Full Requirements List: Mentorship-Focused Student Management System (SMS)

1. Functional Requirements

A. User Management

- **User Roles:**
    - Admin
    - Developer
    - Manager
    - Student Relationship Executive (SRE)
    - Mentor
    - Student
- **Registration & Authentication:**
    - User registration and login via email and password.
    - Role-based access control (RBAC).
    - Password reset functionality.
    - User profiles with editable information.

B. Student Management

- **Student Profiles:**

    - Store student personal information (name, email, etc.).
    - Track student enrollment in specific missions/batches.
    - Manage student status (active, banned, delisted).
- **Mission & Batch Assignment:**
    - Assign students to batches and missions.
    - A student can be part of multiple missions and batches.
    - Assign multiple mentors to each student based on missions.
- **Mentor Assignment:**
    - Assign mentors to students for each mission.
    - Mentors should be able to view only their assigned students.

C. Mission Management

- **Batch and Mission Management:**
    - Create and manage batches.
    - Create and manage missions within batches.
    - Track mission progress and student participation.
- **Mission Progress Tracking:**
    - Track student engagement with missions, including attendance and assignments.

D. Attendance & Assignment Management

- **Daily Attendance:**
    - Students submit daily attendance.
    - Students submit weekly meeting attendance.

- **Assignment Tracking:**
    - SRE will update Assignment sheet. By just copying email list from our main system to SMS system.
    - Track assignment submission deadlines.
    - Automatic point allocation for submissions and attendance.
- **Points Calculation:**
    - Assign points based on attendance and assignment submission.
    - Calculate total points and progress for students.

E. Academic Integrity & Access Control

- **Access Restrictions:**
    - Ban or delist students who violate academic integrity (e.g., cheating or plagiarism).
    - Restrict access to mentorship resources for flagged students.
- **Appeal Process:**
    - Provide an appeal process for students to contest bans or restrictions.

F. SRE Features

- **Low-Performer Monitoring:**
    - Generate reports listing low-performing students based on attendance and assignment submission.
- **Call List Generation:**
    - Generate call lists for low-performing students with suggested scripts/questions.
    - Log call outcomes (e.g., call picked, phone off, no answer, reschedule).

- **Call Logging:**
  - Record call outcomes and feedback during conversations.
  - Track follow-up actions and report progress.

G. Guidelines & Notifications

- **Notice System:**
  - Mentors, SREs, and Managers can send notices or guidelines to students.
  - Notices should be grouped by student batch or mission.
  - Track acknowledgment of notices by students.

H. Reporting & Analytics

- **Reports:**
  - Attendance reports per student, batch, or mission.
  - Assignment submission reports and grades.
  - Low-performing student reports.
  - Mentor performance and engagement reports.
- **Dashboards:**
  - Admin, Manager, and Mentor dashboards displaying relevant data (e.g., student progress, mission status).
  - SRE dashboard displaying low performers and call logs.

---

2. Non-Functional Requirements

A. Performance

- The system should support **2,000–3,000 students per batch**.
- System should be responsive and able to handle **high concurrency** during peak usage (e.g., assignment submission deadlines, attendance submission).

B. Scalability

- The system must be able to scale as the number of students and missions grows.
- Database should handle large volumes of data (e.g., student profiles, attendance records, assignments).
- The backend should handle real-time updates for attendance, assignments, and notifications.

C. Security

- Implement secure authentication and authorization mechanisms using **JWT** or **OAuth**.
- Protect user data and assignments from unauthorized access.
- Ensure secure handling of sensitive data (e.g., student contact details).
- Implement **CSRF, XSS**, and **SQL Injection** protection.

D. Data Integrity

- Use a **relational database** (PostgreSQL) to ensure data integrity and enforce relationships between entities (students, missions, mentors).
- Use **foreign keys** to link students to missions, mentors, and assignments.

E. Availability & Reliability

- The system should ensure high availability, especially during critical times (e.g., exam periods).
- Implement regular **database backups** and recovery mechanisms.

F. Usability

- The UI should be user-friendly, intuitive, and easy to navigate.
- Provide clear error messages and feedback to users during interactions.
- The system should be responsive and accessible across devices (desktop and mobile).

---

3. Tech Stack

A. Frontend

- **Next.js** (React Framework):
    - Server-Side Rendering (SSR) for faster load times.
    - Built-in API routes for simple backend logic.
    - Good scalability and SEO support.

B. Backend

- **Django REST Framework**:
    - Built on Python, Django is robust and supports AI integration easily.
    - Django's ORM simplifies interactions with the PostgreSQL database.
    - Offers built-in security features (e.g., authentication, CSRF protection).

C. Database

- **PostgreSQL** (via Supabase):
    - Relational database with strong querying capabilities.
    - Supports complex relationships and data integrity.

    - Supabase offers an easy-to-use backend platform with real-time features and authentication built-in.

