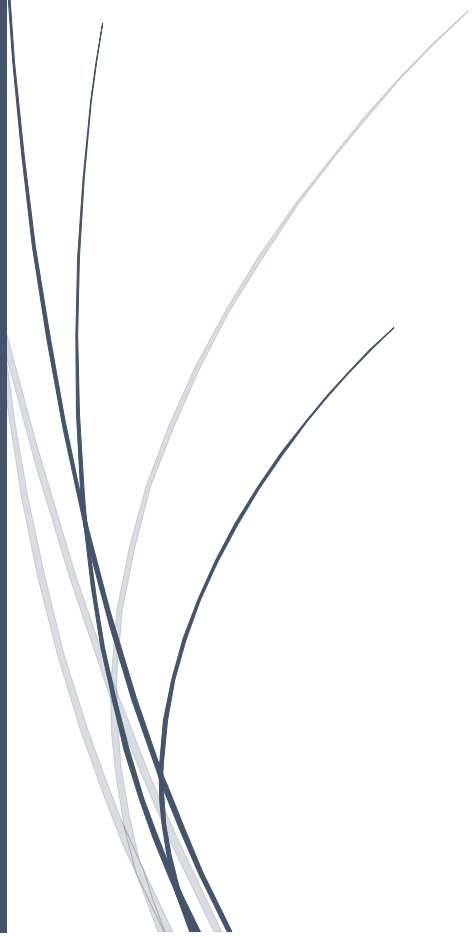
5/10/2021

Movie Database Project

RAD Project - Team Elite



EROL A'NIL, Nathan Tai, Blake Topham
SOUTH METROPOLITAN TAFE

Contents

Figures

Tables

Sprint One

Sprint Two

Source Control Snapshot

GitHub Repository

GitHub Link:

https://github.com/profiteroles/RAD

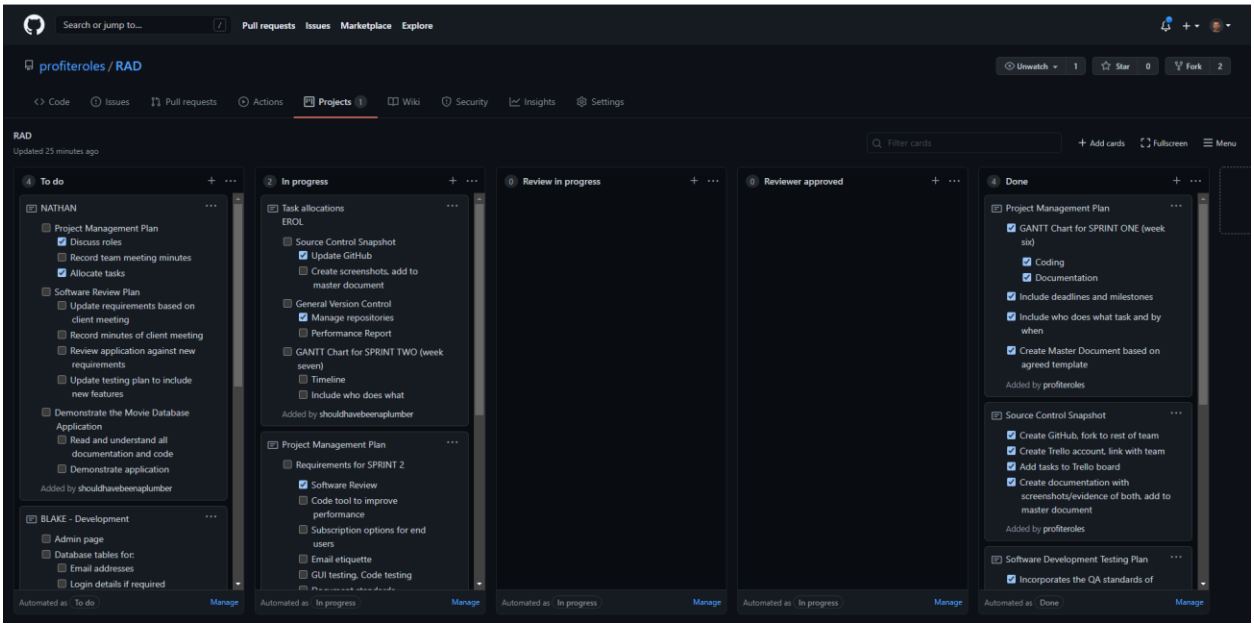
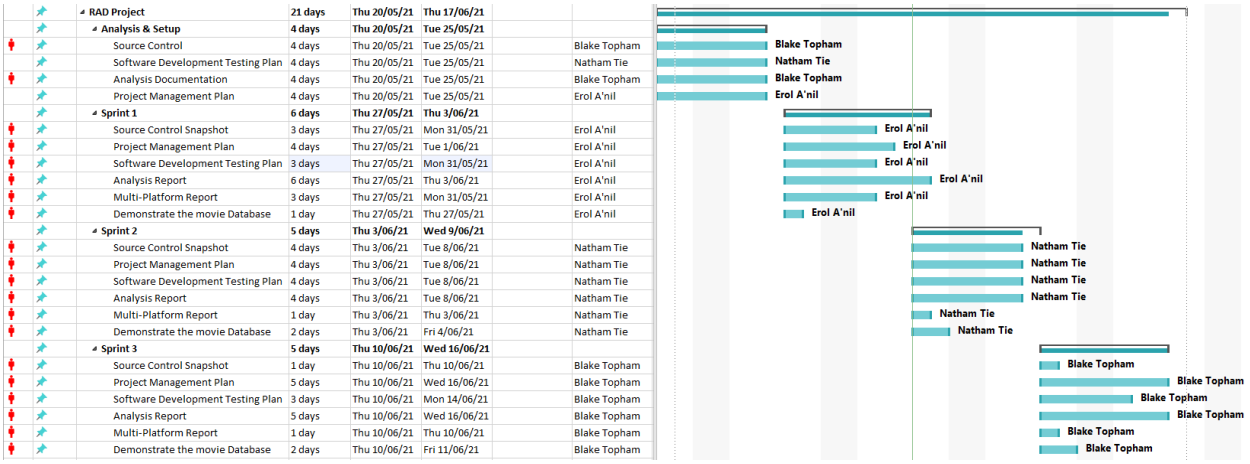


Figure 1- GitHub repository



	RAD Project	21 days	Thu 20/05/21	Thu 17/06/21	
	Analysis & Setup	4 days	Thu 20/05/21	Tue 25/05/21	
	Source Control	4 days	Thu 20/05/21	Tue 25/05/21	Blake Topham
	Software Development Testing Plan	4 days	Thu 20/05/21	Tue 25/05/21	Natham Tie
	Analysis Documentation	4 days	Thu 20/05/21	Tue 25/05/21	Blake Topham
	Project Management Plan	4 days	Thu 20/05/21	Tue 25/05/21	Erol A'nil
	Sprint 1	6 days	Thu 27/05/21	Thu 3/06/21	
	Source Control Snapshot	3 days	Thu 27/05/21	Mon 31/05/21	Erol A'nil
	Project Management Plan	4 days	Thu 27/05/21	Tue 1/06/21	Erol A'nil
	Software Development Testing Plan	3 days	Thu 27/05/21	Mon 31/05/21	Erol A'nil
	Analysis Report	6 days	Thu 27/05/21	Thu 3/06/21	Erol A'nil
	Multi-Platform Report	3 days	Thu 27/05/21	Mon 31/05/21	Erol A'nil
	Demonstrate the movie Database	1 day	Thu 27/05/21	Thu 27/05/21	Erol A'nil
	Sprint 2	5 days	Thu 3/06/21	Wed 9/06/21	
	Source Control Snapshot	4 days	Thu 3/06/21	Tue 8/06/21	Natham Tie
	Project Management Plan	4 days	Thu 3/06/21	Tue 8/06/21	Natham Tie
	Software Development Testing Plan	4 days	Thu 3/06/21	Tue 8/06/21	Natham Tie
	Analysis Report	4 days	Thu 3/06/21	Tue 8/06/21	Natham Tie
	Multi-Platform Report	1 day	Thu 3/06/21	Thu 3/06/21	Natham Tie
	Demonstrate the movie Database	2 days	Thu 3/06/21	Fri 4/06/21	Natham Tie
	Sprint 3	5 days	Thu 10/06/21	Wed 16/06/21	
	Source Control Snapshot	1 day	Thu 10/06/21	Thu 10/06/21	Blake Topham
	Project Management Plan	5 days	Thu 10/06/21	Wed 16/06/21	Blake Topham
	Software Development Testing Plan	3 days	Thu 10/06/21	Mon 14/06/21	Blake Topham
	Analysis Report	5 days	Thu 10/06/21	Wed 16/06/21	Blake Topham
	Multi-Platform Report	1 day	Thu 10/06/21	Thu 10/06/21	Blake Topham
	Demonstrate the movie Database	2 days	Thu 10/06/21	Fri 11/06/21	Blake Topham

Figure 2- GANTT chart sprint two

Performance Report

Definition

Performance report is a documented assessment of performance and capacity of an application or system that is complex, time consuming and expensive to build. Performance test helps to reduce the risk of down time on multi-user interface by conducting tests that use “load” to reveal errors and limitations in the application.

