

MOVIE WEBSITE - MOVIEVERSE

A PROJECT REPORT

Submitted By

AYUSH CHAUDHARY

University Roll No 2100290140043

ABHISHEK DAGAR

University Roll No 2100290140004

AKSHAT GARG

University Roll No 2100290140015

**Submitted in partial fulfilment of the
Requirements for the Degree of**

MASTER OF COMPUTER APPLICATION

**Under the Supervision of
Dr. Shashank Bhardwaj
Associate Professor**



Submitted to

**DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206**

(DEC 2022)

CERTIFICATE

Certified that **Ayush Chaudhary (Roll No 2100290140043)** **Abhishek Dagar (Roll No 2100290140004)** **Akshat Garg (Roll No 2100290140015)** have carried out the project work having “**A Movie Website**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

AYUSH CHAUDHARY
University Roll No 2100290140043

ABHISHEK DAGAR
University Roll No 2100290140004

AKSHAT GARG
University Roll No 2100290140015

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Shashank Bhardwaj
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner

Signature of External Examiner

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The project entitled "Movie Website" which is utilized by the user to see the latest movies and download it. Each and every movie details such as the description about the movie will be stored in the repository system in the form of database. That database could be utilized by the mislaid by the users. Each and every movie can be updated lively using this software. As soon as someone checks the dashboard, details of a particular movie can be viewed by a single click on his name. This software is error free anyone can use this software.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Shashank Bhardwaj** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Tripathi Professor** and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

AYUSH CHAUDHARY
University Roll No 2100290140043

ABHISHEK DAGAR
University Roll No 2100290140004

AKSHAT GARG
University Roll No 2100290140015

List of Chapters

Certificate	1
Abstract	2
Acknowledgement	3
Table of Contents	4
Chapter 1. Introduction	5
1.1 Project description	
1.2 Project Scope	
1.3 Hardware / Software used in Project	
Chapter 2. Feasibility Study	8
2.1 Technical feasibility	
2.2 Operational Feasibility	
2.3 Behavioral Feasibility	
2.4 Operational Feasibility	
Chapter 3. Database Design	10
3.1 Database Tables	
3.2 Flow Chart	
3.3 Use Case Diagram	
3.4 Sequence Diagram	
3.5 Collaborative Diagram	
Chapter 4. Software Development Life Cycle Model	
Chapter 5. Testing	16
Chapter 6. Snapshots	24
Chapter 7. Coding	27
Chapter 8. Conclusion	47

Chapter 9. Limitation of Project

Chapter 10. References 49

Chapter 11. Research Paper 50

CHAPTER 1: INTRODUCTION

1.1 PROJECT DESCRIPTION

The project “Movie website” developed in HTML , CSS and JavaScript is used to provide user with an update of the upcoming latest movies. The user can use this website anytime, anywhere to see their favorite movies. This gives original experience of watching the movie by the user. Software or website developed must be built from user’s point of view. It gives the experience of watching the movie by adding the details in the website. The website is maintained by admin where he/she updates the movie without any delay. Adding new movies is the responsibilities of admin.

1.2 PROJECT SCOPE

It may help to see new movies and collecting upcoming movies details. In a very short time, the collection will be obvious, simple and sensible. It will support in mobile as well as in desktop. You can download new movies anywhere in the world but you just have connected to the internet .

1.3 HARDWARE / SOFTWARE USED IN PROJECT

This section describes the software and hardware requirements of the system.

1.3.1 HARDWARE REQUIREMENTS

Hardware components required for installing all the required software environment and tools are:

- Processor – Intel i3 5th generation or higher
- RAM – Minimum 4 GB, recommended 8 GB
- Disk space - Minimum 10 GB of free disk space

1.3.2 SOFTWARE REQUIREMENTS

The software environment used for developing the application is:

- Front End Tool: Html, CSS, JavaScript, Bootstrap.
- Operating System - Windows 10/11 or Ubuntu 18.04 +
- Code editor – Microsoft Visual Studio Code

CHAPTER 2: FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

2.1 Economic Feasibility:

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditure must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only customized products have to be purchased.

2.2 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

2.3 Operational Feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

CHAPTER 3: DATABASE DESIGN

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an ontology.

3.1 FLOW CHART

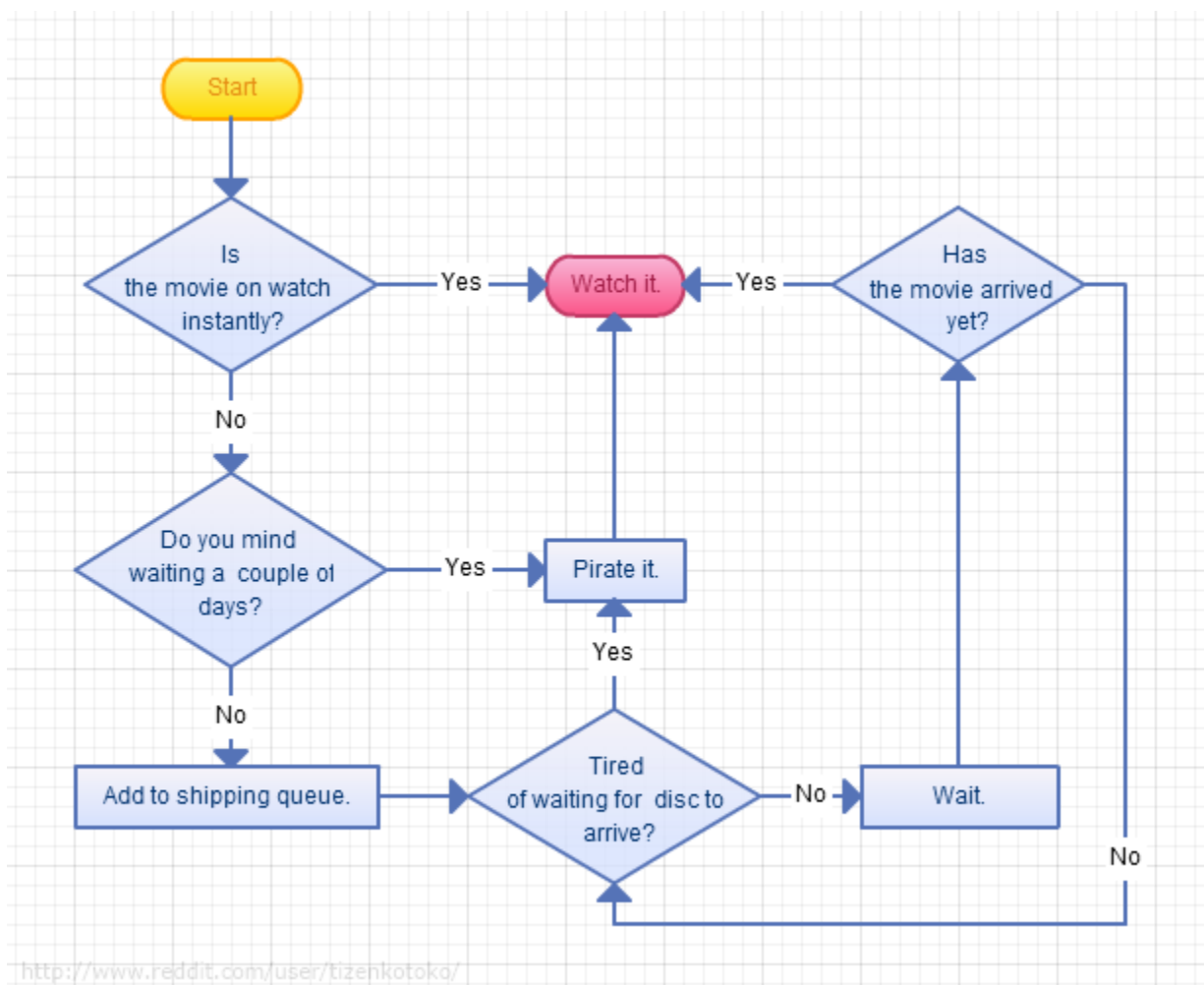


Figure 3.1 (Flowchart for Movie Search Engine)

