CMPE-273 Lab 2 Splitwise

Yusuf Juzar Soni

Link on YouTube: https://youtu.be/AFTvPoaZPzI

Introduction:

Problem Statement:

Mocking the popular bill splitting application "Splitwise" using a diverse technology stack. Splitwise is a mobile app and web platform that helps users share expenses with others. The technologies used in this project are as follows:

- React JS was used for frontend coding.
- Node was used as to implement the API layer etc.
- MongoDB was used to implement the databases.
- Kafka was implemented as a variant of a message queue.

Functional Requirements/ Goals of the System covered:

- A new user would be able to sign up and will be redirected to his dashboard which shows a summary of his transactions (How much he owes, how much he is owed etc.)
- Existing users can log in and would be redirected to their respective dashboards.
- Form based validations have been implemented to check proper inputs
- The user can see a list of groups he is part of, he can also search within that list of groups if he wishes.
- The left navbar also contains links to the recent activity page where the user can view a history of who has added bills into the group. Pagination (from the backend was implemented in recent activity).
- The same navbar also contains a link to the invite list page which displays a list of groups the user has been invited to. The user can accept the invitation, only after accepting the invitation will the group be visible in the users group list.
- The members list also changes based on the invite status.
- A member can create a group by selecting from all the users registered in the app.
- A member can settle up the amount he is owed and the amount he owes.
- A basic profile page is visible that gets the data from the database and displays the data stored in the backend. The user also has the ability to add profile picture and edit profile information.
- The user can add and delete comments on each bill. The can only delete his/her own comments.
- User can only leave group after all expenses have been settled up.

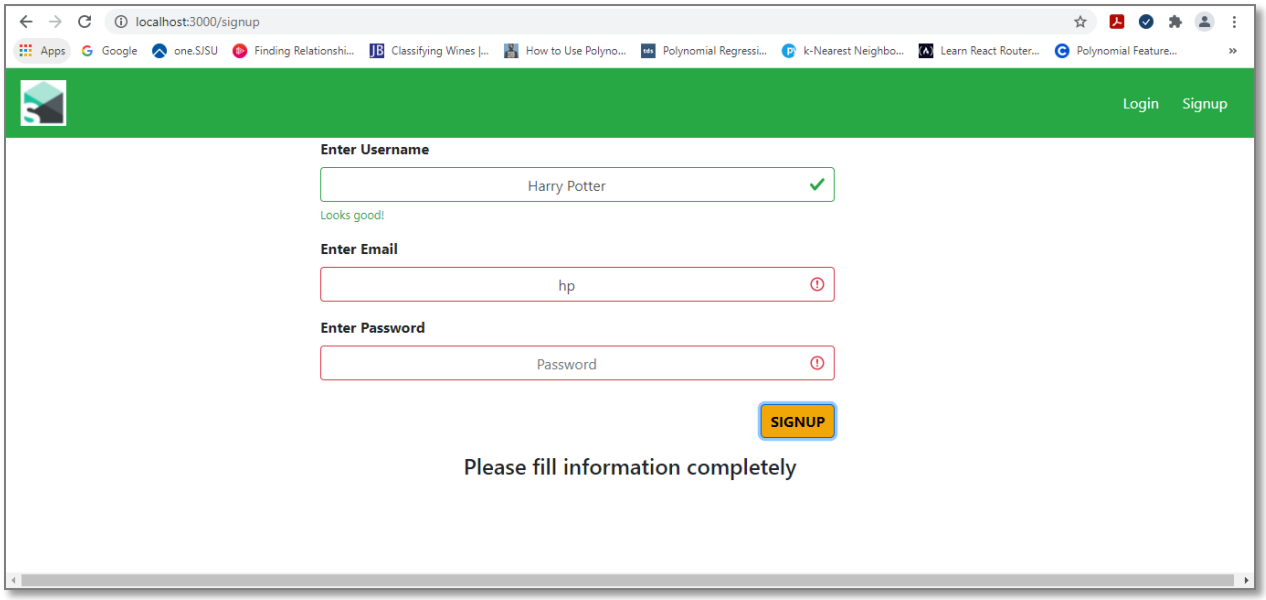
System Design and Database Overview

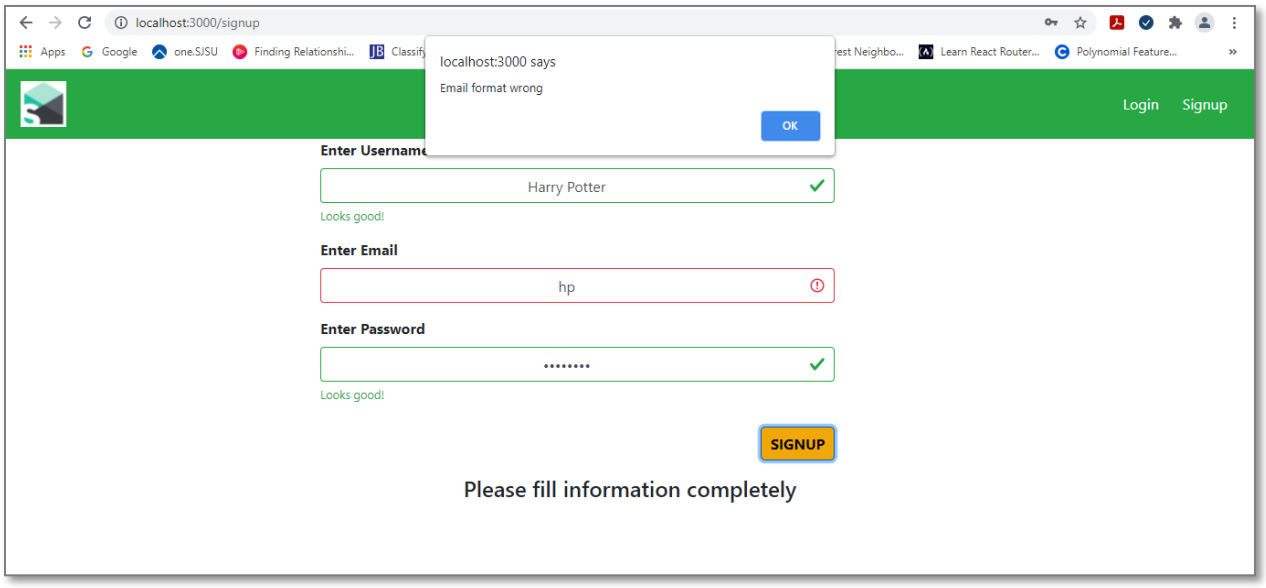
- ReactJS makes get, post, put calls to express backend using routes.
- NodeJS receives the requests and performs MongoDB operations to update the database.
- Session is assigned to a user when he signs up or logs in.
- MongoDB database receives the requests from NodeJS and performs the operations to its tables.
- Backend Sends the response back to React JS to display.
- Having Kafka enables asynchronous communication and helps in scaling.

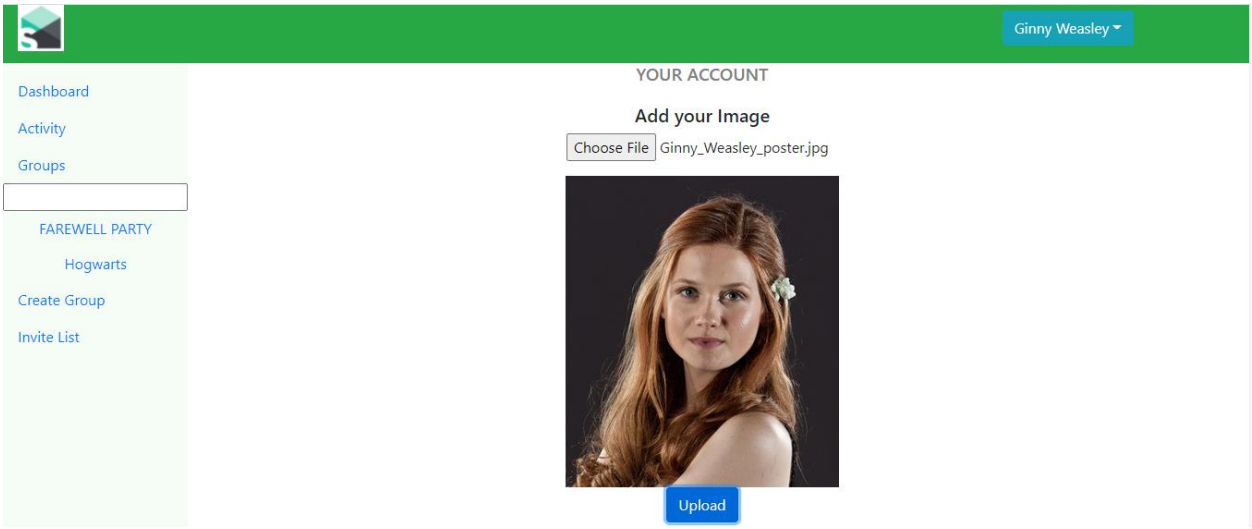
MongoDB Schema:

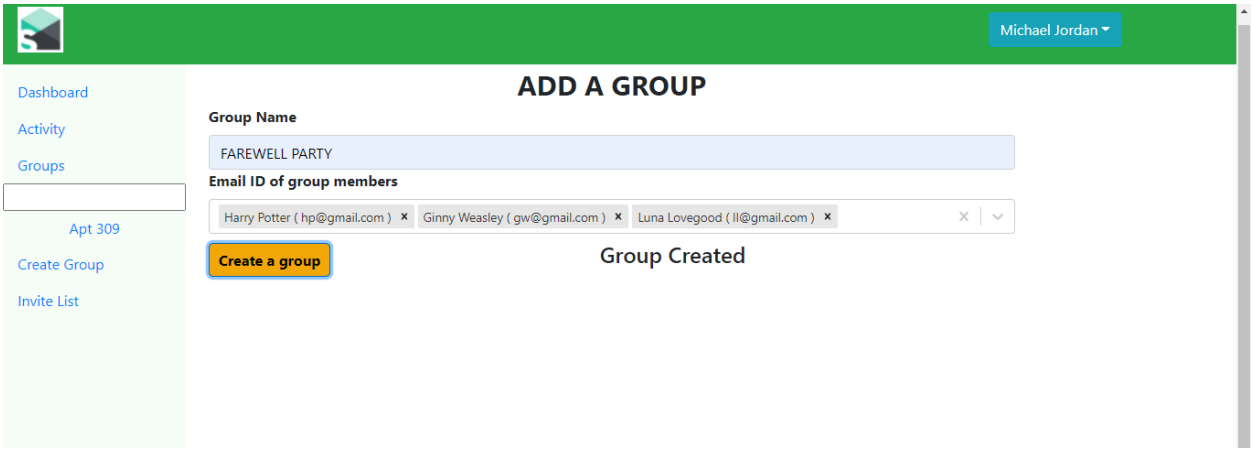
- users (stores user elated information like username, email etc.) **Email is used as primary key and hence is used in all operations.**
- groups (stores group name, group description, group photo, members, invited members, created by etc.)
- bills (store bill related information, bill id, bill name, amount description etc.) primarily used in displaying recent activity.
- transactions (used to store details of transactions and also keeps record of splits etc.)
- comments(comment_id, bill_id, commented_by, comment_body)

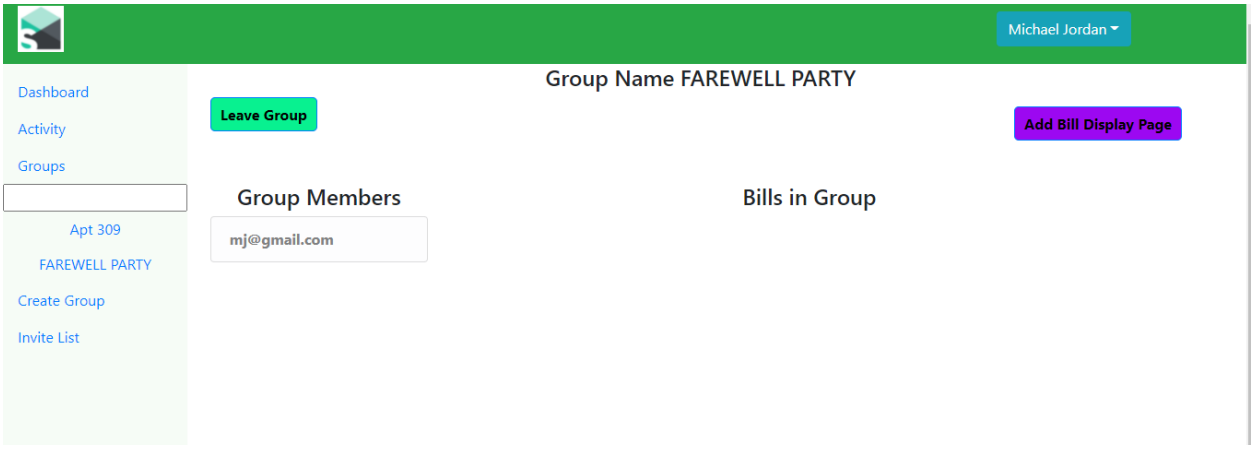
Screenshots of various workflows are added below.