The steps in a performance test are:

- Discovery
- Modelling
- Developing Scripts
- Execution of tests

Goal

1. To clarify metrics and factors that the pages operate on
2. State of assumptions
3. Process description
4. Improvement opportunities

Assumptions

These assumptions should be revised by the team closely related to the business and specific part of the application.

Software and Hardware

- CPU
- Network Connection
- Hard Drive
- Memory
- Version of Operating System
- Version of Software
    - Web Server
    - Database
    - Applications Server

Process Description

All the steps in performance testing matter in making good decisions to make a project successful. These steps include, but aren’t limited to:

- Discovery
- Modelling
- Developing Scripts
- Executing Test

Optimizer Tools

Possible tools for code optimisation/performance improvement would include:

PINBA

A MySQL storage engine that monitors the performance of PHP code and detects bottlenecks in real time. Statistics are displayed in user-friendly, human language. Interface is read only.

PHPLOC

A tool to measure how many lines of code in a PHP file as well as number of classes, files, etc. Ability to generate retroactive reports in CSV format.

```
➜ ~ phploc src
phploc 2.0.4 by Sebastian Bergmann.

Directories                                 3
Files                                       8

Size
  Lines of Code (LOC)                    1858
  Comment Lines of Code (CLOC)            560 (30.14%)
  Non-Comment Lines of Code (NCLOC)      1298 (69.86%)
  Logical Lines of Code (LLOC)            289 (15.55%)
    Classes                               260 (89.97%)
      Average Class Length                 37
      Average Method Length                 9
    Functions                               5 (1.73%)
      Average Function Length               5
    Not in classes or functions            24 (8.30%)

Complexity
  Cyclomatic Complexity / LLOC           0.67
  Cyclomatic Complexity / Number of Methods      7.86
```

Data Need Analysis

Load testing tools have graphic capability. Graphs are just tools but not an actual report, however graphical data aids visualization to guide the stakeholders in consuming actionable information. Learning pattern recognition can take years to acquire where the need exists to recognize the system performance changes after certain load is surpassed, understanding of the limiting resources etc, is an ongoing and changing process.

A performance tester does the following duties:

- Form hypothesis
- Draw tentative conclusions
- Collect information to determine the information needed for the above
- Prepare key visualizations that provide insight into the performance and bottlenecks
- Support report narratives

To perform the above duties a thorough understanding of the following is very important:

- Architecture
- Hard and Soft Resources
- Garbage Collection Algorithms
- Database Performance
- Message Bus Characteristics
- Auxiliary Components of Systems – in complex systems

The full value of the performance test is unlocked when there is collective information from Developers, Operations, database analysts, help desk techs, business stakeholders and all teammates. Few effective steps to successfully achieve this are:

Collecting

To weigh up the validity of the performance results its essential to gather-

- Errors and type of errors
- Pattern of errors
- Obtaining error logs from the application

Measurements from every few seconds helps to understand the granularity of the application and help us to spot the trends and transient conditions in application development and testing.

Aggregating

Measurements with statistics like scatter plots, graphs, data ranges, variance studies to study the data distribution aids in making the report more accurate. Using various levels of granularity provide isolated to collective views of the performance with compared with consistent granularities. This can be an improvement strategy standard.

Visualizing

Comparison studies are done by using key graphical indicators to help us understand what is happening during the testing of the application.

List of the comparative visualizations are

- Check for validity of results by studying **Load data vs Error**
- Recognizing bottlenecks by checking **Load vs bandwidth throughput**
- Study scaling and scaling behavior by studying **Load Vs Response time**
- **Capacity of Infrastructure** i.e. system resources adequacy is determined by
    - Load vs Server CPU
    - JVM heap Memory
    - Input/output latency
    - Database lock contention

Interpreting

Evaluating the data and drawing conclusions from hypothesis can be done by

- Quantitative observations – What can be mainly observed in the data?
- Comparing the observations – Where are the consistencies and inconsistencies?
- Developing hypothesis based on observations
- Testing the hypothesis
- Conclude from the hypothesis by validating

Analyzing

Deciding on actions to be taken by checking if the objectives are met, then determining remediation options at business level, applications level, system requirements and network level, then retest.

Transparency in costs, benefits, and risk is essential, them must be specific and actionable at technical level or at the management level.

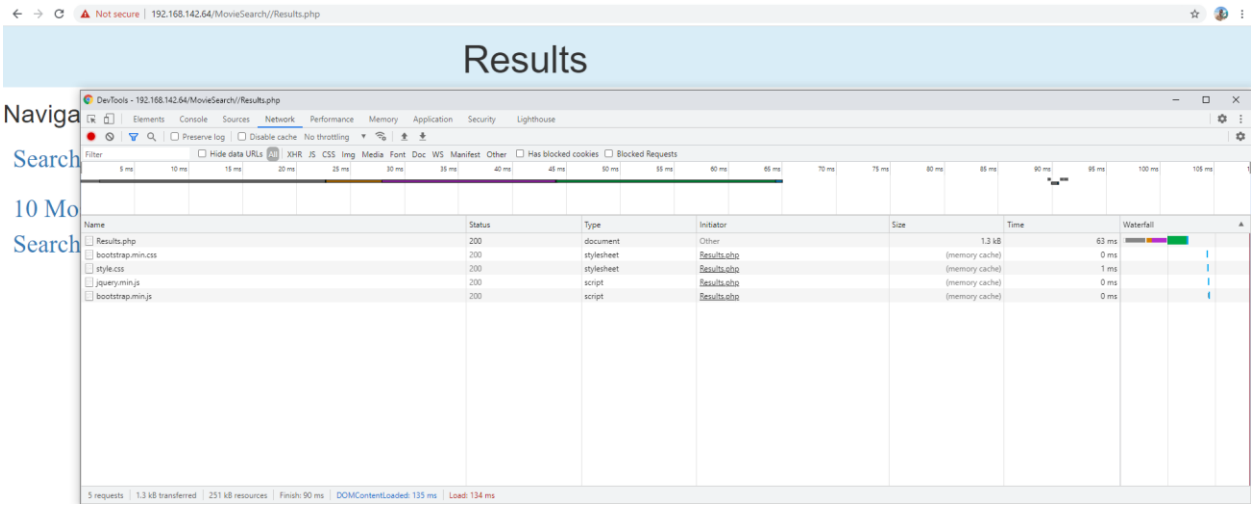
Reporting

This is done by aggregating and presenting the risks, costs, limitations and recommendations to the stakeholders' terms in a short elevator summary or a brief narrative. The report has the following sections:

- Executive summary
- Supporting detail
- Documents associated with the test
- Presentation

