Project Report

on

Non-Profit QuizBot

Course Name: GenAI

Institution Name: Medicaps University – Datagami Skill Based Course

Submitted by:

Sr no	Student Name	Enrolment Number
1.	Ananya Subramanya Rao	EN22CS301120
2.	Anirudh Vyas	EN22CS301131
3.	Anushka Kashyap	EN22CS301180
4.	Atharv Chaturvedi	EN22CS301228
5.	Harshit Panchal	EN23CS3L1010

Group Name: Group 03D2

Project Number: GAI-15

Industry Mentor Name: Rishabh Sir

University Mentor Name: Dr. Hemlata Patel

Academic Year: 2026

Table of Contents

Problem Statement & Objectives

1. Introduction

The non-profit sector relies heavily on donor relationships, fundraising strategies, and community engagement. The non-profit sector is a knowledge-intensive field where professionals and volunteers working in this space and constantly need to stay updated on their knowledge of best practices from donor communication and fundraising strategies to impact reporting and ethical stewardship. A large amount of this knowledge is buried inside day-to-day donor emails, which are read once and forgotten.

However, traditional learning methods lack personalization, contextual relevance, and immediate feedback. There is a clear gap between the information available in day-to-day donor communications and the ability to convert that information into structured learning experiences. Therefore, there is a need for an AI-driven educational system that provides interactive and intelligent educational tools to help individuals to understand and analyse donor communication in depth.

QuizBot bridges this gap by transforming real-world donor emails into intelligent, AI-powered assessments that test comprehension, reinforce learning, and track progress over time. **QuizBot** is an AI-Powered Educational Assessment Platform for the Non-Profit Sector. It is an AI-powered web application built specifically for the **non-profit sector**. It allows users to paste donor emails as input and system automatically converts them into intelligent multiple-choice quizzes using Google Gemini AI. Users take the quiz, get evaluated, receive detailed explanations for each answer, and track their learning progress over time- turning everyday communication into a personalized learning tool.

2. Problem Statement

Develop an interactive AI-driven educational bot for the Non-Profit domain that engages users in targeted assessments based on donor emails, evaluates their responses using a Large Language Model, and provides deep contextual explanations for incorrect answers, creating a personalized learning loop that bridges knowledge gaps and reinforces industry-specific concepts.

Currently, traditional assessments focus only on scoring responses without offering contextual explanations or personalized learning insights. There is no structured system that converts real donor emails into dynamic assessments to evaluate understanding of non-profit best practices, donor stewardship, fundraising strategies, ethical considerations, and impact measurement.

3. Project Objectives

The key objectives of the project QuizBot – Non-Profit Quiz/Tutor Bot is to develop an interactive AI-powered educational assessment platform that:

- Accepts donor email content as User input and automatically generates 3 to 10 contextual MCQ-based quiz questions based on email content using LLM API.
- Uses Google Gemini 2.5 Flash (LLM) to generate quiz questions, evaluate user answers, and produce personalized feedback and explanations for each answer to reinforce learning.
- Store donor email embeddings in a Vector Database (ChromaDB) using Sentence Transformers for semantic search and retrieval.
- Provide detailed AI-generated explanations for both correct and incorrect answers after every quiz.
- Track user quiz history, scores, and learning progress through Firebase Realtime Database.
- Display an analytics dashboard showing performance trends, accuracy rates, and achievements over time.
- Implement secure user authentication using Firebase Authentication with a dual-token session model (Firebase ID Token + backend JWT).

The overall goal is to create a personalized learning loop that bridges knowledge gaps in the non-profit domain.

4. Scope of the Project

The scope of QuizBot — Non-Profit Quiz/Tutor Bot covers the following areas:

A. Functional Scope

- Secure User registration and login using Firebase Authentication.
- Donor email input interface where users can paste email content and select the number of quiz questions (3, 5, 7, or 10).
- Take donor email and generates MCQ-based quizzes using Google Gemini 2.5 Flash LLM API.
- Per-question answer evaluation using LLM with detailed explanations for both correct and incorrect answers.
- Overall performance summary generated by AI after each quiz attempt.
- Score calculation and result display through a dedicated Result Card.
- Quiz history storage per user in Firebase Realtime Database.
- User analytics dashboard showing accuracy rates, total quizzes attempted, and performance trends over time.

B. Technical Scope

- **Frontend:** React 19 with Vite, Tailwind CSS, Zustand (state management), Axios (HTTP client), Framer Motion (animations).
- **Backend:** Python with FastAPI, SQLAlchemy ORM, SQLite database.
- **AI Layer:** Google Gemini 2.5 Flash (LLM) for quiz generation, answer evaluation, and summary generation.
- **Vector Database:** ChromaDB with Sentence Transformers (all-MiniLM-L6-v2, 384-dimensional embeddings) for storing and retrieving donor email context.
- **Authentication:** Firebase Authentication (email/password) with dual-token model - Firebase ID Token + backend JWT (python-jose).
- **Cloud Database:** Firebase Realtime Database for quiz history and user progress storage.
- **Security:** JWT middleware, CORS policy, Pydantic V2 input validation, environment variable-based secret management

The project is currently scoped as a v1.0 web application running in a local environment. Features such as mobile application support, multi-language interface, team or organization-level accounts, and a dedicated caching layer are not included in the current version and are planned for future development.