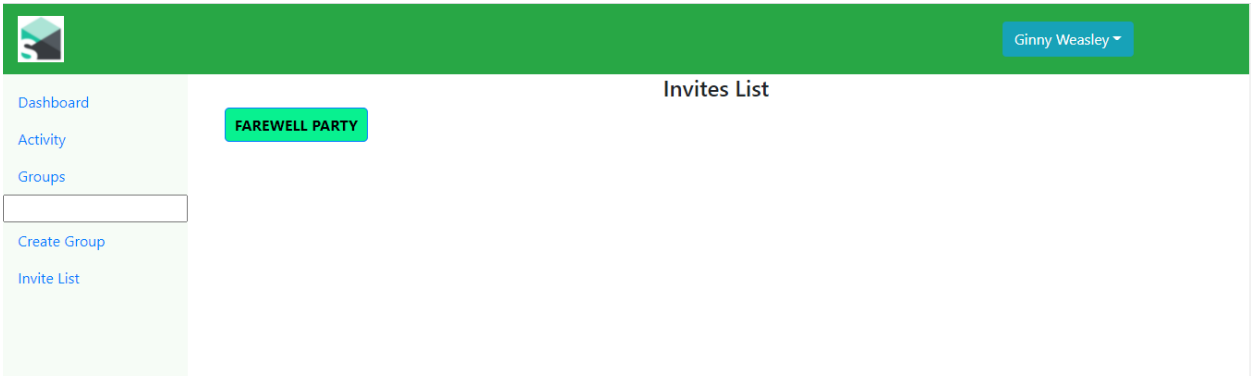


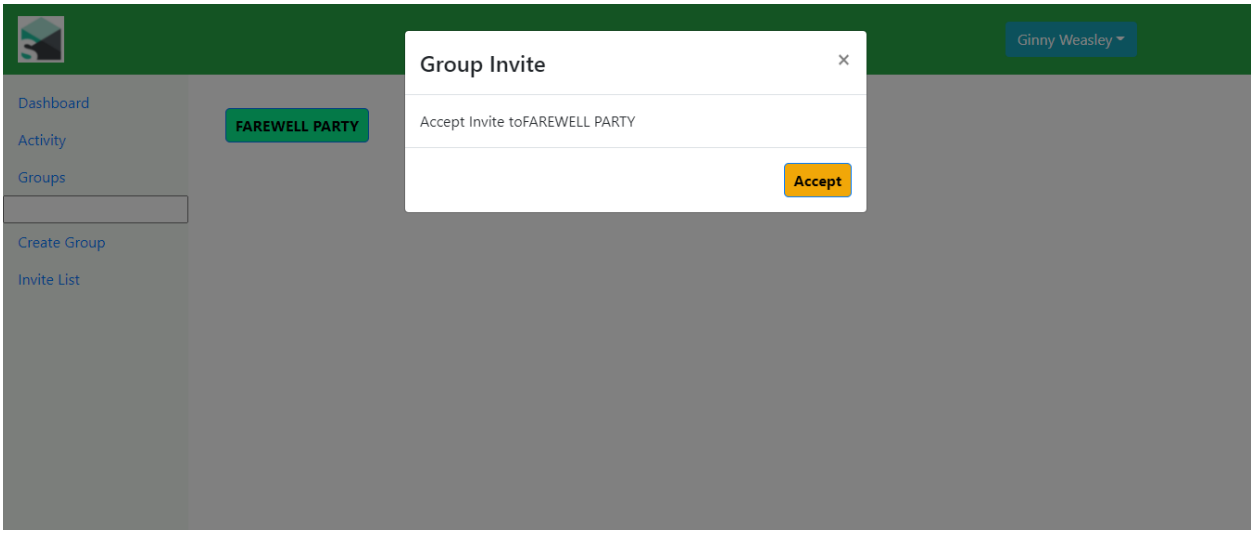

localhost:3000/signup
localhost:3000 says
Email format wrong
OK
Login
Signup
Harry Potter
Looks good!
Enter Email
hp
Enter Password
Looks good!
SIGNUP
Please fill information completely

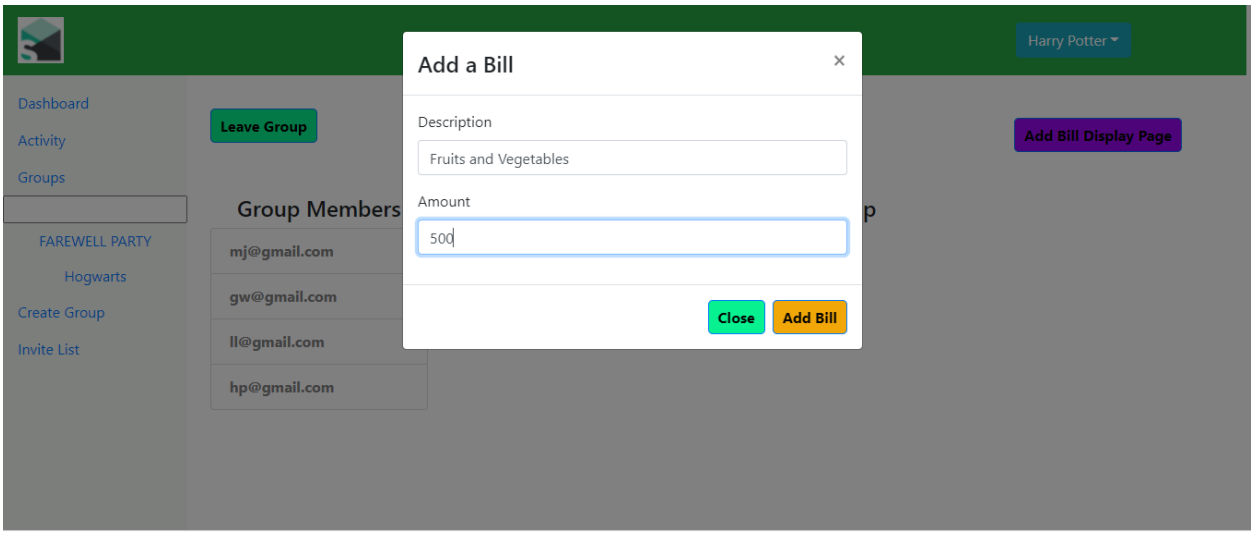

Ginny Weasley
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
YOUR ACCOUNT
Add your Image
Choose File
Ginny_Weasley_poster.jpg
Upload

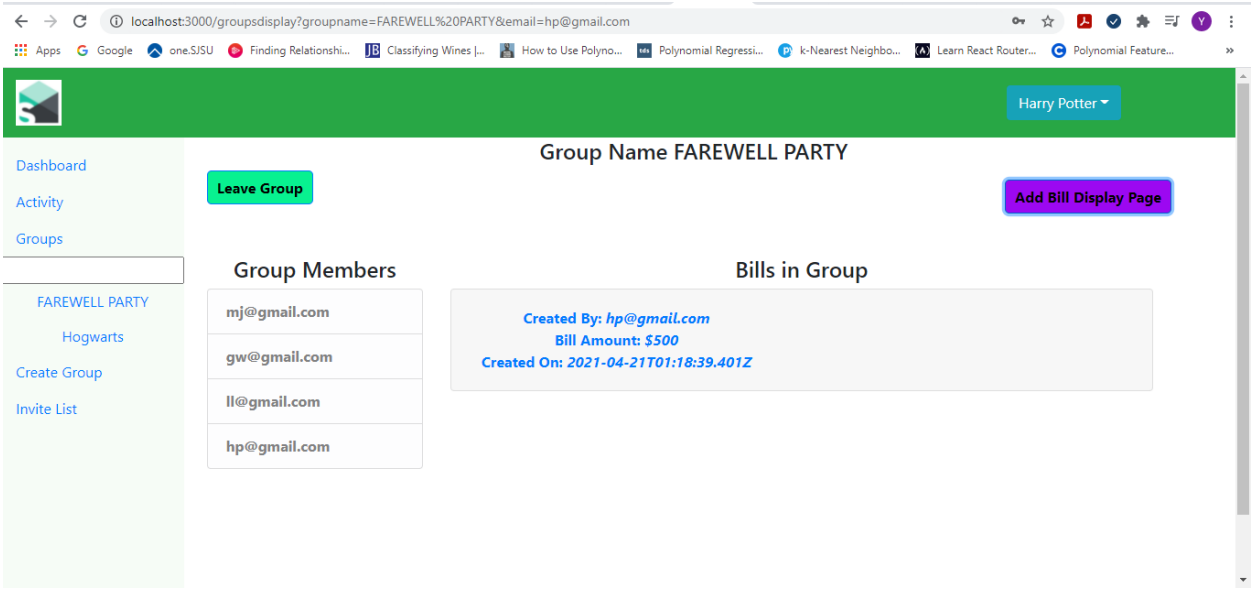

Michael Jordan
Dashboard
Activity
Groups
Apt 309
Create Group
Invite List
ADD A GROUP
Group Name
FAREWELL PARTY
Email ID of group members
Harry Potter (hp@gmail.com)
Ginny Weasley (gw@gmail.com)
Luna Lovegood (ll@gmail.com)
Create a group
Group Created

