**Index of Contents**

**Chapter 1 : Introduction**

**1.1 Background**

**1.2 Problem definition**

**1.3 Scope**

**1.4 Need to define the problem**

**Chapter 2 :**   **Literature Survey**

**2.1 Proposed System**

**2.2 Future Scope**

**Chapter 3: Project Design**

**3.1 System Design**

**3.2 System Architecture**

**3.3 Module wise flow diagram**

**3.4 Software development tools**

**Chapter 4 : Implementation and Testing of project work/ topic**

**4.1 Code**

**4.2 Testing Phases**

**4.3 Responsibilities**

**4.4 Github upload**

**Chapter 5 : Conclusions and further work**

**Chapter 1: Introduction**

**Introduction**:

A Movie Recommendation system is a system that provides movie suggestions to users

based on some dataset. Such a system will predict what movies a user will like based on

the attributes of previously liked movies by that user. Content-Based recommendations

have long been in fashion but they tend to overlook some great suggestions that may not

be covered by mere similarities. To overcome such shortcomings, we will combine

collaborative techniques having a with neural networks to provide users(who have

already rated movies previously) with appropriate suggestions.

In the most general way, recommender systems are algorithms aimed at suggesting

relevant items to users(items being books to read, products to buy, music to listen or as

in our case, movies to watch.)

The most common approach towards recommendation systems has been the Content based recommendations. Content-based techniques develop representations of clients

and items through the investigation of additional data, for example, record content, client

proles and the traits of items, to make suggestions

In short our project will be simple The system will provide recommendations based on

the ratings on commodities that users have previously rated.

**Problem Definition:**

The amount of information generated by the internet is growing at an overwhelming

rate. The fact that more products and services are available to consumers now than ever

has not only compounded the problem but has also increased its scope and diversity.

Businesses are forced to come up with smarter solutions to prevent consumers from

being engulfed by seemingly countless choices and more importantly to be still relevant

to them by directing their attention to commodities they might be interested in.

**Scope of the project:**

• Recommendation systems are introduced to e-commerce to help businesses ,

tackle the challenge.

• Movie recommendation system will provide relevant options to the consumers

instantly.

**Describe the need of this stage in project.**

In today's world where internet has become an important part of human life, the users are facing problems of choosing due to the wide variety of collection. Searching from a motel to good investment options, there is too much information available over the internet. To help the users cope with this information explosion, companies have deployed recommendation systems for guiding their users.

The main purpose of our system is to recommend movies to its users based on their viewing history and ratings that they provide. The system will also recommend their products to specific customers based on the genre of movies they prefer. Collaborative filtering and content based filtering are the prime approaches in providing recommendation to the users. Both of them are best applicable in specific scenarios because of their respective properties.

* 1. **Background**



**Why hybrid Recommendation System?**

* Recommender system has the ability to predict whether a particular user would prefer an item or not .
* Recommender systems are beneficial to both service providers and users .
* They reduce Time of finding and selecting items in an online environment.
* The number of users typically increases when a recommendation system is used.

**Chapter 2 Literature survey**

MOVREC [1] is a movie recommendation system presented by D.K. Yadav et al. based on collaborative filtering approach. Collaborative filtering makes use of information provided by user. That information is analyzed and a movie is recommended to the users which are arranged with the movie with highest rating first. The system also has a provision for user to select attributes on which he wants the movie to be recommended.

Luis M Capos et al. [2] has analyzed two traditional recommender systems i.e. content based filtering and collaborative filtering. As both of them have their own drawbacks he proposed a new system which is a combination of Bayesian network and collaborative filtering. The proposed system is optimized for the given problem and provides probability distributions to make useful inferences.

**2.1 Proposed System:**

Recommendation algorithms mainly follow collaborative filtering, content-based filtering, demographics-based filtering and hybrid approaches.

**Collaborative filtering:-**It recommends items based on the similarity measures between users and items. The system recommends those items that are preferred by similar category of users. Collaborative filtering has many advantages

It is content-independent

In CF people makes explicit ratings so real quality assessment of items is done.

It provides effective recommendations because it is based on user’s similarity rather than item’s similarity.

**Content based filtering:-**It is based on profile of the user’s preference and the item’s description. In CBF, to describe items we use keywords apart from user’s profile to indicate users preferred likes or dislikes. In other words CBF algorithm recommend items or similar to those items that were liked in past. It examines previously rated items and recommends best matching item.

