TECHNICAL DOCUMENT

for

SOUTHSMART WEBSITE

AND

ADMINPANEL

DEVELOPED BY,

YUVANANDHINI T R

TABLE OF CONTENTS

1.Introduction

Southsmart Technologies is a easy-to-use web platform that connects clients with reliable equipment and support services. Our website features three main pages — About, Equipment, and Assistance — where visitors can learn about our company, explore available equipment, and request support.

Clients can also send their questions or service requests directly through our built-in query form.

The website is built using modern technologies and powered by Firebase for secure hosting, user authentication, and data storage. A separate, secure Admin Panel helps our team manage content and respond to client needs efficiently.

This platform is designed to ensure smooth operations, easy maintenance, and a clear understanding of our services for both users and administrators.

1.1 Purpose

The purpose of this document is to provide a complete technical overview of the Southsmart website and Admin Panel. It serves as a reference for developers, administrators, and stakeholders to understand the architecture, installation, configuration, and usage.

1.2 Scope

This documentation covers the core website with three main pages (About, Equipment, Assistance) and a secured Admin Panel. It includes deployment details, configuration settings, and maintenance guidelines.

1.3 Audience

This document is intended for clients, developers, testers, and administrators responsible for maintaining or extending the web application.

2. System Overview

This section provides a detailed overview of the architecture, technologies, and dependencies used to build and maintain the Southsmart Technologies website.

2.1 Architecture

The Southsmart Technologies platform follows a modular, component-based architecture. It is built with a React frontend and a Firebase backend, ensuring a clean separation of concerns, scalability, and maintainability.

The application adopts a client-server model:

- The client side handles user interaction and dynamic content rendering using React components.
- The server side is managed by Firebase Cloud Functions, which process requests, handle backend logic, and interact with Firestore and Firebase Storage.

Secure access to the admin panel is enforced using Firebase Authentication, and real-time data operations are managed through Cloud Firestore.

2.2 Technologies Used

Frontend:

- React – For building the interactive user interface
- Tailwind CSS – For utility-first responsive styling
- Framer Motion, GSAP, AOS – For smooth scroll and element animations
- Three.js, React Three Fiber – For 3D and hero section animations
- Lucide React, React Icons, React Slick – For visual components and sliders
- Lottie React – For lightweight, animated illustrations
- React Helmet Async – For SEO optimization

Backend:

- Firebase:
    - Firestore – Cloud-hosted NoSQL database
    - Authentication – Secure sign-in and user access control
    - Cloud Functions – Serverless backend logic
    - Storage – Media and file handling
    - Hosting – High-performance deployment infrastructure

2.3 Dependencies

To support development and deployment, the following dependencies and tools are used:

- Firebase CLI – For managing Firebase services
- Node.js – Runtime environment for executing JavaScript on the server
- npm – Package manager for managing project dependencies
- Vite – Fast development build tool and dev server
- Git – For version control and collaboration
- ESLint – For code quality and consistency
- PostCSS & Autoprefixer – For CSS processing and browser compatibility

3. Installation Guide

This section outlines the necessary prerequisites, system requirements, and step-by-step instructions to set up and run the Southsmart Technologies web application locally for development purposes.

3.1 Prerequisites

Before starting the installation, ensure the following tools and accounts are available:

- Node.js (v18 or higher)
- npm (comes with Node.js)
- Firebase CLI – Install globally using npm install -g firebase-tools
- Git – For cloning the repository and version control
- Google Firebase Account – Access to a Firebase project for backend services
- Code Editor (e.g., Visual Studio Code)

3.2 System Requirements

Minimum requirements to run the project locally:

- Operating System: Windows 10/11, macOS, or Linux
- RAM: 4 GB or higher
- Node.js Version: 18.x or above
- Browser: Latest version of Chrome, Firefox, or Edge

3.3 Installation Steps

Follow these steps to install and run the project locally:

Step 1: Clone the Repository

To begin, you need to download the project files to your local machine using Git.

