

A-SW Platform

presented by VIASOFT.AI

LEAD INSTITUTION

KITECH

Korea Institute of Industrial Technology

Korea Institute of Industrial Technology

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

PLATFORM OVERVIEW

An integrated development environment for autonomous agricultural machinery, providing CI/CD pipelines, build management, deployment automation, and quality assurance features for collaborative development.

Role

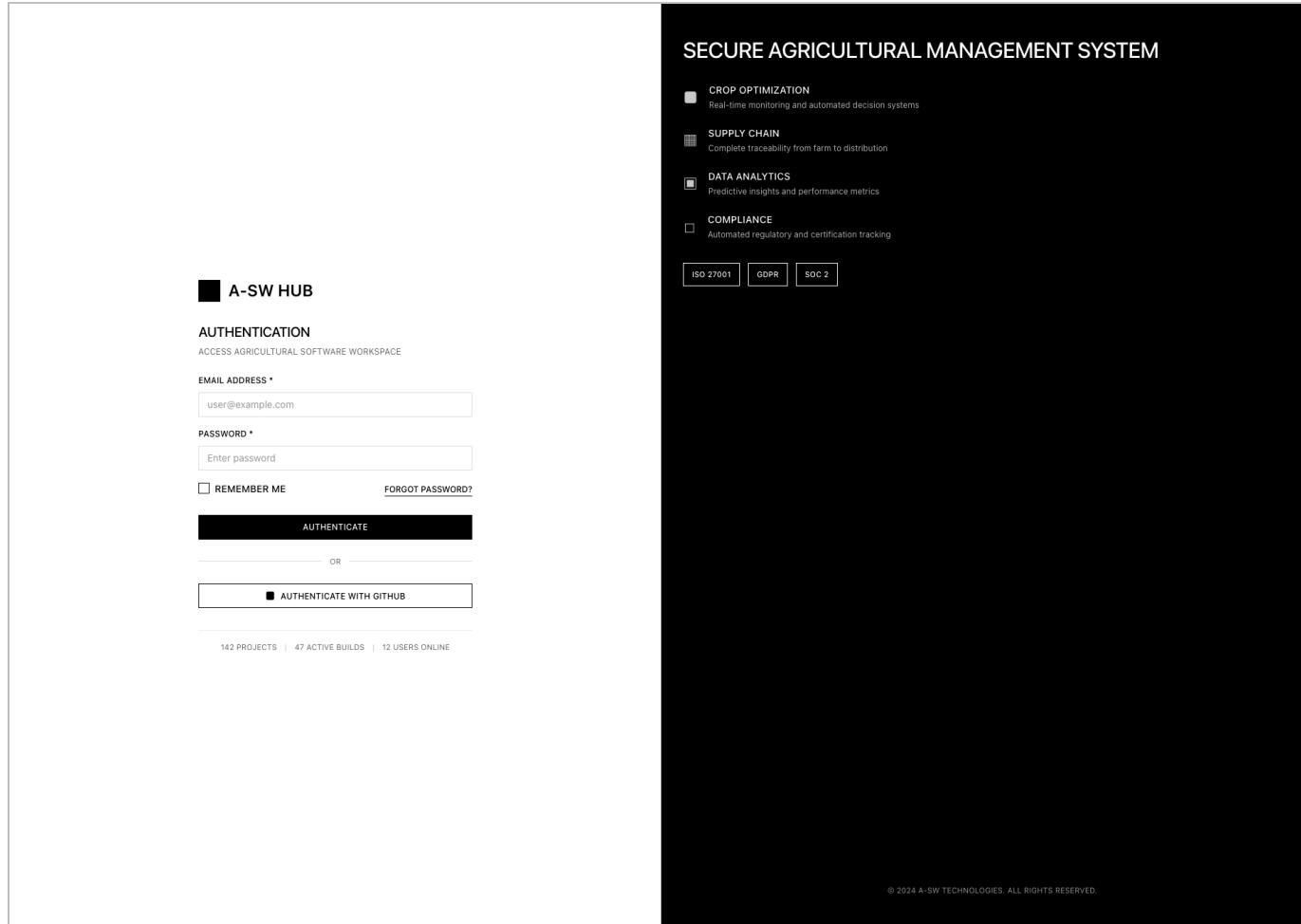
Administrator

Document Date: October 24, 2025

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



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
- ...and 1 more feature

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

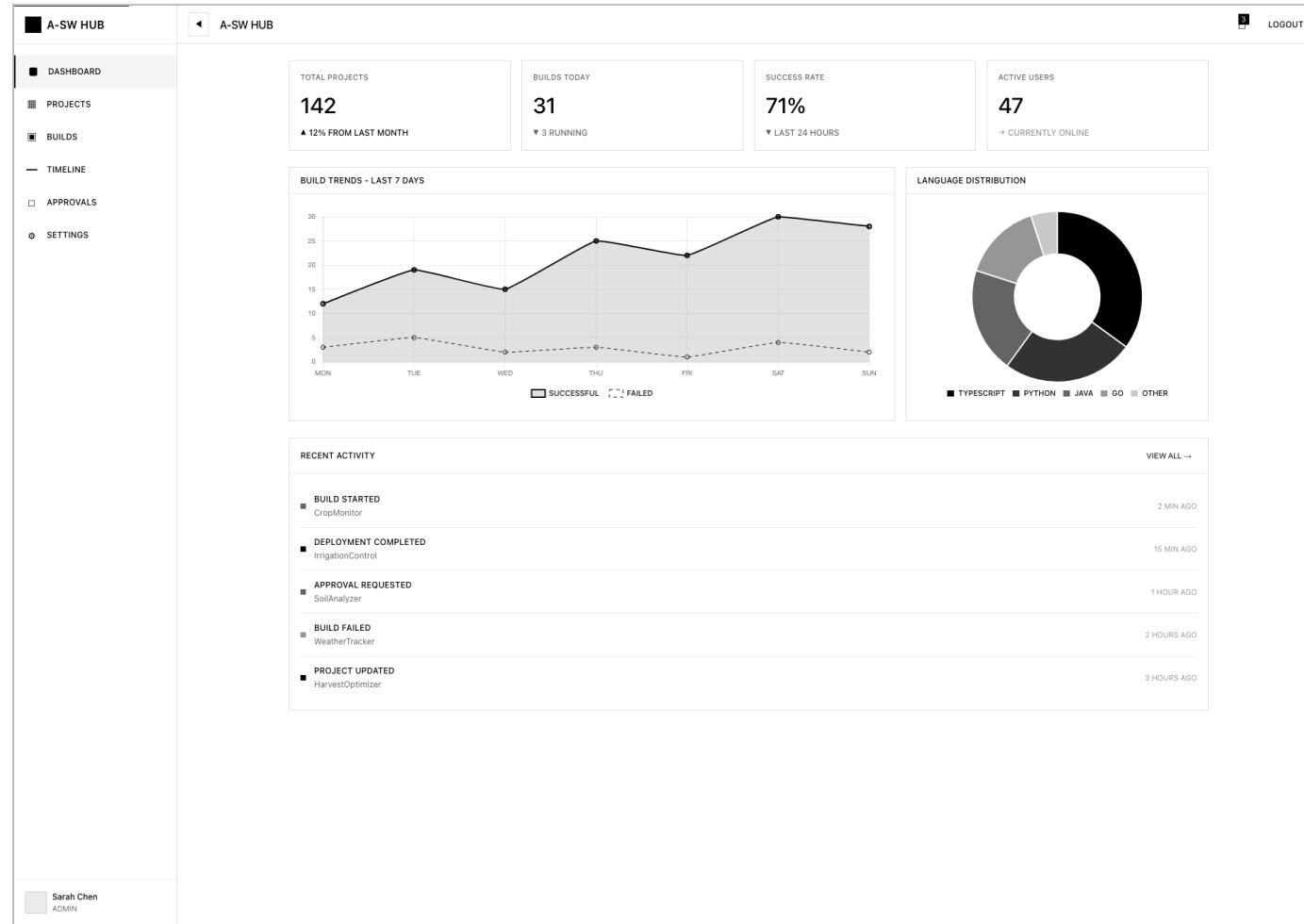
Forgot password link with underline styling

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
- ...and 1 more feature

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		
A-SW HUB	DASHBOARD	SEARCH			CATEGORY	STATUS	LANGUAGE	+ NEW PROJECT
	PROJECTS	Search projects...	ALL	ALL	ALL	ALL	ALL	15 PROJECTS
	BUILDS	Go	Python	Python	Java			
	V2X 협업 플랫폼	Path Optimizer	Path Optimizer	학습 데이터베이스	Task Scheduler			
