**Logo

Description automatically generated**

**PROJECT REPORT ON**

**‘WEB BASED VISUALISATION**

**OF OIL AND GAS INDUSTRY**

**LAS 2.0 DATA’**

**TEAM MEMBERS**

**ARCHIT GARG**

**SHREYA VERMA**

**MUSKAN TYAGI**

**ROHAN RANA**

**DHRUV GOYAL**

**INTRODUCTION**

With the development of strategic oil and gas assets, massive spatiotemporal oil and gas data have been accumulated. Application systems that assist in the storage and management of the voluminous and complex oil and gas datasets are in high demand. The voluminous and various data should be leveraged and turned into information for business decision-making and operation assistance.

During this Summer Training 2021 at ONGC, the project which we worked upon was ‘Web-based Visualisation and full Interactivity of Oil and Gas Industry LAS Data.’

**What is Data Visualisation?**

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

In the world of Big Data, data visualization tools and technologies are essential to analyse massive amounts of information and make data-driven decisions.

**What is LAS Data?**

Log ASCII Standard, LAS, is a format specification for storing and distributing digital well log data. The standard was defined and is maintained by the Canadian Well Logging Society. LAS 2.0 is the most used digital well log distribution format.

The LAS 2.0 file specification defines basic sections that appear in the text file:

~VERSION – (required) LAS version, delimiter for data sections, and wrapping information

~WELL – (required) well name, location, and information on the start and stop depths, step interval between depths, and null value definitions

~CURVE – (required) list of well log curves included in the file

~PARAMETER – data on well elevation, casing depths, etc.

~OTHER –freeform section that includes comments

~ASCII – (required) digital log curve data

**AIM AND MOTIVATION**

**Aim:** To develop a web-based utility that will visualise the LAS 2.0 Data and provide full interactivity over that data as well.

**Motivation:**The explosion of oil and gas related data accompanies exploration and production expansion. The utilization of the large amount of data accumulated over years is a new challenge and critical for efficient and sustainable exploitation. Systems that assist in the storage and management of the voluminous and complex oil and gas datasets are in high demand.

The main motivation to work on this project and take up the ONGC Summer Training is to provide us with a platform to learn and grow and get hand-on experience about how a project is executed and worked upon.

/Log ASCII Standard (LAS) files are a common Oil & Gas industry format for storing and transferring well log data. The data contained within is used to analyze and understand the subsurface, as well as identify potential hydrocarbon reserves./

**OBJECTIVES**

The key objectives of this Training/Project can be listed as:

1. Read LAS 2.0 Data using Python and serve using JSON format in Flask.
2. Create Linear, Log & Depth Track using ‘Highcharts’ front end JavaScript library.
3. Assembling tracks in Well template and Visualization.
4. Creating a Home screen and assembling.
5. Modification of properties of curves.

**ABOUT THE INDUSTRY**

The Oil and Natural Gas Corporation Limited (ONGC) is the largest oil and gas exploration company in India and dominates the hydrocarbon market in India. ONGC's major products included petroleum, crude natural gas, liquefied petroleum gas (LPG), kerosene and petrochemical feedstock. The company enjoyed a dominant position in the country's hydrocarbon sector with 84 per cent market share of crude oil & gas production. ONGC has its headquarters in Dehradun and its operations are spread throughout the country. ONGC is operating in fourteen countries with its fully owned company OVL (ONGC Videsh Limited). OVL was formed to help ONGC secure a strong foothold in the international oil market. With the acquisition of Mangalore Refinery and Petrochemicals Limited (MRPL), ONGC became the first integrated oil company in India.

**Institutes of ONGC**

ONGC conducts institutional research and development in the oil and gas, and other related sectors. It has established separate institutions to undertake specific activities in key areas of exploration, drilling, reservoir management, production technology, ocean engineering, safety and environment protection

**ONGC in Dehradun**

The headquarters of Oil & Natural Gas Corporation LTD (ONGC) is situated in Dehradun along with the following offices.

* Tel Bhavan (Head Quarter)
* KeshavDevMalviya Institute of Petroleum Exploration (KDMIPE)
* Institute of Drilling Technology (IDT)
* ONGC Academy (formerly Institute of Management Development (IMD))
* Geo Data Processing and Interpretation Centre (GEOPIC)
* Exploration & Development Directorate (E & D Dte.)

