CollabQuest

Complete Project Documentation

Academic Year: 2025–2026

Contents

1 Project Overview

CollabQuest is a full-stack web application designed to connect developers based on their skills and help them find compatible collaborators for coding projects.

1.1 Core Purpose

CollabQuest bridges the gap between talented developers by matching them using AI-based compatibility scoring.

2 Technology Stack

2.1 Frontend

React, Vite, React Router, Axios, Tailwind CSS, Context API

2.2 Backend

FastAPI, Uvicorn, Firebase Admin, Google Generative AI, Sentence Transformers

2.3 Database

Firebase Realtime Database (NoSQL)

3 Project Architecture

3.1 Architecture Overview

Three-tier architecture consisting of frontend, backend API, and database layer.

3.2 Data Flow

User → Frontend → Backend → Database → Response → UI Update

4 Backend Documentation

4.1 Backend Structure

main.py, database.py, quiz.py, c_score.py, cleanjson.py

4.2 Core Modules

4.2.1 main.py

Handles routing, authentication, and CORS.

4.2.2 database.py

All Firebase database operations.

4.2.3 quiz.py

AI-based problem generation and evaluation.

4.2.4 c_score.py

Compatibility scoring using NLP embeddings.

4.2.5 cleanjson.py

Cleans AI-generated JSON responses.

5 Frontend Documentation

5.1 Frontend Structure

React + Vite based component-driven architecture.

5.2 Key Components

Home, Login, Register, Profile, Users, Partners, Problems, Verification

5.3 State Management

Global state handled using Context API.

6 Database Schema

6.1 Users Collection

Stores user profile, skills, verification status, teams.

6.2 Teams Collection

Stores team members and projects.

6.3 Problems Collection

Stores coding problems for verification.

7 API Endpoints

7.1 Authentication APIs

POST /signup
POST /login

7.2 Compatibility APIs

POST /compatibility-score
GET /compatibility/compare

7.3 Verification APIs

POST /verification/generate-problem
POST /verification/submit-solution

8 Setup & Installation

8.1 Backend Setup

Install dependencies, configure Firebase, run Uvicorn server.

8.2 Frontend Setup

Install npm packages and run Vite dev server.

9 Development Workflow

9.1 Local Development

Backend and frontend run on separate terminals.

9.2 Best Practices

Modular code, RESTful APIs, reusable components.

10 Features & Functionality

10.1 User Authentication

Signup, login, profile management.

10.2 Skill Discovery

Browse and verify skills through AI problems.

10.3 Developer Discovery

Search and filter developers by skills.

10.4 Compatibility Matching

AI-powered compatibility scoring.

10.5 Team Management

Create teams and collaborate.

11 File Structure

11.1 Backend Files

main.py, quiz.py, c_score.py, database.py

11.2 Frontend Files

App.jsx, Layout.jsx, components/, context/

12 Conclusion

CollabQuest provides a scalable and intelligent platform for developer discovery and collaboration using full-stack technologies and AI.