

VIASOFT.AI

VIA Inc.

ADMINISTRATOR

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

1	Login	3
2	Dashboard	4
3	Projects List	5
4	Project Detail	6
5	Project Configuration	12
6	Project Pipelines	20
7	New Project	25
8	Builds Dashboard	26
9	Deployments	27
10	Approvals	28
11	Timeline	29
12	Quality Metrics	30

Login

A-SW HUB

AUTHENTICATION
ACCESS AGRICULTURAL SOFTWARE WORKSPACE

EMAIL ADDRESS *

PASSWORD *

REMEMBER ME [FORGOT PASSWORD?](#)

AUTHENTICATE

OR

AUTHENTICATE WITH GITHUB

142 PROJECTS | 47 ACTIVE BUILDS | 12 USERS ONLINE

SECURE AGRICULTURAL MANAGEMENT SYSTEM

CROP OPTIMIZATION
Real-time monitoring and automated decision systems

SUPPLY CHAIN
Complete traceability from farm to distribution

DATA ANALYTICS
Predictive insights and performance metrics

COMPLIANCE
Automated regulatory and certification tracking

ISO 27001 **GDPR** **SOC 2**

© 2024 A-SW TECHNOLOGIES. ALL RIGHTS RESERVED.

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

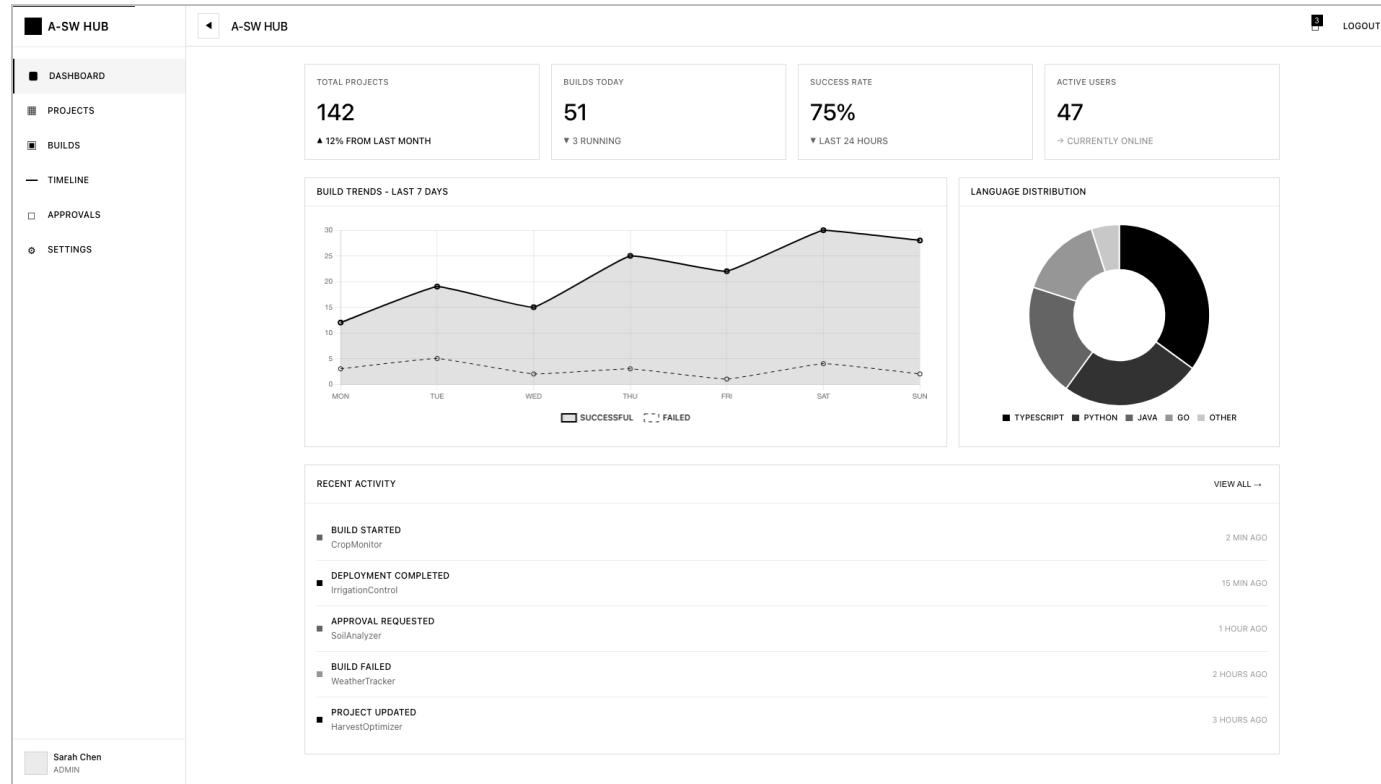
Forgot password link with underline styling

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

A-SW HUB		AGRICULTURAL SOFTWARE PROJECTS				LOGOUT
DASHBOARD		SEARCH	CATEGORY	STATUS	LANGUAGE	+ NEW PROJECT
PROJECTS		Search projects...	ALL	ALL	ALL	15 PROJECTS
Farm Analytics						
V2X 협업 플랫폼	Path Optimizer	Python	Python	Java		
V2X 기반 다수 장비 협업 제어 플랫폼	Multi-agent path optimization for agricultural machinery	학습 데이터베이스	Task Scheduler	Cooperative task allocation and scheduling system		
STARS 89 FORKS 23 CONTRIB 6 UPDATED 10/14/2025	SUCCESS	STARS 145 FORKS 38 CONTRIB 6 UPDATED 10/17/2025	SUCCESS	STARS 189 FORKS 45 CONTRIB 20 UPDATED 10/19/2025	SUCCESS	STARS 98 FORKS 25 CONTRIB 5 UPDATED 10/20/2025
측위 알고리즘	농작업 분석 모듈	Python	C++	Go		
고정밀 측위 및 SLAM 알고리즘	무인차 농작업 분석 및 A-SW 모듈화 파라미터 정립 시스템	통합제어기(ICU)	Git 협업 시스템	Git 기반 A-SW 저장소 및 협업 워크플로우		
WEATHER MONITORING	CROP MANAGEMENT	FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS		
STARS 298 FORKS 78 CONTRIB 8 UPDATED 10/21/2025	SUCCESS	STARS 156 FORKS 34 CONTRIB 12 UPDATED 10/22/2025	SUCCESS	STARS 67 FORKS 12 CONTRIB 10 UPDATED 10/23/2025	SUCCESS	STARS 389 FORKS 98 CONTRIB 18 UPDATED 10/23/2025
AI 미들웨어	경로 제어 시스템	Python	C++	TypeScript	TypeScript	
멀티모달 AI 통합 미들웨어 플랫폼	경로 생성/추종/속도 제어 통합 시스템	인증/권한 시스템	A-SW 통합 플랫폼	무인차량 농작업 소프트웨어 통합 플랫폼 - 1단계 3개년 + 2단계 2개년 국가 R&D 사업		
FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS			
STARS 178 FORKS 42 CONTRIB 9 UPDATED 10/24/2025	FAILED	STARS 203 FORKS 45 CONTRIB 8 UPDATED 10/24/2025	RUNNING	STARS 234 FORKS 67 CONTRIB 8 UPDATED 10/24/2025	SUCCESS	STARS 342 FORKS 89 CONTRIB 45 UPDATED 10/24/2025
Sarah Chen ADMIN						

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

- | 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

The screenshot shows the 'OVERVIEW' tab of the A-SW HUB project detail page. The top navigation bar includes tabs for DASHBOARD, PROJECTS (selected), BUILD, TIMELINE, APPROVALS, and SETTINGS. The main content area has tabs for PROJECTS (selected), MAINTENANCE, PIPELINES, and CONFIGURE. Below these are buttons for CLONE, STAR (89), FORK (23), and DEPLOY.

PROJECT INFORMATION:

Category	FARM ANALYTICS
Language	Go
Visibility	PUBLIC
License	MIT
Created	Jan 1, 2027

REPOSITORY:

URL	https://github.com/asw-hub/v2x-collaboration
Default Branch	main
Branches	2
Last Commit	Oct 14, 2025

METRICS:

89	23
STARS	FORKS
6	75%
CONTRIBUTORS	COVERAGE

DESCRIPTION:

V2X 기반 다수 장비 협업 제어 플랫폼

RECENT COMMITS:

t7fgmnh	feature/new-feature	2h ago
Add unit tests for irrigation controller	Emma Johnson	
ddmune	feature/new-feature	16h ago
Improve error handling in harvest module	Ryan Taylor	

Sarah Chen
ADMIN

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 Detail - Overview

The screenshot displays the 'Project Detail - Overview' page of the A-SW HUB. The top navigation bar includes 'A-SW HUB' (with a dropdown arrow), 'LOGOUT', and tabs for 'PROJECTS' (selected), 'MAINTENANCE', 'PIPELINES', and 'CONFIGURE'. The left sidebar contains links for 'DASHBOARD', 'PROJECTS' (selected), 'BUILDS', 'TIMELINE', 'APPROVALS', and 'SETTINGS'. The main content area is titled 'V2X 협업 플랫폼' and shows the following sections:

- OVERVIEW**: Buttons for 'CLONE', 'STAR (89)', 'FORK (23)', and 'DEPLOY'.
- PROJECT INFORMATION**:

CATEGORY	FARM ANALYTICS
LANGUAGE	Go
VISIBILITY	PUBLIC
LICENSE	MIT
CREATED	Jan 1, 2027
- REPOSITORY**:

URL	https://github.com/asw-hub/v2x-collaboration
DEFAULT BRANCH	main
BRANCHES	2
LAST COMMIT	Oct 14, 2025
- METRICS**:

89	23
STARS	FORKS
6	75%
CONTRIBUTORS	COVERAGE
- DESCRIPTION**: V2X 기반 다수 장비 협업 제어 플랫폼
- RECENT COMMITS**:

t7fgmnh	↳ feature/new-feature	2h ago
Add unit tests for irrigation controller		
◆ Emma Johnson		
ddmune	↳ feature/new-feature	16h ago
Improve error handling in harvest module		
◆ Ryan Taylor		
- User Details**: Sarah Chen (ADMIN)

KEY COMPONENTS

- Quick actions bar with 4 buttons (CLONE, STAR, FORK, DEPLOY)
- 3-column info grid (Project Information, Repository, Metrics)
- Project Information card with 5 rows (Category, Language, Visibility, License, Created)

PROJECT DETAIL - OVERVIEW

OVERVIEW

The Overview tab displays comprehensive project information including quick action buttons, project metadata cards, repository details, metrics dashboard, and recent commit history. The layout uses a 3-column info grid showing Project Information (category, language, visibility, license, created date), Repository details (URL, default branch, branch count, last commit), and Metrics (stars, forks, contributors, coverage percentage). Below the info grid, a description section shows the full project description, followed by a recent commits list with commit messages, authors, timestamps, and SHA hashes.

