

Tutorwise Project Audit

Comprehensive Security & Development Analysis

- Report Date:** November 17, 2025
- Current Snapshot:** November 17, 2025 at 20:20:18
- Previous Baseline:** November 1, 2025 at 10:16:06
- Audit Period:** Comparing current codebase against previous baseline
- Report Version:** v1.0.0
- Generated By:** Tutorwise Audit System

Executive Summary

Overview

TutorWise is an innovative tutoring marketplace platform that reimagines how tutors, clients (parents/students), and agents interact through a unified multi-role ecosystem.

Project Health Dashboard

Metric	Current	Previous	Trend
Health Score	9.0/10	10.0/10	Declined -1.0
Critical Issues	0	0	Stable 0
High Priority	0	0	Stable 0
Medium Priority	0	0	Stable 0

Key Performance Indicators

- Overall Status:** Excellent
- Risk Level:** Low
- Code Quality:** Enterprise Scale
- Security Posture:** 7.5/10

Platform Architecture

Monorepo Structure

- Frontend:** Next.js 14+ (TypeScript, Tailwind CSS, Radix UI) in apps/web/
- Backend:** FastAPI (Python) in apps/api/
- Shared Packages:** TypeScript types in packages/shared-types/
- Deployment:** Vercel (frontend), Railway (backend)

Technology Stack

Core Technologies

- Database:** Supabase PostgreSQL, Neo4j Graph Database, Redis
- Authentication:** Supabase Auth
- Payments:** Stripe Connect
- Testing:** Jest, Playwright, Percy (visual testing)
- AI Integration:** Gemini Pro, Claude Code CLI

Development Tools

- **Package Manager:** npm
- **Monorepo Management:** Turborepo-compatible structure
- **Version Control:** Git with GitHub
- **CI/CD:** GitHub Actions, Vercel, Railway

Key Platform Innovations

1. **Single Account, Multi-Role:** Users can switch between Client, Tutor, and Agent roles seamlessly
2. **Dynamic Dashboards:** Role-based interfaces (Learning Hub, Teaching Studio, Tutoring Agency)
3. **Seven Revenue Streams:**
 - Marketplace listing
 - Reverse marketplace
 - Group sessions
 - Job board
 - Course sales
 - AI tutors
 - Referrals
4. **Network & Connections:** Organize contacts into groups
5. **Anyone Can Refer Anything to Anyone (RATA):** Built-in referral system

Development Activity Analysis

Recent Development Summary

- **Commits:** 1 commits since last audit
- **Files Changed:** 1 files modified
- **Contributors:** 1 active developers

Git Change Analysis

Recent Commits (Last 1 commits)

```
262ce4e feat(wiselists): Implement [My Lists] / [Shared With Me] tabs (v5.7.1 UX pattern)
ecd1aa fix(wiselists): Use API route instead of direct Supabase query in getWiselist
a449109 fix(wiselists): Change grid layout to vertical stack for HubRowCard v1 pattern
ac1c3b8 fix(build): Remove pages/_document.tsx and move Material Symbols font to App Router layout
e816cab feat(wiselists): Redesign WiselistCard to match HubRowCard v1 specification
94853eb fix(wiselists): Fix detail page loading by splitting query into separate calls
f952157 feat(wiselists-v5.7.1): Implement Gap 2 - Growth Engine Referral Integration
97a404a feat(wiselists-v5.7.1): Implement Gap 1 - Save Bridge
93e46bd docs(wiselists): Create comprehensive remediation plan v5.7.1
26ee44b fix(wiselists): Add error state handling to wiselist detail page
772d337 fix: Create profile search API endpoint for network connections
c9f619f fix: Create booking details page to resolve 404 error
cefa37a feat(wiselists): Move helper text to top with 24px spacing
1a1c1c9 fix(widgets): Restore standard 16px padding to align with header
e6b1cc9 fix(widgets): Reduce sidebar widget padding for better fit
02b57fd feat(wiselists): Always display stats rows for better UX and awareness
6abad6a feat(wiselists): Update stats widget to horizontal row design
9f3efdb feat(wiselists): Align Wiselists page with Network/My Students design system
d72a2f8 feat: Always show Free Help Now badge with active/inactive states
a75e635 fix: Add Wiselists to sidebar and make Free Help work without Redis
7f8a1f0 fix(free-help): Fix deprecated Supabase client in Free Help API endpoints
8e494ee feat(v5.5-v5.9): Implement CaaS Engine, Wiselists, WiseSpace, and Free Help Now
```

File Modification Summary

```
apps/web/next-env.d.ts | 1 -
.../app/(authenticated)/wiselists/page.module.css | 10 +-
.../web/src/app/(authenticated)/wiselists/page.tsx | 65 ++++----
.../components/wiselists/WiselistCard.module.css | 158 ++++++-----
.../src/app/components/wiselists/WiselistCard.tsx | 164 ++++++-----
apps/web/src/app/layout.tsx | 8 +-
apps/web/src/lib/api/wiselists.ts | 52 ++++---
```

apps/web/src/pages/_document.tsx | 19 ---
8 files changed, 272 insertions(+), 205 deletions(-)

