NATMS Project Handover Briefing

Naval Aviation Training Management System - Status & Way Forward

Presented by: [Team Lead Name] | Date: [Current Date]

NATMS Project Handover Briefing

Naval AviationTraining Management System –
Status & Way Forward

Presentation Agenda

- Project Overview & Objectives
- System Architecture & Modules
- Technology Stack
- Current Implementation Status
- Gaps & Technical Debt
- Roadmap & Way Forward
- Resource Requirements
- Next Steps

Executive Summary - At a Glance

What is NATMS?

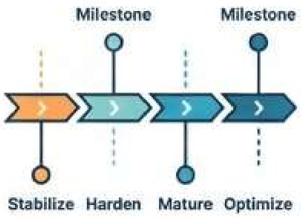
Multi-module web platform for naval aviation training operations management.

Current State: Functional Beta

- ✅ All 5 core modules operational
- ✅ Real business value demonstrated
- ⚠️ Security & operations need hardening

Key Challenge: Not Production-Ready

- ❗ Security gaps
- ❗ Duplicate architecture
- ❗ Missing operational visibility



Recommended Action: 6-Month Roadmap

Phased approach to achieve production readiness with clear milestones.

NATMS –Naval Aviation Training Management System

A centralized platform for managing critical training operations, serving trainees, instructors, directors, training faculty, and system administrators

Facility Management

- Auditorium booking
- Approval workflow

Attendance Tracking

- Instructor presence
- Classroom schedules

Training Materials

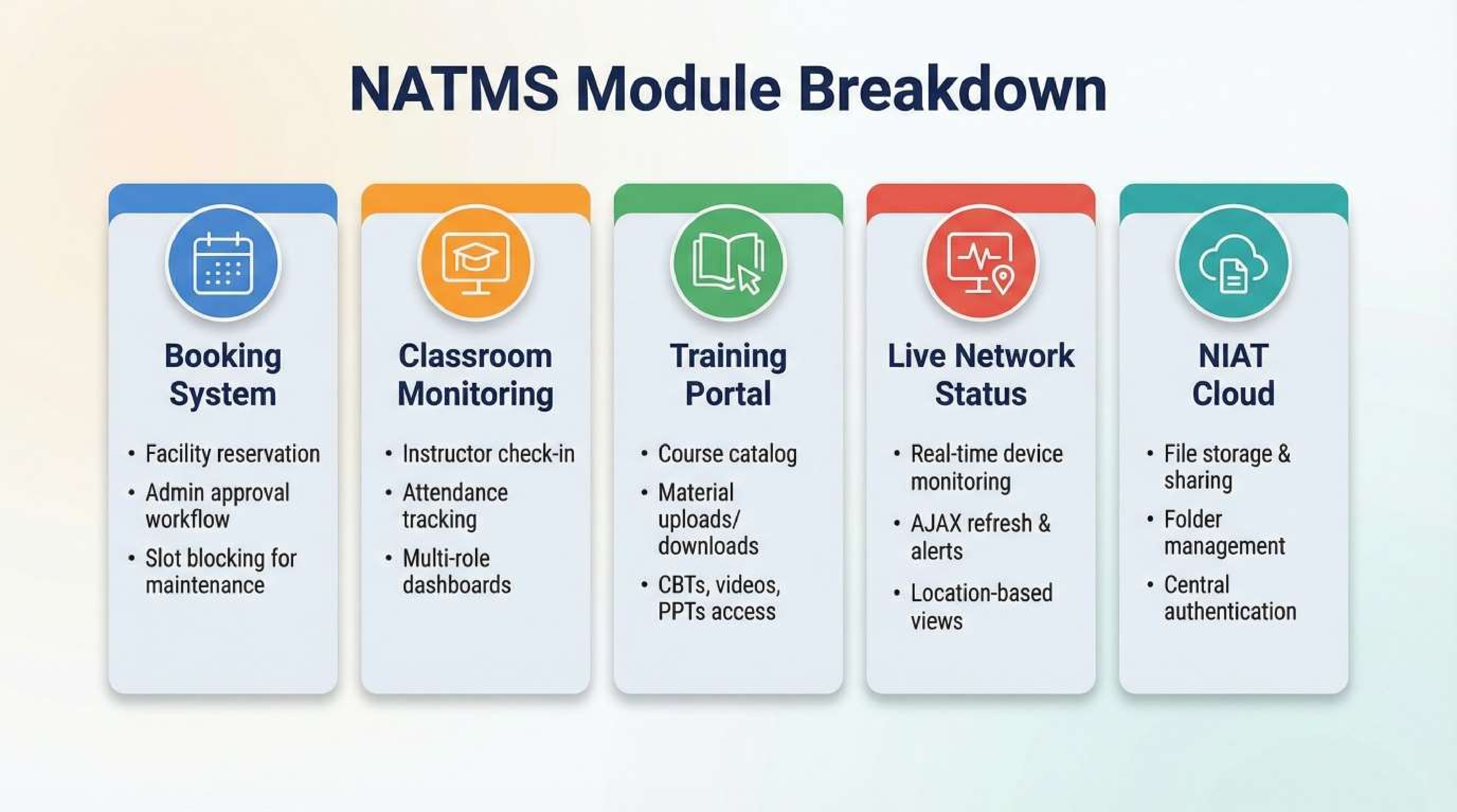
- Centralized course catalog
- PPTs, videos, CBTs