#### BRIEF ABOUT E&D DIRECTORATE

The project work was undertaken at Exploration & Development Directorate (E & D Dte.). The functions of E & D Directorate mainly include planning and monitoring, Exploration & Development activities and also the acreage monitoring in ONGC. E&D Dte can further be categorized into several different groups according to their work areas and the nature of the work they deal with.

**SKILLS ACQUIRED DURING INDUSTRIAL INTERNSHIP PERIOD**

Skills learnt during this entire Training period can be listed as:

**FRONTEND**

**FLASK FRAMEWORK**

**Flask** is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

**SOME ELEMENTS OF FLASK-**

Initialization: flask applications must create an application instance. The web server passes all the requests it receives from clients to objects for handling using a protocol for WSG from flask import Flask app = Flask (\_\_name\_\_) (An application instance is an object of class Flask.)

Routes and View Functions in Flask Framework Instance -Clients send requests to the web server, in turn, sends them to the Flask application instance. The instance needs to know what code needs to run for each URL requested and map URLs to Python functions. The association between a URL and the function that handles it is called a route. The most convenient way to define a route in a Flask application is through the (app.route). Decorator exposed by the application instance, which registers the ‘decorated function,’ decorators are python feature that modifies the behavior of a function.

The index is a view function, and the response can even be a string format HTML.

# Server Startup - The application instance has a ‘run’ method that launches flask’s integrated development webserver -

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

Once the script starts, it waits for requests and services in a loop.

# Local-Host - Run a python script in a virtual environment. Flask starts the server listening on 127.0.0.1 and port 5000 by default. To accept connection from any remote address, use host = ‘0.0.0.0.’

### HTTP Methods

#### Request

To process incoming data in Flask, you need to use the request object, including mime-type, IP address, and data. HEAD: Un-encrypted data sent to server w/o response.

#### GET

Sends data to the server requesting a response body.

#### POST

Read form inputs and register a user, send HTML data to the server are methods handled by the route. Flask attaches methods to each route so that different view functions can handle different request methods to the same URL.

#### Response

Flask invokes a view function. It has to return a response value to the client. HTTP requires it to be more than a string response, a status code.

**IMPORTING PYTHON LIBRARIES**

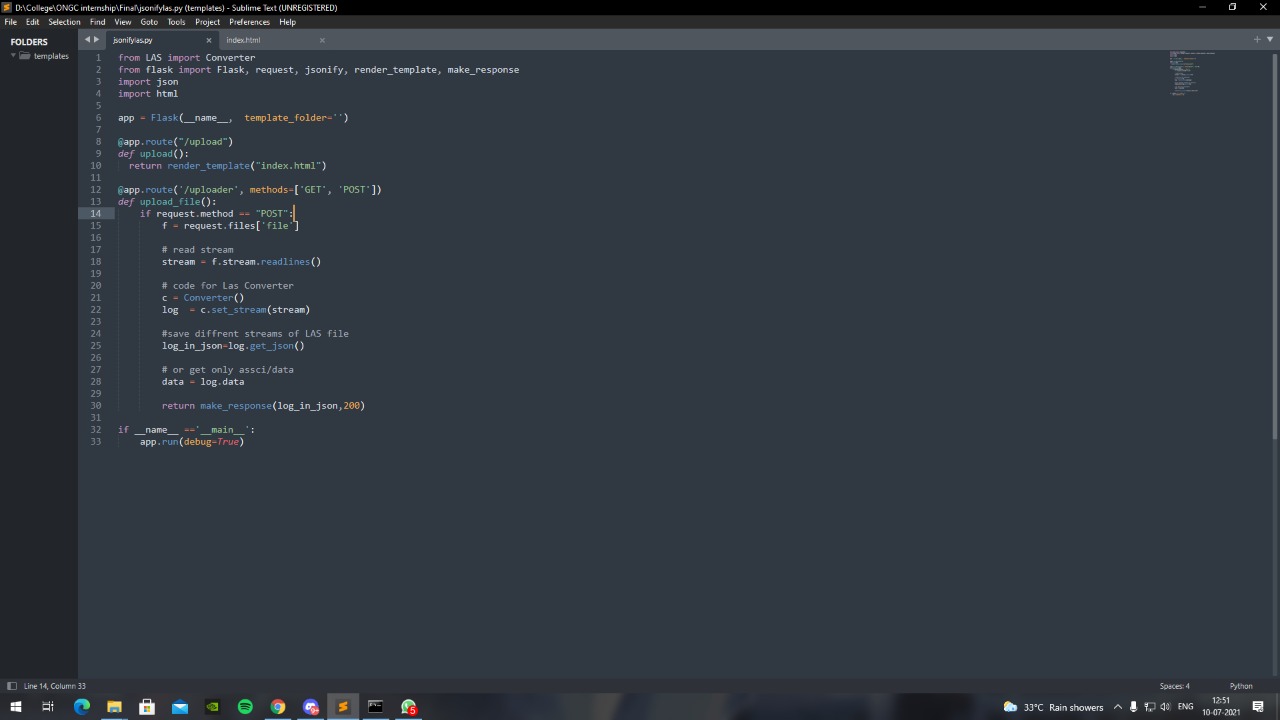
We imported Python libraries such as:

