VIASOFT.AI
VIA Inc.

DEVELOPER

COLLABORATIVE DEVELOPMENT PLATFORM

A-SW HUB

Integrated Collaborative Platform for Autonomous Agricultural Machinery Software Development

---

PROJECT

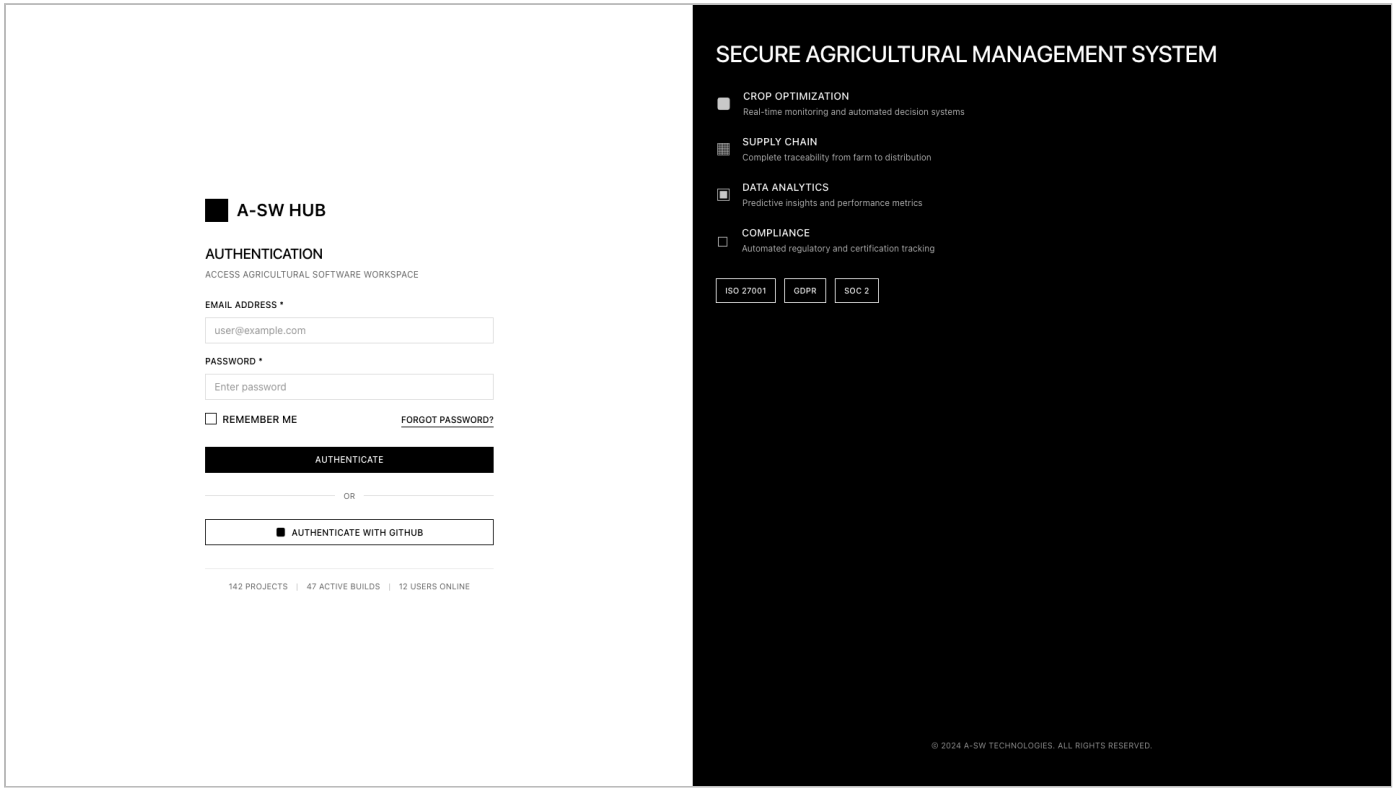
Open-Source A-SW and Collaborative Development Service Platform for Digital Transformation of Agricultural Machinery

SUB-PROJECT

Development and Demonstration of Open A-SW for Unmanned Autonomous Agricultural Operations

TABLE OF CONTENTS

Login



PURPOSE

Secure user authentication gateway for accessing the A-SW Hub platform with role-based credentials and session management.

KEY COMPONENTS

- Email input field with HTML5 validation and autocomplete
- Password input field with visibility toggle and autocomplete
- Remember me checkbox for session persistence

Login

OVERVIEW

The login page provides a secure authentication interface for all A-SW Hub users with a distinctive split-screen industrial design. The left side features a clean white form area with email/password authentication, while the right side displays a black pitch area showcasing the platform's key features. The page supports email-based login, password authentication, GitHub OAuth integration, and includes session persistence through the 'Remember me' option. Real-time platform statistics (142 projects, 47 active builds, 12 users online) are displayed at the bottom. The design follows the extreme Black & White industrial aesthetic with ASCII iconography and zero border-radius throughout.

PURPOSE

Secure user authentication gateway for accessing the A-SW Hub platform with role-based credentials and session management.

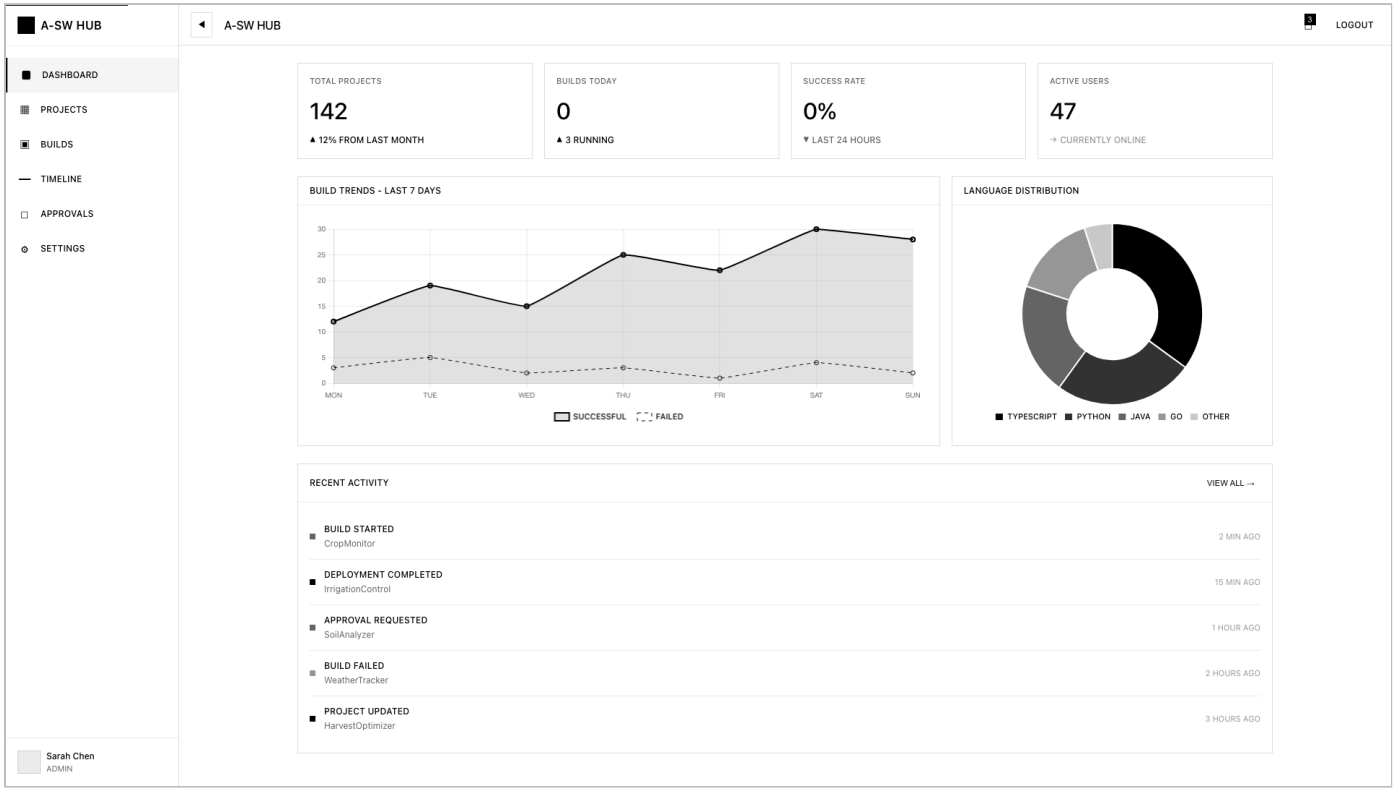
KEY FEATURES

- Secure credential validation with authStore integration
- Session token generation and management
- Role-based access control (Admin, Developer, Viewer)
- GitHub OAuth authentication support
- Session persistence with localStorage integration
- Real-time authentication status feedback with loading states
- Responsive design with mobile optimization (pitch area hidden on mobile)
- Industrial B/W design system with consistent typography
- Comprehensive form validation and error handling
- Automatic navigation to dashboard on successful login

COMPONENTS

COMPONENT
Email input field with HTML5 validation and autocomplete
Password input field with visibility toggle and autocomplete
Remember me checkbox for session persistence
Primary authentication button with loading state
GitHub OAuth integration button with ASCII icon
Platform statistics footer (projects, builds, online users)
Split-screen layout: white form area (left) + black pitch area (right)
A-SW HUB logo with geometric block icon
Forgot password link with underline styling
Security badges display (ISO 27001, GDPR, SOC 2)
Feature showcase cards (Crop Optimization, Supply Chain, Data Analytics, Compliance)
Real-time error message display with monochrome styling

Dashboard



PURPOSE

Centralized overview of all projects, builds, issues, and team activities across the A-SW Hub platform with real-time statistics and interactive visualizations.

