PUBLIC TECHNICAL DOCUMENTATION

Professional Portfolio Platform

A production-deployed cybersecurity portfolio and community platform

Designed, developed, and maintained by

Md Samsudduha Shawon

Release	Version 1.0 - Public edition
Published	August 2026
Live platform	https://meetshawon.com
Source repository	https://github.com/Shawon1024/meetshawon-portfolio
Project status	Active production platform

Cybersecurity / Software Engineering / Infrastructure

DOCUMENT GUIDE

About this documentation

This public document explains the platform's purpose, architecture, capabilities, security approach, and engineering outcomes without exposing operational secrets.

Audience	Recruiters, hiring managers, technical reviewers, collaborators, and interested visitors
Document type	Public technical case study and solution overview
Scope	Application, user experience, data services, security controls, deployment, monitoring, and Drive integration boundary
Excluded	Credentials, private network details, internal paths, recovery secrets, and sensitive operating procedures
Owner	Md Samsudduha Shawon

> **Responsible disclosure:** Cybersecurity content associated with this platform is intended for lawful, authorised, educational, and defensive use. The public documentation intentionally avoids details that would weaken the platform's security posture.

Contents

01 / OVERVIEW

Executive summary

Meet Shawon is a production-deployed professional portfolio and community platform that combines personal branding, technical publishing, account services, moderation, secure communication, and private-cloud integration.

The platform began as a personal portfolio and evolved into a full-stack application designed to demonstrate practical software engineering, cybersecurity awareness, infrastructure integration, and responsible operation. It presents verified professional information while also supporting authenticated community participation and administrative workflows.

The system is built with Next.js, TypeScript, Tailwind CSS, Supabase, Vercel, Cloudflare, Resend, and Sentry. Its public experience includes a responsive portfolio, project case studies, certifications, resume content, a technical blog, professional contact tools, and legal information. Authenticated areas add profiles, reactions, comments, saved content, notification preferences, and account controls. Role-aware administration and moderation support content management, user oversight, appeals, and Drive retention review.

> **Engineering significance:** This is not a static portfolio template. It is an actively maintained production platform with data, identity, communication, monitoring, maintenance, and self-hosted infrastructure boundaries.

Project at a glance

Owner and developer	Md Samsudduha Shawon
Primary purpose	Professional portfolio, technical publishing, community interaction, and practical engineering evidence
Primary domain	meetshawon.com
Application model	Next.js full-stack web application
Data and identity	Supabase authentication, database, and storage services
Deployment	Vercel with Cloudflare-integrated domain and browser security controls
Operational status	Live and continuously improved

02 / CONTEXT

Project background and motivation

The project was created to establish a credible professional presence while providing a real environment for applying software, security, networking, and infrastructure knowledge.

A conventional portfolio can communicate qualifications, but it provides limited evidence of how its owner approaches engineering. This platform was therefore designed to do both: present professional information clearly and demonstrate the ability to design, implement, secure, deploy, monitor, and maintain a multi-feature application.

The problem being addressed

- Professional information was distributed across a CV, social profiles, certificates, code repositories, and project notes.
- Technical learning and infrastructure work needed a structured publishing destination.
- A credible cybersecurity-focused portfolio required more than visual presentation; it needed practical security and operational controls.
- The self-hosted private-cloud project needed a professional web gateway and clear separation from the public portfolio.
- Long-term growth required a maintainable platform that could accommodate new qualifications, content, projects, and services.

Project motivation

The work aligns directly with the developer's progression from a First-Class Computer Science degree into postgraduate Cyber Security Management study. Building the platform created opportunities to connect academic foundations with hands-on work in authentication, data handling, role-based access, browser security, content moderation, deployment, monitoring, and infrastructure integration.

> **Professional intent:** Every major feature was assessed not only for whether it worked, but also for whether it supported a trustworthy, accessible, and sustainable public presence.

03 / DIRECTION

Objectives, scope, and success criteria

The platform was planned as an evolving production system with explicit professional, technical, security, and operational objectives.