KEY FEATURES

- One-click clone URL copy to clipboard
- Star/unstar project with real-time count update
- Fork project functionality
- Quick deploy to default environment
- Category name formatting (hyphens to spaces, uppercase)
- Date formatting for created and last commit dates
- Coverage percentage display with conditional styling
- Monospace font for technical values (URLs, branches, SHAs)
- Recent commits with time ago formatting
- Commit SHA truncation to 7 characters
- Responsive grid layout (stacks to 1 column on mobile)

COMPONENTS

COMPONENT

Quick actions bar with 4 buttons (CLONE, STAR, FORK, DEPLOY)
3-column info grid (Project Information, Repository, Metrics)
Project Information card with 5 rows (Category, Language, Visibility, License, Created)
Repository card with 4 rows (URL in monospace, Default Branch, Branches count, Last Commit)
Metrics card with 2x2 grid (Stars, Forks, Contributors, Coverage %)
Description section with project description text
Recent Commits section with commits list
Commit cards showing message, author, time ago, SHA (7 chars monospace)
Clone URL copy functionality
Star toggle button with active state
Fork button with fork count
Quick Deploy button (primary styling)

Project Detail - Builds

The screenshot shows the 'Builds' section of the A-SW HUB interface. The left sidebar includes links for Dashboard, Projects, Builds, Timeline, Approvals, and Settings. The main header displays 'V2X 협업 플랫폼' and 'MAINTENANCE' buttons, along with 'LOGOUT', 'PIPELINES', and 'CONFIGURE' buttons. The navigation bar below the header includes 'OVERVIEW', 'BUILD HISTORY' (selected), 'STATISTICS', 'DEPLOYMENTS', 'CONTRIBUTORS (2)', and 'SETTINGS'. A search bar at the top of the build history section contains the placeholder 'SEARCH BUILDS...'. Below it are dropdown filters for 'ALL BRANCHES', 'ALL STATUS', 'ALL USERS', and a 'CLEAR' button. The build history table has columns: '#', 'STATUS', 'BRANCH', 'COMMIT', 'TRIGGERED BY', 'DURATION', 'DATE', and 'ACTIONS'. It lists two builds: #1016 and #1031, both marked as 'SUCCESS'. The 'ACTIONS' column for each build contains three small icons: a downward arrow, a circle, and a downward arrow. At the bottom of the page, a user profile for 'Sarah Chen' (ADMIN) is shown.

#	STATUS	BRANCH	COMMIT	TRIGGERED BY	DURATION	DATE	ACTIONS
#1016	SUCCESS	feature/new-feature	t7fgmnh	Emma Johnson	1s	Oct 24, 2025	[Down, Circle, Down]
#1031	SUCCESS	feature/new-feature	ddmunue	Ryan Taylor	1s	Oct 24, 2025	[Down, Circle, Down]

KEY COMPONENTS

- Builds table with 8 columns (Build #, Status, Branch, Commit, Trigger, Duration, Finished, Actions)
- Table header with inverted colors (white on black)
- Build number column with #prefix

PROJECT DETAIL - BUILDS

OVERVIEW

The Builds tab presents a comprehensive view of all build executions for the project in a tabular format. The table shows build number, status with ASCII indicators (✓ success, ✘ failed, ⚡ running, ○ pending), branch name, commit SHA (7 characters), trigger type (push/PR/manual/schedule), duration formatted in minutes and seconds, completion timestamp, and action buttons for viewing logs or retrying failed builds. Running builds display with animated indicators and real-time progress updates.

KEY FEATURES

- Real-time build status updates via buildsStore
- Build table with sortable columns
- ASCII status indicators (✓, ✘, ⚡, ○)
- Clickable build rows navigating to build detail
- View Logs button opening log viewer modal
- Retry button for failed builds (triggers new build)
- Build trigger type identification
- Duration calculation and formatting
- Relative timestamp formatting (5m ago, 2h ago, 3d ago)
- Branch name with monospace styling
- Commit SHA linking to repository
- Running build animation with pulsing indicator
- ...and 2 more features

COMPONENTS

COMPONENT

Builds table with 8 columns (Build #, Status, Branch, Commit, Trigger, Duration, Finished, Actions)
Table header with inverted colors (white on black)
Build number column with #prefix
Status column with ASCII icons and text (SUCCESS/FAILED/RUNNING/PENDING)
Branch name in monospace font
Commit SHA truncated to 7 characters in monospace
Trigger badge (PUSH/PR/MANUAL/SCHEDULE)
Duration formatted as Xm Ys
Timestamp formatted as relative time or full date
Actions column with View Logs and Retry buttons
Running builds with spinner animation (⚡)
Empty state message when no builds exist

Project Detail - Deployments

The screenshot shows the A-SW HUB deployment interface. The left sidebar includes links for DASHBOARD, PROJECTS (selected), BUILDS, TIMELINE, APPROVALS, and SETTINGS. The main header displays the project name "V2X 협업 플랫폼" and maintenance status. The navigation bar includes PROJECTS, MAINTENANCE, OVERVIEW, BUILDS (2), STATISTICS, DEPLOYMENTS (selected), CONTRIBUTORS (2), SETTINGS, PIPELINES, and CONFIGURE. The top right has LOGOUT and a user icon.

DEPLOYMENT HISTORY and **CURRENT ENVIRONMENT STATUS** sections are present. The **ENVIRONMENTS** section lists three environments:

- PRODUCTION**: HEALTHY, Version v2.5.0, Deployed 2 DAYS AGO by DEPLOY BOT. Includes a "ROLLBACK" button.
- STAGING**: DEGRADED, Version v2.6.0-rc1, Deployed 3 HOURS AGO by Sarah Chen. Includes a "ROLLBACK" button.
- DEVELOPMENT**: HEALTHY, Version v2.7.0-dev, Deployed 1 HOUR AGO by Sarah Chen. Includes a "ROLLBACK" button.

RECENT DEPLOYMENTS section lists three recent deployments:

- v2.5.0 → PRODUCTION: 2 DAYS AGO - BY DEPLOY BOT - DURATION: 3M 45S. Status: ✓. Includes a "VIEW DETAILS" link.
- v2.6.0-rc1 → STAGING: 3 HOURS AGO - BY SARAH CHEN - DURATION: 4M 12S. Status: ○. Includes a "VIEW DETAILS" link.
- v2.7.0-dev → DEVELOPMENT: 1 HOUR AGO - BY SARAH CHEN - DURATION: 2M 30S. Status: ✓. Includes a "VIEW DETAILS" link.

A footer at the bottom left shows the user profile: Sarah Chen (ADMIN).

KEY COMPONENTS

- Deployments table with 7 columns (Deployment #, Environment, Version, Status, Deployer, Deployed, Actions)
- Environment badges (PRODUCTION: filled black, STAGING: dashed border, DEVELOPMENT: gray)
- Version numbers in semibold font

PROJECT DETAIL - DEPLOYMENTS

OVERVIEW

The Deployments tab tracks deployment history and current deployment status across all environments. It displays a table of deployments showing deployment ID, environment (Production/Staging/Development), version number, status with indicators, deployer name, deployment timestamp, and action buttons for viewing details or initiating rollback. Active deployments show real-time progress through deployment stages with visual indicators.

KEY FEATURES

- Real-time deployment status tracking
- Multi-environment deployment visibility
- Environment-specific badge styling
- Version number display and tracking
- Deployer information with user details
- Deployment timestamp formatting
- View deployment details with full logs
- One-click rollback for failed deployments
- Rollback confirmation with reason input
- Active deployment progress visualization
- Stage-by-stage deployment tracking
- Deployment history with filtering
- ...and 1 more feature

COMPONENTS

COMPONENT

Deployments table with 7 columns (Deployment #, Environment, Version, Status, Deployer, Deployed, Actions)
Environment badges (PRODUCTION: filled black, STAGING: dashed border, DEVELOPMENT: gray)
Version numbers in semibold font
Status indicators (● DEPLOYED, ⚙ DEPLOYING, ✘ FAILED, ↺ ROLLED BACK)
Deployer name with user icon
Deployment timestamp with relative formatting
Actions buttons (View Details, Rollback)
Active deployment progress bar with stages
Deployment stage indicators (Build, Tests, Deploy, Verify)
Rollback confirmation dialog

Project Detail - Contributors

A-SW HUB

V2X 협업 플랫폼

PROJECTS MAINTENANCE

DASHBOARD PROJECTS BUILDINGS STATISTICS DEPLOYMENTS CONTRIBUTORS (2) SETTINGS PIPELINES CONFIGURE LOGOUT

OVERVIEW

PROJECT CONTRIBUTORS

2 CONTRIBUTORS • 2 TOTAL BUILDS

Emma Johnson

Ryan Taylor

Sarah Chen ADMIN

KEY COMPONENTS

- Contributors grid (auto-fill, minmax(250px, 1fr))
- Contributor cards with avatar, name, role, stats
- Avatar display (image or initials fallback)

PROJECT DETAIL - CONTRIBUTORS

OVERVIEW

The Contributors tab displays all team members and contributors associated with the project in a grid layout. Each contributor card shows their name (or username), role badge (Owner/Maintainer/Developer/Viewer), avatar (or initials if no avatar), contribution statistics (commits, PRs, issues), and last active timestamp. The grid uses a responsive auto-fill layout with minimum 250px card width.

