RAD Development Project

Duy P. Minh Khoa V. Nicholas B.

Contents

[Initial meeting 4](#_Toc89073225)

[Tools 4](#_Toc89073226)

[Communication Avenues 4](#_Toc89073227)

[Doc standardization 4](#_Toc89073228)

[Responsive vs Adaptive 4](#_Toc89073229)

[Framework usage 4](#_Toc89073230)

[Task assignment 4](#_Toc89073231)

[Adaptive VS Responsive Report 5](#_Toc89073232)

[Responsive 5](#_Toc89073233)

[Description 5](#_Toc89073234)

[Benefits 5](#_Toc89073235)

[Drawbacks 5](#_Toc89073236)

[Testing 5](#_Toc89073237)

[Adaptive 5](#_Toc89073238)

[Description 5](#_Toc89073239)

[Benefits 5](#_Toc89073240)

[Drawbacks 6](#_Toc89073241)

[Testing 6](#_Toc89073242)

[Our choice 6](#_Toc89073243)

[Using a framework 6](#_Toc89073244)

[Analysis 7](#_Toc89073245)

[Cite business rules for software development 7](#_Toc89073246)

[CITE Managed Services Quality Insurance 7](#_Toc89073247)

[What is software Quality Assurance 7](#_Toc89073248)

[software Verification 7](#_Toc89073249)

[Software Validation 7](#_Toc89073250)

[Verification vs validation 7](#_Toc89073251)

[Requirements 8](#_Toc89073252)

[Meeting summary nov 8 8](#_Toc89073253)

[Code for modularity 9](#_Toc89073254)

[Extensibility 9](#_Toc89073255)

[Comments, version control and future code maintainence 9](#_Toc89073256)

[Styling Changes 9](#_Toc89073257)

[Styling comments 12](#_Toc89073258)

[Kanban in progress 13](#_Toc89073259)

[Sprint Two 14](#_Toc89073260)

[Performance Report 14](#_Toc89073261)

[Evaluation plan 14](#_Toc89073262)

[Testing Plan 15](#_Toc89073263)

[Testing for sprint two 16](#_Toc89073264)

[Optimization Report 20](#_Toc89073265)

[Testing PLan 21](#_Toc89073266)

[Testing for Sprint three 23](#_Toc89073267)

[Handover 26](#_Toc89073268)

[WCAG REquirements 26](#_Toc89073269)

[Principle 1 – Perceivable 26](#_Toc89073270)

[Operable 27](#_Toc89073271)

[Understandable 29](#_Toc89073272)

[Robust 30](#_Toc89073273)

[Software Review Report 30](#_Toc89073274)

[Quality Assurance & Control 30](#_Toc89073275)

[Code Testing 30](#_Toc89073276)

[Support for future development 31](#_Toc89073277)

[Provide a mapping of the user requirement to the final software product (did you meet the clients expectations and requirements). 31](#_Toc89073278)

# Initial meeting

An initial meeting was held between team members to discuss the Movie Manager Website project on November 1. These topics were discussed:

1. Tools that will be used
2. Communication avenues
3. Docs standardization
4. Results of Responsive VS Adaptive report and preferred methodology
5. Framework usage & quick training
6. Task assignment

## Tools

We discussed the tools we will be using to develop the application during our RAD development cycle. We will be using the provided webserver for deployment and final testing. Individually, we will use a preferred testing tool like XAMPP or USBWebServer. Team members will use our preferred code editors/IDEs.

## Communication Avenues

We decided on using the project GitHub as the primary avenue of communication. We will use the new *GitHub Discussions* section to do this. Members will post in the main discussion thread to communicate.

Using a single program to handle most aspects of the project should help keep information in one place for documentation purposes.

## Doc standardization

We selected the Microsoft Word style Theme “Shaded” for our docs for this project. All docs for the project should use this theme.

## Responsive vs Adaptive

We discussed the outcome of the Adaptive VS Responsive report and collaboratively decided on a conclusion for the report. We decided to use the Responsive methodology with the help of the framework “Bootstrap 4”.

## Framework usage

We ran a quick demo to ensure we were all familiar with basic usage of the framework, showing styling and script setup using Bootstrap’s Content Delivery Network.

## Task assignment

We reviewed the project Kanban board, created using GitHub projects. We distributed the remaining tasks between team members:

* ‘Record suitable testing for the project’
* ‘Adapt prototype to use responsive techniques’
  + Search for movie page
  + Table of search results
  + Graph of top 10 movies

# Adaptive VS Responsive Report

## Responsive

### Description

Responsive websites respond to the user device’s screen layout and size and change the layout of the elements on the webpage automatically. They differ from the alternative “adaptive” approach, where the appropriate UI layout is chosen from a selection of custom-made designs for each screen size.

### Benefits

Responsive layouts are valued because they can produce a functioning and screen-fitting webpage design even on unusual, fringe case screen sizes. When executed properly, websites made this way can look as though they were designed especially for the user’s device. The responsive approach also has the advantage of requiring less individual designs be created by the UI designer.

### Drawbacks

The natural issue with responsive layouts is that on unusual screen sizes, elements can arrange themselves in unintuitive and unintended ways, and the UI may become unintuitive or inaccessible.

### Testing

A responsive UI design needs to be tested on many different screen sizes to ensure proper functionality and layout. Resolving issues that arise from using a responsive layout can prove difficult without

## Adaptive

### Description

Adaptive Design uses static layouts based on breakpoints that don’t respond once they’re initially loaded. Adaptive web design is different from responsive design in that there isn’t one layout that always change but instead there are serval distinct layouts for multiple screen sizes.

### Benefits

Adaptive is easier when it comes to efforts used while coding but adaptive design is not the best when it comes to performance. Adaptive site is easier to create and it is perfect for smaller size business and audience since it would keep the slower load times or lower flexibility from being an issue.

### Drawbacks

Adaptive design offers design which is suited to all the screen size wherefore it makes the loading very slow as it loads all the possible screen layouts for the development. Time and resources consumed in loading the layout makes adaptive websites slower to load which in return makes them less SEO friendly. In order to get high performance from adoptive web design you need to have a wide expanse of research done on devices and this won’t end here as with the advent of every new gadget launch you need to start the development again.

### Testing

Beside testing each CSS files with the corresponding layout there isn’t much testing to be done with adaptive design but adaptive design demand a lot of attention with maintenance that no one have found a decent solution to yet.

## Our choice

We have selected the ‘Responsive’ approach for adapting our website prototype to a multi-platform environment.

Our reasoning for this is as follows:

* Sources indicate that the responsive approach results in a faster-loading website, which is more in line with Search Engine Optimization (SEO) principles.
* Responsive is the more popular approach. We would like to use the industry standard approaches where possible, so responsive is the ideal choice.
* Responsive is more flexible to more different device sizes, which is ideal for deployment on any possible target device.

We would like to apply the framework “Bootstrap 4” to provide standardized formatting for the webpage. Bootstrap will assist in maintaining a standard layout and formatting across all pages, and allow us to create a website that is guaranteed to work on most layouts.

## Using a framework

A responsive web UI design can be achieved easily with a framework like Bootstrap, where elements automatically rearrange based on their proximity to a specific screen size. Bootstrap picks a design from several made by the designer and applies it to the target device. It will apply the design that most closely matches the target device’s screen size. In this way, it combines the responsive approach and the alternative adaptive approach.

# Analysis

## Cite business rules for software development

The following standards apply to the quality of the code itself rather than the quality of the product, which is handled separately. This section will be relevant to the client if they would like us to maintain the product in the future, or would like another actor to maintain the product.

Our software at CITE is developed in accordance with the set of ISO standards we use. These are *“ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes”* for the foreseeable scope of this project.

In addition to these standards, we also follow these best practices to the best of our ability when developing code.

* Informative comments where necessary for maintainability.
* Proper indentation and formatting in accordance with the standard of each of the languages in use.
* Small code size is pursued where possible, and duplicate code kept to a minimum.

By following these standards, we also hope to improve our ability to upscale the product with new features at the client’s request.

## CITE Managed Services Quality Insurance

### What is software Quality Assurance

Software quality assurance is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standards. Software quality assurance incorporates all software development process starting from defining requirements to coding until release. Its prime goal is to ensure quality. The whole process of software quality assurance revolves around two concepts, verification and validation.

### software Verification

Verification is the process of checking or verifying the credentials, data or information to confirm their credibility and accuracy. In software engineering, verification is defined as the process of evaluating software products, to ensure that the development phase is being carried out accurately. It’s performed during the ongoing phase of software development to ensure the detection of defects and faults at an early stage of the life cycle and to determine whether the initial specifications are satisfied.

### Software Validation

Software validation is a process of evaluating software product, to ensure that the software meets the pre-defined and specified business requirements as well as the end users/customers’ demands and expectations. It is performed with the intent to check that the developed software is built as per decided requirements specifications and if it caters to fulfil the customers’ actual needs in the real environment.

### Verification vs validation

Verification refers to the assessment and evaluation of the process or approach while validation is about examination of the developed software product to ensure the fulfilment of the pre-defined and specified requirements, such as software specifications.

|  |  |
| --- | --- |
| Verification | validation |
| Verification evaluates the ongoing development phase and the software product | Validation involves the examination of finally developed software product with respect to pre-defined specified requirements |
| The motive is to check the path and progress of the software development at each stage of the life cycle to ensure the incorporation of all the requirements in the software product | The purpose of carrying out the validation process is to determine whether the final product achieved the requirements that were stated prior to the development or it may need improvement. |
| It is performed over software product which is under the development stage | It takes on final software product, produced after the completion of the development process. |
| Includes static activities such as: reviews, inspections, walkthrough and meetings | Include dynamic testing techniques such as: black box testing, white box testing, grey box testing and acceptance testing. |

## Requirements

The client business is known as *Acme Entertainment Pty Ltd,* or simply *Acme* for the purposes of this report.

The requirements for the project have been reported to us by the Acme liason as follows:

“Acme Entertainment have commissioned a prototype movie database, however they want to review and update this application so it can be used across all the major digital platforms. They require a Multi-Platform Report on the merits of the two design options currently used; adaptive and responsive. Your team must choose the best option and rework the prototype; this modified version is to be presented at the Sprint One assessment point by the first Scrum Master. The development or migration of the movie database can be hosted on the cloud or suitable local server. Ensure your Lecturer is advised on this decision. Conduct and record suitable testing of the completed sprint one development, include this information in the Testing Plan.”

To summarise, Acme has expressed interest in our movie database program prototype. For the first sprint of this project we will be adapting our previously made prototype to work with our choice of responsive or adaptive layout technology, and present this new version to the client for review. Further requirements will be discussed on completion of this first phase.

# Meeting summary nov 8

A second meeting was held on November 8. We discussed the following topics:

1. Code modularity
2. Extensibility of the program
3. Commenting practice
4. Version control
5. Efforts to ensure future maintainability

## Code for modularity

We discussed how the program can be made modular during development by ensuring individual components can be easily decoupled from each other and used individually. We concluded that modularity is not necessarily a major concern, but we will keep avenues open by keeping an eye on coupling between modules, in case requirements change in the future.

## Extensibility

We discussed extensibility in our program. We concluded that the use of responsive technology would aid extensibility by allowing elements to be added and removed easily without upsetting the UI design.

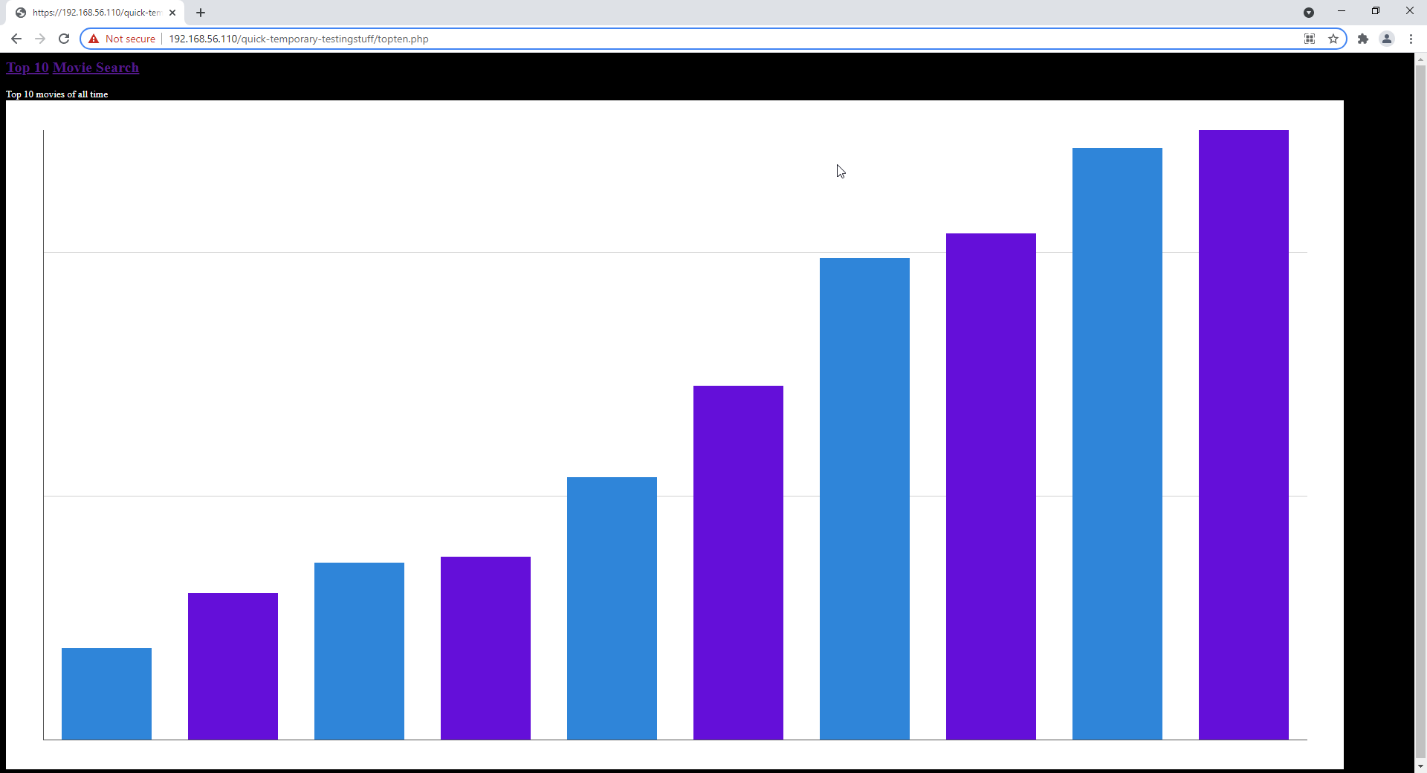
## Comments, version control and future code maintainence

We discussed the role of comments in ensuring the maintainability of our code. We resolved to provide proper comments for all PHP code in the program. Markup code in HTML and CSS will only be commented when particularly hard to understand.

