Experiment 3: Implement Exit Rate web Metric.

🔍 Aim

To write a Python program that simulates user session exits from different web pages and calculates the exit rate (in percentage) for the `signup` and `subscribe` pages based on session data.

🎯 Objective

- To understand the concept of **exit rate** as a key web analytics metric.
- To simulate session data representing user exits from different pages.
- To calculate:
    - Total number of sessions
    - Number of exits at each page
    - Exit rate percentage for each page
- To demonstrate clear separation of logic into input, processing, and output phases without involving UI or database implementation.

Pseudocode

```
from flask import Flask, session, redirect
import sqlite3
import uuid

app = Flask(__name__)
app.secret_key = 'exit-metrics-demo-key'

# ---------- Database Setup ----------
def init_db():
    # DB CALL: connect to database
    # DB CALL: create 'sessions' table with columns 'id' and 'exit_page'
    pass

init_db()

# ---------- Helper ----------
def log_exit(session_id, page):
    # DB CALL: insert or update session ID and exit page in the database
    pass

# ---------- Routes ----------
@app.route('/')
def home():
    # Clear session
```

```
    session.clear()

    # UI/HTML: Return homepage with link to signup and metrics
    return '''

    '''

@app.route('/signup')
def signup():
    # Generate unique session ID
    session['session_id'] = str(uuid.uuid4())

    # UI/HTML: Show signup form with "Exit Without Signing Up" link
    return '''

    '''

@app.route('/subscribe', methods=['POST'])
def subscribe():
    session_id = session.get('session_id')

    if session_id:
        # DB CALL: log exit with page = 'subscribe'
        log_exit(session_id, 'subscribe')
        session.clear()

    # UI/HTML: Show thank you message after subscribing
    return '''

    '''

@app.route('/exit/<page>')
def exit_page(page):
    session_id = session.get('session_id')

    if session_id:
        # DB CALL: log exit with page = <page>
        log_exit(session_id, page)
        session.clear()

    return redirect('/')

@app.route('/metrics')
def metrics():
    # DB CALL: count total number of sessions
    total = 0

    # DB CALL: count exits from signup page
```

```
    exit_signup = 0

    # DB CALL: count exits from subscribe page
    exit_subscribe = 0

    # CALCULATION: Calculate percentage exit rates
    percent_signup = (exit_signup / total) * 100
    percent_subscribe = (exit_subscribe / total) * 100

    # UI/HTML: Display exit counts and percentages
    return f'''

    '''

# ---------- Run ----------
if __name__ == '__main__':
    app.run(debug=True)


# ---------- End of Program ----------
```

Actual code with UI and DB,

```
from flask import Flask, session, redirect
import sqlite3
import uuid

app = Flask(__name__)
app.secret_key = 'exit-metrics-demo-key'

# ---------- Database Setup ----------
def init_db():
    conn = sqlite3.connect('sessions.db')
    c = conn.cursor()
    c.execute('''
        CREATE TABLE IF NOT EXISTS sessions (
            id TEXT PRIMARY KEY,
            exit_page TEXT
        )
    ''')
    conn.commit()
    conn.close()

init_db()

# ---------- Helper ----------
```

```
def log_exit(session_id, page):
    conn = sqlite3.connect('sessions.db')
    c = conn.cursor()
    c.execute("INSERT OR REPLACE INTO sessions (id, exit_page) VALUES (?, ?)",
              (session_id, page))
    conn.commit()
    conn.close()

# ---------- Routes ----------
@app.route('/')
def home():
    # Clear any session, no logging at home
    session.clear()
    return '''
    <h1>Welcome to Our Newsletter</h1>
    <p><a href="/signup">Sign Up</a></p>
    <p><a href="/metrics">View Exit Metrics</a></p>
    '''

@app.route('/signup')
def signup():
    session['session_id'] = str(uuid.uuid4())
    return '''
    <h1>Newsletter Signup</h1>
    <form action="/subscribe" method="POST">
        <input type="email" name="email" required placeholder="Your email">
        <button type="submit">Subscribe</button>
    </form>
    <p><a href="/exit/signup">Exit Without Signing Up</a></p>
    '''

@app.route('/subscribe', methods=['POST'])
def subscribe():
    session_id = session.get('session_id')
    if session_id:
        log_exit(session_id, 'subscribe')
        session.clear()
    return '''
    <h1>Thank You!</h1>
    <p>You've successfully subscribed.</p>
    <p><a href="/">Return Home</a></p>
    '''

@app.route('/exit/<page>')
def exit_page(page):
    session_id = session.get('session_id')
    if session_id:
        log_exit(session_id, page)
```

```
        session.clear()
    return redirect('/')

@app.route('/metrics')
def metrics():
    conn = sqlite3.connect('sessions.db')
    c = conn.cursor()

    # Total sessions recorded
    c.execute("SELECT COUNT(*) FROM sessions")
    total = c.fetchone()[0] or 1  # avoid division by zero

    # Exit counts per page
    c.execute("SELECT COUNT(*) FROM sessions WHERE exit_page = 'signup'")
    exit_signup = c.fetchone()[0]

    c.execute("SELECT COUNT(*) FROM sessions WHERE exit_page = 'subscribe'")
    exit_subscribe = c.fetchone()[0]

    conn.close()

    # Percentages
    percent_signup = (exit_signup / total) * 100
    percent_subscribe = (exit_subscribe / total) * 100

    return f'''
    <h1>Exit Page Metrics</h1>
    <p>Total Sessions Recorded: {total}</p>

    <h2>Exit Rate Details</h2>
    <ul>
        <li>Signup Page: {exit_signup} exits ({percent_signup:.2f}%)</li>
        <li>Subscribe Page: {exit_subscribe} exits ({percent_subscribe:.2f}%)</li>
    </ul>

    <p><a href="/">Return Home</a></p>
    '''

# ---------- Run ----------
if __name__ == '__main__':
    app.run(debug=True)
```