COMPUTER NETWORKS Project

**UBERPOOL**

# 

# Team members -

Harish P B PES1201701435 5F

Pruthvish Eshwar PES1201701629 5F

Ashrith Sridhar PES1201700649 5F

# 

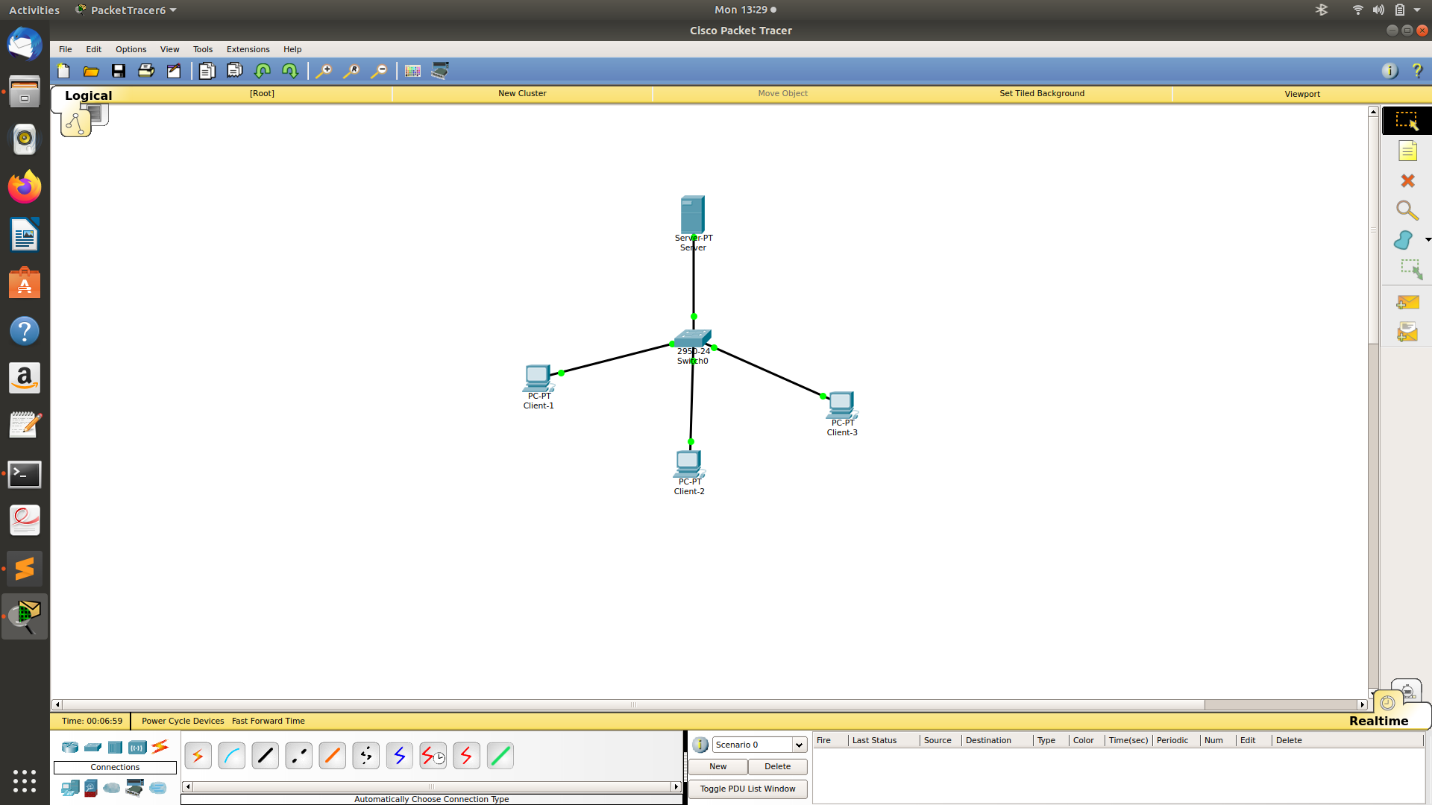
# Problem Statement

Implementation of uberpool or carpooling

# Description

A main server is system which handles all the request and keeps track of the status from consumers ( passenger ) as well as producers ( driver ), it is implemented using socket programming and multithreading in python to handle the problem of critical section and communicate between hosts. The client needs to register to offer a ride or search for a ride. He can offer a ride with username, source, destination and seats available. The consumer can search for ride based on the producer’s manhattans distance and number of seats available.