Proposed Solution

1. Key features

The Non-Profit QuizBot system includes the following key features:

- **Donor Email-Based Assessment** - Uses real donor emails as the primary knowledge input for quiz generation, making assessments contextually relevant to the non-profit domain.
- **AI-Powered Quiz Generation** - Automatically generates 3 to 10 MCQ questions from donor email content using Google Gemini 2.5 Flash LLM API.
- **Contextual Question Formation** - Questions are derived directly from the content, intent, and themes of the donor email provided.
- **Intelligent Answer Evaluation** - Each user response is evaluated by Gemini LLM with a detailed explanation for both correct and incorrect answers, going beyond basic rule-based scoring.
- **AI Performance Summary** - An overall performance summary is generated by Gemini after every quiz attempt.
- **Vector Semantic Search** - Donor email content is embedded using Sentence Transformers and stored in ChromaDB for semantic context retrieval.
- **Progress Tracking & Analytics** - Quiz history, scores, and accuracy rates are stored per user and displayed on a visual analytics dashboard showing performance trends over time.
- **User Authentication & Secure Access** - Secure registration and login via Firebase Authentication using a dual-token model.
- **Responsive Web Interface** - A modern, animated Single Page Application built with React 19 and Tailwind CSS.

2. Overall Architecture and Workflow

QuizBot follows a **3-Tier Client-Server Architecture** with clear separation between the Presentation Layer, Business Logic Layer, and Data Layer.

A. Overall Architecture

1. **Frontend Layer**
    - Built using React 19 with Vite as the build tool
    - Accepts donor email input and number of questions selection
    - Displays generated quiz questions via QuizInterface component
    - Collects user responses and triggers quiz submission
    - Shows per-question explanations, score, and AI performance summary
    - Displays quiz history and performance analytics on the Dashboard

2. **Backend Layer**
    - Developed using FastAPI (Python)
    - Handles all REST API requests from the frontend
    - Manages JWT middleware, CORS policy, and Pydantic V2 input validation
    - Communicates with Gemini LLM API for quiz generation and evaluation
    - Manages quiz generation logic, scoring calculation, and result building
    - Reads and writes data to SQLite and Firebase Realtime Database
3. **AI Layer**
    - Google Gemini 2.5 Flash LLM API for quiz question generation, answer evaluation, and performance summary.
    - Sentence Transformers (all-MiniLM-L6-v2) for converting donor email text into 384-dimensional vectors.
    - ChromaDB as the vector store for storing and retrieving donor email embeddings.
4. **Data Layer**
    - **SQLite** (via SQLAlchemy ORM) - stores users, quizzes, questions, quiz results, and user progress locally.
    - **Firebase Realtime Database** - stores quiz history and performance analytics per user in the cloud.
    - **ChromaDB** - stores donor email embeddings for semantic context retrieval.

B. System Workflow

This creates a continuous personalized learning cycle — from donor email input to AI-evaluated assessment to tracked progress.

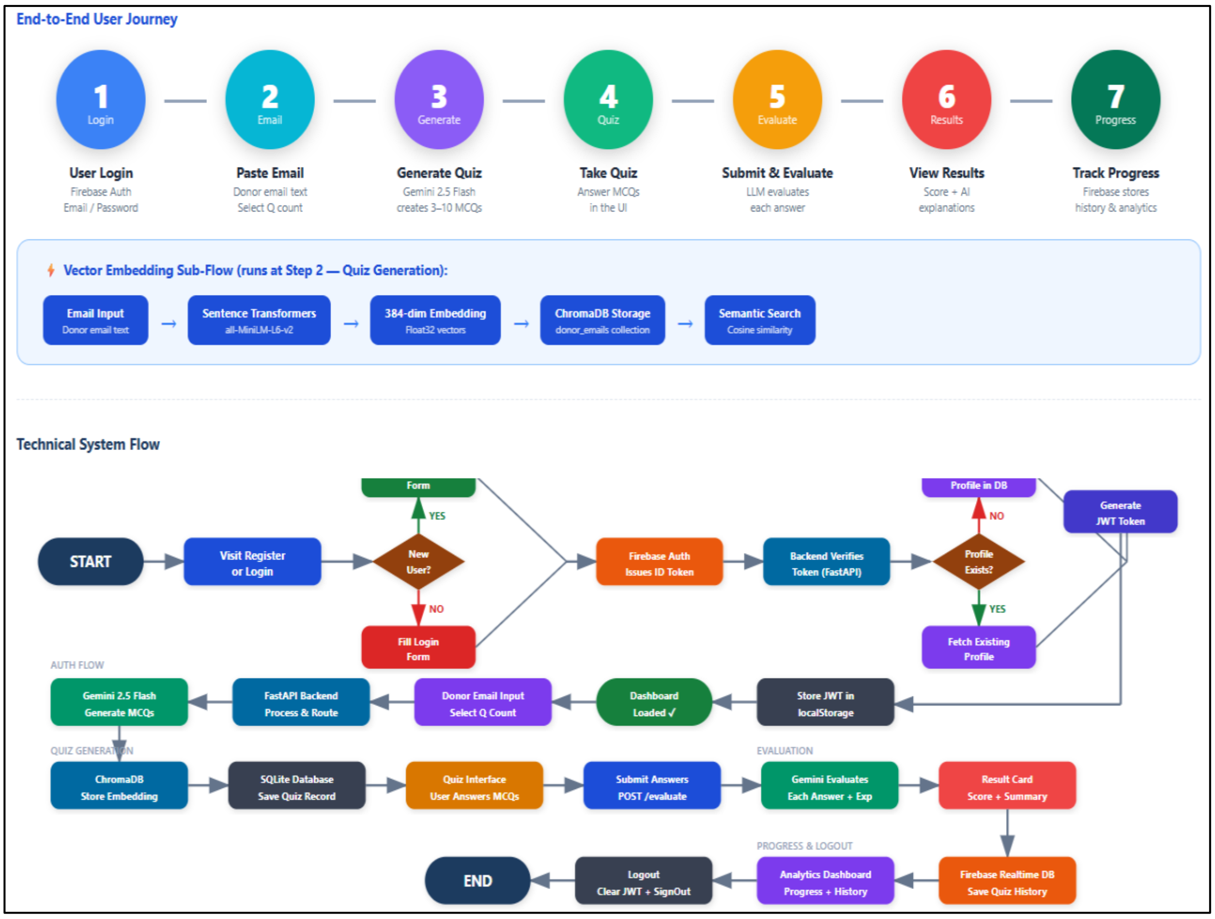
Step 1 — User Authentication The user registers or logs in via Firebase Authentication. A Firebase ID Token is sent to the FastAPI backend, which verifies it using the Firebase Admin SDK and issues a short-lived JWT (1 hour) for all subsequent API calls.

