openFDA – A site for open data research

Links:
Prototype Github Code Repo: https://github.com/shyamrock/openFDAWebApp
Prototype URL: http://104.236.11.72/#/dashboard/overview

Approach for Pool Two – Developer Pool

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

Product owner and lead - Vishal Maheshwari

Additional Project Members:

Technical Architect	Shyam Nagarajan
Front End Developer	Depak Sabu
Dev Ops Engineer	Nabil Soulane
Backend Developer	Vimal Kovath

Our Technology Stack/Framework

Development technology:

Agile Methodology Management	Redmine
Testing Framework	Protractor, Mocha, Chai, Jasmine
Development IDE	Sublime Text 3

The above resources for development technology are all open source and freely available.

Runtime Technology:

Frontend	HTML 5, CSS3, Angular JS
Backend	Node JS, Sails JS
Database	Reddis for Session Store, Mongo DB

We used Digital Ocean IaaS to deploy our code.

We split this project into two sprints. Redmine was used to implement our Agile development approach. All development was done iteratively as per scrum practice. Here's a link to the product backlog and scrum meeting notes. Sprint daily meetings and Sprint retrospectives were both employed to identify issues (bugs, to do items, etc.) with the technology build. These issues were captured in the Redmine management system and then closed by our team.

We used Jenkins CI for the continuous integration and continuous deployment of the GitHub code files. Additionally, a GitHub plugin for Jenkins CI and the web hook component of GitHub was deployed to automate the build directly from GitHub. Here is a link to the document that explains our CI and CD approach. Digital Ocean IaaS was also used as the server platform to host the CI server.

For source control management, we used GitHub repository.

Docker was used as the container for the openFDA prototype. Here is the link to the Docker file located on the Docker hub. For more details about the Docker deployment, please refer to this link.

Assuming Docker is installed in your machine, the following commands will run the application :

docker pull shyamos/openfda_node_webapp

docker run d p 80:1337 shyamos/openfda_node_webapp sails lift

We continuously monitored the deployment infrastructure using our Nagious Monitoring Server (open source server) which also runs on the Digital Ocean IaaS server. Here is a link showing the results of the monitoring.

This prototype as been designed to be easily deployed on any other servers or local machines. This assumes that node.js is and Git is installed. Here are the detailed steps.

Open terminal in Mac or Linux ; Command Line on Windows Systems

1. Install nodejs: https://nodejs.org/download/

2. Install Git: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

3. Run “Git clone: https://github.com/shyamrock/openFDAWebApp.git”.

4. Cd “openFDAWebApp”.

5. Npm install. Run this command with Sudo on a Mac or in Linux.

6. Run “Sails lift”.

7. Open the app in a browser at http://localhost:1317.

Approach for Pool Three – Full Stack Pool

Product owner and lead - Vishal Maheshwari

Additional Project Members:

Technical Architect	Shyam Nagarajan
Front End Developer	Depak Sabu
Dev Ops Engineer	Nabil Soulane
Backend Developer	Vimal Kovath
Interaction Designer	Johnson Eyadiel
Visual Designer	Martin Mathew

Define the Site Use:

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.

Conclusions:

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

Use this link to view the user stories.

Approach UI and UX with Human Centered Design:
Keep things simple and intuitive for the user. Furthermore, our approach is to ensure that the site is accessible across multiple screens and be as inclusive to as many people who may be considered having a "disability". We've kept the site open standard using semantic HTML to structure the content so that support devices such as readers can better determine how to present information to the user. We have also used a responsive web design approach understanding that there are a number of users who will be using hardware devices with specialized keyboards (screen/touch keyboards) or will be using this site in remote areas away from their office space in order to investigate specific information. Please refer to this link for our guide to Human Centered Design.

Please refer to Approach for Pool Two – Developer Pool to review deployment.