

SW 304/406

Dr. Amalia Rusu

Fall 2017

Group 1

Spark City Robotics

September 16, 2018

Prathna Pel

Alexandru Rusu

Jayawanth Katterapu

Spencer Letizia

Kyle Morehead

Jeffrey Kramer

Overview

Our project was to highlight Harding's FIRST robotics Team activities through a website. The website is geared to support the mission of the Bridgeport division group by both advertising their accomplishments and providing easy access of donations.

Requirements

1. **Website supporting the group:** The site provides detail about this group, some of their accomplishments and available resources.
2. **Easy access for donations:** The site provides a single click to paypal on every page to make donating a snap.
3. **Visual appeal:** The site needs to be aesthetically pleasing, easy to use and informative with respect to providing information about the field of engineering.
4. **Improve upon firstinspires.com:** The site needs may leverage content from firstinspires.com but needs to embellish that detail with robotics information and overall provide a more professional experience.

Specifications

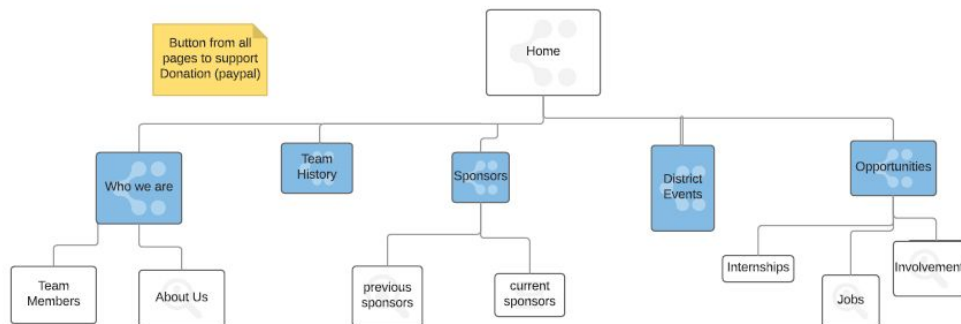
The site meets the following:

- A Home page describing the team, their interest and involvement in Robotics and some of the evolution of the group.
- A Page detailing specifics of 'who we are' as a team
 - Including the mission of the group
 - Information about the competition
 - Specifics on partnership with Fairfield University
- A page dedicated to each of the team members and their biography
- A page with historical detail including links to past events, designs, statistics and scoring metrics
- Pages dedicated to donation detail, previous and current sponsors, and success stories.
- Support on every page of a single button click which redirects the user to a pre-populated PayPal form
- Provides direct links to social media
- A Page providing information about upcoming events
- Pages dedicated to opportunities related to engineering and robotics
 - A form provided for interested members to submit their information
 - A page describing some reasons why college makes sense, detail about Internships in the field, and links to current opportunities
 - A page illustrating a list of jobs in engineering and robotics typical compensation and some of the required education.

Design

The design supports our basic static content based html site. We've kept our overall depth no more than three clicks deep for any page. We have made efforts to employ code reuse of component based include files such as the nav (menu), header, and footer.

Diagram



Implementation

The implementation has emphasised using the building blocks of this course, specifically: HTML5, CSS, and JavaScript.

- Within the HTML we've used features including forms, tables, images, navs, footers, and headers.
- The CSS of the site comes in external, internal and inline
- We have followed a standard directory structure with .html / .css on the root of the project and subdirectories containing images and include files.
- A wealth of graphics
- Basic JavaScript has also been employed to enable including repeated sections of code

Testing

- Each developer has unit tested their individual group of files across multiple browsers after minor and milestone changes.
- The integration of github to our project emphasised continuous integration testing as all developers were frequently pulling the latest changes from git and were then forced to scrutinize and broadcast defects all through the development cycle.
- The application has been frequently run thru both aptana studio and Apache httpd 2.4

Deployment

Deploying the application to the container of your choice is a straightforward process:

- Uncompress (unzip) the application (robotics.jar) and copy the entire directory structure to a directory in your deployment directory. For example on Apache HTTP Server we've deployed the app from ../httpd/Apache2.4/htdocs/Robotic.
- Restart your web server and the homepage of the site will be available at:
`http://<yourServer:yourPort>/Robotic/index.html`

Maintenance

From time to time changes need to be made to the application, here are a few helpful hints:

- The majority of the code is html.
 - Modifying content can be done by non-programmers if you can identify the file and section of what you'd like to modify.
 - Changing the structure of the site will best be done by an experienced web developer
- Graphics are all stored in an `./images` directory off the project root
 - To change an image, simply replace the existing image with the new one of the same name
- 'Shared files' are stored in an `./includes` directory off the project root
 - Changes to code in an include file will be propagated across all pages that include this file
- The application also employs limited javascript.
 - `.js` has enabled us to reduce some of the repetitive sections.
 - The process of extending this include functionality is illustrated in a 'howto.txt' built into the project
- The application has been styled with `.css` in all of an external (`main.css`), internal and inline.
 - Modifying the style of a page will be done through the specific `.css` dedicated to that page or section.
- The website has a 'donate' button on each page which can be modified in the `./include/donateForm.html`. Changes to the detail of how donations are itemized to paypal could be performed here.