	V2X 기반 다수 장비 협업 제어 플랫폼	Multi-agent path optimization for agricultural machinery		농업 AI 학습용 대규모 데이터셋 및 라벨링 플랫폼	Cooperative task allocation and scheduling system			
	FARM ANALYTICS	HARVEST OPTIMIZATION	FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS			
	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	SUCCESS
	Python	Python	C++	Go				
	측위 알고리즘	농작업 분석 모듈	통합제어기(ICU)	Git 협업 시스템				
	고정밀 측위 및 SLAM 알고리즘	무인차량 농작업 분석 및 A-SW 모듈화 파라미터 정립 시스템	농업용 통합제어기 하드웨어 및 소프트웨어 플랫폼	Git 기반 A-SW 저장소 및 협업 워크플로우				
A-SW HUB	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	SUCCESS
	Python	C++	TypeScript	TypeScript				
	AI 미들웨어	경로 제어 시스템	인증/권한 시스템	A-SW 통합 플랫폼				
	멀티모달 AI 융합 미들웨어 플랫폼	경로 생성/추종/속도 제어 통합 시스템	OAuth2/OIDC 기반 통합 인증 시스템	무인차량 농작업 소프트웨어 통합 플랫폼 - 1단계 3개년 + 2단계 2개년 국가 R&D 사업				
	FARM ANALYTICS	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	SUCCESS
	TypeScript	Python	TypeScript					
	HMI 시스템	환경인지 모듈	A-SW Hub Platform					
	농업용 듀얼 Human-Machine Interface 시스템	딥러닝 기반 농업환경 인지 시스템	개방형 A-SW 공동 및 협업 플랫폼					
Sarah Chen ADMIN	FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS	FARM ANALYTICS			
	STARS 234 FORKS 56 CONTRIB 15 UPDATED 10/24/2025	SUCCESS	STARS 412 FORKS 123 CONTRIB 12 UPDATED 10/24/2025	RUNNING	STARS 567 FORKS 156 CONTRIB 25 UPDATED 10/24/2025	SUCCESS		