1. Las Convertor,
2. flask libraries such as jsonify,
3. render\_template,
4. make\_response
5. We created an object of flask and define flask route as upload.

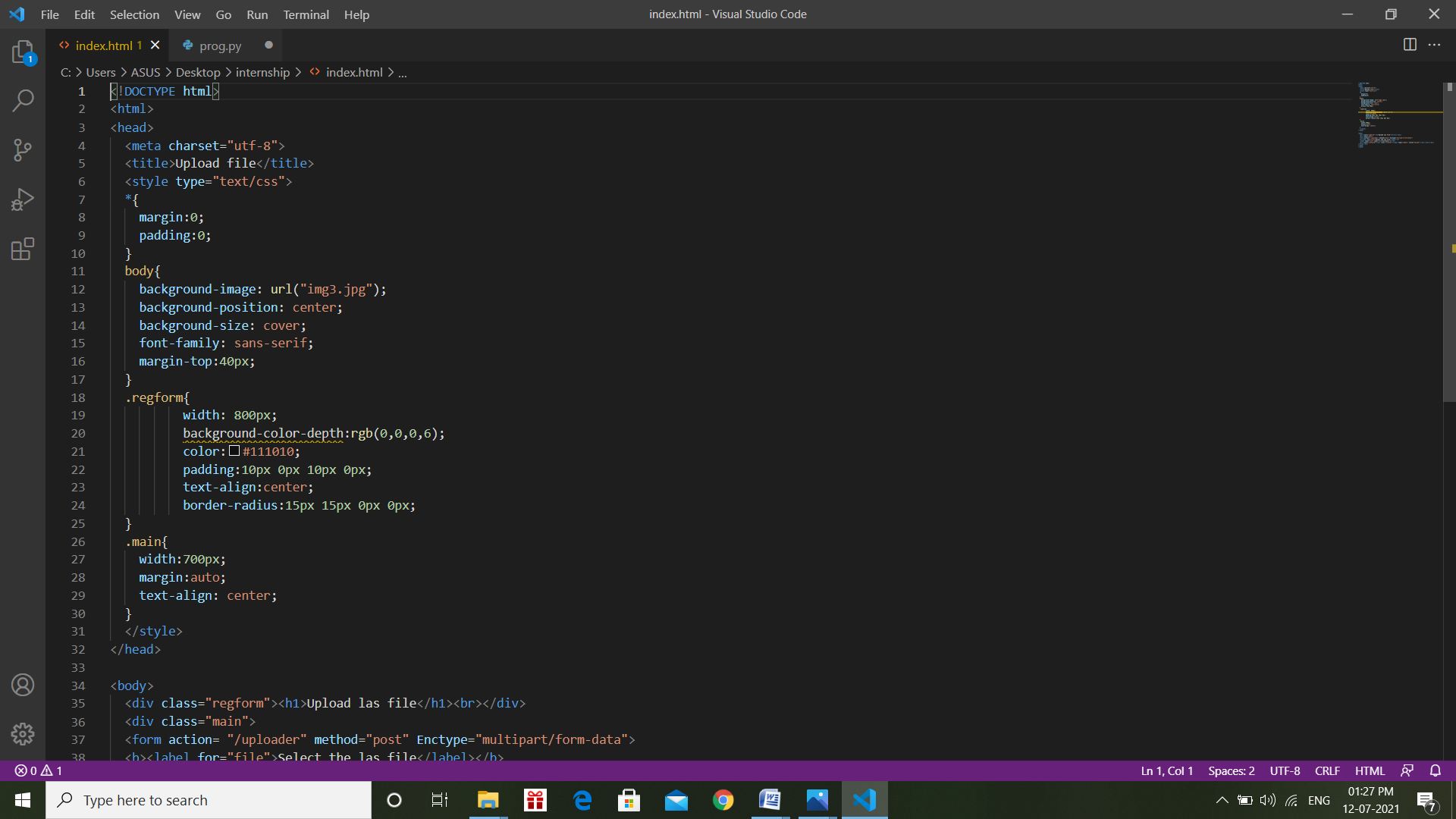
**SERVING LAS DATA OVER API USING JSON FORMAT**

We also create and render an html file index.html for defining the form that is going to be used to upload our las file(oil&gas.las).