Network Monitoring

- Real-time device status
- Across locations

Document Sharing

- Internal cloud storage
- Central authentication



NATMS Module Breakdown

Booking System

- Facility reservation
- Admin approval workflow
- Slot blocking for maintenance

Classroom Monitoring

- Instructor check-in
- Attendance tracking
- Multi-role dashboards

Training Portal

- Course catalog
- Material uploads/ downloads
- CBTs, videos, PPTs access

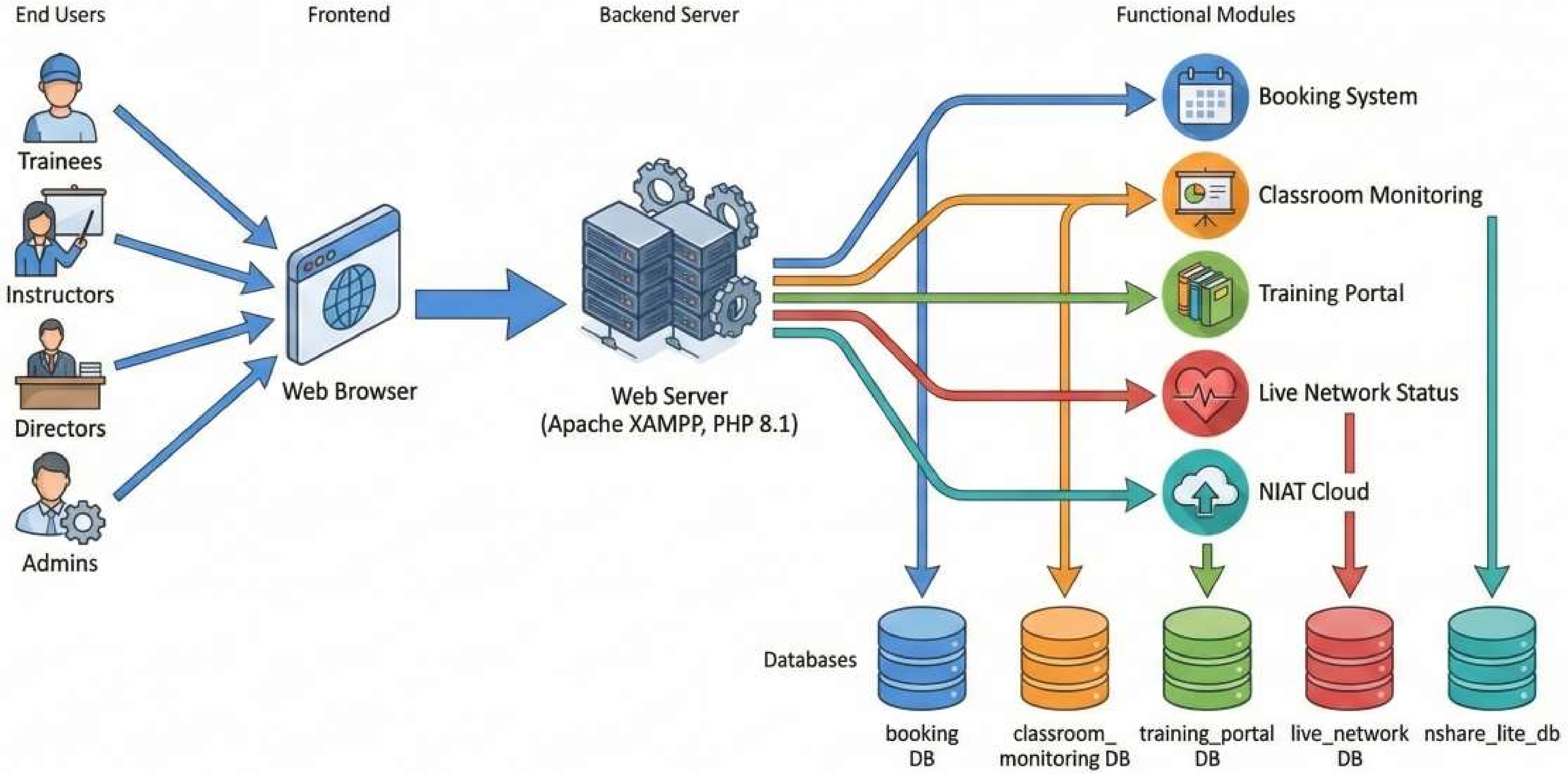
Live Network Status

- Real-time device monitoring
- AJAX refresh & alerts
- Location-based views

NIAT Cloud

- File storage & sharing
- Folder management
- Central authentication

System Architecture Overview



Technology Stack & Infrastructure

Backend

- PHP 8.1+
- MySQL 10.4.32 (MariaDB)
- PDO & MySQLi Access

Frontend

- Tailwind CSS Framework
- Vanilla JavaScript
- Pikaday.js
- Responsive Modern UI