D. File Storage

- **Supabase Storage**:
    - Handles media and document uploads (e.g., student assignments).
    - Integrates seamlessly with PostgreSQL and Supabase’s authentication system.

E. AI Integration (Future)

- **Python Libraries**:
    - TensorFlow, scikit-learn, and PyTorch for machine learning.
    - Hugging Face API for pre-trained NLP models.
- **AI Features**:
    - Personalized learning recommendations.
    - Automated grading system for assignments.
    - Predictive analytics for student performance.

F. Hosting

- **Frontend**: Vercel (for Next.js deployment).
- **Backend**: DigitalOcean or Heroku (for Django deployment).
- **Database**: Supabase (for PostgreSQL database hosting).

---

4. Milestones and Phases

Phase 1: Setup and Core Infrastructure

- Set up the Django project with PostgreSQL and Supabase.
- Create models and migrations for core entities (students, batches, missions).
- Set up the Next.js frontend with authentication and role-based access.

Phase 2: Core Functionality Development

- Implement CRUD for students, batches, missions.
- Implement progress tracking (attendance, assignments, mentor meetings).
- Build the SRE and mentor dashboards for managing students and reports.

Phase 3: Academic Integrity and AI Integration

- Implement plagiarism detection and ban/delist features.
- Integrate AI-based recommendation and analysis (future phase).

Phase 4: Reporting, Analytics, and Notifications

- Develop reporting features (attendance, assignment reports).
- Set up the notice and guidelines system.

Phase 5: Testing, Deployment, and Documentation

- Perform unit and integration testing.
- Deploy to production on DigitalOcean or Heroku.
- Write user documentation and admin guides.

Phase 6: Maintenance and Updates

- Regular updates, bug fixes, and performance optimizations.

- Implement new features like AI-driven chatbots or automated grading.

---

5. Post-Launch & Maintenance

- **Monitoring**: Set up monitoring tools (e.g., Sentry, Prometheus).
- **Security Updates**: Regularly patch vulnerabilities and update dependencies.
- **User Feedback**: Collect and implement user feedback to improve the system.
- **Backups**: Ensure regular backups of data and system configurations.

This full plan gives you a comprehensive roadmap for building your SMS. Each step aligns with your goals of scaling, security, and future AI integration. Let me know if you'd like to dive deeper into any specific area or start with a particular phase!

📋 Frontend Page Development Plan

Priority	Page	Description
1	**Login Page**	Authentication interface for all users.
2	**Home Page**	Landing page providing an overview and navigation to role-specific dashboards.
3	**Admin Dashboard**	Management interface for administrators to oversee system operations.

4	**Developer Dashboard**	Interface for developers to monitor system performance, logs, and handle technical configurations.
5	**Manager Dashboard**	Tools for managers to track team progress, manage resources, and oversee operations.
6	**SRE Dashboard**	Monitoring and alerting interface for Site Reliability Engineers to ensure system stability.
7	**Mentor Dashboard**	Interface for mentors to manage student interactions, track progress, and provide feedback.
8	**Student Dashboard**	Personalized interface for students to access learning materials, assignments, and progress reports.
9	**Settings Page**	Allows users to update personal information, change passwords, and configure preferences.
10	**Error Page**	Custom error handling page to manage unexpected issues gracefully.

🧩 Folder Structure in Next.js 15.3.3

With the introduction of the App Router in Next.js 15.3.3, organizing your project structure is crucial for maintainability and scalability. Here's a suggested folder structure:

app/

├── layout.tsx   # Root layout component

├── page.tsx   # Home page component

```
├── login/
│   └── page.tsx          # Login page component
├── dashboard/
│   ├── layout.tsx        # Dashboard layout
│   ├── admin/
│   │   └── page.tsx      # Admin dashboard page
│   ├── developer/
│   │   └── page.tsx      # Developer dashboard page
│   ├── manager/
│   │   └── page.tsx      # Manager dashboard page
│   ├── sre/
│   │   └── page.tsx      # SRE dashboard page
│   ├── mentor/
│   │   └── page.tsx      # Mentor dashboard page
│   └── student/
│       └── page.tsx      # Student dashboard page
├── settings/
│   └── page.tsx          # Settings page
└── error/
    └── page.tsx          # Error handling page
```

This structure leverages the App Router's capabilities, ensuring a clean and scalable organization of components and pages.

🔐 Implementing Role-Based Access Control (RBAC)

To manage the diverse user roles effectively, implementing Role-Based Access Control (RBAC) is essential. Here's how you can approach it:

- **Define Roles and Permissions**: Clearly outline what each role can access and perform within the application.
- **Assign Roles to Users**: Ensure that each user is assigned a role that aligns with their responsibilities.
- **Enforce Access Control**: Utilize middleware or higher-order components to restrict access to pages and components based on the user's role.

This approach ensures that users have access only to the resources and functionalities pertinent to their roles, enhancing security and user experience.