3.2 DATA FLOW DIAGRAM

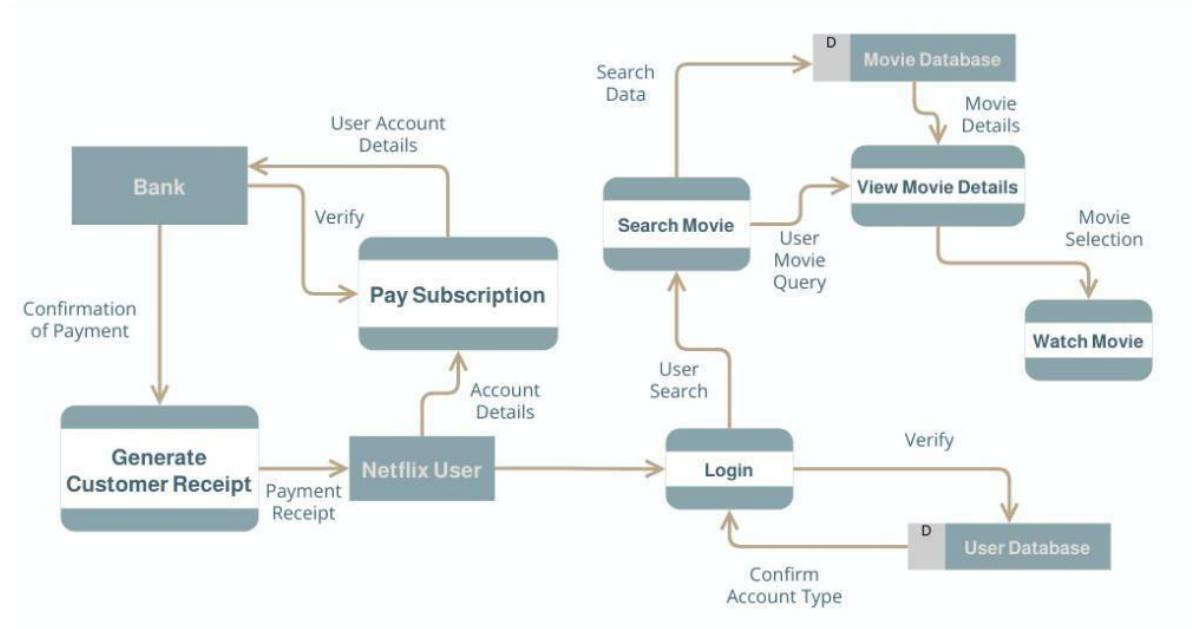


Figure 3.2 (DFD for Movie Download)

3.3 USE CASE DIAGRAM

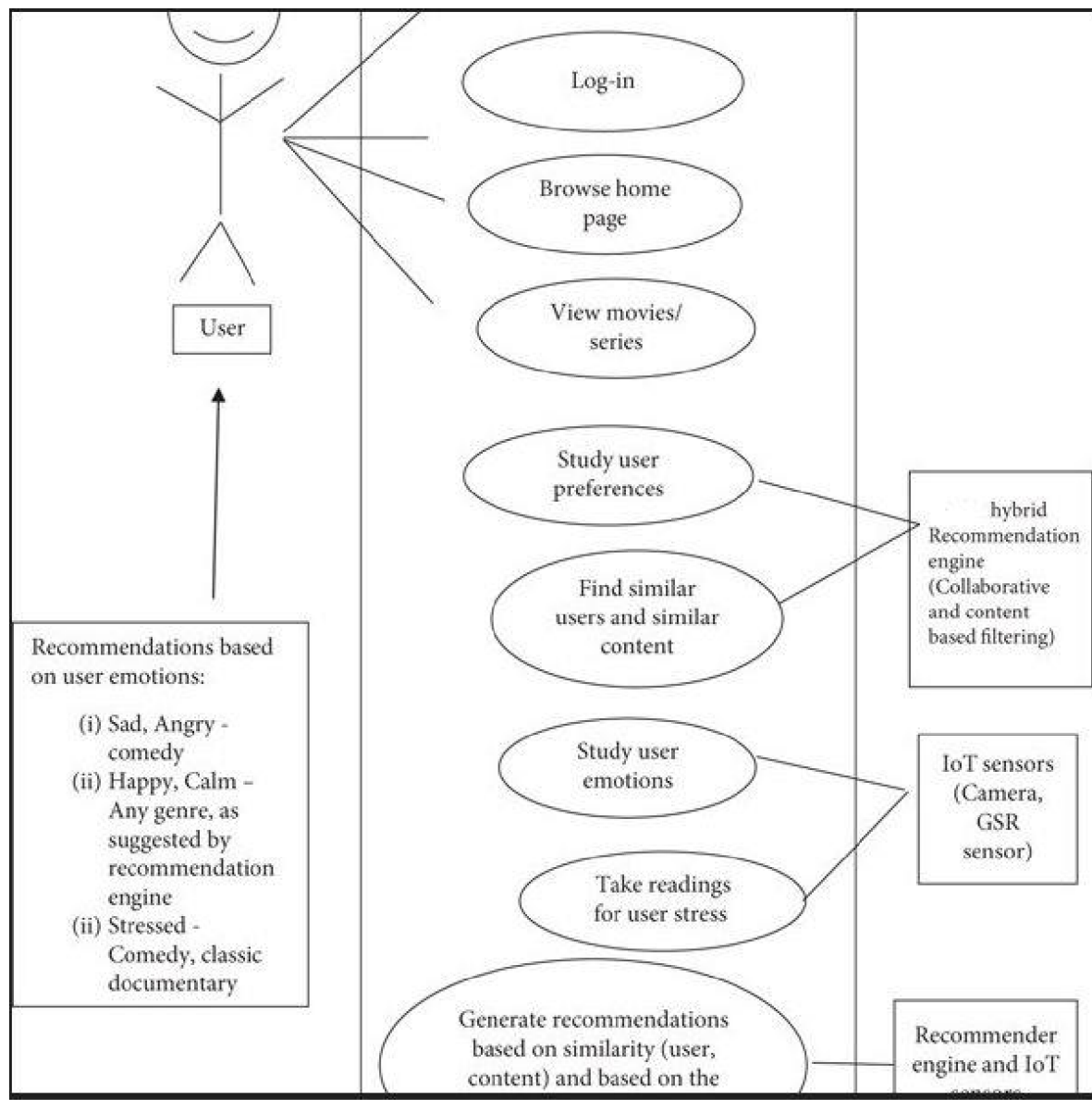


Figure 3.3 (Use Case for Movie Search Engine)

CHAPTER 4: SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

4.1 Waterfall Model

The waterfall model is a well-known structured methodology for software development. The whole process of system development is divided into distinct phases. The model has been introduced in 1970s. Every phase has a unique output. It was the first SDLC model to be used widely. So that, sometime it is referred to Waterfall by SDLC. The waterfall model is used when the system requirements are well known, technology is understood, and the system is a new version of an existing product (Dennis, Wixom and Roth, 2012).

Mainly there are six phases in Waterfall model. If there is a problem faced in any phase of the cycle, the system goes to the previous phase. The phases of Waterfall method is:

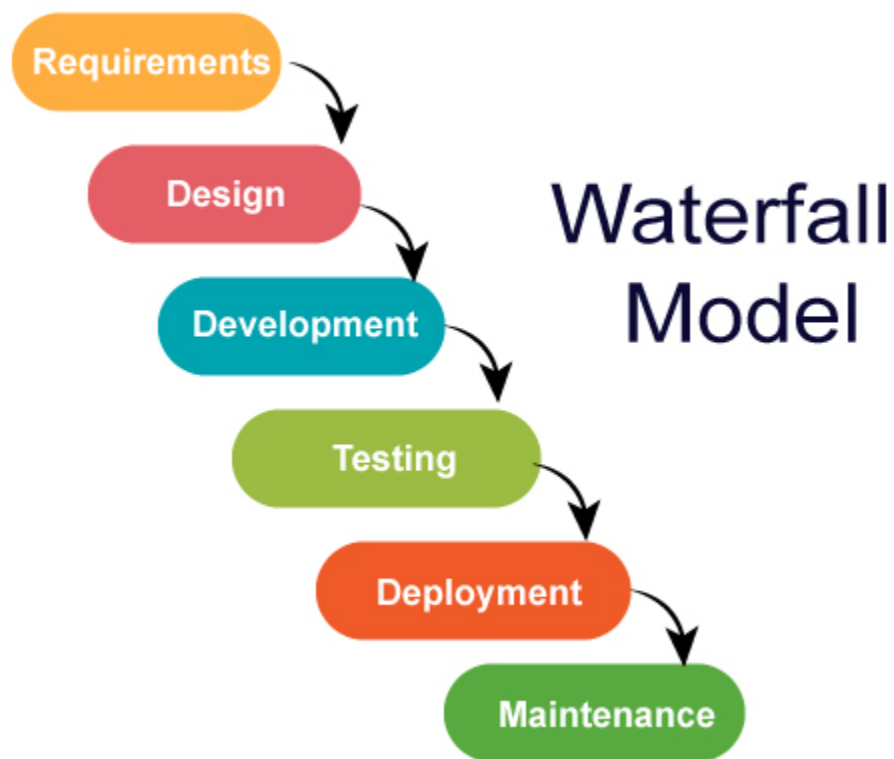


Fig 4.1 (Waterfall Model)

4.2 Requirements Gathering & Analysis:

In this phase, the project title had been selected. The project title for the system was Food Hack. This project starting with brainstorming ideas with supervisor and proposed the title of the project. An abstract and description of the project module has also been done and attached. Besides, the Gantt chart also needed as a guideline and references for the project. This phase is to analyze the existing system and the article of the techniques or method that will be used for this project. In this phase also get all the requirements that are needed to design and develop the new system. Based on the collection of information through article, method and technique that is suitable been decided.