Step 2 — Quiz Generation The user pastes donor email content and selects the number of questions. The backend encodes the email into a 384-dimensional vector using Sentence Transformers and stores it in ChromaDB. The email is then sent to Google Gemini 2.5 Flash with a structured prompt, which returns MCQ questions in JSON format. The quiz is saved to SQLite via SQLAlchemy ORM and returned to the frontend.

Step 3 — Quiz Evaluation The user answers all questions and submits the quiz. The backend loops through each answer, sends it to Gemini for evaluation, and collects the result (correct/incorrect + explanation) for each question. Gemini then generates an overall performance summary. The final score and results are saved to SQLite and written to Firebase Realtime Database.

Step 4 — Progress & Analytics The user can view their quiz history, scores, and performance trends on the analytics dashboard, which reads data from Firebase Realtime Database via the FastAPI backend.

Then, the user can log out at any time, which clears the JWT from localStorage, resets the Zustand store, and calls Firebase signOut() to terminate the session securely.



3. Tools & Technologies Used

QuizBot is built using a modern, carefully selected technology stack that covers every layer of the application — from the user interface to the AI engine to cloud-based data storage. Each tool was chosen based on its specific strengths, compatibility with the project requirements, and suitability for building a scalable, secure, and AI-powered educational platform.

The frontend uses React-based technologies for a responsive user experience, the backend uses Python with FastAPI for high-performance API handling, and the AI layer integrates Google Gemini 2.5 Flash along with ChromaDB for intelligent quiz generation and semantic retrieval.

Category	Tool / Technology	Purpose
Frontend	React 19 + Vite	SPA framework and build tool
	Tailwind CSS	UI styling
	Zustand	Global state management (auth + quiz state)
	Axios	HTTP client with JWT interceptor
	Framer Motion	Page and component animations
	Recharts	Analytics charts and graphs
Backend	Python + FastAPI	REST API backend
	SQLAlchemy ORM	Database models and query handling
	SQLite	Local structured database
	python-jose	JWT token generation and validation
	Pydantic V2	Request/response schema validation
AI / ML	Google Gemini 2.5 Flash	Quiz generation, evaluation, and summary
	Sentence Transformers (all-MiniLM-L6-v2)	Donor email text embedding (384-dim)
	ChromaDB	Vector store for semantic email retrieval
Authentication	Firebase Authentication	Email/password login and registration
	Firebase Admin SDK	Server-side token verification
Database	Firebase Realtime Database	Cloud storage for quiz history and user progress
Security	JWT Middleware	Protects all non-public API endpoints
	Environment Variables (.env)	Secure storage of all API keys and credentials
Version Control	Git + GitHub	Version control and collaboration

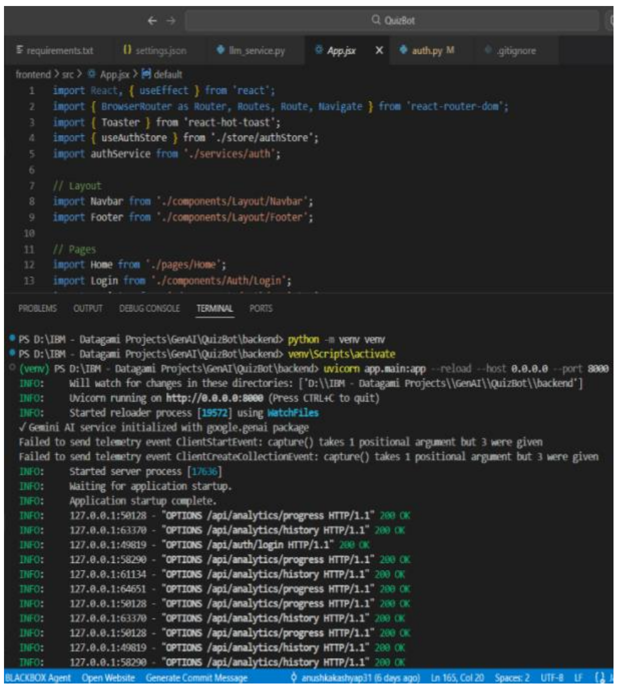
Together, these technologies form a cohesive and well-integrated stack that enables QuizBot to deliver intelligent, AI-evaluated assessments in a secure and responsive web environment. The combination of a stateless FastAPI backend, a dual-token authentication model, and a multi-database strategy ensures the system is reliable, maintainable, and ready for future scalability.

