**MINI PROJECT DOCUMENTATION**

**College Website** created for S**tudents** to choose the best company on particular domain, so for that we are using **Natural language Processing** ,**Web Technologies** and **Machine Learning** which includes **Sentiment Analysis**.

The approach towards the project:

1. The backend part (Sentiment Analysis)
2. The frontend part (User friendly website)
3. Finally establish the relation among them

Technologies/Algorithms/Libraries in each Module:

1. Sentiment Analysis:

* Pandas
* Numpy
* nltk
* nltk.sentiment.vader import SentimentIntensityAnalyzer
* textblob
* words from word cloud
* seaborn
* Matplotlib
* Naive bayes classification
* Support vector machine
* K-means clustering

2.webpage:

* HTML
* CSS
* Java script
* Bootstrap

3.connection between them:

* PHP

**Sentiment analysis**

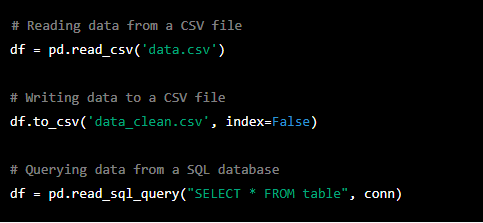
**Pandas:**

Pandas is a popular data analysis library in Python which plays major role in sentiment analysis

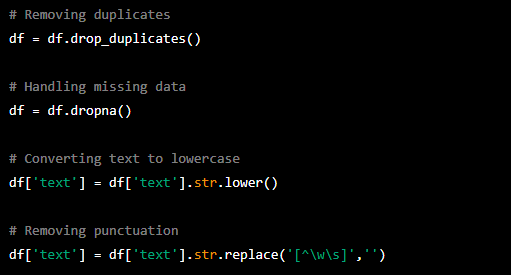
1.Pandas is a Python library that provides easy-to-use data structures and data analysis tools.



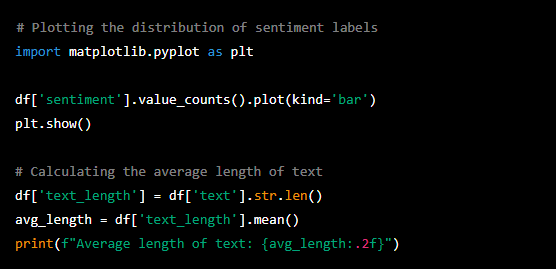
1. Pandas can be used for reading, writing, and manipulating structured data, such as CSV, Excel, and SQL databases.



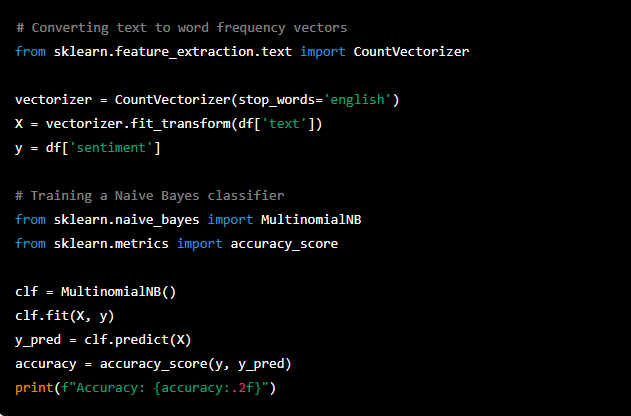
3.Also we can use to remove the duplicates,handling missing data,to remove the punctuations



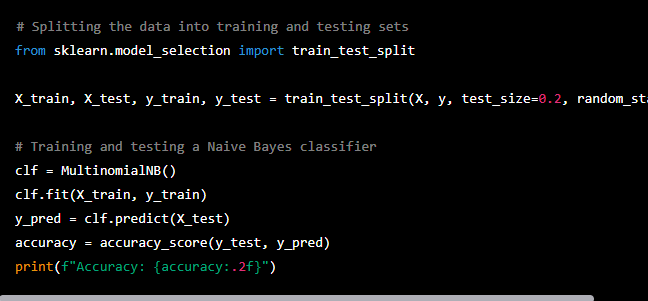
4.Pandas can be used for exploratory data analysis, such as data visualization and statistical analysis, to gain insights into the data before sentiment analysis.



1. Pandas can be used for sentiment analysis by converting the text data into numerical representations, such as word frequency or TF-IDF (term frequency-inverse document frequency), which can be fed into machine learning algorithms for classification.



