

## Scholarship Management Solution

Scholarship match up application built on top of CiviCRM

Our client is a non-profit program that links bright, motivated low-income students with educational and scholarship opportunities at some of the nation's best colleges. It is the provider of the National College Match Program and the College Prep Scholarship.

### Problem

Our client was generating 1000's of applications manually and this was becoming a major bottleneck. That is when they decided to develop a customized application that would automate a large part of the complete process.

### Solution

The cost of building something from scratch while attractive was hugely expensive. In addition, the time taken for such a development would be a couple of years and client requirement was immediate. That is when their technical team zoomed onto CiviCRM as a platform on which they could develop their customized application.

The application used the basic CiviCRM engine and built a customized application that suited the workflow that was required for the college match program. There were two major challenges. The first was in automating the complete process in a simple manner for the applicant. That meant that although the applicant filled in just one form, the same could be automatically customized and sent to the chosen colleges in a format acceptable to the college. The second was in handling the load since the number of applicants during the admission process would be in 100's of thousands. Further since the application form itself was long and needed a couple of hours to fill out, the concurrent load too would be very high.

The first challenge was solved by using custom data fields and the concept of collecting superset data or the cumulative data required by all colleges and then customizing the forms with that subset of data as required by the specific college. The load issue was handled by having a set of 5 servers, out of which 1 served as a backup and the other 4 were connected and handled and serviced requests in a load balancing manner, by directing the request to the server with the least load at that point in time. And the databases were synchronized in real time and so even if the applicant logged out and logged in a minute later, he may be directed to another server but was able to continue the process seamlessly.