Here’s a prioritized breakdown of **Student Dashboard pages**, categorized as **Mandatory (V1)** or **Future (V2+)**, based on common standards and educational best practices:

🥇 V1: Must-Have Pages (Core MVP)

These are essential pages to launch a functional student dashboard:

1. **Dashboard Home** – Overview of mission progress, points, and upcoming deadlines.
2. **Daily Attendance** – Interface to submit attendance and view history.
3. **Assignments Tracker** – List assignments with submission capability and due dates.
4. **My Missions/Batches** – Access current and past mission details and mentoring info.
5. **Mentor Contact / Call Scheduler** – View your mentor’s profile, contact, and session schedule.
6. **Profile & Settings** – Manage personal data, password, and notification preferences.

These core features align with leading educational dashboard essentials like tracking attendance, progress, and communication—key for structured learning and engagement (illuminateed.com).

🥈 V2: Important but Secondary

These features add depth and student empowerment:

- **Resource Center** – Central repository for mission documents, notices, and guidelines.
- **Progress & Analytics** – Visual metrics on attendance, assignment trends, and earned points (tenthousandcoffees.com).
- **Calendar & Schedule** – Shows deadlines, mentor sessions, and attendance logs.

🥉 V3: Nice-to-Have for Future Growth

These can significantly boost engagement and community:

- **Portfolio / Project Showcase** – Students upload work, include mentor feedback.
- **Discussion Forum / Community** – Peer interaction space (innovativementoring.net, watermarkinsights.com).
- **Notifications Center** – Alerts for new assignments, mentor comments, deadlines.
- **Academic Appeals** – Track integrity flags or appeal submissions.
- **Support / Help Center** – Ask SRE for technical or academic assistance.
- **Certificates / Achievements** – Display earned badges or completion status.

📊 Priority Summary Table

Priority	Page	Core Function

V1	Dashboard Home	Overview of key metrics
V1	Daily Attendance	Attendance submission & tracking
V1	Assignments Tracker	View & submit assignments
V1	My Missions / Batches	Mission/batch access and mentor info
V1	Mentor Contact / Scheduler	Mentor communication and scheduling
V1	Profile & Settings	User profile and preferences
V2	Resource Center	Access mission documents and notices
V2	Progress & Analytics	Visual tracking of student performance
V2	Calendar & Schedule	Unified view of schedule elements
V3	Portfolio / Project Showcase	Upload and review project work
V3	Discussion Forum / Community	Peer-to-peer engagement
V3	Notifications Center	Centralized alerts and updates
V3	Academic Appeals	Manage integrity and appeal notifications
V3	Support / Help Center	Submit support requests
V3	Certificates / Achievements	Display earned mission badges or certificates

Thanks for the detailed clarification! Here's a **version-wise breakdown of Student tasks/features**, based on your updated vision — structured to reflect both core functionalities and future enhancements.

🥇 Version 1 (MVP – Must Have)

🚀 Goal: Ensure student engagement, attendance, communication, and profile management.

Feature	Description

📌 **Dashboard Overview**	Personalized dashboard showing points, missions, attendance summary, etc.
📝 **Daily Attendance Submission**	Submit daily attendance (type varies by mission status).
👥 **Mission Participation View**	See enrolled missions and progress details.
📅 **Meeting Attendance Submission**	Submit weekly meeting attendance.
📨 **Notices Viewer**	View notices from mentors/SREs/Managers, sorted by batch/mission.
💬 **Private Ticket to Mentor/SRE**	Raise a private query/request; separate threads for mentors & SREs.
👤 **Profile & Settings**	Manage personal info, password, notification preferences.
📮 **Feature Request Submission**	Suggest new features for the system (goes to admin/manager review).

🥈 Version 2 (Important Enhancements)

> 📈 Goal: Improve self-tracking, peer motivation, and academic accountability.

Feature	Description
📊 **Progress & Analytics**	Visual representation of assignment & attendance stats.
📖 **Mission Resource Center**	Access to uploaded mission content (PDFs, videos, etc.).
🔔 **React to Notices**	Students can acknowledge or react to notices (✅ / ❗ / 👍 etc.).
📁 **Assignment Submission Overview**	View submitted/pending assignments with due dates.
🗣️ **Appeal / Flag Resolution System**	Appeal against academic flags (late submissions, bans, etc.).

🥉 Version 3 (Nice-to-Have / Future-Ready)

> 🧠 Goal: Enable deeper interaction, community learning, and recognitions.

Feature	Description
🏆 **Certificates & Achievements**	Download badges or certificates for mission completion.
👥 **Discussion/Peer Forum (Optional)**	Interact with batchmates (carefully moderated; optional feature).
🗃️ **Project Portfolio / Showcase**	Showcase assignments, feedback, and progress (only viewable by mentors/SRE).
🧠 **Personalized AI Assistant**	Ask questions, review suggestions, etc. (future AI integration).