Core objectives

1. Create a modern and credible professional identity for cybersecurity and software opportunities.
2. Present education, qualifications, skills, projects, technical writing, and career direction in one coherent platform.
3. Demonstrate full-stack development with authentication, data persistence, server routes, email, and administrative workflows.
4. Apply defence-in-depth principles to browser policy, validation, identity, roles, data access, and operational controls.
5. Integrate the public application with a separate self-hosted private-cloud project without merging their operational boundaries.
6. Establish deployment, health monitoring, error reporting, documentation, and maintenance procedures suitable for ongoing operation.

Included scope

- Public portfolio and project case studies
- Resume, education, certifications, and skills presentation
- Technical publishing with categories, tags, search, and engagement
- User identity, profiles, account preferences, and recovery
- Comments, reactions, saved content, notifications, and appeals
- Administration, moderation, and Drive lifecycle review
- Contact and double opt-in newsletter communication
- Deployment, monitoring, health checks, legal pages, and maintenance modes

Success criteria

- A consistent, professional experience across desktop and real mobile devices
- Successful linting, type checking, and production builds
- Correct anonymous, authenticated, moderator, and administrator behaviour
- Secure handling of secrets through environment configuration
- Production availability through the canonical domain with expected redirects and headers
- Clear documentation and a stable release baseline for future development

04 / CAPABILITIES

Platform capabilities

The application brings professional presentation and functional platform services together in one coherent user experience.

Portfolio	Responsive homepage; About journey; skills; certifications; resume; contact; privacy and terms
Projects	Portfolio platform, self-hosted private cloud, and cybersecurity home-lab case studies
Publishing	Article management, Markdown rendering, syntax highlighting, categories, tags, search, and discovery
Engagement	Views, reactions, comments, bookmarks, sharing, related content, and popular articles
Accounts	Registration, authentication, recovery, profiles, preferences, notifications, activity, and security controls
Administration	Posts, users, comments, roles, verification, appeals, moderation, and history
Communication	Turnstile-protected contact form and double opt-in newsletter workflows
Operations	Health endpoint, monitoring, structured maintenance modes, documentation, and release management
Infrastructure	Role-protected gateway and lifecycle review for the separately operated private Drive

Design characteristics

- Dark, security-oriented visual identity using green, cyan, blue, and silver accents
- Reusable components and consistent interaction patterns
- Responsive layouts designed for professional readability rather than decorative excess
- Clear navigation for anonymous, authenticated, moderator, and administrator contexts
- Accessible labels, focus states, alternative text, and semantic page structure
- Explicit empty, loading, success, and error states across interactive areas

05 / ARCHITECTURE

Solution architecture

The platform uses managed application and data services for the public web experience while maintaining a deliberate boundary around the self-hosted Drive service.