**Hybrid recommender:** Hybrid recommender system is the one that combines multiple recommendation techniques together to produce the output. If one compares hybrid recommender systems with collaborative or content-based systems, the recommendation accuracy is usually higher in hybrid systems. The reason is the lack of information about the domain dependencies in collaborative filtering, and about the people’s preferences in content-based system. The combination of both leads to common knowledge increase, which contributes to better recommendations. The knowledge increase makes it especially promising to explore new ways to extend underlying collaborative filtering algorithms with content data and content-based algorithms with the user behavior data.

**Chapter 3: Project Design**

1. **System Design**

The front-end website or app service collects historical data of user-movie

interactions, which are represented in a table of user, item, and numerical

rating tuples.

The collected historical data is stored in storage.

**3.1 System Architecture**

Architecture is a set of structuring principles that enables a system to be comprised of a set of simpler

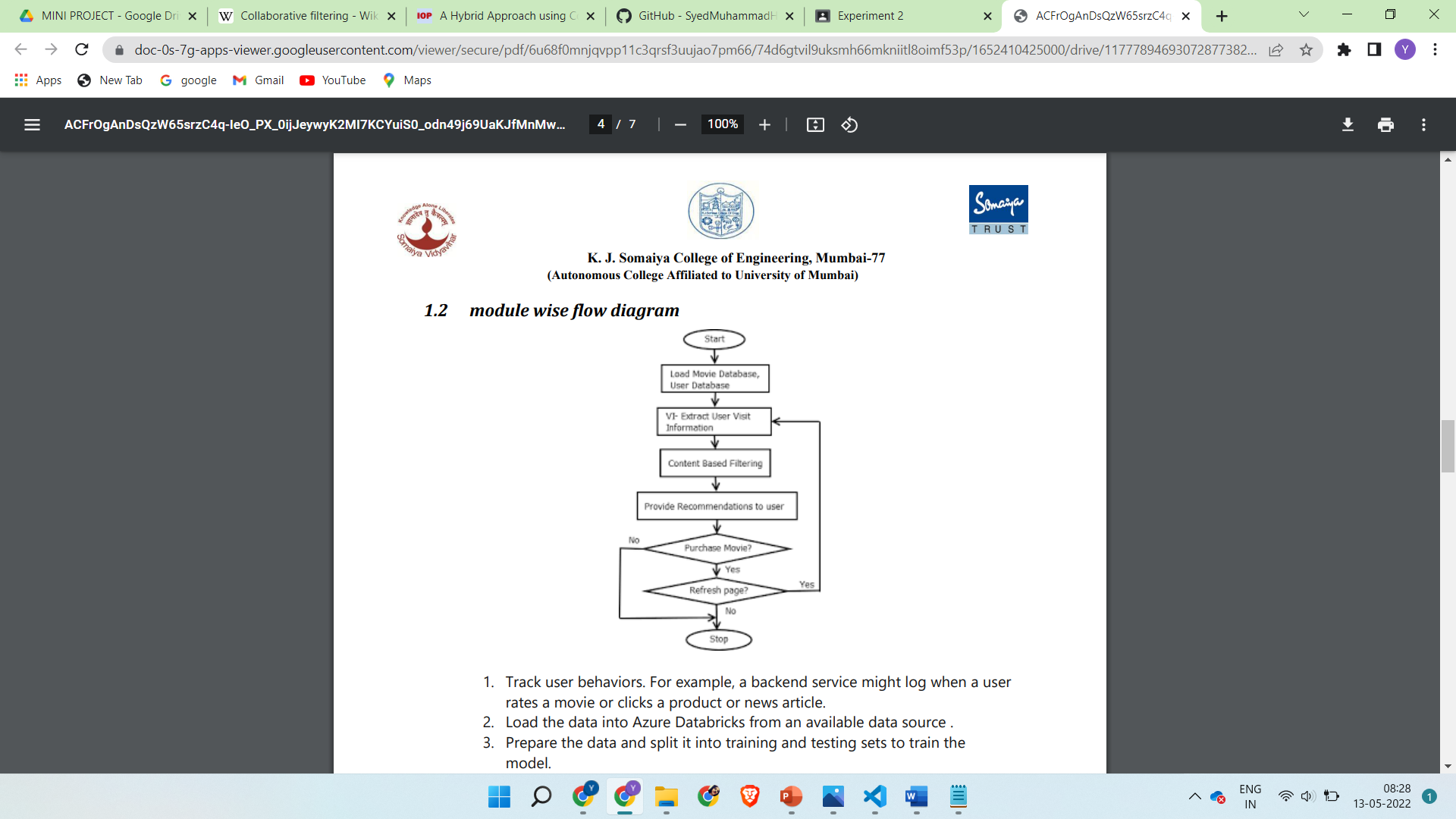
systems each with its own local context that it is independent of but not inconsistent with the context of

the larger system as a whole. Architecture is created to describe the structure of the system to be built

and how that structure supports the business and service-level requirements. In this section we are

going to discuss the components the system is made up of.