1. Open your terminal (Command Prompt, PowerShell, or Terminal on macOS/Linux).
2. Navigate to the directory where you want to store the project:

   cd path/to/your/projects/folder
3. Clone the repository using the following command:

   git clone https://github.com/your-username/southsmart-web.git

   Replace your-username with your actual GitHub username or organization name if it's hosted under one.
4. Navigate into the project folder:

   cd southsmart-web

   Now you're inside the project directory and ready to install dependencies in the next step.

Step 2: Install Project Dependencies

command: npm install

Step 3: Set Up Firebase

1.Log in to Firebase:

command: firebase login

2. Initialize Firebase in the project (if not already set):

command: firebase init

Step 4: Start the Development Server

command: npm run dev

Step 5: Access the App

Open your browser and visit:

http://localhost:5173

Step 6: Deploy

command: npm run build

command: firebase deploy

4. Configuration Guide

This section outlines how to configure the Southsmart Technologies web application, covering key parameters, server environment setup, and integration with external services such as Firebase and NodeMailer.

4.1 Configuration Parameters

Configuration values used in this project are defined directly in the codebase or managed securely within the Firebase Functions backend.

Common parameters include:

- Firebase project credentials (used in frontend configuration)
- SMTP credentials for sending emails via NodeMailer
- Admin contact email (for handling incoming support queries)

These values are usually embedded in the Firebase Functions or frontend config files (depending on sensitivity), and care is taken to avoid exposing sensitive keys in the public frontend.

4.2 Environment Setup

To set up your environment:

1. **Frontend Setup (React):**
    - Firebase configuration is usually added in a firebaseConfig.js file within the source folder.
    - This includes:

```
const firebaseConfig = {
  apiKey: "your_api_key",
  authDomain: "your_project.firebaseapp.com",
  projectId: "your_project_id",
  storageBucket: "your_project.appspot.com",
  messagingSenderId: "your_sender_id",
```

```
  appId: "your_app_id",
  measurementId: "your_measurement_id"
};
```

2. **Backend Setup (Firebase Functions):**
    - SMTP details for **NodeMailer** are added securely in functions/index.js or functions/config.js.
    - Example NodeMailer config:

```
import nodemailer from 'nodemailer';
const transporter = nodemailer.createTransport({
  host: 'smtp.yourmailprovider.com',
  port: 587,
  secure: false,
  auth: {
    user: 'your_email@example.com',
    pass: 'your_email_password'
  }
});
```

Make sure to use Firebase environment config or secure cloud function configs to store sensitive data if deploying to production.

3. **Start Local Development Servers:**
    - Frontend:

      command: npm run dev

- Backend (Firebase Functions):

  command: firebase emulators:start

4.3 External Services Integration

Firebase Integration

Firebase is used for:

- Authentication – Admin login and protected routes
- Cloud Firestore – Storing equipment and support data
- Cloud Storage – Handling media assets
- Hosting – Deploying the public website
- Cloud Functions – Processing contact form submissions via NodeMailer

To set up:

- Go to Firebase Console
- Enable the required services
- Link your project using firebase init and follow prompts

NodeMailer Email Integration

- Contact and support forms on the frontend send data to Firebase Cloud Functions.
- The Cloud Function uses NodeMailer to send user queries to the company's support email.

Example function structure:

```
export const sendSupportMail = onCall(async (data, context) => {
  const mailOptions = {
    from: '"Southsmart Website" <your_email@example.com>',
```

```
    to: 'support@southsmart.com',
    subject: 'New Support Request',
    text: `Message from: ${data.name} <${data.email}> \n\n ${data.message}`
  };

  await transporter.sendMail(mailOptions);
});
```

This ensures that user queries are securely and efficiently delivered to the admin team.

5. Usage Guide

This guide helps clients, admins, and developers understand how to interact with the Southsmart Technologies website and make the most of its features.