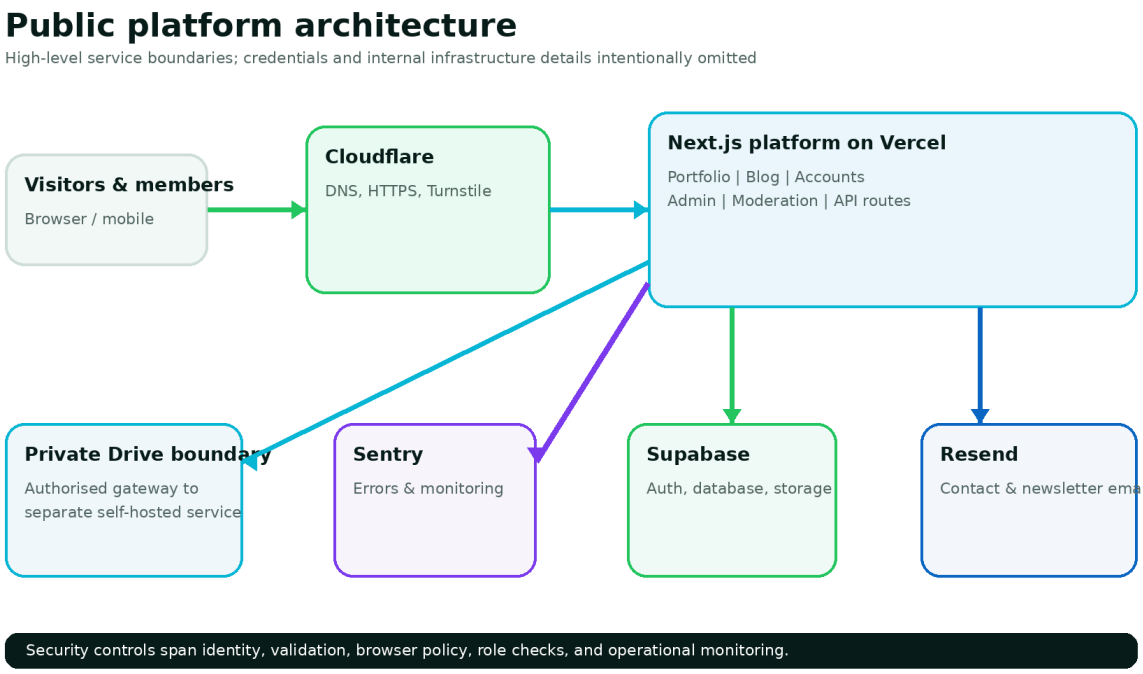

Figure 1. Public high-level architecture of the Professional Portfolio Platform.

Request and service flow

1. Visitors reach the canonical domain through Cloudflare-managed DNS and HTTPS-facing controls.
2. Vercel serves the Next.js application, including public pages, authenticated areas, and server route handlers.
3. Supabase supplies identity, relational data, and configured storage services.
4. Resend supports contact and newsletter communication; Cloudflare Turnstile provides human-verification tokens.
5. Sentry records application errors and operational signals for diagnosis.
6. The Drive hostname and gateway remain logically separate from the portfolio and route authorised users toward the self-hosted service boundary.

> **Boundary principle:** The public application can coordinate identity and access decisions without exposing the underlying storage topology or treating the home infrastructure as part of the public application runtime.

06 / STACK

Technology decisions

Technology choices were driven by maintainability, type safety, managed identity and data services, deployment efficiency, and integration with the wider infrastructure project.

Layer	Technology	Engineering role
Application	Next.js 16 / React	App Router pages, server rendering, route handlers, metadata, and production optimisation
Language	TypeScript	Typed data contracts and safer component/server logic
Interface	Tailwind CSS / Lucide	Responsive styling, reusable visual patterns, and accessible iconography
Identity and data	Supabase	Authentication, PostgreSQL data, storage, and controlled server integration
Deployment	Vercel	Automated production builds, previews, hosting, and runtime logs
Edge and protection	Cloudflare	DNS, HTTPS-facing services, Turnstile, and Drive connectivity
Email	Resend	Transactional contact and newsletter messages
Monitoring	Sentry	Error reporting and operational visibility
Content	React Markdown / Highlight.js	Technical article rendering and code syntax presentation
Infrastructure	TrueNAS SCALE / ZFS	Separate resilient storage platform accessed through the Drive boundary

Decision rationale

The selected stack balances practical learning with production suitability. Next.js consolidates public pages, authenticated experiences, and server endpoints. TypeScript reduces ambiguity across a growing codebase. Supabase avoids rebuilding fundamental identity and database

services while still requiring careful policy and role design. Vercel provides a direct deployment path for the Next.js application, while Cloudflare, Resend, and Sentry add focused capabilities without forcing those responsibilities into the application itself.

07 / EXPERIENCE

Experience and information architecture

The interface is organised around what a recruiter, technical reviewer, reader, registered member, or administrator needs to accomplish.

Public journey

- Understand professional focus and availability from the homepage
- Review education, values, technical direction, and experience
- Inspect project evidence and supporting technologies
- Verify certifications and access public certificate documents
- Read or download the current CV
- Discover technical articles through search, filters, categories, and tags
- Make professional contact or subscribe to confirmed newsletter updates

Member journey

- Create and confirm an account
- Maintain a public profile and preferences
- React to, comment on, save, and share articles
- Review activity and notifications
- Manage account security and deletion options
- Access eligible Drive workflows where authorised

Professional visual system

The visual identity uses a dark green foundation, restrained gradients, bright green action colour, cyan and blue technical accents, white typography, and silver logo details. Components use consistent radii, borders, spacing, hover feedback, and focus treatment. The result is recognisably cybersecurity-oriented without relying on exaggerated visual effects.

> **Mobile principle:** Responsive behaviour is treated as a core requirement. Navigation, account controls, article cards, forms, and project content were tested for narrow screens and real-phone interaction rather than desktop appearance alone.

