

A Project based learning project report on

Topic:

**DATAGRAPH**

**Making life easier..**

SUBMITTED TO ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY’S

llINSTITUTE OF INFORMATION TECHNOLOGY

( AISSMS IOIT )

**IN**

DEPARTMENT OF ENGINEERING SCIENCES

FIRST YEAR AI-DS

BY THE TEAM MEMBERS :

1. KAVERI AHER (E-4)
2. ATHARVA BHOITE (E-12)
3. PRANITA BOBADE (E-15)
4. ARYA CHUNNE (E-20)
5. LAXMIKANT DESHPANDE (E-23)

UNDER THE GUIDANCE OF:

Subject teacher: Head of department

Dr. N.D. Gaikwad sir Dr. P. G. Musrif sir

**

Certificate

**This is to certify the project report entitled as:**

**“DATAGRAPH”**

**Making life easier…**

**Submitted by:**

**Miss. Kaveri Aher (22610004) Mr. Atharva Bhoite (22610012)**

**Miss. Pranita Bobade (22610015) Miss. Arya Chunne (22610020)**

**Mr. Laxmikant Deshpande (22610023)**

is the record of bonafide work carried out by them in fulfillment of the requirement for the Problem based learning (PBL) subject, as prescribed by the institute AISSMS IOIT (autonomous) under the guidance of Savitribai Phule Pune University in the Academic Year 2022-23 .

Subject teacher: Head of department

Dr. N.D. Gaikwad sir Dr. P.G. Musrif sir

**Acknowledgment**

It is my great pleasure in expressing sincere and deep gratitude towards my guide **Prof. Dr.N.D.Gaikwad** from Engineering Science Department for his valuable guidance and constant support throughout this work and help to peruse additional studies . We take this opportunity to thank Head of the Department **Dr.P.G.Musrif** and all staff members of department of Engineering Sciences, AISSMS IOIT, Pune, for cooperation provided by them in many ways. The motivation factor for this work was the inspiration given by our honorable principal **Dr. P.B.Mane**. Lastly I am thankful to those who have directly or indirectly supported for our work.

**Miss. Kaveri Aher Miss. Pranita bobade**

**Mr. Atharva Bhoite Miss.Arya Chunne**

**Mr. Laxmikant Deshpande**

**Department of Engineering Sciences**

**Course objective**

|  |  |
| --- | --- |
| 1. | To inculcate learning approach in Mini Project to develop replicable skill and to improve students attitudes towards learning. |
| 2. | To inculcate learning approach in Mini Project to develop replicable skill and to improve students attitudes towards learning. |

**Course outcomes**

|  |  |
| --- | --- |
| **1.** | To inculcate learning approach in Mini Project to develop replicable skill and to improve students attitudes towards learning. |
| **2.** | Identify projects relevant to societal needs and its solutions |

**INDEX**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE NO.** |
| **1.** | **ABSTRACT** | **6.** |
| **2.** | **INTRODUCTION** | **7.** |
| **3.** | **AIM AND OBJECTIVE** | **9-10** |
| **4.** | **PROBLEM STATEMENT** | **11** |
| **5.** | **LITERATURE SURVEY** | **12** |
| **6.** | **METHODOLOGY** | **13-14** |
| **7.** | **WORKING AND EXPLANATION** | **15-19** |
| **8.** | **SOFTWARE PROGRAMMING** | **20-26** |
| **9.** | **APPENDIX** | **27-28** |
| **10.** | **REFERENCES** | **29** |

**ABSRACT**

**EXCEL FILE TO GRAPHICAL REPRESENTATOR**

**Making life easier…**

**Our application aims to create a user-friendly platform that promotes simplifying work by letting our program to do the work by simply attaching the excel file you need to work on. It incorporates a variety of features to create different kind of graphs and charts with few clicks on our finger tips. These features encourage participants to customise and represent their data by simply putting their excel file into our application and get there charts and graphs in few milliseconds .**

**Also, the application influences attractive interfaces, adaptable designs, and personalized experiences to fulfill to the unique needs and preferences of each user. It takes into consideration factors such as rapidity, flexiblities, and technological familiarity, ensuring inclusivity and accessibility for all users.**

**The implementation of our application encourages inter-professional learning and mutual support. New users can get help with the help of how to use guide Simultaneously, new users can use technological guidance, contemporary insights, and a fresh perspective, by watching the reference youtube video linked in the help menu. You just have to search for the problem you are facing and can get its solution without even leaving the application within the same platform.**

**In conclusion , our application make the task of converting data into graphs and chart easy and instant . Only by putting a ready excel file to the application can perform the task by choosing the requirements .**

**INTRODUCTION**

**When it comes to making charts in colleges and schools , where some teachers and students are not aware of technological things , it is difficult for them to complete the task by learning the features from the scartch . This is not possible to everyone and that even in there busy schedule . in order to solve this problem we have created an application where they just have to input there excel file in our input bar of the application and by just selecting there preferances a completed representation is shown on your screen .**

**With DATAGRAPH, we aim to break down barriers and stereotypes that often divide generations. Whether you're a tech-savvy youngster or a seasoned individual with a wealth of life experiences, this application offers a unique platform where everyone can thrive and engage in fruitful results.**

**This application is based on programs done with the help of matplot library and Pandas library in Python programming language . It is a proper website containing frontend as well as backend . Frontend contains the pages that we see when on our screen when we open an application . Such as the login page , intro page , search bar , contents , intro , etc. The backend includes all the work required to complete the tasks , the programs and the calculations that are performed .**

**For example , In a Calculator we put numbers and values of which we want calculations ,the values input appears on the screen , this is the frontend part of the calculator. And the calculations that took part inside the calculator by which we got the expected answers , all that procedure is backend of the calculator .**

**Through this app, we encourage our users to tap into the wisdom and knowledge and save there time with much efforts taken to complete there tasks .**

