Web Based Queuing Simulator

Contents

1. Introduction:

1.1 What is Queuing Theory:

Queuing theory is a branch of mathematics that studies how lines form, how they function, and why they malfunction. Queuing theory examines every component of waiting in line, including the arrival process, service process, number of servers, number of system places, and the number of customers—which might be people, data packets, cars, or anything else.

1.2 Project Objectives

1.2.1 Business Objectives

We are targeting the outfit industry, entrepreneurs, and a common user. Target will greatly profit from our application; it will be especially useful to Super markets since they will have a platform to market their brands, allowing them to build their businesses and help them in increasing their sales and reduce peoples in queue fastly.

1.2.2 Technology Objectives

We will be using flutter technology for our Website frontend and making it cross-platform. For the backend of our application, we used a dart programming language.

1.3 Project Vision

To develop a Web Based Queuing Simulator that is immersive ,user-friendly and easy to use for the users to help specially for those patients who are suffering from severe pain , injury etc and needs immediate treatment they can use this this web app to look which Hospital have less no of patients or have fast service. Similarly , It will also provide the plate- form for the new entrepreneurs and brands to expand their business and increase their sell. It will also help the user to finding best grocery mart where more peoples come and serves fastly.

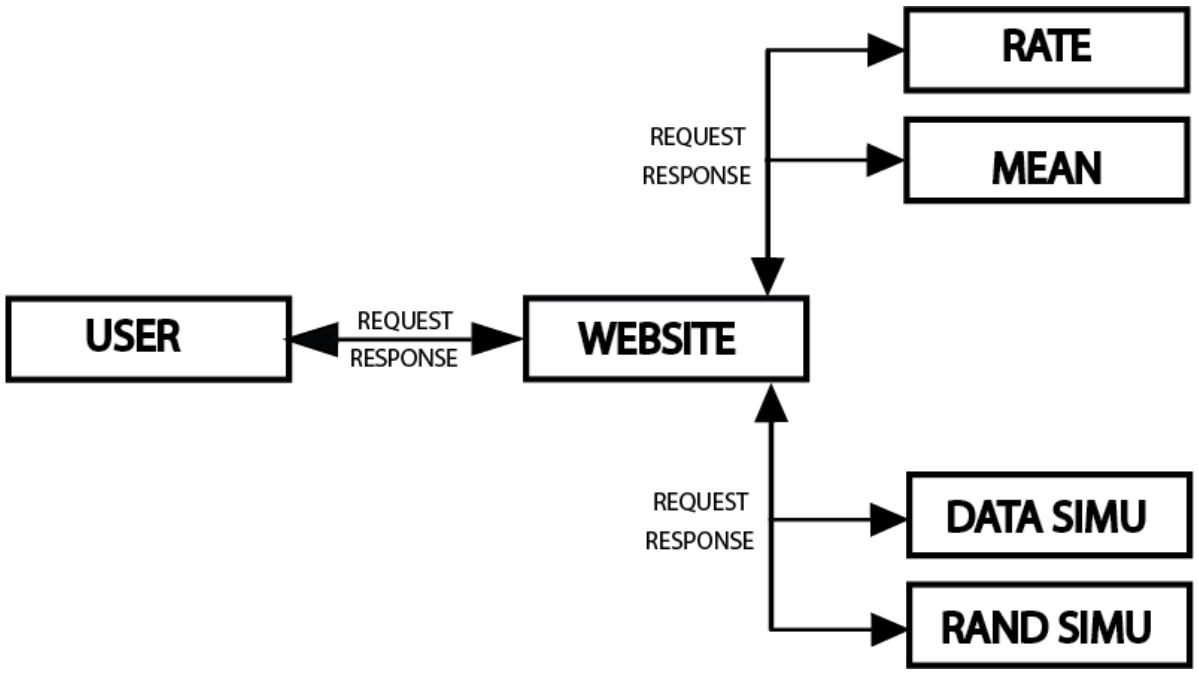
1.4 Project Scope

This website will provide business opportunities to normal user , entrepreneurs and grocery mart brands to enhance their sell. A visitor using this website always know that how much average time at instant a customer pay in mart for shopping, for billing and much more. Mart management can increase their mart's service time , increase or decrease number of server(s) serving queues , make system like that there customers spent less time and do all pretented grocery.

2. System Design:

2.1 System Architecture

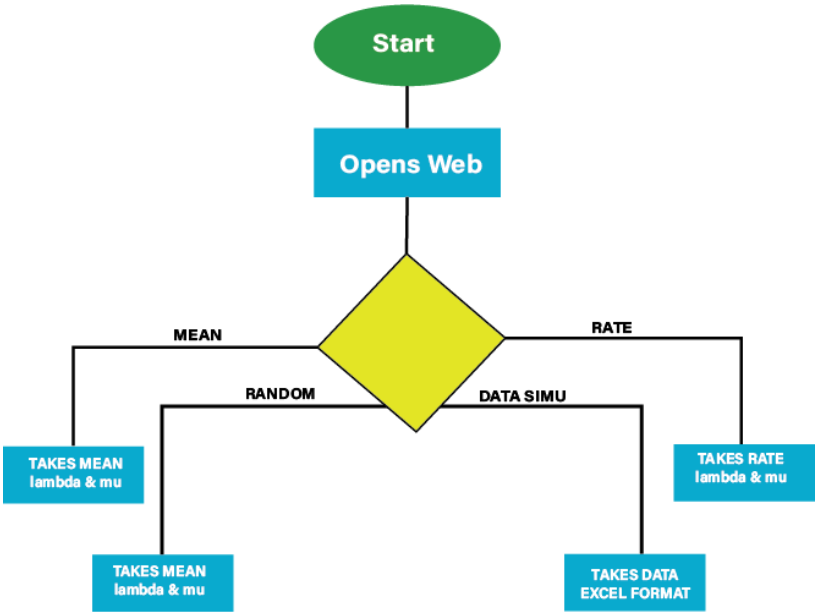
Following figure depicts the website's fundamental design which include four main modules : Rate, Mean, Data Simulation and Random Simulation Methodologies.



