

A
Project Progress Report On
MOVIE REVIEW ANALYSIS AND RECOMMENDATION SYSTEM
*Submitted in partial fulfilment of the requirement for the Award of the
degree of*

Bachelor of Technology
In
INFORMATION TECHNOLOGY

Submitted by
AYUSH SRIVASTAVA(Roll No.:1628413008)
AKHILESH KUMAR PAL(Roll No.:1628413003)
ANUBHAV SRIVASTAVA(Roll No.:1628413005)

Under the supervision of
Mr. Shobhit Kumar
(Assistant Professor)



Department of Information Technology
United Institute Of Technology, Naini, Prayagraj, College Code: 284
(Affiliated to AKTU)
December 2019-20

CERTIFICATE

This is to certify that the synopsis report entitled “Movie Review Analysis And Recommendation System” submitted by Ayush Srivastava(Roll.no 1628413008),Akhilesh kumar.Pal(Roll.no 1628413003) and Anubhav Srivastava(Roll.no 1628413005) in partial fulfillment of the requirement for the award of degree B.Tech in Information Technology at UNITED INSTITUTE OF TECH NOLOGY ,NAINI ALLAHABAD in an original work carried by them in my supervision and guidance.

The matter embodied in the synopsis report is original and has not been submitted to any university/institute for the award of any degree.

Date:

Supervisor

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and believe it contain no matter previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other Institute of higher learning except where due acknowledgement has been made in the text.

Signature:

Name:

Roll.no:

Date:

ACKNOWLEDGEMENT

First of all, I would like to express the deepest sense of gratitude to our mentor **Mr.Shobit kumar (Asst.Prof.CSE Dept)** ,The Head Of Department, **Mr. Abhishek Malviya** for being an outstanding advisor and for excellent guidance. They have been a constant source of encouragement and an inspiring figure in this endeavor. Their support and individual suggestions made this work successful. Their readinesses for consultation at all time, educative comments, concerns and assistance have been individual.

More, I am also thankful to all the other team members for always supporting and helping me in every possible way.

Above all, thanks to the Almighty for giving me such an amazing opportunity to work on a live project and for giving me strength me and courage to accomplish such an challenging project at such an unripe age of my carrier.

Signature:

Name:

Roll No:

Signature:

Name:

Roll No:

Signature:

Name:

Roll No:

ABSTRACT

The project titled “Movie Review Analysis and Recommendation System” websites have emerged as one of the platforms to raise users' opinions and influence the way any business is commercialized. Our system is a movie review system which will generate polarity related to the movies which are released. Generally, the reviews are generated provide star rating of a movie. Users just have to see the ratings which are generated by analyzing the ratings given by other users to that movie and have to take his/her decisions. Such ratings are easily understandable by any user. But they don't give clear idea of how the movie is. They are helpful only in the scenario where if any movie is excellent or very poor. The scenario where movie is average, star ratings prove bit confuse for any user. They don't get clear views of what the other users think of that movie.

TABLE OF CONTENT

CERTIFICATE	ii
DECLARATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
CHAPTER 1: INTRODUCTION	1
1.1 PROBLEM STATEMENT	1
1.2 PROPOSED MODEL	1
1.3 Module Description	1
CHAPTER 2: Feasibility Study	2-3
2.1 Technical Feasibility	2
2.2 Legal Feasibility	2
2.3 Operational Feasibility	2
2.4 Schedule Feasibility	3
2.5 Economic Feasibility	3
CHAPTER 3: Software Model	4-5
3.1 Iterative Model	4
CHAPTER 4: DFD	6-10
4.1 data flow Diagram	6
4.2 Attributes	6
4.3 Zero Level DFD	7
4.4 Level 1 DFD(8
4.5 Level 1 DFD ADMIN	9
4.6 Level 1 DFD Member	10
4.7 Level 1 DFD User	10
CHAPTER 5: E R Diagram	11
CHAPTER 6: Flow Chart	12
CHAPTER 7: Hardware And Software Requirement	
7.1 Software Requirement	13
7.2 Hardware Requirement	13
CHAPTER 8: FUTURE SCOPE	14
CHAPTER 9: Screenshot	15-26

References

CHAPTER 1

1.1 INTRODUCTION:

As we know in today's era people opinion matter more in field of business as well as for the any critic review of products or any sites. So we are doing a review analysis in this and gives the rating according to our analysis.

1.2 Problem Statement:

Now day's number of movie released in a particular day and different person has different choices to watch them but it is really difficult for a user to choose which movie is best for them to invest money and time to watch it. So the people are required a genuine opinion and reaction of people who already know about the movie after the release which helps them to decide which one is better to watch it.

1.3 Proposed Model: Reasons for implementation

People's opinion has become one of the extremely important sources for various services in ever-growing popular social networks. Our system is a movie review system which will generate polarity related to the movies which are released. Generally, the reviews are generated provide star rating of a movie. Users just have to see the ratings which are generated by analyzing the ratings given by other users to that movie and have to take his/her decisions. Such ratings are easily understandable by any user. But they don't give clear idea of how the movie is. They are helpful only in the scenario where if any movie is excellent or very poor. The scenario where movie is average, star ratings prove bit confuse for any user. They don't get clear views of what the other users think of that movie.

So our system is a movie review system which will generate polarity graph related to the movies which are released. The system will generate better polarity after analyzing the user review. Also our system will provide star ratings too.

1.4 Modules Description

- **Admin:** This module contain specific task to manage the activity perform by different user, and to ensure the proper working of system
- **User:** This module will authenticate the users. It makes every user unique. Simply the user has to give input as the movie name and the review analysis in the form of rating and graph will provide as output.
- **Login:** Login module contains the login id and password which will authenticate the user as well as admin and it also provide the time of login and logout.

CHAPTER 2

2.1 Feasibility Study

A feasibility study is an analysis that takes all of a project's relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

The goals of feasibility studies are as follows:

- To understand thoroughly all aspects of a project, concept, or plan
- To become aware of any potential problems that could occur while implementing the project
- To determine if, after considering all significant factors, the project is viable—that is, worth undertaking

2.1.1 Technical feasibility :

Technical feasibility is the assessment of the technical requirements of a project or product to find out what technical resources a project requires. Technical feasibility is not complete until the same technical assessment is done on the company to establish that it has the technical capabilities to carry out the implementation of the project to completion within the required time. It is focused on the available hardware and software to be used for a project

2.1.2 Legal feasibility:

This is an assessment of whether the project meets the legal requirements that exist for implementation. An example is whether a new building meets the requirements of the law, whether the location was suitably chosen and construction is approved by the authorities. Legal feasibility also encompasses the ethical aspects of a project, for example does a new plant dump its waste in a manner that is environmentally friendly. The project may also be a new undertaking that is not covered under any law and so this must also be checked.

2.1.3 Operational feasibility:

Operational feasibility ascertains how well the implementation of a project fits in with the current organizational business structure. The solutions to a current problem must come as close as possible to a perfect fit with the organizational structure and be able to be applied to solve other arising problems. The opportunities that come along the way during the solution implementation must be able to be harnessed for even easier implementation.

2.2.4 Schedule feasibility:

Before you start a project, you have to ask the following questions: When is the project due? Are there legal obligations related to the schedule? Within what time is the project or product viable? Are the deadlines attainable? This is an assessment of the time frame for the delivery of the product and whether or not the product will still be needed after that time.

2.1.5. Economic feasibility

Here, a study is done on the project to see how long it would take for the project to reach break-even point. It is similar to a cost benefit analysis. Every project incurs a cost and it is prudent for financial planning purposes to know exactly when to expect returns. Also, to anticipate the capital required to complete the project. This is not limited to for-profit projects only as they will also incur a financial cost.

CHAPTER 3

3.SOFTWARE MODEL

3.1 Iterative Model:

The **iterative model** is a particular implementation of a software development life cycle (SDLC) that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete. When discussing the **iterative method**, the concept of **incremental development** will also often be used liberally and interchangeably, which describes the incremental alterations made during the design and implementation of each new iteration.