Detailed File Changes

M	apps/web/next-env.d.ts
M	apps/web/src/app/(authenticated)/wiselists/page.module.css
M	apps/web/src/app/(authenticated)/wiselists/page.tsx
M	apps/web/src/app/components/wiselists/WiselistCard.module.css
M	apps/web/src/app/components/wiselists/WiselistCard.tsx
M	apps/web/src/app/layout.tsx
M	apps/web/src/lib/api/wiselists.ts
D	apps/web/src/pages/_document.tsx

Development Team Activity

22 micquan

Latest Itemized Changes Since Last Report

Summary of Changes

Change Type	Count
Files Added	0
Files Modified	7
Files Deleted	1
Files Renamed	0
Total Changes	8

Files Modified

- apps/web/next-env.d.ts
- apps/web/src/app/(authenticated)/wiselists/page.module.css
- apps/web/src/app/(authenticated)/wiselists/page.tsx
- apps/web/src/app/components/wiselists/WiselistCard.module.css
- apps/web/src/app/components/wiselists/WiselistCard.tsx
- apps/web/src/app/layout.tsx
- apps/web/src/lib/api/wiselists.ts

Files Deleted

- apps/web/src/pages/_document.tsx

Project Metrics & Architecture

Codebase Composition

File Type	Count	Percentage
TypeScript	5301	8.5%
React Components	340	0.5%
JavaScript	17823	28.5%
Markdown	1791	2.9%
CSS/Styles	214	0.3%

File Type	Count	Percentage
Configuration	1781	2.8%
Total Files	62575	100%

Code Quality Metrics

- **Lines of Code:** 169,577
- **React Components:** 225
- **Dependencies:** 16 production + 17 development
- **NPM Scripts:** 87 available commands
- **Node.js Version:** Not specified

Security Assessment

- **Environment Security:** Configured
- **Git Security:** Configured
- **Security Documentation:** Missing
- **AI Restrictions:** Implemented
- **Overall Security Score:** 7.5/10

Recent Security Incidents & Remediation

Critical Security Issues (September 30, 2024)

Incident #1: Environment Variable Access Violation

AI agent gained unrestricted access to production secrets across all platforms (Vercel, Railway, Supabase, Neo4j, Redis, Terraform, Google Cloud).

Risk Level: CRITICAL **Impact:** Potential data breach, service disruption, unauthorized access

Incident #2: Project Scope Violation

AI agent performed system-wide file searches outside project boundaries, accessing other projects and directories.

Risk Level: HIGH **Impact:** Unauthorized access to other projects, confusion with wrong accounts, scope creep

Implemented Security Controls

- **Comprehensive AI RBAC system:** tools/change-management/ai-permission-system.js
- **Human approval workflow:** Required for sensitive changes (tools/change-management/approval-workflow.js)
- **Automated audit system:** Daily/weekly reports with email notifications

Recent Achievements

- Onboarding infinite loading bug fixed
- Security framework successfully implemented
- Navigation menu spacing issues resolved
- Auto-save/resume onboarding functionality completed
- API routes cleaned up for frontend-only deployment

Outstanding Issues

Critical

- No critical issues identified 

High Priority

- Security documentation could be expanded
- Consider adding enhanced monitoring dashboard
- Performance optimization review recommended

Development Workflow & Automation

AI-Assisted Development

The project leverages extensive AI automation through Claude Code CLI:

- **Context Engineering:** Comprehensive project context for AI assistance
- **Automated Task Execution:** Integration with Jira and Calendar events
- **Code Generation:** AI-assisted feature implementation
- **Testing Infrastructure:** Automated unit, integration, E2E, and visual tests

Testing Strategy

Testing Layers

- **Unit Tests:** Jest for component and utility testing
- **Integration Tests:** API and database integration verification
- **E2E Tests:** Playwright for full user journey testing
- **Visual Tests:** Percy for UI regression detection

Test Coverage

- Comprehensive test plans in docs/testing/
- Automated test execution in CI/CD pipeline
- Visual regression testing for UI changes

Documentation Structure

Well-organized documentation in docs/:

- **Requirements:** Product requirements and specifications
- **Design:** Architecture and design decisions
- **Development:** Development guides and workflows
- **Testing:** Test strategies and test plans
- **Deployment:** Infrastructure and deployment guides
- **Integration:** Third-party integration documentation
- **Features:** Feature-specific workflows and documentation

Automation Tools

- **Context Management:** Automated context collection for AI
- **Jira Integration:** Task synchronization and automation
- **Google Workspace:** Calendar and Docs integration
- **Confluence:** Documentation synchronization
- **Audit System:** Daily/weekly automated project audits

Trend Analysis

Performance Trends

Health Score Evolution: Declined (-1.0 points)

Issue Resolution Progress: No Change (+0 issues)

