

EduAssess Platform - Master Context Document

Complete Project History & Progress

Last Updated: January 2025

Project Overview

EduAssess Platform is an educational assessment platform for Australian students (Years 2-9) focusing on NAPLAN and ICAS exam preparation. Features a kid-friendly, colorful UI with engaging animations.

GitHub Repository: <https://github.com/Vishwas2018/AssessmentPortal> **Local Development:** <http://localhost:5173>

Tech Stack

Technology	Purpose
React 18	Frontend framework
TypeScript	Type safety
Vite	Build tool
TailwindCSS	Styling
Framer Motion	Animations
Zustand	State management
Supabase	Backend/Auth/Database
React Router v6	Routing
React Hook Form + Zod	Form validation
Lucide React	Icons
canvas-confetti	Celebration animations
Font: Nunito	Kid-friendly typography

COMPLETED FEATURES

Phase 1: Foundation

- Complete project structure with TypeScript
- Mock authentication system (later replaced with real auth)
- Colorful, animated landing page with hero section
- Login page with floating stars animation
- Register page with gradients and validation
- Interactive dashboard with stats cards
- Exams browsing page with filtering
- Protected routes setup
- Mock data for testing (5 exams, users, stats)
- Kid-friendly design (bold colors, emojis, animations)
- Responsive layout for all screen sizes

Phase 2: Authentication & Polish

- Supabase project setup and configuration
- Environment variables (.env) configured
- Database tables created with SQL:
 - `(user_profiles)` - Student info (name, year_level, school, parent_email)
 - `(exams)` - Available NAPLAN/ICAS exams
 - `(questions)` - Individual exam questions
 - `(exam_attempts)` - Tracking exam sessions
 - `(user_progress)` - Aggregated progress per subject
 - `(achievements)` - 10 pre-loaded badges
 - `(user_achievements)` - Earned badges tracker
- Row Level Security (RLS) policies for all tables
- Auto-create profile trigger on signup
- Real email/password authentication with Supabase
- Email verification flow
- Google OAuth integration (fully configured)
- Password reset functionality
- Form validation with Zod schemas
- Password strength indicator (Weak/Medium/Strong)
- Session persistence in localStorage

- Protected routes with auth check
- Profile page (view/edit mode with validation)
- Dashboard displays user's actual name from profile
- Footer component with links, social icons, newsletter signup
- Enhanced landing page with:
 - Hero section with CTA buttons
 - Feature cards with animations
 - Stats section (10,000+ students, etc.)
 - How It Works section (3 steps)
 - Testimonials section (4 testimonials)
 - FAQ section (8 questions with accordion)
 - "Why Choose Us" section (6 features)
 - Final CTA banner

Phase 3: Exam Taking Interface

- ExamStartPage - Pre-exam intro with instructions
- ExamPage - Full test-taking interface (ENHANCED - see below)
- ExamResultsPage - After submission with detailed review
- ExamsPage fetches from Supabase (real data)
- Router configuration with exam routes
- Question text formatting helper function

Phase 3.5: Enhanced Exam UI (NEW)

Left Sidebar Question Navigation

- 5x grid layout for question numbers
- Color-coded status (green=answered, gray=unanswered, indigo=current)
- Orange dot indicator for flagged questions
- Scrollable for large exams
- Stats showing "X Done" and "X Flagged"
- Legend explaining color codes

Improved Exam Header

- Timer progress bar (gradient green to red as time decreases)
- Exam title and subject display
- Floating decorative shapes animation
- Dark indigo gradient background

Enhanced Question Card

- Clean white card design
- Question number badge
- Answered checkmark indicator
- Radio-button style answer selection
- Image support for questions

Hint System

- "Hint" button toggles hint visibility
- Blue highlight box with lightbulb icon
- Shows question hint or partial explanation

Flag for Review

- Flag button to mark questions
- Visual indicator on question nav
- Flagged count shown in stats

Smart Navigation Buttons

- Previous/Next buttons
- **"Next" automatically changes to "Finish" on last question**
- Finish button is green, triggers submit modal

Bottom Progress Bar

- Shows "Your Progress: X of Y questions answered"
- "Finish Exam" button always accessible
- Visual progress bar

Confetti Celebration Modal

- "Great Job! 🎉" message on completion
- Colorful confetti animation
- "See My Results" button

Phase 3.5: Enhanced Results Page (NEW)

Two-Tab Interface

- "Results Summary" tab - Score overview
- "Question Review" tab - Detailed breakdown

Results Summary Tab

- Large animated score circle
- Grade emoji based on percentage

- Encouragement message
- Stats grid: Correct / Incorrect / Unanswered
- Time spent and points earned
- Review Answers / Try Another Exam buttons