Infrastructure

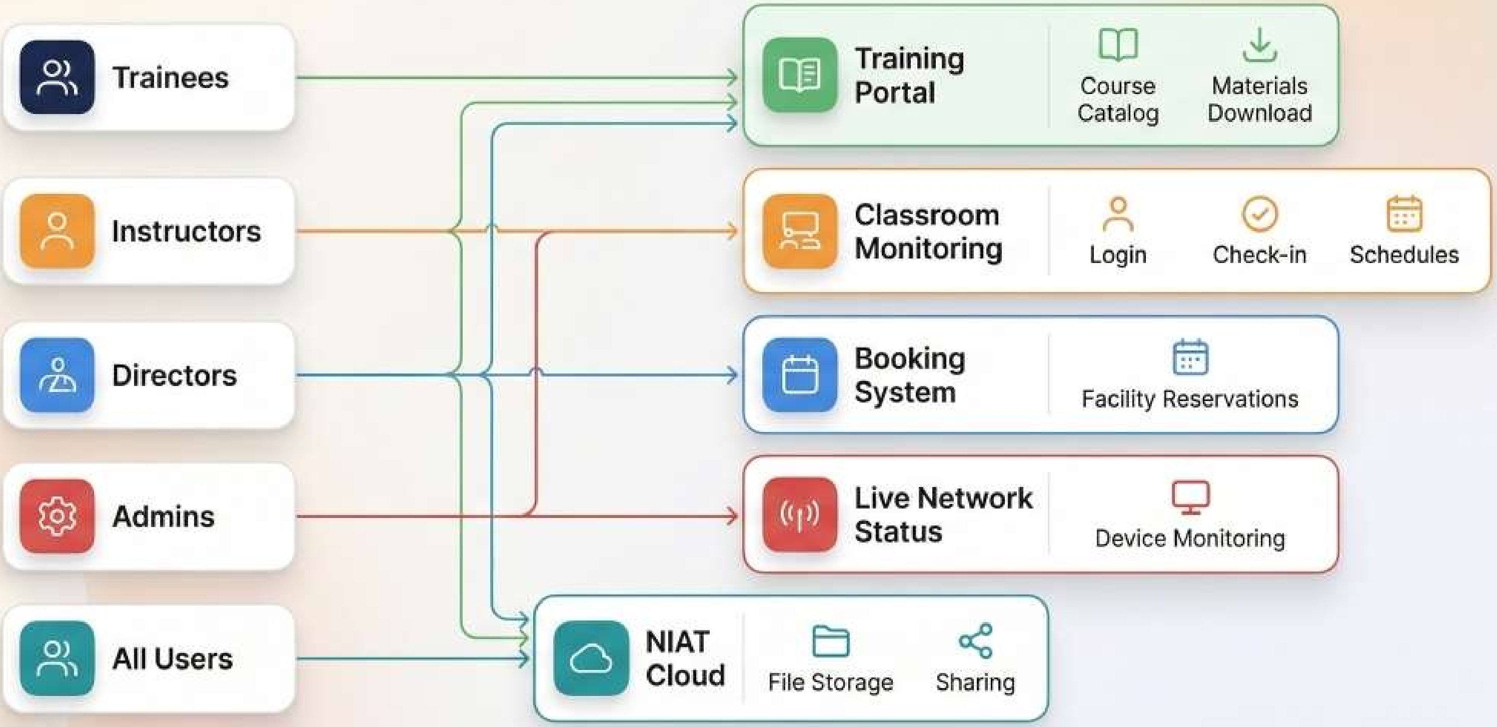
- XAMPP (Apache + PHP + MySQL)
- Ubuntu 24.04 LTS
- Traditional LAMP Stack

Database & Gaps

- 5 Separate Databases
- Notable Gaps:
  - No Docker/ Containerization
  - No CI/CD Pipeline
  - No .env Configuration

Module Ecosystem & User Journeys

Mapping user roles to functional modules and capabilities



Trainees

Instructors

Directors

Admins

All Users

Training Portal

Course Catalog

Materials Download

Classroom Monitoring

Login

Check-in

Schedules

Booking System

Facility Reservations

Live Network Status

Device Monitoring

NIAT Cloud

File Storage

Sharing

Current Current Capabilities - What’s Working

Booking System

- End-to-end reservation workflow
- Admin approval/rejection
- Slot blocking
- Role-based access