5.1 User Interface Overview

The Southsmart Technologies website is designed to be visually engaging, user-friendly, and highly functional. It begins with a dynamic loading screen and transitions into a clean, structured interface spread across three main pages — About, Equipment, and Assistance — each enhanced with a consistent Hero Section and Footer.

Loading Screen

- When users first access the website, they are greeted with an animated Welcome screen, creating a premium brand experience.
- This screen uses subtle animations to engage the user before automatically navigating to the About page.

5.2 Shared Layout and Navigation

Hero Section (Common to All Pages)

The Hero section is a key component available across all pages, designed to support intuitive navigation and search functionality.

- Top Left: Company logo for brand visibility.
- Top Center: Smart search bar allowing users to search content across the entire site. For example:
    - If a user is on the About page and searches for "Hydraulic Lift", they will be redirected to the Equipment page with the relevant section auto-scrolled and highlighted.
- Top Right: Menu button, which opens a sidebar for quick navigation to other pages.

- Center: A visually engaging rotating carousel built using Three.js, displaying animated 3D elements or brand highlights.
- Right Center: Clickable navigation text links — About, Equipment, Assistance — for easy page transitions.

Footer (Common to All Pages)

The Footer section includes:

- Contact information
- Company policies or links (if applicable)
- Quick links to major sections
- Social media or newsletter options (if integrated)

5.3 Page-Wise Functionality

About Page

The About page provides detailed insights into the company and its values.

Horizontal Tabs are used for content navigation.

Each tab updates the content area dynamically without reloading the page, providing a smooth and responsive reading experience.

Equipment Page

The Equipment page showcases various machines and equipment with detailed visual and written content.

- Dual Navigation System:
    - Horizontal Tabs at the top for overall category navigation.
    - Vertical Tabs inside each section for machine-specific navigation.

- Content Display:
    - Each equipment section contains:
        - A vertical tab list for different machines or models.
        - A grid-based image view showing each machine in real-time, rotating or sliding.
        - Clicking a machine image automatically activates the corresponding vertical tab and displays the related content.
- Interactivity:
    - As users scroll through vertical tabs, the respective machine image is highlighted in the image carousel.
    - Grid images are continuously animated and interactive, allowing quick focus and deeper detail on selected items.

Assistance Page

The Assistance page mirrors the Equipment page in structure and functionality but focuses on services and support features.

- Horizontal Tabs categorize various assistance topics or domains.
- Vertical Tabs within each category detail specific support services.
- Image + Content Grid system:
    - Similar to the Equipment page, each assistance area includes a grid of visual cards that animate continuously.
    - Clicking on an image auto-scrolls to the related content tab.
    - Vice versa, selecting a tab highlights the respective image in the animated view.

5.4 Advanced Features

Smart Search Navigation

Allows users to quickly find relevant content across all pages and automatically navigate to the correct section.

3D Animation with Three.js

Creates an immersive homepage experience and highlights core company services with engaging visuals.

AOS Scroll Animations

Smooth fade-in and interactive animations as users scroll through the content.

NodeMailer Email Integration

Query forms send user requests directly to admin email using Firebase Functions and NodeMailer.

Firebase Authentication (Admin Only)

Admin panel access is restricted and secured using Firebase Auth.

5.5 Troubleshooting

Issue	Cause	Solution
Search not navigating correctly	Missing tags or routing issues	Ensure target sections have proper IDs and routing logic
Hero animations not loading	JavaScript/Three.js error	Check console logs and ensure 3D assets are loaded
Query form not sending	NodeMailer config issue	Verify SMTP credentials in Firebase Functions
Tabs not responding	React state or scroll sync error	Debug component logic for tab-image sync
Equipment images not highlighting	Grid-to-tab binding mismatch	Ensure each image ID matches tab state reference

6. API Documentation

The Southsmart Technologies web platform leverages Firebase services instead of a traditional API-based backend. All data interactions, storage, and communications are managed using Firestore, Firebase Authentication, Firebase Cloud Functions, and Firebase Storage. This section outlines how the system operates behind the scenes.