KEY COMPONENTS

- Four statistics cards with trend indicators (up/down/neutral arrows)
- Line chart displaying build trends (successful vs failed) over 7 days using Chart.js
- Doughnut chart showing language distribution across projects

Dashboard

OVERVIEW

The dashboard serves as the main landing page after successful authentication, providing a comprehensive overview of platform activities. It displays four key statistics cards at the top (Total Projects: 142, Builds Today with dynamic counts, Success Rate percentage, Active Users: 47), followed by two Chart.js visualizations arranged in a 2-column grid. The first chart shows build trends over the last 7 days with both successful and failed builds tracked, while the second displays a doughnut chart of programming language distribution (TypeScript: 35%, Python: 25%, Java: 20%, Go: 15%, Other: 5%). Below the charts, a real-time activity feed shows the 5 most recent platform activities including build starts, deployment completions, approval requests, build failures, and project updates, each with status-specific indicators.

PURPOSE

Centralized overview of all projects, builds, issues, and team activities across the A-SW Hub platform with real-time statistics and interactive visualizations.

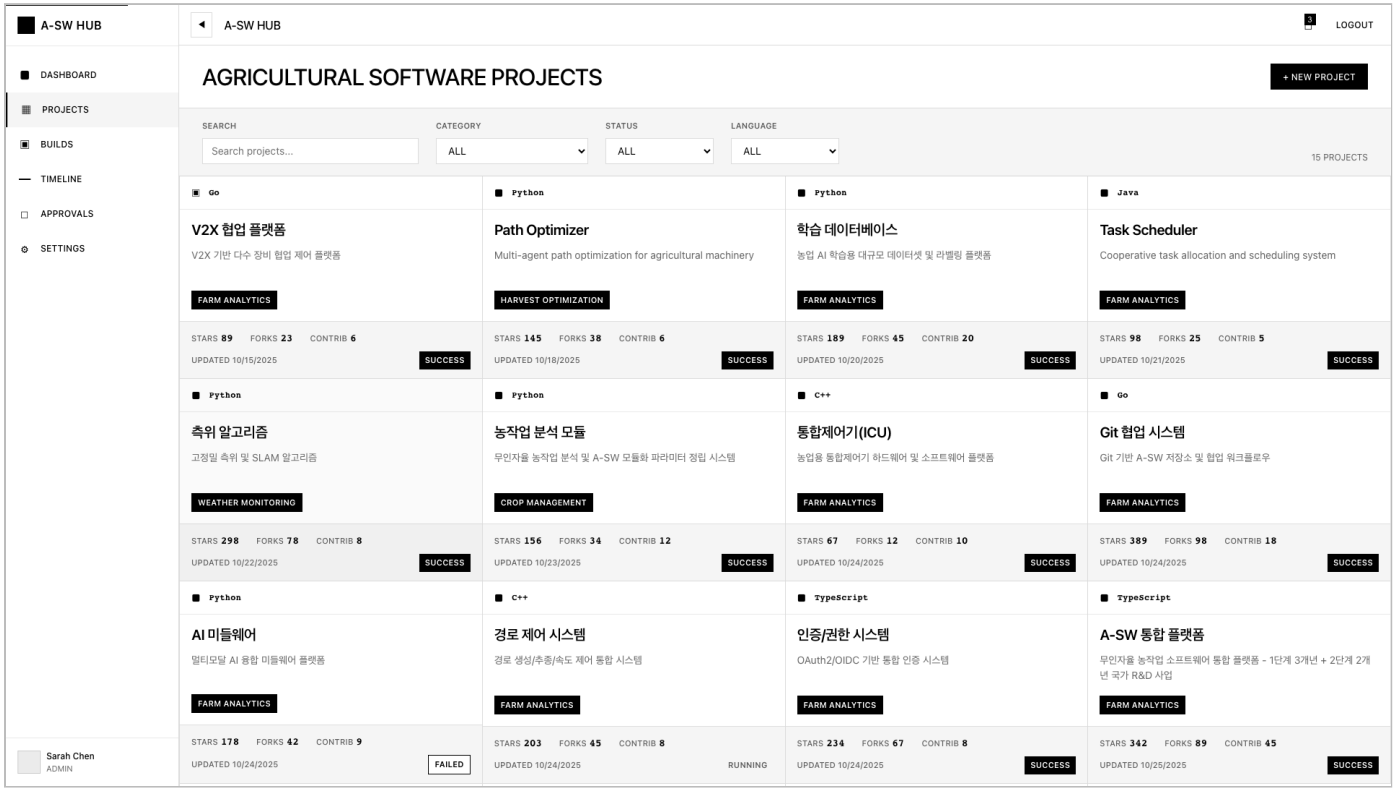
KEY FEATURES

- Real-time data updates from buildsStore and projectsStore
- Role-based content filtering based on user permissions
- Interactive Chart.js visualizations with monochrome styling
- Build metrics calculation (total, running, failed, success rate)
- Activity feed with status-based color coding and icons
- Automatic chart cleanup on component destruction
- Dynamic statistics based on actual build data
- Responsive design with grid collapsing on smaller screens
- Hover interactions on charts for detailed data points
- Background pulse animation for running builds in activity feed

COMPONENTS

COMPONENT
Four statistics cards with trend indicators (up/down/neutral arrows)
Line chart displaying build trends (successful vs failed) over 7 days using Chart.js
Doughnut chart showing language distribution across projects
Real-time activity feed with 5 recent activities
Activity status indicators (running: pulsing animation, success: solid, pending: hollow, failed: subtle)
Panel component wrapper with title and optional actions slot
View All button for accessing complete activity history
Stat component with label, value, trend direction, and trend description
Monochrome chart styling (black lines, fills, and labels)
Responsive grid layout (charts stack on mobile <1024px)

Projects List



PURPOSE

Comprehensive directory of all agricultural software projects with advanced filtering, search capabilities, and grid-based visualization for easy project discovery and navigation.

KEY COMPONENTS

- Page header with title and 'NEW PROJECT' button
- Filters bar with search input (300px width with auto-complete)
- Category dropdown filter with 12 options (all + 11 project categories)

Projects List

OVERVIEW

The projects list page displays all autonomous software projects in a structured grid layout with industrial design aesthetics. The page header includes the title 'AGRICULTURAL SOFTWARE PROJECTS' and a prominent 'NEW PROJECT' button. A comprehensive filters bar provides search functionality with autocomplete, category dropdown (11 categories including crop-management, soil-analysis, irrigation, etc.), status filter (active, maintenance, archived, deprecated), and programming language filter. The main content area uses a responsive grid (auto-fill, 400px minimum) to display project cards, each showing project status with ASCII indicators (■ for active, ▣ for maintenance, □ for archived, ▢ for deprecated), programming language in monospace font, project name, description (truncated to 2 lines), category badge, and footer statistics (stars, forks, contributors). Each card is clickable and navigates to the project detail page.

PURPOSE

Comprehensive directory of all agricultural software projects with advanced filtering, search capabilities, and grid-based visualization for easy project discovery and navigation.