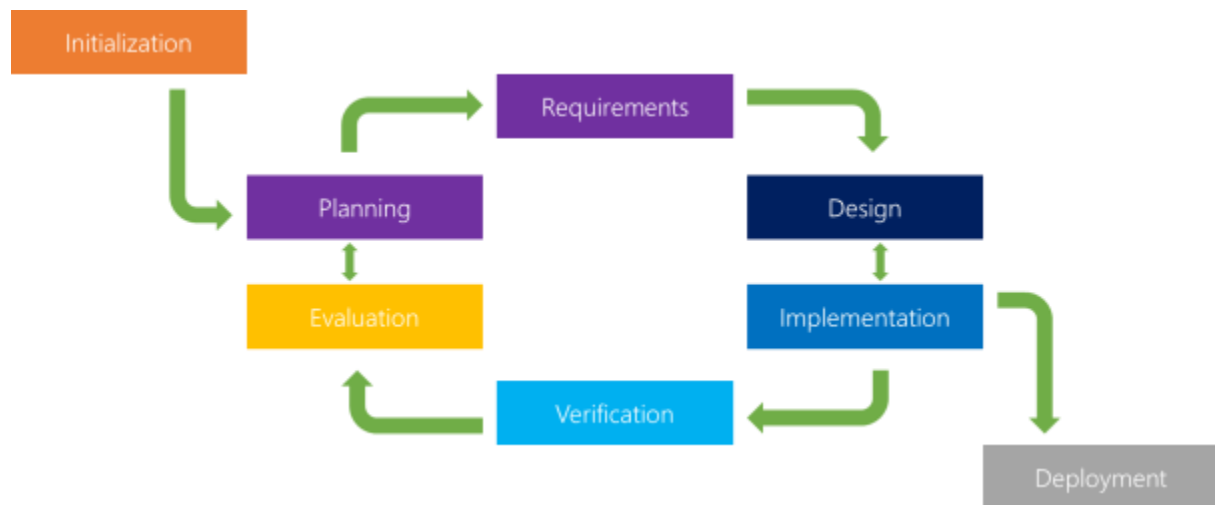
Process in Iterative Model

Unlike the more traditional **waterfall model**, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software. Enhancements can quickly be recognized and implemented throughout each iteration, allowing the next iteration to be at least marginally better than the last.

- **Planning & Requirements:** As with most any development project, the first step is go through an initial **planning** stage to map out the specification documents, establish software or hardware **requirements**, and generally prepare for the upcoming stages of the cycle.
- **Analysis & Design:** Once **planning** is complete, an **analysis** is performed to nail down the appropriate business logic, database models, and the like that will be required at this stage in the project. The **design** stage also occurs here, establishing any technical requirements (languages, data layers, services, etc) that will be utilized in order to meet the needs of the **analysis** stage.

Implementation: With the **planning** and **analysis** out of the way, the actual **implementation** and coding process can now begin. All planning, specification, and design docs up to this point are coded and implemented into this initial iteration of the project.

- **Testing:** Once this current build iteration has been coded and **implemented**, the next step is to go through a series of **testing** procedures to identify and locate any potential bugs or issues that have cropped up.
- **Evaluation:** Once all prior stages have been completed, it is time for a thorough **evaluation** of development up to this stage. This allows the entire team, as well as clients or other outside parties, to examine where the project is at, where it needs to be, what can or should change, and so on.