Results & Output

1. Screenshots / outputs

Figure 1: Frontend Development Server (Vite) - The QuizBot frontend development server running via Vite v7.3.1 on localhost:5173. The terminal confirms the React application is successfully compiled and running.

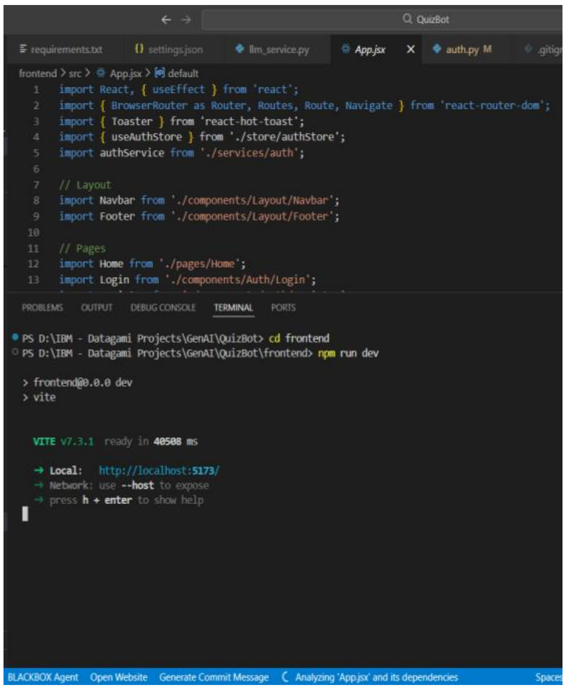
Figure 2: Backend Server & API Logs (Uvicorn + FastAPI) - The FastAPI backend server running via Uvicorn on http://0.0.0.0:8000 with Python 3.11. The terminal confirms successful initialization — including the Gemini AI service and shows live API request logs with HTTP 200 OK responses for endpoints.



```
import React, { useEffect } from 'react';
import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom';
import { Toaster } from 'react-hot-toast';
import { useAuthStore } from './store/authStore';
import authService from './services/auth';

// Layout
import Navbar from './components/Layout/Navbar';
import Footer from './components/Layout/Footer';

// Pages
import Home from './pages/Home';
import Login from './components/Auth/Login';
```

```
PS D:\IBM - Datagami Projects\GenAI\QuizBot\backend> python -m venv venv
PS D:\IBM - Datagami Projects\GenAI\QuizBot\backend> venv\Scripts\activate
(venv) PS D:\IBM - Datagami Projects\GenAI\QuizBot\backend> uvicorn app.main:app --reload --host 0.0.0.0 --port 8000
INFO:     Will watch for changes in these directories: ['D:\\IBM - Datagami Projects\\GenAI\\QuizBot\\backend']
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [19572] using WatchFiles
✓ Gemini AI service initialized with google.genai package
Failed to send telemetry event ClientStartEvent: capture() takes 1 positional argument but 3 were given
Failed to send telemetry event ClientCreateCollectionEvent: capture() takes 1 positional argument but 3 were given
INFO:     Started server process [17636]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:50128 - "OPTIONS /api/analytics/progress HTTP/1.1" 200 OK
INFO:     127.0.0.1:63370 - "OPTIONS /api/analytics/history HTTP/1.1" 200 OK
INFO:     127.0.0.1:49819 - "OPTIONS /api/auth/login HTTP/1.1" 200 OK
INFO:     127.0.0.1:58290 - "OPTIONS /api/analytics/progress HTTP/1.1" 200 OK
INFO:     127.0.0.1:61134 - "OPTIONS /api/analytics/history HTTP/1.1" 200 OK
INFO:     127.0.0.1:64651 - "OPTIONS /api/analytics/progress HTTP/1.1" 200 OK
INFO:     127.0.0.1:50128 - "OPTIONS /api/analytics/progress HTTP/1.1" 200 OK
INFO:     127.0.0.1:63370 - "OPTIONS /api/analytics/history HTTP/1.1" 200 OK
INFO:     127.0.0.1:50128 - "OPTIONS /api/analytics/progress HTTP/1.1" 200 OK
INFO:     127.0.0.1:49819 - "OPTIONS /api/analytics/history HTTP/1.1" 200 OK
INFO:     127.0.0.1:58290 - "OPTIONS /api/analytics/history HTTP/1.1" 200 OK
```

Figure 1



```
import React, { useEffect } from 'react';
import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom';
import { Toaster } from 'react-hot-toast';
import { useAuthStore } from './store/authStore';
import authService from './services/auth';

// Layout
import Navbar from './components/Layout/Navbar';
import Footer from './components/Layout/Footer';

// Pages
import Home from './pages/Home';
import Login from './components/Auth/Login';
```