**The programming languages used to prepare the frontend includes HTML (**HTML is the standard markup language for creating Web pages) **and CSS (CSS is the language we use to style an HTML document.CSS describes how HTML elements should be displayed.). The programming language used in building the backend includes**

**AIM AND OBJECTIVE**

**Aim:**

**The aim of the interactive application is to engage users by providing an enjoyable and meaningful experience that caters to their specific needs and tasks. The application should promote intergenerational interaction, enhance productivity, and offer helpful or educational content.**

**Objectives:**

**User-Friendly Interface:**

**Develop an intuitive and easy-to-use interface that accommodates the varying technological proficiency of both old and young users.**

**Ensure clear and accessible navigation to facilitate smooth interaction with the application.**

**Age-Appropriate Content:**

**Create a diverse range of content suitable for different age groups, considering their technical abilities, needs, and preferences.**

**Offer interactive videos, and educational videos that help the user to overcome the difficulties while using the application, and adaptive to the user's skill level.**

**Interactivity and Collaboration:**

**Incorporate features that includes help desk and reference videos to complete the task without any interuptions .**

**Provide opportunities for old and young users to share experiences, knowledge, and achievements within the application by using this application to get work done of people which helps our user to earn a little as well .**

**Customization and Personalization:**

**Allow users to personalize their representation by selecting preferences, or themes that resonate with their individual tastes and identities.**

**Provide options to adjust colour selection, font sizes, and background settings to accommodate varying needs .**

**Continuous Improvement:**

**Gather user feedback and conduct regular assessments to identify areas for improvement and enhancement in the application.**

**Continuously update and expand the content and features based on user preferences and emerging trends in interactive technology.**

**Accessibility:**

**Ensure the application is accessible to users with less technical awareness and easy to them .**

**Provide support for screen readers, alternative input methods, and other assistive technologies to enable a wider range of users to engage with the application.**

**By aligning the aim and objectives of the application towards engaging both old and young users, the application can create a positive and inclusive user experience that caters to the needs of different age groups and promotes intergenerational interaction and learning.**

**PROBLEM STATEMENT**

**There is a need for a data visualization website that allows users to easily convert Excel and CSV files into visually appealing and interactive graphs. The current options available for data visualization either lack user-friendliness or require extensive technical knowledge, limiting accessibility for non-technical users. Consequently, individuals and organizations face challenges in effectively analyzing and presenting data in a visually compelling manner, hindering decision-making processes and communication of insights.**

**The desired solution should provide an intuitive and user-friendly platform where users can effortlessly upload their Excel or CSV files and transform them into various types of graphs and charts. The system should have robust data processing capabilities to handle large datasets efficiently, ensuring accurate and timely visualizations. Additionally, it should offer customization options, enabling users to modify graph layouts, color schemes, labels, and other visual elements to suit their specific needs and preferences.**

**By providing a streamlined and accessible tool for data visualization, this website will empower users to effectively communicate their data-driven insights, improve decision-making processes, and enhance the overall understanding of complex information through visual representations.**

**LITERATURE SURVEY**

Graph visualization plays a crucial role in understanding and analysing complex data structures. In the realm of Python programming, the matplotlib library offers a comprehensive solution for creating visually appealing and informative graphs.

With matplotlib, developers can effortlessly generate a wide range of graphs, from basic line and bar graphs to advanced scatter plots and network diagrams. The library's versatility, coupled with its intuitive interface, makes it an ideal choice for researchers, data scientists, and programmers alike.

By leveraging matplotlib's rich set of features, users can customize every aspect of their visualizations, including colours, labels, titles, axes, and legends, enabling them to present their data in a visually compelling and insightful manner. Whether it's for exploratory data analysis, scientific research, or data-driven presentations, matplotlib's graph visualization capabilities empower users to effectively communicate complex information with ease.

**METHODOLOGY**

**Program Description:**

The data visualization program leverages the Pandas library, which is widely used for data manipulation and analysis. It provides a convenient interface for loading and processing data from Excel sheets. The program takes a link to an Excel sheet as input and performs the following steps:

a. **Data Extraction:**

The program uses Pandas' read\_excel() and read\_csv() function to read the data from the Excel sheet and store it in a Pandas DataFrame. This allows for efficient data handling and manipulation.

b. **Data Preparation**: Once the data is loaded, the program performs necessary preprocessing steps, such as data cleaning, filtering, and transformation. Pandas' functionalities, such as data cleaning methods (dropna(), fillna()) and data manipulation operations (groupby(), pivot\_table(), etc.), are utilized for this purpose.

c**. Data Visualization**: After preparing the data, the program employs Pandas' built-in plotting capabilities to generate charts and graphs. It utilizes the plot() method with various parameters, such as the type of plot (line, bar, scatter, etc.), x-axis and y-axis data, and optional styling options. These plots provide visual representations of the data, making it easier to identify trends, patterns, and insights.

d. **Output Generation:** The program saves the generated charts and graphs as image files or presents them interactively using a graphical user interface (GUI) for further analysis or sharing with stakeholders.

**Program Benefits:**

The data visualization program developed by our teammates using Pandas offers several advantages:

a. **Easy Data Loading**: The program simplifies the process of loading data from Excel sheets by utilizing Pandas' read\_excel() function. This eliminates the need for manual data extraction and formatting.

b. **Data Cleaning and Manipulation:** The program leverages Pandas' powerful data cleaning and manipulation capabilities to ensure that the visualizations are based on accurate and relevant data. This includes handling missing values, filtering data, and transforming data structures.

c. **Quick and Interactive Visualizations**: With Pandas' built-in plotting functionalities, the program allows for the rapid generation of various charts and graphs. These visualizations facilitate quick data exploration and aid in identifying patterns, correlations, and outliers.

