A Site for open data for research

Objectstream began working on creating useful reports using the openFDA API on (DATE). After looking through the available data, we began to work with an external group of researchers from public and private sectors to understand:

1. What data would provide the most immediate impact for the researchers.
2. What ways the data should be displayed in order to easily understand the information.

The conclusions after meeting with our professional board of researchers was that we needed to focus on the following:

1. Research data should be pulled for food, device and drug recalls.
2. Data should report across multiple timelines (in years)
3. Researcher should be able to enter word(s), phrase(s) in a search field in order to pull the data of interest
4. Researcher should be able to request data to be displayed as various graphs

From these conclusions we created user stories so that we could quickly develop a prototype application in an agile environment. Use this link to view the user stories.

The following documents our approach of development once we initiated resources to develop our prototype. We are following the U.S. Digital Service Playbook for the following documentation:

1. We created a project folder using Redmine to implement our Agile development approach. Here is a link to the project in Redmine.
2. We assigned the following responsibilities to specific team members:
    a. Product Manager– Vishal Maheshwari
    b. Business analysis oversight – Basil George
    c. Technical architect – Shyam Nagaragan
    d. Interaction designer – Johnson Eyadiel
    e. Front End Developer – Depak Sabu
    f. Dev Ops Engineer – Nabil Soulane
    g. Backend Development - Vimal Kovath
    h. Visual Designer – Martin Mathew
3. We set June 26th as our Sprint 1 Completion date.
4. We conducted daily stand up meetings up to June 26th to check on progress and completion of assignments. We documented all to do items and tasks with Redmine Agile plugin. Please use this link to view product backlog.
5. We created a Git Hub repository and added access for all project contributors.
6. We created a CI server running on Digital Ocean IaaS. The CI server was set to automatically update files based on pushes to the Git Hub repository.
7. Team tracked bugs in Redmine

8. We met with our expert panel on (DATE) to review our progress with the stakeholders. We identified issues with layout, reporting and recorded them for our developers to fix.
9. We received word that the timeline was expanded and used the time to add backlog work to Sprint.
10. We set July 3rd as the date for completing the final work on all expanded Sprint.
11. We met with our expert panel on July 3rd to review progress and identify specific user issues with UI/UX and data. Bugs were reported to team using Redmine.
12. Team was given to July 7th to complete all bug fixes and send final push out.

We used the following specified tools for development:

Front End (Client Side) – HTML5, CSS3, Angular JS
Back End (Server Side) – Node JS, Sails JS, MongoDB from Mongo Lab SaaS
Testing Tools –

We specifically separated the server hosting the data from the backend node server to help reduce the risk of failed services.

We also used Reddis from Reddis Lab SaaS for session store.