

Auto-tagging application for student collaboration and networking system

Based on Pinax

Executive Summary

Client wanted to create a world class social networking and collaboration platform for a highly technical group of users. A key feature that client wanted was a quick and easy way for the user to create and add search content.

We decided and proposed using **Pinax** which provides an integrated collection of **Django** applications that provide the most commonly needed social networking features. We also proposed highlighting of relevant content as the user types and creation of dynamic tags.

Business Situation

The target group of users comprised of students from world class top colleges and the overall need was to create a platform which would help the students collaborate and get their work/projects done. In order to make it more effective the application also needed to have good social networking features so that users can know and interact well with each other.

About our Client

Client Custom software development company | **Location** Stanford, California, USA | **Industry** Custom Software Development

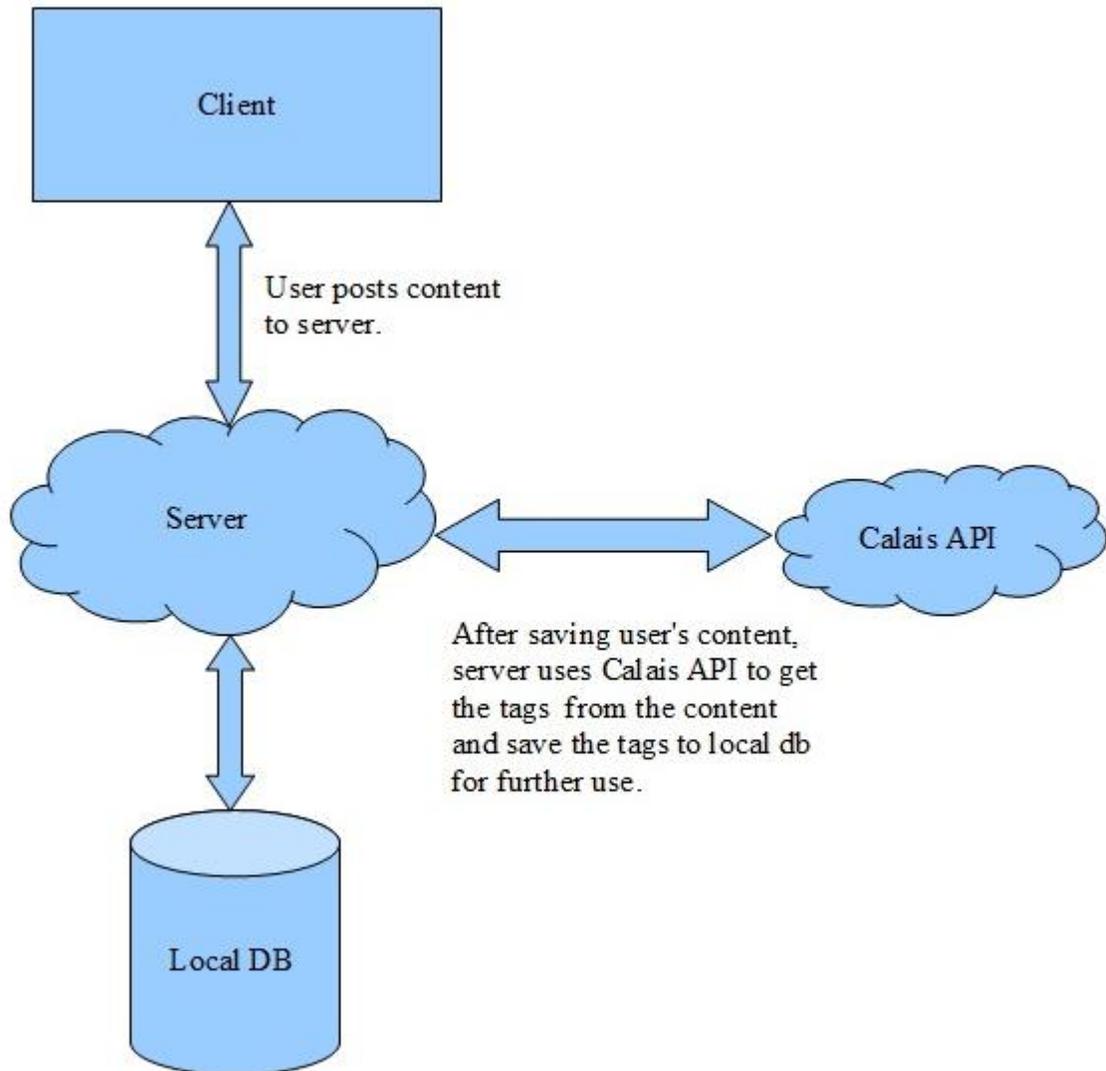
Technologies

Python, Django, Pinax, nginx front end, apache backend, MySQL, GIT, OpenCalais REST API for tagging.