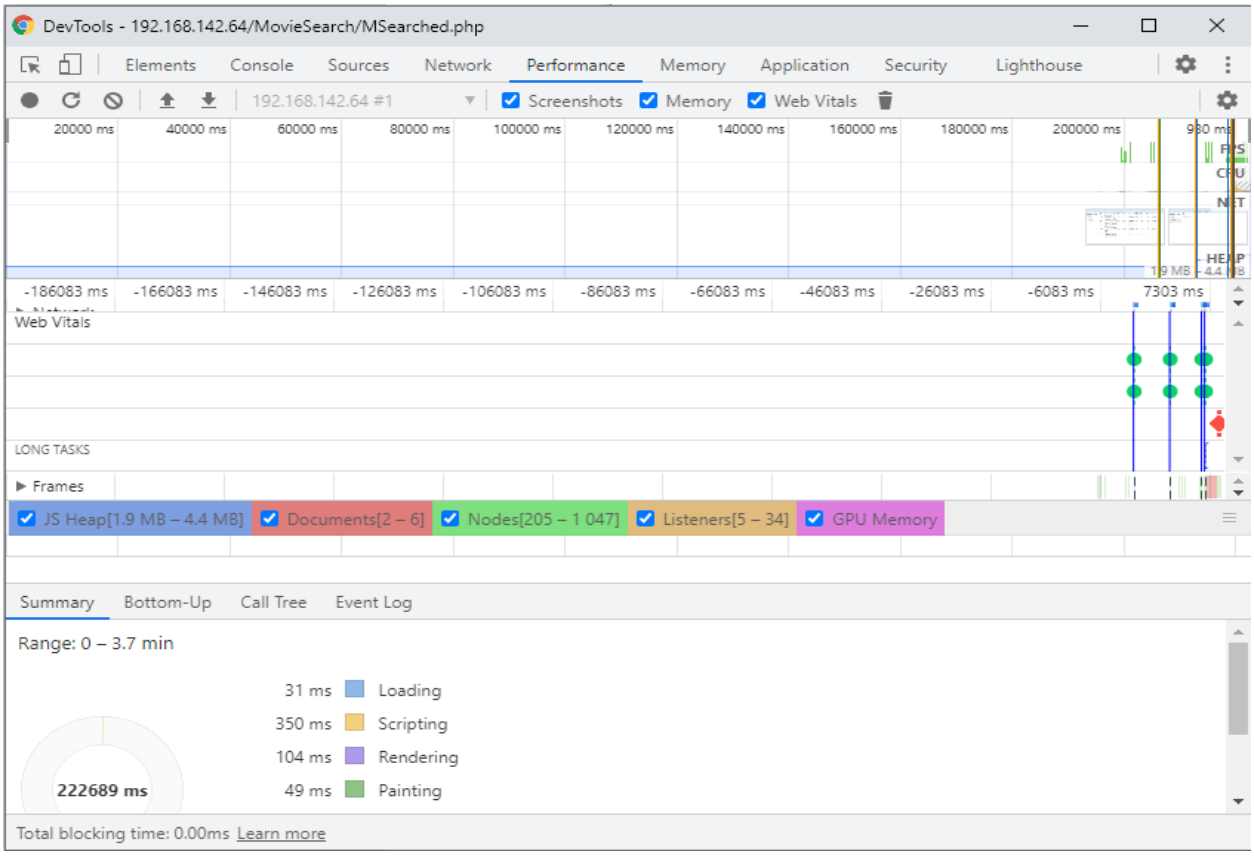
Performance of Results

Response that we get within 65ms.



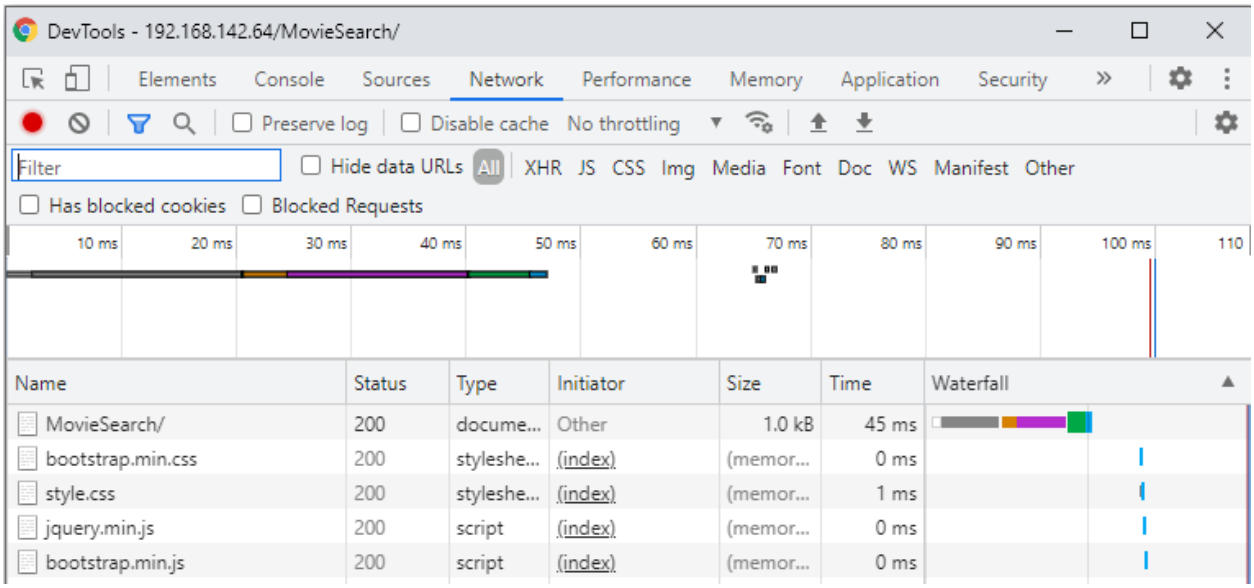
Performance of Top 10 Movies

The completion of top 10 movies takes up to 980ms which includes the loading of a graphical chart.



Performance of Home Page

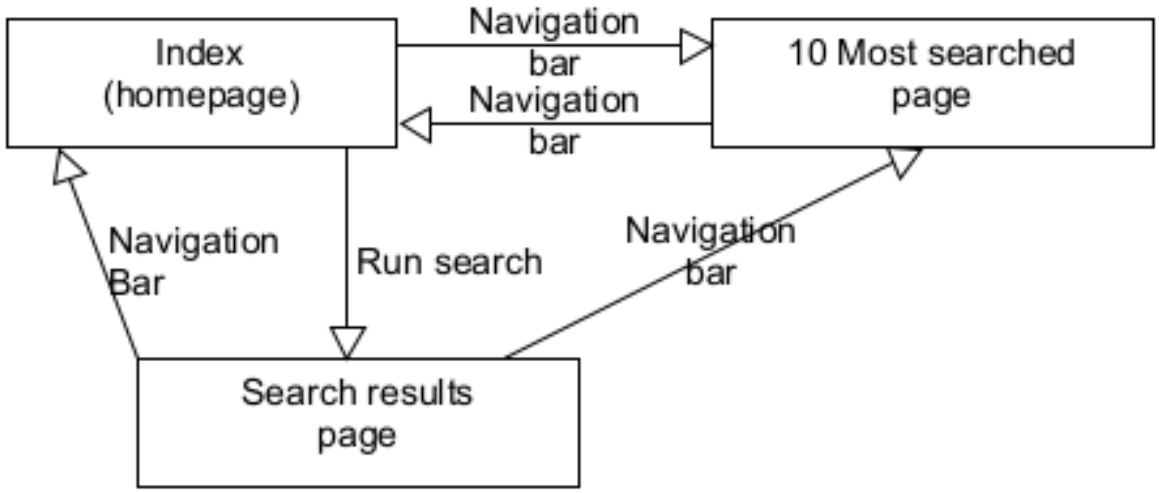
The Home(Index) page loads less than 50ms.



Software Review Plan

Current Architecture

The following navigation chart illustrates the current system structure of the project. The web application is comprised of three pages; Index, Search Results, and Top Ten. The chart demonstrates the relationships and pathways, including their direction.



The Index page is an HTML page which serves as a hub, providing text input fields and buttons. The buttons directly link to each of the two other pages, passing user input data using a “POST” method.

The Results page contains a connection script to the associated database. It generates SQL queries based on search terms posted from the index page, and constructs a table to display all matching entries in the database. A user-friendly message is displayed if no matching results are found.

The Ten Most-Searched Movies page is similar to the search results page. The page contains its own connection script and generates an SQL query to find and return the ten most-searched movies in the database. Movies are displayed in a table identical to that of the results page, and includes a bar chart displaying the titles and search count of each of the top ten movies.

Software Quality Issues

Quality issues can be grouped in the following categories:

Modularity

The two results pages (search results and top ten) contain **repeated code**; both of these pages have their own identical connection scripts.

Portability

The database connection scripts contain **hard coded values** for host, username, password and database name. Given the scope of the project, this may be inconsequential but is worth noting.

Efficiency

The code appears to contain some **redundant variables** and links, a few at most. Impact may be negligible. Further analysis required.

