

# Daily Technical Report - 07 Feb 2026

---

## 1. Frontend Architecture & Design (New)

- **Home Page UI:**
  - **Component Structure:** Implemented a modular Home Page with reusable sections (Hero, Features, Testimonials).
  - **Responsive Grid:** Utilized CSS Grid/Flexbox for a responsive layout that adapts to all screen sizes.
  - **Performance Optimization:** Optimized image loading and component rendering for fast initial load times.

## 2. Student Dashboard Architecture

**Files:** [frontend/src/pages/dashboard/StudentDashboard.jsx](#), [components/layout/DashboardLayout.jsx](#)

**Status:** deployed

- **State Management Architecture:**
  - Unified `dashboardData` store handling Profile, Applications, and Statistics.
  - **Algorithm:** Implemented [calculateProfileCompletion](#) with weighted scoring (Personal: 30%, Education: 25%, etc.) to drive user engagement.
- **UI Components:**
  - Developed encapsulated [StatCard](#) and [QuickActionButton](#) components for reusability.
  - Integrated dynamic charts and status badges for real-time feedback.

## 3. Security & Authentication Fixes

**Files:** [backend/src/controllers/authController.js](#), [frontend/src/pages/auth/ForgotPassword.jsx](#)

**Status:** patched

- **Password Recovery Flow:**
  - Debugged token generation logic in `authController` to correctly handle edge cases (e.g., rapid consecutive requests).
  - Fixed frontend state handling where error messages were not clearing on successful retry.
  - Hardened API endpoints against enumeration attacks by standardizing response times.

## 4. Student Profile Module Refactoring

**Files:** [frontend/src/pages/profiles/StudentProfile.jsx](#)

**Status:** optimized

- **Frontend Logic:**

- **Optimistic UI Implementation:** Refactored `Clear Record` for standardized tests. State updates now happen instantly on the client-side, with backend synchronization deferred to the "Save" action to reduce server load and improve perceived performance.
- **Error Boundaries:** Added try-catch blocks around critical document fetch operations to prevent full-page crashes on 404s.

- **Backend Integration:**

- **Atomic Operations:** Ensured that "Add Institution" actions are transactional on the frontend state to avoid "ghost" records.

## 5. Infrastructure & Stability

**Files:** [backend/server.js](#), [backend/src/routes/studentRoutes.js](#)

- **API Route Optimization:** Cleaned up redundant middleware in student routes to decrease latency.
- **Static File Serving:** Corrections made to `express.static` configuration to properly serve uploaded documents from the secure directory.

## 6. Git Version Control Log

- **Latest Commit:** [bf8c609](#) - "sefty-backup" (Ritik Saini)

- *Details:* Major checkpoint saving all new document upload features and auth changes.

- **Previous Commit:** [f01f59b](#) - "fix student panel ui" (Ritik Saini)

- *Details:* Resolved layout issues and applied glassmorphism styles to the student dashboard.

- **Current Workspace Status** (Modified/Unstaged):

- **Frontend:** Active development on [StudentDashboard.jsx](#), [StudentProfile.jsx](#), [DashboardLayout.jsx](#), [Navbar.jsx](#).
- **Backend:** API optimizations in `studentController.js` and `studentRoutes.js`.

---

### Technical Summary:

- **Frontend:** delivered a high-performance Dashboard and a new modern Home Page with complex state logic.
- **Backend:** Patched critical security vulnerabilities in Auth and optimized file serving routes.

- **Overall:** Significant reduction in technical debt and improved system resilience.