Website Technologies

Front End

Html5, *css* and *javascript* will be used to develop the client side stuff.

We'll need to make a responsive design here (able to look nice on both large screens and mobile phones/tablets), I'm not certain what the best approach here will be, the way I see it we can either develop from scratch (or use my current website as a template for instance) or use something like bootstrap. Personally I'm leaning towards bootstrap, but the people who will actually be designing the front end should be able to work with something they are most comfortable in.

Back End

MySQL database to store data, and *php* to process server side stuff.

There are a couple of options for the php, we can either do everything from scratch (Michael and I have experience in this) or use something like CakePHP which we have never used before, but apparently makes life much easier. This will require further investigation.

Hosting

Michael and I have decided to use a program named heroku to host the website. If you want to access it you will need to make an account at https://dashboard.heroku.com/ and I can add you to our project.

Currently I have hosted a website from a previous subject as a test to make sure both the website and database are working how we want them too. You can see this at:
https://ifb299-group52.herokuapp.com/

Database

This in turn accesses a database with the following access details:
Hostname: us-cdbr-iron-east-04.cleardb.net
Username: b82dd638ef1f83
Password: 07d1dd63

MYSQL workbench is probably the easiest way to access it, so I would just download and install that and connect to it using the details above if you need to.

Remember to use a local copy of the database to make changes first, (probably a good philosophy for everything anyway!) as due to the database not being controlled by git if we change something in a bad way, there is no easy way to recover the old version.

Collaboration Technologies

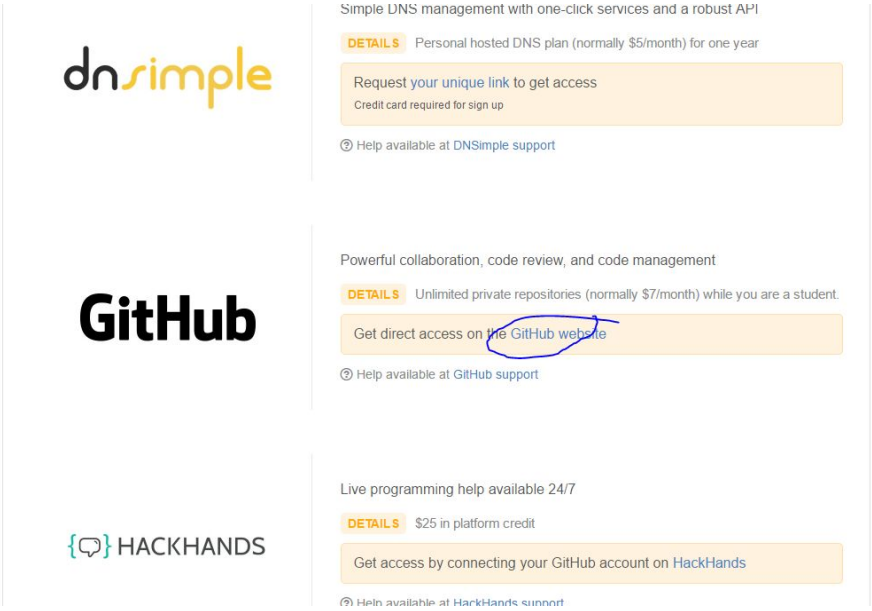
GitHub - Setup

Sorry to the guys who have already setup bitbucket, but unfortunately they only have 4 user accounts max for each project, and we will need 7 (all of us and the tutor).

So we have decided to use github, which when set up as an educational account gives you pretty much unlimited access for everything.

If you set this up the normal way, it can take like a week to get approved, so here are some quick instructions to get an instant account and connect to our project:

1. Go here: https://www.dreamspark.com/Student/Default.aspx and follow the instructions to create and verify your account. I used the @connect email to set it up, and now have it linked to my personal microsoft account
2. Go here: https://www.dreamspark.com/Product/Product.aspx?productid=92 and get the GitHub Student Developer Pack.
3. Click the link that I circled below.



4. I already had an account so at this point it just activated automatically for me, I think you guys will probably have to make an account here.
5. Once you have an account you'll need michael to give you access to it, then just follow the git instructions here at the blackboard git tutorial section, using https://github.com/LittleMik/IFB299 as the remote repository. If you already had linked with the old bitbucket repository, I would suggest just making a new folder rather than trying to change the existing one, I had some issues with that that I didn't really understand too well.
6. Once that is all set up, pushing to the 'master' branch should automatically update heroku, and therefore the application.

Workflow:

We should probably talk about how we are going to work this--I (Greg) am absolutely not the authority on this--but the way I understand it the way we would make changes to the website would be as follows:

1. Use git to pull the online repository to a local version.
2. Make your changes
3. Commit and push it.
4. At this point, it's possible that while you've been working on the website, someone else has as well. This means that the push command will fail. To fix this, the repository will need to be pulled again however using the command git pull --rebase will sync your local repository with the online one, you can google it for more info.
5. At this point either everything will be fine and you can push to the master branch, and the thing you changed will be on our website. However, if someone has edited the same file as you have you will run need to manually edit the file to reflect both your's and the other person's changes.
6. Once you have merged those files, you need to use the add command on the file in question, and then the git rebase --continue command.

As we are a small team, and I think we'll probably be working on different stuff at different times most of the time I doubt this will be an issue too often but it is good to have a plan in place to deal with it.

The below website has much better documentation on the workflow I think we should use, specifically Centralized Workflow:

https://www.atlassian.com/git/tutorials/comparing-workflows/centralized-workflow