Live Network Status

- Real-time device monitoring
- AJAX auto-refresh (30s)
- Location/type filters
- Color-coded status

Classroom Monitoring

- Instructor login & check-in
- Weekly schedule management
- Multi-role dashboards
- Attendance tracking

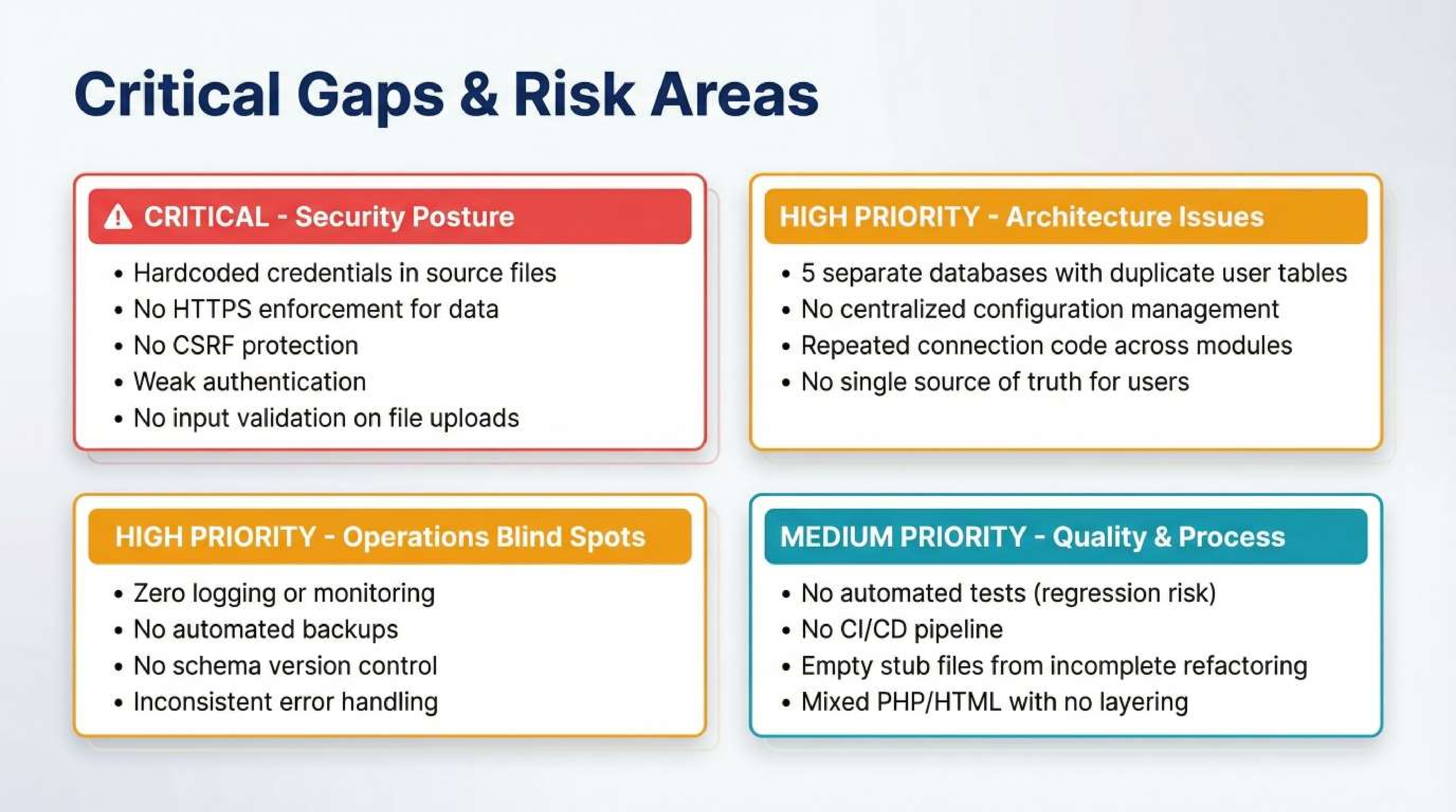
NIAT Cloud

- File upload/download
- Central authentication
- Folder management
- Session management

Training Portal

- Complete course catalog
- Material upload (7 types)
- Download functionality
- User management

Assessment: Core business value is delivered and users can perform their jobs with this system.