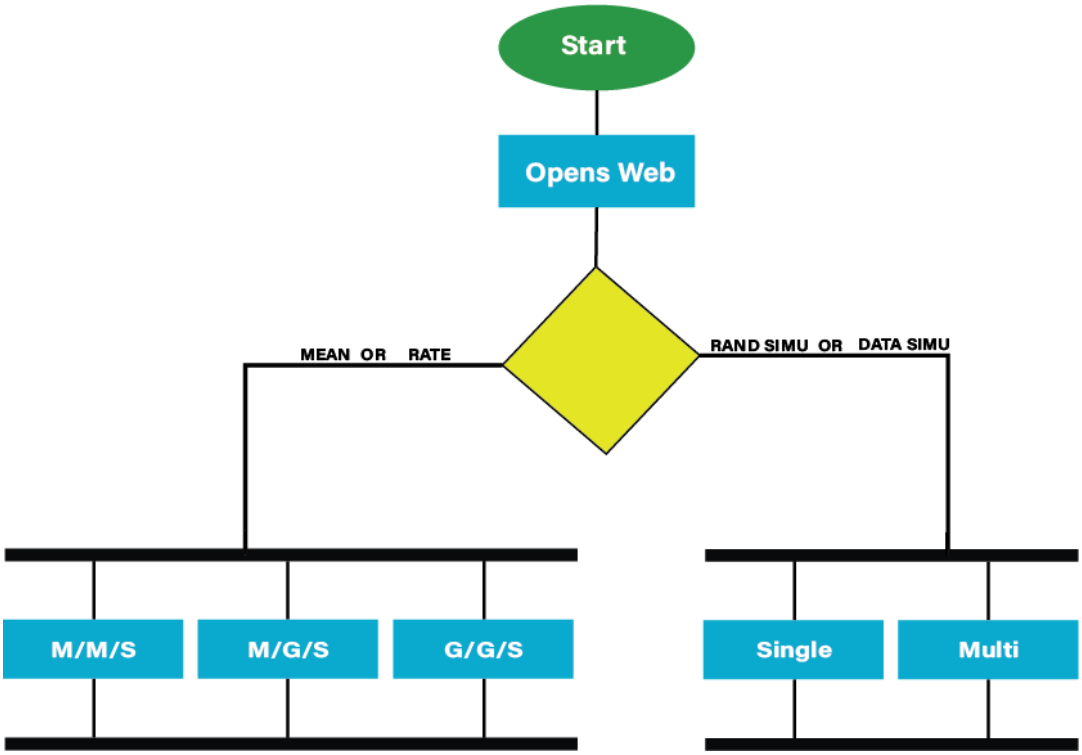
2.2 High Level Design

2.2.1 Data Flow Diagram:

In Terms of inputs and outputs , a dataflow diagram depicts how data is handled by a system. The concentration, as its name suggests, is on the flow of information, namely where it originates from, where it flows, and how it is kept.



2.2.1 Detailed System Diagram:



3. System Implementation:

Implementation is the process of turning a concept into reality. The system implementation is the programming and development of an algorithm into a software component.

3.1 Tools and Technologies

The following tools and technologies were used to create this Web:

- Android Studio
- Vs Code
- Microsoft Excel

3.2 Libraries / Packages

The following libraries and packages are used by our Web:

- cupertino icons: 0.1.2
- google_fonts: 3.0.1
- charts_flutter: 0.12.0

- data_table_2: 2.3.3
- flutter_screenutil: 5.5.3+2
- hovering: 1.0.4
- provider: 6.0.2
- percent_indicator: 4.0.1
- syncfusion_flutter_charts: 20.2.36
- fl_chart: 0.55.0
- count_up: 0.1.4
- odometer: 3.0.0
- http: 0.13.4
- excel: 2.0.0
- universal_io: 2.0.4

3.3 Development Environment / Language Used

Flutter web framework was used to implement the frontend of our website, while Dart was used to implementing the backend and JavaScript was used to write server to convert excel data in API form .

4. System Testing and Evaluation:

4.1 Graphical User Interface Testing

Our website initially starts with split screen in four parts Rate , Mean ,Data Simulation and Random Simulation of Queuing Analysis as shown below:

Figure : First Screen Of Web application

Now we have four options , Rate and Mean methodologies will be used when we know about the average mean and average service OR mean rate and service rate of the considered system , In rate scenario our system takes value of arrival and service as we will give but in case of mean, one will be divide by arrival and service values automatically. In Simulation methodology we will give observed data of individual arrival and service to google excel sheet our web automatically get that data by using API . Similarly, in Random methodology , our website gives us arrival and service of individual customer with the help of backend stochastic process.

On selecting Rate or Mean Method our screen looks like:

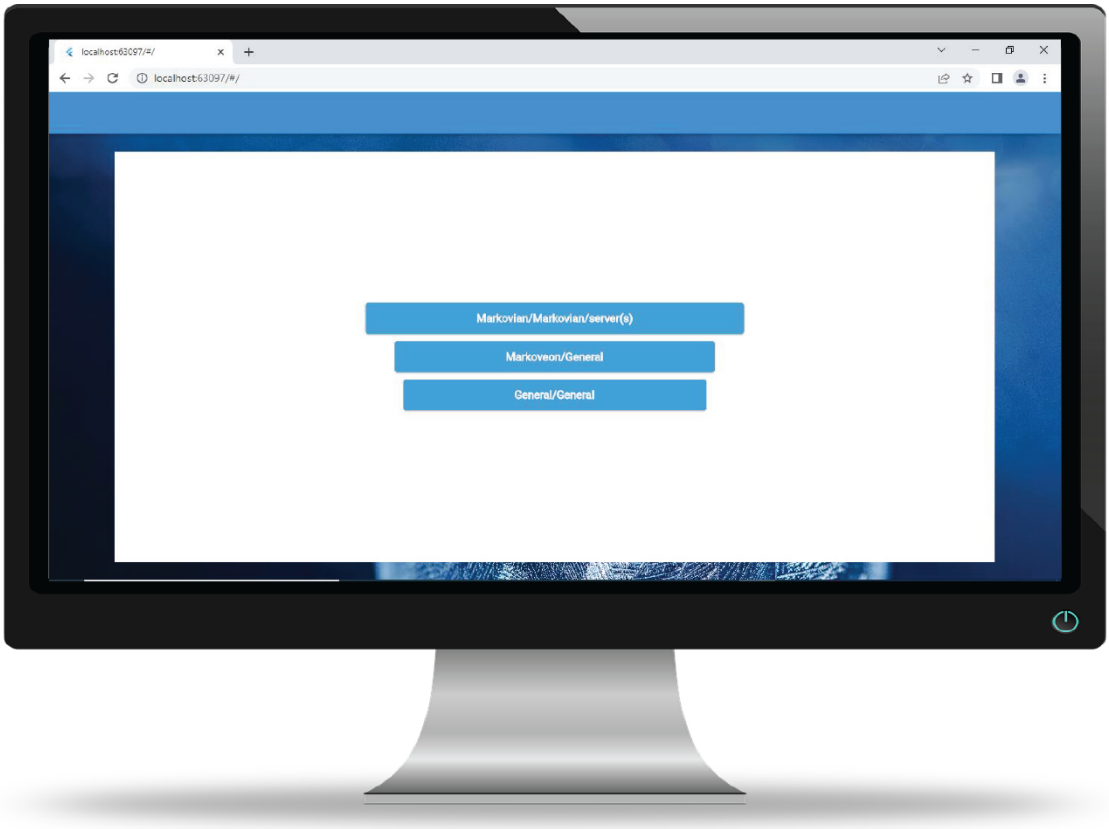
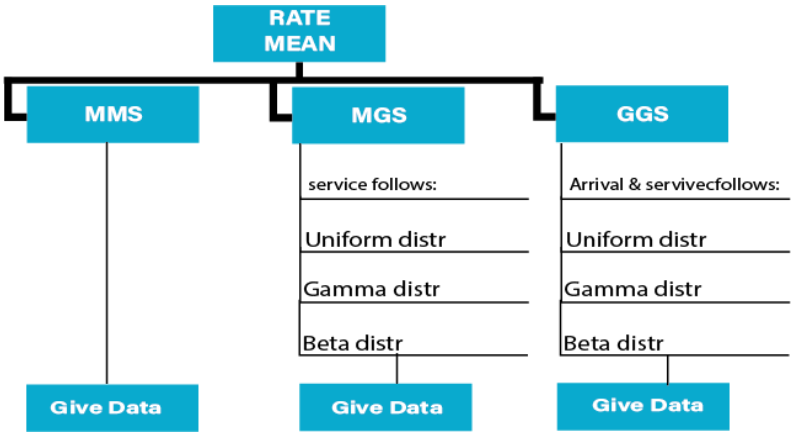


