Tutorwise Project Audit

Comprehensive Security & Development Analysis

- **Report Date**: October 1, 2025
- **Current Snapshot**: October 1, 2025 at 20:34:21
- **Previous Baseline**: October 1, 2025 at 19:00:01
- **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	8.0/10	10.0/10	Declined -2.0
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**: Development Stage
- **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)

```
f8846c4 Move railway.json to project root with correct Dockerfile path
ae6dec6 Fix Railway Dockerfile path configuration
50e60d9 Fix Railway backend deployment configuration
63fb182 Fix onboarding completion - implement database persistence and navigation
c84737a Fix broken onboarding UI - constrain checkmark icon size
82a5152 Fix testing infrastructure across all layers
d7978f7 Schedule daily critical files protection reports via cron
6d9bd98 Implement critical files protection system (Option 3)
c0a0660 Restore all API routes from git history
9c49501 Implement comprehensive project audit system with email automation
f5e52ee Test automatic GitHub → Vercel deployment
c2dc81b Add AI security controls and RBAC system
69407b0 URGENT: Fix onboarding step transition error and implement AI security controls
1d561ee Remove API routes entirely for frontend-only deployment
0e084a9 Temporarily disable all API routes for onboarding deployment
a54df14 Temporarily disable API route to fix production deployment
75af3a3 Fix API route env vars for production deployment
e3f3c3c Remove secret manager files causing build issues
c976afc Fix onboarding auto-resume and navigation issues
61106fa Fix onboarding SSR error with Suspense boundary
2ba27bb Fix TypeScript null check error for searchParams
1f1a4ca Fix TypeScript function declaration order error
eb635a3 Fix TypeScript build error: Add missing OnboardingProgress properties
56bfb9e Implement comprehensive onboarding auto-save and resume functionality
3016184 Fix divider line overlap issue with 12px spacing
7c05989 Fix duplicate divider above 'Become a tutor' section
bbcd6fa Improve NavMenu code quality and consistency
20b0107 Add divider line above 'Become a tutor' section
ca5e3fd Adjust navigation menu divider spacing to 8px
e78059b Fix navigation menu divider with reliable div element and 12px spacing
531ada5 fix: change spacing below 'Grow your business' from 16px to 8px
ab9a4c8 fix: correct spacing - revert My network to 8px, add 8px below 'Grow your
business'
89218c1 fix: add surgical 8px spacing only after 'My network' section
a6ed4c0 fix: standardize vertical spacing and center separators properly
15a7dcf fix: add explicit styling to separator above Account section
15fdba0 fix: ensure divider line appears above Account section
b56f4a5 fix: increase separatorExtra spacing to 24px total
4ef561e fix: add 8px extra vertical spacing where indicated by red lines
c716b39 fix: standardize all navigation separator spacing to 8px
6171ae4 fix: ensure navigation separators are visible with CSS fallbacks
d62f388 fix: add divider line between 'Become an agent' and 'Account' sections
4760dcc fix: ensure divider line always appears after 'My network'
200c696 feat: implement visual testing approach and fix navigation spacing
b56bea0 fix: ensure consistent vertical spacing in navigation menu
c40d0ed fix: add missing divider line under 'My network' in navigation menu
c14d729 fix: enforce onboarding redirect on middleware database errors
```

File Modification Summary

```
.github/workflows/ci.yml                              |   4 +-
 .github/workflows/deploy.yml                         |   2 +-
 RAILWAY_DEPLOYMENT_FIX.md                            |  34 +++
 .../app/components/onboarding/OnboardingWizard.tsx   |  42 +++-
 .../onboarding/steps/CompletionStep.module.css       | 241 ++++++++++++++++++++
 .../components/onboarding/steps/CompletionStep.tsx   |  85 +++++----
 docker-compose.yml                                   |  16 +-
 apps/api/railway.json => railway.json                |   3 +-
 8 files changed, 371 insertions(+), 56 deletions(-)
```

Detailed File Changes

```
M   .github/workflows/ci.yml
M   .github/workflows/deploy.yml
A   RAILWAY_DEPLOYMENT_FIX.md
M   apps/web/src/app/components/onboarding/OnboardingWizard.tsx
A   apps/web/src/app/components/onboarding/steps/CompletionStep.module.css
M   apps/web/src/app/components/onboarding/steps/CompletionStep.tsx
M   docker-compose.yml
R088    apps/api/railway.json   railway.json
```

Development Team Activity

```
46 micquan
```

Latest Itemized Changes Since Last Report

Summary of Changes

Change Type	Count
Files Added	2
Files Modified	5
Files Deleted	0
Files Renamed	0
Total Changes	**8**

Files Added

- `RAILWAY_DEPLOYMENT_FIX.md`
- `apps/web/src/app/components/onboarding/steps/CompletionStep.module.css`

Files Modified

- `.github/workflows/ci.yml`
- `.github/workflows/deploy.yml`

- `apps/web/src/app/components/onboarding/OnboardingWizard.tsx`
- `apps/web/src/app/components/onboarding/steps/CompletionStep.tsx`
- `docker-compose.yml`

Project Metrics & Architecture

Codebase Composition

File Type	Count	Percentage
TypeScript	103	1.5%
React Components	75	1.1%
JavaScript	215	3.0%
Markdown	141	2.0%
CSS/Styles	59	0.8%
Configuration	129	1.8%
Total Files	**7058**	**100%**

Code Quality Metrics

- **Lines of Code**: 39,673
- **React Components**: 14
- **Dependencies**: 8 production + 4 development
- **NPM Scripts**: 66 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/rbac/ai-permission-system.js
- **AI restrictions file**: .ai-restrictions with forbidden actions
- **Project scope limits**: Enforced boundaries at /Users/michaelquan/projects/tutorwise
- **Human approval workflow**: Required for sensitive changes (tools/rbac/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 (-2.0 points)

Issue Resolution Progress: No Change (+0 issues)

Development Velocity: Active development with 1 commits since last audit

Quality Metrics:

- Code base: 39,673 lines across 7058 files
- Component architecture: 14 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: 8.0/10 (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 39,673 lines of code across 7,058 files, the codebase demonstrates Development Stage characteristics with 14 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-2025-10-01-mg8dxn7m **Generation Time**: 2025-10-01T19:34:21.634Z **Report Format**: Unified Markdown + PDF **Data Collection Period**: October 1, 2025 to October 1, 2025 **Next Scheduled Audit**: 10/8/2025

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.