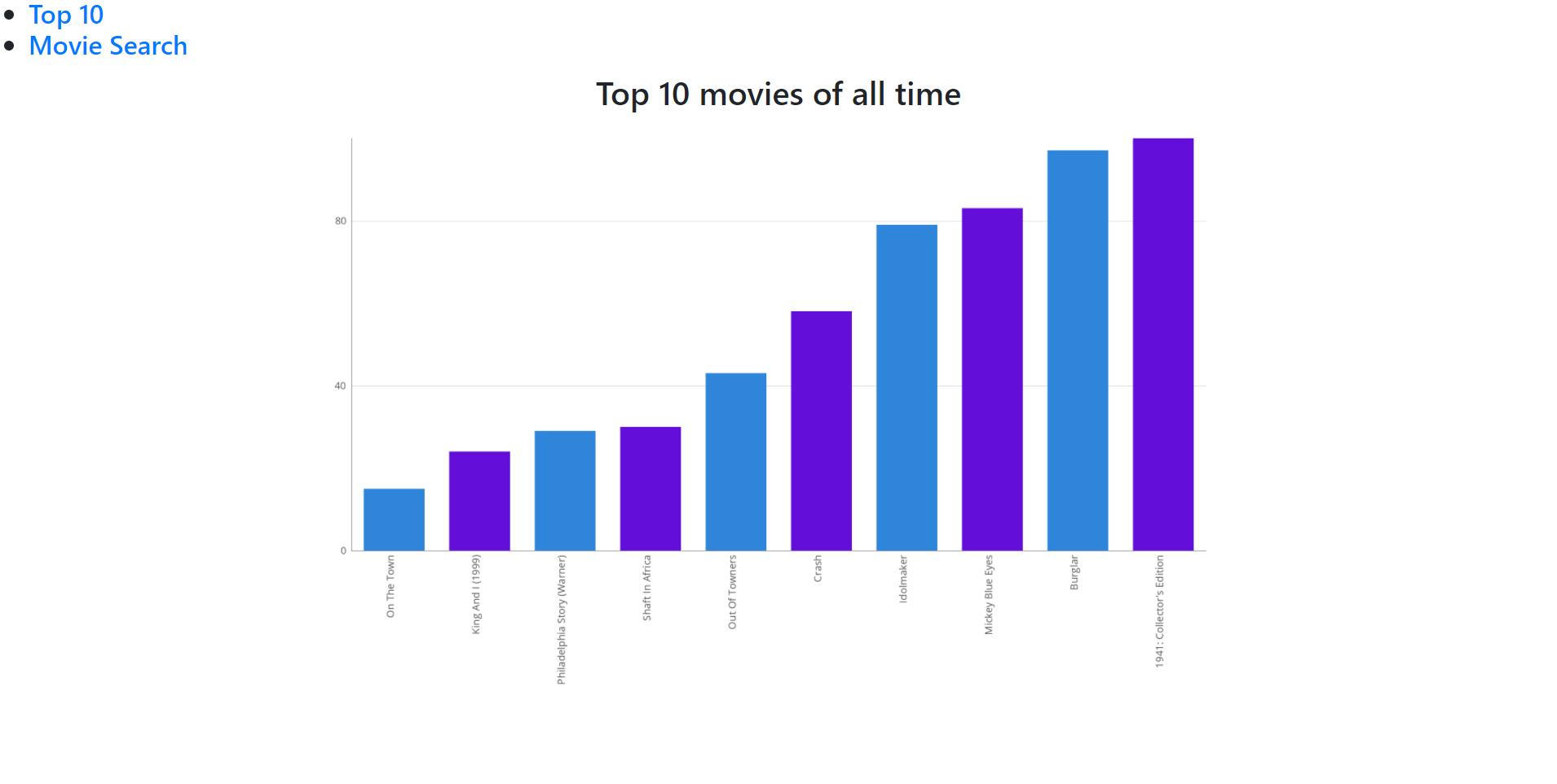
We will control versions by creating a copy of the source files at each milestone. We will store each version on local files, plus a copy will be submitted to cloud storage at the Blackboard website.

# Styling Changes

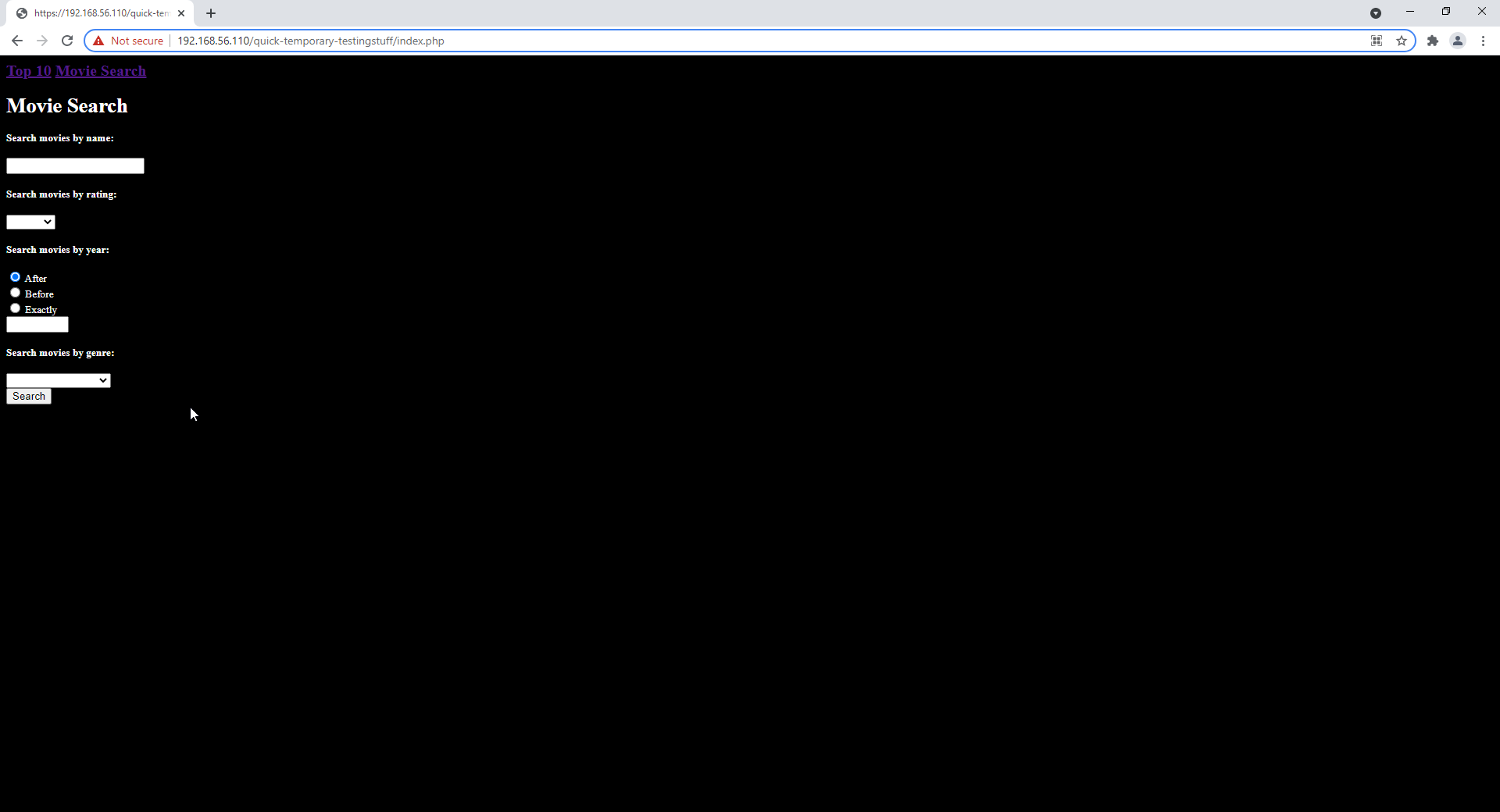
In sprint 1, styling was changed from the initial prototype to a new style. The style review came about during transition to use a Responsive website layout. This section will demo the changes.



