CS 499R Proposal

Contributions to Open Source Software

Name

February 16, 2014

Course Topic

The readings course will be dedicated to a 12-week project for the FOSS non-profit OpenHatch (`http://openhatch.org`). The project is intended to challenge the student and provide real-world experience in contributing to open source while benefiting the overall project. The expected time commitment is 10-15 hours per week.

About the FOSS Project

OpenHatch is a non-profit dedicated to matching prospective free software contributors with communities, tools, and education. The OpenHatch community organizes outreach events on university campuses, seeks out mentors for FOSS newbies, and targets minority participation.

These activities are supported by the project's technical contributors. The OpenHatch platform scrapes the web to aggregate and classify bugs to present them in a more accessible fashion. The project is primarily written in Python using the Django web framework, and data is stored in a MySQL database. A number of different tools and the OpenHatch code itself can be found on github: `https://github.com/openhatch`.

Course Project Proposal

In order to help new organizers get involved in running OpenHatch events, building the following web application is proposed.

Brief explanation

In order to prepare for an on-campus event or the like, an event organizer picks out a set of small bugs that would be considered an appropriate difficulty for the audience. Currently, the mentors handpick these bugs by querying the web interface and copying bugs into a spreadsheet by hand. This is very labor-intensive and limits the organization of these "bite-size" bug sets to inside contributors.

Solution

This project would add a user interface and back-end support to the OpenHatch website for generating one of these "bite-size" bug sets by intelligently querying the database. The application would allow user interaction to customize and store the bug set.

Required skills

This project requires knowledge of (or ability to learn within 3 weeks) Python, Scrapy, MySQL/relational data processing, unit testing, a willingness to think about/make/take feedback on user interface decisions, Javascript, web forms, JSON data APIs.

Proposed timeline

Week 1: Onboarding for project. Meeting with the relevant users and determining user requirements. Deliverable: report of user requirements.

Week 2: Sketch of development work. Story board for user interface. Deliberable: planning material.

Week 3: Begin development. Educational material and mentorship provided for unfamiliar technologies.

Week 4: Development.

Week 5: Deliverable: Prototype of a single-page web app that allows the user to query for and see bugs (loaded from a static file), and select bugs via checkboxes to keep them in the "interested set." The prototype does no data storage and loses all state when the page closes.

Week 6: Development. Deliverable: Mid-term review.

Week 7: Development.

Week 8: Deliverable: Adjustments to single-page web app that let the user add bugs manually, query directly from the database. Still no writing to the database, but reads enabled.

Week 9: Development.

Week 10: Development.

Week 11: Deliverable: Python backend for retrieving and storing the data in the OpenHatch database.

Week 12: Deliverable: project demo and presentation. Brief report on what was accomplished and what future work is needed.