Figure : Rate or Mean method Landing Screen

Now we have to select one option from above three options , if we click on Markoveon/Markoveon it will show us alert box and requires the arrival mean , service mean and number of servers, OR arrival rate, service rate and number of servers while if we click on remaining two options web requires more selection of options for details there is an hierarchy like:



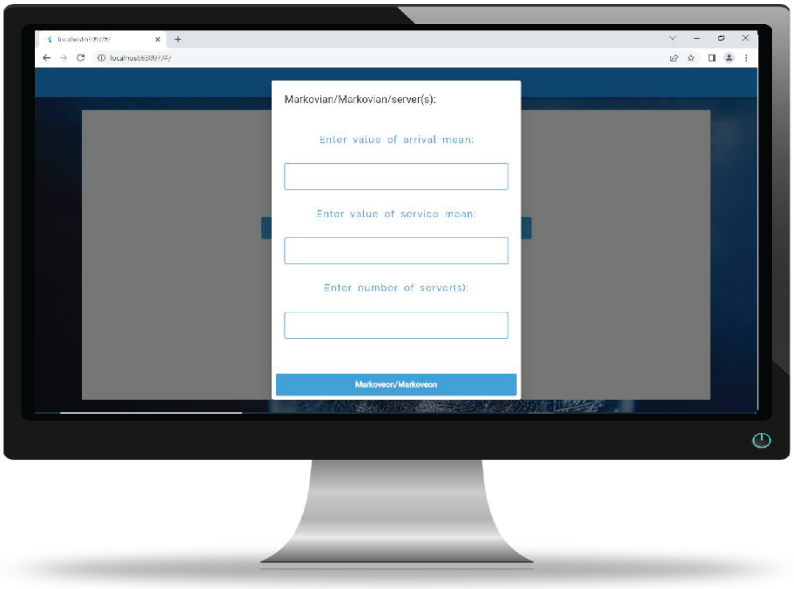


Figure : MMC

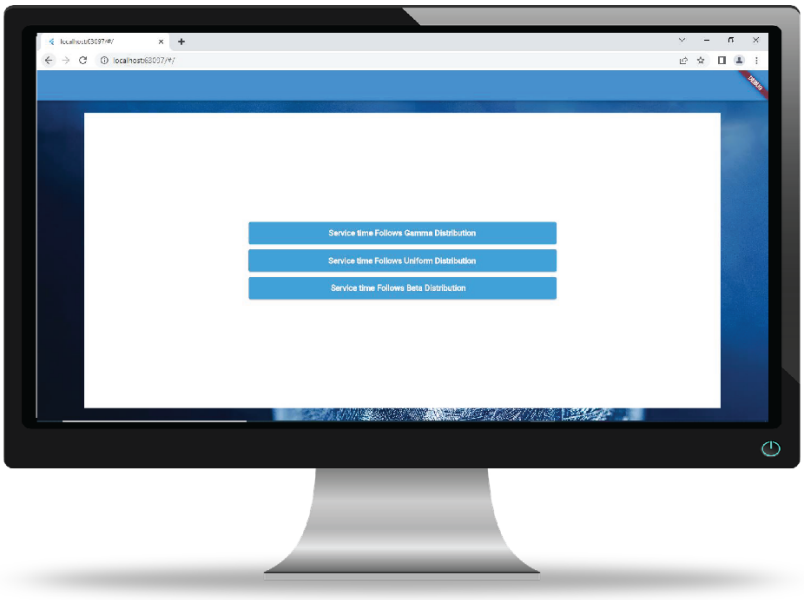


Figure : MGC

Figure : GGC

When we put all data and tell the distributions after clicking simulate button it will give us all statistics of our data like this:

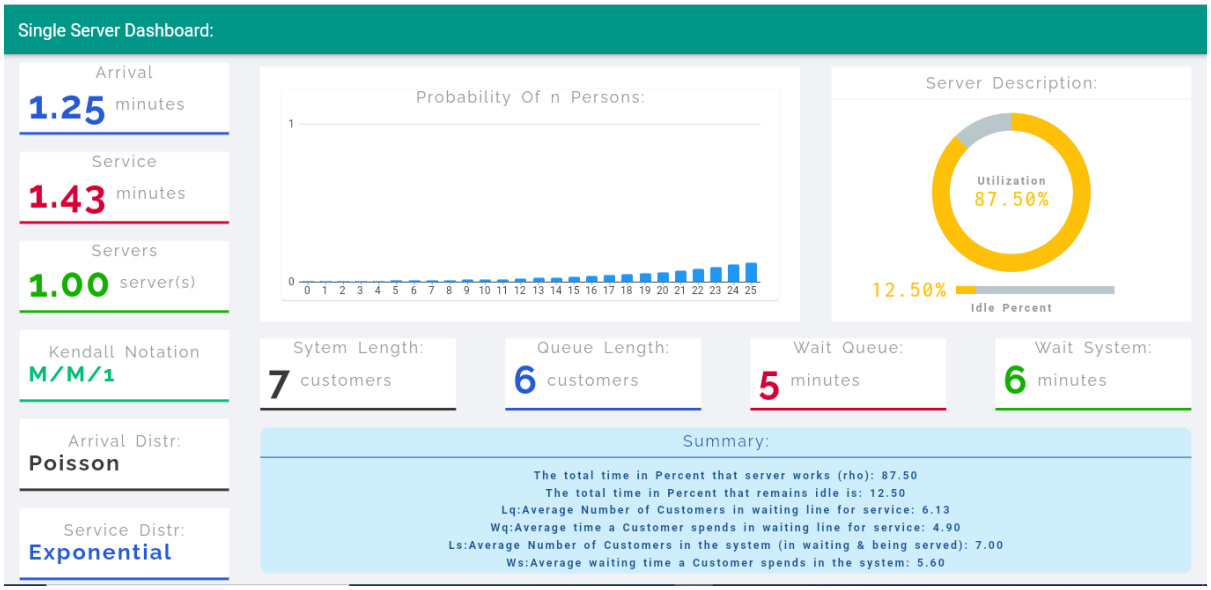


Figure : Single-Server Dashboard

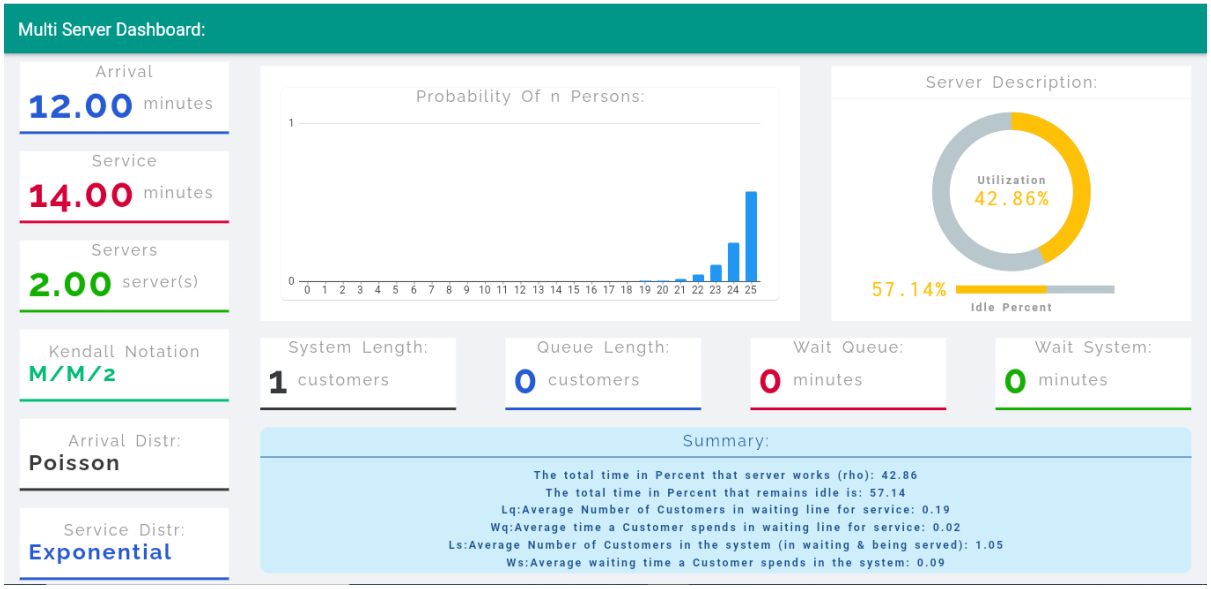


Figure : Multi-Server Dashboard

Here we got all statistics here the histogram is of probability of n persons in system and in this web n = 25 , we get server idle percentage , server utilization percent , system length , queue length , wait queue , wait system and also we got summary of simulation which contains exact value at the bottom of page.

Similarly, when we select Simulation option on first screen of web it will gives us two buttons at the top right of screen like this:

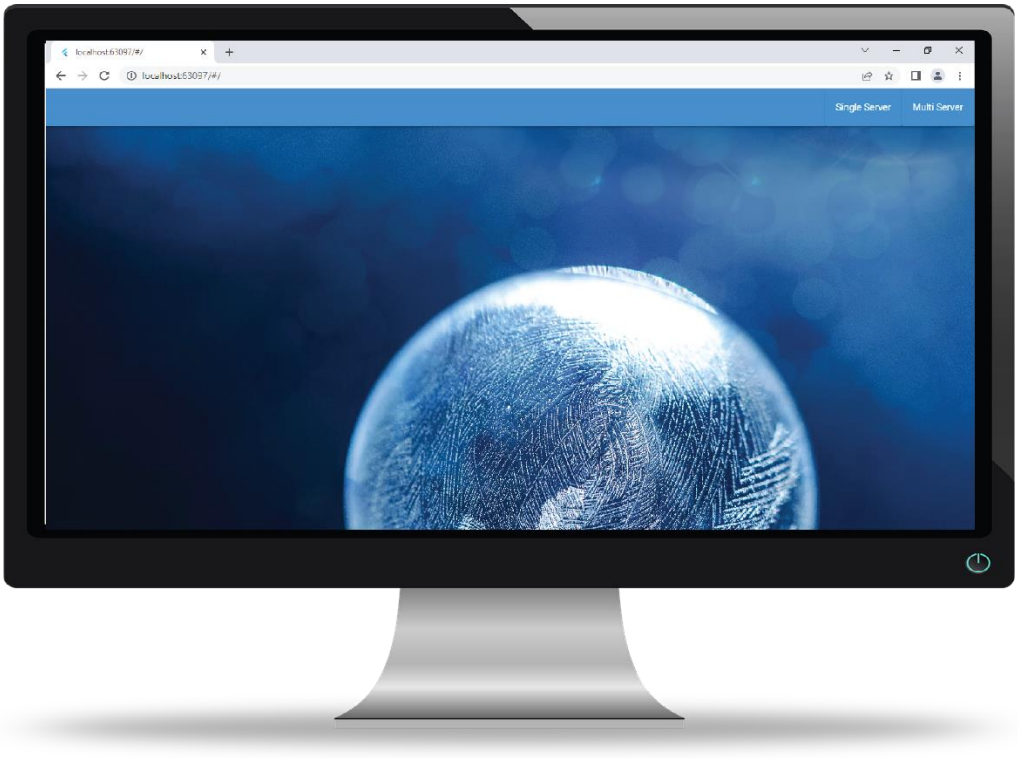


Figure : Simulation Method Screen

We collect data of Imtiaz Super Market , this data we transfer on google Excel sheet and write a JavaScript language based server for converting that excel data in JSON format , that JSON format data will be called through API in our web application.

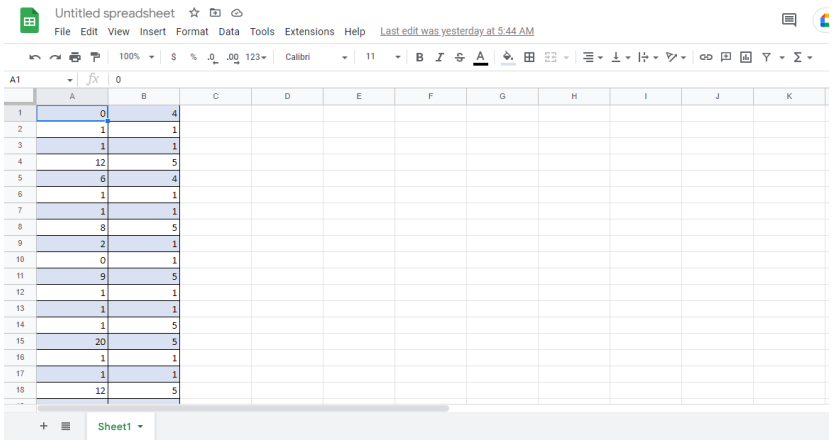


Figure : Data on Excel

```
[{"ArrivalTime":0,"ServiceTime":4},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":12,"ServiceTime":5},{"ArrivalTime":6,"ServiceTime":4},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":8,"ServiceTime":5},{"ArrivalTime":2,"ServiceTime":1},{"ArrivalTime":0,"ServiceTime":1},
{"ArrivalTime":9,"ServiceTime":5},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":5},{"ArrivalTime":20,"ServiceTime":5},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":12,"ServiceTime":5},{"ArrivalTime":1,"ServiceTime":5},{"ArrivalTime":16,"ServiceTime":1},
{"ArrivalTime":7,"ServiceTime":6},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":13,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":6},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":3,"ServiceTime":2},{"ArrivalTime":9,"ServiceTime":2},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":2,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":5,"ServiceTime":3},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":3,"ServiceTime":2},
{"ArrivalTime":8,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":0,"ServiceTime":1},{"ArrivalTime":9,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":3},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":7,"ServiceTime":2},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},
{"ArrivalTime":2,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":12,"ServiceTime":3},{"ArrivalTime":3,"ServiceTime":6},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":4},{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":0,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":8,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":11,"ServiceTime":6},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":2,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":13,"ServiceTime":3},{"ArrivalTime":5,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":5},{"ArrivalTime":2,"ServiceTime":4},
{"ArrivalTime":3,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":8,"ServiceTime":3},{"ArrivalTime":8,"ServiceTime":16},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":5,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":5,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":5},{"ArrivalTime":1,"ServiceTime":4},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":11,"ServiceTime":3},{"ArrivalTime":9,"ServiceTime":5},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":18,"ServiceTime":6},{"ArrivalTime":11,"ServiceTime":4},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":5},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":21,"ServiceTime":6},{"ArrivalTime":6,"ServiceTime":6},{"ArrivalTime":2,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":8,"ServiceTime":2},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":2,"ServiceTime":1},{"ArrivalTime":9,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":7,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":4},
{"ArrivalTime":4,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":2},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":2},{"ArrivalTime":7,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":2},
{"ArrivalTime":14,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":6,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":3},{"ArrivalTime":10,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":4,"ServiceTime":2},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":4},{"ArrivalTime":1,"ServiceTime":1},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":5,"ServiceTime":2},{"ArrivalTime":20,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},
{"ArrivalTime":7,"ServiceTime":4},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":4,"ServiceTime":1},{"ArrivalTime":7,"ServiceTime":2},
{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":3},{"ArrivalTime":4,"ServiceTime":3},{"ArrivalTime":1,"ServiceTime":1},{"ArrivalTime":1,"ServiceTime":1}]
```