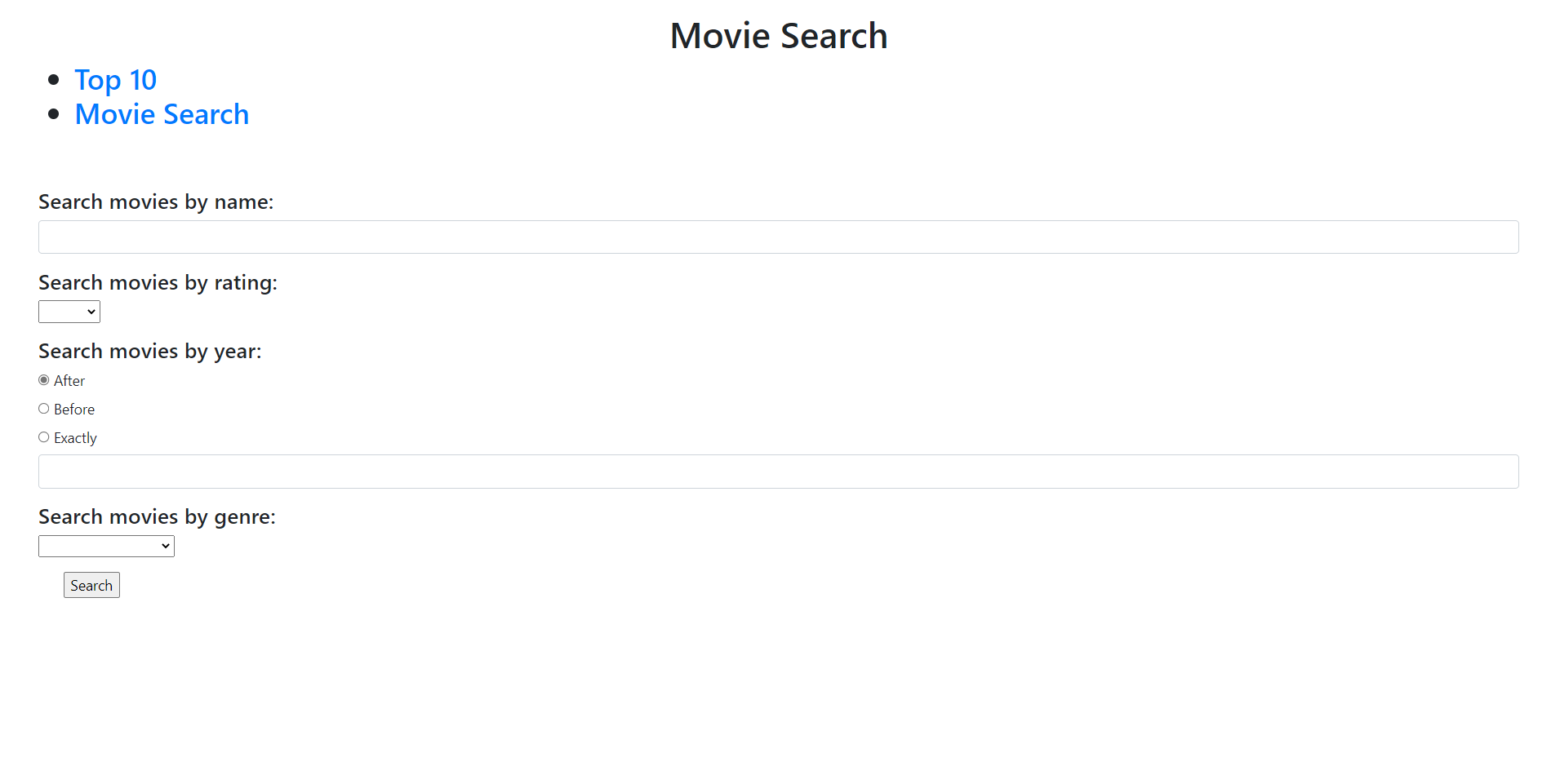
1. Original styling for Graph



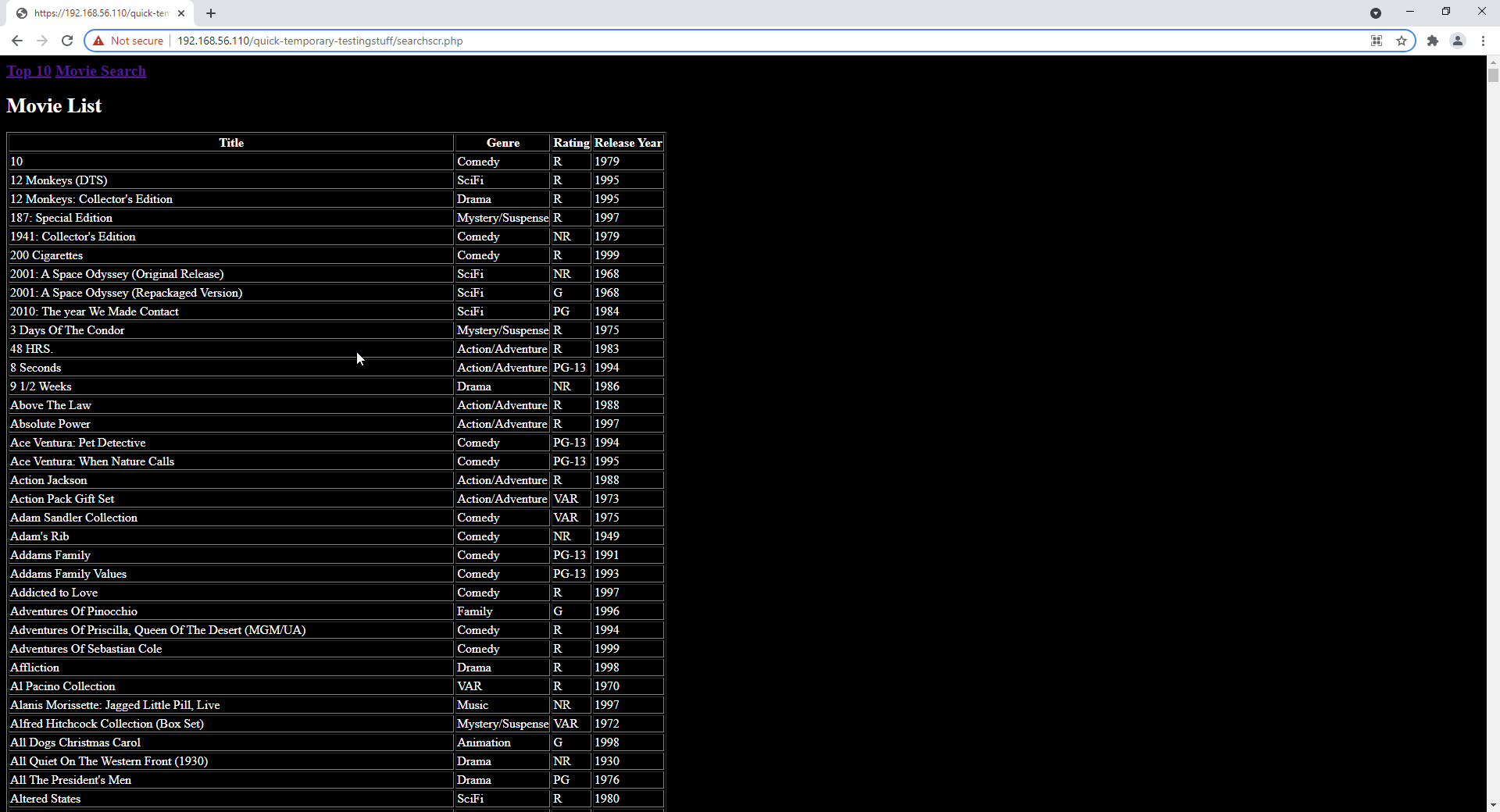
2. New Styling for Graph



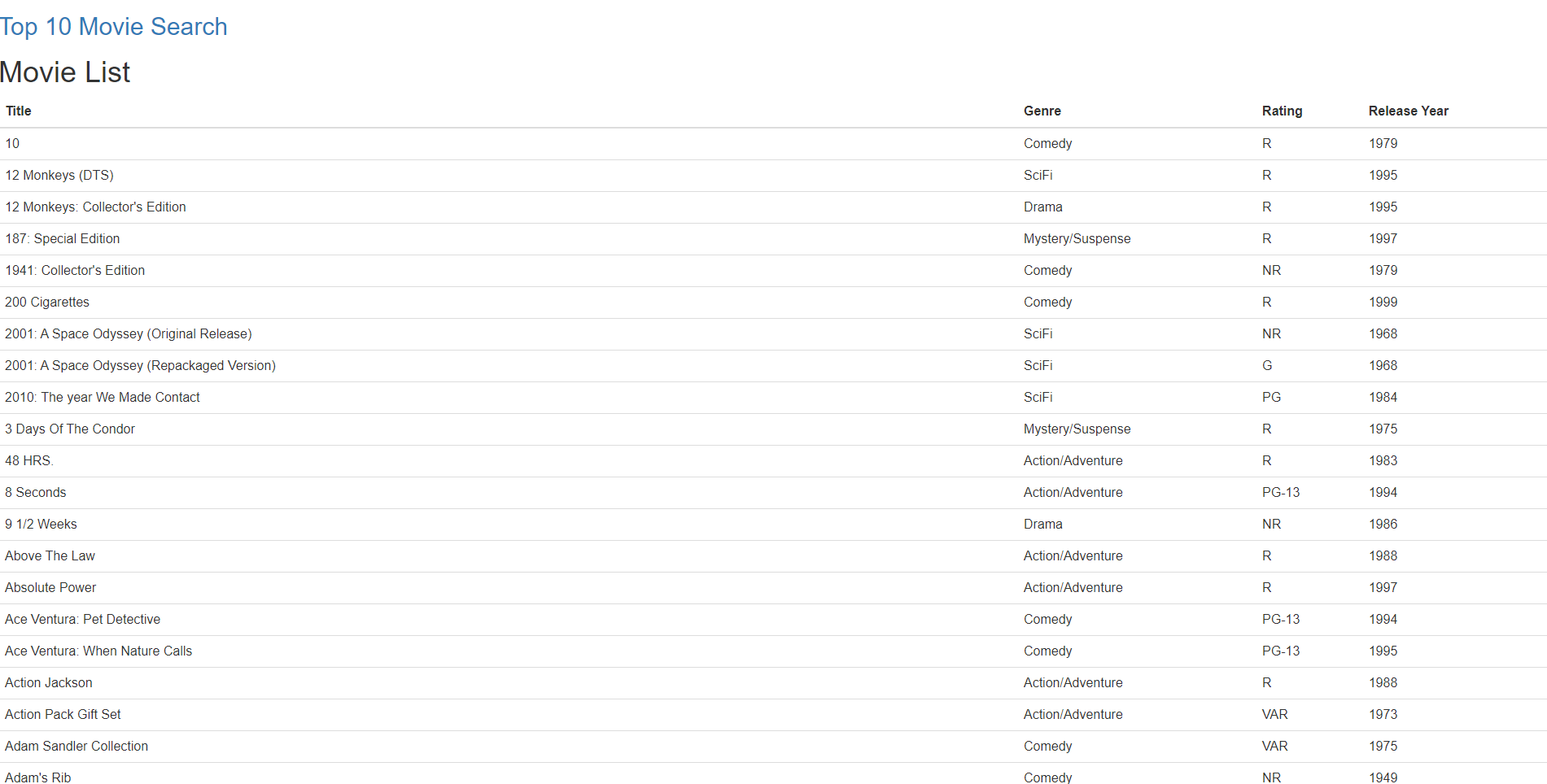
3. Original styling for search page



4. New styling for search page



5. Original styling for search results page



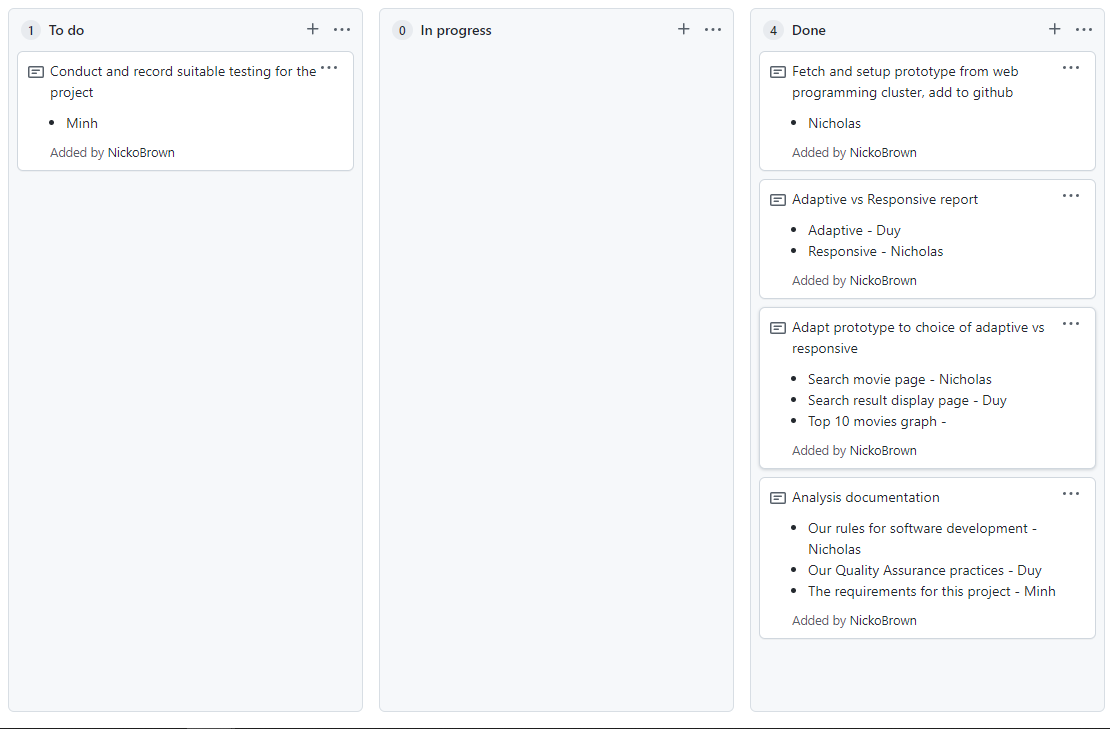
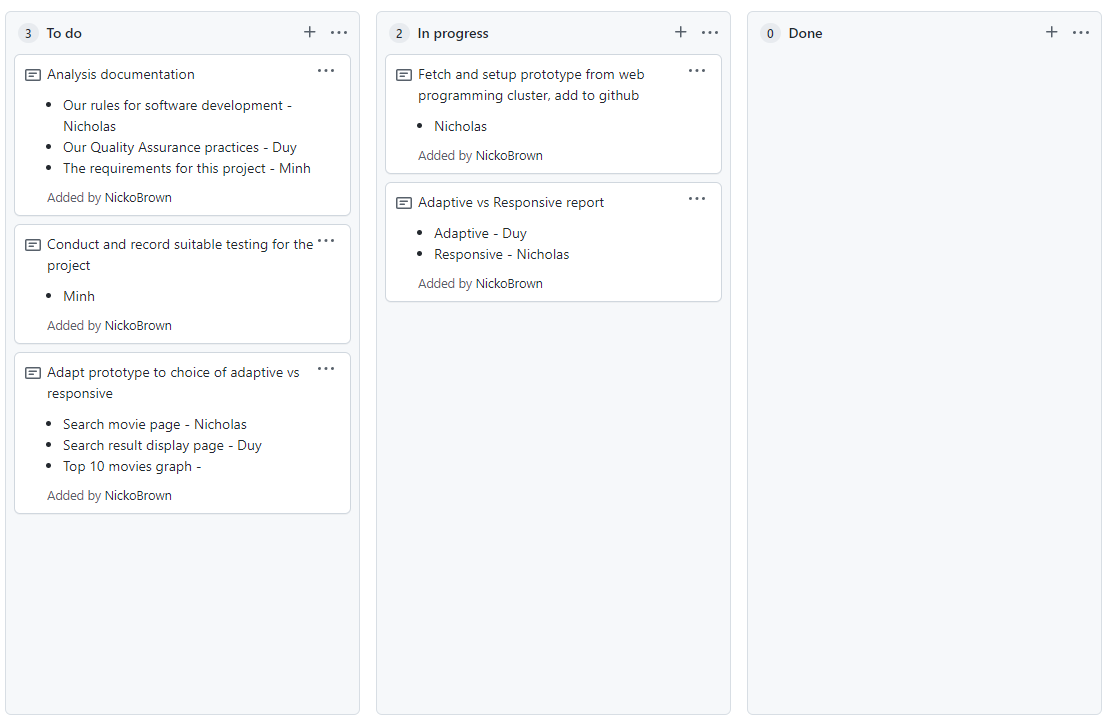
6. New styling for search results page

## Styling comments

The new style improves on the layout and visual pleasantness of the pages. The new style is still subject to change and may include more complex colors and fonts to match user requirements.

# Kanban in progress

We are using a Kanban board provided by GitHub for project management purposes. We are using the board as shown in these images:



# Sprint Two

## Performance Report

We reviewed appropriate code optimizers and performance tools for use in optimizing the movie database project.

Our project uses HTML, PHP and SQL code. Optimizations are only feasible in the PHP and SQL portion of our program, as HTML is a markup language.

For our PHP code, we are looking to find more performant ways to complete repeated operations, like displaying table rows. For our SQL, we are looking to reduce the total number of SQL calls in our program where possible. These steps can be completed manually independent of optimization tools.

Various optimization tools are available for PHP code. They work using various tricks and methods to serve requests faster. Such tools are referred to as PHP Accelerators. We may consider using an accelerator if we encounter performance issues with our webpage during testing. If we choose to employ an accelerator, we will have to install it on our server and set it up properly.

**Alternate PHP Cache** is a popular PHP accelerator notable for being used and maintained by Facebook. We would have to install this tool if we would like to use it.

**Zend OpCache** is an accelerator packaged with PHP version 5.5 and onwards. We should be able to use this tool with some setup to accelerate our webpage.

From research on this topic, it seems as though Zend OpCache is the preferable choice to apply to our program, should we encounter performance issues.

