**Project Report: Blog Fusion**

**Overview**

**Blog Fusion** is a Next.js-based blogging platform that enables users to create, publish, and manage blog posts with both manual and AI-assisted workflows. It features user authentication, subscription plans, and integration with external APIs for AI content generation and payment processing.

**Key Features**

* **User Authentication:**
  + Email/password and Google sign-in via Firebase Auth.
  + Email verification and password reset support.
* **Blog Creation:**
  + Manual editor for custom blog writing.
  + Automated blog generation using AI (Langflow API).
  + Blog posts stored in Firestore.
* **Subscription Plans:**
  + Free, Medium, and Premium plans with different feature sets and post limits.
  + Stripe integration for paid subscriptions.
  + Modal UI for plan selection and upgrade.
* **UI/UX:**
  + Responsive design with Tailwind CSS.
  + Animated sections using Framer Motion.
  + SweetAlert2 for user feedback and error handling.
  + Contact form with EmailJS integration.
* **Additional Functionality:**
  + Saved blogs and user dashboard.
  + FAQ and About sections.
  + Contact information and business hours.

**Technical Stack**

* **Frontend:**
  + Next.js (React)
  + TypeScript
  + Tailwind CSS
  + Framer Motion
  + SweetAlert2
  + EmailJS
* **Backend/Serverless:**
  + Next.js API routes
  + Node.js (for custom server logic)
  + Python Flask (for plagiarism/AI detection, see plagiarismdetector/)
* **Database & Auth:**
  + Firebase Firestore
  + Firebase Authentication
* **Payments:**
  + Stripe API
* **AI Integration:**
  + Langflow API for blog generation
  + Python-based AI detection and humanization (plagiarismdetector/)

**Project Structure**

* **app/**: Main Next.js application (pages, components, API routes)
* **components/**: Shared React components (e.g., Navbar)
* **plagiarismdetector/**: Python Flask backend for AI/plagiarism detection and content humanization
* **public/**: Static assets (images, favicon)
* **styles/**: Global and editor-specific CSS
* **config files**: ESLint, Tailwind, TypeScript, etc.

**Notable Files**

* [homepage.tsx](vscode-file://vscode-app/c:/Users/PMLS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Main landing page with authentication, subscription, and navigation logic.
* [route.ts](vscode-file://vscode-app/c:/Users/PMLS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): API route for AI blog generation.
* [route.ts](vscode-file://vscode-app/c:/Users/PMLS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): API route for Stripe checkout.
* [firebaseConfig.ts](vscode-file://vscode-app/c:/Users/PMLS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Firebase initialization.
* [plagiarism\_detector.py](vscode-file://vscode-app/c:/Users/PMLS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Python backend for AI detection/humanization.

**Integrations**

* **Firebase:** Auth, Firestore, Storage
* **Stripe:** Subscription payments
* **Langflow:** AI blog generation
* **EmailJS:** Contact form handling
* **Python Flask:** AI/plagiarism detection

 It also includes plagiarism removal and humanization feature for AI-generated blog content. When a user generates a blog post using the AI (via the Langflow API), the content is sent to a custom Python Flask backend ([plagiarismdetector](vscode-file://vscode-app/c:/Users/PMLS/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")). This backend analyzes the text for AI-generated patterns and potential plagiarism, then rewrites or "humanizes" the content to improve originality and reduce detection by plagiarism checkers.

* Workflow:
  1. User requests AI-generated content.
  2. The generated text is sent to the Flask backend.
  3. The backend processes the text, detects AI/plagiarism patterns, and rewrites the content.
  4. The humanized content **is returned and displayed to the user for review and publishing.**

**Summary**

Blog Fusion is a full-featured blogging platform combining modern web technologies, AI-powered content creation, and a flexible subscription model. It is designed for ease of use, scalability, and extensibility.