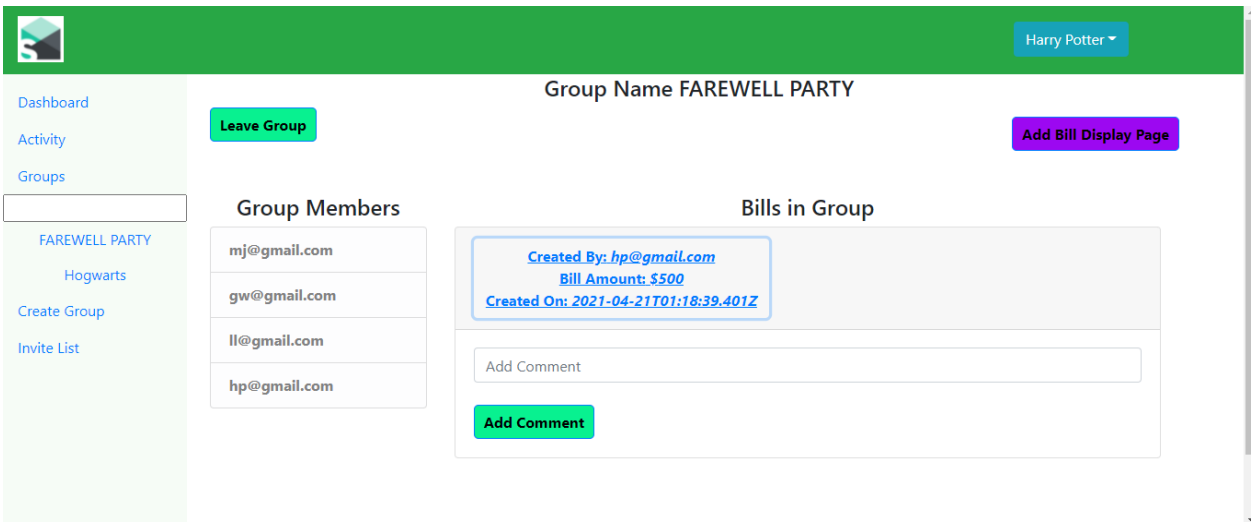

Michael Jordan
Dashboard
Activity
Groups
Apt 309
FAREWELL PARTY
Create Group
Invite List
Group Name FAREWELL PARTY
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
Bills in Group

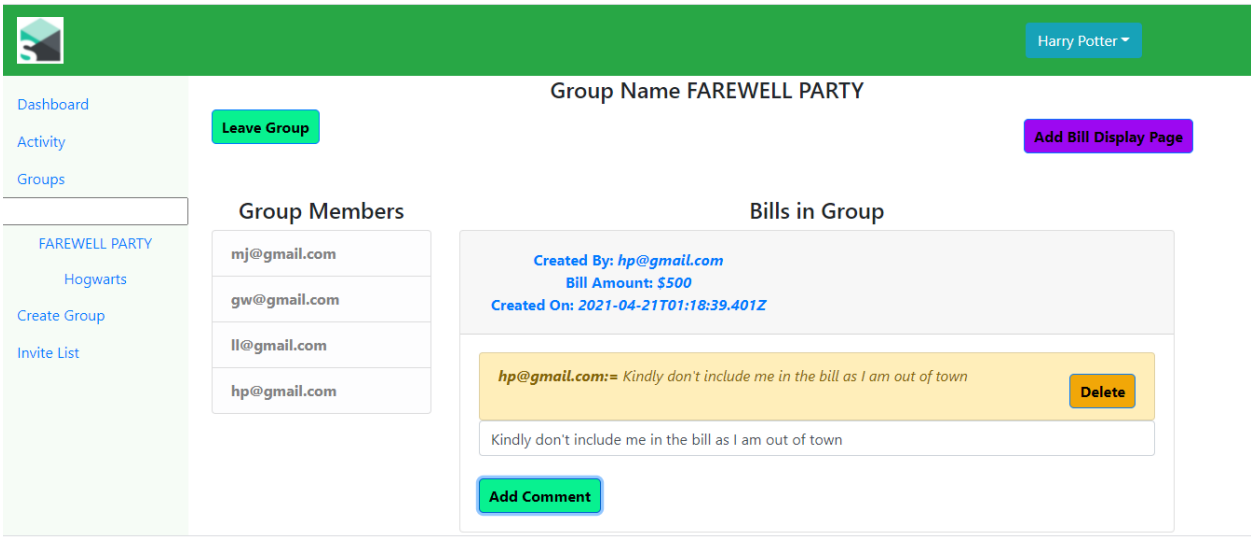

Ginny Weasley
Dashboard
Activity
Groups
Create Group
Invite List
Invites List
FAREWELL PARTY

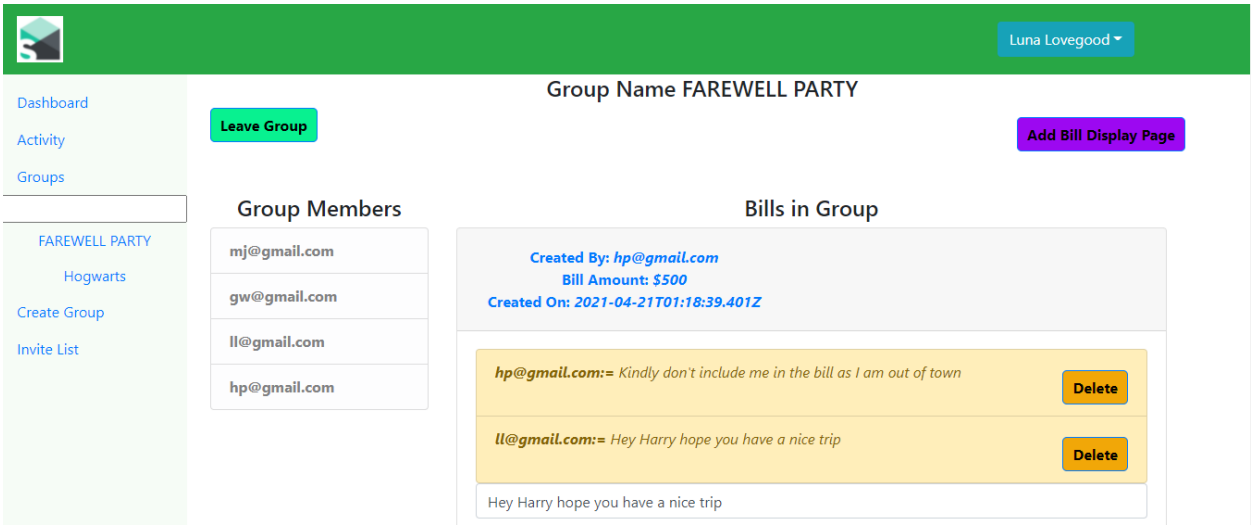

Ginny Weasley
Dashboard
Activity
Groups
Create Group
Invite List
FAREWELL PARTY
Group Invite
Accept Invite toFAREWELL PARTY
Accept

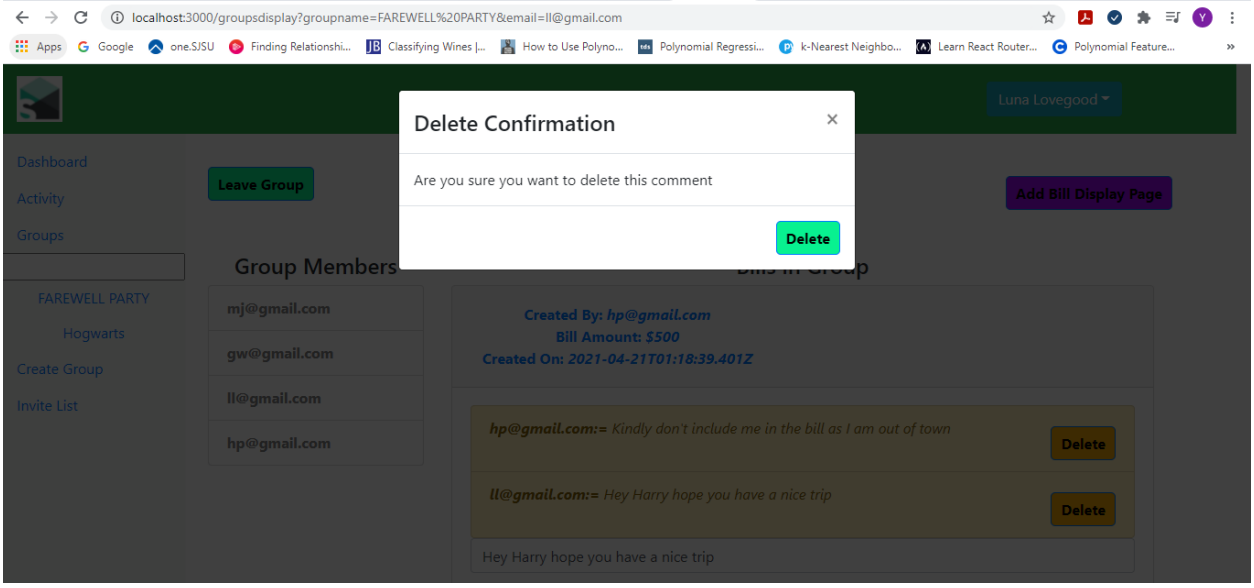

Harry Potter
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Add a Bill
Description
Fruits and Vegetables
Amount
500
Close
Add Bill

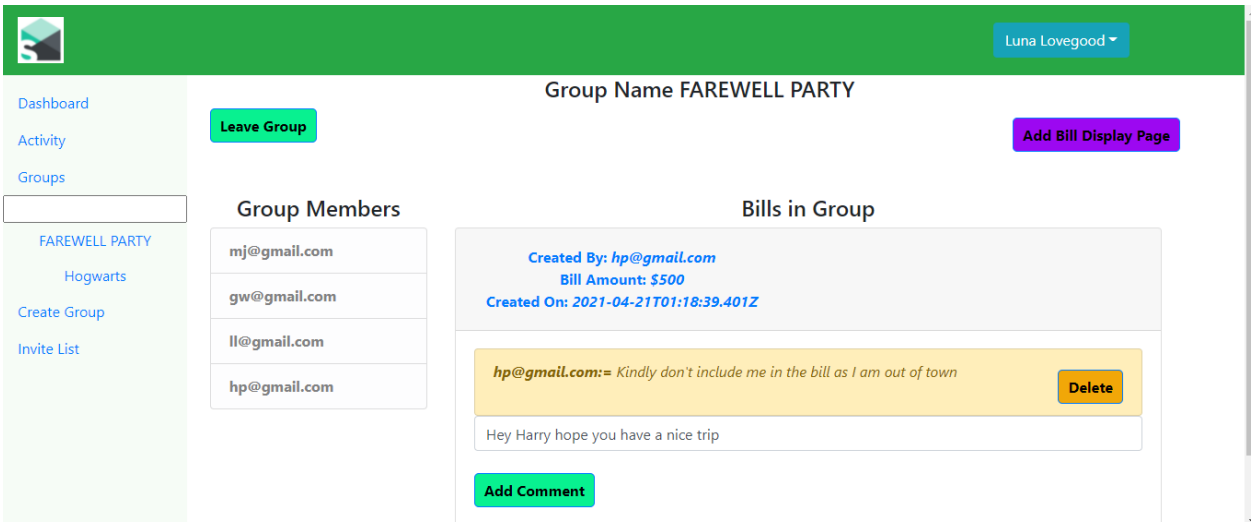

localhost:3000/groupsdisplay?groupname=FAREWELL%20PARTY&email=hp@gmail.com
Harry Potter
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Group Name FAREWELL PARTY
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Bills in Group
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z

