

# GreenSaaS – Email Verification SaaS Platform

## ◆ Introduction

In today's digital ecosystem, securing user registration and access is a fundamental feature of any SaaS (Software as a Service) platform. Email verification is a common but critical method to validate user identity and ensure legitimate usage.

GreenSaaS is a web-based SaaS platform built to simulate this real-world requirement. It allows users to register with their email, receive a verification link, and unlock access to their dashboard only after successful verification. This project was developed with a full-stack approach, using modern technologies and frameworks.

---

## ◆ Abstract

The purpose of GreenSaaS is to build a robust, user-friendly, and secure user onboarding flow. The project handles everything from form input validation to password encryption, email token generation, and email delivery using SMTP protocols. Once a user registers, a verification email is dispatched with a secure link. This link confirms their identity and enables them to access further features of the platform.

The application not only emphasizes backend logic (such as secure hashing, email token management, and database updates), but also offers a modern and responsive front-end experience using Tailwind CSS and dynamic EJS templating.

---

## ◆ Tools & Technologies Used

- **Frontend:** HTML, Tailwind CSS, EJS (Embedded JavaScript)
  - **Backend:** Node.js, Express.js
  - **Database:** MongoDB (via Mongoose)
  - **Email Service:** Nodemailer with Gmail SMTP
  - **Validation & Security:** bcryptjs, validator, dotenv
  - **Deployment-ready Stack:** Environment variables managed via .env file
- 

## ◆ Steps Involved in Building the Project

### 1. Project Initialization

- Created project folder and initialized with npm init.
- Installed required packages (express, mongoose, bcryptjs, nodemailer, ejs, etc.).

### 2. Database Schema Design

- Created a Mongoose schema for the User model with fields for name, email, hashed password, verification status, and timestamps.
- Implemented validation rules, including regex-based email validation and password length enforcement.

### 3. Frontend UI Development

- Developed a responsive and modern UI using Tailwind CSS and Google Fonts.
- Designed two dynamic states: verified and unverified dashboard messages.
- Implemented dark mode support and device responsiveness.

### 4. Signup Form & Input Validation

- Created the /signup route with full input validation using the validator library.
- Prevented duplicate registrations with database uniqueness checks.
- Hashed user passwords using bcryptjs before saving to the database.

### 5. Email Verification System

- Generated unique verification links using the user's MongoDB \_id.
- Configured Nodemailer to send stylized HTML email using Gmail's SMTP server.
- Used .env to secure email credentials and app URLs.

### 6. Token Handling & Verification Route

- Implemented /verify route to handle query token from the verification email.
- Verified the token by matching user \_id and updating verified: true in MongoDB.
- Rendered success messages using EJS, showing a personalized greeting upon verification.

### 7. Dashboard and Final User Flow

- Created a protected /dashboard route for post-verification access.
- Rendered dynamic user data such as name, email, and verification status.
- Finalized logic to redirect unverified users appropriately (in future iterations, authentication middleware would handle this).

---

#### ◆ Conclusion

GreenSaaS serves as a real-world simulation of user onboarding and verification for SaaS platforms. The combination of a polished UI and secure backend ensures both functionality and user trust. This project not only strengthens backend skills such as database integration, email delivery, and password encryption but also emphasizes good UX and interface development. By combining full-stack technologies with security principles, GreenSaaS is an excellent foundation for building more complex SaaS platforms that require robust authentication and onboarding systems.