

NB Student Hub

Technical Development Report - Production Ready

New Brunswick High School Student Job Platform

NB Student Hub - Technical Development Report

Project Name: NB Student Hub (New Brunswick High School Student Job Platform)
Development Period: January 2025 Development Environment: Visual Studio Code + Node.js + Git Developer: K Deployment Status: Production Ready (98% Complete)

Chapter 1: Project Overview

Purpose

Development of a comprehensive employment and volunteer matching platform specifically designed for high school students in the New Brunswick region of Canada.

Key Features

- High school student job search and recruitment posting system
- Volunteer opportunity matching service
- Reference letter management system
- File upload and download system

- Real-time community bulletin board
 - Administrator dashboard and monitoring
 - Contact inquiry management system
-

Chapter 2: Core Technology Stack

Frontend Framework

- Next.js 15.3.5 - React-based full-stack framework
- React 19.0.0 - Latest React component-based UI
- TypeScript - Type safety and enhanced development productivity

Styling Technologies

- Tailwind CSS 3.4.1 - Utility-first CSS framework
- clsx + tailwind-merge - Conditional styling optimization
- Lucide React - Consistent icon system

Backend and Database

- Firebase 11.10.0
 - Firestore Database - NoSQL real-time database
 - Firebase Storage - File storage system
 - Firebase Authentication - User authentication management
- Supabase - Backup database system

Form Management and Validation

- React Hook Form 7.60.0 - High-performance form library
- @hookform/resolvers - Schema-based validation

External Service Integration

- MailerSend API 2.6.0 - Email sending service
- Vercel - Serverless deployment platform

Additional Libraries

- jsPDF - PDF document generation
 - html2canvas - Screen capture functionality
 - PapaParse - CSV file processing
-

Chapter 3: System Architecture

Frontend Architecture

```

src/
├─ app/                                # Next.js App Router
|   ├─ page.tsx                        # Main homepage
|   ├─ admin/                          # Administrator dashboard
|   ├─ job-listings/                   # Job posting list/details
|   ├─ job-postings/                   # Job posting registration
|   ├─ job-seekers/                     # Job seeker registration
|   ├─ volunteer-listings/             # Volunteer activity list
|   ├─ volunteer-postings/             # Volunteer activity registration
|   ├─ community/                      # Community bulletin board
|   ├─ contact/                        # Contact inquiries
|   ├─ my-page/                        # User profile page
|   └─ api/                            # API Routes
├─ components/                         # Reusable components
|   ├─ Navigation.tsx                  # Navigation bar
|   ├─ Footer.tsx                      # Footer
|   ├─ FileManager.tsx                 # File management
|   └─ AdminFileUpload.tsx             # Admin file upload
├─ lib/                                # Utilities and services
|   ├─ firebase.ts                     # Firebase configuration
|   ├─ firebase-services.ts            # Firebase service layer
|   ├─ auth-service.ts                 # Authentication service
|   ├─ email-service.ts                # Email service
|   └─ utils.ts                        # Common utilities
└─ types/
    └─ index.ts                        # TypeScript type definitions

```

Database Design (Firestore)

Collections:

```

├─ jobPostings                        # Job postings
├─ jobSeekers                         # Job applications
├─ jobApplications                    # Application history
├─ volunteerPostings                  # Volunteer postings
├─ volunteerApplications              # Volunteer applications
├─ communityPosts                     # Community posts
├─ contacts                           # Contact inquiries
├─ references                         # Reference letters
├─ eventRegistrations                 # Event registrations
├─ uploadedFiles                      # Uploaded file metadata
└─ logs                              # Activity logs

```

Chapter 4: Key Feature Implementation

Authentication and Security System

- Firebase Authentication-based user management
- Administrator privilege separation system
- Email-based user authentication

File Management System

- Category-based file classification: Documents, Resumes, References, Images, Admin Files
- Upload confirmation modal: Confirmation step to prevent errors
- Secure download: Direct Firebase Storage URL method
- File size limitation: 10MB limit for performance optimization

Real-time Data Synchronization

- Real-time updates using Firestore onSnapshot
- Real-time event participant count display
- Community bulletin board real-time synchronization

Complex Search and Filtering

- Multi-filter by job type, location, and category
- Performance optimization with pagination
- Real-time filtering based on search terms

Administrator Dashboard

- Approval and rejection system
- Activity log monitoring
- File management and statistics

- User management functionality
-

Chapter 5: Development Environment and Tools

IDE and Development Tools

- Visual Studio Code - Primary development environment
- Node.js 18+ - JavaScript runtime
- npm - Package management
- Git - Version control

Development Workflow

```
# Development server execution
npm run dev

# Production build
npm run build

# Code quality inspection
npm run lint

# Firebase backup
npm run backup:firebase
```

Code Quality Management

- ESLint - JavaScript/TypeScript linting
 - TypeScript - Static type checking
 - Git Pre-commit Hooks - Code verification before commit
-

Chapter 6: Deployment and Infrastructure

Deployment Platform

- Vercel - Next.js optimized serverless deployment
- Automatic deployment - Auto build/deploy on GitHub push
- CDN - Global edge cache optimization