4.3 System Design:

The requirements documented in previous phase are studied in this phase and the system design is prepared. All the data or requirement obtained during planning and analysis phase transformed into the design.

4.4 Implementation:

With inputs from system design, the system is developed in several units. Then the units are tested. This phase is where the design will implement into the coding. The system will develop regarding the user and system requirement. In this project, to develop the system will be use Visual Studio Code to code, Bootstrap as framework and Google Firebase Fire store as database and NodeJS for backend. This phase is a critical phase because user part needed to fulfil and to make sure the objectives accomplish.

4.5 Integration & Testing:

The units of the program developed in previous phase are integrated into a system. Then the whole system is tested. This testing phase will test the system to check the error and ensure the function run well as a whole system. Any error or bugs will be fixed and repeated testing the system until all the function can be use.

4.6 Deployment of the system:

This phase is done when the system has successfully done and fulfill all the objective. System can be deployed and finally the system will publish to the user.

4.7 Maintenance:

There are some issues which are found in the client environment. Patches are released to fix those issues.

CHAPTER 5: TESTING

Testing is vital for the success of any software. No system design is ever perfect. Testing is also carried in two phases, first is during the software engineering that is during the module creation, second phase is after the completion of software, this is system testing which verifies that the whole set of programs hanged together.

5.1 White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. All logical parts of the software checked once. Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. White box testing tests all the independent parts within a module and logical decisions on their true and the false side are exercised. All loops and bound within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

5.2 Black Box Testing:

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. Black Box tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

5.3 Alpha Testing:

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

5.4 Beta Testing:

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

5.5 Unit Testing:

Each module is considered independently. It focuses on each until of software as implemented in the source code. It is white box testing.

5.6 Integration Testing:

Integration Testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. Modules are integrated by using the top down approach.

5.7 Validation Testing:

Validation testing was performed to ensure that all the functional and performance requirements are met.

5.8 System Testing:

It is executing programs to check logical changes made in it with intention of finding errors. A system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

5.9 Implementation and Software Specification

5.9.1 Detailed Design of implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures. To evaluate design and operating specifications and/or provide the basis for further modification.

5.9.2 Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

5.9.3 Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job systems, subsystems, and for the system as a whole.

5.9.4 Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

5.9.5 User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

5.9.6 Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

5.9.7 Installation Phase

In this phase the new Computerized system is installed the conversion to new procedures is fully implemented and the potential of the new system is explored.

5.9.8 System Installation

The process of starting the actual use of a system and training user personnel in its operation.

5.9.9 Review Phase

This phase evaluate the successes and failures during a systems development project, and to measure the results of a new Computerized Tran system in terms of benefits and savings projected at the start of the project.

5.9.10 Development Recap

A review of a project immediately after completion to find successes and potential problems in future work.

5.9.11 Post-Implementation Review

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

5.10 THE STEPS IN THE SOFTWARE TESTING

The steps involved during Unit Testing are as follows:

- a. Preparation of the test cases.
- b. Preparation of the possible test data with all the validation checks.
- c. Complete code review of the module.
- d. Actual testing done manually.
- e. Modifications done for the errors found during testing.
- f. Prepared the test result scripts.

5.11 The unit testing done included the testing of the following items:

- 1 Functionality of the entire module/forms.
- 2 Validations for user input.
- 3 Checking of the Coding standards to be maintained during coding.
- 4 Testing the module with all the possible test data.
- 5 Testing of the functionality involving all type of calculations etc.
- 6 Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, We integrated the modules

one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

5.12 The steps involved during System testing are as follows:

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Prepared the test result scripts after rectification of the errors.

5.13 The System Testing done included the testing of the following items:

- Functionality of the entire system as a whole.
- User Interface of the system.
- Testing the dependent modules together with all the possible test data scripts.
- Verification and Validation testing.
- Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing Clients at their end this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

There are other six tests, which fall under special category. They are described below:

- **Peak Load Test:** It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. Forexample, test the system by activating all terminals at the same time.
- **Storage Testing:** It determines the capacity of the system to store transaction data on a disk or in other files.
- **Performance Time Testing:** It determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.
- **Recovery Testing:** This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.
- **Procedure Testing:** It determines the clarity of documentation on operation and users of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.
- **Human Factors Testing:** It determines how users will use the system when processing data or preparing reports.

CHAPTER 6: SNAPSHOTS

6.1 HOMEPAGE

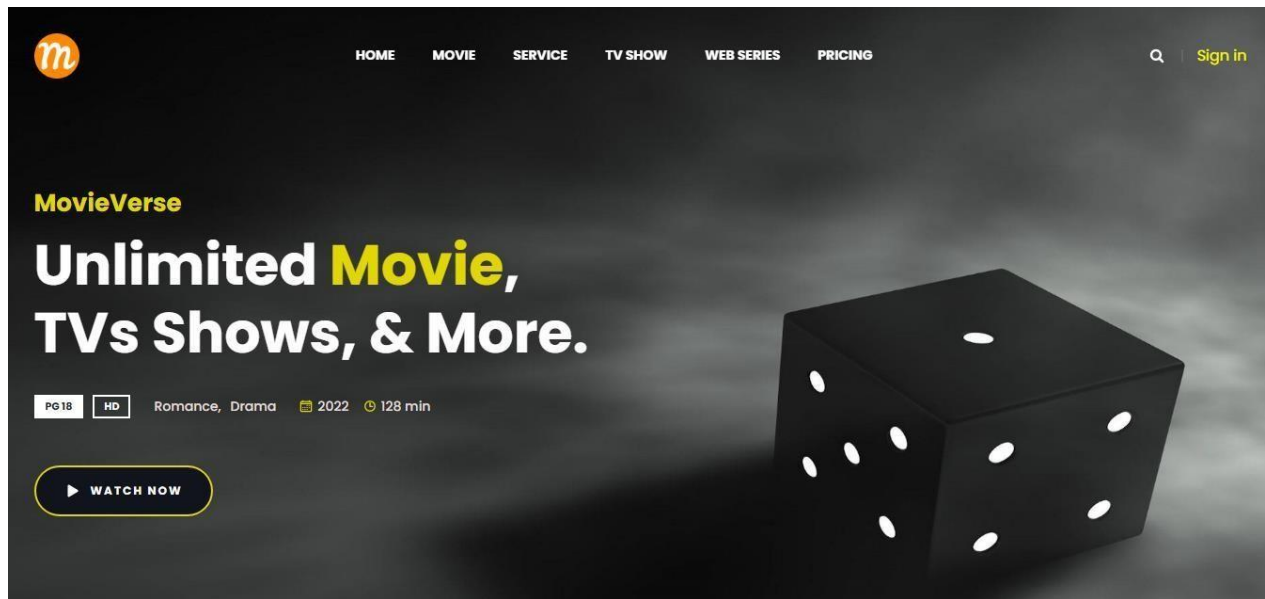


Figure 6.1(Homepage Module)

