# NORMAL VS C SECTION DELIVERY

Developed by :

**P THARUN KRISHNA**

**20231CBD0034**

## PROBLEM STATEMENT

COMPARATIVE ANALYSIS OF
NORMAL DELIVERY VS C-SECTION DELIVERY

## ABSTRACT

THE PROJECT INVOLVES REPRESENTING VISUAL INFORMATION ALONG WITH RESPECTIVE DESCRIPTIONS IN THE FORM OF SERVER BUILT USING FLASK API.
THE REAL-TIME DATA HAS BEEN EXTRACTED FROM THE 5TH NATIONAL FAMILY HEALTH SURVEY REPORT (INDIA), 2019-21. THIS PROJECT INVOLVES USAGE OF MYSQL AND MATPLOTLIB LIBRARY.

# TABLE OF CONTENTS

# DESCRIPTION

THE PROJECT HAS BEEN MODULARIZED BASED ON THE THREE DOMAINS:

1. SQL DATABASE
2. VISUALIZATION
3. FLASK SERVER

FIRSTLY, THE DATA EXTRACTED FROM THE SURVEY REPORT HAS BEEN STORED AS TABULAR DATA IN SQL DATABASE. FUNCTIONS DEFINED TO CONNECT / DISCONNECT DATABASE AND READ TABLE CONTENTS IS DEFINED INTO A PYTHON FILE.

ON THE OTHER HAND, THE FUNCTIONS TO CREATE VISUALS HAVE BEEN MODULARIZED INTO A NEW PYTHON FILE. THIS PROMOTES RE-USABILITY OF THE CODE JUST BY IMPORTING THE CONTENTS OF THE CURRENT FILE.

FINALLY, THE FLASK API IMPLEMENTATION OF THE PROJECT IS STORED AS THE MAIN FILE. ON RUNNING THE SERVER, A DROP-DOWN LIST APPEARS WITH A LIST OF CRITERIA TO ANALYSE THE CURRENT TOPIC. ON CLICKING THE “VIEW GRAPH” BUTTON, IT DISPLAYS THE VISUAL IN A NEW WINDOW. ALSO, THE RESPECTIVE DESCRIPTION APPEARS ON THE WEBSITE ONCE THE VISUAL WINDOW IS CLOSED. THE WEBSITE ALLOWS US VIEW GRAPHS DYNAMICALLY.

## DESCRIPTION

.....CONTINUED

THE PURPOSE OF THE PROJECT IS TO SHOWCASE A COMPARATIVE ANALYSIS BETWEEN NORMAL DELIVERY AND C-SECTION DELIVERY. THIS ANALYSIS SERVES AS AN INFORMATIVE SOURCE ENHANCED BY VISUALS.

THE PARAMETERS DISCUSSED IN THIS ANALYSIS ARE AS FOLLOWS:

1. **BIRTH ORDER** - REFERS TO ORDER OF THE CHILD BEING DELIVERED. EX: FIRST BORN, SECOND BORN, ETC.
2. **DURATION OF STAY** - REFERS TO HOW LONG PATIENTS STAY AT HOSPITALS POST DELIVERY. THIS HIGHLIGHTS ABOUT THE RECOVERY TIME IN BOTH CASES.
3. **RESIDENCE** - STATES HOW PREFERENCE OF C SECTION DELIVERY DIFFERS FROM URBAN TO RURAL AREAS.
4. **WEALTH QUINTILE -** DESCRIBES HOW ECONOMIC STATUS INFLUENCE THE DECISION OF C-SECTION DELIVERY.

## DESIGN AND SOLUTION PLAN

STEP 1:
ORGANIZE THE DIRECTORY IN THE ORDER GIVEN BELOW:

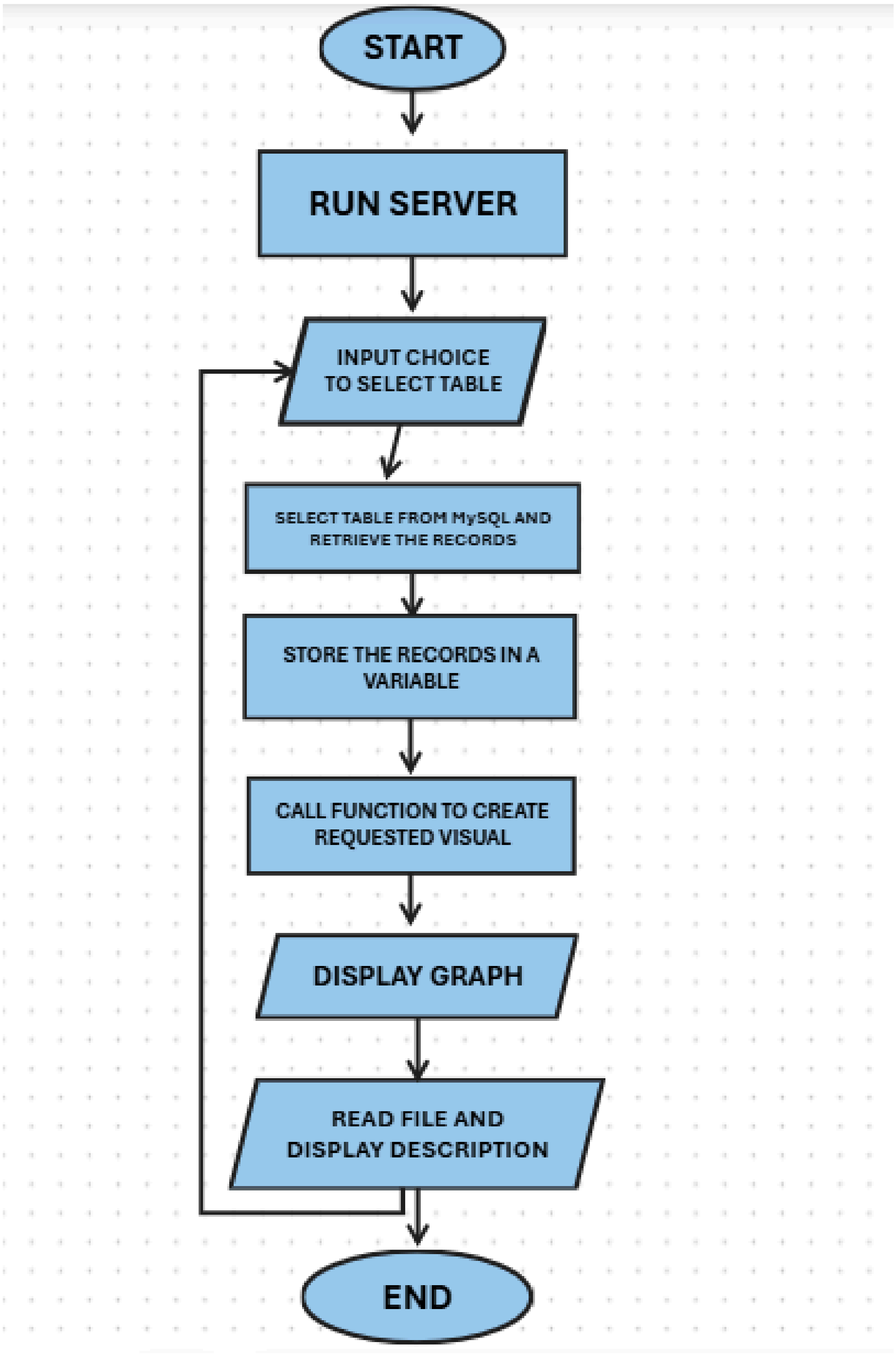
```
Project/
 ├── descriptions/
 │   ├── birth_order_criteria.txt
 │   ├── stay_duration_criteria_public.txt
 │   ├── stay_duration_criteria_private.txt
 │   ├── residence_criteria.txt
 │   └── wealth_quintile_criteria.txt
 ├── templates/
 │   └── index.html
 ├── db_operations.py
 ├── main.py
 └── visuals.py
```

STEP 2: RUN THE main.py file
STEP 3: A SERVER LINK APPEARS. CTRL + CLICK ON THE LINK TO GO TO THE WEBSITE.

### REQUIRED MODULES:

- pymysql
- matplotlib
- flask
- numpy

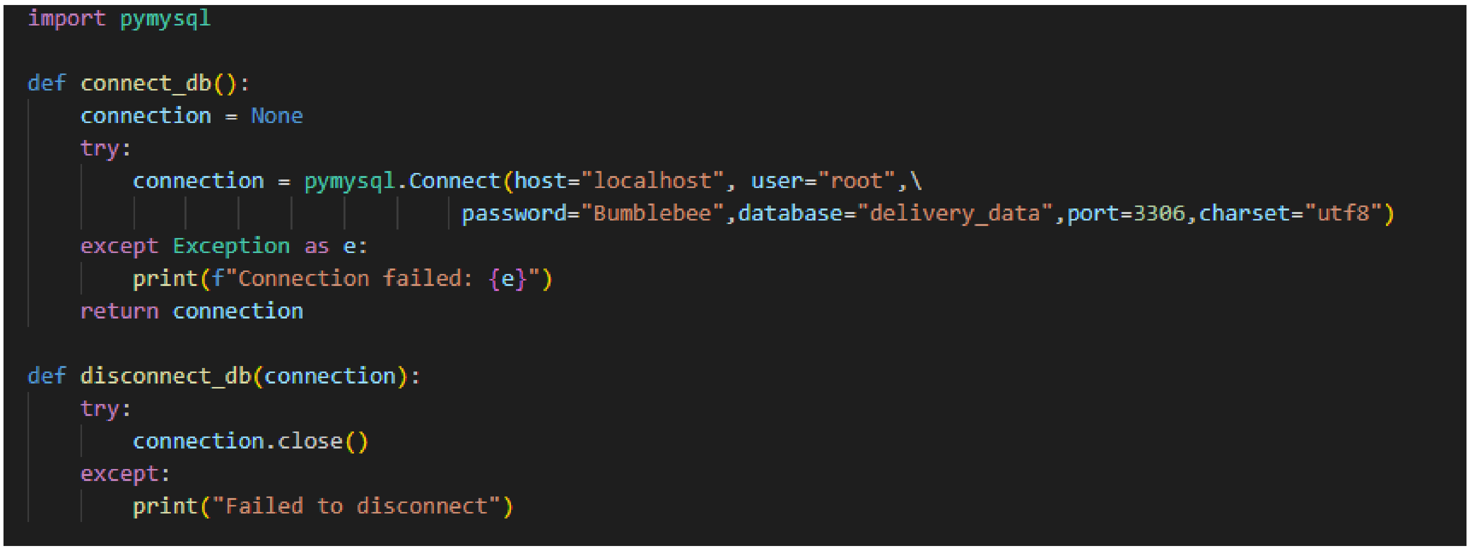
# FLOWCHART


START
RUN SERVER
INPUT CHOICE TO SELECT TABLE
SELECT TABLE FROM MySQL AND RETRIEVE THE RECORDS
STORE THE RECORDS IN A VARIABLE
CALL FUNCTION TO CREATE REQUESTED VISUAL
DISPLAY GRAPH
READ FILE AND DISPLAY DESCRIPTION
END