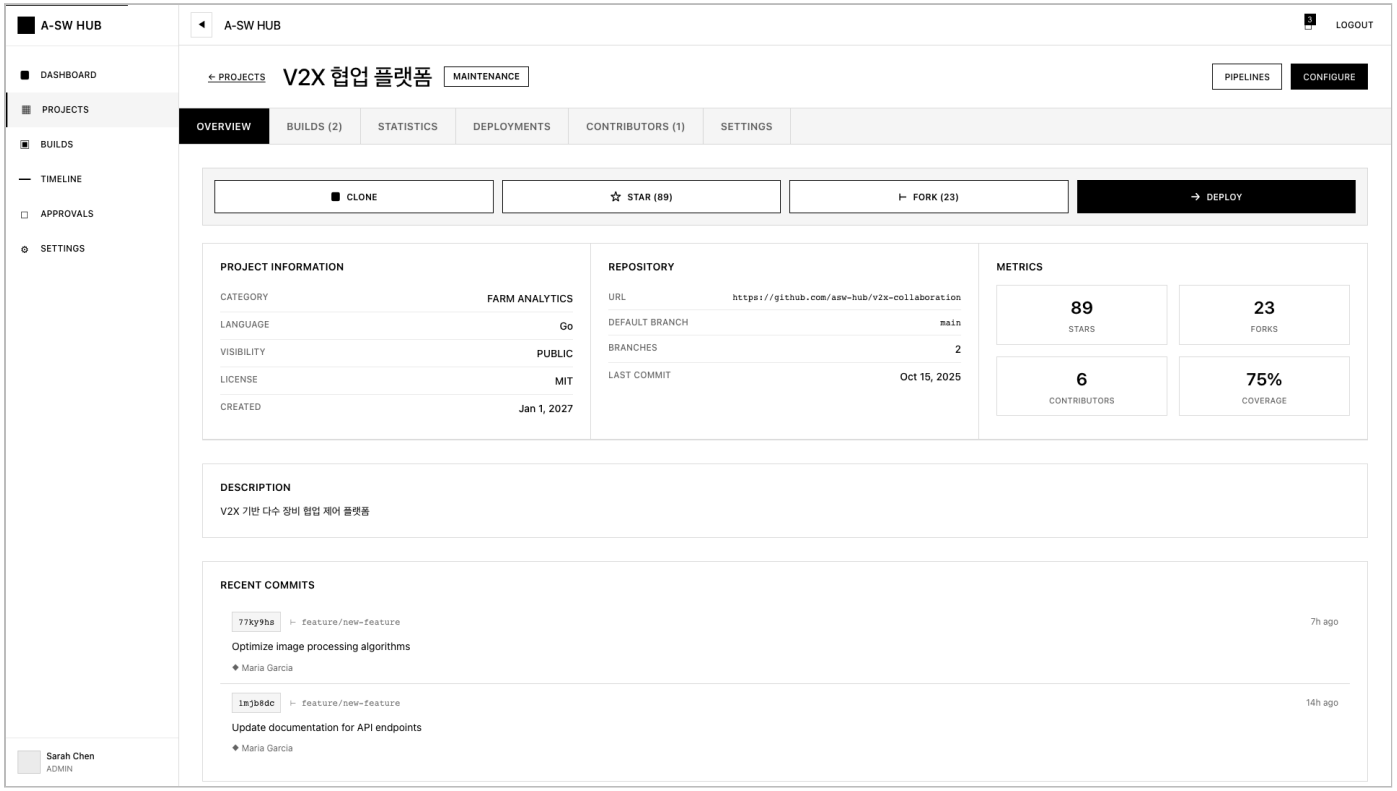
KEY FEATURES

- Real-time project status updates via projectsStore
- Multi-criteria filtering and search with reactive updates
- Search query binding with input event handling
- Category, status, and language filter combinations
- Filtered projects count from filteredProjects store
- Click navigation to project detail page (/projects/{slug})
- ASCII status indicators based on project state
- Responsive grid with border-based layout (no gaps)
- Hover state with subtle background color change
- Project card height fixed at 280px for consistency
- Build status conditional rendering and styling
- Date formatting for last commit information
- *...and 3 more features*

COMPONENTS

COMPONENT
Page header with title and 'NEW PROJECT' button
Filters bar with search input (300px width with auto-complete)
Category dropdown filter with 12 options (all + 11 project categories)
Status dropdown filter (all, active, maintenance, archived, deprecated)
Programming language dropdown filter (all + 10 languages)
Filter results counter showing number of projects
Responsive projects grid (auto-fill, minmax(400px, 1fr))
Project card with header, body, and footer sections
Status indicator with ASCII symbols based on project state
Language badge in monospace font (JetBrains Mono/Fira Code)
Project name heading with semibold weight
Description text with 2-line ellipsis overflow
Category badge with inverted colors (white text on black background)
Statistics row showing stars, forks, and contributors count
Last commit date with formatted display
Build status badge (success or failed) with conditional styling
Empty state message when no projects match filters

Project Detail



PURPOSE

Comprehensive project management hub with tabbed interface providing complete visibility into project overview, builds, deployments, team members, and configuration settings.

KEY COMPONENTS

- Project header with name, status badge (■/▣/□), and action buttons
- Six-tab navigation bar with active state indicators
- Tab content area with conditional rendering based on activeTab

Project Detail

OVERVIEW

The project detail page serves as the central hub for all project-related activities with a sophisticated tabbed interface. The page includes a project header displaying name, status badge, and key metrics, followed by a six-tab navigation system (Overview, Builds, Statistics, Deployments, Contributors, Settings). Each tab provides specialized functionality with detailed information and interactive controls. The page uses reactive Svelte stores for real-time data synchronization and implements role-based access control to show/hide tabs based on user permissions.

PURPOSE

Comprehensive project management hub with tabbed interface providing complete visibility into project overview, builds, deployments, team members, and configuration settings.

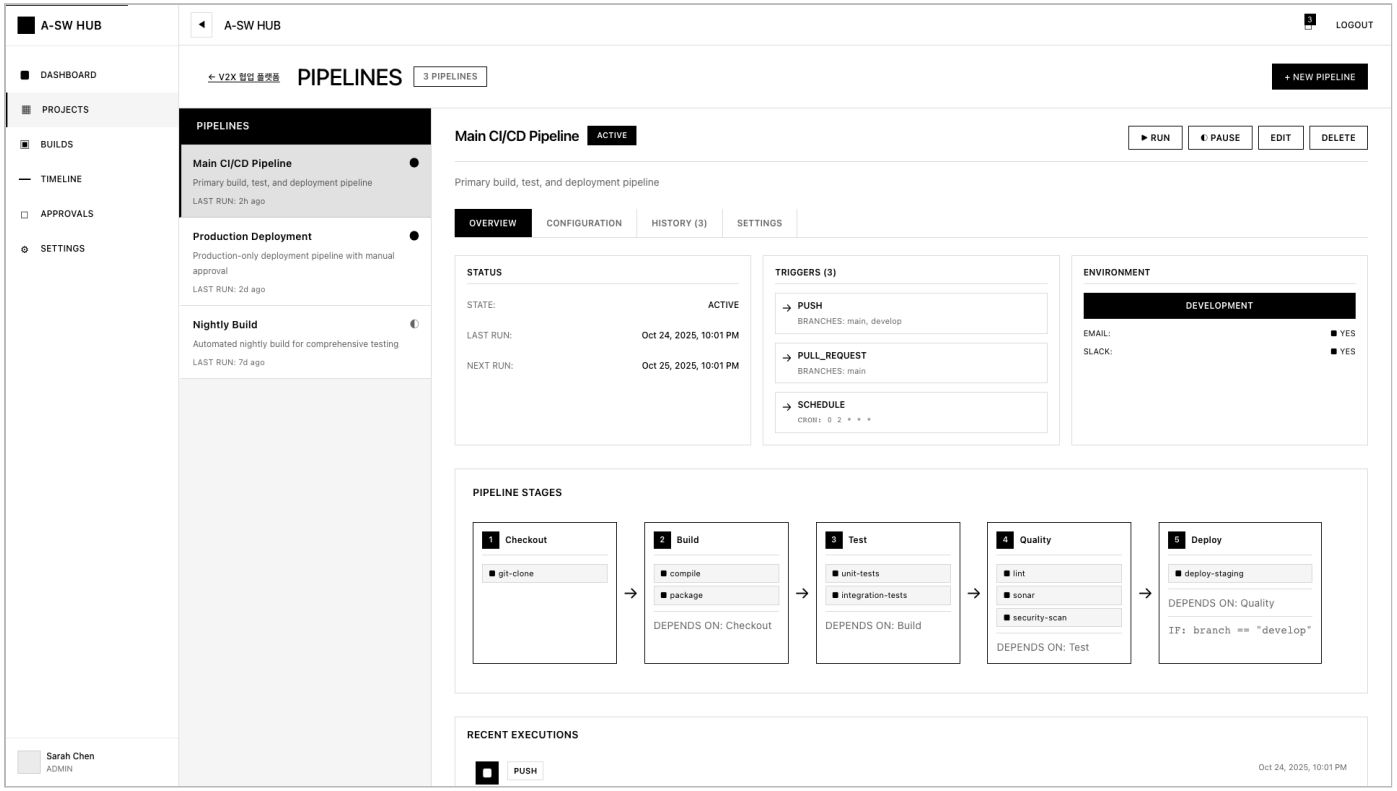
KEY FEATURES

- Six specialized tabs for different project aspects
- Real-time data synchronization via Svelte stores
- Role-based tab visibility and access control
- Tab navigation with active state management
- Responsive design with mobile tab stacking
- Keyboard navigation support
- Deep linking to specific tabs via URL hash

COMPONENTS

COMPONENT
Project header with name, status badge (■/▣/□), and action buttons
Six-tab navigation bar with active state indicators
Tab content area with conditional rendering based on activeTab
Project metrics badges (stars, forks, issues, commits)
Real-time status indicators
Action button group in header (Edit, Settings, Pipeline)
Responsive layout with mobile optimization

Project Pipelines



PURPOSE

CI/CD pipeline management interface with comprehensive visualization of pipeline stages, execution history, configuration editor, and real-time monitoring of pipeline runs.

KEY COMPONENTS

- Four-tab navigation (Overview, Configuration, History, Settings)
- Tab content area with conditional rendering
- Pipeline status indicators

Project Pipelines

OVERVIEW

The project pipelines page provides complete control over continuous integration and deployment pipelines through a four-tab interface. It displays pipeline status, allows configuration editing, shows execution history with logs, and manages pipeline settings.

PURPOSE

CI/CD pipeline management interface with comprehensive visualization of pipeline stages, execution history, configuration editor, and real-time monitoring of pipeline runs.