📊 Summary Table

Priority	Page/Feature	Core Functionality
V1	Dashboard Overview	Quick stats & navigation
V1	Daily Attendance	Submit based on mission status
V1	Meeting Attendance	Weekly submissions
V1	Notices Viewer	View mentor/SRE/manager updates
V1	Ticket System	Ask private questions
V1	Profile & Settings	Manage account info
V1	Feature Request	Suggest improvements
V2	Progress Analytics	Graph-based performance view
V2	Mission Resource Center	Access content (if shared by mentor)
V2	React to Notices	Engagement via reactions

V2	Assignment Tracker	Submission status and timelines
V2	Appeal Management	Resolve integrity issues or submission problems
V3	Certificate Display	View/download earned recognitions
V3	Peer Forum (Optional)	Collaborate or discuss among students
V3	Project Portfolio	Upload & store project-based work
V3	AI Assistant (Future)	Guidance/chatbot for support

Let me know if you'd like to prioritize implementation **based on complexity vs. impact**, or start designing a specific module (like Attendance, Ticket System, or Feature Requests).

Here’s the prioritized breakdown of **Mentor Dashboard pages**, categorized into **Mandatory (V1)**, **Important (V2)**, and **Future enhancements (V3)**, inspired by common mentor portal features (disruptionfactory.org):

🥇 V1: Must-Have (Core MVP)

These essential pages ensure mentors can support students effectively:

1. **Mentor Dashboard Home**
   – Overview of mentees, upcoming sessions, and quick metrics.

2. **My Students**
   – List of assigned mentees with progress status and contact links.

3. **Session Scheduler**
   – Interface to book or manage one-on-one sessions (calendar view + list).

4. **Messaging / Communication**
   – Built-in chat or email links to message mentees and follow up.

5. **Profile & Settings**
   – Personal info, availability preferences, notification settings.

🥈 V2: Important Features

Enhance mentorship quality and tracking:

- **Resources Library**
  – Access mentor guides, learning documents, and best-practices materials (disruptionfactory.org, mentorink.com).

- **Progress Tracker / Analytics**
  – View mentee milestones, engagement metrics, and mentoring KPIs (chronus.com, support.higherlogic.com).

- **Reporting & Export**
  – Export session logs, mentor hours, or notes for reviews.

- **Calendar Integration**
  – Sync mentoring sessions with Google/Outlook calendar.

🥉 V3: Future / Nice-to-Have

Useful as mentoring matures within the platform:

- **Group Mentoring / Workshops**
  – Host group sessions or student cohorts (socialroots.ai).

- **Feedback & Evaluation**
  – Forms for giving structured feedback on mentee progress (mentorink.com).

- **Achievements Showcase**
  – Highlight which mentees completed milestones or certificates.

- **Community & Peer Mentor Forums**
  – Connect with peers, share resources and mentoring techniques.

📊 Priority Summary Table

Priority	Page	Core Functionality and Purpose
V1	Mentor Dashboard Home	Overview metrics, upcoming sessions
V1	My Students	Mentee list with progress and contacts
V1	Session Scheduler	Book/manage mentoring sessions
V1	Messaging / Communication	Built-in chat or email to mentees
V1	Profile & Settings	Manage personal info and availability
V2	Resources Library	Mentor materials, best practices
V2	Progress Tracker / Analytics	Metrics and mentee performance dashboards
V2	Reporting & Export	Export logs and session details
V2	Calendar Integration	Sync sessions with external calendars
V3	Group Mentoring/Workshops	Host and manage cohort sessions
V3	Feedback & Evaluation	Structured feedback forms for mentees
V3	Achievements Showcase	Highlight mentee certificates and completed milestones
V3	Community / Forums	Mentor peer connection and resource sharing

---

This prioritized roadmap ensures your **Mentor Dashboard** starts with powerful, essential tools (V1), builds deeper functionality for mentoring success (V2), and later fosters community and recognition (V3). Let me know which page you'd like to develop next!

✅ Version 1 (MVP - Must Have)

Goal: Empower mentors to monitor and guide assigned students effectively.

Feature	Description
🧭 Mission-Based Student Assignment	Mentors only see students under missions they're assigned to.

📜 Mission Notices	Post important announcements/notices visible to assigned students.
📆 View Attendance	View daily/weekly attendance of students in their assigned missions.
🗂️ Mentor Dashboard	Overview of assigned students, sessions, attendance summary.
🧑‍🏫 Student List with Contact & Progress	Access to list of students with key progress indicators.

🧪 Version 2 (Important Enhancements)

Goal: Streamline communication and basic student queries.

Feature	Description
🎫 Ticket-Based Student Queries	Students can open private tickets (questions/appeals), mentors can reply and close.
🧮 Attendance Insights	Graph-based view of attendance trends (optional charts).
📝 Progress Notes	Mentor can add notes per student (not visible to students).
🧑‍🏫 Schedule View	See upcoming meeting sessions with students (pulled from scheduler).