**3.2 module wise flow diagram**



1. Track user behaviors. For example, a backend service might log when a user

rates a movie or clicks a product or news article.

2. Load the data into Azure Databricks from an available data source .

3. Prepare the data and split it into training and testing sets to train the

model.

4. Evaluate the quality of the model using rating and ranking metrics.

5. When the backend service gets a request from a user, call the

recommendations API hosted in AKS to get the top 3 recommendations and

display them to the user.

**3.2 Future Scope**

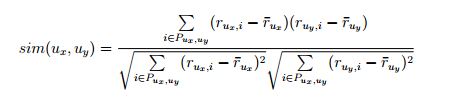
Recommendation systems are introduced to e-commerce to help businesses , tackle the challenge.

Movie recommendation system will provide relevant options to the consumers instantly.

Cosine similarity calculation do not work well when we don't have enough rating for movie or when user's rating for some movie is exceptionally either high or low.As an improvement on this project some other methods such as adjusted cosine similarity can be used to compute similarity.

Adjusted cosine similarity, which issimilar to cosine similarity, is measured by normalizing the user vectors Ux and Uy and computing the cosine of the angle between them. However, unlike cosine similarity, when computing the dot product of the two user vectors, adjusted cosine similarity uses the deviation between each of the user’s item ratings, denoted Ru, and their average item rating, denoted ¯Ru, in place of the user’s raw item rating.

In equation form, the adjusted cosine similarity computation is expressed as:



where Pux,uy represents the subset of items i ∈ I for which both users have rated, Rux,i is the rating of user Ux on item i and Ruy,i in the rating of user Uy on item i. The main advantage of this approach is that in item-based collaborative filtering, the item vectors consist of ratings from different users who often have varying rating scales.

**3.4 Software development tools**

a. an application development too: VS Code

b. HTML authoring tools: VS Code

c. a word processor for documentation: MS Word

d. a tool for drawing diagrams: Dia Software e. automated testing tools: Selenium

**Chapter 4 - Implementation and Testing of project**

**Implementation details:**

The Regression-based Movie Recommender system that's a hybrid of content-based and collaborative filtering Recommender system. Simply rate some movies and get immediate recommendations tailored for you.

**Project Structure**

project

│ README.md

│

└───Data

│ │ Data\_README.txt

│

└───Images

| | img1.png

| | img1.png

| | ...

|

└───Model

│ │ Exploratory data analysis with PostgreSQL.ipynb

│ │ Hybrid recommendation algorithm.ipynb

│ │ PostgreSQL\_Database\_wrapper.py

│ │ Movies\_Datase.pkl

│ │ Movies\_Learned\_Features.pkl

│ │

│ └───images

│ │ img1.png

│ │ img2.png

│ │ ...

│

└───Server

│ README.txt

└───templates

│ │ home.html

│ │ layout.html

│ │ result.html

└───app.py

└───Movies\_Datase.pkl

└───Movies\_Learned\_Features.pkl

└───requirements.txt

└───templates

**Dataset**

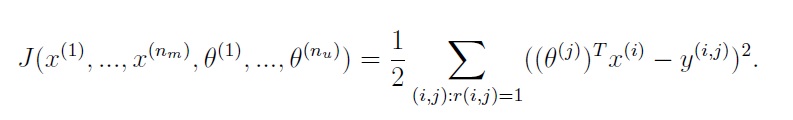
The dataset is provided by GroupLens and can be downloaded from here it contains the following files(links.csv, movies.csv, ratings.csv, and tags.csv)