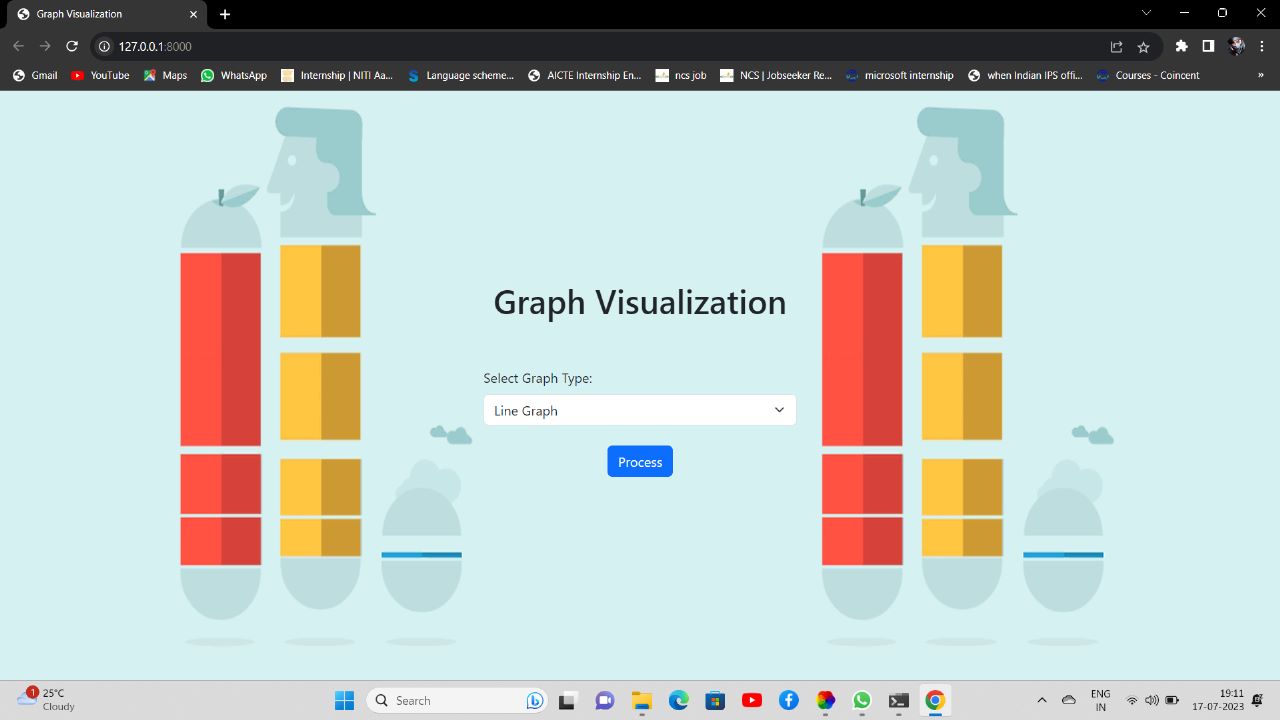
d. **Customization and Flexibility:** While Pandas provides convenient default settings for plotting, the program can also be extended to include customizations using Matplotlib, which is the underlying library behind Pandas' plotting capabilities. This flexibility enables the program to cater to specific visualization requirements.

**WORKING AND EXPLANATION**

**WORKING OF THE WEBSITE :**

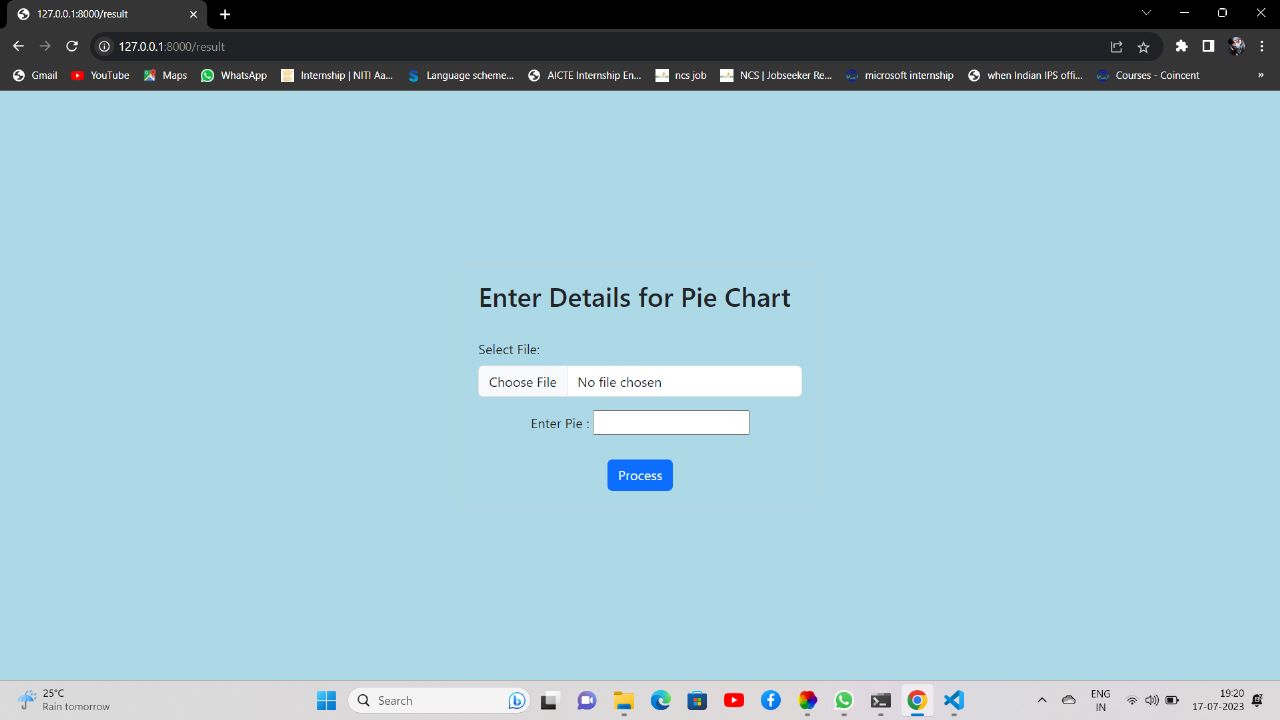
The actual working of the website is in such way step by step working of the website is given below:

**1.**



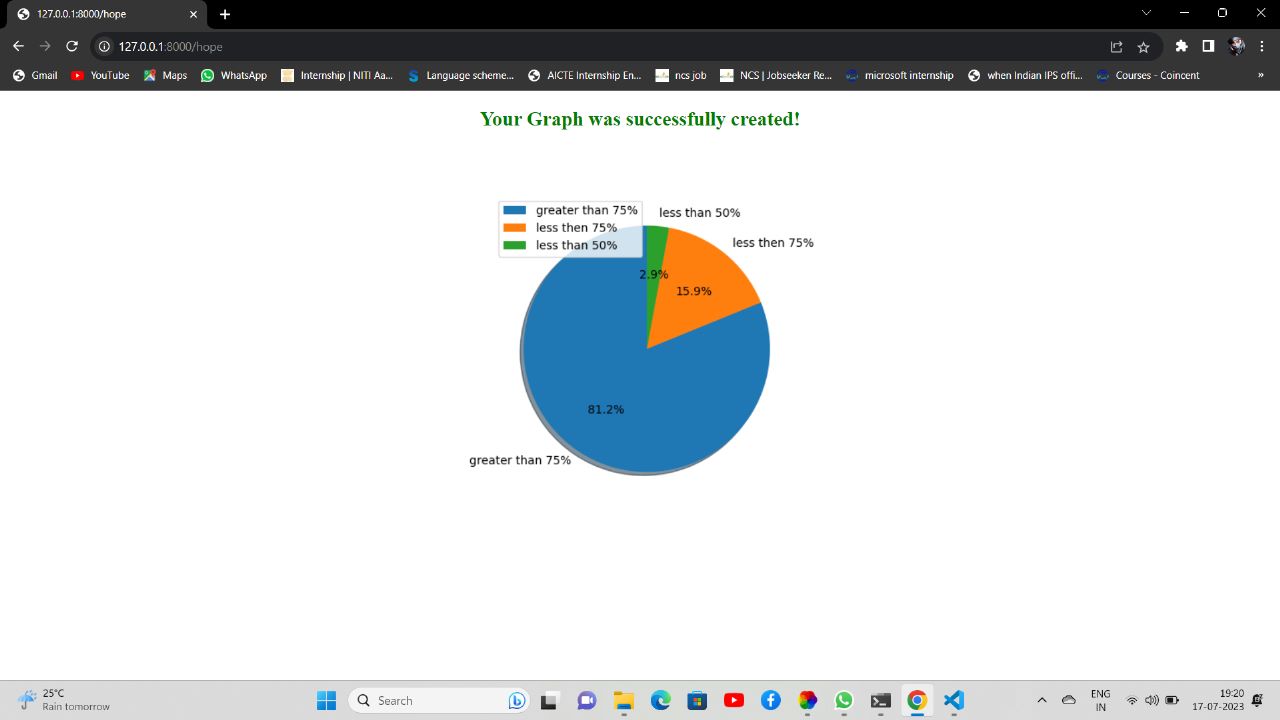
In this step we have to choose the kind of graph we want.

**2.**

****

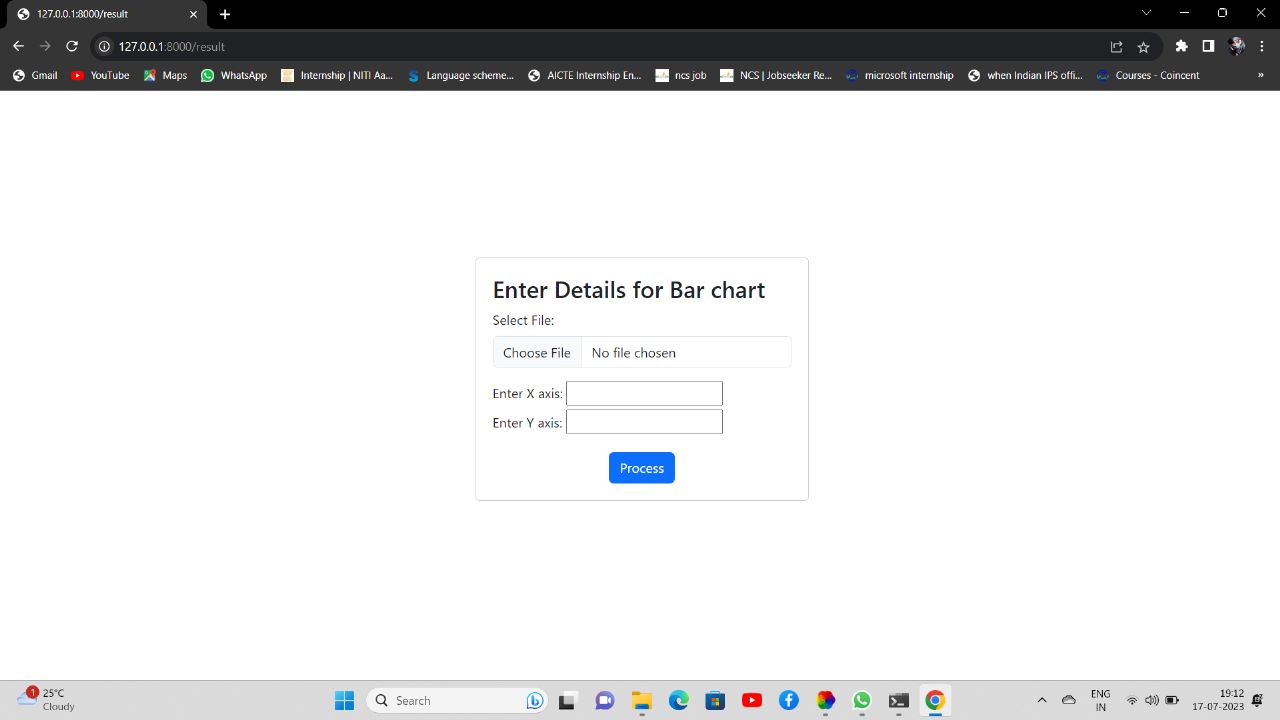
Second step is searching the graph option.

