Final Paper Rubric

Wednesday, March 2, 2016

12:24 PM

Deliverables:

Information on font sizes, spacing, and APA style can be found in the ITM student handbook

\*Note your group may need to add more details in certain sections

Formatting style located starting on page 16

http://appliedtech.iit.edu/sites/sat/files/pdfs/ITM/ITMUndergraduateStudentHandbookFall2015.pdf

Total Points: 500 points (50 elements listed below)

Each section/bullet point is graded on a 3 position scale

10 points

Exactly explained and demonstrated described concept. Additionally provided proper visual authentication, written documentation, and proper citation from textbook.

7 points

Adequately explained and demonstrated concept. Provided visual authentication, written documentation, and proper citation from textbook. Some components were missing and or not clear.

5 points

Somewhat explained and demonstrated concept. Provided visual authentication, written documentation, and proper citation from textbook. Some components were missing and or not clear.

3 points

Inadequately explained and demonstrated concept. Did not provided all necessary visual authentication, written documentation, or proper citation from textbook. Some components were missing and or not clear.

iRL: In Real Life

Illinois Institute of Technology

Seth Carpenter, Brian Semaru, Jigar Patel, Shreyank Patel, Prayag Patel

IRL is an application to bring people together in the same physical space as opposed to the current digital tools that have a tendency to keep people at their screens digitally communicating.

The use case would be as follow. You're alone on a Friday night with nothing to do. You want to instantly find people who are in a similar situation and want some company, but don't want to randomly message every Facebook friend that has a green dot by their name. In the modern age, a green dot does not indicate freedom. In this situation, you need IRL.

The app allows students to sign in using their Hawk credentials. Every user has a time left or they don’t appear in the application. Contact information each student is displayed if they are availability.

I would like to pilot this on our campus which is in great need of more social interaction. If 40-100 people sign up for the application, we would consider this a successful test.

**Technology Choices**

We chose to use Apache for our webserver. It is free, well-documented and easy to install and configure.

We chose to go with MySQL to be our database platform because it's open-source, free and popular. There is also excellent documentation on how to use MySQL with PHP.

We created a responsive design without using any CSS templates. We are using the Eric Meyer reset CSS code; but beyond that, in order to ensure proper display on all devices, no templates (including Bootstrap) will be used.

We chose Ubuntu as our on the CI server and production server because it is free, supports all of our technologies, and is easy to deploy to Eucalyptus machines.

We chose PHP because it plays well with mySQL and is good for processing form data. Also, our teammate, Jigar had experience with PHP from his internship at AT&T.

**Development Tools**

We used our preferred editors to edit files in a consistent Vagrant development environment. Seth used Brackets v1.6, the open-source code editor by Adobe.

Prayag used Sublime text v3, sophisticated text editor for code

Brian used Notepad++ v6.9 because he is too stubborn to learn a new text editor (you don’t want to know how long he used “edit.exe” before he switched to Notepad++!)

**Costs**

Our infrastructure was provided through the school, but if this was done in the real-world, there would be real dollars spent. Figure 1 through 3 show a basic comparison between our setup, AWS, and Azure settings.

**OUR SCHOOL SETUP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **CPU(s)** | **RAM** | **HDD** | **COST** |
| Jenkins CI Server | 1 | 4GB | 60GB |  |
| Production | 2 | 8GB | 40GB |  |
|  |  |  |  |  |
|  |  |  |  | **FREE** |

Figure 1

**AWS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **CPU(s)** | **RAM** | **HDD** | **COSTS** |
| Jenkins CI Server | 2 | 4GB |  | $114.20 |
| Production | 2 | 8GB |  |
| EBS Storage |  |  | 100GB | $5.00 |
|  |  |  |  |  |
|  |  |  | Total | **$119.20/month** |

Figure 2

**AZURE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **CPU(s)** | **RAM** | **HDD** | **COST** |
| Jenkins CI Server | 2 (SUSE) | 3.5GB | 60GB | $69.94 |
| Production | 4 | 7GB | 120GB | $139.87 |
|  |  |  |  |  |
|  |  |  | Total | **$209.81/month** |

Figure 3

**Design and Testing**

We used a typographic scale as taught by Professor Karl Stolley in Human-Computer Interaction. Red font colors were chosen to evoke a connection Illinois Tech. To make sure the interface was readable, a transparent dark background overlays the photo in our background. Rgba CSS values were used for this effect.