# IMPLEMENTATION

## 1. FUNCTIONS DEFINED TO ACCESS MYSQL DATABASE (BACK-END)

### A. CONNECT / DISCONNECT DATABASE



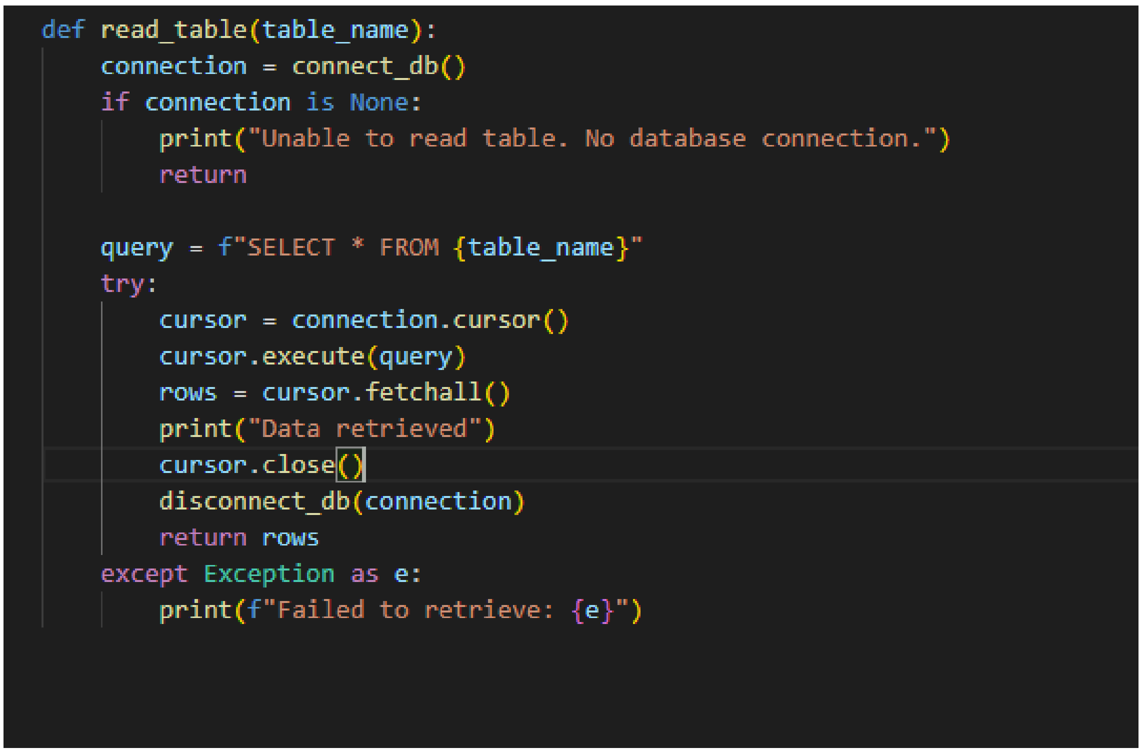
```python
import pymysql

def connect_db():
    connection = None
    try:
        connection = pymysql.Connect(host="localhost", user="root",\
                                     password="Bumblebee",database="delivery_data",port=3306,charset="utf8")
    except Exception as e:
        print(f"Connection failed: {e}")
    return connection

def disconnect_db(connection):
    try:
        connection.close()
    except:
        print("Failed to disconnect")
```

connect_db function is defined to connect to the MySQL Database in order to access the table requested by the user. It requires arguments : host, user, password, database, port and charset.

disconnect_db function is defined to close the connection in order to ensure safety of data and avoid risk of losing/modifying data by intruders.

Both block of codes are implemented using exception handling to ensure smooth execution of program without popping errors

## B. READ TABLE CONTENTS REQUESTED BY USER



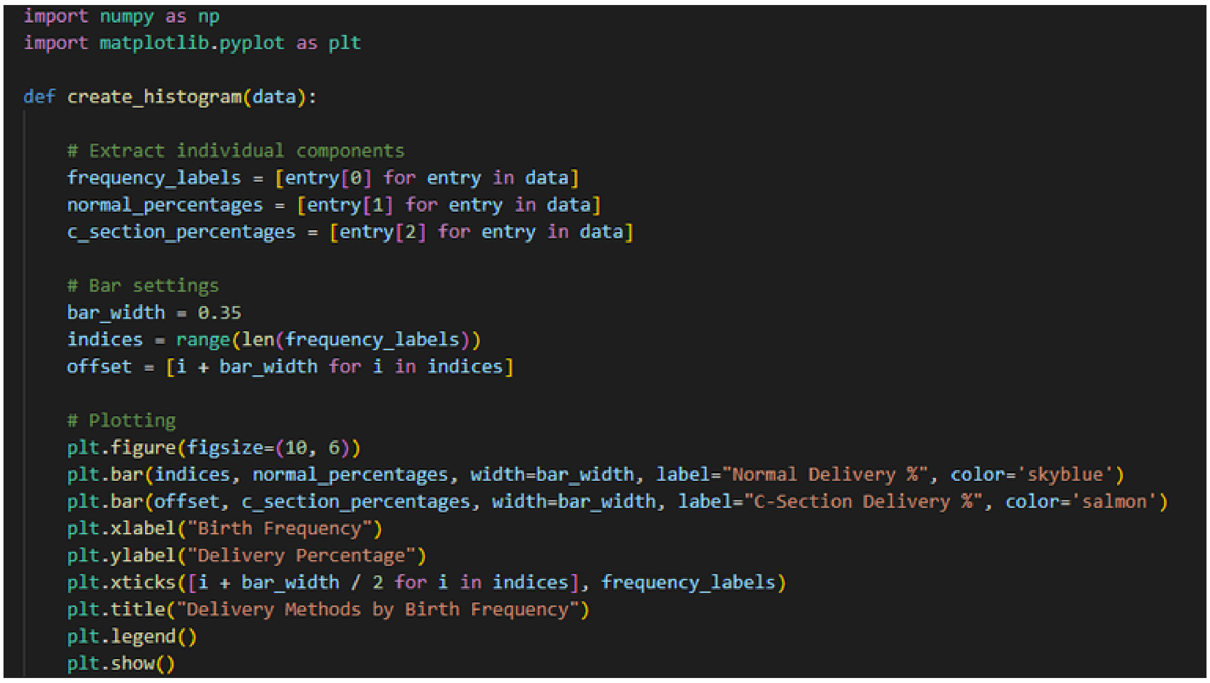
```python
def read_table(table_name):
    connection = connect_db()
    if connection is None:
        print("Unable to read table. No database connection.")
        return

    query = f"SELECT * FROM {table_name}"
    try:
        cursor = connection.cursor()
        cursor.execute(query)
        rows = cursor.fetchall()
        print("Data retrieved")
        cursor.close()
        disconnect_db(connection)
        return rows
    except Exception as e:
        print(f"Failed to retrieve: {e}")
```

read_table function is defined to access the records of the table requested by user. It takes one parameter table_name. This variable is passed down to the query string.

After establishing connection to database, the query is executed by cursor function in MySQL. Using fetchall function, the table records are retrieved and stored in variable “row”. Then the connection is terminated and the rows are returned.