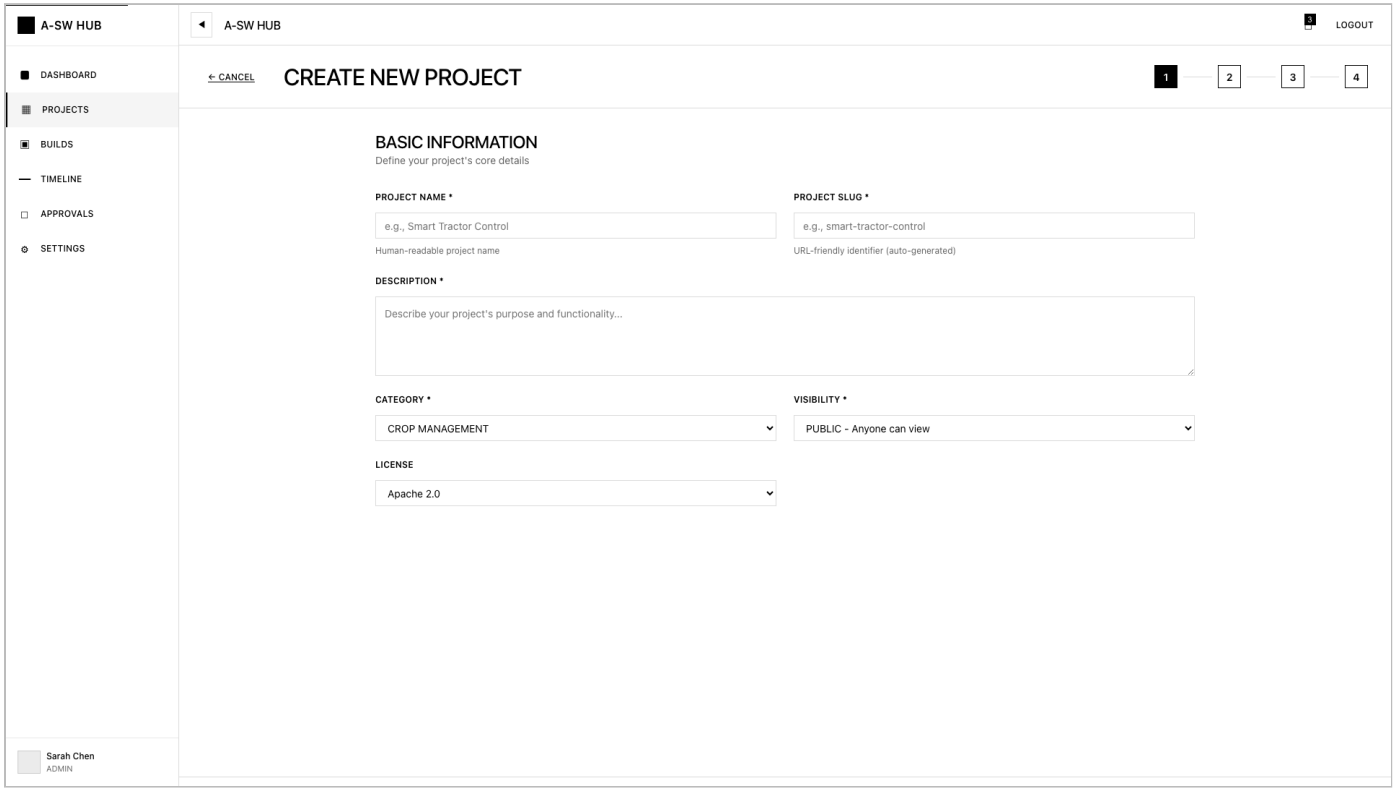
KEY FEATURES

- Four specialized tabs for pipeline management
- Real-time pipeline status via WebSocket
- Pipeline configuration editing
- Execution history with logs
- Pipeline settings management

COMPONENTS

COMPONENT
Four-tab navigation (Overview, Configuration, History, Settings)
Tab content area with conditional rendering
Pipeline status indicators
Action buttons (Run Pipeline, Edit Configuration)
Real-time status updates

New Project



PURPOSE

Guided project creation wizard with four comprehensive steps to set up new agricultural software projects including basic information, repository configuration, CI/CD setup, and review.

KEY COMPONENTS

- Four-step wizard header with step indicators
- Step indicator circles (active, completed, pending states)
- Step connection lines with completion visualization

New Project

OVERVIEW

The new project page guides users through creating a new project with a sophisticated 4-step wizard interface. Step 1 (Basic Information) collects project name with automatic slug generation, description textarea, category selection from 10 agricultural domains, visibility (public/private), and license choice. Step 2 (Repository Setup) offers three options: create new repository with README initialization and .gitignore template selection, connect to existing repository via URL, or import from GitHub/GitLab/Bitbucket with authentication. Step 3 (CI/CD Configuration) enables build trigger selection (on push, on PR, scheduled, manual), environment configuration (ROS2 version, Python version, Node.js version), test settings (unit tests, integration tests, security scan), and quality thresholds (minimum coverage percentage, test pass rate). Step 4 (Review & Create) displays a comprehensive summary of all configured settings organized by section with estimated resource requirements (build time, storage, monthly cost). The wizard includes step validation, progress indicator, navigation controls, and responsive design.

PURPOSE

Guided project creation wizard with four comprehensive steps to set up new agricultural software projects including basic information, repository configuration, CI/CD setup, and review.

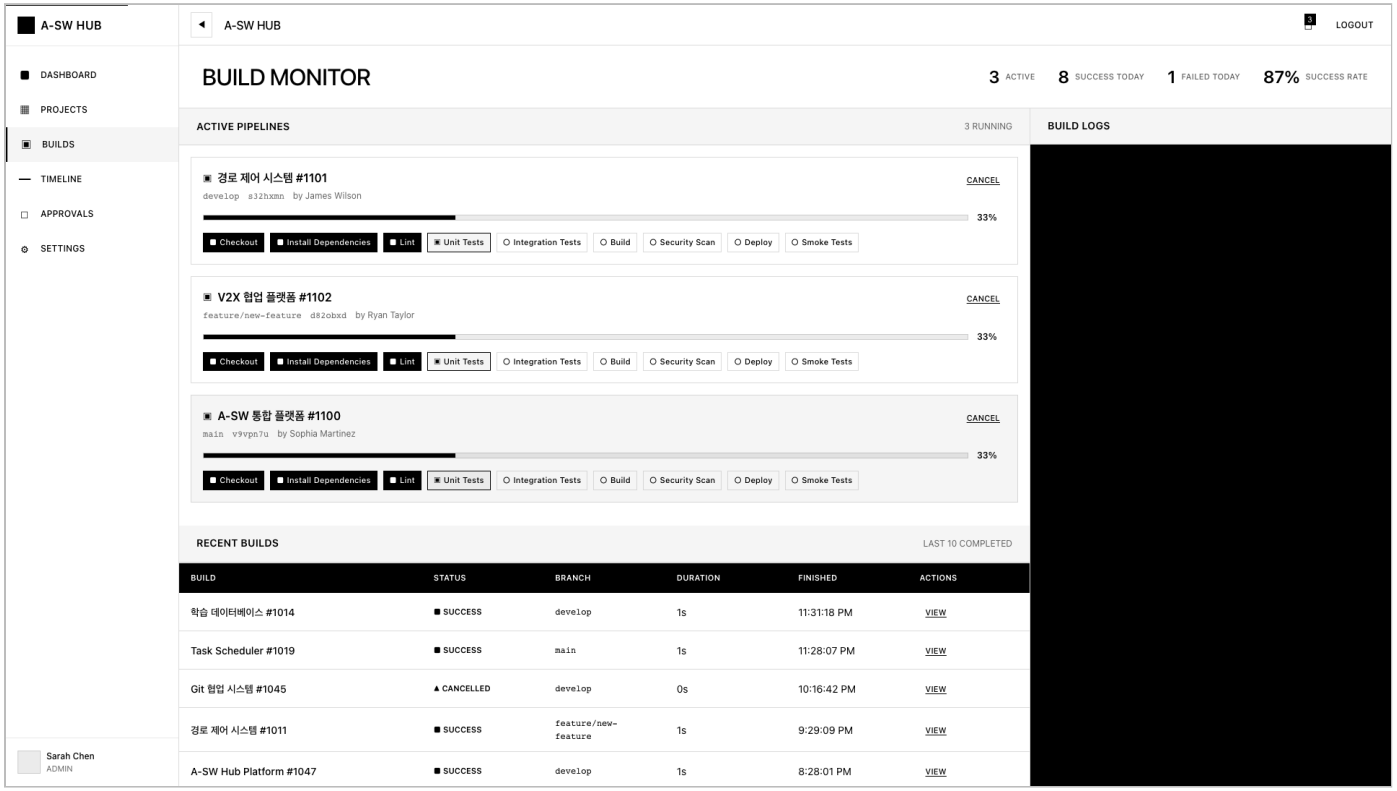
KEY FEATURES

- Guided project setup with step-by-step workflow
- Automatic slug generation from project name
- Three repository connection modes (new, existing, import)
- README initialization option for new repositories
- .gitignore template selection for multiple languages
- Repository URL validation and connection testing
- External platform import (GitHub, GitLab, Bitbucket)
- Comprehensive CI/CD configuration
- Build trigger customization (push, PR, scheduled, manual)
- Multi-language environment support (ROS2, Python, Node.js)
- Quality gate threshold configuration

COMPONENTS

COMPONENT
Four-step wizard header with step indicators
Step indicator circles (active, completed, pending states)
Step connection lines with completion visualization
Cancel button returning to projects list
Step 1 form: name input, slug input (auto-generated), description textarea, category dropdown, visibility dropdown, license dropdown
Step 2 repository options: three radio card choices (new, existing, import)
New repository settings: README checkbox, gitignore template dropdown, default branch input
Existing repository: URL input with validation, test connection button
Import repository: platform selection (GitHub/GitLab/Bitbucket), URL input
Step 3 build triggers: four checkboxes (push, PR, scheduled, manual)
Environment configuration: ROS2 version, Python version dropdowns
Test settings: unit tests, integration tests, security scan checkboxes
Quality thresholds: coverage percentage, pass rate percentage number inputs
Step 4 review sections: Basic Information summary, Repository summary, CI/CD summary
Estimated resources cards: build time, storage required, monthly cost
Footer navigation: Previous button, step counter, Next/Create button
Step validation with disabled Next button on incomplete forms
Wizard content area with fade-in animation on step change

Builds Dashboard



PURPOSE

Real-time monitoring of all CI/CD build pipelines across projects with comprehensive status tracking, log visualization, and build management capabilities.