KEY FEATURES

- Team member listing with roles
- Avatar display with initials fallback
- Automatic initials generation
- Role-based badge styling
- Contribution statistics tracking
- Commit count per contributor
- Pull request count
- Issue activity tracking
- Code review participation count
- Last active time tracking
- Relative time formatting (2h ago, 3d ago)
- Responsive grid layout
- ...and 2 more features

COMPONENTS

COMPONENT

Contributors grid (auto-fill, minmax(250px, 1fr))

Contributor cards with avatar, name, role, stats

Avatar display (image or initials fallback)

Initials generation from name (first letters)

Role badge with color coding (Owner: black, Maintainer: dark gray, Developer: gray, Viewer: light gray)

Contribution stats grid (2x2): Commits, PRs, Issues, Reviews

Last active timestamp with relative formatting

Contributor count badge in tab

Empty state for projects without contributors

Project Detail - Settings

The screenshot shows the 'SETTINGS' tab selected in the navigation bar. The main content area is divided into two columns:

- GENERAL SETTINGS**:
 - PROJECT NAME**: V2X 협업 플랫폼
 - DESCRIPTION**: V2X 기반 다수 장비 협업 제어 플랫폼
 - VISIBILITY**: PUBLIC
- BRANCH PROTECTION**:
 - main**:
 - REQUIRE PULL REQUEST
 - REQUIRE STATUS CHECKS
 - REQUIRE 2 APPROVAL(S)
 - ENFORCE FOR ADMINISTRATORS
 - develop**:
 - REQUIRE PULL REQUEST
 - REQUIRE STATUS CHECKS
 - REQUIRE 1 APPROVAL(S)
 - ENFORCE FOR ADMINISTRATORS

KEY COMPONENTS

- Settings summary grid (2 columns)
- Repository Settings card (URL, default branch, protection status)
- Build Settings card (CI/CD enabled, quality gates status, triggers)

PROJECT DETAIL - SETTINGS

OVERVIEW

The Settings tab provides quick access to essential project configuration options directly within the project detail page. It displays a summary of current settings with direct links to full configuration pages. The tab shows repository settings (URL, branch protection status), build settings (CI/CD enabled status, quality gates), deployment settings (auto-deploy toggles per environment), and notification settings (enabled channels). Each settings section includes a 'Configure' button navigating to the detailed configuration page.

KEY FEATURES

- Settings summary overview
- Quick navigation to full configuration
- Repository settings display
- Branch protection status
- CI/CD enablement status
- Quality gates summary
- Build triggers list
- Auto-deploy status per environment
- Notification channels overview
- Quick toggle for common settings
- Configure button navigation
- Status indicator styling
- ...and 1 more feature

COMPONENTS

COMPONENT
Settings summary grid (2 columns)
Repository Settings card (URL, default branch, protection status)
Build Settings card (CI/CD enabled, quality gates status, triggers)
Deployment Settings card (auto-deploy toggles for environments)
Notification Settings card (enabled channels list)
Configure buttons for each settings section
Status indicators (● ENABLED, ○ DISABLED)
Quick toggle switches for binary settings
External link icons for full configuration pages

Project Configuration

The screenshot shows the 'PROJECT CONFIGURATION' page for a project named 'V2X 협업 플랫폼'. The left sidebar has sections: DASHBOARD, PROJECTS (highlighted), BUILD, TIMELINE, APPROVALS, and SETTINGS. The main area has sections: GENERAL, REPOSITORY, BUILD, DEPLOYMENT, NOTIFICATIONS, ADVANCED, and DANGER ZONE. Under GENERAL, there are fields for PROJECT NAME ('V2X 협업 플랫폼'), DESCRIPTION ('V2X 기반 다수 장비 협업 제어 플랫폼'), CATEGORY ('FARM ANALYTICS'), VISIBILITY ('PUBLIC - ANYONE CAN VIEW'), and LICENSE ('MIT LICENSE'). At the top right are 'LOGOUT', 'CANCEL', and 'SAVE CHANGES' buttons. A user profile at the bottom left shows 'Sarah Chen' and 'ADMIN'.

PURPOSE

Advanced project settings interface providing comprehensive configuration for general settings, repository connections, build pipelines, deployment strategies, notifications, and security policies.

KEY COMPONENTS

- Sidebar navigation (250px fixed width) with 7 sections
- Active section indicator with left border
- Form content area with section-specific forms

PROJECT CONFIGURATION

OVERVIEW

The project configuration page allows administrators and project owners to manage all aspects of project settings through a seven-section sidebar navigation. Each section contains detailed forms and controls for specific configuration areas. The interface implements form validation, dirty state tracking, and confirmation dialogs for destructive operations.

PURPOSE

Advanced project settings interface providing comprehensive configuration for general settings, repository connections, build pipelines, deployment strategies, notifications, and security policies.

KEY FEATURES

- Seven-section configuration management
- Form validation with real-time feedback
- Dirty state tracking
- Unsaved changes warning
- Confirmation dialogs for dangerous operations
- Responsive design with collapsible sidebar on mobile

COMPONENTS

COMPONENT

Sidebar navigation (250px fixed width) with 7 sections

Active section indicator with left border

Form content area with section-specific forms

Save Changes button (sticky at bottom)

Unsaved changes indicator

Confirmation dialogs for destructive actions

Project Configuration - General

The screenshot shows the 'PROJECT CONFIGURATION' page under the 'GENERAL' tab. The left sidebar has 'PROJECTS' selected. The main area displays 'GENERAL SETTINGS' with fields for 'PROJECT NAME' (V2X 협업 플랫폼), 'DESCRIPTION' (V2X 기반 다수 장비 협업 제어 플랫폼), 'CATEGORY' (FARM ANALYTICS), 'VISIBILITY' (PUBLIC - ANYONE CAN VIEW), and 'LICENSE' (MIT LICENSE). A bottom navigation bar shows 'Sarah Chen' and 'ADMIN'.

A-SW HUB

PROJECTS

GENERAL

GENERAL SETTINGS

PROJECT NAME

V2X 협업 플랫폼

DESCRIPTION

V2X 기반 다수 장비 협업 제어 플랫폼

CATEGORY

FARM ANALYTICS

VISIBILITY

PUBLIC - ANYONE CAN VIEW

LICENSE

MIT LICENSE

Sarah Chen

ADMIN

KEY COMPONENTS

- Project name input with validation (required, 3-50 characters)
- Slug input with auto-generation and uniqueness check
- Description textarea (max 500 characters)

PROJECT CONFIGURATION - GENERAL

OVERVIEW

The General settings section manages core project metadata including project name, slug (URL-safe identifier), description, category selection, visibility (public/private/internal), license type, default branch, and repository URL. The form includes validation for required fields, unique slug checking, and automatic slug generation from project name.

KEY FEATURES

- Project name editing with validation
- Automatic slug generation from name
- Slug uniqueness validation via API
- Description character counter
- Category selection from predefined list
- Visibility control (public/private/internal)
- License type selection
- Default branch configuration
- Repository URL display (read-only)
- Form validation with inline errors
- Real-time field validation
- Unsaved changes tracking

COMPONENTS

COMPONENT

- | |
|--|
| Project name input with validation (required, 3-50 characters) |
| Slug input with auto-generation and uniqueness check |
| Description textarea (max 500 characters) |
| Category dropdown (11 options) |
| Visibility dropdown (Public/Private/Internal) |
| License dropdown (MIT, Apache 2.0, GPL v3, BSD, Proprietary, None) |
| Default branch input |
| Repository URL input (read-only if connected) |
| Character counter for description |
| Validation error messages |
| Save button with dirty state indicator |

Project Configuration - Repository

The screenshot shows the 'PROJECT CONFIGURATION' screen for a repository. The left sidebar has a 'REPOSITORY' section selected under 'GENERAL'. The main area displays 'REPOSITORY SETTINGS' for a GitHub repository at <https://github.com/asv-hub/v2x-collaboration>. It shows the 'DEFAULT BRANCH' as 'main' and the 'GIT PROVIDER' as 'GITHUB'. Under 'BRANCH PROTECTION', there are three checkboxes: 'REQUIRE PULL REQUEST BEFORE MERGING', 'REQUIRE STATUS CHECKS TO PASS', and 'REQUIRE SIGNED COMMITS', all of which are unchecked.

KEY COMPONENTS

- Repository connection status indicator (● CONNECTED / ○ DISCONNECTED)
- Repository URL display with Test Connection button
- Webhook management section

PROJECT CONFIGURATION - REPOSITORY

OVERVIEW

The Repository section configures Git repository connections, webhook settings, branch protection rules, and commit signing requirements. It displays connection status, allows testing the repository connection, manages webhook creation/deletion, and configures branch protection rules including required reviews, status checks, and restriction policies.

