

Edutech Platform – Detailed Development Report

Project Name: Edutech Platform – Online Education Platform
Development Period: November 23, 2025 – November 30, 2025
Tech Stack: React + TypeScript + Vite (Frontend) | Express + TypeScript + Prisma (Backend)
Database: PostgreSQL
Deployment: Docker + Nginx

Table of Contents

- 1. Project Overview
- 2. Detailed Development Phases
- 3. Database Schema Evolution Log
- 4. Technical Implementation Details
- 5. Summary

1. Project Overview

1.1 Background

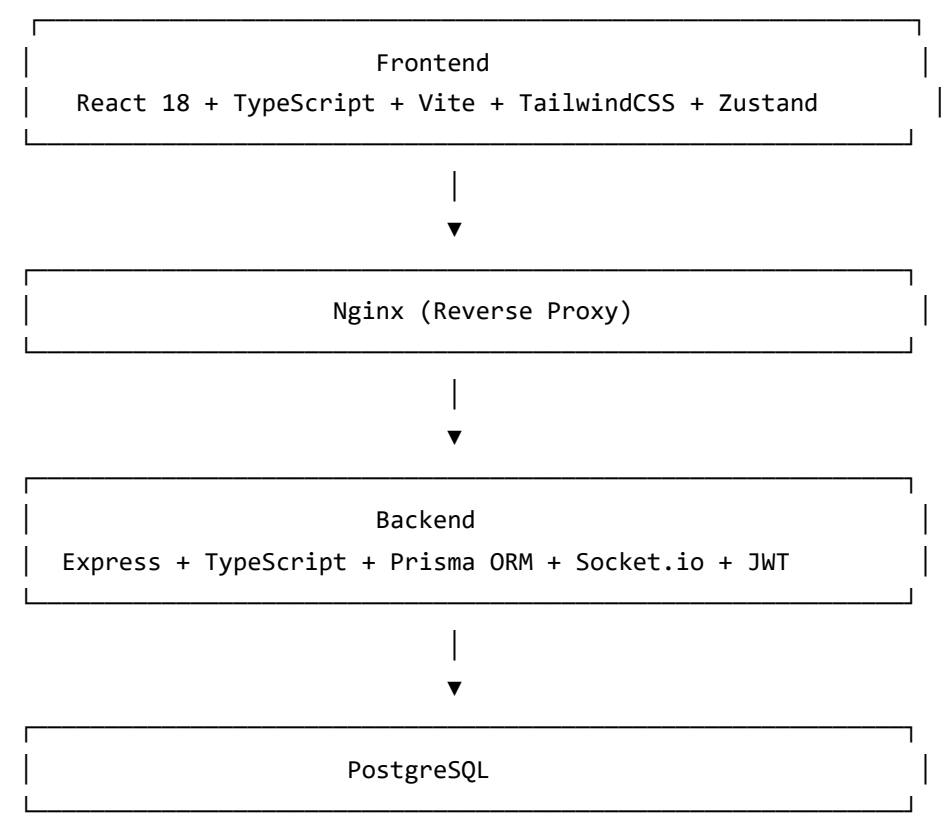
Edutech Platform is a fully featured online education system designed to connect teachers and students. It provides a complete educational ecosystem including course creation, purchasing, learning, reviewing, community interaction, messaging, customer support, and teacher financial management.

1.2 Core Features

Module	Description
User System	Supports three roles: Student, Teacher, Admin
Course System	Supports Live Courses, Recorded Courses, Hybrid Courses
Payment System	Stripe integration, shopping cart, order management

Module	Description
Community	Users can post, comment, like, bookmark, and follow
Real-time Messaging	One-to-one communication between teachers and students
Customer Support	Ticketing system, live support
Wallet System	Teacher earnings tracking and withdrawals
Admin Dashboard	Full management of users, courses, orders, and finance

1.3 System Architecture Diagram



2. Detailed Development Phases

Phase 1: Project Initialization & Base Architecture

Date: Nov 23, 2025
Commit: acd374a
Changes: +18,861 lines (123 files)

Backend Core Modules

Module	Controller	Service	Routes	Functionality
User Authentication	auth.controller.ts	auth.service.ts	auth.routes.ts	Register, Login, Token
Course Management	course.controller.ts	course.service.ts	course.routes.ts	Course CRUD
Teacher Management	teacher.controller.ts	teacher.service.ts	teacher.routes.ts	Teacher Verification
Payment System	payment.controller.ts	payment.service.ts	payment.routes.ts	Stripe Integration
Admin	admin.controller.ts	admin.service.ts	admin.routes.ts	Admin Panel

Frontend Pages

- Home, Login, Register
- Course List, Course Detail
- Student Dashboard, My Courses, Course Learning
- Teacher Dashboard, Create Course, Student Management
- Admin Dashboard, User Management

Middleware

JWT authentication, error handling, rate limiting, file uploads, input validation

Result:

A working full-stack foundation with user login, course browsing, purchasing, and learning.

Phase 2: Bug Fixes & Code Optimizations

Date: Nov 23–24, 2025

Commits: 0998952, 9a3bb0f, fa52b4d, 56ff7be, f819ffd

Commit	Description	Result
0998952	Fixed TypeScript build errors	Project compiles successfully

Commit	Description	Result
9a3bb0f	Cleaned temp files; added course type migration	Cleaner repo
fa52b4d	Improved Admin Dashboard activity feed	Real-time updates
56ff7be	Fixed corrupted Prisma schema	Database functioning normally
f819ffd	Code quality overhaul; strong type safety	90%+ type error reduction

Phase 3: Community Module Development

Date: Nov 25, 2025

Commit: f668ef4

New Tables

- community_tags
- community_posts
- community_comments
- community_post_likes
- community_post_bookmarks
- community_following

New APIs

- POST/GET /api/v1/community/posts – Post CRUD
- POST /api/v1/community/posts/:id/like
- POST /api/v1/community/posts/:id/bookmark
- POST /api/v1/community/users/:id/follow

New Pages

- CommunityHomePage.tsx
- CreatePostPage.tsx
- PostDetailPage.tsx
- CommunityUserProfilePage.tsx

Result:

Users can create posts, comment, like, bookmark, and follow each other.