Extensibility

In its current state, the codebase is quite small and simple. New additions would not be difficult to implement and integrate.

Readability

Code is sufficiently commented in plain terms using minimal technical vocabulary. Appropriate indentation is utilized throughout. Code contains minimal nesting. Naming conventions are consistent and meaningful.

Proposed Improvements

- Remove connection scripts from results pages. Create a dedicated database connection file to be referenced as needed, or place the connection script in a global function.
- Streamline code; check for redundant variables/links and remove if necessary
- Format code with auto-formatting tool

Suggestions/Recommendations (Optional)

- Create optional form on homepage to override default values for database connection with user-defined values (host, user, password, etc.)
- Place code for results table in a global function

Testing

Full program will be tested, including previously tested features to **monitor effect of changes** and ensure new additions have not broken existing modules. The testing methodology, strategy, and metrics are consistent with those detailed in the Sprint One testing plan.

Scope

The new proposed features are detailed in the following revised testing scope:

In Scope

- User authentication
- Display database
- Search database
    - With any combination of parameters
- Top ten movies
- Operation on three different sized devices
    - PC
    - Tablet/Laptop
    - Phone

New features to be tested:

- User subscription
- Unsubscribe option
- Email user with:
    - Newsletter
    - Newsflash updates
- Admin portal
    - Receives email when user unsubscribes
    - Function to remove user details from database
- User experience (NON FUNCTIONAL)
    - Speed of operation to be reviewed in **Performance Report**

Out of Scope

- Login security
- User account recovery
- Thumbnails with theatrical posters/cover art
- User reviews

Test Table

Test Report S2 T1						
Project Name: RAD Sprint One			Test Type: Black Box			
Description: Movie Search Application			Date: 03/06/2021			
Developer(s): Blake Topham, Erol Anil, Nathan Tai			Tester: Blake Topham			
Test Case #	Test Case Name	Test Steps	Test Data	Expected Results	Evidence	Pass?
1	Search by Name	Enter search terms	Search term	Matching results displayed	Figure	☒
2	Search by Name, Genre	Enter search terms	Search term	Matching results displayed	Figure	☒
3	Search by Name, Genre, Rating	Enter search terms	Search term	Matching results displayed	Figure	☒
4	Search by Name, Genre, Rating, Year	Enter search terms	Search term	Matching results displayed	Figure	☒
5	Top Ten Movies	Click “10 Most Searched” button	10 movies with highest search count	10 movies with highest search count displayed	Figure	☒
6	Runs on: Desktop PC	Run page in desktop browser	Page elements	Page runs	Figure	☒
7	Runs on: iPad	Run page in device simulation mode	Page elements	Page runs	Figure	☒
8	Runs on: Galaxy S5	Run page in device simulation mode	Page elements	Page runs	Figure	☒
9	User subscribes	Enter email and click subscribe	Email address	User subscribed, receives newsletter/newsflash	Figure	☒
10	User unsubscribes	Enter email and click unsubscribe	Email address	Email sent to Admin account for database removal	Figure	☒
11	Admin removes user	Remove user entry from database	User details(email, etc.)	User entry removed from database	Figure	☒

Test Completeness Criteria Achieved:	Suspension Criteria Met:
☒ 1. All test cases carried out successfully ☒ 2. All bugs fixed	☐ 1. Vital functionality broken ☐ 2. Error directly prevents further testing ☐ 3. Errors too many or too often ☐ 4. Error makes further testing redundant
☒Testing Complete	☐Testing Suspended

