Phase 2 Deliverables

461L Team Project Contract Team E8

Tyler Pak, Jack Burrus, Lucy Zheng, Tyler Black

Contact List:

Tyler Pak: paktyler@utexas.edu , Github username: tylerpak

Tyler Black (Phase2 Lead): tyler.black@utexas.edu , Github username: guest1GRjBAYo

Lucy Zheng: lucy.zheng@utexas.edu, Github username: lzheng777

Jack Burrus: jackburrus@utexas.edu, Github username: jburrus1

Github repo: https://github.com/UT-SWLab/TeamE8

Deployed app: https://college-basketball-infosite.uc.r.appspot.com/

Expectations

- Everyone will communicate regularly
- Everyone is expected to attend the weekly meetings, and if they can't, communicate why
- Equal amounts of involvement
- Learn from each other

Communication

- 3 Slack report days a week: Monday, Wednesday and Saturday which includes
  - What each member has done
  - What they're working on
  - Any issues
- Weekly meetings via zoom every Friday at 10:00AM
- Code will be shared using Github repository at https://github.com/UT-SWLab/TeamE8

Conflict Resolution

- Minor conflicts will be handled internally through team communication channels (zoom, slack etc.)
- With major conflicts we will request a third party to mediate discussion such as a TA or Professor Eberlein

Team Role Expectations

- Phase Lead: Facilitate team meetings with a balanced approach. The lead should not be a dictator, but still maintain a strong leadership role. The phase lead will also document team meetings.
- Here are the primary responsibilities of the team:
  - Tyler Pak: Frontend design and development
  - Jack Burrus: Frontend design and development
  - Lucy Zheng: Backend development and website deployment
  - Tyler Black: API method development
- All commits to the github repository will be done via pull requests. We will work in pairs to review and check each other's commits to make sure everything is working. Tyler Pak and Jack will check each other's work, and Lucy and Tyler Black will do the same. These pairings were decided because both people in each pair are working on similar parts of the project.

Participation

- Member tasks will be assigned each meeting and each member is expected to complete them within time constraints. If a member is having difficulty with their task, they should communicate that to the rest of the team so that they receive help.

Behavior

- Team members will treat each other with respect at all times and communicate fairly.
- Team members will respect each other's ideas and will always hear each other out.

Technical Report

Tyler Pak, Tyler Black, Jack Burrus, Lucy Zheng

Team E8

Men's College Basketball Internet Database

Contact List:

Tyler Pak : paktyler@utexas.edu , Github username: tylerpak

Tyler Black(Phase2 Lead): tyler.black@utexas.edu , Github username: guest1GRjBAYo

Lucy Zheng: lucy.zheng@utexas.edu, Github username: lzheng777

Jack Burrus: jackburrus@utexas.edu, Github username: jburrus1

Github repo: https://github.com/UT-SWLab/TeamE8

Deployed app: https://college-basketball-infosite.uc.r.appspot.com/

Motivation and Users:

We picked college basketball as our topic because we saw great potential for application with users who need our data to make predictions on game outcomes, or to help build fantasy teams.

User Stories (Completion times on github project board):

Phase1:

1. As a user, I can search for teams, players, or games by name or date the game was played. Users on our site will be able to search for instances of our models and select specific instances with ease.

2. As a user, I can access the games a certain team participated in from their instance page. Users on our site can currently see "featured games" on each team page and select one to take them to the corresponding game page.

3. As a user, I can access a home page that contains navigational links to other pages. Users will initially arrive at a home page that gives them access to every other page on the site via links and navbar.

4. As a user, I can view the tools and APIs used on the site. Users can access and “About” page that includes all the tools we used, and links to all our APIs.

5. As a user, I can access other instances from an instance page. Users can visit instances from other models from an instance page. For example, a user that is on a player instance page, can access that player’s team page, or a game page that the player participated in.

Phase2:

1. As a user, I want to be able to see the entire team's roster when on a team's page. These players should appear as names and I should be able to access that player’s page by clicking on their name.

