CMPE 297 – Web UI Design & Development

Project Report on

"POPCORNGEEKS –

The Social Networking Site for Movie Buffs"

Submitted to:

Prof. Chandrasekar Vuppalapati

Submitted By –

Manav Pavitra Singh (009328254)
Mukhtar Yusuf (009312654)
Sahil Arora (008730033)
Suantak Niangneihoi (008719893)

Fall 2014

***Abstract* – The Internet is exploding with humungous amounts of data being shared by millions of users over the globe. Most of the information and data transmitted and shared over the World Wide Web takes place via social media. Social media platforms can be generic or focused on any particular theme. They can also be presented as websites, web apps, or even mobile apps. The PopcornGeeks project is a social networking platform initially developed as a website with potential to be extended as a mobile app. The current version of the project has features of basic social networking like maintaining one's own profile, connecting with other users, sharing and posting on one's own or other users' profiles, with the main theme being movies. The website allows movie enthusiasts to connect with people that share their similar interest in movies. Using RESTful APIs, additional features like Facebook user mode, information about latest movies and localization may be added to make the project more marketable.**

I. INTRODUCTION

There are a bunch of social media platforms and applications like Facebook, Twitter, Google+ and many more, that link people together. More specific social media like LinkedIn have also been developed, which focus on a particular area of interest i.e. professional marketing and career search.
The PopcornGeeks website focuses on a more fun aspect, movies. It has been developed as a social networking environment that allows people who love movies to come together and discuss their interests, talk about movies, and socialize as they please. Since the project is in its current stage and has a lot of scope for growth, the project has the functionality of a very basic social networking website at present. This is will however be extended to serve a larger audience by giving users the option to join and use the website using Facebook. Users can also search and scour for more updates on past and upcoming movies as well as the latest ones for more hot topics to talk about. For a more globally expanded market, localization will definitely be added to cater to a more diverse target audience.
This project documentation includes various intricacies of the project development process and explains in details the web design and development technologies used and implemented, the client side and server side design and hosting services, how the entire system functions together in a uniform, collaborative workflow, and the design and development protocols and patterns followed throughout the entire process of developing the website.

II. PROJECT DESCRIPTION

The PopcornGeeks website is hosted using the server-side web hosting plan offered by GoDaddy Inc., who also provided the domain name that was customized. The website can be visited at:
www.popcorngeeks.com
Since GoDaddy web hosting does not support Java, the web pages were created in PHP instead of HTML. However, the source code contains a variety of different languages and technologies. All the source files that represent a web page within the website, or a functionality script, all possess a .php extension. The UI code is of course in HTML5 and CSS3. The scripts for different functions are written in PHP (for the backend) and JavaScript (for the client side). The UI themes implement Twitter Bootstrap and jQuery technologies for a more visual and user-friendly experience. Since the website deploys a bunch of asynchronous functions, AJAX is widely implemented for almost every page. The server hosting plan is done in a Linux hosting environment.
The first page that pops up on a user's browser is the index page of the website, called "index.php". This is the main page of the website that any first-time visitor to the website will see. Existing users may log in to the website, after which they can access their accounts and do whatever they please as long as the terms of use are met. In case an existing user forgets a password, a new password may be requested with the help of a password generator. New users are redirected to a sign up page where they can complete the signup process and click the activation link on their email. After this, they are redirected to the website as a logged in user. From thereon, the user can explore the website and check out the different functionalities available.

III. PROJECT REQUIREMENTS

Although most of the project website consists of designing the user interface and developing it, a major part of its features stem from all the backend and middleware functionality, which adds to the website's usability. There are a bunch of requirements needed to set up an environment feasible for the development and smooth functioning of the website.

A. Front End Designing Toolkit

For building the front end, the technologies used are HTML5, CSS3, JavaScript and jQuery for the user interface and functionality of the client side design. Any web development IDE can be used. But since the system deals a lot with RESTful services and a lot of server side hosting, it is preferable to use a tool that can sync the files from the client side to the server side. For this project, Adobe Dreamweaver is used to create and edit the PHP, JavaScript and CSS source files on the local machine. The edited files are then synced to the server side via an FTP (File Transfer Protocol) mechanism.
Alternative IDEs may also be used. Dreamweaver has been used in this project to facilitate and ease the syncing of files to the server. There are also IDEs like Microsoft Web Matrix

that run the web pages on the local machine. However, it does not support embedding of PHP within HTML code. This is supported in Dreamweaver on the server Linux hosting.
For execution, of course, a web browser is required. This project has been tested mostly on Google Chrome, and occasionally on Mozilla Firefox. It should test fine on Safari as well.

B. Domain Name

Obviously to host a website on the Internet, outside the confinements of localhosts, a domain name is required to launch the website as a registered domain. Domain names are usually provided by server side web hosting companies, who normally have paid plans of different tiers. Sometimes they may provide a free domain name along with the web hosting services like a server database, website builder etc. For this project, the domain name was obtained from GoDaddy and was custom named as www.popcorngeeks.com

C. Web Hosting Plan

The source files need to be hosted on a server in order to access the website on the Internet. GoDaddy also provided a server side web hosting plan on a Linux platform along with the domain name. The web hosting plan contains everything from a phpMyAdmin panel where mysql tables can be created, to a file manager that holds all the source files on the server side.