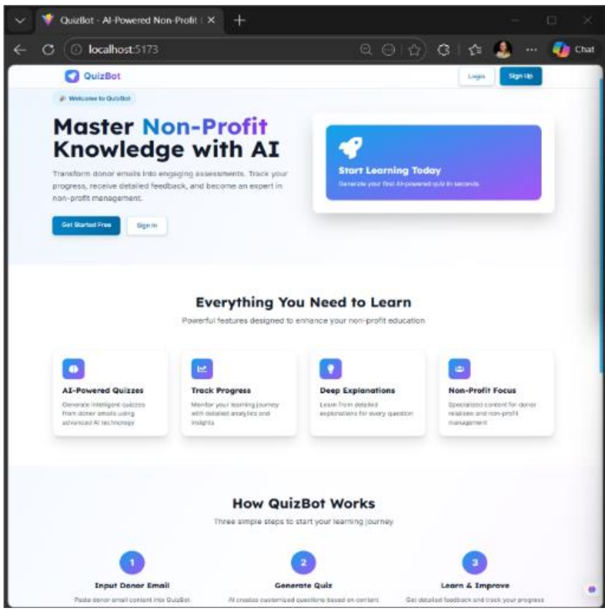
```
PS D:\IBM - Datagami Projects\GenAI\QuizBot> cd frontend
PS D:\IBM - Datagami Projects\GenAI\QuizBot\frontend> npm run dev

> frontend@0.0.0 dev
> vite


  VITE v7.3.1  ready in 40508 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Figure 2

Figure 3: Home / Landing Page - The QuizBot landing page displaying the product introduction, key features overview, and the main call-to-action button to get started.

Figure 4: User Registration Page - The registration interface where new users can create an account using their name, email address, and password via Firebase Authentication.

Figure 5: User Login Page - The login page where registered users authenticate using their email and password. Firebase handles credential verification and issues an ID Token.



Master Non-Profit Knowledge with AI

Transform donor emails into engaging assessments. Track your progress, receive detailed feedback, and become an expert in non-profit management.

Everything You Need to Learn

Powerful features designed to enhance your non-profit education

How QuizBot Works

Three simple steps to start your learning journey

Figure 3

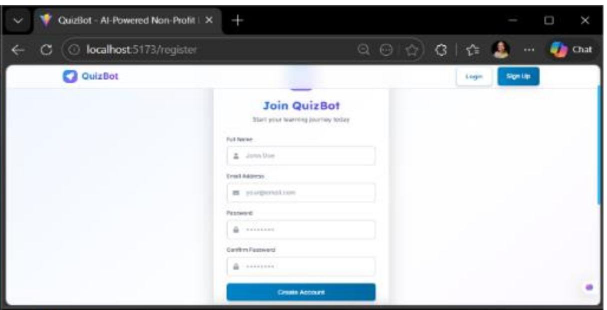


Figure 4

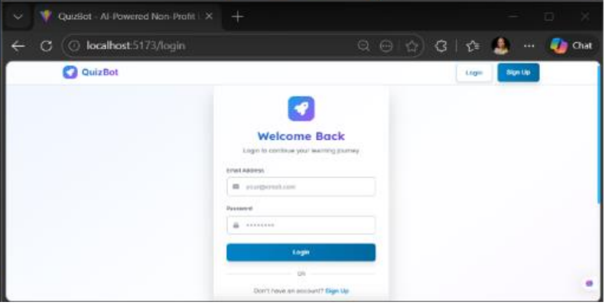

QuizBot - AI-Powered Non-Profit
localhost:5173/login
QuizBot
Login
Sign Up
Welcome Back
Login to continue your learning journey
Email Address
your@email.com
Password
Login
OR
Don't have an account? Sign Up

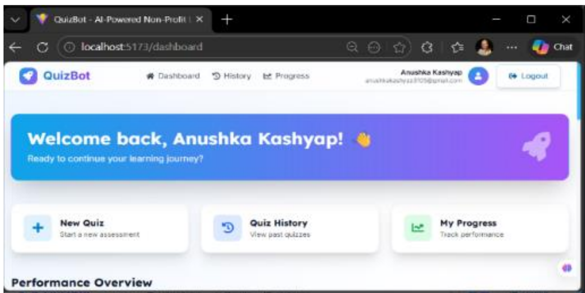
Figure 5

Figure 6: Dashboard - The user dashboard displaying total quizzes attempted, average score, and overall accuracy rate as summary stat cards.

Figure 7: Quiz Page: Email Input The donor email input interface where users paste email content and select the number of MCQ questions before generating a quiz.

Figure 8: Quiz Page: Quiz Interface The quiz interface displaying AI-generated MCQ questions with 4 answer options per question. Users select their answers before submitting.

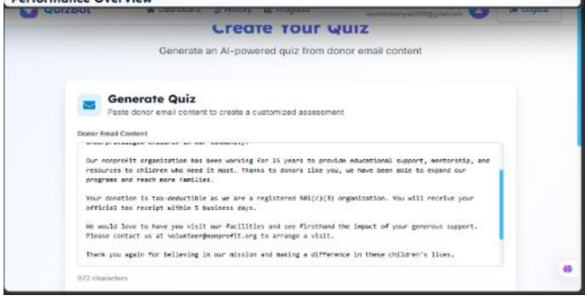
Figure 6



Welcome back, Anushka Kashyap!

Ready to continue your learning journey?

Performance Overview



Create Your Quiz

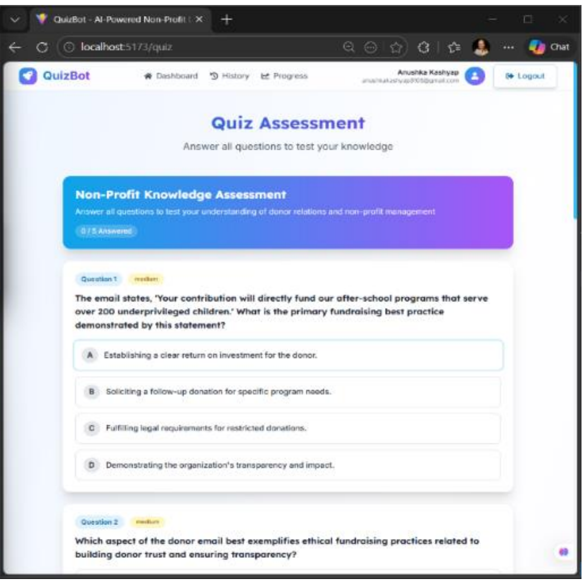
Generate an AI-powered quiz from donor email content

Generate Quiz

Paste donor email content to create a customized assessment