Screenshots

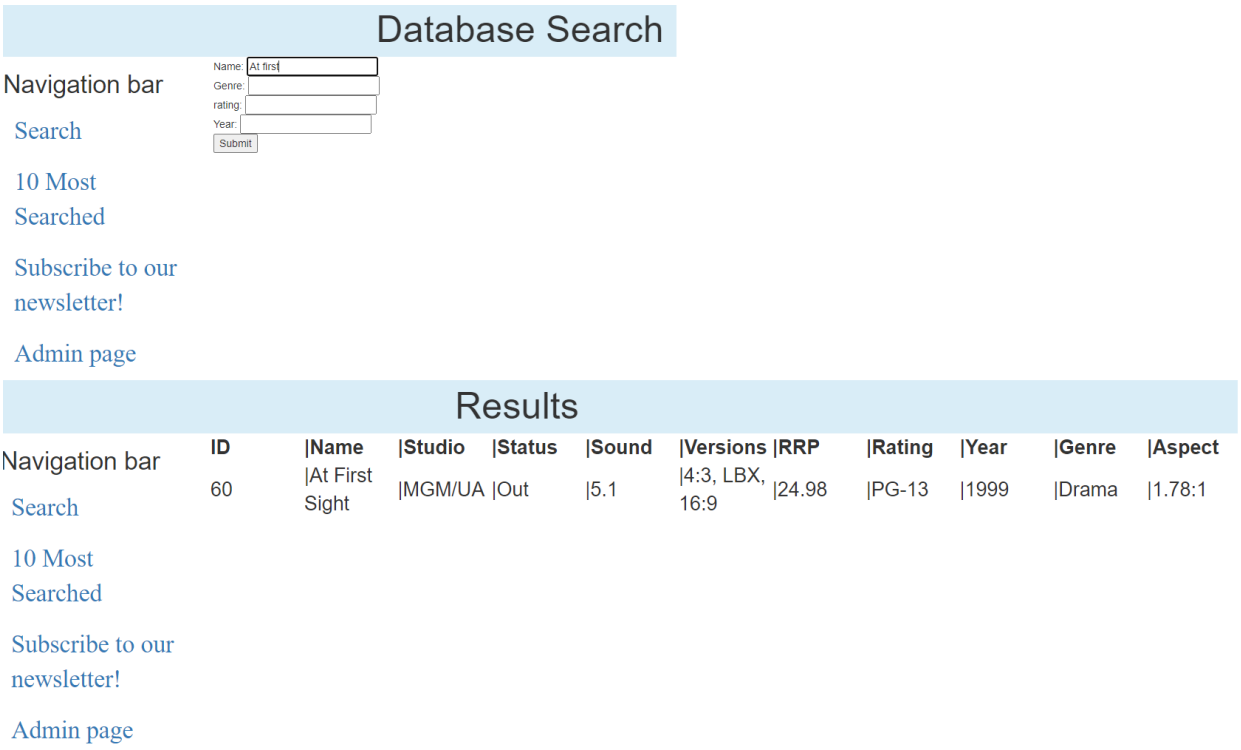


Figure 3- Search by name

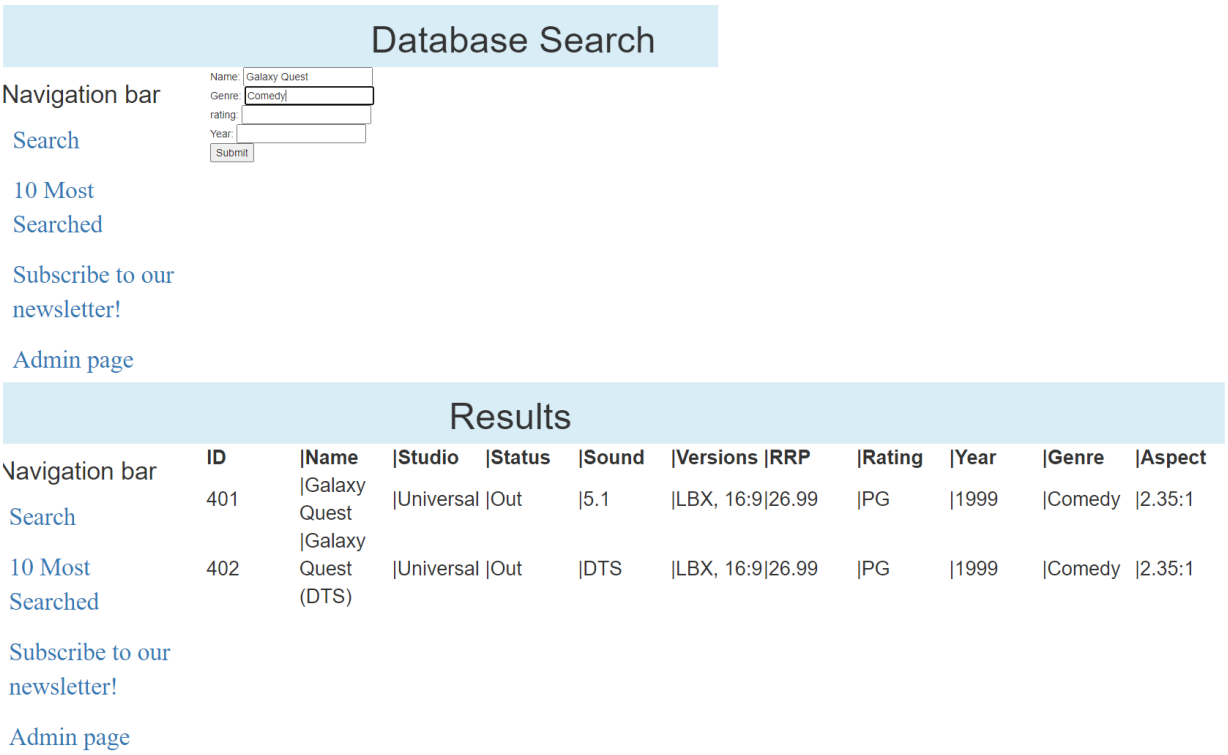


Figure 4- Name, Genre

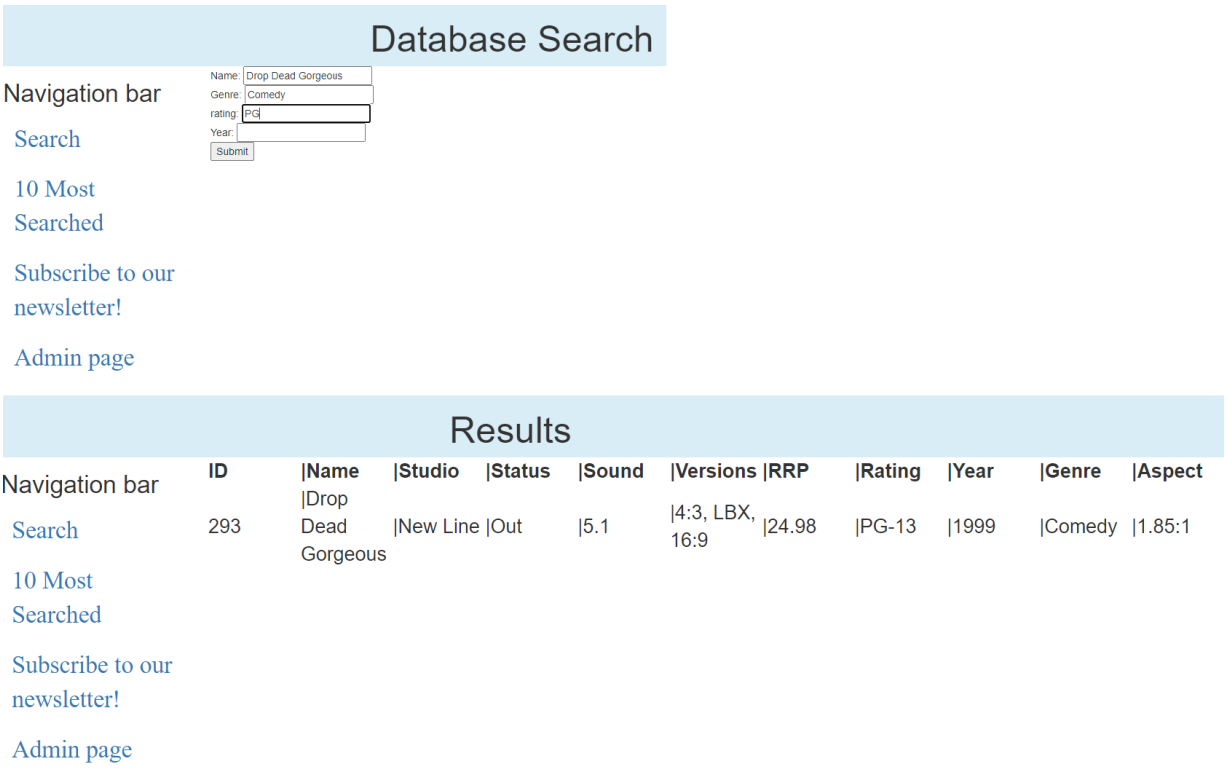


Figure 5-Name, Genre, Rating

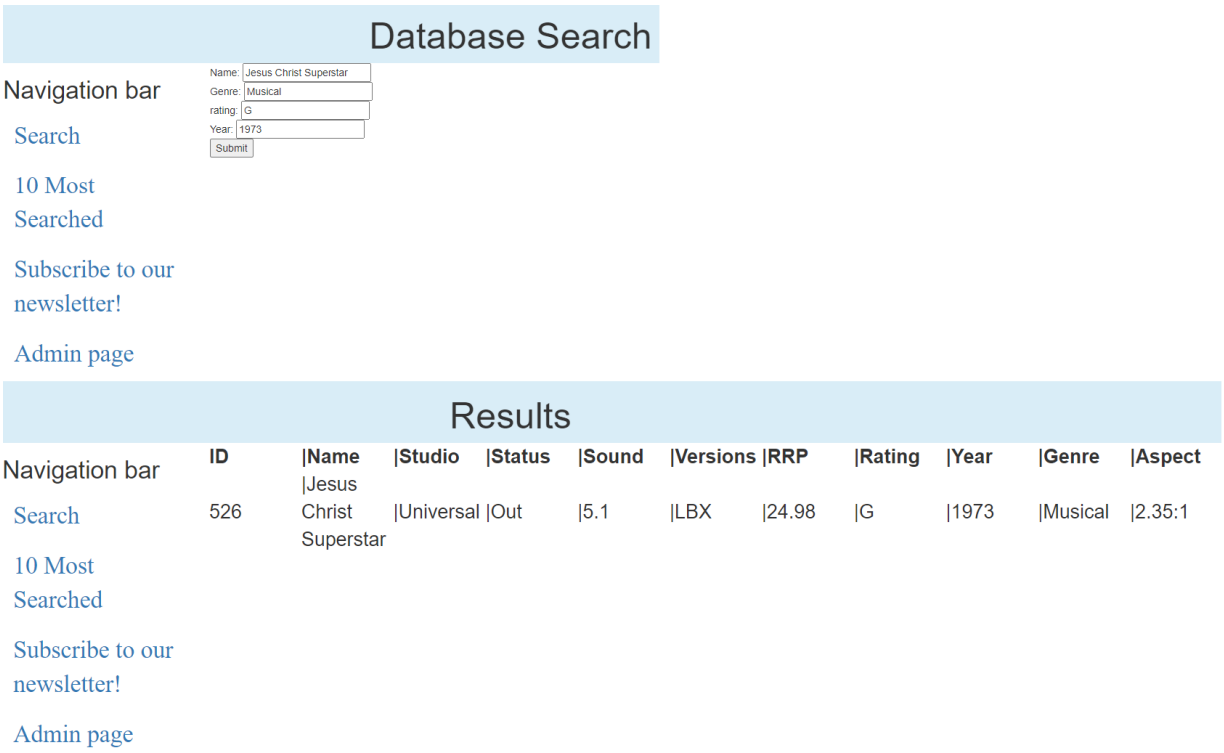


Figure 6- Name, Genre, Rating, Year

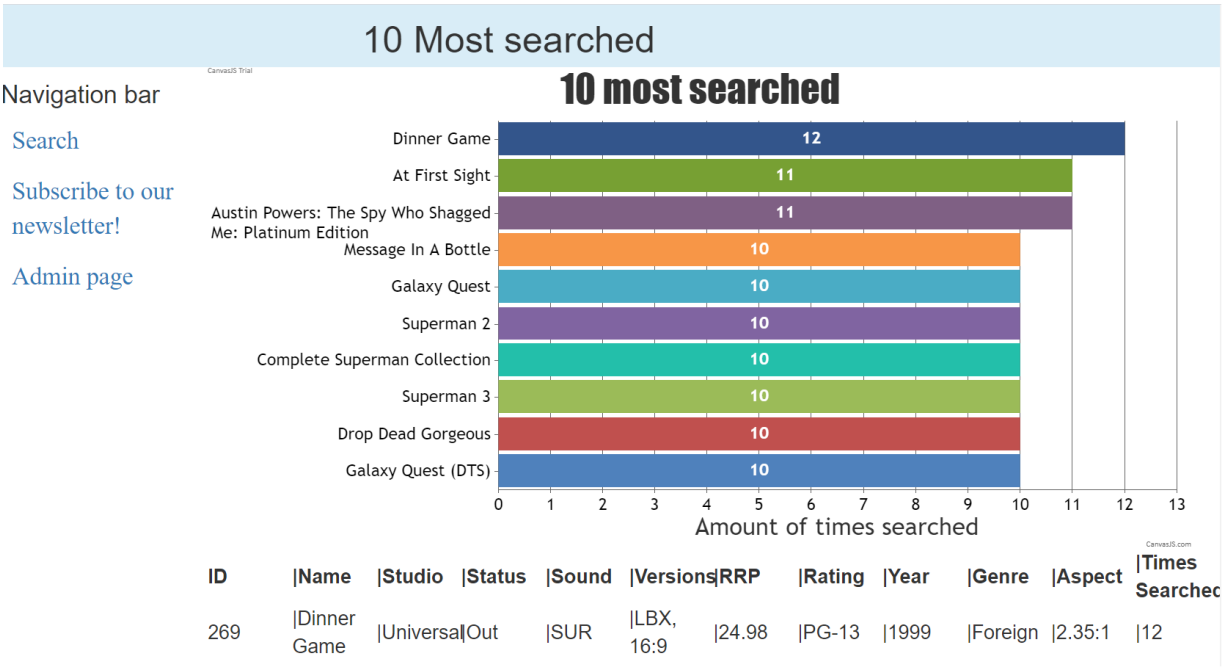


Figure 7- Top ten

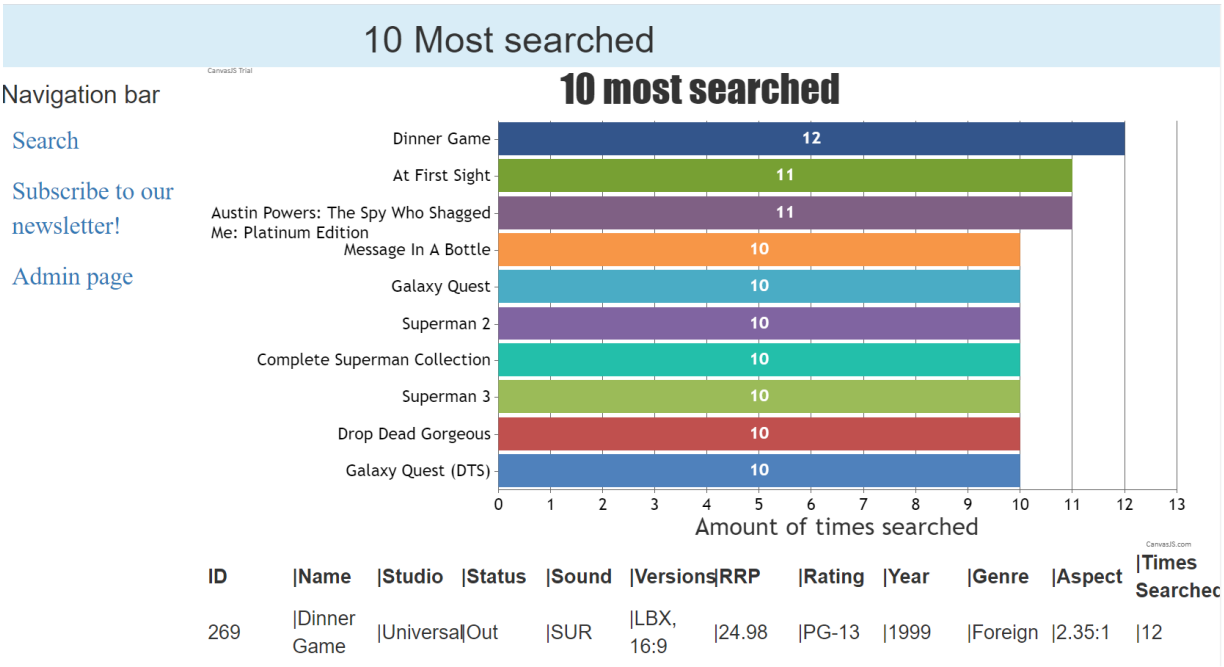


Figure 8- Runs on PC

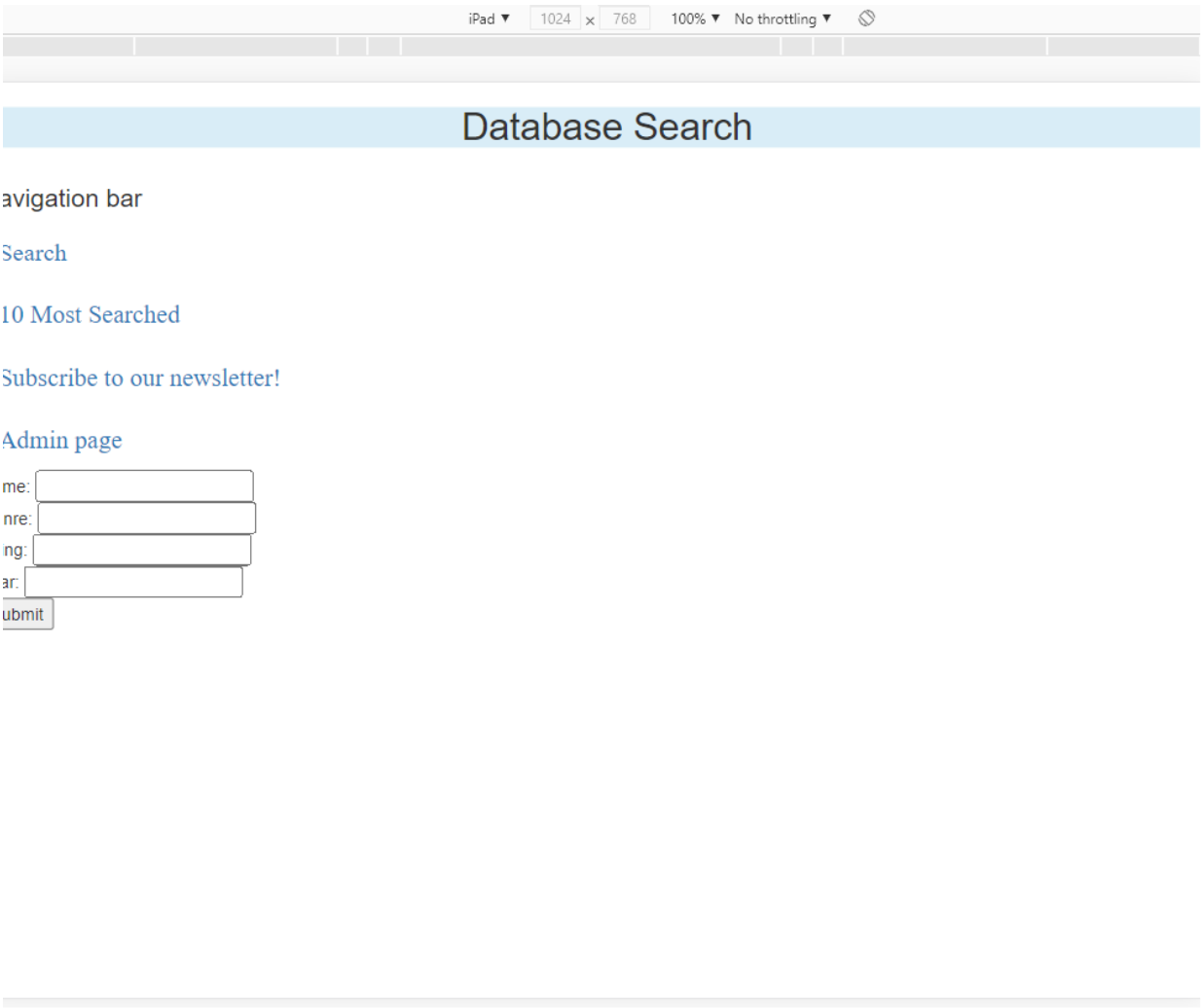


Figure 9- Runs on iPad

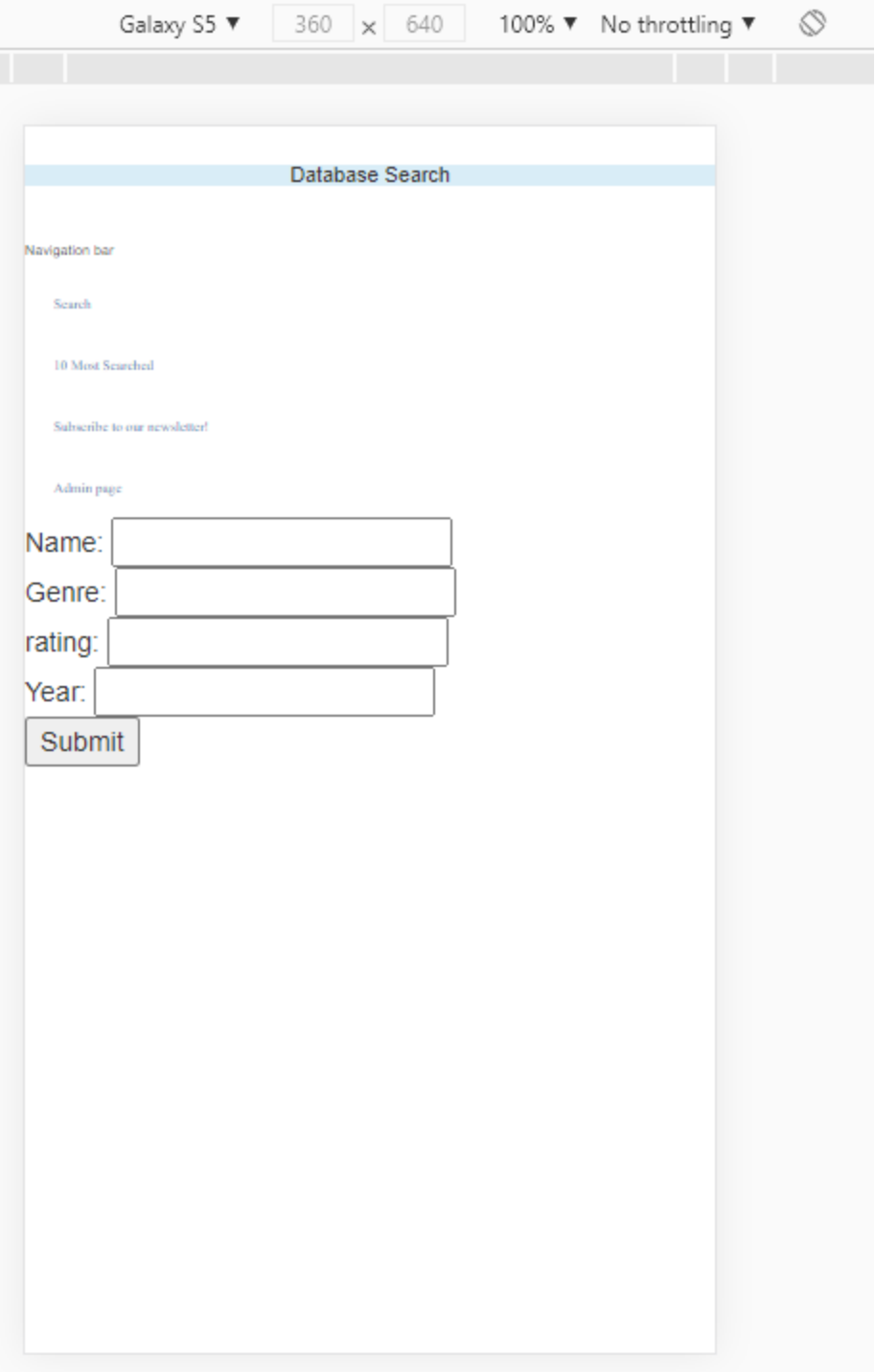


Figure 10- Runs on S5

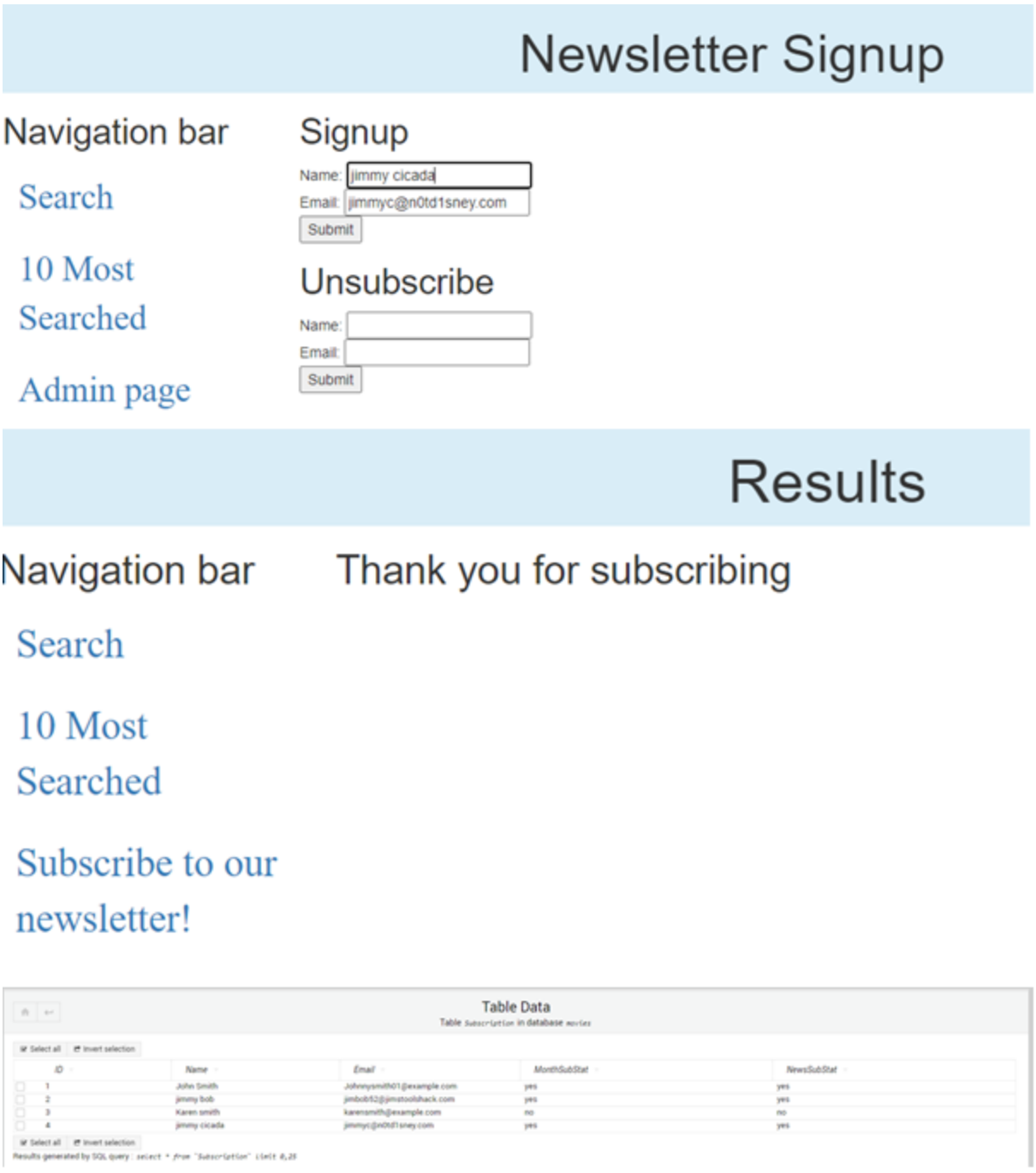


Figure 11- User unsubscribes, database not yet reflected

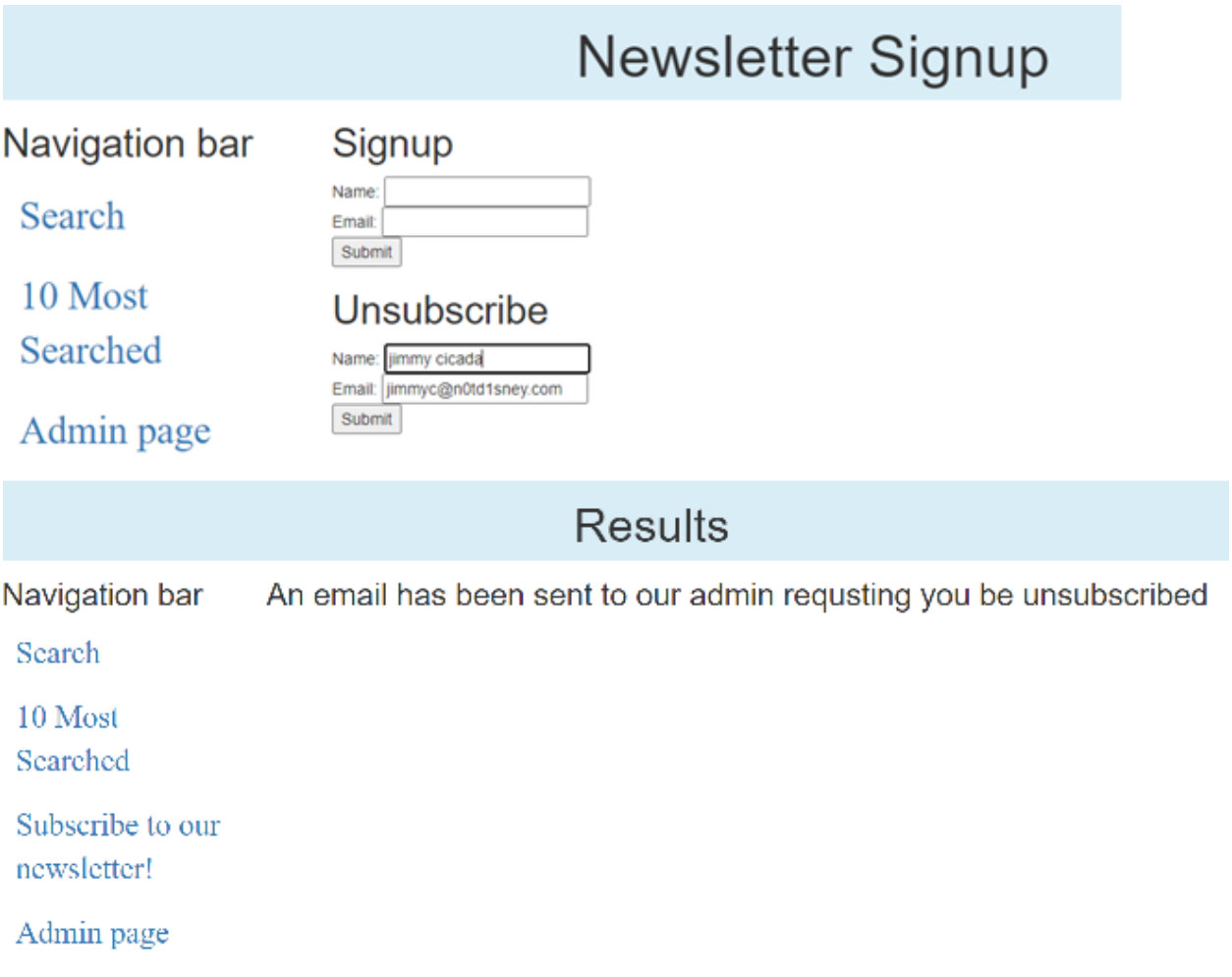


Figure 12- User unsubscribes, admin email

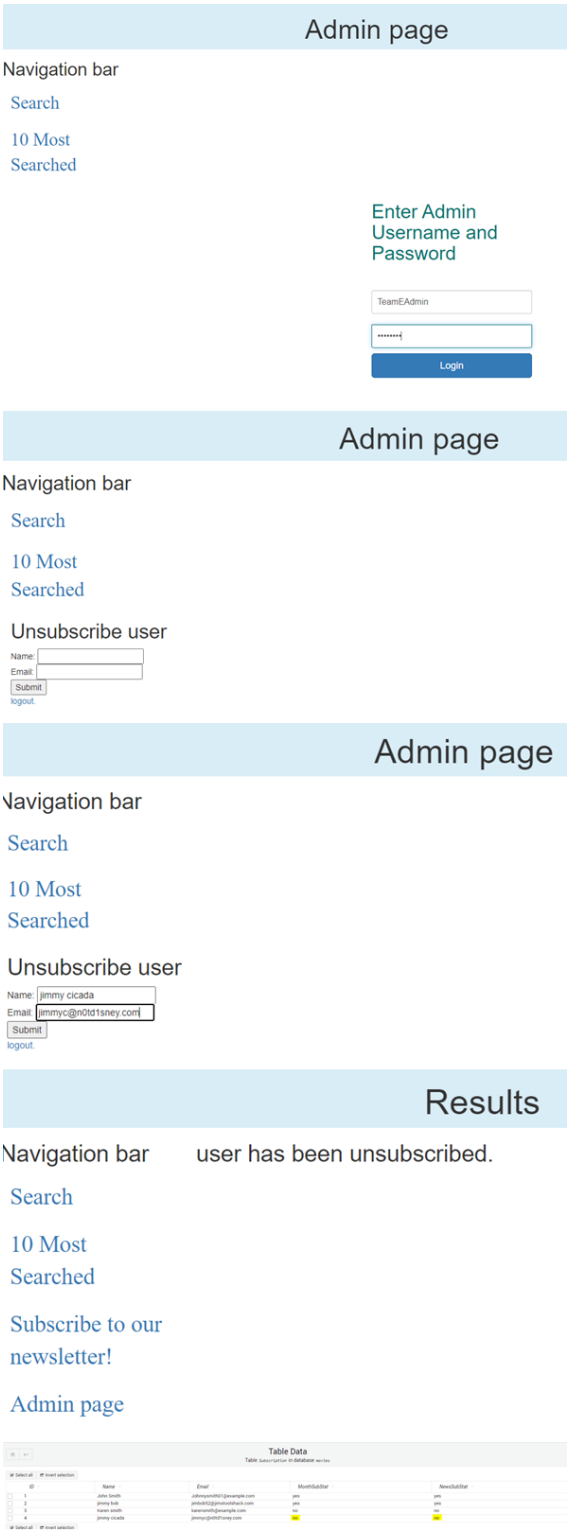