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
- ...and 6 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 'A-SW HUB' project detail page. The top navigation bar includes 'LOGOUT' and tabs for 'PIPELINES' and 'CONFIGURE'. The left sidebar has sections for DASHBOARD, PROJECTS (selected), BUILD, TIMELINE, APPROVALS, and SETTINGS. The main content area has tabs for OVERVIEW (selected), BUILDS (3), STATISTICS, DEPLOYMENTS, CONTRIBUTORS (3), and SETTINGS. Action buttons include 'CLONE', 'STAR (89)', 'FORK (23)', and 'DEPLOY'. The 'PROJECT INFORMATION' section shows Category: V2X 협업 플랫폼, Language: Go, Visibility: PUBLIC, License: MIT, and Created: Jan 1, 2027. The 'REPOSITORY' section shows URL: https://github.com/asw-hub/v2x-collaboration, Default Branch: main, Branches: 2, Last Commit: Oct 14, 2025. The 'METRICS' section shows 89 stars, 23 forks, 6 contributors, and 75% coverage. The 'RECENT COMMITS' section lists three commits by James Wilson, Lisa Anderson, and Emma Johnson. At the bottom, there is a user profile for 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 shows the 'OVERVIEW' tab of the project details. The top navigation bar includes 'A-SW HUB' (with a dropdown), 'PROJECTS' (selected), 'MAINTENANCE' (disabled), 'LOGOUT', 'PIPELINES', and 'CONFIGURE'. The left sidebar has links for 'DASHBOARD', 'PROJECTS' (selected), 'BUILDS', 'TIMELINE', 'APPROVALS', and 'SETTINGS'. The main content area has tabs for 'OVERVIEW' (selected), 'BUILD (3)', 'STATISTICS', 'DEPLOYMENTS', 'CONTRIBUTORS (3)', and 'SETTINGS'. Below these are four buttons: 'CLONE', 'STAR (89)', 'FORK (23)', and 'DEPLOY'. The 'PROJECT INFORMATION' section shows: Category (Farm Analytics), Language (Go), Visibility (PUBLIC), License (MIT), and Created (Jan 1, 2027). The 'REPOSITORY' section shows: URL (<https://github.com/asw-hub/v2x-collaboration>), Default Branch (main), Branches (2), Last Commit (Oct 14, 2025). The 'METRICS' section shows: Stars (89), Forks (23), Contributors (6), and Coverage (75%). The 'DESCRIPTION' section contains the text: 'V2X 기반 다수 장비 협업 제어 플랫폼'. The 'RECENT COMMITS' section lists three commits: 1. '464axt5' by James Wilson, 'Improve error handling in harvest module', 14h ago. 2. 'obeyy8q' by Lisa Anderson, 'Optimize image processing algorithms', 17h ago. 3. 'qxs7oya' by Emma Johnson, 'Improve error handling in harvest module', 18h ago. At the bottom, there is a user profile for '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
- ...and 2 more features

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 the project name 'V2X 협업 플랫폼' and maintenance status. The navigation bar at the top has tabs for Overview, Builds (3), Statistics, Deployments, Contributors (3), and Settings. Below this is a search bar and filter options for branches, status, and users. The central area shows a table of build history with columns: #, Status, Branch, Commit, Triggered By, Duration, Date, and Actions. Three builds are listed: #1011, #1040, and #1023, all marked as SUCCESS. The bottom right corner of the sidebar shows the user profile 'Sarah Chen' and 'ADMIN'.

#	Status	Branch	Commit	Triggered By	Duration	Date	Actions
#1011	SUCCESS	main	464axt5	James Wilson	1s	Oct 24, 2025	[Actions]
#1040	SUCCESS	main	obeyeq	Lisa Anderson	1s	Oct 23, 2025	[Actions]
#1023	SUCCESS	feature/new-feature	qxs7oya	Emma Johnson	1s	Oct 23, 2025	[Actions]

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)
- ...and 5 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), BUILD, TIMELINE, APPROVALS, and SETTINGS. The main header has tabs for PROJECTS (selected), V2X 협업 플랫폼 (Maintenance), PIPELINES, and CONFIGURE. The top navigation bar includes LOGOUT, BACK, and FORWARD buttons.

DEPLOYMENT HISTORY and **CURRENT ENVIRONMENT STATUS** sections are present. The environment status cards show:

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

RECENT DEPLOYMENTS section lists:

- v2.5.0 → PRODUCTION (2 DAYS AGO, BY DEPLOY BOT, DURATION: 3M 45S) with a [VIEW DETAILS](#) link.
- v2.6.0-rc1 → STAGING (3 HOURS AGO, BY SARAH CHEN, DURATION: 4M 12S) with a [VIEW DETAILS](#) link.
- v2.7.0-dev → DEVELOPMENT (1 HOUR AGO, BY SARAH CHEN, DURATION: 2M 30S) with a [VIEW DETAILS](#) link.