```
Our nonprofit organization has been working for 15 years to provide educational support, mentorship, and
resources to children who need it most. Thanks to donors like you, we have been able to expand our
programs and reach more families.

Your donation is tax-deductible as we are a registered 501(c)(3) organization. You will receive your
official tax receipt within 5 business days.

We would love to have you visit our facilities and see firsthand the impact of your generous support.
Please contact us at volunteer@nonprofit.org to arrange a visit.

Thank you again for believing in our mission and making a difference in these children's lives.
```

Figure 7



Quiz Assessment

Answer all questions to test your knowledge

Non-Profit Knowledge Assessment

Answer all questions to test your understanding of donor relations and non-profit management

The email states, 'Your contribution will directly fund our after-school programs that serve over 200 underprivileged children.' What is the primary fundraising best practice demonstrated by this statement?

A Establishing a clear return on investment for the donor.

B Soliciting a follow-up donation for specific program needs.

C Fulfilling legal requirements for restricted donations.

D Demonstrating the organization's transparency and impact.

Which aspect of the donor email best exemplifies ethical fundraising practices related to building donor trust and ensuring transparency?

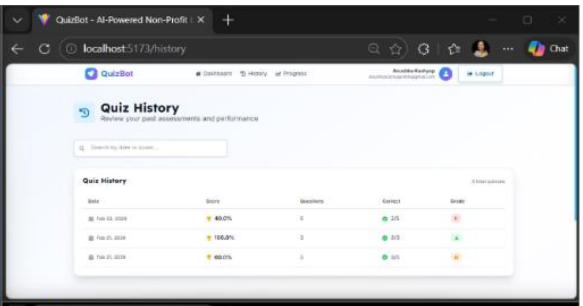
Figure 8

Figure 9: Result Card - The result screen showing the final score, per-question correct/incorrect breakdown, AI-generated explanations for each answer, and an overall performance summary.

Figure 10: History Page - The quiz history page showing a paginated table of all past quiz attempts with scores, dates, and individual result links.

Figure 11: Progress Page - The analytics and progress page showing performance trend charts built with Recharts, tracking user improvement over time.

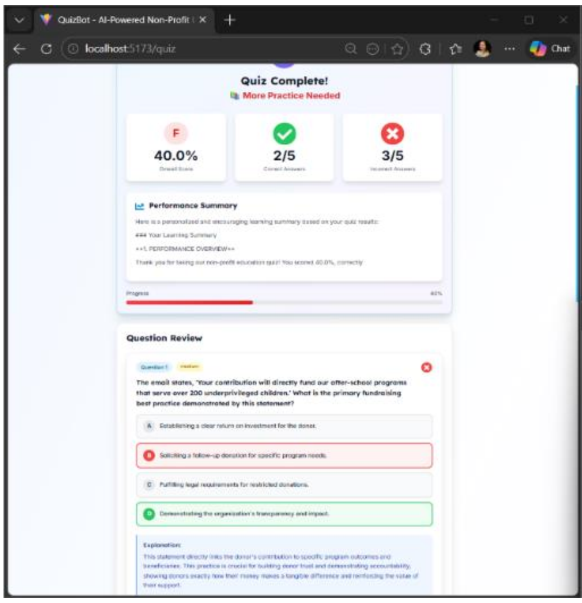
Figure 10



Quiz History
Review your past assessments and performance

Quiz History

Date	Score	Questions	Correct	Grade
[illegible]	40.0%	[illegible]	2/5	[illegible]
[illegible]	100.0%	[illegible]	3/3	A
[illegible]	60.0%	[illegible]	[illegible]	[illegible]



Quiz Complete!
More Practice Needed

40.0%
2/5
3/5

Performance Summary

Your Learning Summary

1. PERFORMANCE OVERVIEW

Progress

Question Review

The email states, 'Your contribution will directly fund our after-school programs that serve over 200 underprivileged children.' What is the primary fundraising best practice demonstrated by this statement?

A Establishing a clear return on investment for the donor.

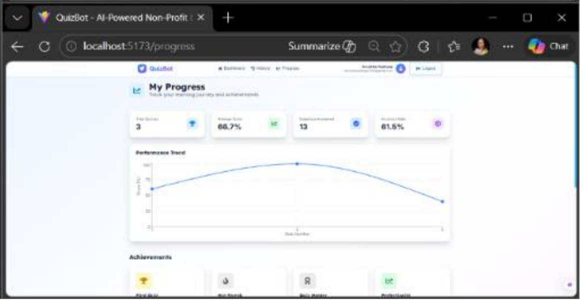
B Soliciting a follow-up donation for specific program needs.

C Fulfilling legal requirements for restricted donations.

D Demonstrating the organization's transparency and impact.

Explanation:

Figure 9



My Progress

3
66.7%
13
81.5%

Performance Trend

Achievements

Figure 11

2. **Reports / Dashboards / Models**

A. Analytics Dashboard - The QuizBot analytics dashboard provides each user with a personalized performance overview including:

- Total number of quizzes attempted.
- Overall accuracy rate (correct answers / total answers).
- Average score across all quiz attempts.
- Performance trend line chart (built with Recharts) showing improvement over time
- Chronological quiz history table with scores and dates.

B. Quiz Result Report - After every quiz submission, the system generates a detailed result report containing:

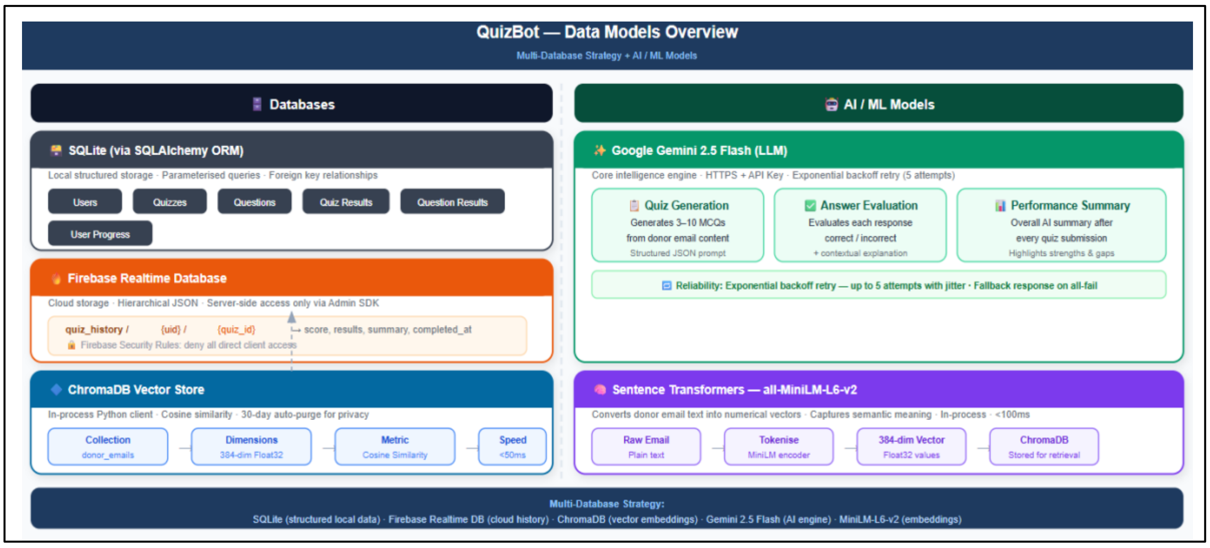
- Final score (number of correct answers out of total questions).
- Per-question breakdown showing the selected answer, correct answer, and AI-generated explanation for each question.

- An overall AI performance summary generated by Google Gemini 2.5 Flash, highlighting strong areas and areas needing improvement.
- A "Try Another Quiz" option that resets the session and navigates back to the quiz page.

C. Data Models - The system uses a multi-database strategy combined with AI and ML models:

Databases:

- **SQLite (via SQLAlchemy ORM)** — stores Users, Quizzes, Questions, Quiz Results, Question Results, and User Progress locally. All tables are connected through foreign key relationships and accessed via parameterised queries to prevent SQL injection.
- **Firebase Realtime Database** — stores quiz history and performance analytics per user in a hierarchical JSON structure under quiz_history/{uid}/{quiz_id}. Accessible only through the FastAPI backend via Firebase Admin SDK — direct client access is denied by Firebase Security Rules.
- **ChromaDB Vector Store** — stores 384-dimensional donor email embeddings in the donor_emails collection using cosine similarity metric for semantic retrieval. Donor email embeddings are automatically purged after 30 days for privacy.


QuizBot — Data Models Overview
Multi-Database Strategy + AI / ML Models
Databases
SQLite (via SQLAlchemy ORM)
Local structured storage · Parameterised queries · Foreign key relationships
Users
Quizzes
Questions
Quiz Results
Question Results
User Progress
Firebase Realtime Database
Cloud storage · Hierarchical JSON · Server-side access only via Admin SDK
quiz_history / {uid} / {quiz_id}
score, results, summary, completed_at
Firebase Security Rules: deny all direct client access
ChromaDB Vector Store
In-process Python client · Cosine similarity · 30-day auto-purge for privacy
Collection donor_emails
Dimensions 384-dim Float32
Metric Cosine Similarity
Speed <50ms
AI / ML Models
Google Gemini 2.5 Flash (LLM)
Core intelligence engine · HTTPS + API Key · Exponential backoff retry (5 attempts)
Quiz Generation
Generates 3–10 MCQs from donor email content
Structured JSON prompt
Answer Evaluation
Evaluates each response correct / incorrect
+ contextual explanation
Performance Summary
Overall AI summary after every quiz submission
Highlights strengths & gaps
Reliability: Exponential backoff retry — up to 5 attempts with jitter · Fallback response on all-fail
Sentence Transformers — all-MiniLM-L6-v2
Converts donor email text into numerical vectors · Captures semantic meaning · In-process · <100ms
Raw Email Plain text
Tokenise MiniLM encoder
384-dim Vector Float32 values
ChromaDB Stored for retrieval
Multi-Database Strategy:
SQLite (structured local data) · Firebase Realtime DB (cloud history) · ChromaDB (vector embeddings) · Gemini 2.5 Flash (AI engine) · MiniLM-L6-v2 (embeddings)

AI / ML Models:

- **Google Gemini 2.5 Flash (LLM)** — the core intelligence engine of the system. Used for three distinct tasks:
  - *Quiz Generation* — generates 3 to 10 contextual MCQ questions from donor email content using a structured prompt.
  - *Answer Evaluation* — evaluates each user response and returns whether it is correct along with a detailed contextual explanation.

- *Performance Summary* — generates an overall AI summary of the user's quiz performance after submission.
- Includes exponential backoff retry logic (up to 5 attempts) with a fallback response on all-fail.

- **Sentence Transformers — all-MiniLM-L6-v2** — converts donor email text into 384-dimensional numerical vectors. These embeddings capture the semantic meaning of the email content and are stored in ChromaDB for context-aware quiz generation.

3. Key Outcomes

- Successfully built a fully functional AI-powered educational assessment platform for the non-profit sector.
- Google Gemini 2.5 Flash accurately generates contextual MCQ questions from real donor email content.
- LLM-based answer evaluation provides meaningful, explanation-rich feedback beyond simple correct/incorrect scoring.
- ChromaDB vector storage enables semantic understanding and retrieval of donor email context.
- Dual-token authentication (Firebase ID Token + JWT) provides secure, stateless session management.
- User quiz history and progress are reliably persisted across sessions via Firebase Realtime Database and SQLite.
- The analytics dashboard gives users a clear view of their learning progress over time.
- Exponential backoff retry logic ensures system reliability even under Gemini API rate limits.

Conclusion