Critical Gaps & Risk Areas

⚠ CRITICAL - Security Posture

- Hardcoded credentials in source files
- No HTTPS enforcement for data
- No CSRF protection
- Weak authentication
- No input validation on file uploads

HIGH PRIORITY - Architecture Issues

- 5 separate databases with duplicate user tables
- No centralized configuration management
- Repeated connection code across modules
- No single source of truth for users

HIGH PRIORITY - Operations Blind Spots

- Zero logging or monitoring
- No automated backups
- No schema version control
- Inconsistent error handling

MEDIUM PRIORITY - Quality & Process

- No automated tests (regression risk)
- No CI/CD pipeline
- Empty stub files from incomplete refactoring
- Mixed PHP/HTML with no layering

Implementation Status Matrix

Comprehensive Heat Map & Readiness Assessment (Overall Status: 40% Ready)

🟢 Complete/Production-Ready 🟡 Partial/Needs Work 🔴 Missing/Critical Gap

	Booking System	Classroom Monitoring	Training Portal	Live Network Status	NIAT Cloud
Core Business Logic	🟢	🟡 10%	🟢 40%	🔴	🟢
User Interface	🟢	🟠	🟡 20%	🟡	🟢
Admin Functions	🟢	🟡 20%	🟢 20%	🔴	🟢
Authentication/Session	🟢	🟡	🟢 10%	🔴 70%	🟢
Input Validation	🟢	🟡	🟢 10%	🔴	🟢
Security Hardening	🔴	🔴	🟡 15%	🟡	🔴
Logging & Monitoring	🟢	🔴	🟢 20%	🔴	🟡 20%
Automated Tests	🟢	🟡 30%	🟢 10%	🔴	🔵
Documentation	🟢	🟡 20%	🟢 10%	🔴	🔵

Presented by: [Team Lead Name] | Date: [Current Date]

Risk Assessment - If Deployed As-Is

Analysis of critical security and operational risks in current implementation.

Credential Leak/Compromise

Probability: Medium | **Impact: CRITICAL** 🔴

- Database breach
- Data loss
- Unauthorized access

Malicious File Upload

Probability: High | **Impact: HIGH** 🔴

- Server compromise
- Malware injection
- Data theft

Data Breach (No HTTPS)