D. Databases

On the backend, there needs to be some form of data store that holds all the information entered for every end user who accesses the website. Since it is a social networking website, users are bound to have their own profiles. Each attribute of every user profile needs to be stored for distinction and uniqueness of users.
The phpMyAdmin panel on the server side, which is essentially an admin database dashboard on the developer's side, provides for creation and maintenance of databases and data tables on the server side. These are all hosted on the server side and are accessible at the backend by the developer via the web hosting plan control panel.

E. Indigo Studio:

The user interface of the website was created with storyboards and wireframes using the Indigo Studio Infragistics tool. This tool is helpful in ascertaining the layout of the web pages and designing what the UI would be like. Indigo Studio can be downloaded either as a 30-day trial version or purchased. It works on both Windows and well as Mac OS.

IV. WEB UI REQUIREMENT PRINCIPLES – VOC, PERSONAS, JOB SHADOWING

The main requirement for the UI was to provide a consistent user experience throughout the website. Also, keeping in mind about the usability and navigability principles the UI is created to provide an easy to use, consistent user experience. It was ensured that these principles were followed as well as the functionality of website isn't compromised at any point.
Job shadowing was done to ensure that we get the targeted audience. Further user personas were made to ensure that the potential users can get what they actually want from the website. For instance, a prototype of the UI was developed and potential users were asked questions so as to know what they would like the most on the "What's New" web page.
After a lot of research and analyzing different user personas, the decision was made to show a list of movies in the theaters along with their critics score, audience score and the movie MPAA rating.

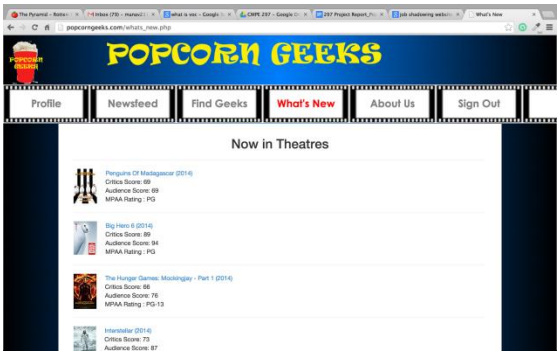


Fig.1. Screen Capture of the "What's New" webpage showing list of movies- critics score, audience score, MPAA rating

V. WEB UI DESIGN PRINCIPLES - STORYBOARDS, WIREFRAMES

A. Storyboard

The concept of the website stems from the idea that people who have a deep interest in talking and discussing about movies need a platform or forum where they can exclusively discuss their topic of interest – movies. This is why the website mainly consists of web pages that facilitate this kind of interaction through social media and digital networking. Except for the initial login page which takes only a little bit of time to log in, or the sign up process, which is also kept to a minimum, the rest of the website content caters more towards meeting the end goal of providing a social networking platform for movies for the end user.
The infrastructure of the web pages and the way they link to each other as a storyboard is better explained as part of the front end architecture and design pattern.

The topic of concentration in this section leans more towards wireframes.

B. Wireframes

The initial wireframes that were proposed at the beginning of initiation of the project development process are slightly different from the end result at present. Since the development process follows the agile methodology, the resultant system is bound to differ from the initial model proposed to be built. The initial wireframes of the website were designed using Indigo Studio and are shown in the following snapshots:

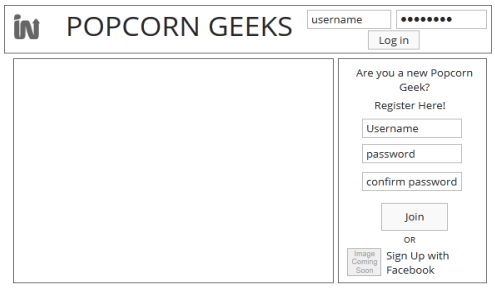


Fig.2. Wireframe for login page (index.php)

This is how the index page was initially supposed to look like. In the actual UI, new user and existing user login options are included in the same form at the center of the page right on top of an image element that spans across the background.

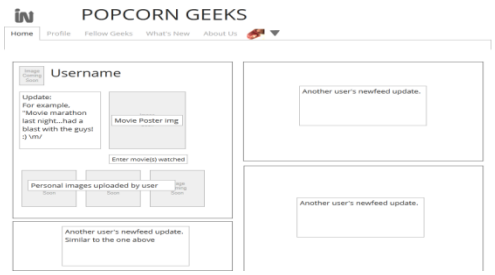


Fig.3. Wireframe for homepage after login

Fig.4. Wireframe for the user profile page

The wireframes above show two different pages for a homepage after the user logs in (which contains the newsfeed and latest posts or updates from other users), and another separate user profile for the user's personal information in regards to movies like favorites and recently watched etc. However, the actual website encompasses both these pages as one, called the user page or user.php.

Fig.5. Wireframe for the FellowGeeks page

The FellowGeeks page was initially meant to be a page where a user could maintain a list of friends or "FellowGeeks". The actual website now has a page called "Find Geeks" (search.php) to find fellow users and add them to the friend list. The list of friends are now included as part of the user profile page, user.php

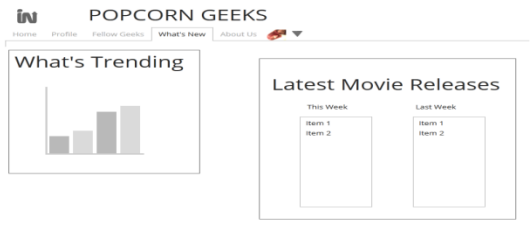


Fig.6. Wireframe for the "What's New" page

Fortunately, this is one page of the end system that has stayed consistent with respect to content. However, the structure is such that the latest movie releases for any particular week can be retrieved and shown on the website. This is done by invoking the IMDb API as part of the PHP script.