## Evaluation plan

The product movie database will be evaluated against a set of standards. During development, we will endeavor to keep our product adherent to all the standards specified to ensure a high-quality product as the end result.

|  |  |  |
| --- | --- | --- |
|  | Standard | Description |
| 1 | Accessibility | * The colors used on the webpages should be visible to colorblind users. * Elements should have proper contrast with one another. * The webpage should be navigable to users with only a keyboard input device. |
| 2 | Readability | * Body text fonts should be easily readable and simple sans-serif fonts. Navigation panel fonts should be easily readable. * Fonts should be in a color that contrasts well with its background. * Content should be laid out logically from top to bottom or left to right. |
| 3 | Usability | * Form controls should have an obvious purpose within their forms. * Form controls should give proper feedback when an incorrect value is submitted. * The navigation panel should be sensible and consist of easy to understand headings. * Content should be accessible in an intuitive way. This can mostly only be assessed by testing. |
| 4 | Aesthetics | The webpage should be pleasing to look at:   * Content should be spread out across the available space. * Fonts should be simple and pleasant colors. * The page background should be a pleasant muted color and contrast well with the text and content. * The navigation panel should complement the background and text color. * A consistent theme should be applied across all pages on the website, using the same / similar colors, fonts, graphics and layouts. |
| 5 | Performance | The webpage should be performant, external factors notwithstanding.   * Requests should complete within 1 second under controlled conditions. “Requests” referring to form submissions and page changes. |

Our final product will be evaluated against these standards. We invite the client to evaluate the website against these standards upon demonstration.

## Testing Plan

When testing, we will test the product according to these test cases.

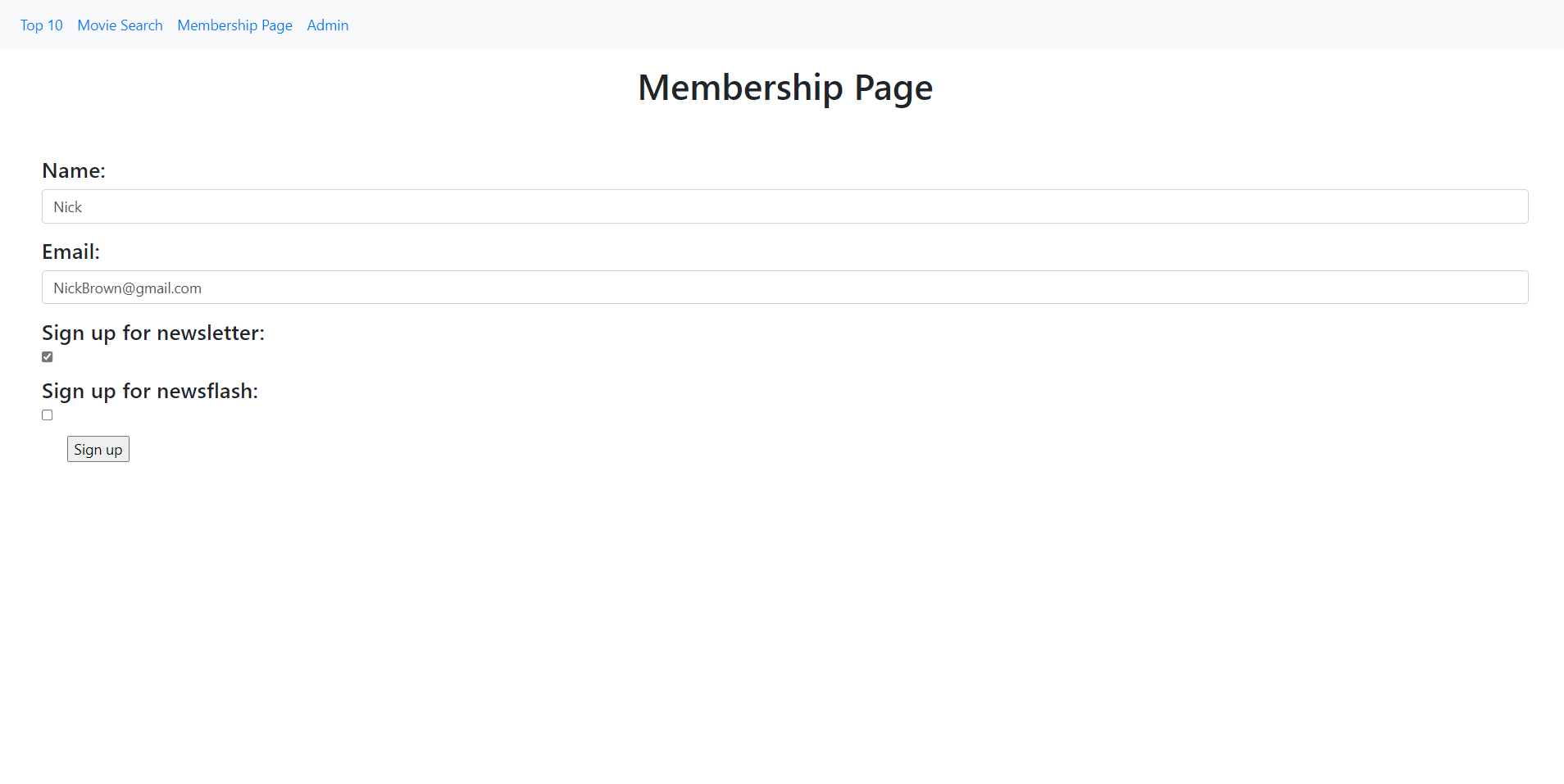
This list of cases is not exhaustive as all requirements for the website have not been delivered at the time of writing. More test cases will be added as more requirements are added or revised.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Expected / Desired Result | Actual Result | Evidence / Resources |
| Search for a movie by Name | Correct movies appear on results table. |  |  |
| Search for a movie using exact Date input | Correct movies appear on results table. |  |  |
| Search for a movie using before Date input | Correct movies appear on results table. |  |  |
| Search for a movie using after Date input | Correct movies appear on results table. |  |  |
| Search for a movie by Genre | Correct movies appear on results table. |  |  |
| Search for a movie by Rating | Correct movies appear on results table. |  |  |
| Add a new user to the database for newsletter correspondence. | The new user is correctly added to the database. |  |  |
| Add a new user to the database for newsflash correspondence. | The new user is correctly added to the database. |  |  |
| Attempt to add user to database with malformed email. | The new user is not added. |  |  |
| Remove a user by email on the admin page. | The user is correctly removed from the admin page. |  |  |
| Remove a non-existent user by email on the admin page. | A user is not removed from the database. |  |  |

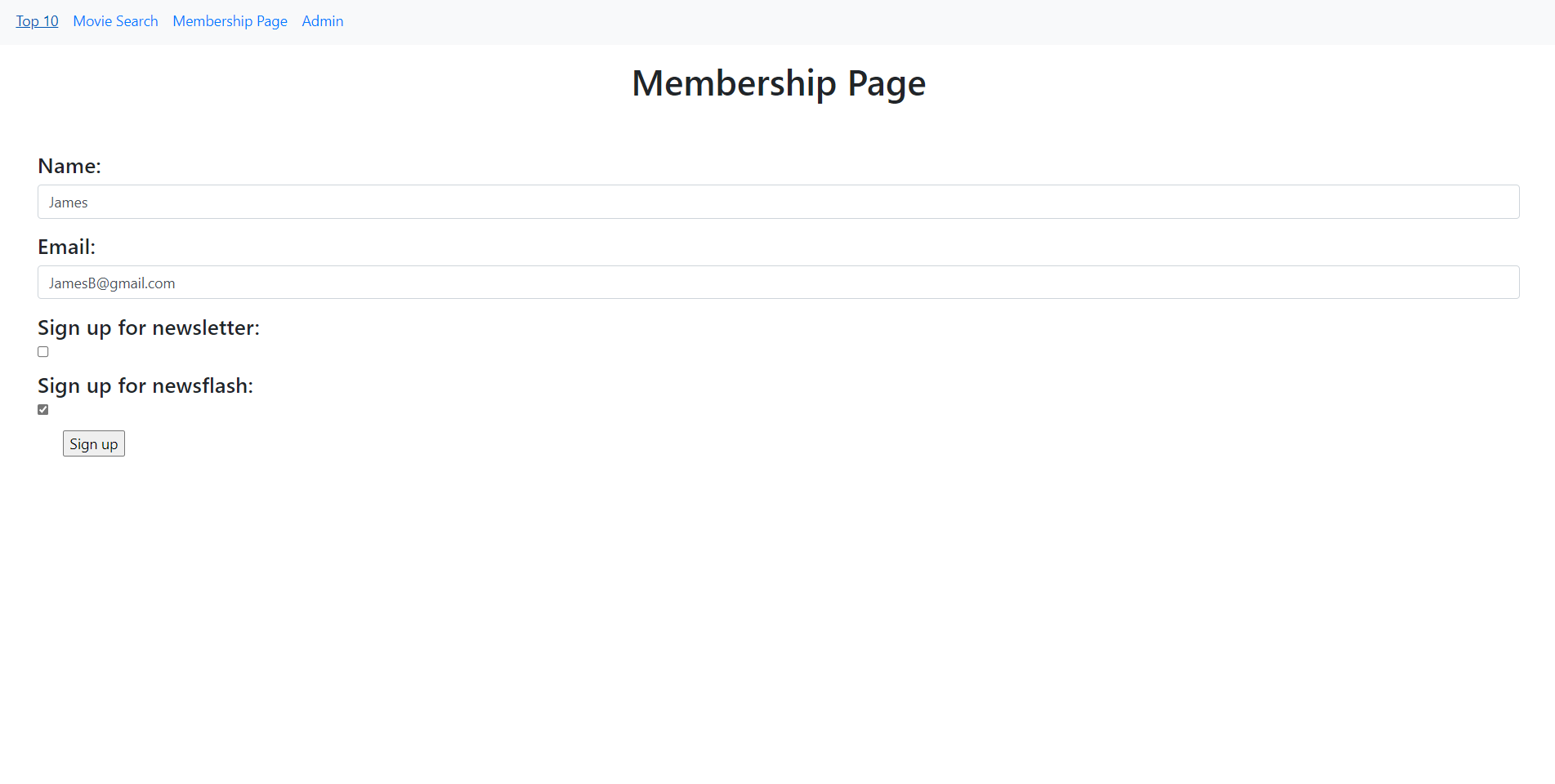
We will use a form of black box testing to test these cases. The tester will attempt a case specified by the testing plan. If the actual result matches the expected result, they will record a screenshot of the result to the table. They will then report deviation from the expected result of each case to the programmer. This will continue until the actual result of each case matches the expected result.

## Testing for sprint two

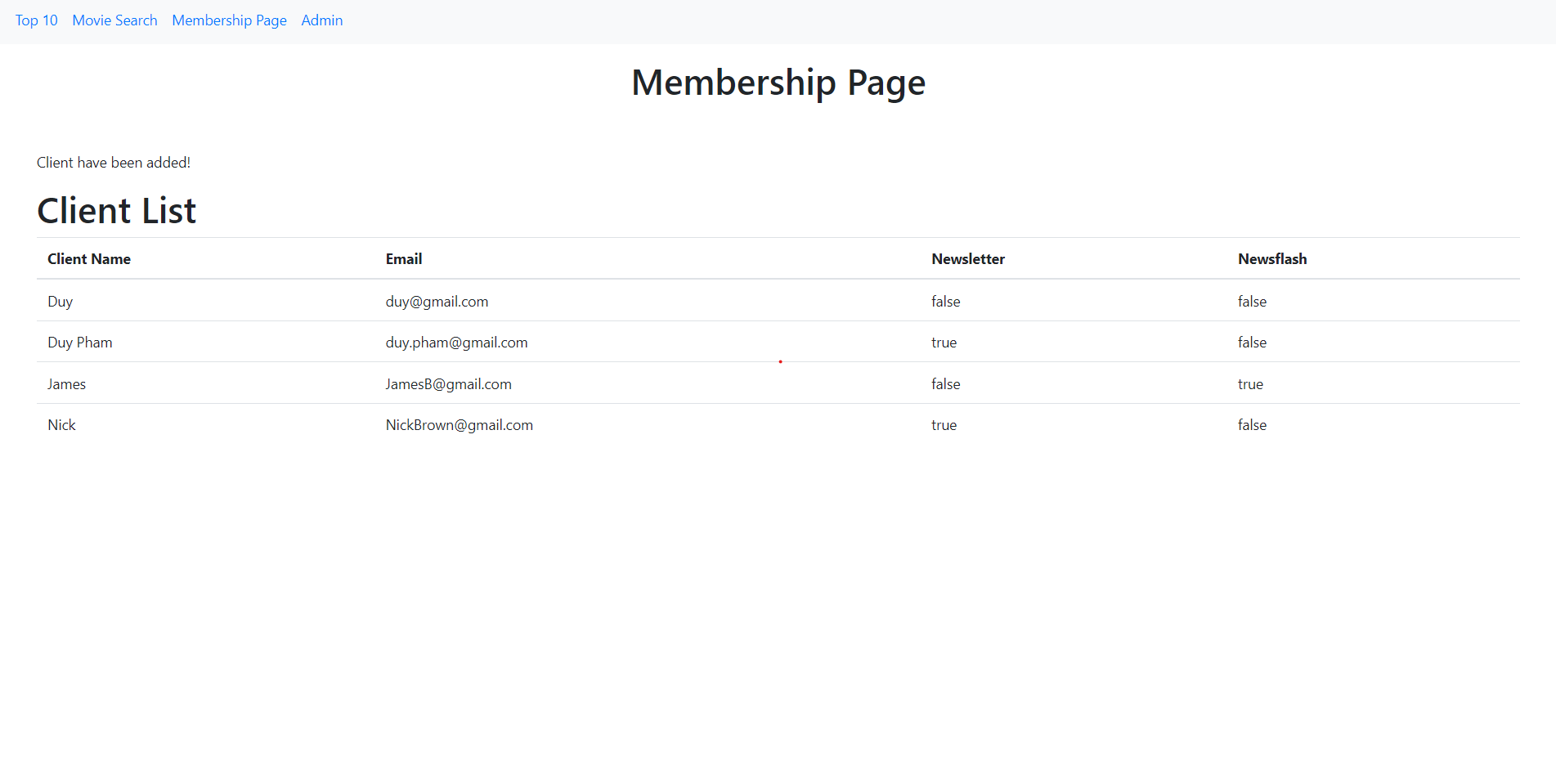
|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Expected / Desired Result | Actual Result | Evidence / Resources |
| Add a new user to the database for newsletter correspondence. | The new user is correctly added to the database. | As Expected. | Screenshot 1, 3 |
| Add a new user to the database for newsflash correspondence. | The new user is correctly added to the database. | As Expected. | Screenshot 2, 3 |
| Attempt to add user to database with malformed email. | The new user is not added. | As Expected. | Screenshot 4 |
| Remove a user by email on the admin page. | The user is correctly removed from the admin page. | As Expected. | Screenshot 5, 6 |
| Remove a non-existent user by email on the admin page. | A user is not removed from the database. | As Expected. | Screenshot 7 (Table did not change |