A user profile for Sarah Chen (ADMIN) is at the bottom left.

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
- ...and 4 more features

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 (3) SETTINGS PIPELINES CONFIGURE

OVERVIEW

PROJECT CONTRIBUTORS

3 CONTRIBUTORS • 3 TOTAL BUILDS

J James Wilson
1 1
COMMITS BUILDS
◆ LAST ACTIVE: 14h ago

L Lisa Anderson
1 1
COMMITS BUILDS
◆ LAST ACTIVE: 17h ago

E Emma Johnson
1 1
COMMITS BUILDS
◆ LAST ACTIVE: 18h ago

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
- ...and 5 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 several sections:

- GENERAL SETTINGS**: Includes fields for PROJECT NAME (V2X 협업 플랫폼), DESCRIPTION (V2X 기반 다수 장비 협업 제어 플랫폼), and VISIBILITY (PUBLIC).
- BRANCH PROTECTION**: Configurable protection rules for branches like main and develop. For 'main', requirements include REQUIRE PULL REQUEST, REQUIRE STATUS CHECKS, REQUIRE 2 APPROVAL(S), and ENFORCE FOR ADMINISTRATORS. For 'develop', requirements include REQUIRE PULL REQUEST, REQUIRE STATUS CHECKS, and REQUIRE 1 APPROVAL(S). A '+ ADD PROTECTION RULE' button is available.
- ENVIRONMENT VARIABLES**: Manages environment variables for the project. Current entries include API_URL (https://api.example.com) and DATABASE_URL (*****).

The sidebar on the left lists project management sections: DASHBOARD, PROJECTS (selected), BUILDS, TIMELINE, APPROVALS, and SETTINGS. The bottom left corner shows the user profile of Sarah Chen (ADMIN).

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
- ...and 4 more features

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 within the A-SW HUB application. The left sidebar contains navigation links for Dashboard, Projects (selected), Builds, Timeline, Approvals, and Settings. The main content area is titled 'PROJECT CONFIGURATION' and shows the 'GENERAL' settings for project 'V2X 협업 플랫폼'. The 'GENERAL SETTINGS' section includes fields for 'PROJECT NAME' (V2X 협업 플랫폼) and 'DESCRIPTION' (V2X 기반 다수 장비 협업 제어 플랫폼). Below these are sections for 'CATEGORY' (FARM ANALYTICS) and 'VISIBILITY' (PUBLIC - ANYONE CAN VIEW). A 'LICENSE' section shows 'MIT LICENSE' selected. At the top right are 'CANCEL' and 'SAVE CHANGES' buttons. The bottom left corner of the sidebar shows a user profile for Sarah Chen (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

A-SW HUB

LOGOUT

CANCEL SAVE CHANGES

PROJECT CONFIGURATION

V2X 협업 플랫폼

GENERAL SETTINGS

PROJECT NAME

The display name for your project

V2X 협업 플랫폼

DESCRIPTION

A brief description of what this project does

V2X 기반 다수 장비 협업 제어 플랫폼

CATEGORY

Project category for organization

FARM ANALYTICS

VISIBILITY

Control who can see this project

PUBLIC - ANYONE CAN VIEW

LICENSE

Open source license (optional)

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)
- ...and 3 more features

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 project named 'V2X 협업 플랫폼'. The left sidebar includes links for Dashboard, Projects, Builds, Timeline, Approvals, and Settings, with 'PROJECTS' currently selected. The main content area displays 'REPOSITORY SETTINGS' with a 'REPOSITORY URL' field containing 'https://github.com/asv-hub/v2x-collaboration', a 'DEFAULT BRANCH' field set to 'main', and a 'GIT PROVIDER' section indicating 'GITHUB'. Under 'BRANCH PROTECTION', there are three checkboxes: 'REQUIRE PULL REQUEST BEFORE MERGING' (unchecked), 'REQUIRE STATUS CHECKS TO PASS' (unchecked), and 'REQUIRE SIGNED COMMITS' (unchecked). The bottom right corner of the page shows the user 'Sarah Chen' with the role 'ADMIN'.

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
- ...and 5 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

Configure how your project is built and tested

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
- ...and 7 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

The screenshot shows the 'PROJECT CONFIGURATION' section of the A-SW HUB interface. The left sidebar includes links for DASHBOARD, PROJECTS (selected), BUILD, TIMELINE, APPROVALS, and SETTINGS. The main area has tabs for GENERAL, REPOSITORY, BUILD, DEPLOYMENT (selected), NOTIFICATIONS, ADVANCED, and DANGER ZONE. Under DEPLOYMENT, there are sections for 'DEPLOYMENT SETTINGS' (Configure automatic deployments and environments), 'AUTO-DEPLOY BRANCH' (main), 'DEPLOYMENT ENVIRONMENT' (PRODUCTION selected), 'DEPLOYMENT COMMAND' (npm run deploy), and 'DEPLOYMENT OPTIONS' (AUTOMATIC DEPLOYMENTS ON PUSH checked, PREVIEW DEPLOYMENTS FOR PULL REQUESTS and ROLLBACK ON FAILURE unchecked). Navigation icons include back, forward, and search. Top right buttons are CANCEL and SAVE CHANGES. Bottom right shows user info: Sarah Chen, ADMIN.