Fig.7. Wireframe for the "About Us" page

The About Us page has also remained unchanged in regards to the content. It basically contains information about the developers of the website, and a form that can be used to contact them.

VI. HIGH-LEVEL ARCHITECTURE DESIGN

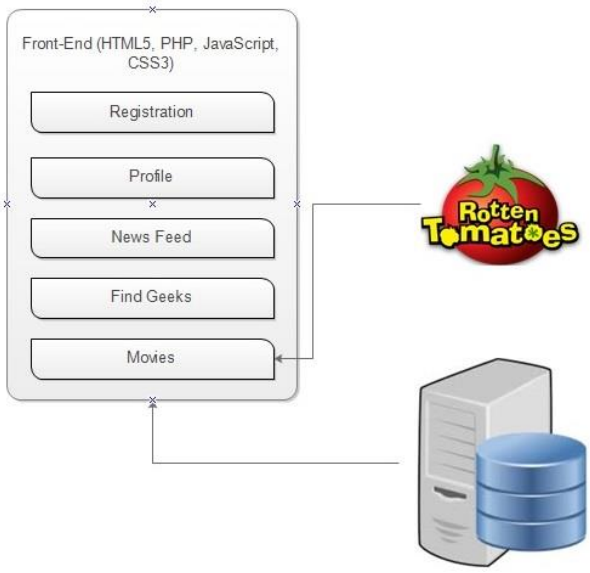


Fig.8. High Level Architecture Design

The front-end is connected to the MySQL database using PHP.
Everything from the user's information to the friends' information is stored in MySQL tables.
The movies list is accessed using the Rotten Tomatoes API.

VII. COMPONENT LEVEL DESIGN

A. Front End

Index Page: It is the homepage for the website, and it contains the scripts for logging the users in. It creates cookies and sessions for the logged in user for 30 days.

Fig.9. Screenshot of the Index Page (index.php)

Signup Page: It contains a form for a new user to register with our website. After filling up the form, an email is sent to the email address specified by the user that contains an activation link. The user's profile gets activated after clicking on that link. He can proceed by logging in to the website.

Fig.10. Screenshot of the Signup Page (signup.php)

Profile Page: After logging in, the user is directed to his profile page that has his profile picture, photo album, friend list and a status box. The user can also visit other people's profile pages, who he can add as friends, unfriend (if they're already friends) or block.

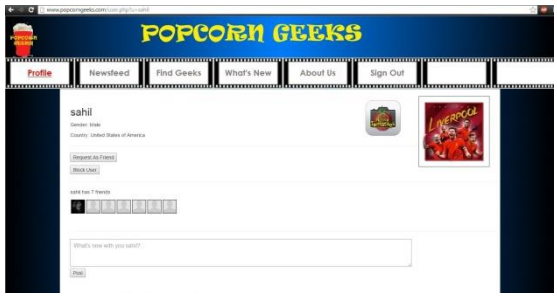


Fig.11. Screenshot of the Profile Page (user.php)

Photos Page: The user can upload photos onto his gallery or view his friends' photos on this page.

Fig.12. Screenshot of the Photos Page (photos.php)

Notifications Page: If someone sends the user a friend request or posts something on his wall, the user will see those events on the notifications or "News Feed" page.

Fig.13. Screenshot of the Newsfeed Page (notifications.php)

Find Geeks (Search) Page: The user can look for his friends on the website by searching for them with their usernames.

Fig.14. Screenshot of the Find Geeks Page (search.php)

What's New Page: This page lists all the movies that are playing in theatres at that time, along with their rotten tomatoes critics' rating.

Fig.15. Screenshot of the What's New Page (whats_new.php)

About Us Page: This page has info about the people who developed the website and there is a contact form for the user to inform the developers about any feedback for the website.

Fig.16. Screenshot of the About Us Page (about.php)

B. Backend

The back-end consists of a MySQL database, created and accessed using PHP scripts.

VIII. SEQUENCE OR WORKFLOW

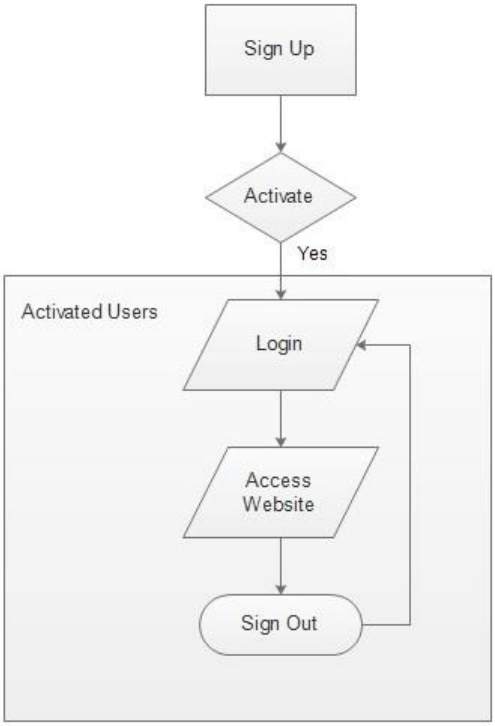


Fig.17.The Workflow Diagram

IX. HTML5 FEATURES USED

The pages of the user interface, though stored as PHP files, implement AJAX and thus contain different web development languages and scripts embedded in each. Most of the client side source code has HTML script, with referenced JS and

