

Use a defect tracking tool to manage software defects in your project (5%)

Major Defect Fixes

Date reported	4-19-2022
Who reported it	Aditi
Brief description of defect	Initially the remove function only took tab name as a parameter but then I realized the unique records that were added could not be deleted with just tab name.
Date fixed	4-20-2022
Who fixed it	Aditi
Brief description of how it was fixed	Then in the remove function movie name and tab name was passed as the two parameters.

Date reported	4-18-2022
Who reported it	Aditi
Brief description of defect	After adding rows to the table of different tabs, the heading row became missing.
Date fixed	4-18-2022
Who fixed it	Aditi
Brief description of how it was fixed	In the function that displays populated table header row were added along with the movie rows.

Date reported	4-19-2022
Who reported it	Aditi
Brief description of defect	XML is organised in tree structure form which can be accessed by using dom object but we have to iterate through all nodes till we find the required node.
Date fixed	4-19-2022
Who fixed it	Aditi

Brief description of how it was fixed	This was fixed by using X path. X path was constructed from root node till the collection of movies for given tab. Therefore due to this the search became fast and narrow.
---------------------------------------	---

Date reported	4-21-2022
Who reported it	Aditi
Brief description of defect	syntax error with a function was named getFileContaint when it was referenced as getFileContent
Date fixed	4-21-2022
Who fixed it	Aditi
Brief description of how it was fixed	Changed the function initialization name

Date reported	4-21-22
Who reported it	Aditi
Brief description of defect	missing initialization of the openTab function causing syntax error invalid function
Date fixed	4-21-22
Who fixed it	Aditi
Brief description of how it was fixed	fixed the function syntax initializing the openTab function

Date reported	4-22-22
Who reported it	Aditi
Brief description of defect	For loop not looping, syntax error
Date fixed	4-22-22
Who fixed it	Aditi
Brief description of how it was fixed	the final part of the for parameters was i instead of i++, once changed it was fixed

Date reported	4-22-22
Who reported it	Huyen
Brief description of defect	Cannot import firestore
Date fixed	4-22-22
Who fixed it	Huyen
Brief description of how it was fixed	Install the firebase package

Date reported	4-23-22
Who reported it	Huyen
Brief description of defect	Api has limited number of fetching
Date fixed	4-22-22
Who fixed it	Huyen
Brief description of how it was fixed	Use another api

Date reported	4-24-22
Who reported it	Huyen
Brief description of defect	Cannot call function from script with type="module"
Date fixed	4-24-22
Who fixed it	Huyen
Brief description of how it was fixed	Use addEventListener instead

Describe how your team integrated code from the team members. Which Integration

Strategy do you think your team used? Explain why. (250-350 words – 5%) - DONE

During our final project, our team was able to integrate code from each other by using GitHub's code merging features, and by using simple Emails. We used email, because it was a simple and easy method to go with to send code files to one another. The integration strategy that we used for this project was the All-At-Once Integration. Using this method, we combined each member's code together by sharing files via email. After everyone received and downloaded each other's integration parts, we worked to make the needed changes to the code via cross collaboration by using zoom, using groupme, and by comparing commits by using Github's code integration feature. There were a few instances where some parts of the code failed to work with each other, so minor tweaks were needed. For example: Some functions and classes didn't merge properly because of differences between written code of team members. In retrospect, this was a failure of the type of integration method that we chose to use. The all-at-once integration strategy that we used ended up causing us more errors that we had to fix, which caused us extra time delays in finishing up our project. After we fixed all of the errors, we once again used GitHub's code merge to make sure everything merged together properly, and avoid any extra complications. In the future, and especially with any larger projects, we recognize that it would be best to avoid using the all-at-once integration again, since it did cause a few errors for us, which would only multiply if our project increased in magnitude.

Deployment plan (500-700 words - 5%) - DONE

In order to deploy our project to users, we will need to run extensive testing on our project to make sure it is fully ready to launch. Additionally we will need to purchase the domain that our site will run off of, as well as servers to store user data and movie data. We would also need to purchase rights to use the iMDB API for commercial use, since the current model we are using is only licensed for personal and non-commercial use.

For testing, this would begin with beta testing done by external users, after we complete alpha testing, which would be done internally by us. This would take us a few weeks to complete, and could potentially require monetary incentives to testers if we cannot find sufficient testers. But, assuming we have sufficient testers, beta testing would cost us only our time and effort put into fixing any potential bugs found by users.

For the domain and server space, we would start by buying a new domain name similar to our application name in order to be easily recognizable to our market. The costs of buying a domain name are about \$20.00 a year, with some places offering domain hosting for just a few dollars a month. So a good budget for domain name and hosting would be around \$100. To start out, we could use mySQL

servers for free, but if we need more storage space, speed, or security, we could upgrade to a paid server to store user data.

