



Sarhah_App

A RESTful API for anonymous messaging and user management, built using [Node.js](#), [Express](#), and [MongoDB Atlas](#).
This API supports authentication, account management, and secure message handling.

by Mustafa M. Abdelaziz

GitHub: [MustafaDols](#)

Contents

01. System Overview

High-level architecture and technology stack overview of the Node.js backend system

02. Authentication Module

Detailed breakdown of user authentication, authorization, and security middleware implementation

03. API Endpoints

Comprehensive documentation of RESTful endpoints for user management and messaging services

04. Security & Middleware

Security implementations including rate limiting, CORS, helmet, and data validation strategies

System Architecture Overview

Technology Stack

Built on Node.js with Express.js framework, featuring MongoDB integration, JWT authentication, and comprehensive middleware pipeline for security and validation



Express.js Server

RESTful API design with modular route structure and middleware pipeline for scalable backend services



MongoDB Integration

Database connection management with transaction support and optimized query patterns for user and message data



Authentication System Design

JWT Token System

Stateless authentication with access and refresh tokens for secure user sessions and API access

S

W

Role-Based Access

Granular permission system with Super Admin, Admin, and User roles for secure resource access

Security Middleware

Rate limiting, input validation, and CORS protection against common web vulnerabilities and attacks

T

O

Multi-Provider Auth

Google OAuth integration alongside traditional email/password authentication for flexible user onboarding

User API Endpoints

RESTful User Management

Comprehensive CRUD operations for user accounts with secure authentication and profile management capabilities

Endpoint	Method	Auth Required	Description
/users/signup	POST	No	Register new user account with email validation
/users/signin	POST	No	Authenticate user and return JWT tokens
/users/confirm	PUT	No	Verify email address via confirmation token
/users/logout	POST	Yes	Invalidate user session and tokens
/users/refresh-token	POST	No	Generate new access token using refresh token

Account Management Endpoints

User Profile Operations

Advanced account management including password reset, profile updates, and secure file upload capabilities

Endpoint	Method	Auth Required	Description
/users/update	PUT	Yes	Update user profile information
/users/updatePassword	PUT	Yes	Change account password
/users/resetPassword	PUT	No	Reset password via email token
/users/upload-profile	POST	Yes	Upload profile picture to Cloudinary
/users/delete/:userId	DELETE	Yes	Delete user account permanently

Messaging System API

Real-time Communication

Direct messaging system enabling users to send and retrieve messages with efficient data retrieval patterns

Endpoint	Method	Auth Required	Description
/messages/send/:receiverId	POST	Yes	Send direct message to specific user
/messages/get/:receiverId	GET	Yes	Retrieve conversation history with user
/messages/	GET	Yes	List all user conversations
/messages/unread	GET	Yes	Count unread messages across all conversations

Security Implementation

100%

HTTPS Only

Authentication Security

- Multi-layered security approach with JWT tokens, bcrypt password hashing, and secure session management
- Bcrypt password hashing with salt rounds
- JWT token expiration and refresh mechanism
- Secure HTTP-only cookie storage

15min

Token Expiry

API Protection

- Comprehensive middleware stack preventing common attacks and ensuring data integrity
- Rate limiting per IP and user
- Input validation with Joi schemas
- CORS whitelist configuration