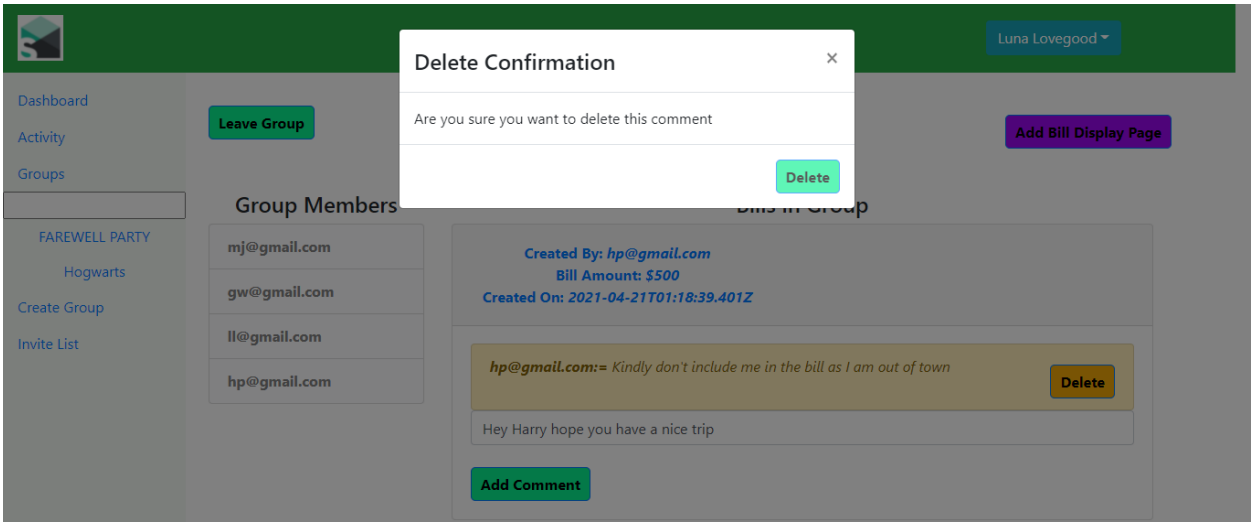

Harry Potter
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Group Name FAREWELL PARTY
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Bills in Group
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z
Add Comment
Add Comment


Harry Potter
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Group Name FAREWELL PARTY
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Bills in Group
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z
hp@gmail.com:= Kindly don't include me in the bill as I am out of town
Delete
Kindly don't include me in the bill as I am out of town
Add Comment


Luna Lovegood
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Group Name FAREWELL PARTY
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Bills in Group
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z
hp@gmail.com:= Kindly don't include me in the bill as I am out of town
Delete
ll@gmail.com:= Hey Harry hope you have a nice trip
Delete
Hey Harry hope you have a nice trip

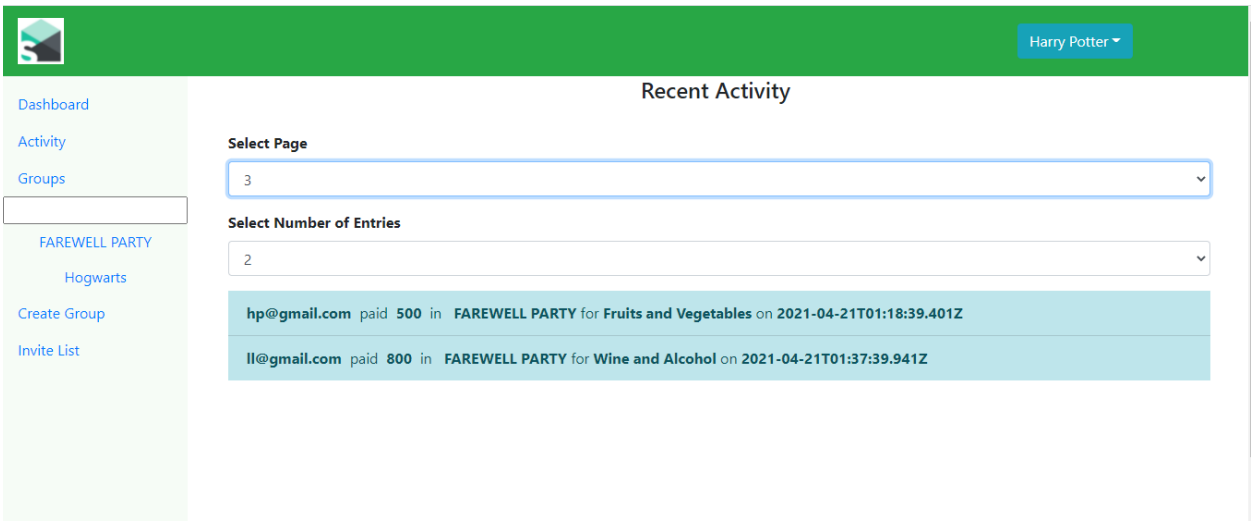

localhost:3000/groupsdisplay?groupname=FAREWELL%20PARTY&email=ll@gmail.com
Apps
Google
one.SJSU
Finding Relationshi...
Classifying Wines |...
How to Use Polyno...
Polynomial Regressi...
k-Nearest Neighbo...
Learn React Router...
Polynomial Feature...
Luna Lovegood
Delete Confirmation
Are you sure you want to delete this comment
Delete
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z
hp@gmail.com:= Kindly don't include me in the bill as I am out of town
Delete
ll@gmail.com:= Hey Harry hope you have a nice trip
Delete
Hey Harry hope you have a nice trip

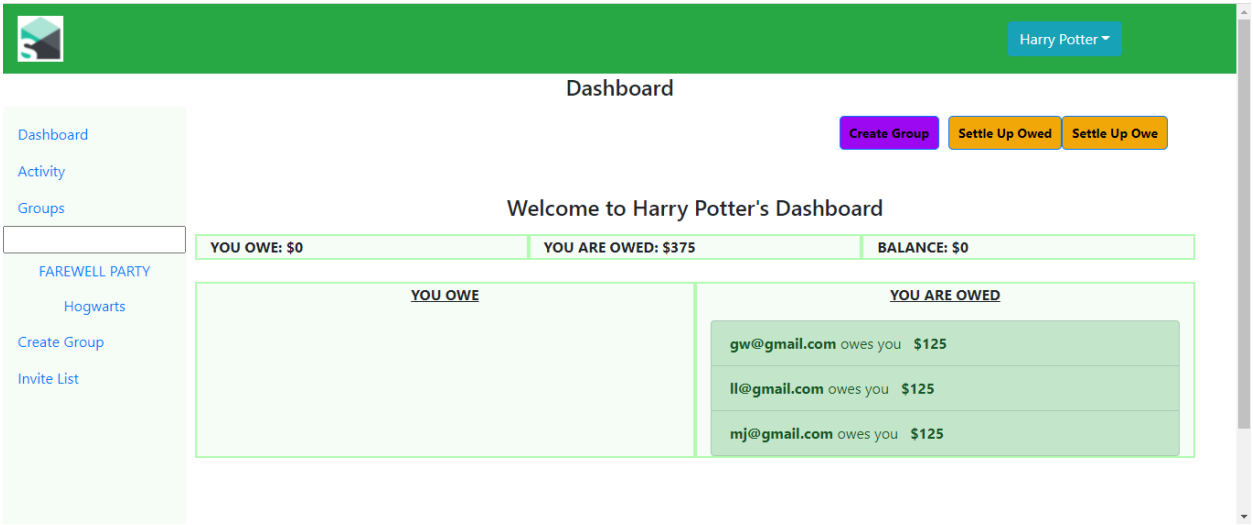

Luna Lovegood
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Group Name FAREWELL PARTY
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Bills in Group
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z
hp@gmail.com:= Kindly don't include me in the bill as I am out of town
Delete
Hey Harry hope you have a nice trip
Add Comment

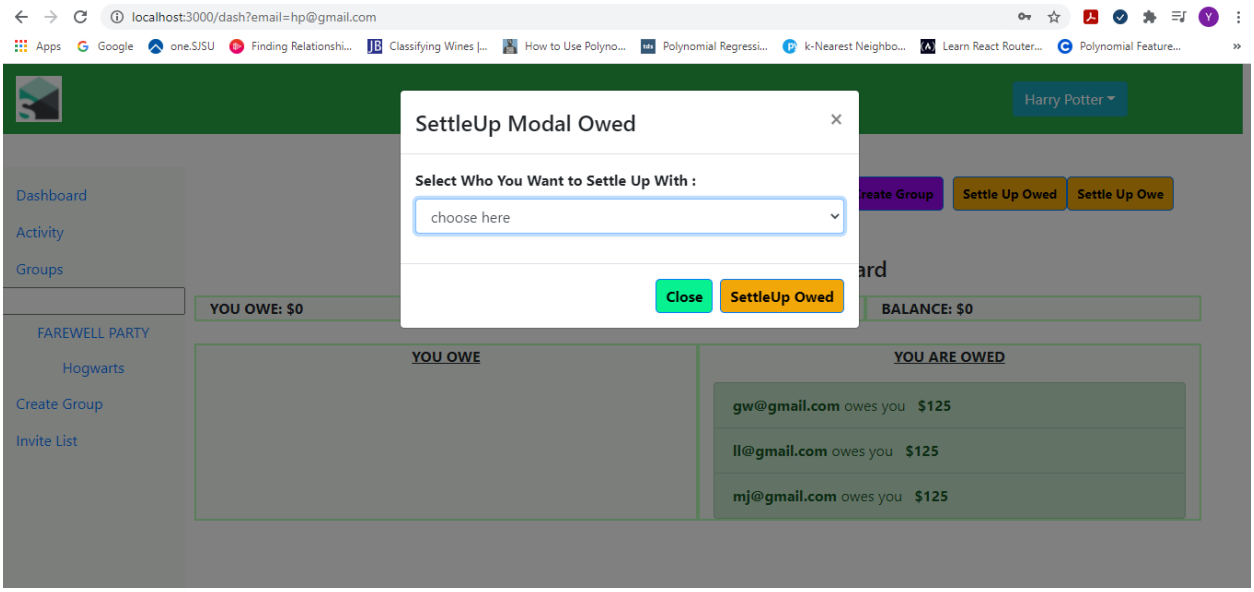

Luna Lovegood
Delete Confirmation
Are you sure you want to delete this comment
Delete
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Leave Group
Add Bill Display Page
Group Members
mj@gmail.com
gw@gmail.com
ll@gmail.com
hp@gmail.com
Created By: hp@gmail.com
Bill Amount: $500
Created On: 2021-04-21T01:18:39.401Z
hp@gmail.com:= Kindly don't include me in the bill as I am out of town
Delete
Hey Harry hope you have a nice trip
Add Comment

