**Single-Server Queuing Model in Mathematics**

The **Single-Server Queuing Model** is a mathematical model used to describe systems where a single server processes customers (or jobs) one at a time. This model is particularly useful in scenarios such as banks, restaurants, and support centres where customers arrive and wait for service in a queue.

Mathematically, the model is characterized by the following:

* **Arrival rate (λ)**: The rate at which customers arrive in the queue. It's typically modelled as a Poisson process, meaning that the time between customer arrivals follows an exponential distribution.
* **Service rate (μ)**: The rate at which the server can serve customers. This is also modelled as an exponential distribution, where the service time is random but follows a certain average.
* **Queue Length (L)**: The number of customers in the queue, including the one being served.
* **Utilization (ρ)**: The fraction of time the server is busy. It is calculated as ρ = λ / μ.

The single-server queuing model follows the notation **M/M/1**, where:

* The first **M** stands for Markovian (memoryless) arrival process (Poisson process).
* The second **M** stands for the memoryless service process (exponentially distributed service times).
* **1** indicates a single server.

This model is widely used in operations research and performance analysis to optimize the design of service systems.

**1. Introduction to the HTML Project: Bakery Queue Simulation**

The **Bakery Queue Simulation** is an interactive web-based application developed using HTML, CSS, and JavaScript. The project simulates a queue system where customers arrive at a bakery, wait in line, and are served by a baker one by one. This simulation represents a **Single-Server Queuing Model**.

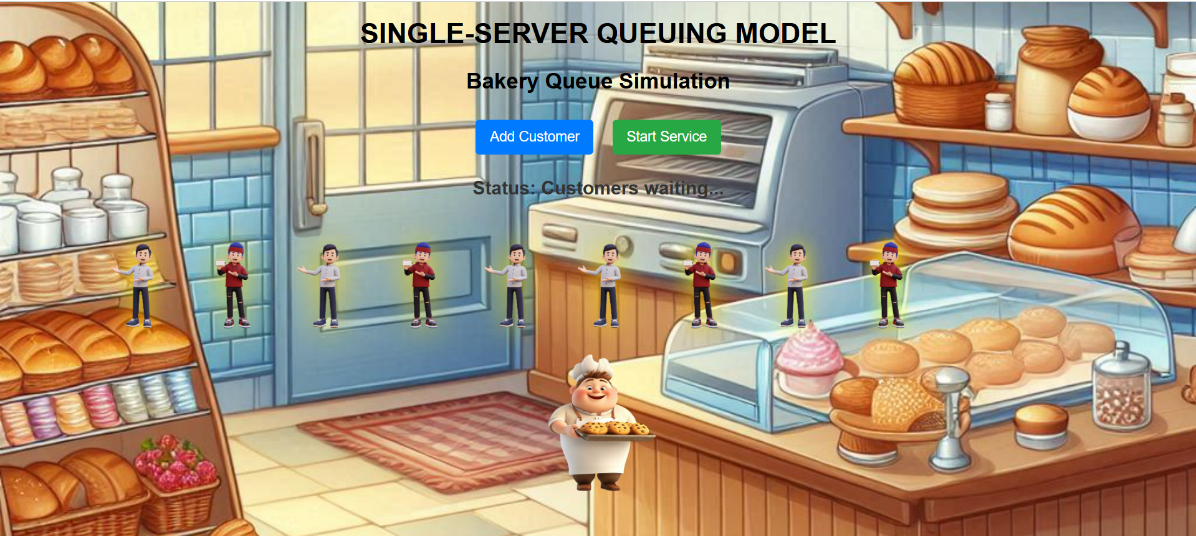
The system has two main functions:

* **Customer Arrival**: Customers arrive at the bakery and stand in a queue, awaiting service.
* **Customer Service**: The server (baker) serves the customers one by one. When a customer is served, a visual effect (fireworks) is triggered to represent successful service.



**a) Adding a customer**

* When the "Add Customer" button is clicked, a new customer (represented by an image) is added to the queue.
* Each customer is animated into the queue, moving from the left to their assigned position.



**b) Serving a customer**

* When the "Start Service" button is clicked, the first customer in the queue is selected to be served.
* The customer is animated to a larger size to indicate they are being served, and the queue moves forward.
* After serving, a firework animation is triggered, and the customer disappears from the queue.



**2. Project Flow**

1. **No Customers in Queue**: The bakery is initially empty, with the status "Waiting for customers."
2. **Customers Arriving**: Customers start arriving when the "Add Customer" button is clicked. They wait in line until the server is ready to serve them.
3. **Customer Service**: The server starts serving customers one by one. Each time a customer is served, a firework effect is displayed to mark the event.
4. **Queue Updates**: The status updates accordingly to inform if there are customers in the queue or if there are no customers left to serve.

**3. Code Overview**

The code consists of:

* **HTML Structure**: Basic layout, including buttons for adding customers and starting service, and containers for displaying the queue and server.
* **CSS Styling**: Custom styles for the bakery background, customer appearance, server image, and animations (e.g., glowing customers, bouncing effects).
* **JavaScript Logic**: Handles customer addition to the queue, queue movement, customer service, and fireworks effect when a customer is served.

**4. Conclusion**

The **Bakery Queue Simulation** demonstrates the implementation of the **Single-Server Queuing Model** using basic web technologies. By simulating customer arrivals and services in a queue, this project provides insight into how queuing systems work in real-world scenarios, such as bakeries, hospitals, and service centers. The interactive and animated features of the simulation enhance user engagement and understanding of the queuing process.