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
- ...and 5 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, Builds, Timeline, 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 an email input field containing 'dev-team@example.com' and an '+ ADD RECIPIENT' button. The top right features 'LOGOUT', 'CANCEL', and 'SAVE CHANGES' buttons.

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
- ...and 5 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' screen 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 DEPENDABOT 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 (v2x-collaboration), CREATED (1/1/2027), and LAST UPDATED (10/24/2025). The top right of the screen features a 'LOGOUT' button, and at the bottom left, there is a user profile for '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
- ...and 3 more features

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 following details:

- Header:** A-SW HUB, LOGOUT
- Left Sidebar:** DASHBOARD, PROJECTS (selected), BUILD, TIMELINE, APPROVALS, SETTINGS. A-SW HUB is also listed under PROJECTS.
- Page Title:** PROJECT CONFIGURATION
- Content Area:** PROJECT CONFIGURATION
- DANGER ZONE Section:** Irreversible and destructive actions. Contains three buttons:
 - TRANSFER PROJECT OWNERSHIP:** Transfer this project to another user or organization. Button: TRANSFER
 - ARCHIVE PROJECT:** Make this project read-only. Can be undone later. Button: ARCHIVE
 - DELETE PROJECT:** Permanently delete this project. This action cannot be undone. Button: DELETE PROJECT
- Buttons at the top right:** CANCEL, SAVE CHANGES
- User Information:** Sarah Chen, 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
- ...and 4 more features

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' application interface for managing project pipelines. The left sidebar includes links for Dashboard, Projects, Builds, Timeline, Approvals, and Settings. The main area displays the 'PIPELINES' section with three pipelines listed: 'Main CI/CD Pipeline' (ACTIVE), 'Production Deployment', and 'Nightly Build'. The 'Main CI/CD Pipeline' is described as a 'Primary build, test, and deployment pipeline' with a 'LAST RUN: 2h ago'. The 'Production Deployment' pipeline is a 'Production-only deployment pipeline with manual approval' with a 'LAST RUN: 2d ago'. The 'Nightly Build' pipeline is an 'Automated nightly build for comprehensive testing' with a 'LAST RUN: 7d ago'. The central panel for the 'Main CI/CD Pipeline' shows its configuration tabs: Overview (selected), Configuration, History (3), and Settings. The Overview tab displays the pipeline's status (ACTIVE), triggers (PUSH, PULL_REQUEST, SCHEDULE), environment (DEVELOPMENT, EMAIL: NO, SLACK: NO), and stages (Checkout, Build, Test, Quality, Deploy). The Pipeline Stages section shows the sequential flow from Checkout to Deploy. The RECENT EXECUTIONS section lists three completed runs: a GitHub webhook push at 12:38 PM on Oct 24, a manual run by John Developer at 08:38 AM on Oct 24, and a scheduled run at 02:38 PM on Oct 23.

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 Project Pipelines Overview 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): 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: Stage 1 (Checkout) for Stage 2 (Build); Stage 2 (Build) for Stage 3 (Test); Stage 3 (Test) for Stage 4 (Quality); Stage 4 (Quality) for Stage 5 (Deploy).
 - Stage 1: Checkout
 - git-clone
 - Stage 2: Build
 - compile
 - package
 - Stage 3: Test
 - unit-tests
 - integration-tests
 - Stage 4: Quality
 - lint
 - sonar
 - security-scan
 - Stage 5: Deploy
 - deploy-staging
- 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 (a3f2c1b): feat: Add new authentication module. BY: GitHub Webhook - DURATION: 5m 0s. Oct 24, 2025, 12:38 PM.
- MANUAL (b2e4d7a): fix: Resolve database connection issue. BY: John Developer - DURATION: 3m 0s. Oct 24, 2025, 08:38 AM.
- SCHEDULE (c7a92f3e): chore: Update dependencies. BY: Scheduled Trainer - DURATION: 6m 0s. Oct 23, 2025, 02:38 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
- ...and 3 more features

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' interface with the 'PIPELINES' tab selected. On the left, a sidebar lists '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 detailed below:

- OVERVIEW**: Primary build, test, and deployment pipeline. Last run: 2h ago.
- CONFIGURATION**:
 - TRIGGERS**: Trigger 1 (PUSH, branches: main, develop), Trigger 2 (PULL_REQUEST, branches: main), Trigger 3 (SCHEDULE, schedule: 0 2 * * *)
 - STAGES**:
 - STAGE 1**: Checkout (1 JOBS: git-clone)
 - STAGE 2**: Build (2 JOBS: compile, package)
 - STAGE 3**: Test (2 JOBS: unit-tests, integration-tests)
 - STAGE 4**: Quality (3 JOBS)
- HISTORY (3)**: Shows the last three runs.
- SETTINGS**: Pipeline settings.

