


MentorCRM

Michael Bernabei, Akash Devgun, Jose Vieitez

A close-up, slightly blurred image of a laptop screen. The screen shows a data visualization interface. At the top, there is a line graph with a blue line and several data points. Below the graph, there is a pie chart with a blue and green segment. The MentorCRM logo, which is a blue circle with a white 'M' inside, is visible on the right side of the screen. The text 'MentorCRM's Mission: Your mentors at your fingertips' is overlaid in white, bold, sans-serif font. The laptop keyboard is partially visible at the bottom of the frame.

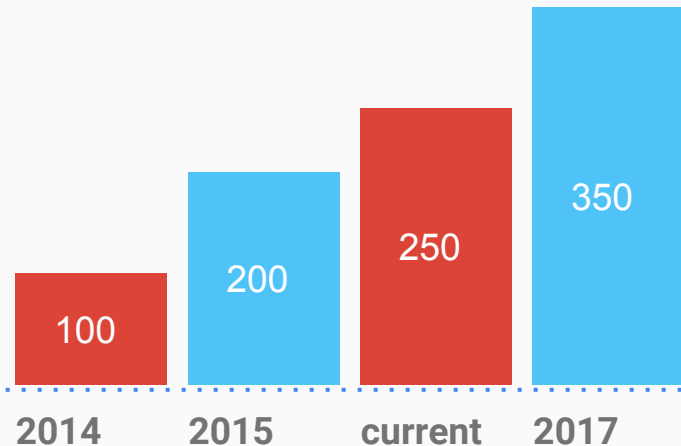
MentorCRM's Mission: Your mentors at your fingertips

