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 Techonology

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 and digitally communicate.

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 this situation, you need IRL.

The app would allow students to sign in using their Hawk credentials. Every user would have a current status, free or invisible. You would then be able to choose whether you wanted to broadcast to all people or just a list of friends. From this broadcast group, free people within a specified radius would appear. You would then choose a person or multiple people. If both ends accepted the request, then each user's status automatically changes to invisible.

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.

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.

• (6) Describe and explain the development tools you chose and why

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.

**OUR SCHOOL SETUP**

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

**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** |

**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** |

• UI/UX design outlines

* Present your outlines and mockup designs

• (9) Show the change over time of your UI/UX based on feedback and bug tests

• (10) Justify and explain your UI/UX design choices for each page in your application

 Based on ideas learned in ITMD 434/362

We used a typographic scale (DO THIS SETH) as taught by 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.

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

Our infrastructure is deployed via shell scripts.

We are using Phing to lint the PHP code

Infrastructure

• Use of load-balancer

o (28) Deploy a Eucalyptus Load-Balancer (ELB) in front of your application

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

• This must match your Visio diagrams

We are using

• Use of EBS storage

o (30) Databases will be installed but the actual database will be moved to an EBS (Elastic Block Store) during configuration/installation

• Database functions (Application)

o (31) Create Master/Slave replication in your database

• (32) Separate your application to send your database writes to the SQL Master

• (33) Separate your application to send your database reads to the SQL Slave

• Complete Visio diagram of infrastructure

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

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

Application Developers:

• CAS authentication

o (35) Users authenticate through @hawk.iit.edu (You will be connected to OTS who will give you the necessary libraries)

o (36) Implies you will need to use session as well in your application

• Person responsible for testing and security will need to prove that Http Session is in place and working

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.

• Application must be tested with real data; valid data of consequence (Testing and Security)

o (40) Describe the method used to create valid test data.

o Create pre-populate items/questions/schemas for usage testing

• <https://github.com/marak/Faker.js/>

• Must have operational introspection (Infrastructure)

o (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

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**