KEY COMPONENTS

- Page header with 'BUILD MONITOR' title and statistics (4 stat cards)
- Statistics: Active count, Success Today, Failed Today, Success Rate percentage
- Split-view layout: Active builds section (left, scrollable) + Log viewer (right, fixed 500px)

Builds Dashboard

OVERVIEW

The builds dashboard aggregates all build activities across the platform in a sophisticated split-view interface. The left panel displays active builds with real-time progress indicators, stage-by-stage status (Build → Unit Tests → Integration Tests → Deploy → Smoke Tests → Health Check), progress bars, and expandable details showing project name, version, environment, branch, commit SHA, triggering user, and changelog. The right panel features a terminal-style log viewer with a black background and green text, displaying logs organized into collapsible sections with status-specific colors (green for success, red for errors, yellow for warnings, cyan for running). The page includes header statistics showing active builds count, success/failed counts for today, and overall success rate. A recent builds table at the bottom shows the last 10 completed builds with sortable columns, status badges, durations, and action buttons (View, Retry for failed builds).

PURPOSE

Real-time monitoring of all CI/CD build pipelines across projects with comprehensive status tracking, log visualization, and build management capabilities.

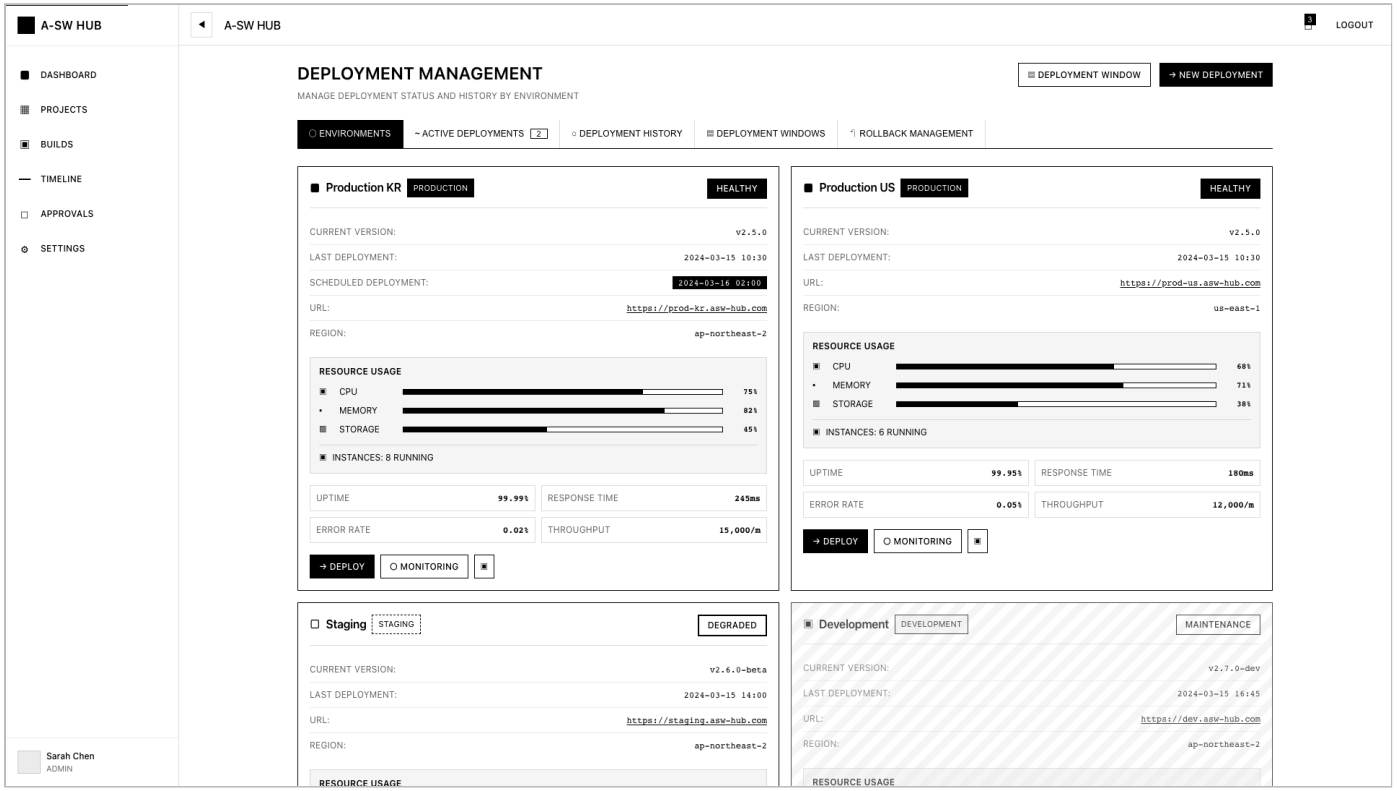
KEY FEATURES

- Multi-project build monitoring with centralized view
- Real-time build status tracking via WebSocket updates
- Build progress calculation across all stages
- Stage-by-stage execution visualization
- Terminal-style log viewer with monospace font
- Collapsible log sections for organized viewing
- Syntax-highlighted logs with color coding by level
- Build cancellation for running pipelines
- Failed build retry functionality
- Build history with last 10 completions
- Sortable table columns for easy navigation
- Duration formatting (hours, minutes, seconds)
- *...and 8 more features*

COMPONENTS

COMPONENT
Page header with 'BUILD MONITOR' title and statistics (4 stat cards)
Statistics: Active count, Success Today, Failed Today, Success Rate percentage
Split-view layout: Active builds section (left, scrollable) + Log viewer (right, fixed 500px)
Active pipelines section with pipeline cards (dashed border animation for in_progress)
Pipeline card: project name, build number, branch, commit SHA (7 chars), triggered by user
Progress bar with percentage and animated fill transition
Stage indicators: 6 stages with status icons (✓ passed, ✕ failed, ◌ running, ○ pending)
Stage status badges with appropriate colors and icons
Cancel button for running builds (stops execution)
Recent builds table with 6 columns (Build, Status, Branch, Duration, Finished, Actions)
Table header with inverted colors (white on black)
Status badges in table rows with ASCII icons
Retry button for failed builds (visible only on failure)
View button navigating to detailed build page
Log viewer with black terminal background (#000000)
Collapsible log sections with expand/collapse controls
Expand All / Collapse All buttons in log header
Log sections with status-specific border colors
Log lines with syntax highlighting (ERROR: red, WARNING: yellow, SUCCESS: green, INFO: cyan)
Section headers with toggle icon (▼ expanded, ▶ collapsed), status icon, name, duration

Deployments



PURPOSE

Comprehensive deployment management dashboard showing deployment status, history, and controls across all environments (development, staging, production) with rollback capabilities and deployment window scheduling.

KEY COMPONENTS

- Page header with title 'DEPLOYMENT MANAGEMENT' and action buttons (Deployment Window, New Deployment)
- Five-tab navigation (Environments, Active Deployments, History, Windows, Rollback)
- Tab badges showing active deployment count

Deployments

OVERVIEW

The deployments page provides complete visibility into deployment activities across the platform through a sophisticated 5-tab interface. The Environments tab (Overview) displays 4 environment cards (Production KR, Production US, Staging, Development) each showing status (healthy/degraded/down/maintenance), current version, last deployment date, scheduled next deployment, URL, region (AWS/GCP), resource usage graphs (CPU, Memory, Storage with progress bars), instance count, and key metrics (uptime %, response time ms, error rate %, throughput/min). The Active Deployments tab shows currently running deployments with real-time progress through 6 check stages (Build → Unit Tests → Integration Tests → Deploy → Smoke Tests → Health Check), displaying project name, version, environment, branch, commit, deployer, timestamp, and change log with artifact information. The Deployment History tab provides a filterable table of past deployments with status, duration, and rollback options. The Deployment Windows tab schedules deployment time slots with environment and project associations. The Rollback Management tab lists all rollback-eligible deployments from the last 30 days with version comparison and one-click rollback execution.

PURPOSE

Comprehensive deployment management dashboard showing deployment status, history, and controls across all environments (development, staging, production) with rollback capabilities and deployment window scheduling.