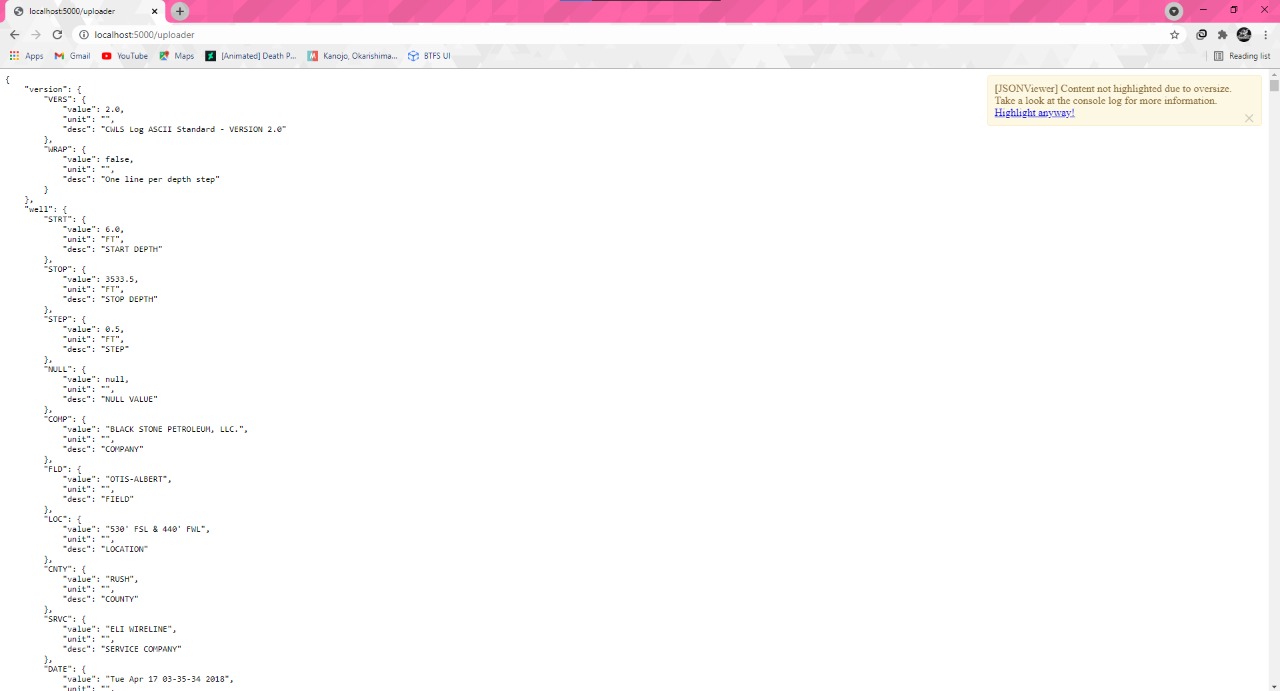
Subsequently we use ‘run’ method to launch flask’s integrated development web server.