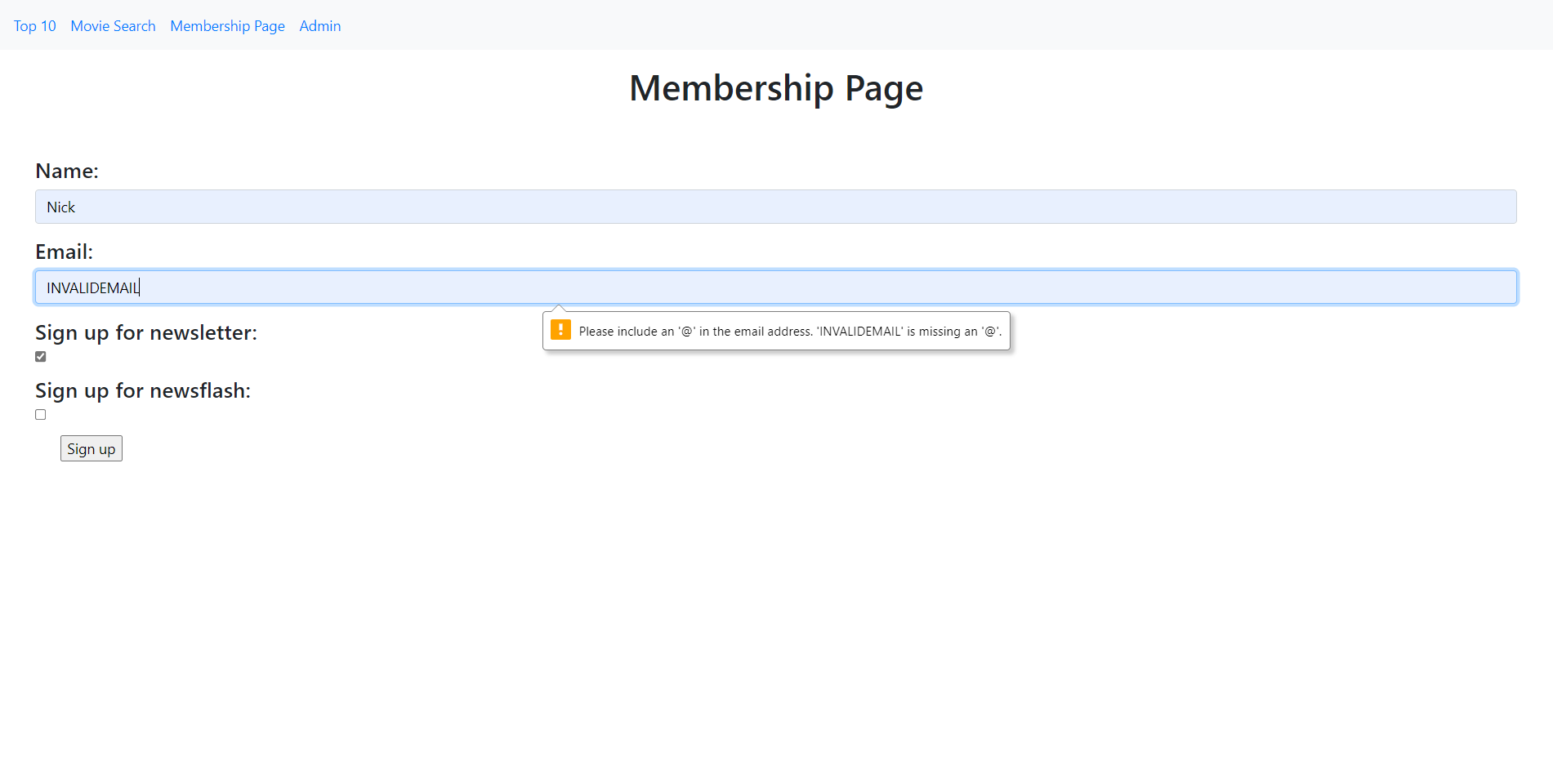
Screenshot 1



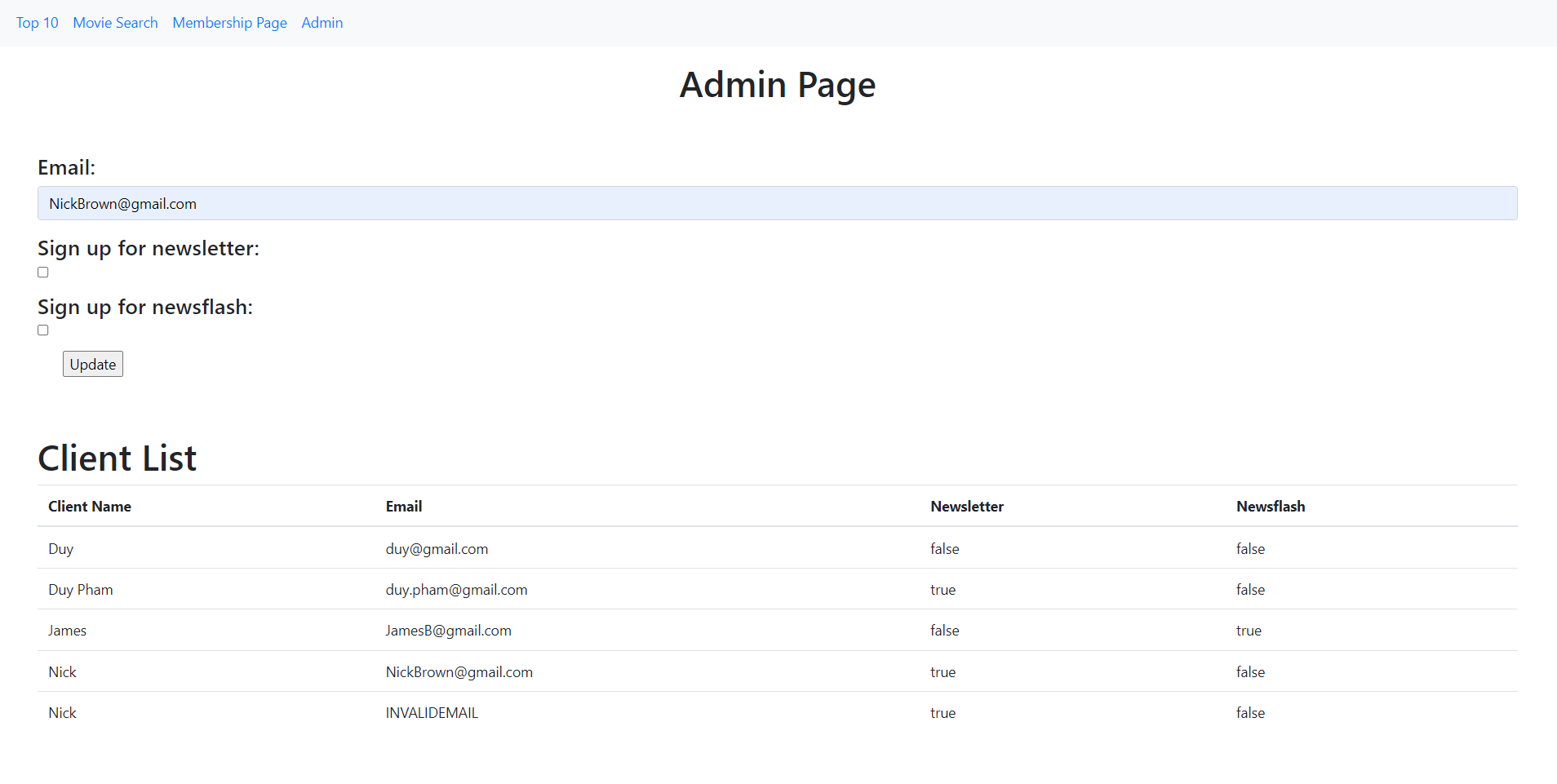
Screenshot 2



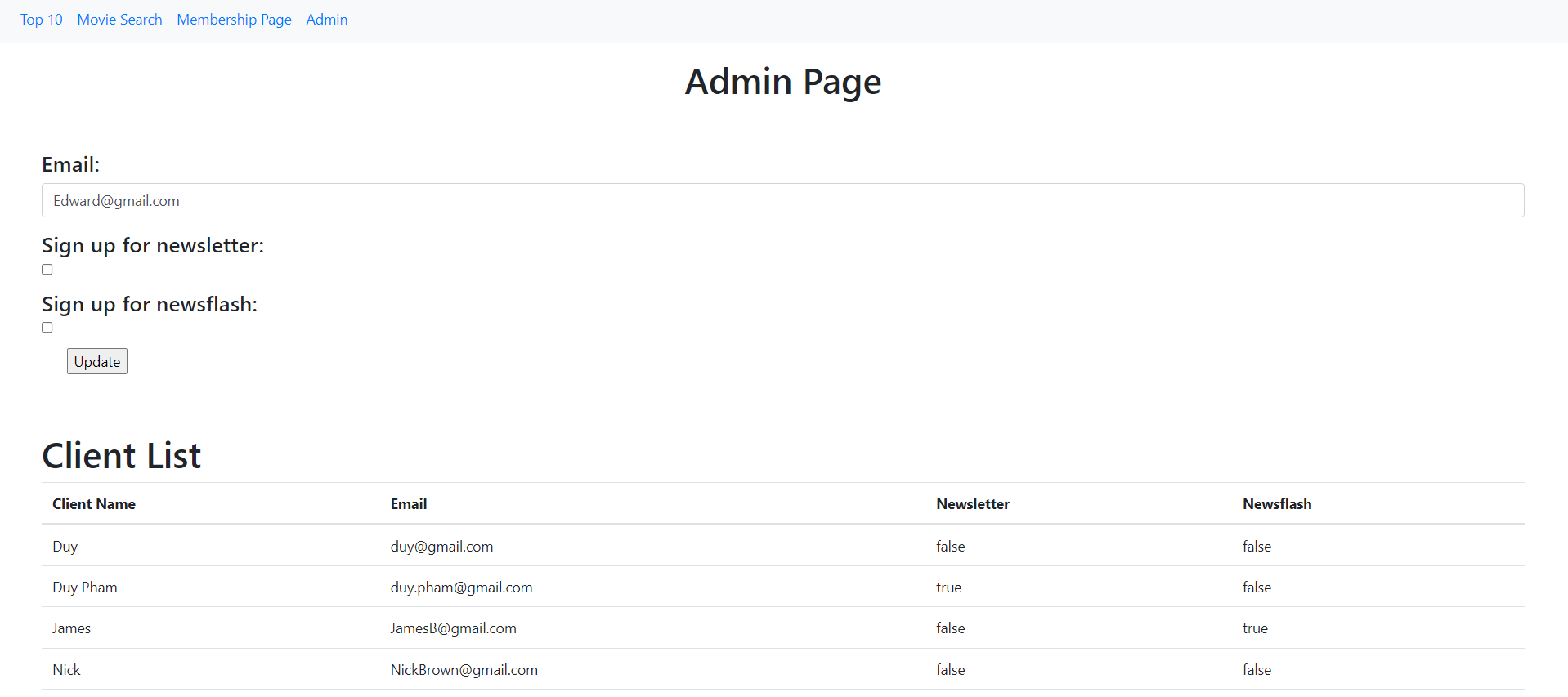
Screenshot 3



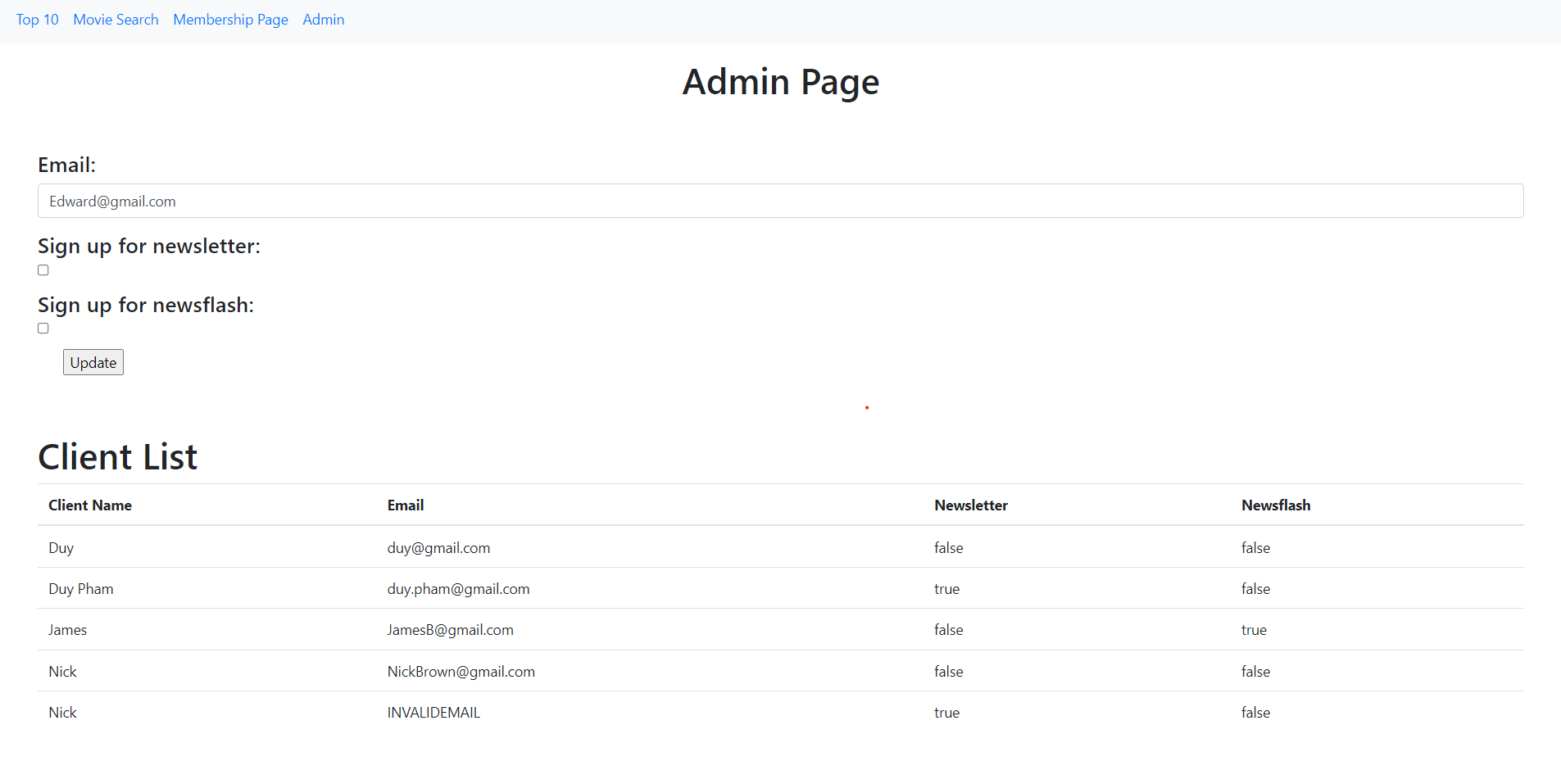
Screenshot 4



Screenshot 5



Screenshot 6

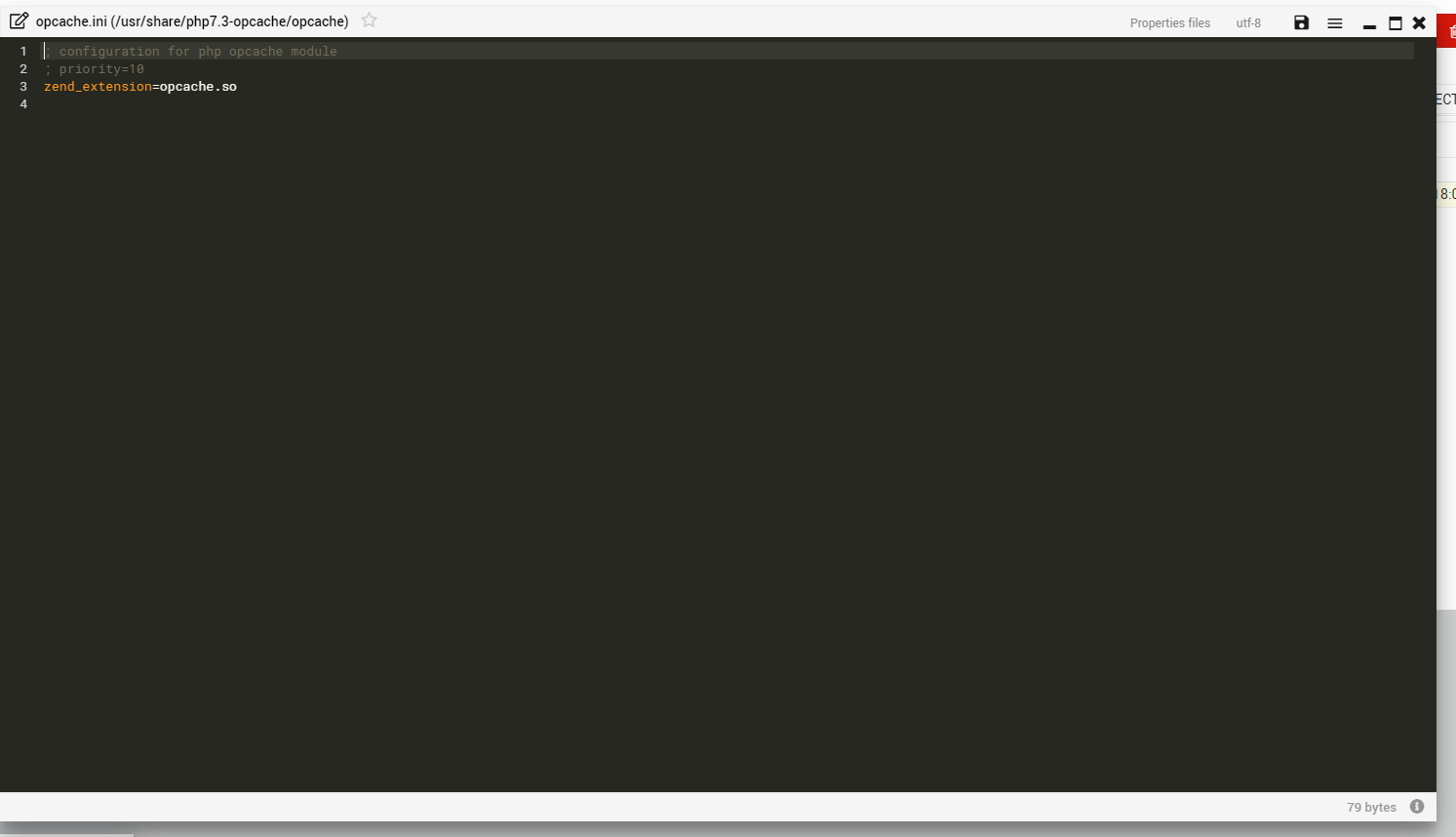


Screenshot 7

## Optimization Report

Investigation was done into the installation of a PHP accelerator to optimize our program to improve performance.

Upon investigation into our installation of PHP, I learned that the “Zend Optimizer” OpCaching service was already included and enabled in the server’s installation of PHP. This is a standard PHP accelerator utility. We evaluated this as sufficient use of a performance enhancing utility for this project.



Builtin Opcaching configuration on the server

At the time of writing, performance concerns are a lower risk factor in the development process. If this situation changes, this topic will need revisiting to further address performance concerns in more detail.

In the case that performance becomes an issue, our next avenue of investigation is manual analysis and optimization of our table generating code.

## Testing PLan

When testing, we will test the product according to these test cases.

This list of cases is not exhaustive as all requirements for the website have not been delivered at the time of writing. More test cases will be added as more requirements are added or revised.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Expected / Desired Result | Actual Result | Evidence / Resources |
| Search for a movie by Name | Correct movies appear on results table. |  |  |
| Search for a movie using exact Date input | Correct movies appear on results table. |  |  |
| Search for a movie using before Date input | Correct movies appear on results table. |  |  |
| Search for a movie using after Date input | Correct movies appear on results table. |  |  |
| Search for a movie by Genre | Correct movies appear on results table. |  |  |