## 2. FUNCTIONS DEFINED TO CREATE VISUALS

### A. HISTOGRAM



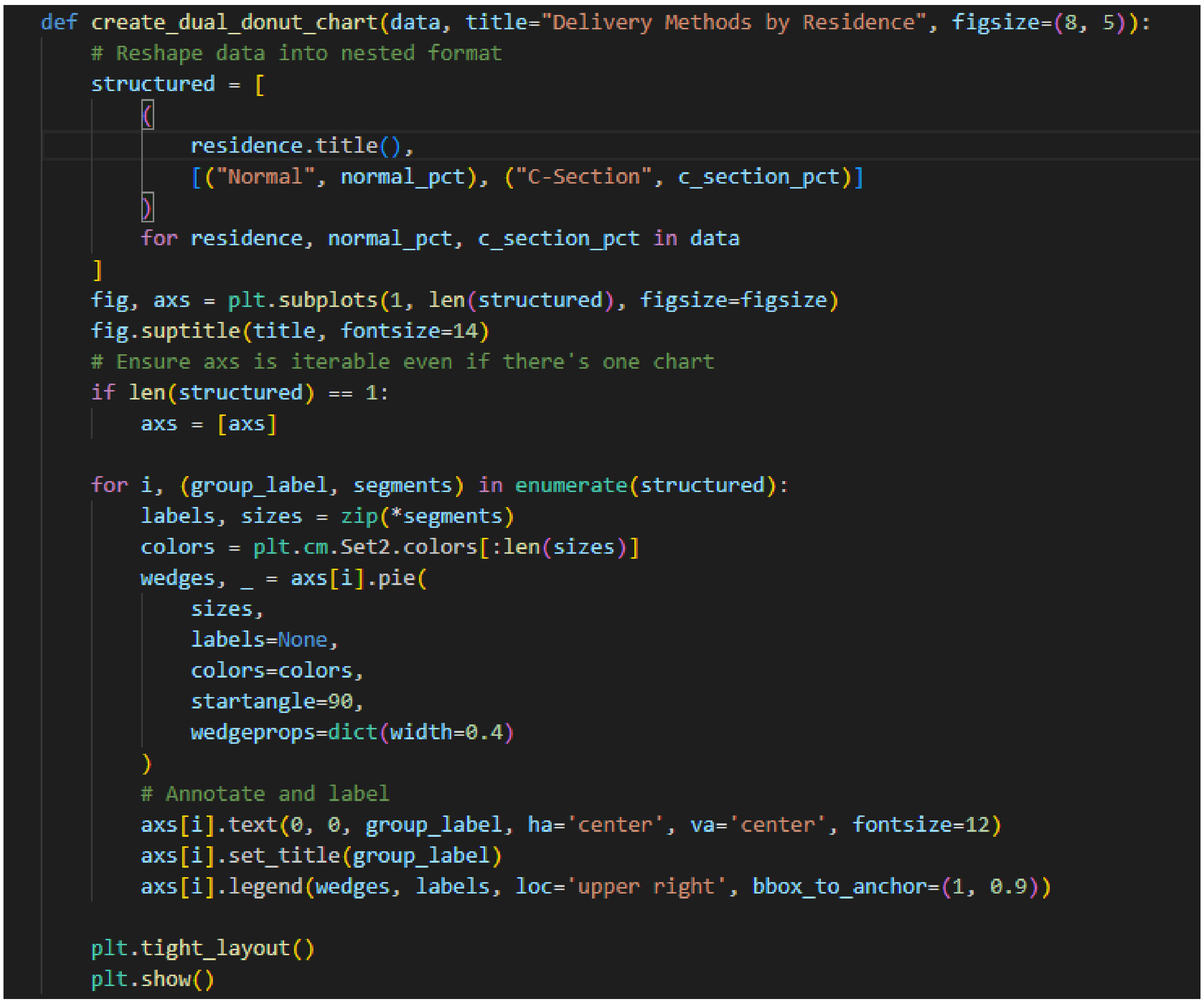
```python
import numpy as np
import matplotlib.pyplot as plt

def create_histogram(data):

    # Extract individual components
    frequency_labels = [entry[0] for entry in data]
    normal_percentages = [entry[1] for entry in data]
    c_section_percentages = [entry[2] for entry in data]

    # Bar settings
    bar_width = 0.35
    indices = range(len(frequency_labels))
    offset = [i + bar_width for i in indices]

    # Plotting
    plt.figure(figsize=(10, 6))
    plt.bar(indices, normal_percentages, width=bar_width, label="Normal Delivery %", color='skyblue')
    plt.bar(offset, c_section_percentages, width=bar_width, label="C-Section Delivery %", color='salmon')
    plt.xlabel("Birth Frequency")
    plt.ylabel("Delivery Percentage")
    plt.xticks([i + bar_width / 2 for i in indices], frequency_labels)
    plt.title("Delivery Methods by Birth Frequency")
    plt.legend()
    plt.show()
```

create_histogram function is defined to plot a histogram based on assigned data. As the data retrieved from MySQL is in the form of structured tuples, the labels and percentages are stored in new variables using list comprehension.

Bar width and space between consecutive bars also can be set in the code.

Finally the histogram is plotted using matplotlib.pyplot library. Values and labels are assigned to the axes along with creation of legend. The plt.show() method displays the graph

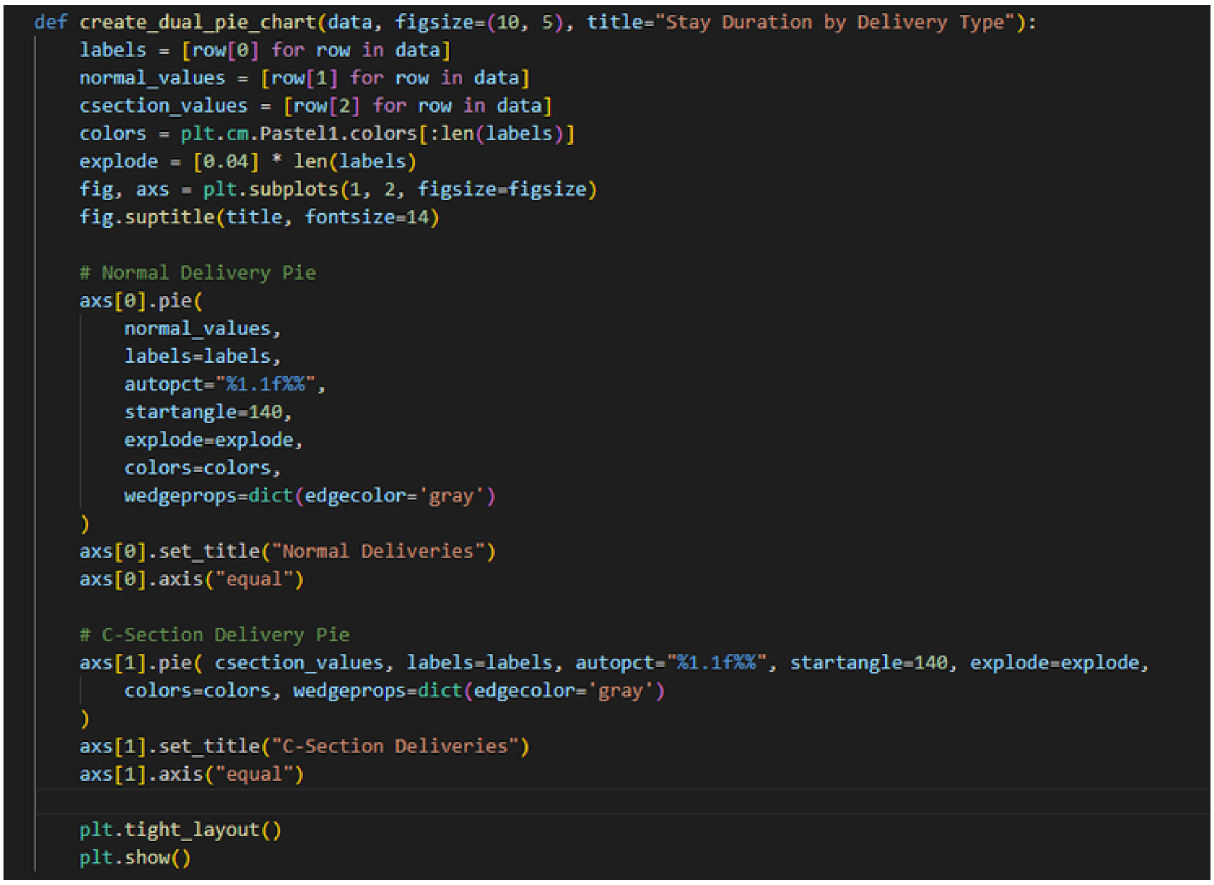
## B. DUAL DONUT CHART



```python
def create_dual_donut_chart(data, title="Delivery Methods by Residence", figsize=(8, 5)):
    # Reshape data into nested format
    structured = [
        (
            residence.title(),
            [("Normal", normal_pct), ("C-Section", c_section_pct)]
        )
        for residence, normal_pct, c_section_pct in data
    ]
    fig, axs = plt.subplots(1, len(structured), figsize=figsize)
    fig.suptitle(title, fontsize=14)
    # Ensure axs is iterable even if there's one chart
    if len(structured) == 1:
        axs = [axs]

    for i, (group_label, segments) in enumerate(structured):
        labels, sizes = zip(*segments)
        colors = plt.cm.Set2.colors[:len(sizes)]
        wedges, _ = axs[i].pie(
            sizes,
            labels=None,
            colors=colors,
            startangle=90,
            wedgeprops=dict(width=0.4)
        )
        # Annotate and label
        axs[i].text(0, 0, group_label, ha='center', va='center', fontsize=12)
        axs[i].set_title(group_label)
        axs[i].legend(wedges, labels, loc='upper right', bbox_to_anchor=(1, 0.9))

    plt.tight_layout()
    plt.show()
```

create_dual_donut_chart function is defined to display two donut charts side to side to enable comparison of two different parameters.
Firstly, the structured tuples given as input is processed to make it suitable for plotting donut chart. Also the data is divided based on two parameters.
Each parameter data is plotted and color palette is assigned to it for easier identification.
Labels are aligned such that to fit the plot appropriately.
Subsequently, a legend is also created for identification purpose.

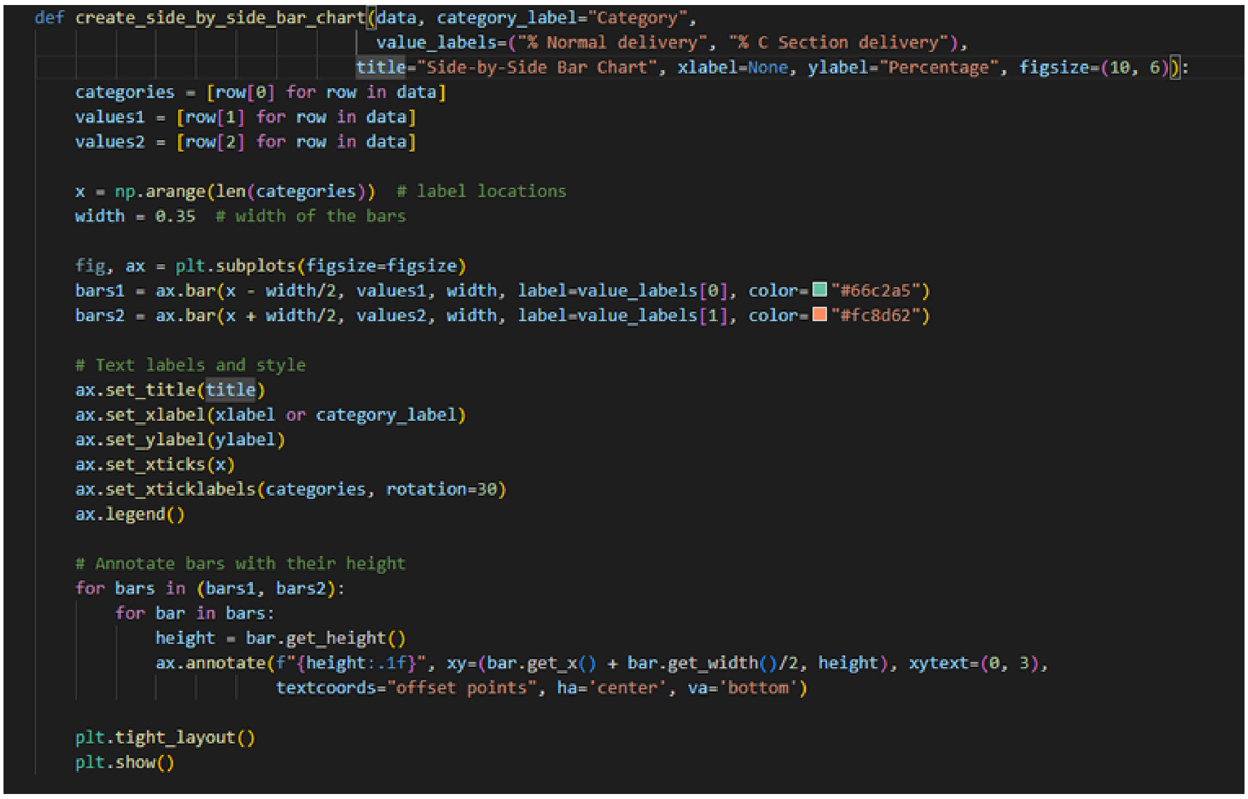
## C. DUAL PIE CHART



```python
def create_dual_pie_chart(data, figsize=(10, 5), title="Stay Duration by Delivery Type"):
    labels = [row[0] for row in data]
    normal_values = [row[1] for row in data]
    csection_values = [row[2] for row in data]
    colors = plt.cm.Pastel1.colors[:len(labels)]
    explode = [0.04] * len(labels)
    fig, axs = plt.subplots(1, 2, figsize=figsize)
    fig.suptitle(title, fontsize=14)

    # Normal Delivery Pie
    axs[0].pie(
        normal_values,
        labels=labels,
        autopct="%1.1f%%",
        startangle=140,
        explode=explode,
        colors=colors,
        wedgeprops=dict(edgecolor='gray')
    )
    axs[0].set_title("Normal Deliveries")
    axs[0].axis("equal")

    # C-Section Delivery Pie
    axs[1].pie( csection_values, labels=labels, autopct="%1.1f%%", startangle=140, explode=explode,
        colors=colors, wedgeprops=dict(edgecolor='gray')
    )
    axs[1].set_title("C-Section Deliveries")
    axs[1].axis("equal")

    plt.tight_layout()
    plt.show()
```