CSS included. They also include jQuery/Bootstrap libraries for a better UI design.
As for the HTML5 features used, all the basic HTML tags are always included, namely:

- <head>
- <title>
- <script> for including the external JS libraries
- <style> to include external CSS
- <body>
- <html> (of course)

Apart from these essentially basic tags, some new HTML5 tags are also used for a better web design structure. These include:

- **<aside> :** Used in the about_us.php UI page to put the contact form at the side of the main body, as a separate element.
- **<header> :** This contains the logo and the main heading at the top of every page. The header element for the website is included on all pages, internal and external.
- **<footer> :** The footer is also contained on all pages but have a different structure for internal pages and a different one for the external ones. The internal pages (after login) have a footer that contains links to all the other pages within the website. The external pages (before signup or login) have a different, limited footer that contains only a link to the about_us.php page which can be public. The link to the About Us page internally is different (about_us_in.php). It contains the contact form, which is not available for public view for security reasons.

The generic HTML structure for most of the pages can be shown in the following syntax:

```
<html>
  <head>
    <title></title>
    <style></style> //external CSS
    <script></script> //JS
  </head>

  <body>
          <header></header> //standard header
          <div></div> //main body
          <footer></footer> //standard footer
  </body>
</html>
```

X. INTERFACES – RESTFUL & SERVER SIDE DESIGN

RESTful APIs

Restful APIs of rottentomatoes.com were used to get all the movies data. A developer account with Rotten Tomatoes was created that provided the developer key to use their APIs. PHP script is used to get the data from their website.

- ***whats_new.php:*** This script gets the response in JSON format. The response is then decoded and data is formatted using PHP to represent the list of movies that are in theatres along with some other movies' data.

Server Side Design

The server consists of a MySQL database that houses several SQL tables.
There are different tables for storing user info, friends, photos, notifications and blocked users. The website connects to the database using PHP scripts. And all the data is posted to the PHP scripts using AJAX, enabling the website to communicate with the database.
The back-end functionality uses the following PHP references:

- Check_login_status - This script is used in most of the pages. It checks for whether there are live cookies and sessions for that user. If the user isn't logged in, he is redirected to the index page.
- Db_conx - This script connects the website to the MySQL database.
- Image_resize - This scripts runs whenever there is a photo upload. It resizes the image according the specified dimensions.
- Photo_system - This is the script that uploads the the user's picture in his folder in the database.
- Status_system - This script is used to post a status, post on a friend's wall or reply to a post.
- Friend_system - The "add friend" and "unfriend" functionalities of the website is handled by this script.
- Block_system - This script handles the "block" and "unblock" functionalities of the website.

XI. CLIENT SIDE DESIGN

The UI features implemented using HTML5 across the website are numerous. Most of the basic and general HTML5 elements are used in building the client side of the website.

A. Source Code Format

All the source files are written in HTML, including CSS, JavaScript, PHP, Bootstrap, jQuery and AJAX within and collaborating all of those technologies together to form an asynchronous, user-friendly front end. The source files are however, stored as PHP files on the server backend instead of HTML, one reason being that the GoDaddy server supports PHP but not Java/HTML. PHP scripts are also far easier to parse and include in each UI source file. All the pages that are

included as part of the website are PHP files with HTML code embedded in it. The HTML in that then of course has JavaScript, CSS and other references. There are also other PHP scripts which are not part of the UI but are used for running certain functionalities of front end objects at the backend like retrieving a request or parsing HTML code contained in another PHP script.
Some pages are formatted using the latest HTML5 tags like <aside>, <header>, <footer> and so on.

B. JavaScript References

JavaScript and JavaScript libraries are widely used across the project. They are embedded within HTML and PHP codes in <script> tags.
There are two main custom libraries:

- main.js
- ajax.js

These libraries are referenced throughout different source files across the website.
External and open-source JavaScript references to Bootstrap and jQuery libraries are also used for designing the UI using Bootstrap elements.

C. CSS References

There are two main custom CSS3 files in the project that are referenced in each of the source files pertaining to the UI, as external CSS:

- style.css
- login.css

The "style.css" file contains all the stylesheet code for each UI web page. It basically contains the UI design code for the header, footer and the main body of each page, which is referenced on practically every page.
The index.php page looks a bit different from the other pages after logging in. Logically, there are two types of web pages in the entire website, based on their appearance:

1) ***Before logging in:*** These are the pages that are visible to new users before and while they complete the signup process, as well as for users who have forgotten their passwords and request for a new one. These pages reference both the style.css as well as the login.css for their UI design, the reason being that even though their primary CSS is contained in login.css (since these pages are all part of the login process), the design for the background and main logo with the header are all contained in style.css. The pages under this category include:
    - index.php
    - signup.php
    - forgot_pass.php
    - login.php
2) ***After logging in:*** These pages are the ones visible to an existing user after logging in or even to new users who log in for the first time. Only the style.css is included for these pages. But since certain pages may also refer to some of the login page elements, login.css is also included just to be safe. It is not parsed if there are no elements or objects in that file with an ID that call the CSS. These pages include:
    - user.php (Profile tab)
    - notifications.php (Newsfeed tab)
    - search.php (Find Geeks tab)
    - whats_new.php (What's New tab)
    - about_us.php (About Us tab)

D. PHP References

There are certain PHP files which contain only scripts that are to be parsed by the pages containing UI code. These PHP referenced files are listed as follows:

- template_country_list.php
- template_status.php

These different PHP files are not part of the UI but are vital in performing functions related to the UI elements. These functions deploy the necessary collaborations that need to be made between the front end and the backend.