| Search for a movie by Rating | Correct movies appear on results table. |  |  |
| Add a new user to the database for newsletter correspondence. | The new user is correctly added to the database. |  |  |
| Add a new user to the database for newsflash correspondence. | The new user is correctly added to the database. |  |  |
| Attempt to add user to database with malformed email. | The new user is not added. |  |  |
| Remove a user by email on the admin page. | The user is correctly removed from the admin page. |  |  |
| Remove a non-existent user by email on the admin page. | A user is not removed from the database. |  |  |
| Testing admin log in with wrong password | Reload admin log in page |  |  |
| Testing admin log with correct password | Take you to admin page |  |  |
| Testing rating from movie search page | Update the selected movie rating in the database |  |  |
| Testing rating without selecting a rating | Echo score is empty |  |  |
| Testing if top 10 is consistent with the database | Updated top 10 is shown |  |  |

We will use a form of black box testing to test these cases. The tester will attempt a case specified by the testing plan. If the actual result matches the expected result, they will record a screenshot of the result to the table. They will then report deviation from the expected result of each case to the programmer. This will continue until the actual result of each case matches the expected result.

## Testing for Sprint three

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Expected / Desired Result | Actual Result | Evidence / Resources |
| Testing admin log in with wrong password | Reload admin log in page | As expected. | Figure 7 |
| Testing admin log with correct password | Take you to admin page | As expected. | Figure 8 |
| Testing rating from movie search page and if the top 10 graph is been updated | Update the selected movie rating in the database | As expected. | Figure 9,10,11,12 |
| Testing rating without selecting a rating | Echo score is empty | As expected. | Figure 13 |