KEY FEATURES

- Repository connection status display
- Connection testing functionality
- Webhook automatic creation
- Webhook URL generation
- Webhook secret management
- Webhook regeneration
- Branch protection configuration
- Protected branches management
- Required reviews count setting
- Status checks requirement
- Required status checks list
- Signed commits enforcement
- ...and 2 more features

COMPONENTS

COMPONENT
Repository connection status indicator (● CONNECTED / ○ DISCONNECTED)
Repository URL display with Test Connection button
Webhook management section
Webhook URL display (generated)
Webhook secret display (masked, copy button)
Regenerate Webhook button
Branch protection rules section
Protected branches list (tags with remove button)
Add Branch input with Add button
Required reviews count input (0-6)
Require status checks before merge checkbox
Required status checks list (add/remove)
Require signed commits checkbox
Include administrators checkbox

Project Configuration - Build

A-SW HUB

V2X 협업 플랫폼

PROJECT CONFIGURATION

LOGOUT

CANCEL

SAVE CHANGES

DASHBOARD

PROJECTS

BUILDS

TIMELINE

APPROVALS

SETTINGS

Sarah Chen
ADMIN

GENERAL

REPOSITORY

BUILD

DEPLOYMENT

NOTIFICATIONS

ADVANCED

DANGER ZONE

INSTALL COMMAND

Command to install dependencies

```
npm install
```

BUILD COMMAND

Command to build your project

```
npm run build
```

TEST COMMAND

Command to run tests

```
npm test
```

OUTPUT DIRECTORY

Directory where build artifacts are generated

```
dist
```

BUILD ENVIRONMENT

Environment variables for build process

◆ ENVIRONMENT VARIABLES CAN BE MANAGED IN THE PROJECT SETTINGS TAB

KEY COMPONENTS

- Build triggers checkbox group (4 options: Push, PR, Schedule, Manual)
- Schedule input (cron syntax) with validation
- Environment variables key-value editor

PROJECT CONFIGURATION - BUILD

OVERVIEW

The Build section configures CI/CD pipeline settings including build triggers (on push, on PR, scheduled, manual), environment variables, build scripts, quality gates with thresholds, and artifact retention policies. It includes a script editor for build commands and comprehensive quality gate configuration for code coverage, test pass rate, security scans, and complexity thresholds.

KEY FEATURES

- Build trigger configuration
- Scheduled builds with cron syntax
- Cron validation and parsing
- Environment variables management
- Variable value masking
- Variable visibility control (secret/visible)
- Build script editing
- Multi-language script support
- Quality gate thresholds
- Code coverage requirement
- Test pass rate requirement
- Security scan enforcement
- ...and 4 more features

COMPONENTS

COMPONENT
Build triggers checkbox group (4 options: Push, PR, Schedule, Manual)
Schedule input (cron syntax) with validation
Environment variables key-value editor
Add Variable button
Variable cards (name, value masked, visibility toggle, delete button)
Build script textarea with line numbers
Script language dropdown (Bash, Python, Node.js, Go)
Quality gates section
Code coverage threshold input (0-100%)
Test pass rate threshold input (0-100%)
Security scan required checkbox
Max complexity threshold input
Artifact retention section
Retention period input (days)
Artifact size limit input (MB)

Project Configuration - Deployment

A-SW HUB

V2X 협업 플랫폼

LOGOUT

CANCEL

SAVE CHANGES

PROJECT CONFIGURATION

DASHBOARD

PROJECTS

BUILDS

TIMELINE

APPROVALS

SETTINGS

Sarah Chen
ADMIN

GENERAL

REPOSITORY

BUILD

DEPLOYMENT

NOTIFICATIONS

ADVANCED

DANGER ZONE

DEPLOYMENT SETTINGS

Configure automatic deployments and environments

AUTO-DEPLOY BRANCH

Branch that triggers automatic deployments

main

DEPLOYMENT ENVIRONMENT

Target environment for deployments

PRODUCTION

DEPLOYMENT COMMAND

Command to deploy your application

npm run deploy

DEPLOYMENT OPTIONS

AUTOMATIC DEPLOYMENTS ON PUSH

PREVIEW DEPLOYMENTS FOR PULL REQUESTS

ROLLBACK ON FAILURE

KEY COMPONENTS

- Environments list (cards)
- Add Environment button
- Environment cards (name, URL, status, Configure button)

PROJECT CONFIGURATION - DEPLOYMENT

OVERVIEW

The Deployment section manages deployment environments, strategies, and policies. It allows adding/removing environments, configuring deployment strategies (rolling, blue-green, canary), setting auto-deploy rules, configuring rollback policies, and managing deployment windows. Each environment can have specific settings for deployment strategy, health checks, and rollback triggers.

KEY FEATURES

- Multiple environment management
- Environment CRUD operations
- Deployment strategy selection
- Auto-deploy configuration
- Branch-based auto-deploy
- Health check configuration
- Health check interval setting
- Rollback policy selection
- Automatic rollback triggers
- Rollback threshold configuration
- Deployment window scheduling
- Maintenance window management
- ...and 2 more features

COMPONENTS

COMPONENT
Environments list (cards)
Add Environment button
Environment cards (name, URL, status, Configure button)
Environment configuration modal
Environment name input
Environment URL input
Deployment strategy dropdown (Rolling, Blue-Green, Canary)
Auto-deploy toggle
Auto-deploy branches input (comma-separated)
Health check URL input
Health check interval input (seconds)
Rollback policy dropdown (Manual, Auto on Failure, Auto on Health Check)
Rollback trigger threshold input
Deployment window section
Allowed deployment times (day/time selection)
Maintenance window scheduling

Project Configuration - Notifications

The screenshot shows the 'PROJECT CONFIGURATION' screen under the 'NOTIFICATIONS' section. The left sidebar includes links for DASHBOARD, PROJECTS, BUILD, APPROVALS, and SETTINGS, with 'SETTINGS' currently selected. The main area displays 'NOTIFICATION SETTINGS' and 'NOTIFICATION CHANNELS' (Email Notifications checked, Slack Notifications unchecked). It also shows 'NOTIFICATION TRIGGERS' (Successful Builds unchecked, Failed Builds checked) and a 'RECIPIENTS' section with the email 'dev-team@example.com' entered. A user profile for 'Sarah Chen' (ADMIN) is visible at the bottom.

KEY COMPONENTS

- Notification channels list
- Email notifications section
- Email enabled toggle

PROJECT CONFIGURATION - NOTIFICATIONS

OVERVIEW

The Notifications section configures notification channels and event subscriptions. It allows enabling/disabling notification channels (Email, Slack, Webhooks), setting up channel-specific configuration (email addresses, Slack webhook URLs, custom webhook endpoints), and selecting which events trigger notifications (build events, deployment events, approval events, issue events, PR events). Each event type can be configured independently per channel.

KEY FEATURES

- Multi-channel notifications
- Email notifications configuration
- Multiple recipient support
- Slack integration
- Slack webhook configuration
- Channel-specific routing
- Custom webhook integration
- Event-based notifications
- Granular event selection
- Per-channel event subscription
- Test notification functionality
- Notification delivery verification
- ...and 2 more features

COMPONENTS

COMPONENT
Notification channels list
Email notifications section
Email enabled toggle
Recipient emails textarea (one per line)
Slack notifications section
Slack enabled toggle
Slack webhook URL input
Slack channel input
Custom webhooks section
Webhook enabled toggle
Webhook URL input
Webhook events checkboxes
Event subscription section
Event categories (Build, Deploy, Approval, Issue, PR)
Event-specific checkboxes per channel
Test notification button per channel

Project Configuration - Advanced

The screenshot shows the 'PROJECT CONFIGURATION' page with the 'ADVANCED' tab selected in the left sidebar. The main content area includes sections for 'ADVANCED SETTINGS', 'AUTOMATION', 'SECURITY', and 'PROJECT METADATA'. The 'AUTOMATION' section contains checkboxes for 'ENABLE AUTO-MERGE FOR PULL REQUESTS' (unchecked), 'REQUIRE STATUS CHECKS BEFORE MERGE' (checked), and 'ENABLE DPENDABOT FOR DEPENDENCY UPDATES' (unchecked). The 'SECURITY' section contains a checkbox for 'ALLOW PRIVATE NPM MODULES' (unchecked). The 'PROJECT METADATA' section displays read-only information: PROJECT ID (PRJ-KITECH-003), SLUG (vzx-collaboration), CREATED (1/1/2027), and LAST UPDATED (10/24/2025). The top right of the page features 'LOGOUT', 'CANCEL', and 'SAVE CHANGES' buttons. The bottom left corner shows the user profile of Sarah Chen (ADMIN).

KEY COMPONENTS

- Project transfer section
- Transfer to user/organization dropdown
- Transfer button with confirmation

PROJECT CONFIGURATION - ADVANCED

OVERVIEW

The Advanced section provides additional configuration options including project transfer, repository mirroring, custom domain setup, API access control, webhook secrets rotation, and integration with external tools. It includes technical settings for advanced users and administrators.

KEY FEATURES

- Project transfer to another owner
- Transfer confirmation workflow
- Repository mirroring
- Mirror synchronization scheduling
- Custom domain configuration
- DNS verification
- Domain SSL certificate
- API rate limiting
- IP whitelist for API access
- API key scope management
- Webhook secret rotation
- Bulk secret regeneration

COMPONENTS

COMPONENT
Project transfer section
Transfer to user/organization dropdown
Transfer button with confirmation
Repository mirroring section
Mirror repository URL input
Mirror sync schedule dropdown
Enable mirroring toggle
Custom domain section
Custom domain input
DNS configuration instructions
Verify domain button
API access section
API rate limit input
Allowed IP addresses textarea
API key scopes checkboxes
Webhook secrets section
Rotate all secrets button

Project Configuration - Danger Zone

The screenshot shows the A-SW HUB interface with the 'PROJECTS' tab selected in the sidebar. The main content area is titled 'PROJECT CONFIGURATION' and contains a 'DANGER ZONE' section. This section is enclosed in a dashed red border and contains three options: 'TRANSFER PROJECT OWNERSHIP', 'ARCHIVE PROJECT', and 'DELETE PROJECT'. Each option has a brief description and a corresponding button ('TRANSFER', 'ARCHIVE', or 'DELETE PROJECT'). At the top right of the danger zone section are 'CANCEL' and 'SAVE CHANGES' buttons. The bottom left corner of the danger zone section shows the user's name, Sarah Chen, and role, ADMIN.