We used Trello to assign tasks to people and report bugs. We used the “Assign Members” feature to give a task to a particular teammate. This was adjusted later if someone else took on the card. Our build notifications went to the Slack channel named team-3-builds.

• Build Tool and Continuous Integration (CI) Server

* (17) Document and describe how your Build tool will interact with your CI Server and explain how your infrastructure will be deployed via your software pipeline.

**Tools we Used**

We are using a eucalyptus machines to run a Jenkins CI Server as well a Production machine running the LAMP stack.

The Jenkins CI Server was setup to lint the php files using a Phing plugin, and if no syntax errors were found, it would push the new code to production. While best practice would have been to push to a practice environment prior to production, the team opted to push directly to production since we do not have an active user-base at this time. Once we expand the project and obtain an active user-base, we will most likely add a test server to allow the developer to check their work manually before pushing to production.

Our infrastructure is deployed via shell scripts.

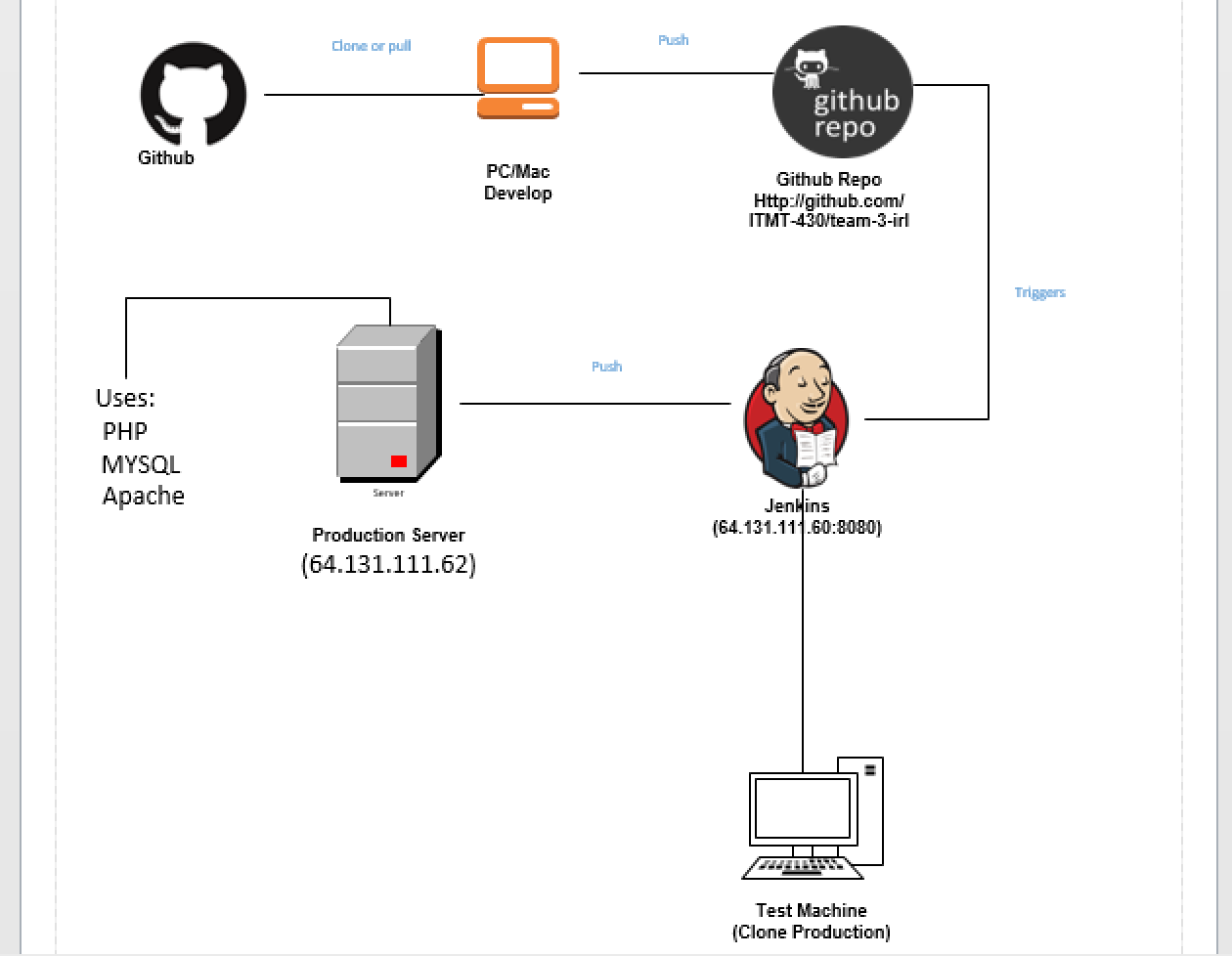
These shell scripts were designed to be run without downloading. A user could use curl to send the scripts to the bash shell. A configuration script for Jenkins and our Production server were created to allow automated configuration, and a master deployment script was created to automatically deploy the Eucalyptus machines and then configure those machines once the admin confirmed connectivity. One thing that could have been improved with the scripts is checking the connectivity programmatically; however, when we attempted to implement this feature we ran into a number of bugs that caused our scripts to run 5 instances for every 1 that it was supposed to start. We did have a good idea of how to fix these errors; however, it wasn’t feasible in the amount of time that we had to complete the scripts along with a few other bugs that we ran into. Since we could obtain a working product without it, this feature was tabled for later in the event that we wanted to continue the project after this semester.