💡 Version 3 (Optional/Nice-to-Have)

Goal: Increase mentor engagement, mentorship impact, and insights.

Feature	Description
📊 Mentor Performance Analytics	Dashboard showing student outcomes, session counts, ticket response times.
📌 Mission Resource Upload	Mentor can upload mission-specific PDFs, slides, videos for students.
🏅 Student Recognition	Ability to give small badges or shout-outs to students (manual).

❌ Avoid (for now)

Feature	Reason
💬 Real-Time Chat	Hard to moderate and scale. Ticket system is better for now.
🛠️ Complex Scheduling System	Use simple booking/calendar-style interface initially.

Would you like help **structuring the mentor dashboard** now that the feature set is clear?

Below is a comprehensive roadmap for the **Developer Dashboard**, covering **Version 1 through Version 4+**. Each stage outlines key pages and features aligned with industry best practices (guide.pidash.cornell.edu).

🥇 V1: Core MVP (Must-Have)

Essential pages to get the Developer Portal up and running:

- **Developer Dashboard Home**
  Overview of API usage, error rates, and deployments.

- **API Keys Management**
  Create, view, rename, and revoke API keys—fundamental for secure access (neosalpha.com).

- **Webhooks / Event Hooks**
  Manage webhook endpoints and event configurations (docs.webmethods.io).

- **Logs & Metrics**
  View recent API logs, request traces, and monitor performance (moesif.com).

- **Profile & Settings**
  Manage account info, security settings (MFA), and notification preferences .

🥈 V2: Developer Experience & Utility Boosters

Features that deepen usability and self-service:

- **API Documentation + Interactive Console**
  Inline documentation and API testing environments (moesif.com).
- **Developer Onboarding Flows**
  Sign-up approval, sandbox access, and guided first steps (neosalpha.com).
- **Usage Dashboards & Analytics**
  Track API usage trends, unique users, error trends .
- **Billing & Quotas Overview**
  Usage limits, quota allocation, and billing summaries.
- **Change Log / Release Notes**
  View recent API updates and version history.

🥉 V3: Integrations & Advanced Control

Expanding into ecosystem and advanced features:

- **Calendar & Schedule (Webhooks)**
  Timeline view for scheduled webhook events.
- **API Versioning & Lifecycle Management**
  Manage sandbox/stable/live versions, deprecation notices.
- **SDK Downloads**
  Provide client libraries for Node, Python, Java, etc. (microsoft.com, github.com, zitadel.com, docs.webmethods.io, learn.microsoft.com).
- **Support / Feedback Portal**
  Submit issues, request help, or ask for feature integrations.

🚀 V4+: Enterprise and Growth-Grade Features

For mature, scalable, and enterprise-ready portals:

- **Team & Role Management**
  Multi-user roles, permissions, team membership.

- **Billing & Invoicing Tools**
  Viewable invoices, exports, and API-based billing management.

- **Self-Service Marketplace**
  App/plugin listings, extensions, or partner integrations.

- **Automation & Workflow Tools**
  Webhook testing sandbox, CI/CD integration via REST.

- **Marketplace / Partner Integrations**
  Enable third-party extensions to interact with your platform.

📊 Summary Table

Version	Page/Feature Category	Key Functionality Overview
V1	- Dashboard Home- API Keys- Webhooks- Logs & Metrics- Profile & Settings	Essential access, security, visibility
V2	- API Docs & Console- Onboarding- Analytics dashboards- Billing / Quotas- Release Notes	Better self-service & insights
V3	- Webhook Calendar- API Versioning- SDK Downloads- Support Portal	Integrations & extended control
V4+	- Team roles- Invoicing- Marketplace & automation	Enterprise-grade ecosystem

---

This roadmap provides a clear path from a **functional developer portal** (V1) to a **powerful, enterprise-grade platform** (V4+). Let me know which level you'd like to dive into next—I can help wire up UI scaffolds, API routes, or database models!

🧑‍💻 Developer Role — Version-Wise Feature List

🥇 V1 – Core Essentials

These are required to support and debug the platform during MVP and early production.

Feature	Description
🔍 **System Dashboard**	Overview of system health, version, and critical notices.
📜 **View Logs (Basic)**	View application logs, error traces, and warnings.
🧪 **Test API Endpoints**	Built-in UI to test selected endpoints (auth, student fetch, etc.)
📈 **Monitor Performance**	Basic metrics (load times, API latency, error rates).
🧾 **Migration Viewer**	Review and verify database migrations and current schema.

🥈 V2 – Operational & Debugging Tools

More advanced tools for day-to-day backend ops and data management.

Feature	Description
🛠️ **Job Queue Monitor**	View running background jobs (emails, assignment syncs, etc.).
🔁 **Sync Tools**	Manually trigger syncs from external systems (assignment sheets, etc.).
🗄️ **Database Inspector**	Search and filter key records (users, sessions, logs).
🔒 **Permission Audit**	View and trace role permissions for users (RBAC checker).