6.2 CATEGORIES PAGE

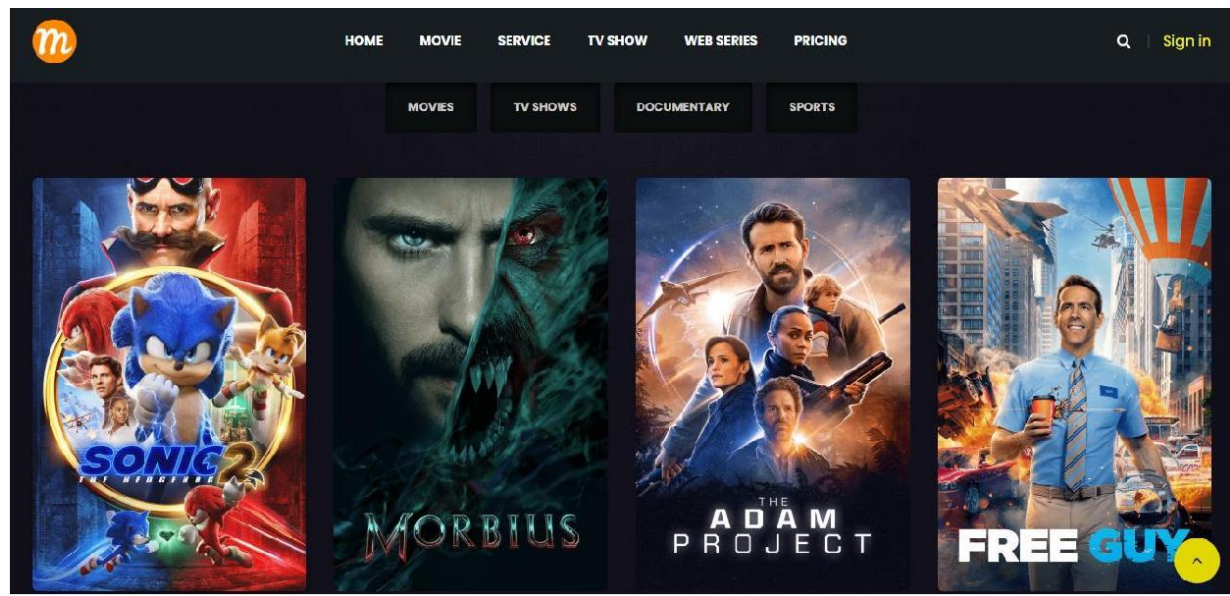


Figure 6.2 (Category Module)

6.2 MOVIE SEARCH

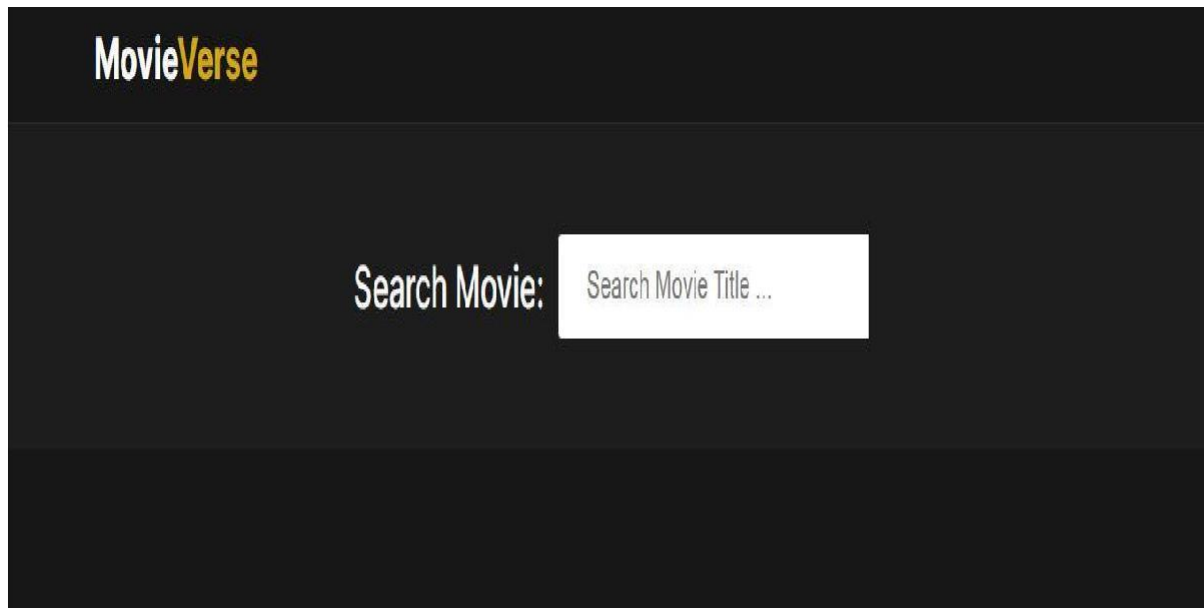


Figure 6.3(Search Module)

6.3 SERVICE

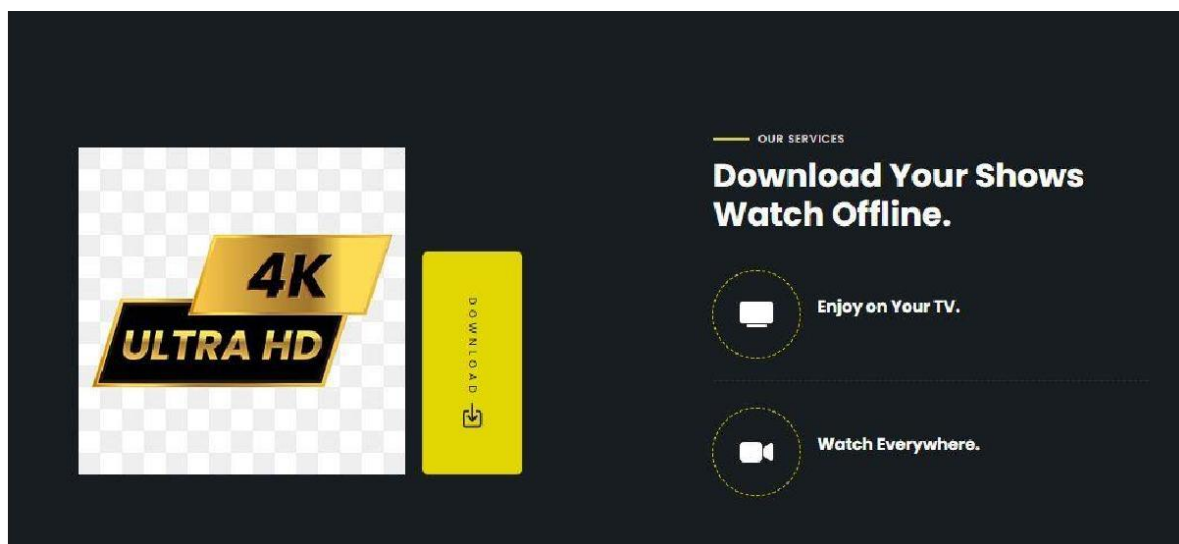


Figure 6.4 (Service Module)

CHAPTER 7 CODING

```
<!-- #UPCOMING-->

<section class="upcoming" id="MOVIE">
  <div class="container">
    <div class="flex-wrapper">
      <div class="title-wrapper">
        <p class="section-subtitle">Online Streaming</p>
        <h2 class="h2 section-title">Upcoming Movies</h2>
      </div>

      <ul class="filter-list">
        <li>
          <button class="filter-btn">Movies</button>
        </li>

        <li><button class="filter-btn">TV Shows</button></li>
        <li><button class="filter-btn">Anime</button></li>

      </ul>
    </div>
    <ul class="movies-list has-scrollbar">
      <li>
        <div class="movie-card">
          <a href="/the_northman_details.html">
            <figure class="card-banner">
              
            </figure>
          </a>

          <div class="title-wrapper">
            <a href="/the_northman_details.html">
              <h3 class="card-title">The Northman</h3>
            </a>

            <time datetime="2022">2022</time>
          </div>
          <div class="card-meta">
            <div class="badge badge-outline">HD</div>
            <div class="duration">
              <ion-icon name="time-outline"></ion-icon>
              <time datetime="PT137M">137 min</time>
            </div>

            <div class="rating">
              <ion-icon name="star"></ion-icon>
```

```

<data>8.5</data>
</div>
</div>
</div>
</li>

<li>
<div class="movie-card">
<a href="/strange_details.html">
<figure class="card-banner">

</figure>
</a>

<div class="title-wrapper">
<a href="/strange_details.html">
<h3 class="card-title">Doctor Strange in the Multiverse of Madness</h3>
</a>
<time datetime="2022">2022</time>
</div>

<div class="card-meta">
<div class="badge badge-outline">4K</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT126M">126 min</time>
</div>
<div class="rating">
<ion-icon name="star"></ion-icon>
<data>NR</data>
</div>
</div>
</div>
</li>

<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>

<div class="title-wrapper">
<a href="#">
<h3 class="card-title">Memory</h3>
</a>

```

```

<time datetime="2022">2022</time>
</div>

<div class="card-meta">
<div class="badge badge-outline">2K</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>

<time datetime="">N/A</time>
</div>
<div class="rating">
<ion-icon name="star"></ion-icon>
<data>NR</data>
</div>
</div>
</div>
</div>
</li>

<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>

<div class="title-wrapper">
<a href="#">
<h3 class="card-title">The Unbearable Weight of Massive Talent</h3>
</a>
<time datetime="2022">2022</time>
</div>
<div class="card-meta">
<div class="badge badge-outline">HD</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT107M">107 min</time>
</div>
<div class="rating">
<ion-icon name="star"></ion-icon>

<data>NR</data>
</div>
</div>
</div>
</div>
</li>
</ul>
</div>
</section>