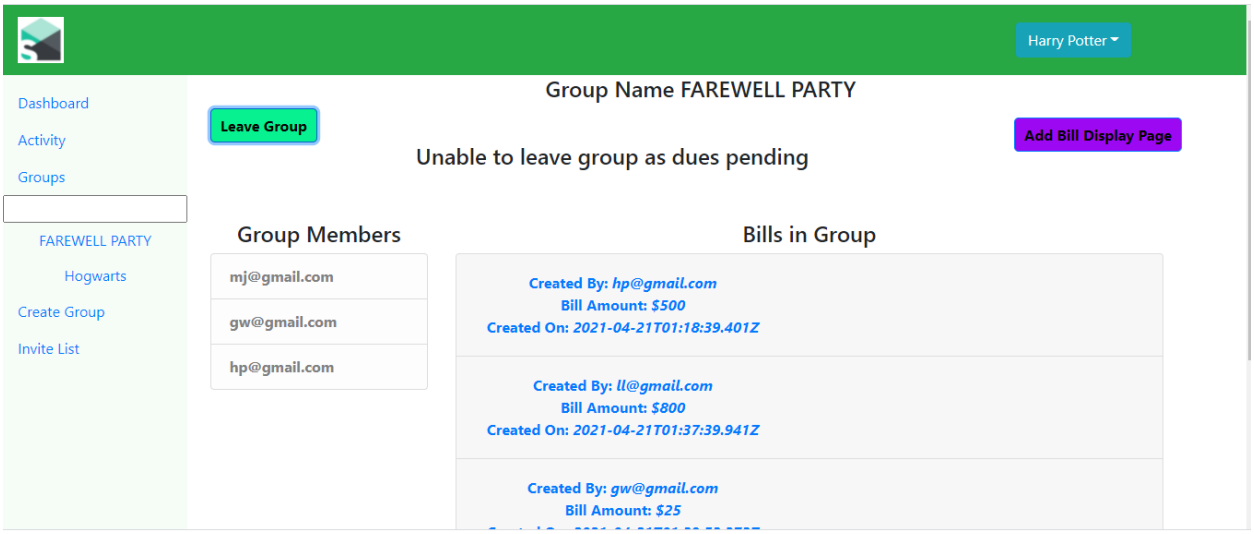

Harry Potter
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Recent Activity
Select Page
1
Select Number of Entries
2
hp@gmail.com paid 500 in FAREWELL PARTY for Fruits and Vegetables on 2021-04-21T01:18:39.401Z
ll@gmail.com paid 800 in FAREWELL PARTY for Wine and Alcohol on 2021-04-21T01:37:39.941Z


Harry Potter
Recent Activity
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Select Page
2
Select Number of Entries
5
hp@gmail.com paid 500 in FAREWELL PARTY for Fruits and Vegetables on 2021-04-21T01:18:39.401Z
ll@gmail.com paid 800 in FAREWELL PARTY for Wine and Alcohol on 2021-04-21T01:37:39.941Z
gw@gmail.com paid 25 in FAREWELL PARTY for Balloons on 2021-04-21T01:39:53.373Z
hp@gmail.com paid 500 in FAREWELL PARTY for Spoons and Plates on 2021-04-21T01:41:37.766Z
hp@gmail.com paid 350 in FAREWELL PARTY for Board Games on 2021-04-21T01:42:04.730Z


Harry Potter
Recent Activity
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Select Page
3
Select Number of Entries
2
hp@gmail.com paid 500 in FAREWELL PARTY for Fruits and Vegetables on 2021-04-21T01:18:39.401Z
ll@gmail.com paid 800 in FAREWELL PARTY for Wine and Alcohol on 2021-04-21T01:37:39.941Z

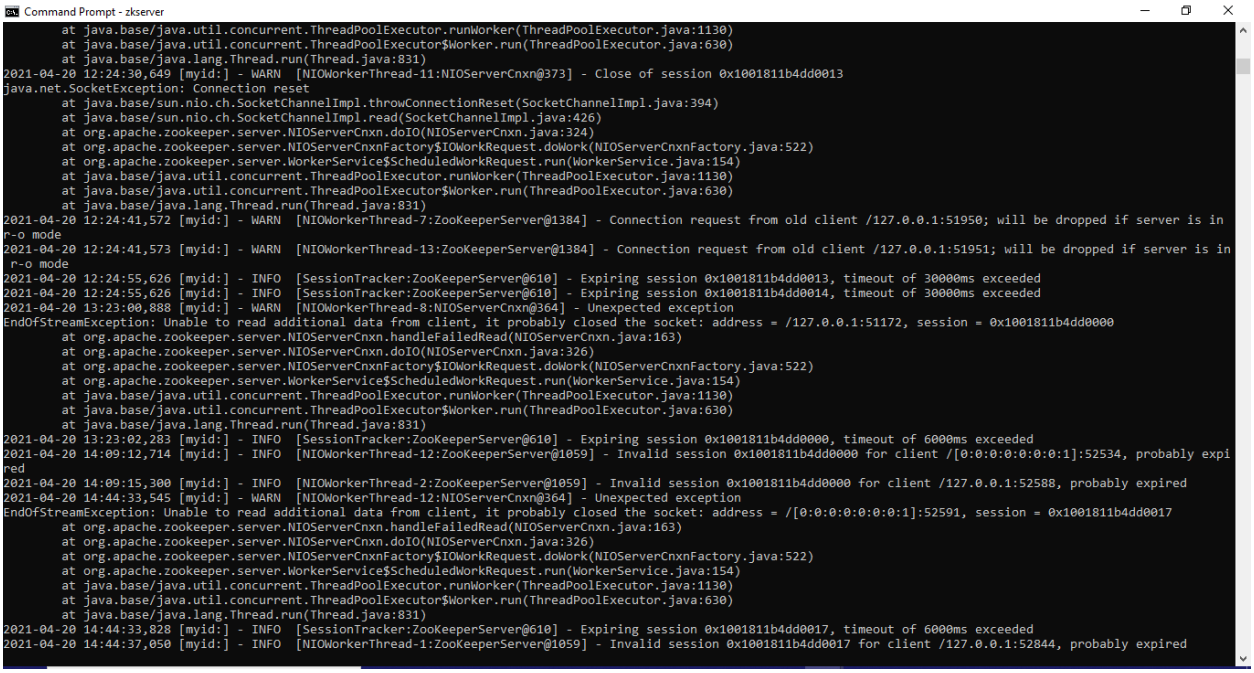

Harry Potter
Dashboard
Dashboard
Activity
Groups
FAREWELL PARTY
Hogwarts
Create Group
Invite List
Create Group
Settle Up Owed
Settle Up Owe
Welcome to Harry Potter's Dashboard
YOU OWE: $0
YOU ARE OWED: $375
BALANCE: $0
YOU OWE
YOU ARE OWED
gw@gmail.com owes you $125
ll@gmail.com owes you $125
mj@gmail.com owes you $125

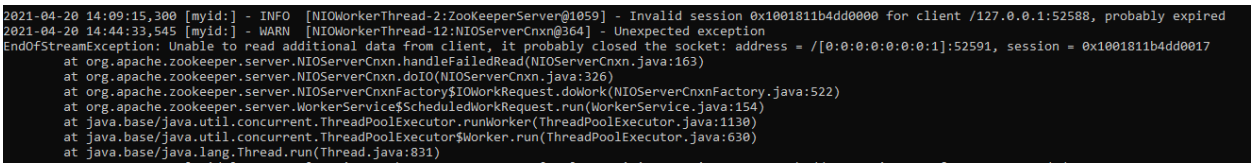




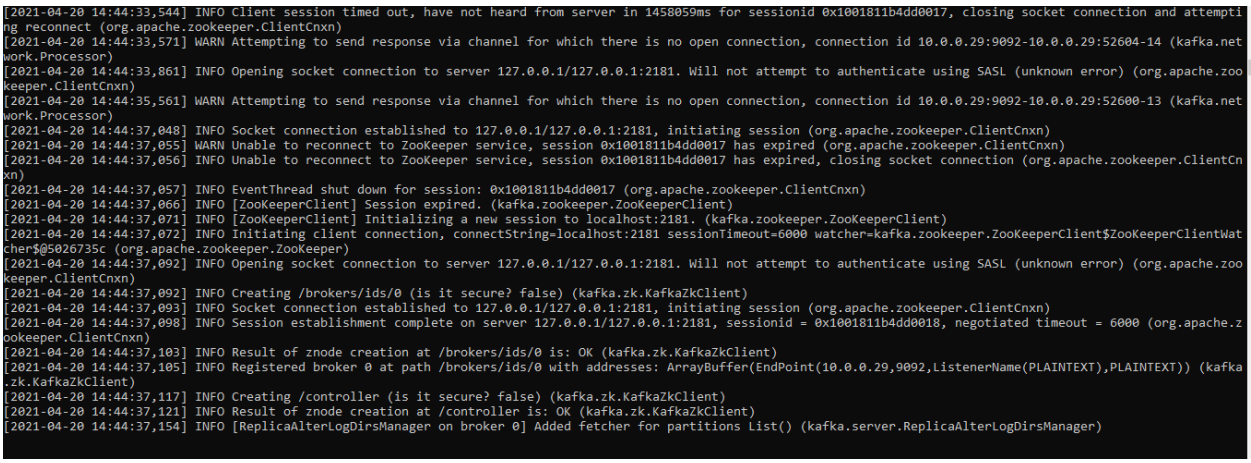
KAFKA SCREENSHOTS:

ZOOKEPER:


Command Prompt - zkserver
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:630)
at java.base/java.lang.Thread.run(Thread.java:831)
2021-04-20 12:24:30,649 [myid:] - WARN [NIOWorkerThread-11:NIOServerCnxn@373] - Close of session 0x1001811b4dd0013
java.net.SocketException: Connection reset
at java.base/sun.nio.ch.SocketChannelImpl.throwConnectionReset(SocketChannelImpl.java:394)
at java.base/sun.nio.ch.SocketChannelImpl.read(SocketChannelImpl.java:426)
at org.apache.zookeeper.server.NIOServerCnxn.doIO(NIOServerCnxn.java:324)
at org.apache.zookeeper.server.NIOServerCnxnFactory$IOWorkRequest.doWork(NIOServerCnxnFactory.java:522)
at org.apache.zookeeper.server.WorkerService$ScheduledWorkRequest.run(WorkerService.java:154)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:630)
at java.base/java.lang.Thread.run(Thread.java:831)
2021-04-20 12:24:41,572 [myid:] - WARN [NIOWorkerThread-7:ZooKeeperServer@1384] - Connection request from old client /127.0.0.1:51950; will be dropped if server is in r-o mode
2021-04-20 12:24:41,573 [myid:] - WARN [NIOWorkerThread-13:ZooKeeperServer@1384] - Connection request from old client /127.0.0.1:51951; will be dropped if server is in r-o mode
2021-04-20 12:24:55,626 [myid:] - INFO [SessionTracker:ZooKeeperServer@610] - Expiring session 0x1001811b4dd0013, timeout of 30000ms exceeded
2021-04-20 12:24:55,626 [myid:] - INFO [SessionTracker:ZooKeeperServer@610] - Expiring session 0x1001811b4dd0014, timeout of 30000ms exceeded
2021-04-20 13:23:00,888 [myid:] - WARN [NIOWorkerThread-8:NIOServerCnxn@364] - Unexpected exception
EndOfStreamException: Unable to read additional data from client, it probably closed the socket: address = /127.0.0.1:51172, session = 0x1001811b4dd0000
at org.apache.zookeeper.server.NIOServerCnxn.handleFailedRead(NIOServerCnxn.java:163)
at org.apache.zookeeper.server.NIOServerCnxn.doIO(NIOServerCnxn.java:326)
at org.apache.zookeeper.server.NIOServerCnxnFactory$IOWorkRequest.doWork(NIOServerCnxnFactory.java:522)
at org.apache.zookeeper.server.WorkerService$ScheduledWorkRequest.run(WorkerService.java:154)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:630)
at java.base/java.lang.Thread.run(Thread.java:831)
2021-04-20 13:23:02,283 [myid:] - INFO [SessionTracker:ZooKeeperServer@610] - Expiring session 0x1001811b4dd0000, timeout of 6000ms exceeded
2021-04-20 14:09:12,714 [myid:] - INFO [NIOWorkerThread-12:ZooKeeperServer@1059] - Invalid session 0x1001811b4dd0000 for client /[0:0:0:0:0:0:0:1]:52534, probably expired
2021-04-20 14:09:15,300 [myid:] - INFO [NIOWorkerThread-2:ZooKeeperServer@1059] - Invalid session 0x1001811b4dd0000 for client /127.0.0.1:52588, probably expired
2021-04-20 14:44:33,545 [myid:] - WARN [NIOWorkerThread-12:NIOServerCnxn@364] - Unexpected exception
EndOfStreamException: Unable to read additional data from client, it probably closed the socket: address = /[0:0:0:0:0:0:0:1]:52591, session = 0x1001811b4dd0017
at org.apache.zookeeper.server.NIOServerCnxn.handleFailedRead(NIOServerCnxn.java:163)
at org.apache.zookeeper.server.NIOServerCnxn.doIO(NIOServerCnxn.java:326)
at org.apache.zookeeper.server.NIOServerCnxnFactory$IOWorkRequest.doWork(NIOServerCnxnFactory.java:522)
at org.apache.zookeeper.server.WorkerService$ScheduledWorkRequest.run(WorkerService.java:154)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:630)
at java.base/java.lang.Thread.run(Thread.java:831)
2021-04-20 14:44:33,828 [myid:] - INFO [SessionTracker:ZooKeeperServer@610] - Expiring session 0x1001811b4dd0017, timeout of 6000ms exceeded
2021-04-20 14:44:37,050 [myid:] - INFO [NIOWorkerThread-1:ZooKeeperServer@1059] - Invalid session 0x1001811b4dd0017 for client /127.0.0.1:52844, probably expired