🥉 V3 – DevOps & Automation Helpers

System growth requires automation, testing, and resilience tooling.

Feature	Description
🚨 **Alert Management**	Set alerts on failure conditions (downtime, API error spikes).
⛓ **API Rate Limiting Panel**	Simulate or inspect limits, throttling behavior.
🔄 **CI/CD Integration Dashboard**	View deploy status, last commit metadata, and environment info.
🧪 **Test Suite Runner**	Trigger automated test cases directly from dashboard (integration/unit tests).

🥇 Optional Developer Dashboard Page Paths

```
📁 src/app/(dashboard)/dashboard/developer/
├── page.tsx          # Main Developer Dashboard
├── logs/
│   └── page.tsx      # Log viewer
├── monitor/
│   └── page.tsx      # Performance & health monitor
├── jobs/
│   └── page.tsx      # Background job monitor
├── apitest/
│   └── page.tsx      # Endpoint test UI
├── inspector/
│   └── page.tsx      # DB data view
├── tools/
│   └── page.tsx      # Utilities (syncs, migrations)
```

Would you like me to generate folder commands and starter code for these pages too?

Here's an enhanced and fully structured roadmap of **Admin Dashboard pages**, covering **Versions 1–4+**, complete with clear justifications and alignment to best practices (aspirity.com):

🎯 V1: Essential Core Pages

Must-have pages that form the backbone of any admin panel:

1. **Admin Dashboard Home**
   – Central overview featuring key metrics like user counts, recent activity, alerts, and performance trends (bootstrapdash.com).

2. **User Management**
   – Create, edit, assign roles to users; includes search, filtering, and permissions control (bootstrapdash.com).

3. **Content / Data Management**
   – CRUD interfaces for managing core entities (courses, notices, mission records), possibly with version control (aspirity.com).

4. **System Logs & Monitoring**
   – Access audit logs, error tracking, and system activity to ensure security and traceability (incora.software).

5. **Settings & Permissions**
   – Site-wide settings (appearance, email templates, integrations), plus role-based access and security configs (bootstrapdash.com).

🥈 V2: Advanced Management & Visibility

Add-ons that enhance transparency, security, and oversight:

- **Analytics & Reporting**
  – Visual dashboards for trends (enrollment, assignments submission rates, platform usage) .

- **Notifications & Alerts**
  – Configurable alerts for important events (e.g. banned students, failed missions, system errors) .

- **Integrations & Webhooks**
  – Manage connectors to other systems (e.g. learning tools, email services) (aspirity.com).

🥉 V3: Moderation & Community Features

Tools for governance and community building:

- **Academic Integrity / Appeals**
  – Monitor flagged students, review academic violations, and process appeals.

- **Support & Feedback Center**
  – Admin-facing ticketing interface or forum for internal/external support (community.fandom.com, multipurposethemes.com).

- **Audit Trail Viewer**
  – Detailed logs of admin actions with filters and rollback support (keycloak.org, docs.redhat.com).

🚀 V4+: Enterprise-Grade Functionality

Scaling up to a powerful, customizable, enterprise-ready admin panel:

- **Role / Team Management**
  – Multi-tenant support, custom roles, hierarchical permissions, and team assignment.

- **Configurable Workspaces**
  – Drag-and-drop dashboards tailored per role, modular UI setups (five.co).

- **Self-Service Tools**
  – Admins can configure UI options themselves (themes, layout, language settings) (help.ivanti.com).

- **Audit & Compliance Reporting**
  – Built-in reporting for GDPR, FERPA, or other compliance requirements.

📊 Summary Table

Version	Page/Feature Category	Functionality Overview

V1	Dashboard Home, User Mgmt, Content Data, Logs, Settings	Essential administrative control & security
V2	Analytics, Alerts, Integrations	Insight, monitoring, external system linkage
V3	Academic Integrity, Support Center, Audit Tools	Governance, appeals, admin transparency
V4+	Teams, Workspaces, Self-Service Admin Tools, Compliance	Enterprise scaling, security, customization

---

✅ Why This Structure Works

- **Based on admin panel best practices**: easy navigation, CRUD operations, analytics, and performance (five.co, help.ivanti.com, aspirity.com).
- **Scalable and modular**: start with essential features, and grow per organizational needs .
- **Security-first**: includes audit logs, role-based access, and system settings .

🥇 V1 — Core Admin Functionality (Launch-Ready)

1. **User Management & RBAC**
   - Create/update/delete users
   - Assign roles, handle bans/delisting
2. **Mission & Batch Management**
   - CRUD operations for missions and batches
   - Assign students to batches and mentors per mission
3. **Notices & Communication**
   - Create batch- or mission-specific notices

- Track student acknowledgments

4. **Basic Reporting & Analytics**
   - Simple exportable summaries for attendance, assignments, etc.