```

SERVICE

<!-- #SERVICE-->

```
<section class="service" id="SERVICE">
<div class="container">
<div class="service-banner">
<figure>

</figure>

<a href="/images/ultra hd.png" download class="service-btn">
<span>Download</span>
<ion-icon name="download-outline"></ion-icon>
</a>
</div>

<div class="service-content">
<p class="service-subtitle">Our Services</p>
<h2 class="h2 service-title">Download Your Shows Watch Offline.</h2>
<p class="service-text">
</p>

<ul class="service-list">
<li>
<div class="service-card">
<div class="card-icon">
<ion-icon name="tv"></ion-icon>
</div>

<div class="card-content">
<h3 class="h3 card-title">Enjoy on Your TV.</h3>
<p class="card-text">
</p>
</div>
</div>
</li>
<li>
<div class="service-card">
<div class="card-icon">
<ion-icon name="videocam"></ion-icon>
</div>
<div class="card-content">
<h3 class="h3 card-title">Watch Everywhere.</h3>
<p class="card-text">
</p>
</div>
</div>
```

```

</div>
</li>
</ul>

</div>
</div>
</section>

```

TOP RATED

```

<!--#TOP RATED-->

```

```

<section class="top-rated" id="TV SHOW">
<div class="container">
<p class="section-subtitle">Online Streaming</p>
<h2 class="h2 section-title">Top Rated Movies</h2>
<ul class="filter-list">
<li>
<button class="filter-btn">Movies</button>
</li>

<li>
<button class="filter-btn">TV Shows</button>
</li>

<li>
<button class="filter-btn">Documentary</button>
</li>

<li>
<button class="filter-btn">Sports</button>
</li>
</ul>

<ul class="movies-list">
<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>

<div class="title-wrapper">
<a href="#">

```

```

<h3 class="card-title">Sonic the Hedgehog 2</h3>
</a>
<time datetime="2022">2022</time>
</div>
<div class="card-meta">
<div class="badge badge-outline">2K</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>

<time datetime="PT122M">122 min</time>
</div>
<div class="rating">
<ion-icon name="star"></ion-icon>
<data>7.8</data>
</div>
</div>
</div>
</li>

<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>

<div class="title-wrapper">
<a href="#">
<h3 class="card-title">Morbius</h3>
</a>

<time datetime="2022">2022</time>
</div>
<div class="card-meta">
<div class="badge badge-outline">HD</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT104M">104 min</time>
</div>
<div class="rating">
<ion-icon name="star"></ion-icon>
<data>5.9</data>
</div>
</div>
</div>
</li>

```



```
<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>
```

```
<div class="title-wrapper">
```

```
<a href="#">
<h3 class="card-title">The Adam Project</h3>
</a>
```

```
<time datetime="2022">2022</time>
</div>
<div class="card-meta">
<div class="badge badge-outline">4K</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT106M">106 min</time>
</div>
```

```
<div class="rating">
<ion-icon name="star"></ion-icon>
<data>7.0</data>
</div>
</div>
</div>
</li>
```

```
<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>
<div class="title-wrapper">
<a href="#">
<h3 class="card-title">Free Guy</h3>
</a>
<time datetime="2021">2021</time>
</div>
```

```
<div class="card-meta">
```

```

<div class="badge badge-outline">4K</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT115M">115 min</time>
</div>

<div class="rating">
<ion-icon name="star"></ion-icon>
<data>7.7</data>
</div>
</div>
</div>
</li>
<li>

<div class="movie-card">
<a href="#">
<figure class="card-banner">


</figure>
</a>

<div class="title-wrapper">
<a href="#">
<h3 class="card-title">The Batman</h3>
</a>

<time datetime="2022">2022</time>
</div>
<div class="card-meta">
<div class="badge badge-outline">4K</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT176M">176 min</time>
</div>

<div class="rating">
<ion-icon name="star"></ion-icon>
<data>7.9</data>
</div>
</div>
</div>
</li>
<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

```

```

</figure>
```

```
</a>
```

```
<div class="title-wrapper">
```

```
<a href="#">
```

```
<h3 class="card-title">Uncharted</h3>
```

```
</a>
```

```
<time datetime="2022">2022</time>
```

```
</div>
```

```
<div class="card-meta">
```

```
<div class="badge badge-outline">HD</div>
```

```
<div class="duration">
```

```
<ion-icon name="time-outline"></ion-icon>
```

```
<time datetime="PT116M">116 min</time>
```

```
</div>
```

```
<div class="rating">
```

```
<ion-icon name="star"></ion-icon>
```

```
<data>7.0</data>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</li>
```

```
<li>
```

```
<div class="movie-card">
```

```
<a href="#">
```

```
<figure class="card-banner">
```

```

```

```
</figure>
```

```
</a>
```

```
<div class="title-wrapper">
```

```
<a href="#">
```

```
<h3 class="card-title">Death on the Nile</h3>
```

```
</a>
```

```
<time datetime="2022">2022</time>
```

```
</div>
```

```
<div class="card-meta">
```

```
<div class="badge badge-outline">2K</div>
```

```
<div class="duration">
```

```
<ion-icon name="time-outline"></ion-icon>
```

```

<time datetime="PT127M">127 min</time>
</div>

<div class="rating">
<ion-icon name="star"></ion-icon>
<data>6.5</data>
</div>
</div>
</div>
</li>

<li>
<div class="movie-card">
<a href="#">
<figure class="card-banner">

</figure>
</a>

<div class="title-wrapper">
<a href="#">
<h3 class="card-title">The King's Man</h3>
</a>
<time datetime="2021">2021</time>
</div>
<div class="card-meta">
<div class="badge badge-outline">HD</div>
<div class="duration">
<ion-icon name="time-outline"></ion-icon>
<time datetime="PT131M">131 min</time>

</div>

<div class="rating">
<ion-icon name="star"></ion-icon>
<data>7.0</data>
</div>
</div>
</div>
</li>
</ul>
</div>
</section>

```

FOOTER

```
<!-- #FOOTER-->
```

```
<footer class="footer">
<div class="footer-top">
<div class="container">
<div class="footer-brand-wrapper">
<a href="/index.html" class="logo">

</a>
```

```
<ul class="footer-list">
<li>
<a href="/index.html" class="footer-link">Home</a>
</li>
```

```
<li>
<a href="#" class="footer-link">Movie</a>
</li>
<li>
<a href="#" class="footer-link">TV Show</a>
</li>
```

```
<li>
<a href="#" class="footer-link">Web Series</a>
</li>
<li>
<a href="#" class="footer-link">Pricing</a>
</li>
</ul>
</div>
```

```
<div class="divider"></div>
<div class="quicklink-wrapper">
```

```
<ul class="quicklink-list">
<li>
<a href="#" class="quicklink-link">Faq</a>
</li>
```

```
<li>
<a href="#" class="quicklink-link">Help center</a>
</li>
```

```
<li>
```

```
<a href="#" class="quicklink-link">Terms of use</a>
</li>
```

```
<li>
<a href="#" class="quicklink-link">Privacy</a>
</li>
</ul>
```

```
<ul class="social-list">
<li>
<a href="#" class="social-link">
<ion-icon name="logo-facebook"></ion-icon>
</a>
</li>
```

```
<li>
<a href="#" class="social-link">
<ion-icon name="logo-twitter"></ion-icon>
</a>
</li>
```

```
<li>
<a href="#" class="social-link">
<ion-icon name="logo-pinterest"></ion-icon>
</a>
</li>
```

```
<li>
<a href="#" class="social-link">
<ion-icon name="logo-linkedin"></ion-icon>
</a>
</li>
```

```
</ul>
</div>
</div>
</div>
```

```
<div class="footer-bottom">
<div class="container">
<p class="copyright">
&copy; 2022 <a href="#"></a>. All Rights Reserved
```

```
</p>
```

```

```

</div>

</div>

</footer>

<!-- #GO TO TOP-->

<ion-icon name="chevron-up"></ion-icon>

<!-- custom js link-->

<script src="./main.js"></script>

<script src="./main1.js"></script>

<!-- ionicon link-->

<script type="module"

src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>

<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>

</body>

</html>

MOVIE SEARCH

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Movie Search</title>

<!-- font awesome icons cdn -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css" integrity="sha512-
Fo3rlrZj/k7ujTnHg4CGR2D7kSs0v4LLanw2qksYuRIEzO+tcaEPQogQ0KaoGN26/zrn20ImR1
DfuL WnOo7aBA==" crossorigin="anonymous" referrerpolicy="no-referrer" />

<!-- custom css -->
<link rel = "stylesheet" href = "./Movie_search.css">
</head>
<body>