08 / IDENTITY

Identity, access, and community

Identity functionality supports professional participation while maintaining distinct anonymous, member, moderator, and administrator responsibilities.

Account capabilities

- Email-based registration and authentication
- Email confirmation and account-recovery workflows
- Profile image, name, username, biography, and visual profile controls
- Public profile routes based on username
- Notification and content preferences
- Saved content and account activity
- Account security and deletion controls

Role-aware access

Anonymous visitor	Public portfolio, projects, published articles, contact, legal pages, and sign-in/register access
Registered member	Profile, engagement, saved content, notifications, and account controls
Moderator	Authorised moderation workflows and history appropriate to the assigned role
Administrator	Content, users, comments, appeals, role/verification, and Drive-retention oversight

Administrative interfaces present a user's profile image, human-readable name, username, verification state, and role to reduce ambiguity during sensitive actions. Protected pages verify the current session and required role rather than relying on navigation visibility alone.

09 / CONTENT GOVERNANCE

Publishing, administration, and moderation

The publishing system supports structured technical communication, while administration and moderation provide the controls needed to operate community features responsibly.

Technical publishing

- Administrative creation and editing of articles
- Draft, scheduled, and published content states where configured
- Markdown body content and syntax-highlighted code
- Cover image and alternative-text management
- Categories, tags, excerpts, slugs, and publication metadata
- Search, filtering, related content, and popularity-based discovery
- Views, reactions, comments, and bookmarks displayed as engagement evidence

Moderation and accountability

- Comment review and removal
- Content and user moderation workflows
- Account restriction handling
- Member appeal submission and administrative review
- Reviewer identity and response recording
- Moderation history for operational accountability

> **Governance principle:** Community functionality is paired with oversight. Engagement features were not treated as isolated interface elements; they were designed alongside administrative review, user context, and appeal mechanisms.

10 / COMMUNICATION

Newsletter, contact, and communication

Public communication features are designed to be professional, consent-aware, and resistant to automated abuse.

Contact workflow

1. The visitor submits a labelled professional enquiry form.
2. Client-side checks provide immediate usability feedback.
3. Cloudflare Turnstile supplies a human-verification token.
4. The server route validates the request and verification result.
5. Resend delivers the enquiry through the configured professional email channel.
6. The visitor receives a clear success or error state without exposing internal details.

Newsletter workflow

1. The visitor enters an email address and explicitly confirms consent.

2. Turnstile verification is required before the subscription request is processed.
3. A pending subscriber record is created or updated as appropriate.
4. A signed confirmation link is sent by email.
5. The confirmation endpoint verifies the token before activating the subscription.
6. A dedicated unsubscribe workflow allows consent to be withdrawn.

Newsletter routes are deliberately unavailable during full maintenance. The urgent contact route remains available so important enquiries can still reach the platform owner while non-essential processing is paused.

11 / INFRASTRUCTURE BOUNDARY

Private Drive integration

The Drive integration connects the portfolio's identity layer with a separate self-hosted storage project while preserving clear operational separation.

Integration goals

- Provide a professional entry point for authorised Drive users
- Use the website identity and role context to control eligibility
- Support individual user storage and quota workflows
- Present clear access-denied and account-state experiences
- Allow administrators to review suspended accounts and retention deadlines
- Keep the storage system operationally independent from the public web deployment

Separation of responsibilities

Portfolio platform	Identity context, eligibility, gateway pages, lifecycle records, and administrative review
Drive worker	Controlled background coordination between approved application requests and infrastructure services
Self-hosted service	File access, storage quotas, user storage, and resilient data services
TrueNAS / ZFS	Underlying storage, dataset organisation, resilience, monitoring, and recovery planning

> **Public-documentation limit:** Exact service credentials, internal network addressing, storage paths, worker secrets, and destructive lifecycle procedures are intentionally excluded.

12 / ASSURANCE

Security and privacy engineering

Security is implemented through multiple complementary controls rather than a single feature or product.