🥈 V2 — Enhanced Visibility & Governance (Next Release)

5. **System Logging & Audit Trail (Lite Logs)**
   - Track login history, permission changes, and critical actions
   - Provide transparency and basic auditability aligned with compliance needs (zengrc.com, strongdm.com, datadoghq.com)
6. **Settings & Access Control Dashboard**
   - Admin UI for global settings (themes, email templates, feature flags)
   - RBAC interface to view and manage permissions

These features solidify administrative oversight and support compliance practices, with audit logs providing valuable accountability (zengrc.com).

🥉 V3 — Optional Enhancements

7. **Academic Integrity & Appeals**
   - Manage flagged students and events, support appeals
8. **External Integrations**
   - Settings interfaces for plagiarism tools, LMS connectivity
9. **Bulk Operations UI**

- CSV imports for users, bulk assignments, and bulk updates

📊 Updated Roadmap Table

Version	Feature Area	Description
V1	Users, Batches, Communication	Core admin workflows and data control
V2	Lite Logs, Settings & Control	Auditability, governance, and system-level configurations
V3	Appeals, Integrations, Bulk Ops	Advanced processes and scaling management

---

🔑 Why Lite Logs Shifted to V2

- **Auditability is critical**: It deters misuse and supports transparency without needing complete audit trails v1.
- **Compliance readiness**: Helps prepare for regulatory needs like SOC2 through basic monitoring and accountability (coreview.com, youtube.com, reddit.com, stackoverflow.com, pangea.cloud, auditboard.com, zengrc.com).

Certainly! Here's the enhanced SRE (Student Relationship Executive) dashboard plan, incorporating the ability to manage student batches and missions, as well as prioritized features:

🥇 V1 – Core MVP (Essential Features)

1. **SRE Dashboard Home**

  - Real-time snapshot: total students, flagged/at-risk students, pending calls, active batches, and scheduled missions.

2. **Low-Performer / At-Risk List**

   - Prioritized student list based on attendance and performance threshold with risk labels and quick action buttons.

3. **Call Scheduler & Log**

   - Generate call lists (e.g., “At-Risk Batch A”), make outreach calls, log results, and set follow-up reminders.

4. **Communication Center**

   - Inbox-like interface for structured calls/emails, pre-defined templates, and call outcome tracking (tryvirtually.com, velvetjobs.com).

5. **Student Assignment & Batch Management**

   - Tool to assign students to missions and batches; bulk updates (e.g. CSV upload) for reorganizing assignments (velvetjobs.com).

6. **Profile & Settings**

   - Manage personal information, call scripts, notification preferences, and batch assignment templates.

🥈 V2 – Enhanced Operations

- **Automated Alerts & Flags**

  - System-generated alerts for attendance decline, missed assignments, or batch imbalance.

- **Performance Reports**

  - Downloadable analytics for outreach effectiveness: call logs vs. student engagement trends.

- **Resource Broadcast**
  - Bulk send notices or resources to students by batch or mission, with acknowledgment tracking.
- **Follow-Up Center**
  - View and manage upcoming and overdue follow-ups by batch or individual student.

🚀 V3+ – Community & Engagement Expansion

- **Feedback & Surveys**
  - Aftercalls or batch activities, collect student feedback; sentiment tracking over time.
- **Intervention Workflow**
  - Predefined steps for escalation: e.g., counselor referral → mentor alert → remedial action; with audit logs.
- **Support Ticketing**
  - Log system, academic, or personal issues; track resolution status with student and staff assignments.
- **Motivational Badges**
  - Award and display badges (e.g., Consistent Attendance) and mission milestones in the student-facing UI.

📊 Summary Table

Version	Features

V1	Dashboard Home, Risk List, Call Scheduler & Log, Communication Center, Batch/Mission Assignment, Profile & Settings
V2	Automated Alerts, Engagement Reports, Resource Broadcast, Follow-Up Overview
V3+	Surveys & Sentiment, Intervention Workflows, Support Tickets, Motivational Badges

---

🧩 Why Student Assignment is Vital

- SREs frequently need to **organize or reassign students** to different batches/missions for tracking, personalized outreach, and academic integrity (steamsonline.com, zoho.com, tryvirtually.com, campustechnology.com).
- Supporting **batch-level actions** prevents repetitive manual work and standardizes assignment processes.

Let me know which part you'd like to start building—whether scaffolding the student assignment page or setting up the call logging UI!