E. Navbar

The navbar is consistent across all web pages after the user has logged in. It consists of the different tabs to inter-navigate between pages. The navbar here is not designed using Bootstrap. Instead, it is a manually designed element with embedded images of filmstrips to form a "movie" like aesthetic visualization for the website. Since the navbar here helps navigate across the internal pages, it is not available on the external web pages, i.e. the pages seen before login or sign up.

F. Header and Footer

The <header> element essentially consists of the logo and the heading of the website. The logo has been designed manually using Adobe Photoshop and Adobe Fireworks. The original image was manually clicked and edited using those photography and picture tools. The heading and the logo are placed as part of the header and are included in every page of the website.
After user login, within the internal pages, the logo also acts as a button which redirects to the user's homepage i.e. profile page. For the external pages, the logo redirects to index.php when clicked on.
The <footer> element is also part of each page but different for internal and external pages. Since the footer consists of links that navigate to different pages within the website, it is

not a good security measure to have a standard footer for all pages. To resolve this issue, there are two footers created. The footer with links to all the web pages is used only within the internal pages. As for the external pages, the footer contains only the link to about_us.php. The standard copyrights are, of course, included on all pages. The About Us page that the external pages link to is also different from the About Us page that the internal pages link to. The internal About Us page is also called about_us_in.php, which has the contact form that is unavailable to the external pages.

XII. TESTING (UI OR STRESS TEST)

UI Testing

The main purpose of UI testing is to ensure that the GUI of the website responds as expected by the user. For our project, we've decided to focus on navigation between the webpages of our website. This is very important because certain pages are only accessible to logged in users, and it would be inappropriate to have users who are logged out accessing such pages, or even worse, users that aren't even registered on the site at all. We approach our UI testing using a manual, model-based approach in which a di-graph is constructed to represent the expected behavior of the website in terms of page navigation. Test cases are then constructed by producing linked lists from the nodes in the graph. For each test case, we specify the expected result, perform the test manually, and then record the actual result. The test case passes if the actual result matches the expected result and fails if the actual result is different from the expected result. The pages that would be tested are the following:

- Index page
- Sign up page
- Profile page
- Photos page
- Search page
- What's new page
- About us page

The model is given below.

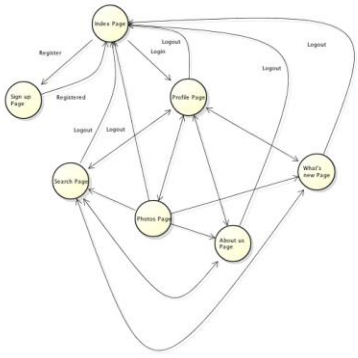


Fig.18. UI Testing Model - GUI Transition Graph

Test Cases

*Please refer to Table.1. at the end of the report

Markup Validation

When referenced and tested on the open-source markup validation testing tool online: http://validator.w3.org/,
The website showed the following results:

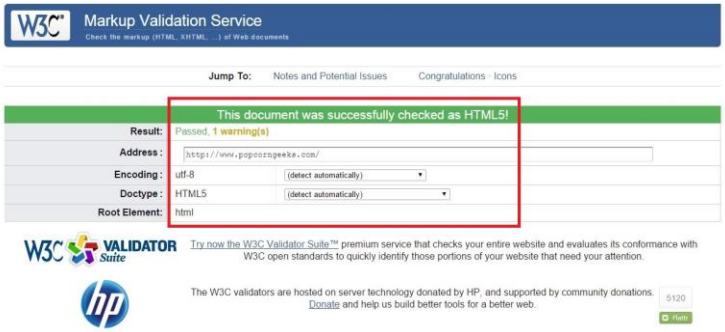


Fig.19. Screen Capture of the HTML Markup Validation Result

The result was good with no errors, and just one warning which was not of major importance:

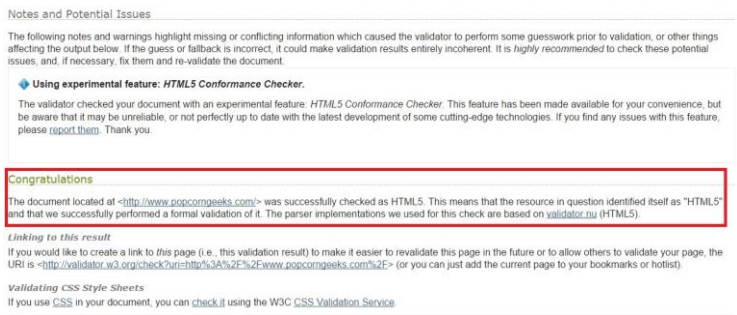


Fig.20. Screen Capture of the HTML Markup Validation Result with Warning Message

XIII. CROSS BROWSER COMPATIBILITY

The website has been tested for different browsers on various platforms. In terms of functionality, it works fine on all platforms, both desktop and mobile. However, in terms of visualization, it is only compatible on desktop browsers. It still runs on mobile browsers but does not look as good as it should. This is an ongoing work in progress part of the project. Currently, the visual mobile compatibility is a future scope. The different outcomes across different browsers can be shown in the following screen captures:

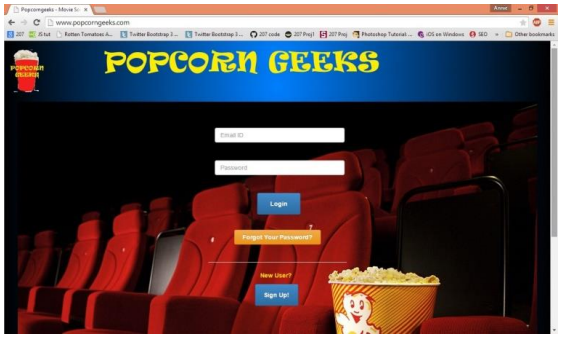


Fig.21. Screen Capture of the website on Google Chrome

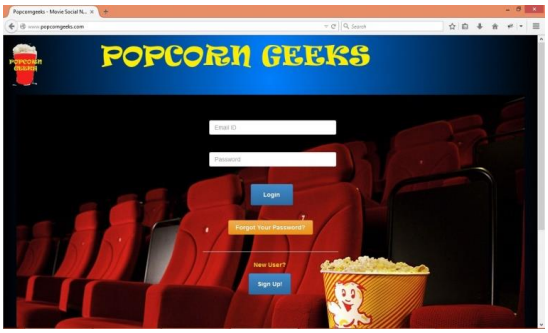


Fig.22. Screen Capture of the website on Mozilla Firefox

Fig.23. Screen Capture of the website on Internet Explorer

As observable in the screenshots above, the website runs perfectly fine on all popular desktop browsers.

XIV. JAVASCRIPT LIBRARIES

A. jQuery.js

It is included to implement bootstrap, since bootstrap is only an extension of all the functionality that jQuery offers.

B. Bootstrap.js

Bootstrap is an HTML, CSS and JavaScript framework used to make responsive websites. It contains an extensive documentation of common HTML features and UI elements. It scales the website efficiently and effectively. Also, bootstrap is open source, so we can exploit even deeper functionality.
For our website, we used various bootstrap features like data-toggle, panel, buttons, modals etc.

C. Main.js (Custom made)

We created this library in order to make the code look more efficient. The most commonly used javascript element is "document.getElementById". We created a library and replaced that with a "_". It not only makes the code look cleaner but also makes life easy for the developer.

D. Ajax.js (Custom made)

Ajax is used extensively throughout the whole website. Instead of writing the whole code for creating a new XMLHttpRequest every time, we simply created a library that makes the code look better and also eliminates the need to write tens of lines of code.

XV. PAGINATION

Pagination patterns for this website are performed in a different way than the conventional patterns. There are two media for pagination across the different web pages – one being the "navbar" that contains all the tabs to different pages, and the other one being the footer that contains links to each other page. Pagination patterns can be seen only within the internal web pages.

Pagination via the navbar can be shown below:

Fig.24. Screen Capture of the Navbar

Pagination can also be done via the footer as shown below:

Fig.25. Screen Capture of the Footer

XVI. SEARCH ENGINE OPTIMIZATION

It's a technique which helps search engines find and rank your site higher than the millions of other sites in response to a search query.

Basically search engines are text-driven they crawl the Web to see what is there which is performed by a software, called a crawler or a spider (or Googlebot, as is the case with Google) After a page is crawled, the next step is to index its content. The indexed page is stored in a giant database, from where it can be retrieved later. In identification the words and expressions that best describe the page are assigned to particular keywords

Sometimes they might not get the meaning of a page right but if you help them by optimizing it, it will be easier for them to classify your pages correctly and for you – to get higher rankings.

When a search request comes, the search engine processes it – i.e. it compares the search string in the search request with the indexed pages in the database. Since it is likely that more than one page (practically it is millions of pages) contains the search string, the search engine starts calculating the relevancy of each of the pages in its index with the search string.

There are many algorithms to calculate relevancy. All these algorithms have different relative weights for common factors like keyword density (number of times a keyword is used), links, or metatags. That is why different search engines provide different search results pages for the same search string.

The last step in search engines' activity is retrieving the results. It is nothing but simply displaying them in browser – i.e. the endless pages of search results that are sorted from the most relevant to the least relevant sites.

Search Engine optimization is a very important aspect for getting traffic. As popcorngeeks.com is a social networking website, getting more number of users can only be possible if the search results for the keywords get the website to a considerable rank. Since, SEO takes time it will be done progressively. Though some on site SEO has been performed by using meta tags, title tag, header tags. As we see in the screen capture below when we used the keywords "popcorn geeks", the website came on fourth page of google search engine.