Application and browser controls

- Content Security Policy restricting executable, connection, frame, image, font, and media origins
- Strict transport and browser hardening headers in production
- Frame-ancestor restrictions and explicit denial of framing
- Restricted browser permissions for camera, microphone, and geolocation
- Same-origin opener policy and referrer controls
- Server-side validation for contact, newsletter, and privileged operations
- Turnstile verification on abuse-sensitive public forms

Identity and data controls

- Supabase session validation for protected routes
- Role checks for administrative and moderation access
- Row-level and ownership-oriented data policies where applicable
- Server-only use of privileged credentials
- Public environment values separated from server secrets
- Signed, purpose-bound newsletter tokens with expiry handling
- No secrets committed to the public repository; repository history scanned for recognised leaks

Privacy approach

- Public profiles display only information intended for public presentation
- Newsletter consent and subscription lifecycle are explicit
- Privacy and Terms pages explain platform use and expectations
- Administrative interfaces expose only the context necessary for authorised work
- Public documentation excludes private operational details

> **Security posture:** The platform is security-conscious and continuously reviewed; it does not claim that any evolving application is invulnerable. Monitoring, updates, testing, and responsible maintenance remain part of the control model.

13 / OPERATIONS

Deployment, monitoring, and maintenance

Operational controls make it possible to deploy confidently, detect problems, communicate service state, and restore normal operation safely.

Deployment workflow

1. Changes are reviewed locally through Git status and diff inspection.
2. Linting, TypeScript production build, and whitespace checks are completed.
3. Explicit files are staged and committed with a descriptive message.
4. The main branch is pushed to GitHub and deployed through Vercel.
5. Production routes, health response, redirects, and security headers are verified.
6. The stable production baseline is recorded through tagged GitHub releases.

Operational visibility

- Vercel build and runtime logs
- Sentry application error monitoring
- A dedicated health endpoint
- Supabase service and data visibility
- Resend delivery diagnostics
- Cloudflare and Turnstile operational signals
- GitHub dependency and security alerts

Maintenance states

Normal	Full platform available; optional general site notice
Scheduled	Full platform remains available with maintenance timing and expected-return notice
Maintenance	Main pages return a maintenance experience; urgent contact and health checks remain available; newsletter processing pauses

The Drive domain is excluded from the main application's maintenance gate because it is a separate service boundary. Its actual availability still depends on the independent infrastructure environment.

14 / QUALITY

Quality assurance and validation

Quality was assessed through automated checks, route verification, browser inspection, and role-specific functional testing.

Automated and build checks

- ESLint with zero unresolved errors or warnings at the release baseline
- TypeScript checking during the production build
- Optimised Next.js production compilation
- Static and dynamic route generation verification
- Git whitespace and patch-integrity checks
- Repository secret-history scan using Gitleaks

Functional verification

- Public route HTTP status checks
- Canonical-domain and www redirect behaviour
- Health endpoint response
- Security-header inspection
- Authentication, account recovery, and role-aware routing
- Blog publishing, discovery, engagement counts, and image behaviour
- Contact and newsletter confirmation/unsubscribe paths
- Scheduled and active maintenance behaviour
- Drive gateway access and administrative retention presentation

Experience verification

- Desktop and mobile layout inspection
- Real-phone navigation and independent menu scrolling
- Keyboard-focus visibility and meaningful labels
- Image alternative text and responsive image sizing
- Loading, empty, error, success, and disabled states
- Largest Contentful Paint image-priority adjustments where observed

15 / PROBLEM SOLVING

Engineering challenges and solutions

Several issues required diagnosis across framework behaviour, browser security, responsive interaction, data presentation, and operational workflow.

Mobile navigation

Challenge. The menu could become inaccessible when opened after the page had been scrolled.

Resolution. The menu was redesigned as a viewport-controlled layer with background scroll locking, independent overflow, and safe-area handling.

Local mobile CSS

Challenge. Production-oriented HTTPS upgrading interfered with HTTP development testing on a real phone.

Resolution. The insecure-request upgrade directive was retained for production but omitted deliberately in development.

Blog engagement

Challenge. Cards originally showed views without reaction or comment totals.

Resolution. Queries and interfaces were extended to provide complete counts with explicit zero states.

Image performance

Challenge. The above-the-fold blog image generated LCP priority warnings.

Resolution. Loading priority was applied selectively to the genuine leading image instead of every card.

Identity consistency

Challenge. Legacy display-name fields remained in administrative interfaces after the profile model evolved.

Resolution. Admin and moderation areas were normalised around profile image, human name, and username.

Maintenance continuity

Challenge. A simple notice was insufficient for planned outages and essential-service exceptions.