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
- ...and 4 more features

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 A-SW HUB interface for managing project pipelines. On the left is a sidebar with navigation links: DASHBOARD, PROJECTS, BUILDS, TIMELINE, APPROVALS, and SETTINGS. The main area is titled "PIPELINES" and shows "3 PIPELINES". The first pipeline listed is "Main CI/CD Pipeline" (ACTIVE), which is a "Primary build, test, and deployment pipeline" last run 2h ago. The second pipeline is "Production Deployment", a "Production-only deployment pipeline with manual approval" last run 2d ago. The third pipeline is "Nightly Build", an "Automated nightly build for comprehensive testing" last run 7d ago. Below these is a section titled "EXECUTION HISTORY" showing "3 TOTAL EXECUTIONS". It lists three entries:

- SUCCESS**: Triggered by PUSH, triggered by GitHub Webhook, duration 5m 0s, commit a3f2c1b, message feat: Add new authentication module. Last run on Oct 24, 2025, 12:38 PM.
- FAILED**: Triggered by MANUAL, triggered by John Developer, duration 3m 0s, commit b2e4d7a, message fix: Resolve database connection issue. Last run on Oct 24, 2025, 08:38 AM.
- SUCCESS**: Triggered by SCHEDULE, triggered by Scheduled Trigger, duration 6m 0s, commit c7a9f3e, message chore: Update dependencies. Last run on Oct 23, 2025, 02:38 PM.

At the bottom left is a user profile for Sarah Chen (ADMIN). At the top right are "LOGOUT" and "+ NEW PIPELINE" buttons.

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
- ...and 4 more features

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' dashboard with the 'PIPELINES' section selected. The main pipeline listed is 'Main CI/CD Pipeline' (ACTIVE), which is described as a 'Primary build, test, and deployment pipeline'. It was last run 2 hours ago. Below this, there are three other pipeline cards: 'Production Deployment' (Production-only deployment pipeline with manual approval, last run 2 days ago), 'Nightly Build' (Automated nightly build for comprehensive testing, last run 7 days ago), and 'V2X 험연 플랫폼' (V2X 험연 플랫폼).

GENERAL SETTINGS

- Pipeline Name:** Main CI/CD Pipeline
- Description:** Primary build, test, and deployment pipeline
- Status:** ACTIVE

METADATA

- Created:** Jul 26, 2025, 02:38 PM
- Updated:** Oct 23, 2025, 02:38 PM
- Pipeline ID:** pipeline-1
- Environment:** DEVELOPMENT

DANGER ZONE

- PAUSE PIPELINE:** Temporarily disable this pipeline (with a PAUSE button)
- DELETE PIPELINE:** Permanently delete this pipeline and all its history (with a DELETE button)

Sarah Chen
ADMIN

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
- ...and 6 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
...and 3 more components

New Project

The screenshot shows the 'CREATE NEW PROJECT' wizard on step 1, 'BASIC INFORMATION'. The left sidebar shows 'A-SW HUB' with 'PROJECTS' selected. The main area has a title 'CREATE NEW PROJECT' with a 'CANCEL' link. A step indicator at the top right shows '1' (active), '2', '3', and '4'. The 'BASIC INFORMATION' section includes fields for 'PROJECT NAME' (e.g., Smart Tractor Control), 'PROJECT SLUG' (e.g., smart-tractor-control), 'DESCRIPTION' (Describe your project's purpose and functionality...), 'CATEGORY' (CROP MANAGEMENT), 'VISIBILITY' (PUBLIC - Anyone can view), and 'LICENSE' (Apache 2.0). A user profile for 'Sarah Chen' (ADMIN) is at the bottom.

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)
- ...and 10 more features

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
...and 1 more component

Builds Dashboard

The screenshot shows the 'BUILD MONITOR' section of the 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, 9 SUCCESS TODAY, 0 FAILED TODAY, and 87% SUCCESS RATE. Below this is a 'ACTIVE PIPELINES' section with three entries:

- A-SW 통합 플랫폼 #1100 (main branch) - Status: 33% (progress bar), Actions: Checkout, Install Dependencies, Lint, Unit Tests, Integration Tests, Build, Security Scan, Deploy, Smoke Tests.
- 측위 알고리즘 #1102 (feature/new-feature branch) - Status: 33% (progress bar), Actions: Checkout, Install Dependencies, Lint, Unit Tests, Integration Tests, Build, Security Scan, Deploy, Smoke Tests.
- 경로 제어 시스템 #1101 (main branch) - Status: 33% (progress bar), Actions: Checkout, Install Dependencies, Lint, Unit Tests, Integration Tests, Build, Security Scan, Deploy, Smoke Tests.

Below this is a 'RECENT BUILDS' section titled 'LAST 10 COMPLETED' with a table:

BUILD	STATUS	BRANCH	DURATION	FINISHED	ACTIONS
Task Scheduler #1014	■ SUCCESS	main	1s	2:02:28 PM	VIEW
A-SW 통합 플랫폼 #1023	■ SUCCESS	develop	1s	2:07:29 PM	VIEW
학습 데이터베이스 #1038	■ SUCCESS	develop	1s	2:01:41 PM	VIEW
환경인지 모듈 #1029	■ SUCCESS	main	1s	1:07:38 PM	VIEW
측위 알고리즘 #1001	■ SUCCESS	main	1s	1:11:37 PM	VIEW
환경인지 모듈 #1048	▲ CANCELLED	main	1s	12:00:04 PM	VIEW
경로 제어 시스템 #1007	■ SUCCESS	main	1s	11:09:14 AM	VIEW
Git 힙업 시스템 #1034	■ SUCCESS	develop	1s	11:03:06 AM	VIEW
농작업 분석 모듈 #1042	■ SUCCESS	main	1s	11:08:15 AM	VIEW
통합제어기(ICU) #1045	■ SUCCESS	feature/new-feature	1s	10:05:16 AM	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
- ...and 11 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
...and 6 more components