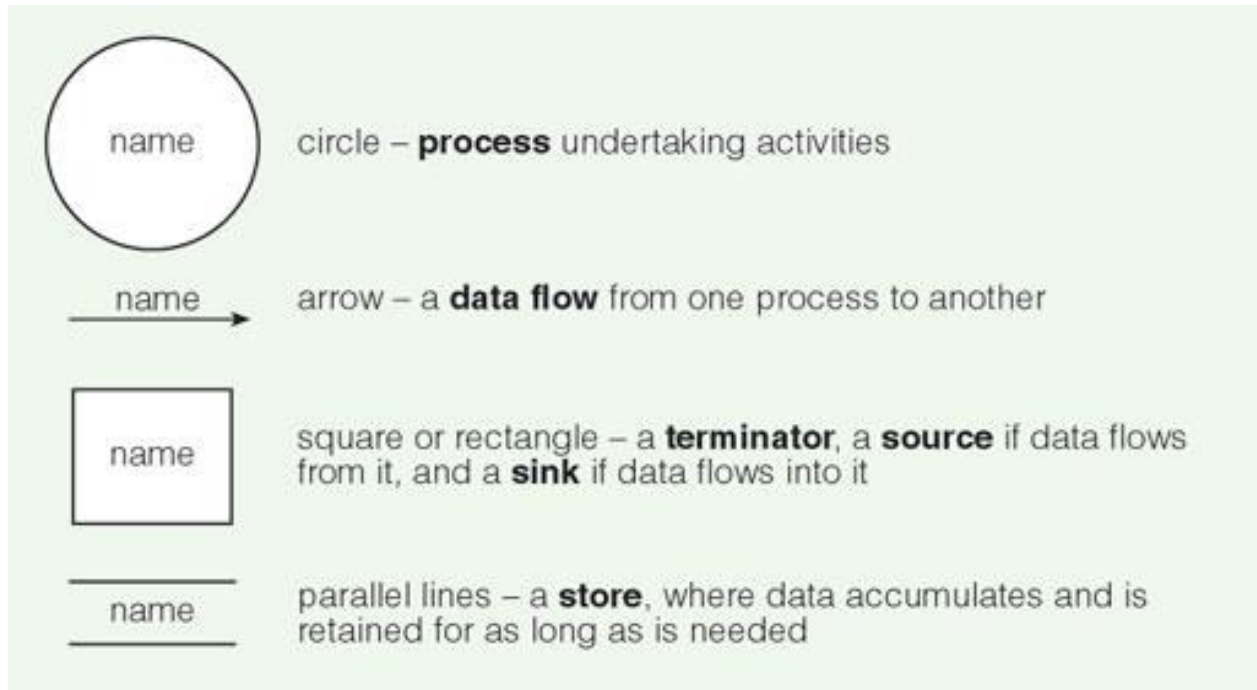


CHAPTER 4

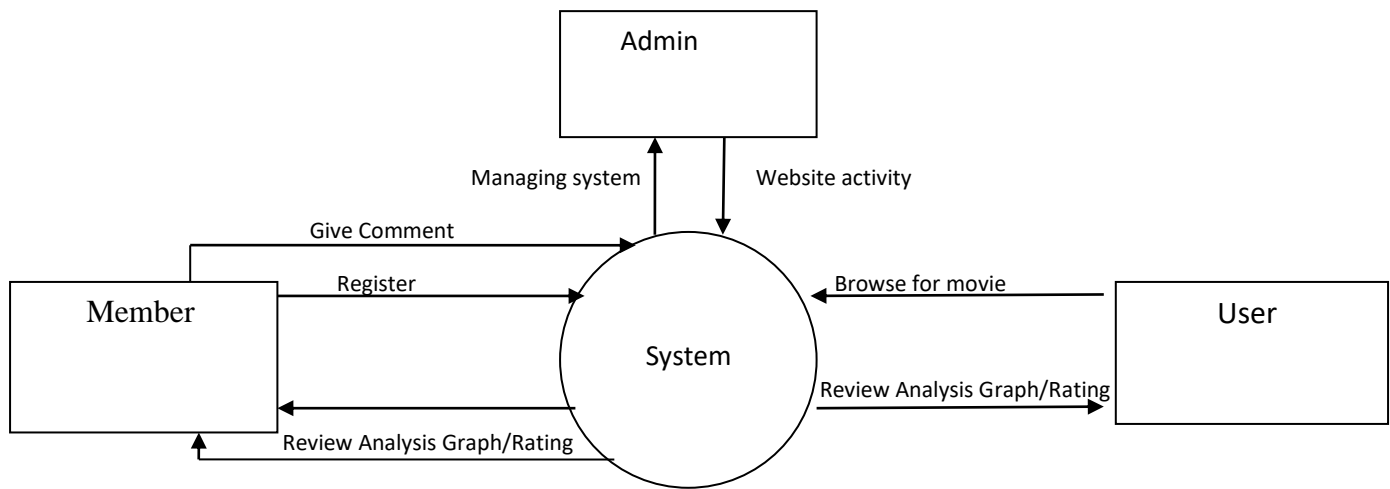
DATA FLOW DIAGRAM

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

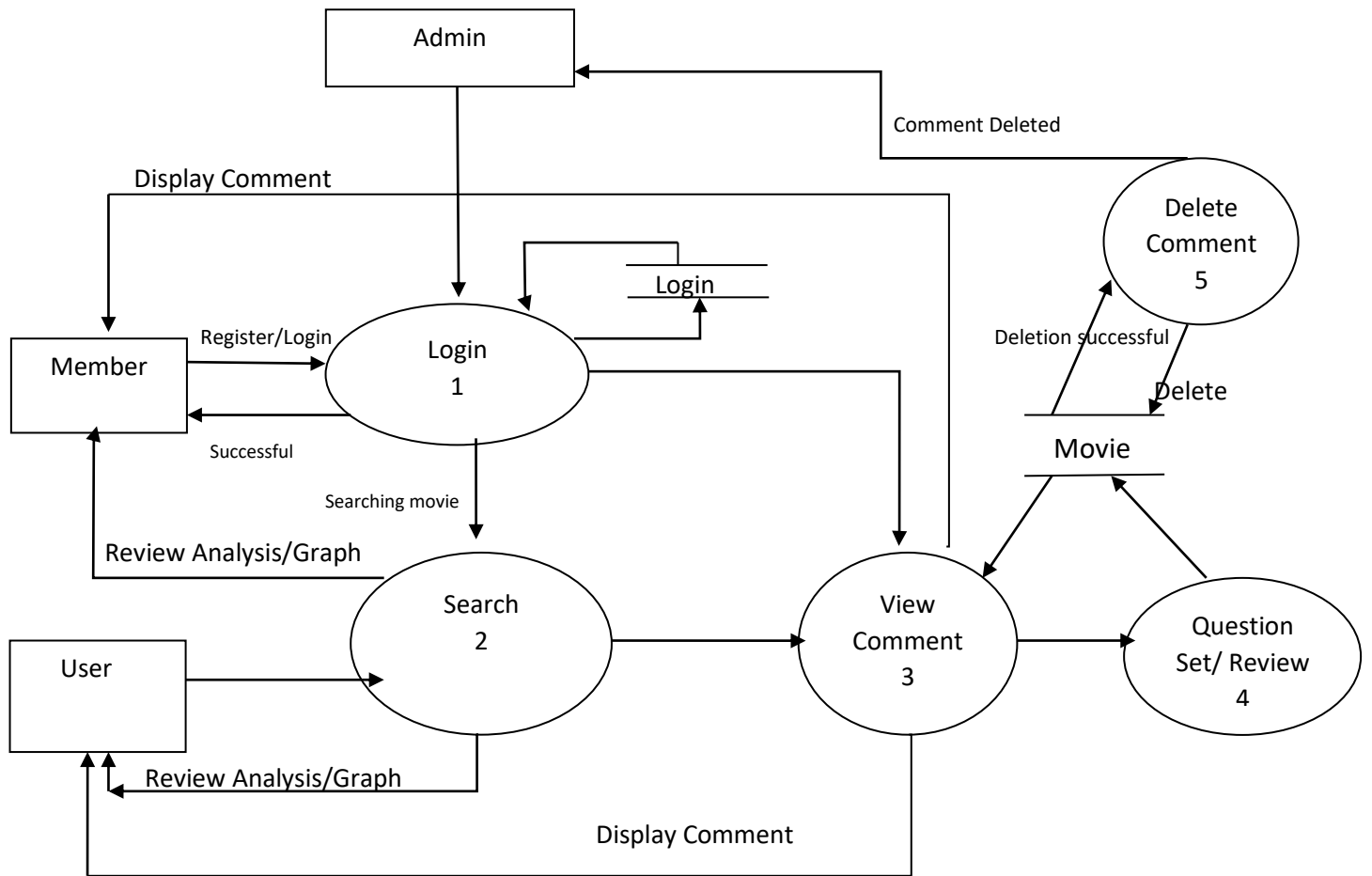
ATTRIBUTE



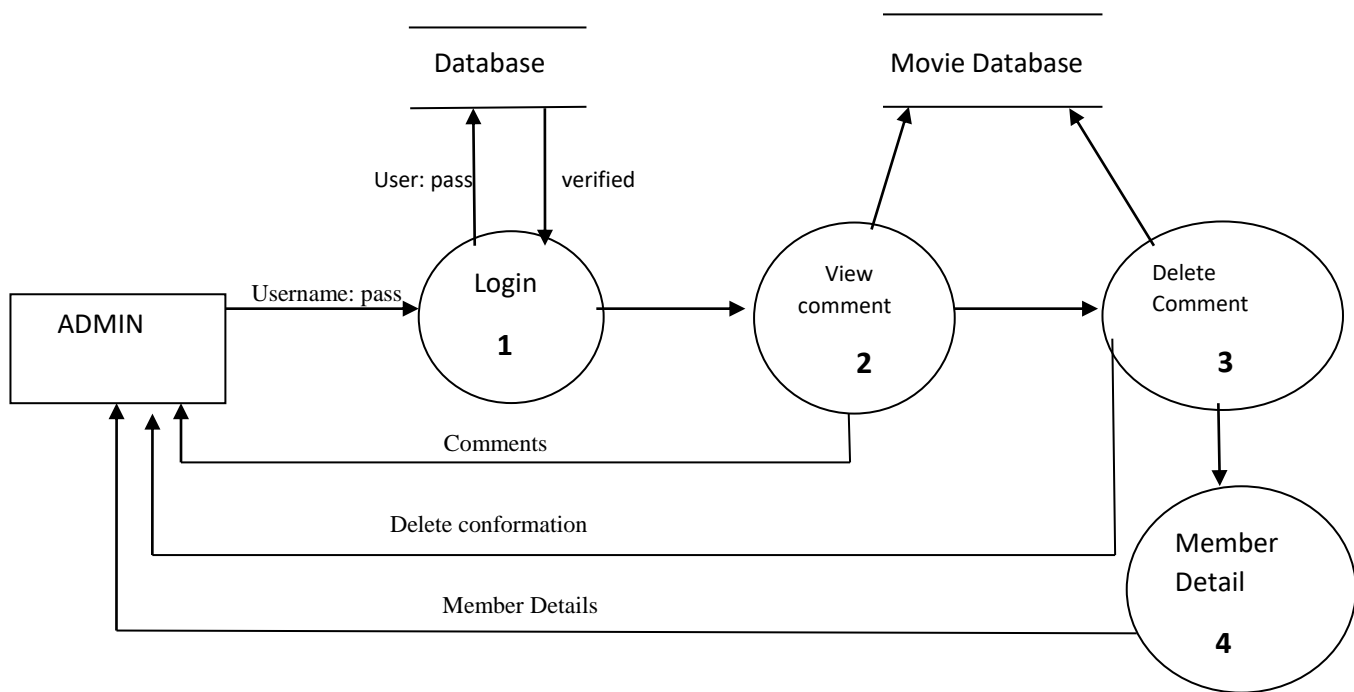
DATA FLOW DIAGRAM



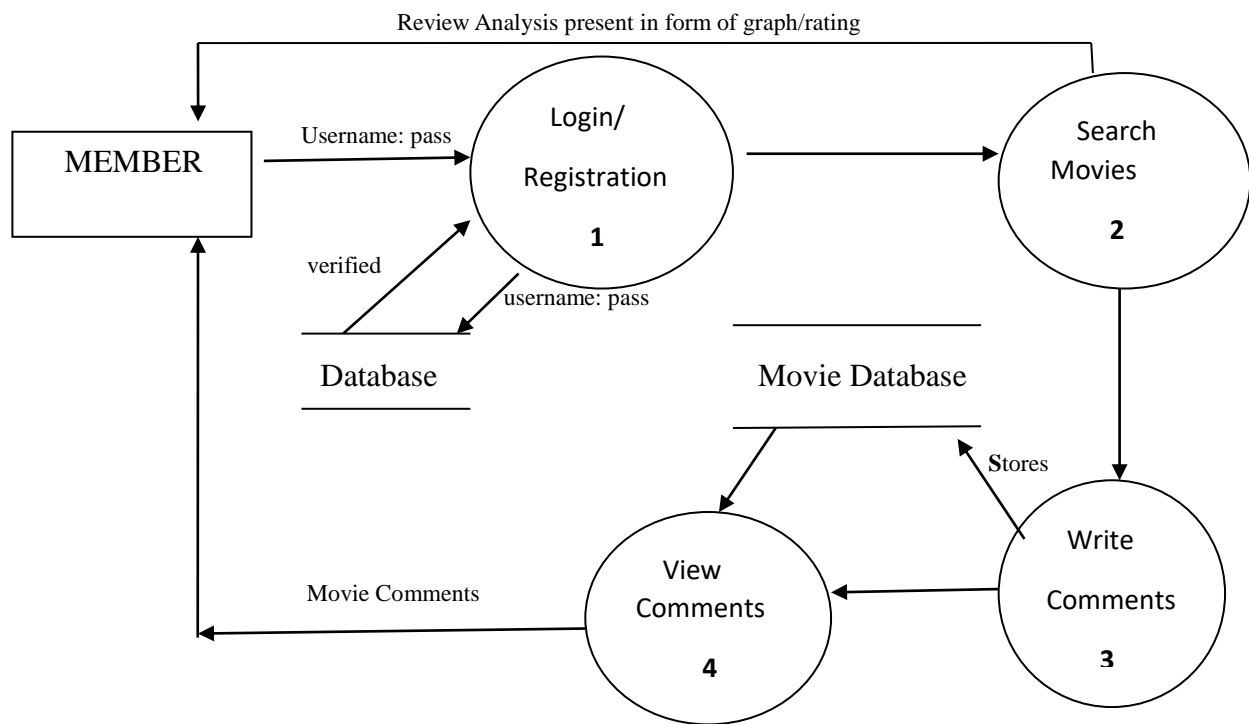
ZERO LEVEL DFD