KEY COMPONENTS

- Danger zone container (dashed red border)
- Archive project section
- Archive description text

PROJECT CONFIGURATION - DANGER ZONE

OVERVIEW

The Danger Zone section contains destructive operations including project archiving, project transfer, and project deletion. Each operation requires explicit confirmation and some operations require typing a confirmation phrase. Archived projects become read-only, transfers require recipient acceptance, and deletion is permanent and irreversible.

KEY FEATURES

- Project archiving (makes read-only)
- Archive with confirmation dialog
- Unarchive functionality
- Project transfer to another user
- Transfer with recipient confirmation
- Transfer acceptance workflow
- Project deletion (permanent)
- Delete with typed confirmation
- Confirmation phrase validation
- Multi-step delete confirmation
- Warning about data loss
- List of deletion consequences
- ...and 1 more feature

COMPONENTS

COMPONENT
Danger zone container (dashed red border)
Archive project section
Archive description text
Archive Project button
Archive confirmation dialog
Transfer project section
Transfer to input (username/email)
Transfer Project button
Transfer confirmation with recipient display
Delete project section
Delete consequences warning list
Delete Project button
Delete confirmation dialog
Confirmation phrase input (type project name)
Final Delete button (disabled until phrase matches)

Project Pipelines

The screenshot shows the A-SW HUB Project Pipelines interface. On the left, a sidebar navigation includes DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The main area displays three pipelines: Main CI/CD Pipeline (ACTIVE), Production Deployment, and Nightly Build. The Main CI/CD Pipeline is selected and shown in detail. It has an ACTIVE status, last ran on Oct 24, 2025, at 09:11 PM, and will run again on Oct 26, 2025, at 09:11 PM. It triggers on PUSH (branches: main, develop), PULL_REQUEST (branches: main), and SCHEDULE (CRON: 0 2 * * *). The environment is set to DEVELOPMENT, with EMAIL and SLACK both set to YES. The pipeline stages are: 1 Checkout (git-clone), 2 Build (compile, package), 3 Test (unit-tests, integration-tests), 4 Quality (lint, sonar, security-scan), and 5 Deploy (deploy-staging). Stage 5 depends on stage 4 and has an IF condition: branch == "develop". Recent executions show a successful PUSH run on Oct 24, 2025, at 09:11 PM. The top right corner shows a LOGOUT button.

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

Project Pipelines - Overview

The screenshot shows the A-SW HUB interface for Project Pipelines. The left sidebar includes links for DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The main area displays three pipelines:

- Main CI/CD Pipeline** (ACTIVE): Primary build, test, and deployment pipeline. Last run: 2h ago. Triggers: PUSH (branches: main, develop), PULL_REQUEST (branches: main), SCHEDULE (CRON: 0 2 * * *). Environment: DEVELOPMENT (Email: YES, Slack: YES). Pipeline Stages: Checkout (git-clone), Build (compile, package), Test (unit-tests, integration-tests), Quality (lint, sonar, security-scan), Deploy (deploy-staging). DEPENDS ON: Quality, IF: branch == "develop".
- Production Deployment**: Production-only deployment pipeline with manual approval. Last run: 2d ago.
- Nightly Build**: Automated nightly build for comprehensive testing. Last run: 7d ago.

Recent Executions: PUSH (Oct 24, 2025, 09:11 PM).

KEY COMPONENTS

- Pipeline flow diagram with stage cards
- Stage cards showing name, status icon, duration
- Stage status icons (○ pending, ⚡ running, ✓ success, ✘ failed)

PROJECT PIPELINES - OVERVIEW

OVERVIEW

The Overview tab displays the current pipeline execution status with a visual flow diagram showing all stages, their current status, and execution duration. It includes a pipeline run summary with branch, commit SHA, trigger type, start time, and overall status. Active pipeline runs show real-time progress with animated indicators for running stages.

KEY FEATURES

- Visual pipeline flow representation
- Real-time stage status updates
- Stage-by-stage progress tracking
- Running stage animation (spinner)
- Stage duration calculation
- Overall pipeline status
- Branch and commit display
- Trigger type identification
- Manual pipeline execution
- Cancel running pipeline
- Stage logs quick access
- Pipeline run summary

COMPONENTS

COMPONENT

Pipeline flow diagram with stage cards

Stage cards showing name, status icon, duration

Stage status icons (○ pending, ◁ running, ✓ success, ✘ failed)

Stage connection lines with flow arrows

Pipeline run summary panel

Branch and commit information

Trigger type badge (Push/PR/Manual/Schedule)

Start time and duration

Overall status badge with color coding

Run Pipeline button (primary)

Cancel Pipeline button (running only)

[View Logs](#) link for each stage

Project Pipelines - Configuration

The screenshot shows the A-SW HUB Project Pipelines configuration page. The left sidebar includes links for DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The main area displays three pipelines: Main CI/CD Pipeline (ACTIVE), Production Deployment, and Nightly Build. The Main CI/CD Pipeline is selected, showing its configuration details. The configuration tabs include OVERVIEW, CONFIGURATION (selected), HISTORY (3), and SETTINGS. The TRIGGERS section lists three triggers: TRIGGER 1 (PUSH, branches: main, develop), TRIGGER 2 (PULL_REQUEST, branches: main), and TRIGGER 3 (SCHEDULE, schedule: 0 2 * * *). The STAGES section shows two stages: STAGE 1 (CHECKOUT, 1 JOBS: git-clone) and STAGE 2 (BUILD, 2 JOBS: compile, package). The pipeline status is shown as Primary build, test, and deployment pipeline, last run 2 hours ago. The user profile at the bottom is Sarah Chen, ADMIN.

KEY COMPONENTS

- YAML editor with syntax highlighting
- Line numbers and code folding
- Validation messages panel

PROJECT PIPELINES - CONFIGURATION

OVERVIEW

The Configuration tab provides a YAML editor for defining pipeline stages, jobs, scripts, and dependencies. It includes syntax highlighting, validation, and a visual preview of the pipeline structure. The editor supports defining multiple stages with parallel and sequential jobs, environment variables, artifacts, and caching configuration.

KEY FEATURES

- YAML-based pipeline definition
- Syntax highlighting for YAML
- Real-time validation
- Pipeline structure preview
- Dependency visualization
- Template library access
- Configuration import/export
- Multi-stage pipeline support
- Parallel job execution config
- Environment variables definition
- Artifacts configuration
- Caching rules setup
- ...and 1 more feature

COMPONENTS

COMPONENT
YAML editor with syntax highlighting
Line numbers and code folding
Validation messages panel
Save Configuration button
Pipeline structure preview
Stage list with job counts
Visual dependency graph
Example templates dropdown
Load Template button
Export configuration button
Import configuration button
Validation status indicator

Project Pipelines - History

The screenshot shows the 'PIPLINES' section of the A-SW HUB application. On the left, a sidebar lists navigation options: DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The current user is Sarah Chen (ADMIN).

The main area displays three pipelines:

- Main CI/CD Pipeline** (ACTIVE): Primary build, test, and deployment pipeline. Last run: 2h ago.
- Production Deployment**: Production-only deployment pipeline with manual approval. Last run: 2d ago.
- Nightly Build**: Automated nightly build for comprehensive testing. Last run: 7d ago.

Below these, the 'EXECUTION HISTORY' section shows three total executions:

- SUCCESS** (Main CI/CD Pipeline): Triggered by PUSH, GitHub Webhook, Duration: 5m 0s, Commit: a3f2c1b, Message: feat: Add new authentication module. Last run: Oct 24, 2025, 09:11 PM. [VIEW DETAILS](#)
- FAILED** (Production Deployment): Triggered by MANUAL, John Developer, Duration: 3m 0s, Commit: b2e4d7a, Message: fix: Resolve database connection issue. Last run: Oct 24, 2025, 05:11 PM. [VIEW DETAILS](#)
- SUCCESS** (Nightly Build): Triggered by SCHEDULE, Scheduled Trigger, Duration: 6m 0s, Commit: c7a9f3e, Message: chore: Update dependencies. Last run: Oct 23, 2025, 11:11 PM. [VIEW DETAILS](#)

At the top right, there are buttons for LOGOUT, + NEW PIPELINE, RUN, PAUSE, EDIT, and DELETE.

KEY COMPONENTS

- Filter controls bar (Status, Branch, Date Range, Triggered By)
- Pipeline runs table with 8 columns
- Run number column with #prefix

PROJECT PIPELINES - HISTORY

OVERVIEW

The History tab displays a chronological list of all pipeline executions with filtering capabilities by status, branch, date range, and triggered by user. Each execution entry shows pipeline run number, status, branch, commit SHA, duration, start time, and provides access to full logs and re-run functionality for failed pipelines.

KEY FEATURES

- Pipeline execution history
- Multi-criteria filtering
- Status filtering (all/success/failed/cancelled)
- Branch filtering with autocomplete
- Date range selection
- User-based filtering
- Sortable table columns
- Detailed logs access
- Failed pipeline re-run
- Commit SHA copy
- Pagination support
- Results per page control
- ...and 1 more feature

COMPONENTS

COMPONENT
Filter controls bar (Status, Branch, Date Range, Triggered By)
Pipeline runs table with 8 columns
Run number column with #prefix
Status column with badge and icon
Branch name in monospace
Commit SHA (7 chars) with copy button
Duration formatted as Xh Ym Zs
Start time with relative formatting
Actions column (View Logs, Re-run)
Pagination controls
Results per page selector
Empty state for no history

Project Pipelines - Settings

The screenshot shows the 'A-SW HUB' interface with the 'PIPELINES' tab selected. The main area displays three pipelines: 'Main CI/CD Pipeline' (ACTIVE), 'Production Deployment', and 'Nightly Build'. The 'Main CI/CD Pipeline' is detailed in the center panel, showing its status as 'Primary build, test, and deployment pipeline' with a 'LAST RUN: 2h ago'. The 'SETTINGS' tab is active, revealing sections for 'GENERAL SETTINGS' and 'METADATA'. In 'GENERAL SETTINGS', fields include 'PIPELINE NAME' (Main CI/CD Pipeline), 'DESCRIPTION' (Primary build, test, and deployment pipeline), and 'STATUS' (ACTIVE). In 'METADATA', fields show 'CREATED' (Jul 26, 2025, 11:11 PM) and 'UPDATED' (Oct 23, 2025, 11:11 PM). A 'DANGER ZONE' section is present at the bottom.