**3.**



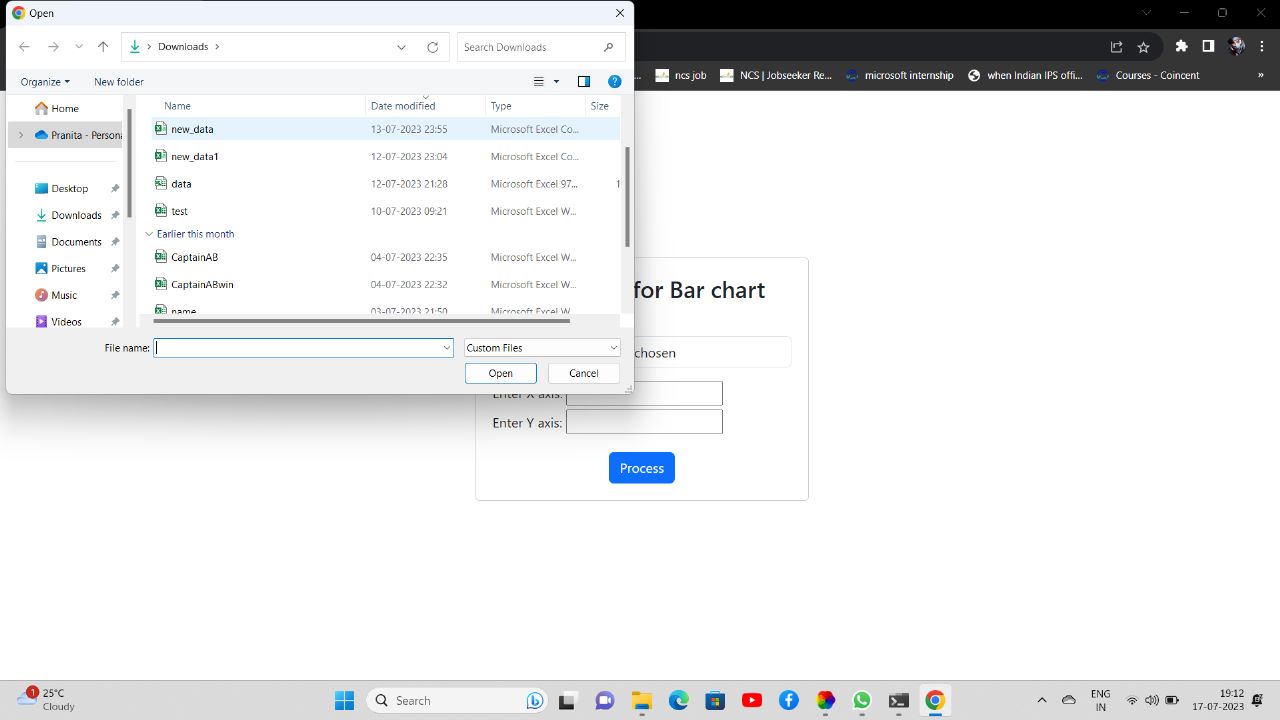
In third step the options occur one the page.

**4.**

****

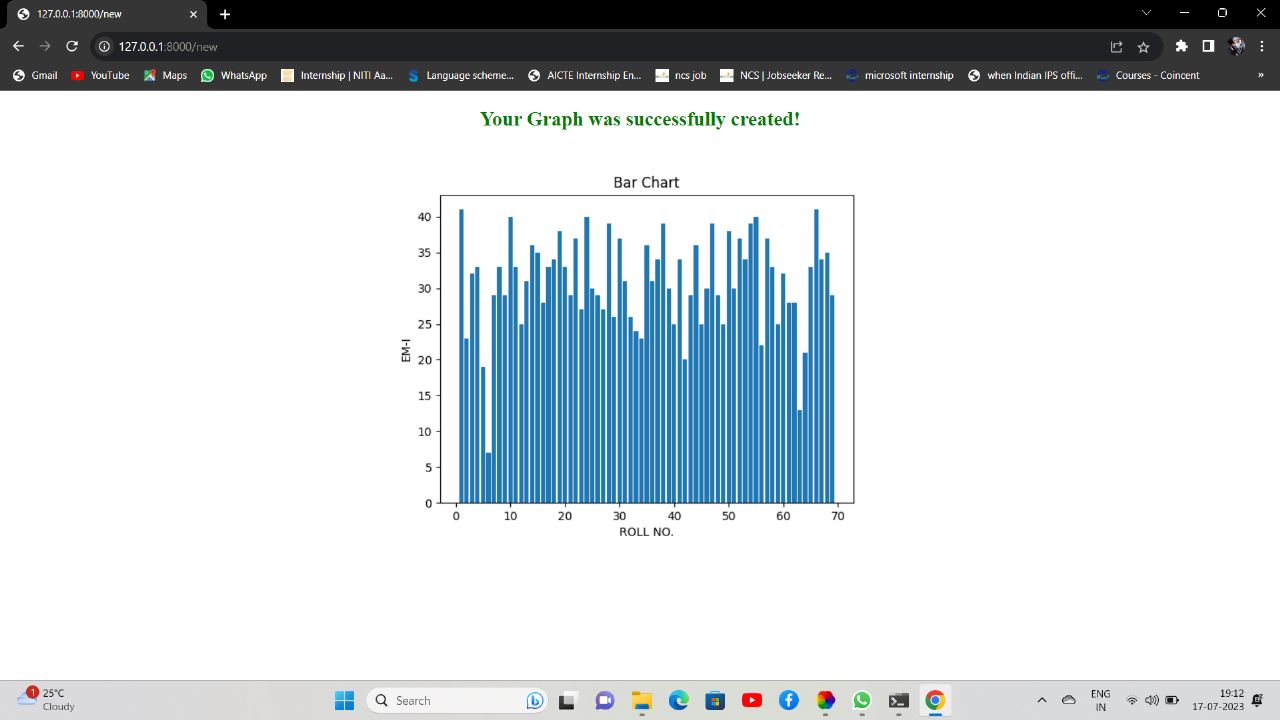
In fourth step we chose the type of graph we want and now moving forward to choosing file either CSV OR EXCEL.