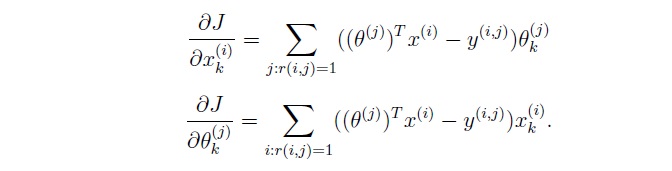
**About Hybrid Algorithm**

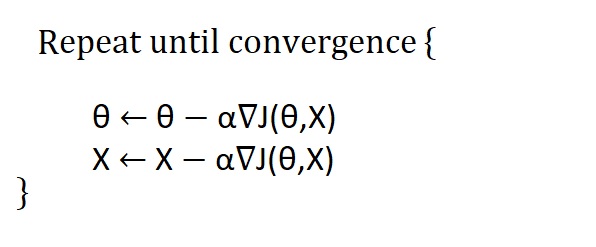
The hybrid recommendation system consists of

**Step 1. Collaborative filtering**

The main objective of collaborative filtering at this step is to learn features for different movies. The implementation of Collaborative filtering here performs "Feature learning" Using a variation of multivariate regression with gradient descent as an optimization algorithm, it takes as input user-item interaction matrix and simultaneously learns both the parameters for different users and features for different movies .

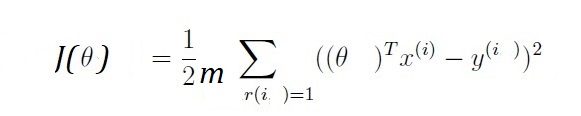


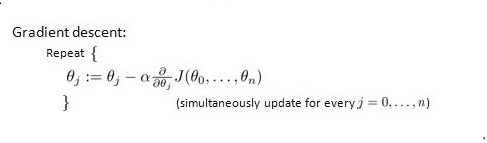




**Step 2. Content-based filtering**

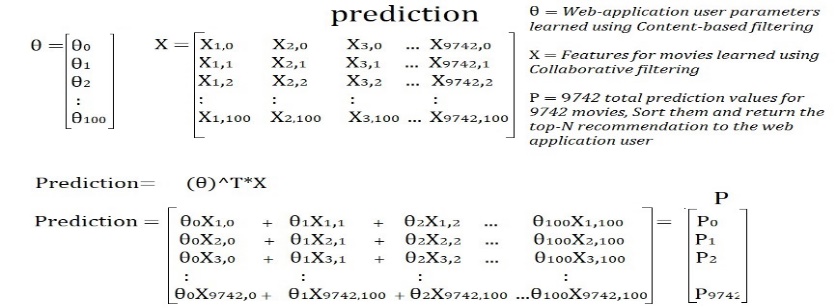
The content-based filtering here again is going to be an extension of multivariate regression but unlike collaborative filtering here I'm going to use the features for movies learned using collaborative filtering now to learn online web-application user parameter using content-based filtering thats unique to the user based on his/her web application movie ratings.





**Step 3. Prediction**

Prediction (uses both the Features for movies learned using collaborative filtering and the Parameters unique to user learned using content-based filtering to recommend top-N recommendation) The prediction uses both the vectors for movies learned using collaborative filtering and the parameter unique to user learned using content-based filtering to recommend top-N recommendation



**Testing of User interfaces:**

**Test plan: Write 5 to 6 test cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Description** | **Intended result** | **Actual result** | **Completed by** |
| Bug-less Navigation Menu | The navigation menu should direct you to the predict of the movie you want to watch and make, the options menu and the exit button to exit the website | Navigation System Runs Smoothly without any errors | Navigation System Runs Smoothly without any errors | **Yash Gavade** |
| Bug-less Start/Exit Menu | The pause/play menu should open up and freeze the website in the system while playing the prediction when the user chooses to and this should allow the user to exit the level or continue the process. | Start/Exit  menu Runs Smoothly without any errors | Start/Exit  menu Runs Smoothly without any errors | **Yash Gavade** |
| Bug-less Recommendation System | The website is implemented without bugs and its work well with the surrounding terrain. | The car runs smoothly without any errors in it control | The car runs smoothly without any errors in its control | **Rohan Dsouza** |
|  |  |  |  |  |

**Responsibilities:**

|  |  |  |
| --- | --- | --- |
| **Task** | **Member1**  **Rohan Dsouza** | **Member2**  **Yash Gavade** |
| **Program:** |  |  |
| **Design** |  | X |
| **Coding** | X |  |
| **Html , Css :** |  |  |
| **Level Design** |  | X |
| **Asset Implementation** |  | X |
| **Coding** |  |  |
| **Program:** |  |  |
| **Bug Tester** | X |  |
| **Engine System** | X | X |
| **Content Sytem** |  | X |
| **Hybrid Sytem** | X |  |
| **Movie Bot** |  | X |
| **Testing:** |  |  |
| **Testing approach** | X |  |
| **Test Cases** |  | X |
| **Presentation:** |  |  |
| **Powerpoint** |  | X |
| **Report** | X |  |