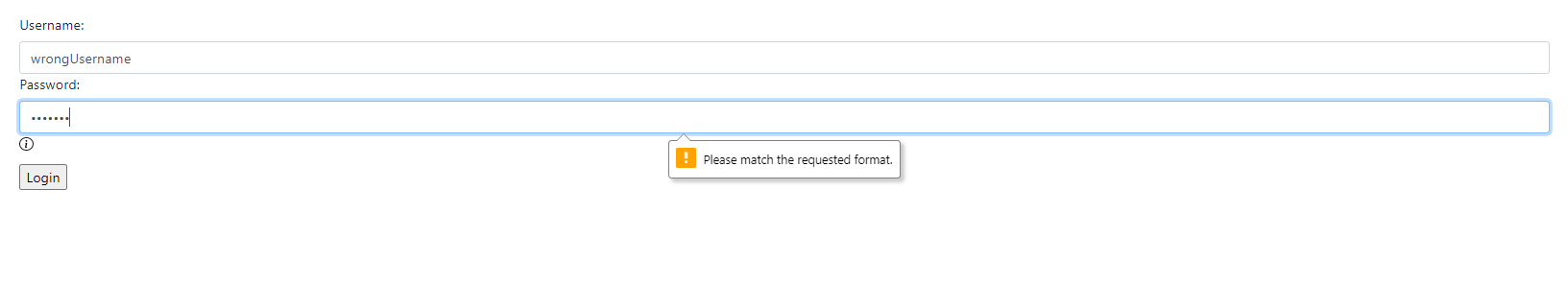


Figure 7

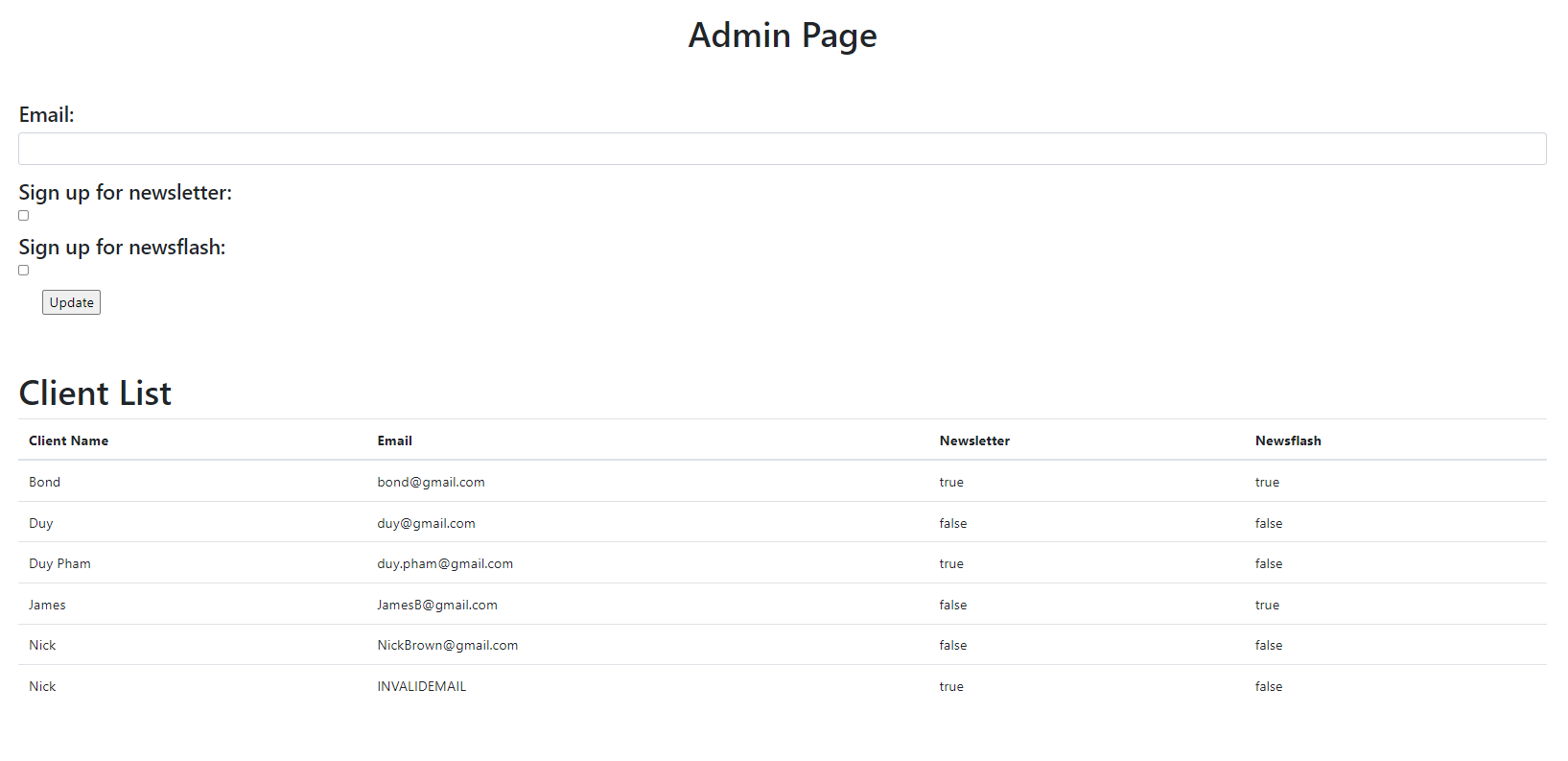


Figure 8

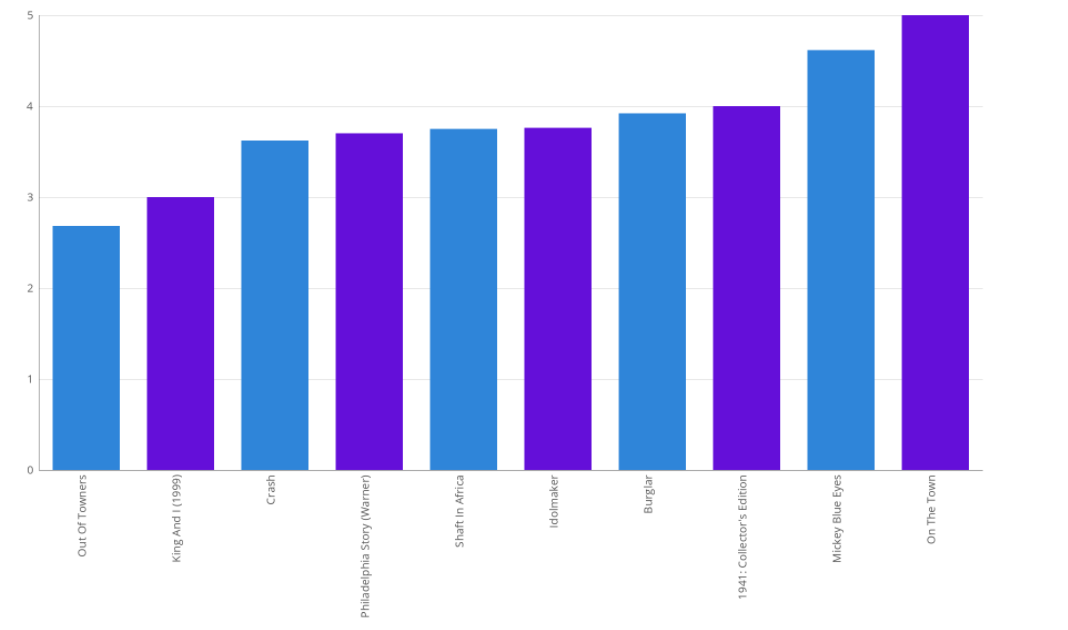


Figure 9



Figure 10



Figure 11

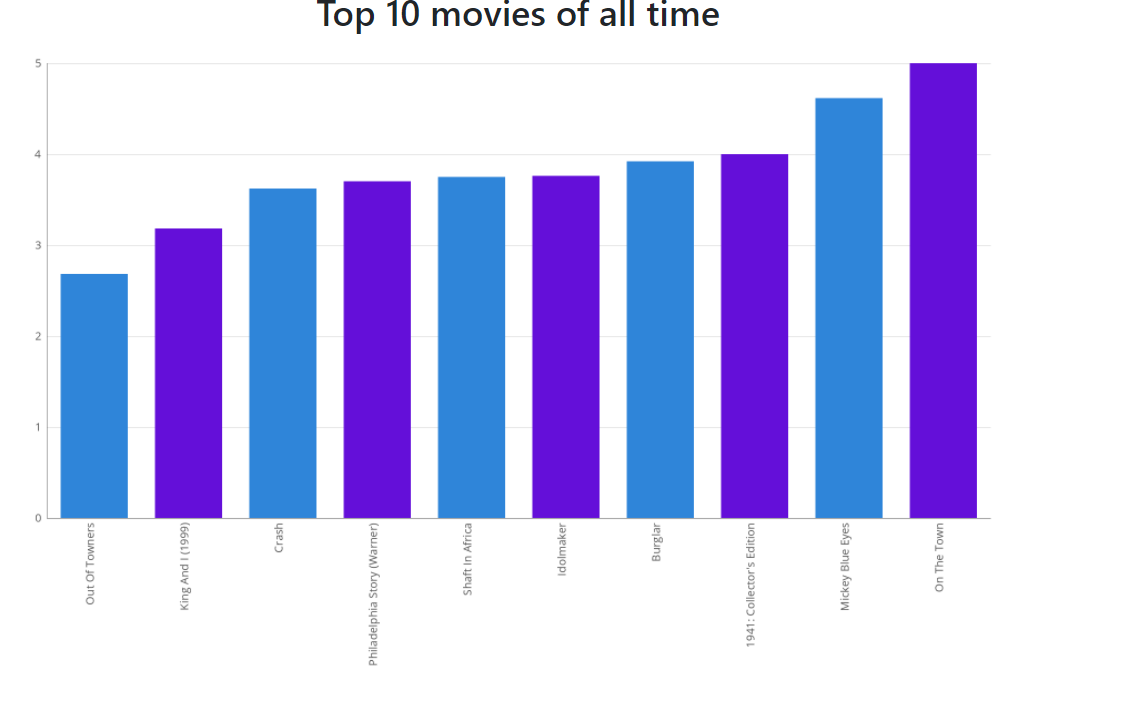


Figure 12

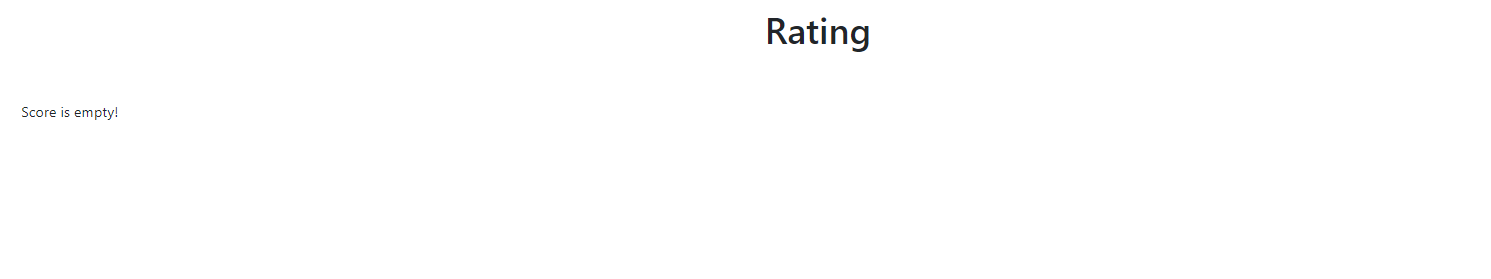


Figure 13

# Handover

## WCAG REquirements

### Principle 1 – Perceivable

Information and user interface components must be presentable to users in ways they can perceive

|  |  |
| --- | --- |
| Success Criteria | WCAG Guide Line |
| * + 1. Non-text Content | Images, form image buttons, and image map hot spots have appropriate, equivalent alternative text.  Images that do not convey content, are decorative, or contain content that is already conveyed in text are given empty alternative text (alt="") or implemented as CSS backgrounds. All linked images have descriptive alternative text.  Equivalent alternatives to complex images are provided in context or on a separate linked page.  Form buttons have a descriptive value.  Form inputs have associated text labels. |
| 1.3.1 Info and Relationships | Semantic markup is used to designate headings (<h1>), regions/landmarks, lists (<ul>, <ol>, and <dl>), emphasized or special text (<strong>, <code>, <abbr>, <blockquote>, for example), etc. Semantic markup is used appropriately.  Tables are used for tabular data and data cells are associated with their headers. Data table captions, if present, are associated to data tables.  Text labels are associated with form input elements. Related form elements are grouped with field set/legend. ARIA labelling may be used when standard HTML is insufficient. |
| 1.3.2 Meaningful Sequence | The reading and navigation order (determined by code order) is logical and intuitive. |
| 1.3.3 Sensory Characteristics | Instructions do not rely upon shape, size, or visual location (e.g., "Click the square icon to continue" or "Instructions are in the right-hand column").  Instructions do not rely upon sound (e.g., "A beeping sound indicates you may continue."). |
| 1.4.1 Use of Color | Color is not used as the sole method of conveying content or distinguishing visual elements.  Color alone is not used to distinguish links from surrounding text unless the contrast ratio between the link and the surrounding text is at least 3:1 and an additional distinction (e.g., it becomes underlined) is provided when the link is hovered over and receives focus. |
| ~~1.4.2 Audio Control~~ | ~~A mechanism is provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.~~ |

### Operable

User interface components and navigation must be operable

|  |  |
| --- | --- |
| Success Criteria | WCAG Guide Line |
| 2.1.1 Keyboard | All page functionality is available using the keyboard, unless the functionality cannot be accomplished in any known way using a keyboard (e.g., free hand drawing).  Page-specified shortcut keys and access keys (access key should typically be avoided) do not conflict with existing browser and screen reader shortcuts. |
| 2.1.2 No Keyboard Trap | Keyboard focus is never locked or trapped at one particular page element. The user can navigate to and from all navigable page elements using only a keyboard. |
| 2.2.1 Timing Adjustable | If a page or application has a time limit, the user is given options to turn off, adjust, or extend that time limit. This is not a requirement for real-time events (e.g., an auction), where the time limit is absolutely required, or if the time limit is longer than 20 hours. |
| 2.2.2 Pause, Stop, Hide | Automatically moving, blinking, or scrolling content (such as carousels, marquees, or animations) that lasts longer than 5 seconds can be paused, stopped, or hidden by the user.  Automatically updating content (e.g., a dynamically-updating news ticker, chat messages, etc.) can be paused, stopped, or hidden by the user or the user can manually control the timing of the updates. |
| 2.3.1 Three Flashes or Below Threshold | No page content flashes more than 3 times per second unless that flashing content is sufficiently small and the flashes are of low contrast and do not contain too much red. |
| 2.4.1 Bypass Blocks | A link is provided to skip navigation and other page elements that are repeated across web pages.  A proper heading structure and/or identification of page regions/landmarks may be considered a sufficient technique. Because navigating by headings or regions is not supported in most browsers. |
| 2.4.2 Page Titled | The web page has a descriptive and informative page title. |
| 2.4.3 Focus Order | The navigation order of links, form elements, etc. is logical and intuitive. |
| 2.4.4 Link Purpose | The purpose of each link (or form image button or image map hotspot) can be determined from the link text alone, or from the link text and its context (e.g., surrounding text, list item, previous heading, or table headers).  Links (or form image buttons) with the same text that go to different locations are readily distinguishable. |

### Understandable

Information and operation of user interface must be understandable

|  |  |
| --- | --- |
| Success Criteria | WCAG Guide Line |
| 3.1.1 Language of Page | The language of the page is identified using the HTML lang attribute (e.g., <html lang="en">). |
| 3.2.1 On Focus | When a page element receives focus, it does not result in a substantial change to the page, the spawning of a pop-up window, an additional change of keyboard focus, or any other change that could confuse or disorient the user. |
| 3.2.2 On Input | When a user inputs information or interacts with a control, it does not result in a substantial change to the page, the spawning of a pop-up window, an additional change of keyboard focus, or any other change that could confuse or disorient the user unless the user is informed of the change ahead of time. |
| 3.3.1 Error Identification | Required form elements or form elements that require a specific format, value, or length provide this information within the element's label.  Form validation errors are efficient, intuitive, and accessible. The error is clearly identified, quick access to the problematic element is provided, and the user can easily fix the error and resubmit the form. |
| 3.3.2 Labels or Instructions | Sufficient labels, cues, and instructions for required interactive elements are provided via instructions, examples, properly positioned form labels, and/or field sets/legends. |

### Robust

Content must be robust enough that it can be interpreted by a wide variety of user agents, including assistive technologies

|  |  |
| --- | --- |
| Success Criteria | WCAG Guide Line |
| 4.1.1 Parsing | Significant HTML validation/parsing errors are avoided. Check at <http://validator.w3.org/> |
| 4.1.2 Name, Role, Value | Markup is used in a way that facilitates accessibility. This includes following the HTML specifications and using forms, form labels, frame titles, etc. appropriately.  ARIA is used appropriately to enhance accessibility when HTML is not sufficient. |

# Software Review Report

## Quality Assurance & Control

During the development process, we developed a test plan which was then put into motion during the testing phase to counter foreseeable issues during development. We also tested extensively during the development phase to ensure our code and design were functioning as intended.

## Code Testing

The code testing in the testing phase was performed according to this structure:

1. Select a test case detailed in the testing plan.
2. Perform the test defined in the testing plan.
3. Compare the expected and actual results of the test.
4. Adjust the code if the test failed and repeat, OR
5. Take a screenshot as evidence if the test succeeded.

## Support for future development

If future modification is required, we are confident in our ability to extend our current code with further requirements.

* Appropriate comments are applied to our PHP code, but not our HTML code.
* Code is not well documented, as it might be using a documentation generator utility. If the scope of the project ever increases in the future, we may consider adding this kind of documentation. For the current scale of the project, we believe commenting will suffice for code documentation.
* We will be able to append new requirements and sprints to our documentation painlessly. This will consist of:
  + Adding a new project Kanban board to the project GitHub page.
  + Creating new requirements and testing documentation for the new sprint.
  + Appending the new documentation to the master document.