5.



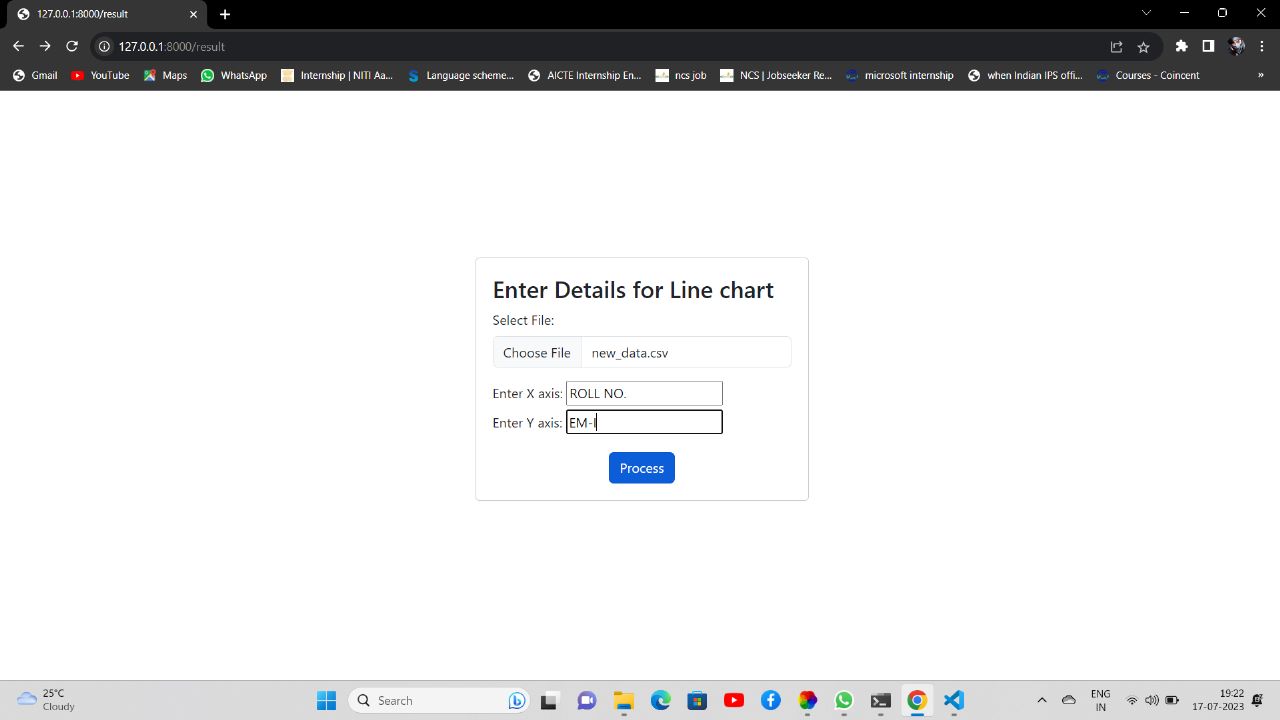
Then in the fifth step choose the file and open it.

6.



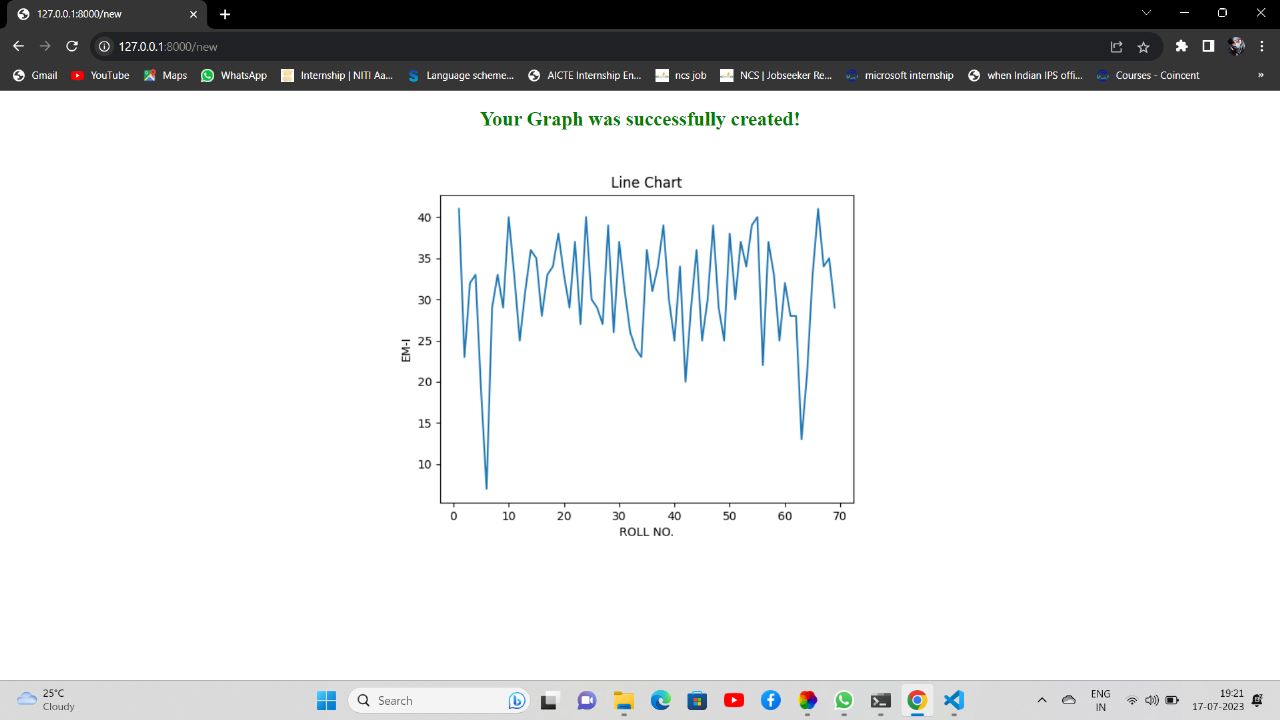
Choose the file and open it.