Figure : Data in JSON format

When we click on single server button our program simulate the data with server = 1 and shown like this

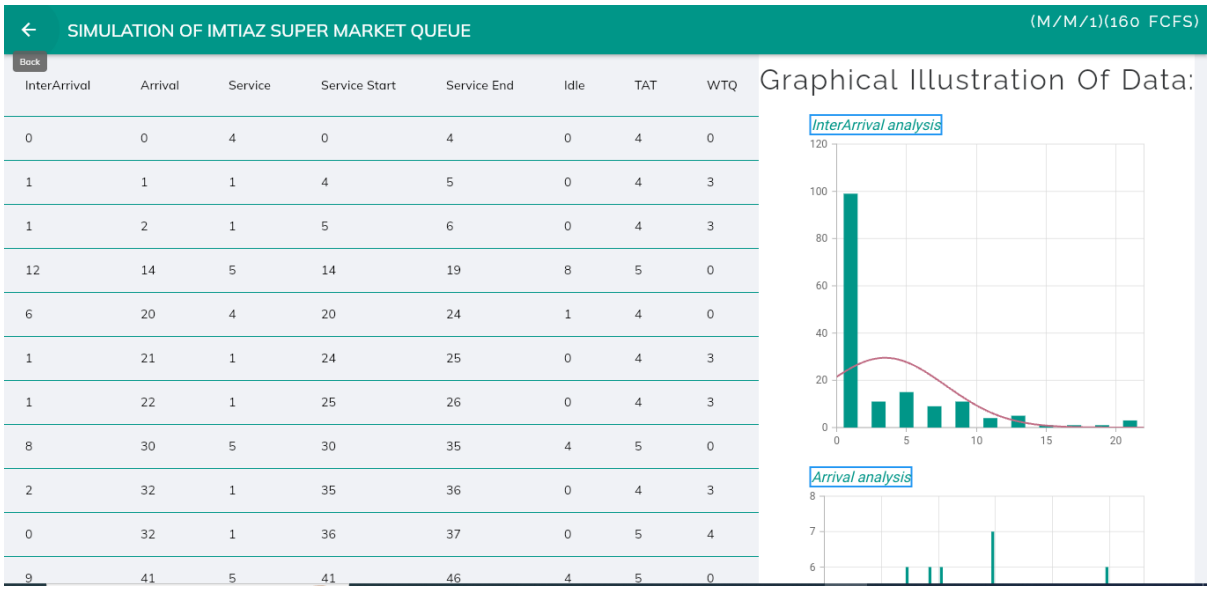
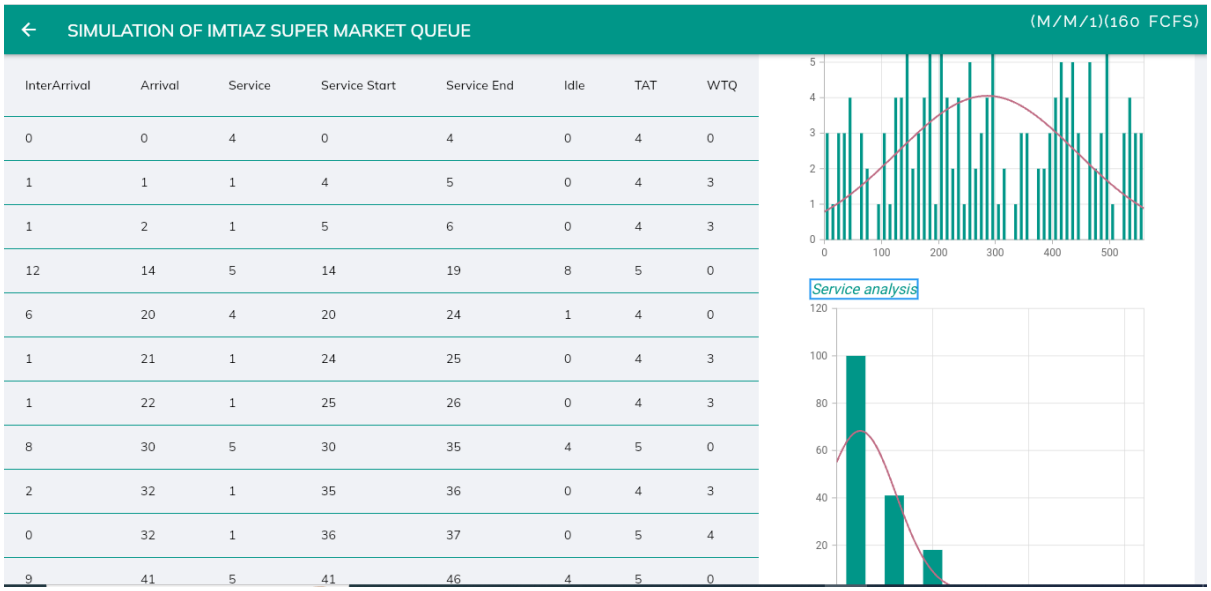
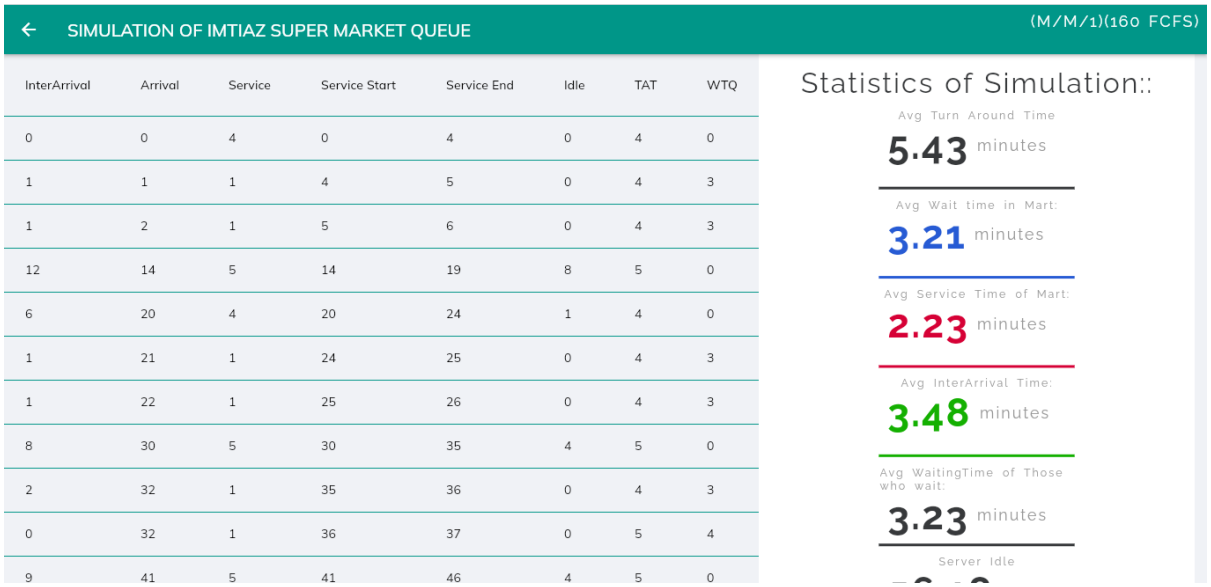