**Infrastructure**

o (29) Use Elastic (permanent) IPs for all designed infrastructure pieces.

• This must match your Visio diagrams

• Complete Visio diagram of infrastructure



o (34) Use the AWS stencils for Visio to represent any Cloud items

• https://aws.amazon.com/architecture/icons/

We used Visio to design the structure of the whole interface.

We used AWS stencils to show the workflow of our process with icons.

The main process we used for designing the website was Fluid UI. Team member Shreyank worked on this design process alongside our Project Manager, Seth, with his guidelines.

**Application Developers**

We decided not to allow anonymous use of the site due to security concerns. At this point, we don’t want people who aren’t affiliated with the University to have access to the information therein.

Someone who is assigned administrator privileges gets a new section in the settings menu. They are able to Assign and remove other administrators as well as reset the time of other users that are logged in.

• Must have operational introspection (Infrastructure)

* (41) Website page where sys-admins can turn on or off features of software

o (42) Must have a database save/restore backup feature for entire site

o (43) Must have a feature that turns the site into read-only (no uploads) with the push of a button by system administrator

• This requires you to turn UI elements visible and invisible

**Admin Feature**

We started off the application with 5 test users and added additional test user through CAS authentication for testing purposes. Originally, CAS was only on a test environment so we had those 6 users test with their hawk credentials. Since our goal is a campus wide usage of this application, we will have to work with OTS and get our application live so students can use this platform to come together. We do have system admins with few privileges where they can assign or remove other admins. They can reset or kill the time for the people that are online and available.

**Operations (Ops)**

• Operations person must decide metrics to capture

o (44) Describe the type and nature of the metrics your application will collect and display

o (45) Explain which tools your will use for and why

o (46) Must perform benchmarks and baseline recordings of your application underload and at rest.

Security and Testing

• (47) Must prove that system is SQL injection and OS is secure (firewall ports and uname/passwords)

• (48) Must encrypt the content of databases

**References**