Deployments

The screenshot shows the 'DEPLOYMENT MANAGEMENT' section of the A-SW HUB interface. It features a sidebar with navigation links like Dashboard, Projects, Builds, Timeline, Approvals, and Settings. The main area displays four environment cards:

- Production KR (Production)**: Status: HEALTHY. Current Version: v2.5.0. Last Deployment: 2024-03-15 10:30. Scheduled Deployment: 2024-03-16 02:00. URL: <https://prod-kr.aww-hub.com>. Region: ap-northeast-2. Resource Usage: CPU 75%, Memory 82%, Storage 45%. Instances: 8 RUNNING. Metrics: Uptime 99.99%, Response Time 245ms, Error Rate 0.02%, Throughput 15,000/m. Buttons: → DEPLOY, ○ MONITORING.
- Production US (Production)**: Status: HEALTHY. Current Version: v2.5.0. Last Deployment: 2024-03-15 10:30. Scheduled Deployment: 2024-03-16 02:00. URL: <https://prod-us.aww-hub.com>. Region: us-east-1. Resource Usage: CPU 68%, Memory 71%, Storage 38%. Instances: 6 RUNNING. Metrics: Uptime 99.95%, Response Time 180ms, Error Rate 0.05%, Throughput 12,000/m. Buttons: → DEPLOY, ○ MONITORING.
- Staging (Staging)**: Status: DEGRADED. Current Version: v2.6.0-beta. Last Deployment: 2024-03-15 14:00. URL: <https://staging.aww-hub.com>. Region: ap-northeast-2. Resource Usage: CPU 45%, Memory 52%, Storage 25%. Instances: 2 RUNNING. Metrics: Uptime 98.5%, Response Time 320ms, Error Rate 2.5%, Throughput 3,000/m. Buttons: → DEPLOY, ○ MONITORING.
- Development (Development)**: Status: MAINTENANCE. Current Version: v2.7.0-dev. Last Deployment: 2024-03-15 16:45. URL: <https://dev.aww-hub.com>. Region: ap-northeast-2. Resource Usage: CPU 20%, Memory 25%, Storage 15%. Instances: 1 RUNNING. Metrics: Uptime 95%, Response Time 450ms, Error Rate 5%, Throughput 500/m. Buttons: → DEPLOY, ○ MONITORING.

In the bottom left corner, there is a user profile for Sarah Chen (ADMIN).

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)

...and 13 more features

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)

...and 11 more components

2025 농업기계 디지털 전환형 개방형 A-SW 오픈소스 협력 개발 서비스 플랫폼 구축 사업

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, and 0 APPROVED TODAY, with a total of 3 REQUESTS. The main content area is divided into two columns: 'APPROVAL LIST' (left, scrollable) and 'APPROVAL DETAILS' (right, fixed width).
APPROVAL LIST:

- Production Deployment - Smart Tractor v2.1.0** (Urgent, 30m ago)
Deploy critical security patch to production environment.
PROJECT: smart-tractor-control ENV: PRODUCTION BY: John Doe
✓ Security ✓ Tests △ Coverage ✓ Performance ✓ Code Quality
[APPROVE] [REJECT]
- Staging Deployment - Irrigation Controller v1.5.0** (High, 2h ago)
New feature: Smart water management based on weather forecast.
PROJECT: irrigation-controller ENV: STAGING BY: Sarah Chen
✓ Security ✓ Tests ✓ Coverage ✓ Performance ✓ Code Quality
[APPROVE] [REJECT]
- Configuration Update - Database Connection Pool** (Medium, 4h ago)
Increase connection pool size for harvest season load.
PROJECT: harvest-optimizer ENV: PRODUCTION BY: Mike Kim
✓ Security ✖ Tests ✓ Coverage △ Performance ✓ Code Quality
[APPROVE] [REJECT]

APPROVAL DETAILS:

APPROVAL DETAILS

APPROVE [REJECT] HOLD

QUALITY GATE CHECKS

- ✓ Security 100/100
- ✓ Tests 342/342 tests passed
- △ Coverage 78/80 Below threshold
- ✓ Performance 95/90
- ✓ Code Quality 85/80

CHANGES

- MODIFIED src/navigation/auto_pilot.py
- ADDED src/security/patch.py
- MODIFIED requirements.txt

REQUEST INFORMATION

TYPE	PRIORITY
DEPLOYMENT	URGENT
ENVIRONMENT	
PRODUCTION	smart-tractor-control
REQUESTED BY	John Doe
BUILD ID	build123
REQUESTED AT	10/24/2025, 2:09:02 PM