🥇 V1 — Core Operational Features (Must-Have)

These tasks define the essential daily responsibilities of an SRE:

1. **Student Assignment Management**
    - Assign students to batches and missions

   - Allocate mentors to students per mission

2. **Assignment Email List Generation**
   - Upload day-wise assignment email lists
   - System updates attendance/submission status dynamically

3. **Call List Automation**
   - System generates prioritized call lists based on performance flags
   - SRE views suggested students for contact

4. **Interactive Call Logging Form**
   - A dynamic table displaying student name, phone, batch/mission
   - SRE records outcome (picked/no answer), responses, and remarks

5. **Batch/Mission Access Management**
   - Update students' access status within missions after appeals or infractions

6. **Notice & Announcement Tools**
   - Create motivational or informational notices for specific batches/missions
   - Track acknowledgments after dissemination

🥈 V2 — Efficiency & Insight Enhancements (Should-Have)

Ideal additions when core operations are stable:

7. **Automated Alert Generation**
   - System flags students behind by criteria (e.g., no submission for X days)

    - Auto-creates call or follow-up lists

8. **SRE Dashboard Analytics**
    - Visual reports: call outcomes, response rates, outreach effectiveness
9. **Scheduled Follow-Up Workflow**
    - SRE can mark follow-up dates, reminders, and track pending calls
10. **Batch Notices Acknowledgement Reports**

- View response status to notices per batch/mission

🥉 V3+ — Engagement, Community & Optimization (Could-Have)

For deeper student engagement and scalable interaction:

11. **Feedback Surveys Post-Calls or Missions**
    - Embed survey forms in the call flow for real-time sentiment data
12. **Intervention Workflows & Escalations**
    - Predefined escalation plans for recurring low performance cases
13. **Support Ticketing System**
    - Log student issues or grievances, assign, and track resolution
14. **Automated Email Workflow**
    - Use templates for reminders or motivational nudges to students
15. **Performance Gamification**
    - Badges or recognition tied to improved attendance/assignment completion

---

📊 Consolidated Priority Overview

Version	Feature Set	Description
V1	Core Operations	Assignment, email lists, calls, access
V2	Data-Driven Enhancements	Alerts, analytics, follow-ups, reports
V3+	Community & Automation	Surveys, escalations, ticketing, badges

---

🥇 Version 1 (MVP – Must Have)

🎯 Goal: Enable managers to build structure (batches + missions) and manage overall mentorship ecosystem.

Feature	Description
📋 **Dashboard Overview**	Summary of active batches, mission stats, mentor/student counts, etc.
🏗️ **Batch Management**	Create/edit batches (with titles, codes, status).
🧭 **Mission Management**	Create missions, assign them to batches.
👨‍🏫 **Mentor Assignment**	Assign mentors to specific missions.
🔍 **Student View (Read-Only)**	View student performance and progress under each mission.
🗂️ **Mission Status Tracking**	Monitor mission progress and participation rate.

🥈 Version 2 (Moderate Impact – Strategic Layer)

🔧 Goal: Optimize system usage, mentor allocation, and batch scaling.

Feature	Description
📊 **Mission Performance Analytics**	Charts for attendance, assignment stats across missions/batches.
🤝 **Mentor Load Analysis**	View number of mentees per mentor, workload balance.
🔁 **Mentor Reassignment Tool**	Shift mentor-student pairs dynamically per mission needs.
📑 **Batch Archiving**	Archive inactive or completed batches.
💼 **Mentorship Effectiveness Report**	Metrics to measure engagement and mentor effectiveness.

🥉 Version 3 (Advanced Governance)

> 🔒 Goal: Expand control, communication, and strategic decisions.

Feature	Description
📣 **Notice Broadcasting**	Send global/system-wide notices to all batches or roles.
📁 **Resource Sharing Control**	Upload or distribute learning resources across batches.
🧾 **Mission-Level Guidelines Editor**	Define do's & don'ts or expectations per mission.
📑 **Cross-Batch Comparison Report**	Analyze performance across multiple batches.
👥 **Mentor Feedback Collection**	Collect student feedback on mentor behavior/quality.

🧠 Future Versions (V4+)

> 🧬 Goal: Make the Manager role predictive, proactive, and decision-driven.

Feature	Description

🧠 **AI Suggestions for Mentor Pairing**	Recommend mentor-student matching using past data.
📆 **Automated Mentor Scheduling**	Auto-generate mentor availability slots for SREs/students.
🛠️ **Custom Dashboard Builder**	Managers can configure their own dashboard widgets.
📬 **Actionable Alerts**	Get alerts for batch inactivity, overdue assignments, or mentor gaps.

📊 Summary Table

Priority	Page/Feature	Core Functionality
V1	Dashboard Overview	Snapshot of system activity
V1	Batch Management	Create/edit student groups
V1	Mission Management	Define scopes of learning and assign mentors
V1	Mentor Assignment	Match mentors to missions
V1	Student Read-Only View	Track students in batches
V2	Performance Reports	Mission & mentor stats
V2	Mentor Load Tools	Identify overburdened mentors
V2	Batch Archiving	Clean up inactive data
V3	Notice System	Send announcements
V3	Resource Control	Upload materials per mission
V3	Cross-Batch Reporting	System-wide metrics
V3	Mentor Feedback System	Quality review
V4	AI Pairing	Smart mentor-student recommendations
V4	Automated Scheduler	Intelligent mentor slot planning
V4	Alert System	Smart notifications for action