Domain and SSL

- Domain: nbhischooljobs.com (planned)
- SSL Certificate: Automatically provided by Vercel (Let's Encrypt)
- HTTPS: All communication encrypted

Environment Variable Management

```
NEXT_PUBLIC_FIREBASE_API_KEY=***  
NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=***  
NEXT_PUBLIC_FIREBASE_PROJECT_ID=***  
NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET=***  
MAILERSEND_API_TOKEN=***
```

Chapter 7: Performance Optimization

Next.js Optimization Techniques

- Enhanced routing performance using App Router
- Fast page loading with Static Generation
- Automatic image optimization
- Automatic code splitting

Firebase Optimization

- Query performance improvement with composite indexes
- Index burden reduction with client-side sorting
- Large data processing with pagination

Build Optimization Results

Route (app)	Size	First Load JS
└ ○ /	6.5 kB	281 kB
└ ○ /admin	140 kB	414 kB
└ ○ /job-listings	4.11 kB	282 kB
└ ○ /contact	3.69 kB	278 kB
└ ○ /my-page	4.83 kB	279 kB
Total: 17 pages		
Build time: ~11 seconds		

Chapter 8: Major Problem Solutions

Firebase Index Optimization

Problem: Index errors occurring in composite queries
 Solution: Created 6 composite indexes to optimize all queries

File Deletion Path Issues

Problem: Deletion failure due to Storage path mismatch
 Solution: Built category-based path system

Email Service Stability

Problem: Intermittent MailerSend API errors Solution: Dual safety net system (Firebase + Email)

User Experience Improvement

Problem: Accidental file uploads Solution: Built upload confirmation modal system

Chapter 9: Project Achievements

Feature Completion

- Overall progress: 98% complete
- Core features: 100% implementation complete
- Testing complete: All core functions verified
- Deployment ready: Production Ready

Technical Achievements

- Zero-error build achieved
- TypeScript 100% applied
- Responsive design fully supported
- Real-time features implemented

User-Centered Development

- Canadian localization complete (English UI)
 - High school student-friendly interface
 - Intuitive navigation structure
 - Mobile optimization complete
-

Chapter 10: Future Improvement Plans

Phase 1: Domain Connection (Immediate)

- nbhischooljobs.com domain connection
- DNS setup and SSL certificate application

Phase 2: Advanced Features (Short-term)

- Email notification system enhancement
- Detailed statistics dashboard addition
- Advanced search filter expansion
- PWA (Progressive Web App) application

Phase 3: Extended Features (Medium-term)

- AI-based matching system
- Real-time chat functionality
- Calendar integration system
- Point/badge system

Phase 4: Platform Expansion (Long-term)

- Expansion to other Canadian provinces
 - Online education content integration
 - Mentoring system construction
 - Employment rate analysis dashboard
-

Chapter 11: Development Languages and

Technologies

Primary Programming Languages

- HTML5 - Semantic markup structure
- CSS3 - Modern styling with Flexbox and Grid
- JavaScript ES6+ - Modern JavaScript features
- TypeScript - Static typing for enhanced code quality

Framework and Library Details

- React - Component-based UI development
- Next.js - Full-stack React framework with server-side rendering
- Tailwind CSS - Utility-first CSS framework for rapid styling
- Firebase SDK - Backend as a Service integration

Development Environment Setup

- Visual Studio Code - Primary IDE with extensions:
 - TypeScript support
 - ESLint integration
 - Prettier code formatting
 - Git integration
 - Firebase tools
 - Node.js - JavaScript runtime environment
 - npm - Package manager for dependency management
 - Git - Version control system
-

Chapter 12: Code Structure and Organization

Component Architecture

- Functional components with React Hooks
- TypeScript interfaces for type safety
- Modular component design for reusability
- Separation of concerns between UI and business logic

State Management

- React useState for local component state
- React useEffect for side effects and lifecycle management
- Custom hooks for shared logic
- Firebase real-time listeners for global state

API Integration

- RESTful API design with Next.js API routes
 - Firebase SDK for database operations
 - MailerSend API for email services
 - Error handling and loading states
-

Conclusion

The NB Student Hub project has been successfully developed using modern web technology stack to expand employment and volunteer opportunities for high school students in the New Brunswick region of Canada.

Core Achievements

1. Complete full-stack web application implementation
2. Real-time database-based dynamic functionality
3. Scalable architecture design

- 4. User-centered UX/UI implementation
- 5. Production Ready status achieved

Technical Excellence

- TypeScript 100% application for code stability
- Next.js App Router utilization for latest development patterns
- Scalable infrastructure with Firebase + Vercel combination
- Responsive design supporting all devices

This project is expected to contribute to career development and community participation enhancement for Canadian high school students as a platform optimized for local needs.

Report Date: January 15, 2025 Last Update: Just before final deployment Document Version: v1.0

Development Environment

Visual Studio Code + Node.js + React + TypeScript
Built for Canadian High School Students