create_dual_pie_chart function is defined to display two pie charts side to side to enable comparison of two different parameters.
Firstly, the structured tuples given as input is processed to make it suitable for plotting pie chart. Also the data is divided based on two parameters.
Each parameter data is plotted and color palette is assigned to it for easier identification.
Labels are aligned such that to fit the plot appropriately.
Subsequently, a legend is also created for identification purpose.

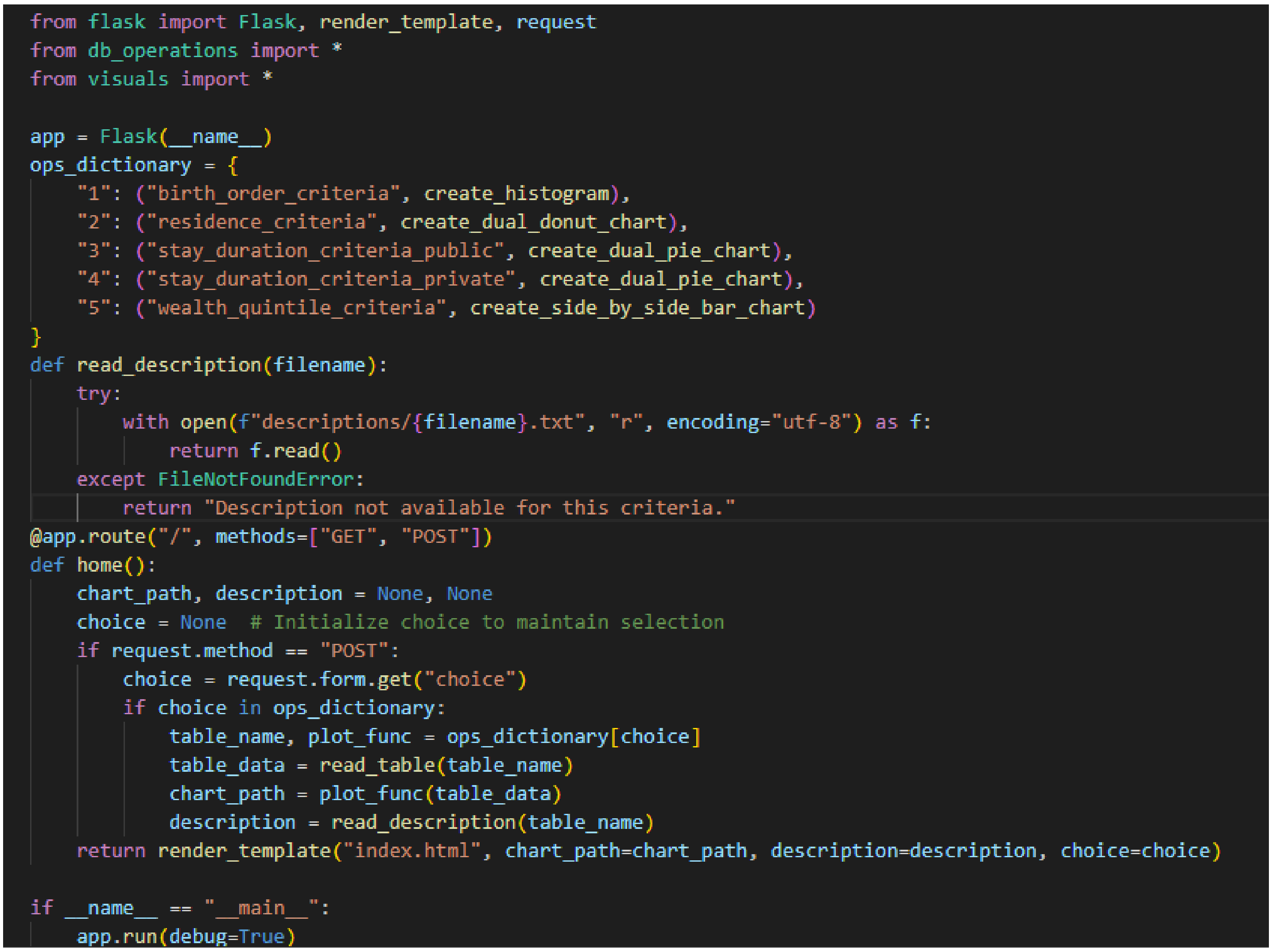
## D. SIDE BY SIDE BAR CHART



```python
def create_side_by_side_bar_chart(data, category_label="Category",
                                  value_labels=("% Normal delivery", "% C Section delivery"),
                                  title="Side-by-Side Bar Chart", xlabel=None, ylabel="Percentage", figsize=(10, 6)):
    categories = [row[0] for row in data]
    values1 = [row[1] for row in data]
    values2 = [row[2] for row in data]

    x = np.arange(len(categories))  # label locations
    width = 0.35  # width of the bars

    fig, ax = plt.subplots(figsize=figsize)
    bars1 = ax.bar(x - width/2, values1, width, label=value_labels[0], color="#66c2a5")
    bars2 = ax.bar(x + width/2, values2, width, label=value_labels[1], color="#fc8d62")

    # Text labels and style
    ax.set_title(title)
    ax.set_xlabel(xlabel or category_label)
    ax.set_ylabel(ylabel)
    ax.set_xticks(x)
    ax.set_xticklabels(categories, rotation=30)
    ax.legend()

    # Annotate bars with their height
    for bars in (bars1, bars2):
        for bar in bars:
            height = bar.get_height()
            ax.annotate(f"{height:.1f}", xy=(bar.get_x() + bar.get_width()/2, height), xytext=(0, 3),
                        textcoords="offset points", ha='center', va='bottom')

    plt.tight_layout()
    plt.show()
```

create_side_by_side_bar_chart function is defined to plot a bar graph for two parameters in the same plot area.
Firstly, the data received as structured tuples is processed to new variables using list comprehension.
Secondly, the width of each bars and labels are set accordingly. The labels are rotated by specific angle for aesthetic purpose.
Finally, the bar graph is plotted with appropriate labels, legend and spacing.

## 3. FLASK OPERATIONS



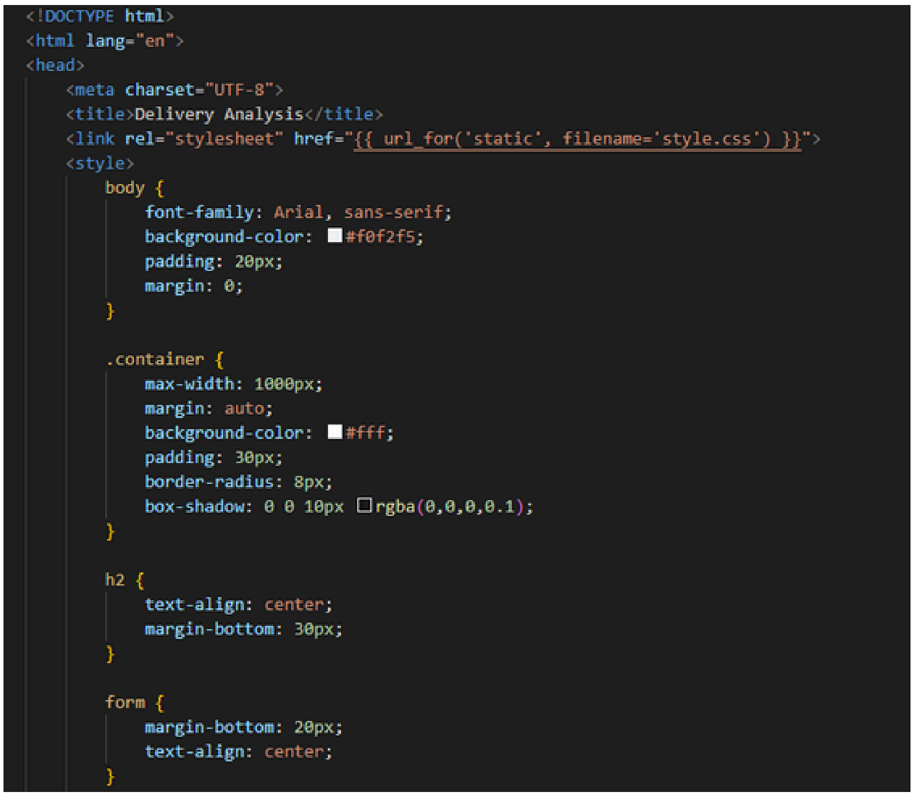
```python
from flask import Flask, render_template, request
from db_operations import *
from visuals import *

app = Flask(__name__)
ops_dictionary = {
    "1": ("birth_order_criteria", create_histogram),
    "2": ("residence_criteria", create_dual_donut_chart),
    "3": ("stay_duration_criteria_public", create_dual_pie_chart),
    "4": ("stay_duration_criteria_private", create_dual_pie_chart),
    "5": ("wealth_quintile_criteria", create_side_by_side_bar_chart)
}
def read_description(filename):
    try:
        with open(f"descriptions/{filename}.txt", "r", encoding="utf-8") as f:
            return f.read()
    except FileNotFoundError:
        return "Description not available for this criteria."
@app.route("/", methods=["GET", "POST"])
def home():
    chart_path, description = None, None
    choice = None  # Initialize choice to maintain selection
    if request.method == "POST":
        choice = request.form.get("choice")
        if choice in ops_dictionary:
            table_name, plot_func = ops_dictionary[choice]
            table_data = read_table(table_name)
            chart_path = plot_func(table_data)
            description = read_description(table_name)
    return render_template("index.html", chart_path=chart_path, description=description, choice=choice)

if __name__ == "__main__":
    app.run(debug=True)
```

Prior to flask instructions, a dictionary is defined which contains a list of table name and visual name as elements. read_description function is defined to read a file from the directory in order to display description along with the displayed graph.