2. As a user, I want to see individual player stats on each game page. These stats should appear in a readable manner and be relevant to the subject

3. As a user, I want to search instances by name. There should be a search bar where I can search for instances easily and get results in a usable way.

4. As a user, I want to be able to see players' profile pictures. These pictures should be visible and displayed in an appealing manner.

5. As a user, I want to be able to navigate the site intuitively. This means I should be able to access all model pages easily and move through results using pagination.

Testing:

Selenium Tests

There are four basic tests for website navigation as well as three more complex tests.

The basic tests, test_about(), test_playerModel(), test_teamModel(), and test_gameModel(), test navigation using the navbar to the about page, 1st player page, 1st team page, and 1st game page respectively. Since the navbar is inherited by every page, the tests all run from the home page, but should work regardless of the page they start on.

The complex tests, test_playerInstances(), test_teamInstances(), and test_gameInstances(), test every link on each player, team, and game instances on the first page of each model to make sure there are no broken links. Ideally, these would check every instance page, but currently the tests take a very long time to run, so we limited them to the first pages of each model.

Unittest

The API was tested using Unittest in python. The API sends HTTP requests to the ESPN database and receives back JSON data. The test was used to verify all of the data that the API returned.

Models:

Our models are players, teams, and games. We currently are only using images as multimedia, but will add more in following phases

Tools:

MongoDB- Database used to store all information collected from the APIs we used.

Bootstrap4- Used for making the website responsive and adding helpful features like the navbar.

Flask- Used for site framework and adding some functionality to the website.

CSS- Used for styling the website to make it look nice and to have a consistent look across the site.

Google Cloud- Used to deploy the website.

Selenium- Used for testing site navigation and checking for broken links.

Git/Github- Used for version control and sharing code.

Sources used for references:

https://www.w3schools.com/bootstrap4/

https://www.tutorialspoint.com/mongodb/index.htm

https://www.youtube.com/watch?v=Z1RJmh_OqeA

APIs:

https://github.com/md130330/Ncaabballstats

_____NCAA BBall Stats provides extra player statistics, per game.

https://sportsdata.io/

Sports Data IO gives fantasy data for players and teams.

http://www.espn.com/apis/devcenter/docs/

ESPN provides most of our data for Teams, Players, and Games.

Phase1 Reflection:

Our team learned a lot in phase 1. Tyler Black learned about HTTP requests and json formatting and accessing. He learned that HTTP requests take much longer than any other part of the API so he realized that optimizing the speed of the API was something he would need to focus on. Lucy learned how to write functions that return data from the API. Jack and Tyler Pak learned how to use bootstrap to create responsive websites that are easy to navigate. We also learned a lot about each of our individual strengths and weaknesses as programmers, and how to work with each other to create a product.

Five things that we struggled with and need to improve:

1. API speed optimization when dealing with HTTP requests
2. Database organization with MongoDB
3. Presentation of information in a visually appealing manner on the website
4. Supplementary data and information for webpages
5. Reviewing each other's work

Five things that worked well:

1. Communication was great between all team members; we all knew what each of us were responsible for
2. Creating the basic website components
3. Deploying the website
4. Team meetings were very efficient and deliberate
5. Individual initiative; all team members constantly looked for ways to improve the product throughout development

Phase 2 Reflection:

Our team refined knowledge in phase 2. Tyler Pak used jinja2 to display data from the database to the webpage. Jack learned how to test website navigation with selenium and realized that selenium is intuitive once he understood it. Lucy struggled with writing the database at first, but learned MongoDB through various online resources. Tyler Black learned a lot about regex in this phase and used it to get data that not was not officially offered by the API.

Five things that we struggled with:

1. Initial understanding of the tools we needed to use
2. Initial time management. A lot of us were busy early on with other classes
3. Clear understanding of specific requirements of product
4. Page loading times
5. Efficiency of code in certain methods

Five things that went well:

1. Communication between frontend and backend. All data that we needed was easily transferred from API to database to webpage
2. Communication between team members was great
3. Work ethics. All team members put in a good deal of work and carried their weight
4. All tests provided good results in terms of site functionality and navigation
5. Team meetings were efficient and purposeful