Probability: High | **Impact: CRITICAL** 🔴

- Sensitive training data exposed
- Data in transit vulnerable

User Sync Issues

Probability: High | **Impact: MEDIUM** 🟡

- Admin overhead
- User frustration
- Access conflicts

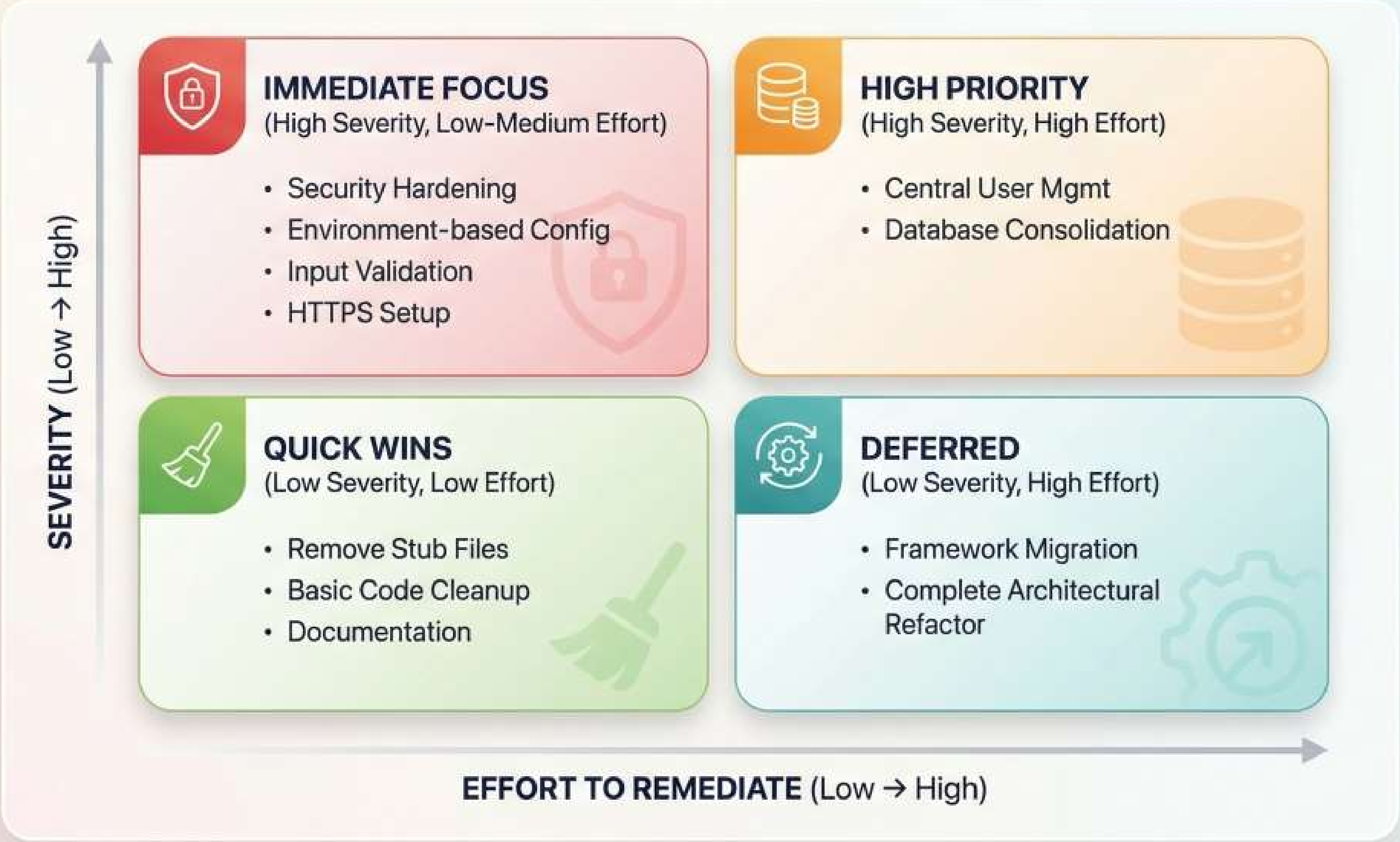
Production Outage

Probability: Medium | **Impact: HIGH** 🔴

- Training disruption
- No troubleshooting capability

⚠ **Bottom Line: System demonstrates value but is NOT production-ready from security and operations standpoint.**

Technical Debt & Effort Matrix



IMMEDIATE FOCUS
(High Severity, Low-Medium Effort)

- Security Hardening
- Environment-based Config
- Input Validation
- HTTPS Setup

HIGH PRIORITY
(High Severity, High Effort)

- Central User Mgmt
- Database Consolidation

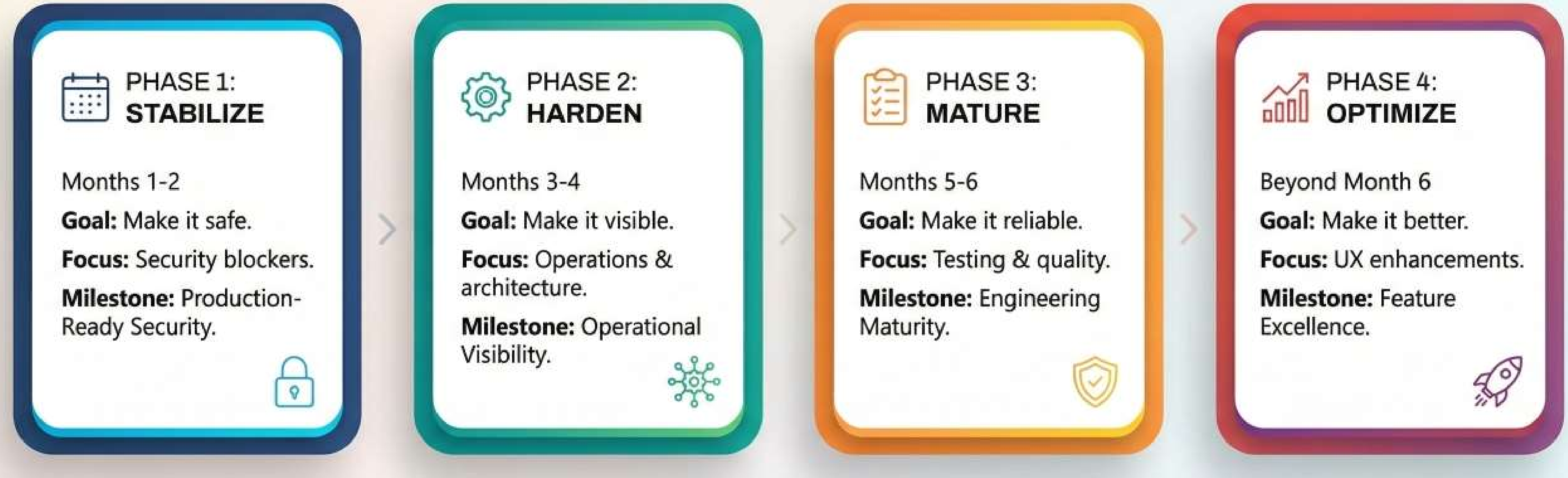
QUICK WINS
(Low Severity, Low Effort)

- Remove Stub Files
- Basic Code Cleanup
- Documentation

DEFERRED
(Low Severity, High Effort)

- Framework Migration
- Complete Architectural Refactor

Phased Approach to Production Readiness



PHASE 1:
STABILIZE

Months 1-2

Goal: Make it safe.

Focus: Security blockers.

Milestone: Production-Ready Security.

PHASE 2:
HARDEN

Months 3-4

Goal: Make it visible.

Focus: Operations & architecture.

Milestone: Operational Visibility.