Figure 13- Admin unsubscribes user

Bug Triage 2

Meeting Minutes

08/06/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 10/06/2021, 9:00AM, Murdoch Campus

1. Bug Report

One issue identified and discussed. Not necessarily a bug. Unsubscribe emails going straight to spam folder.

2. Triage (Bug Priority)

Sole issue takes top priority by default.

#ID	DESCRIPTION	SEVERITY	FREQUENCY	RISK	PRIORITY
2	Unsubscribe emails going straight to spam folder	LOW	HIGH	LOW	HIGH

Team Meeting 2

Meeting Minutes

03/06/2021

Present: Blake Topham, Erol Anil, Nathan Tai

Next meeting: 10/06/2021, 9:00AM, Murdoch Campus

1. Team Role Assignment

ANIL- Configuration Manager

Source Control Snapshot

- Update GitHub
- Create screenshots, add to master document

General Version Control

- Manage repositories

Performance Report

GANTT Chart for SPRINT TWO (week seven)

- Timeline
- Include who does what

BLAKE- Developer

Development

- Admin page
- Database tables for:
    - Email addresses
    - Login details if required
- Unsubscribe feature
    - Sends email to admin account
    - Do not allow user to directly manipulate database
    - Admin account required to remove database entries
- Separate options for user to select:
    - Monthly newsletter
    - Newsflash notification email

NATHAN- Scrum Master

Project Management Plan

- Discuss roles
- Record team meeting minutes
- Allocate tasks
- Produce documentation detailing task allocations

Software Review Plan

- Update requirements based on client meeting
- Record minutes of all meetings (client, task allocations, bug triage)
- Review application against new requirements
- Update testing plan to include new features

Demonstrate the Movie Database Application

- Read and understand all documentation
- Read and understand all code
- Demonstrate application in 3 different sized device environments

Client Meeting 2

Meeting Minutes

03/06/2021

Present: Client, Blake Topham, Erol Anil, Nathan Tai

Next meeting: 10/06/2021, 9:30AM, Murdoch Campus

1. Review Requirements

- Project requirements discussed and documented.
- Project planning commenced.
- Additions requested.
    - Subscription feature, choice of:
        - Monthly newsletter
        - Newsflash notifications
- Requirements noted and agreed.

2. User Experience

- Performance review requested
    - Speed
    - Usability