**Major problems: Write related to problems faced in the project**

* The recommender system adopts the classic B/S architecture mode, which can be divided into three parts: presentation layer, business logic layer, and data access layer and due to many bugs and errors it was not possible to link to each other so we re created the data set and flipped the movie and predict function.
* Flipping Data set: The implementation of the car controller was very buggy as the wheel collider of the car kept interfering with the body of the car and this caused the car to flip over in the game without any input. The problem was solved by importing an example movie from the controller package and replacing the body of the movie with one which we have created and rearranging the placement of the wheels in order to fit the structure of the data set.
* Completely Broken and not Working: The pervious 2 iterations of the respawn system were very movie. The first and second iteration focused on predicting the movie from one place in the system to another. The issue was that it only predict the body of the movie while the prediciting , movie and follow images of the movie stayed in place. To solve that another implementation of the same method was used but the same if not more broken result was obtained because this time, apart from the result of the first iteration. The latest implementation requires the car to be converted to a prefab (Like an Object in Java) and once that is done, through code the predict and the follow movie is destroyed when it enters a collision and then it is recreated in the re predict movies.
* Sytem package without documentation: Though the system asset package had a lot of good looking elements, the documentation which was created for it did not show us how to implement it and so it took a lot of time to figure out how to implement it and use it with the system of options in Unity.

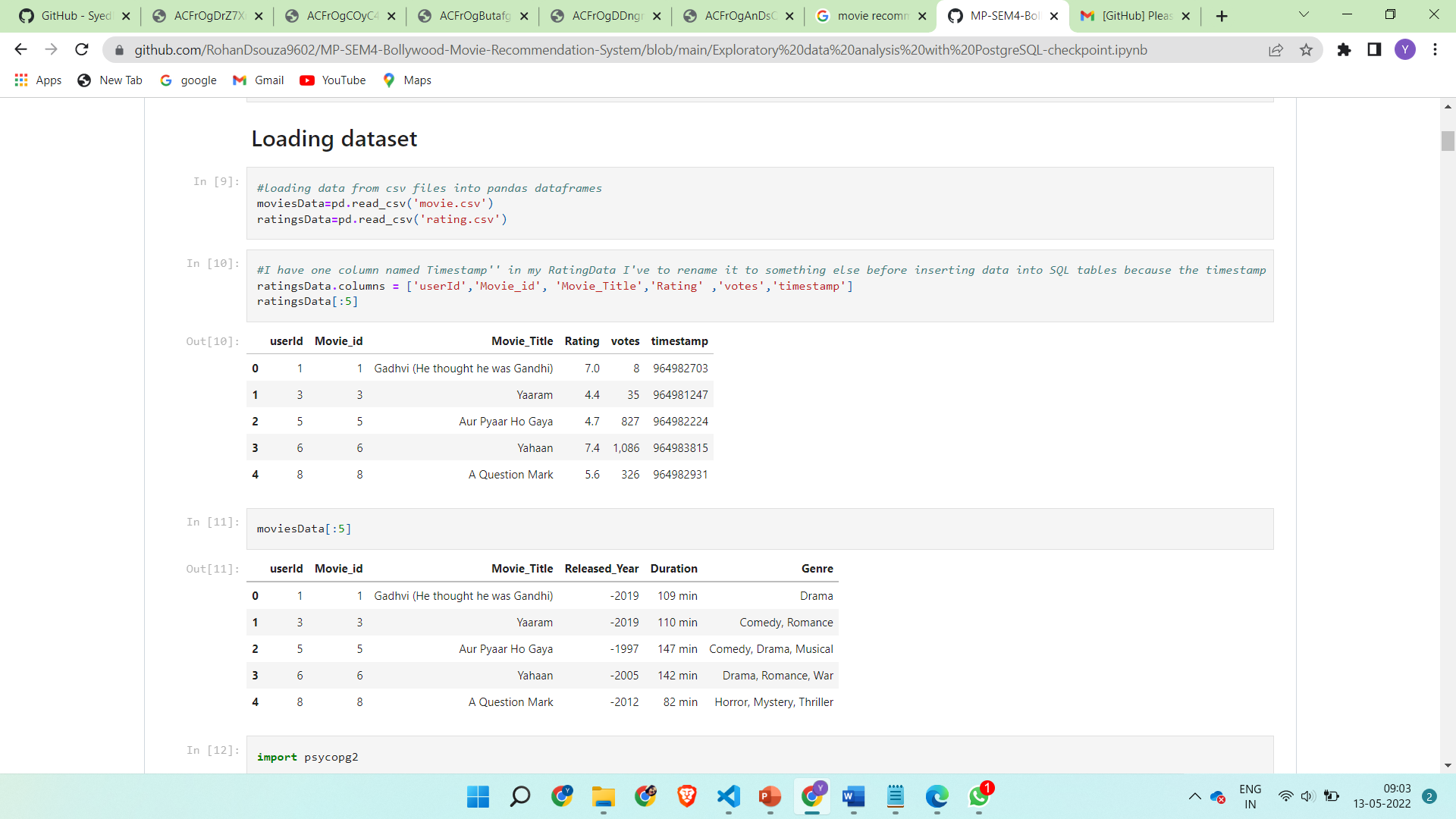
**Software used:** Vs code , Google Colab , VS Code, Blender, Html , Css , Microsoft Word, Google Slides.