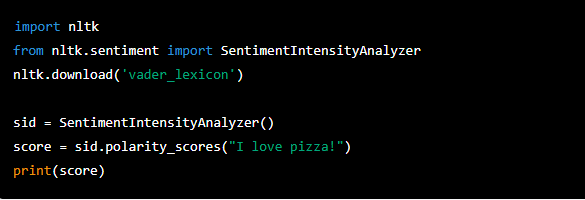
1. Finally, Pandas can be used for evaluating the performance of sentiment analysis models by comparing the predicted results with the actual results.



NLTK/VADER:

NLTK (Natural Language Toolkit) is a popular Python library used for natural language processing tasks, including sentiment analysis. In sentiment analysis, the goal is to determine the sentiment or emotion expressed in a given text. Here are some examples of how NLTK can be used for sentiment analysis in short sentences:

Text: "I love pizza!" Sentiment: Positive Code:



Output: {'neg': 0.0, 'neu': 0.0, 'pos': 1.0, 'compound': 0.6369}

In this example, NLTK's SentimentIntensityAnalyzer is used to analyze the sentiment of the sentence "I love pizza!". The output shows a positive sentiment score of 1.0, indicating a highly positive sentiment.

**NAIVE BAYES CLASSIFICATION:**

* Naive Bayes is a popular machine learning algorithm used for sentiment analysis tasks. It's based on Bayes' theorem, which calculates the probability of an event occurring based on prior knowledge of conditions that might be related to the event.
* In sentimental analysis, the algorithm uses a set of labeled data (i.e., data with known sentiments) to train a model, which can then be used to classify the sentiment of new, unlabeled data. The algorithm assumes that the presence or absence of certain words or phrases in a text document can be used to determine the sentiment of the text.
* The "naive" aspect of Naive Bayes comes from the assumption that the presence of each word or feature in a document is independent of the presence of other words or features. This simplifies the calculations involved and makes the algorithm efficient and fast.
* To use Naive Bayes for sentiment analysis, the algorithm first calculates the probability of a document belonging to each sentiment category (e.g., positive or negative) based on the occurrence of specific words or phrases in the document. It then assigns the document to the sentiment category with the highest probability.

Bayes' theorem is expressed mathematically as:

* P(Class | Features) = P(Features | Class) \* P(Class) / P(Features)

**where:**

1. P(Class | Features) is the posterior probability of the document belonging to a particular class given the observed features.
2. P(Features | Class) is the likelihood of observing the features given the class.
3. P(Class) is the prior probability of the class.
4. P(Features) is the probability of observing the features in the document.

In sentiment analysis, the features are typically the words or phrases present in the document, and the classes are the different sentiment categories (e.g., positive, negative, neutral).To classify a new document, the algorithm calculates the posterior probability of the document belonging to each class and assigns the document to the class with the highest probability.

# Naive bayes Algorithem How it will works:

* In sentiment analysis, the Naive Bayes algorithm works by using a set of labeled data to train a model that can predict the sentiment of new, unlabeled data.
* Once the model is trained, it can be used to predict the sentiment of new, unlabeled documents by calculating the probability of each sentiment category given the words or features present in the document. The document is then classified as belonging to the sentiment category with the highest probability.
* The Naive Bayes algorithm is particularly effective in sentiment analysis because it can handle large numbers of features (i.e., words or phrases) and can deal with sparse data, where many features are not present in a particular document. Additionally, the algorithm is relatively fast and can be trained on relatively small datasets.
* However, one limitation of Naive Bayes in sentiment analysis is that it assumes independence between features, which may not always be the case in natural language text. Nonetheless, the algorithm is widely used in practice and often serves as a baseline for more complex sentiment analysis models.