KEY COMPONENTS

- Pipeline timeout section
- Global timeout input (minutes)
- Stage timeout input (minutes)

PROJECT PIPELINES - SETTINGS

OVERVIEW

The Settings tab manages pipeline-wide configuration including timeout settings, retry policies, artifact retention, concurrent execution limits, notification rules, and approval requirements. It provides controls for optimizing pipeline performance and reliability.

KEY FEATURES

- Global timeout configuration
- Stage-specific timeouts
- Job-level timeout control
- Automatic retry policies
- Retry attempt limits
- Failure type selection for retry
- Artifact retention management
- Artifact size limits
- Concurrent execution control
- Pipeline queue management
- Success notifications
- Failure notifications
- ...and 3 more features

COMPONENTS

COMPONENT
Pipeline timeout section
Global timeout input (minutes)
Stage timeout input (minutes)
Job timeout input (minutes)
Retry policy section
Max retries input (0-3)
Retry on failure types checkboxes
Artifacts section
Artifact retention period input (days)
Max artifact size input (MB)
Concurrent execution section
Max concurrent pipelines input
Queue behavior dropdown (Wait/Cancel Old)
Notifications section
Notify on success toggle
Notify on failure toggle
Notify users list
Approval requirements section
Require approval toggle
Approvers list

New Project

The screenshot shows the 'CREATE NEW PROJECT' page within the A-SW HUB interface. On the left is a vertical sidebar with navigation links: DASHBOARD (selected), PROJECTS (highlighted in blue), BUILDS, TIMELINE, APPROVALS, and SETTINGS. The main content area has a header 'CREATE NEW PROJECT' with a 'CANCEL' link. It features a 'BASIC INFORMATION' section with the sub-instruction 'Define your project's core details'. The form includes fields for 'PROJECT NAME *' (e.g., Smart Tractor Control), 'PROJECT SLUG *' (e.g., smart-tractor-control), 'DESCRIPTION *' (a text area for project purpose), 'CATEGORY *' (CROP MANAGEMENT), 'VISIBILITY *' (PUBLIC - Anyone can view), and 'LICENSE' (Apache 2.0). A step indicator at the top right shows '1' (active), '2', '3', and '4'. The bottom right corner of the form area shows the user profile 'Sarah Chen ADMIN'. The overall design is clean with a light gray background and white form elements.

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

The screenshot shows the 'BUILD MONITOR' section of the A-SW HUB dashboard. On the left, a sidebar menu includes 'DASHBOARD', 'PROJECTS', **BUILDS**, 'TIMELINE', 'APPROVALS', and 'SETTINGS'. The main area has a title 'BUILD MONITOR' with statistics: 3 ACTIVE, 8 SUCCESS TODAY, 1 FAILED TODAY, and 87% SUCCESS RATE. It features a 'ACTIVE PIPELINES' section with three entries: 'A-SW Hub Platform #1100' (status: 33%, progress bar), 'Path Optimizer #1102' (status: 33%, progress bar), and 'HMI 시스템 #1101' (status: 33%, progress bar). Below this is a 'RECENT BUILDS' table titled 'LAST 10 COMPLETED'.

BUILD	STATUS	BRANCH	DURATION	FINISHED	ACTIONS
A-SW 통합 플랫폼 #1033	FAILED	develop	2s	10:45:15 PM	RETRY VIEW
Git 험업 시스템 #1023	SUCCESS	develop	1s	10:38:56 PM	VIEW
학습 데이터베이스 #1047	SUCCESS	develop	1s	9:41:14 PM	VIEW
V2X 험업 플랫폼 #1049	SUCCESS	develop	1s	9:31:42 PM	VIEW
V2X 험업 플랫폼 #1050	SUCCESS	develop	1s	9:42:41 PM	VIEW

In the bottom right corner of the dashboard area, there is a large black rectangular placeholder for 'BUILD LOGS'.

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)

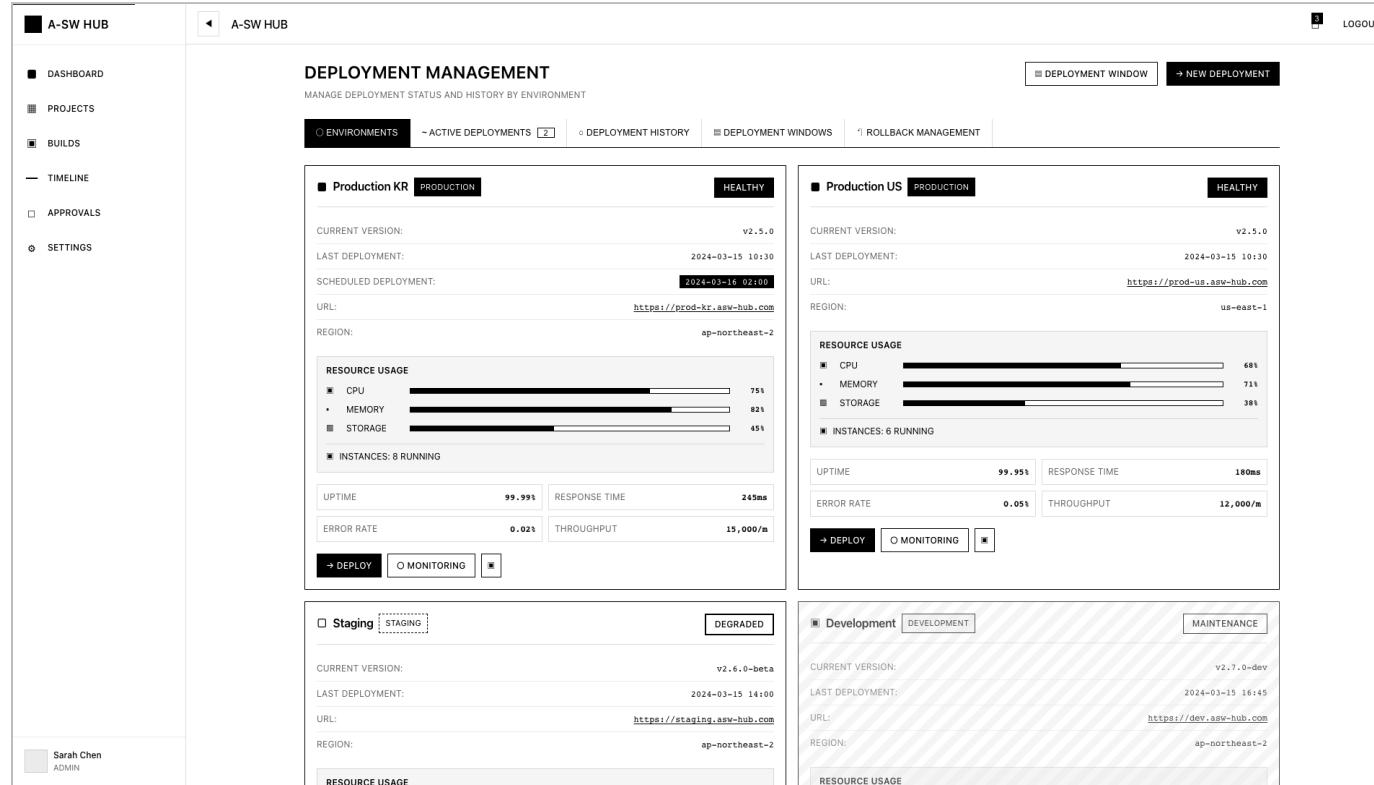
Page 50 / 72
▪ ...and 8 more features

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



The screenshot shows the 'DEPLOYMENT MANAGEMENT' dashboard for the A-SW HUB. The left sidebar includes links for DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS, along with a user profile for Sarah Chen (ADMIN). The main area displays four environment cards: Production KR (Production, HEALTHY), Production US (Production, HEALTHY), Staging (Staging, DEGRADED), and Development (Development, MAINTENANCE). Each card provides deployment details (version, last deployment, URL, region), resource usage (CPU, Memory, Storage), and performance metrics (Uptime, Response Time, Error Rate, Throughput). Action buttons include 'DEPLOY', 'MONITORING', and 'ROLLBACK'.

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)

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

The screenshot shows the 'APPROVAL QUEUE' page of the A-SW HUB. On the left is a sidebar with navigation links: DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS (which is selected and highlighted in grey), and SETTINGS. The main area has a header 'A-SW HUB' and a 'LOGOUT' button. Below the header is a stats card showing 1 URGENT, 3 PENDING, 0 APPROVED TODAY, and 3 REQUESTS. The main content area is divided into two columns. The left column lists three deployment requests with their priority levels (URGENT, HIGH, MEDIUM) and creation times (30m ago, 2h ago, 4h ago). Each request card contains a summary, project details, and a row of status indicators (Security, Tests, Coverage, Performance, Code Quality) with checkmarks or error icons. The right column displays 'APPROVAL DETAILS' for each request, including 'QUALITY GATE CHECKS' (Security 100/100, Tests 342/342 passed, Coverage 78/80 below threshold, Performance 95/90, Code Quality 85/80), 'CHANGES' (list of modified files like auto_pilot.py, patch.py, requirements.txt), and 'REQUEST INFORMATION' (Deployment: URGENT, Environment: PRODUCTION, Project: smart-tractor-control, Requested By: John Doe, Requested At: 10/24/2025, Build ID: build123). Approval buttons ('APPROVE', 'REJECT') are located at the bottom of each request card.