Followed by this, the flask implementation is defined. The implementation is carried out only “POST” method as we are only displaying the contents.
On executing the python file, a link appears. The link takes us to an external website server for operations.
On obtaining choice as input, its respective functions are called out and displayed.

## 4. HTML IMPLEMENTATION (FRONT-END)

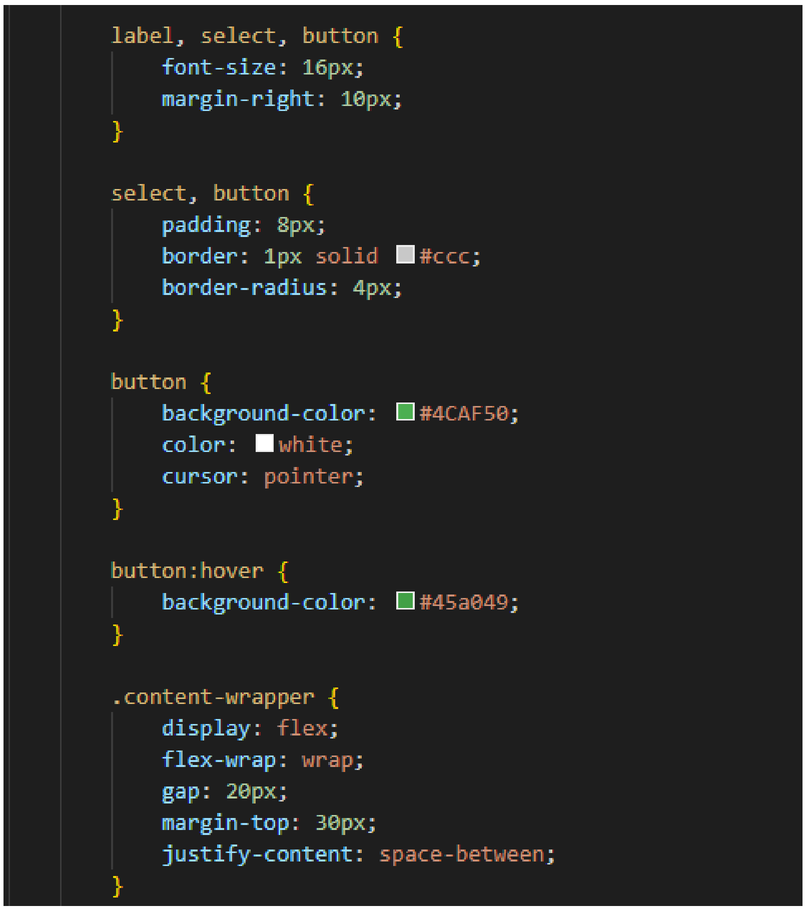
### A. DISPLAY TEXT ON WEB-PAGE



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Delivery Analysis</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f0f2f5;
            padding: 20px;
            margin: 0;
        }

        .container {
            max-width: 1000px;
            margin: auto;
            background-color: #fff;
            padding: 30px;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }

        h2 {
            text-align: center;
            margin-bottom: 30px;
        }

        form {
            margin-bottom: 20px;
            text-align: center;
        }
```

HTML code is structured to display the textual contents on the website which appears from the generated server address.
The headings and text included in the website are structured and aligned by this block of code.

### B. BUTTONS AND DROP-DOWN LIST

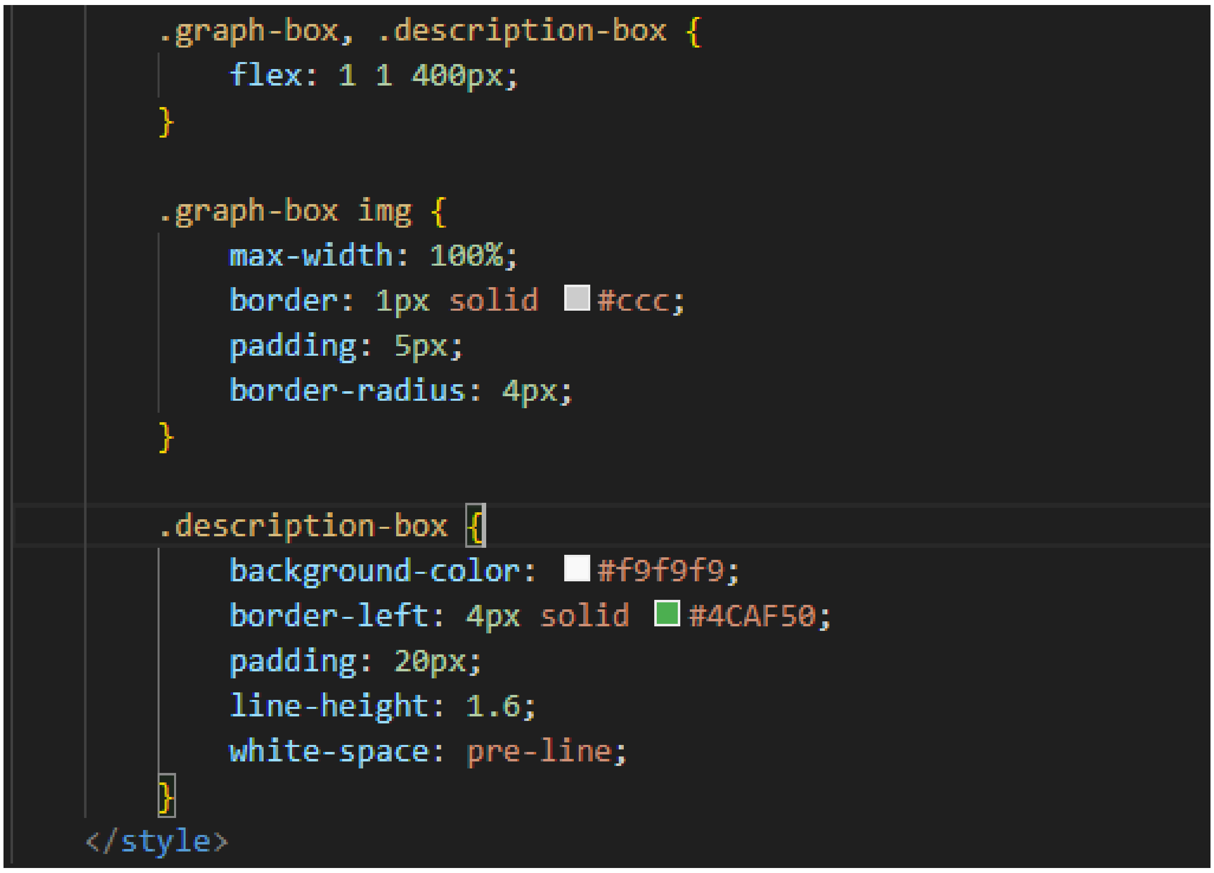


```css
label, select, button {
    font-size: 16px;
    margin-right: 10px;
}

select, button {
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

button {
    background-color: #4CAF50;
    color: white;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

.content-wrapper {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    margin-top: 30px;
    justify-content: space-between;
}
```

HTML code is structured to create the “View Graph” button in the website. This button acts as the trigger which display graphs and their respective descriptions.

The drop down list is created to enable the user to choose his/her choice of parameter to be displayed.

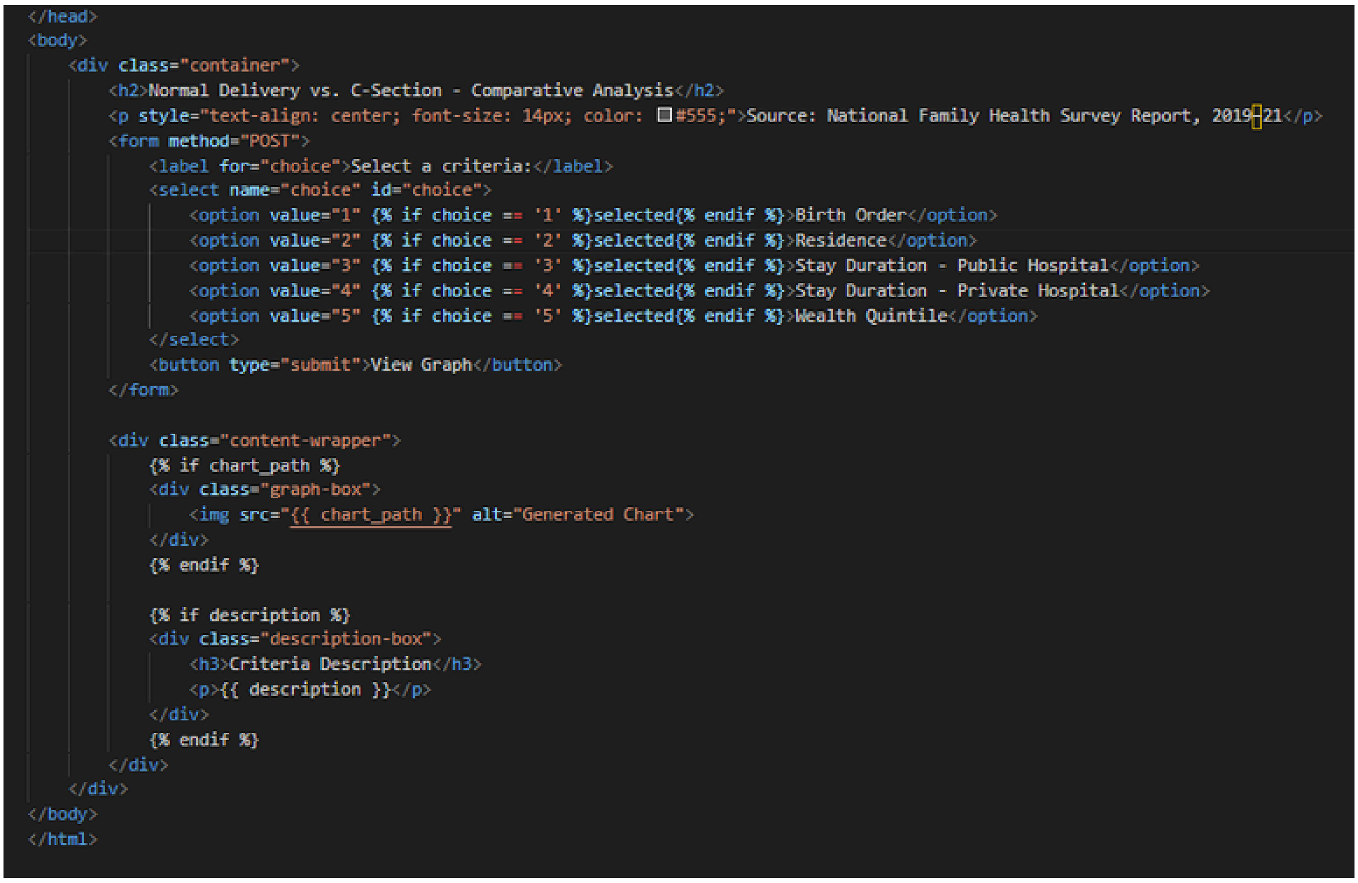
## C. NEW WINDOW TO DISPLAY VISUALS