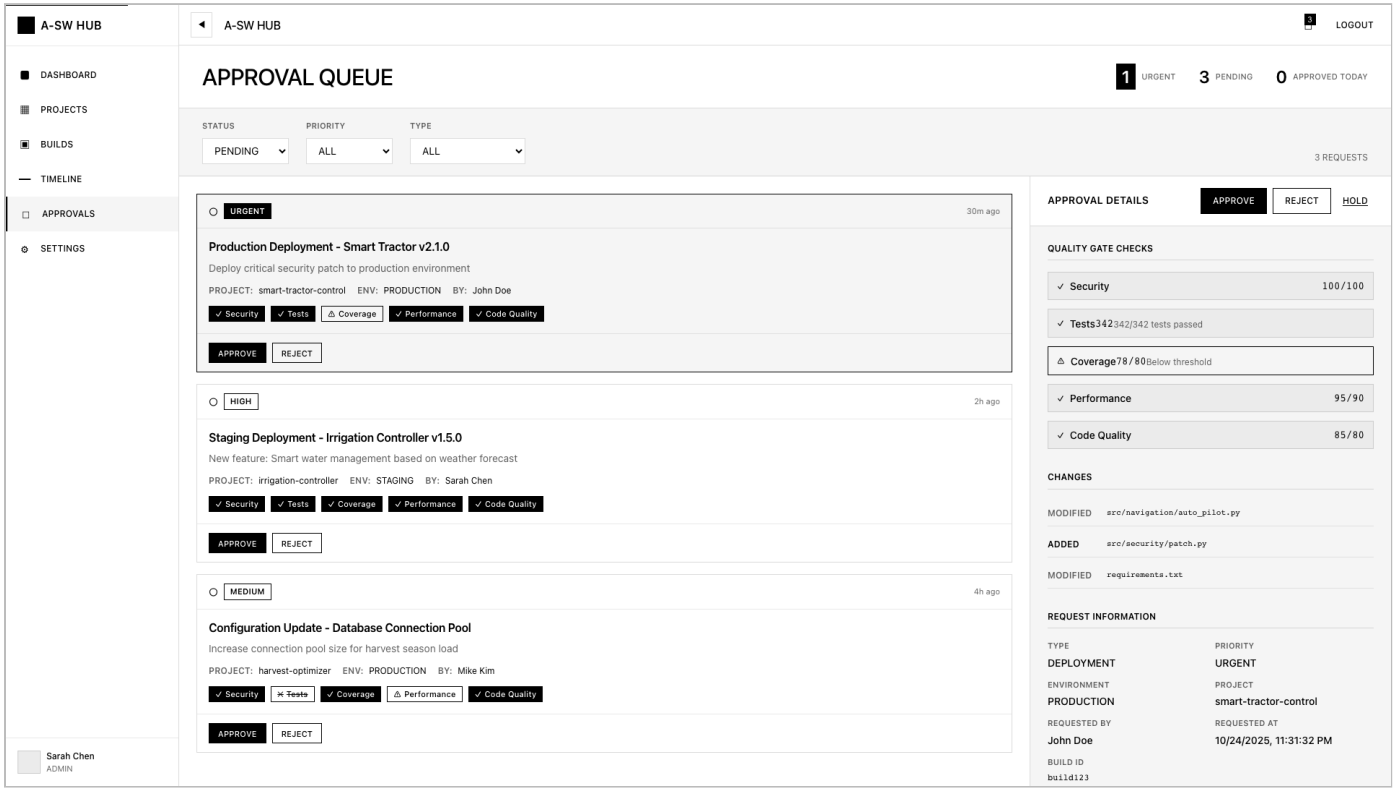
KEY FEATURES

- Multi-environment deployment tracking (4 environments)
- Real-time deployment status monitoring
- Resource usage visualization with progress bars
- Performance metrics display (uptime, response time, error rate, throughput)
- One-click deployment triggers per environment
- Six-stage deployment pipeline with progress tracking
- Deployment pause and resume functionality
- Automated rollback capabilities for failed deployments
- Deployment history with filtering (environment, project, status)
- Export deployment history to CSV/JSON

COMPONENTS

COMPONENT
Page header with title 'DEPLOYMENT MANAGEMENT' and action buttons (Deployment Window, New Deployment)
Five-tab navigation (Environments, Active Deployments, History, Windows, Rollback)
Tab badges showing active deployment count
Environment cards grid (2 columns) with status, version, and metrics
Environment status indicator (HEALTHY: filled, DEGRADED: bordered, DOWN: strikethrough, MAINTENANCE: pattern)
Environment type badges (PRODUCTION: filled black, STAGING: dashed border, DEVELOPMENT: gray)
Resource usage section with 3 progress bars (CPU, Memory, Storage) and percentage labels
Instances count display with running status
Environment metrics grid (2x2): Uptime %, Response Time ms, Error Rate %, Throughput/min
Environment action buttons (Deploy, Monitoring, Menu)
Active deployment cards with real-time status
Deployment status badge with spinner animation for in_progress
6-stage deployment checks with status icons (✓ passed, ✕ failed, ◌ running, ○ pending)
Deployment metadata (deployer, approver, start time, elapsed time)
Expandable deployment details with changelog and artifacts
Deployment action buttons (Pause/Resume, Rollback, Logs)
History filters (Environment, Project, Status dropdowns + Export button)

Approvals



PURPOSE

Centralized approval workflow management for deployment requests, merge requests, and configuration changes with comprehensive quality gate checks and priority-based queue management.

KEY COMPONENTS

- Page header with title 'APPROVAL QUEUE' and 3 statistics cards (Urgent: highlighted, Pending, Approved Today)
- Filters bar with 3 dropdowns (Status, Priority, Type) and results counter
- Split-view layout: Approval list (left, scrollable) + Approval details (right, 500px fixed)

Approvals

OVERVIEW

The approvals page provides a sophisticated interface for managing all pending approval requests across projects in a split-view layout. The left panel displays the approval queue as a scrollable list of cards, each showing priority badge (URGENT: filled black, HIGH: bordered, MEDIUM/LOW: hollow), approval type (deployment, merge, release, configuration, access), request title, description, project name, environment, requestor name, and time ago (formatted as '30m ago', '2h ago', etc.). Each card includes 5 quality check badges (Security, Tests, Coverage, Performance, Code Quality) with status indicators (✓ PASS: filled, ⚠ WARN: bordered, ✕ FAIL: strikethrough). The right panel shows detailed information for the selected approval including expandable quality gate checks with threshold values and details, file changes list with type indicators (ADDED, MODIFIED, DELETED), request metadata, and approval action buttons (Approve, Reject with reason, Hold). Header statistics show counts for urgent, pending, and approved today. Filters allow sorting by status, priority, and type.

PURPOSE

Centralized approval workflow management for deployment requests, merge requests, and configuration changes with comprehensive quality gate checks and priority-based queue management.

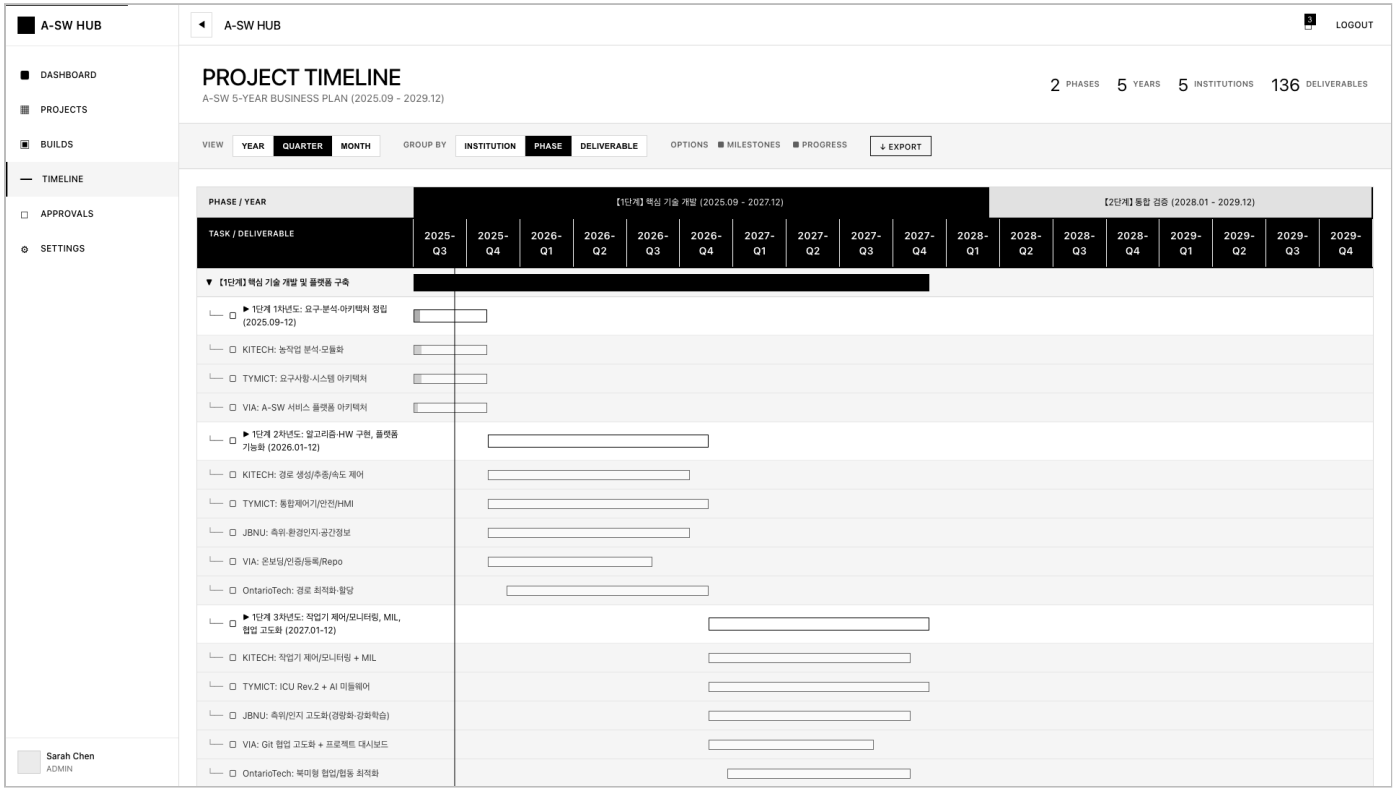
KEY FEATURES

- Centralized approval management across all projects
- Priority-based queue (URGENT, HIGH, MEDIUM, LOW)
- Five approval types (deployment, merge, release, configuration, access)
- Comprehensive quality gate checks with pass/warn/fail states
- Quality metrics display (Security, Tests, Coverage, Performance, Code Quality)
- Threshold-based quality validation
- File changes visualization with type indicators
- Approval action workflow (Approve, Reject with reason, Hold)
- Rejection reason requirement with textarea dialog
- Real-time approval status updates
- Time ago formatting (minutes, hours, days)