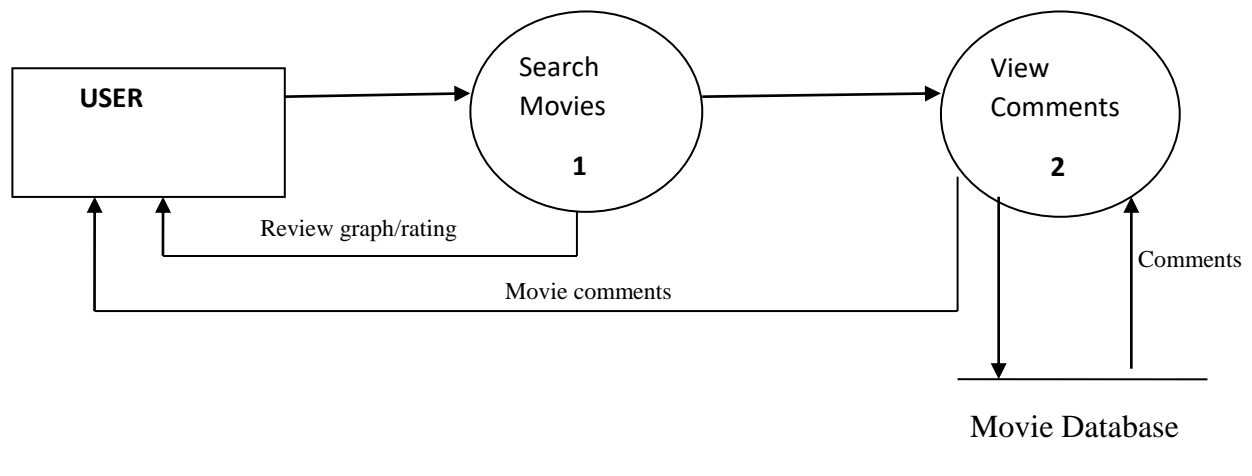
Level 1 Data Flow Diagram



LEVEL 1 DFD FOR ADMIN



LEVEL 1 DFD FOR MEMBER

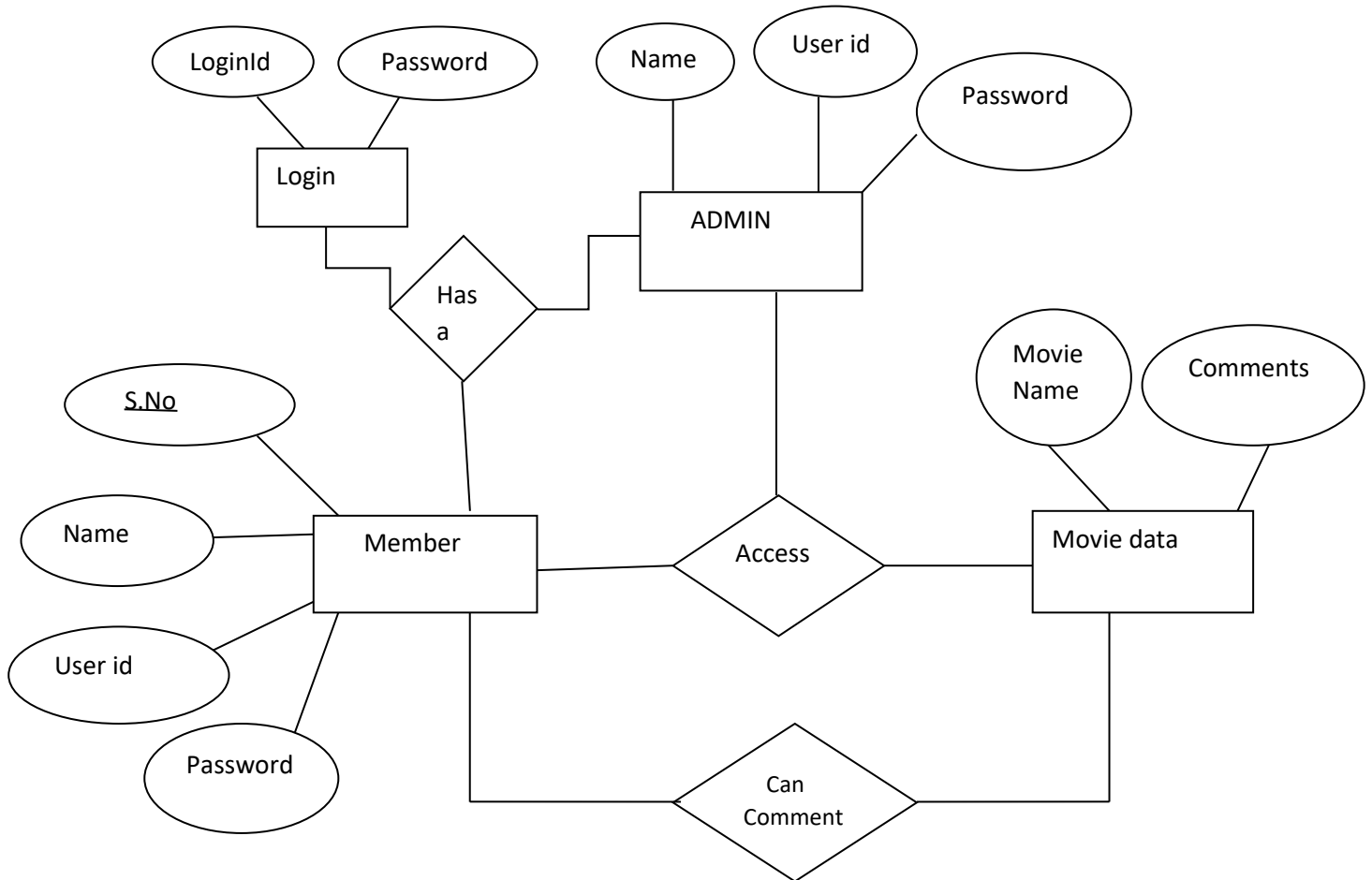


LEVEL 1 DFD FOR USER

CHAPTER 5

ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set

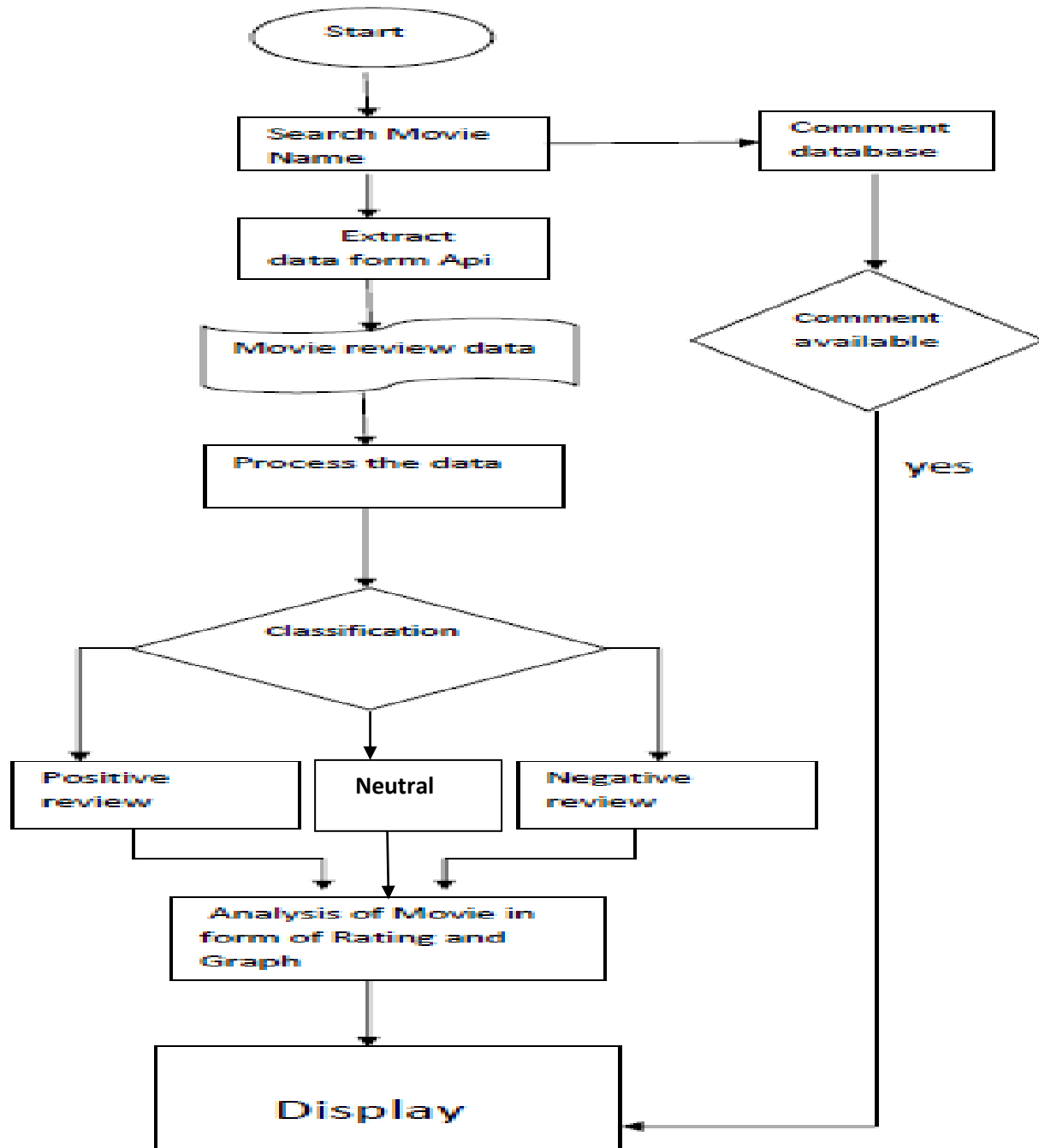


ER DIAGRAM

CHAPTER 6

FLOW CHART