**Tools: (Used for testing if any)**

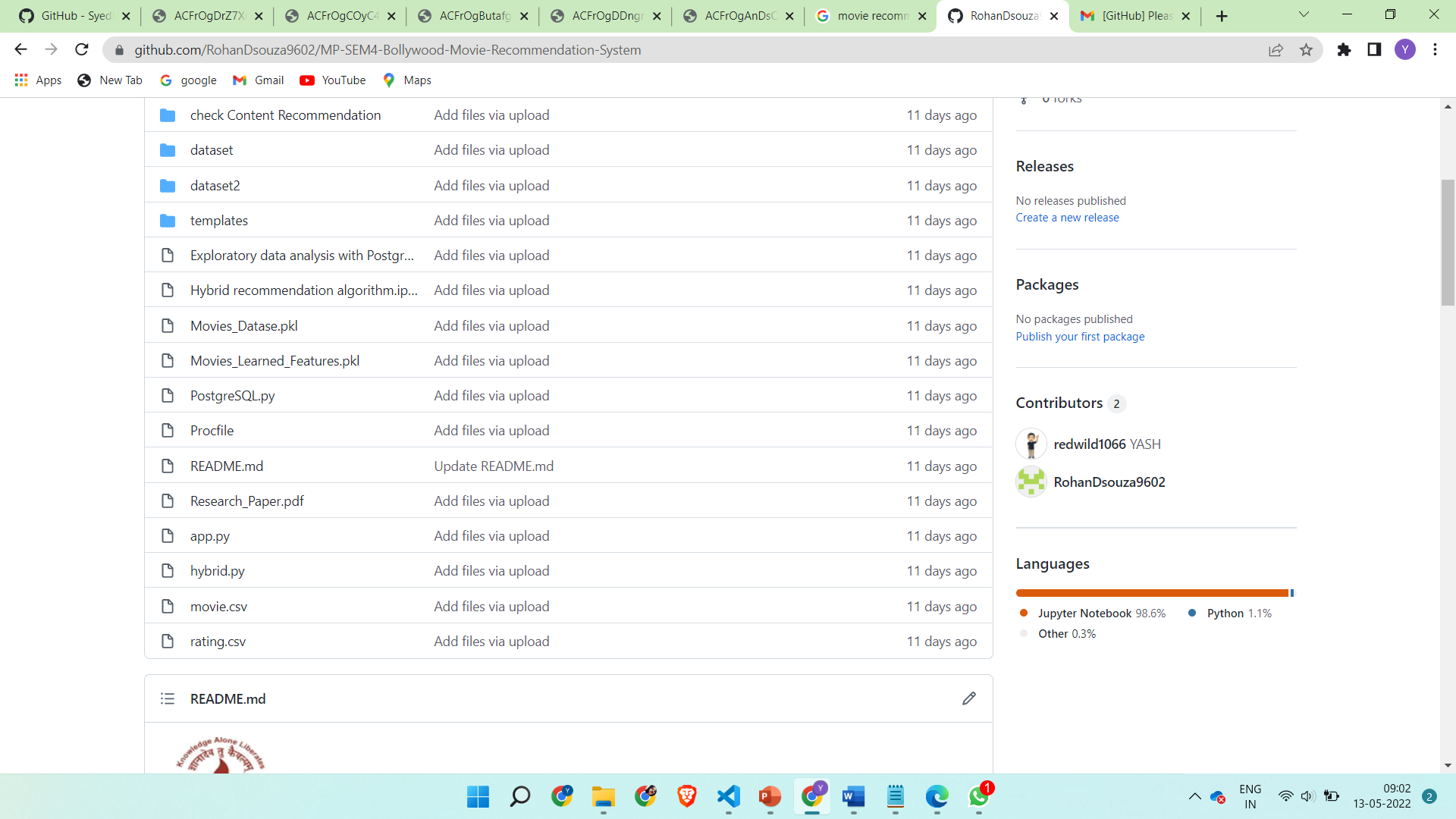
|  |  |
| --- | --- |
| **Tool** | **Description** |
| Vs code | Vs code was used in order to run the engine and obtain some inference to use as a test case to solve the issues in the System |
| Youtube | Youtube was used in order to gain knowledge about any issue/ bug that was encountered in the game |
| Stack Overflow | Stack Overflow was used to fixed any coding issues encountered while programming the logic of the game |