# Basic code of naive bayes algorithm:

import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('user\_data.csv')

x = dataset.iloc[:, [2, 3]].values

y = dataset.iloc[:, 4].values

# Splitting the dataset into the Training set

and Test set from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.25, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler sc = StandardScaler()

x\_train = sc.fit\_transform(x\_train)

x\_test = sc.transform(x\_test)

# Fitting Naive Bayes to the Training set

from sklearn.naive\_bayes import GaussianNB

classifier = GaussianNB()

classifier.fit(x\_train, y\_train)

# Predicting the Test set results

y\_pred = classifier.predict(x\_test)

# Making the Confusion Matrix from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)

# Visualising the Training set results

from matplotlib.colors import ListedColormap

x\_set, y\_set = x\_train, y\_train

X1, X2 = nm.meshgrid(nm.arange(start = x\_set[:, 0].min() - 1,

stop = x\_set[:, 0].max() + 1, step = 0.01), nm.arange(start = x\_set[:, 1].min() - 1, stop = x\_set[:, 1].max() + 1, step = 0.01)) mtp.contourf(X1, X2, classifier.predict(nm.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape), alpha = 0.75, cmap = ListedColormap(('purple', 'green'))) mtp.xlim(X1.min(), X1.max()) mtp.ylim(X2.min(), X2.max())

# Visualising the Test set results

from matplotlib.colors import ListedColormap x\_set, y\_set = x\_test, y\_test X1, X2 = nm.meshgrid(nm.arange(start = x\_set[:, 0].min() - 1, stop = x\_set[:, 0].max() + 1, step = 0.01), nm.arange(start = x\_set[:, 1].min() - 1, stop = x\_set[:, 1].max() + 1, step = 0.01)) mtp.contourf(X1, X2, classifier.predict(nm.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape), alpha = 0.75, cmap = ListedColormap(('purple', 'green'))) mtp.xlim(X1.min(), X1.max()) mtp.ylim(X2.min(), X2.max()) for i, j in enumerate(nm.unique(y\_set)): mtp.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1], c = ListedColormap(('purple', 'green'))(i), label = j) mtp.title('Naive Bayes (test set)') mtp.xlabel('Age') mtp.ylabel('Estimated Salary') mtp.legend() mtp.show() for i, j in enumerate(nm.unique(y\_set)): mtp.scatter(x\_set[y\_set == j, 0], x\_set[y\_set == j, 1], c = ListedColormap(('purple', 'green'))(i), label = j)

mtp.title('Naive Bayes (Training set)')

mtp.xlabel('Age')

mtp.ylabel('Estimated Salary')

mtp.legend()

mtp.show()

#### Advantages:

* Simple and easy to implement
* Fast and efficient
* Handles high-dimensional data:
* Robust to noise and irrelevant features:

# Disadvantages:

* Limited expressiveness:
* Assumption of feature independence
* Limited ability to handle out-of-vocabulary words:

# SVM(SUPPORT VECTOR MACHINE)

* Support Vector Machine (SVM) is a powerful machine learning algorithm that is commonly used for classification tasks, including sentiment analysis. The goal of SVM is to find the best hyperplane that can separate data points from different classes with maximum margin. In the context of sentiment analysis, SVM is used to classify text data into positive, negative, or neutral sentiment classes.
* The basic idea of SVM is to represent the text data as feature vectors in a high-dimensional space. Each feature corresponds to a word or a group of words in the text, and the value of the feature represents the importance or frequency of that word in the text. The SVM algorithm then tries to find the hyperplane that can separate the positive and negative sentiment classes with maximum margin. The margin is the distance between the hyperplane and the closest data points from each class.

#### HOW IT WILL BE WORKS:

* Support Vector Machine (SVM) algorithm works in sentiment analysis by using a trained model to classify text data into positive, negative, or neutral sentiment classes. The SVM algorithm works by representing the text data as feature vectors in a high-dimensional space. Each feature corresponds to a word or a group of words in the text, and the value of the feature represents the importance or frequency of that word in the text.
* The SVM algorithm then tries to find the best hyperplane that can separate the positive and negative sentiment classes with maximum margin. The margin is the distance between the hyperplane and the closest data points from each class. The SVM algorithm tries to maximize the margin while minimizing the misclassification error.
* In order to train an SVM model for sentiment analysis, a dataset of labeled text data is needed. The dataset is divided into a training set and a validation set. The SVM algorithm then uses the training set to learn the optimal hyperplane that separates the positive and negative sentiment classes with maximum margin. The validation set is used to test the performance of the trained model and adjust the hyperparameters if necessary.
* Once the SVM model is trained, it can be used to classify new text data into positive, negative, or neutral sentiment classes. The text data is first converted into a feature vector using the same feature extraction method that was used during training. The SVM algorithm then uses the learned hyperplane to classify the feature vector into one of the sentiment classes.
* However, SVM can be computationally expensive, particularly for large datasets or high-dimensional feature spaces. Additionally, SVM requires careful selection of hyperparameters, such as the kernel function and regularization parameter, which can significantly affect the performance of the algorithm. Finally, SVM can be sensitive to imbalanced classes in the training data, where one sentiment category may have significantly more examples than the others. This can result in biased sentiment analysis results.