Phase 4: Customer Support & Refund System

Date: Nov 26, 2025

Commit: ff1ea24

Changed: 80 files

New Enums

- RefundMethod: ORIGINAL_PAYMENT, WALLET, BANK_TRANSFER
- SupportTicketStatus: OPEN, IN_PROGRESS, RESOLVED, CLOSED
- SupportTicketPriority: LOW, MEDIUM, HIGH, URGENT

New Tables

- support_tickets
- support_ticket_messages

New Backend Modules

Module	File	Functionality
Support	support.*	Users submit tickets
Admin Support	support-admin.*	Admin handles tickets
Refund	refund-admin.*	Refund review & processing

New Frontend Components

- CustomerServiceModal.tsx
- CustomerSupportChat.tsx
- RefundModal.tsx
- SupportTicketsManagement.tsx
- RefundsManagement.tsx

Result:

Complete ticketing system + multiple refund methods.

Phase 5: Teacher Profile Completion System

Date: Nov 26, 2025

Commits: cfd48c2, 15f0cac

New Teacher Profile Fields

- teaching_experience
- education_background
- certifications[]
- languages[]
- registration_status

New Components

- TeacherProfileCompletionForm.tsx
- ProfileCompletionPage.tsx
- TeacherPendingPage.tsx
- TeacherApprovedRoute.tsx

Result:

Teachers must complete their profiles and be approved before creating courses.

Phase 6: Real-time Messaging System

Date: Nov 27, 2025

Commit: dd1c76d

New Tables

- message_threads
- messages
- _ThreadParticipants

APIs

- GET /api/v1/messages/threads
- POST /api/v1/messages/threads/:id/messages
- PUT /api/v1/messages/threads/:id/read

New Pages

- MessagesPage.tsx
- TeacherStudentManagementPage.tsx

Result:

Teachers and students can communicate via one-to-one chat.

Phase 7: Enhanced User Management & Commission System

Date: Nov 27, 2025

Commit: 570ed02

User Table Additions

- is_active
- last_login
- login_count

Teacher Profiles Additions

- commission_rate (default: 0.8)

New Admin Features

- BatchOperationsModal.tsx
- UserActionsMenu.tsx
- UserDetailsModal.tsx
- FinancialsManagement.tsx
- RevenueAnalytics.tsx

Result:

Admins can manage users in bulk and configure commission rates.

Phase 8: Teacher Wallet System

Date: Nov 28–29, 2025

Commits: f91f138, c191387

New Tables

- wallets
- wallet_transactions
- payout_methods
- payout_requests

API Endpoints

- GET /api/v1/wallet

- GET /api/v1/wallet/transactions
- POST /api/v1/wallet/payout-methods
- POST /api/v1/wallet/payout-requests

Supported Payout Methods

- BANK_TRANSFER
- GRABPAY
- TOUCH_N_GO
- PAYPAL

Migration Scripts

- migrate-earnings-to-wallet.ts
- sync-historical-earnings.ts

Result:

Teachers can view earnings, add payout methods, and submit withdrawal requests.

Phase 9: System Integration & Optimization

Date: Nov 29–30, 2025

Commits: 98d8686, 7a15d1b

Optimizations

- Full validation with Joi schemas
- Stronger frontend type safety
- Unified error handling
- Skeleton loaders
- Improved responsive layouts

Result:

Significant improvements in system stability and overall UX.

3. Database Schema Evolution Log

Migration File	Date	Description
initial_migration	11-22	Initial schema

Migration File	Date	Description
add_course_type	11-23	Course type enum
fix_relations_final	11-23	Relation fixes
add_community_models	11-24	Community module
add_support_tickets_and_refund	11-25	Support & refund tables
add_extended_teacher_profile	11-25	Additional teacher fields
add_teacher_registration_status	11-25	Teacher approval status
add_profile_submission_payload	11-26	Submission snapshot
add_messaging_models	11-26	Messaging system
add_message_read_map	11-26	Read tracking
add_message_unread_tracking	11-26	Unread status
add_user_management_fields	11-27	User management fields
add_commission_rate	11-27	Commission rate
add_wallet_models	11-27	Wallet system

4. Final Feature Checklist

Feature	User	Teacher	Admin
User Authentication	✓	✓	✓
Course Browsing	✓ Search & Purchase	✓ Create & Manage	✓ Review
Learning	✓	–	–
Community	✓	✓	–
Messaging	✓ Contact Teacher	✓ Contact Students	–

Feature	User	Teacher	Admin
Support Tickets	✓ Submit	–	✓ Handle Tickets
Refunds	✓ Request Refund	–	✓ Review Refunds
Wallet	–	✓ Earnings & Withdrawals	✓ Audit Withdrawals
Analytics	–	✓ Revenue Stats	✓ Platform-wide Stats

5. Summary

Development Duration

- **Total:** 8 days (Nov 23 – Nov 30, 2025)
- **Total Code:** ~25,000+ lines
- **Commits:** 17

Technical Highlights

- Full-stack TypeScript for strict type safety
- Modular Controller-Service-Route architecture
- Prisma ORM with well-structured migrations
- Real-time communication via Socket.io
- Full Stripe payment integration
- Docker-based one-click deployment

Future Improvements

- Redis caching
- Unit test coverage
- CI/CD pipeline
- Performance monitoring