COMPONENTS

COMPONENT
Page header with title 'APPROVAL QUEUE' and 3 statistics cards (Urgent: highlighted, Pending, Approved Today)
Filters bar with 3 dropdowns (Status, Priority, Type) and results counter
Split-view layout: Approval list (left, scrollable) + Approval details (right, 500px fixed)
Approval cards in queue with selectable/selected states
Card header: status icon (○ pending, ■ approved, □ rejected, ▣ on-hold), priority badge, time ago
Priority badges with styling (URGENT: black background, HIGH: bordered, MEDIUM/LOW: normal)
Card body: title (16px semibold), description (14px muted), metadata row (project, env, requestor)
Quality checks row: 5 check badges with status icons and names
Quality check badges (PASS: filled, WARN: bordered, FAIL: strikethrough)
Card actions (visible only for pending): Approve and Reject buttons
Details panel header with 'APPROVAL DETAILS' title and action buttons
Quality gate checks section with expandable rows showing name, value, threshold, details
Quality row styling (PASS: surface background, WARN: bordered, FAIL: fail background)
Changes section listing modified/added/deleted files with syntax highlighting
Change type labels (ADDED: foreground color, MODIFIED: muted, DELETED: strikethrough)
File paths in monospace font
Request information grid (2 columns) showing Type, Priority, Environment, Project, Requested By, Requested At, Build ID
Approve button (primary, green when enabled)

Timeline



PURPOSE

Interactive Gantt chart visualization of project timelines, milestones, and deliverables across the 5-year A-SW initiative (2025.09 - 2029.12) with comprehensive tracking of 136 deliverables for all participating institutions.

KEY COMPONENTS

- Page header with title 'PROJECT TIMELINE' and subtitle showing date range
- Statistics panel: 4 stat cards (2 Phases, 5 Years, 5 Institutions, 136 Deliverables)
- Controls bar with 3 sections: View mode (YEAR/QUARTER/MONTH buttons), Group by (INSTITUTION/PHASE/DELIVERABLE buttons), Options (Milestones and Progress checkboxes)

Timeline

OVERVIEW

The timeline page provides a sophisticated Gantt chart visualization of the entire 5-year A-SW business plan using actual project data. The interface displays a phase indicator bar at the top showing Phase 1 (60%: 2025.09-2027.12 - Core Technology Development) and Phase 2 (40%: 2028.01-2029.12 - Integration & Verification). The header shows key statistics: 2 phases, 5 years, 5 institutions (KITECH, TYMICT, JBNU, VIA, OntarioTech), and 136 total deliverables. Users can switch between YEAR (showing 5 annual periods), QUARTER (showing Q3 2025 + 16 quarters), and MONTH (showing SEP-DEC 2025 + 48 months) views. The Gantt chart displays hierarchical tasks grouped by institution/phase/deliverable with expandable/collapsible groups, progress bars, milestone diamonds, and dependency lines. A 'TODAY' marker shows current progress. Task bars use different styles: parent tasks (solid black), completed (filled), in-progress (diagonal stripes), planned (hollow), delayed (dashed). A comprehensive legend explains all visual indicators. The bottom section displays institution summary cards with deliverable counts and progress percentages.

PURPOSE

Interactive Gantt chart visualization of project timelines, milestones, and deliverables across the 5-year A-SW initiative (2025.09 - 2029.12) with comprehensive tracking of 136 deliverables for all participating institutions.

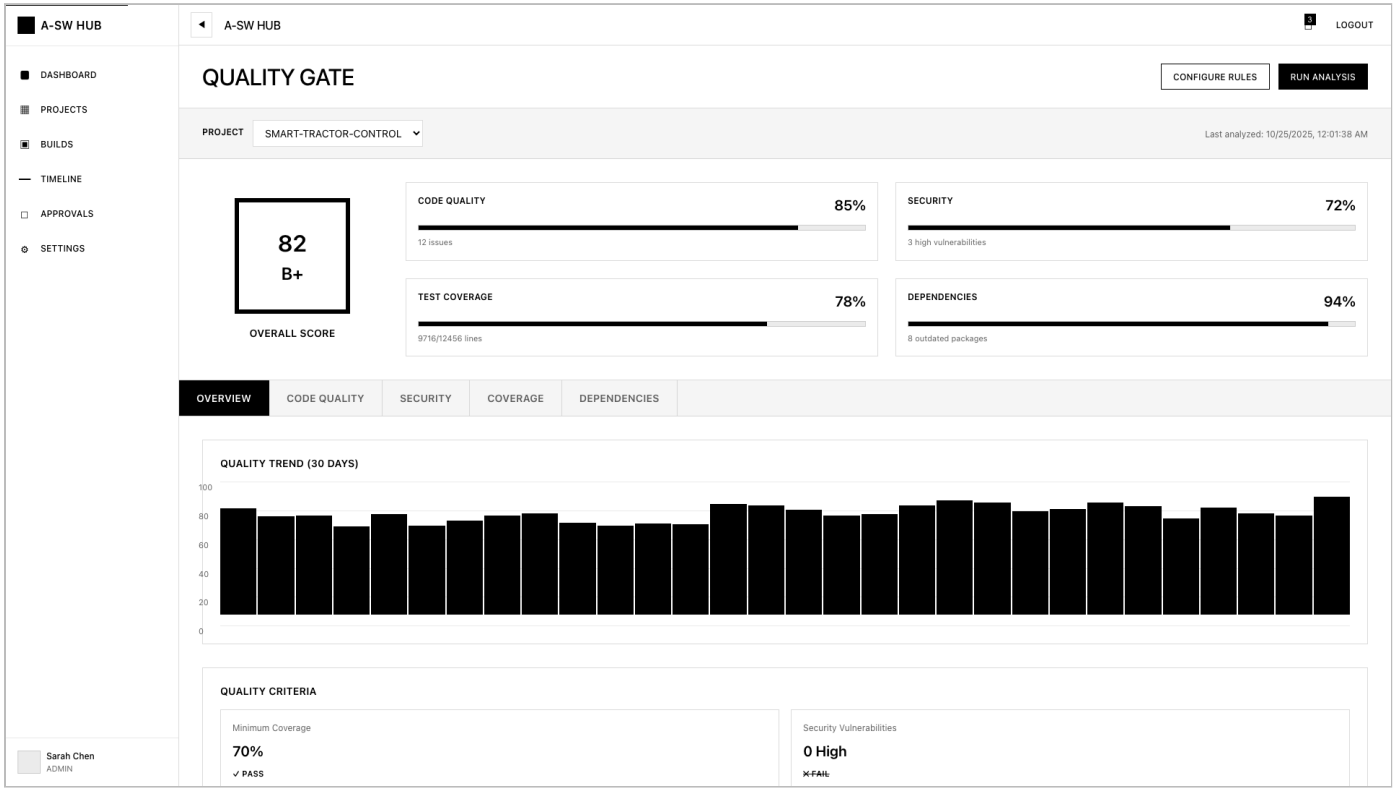
KEY FEATURES

- Five-year timeline visualization (2025.09 - 2029.12)
- 136 deliverables tracking across phases
- Three view modes: YEAR (5 periods), QUARTER (18 quarters), MONTH (52 months)
- Three grouping options: INSTITUTION, PHASE, DELIVERABLE
- Hierarchical task structure with expand/collapse functionality
- Real-time progress tracking with visual indicators
- Milestone visualization with date positioning
- Task dependency relationship visualization
- Gantt data integration from businessPlan module
- Institution filtering (KITECH, TYMICT, JBNU, VIA, OntarioTech)
- Task hierarchy tree view with indentation

COMPONENTS

COMPONENT
Page header with title 'PROJECT TIMELINE' and subtitle showing date range
Statistics panel: 4 stat cards (2 Phases, 5 Years, 5 Institutions, 136 Deliverables)
Controls bar with 3 sections: View mode (YEAR/QUARTER/MONTH buttons), Group by (INSTITUTION/PHASE/DELIVERABLE buttons), Options (Milestones and Progress checkboxes)
Export button for downloading timeline data
Phase indicator bar with 2-segment display (Phase 1: black background, Phase 2: gray background)
Timeline scale header with sticky positioning showing time units based on view mode
Task column header (300px fixed width) with task names and hierarchy
Timeline bars area with horizontal task bars and progress fills
Expandable/collapsible task groups with toggle icons (▼ expanded, ► collapsed)
Group header rows with gray background and border bottom
Child task rows with indent indicators (└──) and status icons
Status icons: ■ completed, ▣ in-progress, □ planned, ▲ delayed
Task bars with different styles: solid (completed), striped (in-progress), hollow (planned), dashed (delayed)
Progress fill overlays (30% opacity) when showProgress is enabled
Milestone diamonds (◆) positioned at specific dates
TODAY marker with vertical line and label
Legend at bottom with 6 items explaining visual indicators
Progress indicator characters: ■▒▤▩·

Quality Metrics



PURPOSE

Comprehensive code quality analysis and reporting dashboard providing detailed insights into code quality, security vulnerabilities, test coverage, and dependency health with AI-powered recommendations.

