Decision Matrix

FRONTEND	How seamless are the user interactions	How scalable is writing more code for this	How robust is the language's package system	Easiest to deploy	Learning Curve for New Developers	How large is the developer community/support	Performance (speed and efficiency for animations, UI responsiveness)	Result
Swift	3 (native)	1	1	1	1	2	3	13
React Native	2 (not native)	3	3	3	3	3	2	19
Flutter	2 (not native)	2	2	2	2	2	2	14

DATABASE	Is this simple to use in APIs?	Is it relational? (prefer many-to-many for a friending system)	Is this scalable for handling large amounts of user data and planning information	How fast are the queries and mutation for user experience (depending on our app's use case)	Support for real-time updates (for live collaboration/trip planning)?	Built-in authentication and security features?	Result
Supabase	3	4	3	3	3	2	18
MongoDB	2	1	2	2	2	2	11
Firebase	3	2	2	2	3	3	15
Neo4j	1	4	4	1	1	1	12

BACKEND	How simple is the backend	How lightweight and flexible is the framework for custom use cases?	How well does it perform for small, serverless, or microservice-based architectures?	How seamless is the integration between frontend and backend?	Scalability for handling large concurrent users?	How well does it support background jobs & async tasks?	Developer experience & ecosystem (docs, tutorials, active community)?	Result
Django	4	1	2	2	2	3	3	17
tRPC	3	4	4	4	4	2	4	25
NestJS	3	2	3	4	3	3	3	21
Laravel	4	1	2	2	2	2	2	15