```
        .graph-box, .description-box {
            flex: 1 1 400px;
        }

        .graph-box img {
            max-width: 100%;
            border: 1px solid #ccc;
            padding: 5px;
            border-radius: 4px;
        }

        .description-box {
            background-color: #f9f9f9;
            border-left: 4px solid #4CAF50;
            padding: 20px;
            line-height: 1.6;
            white-space: pre-line;
        }
    </style>
```

HTML code for displaying the graph and respective description. This block of code enables us to create a new window to display graph and also has an option to save the graph to local device as image in png format.

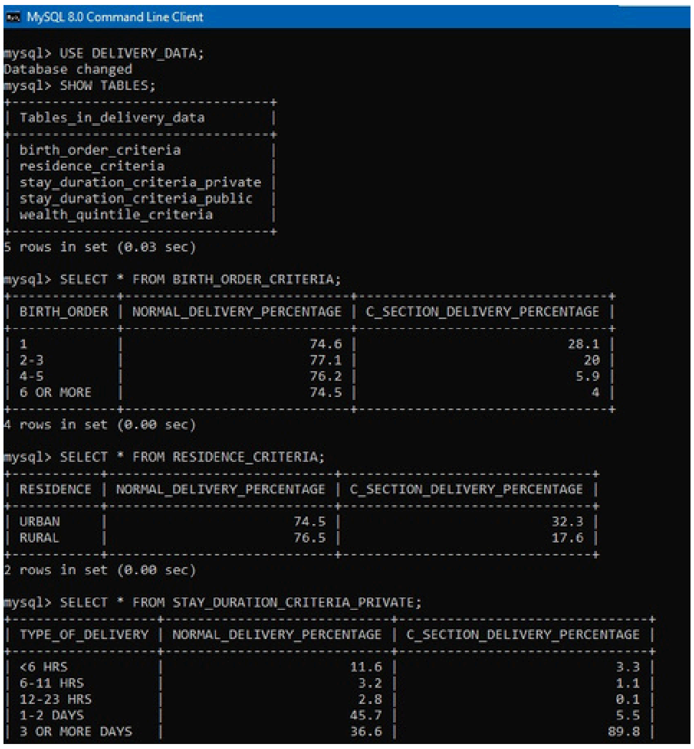
## D. DISPLAY RESPECTIVE DESCRIPTIONS OF THE GRAPH



```html
</head>
<body>
    <div class="container">
        <h2>Normal Delivery vs. C-Section - Comparative Analysis</h2>
        <p style="text-align: center; font-size: 14px; color: ■#555;">Source: National Family Health Survey Report, 2019–21</p>
        <form method="POST">
            <label for="choice">Select a criteria:</label>
            <select name="choice" id="choice">
                <option value="1" {% if choice == '1' %}selected{% endif %}>Birth Order</option>
                <option value="2" {% if choice == '2' %}selected{% endif %}>Residence</option>
                <option value="3" {% if choice == '3' %}selected{% endif %}>Stay Duration - Public Hospital</option>
                <option value="4" {% if choice == '4' %}selected{% endif %}>Stay Duration - Private Hospital</option>
                <option value="5" {% if choice == '5' %}selected{% endif %}>Wealth Quintile</option>
            </select>
            <button type="submit">View Graph</button>
        </form>

        <div class="content-wrapper">
            {% if chart_path %}
            <div class="graph-box">
                <img src="{{ chart_path }}" alt="Generated Chart">
            </div>
            {% endif %}

            {% if description %}
            <div class="description-box">
                <h3>Criteria Description</h3>
                <p>{{ description }}</p>
            </div>
            {% endif %}
        </div>
    </div>
</body>
</html>
```

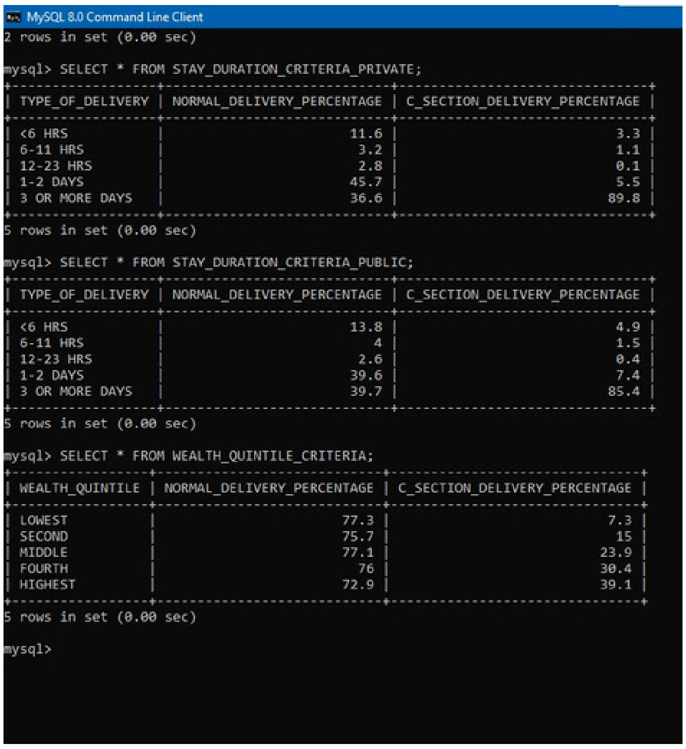
HTML code to display the heading text and the respective description of the requested graph, read from txt file using file handling method, when the graph window appears.
The code is also set such that the drop down list should not reset.

# OUTPUT SCREENSHOTS



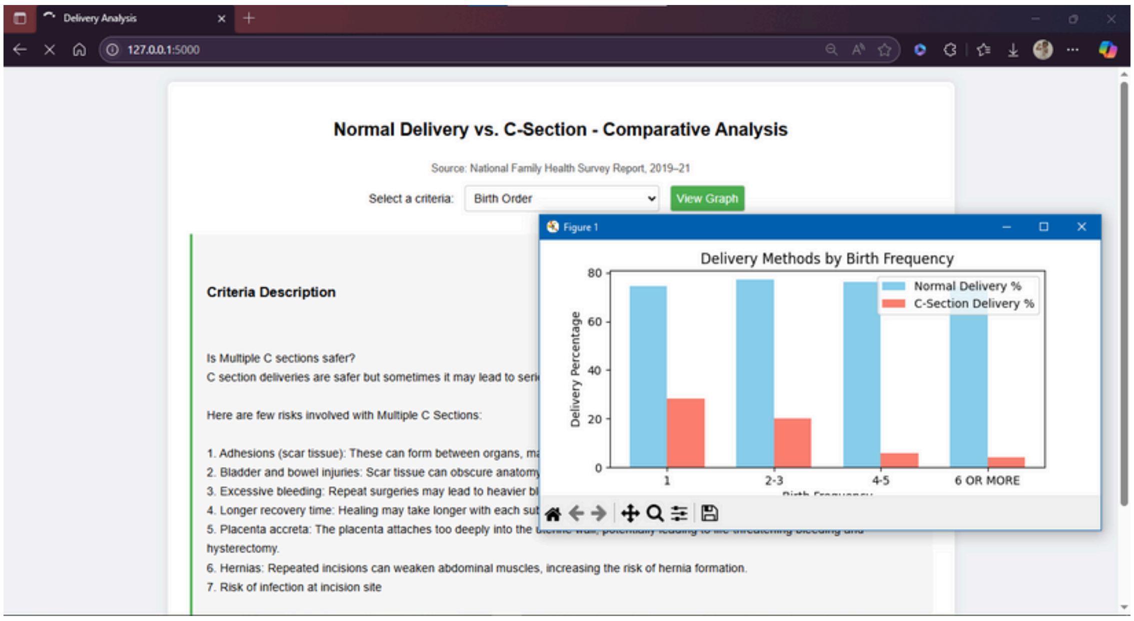
MySQL 8.0 Command Line Client