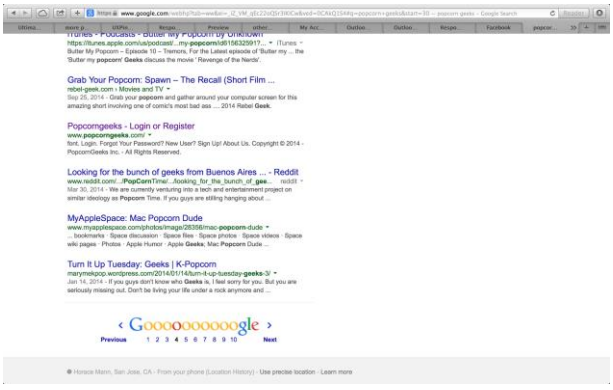


Fig.26. Screen Capture of Search Engine Results when Keywords used- popcorn geeks

And, upon using the the keywords "popcorn geeks movie networking" it came on the first page of google.

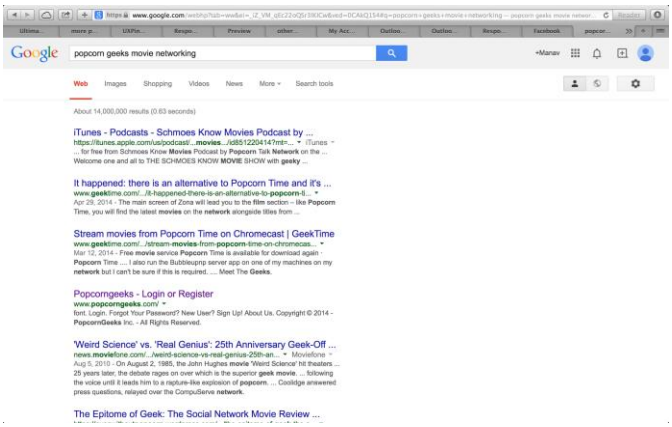


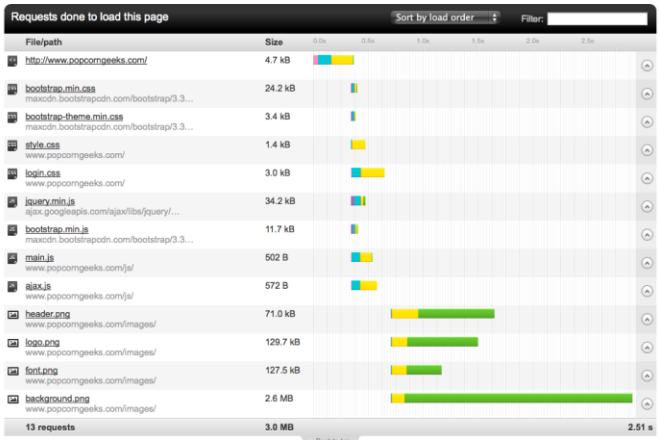
Fig.27. Screen Capture of Search Engine Results when Keywords used- popcorn geeks movie networking

XVII. PROFILING

Profiling is generally used to dynamically analyze the performance of a software or website usually including measurements like space/time complexity, speed, tracking of function calls (frequency and duration), etc. For our project however, we focused on the details of the relationship between load speed, size of files, and number of requests made, as well as DNS health. We selected pingdom tools as our profiler as it's one of the best, easy to use, and free profilers out there. Screenshots of the results are shown below.

No. of Requests and time taken

In terms of no. of requests and time taken, the website takes about 2.5s to load after 13 requests are made.



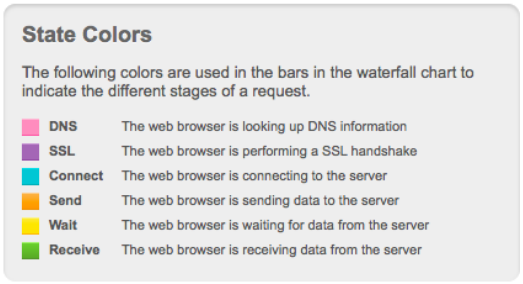


Fig.28. Profiling based on number of requests and time taken

Performance Grade

The performance score for the website is about 85%. One way in which this can be drastically improved is to use browser caching more effectively. From the diagram below, it is seen that the website's score for the leverage of browser caching is pretty low.