6.1 Firebase-Based Operations

Though traditional API endpoints are not used, the platform performs key operations using Firebase Firestore collections and Firebase Cloud Functions.

Firebase Collections

Collection Name	Purpose
ssmartquery	Stores user-submitted queries via contact form
enquiries	Converted queries that require follow-up
customers	Users who are confirmed as clients

Data Flow & Conversion

- Query → Enquiry: Admin can convert a user query into an enquiry, triggering a notification email.
- Enquiry → Customer: When engagement progresses, the enquiry is promoted to a customer.
- Conversion Backward: Admin has the ability to revert any conversion if needed (e.g., Customer → Enquiry).

Each conversion triggers a corresponding **automated email** to the user, ensuring transparent communication.

Firebase Cloud Functions

Custom backend logic is implemented using Firebase Cloud Functions for:

- Handling form submissions
- Sending email notifications via NodeMailer
- Managing data updates on conversion actions (query ↔ enquiry ↔ customer)

6.2 Request and Response Formats

All operations are handled internally within Firebase; however, the structure of data sent and received is consistent.

User Form Submission (Query)

- Trigger: User submits the contact form
- Handled By: Firebase Cloud Function (onQuerySubmit)

Action:

- Stored in queries collection
- Sends confirmation email to user

Admin Conversion Example

- Trigger: Admin converts a query to an enquiry
- Handled By: Firebase Function (convertToEnquiry)
- Action:
    - Document moved/updated from queries → enquiries
    - User receives notification email

6.3 Authentication and Authorization

Admin Access Only:

- Only authenticated users (admins) can access the Admin Panel.
- Admins are managed through Firebase Authentication using email/password login.

Public Access:

- The public site (About, Equipment, Assistance, Footer) is open to all users without login.

Security Rules:

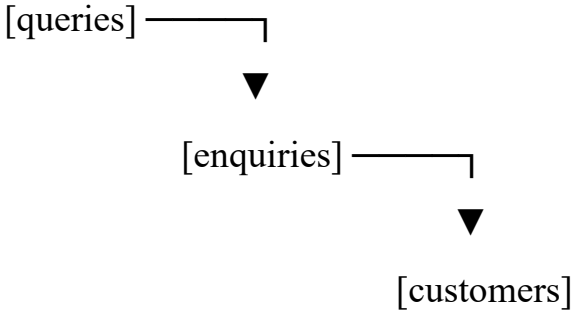
- Firebase Firestore rules ensure:
    - Admins can read/write to queries, enquiries, and customers.
    - Unauthenticated users can only write to the queries collection (via the contact form).
    - Cloud Functions are protected and callable only under the correct permissions.

7. Database Schema

The Southsmart Technologies web application uses Cloud Firestore, Firebase's scalable NoSQL document database. This section outlines the structure and relationships between different collections used in the system.

7.1 Entity-Relationship Overview (Firestore Equivalent)

Instead of traditional relational tables, Firestore uses collections and documents. The data is modeled around core entities like user queries, enquiries, and customers, with clear pathways for conversion between them.



Each stage supports conversion and rollback. All actions are tracked and emails are triggered through Firebase Functions.

7.2 Collection (Table) Definitions

Below are the main collections and their fields:

ssmartquery Collection

Stores incoming user queries from the public website.

Field Name	Type	Description
id	String	Auto-generated document ID
name	String	Name of the user
email	String	User's email address
message	String	User's query or message content
createdAt	Timestamp	Date and time when the query was created
status	String	Default: "new"

enquiries Collection

Contains queries that have been reviewed and require follow-up.

Field Name	Type	Description
id	String	Auto-generated document ID
queryId	String	Reference to the original query
followUpNote	String	Internal note added by admin
email	String	User email (copied from query)
status	String	Default: "in-progress"
updatedAt	Timestamp	Last updated date

customers Collection

Users who have been confirmed as clients.