Development Velocity: Active development with 1 commits since last audit

Quality Metrics:

- Code base: 169,577 lines across 62575 files
- Component architecture: 225 React components identified

Strategic Recommendations

Immediate Actions (Next 24 hours)

1. Monitor Deployment Pipeline

- Verify automated deployments are running smoothly
- Check deployment logs for any warnings
- Validate environment variables are up to date

2. Improve Security Posture

- Complete security documentation
- Review and update access controls
- Validate security controls in practice

Short-term Goals (Next Week)

1. Enhanced Monitoring Setup

- Consider adding deployment monitoring dashboard
- Set up performance metrics tracking
- Implement error tracking and alerting

2. Documentation Expansion

- Complete comprehensive security audit
- Document security controls and procedures
- Validate AI RBAC system effectiveness

3. Testing Infrastructure

- Implement automated testing for deployment pipeline
- Expand E2E test coverage
- Set up continuous visual regression testing

4. Documentation Updates

- Update deployment documentation
- Document security incident response procedures
- Maintain architectural decision records

Long-term Strategy (Next Month)

1. Deployment Monitoring & Observability

- Implement deployment health monitoring
- Set up error tracking and alerting
- Create deployment metrics dashboard

2. Security Hardening

- Regular security audits (weekly/monthly)
- Penetration testing for critical paths
- Security training for development team

3. Architecture Evolution

- Plan for scalability improvements
- Evaluate microservices opportunities
- Optimize database queries and caching

4. CI/CD Enhancement

- Automated quality gates
- Performance benchmarking in CI
- Automated dependency updates

Conclusion

Overall Assessment

TutorWise is a **well-architected, ambitious platform** with strong technical foundations. The monorepo structure, comprehensive testing infrastructure, and AI-assisted development workflow are notable strengths that position the project for long-term success.

Key Strengths

- **Innovative Architecture:** Single account, multi-role system with dynamic dashboards
- **Comprehensive Testing:** Full test coverage across unit, integration, E2E, and visual layers
- **Modern Tech Stack:** Next.js, FastAPI, Supabase, Neo4j, and Stripe Connect integration
- **AI-Assisted Development:** Advanced automation with Claude Code CLI and context engineering
- **Robust Security:** Recent incidents led to implementation of strong governance controls

Focus Areas

Current Health: 9.0/10 (declining from 10.0/10) - Good

Primary Concerns:

1. Deployment configuration (Vercel account setup)
2. GitHub → Vercel auto-deployment integration
3. Security documentation completion

Security Posture: 7.5/10 - Recent security incidents resulted in comprehensive RBAC system and AI restrictions. Ongoing validation required.

Technical Excellence: With 169,577 lines of code across 62,575 files, the codebase demonstrates Enterprise Scale characteristics with 225 React components and modern TypeScript patterns.

Path Forward: Main focus areas are deployment automation cleanup and security validation. The strong technical foundation supports rapid feature development once deployment pipeline is stabilized.

Next Steps

1. Review and address immediate action items listed above
2. Monitor project health metrics through automated daily audits
3. Continue security hardening and documentation improvements
4. Maintain code quality standards and testing practices

Audit Methodology

Data Sources

- **Git Repository:** Commit history, file changes, contributor activity
- **File System:** Code metrics, file composition, project structure
- **Configuration:** Dependencies, scripts, environment setup
- **Security:** Access controls, documentation, restrictions
- **Previous Audits:** Historical comparison and trend analysis

Analysis Techniques

- Statistical trend analysis comparing current vs. previous metrics
- Git diff analysis for code change impact assessment
- File system scanning for project composition metrics
- Security posture evaluation based on best practices
- Risk assessment using weighted priority scoring

Report Validation

- Cross-referenced with previous audit findings
 - Verified against project documentation
 - Validated with automated metric collection
 - Reviewed for accuracy and completeness
-

Report Metadata

Report ID: AUDIT-2025-11-17-mi3l9s40 **Generation Time:** 2025-11-17T20:20:18.960Z **Report Format:** Unified Markdown + PDF **Data Collection Period:** October 1, 2025 to Unknown **Next Scheduled Audit:** 11/24/2025

Automated Distribution: Email sent to tutorwiseapp@gmail.com **Archive Location:** docs/project-audit/ **Backup Location:** tools/snapshots/

References

Documentation

- Project Repository: /Users/michaelquan/projects/tutorwise
- Previous Audits: docs/project-audit/
- Security Guidelines: .ai-restrictions
- Change Management: docs/ai-change-management.md

Tools & Systems

- Audit System: tools/scripts/project-audit.js
- Email Notifications: tools/scripts/email/send-audit-email.js
- Snapshot Storage: tools/snapshots/
- Cron Schedule: Daily at 6:00 AM and 7:00 PM

Contact Information

- Email: tutorwiseapp@gmail.com
- Project Admin: @tutorwiseapp
- Product Engineer: @micquan

This report was automatically generated by the Tutorwise Project Audit System. For questions or concerns, please contact the development team.