```
mysql> USE DELIVERY_DATA;
Database changed
mysql> SHOW TABLES;
+-----------------------------+
| Tables_in_delivery_data     |
+-----------------------------+
| birth_order_criteria        |
| residence_criteria          |
| stay_duration_criteria_private |
| stay_duration_criteria_public |
| wealth_quintile_criteria    |
+-----------------------------+
5 rows in set (0.03 sec)

mysql> SELECT * FROM BIRTH_ORDER_CRITERIA;
+-------------+----------------------------+-------------------------------+
| BIRTH_ORDER | NORMAL_DELIVERY_PERCENTAGE | C_SECTION_DELIVERY_PERCENTAGE |
+-------------+----------------------------+-------------------------------+
| 1           |                       74.6 |                          28.1 |
| 2-3         |                       77.1 |                            20 |
| 4-5         |                       76.2 |                           5.9 |
| 6 OR MORE   |                       74.5 |                             4 |
+-------------+----------------------------+-------------------------------+
4 rows in set (0.00 sec)

mysql> SELECT * FROM RESIDENCE_CRITERIA;
+-----------+----------------------------+-------------------------------+
| RESIDENCE | NORMAL_DELIVERY_PERCENTAGE | C_SECTION_DELIVERY_PERCENTAGE |
+-----------+----------------------------+-------------------------------+
| URBAN     |                       74.5 |                          32.3 |
| RURAL     |                       76.5 |                          17.6 |
+-----------+----------------------------+-------------------------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM STAY_DURATION_CRITERIA_PRIVATE;
+------------------+----------------------------+-------------------------------+
| TYPE_OF_DELIVERY | NORMAL_DELIVERY_PERCENTAGE | C_SECTION_DELIVERY_PERCENTAGE |
+------------------+----------------------------+-------------------------------+
| <6 HRS           |                       11.6 |                           3.3 |
| 6-11 HRS         |                        3.2 |                           1.1 |
| 12-23 HRS        |                        2.8 |                           0.1 |
| 1-2 DAYS         |                       45.7 |                           5.5 |
| 3 OR MORE DAYS   |                       36.6 |                          89.8 |
```



MySQL 8.0 Command Line Client

```
2 rows in set (0.00 sec)

mysql> SELECT * FROM STAY_DURATION_CRITERIA_PRIVATE;
+------------------+----------------------------+-------------------------------+
| TYPE_OF_DELIVERY | NORMAL_DELIVERY_PERCENTAGE | C_SECTION_DELIVERY_PERCENTAGE |
+------------------+----------------------------+-------------------------------+
| <6 HRS           |                       11.6 |                           3.3 |
| 6-11 HRS         |                        3.2 |                           1.1 |
| 12-23 HRS        |                        2.8 |                           0.1 |
| 1-2 DAYS         |                       45.7 |                           5.5 |
| 3 OR MORE DAYS   |                       36.6 |                          89.8 |
+------------------+----------------------------+-------------------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM STAY_DURATION_CRITERIA_PUBLIC;
+------------------+----------------------------+-------------------------------+
| TYPE_OF_DELIVERY | NORMAL_DELIVERY_PERCENTAGE | C_SECTION_DELIVERY_PERCENTAGE |
+------------------+----------------------------+-------------------------------+
| <6 HRS           |                       13.8 |                           4.9 |
| 6-11 HRS         |                          4 |                           1.5 |
| 12-23 HRS        |                        2.6 |                           0.4 |
| 1-2 DAYS         |                       39.6 |                           7.4 |
| 3 OR MORE DAYS   |                       39.7 |                          85.4 |
+------------------+----------------------------+-------------------------------+
5 rows in set (0.00 sec)

mysql> SELECT * FROM WEALTH_QUINTILE_CRITERIA;
+-----------------+----------------------------+-------------------------------+
| WEALTH_QUINTILE | NORMAL_DELIVERY_PERCENTAGE | C_SECTION_DELIVERY_PERCENTAGE |
+-----------------+----------------------------+-------------------------------+
| LOWEST          |                       77.3 |                           7.3 |
| SECOND          |                       75.7 |                            15 |
| MIDDLE          |                       77.1 |                          23.9 |
| FOURTH          |                         76 |                          30.4 |
| HIGHEST         |                       72.9 |                          39.1 |
+-----------------+----------------------------+-------------------------------+
5 rows in set (0.00 sec)