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

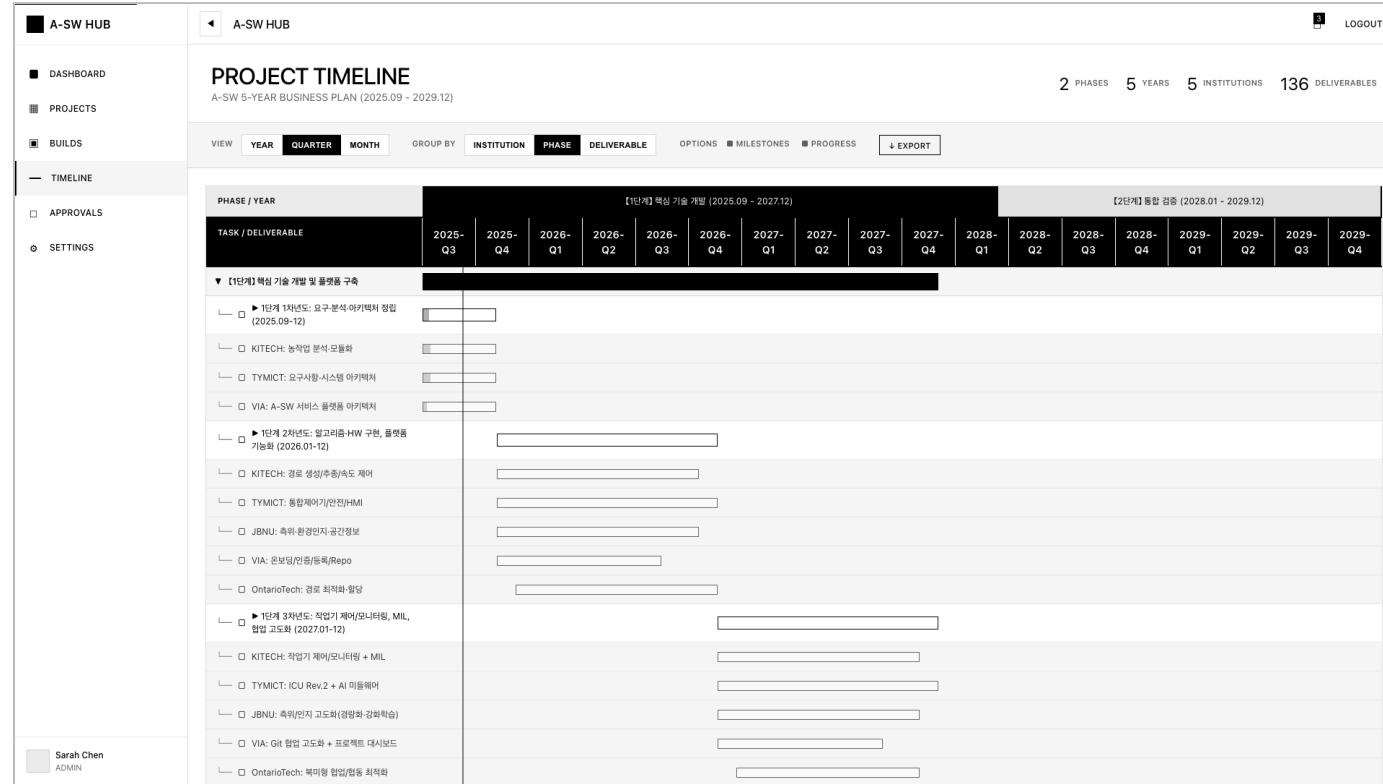
Change type labels (ADDED: foreground color, MODIFIED: muted, DELETED: strikethrough)

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, TYSMCT, 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

The screenshot shows the 'QUALITY GATE' dashboard for the project 'SMART-TRACTOR-CONTROL'. The overall score is 82 (B+). Key metrics include Code Quality (85%, 12 issues), Security (72%, 3 high vulnerabilities), Test Coverage (78%, 9716/2456 lines), and Dependencies (94%, 8 outdated packages). A bar chart shows the quality trend over 30 days. Below, there are sections for Quality Criteria (Minimum Coverage: 70%, PASS) and Security Vulnerabilities (0 High, 0 Fail).

A-SW HUB

LOGOUT

QUALITY GATE

PROJECT SMART-TRACTOR-CONTROL

Last analyzed: 10/24/2025, 11:11:46 PM

OVERALL SCORE: 82 (B+)

CODE QUALITY: 85% (12 issues)

SECURITY: 72% (3 high vulnerabilities)

TEST COVERAGE: 78% (9716/2456 lines)

DEPENDENCIES: 94% (8 outdated packages)

OVERVIEW CODE QUALITY SECURITY COVERAGE DEPENDENCIES

QUALITY TREND (30 DAYS)

QUALITY CRITERIA

Minimum Coverage: 70% ✓ PASS

Security Vulnerabilities: 0 High ✘ FAIL

Sarah Chen
ADMIN

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

The screenshot shows the 'SETTINGS' page under the 'A-SW HUB' header. The left sidebar has tabs for DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS, with SETTINGS selected. The main content area is titled 'USER PROFILE' with the subtitle 'MANAGE YOUR ACCOUNT INFORMATION'. It contains fields for USERNAME ('admin'), EMAIL ADDRESS ('admin@asw-hub.kr'), FULL NAME ('System Administrator'), DEPARTMENT ('IT OPERATIONS'), ROLE ('SYSTEM ADMIN'), PHONE NUMBER ('+82-10-0000-0000'), LOCATION ('SEOUL, KR'), and JOIN DATE ('2024-01-01'). Below these is a BIO field containing 'Agricultural software platform administrator'. At the top right are 'LOGOUT' and 'SAVE CHANGES' buttons. At the bottom left is a user profile card for 'Sarah Chen' (ADMIN).

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

System Settings - Profile

A-SW HUB

A-SW HUB

LOGOUT

SETTINGS

SYSTEM CONFIGURATION AND USER PREFERENCES

▶ SAVE CHANGES

USER PROFILE

MANAGE YOUR ACCOUNT INFORMATION

USERNAME	EMAIL ADDRESS
admin	admin@asw-hub.kr
FULL NAME	DEPARTMENT
System Administrator	IT OPERATIONS
ROLE	PHONE NUMBER
SYSTEM ADMIN	+82-10-0000-0000
LOCATION	JOIN DATE
SEOUL, KR	2024-01-01
BIO	
Agricultural software platform administrator	

Sarah Chen
ADMIN

KEY COMPONENTS

- 8 input fields in 2-column grid
- Username input (unique, 3-20 chars, alphanumeric + underscore)
- Email input (with validation)

SYSTEM SETTINGS - PROFILE

OVERVIEW

The Profile tab manages user information including username (unique identifier), email address, full name, department, role (read-only, assigned by admin), phone number, location, join date (read-only), and bio (max 500 characters). The form uses a 2-column grid layout with full-width bio field at the bottom.

KEY FEATURES

- User profile editing
- Username uniqueness validation
- Email format validation
- Phone number format validation
- Bio character counter
- Read-only fields for role and join date
- 2-column responsive grid
- Form validation with inline errors
- Unsaved changes detection

COMPONENTS

COMPONENT
8 input fields in 2-column grid
Username input (unique, 3-20 chars, alphanumeric + underscore)
Email input (with validation)
Full name input
Department input
Role display (read-only with badge)
Phone number input (with format validation)
Location input
Join date display (read-only, formatted)
Bio textarea (full width, max 500 chars)
Character counter for bio
Save button at bottom

System Settings - Security

The screenshot shows the 'SETTINGS' section of the A-SW HUB interface. The left sidebar has a 'SETTINGS' tab selected. The main area displays 'SECURITY SETTINGS' with sections for 'PASSWORD' (last changed: 2024-01-01, minimum length: 12 characters), 'TWO-FACTOR AUTHENTICATION' (status: ENABLED, method: APP), and 'API KEYS' (two entries: 'CI/CD Pipeline' and 'Monitoring Service', each with a 'REVOKE' button). The 'ACTIVE SESSIONS' section shows a single session for 'MacBook Pro' (Seoul, South Korea | 192.168.1.100 | Chrome 120.0, last active: 2025-10-24). A 'SAVE CHANGES' button is at the top right.

KEY COMPONENTS

- Password section: last changed date, Change Password button, collapsible form (current, new, confirm)
- Password strength indicator
- Two-Factor Auth section: status toggle, method display, Enable/Disable button

SYSTEM SETTINGS - SECURITY

OVERVIEW

The Security tab handles password changes, two-factor authentication, API key management, and active session monitoring. Password section shows last changed date and includes a collapsible form for changing password. Two-factor authentication can be enabled/disabled with method selection (TOTP/SMS/App). API keys section allows generating new keys with custom permissions and expiration dates, displaying existing keys with masked values, and revoking keys. Active sessions section lists all current sessions with device, location, IP, browser info and allows terminating individual sessions.

KEY FEATURES

- Password change with strength validation
- Current password verification
- Two-factor authentication setup
- TOTP/SMS/App method selection
- QR code generation for TOTP
- Backup codes generation
- API key creation with custom permissions
- API key expiration dates
- API key masking for security
- API key copy to clipboard
- API key revocation
- Active session monitoring
- ...and 3 more features

COMPONENTS

COMPONENT

- | |
|---|
| Password section: last changed date, Change Password button, collapsible form (current, new, confirm) |
| Password strength indicator |
| Two-Factor Auth section: status toggle, method display, Enable/Disable button |
| 2FA setup wizard with QR code for TOTP |
| API Keys section: New Key button, keys list, Revoke buttons |
| API key generation form: name input, permissions checkboxes (6 scopes), expiration date |
| API key cards: name, masked key (first 10 + last 4), created/expires dates, Revoke button |
| Copy key to clipboard button |
| Active Sessions section: sessions list, End Session buttons |
| Session cards: device, location, IP, browser, last active, current badge |

System Settings - Notifications

The screenshot shows the 'NOTIFICATIONS' section of the 'SETTINGS' page in the A-SW HUB. The left sidebar includes links for DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The SETTINGS link is currently selected. The main content area is titled 'NOTIFICATION PREFERENCES' with the subtitle 'CONFIGURE HOW YOU RECEIVE ALERTS'. It features four sections: EMAIL, PUSH, SLACK, and DO NOT DISTURB. Each section contains a list of event types with checkboxes for enabling or disabling them. A 'SAVE CHANGES' button is located in the top right corner.