7.



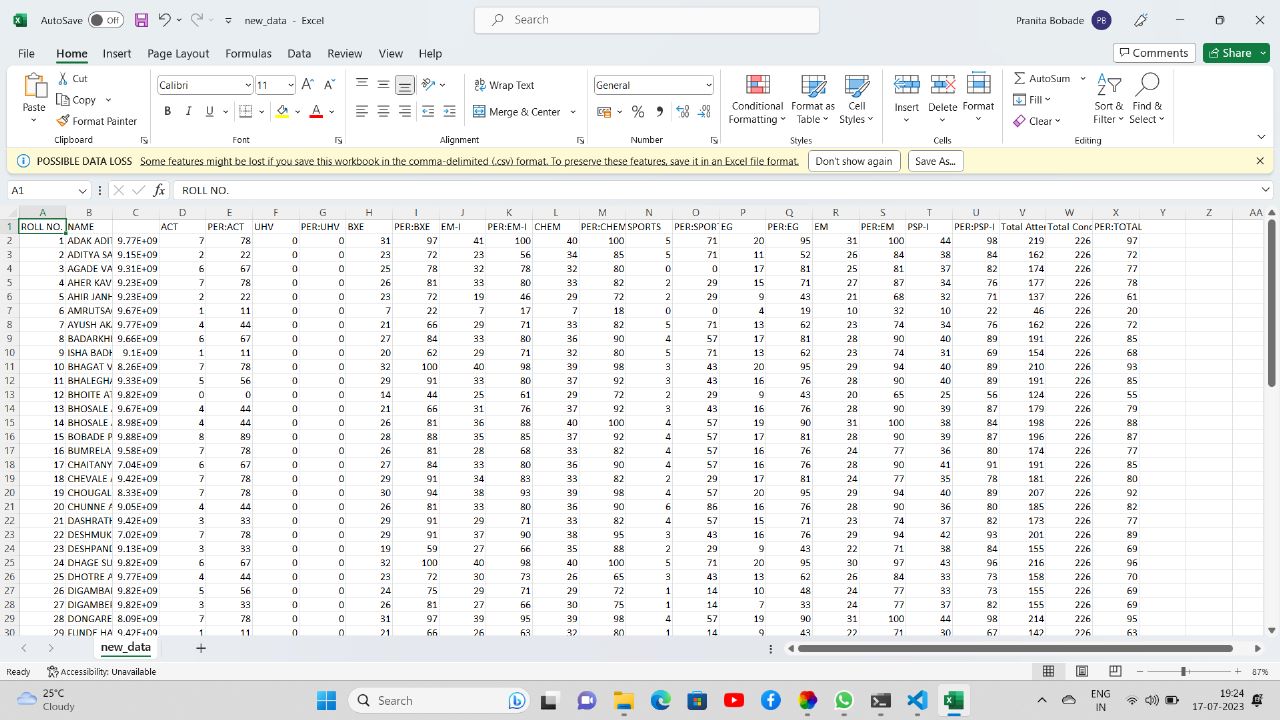
After selecting open the file.

8.



Click to proceed to go forward and to get the required graph.

This is the sample file that we used to take data from:



**SOFTWARE PROGRAMMING**

**The program code to the frontend our Website is as follows :**

<!DOCTYPE html>

<html>

<head>

<title>Graph Visualization</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">

<style>

body { background-image: url("C:\Users\LAXMIKANT\Downloads\bcg.webp");

size: 100px ;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

</style>

</head>

<body >

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

<link

rel="stylesheet"

href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"

/>

</body>

<form method='post' action='result' enctype='multipart/form-data' autocomplete="off">

<div class="container">

<h1 class="text-center"> Graph Visualization </h1><br><br>

</div>

<div class="mb-3">

<label for="graphType" class="form-label"> Select Graph Type: </label>

<select id="graphType" name='type\_g' class="form-select">

<option value="line" name='line'> Line Graph </option>

<option value="bar" name='bar'> Bar Graph </option>

<option value="pie" name='pie'> Pie Chart </option>

</select>

</div>

<div class="mb-3">

<label for="fileInput" class="form-label"> Select File: </label>

<input type="file" id="fileInput" accept=".csv, .xlsx, .xls" class="form-control" name='file'>

</div><br>

<div class="mb-3">

<label for="X" class="form-label" style="margin-left: 20px;"> Enter X axis: </label>

<input type='text' name='X-axis'>

<label for="Y" class="form-label"> Enter Y axis: </label>

<input type='text' name='Y-axis'>

</div>

<center>

<div>

<label for="pie" class="form-label"> Enter Pie : </label>

<input type='text' name='pie\_var'><br><br>

</div>

</center>

<center><input type='submit' value='Process' class="btn btn-primary"></center>

</form>

</form>

<script>

{% if error %}

alert('{{ error }}')

{% endif %}

</script>

</body>

</html>

**The working of this code can be explained as:**

This code represents an HTML form for graph visualization. Let's break down the different parts of the code:

The code begins with the HTML doctype declaration <!DOCTYPE html>, which specifies that this is an HTML document.

Inside the <head> element, there is a <title> tag that sets the title of the web page to "Graph Visualization".

The code includes a CSS file from the Bootstrap framework CDN (https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css) to style the page using Bootstrap classes.

There is an inline <style> block that defines some additional CSS styles for the <body> element. It sets the background image using the background-image property and centers the content on the page using flexbox properties.