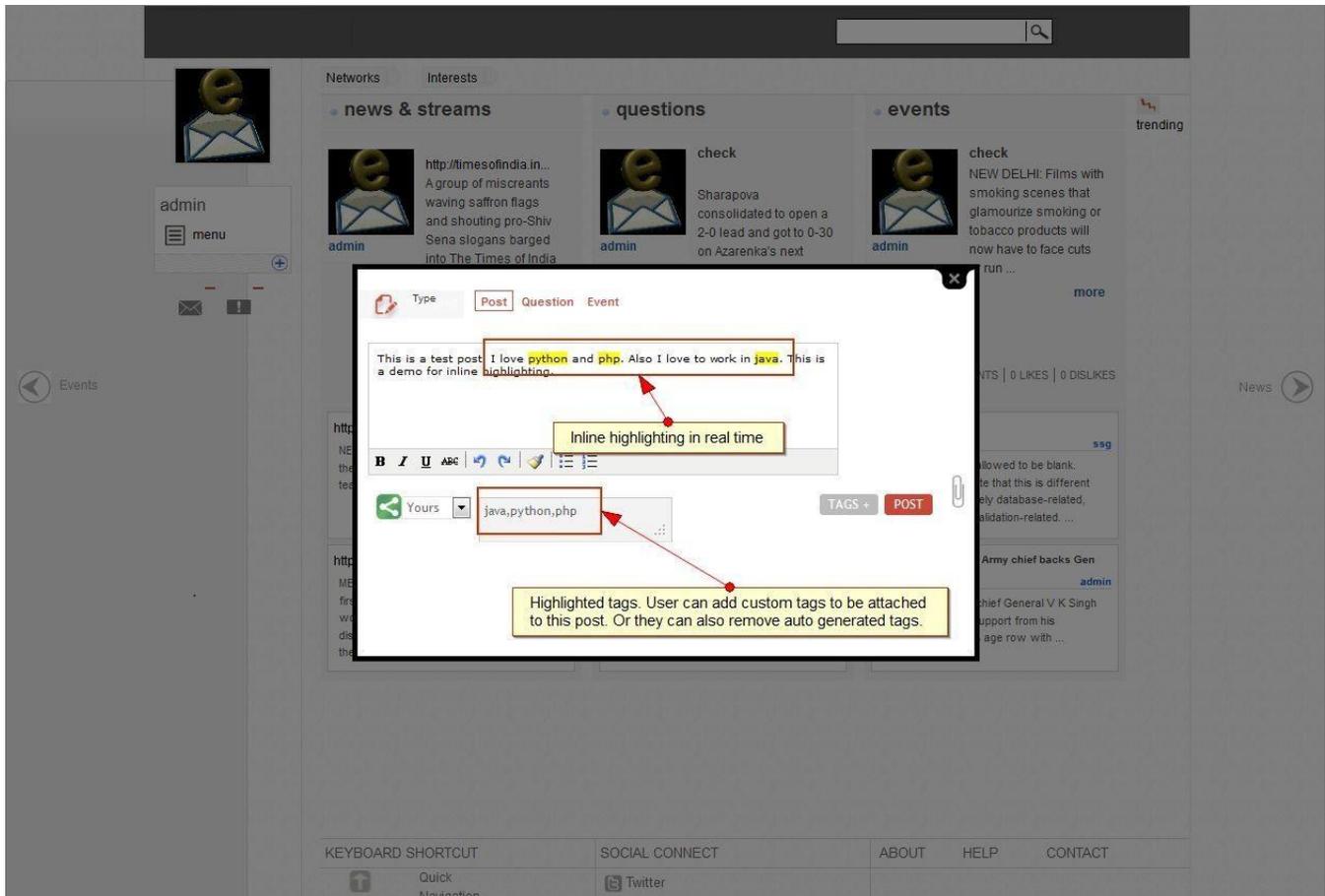
Our Solution

We designed and implemented a Python Django based application that makes the tagging process automatic. Whenever a user posts either a blog or question or event, the content gets tagged automatically. We used OpenCalais (<http://www.opencalais.com/>) API to get the tags from the content that the user enters. We also used Django Supertagging App (<http://readthedocs.org/projects/django-supertagging/>) to make all content tagging processes automatic. This included implementing inline highlighting of content as user types in the content. Along with Autotagging we also allowed users to enter their own tags to make the app as flexible and user friendly as possible. We used heavy Ajax implementation and used caching to avoid repeated calls to the OpenCalais API. Once the module was ready we integrated the code with design templates provided by the client.

Architecture of the application Diagram



Application home page



Future relationship

The client was pleased with Mindfire's effort and reckoned that they were happy to have discovered a professional offshore IT unit. We shall continue to be the service provider for the next versions of the client's product. They have not only allocated the support and maintenance work of the current system to Mindfire but have also chosen us for future customization work.