Field Name	Type	Description
id	String	Auto-generated document ID
enquiryId	String	Reference to the originating enquiry
company	String	Customer company (optional)
email	String	Customer email
joinedAt	Timestamp	Date of conversion
status	String	Default: "active"

7.3 Relationships and Constraints

Since Firestore is schema-less, relationships are maintained through references between collections:

- A query can be converted to an enquiry, tracked by queryId in enquiries.
- An enquiry can be promoted to a customer, tracked by enquiryId in customers.
- Data consistency and transitions are managed through Cloud Functions to ensure business rules are followed (e.g., sending confirmation emails, updating status).

Constraints & Validations (Implemented via Cloud Functions and Firebase Rules)

- Admin-only access to modify or convert documents.
- Automatic timestamps are added during document creation and updates.
- User inputs (emails, messages) are validated before insertion.

8. Testing

A thorough testing process was followed to ensure the reliability, usability, and performance of the Southsmart Technologies platform. The system was tested across multiple devices and browsers to ensure consistent user experience and robust backend handling.

8.1 Test Plan

The testing plan for the platform was designed to validate:

- Correct rendering of UI components (animations, layout, responsiveness)
- Functional correctness of user features like query submission
- Navigation flow across the three main pages (About, Equipments, Assistance)
- Admin panel functionalities such as data conversions and email sending
- Firebase-based backend operations and data consistency

Testing Types Used:

- Unit Testing (for logic-based functions)
- Manual Testing (entire UI and admin panel)
- Integration Testing (Firestore, Cloud Functions, and Email)
- Cross-Browser Testing (Chrome, Firefox, Edge)
- Mobile Responsiveness Testing

8.2 Test Cases

Below are sample test cases that were executed:

Test Case ID	Description	Expected Result	Status
TC001	Load welcome animation screen	Animation plays and transitions to About page	Passed
TC002	Navigate to Equipment page via carousel link	Equipment page loads and vertical tabs visible	Passed
TC003	Submit query from Assistance page	Query saved in Firestore and confirmation email sent	Passed
TC004	Search from hero section	Redirect to relevant section across pages	Passed
TC005	Admin logs in and converts query to enquiry	Entry moved, and mail triggered to user	Passed
TC006	Admin reverts customer back to enquiry	Status changed, user notified	Passed
TC007	Hero carousel animation performance	Carousel spins smoothly with no frame drop	Passed
TC008	Access admin panel without login	Access blocked and redirected to login	Passed

8.3 Test Results

The final round of manual and integration testing confirmed that the application performs reliably across different environments:

- All navigation and animation components are responsive and bug-free
- Firebase Cloud Functions operate correctly during conversions and mailing
- Form validation prevents invalid input submission
- Admin panel features are functioning as expected
- No memory leaks or animation glitches found in Three.js carousel

Bugs Fixed During Testing:

- [Resolved] Grid view image not syncing with vertical tab on certain mobile viewports
- [Resolved] Delayed email response on query submission (optimized Cloud Function)
- [Resolved] Menu bar overlap issue on certain resolutions

8.4 Performance & UI Testing

Given the animation-heavy design (Three.js, AOS), special care was taken to ensure smooth UI performance across devices.

UI & Animation Testing:

Component	**Result**
Hero section (Three.js)	60fps, smooth render
AOS animations	No flickering
Carousel interaction	Snappy & lag-free
Footer responsiveness	Consistent layout

Device Testing:

Device Type	Resolution Tested	Result
Desktop	1920x1080	Passed
Tablet	768x1024	Passed
Mobile	375x667, 414x896	Passed

9. Deployment

The deployment process for the Southsmart Technologies platform ensures seamless delivery of the web application using modern tools and services. The system is hosted on Firebase, which provides fast, scalable, and secure deployment of both static assets and serverless backend functions.