## Topology



## Modules developed

key modules implemented -

* Registration and login of producers and consumers
* Authentication of producer/consumer before login
* Making a system independent user login
* Book-keep module which stores all the related information about each consumer and producer

## Multiple scenarios

1. consumer has registered but there are no drivers registered at the moment

* if there are no registered drivers
* The consumer will be notified with status of ‘ No drivers available at the moment ‘ message

1. consumer has registered but there are no drivers are free or available at the moment

* If the number of seats he requested is more than the available seats
* The consumer will be notified with status of ‘ No drivers available at the moment ‘ message

1. producer has registered and logged in but there are no customers are available at the moment

* If all the customers have already been allocated to other drivers or there are no customers at the moment
* The producer will be notified with status of ‘ No customers available ‘ message

1. There are than one customers

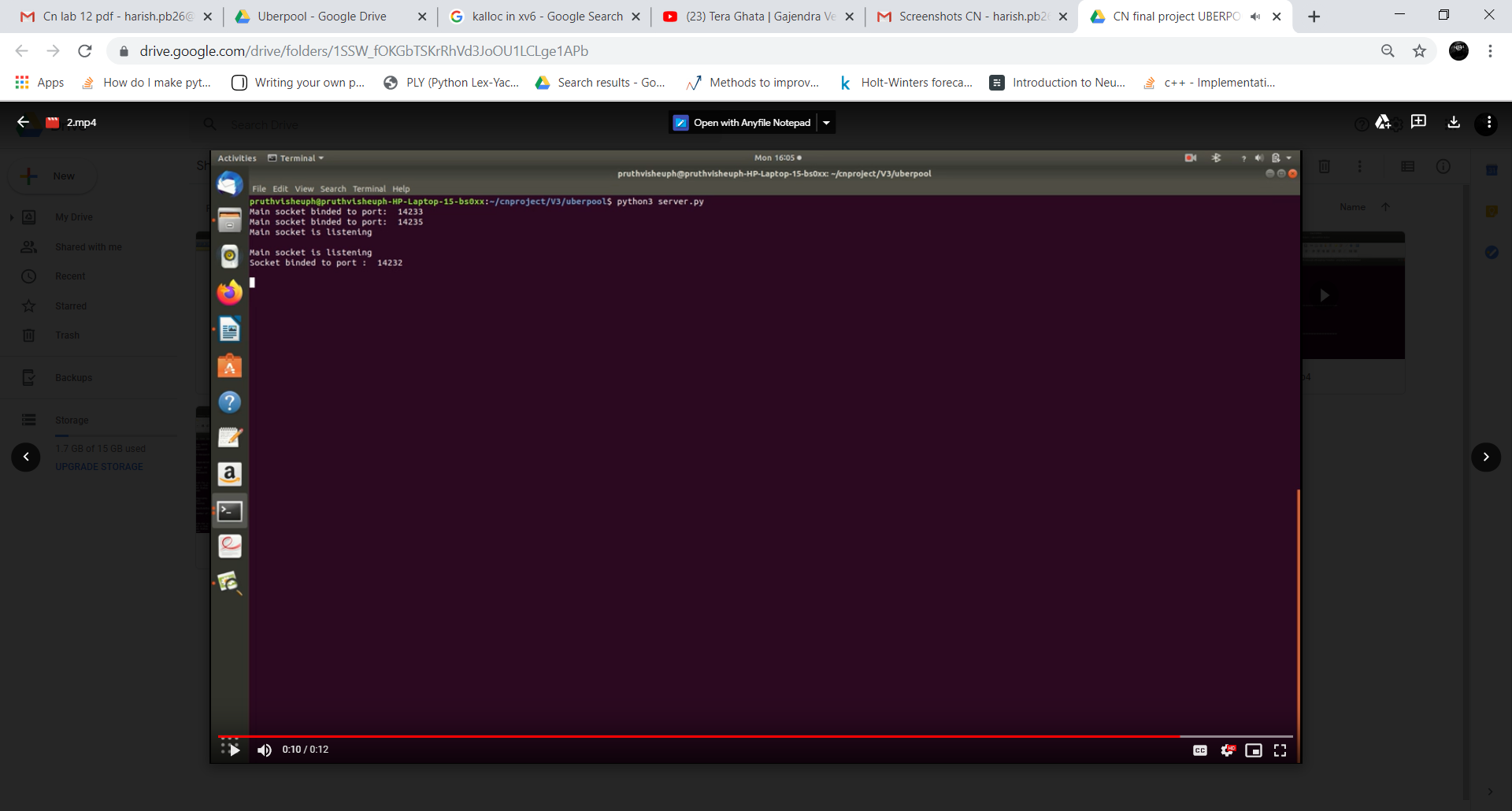
* The customers will be allocated to one or more producers based on the manhattans distance and no of seats the producer offers
* The producer and consumer will be notified with status of ‘ Driver:- <nameofdriver>