## Provide a mapping of the user requirement to the final software product (did you meet the clients expectations and requirements).

|  |  |
| --- | --- |
| User Requirements | Solution |
| Multi-platform report | We design the application using responsive approach which is more user friendly and SEO friendly. |
| Adapt prototype to responsive approach | In order to implement responsive approach to our website we utilize bootstrap. |
| Implement membership page | We implemented a form to capture people email and whether or not they want newsletter or newsflash notification. Figure 1 |
| Implement an option to remove subscription | We implemented a way for the employee to remove people from their newsletter or newsflash notification option. Figure 2 |
| Implement a password system for employee | To improve the security of the client database we have implemented a login system for employee. Figure 3 |
| Implement an admin system to update employee password | We implemented an admin system to update employee password. Figure 4 and 5 |
| Implement a rating system | We implemented a rating system which the user can rate the movie they like. Figure 6 |
| Implement streaming table | In order to keep the top 10 table updated we have implemented an auto refresh feature. |

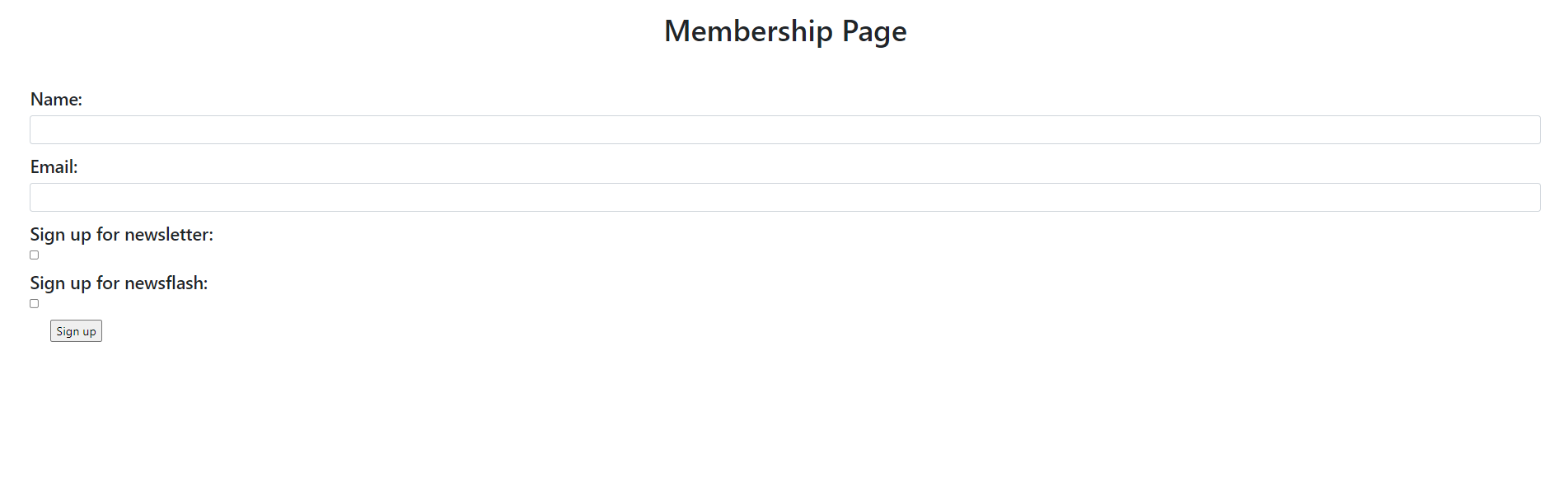


Figure 1

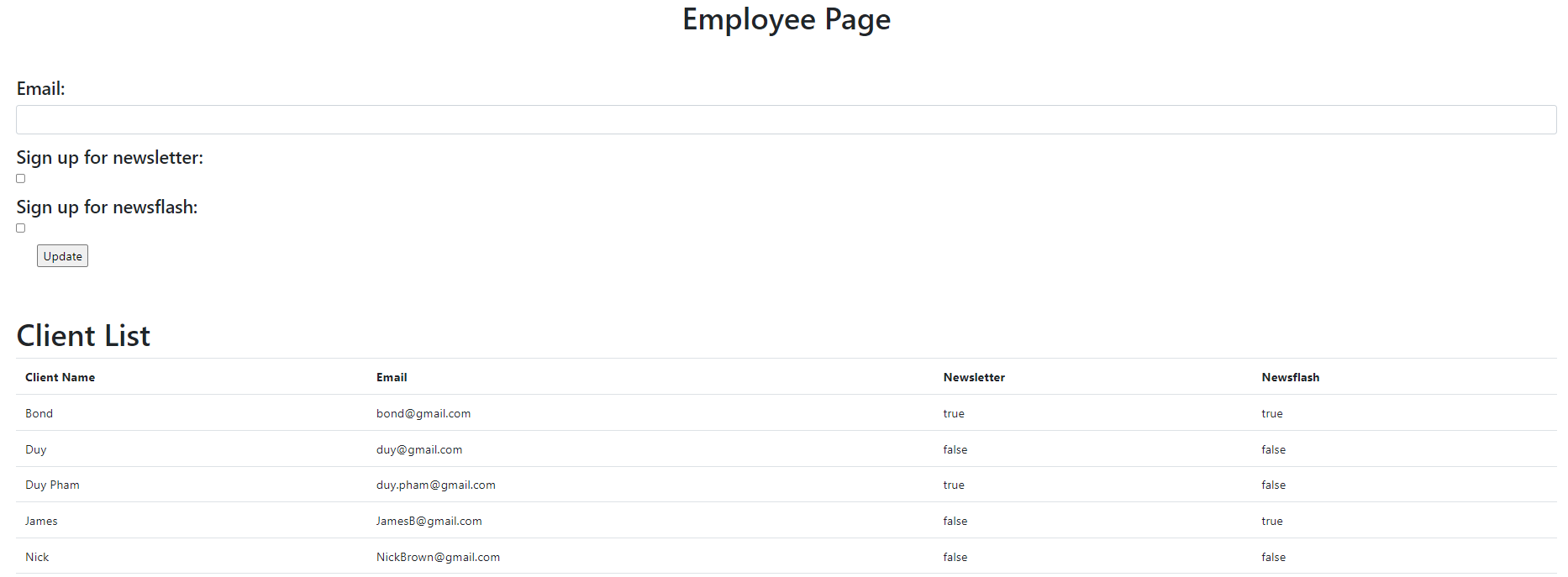


Figure 2

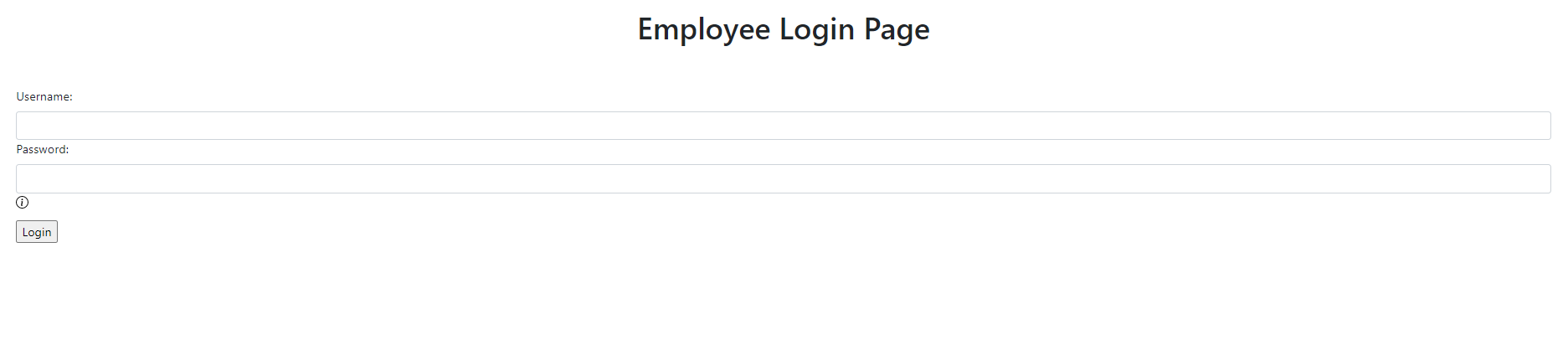


Figure 3

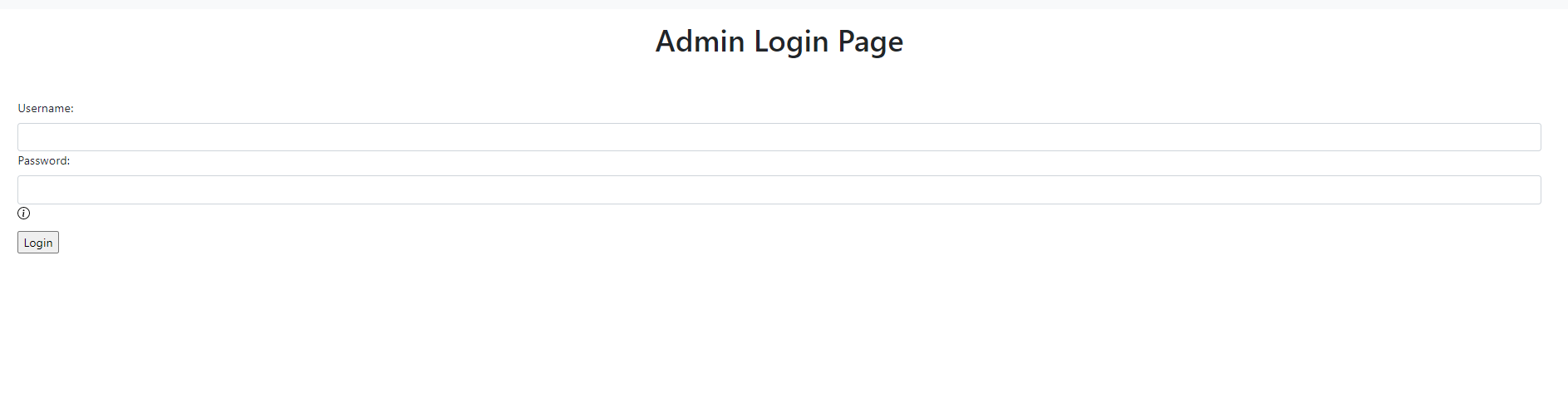


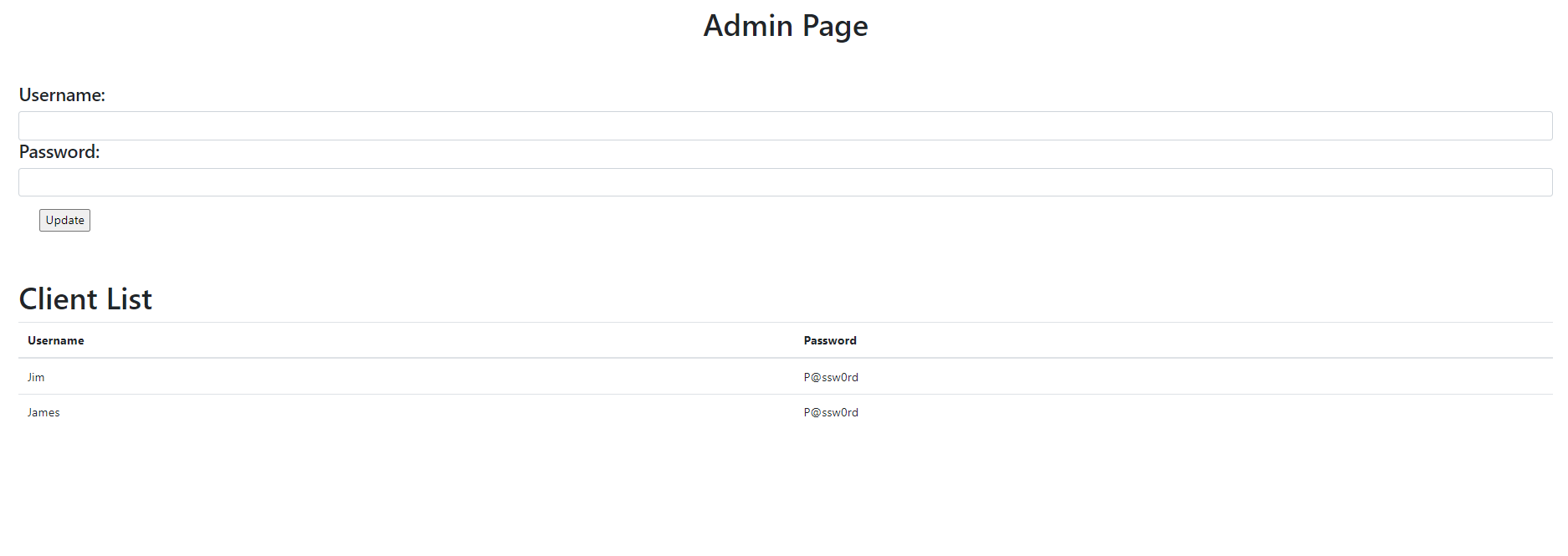
Figure 4 

Figure 5

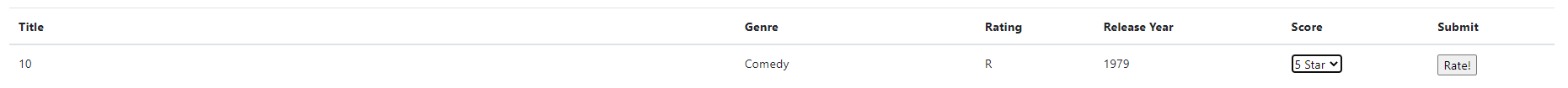


Figure 6