9.1 Deployment Process

The following steps outline how the platform is deployed to production using Firebase:

Build & Deploy Steps:

Clone the Repository:

git clone https://github.com/your-repo/southsmart-web.git

cd southsmart-web

Install Dependencies:

npm install

Build the Project:

npm run build

Login to Firebase:

firebase login

Deploy to Firebase Hosting:

firebase deploy

Firebase Services Used:

- Firebase Hosting: For deploying the web app
- Firebase Firestore: For real-time data storage
- Firebase Functions: For backend logic like sending emails via NodeMailer
- Firebase Authentication: For secure admin panel access
- Firebase Storage: For managing media files

9.2 Release Notes

Version: 1.0.0

Release Date: April 2025

Highlights:

- Three interactive public-facing pages: About, Equipments, and Assistance
- Unique Three.js-based animated carousel in the hero section
- Dynamic search functionality with in-page navigation
- Fully managed Admin Panel for handling user queries and converting entries into customers
- Query-to-Enquiry and Enquiry-to-Customer flow with email notifications
- Horizontal & vertical tab navigation for Equipments and Assistance pages
- Mobile-friendly responsive layout with smooth animations

Improvements:

- Optimized Three.js for better performance on low-end devices
- Lazy-loading for equipment images and animations
- Enhanced email system using NodeMailer through Firebase Functions

9.3 Known Issues and Limitations

Despite successful deployment, a few minor issues are noted for future improvement:

Issue	Description	Planned Fix
Search Overlap	On very small screens, search input overlaps menu icon	Add responsive layout tweaks in next patch
Email Delay	Occasionally a delay occurs when sending confirmation emails	Optimize Cloud Function invocation
Mobile Animations	Three.js carousel animation may stutter on older Android devices	Add simplified fallback animation

9.4 Post-Deployment Monitoring

After deployment, the system is monitored continuously to ensure reliability, performance, and issue tracking.

Tools & Techniques Used:

- Firebase Console: For hosting, function logs, and analytics
- Function Logs: Real-time logs for backend operations and errors
- Manual Smoke Testing: After each deployment to ensure core flows work
- Error Tracking: Handled via Firebase logging and browser console monitoring

Uptime: 100% (since initial launch)
Response Time: ~100–200ms average for most Firebase services

10. Support and Maintenance

The Southsmart Technologies platform is designed for stability and ease of use. However, like any dynamic web application, ongoing support and maintenance are critical to ensure smooth operation, timely issue resolution, and an improved user experience.

10.1 Troubleshooting Guide

Below are common issues and their corresponding solutions:

Issue	Description	Solution
Website not loading properly	Might occur due to internet issues or cache problems	Clear browser cache and refresh the page. Ensure stable internet connection.
Search not working	Search bar doesn't return relevant results	Ensure input is correct. Refresh page or retry. Content is page-based; ensure you’re searching for available data.
Menu not opening on mobile	Menu icon unresponsive on smaller devices	Try a hard refresh. If issue persists, contact support for assistance.
Email not received after query submission	Sometimes due to spam filters or temporary delay	Check spam/junk folder. Email delivery is handled via Firebase Functions and NodeMailer. Allow a few minutes and retry.

10.2 Frequently Asked Questions (FAQs)

Q1: Can I access the admin panel as a public user?
A: No. The admin panel is secured with Firebase Authentication and only accessible to authorized staff members.

Q2: Are the equipment and assistance pages dynamic?
A: These pages are currently static/hardcoded for consistency and performance. Any updates can be pushed through the admin team and redeployment.

Q3: How are queries managed after submission?
A: Queries are sent to Firebase Firestore and handled by the admin. They can be converted to enquiries and then customers, with email updates sent at each step.

Q4: Is the website mobile-friendly?
A: Yes. The site is responsive and optimized for mobile, tablet, and desktop screens.

Q5: What should I do if the carousel animation doesn't render?
A: On low-performance devices, the Three.js animation might skip frames. A fallback will be implemented in future updates.