2021-04-20 14:09:15,300 [myid:] - INFO [NIOWorkerThread-2:ZooKeeperServer@1059] - Invalid session 0x1001811b4dd0000 for client /127.0.0.1:52588, probably expired
2021-04-20 14:44:33,545 [myid:] - WARN [NIOWorkerThread-12:NIOServerCnxn@364] - Unexpected exception
EndOfStreamException: Unable to read additional data from client, it probably closed the socket: address = /[0:0:0:0:0:0:0:1]:52591, session = 0x1001811b4dd0017
at org.apache.zookeeper.server.NIOServerCnxn.handleFailedRead(NIOServerCnxn.java:163)
at org.apache.zookeeper.server.NIOServerCnxn.doIO(NIOServerCnxn.java:326)
at org.apache.zookeeper.server.NIOServerCnxnFactory$IOWorkRequest.doWork(NIOServerCnxnFactory.java:522)
at org.apache.zookeeper.server.WorkerService$ScheduledWorkRequest.run(WorkerService.java:154)
at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:630)
at java.base/java.lang.Thread.run(Thread.java:831)

KAFKA:


[2021-04-20 14:44:33,544] INFO Client session timed out, have not heard from server in 1458059ms for sessionid 0x1001811b4dd0017, closing socket connection and attempting reconnect (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:33,571] WARN Attempting to send response via channel for which there is no open connection, connection id 10.0.0.29:9092-10.0.0.29:52604-14 (kafka.network.Processor)
[2021-04-20 14:44:33,861] INFO Opening socket connection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error) (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:35,561] WARN Attempting to send response via channel for which there is no open connection, connection id 10.0.0.29:9092-10.0.0.29:52600-13 (kafka.network.Processor)
[2021-04-20 14:44:37,048] INFO Socket connection established to 127.0.0.1/127.0.0.1:2181, initiating session (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,055] WARN Unable to reconnect to ZooKeeper service, session 0x1001811b4dd0017 has expired (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,056] INFO Unable to reconnect to ZooKeeper service, session 0x1001811b4dd0017 has expired, closing socket connection (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,057] INFO EventThread shut down for session: 0x1001811b4dd0017 (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,066] INFO [ZooKeeperClient] Session expired. (kafka.zookeeper.ZooKeeperClient)
[2021-04-20 14:44:37,071] INFO [ZooKeeperClient] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2021-04-20 14:44:37,072] INFO Initiating client connection, connectString=localhost:2181 sessionTimeout=6000 watcher=kafka.zookeeper.ZooKeeperClient$ZooKeeperClientWatcher$@5026735c (org.apache.zookeeper.ZooKeeper)
[2021-04-20 14:44:37,092] INFO Opening socket connection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error) (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,092] INFO Creating /brokers/ids/0 (is it secure? false) (kafka.zk.KafkaZkClient)
[2021-04-20 14:44:37,093] INFO Socket connection established to 127.0.0.1/127.0.0.1:2181, initiating session (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,098] INFO Session establishment complete on server 127.0.0.1/127.0.0.1:2181, sessionid = 0x1001811b4dd0018, negotiated timeout = 6000 (org.apache.zookeeper.ClientCnxn)
[2021-04-20 14:44:37,103] INFO Result of znode creation at /brokers/ids/0 is: OK (kafka.zk.KafkaZkClient)
[2021-04-20 14:44:37,105] INFO Registered broker 0 at path /brokers/ids/0 with addresses: ArrayBuffer(EndPoint(10.0.0.29,9092,ListenerName(PLAINTEXT),PLAINTEXT)) (kafka.zk.KafkaZkClient)
[2021-04-20 14:44:37,117] INFO Creating /controller (is it secure? false) (kafka.zk.KafkaZkClient)
[2021-04-20 14:44:37,121] INFO Result of znode creation at /controller is: OK (kafka.zk.KafkaZkClient)
[2021-04-20 14:44:37,154] INFO [ReplicaAlterLogDirsManager on broker 0] Added fetcher for partitions List() (kafka.server.ReplicaAlterLogDirsManager)

```
Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Checkout>cd C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
Created topic "test".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic response_topic
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic "response_topic".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic post_book
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic "post_book".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic getGroups
Created topic "getGroups".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>
```

```
Command Prompt
or 1 --partitions 1 --topic test
Created topic "test".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-fact
or 1 --partitions 1 --topic response_topic
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issu
es it is best to use either, but not both.
Created topic "response_topic".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-fact
or 1 --partitions 1 --topic post_book
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issu
es it is best to use either, but not both.
Created topic "post_book".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-fact
or 1 --partitions 1 --topic getGroups
Created topic "getGroups".

C:\kafka_2.11-1.1.0\kafka_2.11-1.1.0\bin\windows>
```

```
npm
This is groups user is part of [ { groupsPartOf: [ 'Group 310', 'GROCERY' ] } ]
after handle{ groupsPartOf: [ 'Group 310', 'GROCERY' ] }
producer.send { response_topic: { '0': 2 } }
message received for getGroups  { handle_request: [Function: handle_request] }
"{\"correlationId\":\"3633e309b7653418385e6756a2b28def\",\"replyTo\":\"response_topic\",\"data\":{}}"
data in server.js
In handle request: for get groups{}
undefined
This is groups user is part of []
after handle
producer.send { response_topic: { '0': 3 } }
message received for getGroups  { handle_request: [Function: handle_request] }
"{\"correlationId\":\"b73af7ba8e267013d98644e34c0b34e5\",\"replyTo\":\"response_topic\",\"data\":{\"email\":\"logan1@gma
il.com\"}}"
data in server.js
In handle request: for get groups{"email":"logan1@gmail.com"}
logan1@gmail.com
This is groups user is part of [ { groupsPartOf: [ 'Group 310', 'GROCERY' ] } ]
after handle{ groupsPartOf: [ 'Group 310', 'GROCERY' ] }
producer.send { response_topic: { '0': 4 } }
message received for getGroups  { handle_request: [Function: handle_request] }
"{\"correlationId\":\"3cb9431d8034b6a6808e04960eb629c7\",\"replyTo\":\"response_topic\",\"data\":{\"email\":\"logan1@gma
il.com\"}}"
data in server.js
In handle request: for get groups{"email":"logan1@gmail.com"}
logan1@gmail.com
This is groups user is part of [ { groupsPartOf: [ 'Group 310', 'GROCERY' ] } ]
after handle{ groupsPartOf: [ 'Group 310', 'GROCERY' ] }
producer.send { response_topic: { '0': 5 } }
```

SAMPLE SCREEN SHOTS OF DOCUMENTS

Users:


```
_id: ObjectId("60832e86e3b5e3874cf60e46")
phonenumber: "123-456-7890"
currency: "USD"
timezone: "(GMT-08:00) Pacific Time (US & Canada)"
language: "English"
photostring: "https://splitwiseyusuf123.s3.us-east-2.amazonaws.com/bff2588e0d7f9fb69..."
› groupsPartOf: Array
› groupsInvitedTo: Array
name: "Emily Blunt"
email: "eb@gmail.com"
password: "$2a$10$g29ag7NYbUW8DftTiDfW2OLZwNXTEzeG6r35GVlfPXv6P2caTb5i."
__v: 0
```


Bills:


```
_id: ObjectId("608335eeb4baed8cf0df1c2b")
bill_amount: 20
bill_desc: "Event 1"
created_by: "et@gmail.com"
created_in: "Team Event"
created_time: 2021-04-23T21:02:38.112+00:00
__v: 0
```


Groups:


```
_id: ObjectId("60833147e3b5e3874cf60e4b")
group_photostring: "default.jpg"
› bills: Array
› members: Array
› invitedMembers: Array
group_name: "Team Event"
created_by: "et@gmail.com"
created_time: 2021-04-23T20:42:47.368+00:00
__v: 0
```

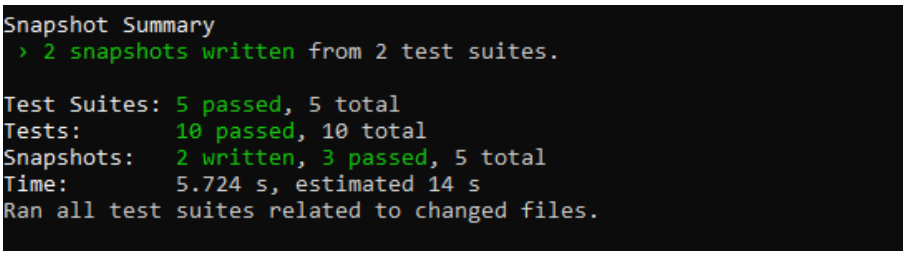

Transaction:


```
_id: ObjectId("608335eeb4baed8cf0df1c2c")
transaction_amount: 4
sender: "et@gmail.com"
transaction_in: "Team Event"
receiver: "et@gmail.com"
__v: 0
```

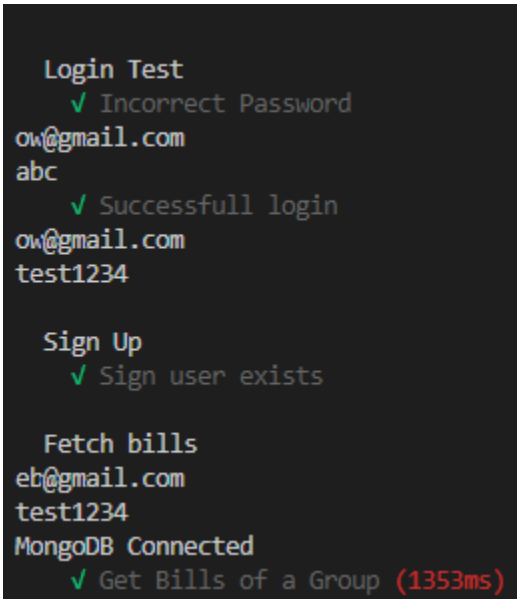
Comments:

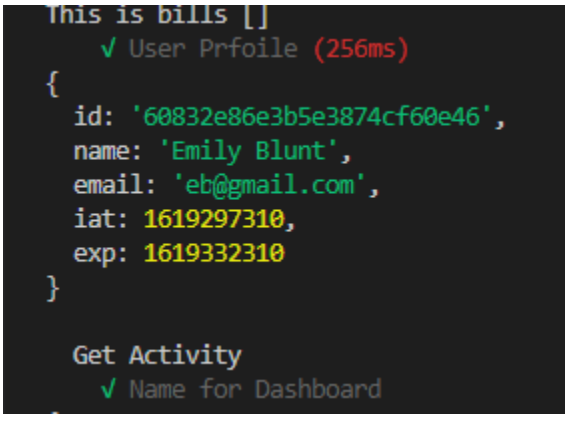
TESTING:

1: Frontend Testing using React Testing Library (Summary of Tests)



2: Mocha and Chai Backend Testing:

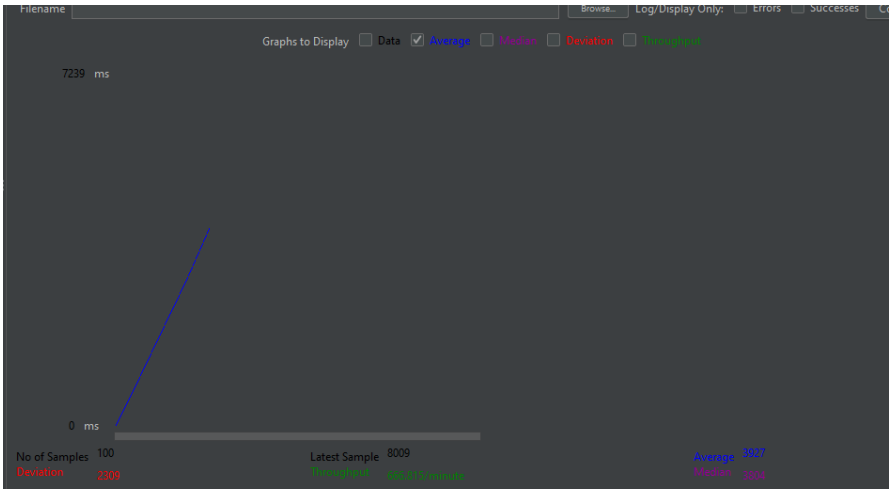




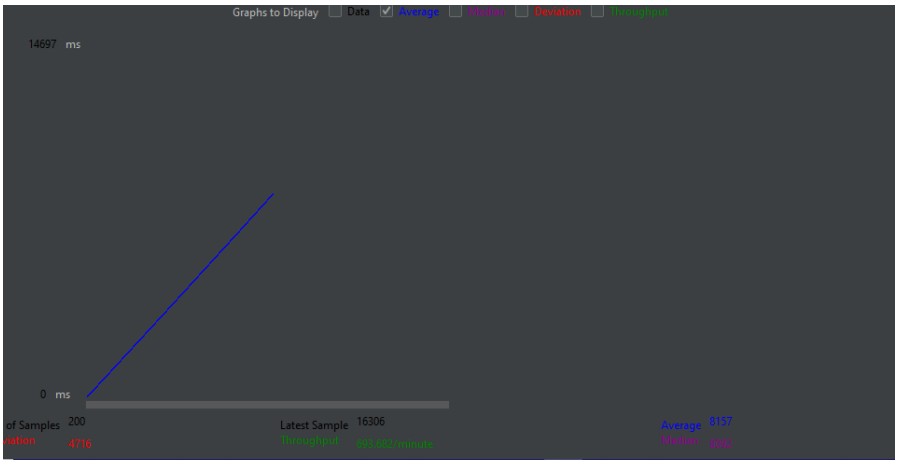
3: JMeter testing.

<u>BEFORE POOLING:</u>

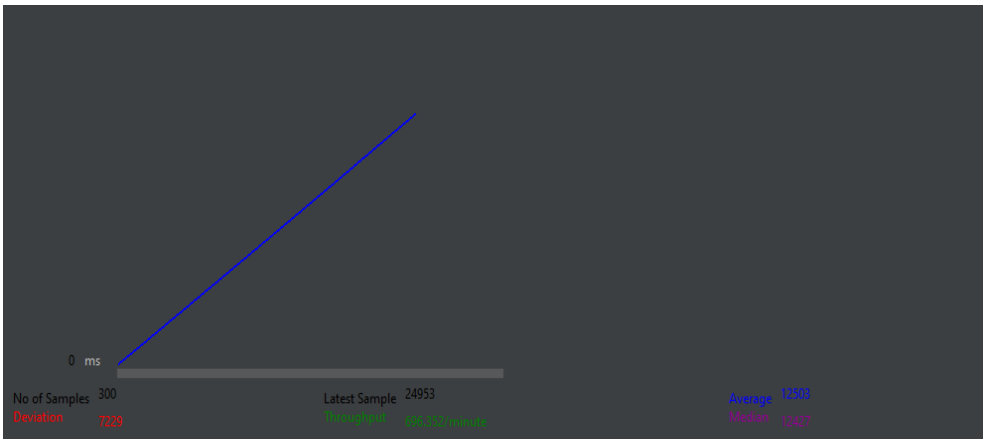
100 Requests



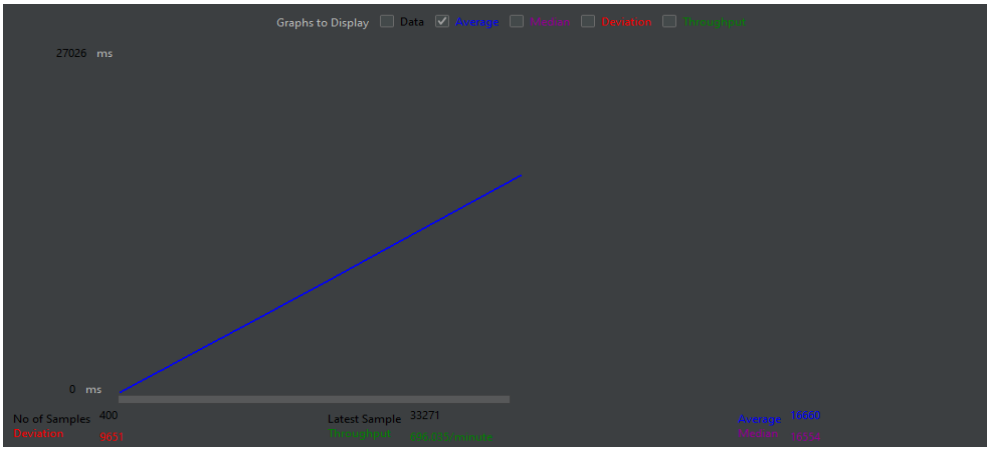
2: 200 Requests



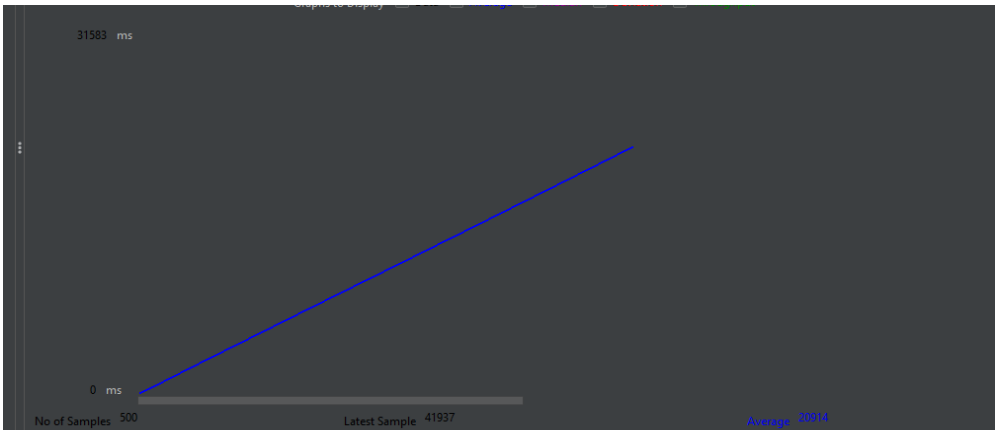
3: 300 Requests



4: 400 Requests

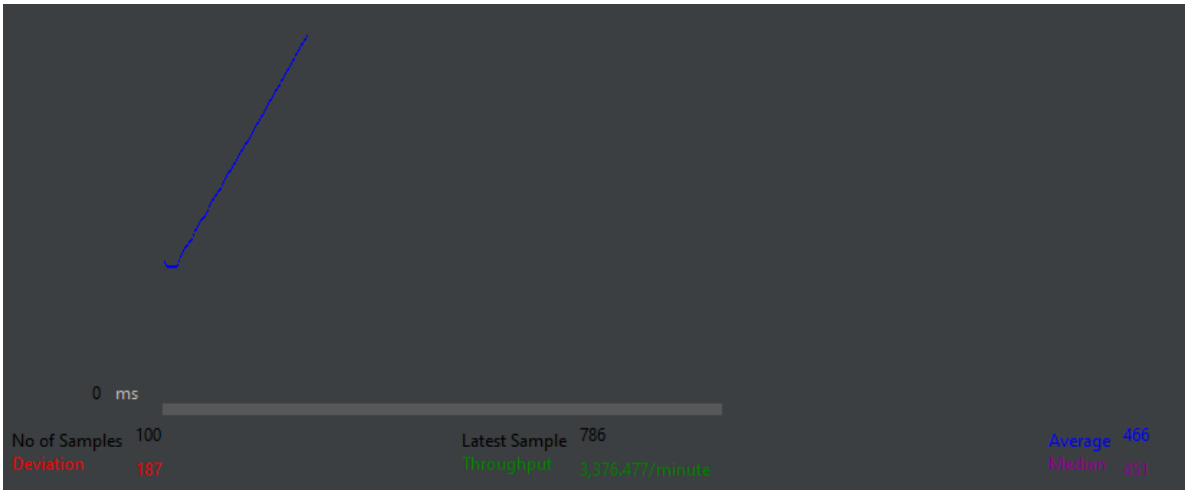


5: 500 Requests

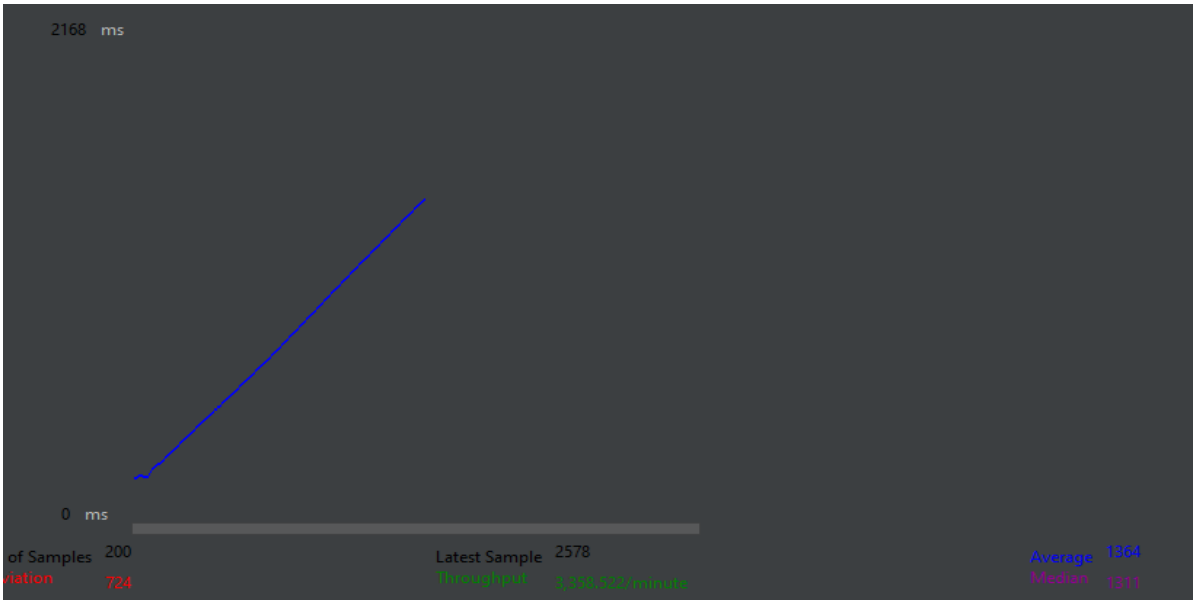


AFTER POOLING:

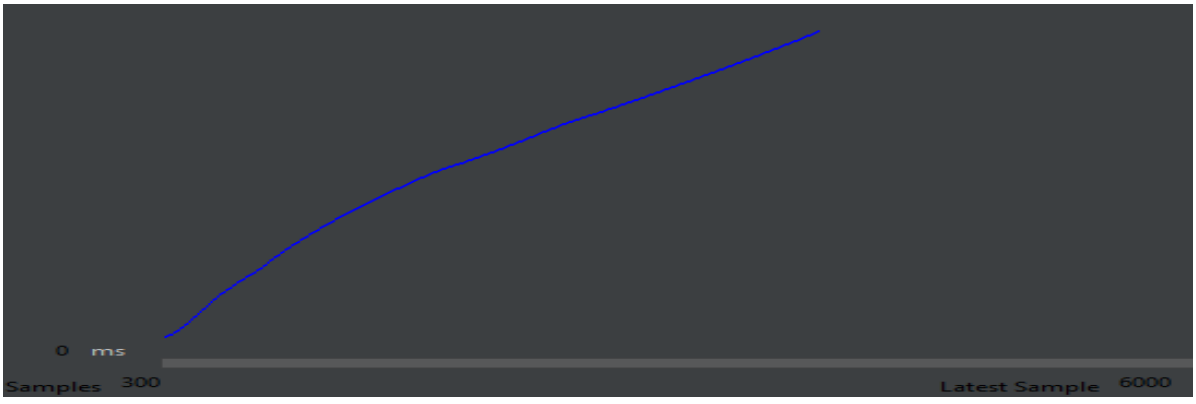
1:100 Requests



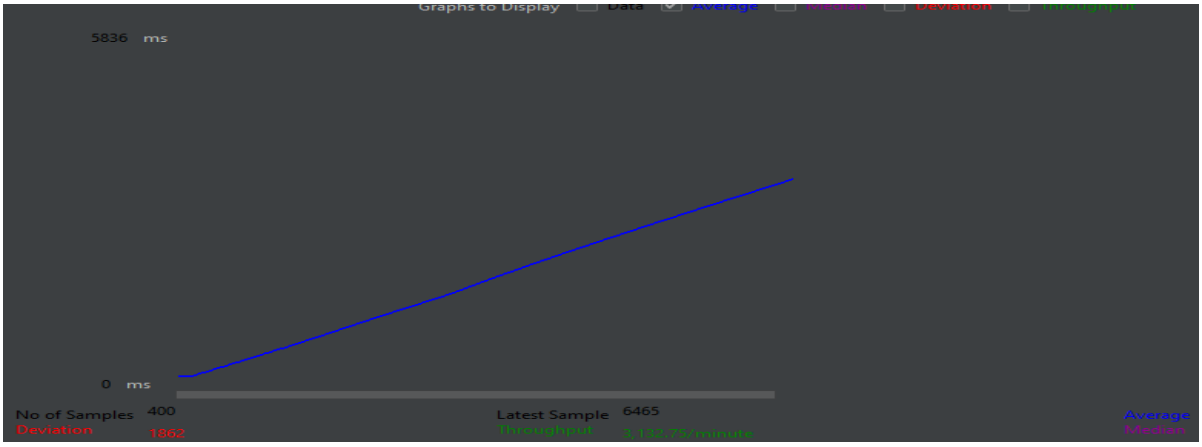
2:200 Requests



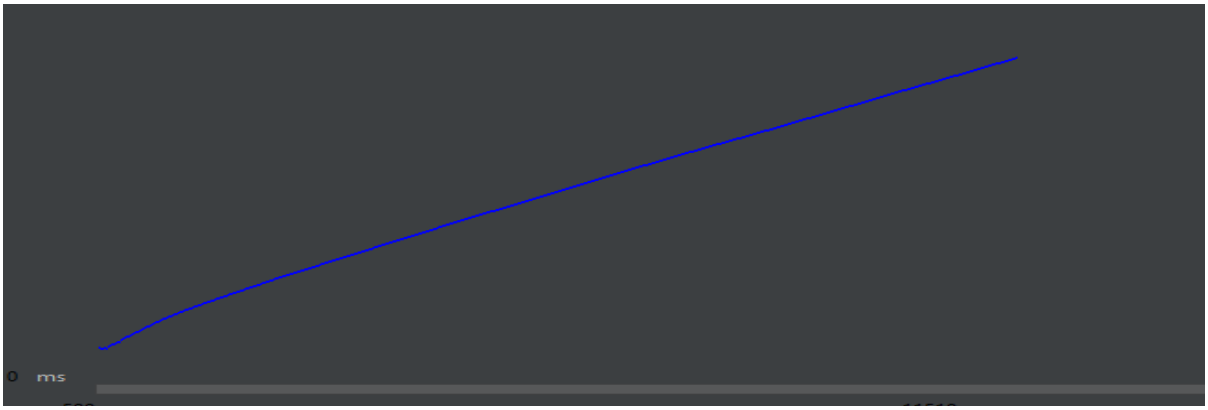
3: 300 Requests



4: 400 Requests



5:500 Requests



QUESTIONS:

1. Compare passport authentication process with the authentication process used in Lab1.

Lab 1 used session and cookie-based authentication. Cookie based authentication is stateful. i.e. an authentication record must be maintained both on the server as well as the client side. The server needs to keep a note of active sessions in a database, while on the front-end a cookie is created that holds a session identifier. The drawback of this approach is that cookies take up memory in the database.

Lab 2 uses a token-based authentication approach. Authentication is mostly done based on JSON Web Tokens (JWTs). While there are many ways to implement tokens, JWTs are the most popular. Token-based authentication is stateless. The server does not keep a record of which users are logged in. Instead, every request to the server is accompanied by a JWT token which the server uses to verify the authenticity of the request. The token is generally sent as an authorization header in the form of Bearer {JWT}. The flow would be as follows: User Logs In=>Server verifies the credentials and returns a signed token (Token is generally stored in Local Storage) => All following requests are made with authorization header set to token.

2. Compare performance with and without Kafka. Explain in detail the reason for difference in performance.

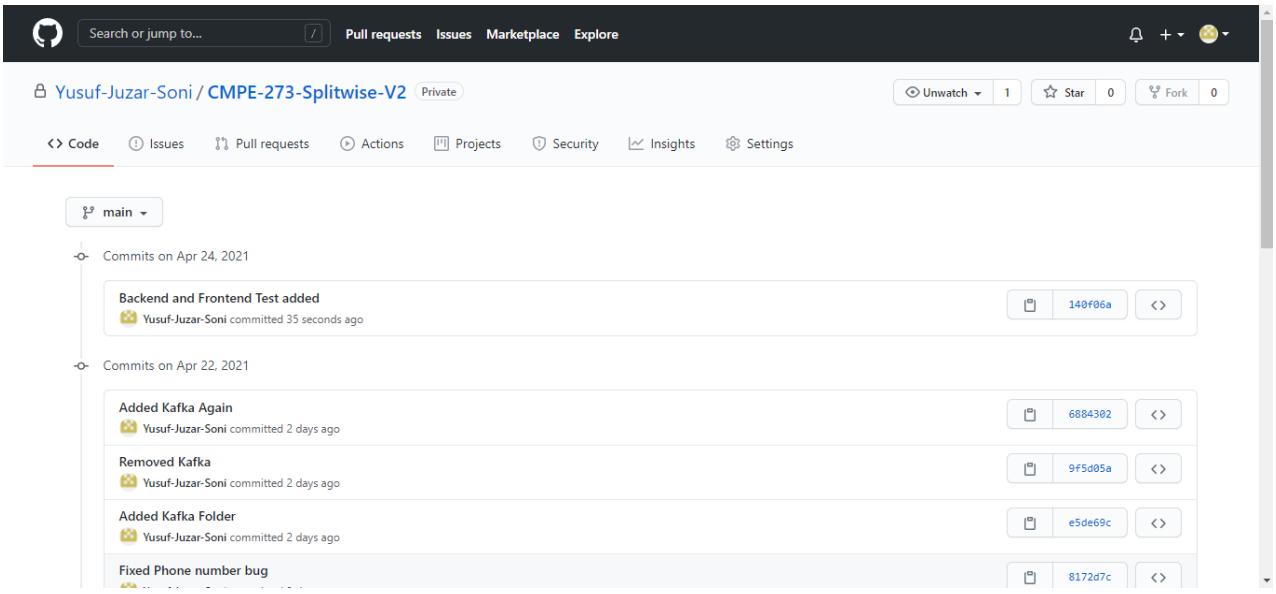
Kafka is a stream processing software that acts as a broker between two parties the consumer and the consumer. It is a pub-sub messaging system which allows exchanging of data between applications, servers,

and processors as well. This process allows different applications to limit the parallel execution of the data, where one record executes without waiting for the output of the previous record. Therefore, Kafka enables the user to simplify the task of the streaming process and parallel execution. Kafka provides a plethora of benefits like asynchronous execution, low latency, high throughput, fault tolerance and easy scalability. Moreover, since Kafka acts as a broker it also maintains order between producer and consumer. Usually producers are faster than consumers, had they been communicating directly, there would be loss of data if the consumer is slower than the producer. In Kafka, consumer processes messages only when they are available. No component system is ever stalled waiting for another.

3. If given an option to implement MySQL and MongoDB both in your application, specify which part of data of the application you will store in MongoDB and MySQL respectively.

MongoDB is a NoSQL database having advantage in storing key-value pair kind of data. MongoDB does not enforce any kind of structure in its documents. It in fact encourages denormalization. If you require some data in a particular document you are encouraged to put the data there, as opposed to SQL where we have to maintain relationships between tables via foreign keys. In our case, modules like Bills, Transactions and Comments are ideal MongoDB candidates as it is easier to put associated data from other collections into them. On the other hand, modules like Users and Groups that have a many to many relationships amongst them and have a relatively fixed structure are good candidates for SQL Databases.

GIT COMMIT HISTORY



Commits on Apr 19, 2021

leave group and Comments Done
Yusuf-Juzar-Soni committed 5 days ago — 30b1e19

Commits on Apr 18, 2021

Frontend code, comments and leavegroup pending
Yusuf-Juzar-Soni committed 6 days ago — 6e4a6eb

Commits on Apr 17, 2021

Complete APIs
Yusuf-Juzar-Soni committed 7 days ago — f74228b

Backend Code Iteration 1
Yusuf-Juzar-Soni committed 7 days ago — 6b65d1f

Commits on Apr 6, 2021

setting up intial code
Yusuf-Juzar-Soni committed 18 days ago — 33c07ad

Initial commit
Yusuf-Juzar-Soni committed 18 days ago — Verified — 70752bf