<div class = "wrapper">

<!-- logo -->
<div class = "logo">
<div class = "container">
<p>Movie<span>Verse</span></p>
</div>
</div>

<!-- end of logo -->
<!-- search container -->

<div class = "search-container">
<div class = "search-element">
<h3>Search Movie:</h3>
<input type = "text" class = "form-control" placeholder="Search Movie Title ..." id = "movie-
search-box" onkeyup="findMovies()" onclick = "findMovies()">

<div class = "search-list" id = "search-list">

<!-- list here -->
<!-- <div class = "search-list-item">

<div class = "search-item-thumbnail">
```



```
<img src = "medium-cover.jpg">
</div>
```

```
<div class = "search-item-info">
<h3>Guardians of the Galaxy Vol. 2</h3>
<p>2017</p>
</div>
</div> -->
</div>
</div>
</div>
```

```
<!-- end of search container -->
```

```
<!-- result container -->
<div class = "container">
<div class = "result-container">
<div class = "result-grid" id = "result-grid">
```

```
<!-- movie information here -->
<!-- <div class = "movie-poster">
```

```
<img src = "medium-cover.jpg" alt = "movie poster">
</div>
```

```
<div class = "movie-info">
<h3 class = "movie-title">Guardians of the Galaxy Vol. 2</h3>
<ul class = "movie-misc-info">
<li class = "year">Year: 2017</li>
<li class = "rated">Ratings: PG-13</li>
<li class = "released">Released: 05 May 2017</li>
</ul>
<p class = "genre"><b>Genre:</b> Action, Adventure, Comedy</p>
<p class = "writer"><b>Writer:</b> James Gunn, Don Abnett, Andy Lanning</p>
<p class = "actors"><b>Actors:</b> Chris Pratt, Zoe Saldana, Dave Bautista</p>
<p class = "plot"><b>Plot:</b> The Guardians struggle to keep together as a team while dealing
with their personal family issues, notably Star-Lord's encounter with his father the ambitious
celestial being Ego.</p>
<p class = "language"><b>Language:</b> English</p>
<p class = "awards"><b><i class = "fas fa-award"></i></b> Nominated for 1 Oscar</p>
</div> -->
</div>
</div>
</div>
```

```
<!-- end of result container -->
</div>

<!-- movie app js -->
<script src = "Movie_search.js"></script>
</body>
</html>
```

JAVASCRIPT

```
// Titles: https://omdbapi.com/?s=thor&page=1&apikey=fc1fef96
// details: http://www.omdbapi.com/?i=tt3896198&apikey=fc1fef96

const movieSearchBox = document.getElementById('movie-search-box');
const searchList = document.getElementById('search-list');
const resultGrid = document.getElementById('result-grid');

// load movies from API
async function loadMovies(searchTerm){
  const URL = `https://omdbapi.com/?s=${searchTerm}&page=1&apikey=fc1fef96`;
  const res = await fetch(`${URL}`);
  const data = await res.json();
  // console.log(data.Search);
  if(data.Response === "True") displayMovieList(data.Search);
}

function findMovies(){
  let searchTerm = (movieSearchBox.value).trim();
  if(searchTerm.length > 0){
    searchList.classList.remove('hide-search-list');
    loadMovies(searchTerm);
  } else {
    searchList.classList.add('hide-search-list');
  }
}

function displayMovieList(movies){
  searchList.innerHTML = "";
  for(let idx = 0; idx < movies.length; idx++){
    let movieListItem = document.createElement('div');
    movieListItem.dataset.id = movies[idx].imdbID; // setting movie id in data-id
    movieListItem.classList.add('search-list-item');
    if(movies[idx].Poster !== "N/A")
      moviePoster = movies[idx].Poster;
    else
      moviePoster = "image_not_found.png";

    movieListItem.innerHTML = `
    <div class = "search-item-thumbnail">
      <img src = "${moviePoster}">
    </div>
    <div class = "search-item-info">
      <h3>${movies[idx].Title}</h3>
      <p>${movies[idx].Year}</p>
    </div>
    `;
    searchList.appendChild(movieListItem);
  }
}
```

```

    }
    loadMovieDetails();
}

function loadMovieDetails(){
    const searchListMovies = searchList.querySelectorAll('.search-list-item');
    searchListMovies.forEach(movie => {
        movie.addEventListener('click', async () => {
            // console.log(movie.dataset.id);
            searchList.classList.add('hide-search-list');
            movieSearchBox.value = "";
            const result = await
fetch(`http://www.omdbapi.com/?i=${movie.dataset.id}&apikey=fc1fef96`);
            const movieDetails = await result.json();
            // console.log(movieDetails);
            displayMovieDetails(movieDetails);
        });
    });
}

function displayMovieDetails(details){
    resultGrid.innerHTML = `
    <div class = "movie-poster">
        <img src = "${(details.Poster !== "N/A") ? details.Poster : "image_not_found.png"}" alt =
"movie poster">
    </div>
    <div class = "movie-info">
        <h3 class = "movie-title">${details.Title}</h3>
        <ul class = "movie-misc-info">
            <li class = "year">Year: ${details.Year}</li>
            <li class = "rated">Ratings: ${details.Rated}</li>
            <li class = "released">Released: ${details.Released}</li>
        </ul>
        <p class = "genre"><b>Genre:</b> ${details.Genre}</p>
        <p class = "writer"><b>Writer:</b> ${details.Writer}</p>
        <p class = "actors"><b>Actors:</b> ${details.Actors}</p>
        <p class = "plot"><b>Plot:</b> ${details.Plot}</p>
        <p class = "language"><b>Language:</b> ${details.Language}</p>
        <p class = "awards"><b><i class = "fas fa-award"></i></b> ${details.Awards}</p>
    </div>
    `;
}

window.addEventListener('click', (event) => {
    if(event.target.className !== "form-control"){
        searchList.classList.add('hide-search-list');
    }
});

```

SIGN UP

```
<!DOCTYPE html>
<html>
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box}

/* Full-width input fields */
input[type=text], input[type=password] {
width: 100%;
padding: 15px;
margin: 5px 0 22px 0;
display: inline-block;
border: none;
background: #f1f1f1;
}

input[type=text]:focus, input[type=password]:focus {
background-color: #ddd;
outline: none;
}

hr {
border: 1px solid #f1f1f1;
margin-bottom: 25px;
}

/* Set a style for all buttons */
button {
background-color: #04AA6D;
color: white;
padding: 14px 20px;
margin: 8px 0;
border: none;
cursor: pointer;
width: 100%;
opacity: 0.9;
}

button:hover {
opacity: 1;
}

/* Extra styles for the cancel button */
.cancelbtn {
padding: 14px 20px;
background-color: #f44336;
```

```

}

/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {

    float: left;
    width: 50%;
}

/* Add padding to container elements */
.container {
    padding: 16px;
}

/* Clear floats */
.clearfix::after {
    content: "";
    clear: both;
    display: table;
}

/* Change styles for cancel button and signup button on extra small screens */
@media screen and (max-width: 300px) {
    .cancelbtn, .signupbtn {
        width: 100%;
    }
}
</style>
<body>

<form action="/action_page.php" style="border: 1px solid #ccc">
<div class="container">
<h1>Sign Up</h1>
<p>Please fill in this form to create an account.</p>
<hr>

<label for="email"><b>Email</b></label>
<input type="text" placeholder="Enter Email" name="email" required>

<label for="psw"><b>Password</b></label>
<input type="password" placeholder="Enter Password" name="psw" required>

<label for="psw-repeat"><b>Repeat Password</b></label>
<input type="password" placeholder="Repeat Password" name="psw-repeat" required>

<label>
<input type="checkbox" checked="checked" name="remember" style="margin-bottom: 15px">
Remember me
</label>

```

```
<p>By creating an account you agree to our <a href="#" style="color: dodger blue">Terms &
Privacy</a>.</p>
```

```
<div class="clear fix">
<button type="button" class="cancel btn">Cancel</button>
<button type="submit" class="sign up btn">Sign Up</button>
</div>
</div>
```