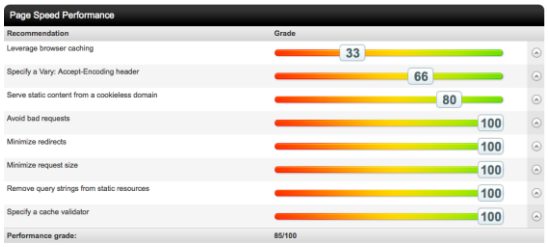


Fig.29. Profiling based on Performance Grade

Analysis

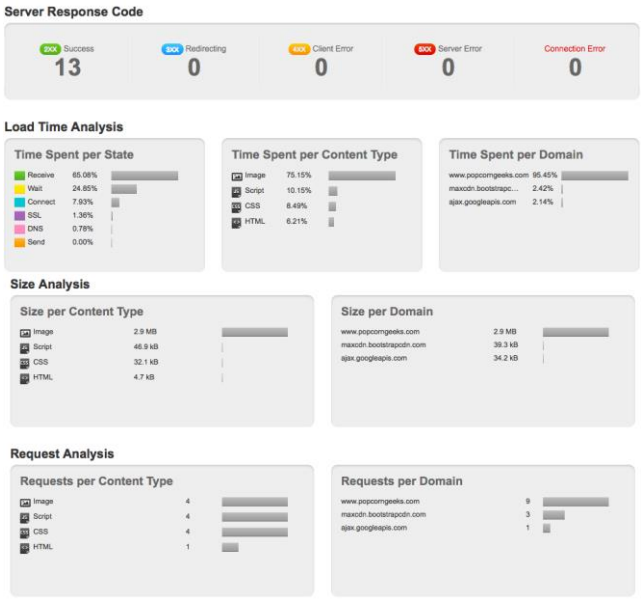


Fig.28. Profiling based on Analysis

DNS Health

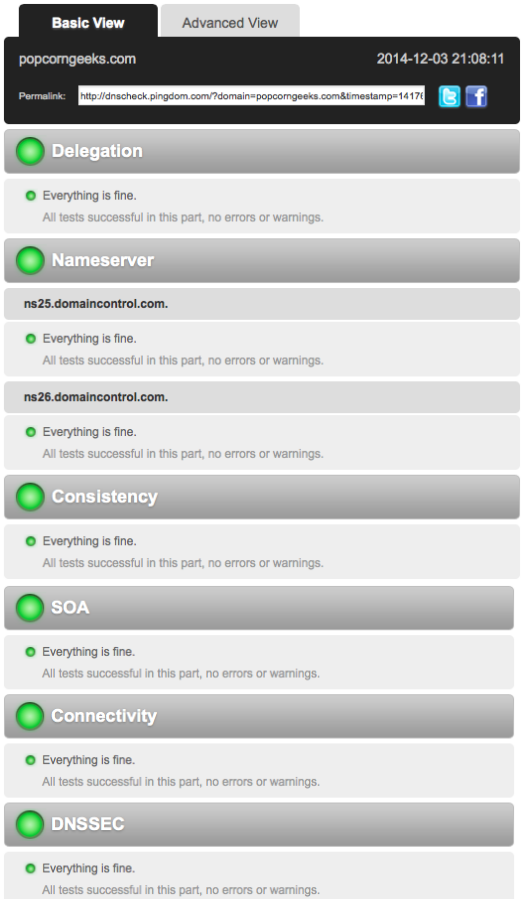


Fig.29. Profiling based on DNS Health

XVIII. CONCLUSION

In this project, we not only applied what was learned in class, but also extra material in web and UI development (especially in building a social network). We went from generating an idea that taps into one of the major trends in technology today (social networking), and combined it with one of the major sectors of the entertainment industry (movies), all laying together a very solid foundation for a successful project. We then performed requirements gathering using personas and job shadowing, moved on to design using storyboards, flowcharts, sketching, wireframes, and mock-ups, then implemented the design using the new HTML5 learned in class, CSS, PHP, JavaScript, SQL, and AJAX. We also learned testing, and performed a model-based UI test for the website. With all this, we've been able to come up with a basic, fully functional social networking site for movies, but we intend to take it beyond the classroom and turn it into a fully-fledged social network that delivers a unique and compelling user experience.

Test Case ID	Test Case Name	Preconditions	Input	Expected Results	Actual Results	Pass/Fail
1	Navigation to Profile Page from Index Page	Logged in	Email, password, click on login button	Navigation to profile page with profile details	Navigation to profile page with profile details	Pass
2	Navigation to About Us Page from Index Page	None	Enter About Us URL in address bar	Redirection to log in page for login	Redirection to login page for login	Pass
3	Navigation to Sign Up Page from Index Page	None	Click on Sign Up button	Redirection to Sign Up Page for registration	Redirection to Sign Up page for registration	Pass
4	Navigation to Photos Page from own profile	Logged In	Login, click on photos icon on profile	Redirection to photos page allowing the upload of photos	Redirection to photos page allowing the upload of photos	Pass
5	Navigation to Photos Page from profile of someone else	None	Search other user, click username, click photos icon on profile	Redirection to photos page not allowing the upload of photos	Redirection to photos page not allowing the upload of photos	Pass
6	Navigation to news feed page from profile page	Logged In	Login, click on news feed link in header	Redirection to news feed page showing notifications and friend requests	Redirection to news feed page showing notifications and friend requests	Pass
7	Navigation to what's new page from profile page	Logged In	Click on what's new link in header	Redirection to what's new page showing a list of latest movies	Redirection to what's new page showing a list of latest movies	Pass

Table.1. Test Cases for the website *

WORK DISTRIBUTION

Project Work

Manav Singh – RESTful API using Rotten Tomatoes

Mukhtar Yusuf – Tried Facebook API integration. Currently unable to sync with our MySQL database on server side, hence put as future scope. Also worked on QA, testing and profiling of website.

Sahil Arora – Backend scripting using PHP. Integrated HTML with AJAX, JavaScript, jQuery. Managed and created backend databases. Added backend functionality with respect to front end.

Suantak Niangneihoi – Designed UI for all pages using HTML5, CSS3, Twitter Bootstrap, jQuery. Front end scripting with JS. Made logo and layout for web pages.

Project Report

Manav Singh - Web UI Requirement Principles – VOC, Personas, Job Shadowing, interfaces (RESTful API), SEO

Mukhtar Yusuf – Testing, Profiling

Sahil Arora – High level architecture design, component level design, sequence/workflow, interfaces (server side design), JavaScript libraries

Suantak Niangneihoi - Abstract, Introduction, Project Description, Web UI Design Principles, HTML% features used, client side design, cross browser compatibility, pagination