PHASE 3:
MATURE

Months 5-6

Goal: Make it reliable.

Focus: Testing & quality.

Milestone: Engineering Maturity.

PHASE 4:
OPTIMIZE

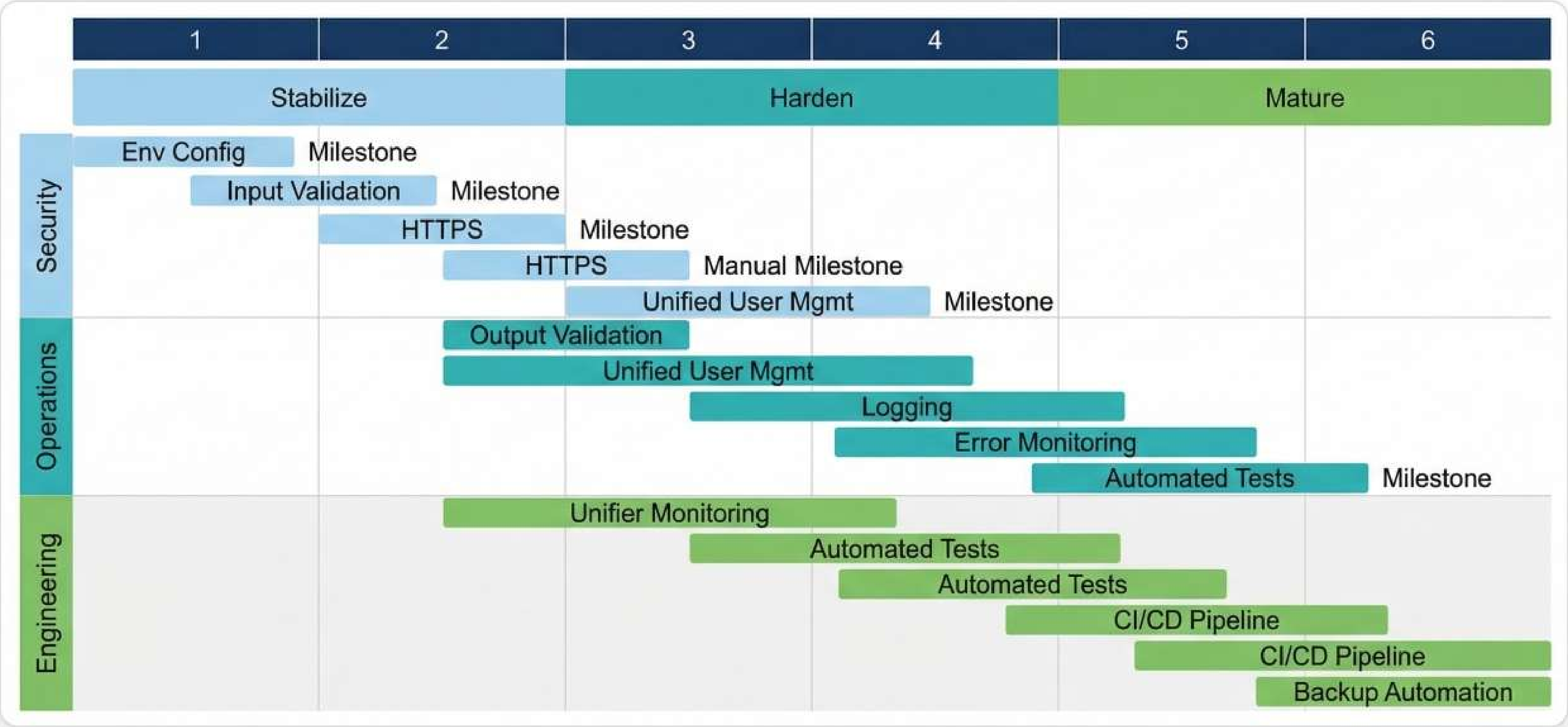
Beyond Month 6

Goal: Make it better.

Focus: UX enhancements.

Milestone: Feature Excellence.

6-Month Roadmap - Detailed Timeline



1 2 3 4 5 6

Stabilize

Harden

Mature

Security

Env Config Milestone

Input Validation Milestone

HTTPS Milestone

HTTPS Manual Milestone

Unified User Mgmt Milestone

Operations

Output Validation

Unified User Mgmt

Logging

Error Monitoring

Automated Tests Milestone

Engineering

Unifier Monitoring

Automated Tests

Automated Tests

CI/CD Pipeline

CI/CD Pipeline

Backup Automation

Phase 1: STABILIZE - Make it Safe (Months 1-2)

Environment-Based Configuration

- Use .env file for all credentials
- Remove hardcoded passwords
- Eliminate credential leak risk

Input Validation & Sanitization

- File upload checks (size, MIME, extension)
- Form inputs sanitize, escape, type-cast
- Block injection attacks

HTTPS Enablement

- SSL/TLS certificate installation
- Force HTTPS redirects
- Protect data in transit

Centralized DB Connection

- Single connection abstraction layer
- DRY principle compliance
- Easier debugging

Resources Required:

2 Developers × 8 Weeks

1 Part-time Security Reviewer

Success Criteria:

- Zero hardcoded credentials
- 100% file uploads validated
- HTTPS on all pages
- Security audit passing

Phases 2 & 3 - Harden & Mature

Phase 2 (Months 3-4): HARDEN

Goal: Operational Visibility

- Make the system visible.