KEY COMPONENTS

- Page header with title 'QUALITY GATE' and action buttons (Configure Rules, Run Analysis)
- Project selector dropdown with last analysis timestamp
- Score dashboard grid: Overall score card (160px circle) + 4 category cards

Quality Metrics

OVERVIEW

The quality metrics page provides in-depth code quality analysis through a sophisticated tabbed interface with visual score representations. The dashboard begins with project selection and analysis timestamp display, followed by a prominent overall score section showing a large circular score indicator (82/100 with B+ grade) and four category scores with progress bars: Code Quality (85% - 12 issues), Security (72% - 3 high vulnerabilities), Test Coverage (78% - 9716/12456 lines), and Dependencies (94% - 8 outdated). Five tabs provide detailed analysis: Overview tab displays a 30-day quality trend bar chart, quality criteria checklist (4 items with pass/warn/fail states), and AI recommendations (3 cards with icons); Code Quality tab shows total issues count, duplicate lines, complex functions table with complexity scores, function names, file locations, and line numbers; Security tab presents vulnerability summary (high/medium/low counts), security hotspots cards with type, severity, location, description, and action buttons; Coverage tab displays line and branch coverage with circular progress rings, and a list of low-coverage files with coverage bars; Dependencies tab shows package statistics, license compatibility grid with incompatible licenses highlighted, and SBOM (Software Bill of Materials) generation options.

PURPOSE

Comprehensive code quality analysis and reporting dashboard providing detailed insights into code quality, security vulnerabilities, test coverage, and dependency health with AI-powered recommendations.

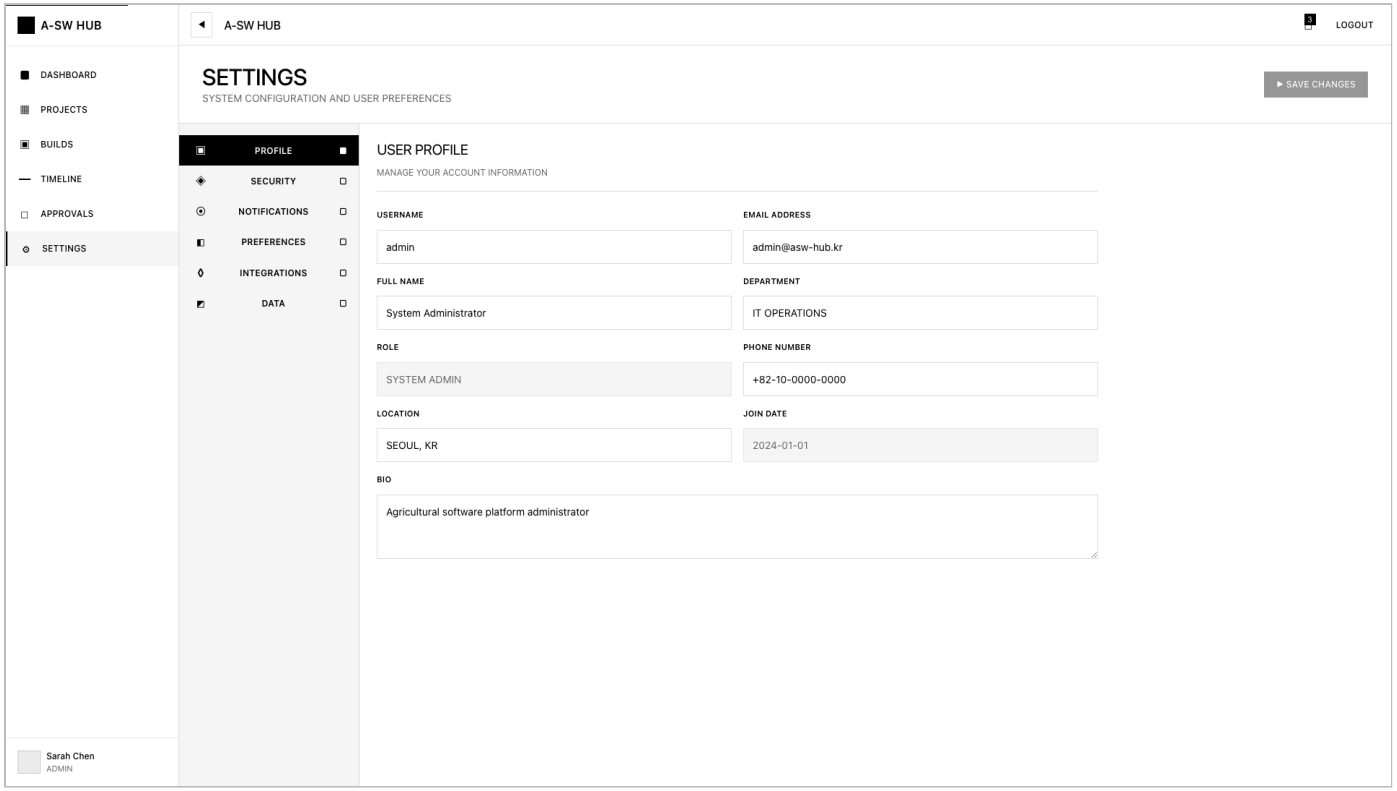
KEY FEATURES

- Real-time quality scoring and grading (A-F scale)
- Project-specific quality analysis with historical data
- 30-day quality trend visualization with bar chart
- Code quality metrics (issues, duplication, complexity)
- High complexity function identification with thresholds
- Security vulnerability scanning (high/medium/low classification)
- Security hotspot detection with severity levels
- SQL injection, path traversal, weak crypto detection
- Line and branch coverage measurement

COMPONENTS

COMPONENT
Page header with title 'QUALITY GATE' and action buttons (Configure Rules, Run Analysis)
Project selector dropdown with last analysis timestamp
Score dashboard grid: Overall score card (160px circle) + 4 category cards
Overall score circle with large score number (32px), grade letter (24px), border thickness indicating grade
Grade styling (A: filled, B: 6px border, C: 4px border, D: 2px border, F: 2px border)
Category score cards (2x2 grid) with name, percentage, progress bar, metadata
Five-tab navigation (Overview, Code Quality, Security, Coverage, Dependencies)
Overview tab: 30-day trend chart with bars, quality criteria grid (2x2), AI recommendations (3 cards)
Trend chart: 6 horizontal grid lines with labels (0-100), 30 vertical bars with hover tooltips
Quality criteria cards showing criterion name, target value, pass/warn/fail status
AI recommendation cards with emoji icons (💡 refactoring, 🔒 security, 🧪 testing), title, description
Code Quality tab: 3 metric cards (Issues, Duplicate Lines, Complex Functions), high complexity functions table
Functions table: 5 columns (Function, Complexity, File, Line, Action), complexity badges (normal/high: >20)
Security tab: 3 vulnerability count cards (High: filled black, Medium: 2px border, Low: normal), security hotspots list
Hotspot cards with type, severity badge, file location (monospace), description, action buttons (View Code, Fix Suggestion)
Coverage tab: 2 coverage metrics with SVG circular progress rings (120px), metric values inside rings, detailed stats below

System Settings



PURPOSE

Comprehensive platform-wide settings management for user profile, security, notifications, preferences, integrations, and data privacy with granular control over all system configurations.

KEY COMPONENTS

- Page header with title 'SETTINGS', subtitle, unsaved changes indicator (● UNSAVED CHANGES), Save Changes button
- Sidebar navigation (250px fixed) with 6 tab buttons (Profile, Security, Notifications, Preferences, Integrations, Data)
- Tab buttons with ASCII icons (▣ profile, ◈ security, ⦿ notifications, ◧ preferences, ◊ integrations, ◩ data)

System Settings

OVERVIEW

The settings page provides extensive configuration options across six major categories through a sophisticated sidebar navigation with tabbed content areas. Each tab contains detailed forms and controls for specific configuration areas including user profile management, security settings, notification preferences, UI customization, external service integrations, and data privacy controls.

PURPOSE

Comprehensive platform-wide settings management for user profile, security, notifications, preferences, integrations, and data privacy with granular control over all system configurations.

KEY FEATURES

- Six-category settings management
- Sidebar navigation with active state
- Form validation with real-time feedback
- Dirty state tracking
- Unsaved changes warning
- Responsive design with collapsible sidebar on mobile

COMPONENTS

COMPONENT
Page header with title 'SETTINGS', subtitle, unsaved changes indicator (● UNSAVED CHANGES), Save Changes button
Sidebar navigation (250px fixed) with 6 tab buttons (Profile, Security, Notifications, Preferences, Integrations, Data)
Tab buttons with ASCII icons (▣ profile, ◈ security, ⦿ notifications, ◧ preferences, ◊ integrations, ◩ data)
Active tab indicator (filled black background, left border)
Tab content area with section-specific forms
Form dirty state tracking indicator
Confirmation dialogs for destructive actions