A **flowchart** is a type of diagram that represents a workflow or process. A **flowchart** can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The **flowchart** shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.



CHAPTER 7

7.1 Software Requirements:

7.1.1 Front-End

- Django
- HTML

7.1.2 Back-End

- **Python**
- **MySql**

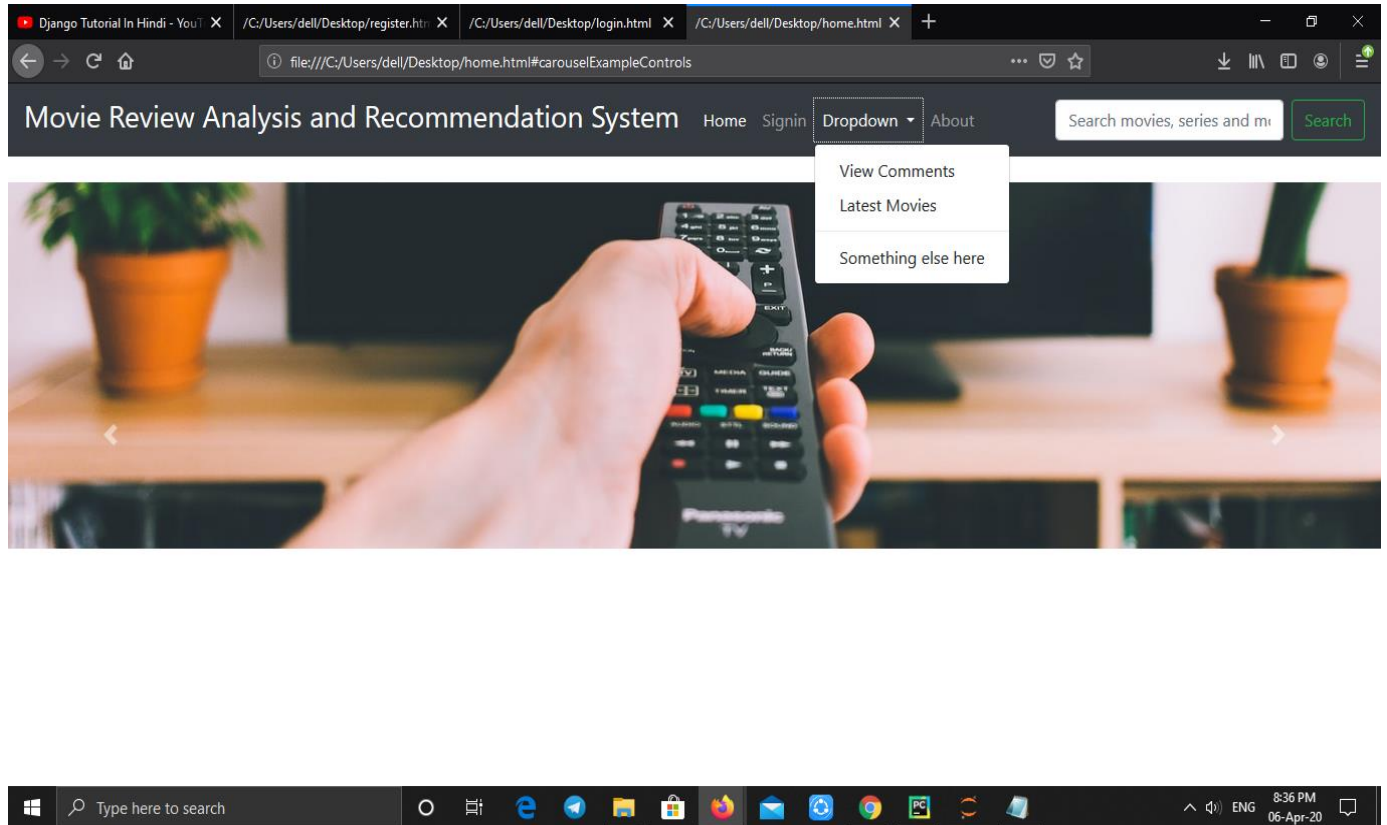
7.2 Hardware Components:

- Processor – i3
- Hard Disk – 5 GB
- Memory – 1GB RAM

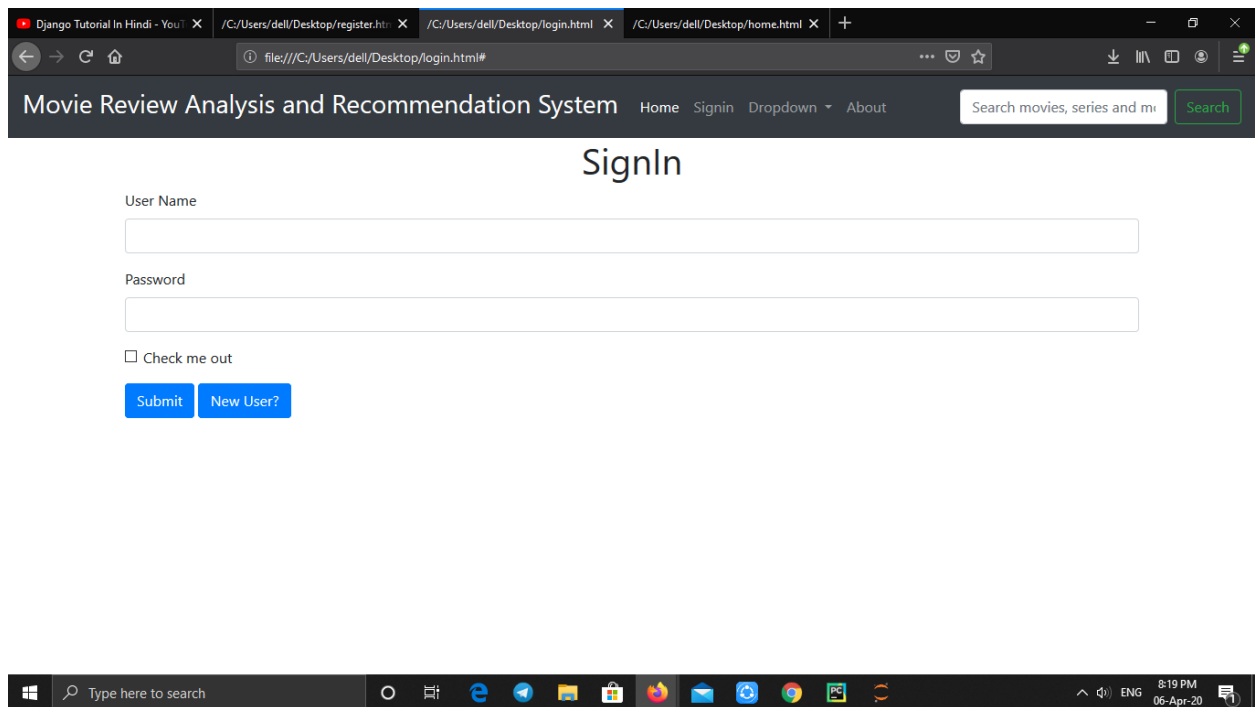
Future Scope:

- This project will give clear view about the movie how the movie is and what the people think about that particular movie the whole analysis will be provided to the user which will be beneficial for them to choose a good movie to watch.
- In this system a person can search all categories of movies.
- In future we are able to provide booking of movie tickets.
- We also provide the reward points to our member for the reviews.

SCREENSHOTS



Home page



Sign in page

The screenshot shows a web browser window with the address bar displaying `file:///C:/Users/dell/Desktop/register.html`. The page title is "Movie Review Analysis and Recommendation System". The navigation bar includes links for "Home", "Signin", "Dropdown", and "About", along with a search bar containing the text "Search movies, series and m..." and a "Search" button.

Register

The registration form consists of the following fields and elements:

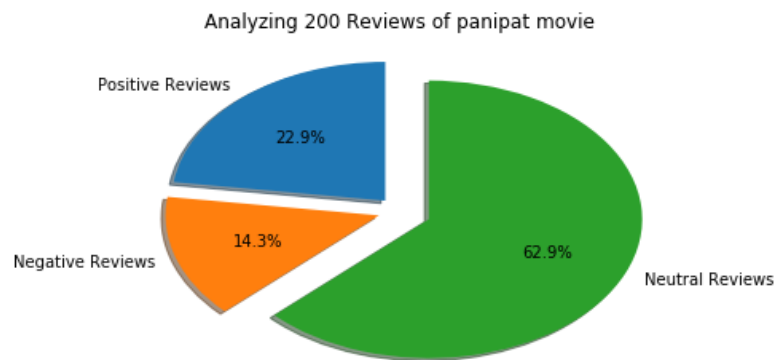
- First name:** A text input field with a green checkmark icon on the right.
- Last name:** A text input field with a green checkmark icon on the right.
- Username:** A text input field with an "@" icon on the left and a red error icon on the right. Below the field, the message "Please choose a username." is displayed in red.
- Password:** A text input field with a red error icon on the right. Below the field, the message "Please provide a valid password." is displayed in red.
- Agree to terms and conditions:** A checkbox followed by the text "Agree to terms and conditions" and "You must agree before submitting." in red.
- Submit form:** A blue button.

The Windows taskbar at the bottom shows the search bar with the text "Type here to search", several application icons, and the system clock displaying "8:20 PM 06-Apr-20".

For new user register

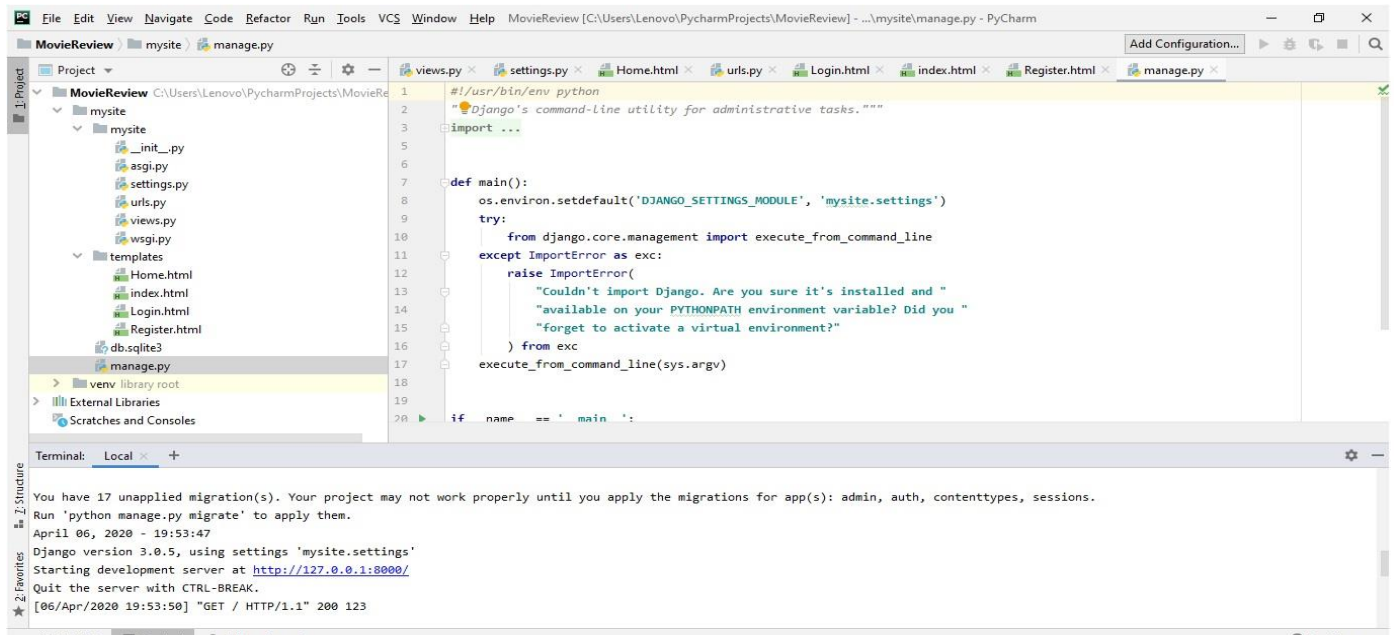

```
if __name__ == '__main__':  
    # calling main function  
    main()
```