Key Deliverables

- Centralized User Management (Merge 5 tables)
- CSRF Protection Implementation
- Centralized Logging Framework
- Error Monitoring & Alerting
- Optional APM Tool Integration

Resources & Milestone

2 Developers + DevOps Support

Milestone: Can monitor & troubleshoot production.

Phase 3 (Months 5-6): MATURE

Goal: Engineering Best Practices

- Make the system reliable.

Key Deliverables

- Automated Testing (Unit/Integration, 70% coverage)
- Database Migration Tool
- CI/CD Pipeline (GitHub Actions)
- Automated Backup System
- Monthly Restore Drills

Resources & Milestone

2 Developers for 8 Weeks

Milestone: Code follows industry standards.

Resource Requirements & Budget

👥 Team Allocation

- 👥 **2 Full-Time Developers** (6 months, PHP/MySQL)
- 👨‍💻 **1 Part-Time Security Reviewer** (Months 1-2, 25%)
- ⚙️ **1 Part-Time DevOps Engineer** (Months 3-4, 25%)

☁️ Infrastructure & Costs

🗄️ **Staging Server:** Mirrors production

🔒 **SSL Certificates:** **FREE** (Let's Encrypt)

💽 **Backup Storage:** ~$50/month (50GB off-site)

📈 **Optional APM Tool:** ~$100/month

💵 Total 6-Month Budget

< $1,500

Personnel uses existing allocation.
Infra: ~$500 one-time + $150/month ongoing.

💡 Alternative Approach

3-Month MVP Hardening: Focus on Phase 1 + critical Phase 2 items
Trade-off: Delays testing/CI/CD for faster production safety.

Success Metrics - How We’ll Measure Progress

Phase 1 (Month 2) - STABILIZE

- ✓ Zero hardcoded credentials in source code
- ✓ 100% of file uploads validated
- ✓ HTTPS enabled on all pages
- ✓ Security audit score passing with no critical vulnerabilities

Phase 2 (Month 4) - HARDEN

- ✓ Single user database with no duplicates
- ✓ All errors logged to centralized system
- ✓ MTTR (Mean Time to Respond) under 15 minutes
- ✓ 100% of admin actions audited

Phase 3 (Month 6) - MATURE

- ✓ Test coverage greater than or equal to 70%
- ✓ Zero manual deployment steps with full CI/CD
- ✓ Schema changes version-controlled
- ✓ Backup restore tested and documented

Immediate Next Steps

Naval Aviation Training Management System - Status & Way Forward

This Week

- ✓ Secure management approval for 6-month roadmap
- ✓ Confirm team allocation of 2 FTE developers
- ✓ Approve minimal budget (~$1,500 for 6 months)

Next 2 Weeks

- ✓ Kick off Phase 1 security hardening sprint
- ✓ Set up project tracking using Jira or Trello board
- ✓ Define success criteria and metrics dashboard

Key Decision & Recommendation

Key Decision:
Full 6-month roadmap versus 3-month MVP approach.

Recommendation:
Full roadmap for sustainable long-term solution.

Call to action: Discuss questions and secure approval to proceed.

Presented by: [Team Lead Name] | Date: [Current Date]

What Happens If We Delay?

Scenario Analysis

Security Breach

Hardcoded credentials + weak validation = inevitable compromise.

User Frustration

Sync issues between duplicate databases = admin overhead.

Data Loss

No backups + no monitoring = catastrophic failure potential.

Maintenance Nightmare

No tests + no CI/CD = every change risks regression.

Bottom Line

The longer we wait, the more expensive and risky remediation becomes. Phase 1 is non-negotiable.

Summary - Clear Path Forward

Current State

✅ **Functional platform**, real business value.

⚠️ **Security & operational gaps** prevent deployment.

Where We Are Going →

Stabilize → Harden → Mature → Optimize

6-month approach: Production-Ready Security, Operational Visibility, Engineering Maturity.

What We Need

👥 **2 FTE Developers** (existing team)

💰 **Minimal Budget: ~$1,500** for 6 months

✅ **Management Approval**

The Ask

Approve the roadmap and authorize Phase 1 kickoff.

Confidence statement: The roadmap is clear, risks are manageable, and ROI is high. We have a proven path to production readiness.