An additional cost we could face would be if we would want to release our product as a mobile app, in addition to releasing it as a website. This would take time to do, since we would have to turn our current product, developed with javascript, into a mobile application, something that a few of us do not have experience with thus far. Additionally, we would face costs of roughly \$125 for the first year we would have our mobile app launched on both Google Play and the Apple app store, and \$100 a year after that to keep it on the Apple app store. However, we do think that this development would be beneficial, since we could allow users to sync their accounts from a mobile account to a web account so they can access their movie lists wherever they happen to be.

For the actual deployment and release of our product, we would like to spend money and put effort into advertising. The first step in doing this is to figure out our market. We believe that our product is geared to anyone who enjoys watching movies so that they can see statistics, keep track of what they watch, and find new movies that they might enjoy watching. Additionally, we could market to critics/film reviewers so that they can plan what movies to watch next, and review the ones they have already watched.

To advertise to these individuals, we believe that advertisements in movie theaters and streaming sites (like netflix, hulu, hbo, etc.) would be a great way to gain users, since people who enjoy movies are often found in theaters and on streaming sites. For theaters, we could advertise both with still ads, such as posters, in person, such as set up a booth where we could talk to potential customers, or, once we gain a larger user base and subsequently, a higher budget, we could run short ads on the big screen before the movies begin.

o Maintenance plan (500-700 words - 5%) - DONE

In order to maintain our product, there are a few things we would need to consider. Firstly, we would need to figure out the costs of maintaining our domain name, server, and potential mobile application. Secondly, as our application grows in popularity, we will need to keep updating it, and would potentially need to hire new developers to help maintain it and add new features. Additionally, we would need to spend time and put effort into regression testing as we add new features, and as we test new features.

The clear cut costs of maintaining our product include keeping our domain name for our website, maintaining our web hosting and servers, and potentially keeping our app up on the app stores. Keeping our domain name and hosting wouldn't be too costly, since purchasing the domain name is a one time fee. The hosting would keep up a monthly cost that would vary depending on the popularity of

our application. At first, the monthly cost would be between \$5-15 depending on how we plan on hosting it. However, once our popularity grows, we would likely need a dedicated hosting plan with a server, which would likely cost up to \$150 a month. As for keeping our mobile app up on app stores, the google play store is a one-time fee, so maintaining our status there is easy. However, Apple's app store requires an annual fee of \$99 to keep your app up on their store.

A more subjective cost comes from the potential to hire new developers for our product. An entry level software developer would cost around \$80,000 a year, subjective to negotiation. Experienced software developers would be significantly more expensive. Considering we are not currently being paid, another cost would be giving the five of us software developer salaries as well.

In order to maintain our product, we would need to improve our project to keep our user base interested which could include adding new features, a way to add friends and see their movie lists, a way to review movies, and even more. It would cost both time and energy to both add the features to our product, and to test and launch them. Adding a new feature would require us to first make the new feature and test how it works alone, then we would need to integrate it to the existing application, which could take some time depending on the complexity of the feature we plan to add. Once we have the feature integrated, we would need to go about testing the feature again, as well as testing all of the existing features that the new one interacts with, just to make sure no new bugs were introduced during integration. After we complete this, we could launch the new feature to be tested by users, and incorporate any feedback we may receive. While this would not likely have a great monetary cost to do the launch, the development, integration, and testing would all require software developers, which would be included in the aforementioned software developer salary costs.

o Code Review (500-700 words – 5%)

Undertake, as part of a team activity, an inspection of a medium-size code segment using either a Walkthrough or Inspection as described in the lecture on Software Verification

Following Medium SizeCode Block is being reviewed for errors and faults.

```
function populateMovieListFromXML(xmlstring, tabName) {
    parser = new DOMParser();
    var xmlDoc = parser.parseFromString(xmlstring,"text/xml");
    //var xmlDoc = xml.responseXML;
    var txt =
    "<tr><td><b>Title</b></td><td><b>Ratings</b></td><td><b>Year</b></td><td><b>Genre</b></td><t
    d><b>Language</b></td>\
    <td><b>Remove</b></td></tr>";
```

```

path = "/User/MovieRecommendation/" + tabName + "/Movie";
console.log(path);
if (xmlDoc.evaluate) {
    var nodes = xmlDoc(path, xmlDoc, null, XPathResult.ANY_TYPE, null);
    var result = nodes.iterateNext();
    var elem;
    var genre;
    var title;
    var ratings;
    var year;
    var lang;
    while (result) {
        genre = result.getAttribute('genre');
        title = result.getElementsByTagName("title")[0].childNodes[0].nodeValue;
        ratings = result.getElementsByTagName("ratings")[0].childNodes[0].nodeValue;
        year = result.getElementsByTagName("year")[0].childNodes[0].nodeValue;
        lang = result.getElementsByTagName("title")[0].getAttribute('lang');

        //console.log("result innerHTML" + result.innerHTML);
        //console.log('genre=' + genre);
        //console.log("title=" + title);
        //console.log("ratings=" + ratings);
        //console.log("year=" + year);
        //console.log("lang=" + lang);

        txt += "<tr><td>" + title + "</td><td>" + ratings + "</td><td>" + year +
            "</td><td>" + genre + "</td><td>" + "</td><td><button
onclick='removeEntryfromXML(\"" + tabName + "\",\"" + title +
            "\"')>Remove</button></td></tr>";

        result = nodes.iterateNext();
    }
    //console.log(txt);
    //WatchedMoviesTable
    var completeTable = document(tabName + "Table");

    completeTable.innerHTML = txt;
}
}

function saveXmlData()
{
    console.log("saveXmlData");
    if(xmlDataDoc!="")
    {
        saveToFile("MovieRecommendation.xml",xmlDataDoc)
    }
}

function loadXmlData()
{
    console.log("loadXmlDat()");
    if(xmlDataDoc=="")
    {
        getFileContaint("MovieRecommendation.xml")
        //while(xmlDataDoc=="") sleep(250);
    }
}

