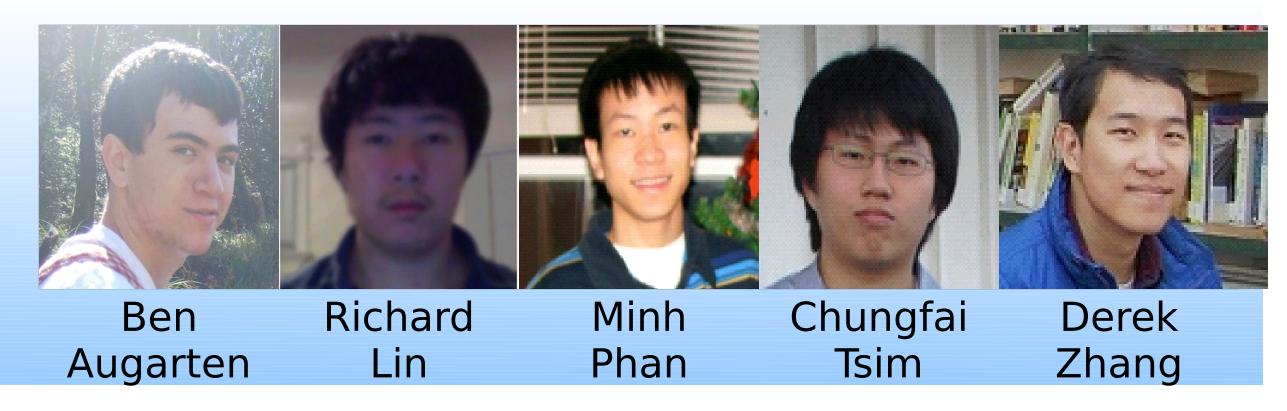


"To reduce world hunger and improve nutrition by providing small farmers with innovative drip irrigation kits"



The Problem

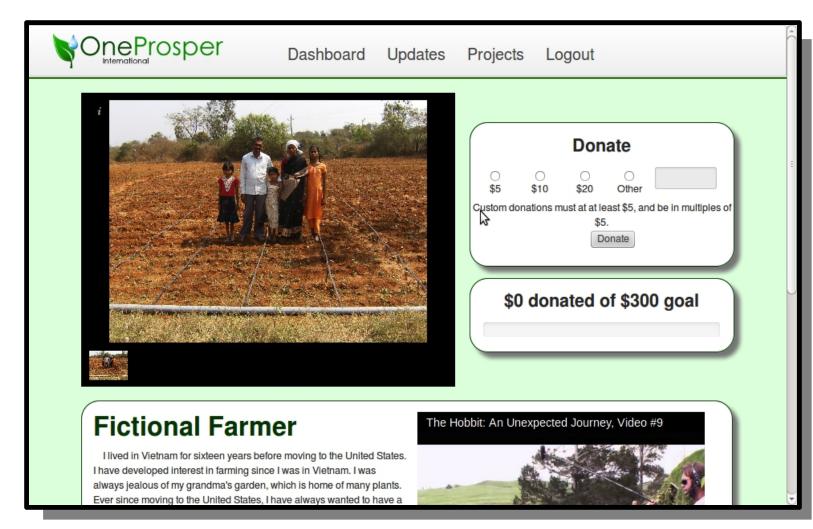
- •The problem: need to raise money to provide drip irrigation kits to farmers in India
- •Want a fun and interactive method to raise money

The Solution

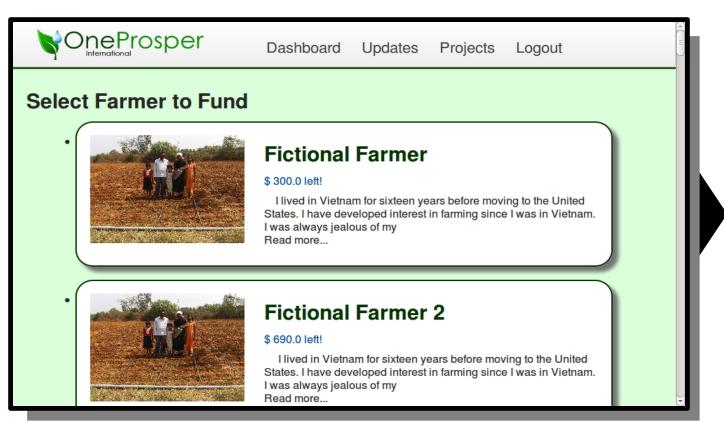
- •Goal: create a crowdfunding website where individuals can take action and donate
- Let donors see the farmer they are donating to and the impact of their contribution
- Add a social aspect to allow users to email friends to donate
- •Track the amount of money raised by each user's campaigns

Features

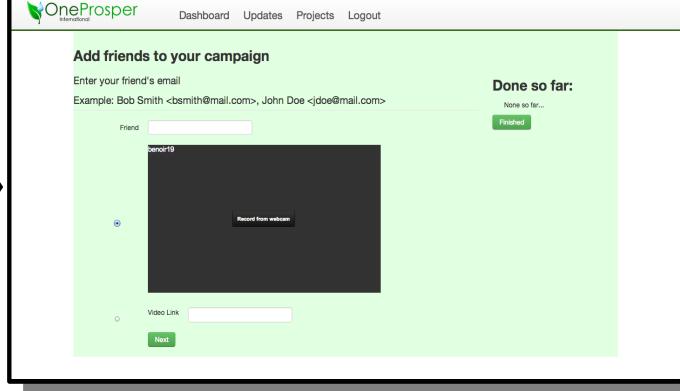
- Administrators can create projects pages for farmers
- •Each project page can have YouTube videos and pictures of the farmer
- Users can donate to farmers
- •Each donation is associated with a specific farmer, and funding for each farmer can be tracked



Project statistics

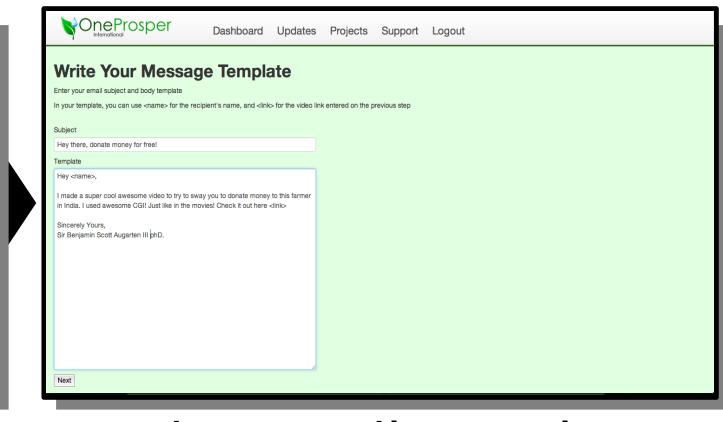


Select farmer



Add friends and videos

•Users can start a campaign for a farmer, soliciting donations from their friends through a personalized video and email



Write email template



Send from your address,

Design Decisions

- Leverage existing gems instead of reinventing the wheel
- •User accounts and authentication, YouTube API, Stripe payment processing, pagination, and email validation all rely on gems
- •Common user accounts for both donors and campaigners
- Keep all routes RESTful
- •DRY out code using helper methods

Challenges

- Testing Javascript
- •Testing external services like Stripe
- Integrating with external services, including payment processor, YouTube, and Dropbox
- Merge nightmares caused by overlapping features

Lessons Learned

- •Start off with Javascript test suite is intimidating to set up in the middle of a feature
- •RTFM (Read the Friendly Manual) when working with external APIs
- •Be aware of teammate's features and predict points of conflict

The Campaign Process