

# 1. Technical Feasibility

Component	Availability / Current Status	Notes / Requirements
Backend	Node.js — Available	Node.js runtime environment established; further scaling.
Data Sources	Prototype implemented using the GitHub API	GraphQL integration pending. The team should enhance expertise in GitHub GraphQL schema queries and rate-limit handling for continuous data collection.
Database	PostgreSQL — Available	Relational structure suited for analytics and social data; ensure proper indexing, caching, and partitioning for large-scale trend and feed data.
Frontend	Vue.js + Tailwind CSS — Available	Reactive UI design framework ready; implement modular components for feed, charts, and dashboards ensuring consistency across devices.
Visualization	Library to be selected	Integrate with a Vue-compatible visualization library (Chart.js, Apex-Charts, or D3.js) for analytics dashboards, trends, and GitHub metrics visualization.
Authentication	Custom accounts + GitHub OAuth	Enable app-native authentication plus OAuth via GitHub; store secure tokens and manage user sessions following OAuth 2.0 standards.
Hosting / Deployment	Cloud-based — Pending final selection	Candidate platforms: Vercel, Netlify, or AWS. Must support scalable backend APIs, secure data storage.
Social Feed UX	Design phase in progress	Plan for interactive and real-time community feed using Vue components and Socket.io; include infinite scroll, likes, and commenting features.
Hardware / Infrastructure	Adequate for current development	Development laptops and cloud resources are sufficient.

Table 1: Technical Feasibility

Future Feature	Purpose	How It Integrates	Example Implementation
<b>Community Skillboard / Needed Skills Section</b>	Show in-demand skills or projects in the developer world	Uses existing feed and visualization layers; adds a “Skills” database table and a “Trending Skills” API	Dashboard showing “Top 10 Emerging Skills This Week.”
<b>LLM-powered Project Generator</b>	Suggest potential projects for users based on their interests or GitHub activity	Connects an LLM API (e.g., OpenAI, Anthropic) to the backend; stores curated prompts and responses	“Based on your GitHub repos, here are 3 project ideas in AI tools.”
<b>Professional Collaboration Space</b>	Matches users with similar or complementary skills and goals	Extends feed and authentication logic; adds user skills and collaboration interest tables	Suggests users with overlapping interests or complementary skills.
<b>Multi-platform Data Integration</b>	Enriches developer profiles using multiple sources such as LinkedIn, Hugging-Face, or Kaggle	Adds new data source microservices that fetch and merge external profile data	Combines user’s GitHub and LinkedIn data to create a unified developer profile.

Table 2: Future “Dream” Features — and How They Fit