**4.4 Github**

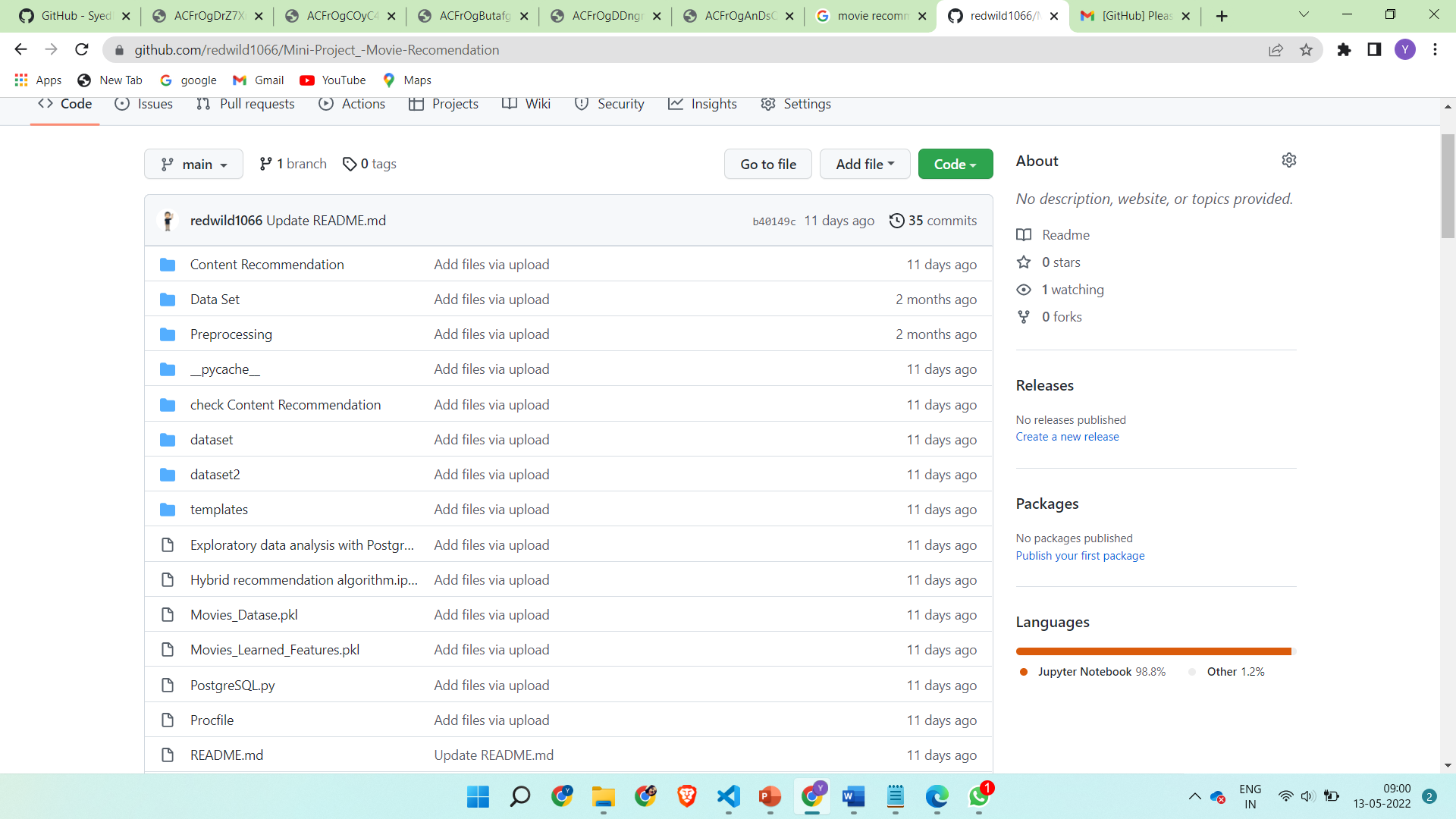
Created a github repository:



Added project members as collaborators:



Pushing files to the repository:



**Chapter 5 Conclusions and further work :**

we have introduced Hybrid Movie Recommendation system , a recommender system for movies.