Question Review Tab

- **Filter buttons:** All, Correct, Incorrect, Unanswered
 - Expandable question cards (click to expand)
 - Color-coded status badges
 - Shows all answer options with:
 - Green highlight for correct answer
 - Red highlight for user's wrong answer
 - "Your Answer" vs "Correct Answer" comparison boxes
 - Blue explanation box for each question
-

Current Project Structure

```

src/
  └── components/
    └── layout/
      ├── Navbar.tsx      # Navigation with user profile dropdown
      ├── Footer.tsx      # Full footer with links & newsletter
      ├── RootLayout.tsx   # Root wrapper, initializes auth
      └── ProtectedLayout.tsx # Auth guard for protected routes
  └── pages/
    ├── HomePage.tsx     # Enhanced landing page
    ├── LoginPage.tsx    # Real Supabase login + Google OAuth
    ├── RegisterPage.tsx  # Real Supabase signup + validation
    ├── ForgotPasswordPage.tsx # Password reset
    ├── DashboardPage.tsx # Shows user's name, stats
    ├── ExamsPage.tsx     # Browse exams (LEFT SIDEBAR FIXED)
    ├── ExamStartPage.tsx # Pre-exam instructions
    ├── ExamPage.tsx      # Test-taking interface (ENHANCED)
    ├── ExamResultsPage.tsx # Results with filtering (ENHANCED)
    ├── ResultsPage.tsx   # All past results
    └── ProfilePage.tsx   # View/edit user profile
  └── lib/
    ├── supabase.ts       # Supabase client config
    └── auth.ts           # Auth service functions

```

```
|   └── validations.ts    # Zod schemas for forms
|   └── router.tsx        # All routes configured
|   └── store/
|       └── index.ts      # Zustand stores (auth, exam, UI)
|   └── types/
|       ├── index.ts      # App TypeScript types
|       └── supabase.ts    # Database types
|   └── data/
|       ├── mockData.ts    # Test users, exams, stats
|       └── constants/index.ts # Routes, subjects, EXAM_TYPES, YEAR_LEVELS
|   └── utils/
|       └── helpers.ts     # Formatting functions
```

🔒 Supabase Configuration

Environment Variables (.env)

```
VITE_SUPABASE_URL=https://ssasvooplzpscaafuedt.supabase.co
VITE_SUPABASE_ANON_KEY=[YOUR_ANON_KEY]
```

Database Tables

All tables have Row Level Security (RLS) enabled:

1. **user_profiles** - Linked to auth.users, auto-created on signup
 - id, full_name, display_name, year_level, school_name, parent_email, avatar_url
2. **exams** - Stores exam metadata
 - id, title, exam_type (NAPLAN/ICAS), subject, year_level, duration_minutes, total_questions, difficulty, is_active, is_free
3. **questions** - Individual questions per exam
 - id, exam_id, question_number, question_text, question_type, options (JSONB), correct_answer, explanation, points, image_url, hint
4. **exam_attempts** - Records each attempt
 - id, user_id, exam_id, started_at, completed_at, time_spent_seconds, score, total_points, percentage, status, answers (JSONB)
 - **Note:** flagged column NOT in database (stored locally during exam only)
5. **user_progress** - Aggregated stats per subject

6. **achievements** - 10 pre-loaded badges
7. **user_achievements** - Junction table for earned badges

Auth Providers Enabled

- Email/Password
 - Google OAuth (configured in Google Cloud Console)
-

📋 Routes Configuration

typescript

```
// Public Routes
/
  → HomePage (landing page)
/login → LoginPage
/register → RegisterPage
/forgot-password → ForgotPasswordPage

// Protected Routes (require auth)
/dashboard → DashboardPage
/exams → ExamsPage (exam listing)
/exam/:examId/start → ExamStartPage
/exam/:examId/take/:attemptId → ExamPage (ENHANCED)
/exam/:examId/results/:attemptId → ExamResultsPage (ENHANCED)
/results → ResultsPage (all past results)
/profile → ProfilePage
```

🎨 Design System

Colors (TailwindCSS)

- Primary: Indigo/Purple gradient (`(from-indigo-500 to-purple-600)`)
- Success: Green (`(green-500)`, `(emerald-500)`)
- Warning: Orange/Yellow (`(orange-500)`, `(amber-500)`)
- Error: Red (`(red-500)`, `(rose-500)`)
- Backgrounds:
 - Exam page: Dark indigo gradient (`(from-indigo-600 via-indigo-700 to-indigo-800)`)
 - Other pages: Soft gradients (`(from-blue-50 via-purple-50 to-pink-50)`)