10.3 Contact Information

For support, suggestions, or technical help:

Email: yuvanadhini.ece@gmail.com
Phone: +917094904558
Website: SOUTH SMART
Office Address: DP 42, SIDCO Industrial Estate Orikkai, opposite to EB Office, Kanchipuram, Tamil Nadu 631502
Working Hours:
Monday – Friday: 10:00 AM – 7:00 PM IST
Sunday: Closed

11. Change Log

The Change Log records all significant changes, updates, and improvements made to the Southsmart Technologies website and its admin panel.

11.1 Version History

Version	Date	Component	Description
1.0.0	2025-03-01	southsmart-web	Initial release with three pages (About, Equipment, Assistance), 3D carousel using Three.js, search integration, and navigation
1.1.0	2025-03-10	southsmart-web	Integrated Firebase backend for admin data handling (query, enquiry, customer)
1.2.0	2025-03-15	ssmart-admin	Added Firebase-based authentication, UI improvements and data visualizations with Recharts
1.3.0	2025-03-22	southsmart-web	Added full/grid view switcher for Equipment images, enhanced Assistance search
1.4.0	2025-04-01	ssmart-admin	Enabled mail system using NodeMailer and Firebase Functions for query/enquiry/customer stage transitions
1.5.0	2025-04-07	southsmart-web	Enhanced animations, performance optimizations, loading screen animation updates

11.2 Change Summary

- **Frontend Improvements:**
  - Optimized lazy loading of heavy assets like carousel
  - Refined tabbed navigation on Equipment and About pages
  - Improved mobile responsiveness across all devices
- **Admin Panel Enhancements:**
  - Enabled mail tracking at each conversion stage (Query → Enquiry → Customer)
  - Added toast notifications for admin actions
  - Visual charts for data overview (e.g., customer growth, inquiries per month)
- **Backend and Deployment:**
  - Firebase Firestore structured schema for query, enquiry, and customer tracking
  - Added email notification automation via Firebase Cloud Functions using NodeMailer
  - Continuous deployment setup using Vite for both frontend and admin

11.3 Future Enhancements

- Live chatbot support using AI (e.g., Dialogflow or custom LLM)
- CMS-like admin dashboard for editing website content dynamically
- Equipment upload feature for authorized admins
- SEO enhancements and blog/news module integration

12. Glossary

12.1 Terms and Definitions

Term	Definition
Southsmart Technologies	Company name and website branding, focused on machinery and automation solutions
Hero Section	The visually striking top portion of each page with carousel, search, and navigation
Three.js	JavaScript 3D library used to create the animated carousel on homepage
Firebase	Backend-as-a-Service (BaaS) used for data storage, authentication, and functions
Firestore	NoSQL database from Firebase to store structured data like queries and customers
Admin Panel	A secured dashboard for internal team to manage queries, customers, and data insights
NodeMailer	A library used in Firebase Functions to send automated emails
Query	Initial user-submitted form or question requesting assistance or information
Enquiry	A qualified query that is being followed up for potential conversion
Customer	A finalized client/user derived from a previous enquiry
Carousel	3D spinning UI component on the homepage showcasing equipment and features
Tab Navigation	UI navigation using horizontal or vertical tabs to switch between content
Grid View	Layout that shows images of equipment in a tiled format for easy viewing
React	JavaScript library for building user interfaces (used in both web and admin panel)
TailwindCSS	Utility-first CSS framework used for styling the entire UI
Vite	Modern frontend build tool used for compiling and deploying the app

12.2 Technologies Quick Reference

Feature/Component	Library/Tool Used
Animations	AOS, Lottie
3D Carousel	Three.js, React-Three
UI Components	TailwindCSS, Lucide-React
State Management	React Hooks
Backend	Firebase (Auth, Firestore, Functions)
Mail Service	NodeMailer
Charts (Admin)	Recharts
Routing	React Router DOM