Enter a search query... panipat movie
Enter no. of Reviews to analyze 200
Positive Reviews percentage: 22.857142857142858 %
Negative Reviews percentage: 14.285714285714286 %
Neutral Reviews percentage: 62.857142857142854 %



Output of searched movie

Coding section



The screenshot shows the PyCharm IDE with the `manage.py` file open. The file contains the following code:

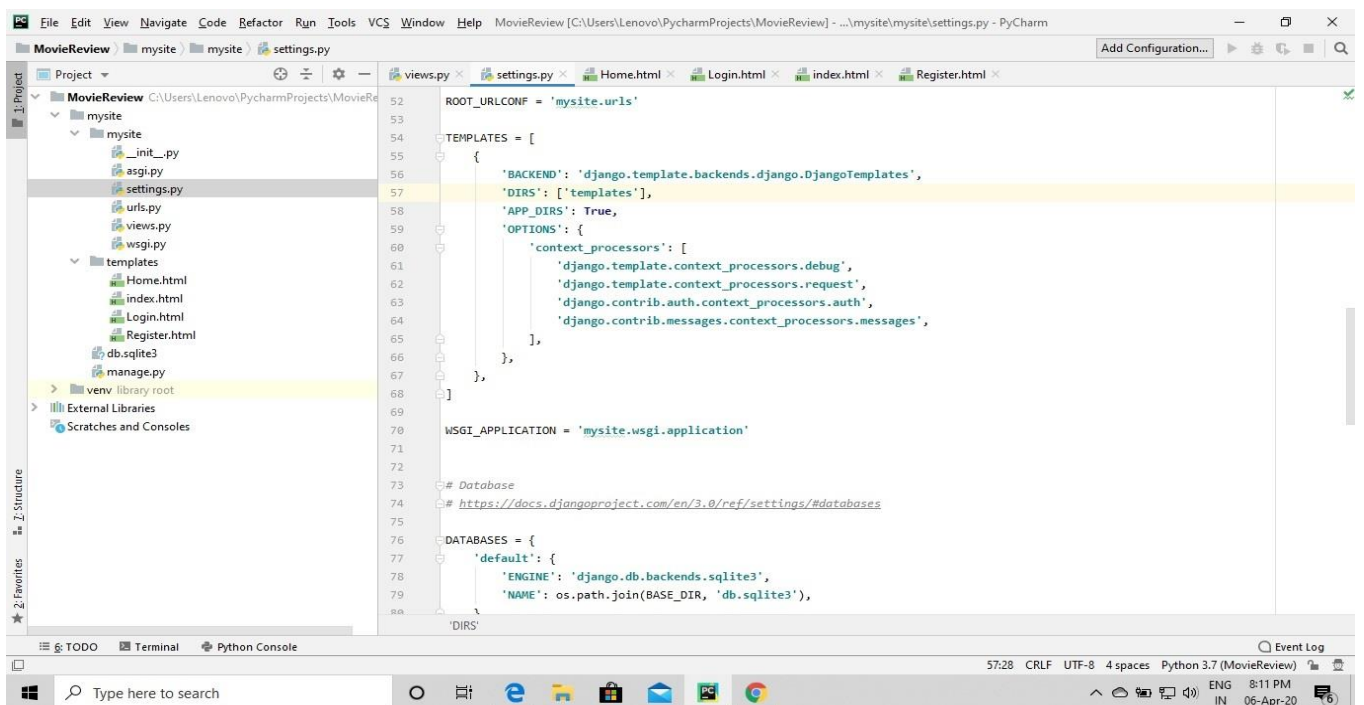
```
#!/usr/bin/env python
# Django's command-line utility for administrative tasks.
import ...

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'mysite.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

The terminal output shows the following messages:

```
You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
April 06, 2020 - 19:53:47
Django version 3.0.5, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[06/Apr/2020 19:53:50] "GET / HTTP/1.1" 200 123
```



The screenshot shows the PyCharm IDE with the `settings.py` file open. The file contains the following code:

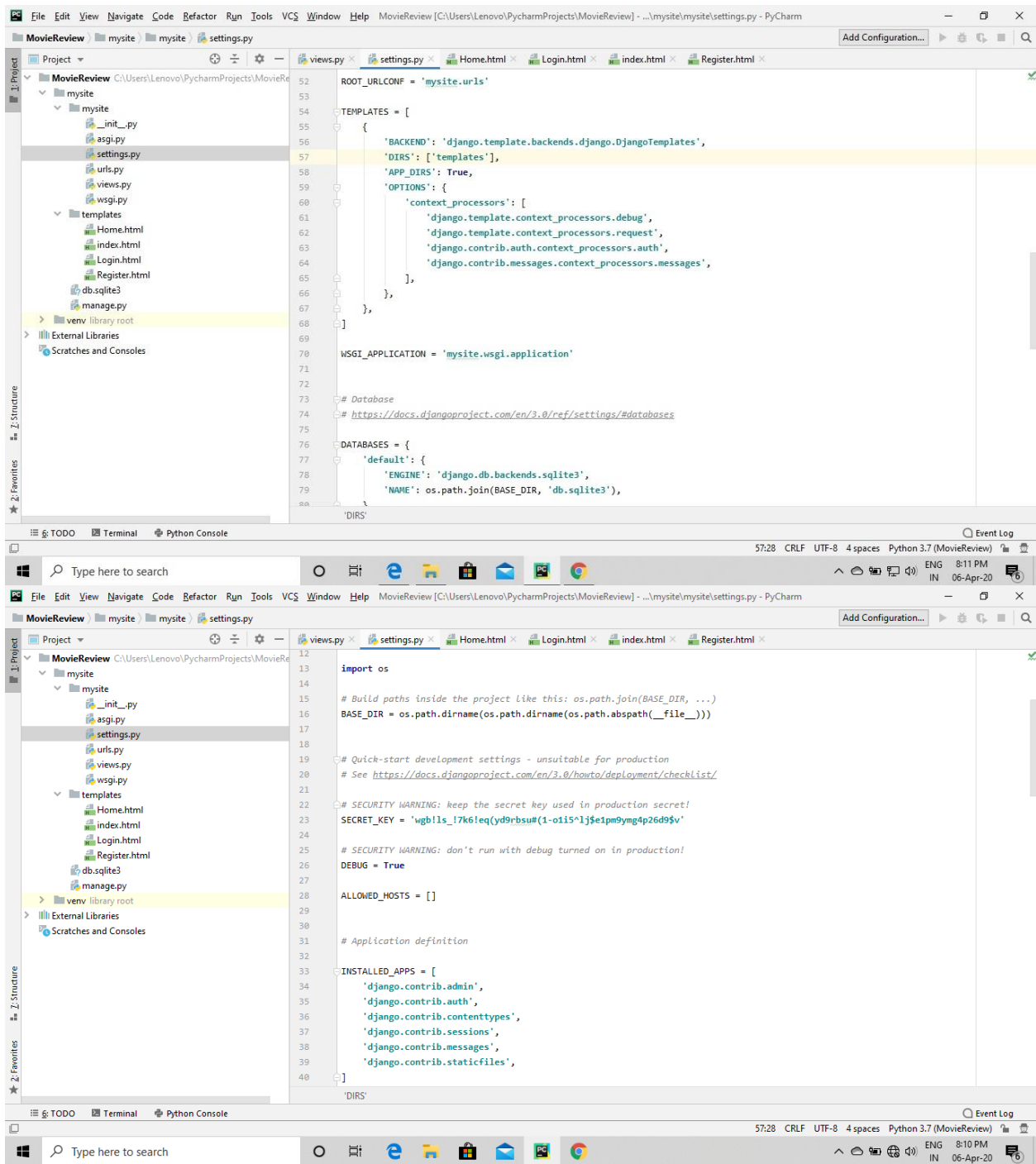
```
ROOT_URLCONF = 'mysite.urls'