#### BASIC CODE OF SENTIMENTAL ANALYSIS

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn import svm

from sklearn.metrics import classification\_report

# Define the training and testing data

train\_data = ["This is a positive review", "I loved this movie", "The acting was excellent", "I didn't like this book"]

train\_labels = [1, 1, 1, 0] # 1: positive, 0: negative

test\_data = ["This is a negative review", "The movie was boring", "The book was great"]

test\_labels = [0, 0, 1]

# Convert text data to numerical features using TF-IDF vectorizer

vectorizer = TfidfVectorizer()

train\_features = vectorizer.fit\_transform(train\_data)

test\_features = vectorizer.transform(test\_data)

# Create an SVM classifier and fit the training data

svm\_classifier = svm.SVC(kernel='linear')

svm\_classifier.fit(train\_features, train\_labels)

# Use the trained SVM classifier to predict the sentiment of the testing data

predicted\_labels = svm\_classifier.predict(test\_features)

# Evaluate the performance of the SVM classifier using classification report

print(classification\_report(test\_labels, predicted\_labels))

#### ADVANTAGES:

* Effective in high-dimensional space
* Robust to noise and irrelevant features
* Can handle non-linear decision boundaries
* Interpretable

# DISADVANTAGES:

* High computational requirements
* Requires careful tuning of hyperparameters
* Sensitivity to imbalanced classes

**K-means clustering algorithm:**

1. K-means clustering is an unsupervised learning algorithms ,which group the unlabeled datasets into different clusters.
2. Algorithms:
3. select the no.of k to decide the no.of clusters.
4. Select random k points or centroids.
5. Assign each data point to their closest centriod.
6. Calculate the variance and place a new centroid of each cluster.
7. Repeat the third steps , which means reassign each datapoint to the new closest centroid of each cluster.
8. If any reassignment occurs then goto step 4 else go to finish.
9. The model is ready.

Here we are going to use 5 clusters that is:

* 0-1→ very poor
* 1-2→poor
* 2-3 → average
* 3-4→ Good
* 4-5 → Excellent

**Webpage:**

**HTML(Hyper Text Markup Language):**

* An HTML element is defined by a start tag, some content, and an end tag <tagname> Content goes here... </tagname>.
* All HTML documents must start with a document type declaration: **<!DOCTYPE html>**.
* The HTML document itself begins with <html> and ends with </html>.
* The visible part of the HTML document is between <body> and </body>.
* HTML tags are not case sensitive: <P> means the same as <p>.

**# HTML HEADINGS:**

HTML headings are defined with the <h1> to <h6> tags.<h1> defines the most important heading. <h6> defines the least important heading

**EX:**<h1>This is heading 1</h1>To<h6>This is heading 6</h6>

**# HTML PARAGRAPHS:**

HTML paragraphs are defined with the <p>tag.Browsers automatically add a single blank line before and after each <p> element.

**EX:**<p>This is a paragraph.</p>

**# HTML LINKS:**

These are defined with the <a> tag.The link's destination is specified in the href attribute.Attributes are used to provide additional information about HTML elements

**EX:**<a href="https://www.w3schools.com">This is a link</a>

**# HTML IMAGES:**

These are defined with the <img> tag.The source file (src), alternative text (alt), width, and height are provided as attributes.

**EX:**<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142">

**# HTML ATTRIBUTES:**

Attributes usually come in name/value pairs like:name="value".These are always specified in the start tag only.It provides additional information about elements.

**EX:**src,href,title,lang,style,alt,width,height, etc

**# HTML TITLE PAGE:**

Every web page should have a page title to describe the meaning of the page.The <title> element adds a title to your page It defines the title in the browser toolbar

**EX: <title>HTML Tutorial</title>**

**# HTML TABLES:**

The HTML tables are created using the <table> tag in which the <tr> tag is used to create table rows and <td> tag is used to create data cells.<th> tag is used to create table headings.There are two attributes called *cellpadding* and *cellspacing* which you will use to adjust the white space in your table cells. The cellspacing attribute defines space between table cells, while cellpadding represents the distance between cell borders and the content within a cell and colspan attribute is to merge two or more columns into a single column and rowspan is to merge two or more rows