Typography

- Font: Nunito (configured in Tailwind)
- Headings: `font-black` (900 weight) or `font-bold`
- Body: `font-medium` or `font-semibold`

Component Patterns

- Rounded corners: `rounded-1xl`, `rounded-2xl`, `rounded-3xl`
 - Shadows: `shadow-lg`, `shadow-xl`, `shadow-2xl`
 - Borders: `border-2 border-gray-200`
 - Animations: Framer Motion for transitions
 - Kid-friendly emojis throughout 🎉 🎨 🚀 🏆
-



Common Commands

```
bash
```

```
# Start development server
```

```
npm run dev
```

```
# Build for production
```

```
npm run build
```

```
# Preview production build
```

```
npm run preview
```

```
# Git commands
```

```
git add .
```

```
git commit -m "message"
```

```
git push
```



Known Issues / Notes

1. **Flagged Questions:** Stored in React state only, NOT persisted to database
 - To persist, add `flagged TEXT[]` column to `exam_attempts` table

2. **Mock Data Still Used:** Dashboard uses some mock stats - needs real Supabase data
 3. **Router Warning:** React Router v7 future flag warning appears in console (cosmetic only)
-

REMAINING TASKS (Phase 4+)

Priority 1: Connect Real Data

- Dashboard to show real exam attempts from Supabase
- Profile page to show real stats
- Results page to show exam history from database

Priority 2: Additional Features

- Payment integration (Stripe) for premium plans
- Parent dashboard / progress reports via email
- Profile picture upload to Supabase Storage
- Password change in profile settings
- More question types (fill-blank, short answer, essay)
- More exam content

Priority 3: Polish & Deploy

- Loading skeletons for better UX
 - Error boundaries for crash handling
 - SEO meta tags
 - Deploy to Vercel/Netlify
 - Custom domain setup
-

Testing the App

Test Account

Create a new account with any email you can verify, or use Google OAuth.

Test Flow

1. Visit <http://localhost:5173>
2. Click "Sign Up Free" → Create account
3. Verify email (check inbox)
4. Login → Dashboard shows your name

5. Go to Exams → Browse available exams
 6. Click an exam → See intro page with instructions
 7. Click "Start Exam" → Take the test
 8. Answer questions, use left sidebar to navigate
 9. Flag questions for review with  button
 10. Click "Hint" to see hints
 11. On last question, "Next" becomes "Finish"
 12. Submit → See confetti celebration! 
 13. View results with filtering (All/Correct/Incorrect/Unanswered)
 14. Expand questions to see explanations
-

Prompt to Start New Chat

Copy this to continue development:

I'm continuing development on EduAssess Platform - an educational assessment app for Australian students (Years 2-9) for NAPLAN/ICAS exam preparation.

****Tech Stack:**** React 18, TypeScript, Vite, TailwindCSS, Framer Motion, Zustand, Supabase, React Hook Form + Zod

****GitHub:**** <https://github.com/Vishwas2018/AssessmentPortal>

****Completed:****

- Phase 1: Foundation (project structure, all pages, routing, mock data)
- Phase 2: Real Supabase auth (email/password, Google OAuth, email verification, password reset), profile page, enhanced landing page
- Phase 3: Full exam taking interface with:
 - Left sidebar question navigation (5-column grid)
 - Timer with progress bar
 - Hint system
 - Flag for review
 - Auto-save every 30 seconds
 - "Next" becomes "Finish" on last question
 - Confetti celebration on completion
- Phase 3.5: Enhanced results page with:
 - Two tabs (Summary / Question Review)
 - Filter by: All, Correct, Incorrect, Unanswered
 - Expandable question cards with explanations
 - Your Answer vs Correct Answer comparison

****Database:**** 7 tables in Supabase (user_profiles, exams, questions, exam_attempts, user_progress, achievements, user_achievements) with RLS policies

****Current State:**** The exam taking flow works end-to-end. Users can browse exams, start an exam, answer questions with hints, flag for review, submit, see confetti celebration, and review results with filtering.