Figure : Simulation of Single-Server



InterArrival	Arrival	Service	Service Start	Service End	Idle	TAT	WTQ
0	0	4	0	4	0	4	0
1	1	1	4	5	0	4	3
1	2	1	5	6	0	4	3
12	14	5	14	19	8	5	0
6	20	4	20	24	1	4	0
1	21	1	24	25	0	4	3
1	22	1	25	26	0	4	3
8	30	5	30	35	4	5	0
2	32	1	35	36	0	4	3
0	32	1	36	37	0	5	4
9	41	5	41	46	4	5	0

Figure : Simulation of Single-Server



InterArrival	Arrival	Service	Service Start	Service End	Idle	TAT	WTQ
0	0	4	0	4	0	4	0
1	1	1	4	5	0	4	3
1	2	1	5	6	0	4	3
12	14	5	14	19	8	5	0
6	20	4	20	24	1	4	0
1	21	1	24	25	0	4	3
1	22	1	25	26	0	4	3
8	30	5	30	35	4	5	0
2	32	1	35	36	0	4	3
0	32	1	36	37	0	5	4
9	41	5	41	46	4	5	0

Figure : Simulation of Single-Server

For Simulation multi-server Simulation Results like:

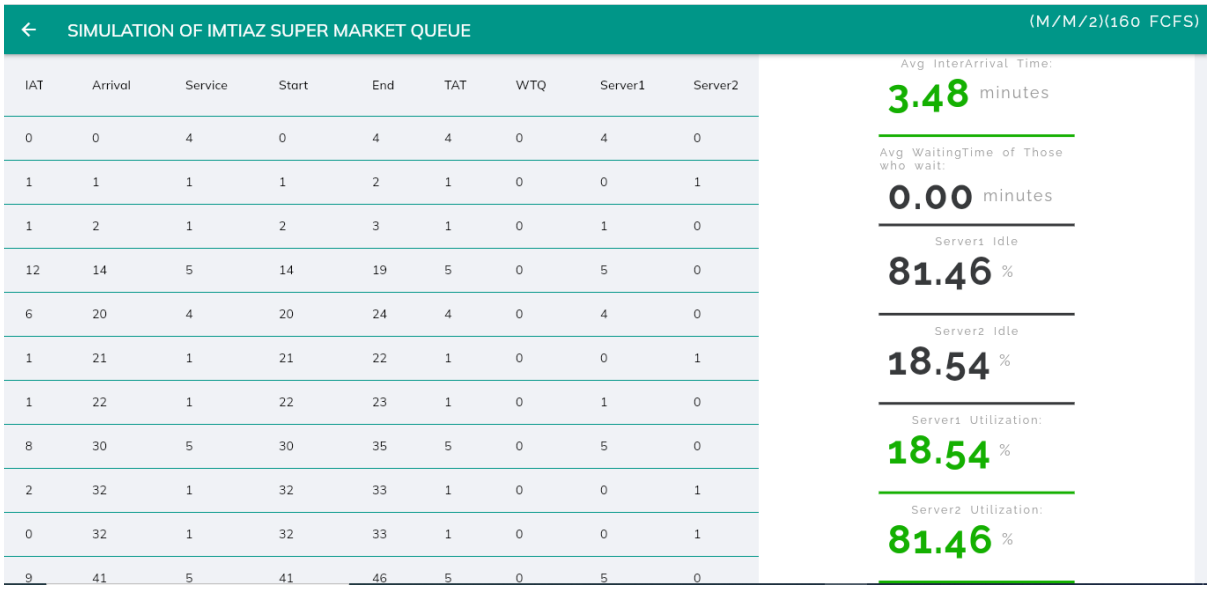


Figure : Simulation of Multi-Server

Similarly, when we click on last option (Random), it will look like this:

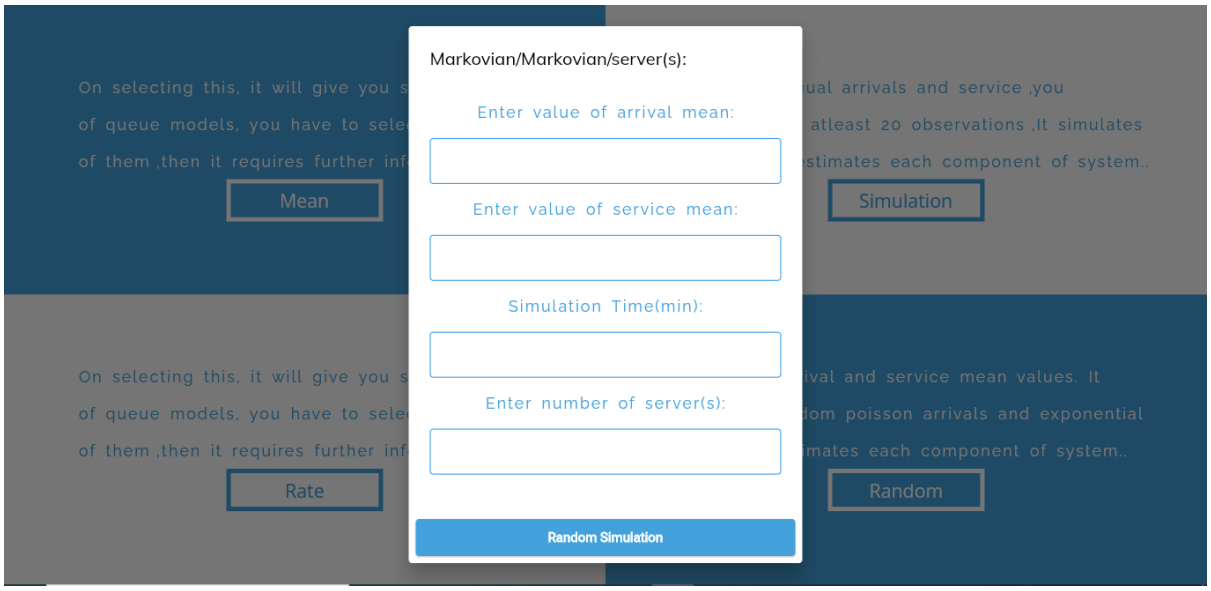
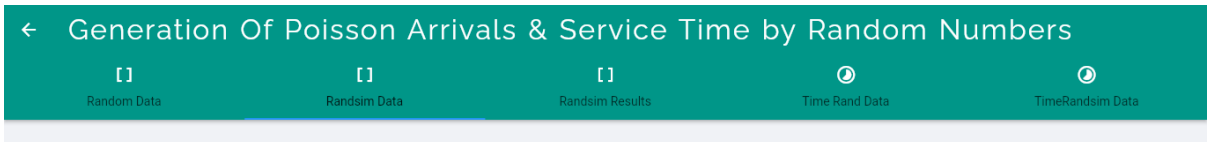


Figure : Random Simulation Inputs

For single-server Random Simulation Results like:

Cummulative Probability	Probability Lookup	Random Variables	Random Numbers	Inter Arrival Time	Arrival Time	Service Time
0.25975502118524524	0	0	0.8752006301249515	0	0	0
0.5084347733353394	0.25975502118524524	1	0.17524755168876194	0	0	0
0.7272729552274222	0.5084347733353394	2	0.12130484910008299	0	0	1
0.8717061552761969	0.7272729552274222	3	0.2290453954818925	0	0	0
0.9479668849019499	0.8717061552761969	4	0.3339359093582568	1	1	3
0.9815216059372812	0.9479668849019499	5	0.04317252562583285	0	1	5
0.9941765292991777	0.9815216059372812	6	0.122797161860984	0	1	3
0.9983526540086035	0.9941765292991777	7	0.5531594278166472	2	3	2
0.9995776505900351	0.9983526540086035	8	0.17571434288343513	0	3	2

Figure: Generation of Arrival & Service Time



InterArrival	Arrival	Service	Service Start	Service End	Idle	TAT	WTQ
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	0	0	1	1	0	1	1
1	1	3	1	4	0	3	0
0	1	5	4	9	0	8	3
0	1	3	9	12	0	11	8
2	3	2	12	14	0	11	9
0	3	2	14	16	0	13	11

Figure: SS,SE,TAT,WTQ Calculation

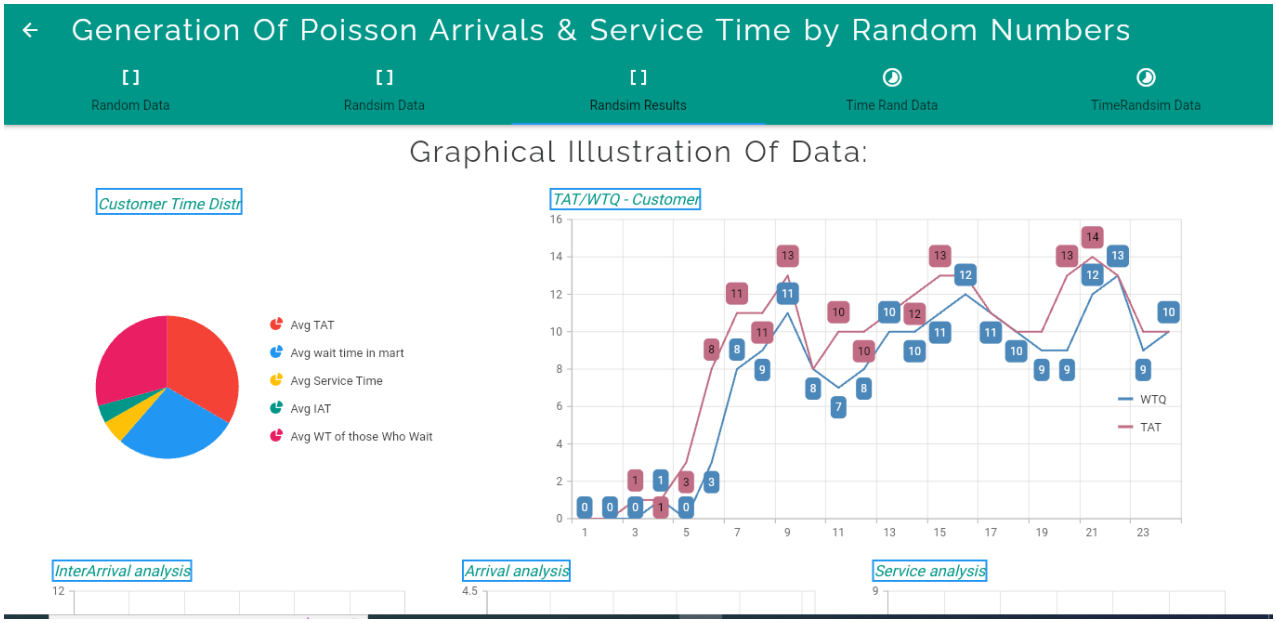


Figure: Graphical Representation of RandomSim results I

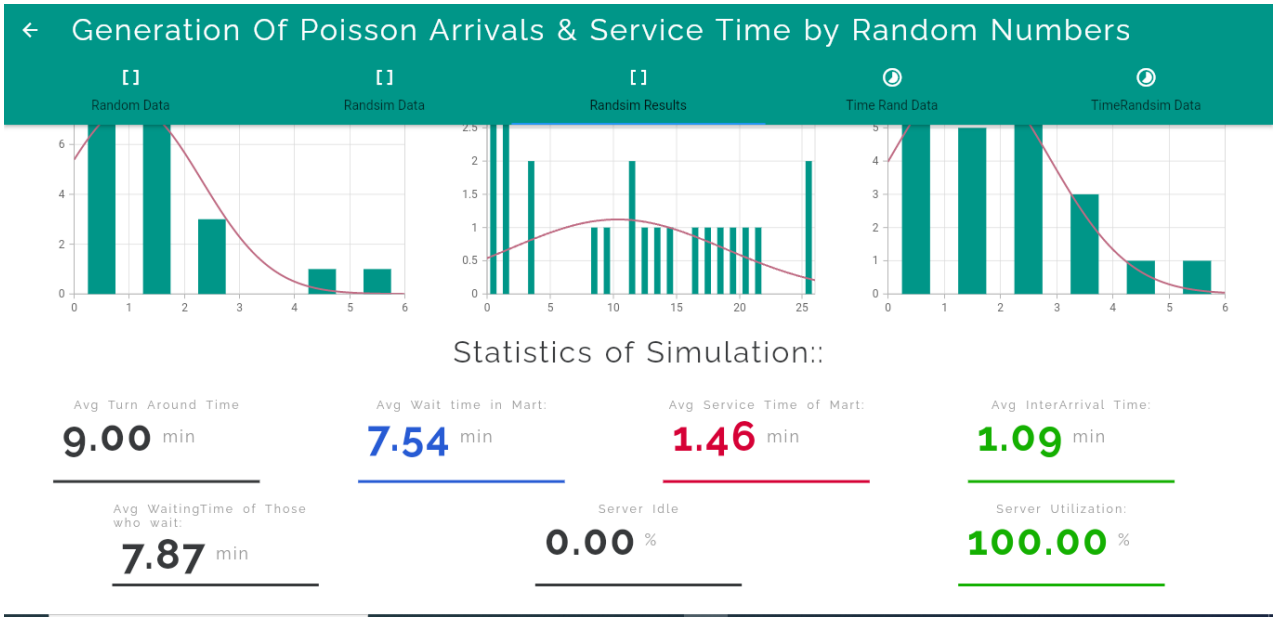


Figure: Graphical Representation of RandomSim results II

For multi-server Random Simulation Results will be look like above shown results.

4.2 Usability Testing

Usability testing is the process of evaluating a design's usability with a set of users who are representative of the target audience. It generally entails watching people attempt to perform activities and may be used in a variety of designs. It is commonly repeated from the beginning of a product's development till its release. Usability testing follows specific guidelines, one of which is that the application should be simple to comprehend and use. Each button's purpose and function must be understood by the user. Users were able to comprehend the functions of buttons and how to accomplish them in this application, and they were able to quickly augment the objects according to their preferences. The user interface should deliver a pleasurable experience. The color scheme chosen in the application is simple and attractive.

4.2.1 Visibility

We selected white and blue color for the backdrop for our application and white for the text and blue for the buttons and give an appropriate size to the buttons so that everything is more apparent and provides the user with a pleasant experience while using it.

4.2.2 Feedback

When buttons are pressed, their color changes to indicate that a certain action has been taken.

4.2.3 Inconsistency

Our application is consistent and follows widely accepted procedures.