EX: <table border = "1">

<tr>

<td>Row 1, Column 1</td>

<td>Row 1, Column 2</td>

</tr>

<tr>

<td>Row 2, Column 1</td>

<td>Row 2, Column 2</td>

</tr>

</table>

**# HTML IFRAMES:**

You can define an inline frame with HTML tag <iframe>. The <iframe> tag is not somehow related to <frameset> tag, instead, it can appear anywhere in your document. The <iframe> tag defines a rectangular region within the document in which the browser can display a separate document, including scrollbars and borders. An inline frame is used to embed another document within the current HTML document.The src attribute is used to specify the URL of the document that occupies the inline frames

**EX:<iframe src = "/html/menu.htm" width = "555" height = "200">**

**Sorry your browser does not support inline frames.**

**</iframe>**

**# HTML LAYOUT ELEMENTS:**

HTML has several semantic elements that define the different parts of a web page:

|  |  |
| --- | --- |
| HTML5 Semantic Elements | * <header> - Defines a header for a document or a section * <nav> - Defines a set of navigation links * <section> - Defines a section in a document * <article> - Defines an independent, self-contained content * <aside> - Defines content aside from the content (like a sidebar) * <footer> - Defines a footer for a document or a section * <details> - Defines additional details that the user can open and close on demand |

**# HTML FORMS:**

The HTML <form> element is used to create an HTML form for user input. For example, during user registration you would like to collect information such as name, email address, credit card, etc. A form will take input from the site visitor and then will post it to a back-end application

**EX:<form >**

**User ID : <input type = "text" name = "user\_id" />**

**<br>**

**Password: <input type = "password" name = "password" />**

**</form>**

**# BUTTON CONTROLS:**

there are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to button. The type attribute can take the following values −

* **SUBMIT:**this creates a button that automatically submits a form.
* **RESET:**This creates a button that automatically resets form controls to their initial values.
* **BUTTON:**This creates a button that is used to trigger a client-side script when the user clicks that button.
* **IMAGE:** This creates a clickable button but we can use an image as the background of the button.

**# HTML MARQUEES:**

An HTML marquee is a scrolling piece of text displayed either horizontally across or vertically down your webpage depending on the settings. This is created by using the HTML <marquees> tag.

**EX:<marquee direction = "up">T**his text will scroll from bottom to up**</marquee>**

**# HTML LISTS:**

HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. Lists may contain −

**<ul>** − An unordered list. This will list items using plain bullets.

**<ol>** − An ordered list. This will use different schemes of numbers to list your items.

**<dl>** − A definition list. This arranges your items in the same way as they are arranged in a dictionary**.**

**# HTML FORMATTING ELEMENTS:**

Formatting elements were designed to display special types of text:

* <b> - Bold text
* <strong> - Important text
* <i> - Italic text
* <em> - Emphasized text
* <mark> - Marked text
* <small> - Smaller text
* <del> - Deleted text
* <ins> - Inserted text
* **<sub> - Subscript text**
* **<sup> - Superscript text**

**CSS(CASCADING STYLE SHEETS):**

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files
* CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document
* It handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.
* CSS can be added to HTML documents in 3 ways:

**Inline** - by using the style attribute inside HTML elements

**Internal** - by using a <style> element in the <head> section

**External** - by using a <link> element to link to an external CSS file

**# INLINE CSS:**

An inline CSS is used to apply a unique style to a single HTML element.It uses the style attribute of an HTML element. Here is an example sets the text color of the <h1> element to blue, and the text color of the <p> element to red

**EX:<h1 style="color:blue;">A Blue Heading</h1>**

**<p style="color:red;">A red paragraph.</p>**

**# INTERNAL CSS:**

An internal CSS is used to define a style for a single HTML page.It is defined in the <head> section of an HTML page, within a <style> element.The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powder blue" background color

**EX:<head>**

**<style>**

**body {background-color: powderblue;}**

**h1 {color: blue;}**

**p {color: red;}**

**</style>**

**</head>**

**# EXTERNAL CSS:**

An external style sheet is used to define the style for many HTML pages.To use an external style sheet, add a link to it in the <head> section of each HTML page The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

EX:<head>

<link rel="stylesheet" href="styles.css">

</head>

**# CSS STYLES:**

BACKGROUND-COLOR:The CSS background-color property defines the background color for an HTML element.