```

```

}
}

// open the tab when user click on it
function(evt, tabName) {
  console.log("openTab:"+tabName);
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablinks");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace("active", "");
  }
  document.getElementById(tabName).style.display = "block";
  evt.currentTarget.className += "active";
  //-----
  //action only when a tab (except for home) is clicked
  if (tabName != "Home")
  {
    if(xlmDataDoc=="")
    {
      getFileContaint("MovieRecommendation.xml")
      //while(xlmDataDoc=="") sleep(250);
    }
    //sleep(250);
    populateMovieListFromXML(xlmDataDoc, tabName);
  }
}

```

Document the following:

- List of faults detected
- Description of each fault
- Who was assigned to fix each fault
- How the fault was fixed

```
function populateMovieListFromXML(xmlstring, tabName)
```

Assigned: Aditi

Function already existed, to fix this we had to go back in the code and fix the function that interfered with having a duplicate name. After that, had to fix errors regarding the other function name change throughout the application.

```
parser = new DOMParser();
```


Assigned: Aditi

DOMParser was showing as invalid. Solution was to find out where the code was breaking, turns out code was break above this line segment which in return caused DOMParser to not work.

```
var txt =
"<tr><td><b>Title</b></td><td><b>Ratings</b></td><td><b>Year</b></td><td><b>Genre</b></td><td><b>Language</b></td>\
<td><b>Remove</b></td></tr>";
```

Assigned: Aditi

Invalid syntax in this code. It was missing ` </td>`, causing the code to break. Solution was to add the missing ` </td>` to fix functionality.

```
var nodes = xmlDoc(path, xmlDoc, null, XPathResult, null);
```

Assigned: Aditi

Type was missing in the following code segment. The fix was by adding `ANY_TYPE` After "null". This fixes the code break that happened at this particular segment.

```
while (result)
```

Assigned: Aditi

Function already existed, to fix this we had to go back in the code and fix the function that interfered with having a duplicate name.

```
title = result.getElementsByTagName("title")[0].nodeValue;
```

Assigned: Aditi

Child Node value missing in the `getElementsByTagName`. The fix is adding the value after title `.childNodes[0]` without this the code will continue breaking with invalid syntax error.

```
txt += "<tr><td>" + title + "</td><td>" + ratings + "</td><td>" + year +
      "</td><td>" + genre + "</td><td></td><td><button
onclick='removeEntryfromXML(\"" + tabName + "\",\"" + title +
      "\"')>Remove</button></td></tr>";
```

Assigned: Aditi

Invalid syntax in this code. It was missing the next category `</td><td></td><td><butto` It was also missing quotations causing the program to interrupt because no string value was added.

```
var completeTable = document(tabName + "Table");
```

Assigned: Aditi

Following is missing .getelementby after `document` causing syntax error and library error since it wasn't didn't know what to do with the document. After adding .getelementby it is able to grasp tabName + "Table"

```
function saveXmlData()  
{  
  console.log("saveXmlData");  
  if(xmlDataDoc!="")  
  {  
    saveToFile("MovieRecommendation.xml",xmlDataDoc)  
  }  
}
```

Assigned: Aditi

Invalid syntax in the above code segment, the solution is adding a closing bracket at the end of the segment.

```
getFileContaint("MovieRecommendation.xml")
```

Assigned: Aditi

Invalid function, instead of containt, is supposed to say content. This will cause syntax error and will not let the program go further.

```
function (evt, tabName) {  
  console.log("openTab:"+tabName);
```

Assigned: Aditi

Function missing on the top openTab function is suppose to be implemented. This will cause syntax error for invalid function. To fix this simply add "openTab" function at the top.

```
tablinks = document.getElementsByClassName("tablinks");  
for (i = 0; i < tablinks.length; i)
```

Assigned: Aditi

Invalid string causing break in the code. i is missing ++ in the length causing syntax error for the value. It causing tablinks length to have no value. It is also missing bracket at the end. Simple adding opening bracket for next segment will fix the issue.