Implementation Plan

AlumniConnect

5 Step Plan

1. Make Heroku app and connect to GitHub (complete)
2. Connect pgAdmin to Heroku (complete)
3. Migrate phpmyAdmin to pgAdmin (complete)
4. Configure Heroku to launch the website (incomplete)
5. Make appropriate changes to code (on going)

Step 1: Make Heroku app and connect to GitHub

Heroku is a salesforce cloud platform that allows personal accounts to deploy apps to the web for free. For AlumniConnect, we will need the HTML, JS, CSS, and Node.js files hosted on GitHub as well as a free add-on, postgresql. Apps can be connected to GitHub under appname>Deploy>App connected to GitHub.

Step 2: Connect pgAdmin to Heroku

The mySQL system originally used for AlumniConnect, phpMyAdmin, was unfortunately very difficult to connect to Heroku, so we will need to migrate to pgAdmin which uses PostgreSQL instead of mySQL. To do this we must create a **postgres Database** (under Resources), install the **postgreSQL client**, and connect it to heroku database (information under Settings>Database Credentials). Links in speaker notes.

Step 3: Migrate phpmyAdmin to pgAdmin

Since these database engines use slightly different syntax and keywords, the database migration cannot be done automatically. First, the tables should be created manually using pgAdmin's table creation dialogue (databasename> Schemas>public>Tables (right click)>Create>Table…). Next, the values can be exported from phpmyAdmin as SQL (tablename>Operations>Go) and the INSERT statement in the phpmyAdmin file can be used to populate the pgAdmin table (tableName (right click)>Scripts>Insert Script). To confirm that the data was migrated run a Select Script on the table.

Step 4: Configure Heroku to launch the website

In order to deploy the AlumniConnect app, Heroku needs the GitHub repository to be formatted in a specific way consistent with the buildpack desired. In this case, since the backend of AlumniConnect is Node.js, the Git repository must follow node.js buildpack rules. The easiest way to do this is to migrate the project to something like AndroidStudio which will automatically format files and then re-upload to GitHub. After this build is successful, the deployed page will open the main (whatever that is designated to be) page.

Step 5: Make appropriate changes to code

A number of things may break when deploying the app:

1. Links to file must be URLs not local files
2. Node.js backend needs to connect to the Heroku Database
3. Node.js backend may not be able to listen on ports, and may need to be called as a URL instead of a port.
4. JavaScript API calls to the Node.js needs to interpret the JSON in the same way
5. User testing is enabled due to deployment and will add necessary revisions