EX:<body style="background-color:powderblue;">

TEXT-COLOR:The CSS color property defines the text color for an HTML element

EX:<p style="color:red;">This is a paragraph.</p>

TEXT-SIZE:font-size property defines the text size for an HTML element

EX:<p style="font-size:160%;">This is a paragraph.</p>

TEXT-ALIGNMENT:text-align property defines the horizontal text alignment for an HTML element

EX: <p style="text-align:center;">Centered paragraph.</p>

FONTS:The CSS font-family property defines the font to be used for an HTML element

EX:<p style="font-family:courier;">This is a paragraph.</p>

**JAVASCRIPT:**

* JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complementary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform**.**
* JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as a Javascript Programmer.
* Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.
* Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel

**# JAVASCRIPT FUNCTIONS:**

Before we use a function, we need to define it. The most common way to define a function in JavaScript is by using the function keyword, followed by a unique function name, a list of parameters (that might be empty), and a statement block surrounded by curly braces.

**EX:**<script type = "text/javascript">

<!--

function functionname(parameter-list) {

statements

}

//-->

</script>

**# JAVASCRIPT EVENTS:**

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

**ONCLICK EVENTS:**

This is the most frequently used event type which occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type

**EX:**<body>

<p>Click the following button and see result</p>

<form>

<input type = "button" onclick = "sayHello()" value = "Say Hello" />

</form>

</body>

**ONSUBMIT EVENTS:**

Onsubmit is an event that occurs when you try to submit a form. You can put your form validation against this event type

**EX:**<body>

<form method = "POST" action = "t.cgi" onsubmit = "return validate()">

.......

<input type = "submit" value = "Submit" />

</form>

</body>

**ON MOUSE OVER & ON MOUSE OUT:**

These two event types will help you create nice effects with images or even with text as well. The on mouseover event triggers when you bring your mouse over any element and the on mouseout triggers when you move your mouse out from that element

**EX:**<body>

<p>Bring your mouse inside the division to see the result:</p>

<div onmouseover = "over()" onmouseout = "out()">

<h2> This is inside the division </h2>

</div></body>

**PHP:**

PHP started out as a small open source project that evolved as more and more people found

out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Pre-processor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage
* dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL,
* Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module
* on the Unix side. The MySQL server, once started, executes even very complex queries
* with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4
* added support for Java and distributed object architectures (COM and CORBA), making
* n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

**Common Uses of PHP:**

PHP performs system functions, i.e. from files on a system it can create, open, read, write,

and close them. The other uses of PHP are:

* PHP can handle forms, i.e. gather data from files, save data to a file, thru email you
* can send data, return data to the user.
* You add, delete, modify elements within your database thru PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.

It can encrypt data.

**Characteristics of PHP:**

Five important characteristics make PHP's practical nature possible:

* Simplicity
* Efficiency
* Security
* Flexibility
* Familiarity

All PHP code must be included inside one of the three special markup tags ate are recognized

by the PHP Parser.

**Php Syntax:**

*<?php PHP code goes here ?>*

*<? PHP code goes here ?>*

*<script language="php"> PHP code goes here </script>*

***Php Variables:***

***PHP has a total of eight data types which we use to construct our variables:***

* Integers: are whole numbers, without a decimal point, like 4195.
* Doubles: are floating-point numbers, like 3.14159 or 49.1.
* Booleans: have only two possible values either true or false.
* NULL: is a special type that only has one value: NULL.
* Strings: are sequences of characters, like 'PHP supports string operations.'
* Arrays: are named and indexed collections of other values.
* Objects: are instances of programmer-defined classes, which can package up both
* other kinds of values and functions that are specific to the class.
* Resources: are special variables that hold references to resources external to PHP

(such as database connections)***.***

**PHP – The if Statement**

The if statement executes some code if one condition is true.

**Syntax:**

if (condition) {

code to be executed if condition is true;

}

The PHP switch Statement

Use the switch statement to select one of many blocks of code to be executed.