4.2.4 Affordance

Every action the user makes is accompanied by suggestions and help from the application.

5. Working:

Basically this Queuing Simulator categorized into two main function of queuing simulation. First functionality takes single valued arrival and service , on the basis of given arrival service mean or rate Queuing Simulator estimates all the statistics of queuing system such as server(s) utilization and idle time, average length of queue ,average time a customer wait in queue for service, average length of system, average time customer spends in system etc. The second function does two separate function in it, one is we put observed data of any mart or somewhere (data contains interarrival time and service time of individual customers) simulator will calculate all the statistics of considered system and also shows results graphically , the last option is to give simulator only arrival mean and service mean simulator itself find all the individual arrivals and services and findout all statistics of the observed system and show results

graphically as well. In last option we will also give a specific time in min at which our simulation stop.

6. Formulas:

M/M/1	**M/M/c**
$\rho = \frac{\lambda}{\mu}$	$\rho = \frac{\lambda}{c\mu}$
$L_q = \frac{\rho^2}{1-\rho}$	$L_q = \frac{P_0 \left(\frac{\lambda}{\mu}\right)^c \rho}{c!\,(1-\rho)^2}$
$L_s = \frac{\lambda}{\mu - \lambda}$	$W_s = W_q + \frac{1}{\mu}$
$W_s = \frac{1}{\mu-\lambda}$	$P_0 = \frac{1}{\sum_{m=0}^{c-1} \frac{(c\rho)^m}{m!} + \frac{(c\rho)^c}{c!\,(1-\rho)}}$
$W_q = \frac{\lambda}{\mu(\mu-\lambda)}$	$W_q = \frac{L_q}{\lambda}$
$L_n = \frac{\mu}{\mu-\lambda}$	$L_s = L_q + \frac{\lambda}{\mu}$
M/G/1	**M/G/c**
$\rho = \frac{\lambda}{\mu}$	$W_q^{M/G/C} \approx W_q^{M/M/C} \frac{{C_a}^2 + {C_s}^2}{2}$
$L_q = \frac{\lambda^2(\sigma_s)^2 + \rho^2}{2(1-\rho)}$	$L_q = \lambda W_q^{M/G/C}$

$W_q = \frac{L_q}{\lambda}$	$W_s = W_q + \frac{1}{\mu}$
$W_s = W_q + \frac{1}{\mu}$	$L_s = \lambda W_s$
$L_s = \lambda W_s$	

Turn Around Time = Service End – Arrival Time
Wait Time = Turn Around Time – Service Time
Server Utilization = Service End Time – Service Start Time
Response Time = Service Start Time – Arrival Time
Response Time = Service Start Time – Arrival Time
Avg Time Cust Spend in Mart = $\frac{Sum\ of\ Turn\ Around\ Time}{number\ of\ customers}$
Average Wait Time = $\frac{Sum\ of\ Wait\ Time}{number\ of\ customers}$
Average Response Time = $\frac{Sum\ of\ Response\ Time}{number\ of\ customers}$
Mean Service Time = $\frac{Sum\ of\ Service\ Time}{number\ of\ customers}$
Average Time B/W Arrival = $\frac{Sum\ of\ Inter\ Arrival\ Time}{number\ of\ customers-1}$

$$\textbf{Avg Waiting Time of those who wait} = \frac{\textbf{Sum of Wait Time}}{\textbf{number of customers} - 1}$$

7. Data Collection Challenges :

Think of a project idea and creating the project live seems Sounds simple enough, doesn't it? And the pre-requirements for the project in our case we are making a queueing simulation

For supermarket that purpose, we need data for the simulation project but getting data from someone or collecting it is a nightmare because it is very difficult to choose what is the collection process like for example by observing the customers in the system or by interviewing them individually customers but following this lengthy process it takes so much time. The most important thing is the mart entering time by the customer and the time taken by the server or counter to provide the service to the customer. So our team chose to observe customers' time entering the mart and after they complete their shopping standing in the queue waiting for their turn. We observe the customer service time at the counter but unfortunately, the plan didn't go as expected. The list we think that this supermart allowed us to observe the customer's time. The first on our list is Imtiaz supermarket, the second chase up supermarket, the third one is al-jaddeed supermart. All the marts in the list reject without listening to our information regarding project data collection from us. So we plan that the only way is that intimately observe the customer's that the time taken by the server or counter to provide the service to each customer. And the entering time by each customer in supermart is observed by waiting outside. By following this method we successfully collected 160 Observation (interarrival and service time).

C#	IAT	Ser	C#	IAT	Ser	C#	IAT	Ser	C#	IAT	Ser
1	0	4	44	9	3	87	1	1	130	1	2
2	1	1	45	1	3	88	1	1	131	14	3
3	1	1	46	1	1	89	1	5	132	1	1
4	12	5	47	1	1	90	1	4	133	1	1
5	6	4	48	7	2	91	1	1	134	6	1
6	1	1	49	1	1	92	1	1	135	1	1
7	1	1	50	1	3	93	11	3	136	1	1
8	8	5	51	2	3	94	9	5	137	4	3
9	2	1	52	1	1	95	1	1	138	10	3
10	0	1	53	1	1	96	18	6	139	1	1
11	9	5	54	12	3	97	11	4	140	1	1
12	1	1	55	3	6	98	1	1	141	4	2
13	1	1	56	1	1	99	1	1	142	1	1
14	1	5	57	1	4	100	1	5	143	1	1
15	20	5	58	1	3	101	1	1	144	4	4
16	1	1	59	0	1	102	1	1	145	1	1
17	1	1	60	1	1	103	21	6	146	1	1
18	12	5	61	8	3	104	6	6	147	5	2
19	1	5	62	1	1	105	2	1	148	20	3
20	16	1	63	1	1	106	1	1	149	1	1
21	7	6	64	1	3	107	8	2	150	1	3
22	1	1	65	11	6	108	1	1	151	7	4
23	1	1	66	1	1	109	2	1	152	1	1
24	13	1	67	1	1	110	9	1	153	1	1
25	4	6	68	2	3	111	1	1	154	4	1
26	1	1	69	1	1	112	1	1	155	7	2
27	1	3	70	1	1	113	1	3	156	1	1
28	3	2	71	13	3	114	7	3	157	1	3
29	9	2	72	5	3	115	1	1	158	4	3
30	1	1	73	1	1	116	1	1	159	1	1
31	1	1	74	1	5	117	4	3	160	1	1
32	2	3	75	2	4	118	1	1			
33	1	1	76	3	1	119	1	1			
34	1	3	77	1	1	120	4	4			
35	5	3	78	8	3	121	4	1			
36	1	1	79	8	16	122	1	1			
37	1	1	80	1	1	123	1	1			
38	1	1	81	1	3	124	1	2			
39	1	1	82	5	3	125	1	1			
40	3	2	83	1	1	126	1	1			
41	8	3	84	1	1	127	4	2			
42	1	1	85	1	1	128	7	3			
43	0	1	86	5	3	129	1	1			

AT : interarrival time

Ser : service time

C# : customer no

IAT unit : minutes

Ser unit : minutes

8. Conclusion:

The main goal of this system is to research and pave the way for the use of Queuing Theory in the real world environment as well as to provide business opportunities to Grocery marts , Shopping marts etc. It will introduce a change in inventory systems, hospital systems and marts system. From our web application management of System can get the drawbacks in queuing environment , can better queuing system , can reduce number of people and increase revenue of people means can serve more peoples in less time.