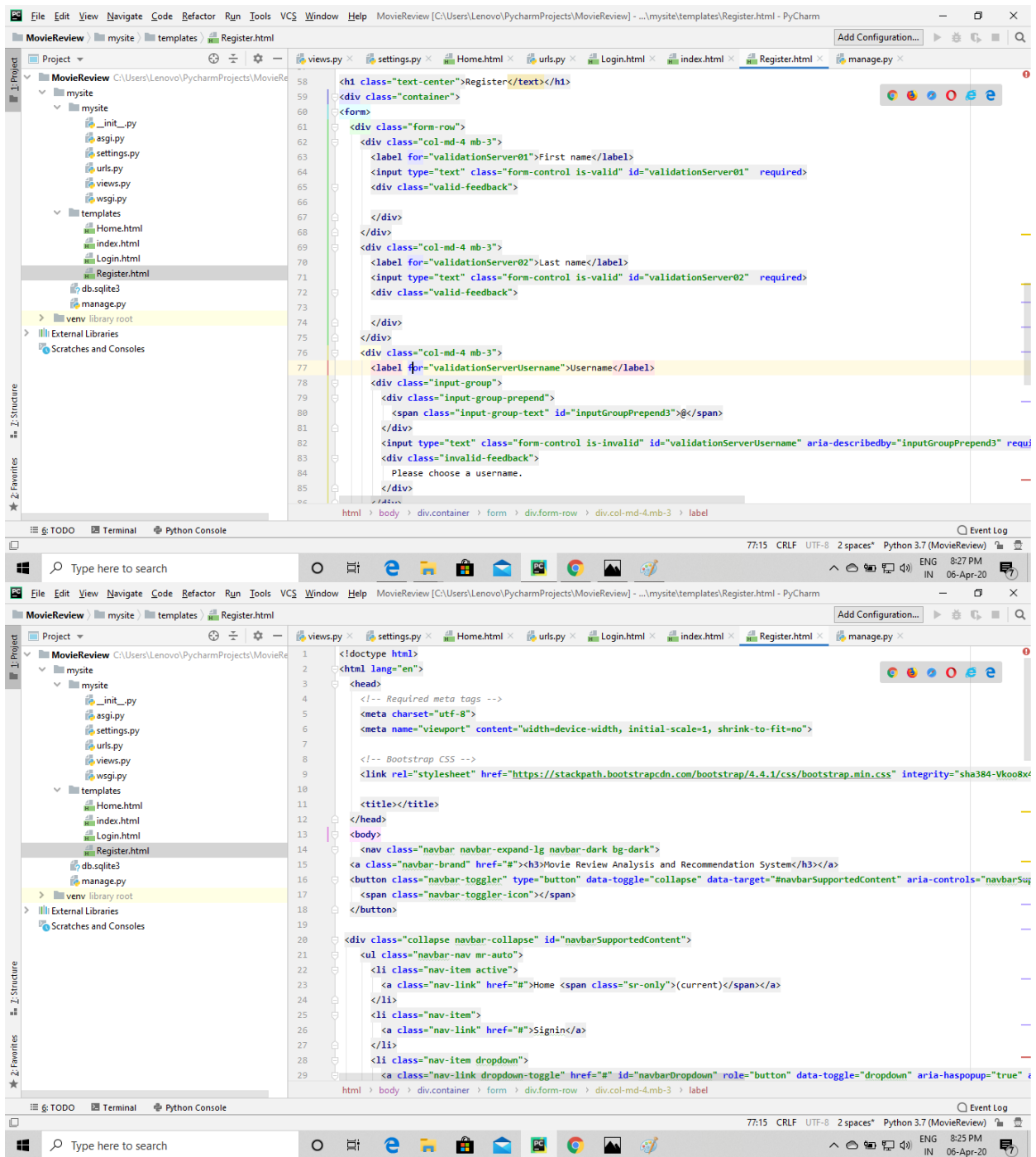
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'mysite.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```





```

import re

import tweepy

from tweepy import OAuthHandler

from textblob import TextBlob

import matplotlib.pyplot as plt


class TwitterClient(object):

    """
    Generic Twitter Class for sentiment analysis.
    """

    def __init__(self):
        """
        Class constructor or initialization method.
        """

        # keys and tokens from the Twitter Dev Console
        consumer_key = 'ATfKfmlU88EffYQzIF8SQhiYX'

        consumer_secret =
'dy8ao96WUqEFjEGi7dnIEiAT2Vq6EDDkXPOv0k50DcSr9dnwsz'

        access_token = '180739880-
ncdWLjX4vDYfaCh1Foj2dltv6TNOPa31Nxb4zvXy'

        access_token_secret =
'3vevh5nD8Z6IXIHn2XcrD7pnxtMVTZRbwu49rPLVRNW1n'


        # attempt authentication

        try:

```

```

    # create OAuthHandler object
    self.auth = OAuthHandler(consumer_key, consumer_secret)

    # set access token and secret
    self.auth.set_access_token(access_token, access_token_secret)

    # create tweepy API object to fetch tweets
    self.api = tweepy.API(self.auth)

except:
    print("Error: Authentication Failed")


def clean_tweet(self, tweet):
    """
    Utility function to clean tweet text by removing links, special characters
    using simple regex statements.
    """
    return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\w+\S+)",
        " ", tweet).split())


def get_tweet_sentiment(self, tweet):
    """
    Utility function to classify sentiment of passed tweet
    using textblob's sentiment method
    """
    # create TextBlob object of passed tweet text
    analysis = TextBlob(self.clean_tweet(tweet))

```

```

# set sentiment
if analysis.sentiment.polarity > 0:
    return 'positive'
elif analysis.sentiment.polarity == 0:
    return 'neutral'
else:
    return 'negative'

def get_tweets(self, query, count = 10):
    """
    Main function to fetch tweets and parse them.
    """
    # empty list to store parsed tweets
    tweets = []

    try:
        # call twitter api to fetch tweets
        fetched_tweets = self.api.search(q = query, count = count)

        # parsing tweets one by one
        for tweet in fetched_tweets:
            # empty dictionary to store required params of a tweet
            parsed_tweet = {}

```

```

# saving text of tweet
parsed_tweet['text'] = tweet.text

# saving sentiment of tweet
parsed_tweet['sentiment'] = self.get_tweet_sentiment(tweet.text)


# appending parsed tweet to tweets list
if tweet.retweet_count > 0:
    # if tweet has retweets, ensure that it is appended only once
    if parsed_tweet not in tweets:
        tweets.append(parsed_tweet)
    else:
        tweets.append(parsed_tweet)


# return parsed tweets
return tweets


except tweepy.TweepError as e:
    # print error (if any)
    print("Error : " + str(e))


def main():
    # creating object of TwitterClient Class
    api = TwitterClient()

    # calling function to get tweets

```



```

q = input("Enter a search query... ")
n = int(input("Enter no. of Reviews to analyze "))
tweets = api.get_tweets(query = q, count = n)

# picking positive tweets from tweets
ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'positive']

# percentage of positive tweets
print("Positive Reviews percentage: {}
%".format(100*len(ptweets)/len(tweets)))

# picking negative tweets from tweets
ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'negative']

# percentage of negative tweets
print("Negative Reviews percentage: {}
%".format(100*len(ntweets)/len(tweets)))

# percentage of neutral tweets
print("Neutral Reviews percentage: {} %"
      .format(100*(len(tweets) - len(ntweets) - len(ptweets))/len(tweets)))

sizes = [len(ptweets),len(ntweets),len(tweets) - len(ntweets) - len(ptweets)]
label = ['Positive Reviews','Negative Reviews','Neutral Reviews']
explode = (0.1,0.1,0.15)

plt.pie(sizes,explode=explode,labels=label,autopct='%1f%%'
        ,shadow=True,startangle=90)

plt.title(f"Analyzing {n} Reviews of {q}")
plt.show()

```

```
if __name__ == "__main__":  
    # calling main function  
    main()
```

References:

1. <http://www.w3schools.com>
2. <http://www.geeksforgeeks.org>