mysql>
```



**Normal Delivery vs. C-Section - Comparative Analysis**

Source: National Family Health Survey Report, 2019–21

**Criteria Description**

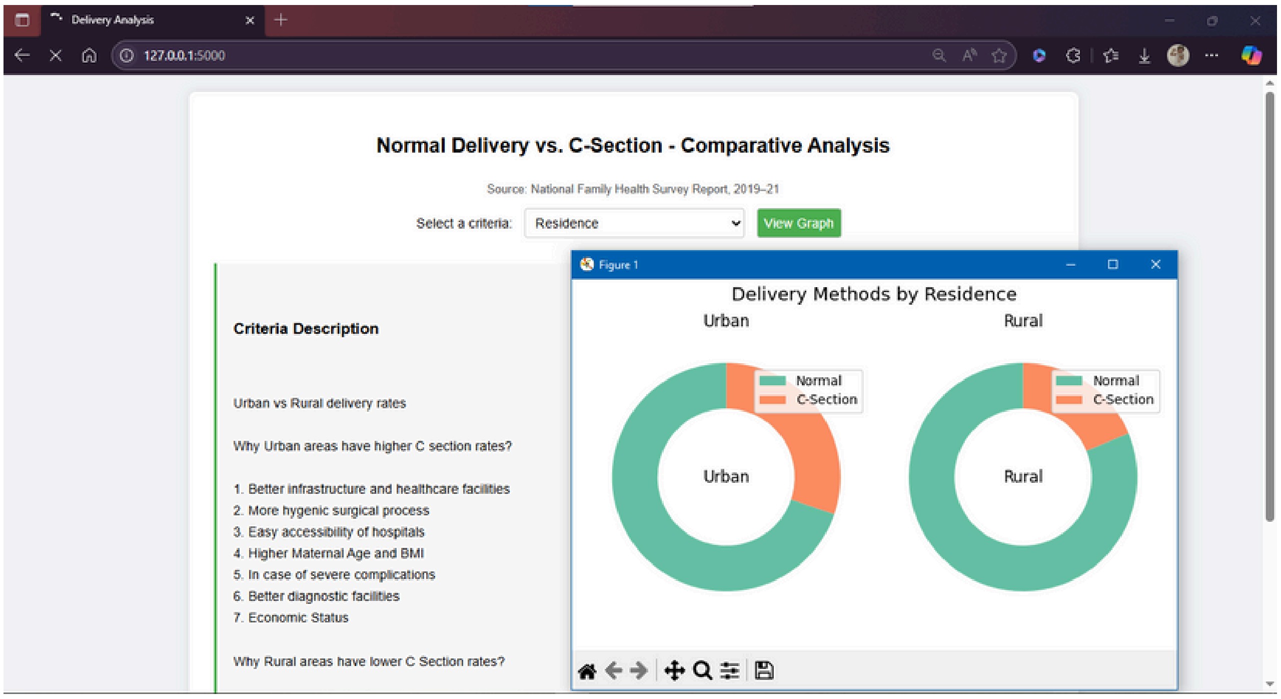
Is Multiple C sections safer?
C section deliveries are safer but sometimes it may lead to seri

Here are few risks involved with Multiple C Sections:

1. Adhesions (scar tissue): These can form between organs, ma
2. Bladder and bowel injuries: Scar tissue can obscure anatomy
3. Excessive bleeding: Repeat surgeries may lead to heavier bl
4. Longer recovery time: Healing may take longer with each sut
5. Placenta accreta: The placenta attaches too deeply into the
hysterectomy.
6. Hernias: Repeated incisions can weaken abdominal muscles, increasing the risk of hernia formation.
7. Risk of infection at incision site

# OUTPUT SCREENSHOTS

**.....CONTINUED**



**Normal Delivery vs. C-Section - Comparative Analysis**

Source: National Family Health Survey Report, 2019–21

Select a criteria: Residence View Graph

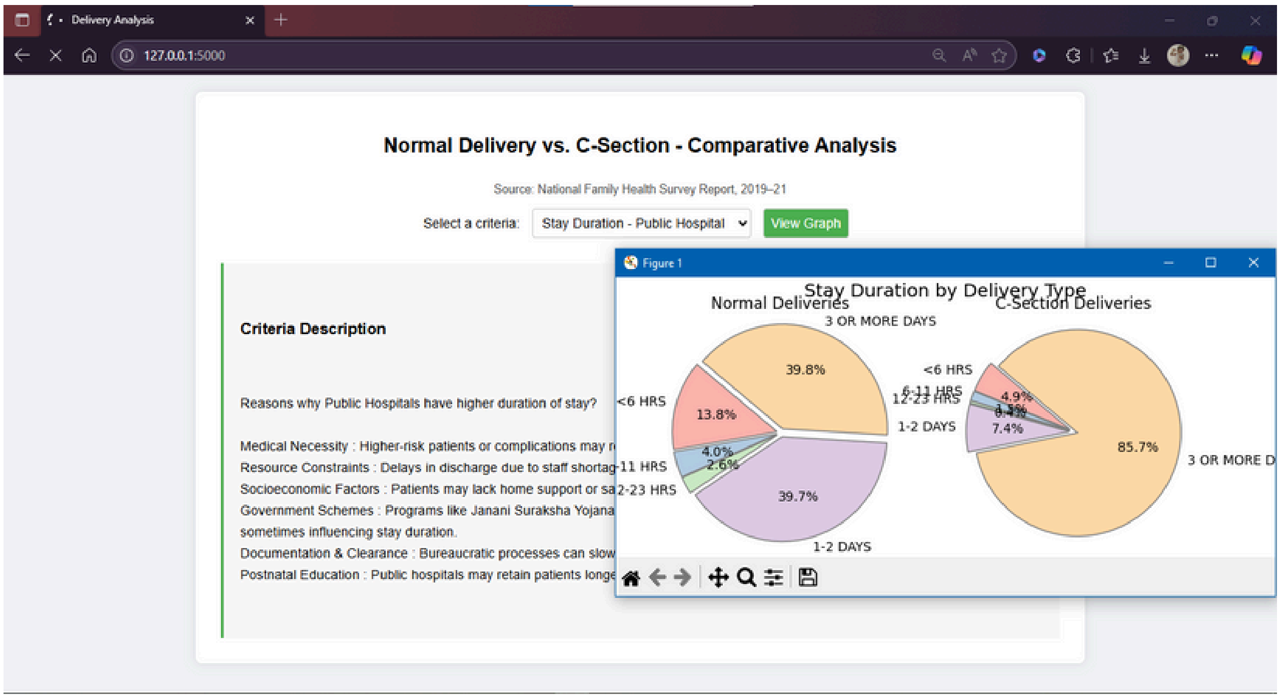
**Criteria Description**

Urban vs Rural delivery rates

Why Urban areas have higher C section rates?

1. Better infrastructure and healthcare facilities
2. More hygenic surgical process
3. Easy accessibility of hospitals
4. Higher Maternal Age and BMI
5. In case of severe complications
6. Better diagnostic facilities
7. Economic Status

Why Rural areas have lower C Section rates?



**Normal Delivery vs. C-Section - Comparative Analysis**

Source: National Family Health Survey Report, 2019–21

Select a criteria: Stay Duration - Public Hospital View Graph

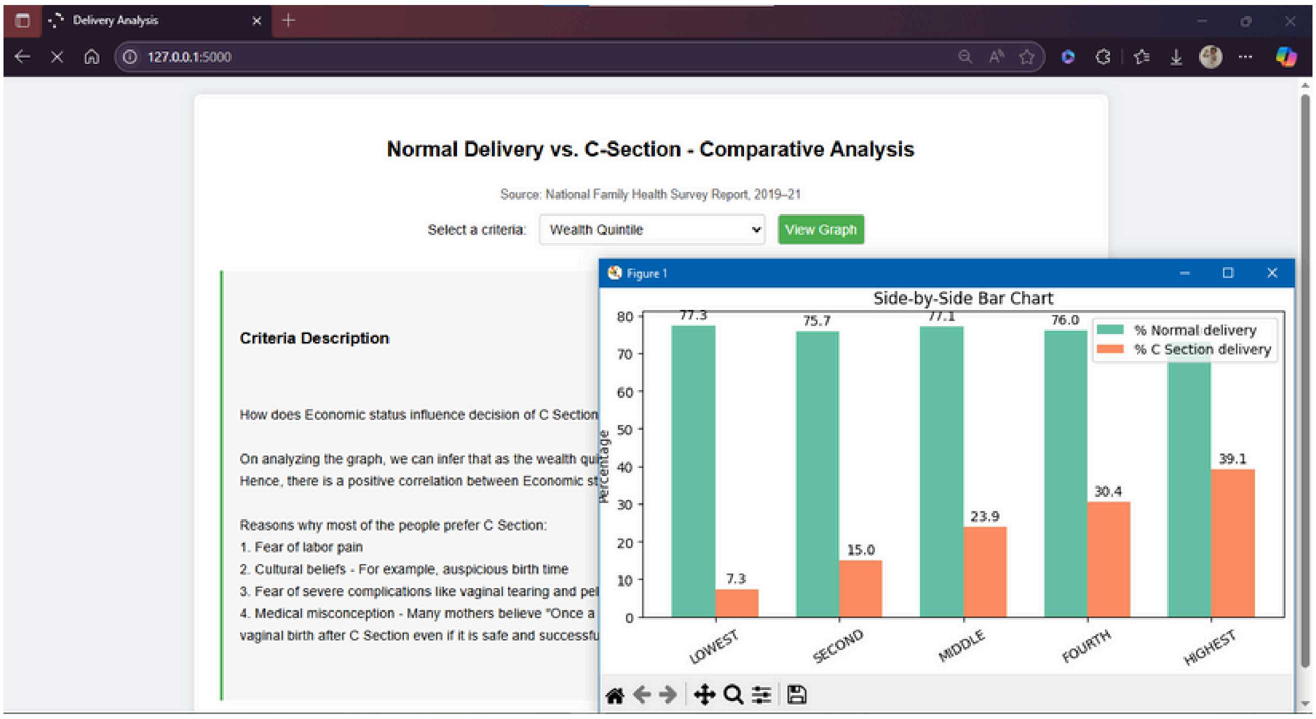
**Criteria Description**

Reasons why Public Hospitals have higher duration of stay?

Medical Necessity : Higher-risk patients or complications may r
Resource Constraints : Delays in discharge due to staff shortag
Socioeconomic Factors : Patients may lack home support or sa
Government Schemes : Programs like Janani Suraksha Yojana
sometimes influencing stay duration.
Documentation & Clearance : Bureaucratic processes can slow
Postnatal Education : Public hospitals may retain patients longe

# OUTPUT SCREENSHOTS

**.....CONTINUED**



**Normal Delivery vs. C-Section - Comparative Analysis**

Source: National Family Health Survey Report, 2019–21

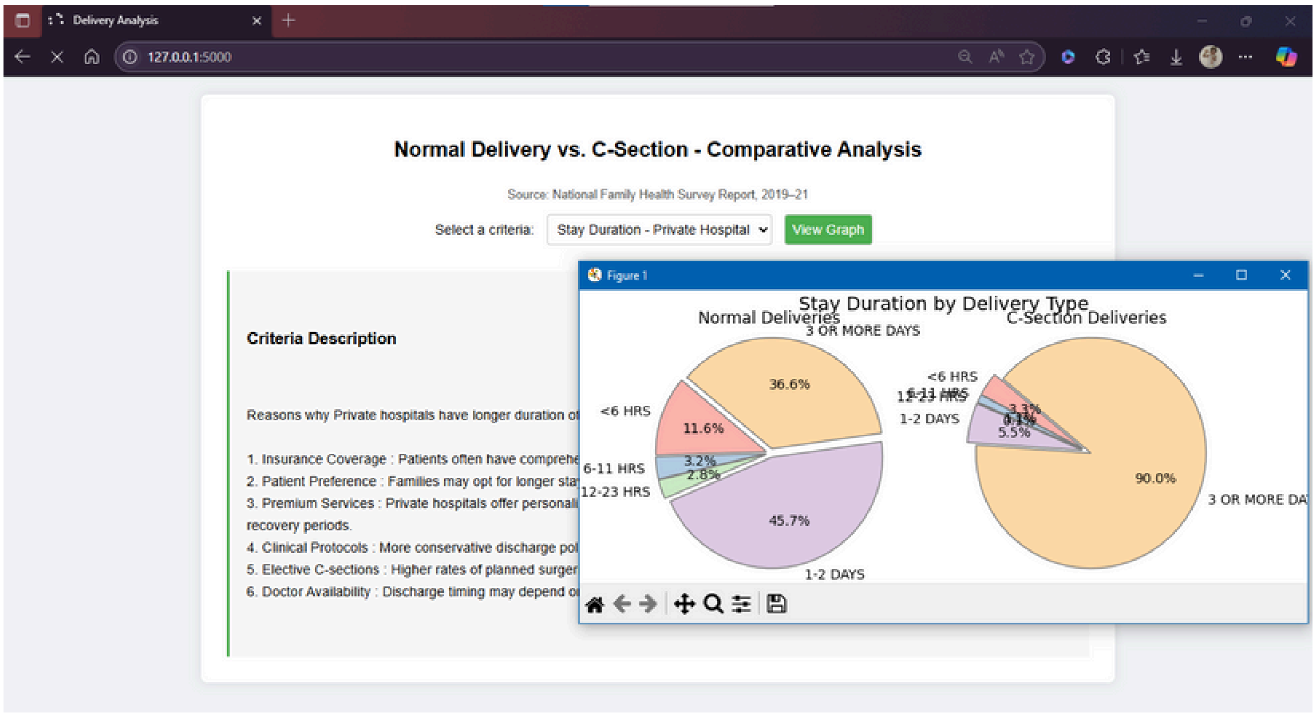
Select a criteria: Wealth Quintile  View Graph

**Criteria Description**

How does Economic status influence decision of C Section

On analyzing the graph, we can infer that as the wealth qu
Hence, there is a positive correlation between Economic st

Reasons why most of the people prefer C Section:
1. Fear of labor pain
2. Cultural beliefs - For example, auspicious birth time
3. Fear of severe complications like vaginal tearing and pel
4. Medical misconception - Many mothers believe "Once a
vaginal birth after C Section even if it is safe and successfu



**Normal Delivery vs. C-Section - Comparative Analysis**

Source: National Family Health Survey Report, 2019–21

Select a criteria: Stay Duration - Private Hospital  View Graph

**Criteria Description**

Reasons why Private hospitals have longer duration of

1. Insurance Coverage : Patients often have comprehe
2. Patient Preference : Families may opt for longer sta
3. Premium Services : Private hospitals offer personali
recovery periods.
4. Clinical Protocols : More conservative discharge pol
5. Elective C-sections : Higher rates of planned surger
6. Doctor Availability : Discharge timing may depend o

## CLOSURE

THIS PROJECT OFFERS A COMPREHENSIVE COMPARISON BETWEEN NORMAL AND C-SECTION DELIVERIES THROUGH DATA-DRIVEN VISUALS AND INSIGHTS. LEVERAGING REAL-WORLD STATISTICS FROM THE 5TH NATIONAL FAMILY HEALTH SURVEY, IT SHOWCASES REGIONAL, ECONOMIC, AND MEDICAL INFLUENCES ON DELIVERY METHOD PREFERENCES. BY MODULARIZING THE IMPLEMENTATION ACROSS SQL, VISUALIZATION, AND FLASK-BASED UI, IT HIGHLIGHTS BOTH THE TECHNICAL FEASIBILITY AND THE SOCIAL RELEVANCE OF HEALTH-BASED ANALYTICS. THE INSIGHTS PRESENTED AIM TO INFORM BETTER MATERNAL HEALTHCARE DECISIONS THROUGH CLEAR VISUAL STORYTELLING.

THIS PROJECT CAN BE EXTENDED BY INTRODUCING SOME MORE UNIQUE PARAMETERS LIKE MOTHER’S AGE AT DELIVERY TIME, DELIVERY BY SKILLED/UNSKILLED PERSON, ETC. ALSO THIS CAN BE CONVERTED INTO AN INTERACTIVE APP WITH MORE INFORMATION AND FEATURES

## BIBLIOGRAPHY

- NATIONAL FAMILY HEALTH SURVEY (NFHS-5), 2019–21, MINISTRY OF HEALTH AND FAMILY WELFARE, GOVERNMENT OF INDIA HTTPS://WWW.RCHIIPS.ORG/NFHS/NFHS-5REPORTS/INDIA.PDF
- MYSQL 8.0 DOCUMENTATION HTTPS://DEV.MYSQL.COM/DOC/
- MATPLOTLIB LIBRARY DOCUMENTATION HTTPS://MATPLOTLIB.ORG/STABLE/CONTENTS.HTML
- FLASK WEB FRAMEWORK DOCUMENTATION HTTPS://FLASK.PALLETSPROJECTS.COM/EN/LATEST/

# Thank You