****Remaining Tasks:****

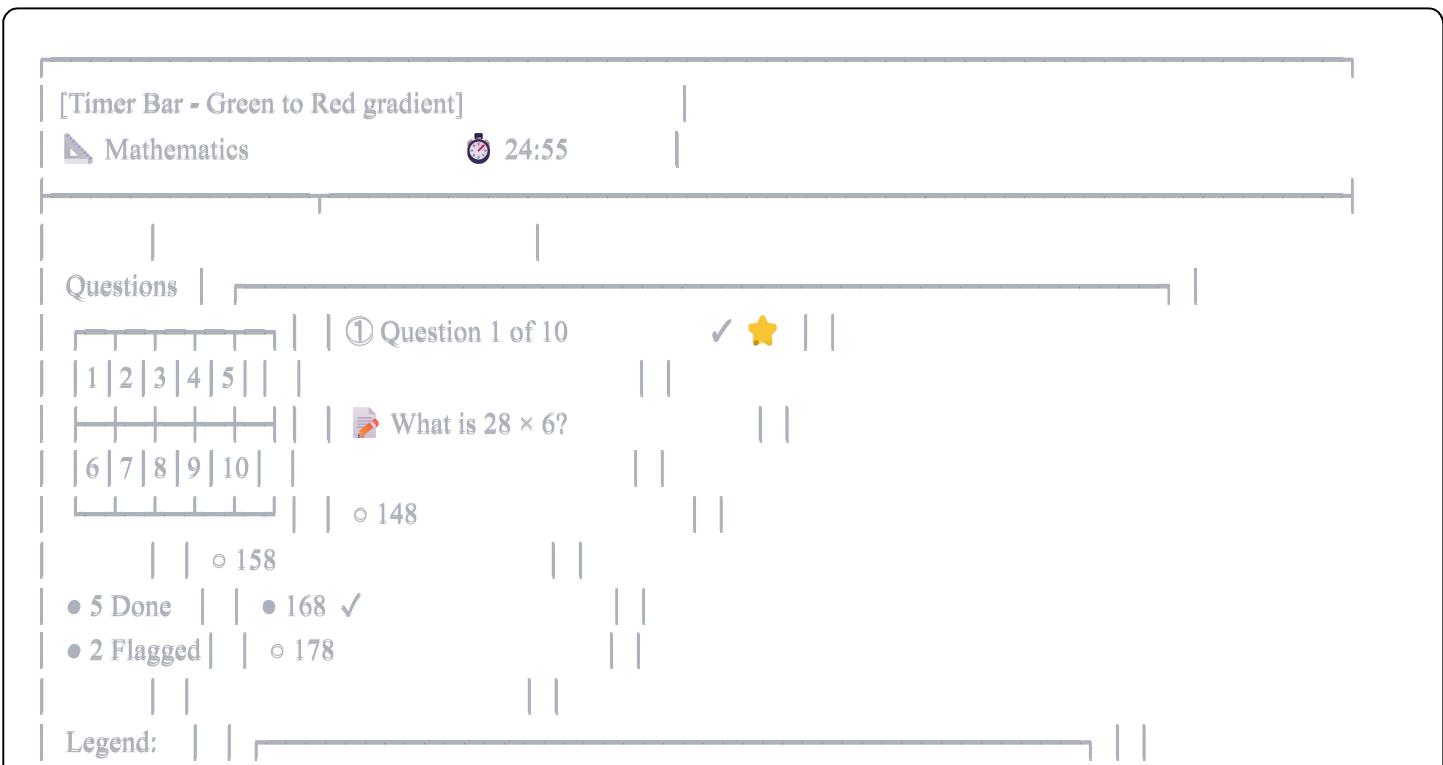
1. Connect Dashboard to real exam_attempts data
2. Add more exam content / questions
3. Payment integration (Stripe) for premium plans
4. Deploy to production

What would you like help with?

📁 Key Files Reference

File	Purpose
src/lib/router.tsx	All route definitions
src/lib/supabase.ts	Supabase client
src/lib/auth.ts	Auth functions (signIn, signUp, etc.)
src/lib/validations.ts	Zod form schemas
src/store/index.ts	Zustand state management
src/pages/ExamPage.tsx	Main exam taking interface (ENHANCED)
src/pages/ExamResultsPage.tsx	Results with filtering (ENHANCED)
src/pages/ExamsPage.tsx	Exam listing with filters
src/pages/HomePage.tsx	Landing page
src/data/constants/index.ts	EXAM_TYPES, YEAR_LEVELS, SUBJECTS arrays

📊 ExamPage.tsx Features Summary



■ Answered | | |  Helpful Hint: $28 \times 6 = 168\dots$

□ Not done | | |

■ Current | | |

● Flagged | | [ Flag] [ Hint] [\leftarrow Previous] [Next \rightarrow] |

 Your Progress: 5 of 10 questions answered [Finish Exam ✓]

End of Master Context Document

Last Updated: January 2025 Status: Phase 3.5 Complete - Enhanced Exam UI Next: Phase 4 - Real Data Integration & Payment