Implementation Plan

# INTRODUCTION

## Purpose

We want to make a software that can analyse user reviews for a given movie by different people. In this way we can deduce overall sentimental orientation of the movie.

To gather reviews and analyse them, we plan to make a web application that can call different API services for collecting reviews. Then we plan to use simple machine learning algorithm for creating a model than can analyse the reviews and output overall sentiment (positive or negative).

## System Overview

The system will take movie name as input from the user. Then the software will scrape different websites on internet, will call various API’s to gather information about the movie and comments on the movie from different users of various age groups and gender. Then the software will make sheet out of this details and will analyse and plot graphs to show all these details.

# MANAGEMENT OVERVIEW

## Description of Implementation

The web application has two parts :

1. Front End Part : Where the user will enter the movie name and details about the movie will be shown.
2. Back End Part : Where various APIs will be called for gathering information, reviews will be analysed and graphs will be plotted and shown to the front end.

We plan to use **Django** for implementing this web application. The front end part will be made in plain html and css. For back end part we plan to use the python. Then we’ll integrate the front end and back end part using Django MVC framework.

The system will be implemented using instant-on approach.

## Major Tasks

* 1. **Creating Front End that can take User input :** Designing the index page that will contain a input field for movie name. User can write a movie name and then submit the request. A simple pc is required to make the index page and a web browser is required to test the code.
  2. **Searching for API services that can return the user reviews** **:** Exploring different APIs that can possible return user reviews for a given movie along with user information like age, gender, location etc. A web browser is required for this purpose and active internet connection.
  3. **Writing Machine Learning Algorithm for analysing the reviews :** After getting the reviews, we need to classify it as positive or negative. We plan to use simple keyword based algorithm, that will check percentage of positive and negative words in the review and classify it according to higher percentage words. For this purpose a PC and active internet connection is required.
  4. **Creating models that can plot graphs on the basis of analysed review :** Once we get the overall sentimental orientation of the movie, we plan to plot the graphs to display the results. For this purpose a PC and active internet connection is required. We’re planning to use some library for plotting the graphs if needed.
  5. **Creating Backend Server :** After the getting the movie name as input, the web browser will send a POST request to the backend. The backend will call different APIs with query parameter as the movie name and collect the reviews. Then will call the movie analyser API for analysing and classifying the movies. After getting the sentiment it will call the graph plotter API for plotting the graphs to show out the results. We’re planning to use Django for creating the backend server and merging it with the front end part.

## Implementation Schedule

| **TASK** | **DATE** |
| --- | --- |
| Creating Front End for User Input | 28th February |
| Searching for API Services | 3rd March |
| Writing the Machine Learning Model for analysis reviews | 10th March |
| Creating Models for plotting graphs | 14th March |
| Creating the Backend Server | 24th March |
| Merging Backend with Front End | 30th March |

# IMPLEMENTATION SUPPORT

## Hardware, Software

### Hardware

Computer, LAN or Wi-Fi Connection

### Software

Django, Web Browser, Visual Studio Code, Windows operating system

# IMPLEMENTATION REQUIREMENTS BY SITE

This section describes specific implementation requirements and procedures. If these requirements and procedures differ by site, repeat these subsections for each site; if they are the same for each site, or if there is only one implementation site, use these subsections only once. The “X” in the subsection number should be replaced with a sequenced number beginning with I. Each subsection with the same value of “X” is associated with the same implementation site. If a complete set of subsections will be associated with each implementation site, then “X” is assigned a new value for each site.

## Movie Rating Analysis System

### Site Requirements

* Hardware Requirements - PC, Internet Connection (LAN or Wi-Fi)
* Software Requirements - Operating System (Windows, Linux, MacOS), Text Editor (Visual Studio Code, Sublime text, Notepad), Web Browser (Chrome, Firefox)
* Data Requirements -Movie reviews along with user details (age, gender, location etc). Reviews should be in english language.

### Site implementation Details

The web application has two parts :

1. Front End Part : Where the user will enter the movie name and details about the movie will be shown.
2. Back End Part : Where various APIs will be called for gathering information, reviews will be analysed and graphs will be plotted and shown to the front end.

We plan to use **Django** for implementing this web application. The front end part will be made in plain html and css. For back end part we plan to use the python. Then we’ll integrate the front end and back end part using Django MVC framework.

We’ll be using Windows operating system and Visual studio code for implementing and programming the movie rating analysis system.

# Acceptance CriteriA

## The web application should take any valid movie name and show the results.

1. The web application should fetch results in linear time.
2. The software should not crash while fetching the results.
3. The website should take parallel requests from different users.