customers:-

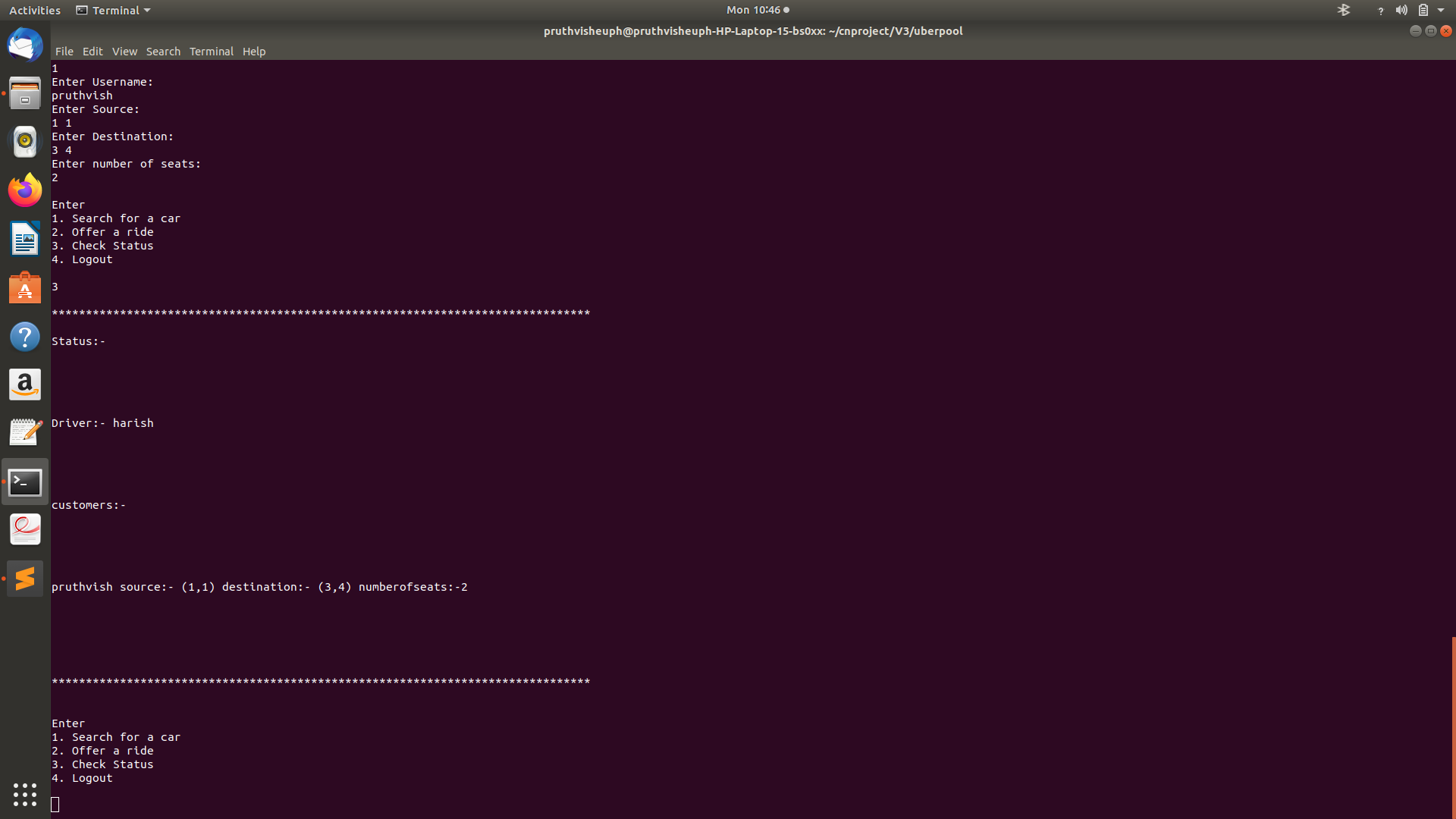
<nameofcustomer> source:- <source> destination:- <destination> numberofseats:- <num> ‘ message

