IAM user with limited permissions, following the principle of least privilege.

**HTTPS Enforced:** In production, AWS Amplify automatically enforces HTTPS, encrypting all data in transit between the user and the server.

**Input Sanitization:** The backend API should sanitize all inputs to prevent cross-site scripting (XSS) and other injection attacks before storing them in the database.

# Support

For technical support, modifications, or questions regarding this project, please use the following contact information.

**Technical Contact:** [apmkhwanazi@gmail.com](mailto:apmkhwanazi@gmail.com)

**Digilink IT Solutions**

Project Documentation

**Company:** Digilink IT Solutions

**Website:** (Presumably the one this project builds)

1. Introduction

**Project Version:** 1.0.0

**Status:** Production Ready ✅

2. Core Features

3. Application Workflow

4. Technology Stack

Documentation generated on: 2025-09-10. Based on project README file, version 1.0.0.

© 2025 Digilink IT Solutions. All Rights Reserved.

6. Setup & Configuration

5. Project Structure

7. Deployment Guide

[k.ai/prod/analysis/2025-09-10/8752163153518422086/19657470387 ab55ff591cb59c41e6b.md](https://static-us-img.skywork.ai/prod/analysis/2025-09-10/8752163153518422086/1965747038784704512_c47c7c7383225ab55ff591cb59c41e6b.md)

Reference

[1] README

[https://static-us-img.skywor 84704512\_c47c7c7383225](https://static-us-img.skywork.ai/prod/analysis/2025-09-10/8752163153518422086/1965747038784704512_c47c7c7383225ab55ff591cb59c41e6b.md)

9. Design System

8. Interactive Features

10. Monitoring & Maintenance

11. Security

12. Support