Inside the <body> element, there is a reference to an external CSS file style.css, which presumably contains custom styles for the page. However, this reference appears to be misplaced and should ideally be placed within the <head> element.

The <form> element is used to create a form for user input. It has the attributes method='post' and action='result', indicating that the form data will be submitted to a server-side script located at the "result" endpoint.

The form is contained within a <div class="container"> element, which provides a responsive container for the form content.

There is an <h1> heading with the text "Graph Visualization" inside the container.

The form includes several form elements for user input:

A dropdown menu (<select>) with the id "graphType" and the name "type\_g", allowing the user to select the type of graph (line graph, bar graph, or pie chart) they want to visualize.

An <input type="file"> element with the id "fileInput" and the name "file" for selecting a file (accepting .csv, .xlsx, and .xls file types).

Two text inputs (<input type="text">) for entering the X and Y axis labels.

Another text input for entering the "pie" variable (presumably for the pie chart).

A submit button (<input type="submit">) with the value "Process" is included inside a centered <center> element.

The closing </form> tag is duplicated and should be removed.

Finally, there is a <script> block that checks for an "error" variable and displays an alert with the error message if it exists.

Overall, this code sets up a form for users to input data and select options for graph visualization, which can be submitted to a server-side script for further processing and rendering of the desired graph.

**Explanation for backend is as follows:**

We have used Django framework for backend programming in Python. Django is a popular web framework that is commonly used for backend development. It provides numerous features and tools for building robust and scalable web applications. Some key features of Django for backend development include:

1. **Model-View-Template (MVT) Architecture**: Django follows the MVC architectural pattern, which helps in separating the application logic, data models, and user interface.

2. **Object-Relational Mapping (ORM**): Django's ORM allows developers to interact with the database using Python objects, making database operations intuitive and efficient.

3. **URL Routing**: Django provides a powerful URL routing system that maps URLs to the appropriate views, allowing for easy navigation and request handling.

4. **Template Engine:** Django includes a template engine that allows developers to create dynamic and reusable templates, making it easier to generate HTML dynamically.

5. **Form Handling:** Django simplifies form handling by providing a form validation framework that helps with data input validation and processing.

6. **Admin Interface:** Django offers a fully-featured admin interface that allows developers to manage data and perform CRUD (Create, Read, Update, Delete) operations on models without writing custom code.

7. **Security:** Django includes various security features such as protection against common web vulnerabilities like cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection.

8**. Caching**: Django provides built-in support for caching to improve the performance of web applications by reducing the load on the database.

9. **Internationalization and Localization:** Django has excellent support for handling multiple languages and locales, making it easier to build applications for different regions and languages.

**APPENDIX**

The provided code is a starting point for creating a project that aims to visualize data from an Excel or CSV file using graphs. However, the code you shared only includes the front-end part, which is responsible for creating the form and capturing user input. To complete the project and achieve the desired functionality, you would need to implement the back-end logic to handle the form submission, read the data from the uploaded file, and generate the corresponding graphs.

Here's an overview of the steps you would typically need to follow to complete the project:

1. **Form Submission Handling:** Set up a server-side script (e.g., using a programming language like Python, Node.js, or PHP) to handle the form submission. This script will receive the form data, including the uploaded file and other inputs, and process it further.

2. **File Processing**: In the server-side script, use appropriate libraries or modules to read the data from the uploaded file (e.g., pandas in Python for CSV and Excel files). Extract the necessary data for graph visualization based on the user's input, such as the X-axis and Y-axis columns.

3. **Graph Generation:** Utilize a graphing library or module (such as Matplotlib or Chart.js) to generate the desired graphs based on the extracted data. Depending on the selected graph type (line graph, bar graph, or pie chart), use the appropriate functions and configurations to create the visualizations.

4. **Render and Display**: Once the graphs are generated, you can choose to display them on the same page or redirect the user to a new page showing the visualizations. This step involves integrating the graph images or interactive graphs into the HTML response sent back to the user's browser.

5**. Error Handling**: Implement appropriate error handling and validation throughout the process to handle cases such as invalid file formats, missing data, or any other errors that may occur during file processing or graph generation.

It's important to note that the code you provided is just the front-end part and does not include the complete implementation for processing the data and generating the graphs. The back-end implementation would require additional code to handle the form submission and perform the necessary data processing and graph generation steps.

Remember to choose the programming language, frameworks, and libraries that suit your project requirements and skillset. Additionally, ensure that you follow best practices for security and file handling when working with user-submitted files.

**REFERENCES**

* Iliev, I., & Dimitrov, G. P. (2014, May). Front end optimization methods and their effect. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (pp. 467-473). IEEE.

This paper gives us an idea about how html and CSS contributes in frontend in web-development.

* Ari, N., & Ustazhanov, M. (2014, September). Matplotlib in python. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)* (pp. 1-6). IEEE.

This paper gives whole knowkedge about the module matplotlib.

* Sadiku, M., Shadare, A. E., Musa, S. M., Akujuobi, C. M., & Perry, R. (2016). Data visualization. *International Journal of Engineering Research and Advanced Technology (IJERAT)*, *2*(12), 11-16.

This research paper gives the whole knowledge about the data visualization concept.

* McPhee, J. J. (1996). On the use of linear graph theory in multibody system dynamics. *Nonlinear Dynamics*, *9*, 73-90.

This research page explains the advantages and usefulness of line graphs in visualising data.

* Chen, D. Y. (2017). *Pandas for everyone: Python data analysis*. Addison-Wesley Professional.

This explains the use of Pandas library, how it introduces new data types and how it is useful in data analysis.

* https://support.microsoft.com/

Excel is an incredibly powerful tool for getting meaning out of vast amounts of data. But it also works really well for simple calculations and tracking almost any kind of information. The key for unlocking all that potential is the grid of cells. Cells can contain numbers, text, or formulas. You put data in your cells and group them in rows and columns. That allows you to add up your data, sort and filter it, put it in tables, and build great-looking charts. Let’s go through the basic steps to get you started.