Use-Case for backend -technologies

Structured Query Language (SQL) happens to be the more structured, rigid way of storing data, like a phone book. For a relational database to be effective, you'll have to store your data in a very organized fashion. ACID (Atomicity, Consistency, Isolation, Durability) compliance reduces anomalies and protects the integrity of your database by suggesting precisely how transactions interact with the database. Often NoSQL databases sacrifice ACID compliance with processing speed and flexibility.

But ..

The main problem with SQL is scaling it as your database grows. You see, even though scalability is usually tested in production environments, it's often lower than NoSQL databases. Sharing is quite problematic as well. NoSQL databases ensure data doesn't become the bottleneck when all of the other components of your server-side application are designed to be seamless and fast.

Our scenario

The main motivation behind test right is to automate the entrance examination process from huge chunk of questions and to make sure that that predictability of a question to appear in an examination is reduced. We need the algorithms which serve the question to have access to huge amount of questions. The database should be very capable of handling huge amount of data and serving set of questions based on the filter applied. According to stack-exchange serve Node remains the most popular technology in back-end currently it also will ensure that the entire codebase will be in single language i.e. JavaScript which again is the most popular programming language in the world so that mean we can easily find the js developer also the language itself has very small learning curve and experienced developers can easily pick it up in very less time.