```
</form>
```

```
</body>
</html>
```

CHAPTER 8: CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. User friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame word that enables the manager to make reasonable estimates made within a limited timeframe at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points

- A description of the background and context of the project and its relation to word already done in the area.
- Made statement of the aims and objectives of the project.
- The description of purpose, scope, and applicability.
- We define the problem on which we are working on the project.
- We describe the requirement specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system.
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.

CHAPTER 9: LIMITATION OF PROJECT

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by me. Though the software presents a broad range of options to its users some intricate options could not be covered into it, partly because of logistic and partly due to lack of sophistication. Paucity of time was also major constraint, thus it was not possible to make the software foolproof and dynamic. Lack of time also compelled me to ignore some part such as storing old result of the candidate etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at the first instance. The user is provided help at each step for his convenience in working with the software.

CHAPTER 10: REFERENCES

1. <http://getbootstrap.com/>
2. www.google.com
3. www.mysql.com
4. www.w3schools.com
5. www.youtube.com
6. <https://codingartistweb.com/2022/07/movie-guide-app-with-javascript/>
7. <https://lambda.uta.edu/cse4392/project3.html>
8. <https://makeschool.org/mediabook/oa/tutorials/rotten-potatoes---movie-reviews-with-express-js/start-an-express-project/>
9. https://www.w3schools.com/html/html_scripts.asp
10. <https://www.geeksforgeeks.org/which-tag-is-used-to-represent-progress-of-a-task-in-html-meter-tag/?ref=rp>

CHAPTER 11: RESEARCH PAPER

11.1 Abstract

In this paper, we present Film Trust, a website that integrates Semantic Web-based social networks, augmented with trust, to create predictive movie recommendations. We show how these recommendations are more accurate than other techniques in certain cases, and discuss this technique as a mechanism of Semantic Web interaction.

The Linked Movie Database (Linked MDB) project provides a demonstration of the first open linked dataset connecting several major existing (and highly popular) movie web resources. The database exposed by Linked MDB contains millions of RDF triples with hundreds of thousands of RDF links to existing web data sources that are part of the growing Linking Open Data cloud, as well as to popular movie-related web pages such as IMDb. Linked MDB uses a novel way of creating and maintaining large quantities of high quality links by employing state-of-the-art approximate join techniques for finding links, and providing additional RDF metadata about the quality of the links and the techniques used for deriving them.

11.2 Introduction

Trust in social networks on the Semantic Web is a topic that has gained increased interest in the last few years. Using FOAF as the basis for the social network, trust has been encoded using the FOAF Trust Module¹ or computed from other information. With these trust values as a starting place, several algorithms for inferring trust relationships have been introduced. This analysis of Semantic Web-based social networks has produced results, but their usefulness in the space of user interaction has not been fully addressed.

In this paper, we present Film Trust, a website that integrates Semantic Web-based social networking into a movie recommender system. We begin with a description of the Film Trust

website, followed by an analysis of its features. Tidal Trust, a trust network inference algorithm, is used as the basis for generating predictive ratings personalized for each user. The accuracy of the recommended ratings is shown to outperform both a simple average rating and the ratings produced by a common recommender system algorithm. Theoretically and through a small user study, some evidence is also established that supports a user benefit from ordering reviews based on the users' trust preferences.

11.2.1 Background and Related Work

Social Network data, represented using the FOAF Vocabulary[1], is some of the most prevalent data on the Semantic Web. Tidal Trust[2] is an algorithm for inferring trust relationships. Using a recursive search with weighted averages, it can take two people in the network and generate a recommendation about how much one person should trust the other, based on the paths that connect them in the network, and the trust ratings on those paths. Part of our recommender system relies on inferred trust ratings, and this is the algorithm that is used there.

Recommender systems help users identify items of interest. These recommendations are generally made in two ways: by calculating the similarity between items and recommending items related to those in which the user has expressed interest, or by calculating the similarity between users in the system and recommending items that are liked by similar users. This latter method is also known as collaborative filtering.

Collaborative filtering has been applied in many contexts, and Film Trust is not the first to attempt to make predictive recommendations about movies. Movie Lens [5], Recommends [6], and Film- Conseil [7] are just a few of the websites that implement recommender systems in the context of films. Herlocker, et al. [8] present an excellent overview of the goals, datasets, and algorithms of collaborative filtering systems. However, Film Trust is unlike the approach taken in many collaborative filtering recommender systems in that its goal is not to present a list of good items to users; rather, the recommendations are generated to suggest how much a given user may be

interested in an item that the user already found. For this to work, there must be a measure of how closely the item is related to the user's preferences.

11.2.2 The Film Trust Website

The social networking component of the website requires users to provide a trust rating for each person they add as a friend. When creating a trust rating on the site, users are advised to rate how much they trust their friend about movies. In the help section, when they ask for more help, they are advised to, "Think of this as if the person were to have rented a movie to watch, how likely it is that you would want to see that film."

Part of the user's profile is a "Friends" page. In the Film Trust network, relationships can be one-way, so users can see who they have listed as friends, and vice versa. If trust ratings are visible to everyone, users can be discouraged from giving accurate ratings for fear of offending or upsetting people by giving them low ratings. Because honest trust ratings are important to the function of the system, these values are kept private and shown only to the user who assigned them.

The other features of the website are movie ratings and reviews. Users can choose any film and rate it on a scale of a half star to four stars. Because the inferred trust values reflect how much the user should trust the opinions of the person rating the movie, the weighted average of movie ratings should reflect the user's opinion. If the user has an opinion that is different from the average, the rating calculated from trusted friends – who should have similar opinions – should reflect that difference. Similarly, if a movie has multiple reviews, they are sorted according to the inferred trust rating of the author. This presents the reviews authored by the most trusted people first to assist the user in finding information.

11.3 Site Personalization

11.3.1 Computing Recommended Movie Ratings

One of the features of the Film Trust site that uses the social network is the "Recommended Rating" feature. As figure 7.3 shows, users will see this in addition to the average rating given to a particular movie.