index.html



Serving las data over API using JSONFormat



**BACKEND**

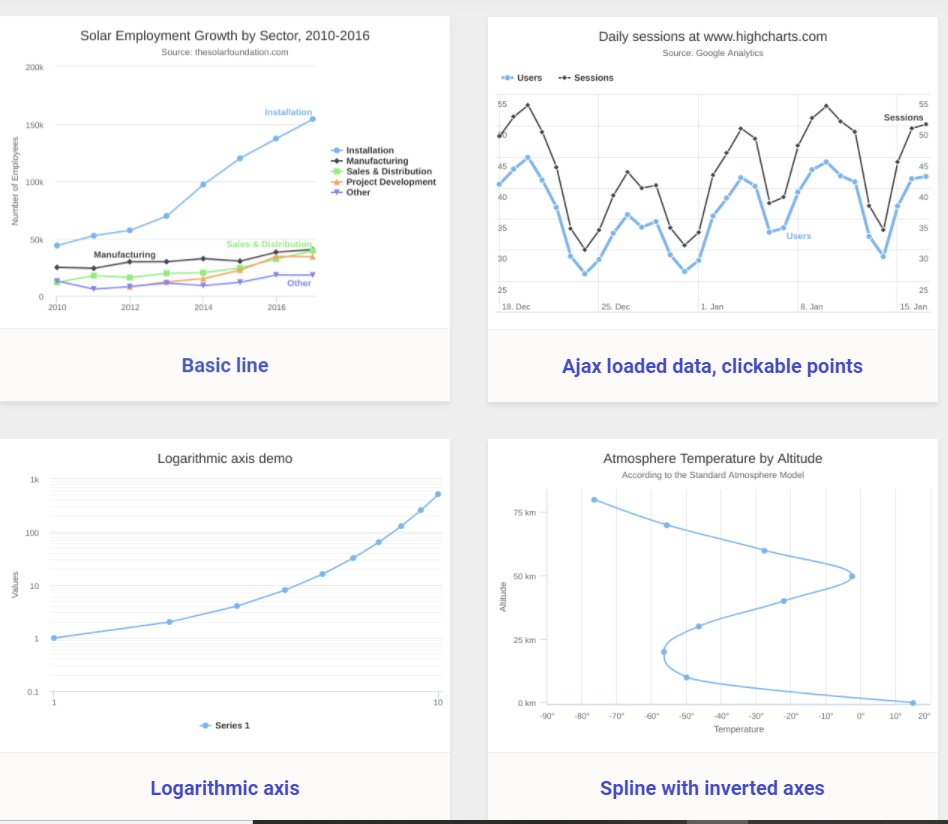
**JAVASCRIPT ‘HIGHCHARTS’ LIBRARY**

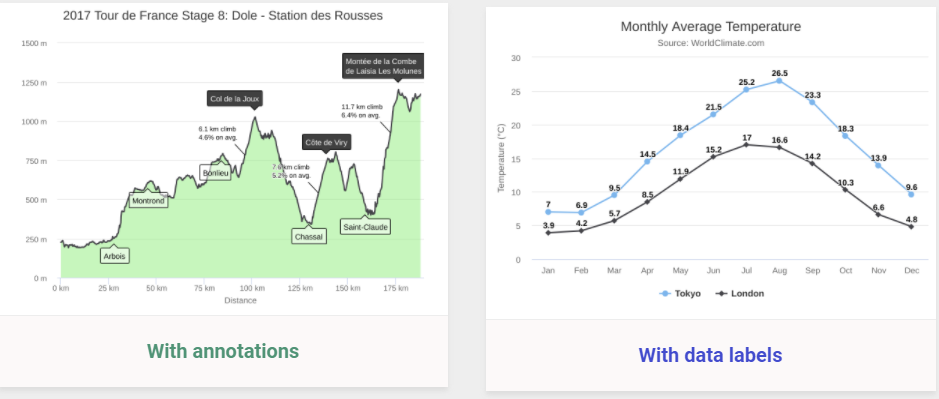
We learnt how to create different types of graphs using this library, how to perform different functions over them and how to format them according to our needs.

**JavaScript:** JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.

**Highcharts:** Highcharts is a library written purely in JavaScript used to enhance web applications by inserting charts capable of interaction. ... You can choose highchart pie charts, highchart line charts, spline charts, area charts, highchart bar charts, etc., to represent your data.

Some of the common Types of Highcharts-

****

****

**JavaScript files (Graphs)**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**HTML**

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behaviour (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

Text

Description automatically generated

**CSS**

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.CSS is used to control the style of a web document in a simple and easy way.

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

**HOME PAGE**

**A picture containing diagram

Description automatically generated**

**LEARNING OUTCOMES**

* Developed and worked over this project in a limited time span of 1 month.
* Learnt new skills and technologies.
* Got hands on experience on the working of new technologies and frameworks such as Flask, LAS Data, High charts JavaScript, etc.
* Learnt how to work individually and as a team since this whole training was on virtual platform.

**CONCLUSION**

* We were successfully able to visualise LAS 2.0 Data over this web-based utility that we created.
* The files are successfully being uploaded and visualised on the Home Page.