Resolution. A three-mode operating model was added with 503 responses, retry/no-index behaviour, urgent contact, and Drive separation.

Newsletter integrity

Challenge. Subscription needed consent, human verification, confirmation, and safe repeated states.

Resolution. A double opt-in workflow with signed purpose tokens, expiry checks, status transitions, and unsubscribe support was implemented.

Outcomes and competencies demonstrated

The completed platform provides evidence across software engineering, cybersecurity awareness, infrastructure, communication, and responsible operations.

Delivered outcomes

- A live, branded professional platform under a long-term custom domain
- A maintainable multi-route application with public and protected experiences
- A technical publishing and community system with operational governance
- A documented maintenance and recovery approach
- A secure communication workflow for contact and newsletter updates
- A practical identity bridge to a separate self-hosted private-cloud project
- A public repository with branch protection, release tagging, documentation, and clean secret scanning

Technical competencies

Software engineering	Component architecture, typed interfaces, server/client boundaries, route handlers, state management, and reusable design
Cybersecurity	Defence in depth, validation, role checks, RLS awareness, CSP, secure headers, secret separation, and responsible disclosure

Data and identity	Authentication flows, profiles, relational content, engagement data, subscriber state, and administrative context
Infrastructure	DNS, deployment, service boundaries, self-hosted storage integration, monitoring, and maintenance
Quality	Linting, type checks, production builds, HTTP verification, mobile testing, and regression diagnosis
Professional practice	Documentation, public communication, legal pages, release management, and continuous improvement

> **Portfolio value:** The project demonstrates the ability to take an idea from requirements and visual identity through implementation, production deployment, troubleshooting, documentation, and ongoing operation.

17 / EVOLUTION

Limitations and roadmap

The platform is stable for its current purpose but remains an evolving personal system rather than a finished commercial product.

Current limitations

- Automated test coverage should continue to expand beyond build and manual functional verification.
- Accessibility assessment should continue with broader assistive-technology and real-user testing.
- The cybersecurity home-lab project remains in development and should not be overstated before operational evidence exists.
- Drive resilience, recovery testing, and user-facing functionality remain areas of continued engineering work.
- Community and newsletter operations require continued privacy, moderation, and deliverability review as usage grows.

Planned development

- Publish additional technical articles and project reports
- Add newly completed and publicly verifiable credentials
- Update the CV and professional narrative as experience develops
- Replace the home-lab placeholder with genuine architecture, exercises, and results

- Increase automated testing and accessibility evidence
- Continue application, dependency, and policy security reviews
- Strengthen Drive recovery workflows and operational documentation
- Maintain public documentation alongside future tagged releases

18 / CLOSING

Conclusion

The Professional Portfolio Platform demonstrates how a personal website can become a credible engineering system when presentation, functionality, security, infrastructure, and operations are treated as one design problem.

The result is a platform that communicates professional direction while providing concrete technical evidence. Its value comes not only from the features visible to visitors, but from the engineering discipline behind them: typed application logic, identity and data controls, community governance, email verification, browser hardening, monitoring, maintenance planning, release management, and a carefully bounded connection to self-hosted infrastructure.

The project will continue to evolve alongside further cybersecurity study, technical qualifications, infrastructure work, and professional experience. Future changes will be evaluated against the same principles that shaped the initial production release: accuracy, security consciousness, accessibility, maintainability, and responsible public communication.

> **Final perspective:** Meet Shawon is both a professional platform and a continuing engineering record - a place where qualifications, projects, technical writing, and operational learning are connected through a real production system.

Public project links

Live platform	https://meetshawon.com
Project case study	https://meetshawon.com/projects/professional-portfolio
Source repository	https://github.com/Shawon1024/meetshawon-portfolio
Technical blog	https://meetshawon.com/blog
Professional contact	https://meetshawon.com/contact
Email	contact@meetshawon.com

Glossary

CSP	Content Security Policy - browser rules controlling permitted content and connection sources
RLS	Row Level Security - database policies that restrict which records a caller may access
SSR	Server-side rendering - producing page output on the server for a request
ZFS	A storage filesystem and volume-management technology used in the private-cloud project
Double opt-in	A subscription process requiring an email owner to confirm before activation
503	HTTP status indicating temporary service unavailability, used during active maintenance