It allows a user to select his choices from a given set of attributes and then recommend him a movie list based on the cumulative weight of different attributes.

By the nature of our system, it is not an easy task to evaluate the performance since there is no right or wrong recommendation; it is just a matter of opinions. Based on informal evaluations that we carried out over a small set of movie data set .

We would like to work on a larger data set which will enable more meaningful results using our system.

A hybrid approach is taken between context based filtering and collaborative filtering to implement the system. This approach overcomes drawbacks of each individual algorithm and improves the performance of the system. Techniques like Clustering, Similarity and Classification are used to get better recommendations thus increasing precision and accuracy. In future we can work on hybrid recommender using clustering and similarity for better performance. Our approach can be further extended to other domains to recommend songs, video, venue, news, books, tourism and e-commerce sites, etc.

**Future work**

Recommender systems can be a very powerful tool in a company’s brand image , and future developments are going to increase business value even further.

Some of the applications include being able to anticipate seasonal purchases based on recommendations, determine important purchases, and give better recommendations to customers which can increase retention and brand loyalty.

Most businesses will use Recommendation system in future as it will increase the value of the product .

**Bibliography :**

[1]. [H. Zhang, F. Min, D. Ślęzak and B. Shi, "Cost-sensitive regression-based recommender system," 2015 International Conference on Machine Learning and Cybernetics (ICMLC), 2015, pp. 253-258, doi: 10.1109/ICMLC.2015.7340931.](https://ieeexplore.ieee.org/document/7340931)

<br/>

[2]. [G. Lekakos and P. Caravelas, "A hybrid approach for movie recommendation", Multimedia tools and applications, vol. 36, no. 1–2, pp. 55-70, 2008.

updting soon]()

<br/>

[3]. [P. Cremonesi, R. Turrin and F. Airoldi, "Hybrid algorithms for recommending new items", Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 33-40, 2011.](https://dl.acm.org/doi/10.1145/2039320.2039325)

**Acknowledgement**

**University of Mumbai**

**Gesture Controlled Virtual Mouse**

Submitted at the end of semester IV in partial fulfillment of requirements

**Of Bachelors in Technology in Computer Engineering**

by

**ROHAN D’SOUZA**

**Roll No: 16010120013,**

**YASH GAVADE**

**Roll No: 16010120015,**

Guide

****

**Department of Computer Engineering**

**Department of Computer Engineering**

**K. J. Somaiya College of Engineering, Mumbai-77**

**(Autonomous College Affiliated to University of Mumbai)**

**Batch 2020 -2024**

**K. J. Somaiya College of Engineering, Mumbai-77**

**Certificate**

This is to certify that the MINIPROJECT report entitled **A Hybrid Bollywood Movie Recommendation System** submitted by Rohan D’souza , Yash Gavade at the end of semester IV of SY B. Tech are bona fide record for partial fulfillment of requirements for the degree of Bachelors in Computer Engineering of University of Mumbai

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Guide Head of the Department

Date: 7-05-2022

Place: Mumbai-77

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**Certificate of Approval of Examiners**

We certify that this MiniProject report entitled **A Hybrid Bollywood Movie Recommendation System** is bona fiderecord of Mini project work done by by Rohan D’souza , Yash Gavade during semester IV. This Mini project work is submitted at the end of semester IV in partial fulfillment of requirements for the degree of Bachelors in Technology in Computer Engineering of University of Mumbai.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner 2

Date:

Place: Mumbai-77

**K. J. Somaiya College of Engineering, Mumbai-77**

(Autonomous College Affiliated to University of Mumbai)

**DECLARATION**

We declare that this written report submission represents the work done based on our and / or others’ ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

|  |  |
| --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **Roll No. 16010120013** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Signature of the Student**  **Roll No. 16010120015** |

**Date: 13-05-2022**

**Place: Mumbai-77**