**Syntax:**

switch (n) {  
  case label1:  
    code to be executed if n=label1;  
    break;  
  case label2:  
    code to be executed if n=label2;  
    break;  
  case label3:  
    code to be executed if n=label3;  
    break;  
    ...  
  default:  
    code to be executed if n is different from all labels;  
}

Variables:

$name=”jaya”;

$age=25;

Php is a loosely typed language.

echo "type of data is".gettype($name)."<br>";

3 types of variables:-

a)Local variable

b)Global variable

c)Static variable

$page="first"; // global variable

function jk()

{

$password="jaya"; // local variable

echo $password;

global $page;

echo $page;

static $age=25;

$age++;

echo $age;

}

echo $password; //error

jk();

jk();

jk();

Constants:

1) const A="krishna";

echo A;

2) define("MESSAGE","cse people are good people");

echo MESSAGE;

Data types:

Primitive data type: integer, float, string, boolean

Compound data type: arrays, object,

Resource, null etc.,

Array’s:

Arrays are heterogeneous data types

$a=array(1,2,3.5,"hi");// indexed array's

echo $a[2];

$b=array("hi"=>1,"hello"=>2);// Associative array's

echo $b["hi"];

Objects:-

class A

{

}

A obj=new A(); obj is an object String’s:- " " or ' ' . (dot) => concatenation; String methods: strtolower($name); echo strtoupper($name); Form Handling: Form method is get use $\_GET[] Form method is post use $\_POST[]; GET POST 1) Send through URL 1) Send through Body 2) not secure 2)Secure 3) Sent small amount of data 3)Large data. Example:- index.html:

Top of Form

Name:   
Age:



Bottom of Form

jk.php:

A obj=new A(); obj is an object

String’s:-

" " or ' '

. (dot) => concatenation;

String methods:

strtolower($name);

echo strtoupper($name);

Form Handling:

Form method is get use $\_GET[]

Form method is post use $\_POST[];

**GET** **POST**

1) Send through URL 1) Send through Body

2) not secure 2)Secure

3) Sent small amount of data 3)Large data.

Example:-

index.html:

<form action="jk.php" method="post">

Name: <input type="text" name="n"><br>

Age:<input type="text" name="age"><br>

<input type="submit" value="submit" name="">

</form>

jk.php:

<body>

<?php

$name=$\_POST["n"];

$age=$\_POST["age"];echo "Welcome to Login page".$name;

?>

</body>

Control, looping statements with examples:

Control statements: If, if else, elseif, switch

Looping statements: For, foreach, while and do… while.

<!DOCTYPE html>

<html>

<body bgcolor="yellow">

<?php

$name=$\_POST["n"];

$age=$\_POST["age"];

echo "Welcome to Login page".$name;

if($age>19)

echo "valid age";

elseif ($age<14) {

echo "<14";

}else{

echo ">14 and <19";

}

switch ($age) {

case >14:

// code...

break;

default:

// code.. break;

}

for($i=0;$i<$age;$i++)

{

echo $i;

}

$c=array(1,2,3,4,5,6);

foreach($c as $k)

{

echo $k;

}

while ($age!=0) {

echo $age;

$age--;

}

do{

echo $age;

$age--;

}while($age!=0);

?>

</body>

</html>

PHP Introduction - need - variables - Data types - array's - Form

Controls - Practical: https://youtu.be/dcNDmCSjib4

PHP Advanced:

File Inclusion:

Writing one code (.html, .php) in another php file.

Include: error + remaining loc is executed

Require : error then stops the execution.

Adv:

Reusability

Updations are easy

<?php

include "header.php"; or include(“header.php”);

echo "Body of careers page"."<br>";

include "footer.php";

if($age>5)

include "moreage.php";

else

include "lessage.php";

?>

Inplace of include, you can use require to include the file. The

difference is: At the time of Error, include will show warning and

continue its execution whereas require will stop execution there itselfFile system:

Files which are available in the system.

Operations:

1) Open - fopen()

2) Read- character - fgetc(), line by line- fgets(), entire file- fread().

3) Write- fwrite(), fputs()

4) Update - a mode

5) Close - fclose().

Modes: r,r+,w,w+,a,a+

R+: read, write from starting point

W+: read, write (overwritten)

A+: read, write ( Added)

<?php

$fp=fopen("data.php", "r+");

echo fread($fp, 10)

echo fread($fp, filesize("data.php"));

echo fgets($fp);

while (!feof($fp)) {

echo fgetc($fp);

}

echo fwrite($fp, "Student");

fclose($fp);

$fp=fopen("data1.php", "w+");

echo fread($fp, 5);

echo fwrite($fp, "jaya krishna cse",8);

echo fputs($fp, "rgukt is good located at nuzvid",8);

fclose($fp);

$fp=fopen("data.php","a+");echo fread($fp, 4);

fwrite($fp, "end of the file");

echo fwrite($fp, "is a good branch");

fclose($fp);

?>

File inclusion and File system: https://youtu.be/evwGiIdHHqk

18-8-2022

Parsing of Directories:

Used to list out all the files in the directory - sub directory - sub sub

directory etc..