NOTIFICATION PREFERENCES
CONFIGURE HOW YOU RECEIVE ALERTS

EMAIL

- ENABLED
- BUILDS
- APPROVALS
- UPDATES
- REVIEWS
- DEPLOYMENTS
- SECURITY
- MENTIONS

PUSH

- ENABLED

SLACK

- ENABLED
- BUILDS
- APPROVALS
- UPDATES
- REVIEWS
- DEPLOYMENTS
- SECURITY
- MENTIONS

DO NOT DISTURB

- ENABLED

Sarah Chen
ADMIN

KEY COMPONENTS

- 4 channel sections (Email, Slack, In-app, SMS)
- Channel enable toggle for each
- Event subscription grid (2 columns) per channel

SYSTEM SETTINGS - NOTIFICATIONS

OVERVIEW

The Notifications tab configures multi-channel notification preferences including Email, Slack, In-app, and SMS channels. Each channel can be enabled/disabled and configured with event-specific subscriptions. Users can select which events trigger notifications (builds, deployments, approvals, issues, mentions) per channel. A Do Not Disturb feature allows scheduling quiet periods with start and end times.

KEY FEATURES

- Multi-channel notifications
- Per-channel enable/disable
- Granular event selection
- Build notifications (started/completed/failed)
- Deployment notifications
- Approval workflow notifications
- Issue tracking notifications
- Mention notifications
- Do Not Disturb scheduling
- Timezone-aware DND
- Test notification functionality
- Channel-specific event subscriptions

COMPONENTS

COMPONENT
4 channel sections (Email, Slack, In-app, SMS)
Channel enable toggle for each
Event subscription grid (2 columns) per channel
Event checkboxes: Builds (started/completed/failed), Deployments (started/completed/failed), Approvals (requested/approved/rejected), Issues (opened/assigned/closed), Mentions
Do Not Disturb section
DND enable toggle
Start time input (HH:MM)
End time input (HH:MM)
Timezone display
Test notification button per channel

System Settings - Preferences

The screenshot shows the 'SETTINGS' page under the 'A-SW HUB' header. The left sidebar includes links for DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The SETTINGS link is currently selected. The main content area is titled 'USER PREFERENCES' with the subtitle 'CUSTOMIZE YOUR EXPERIENCE'. It contains several dropdown menus for configuration:

- THEME: DARK (selected)
- LANGUAGE: ENGLISH
- TIMEZONE: SEOUL
- DATE FORMAT: YYYY-MM-DD
- TIME FORMAT: 24 HOUR
- FIRST DAY OF WEEK: MONDAY
- NUMBER FORMAT: 1,000.00
- DEFAULT VIEW: DASHBOARD

A 'SAVE CHANGES' button is located in the top right corner. In the bottom left corner of the main area, there is a user profile for 'Sarah Chen' (ADMIN).

KEY COMPONENTS

- Theme dropdown (Dark/Light/System)
- Language dropdown (4 languages)
- Timezone dropdown (50+ major cities)

SYSTEM SETTINGS - PREFERENCES

OVERVIEW

The Preferences tab customizes UI and regional settings including theme selection (Dark/Light/System), language (EN/KO/JA/ZH), timezone (major cities), date format (YYYY-MM-DD/DD-MM-YYYY/MM-DD-YYYY), time format (12h/24h), first day of week, number format, and default landing page view. All settings have immediate preview.

KEY FEATURES

- Theme selection (dark/light/system)
- Multi-language support
- Timezone configuration
- Date format customization
- Time format selection
- First day of week preference
- Number format localization
- Default landing page
- Real-time preview
- Format examples display

COMPONENTS

COMPONENT

- | |
|--|
| Theme dropdown (Dark/Light/System) |
| Language dropdown (4 languages) |
| Timezone dropdown (50+ major cities) |
| Date format dropdown (3 formats with examples) |
| Time format dropdown (12h/24h with examples) |
| First day of week dropdown (Sunday/Monday) |
| Number format dropdown (1,234.56 / 1.234,56 / 1 234,56) |
| Default view dropdown (Dashboard/Projects/Builds/Timeline) |
| Preview section showing current selections |

System Settings - Integrations

The screenshot shows the 'SETTINGS' page under the 'INTEGRATIONS' section. The left sidebar has a 'SETTINGS' tab selected. The main area displays five integration cards:

- GITHUB**: Connected. Details: USERNAME asw-hub, ORGANIZATIONS asw-org, WEBHOOKS https://api.asw-hub.kr/webhooks/github. Buttons: CONFIGURE, DISCONNECT.
- GITLAB**: Disconnected. Button: CONNECT.
- JENKINS**: Connected. Details: URL https://jenkins.asw-hub.kr, USERNAME admin. Buttons: CONFIGURE, DISCONNECT.
- SLACK**: Connected. Details: WORKSPACE ASW-HUB, CHANNEL #deployments, WEBHOOKURL https://hooks.slack.com/services/xxx. Buttons: CONFIGURE, DISCONNECT.
- JIRA**: Disconnected. Button: CONNECT.
- SONARQUBE**: Connected. Details: URL https://sonar.asw-hub.kr, TOKEN sqp_XXXXXXXXXX, QUALITYGATE ASW Quality Gate. Buttons: CONFIGURE, DISCONNECT.

At the top right are 'LOGOUT' and 'SAVE CHANGES' buttons. At the bottom left is a user profile for 'Sarah Chen'.

KEY COMPONENTS

- 5 integration cards in 2-column grid
- Integration cards: name, status (● CONNECTED / ○ DISCONNECTED), info display, action buttons
- GitHub integration: organization, repositories count, Connect/Disconnect

SYSTEM SETTINGS - INTEGRATIONS

OVERVIEW

The Integrations tab manages connections to external services including GitHub, GitLab, Jenkins, Slack, and SonarQube. Each integration card shows connection status, configured values, and Connect/Disconnect buttons. Configuration panels allow entering credentials, URLs, and testing connections. Integration status is tracked and displayed with visual indicators.

KEY FEATURES

- External service integrations
- OAuth authentication support
- Connection status tracking
- GitHub integration (repos, webhooks)
- GitLab integration (projects, pipelines)
- Jenkins integration (jobs, builds)
- Slack integration (notifications, commands)
- SonarQube integration (quality metrics)
- Connection testing
- Credential management
- Disconnect with confirmation

COMPONENTS

COMPONENT
5 integration cards in 2-column grid
Integration cards: name, status (● CONNECTED / ○ DISCONNECTED), info display, action buttons
GitHub integration: organization, repositories count, Connect/Disconnect
GitLab integration: instance URL, groups count, Connect/Disconnect
Jenkins integration: server URL, jobs count, Connect/Disconnect
Slack integration: workspace name, channel, Connect/Disconnect
SonarQube integration: server URL, projects count, Connect/Disconnect
Connection configuration modals
OAuth authentication flows
Test connection buttons

System Settings - Data & Privacy

The screenshot shows the 'SETTINGS' page under the 'DATA' section of the A-SW HUB. The left sidebar includes links for Dashboard, Projects, Builds, Timeline, Approvals, and Settings. The main content area has a header 'SETTINGS' and 'SYSTEM CONFIGURATION AND USER PREFERENCES'. It features sections for 'DATA & PRIVACY' (Manage your data and privacy settings), 'DATA RETENTION' (with fields for Logs (Days) set to 30, Builds (Days) set to 90, Metrics (Days) set to 365, and Artifacts (Days) set to 7), 'DATA EXPORT' (Request Data Export button), 'EXPORT HISTORY' (JSON file from 2024-10-01 completed, with a download link), 'PRIVACY SETTINGS' (checkboxes for Data Processing Agreement, Marketing Emails, and Analytics Tracking, all checked), and a 'DANGER ZONE' section (Irreversible Actions - Proceed with caution). A 'LOGOUT' button is in the top right.

KEY COMPONENTS

- Data Retention section: 4 number inputs (days) for builds, logs, artifacts, analytics
- Data Export section: Request Export button, format selection dialog, export history table
- Export history table: Format, Requested At, Status, Download button

SYSTEM SETTINGS - DATA & PRIVACY

OVERVIEW

The Data & Privacy tab manages data retention policies, data export requests, and privacy settings. Users can configure retention periods for different data types (builds, logs, artifacts, analytics), request data exports in various formats (JSON/CSV/XML), view export history, and manage GDPR settings. A Danger Zone section allows account deletion with confirmation.

KEY FEATURES

- Data retention policy configuration
- Per-type retention periods
- Data export requests
- Multi-format exports (JSON/CSV/XML)
- Export history tracking
- Export download links
- GDPR compliance settings
- Data processing agreement
- Marketing email opt-in/out
- Analytics tracking control
- Data subject access requests
- Account deletion
- ...and 2 more features

COMPONENTS

COMPONENT

Data Retention section: 4 number inputs (days) for builds, logs, artifacts, analytics

Data Export section: Request Export button, format selection dialog, export history table

Export history table: Format, Requested At, Status, Download button

Privacy Settings section: 3 checkboxes (data processing agreement, marketing emails, analytics tracking)

GDPR section: data subject rights information, request data access button

Danger Zone: dashed red border, Delete Account button

Delete confirmation dialog: warning text, type 'DELETE MY ACCOUNT' input, final Delete button