5MB

File Upload Limit

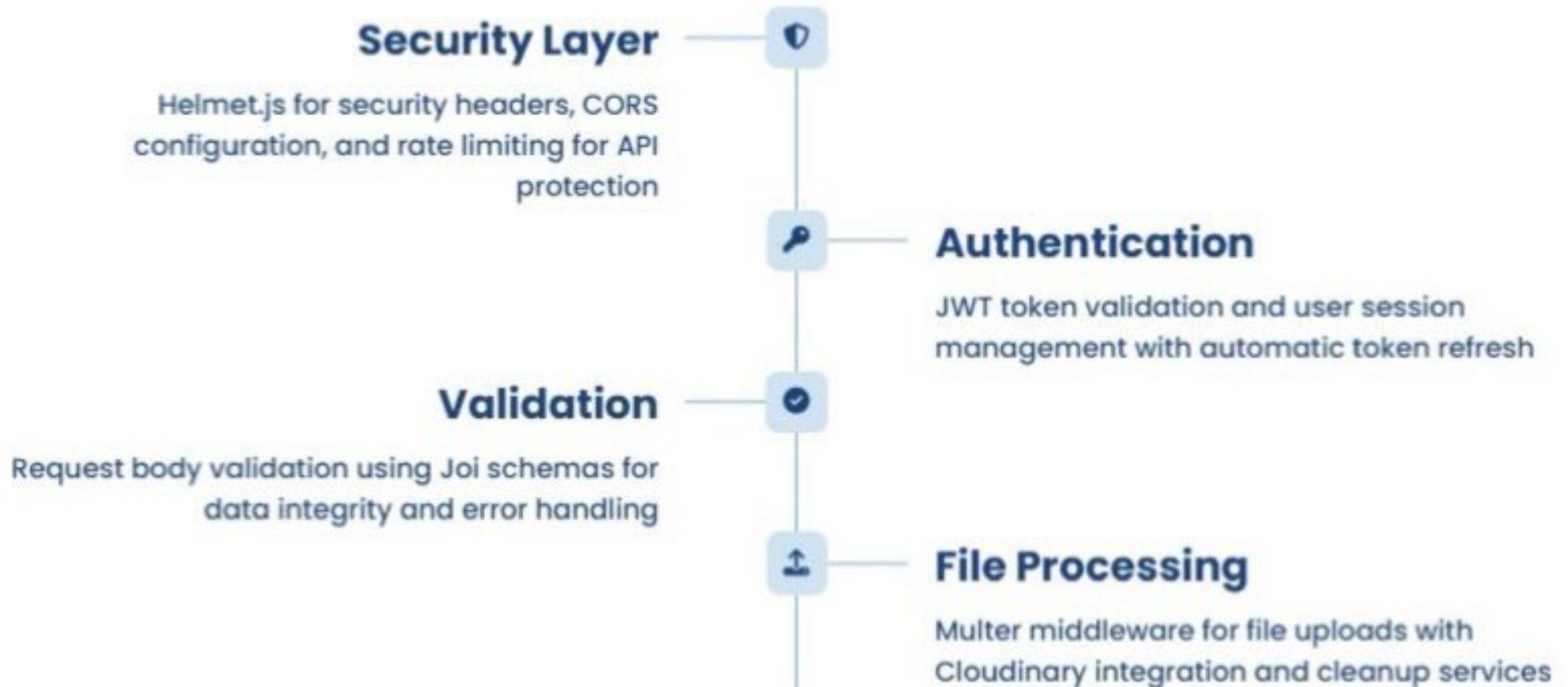
File Security

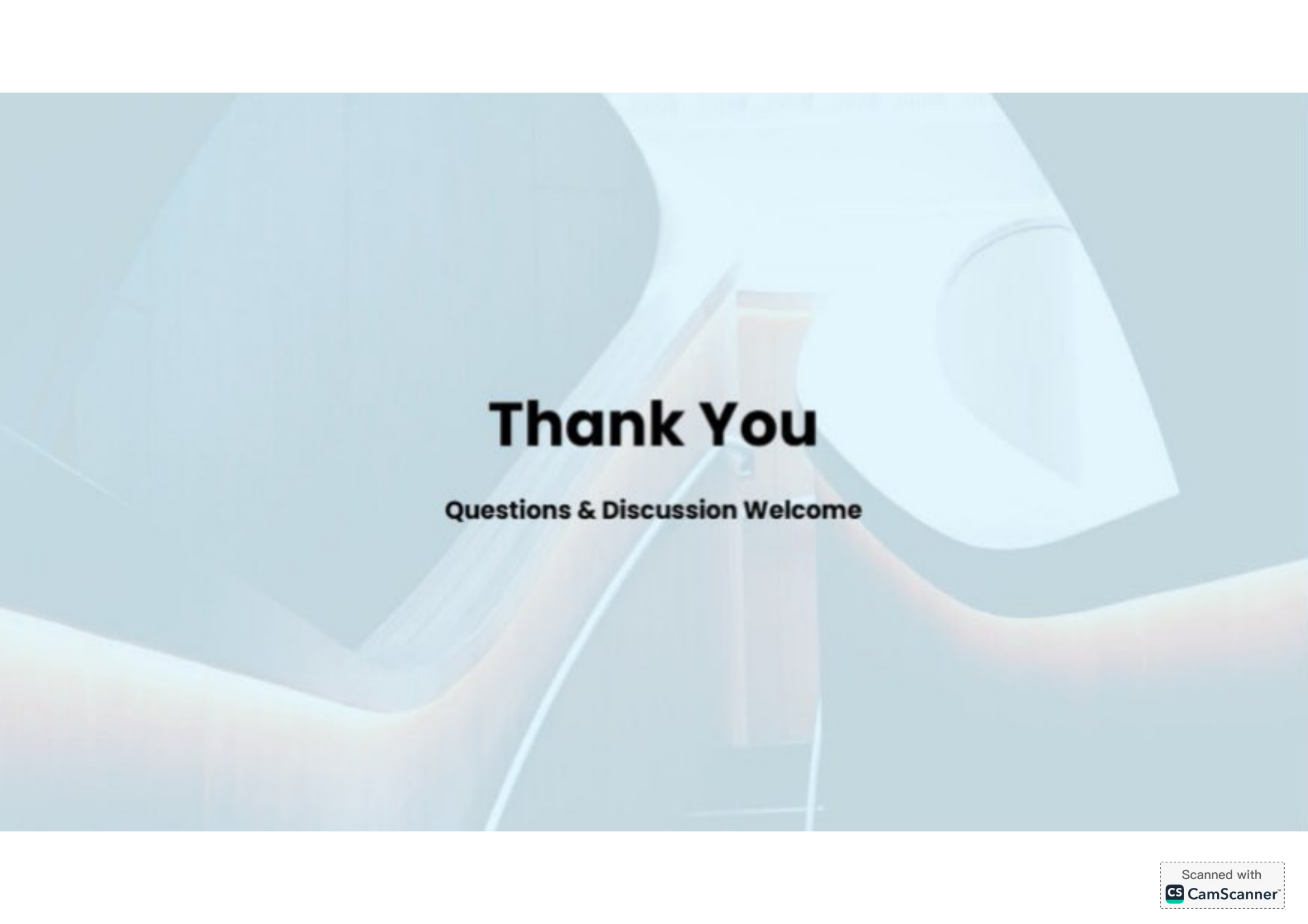
- Secure file upload handling with Cloudinary integration and automatic cleanup
- File type validation and size limits
- Cloudinary secure storage with CDN
- Automatic old file deletion

10req/min

Rate Limit

Middleware Pipeline Flow



The background of the slide is a faded, light blue image showing a hand holding a pen, poised to write on a document. The document has some faint, illegible text and a circular graphic element.

Thank You

Questions & Discussion Welcome