The problem

Mentor management inside organizations is a game of excel sheets and email memos

Hundreds of mentors, hundreds of mentor - mentee interactions, one admin

Mentor Database Growth





The solution

MentorCRM

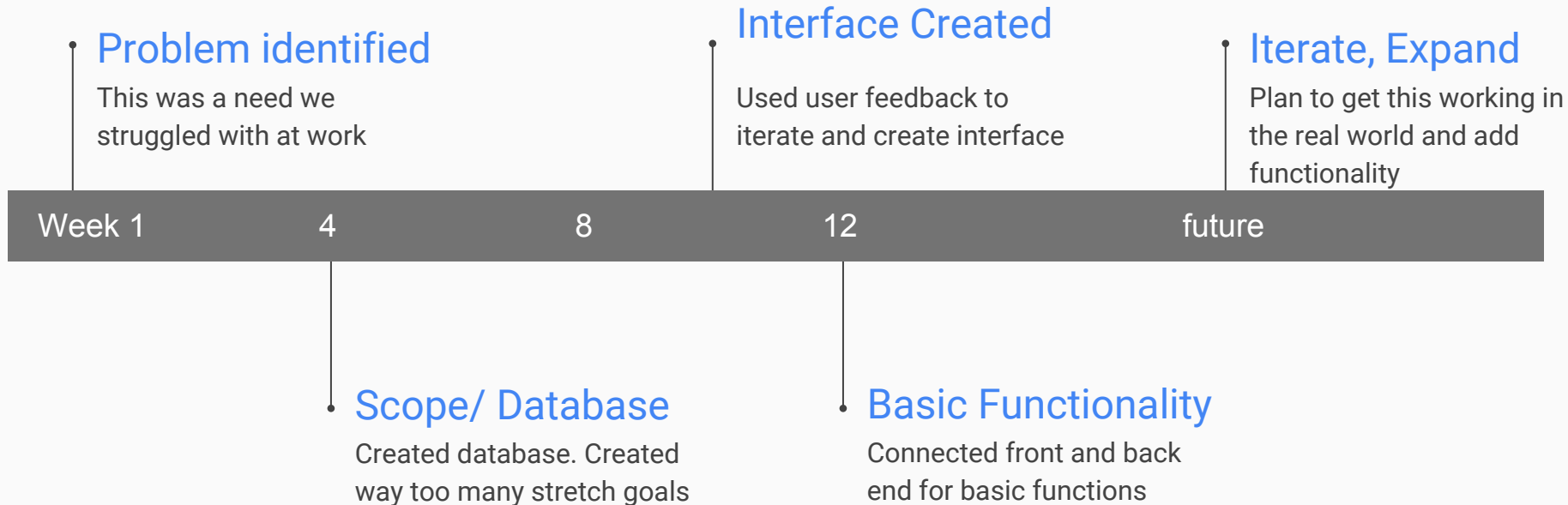
One place for all mentors

Easily Searchable

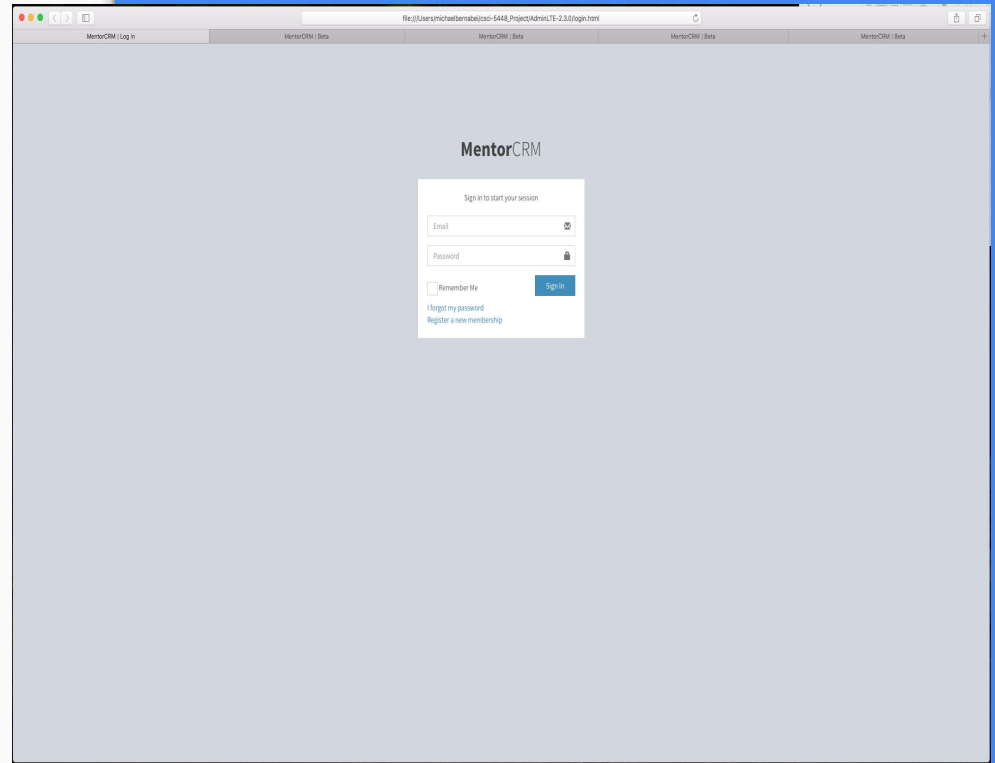
Tracks: ratings,
engagements, feedback

Milestones

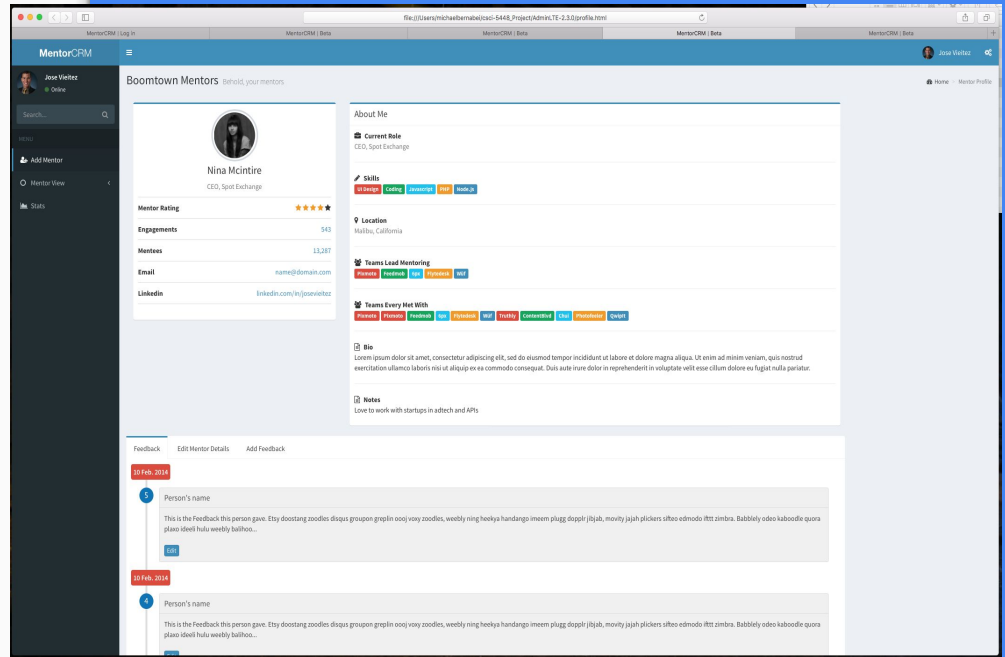
Show where you are in the process and what's left to tackle



Login



Profile



Adding Mentor

MentorCRM | Log in | MentorCRM | Beta | MentorCRM | Beta | MentorCRM | Beta | MentorCRM | Beta | MentorCRM | Beta

MentorCRM

Jose Veltez
Online

Search...

MENU

- Add Mentor
- Mentor View
- Stats

Boomtoun Mentors Behold, your mentors

Auto import using LinkedIn

LinkedIn

Email

Cancel Import

Manually Add Mentor Info

Name

Email

LinkedIn

Role

Location

Bio

Skills

Notes

no file selected Choose File

Photo

Upload Mentor Photo (Square only)

Cancel Add

Copyright © 2016 Michael, Akash, Jose. All rights reserved.

For support contact: jose.veltez@colorado.edu

Stats



Design Pattern

MVC

Being a web application we elected to use MVC

We used Django to implement MVC

We created a Django project and created a Mentor app within that project

MVC via Django

Django uses slightly different terminology when describing MVC, e.g. views are implemented by templates and the controller is the framework itself.

Django Model

Within a Django App there is a models.py file where each model class is defined.

To create a database containing the desired models simply issue the command: `python manage.py migrate`

A relational database will then be automatically created containing the models from the models.py file.

We spent an extensive amount of time on database/model design, therefore mapping our design to code was trivial

Django View

Django has a `views.py` file but it is not a view in the traditionally since. The `views.py` actually delegates to a Django template which displays the data.

Therefore Django Templates are actually the View in MVC

Django Controller

There is no real “Controller” file in Django.

The Framework is the controller

Connections between the views and models are made via a `urls.py` file