Some of the useful predefined functions are:

mkdir(): create directory

copy(): source to destination

file\_exists(): file/directory

is\_dir(): to check whether the directory or not

is\_file():to check whether the file or not

scandir(): to list out all the files, directories in the given directory and

returns in the form of an array.

create a directory with the name cse3:

$d="cse3";

if(file\_exists($d))

{

echo "folder already exists";

}

else

{if(mkdir($d))

{

echo "created folder successfully";

}

else

{

echo "there is a problem in the creation";

}

echo "given input is not a file";

}

Copying the file data from one file to another:

$s="file.php";

$d="cse3/data.php";

if(file\_exists($s))

if(copy($s, $d))

echo "copy success";

else

echo "there is a problem in copying";

else

echo "file not exists";

list out all the files of the given directory:

// $d="cse3";

function filesCheck($d)

{

if(is\_dir($d))

{

$files=scandir($d) $files=array\_diff($files, array('.','..'));

foreach($files as $file)

{

if(is\_file("$d/$file"))

echo $file."<br>";

elseif (is\_dir("$d/$file")) {

// echo $file;

filesCheck("$d/$file");

}

}

}

else

{

echo "given input is not directory";

}

}

filesCheck("cse3");

File Uploading:

File: pdf/doc/csv/image/ etc.,

1) System directory:

Create a form:

Compulsory method should be post (as large amounts of data)

and sending data in the form of enctype="multipart/form-data".

While retrieving the file, specially use $\_FILE[‘file\_name’] to work

with the uploaded file and it will return an array.

Ex: $name=$\_FILES['fn'];

print\_r($name)

Output:- Array ( [name] => 2012-01-16-112.jpg [full\_path] => 2012-01-16-112.jpg [type] => image/jpeg [tmp\_name] => C:\xampp\tmp\phpC2FD.tmp [error] => 0 [size] => 361694 ).

To list out all values: foreach($name as $n) echo $n."  
"; To list out specific value like size of uploaded file: $size=$name['size']; echo $size; Or $size=$\_FILES['fn']['size']; echo $size; move\_uploaded\_file(source,destination): used to upload file Code to upload file in the upload folder: jk.php:

Top of Form



Bottom of Form

U.php: Output:-

Array ( [name] => 2012-01-16-112.jpg [full\_path] =>

2012-01-16-112.jpg [type] => image/jpeg [tmp\_name] =>

C:\xampp\tmp\phpC2FD.tmp [error] => 0 [size] => 361694 ).

To list out all values:

foreach($name as $n)

echo $n."<br>";

To list out specific value like size of uploaded file:

$size=$name['size'];

echo $size;

Or

$size=$\_FILES['fn']['size'];

echo $size;

move\_uploaded\_file(source,destination): used to upload file

Code to upload file in the upload folder:

jk.php:

<form action="u.php" method="post"

enctype="multipart/form-data">

<input type="file" name="fn"><br><br>

<input type="submit" name="sub" value="Upload file">

</form>

U.php:

<?php$tlocation=$\_FILES['fn']['tmp\_name']; // temporary file location

name

$name=$\_FILES['fn']['name']; //name of the file

$dlocation="upload/".$name;

if(move\_uploaded\_file($tlocation, $dlocation))

echo "file uploaded successfully";

else

echo "there is a problem in uploading";

?>

2) Database.

Parsing of directories and File Uploading:

https://youtu.be/sqDSjBZX-5g

20-8-2022

**File Downloading:**

Downloading of a file to the local system.

<a href="images/b.jpeg">Download image</a>

<a href="images/wt.pdf">Download wt file</a>

<a href="images/makedigitize.csv">Download CSV file</a>

Images or documents can't be downloaded directly in some

browsers. So, we need to download forcefully.

The .csv, .zip, .exe automatically downloaded.

readfile(): reads a file and writes it to the browser i.e., output bufferheader():- It sends a raw http to the client + a lot of options are there.

Among these one is content-disposition to save the data that you are

sending.

header("content-disposition:attachment;filename=".basename($n

ame).";");

Filename: the name given to the downloaded file.

basename(): Actual name of the document.

Cse.php:(Create a image gallery and providing download option)