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
- ...and 14 more features

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

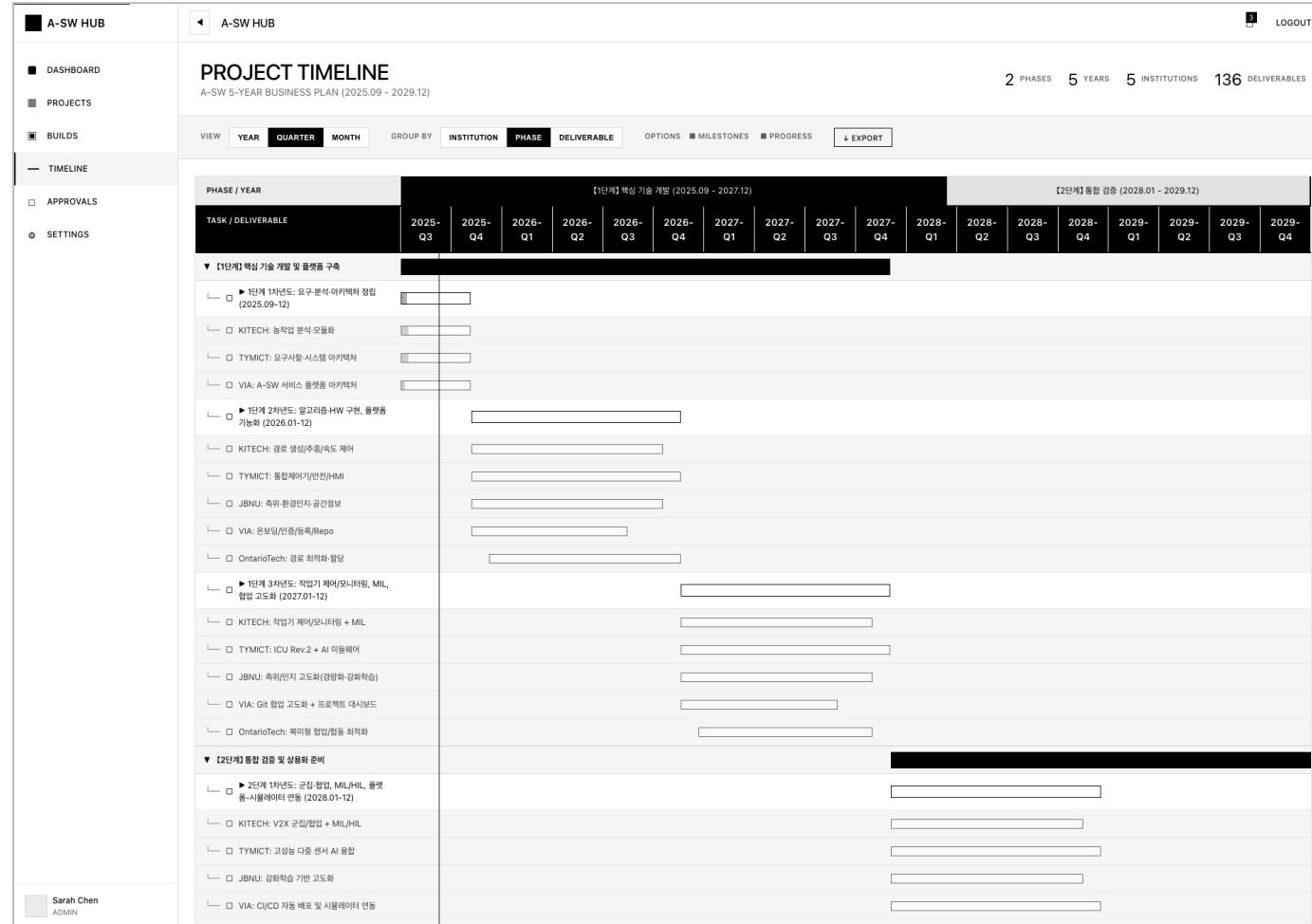
Change type labels (ADDED: foreground color, MODIFIED: muted, DELETED: strikethrough)

File paths in monospace font

Request information grid (2 columns) showing Type, Priority, Environment, Project, Requested By, Requested At, Build ID

...and 6 more components 2025 농업기계 디지털 전환형 개방형 A-SW 오픈소스 협력 개발 서비스 플랫폼 구축 사업

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, TYMICK, JBU, 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
- ...and 14 more features

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

...and 4 more components

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 quality trend over 30 days. Below, quality criteria are listed: Minimum Coverage (70%, PASS), Security Vulnerabilities (0 High, FAIL), Code Duplication (< 5%, PASS), and Complexity (< 20, WARN). The bottom section is labeled 'AI RECOMMENDATIONS'.

A-SW HUB

LOGOUT

QUALITY GATE

PROJECT SMART-TRACTOR-CONTROL

Last analyzed: 10/24/2025, 2:39:08 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)

Minimum Coverage: 70% ✓ PASS

Security Vulnerabilities: 0 High ✗ FAIL

Code Duplication: < 5% ✓ PASS

Complexity: < 20 △ WARN

Sarah Chen ADMIN

AI RECOMMENDATIONS

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' currently 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). A bio field contains the text 'Agricultural software platform administrator'. At the top right is a 'LOGOUT' button, and at the bottom right is a 'SAVE CHANGES' button with a circular progress indicator.

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