QuizBot - Non-Profit Quiz/Tutor Bot is an AI-powered educational assessment platform designed and developed specifically for the non-profit sector as part of the Datagami Skill Based Course at Medicaps University. The system successfully transforms real-world donor email content into intelligent, contextual multiple-choice quizzes using Google Gemini 2.5 Flash. Each quiz is evaluated by the same LLM, which provides detailed explanations for both correct and incorrect answers, generates an overall performance summary after every attempt, and tracks learning progress over time through a secure and responsive web application. This moves the platform beyond simple scoring and into genuine, explanation-driven learning, creating a personalized learning loop for non-profit professionals and volunteers.

From an architectural standpoint, the project implements a clean 3-tier Client-Server Architecture. The frontend is built with React 19, Vite, Tailwind CSS, and Zustand for state management. The backend is developed using FastAPI with SQLAlchemy ORM and SQLite for structured local storage. The AI layer integrates Google Gemini 2.5 Flash for content generation and evaluation, Sentence Transformers for semantic email embedding, and ChromaDB as the vector store. Firebase Authentication handles secure user login using a dual-token model, and Firebase Realtime Database stores quiz history and analytics in the cloud. The system demonstrates how Generative AI can be applied meaningfully in an educational context — not just as a chatbot, but as a structured assessment engine that understands content, evaluates responses, and guides users.

Key learnings from this project include:

- Hands-on experience integrating a Large Language Model (Google Gemini 2.5 Flash) into a real-world application for dynamic content generation and evaluation
- Understanding of vector embeddings and semantic search using Sentence Transformers and ChromaDB.
- Practical implementation of a dual-token authentication model using Firebase Authentication and JWT.
- Building a secure REST API with FastAPI including middleware, input validation, and error handling.
- Managing a multi-database strategy combining SQLite, Firebase Realtime Database, and ChromaDB for different data needs.
- Designing and building a responsive Single Page Application with React 19, Zustand state management, and Tailwind CSS.

Overall, QuizBot successfully achieves its goal of creating an intelligent, personalized, and domain-specific learning tool that bridges the gap between day-to-day donor communication and structured educational assessment.

Future Scope & Enhancements

While QuizBot v1.0 is fully functional as a local web application, several enhancements are planned for future versions:

- **Cloud Deployment** - Deploy the FastAPI backend and React frontend to a cloud platform (such as AWS, Google Cloud, or Vercel) to make QuizBot accessible publicly without local setup.
- **Mobile Application** - Develop a mobile version of QuizBot for Android and iOS to allow users to take quizzes on the go.
- **Multi-Language Support** - Extend the platform to support donor emails and quiz generation in multiple languages, broadening accessibility for international non-profit organizations.
- **Student Learning Mode** - Extend QuizBot beyond the non-profit domain to support students by accepting lecture notes, textbook chapters, and study material as input, enabling AI-generated revision quizzes for exam preparation across any academic subject.
- **Rate Limiting & API Throttling** - Implement request rate limiting on the FastAPI backend to prevent abuse and manage Gemini API usage costs.
- **Organization / Team Accounts** - Add support for team-level accounts where non-profit organizations can manage multiple users, track collective progress, and compare performance across team members.
- **Support for Multiple Document Types** - Extend input support beyond donor emails to include grant proposals, impact reports, newsletters, and fundraising documents.
- **Migration to PostgreSQL** - Migrate from SQLite to PostgreSQL for production-grade scalability when concurrent users increase beyond the current local scope.