The "Recommended Rating" is personalized using the trust values for the people who have rated the film (the raters). The process for calculating this rating is very similar to the process for calculating trust ratings presented in chapter 6. First, the system searches for raters that the source knows directly. If there are no direct connections from the user to any raters, the system moves one step out to find connections from the user to raters of path length 2. This process repeats until a path is found. The opinion of all raters at that depth are considered. Then, using Tidal Trust, the trust value is calculated for each rater at the given depth. Once every rater has been given an inferred trust value, only the ones with the highest ratings will be selected; this is done by simply finding the maximum trust value calculated for each of the raters at the selected depth, and choosing all of the raters for which that maximum value was calculated. Finally, once the raters have been selected, their ratings for the movie (in number of stars) are averaged. For the set of selected nodes S , the recommended rating r from node s to movie m is the average of the movie ratings from nodes in S weighted by the trust value t from s to each node:

11.3.2 Determining the Accuracy of Recommended Ratings

For each movie the user has rated, the recommended rating can be compared to the actual rating that the user assigned. In this analysis, we also compare the user's rating with the average rating for the movie, and with a recommended rating generated by an automatic collaborative filtering (ACF) algorithm. There are many ACF algorithms, and one that has been well tested, and which is used here, is the classic user-to-user nearest neighbor prediction algorithm based on Pearson Correlation [5]. If the trust-based method of calculating ratings is best, the difference between the personalized rating and the user's actual rating should be significantly smaller than the difference between the actual rating and the average rating.

On first analysis, it did not appear that the personalized ratings from the social network offered any benefit over the average. The difference between the actual rating and the recommended rating (call this ∂r) was not statistically different than the difference between the user's actual rating and the average rating (call this ∂a). The difference between a user's actual rating of a film and the ACF calculated rating (∂cf) also was not better than ∂a in the general case. A close look at the data suggested why. Most of the time, the majority of users' actual ratings are close to the average. This is most likely due to the fact that the users in the Film Trust system had all rated the AFI Top 50 movies, which received disproportionately high ratings. A random sampling of movies showed that about 50% of all ratings were within the range of the mean \pm a half star (the smallest possible increment). For users who gave these near-mean ratings, a personalized rating could not offer much benefit over the average.

However, the point of the recommended rating is more to provide useful information to people who disagree with the average. In those cases, the personalized rating should give the user a better recommendation, because we expect the people they trust will have tastes similar to their own [10].

To see this effect, ∂a , ∂cf , and ∂r were calculated with various minimum thresholds on the ∂a value. If the recommended ratings do not offer a benefit over the average rating, the ∂r values

will increase at the same rate the ∂a values do. The experiment was conducted by limiting ∂a in increments of 0.5.

Figure 11.3.2 illustrates one of the examples where the recommended value reflects the user's tastes. "A Clockwork Orange" is one of the films in the database that has a strong collective of users who hated the movie, even though the average rating was 3 stars and many users gave it a full 4-star rating. For the user shown, $\partial a = 2.5$ – a very high value – while the recommended rating exactly matches the user's low rating of 0.5 stars. These are precisely the type of cases that the recommended rating is designed to address.

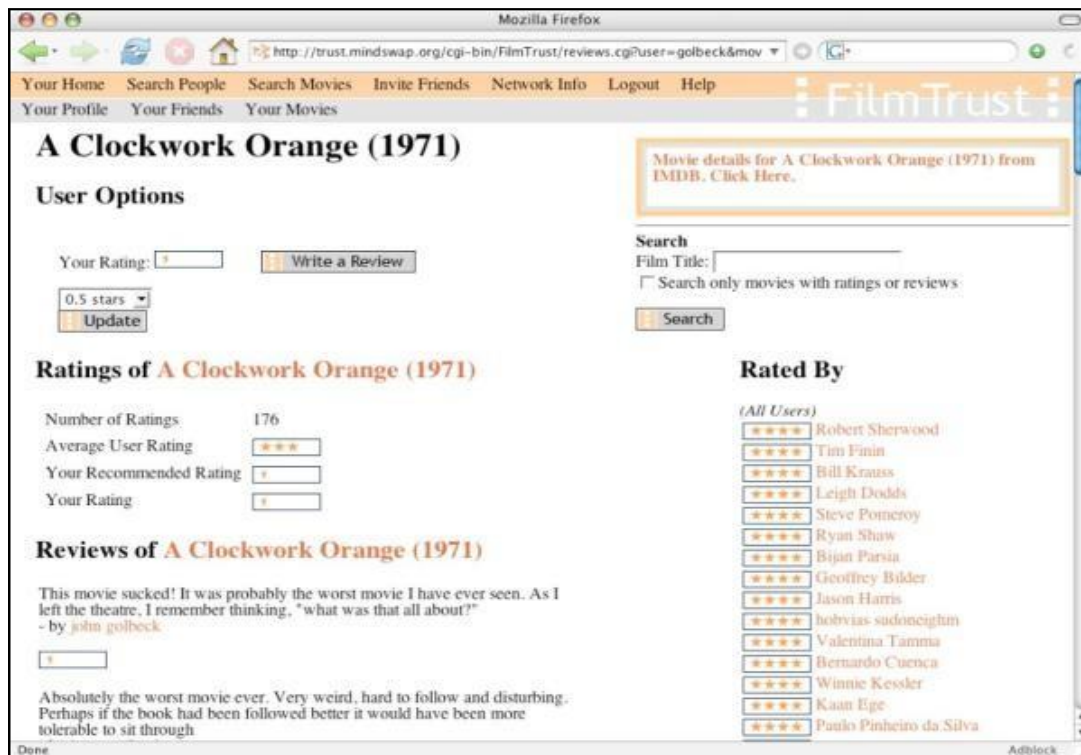


Fig 11.3.2 (Ratings)

A user's view of the page for "A Clockwork Orange," where the recommended rating matches the user's rating, even though ∂a is very high ($\partial a = 2.5$).

11.3.3 Presenting Ordered Reviews

In addition to presenting personalized ratings, the experience of reading reviews is also personalized. The reviews are presented in order of the trust value of the author, with the reviews from the most trustworthy people appearing at the top, and those from the least trustworthy at the bottom. The expectation is that the most relevant reviews will come from more trusted users, and thus they will be shown first.

Unlike the personalized ratings, measuring the accuracy of the review sort is not possible without requiring users to list the order in which they suggest the reviews appear. Without performing that sort of analysis, much of the evidence presented so far supports this ordering. The definition of trust has been used to support many of the calculations made throughout this dissertation. That definition also supports the ordering of reviews. Trust with respect to movies means that the user believes that the trusted person will give good and useful information about the movies. The analysis also suggests that more trusted individuals will give more accurate information. It was shown there that trust correlates with the accuracy of ratings. Reviews will be written in line with ratings (i.e. a user will not give a high rating to a movie and then write a poor review of it), and since ratings from highly trusted users are more accurate, it follows that reviews should also be more accurate.

A small user study with 9 subjects was run on the Film Trust network. Preliminary results show a strong user preference for reviews ordered by the trustworthiness of the rater, but this study must be extended and refined in the future to validate these results.

11.4 Conclusions and Discussion

Within the Film Trust website, trust in social networks has been used to personalize the user experience. Trust took on the role of a recommender system forming the core of an algorithm to create predictive rating recommendations for movies. The accuracy of the trust-based predicted ratings in this system is significantly better than the accuracy of a simple average of the ratings

assigned to a movie and also the recommended ratings from a Person- correlation based recommender system.

Overall, we believe that Film Trust is an example of how the Semantic Web, and Semantic trust networks in particular, can be exploited to refine the user experience. By using the Semantic Web data in computations, interaction with the Semantic Web becomes integrated into common tasks, and enhances existing tools.

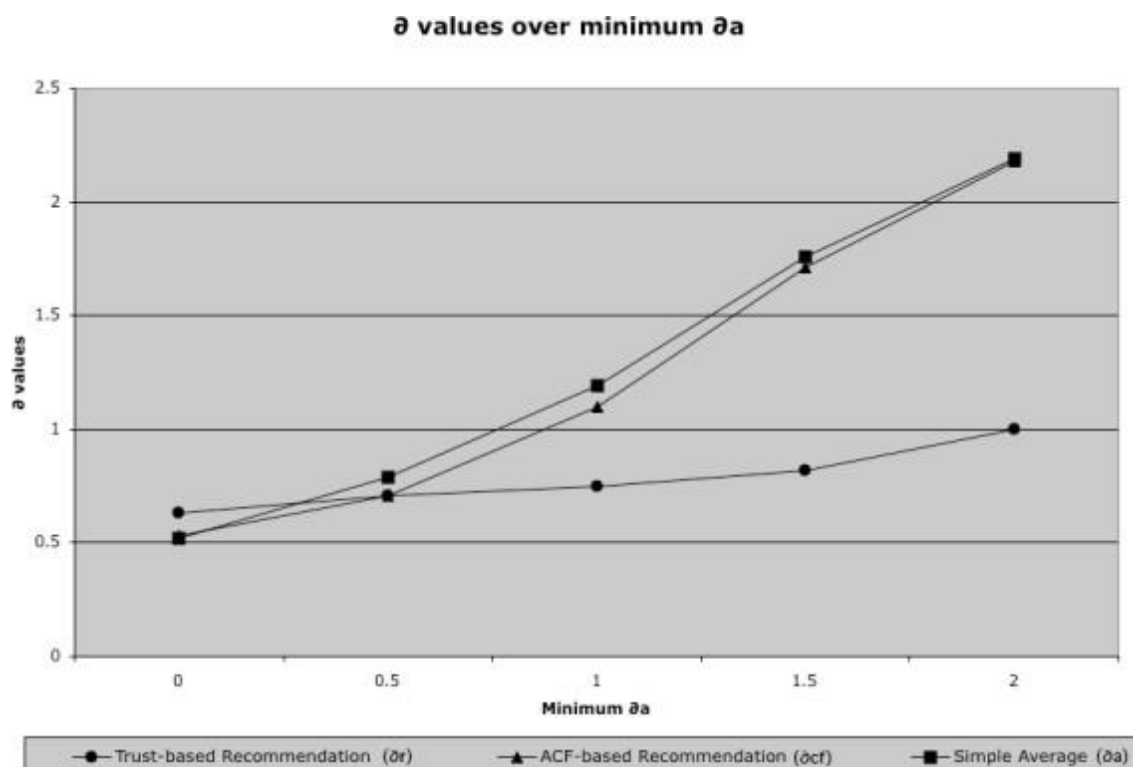


Fig 11.4 (Rating Graph)

11.5 Acknowledgements

This work, conducted at the Maryland Information and Network Dynamics Laboratory Semantic Web Agents Project, was funded by Fujitsu Laboratories of America -- College Park, Lockheed Martin Advanced Technology Laboratory, NTT Corp., Kevric Corp., SAIC, the National Science Foundation, the National Geospatial-Intelligence Agency, DARPA, US Army Research Laboratory, NIST, and other DoD sources.

11.6 References

- a. FOAF Vocabulary: <http://xmlns.com/foaf/0.1/>
- b. Golbeck, Jennifer (2005) "Computing and Applying Trust in Web-Based Social Networks. D Dissertation, University of Maryland, College Park.
- c. Sinha, R., and Swearingen, K. (2001) "Comparing recommendations made by online systems and friends." In Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries Dublin, Ireland.
- d. Swearingen, K. and R. Sinha. (2001) "Beyond algorithms: An HCI perspective on recommender systems," Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems, New Orleans, Louisiana.
- e. Herlocker, Jonathan L., Joseph A. Konstan, John Riedl, Explaining collaborative filtering recommendations, Proceedings of the 2000 ACM conference on Computer supported cooperative work, p.241-250, December 2000, Philadelphia, Pennsylvania, United States.
- f. Garden, Matthew, and Gregory Dudek (2005) Semantic feedback for hybrid recommendations in Recommends. Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE05), Hong Kong, China, March 2005.