## Sample input and output

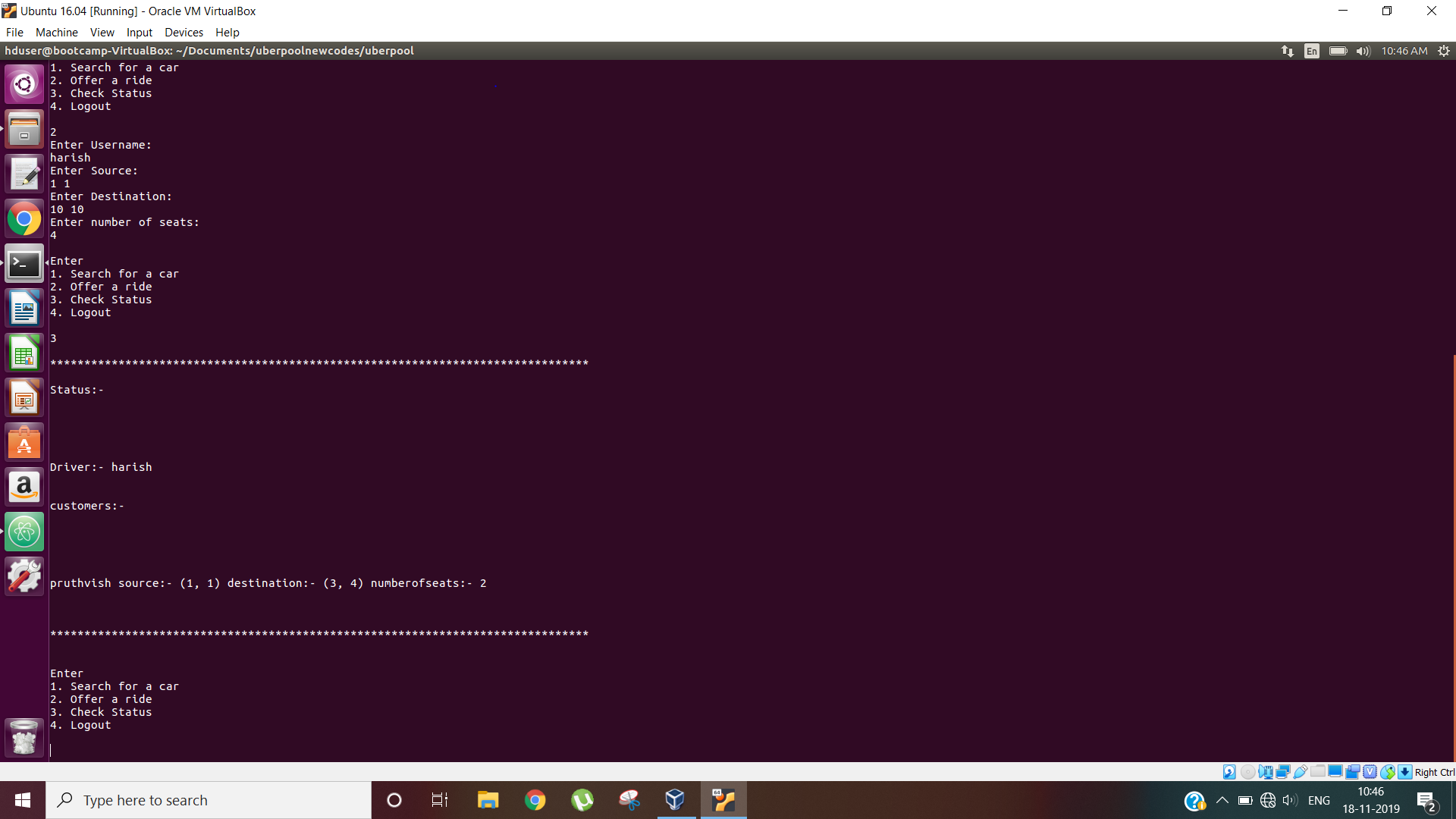
Server –



Consumer ( Client ) –



Producer ( Driver ) –



## Networks concepts used

* Socket Programming
* Open System Interconnection Layers
* API
* protocols
* encryption and decryption

## 

## Other Concepts used

* Deadlock prevention and handling multiple connections from producer and consumer to the server by using locks.
* How to assign driver to different consumer while satisfying the constraints using graph algorithms.
* Headers – the information about each consumer and producer is stored in consumer database and producer database which stores information like email, source, destination and numberofseats offered/required database here is a simple text file which can be accessed only through authentication.

## Problems

* Non Persistent storage – server loses all the data when connection is lost,that is if the server program stops all the regitered users will not be stored
* We can overcome this problem by using sql databases and loading data on startup or start of the server every time but this could increase load time and might slow down the server initially and users might need to wait for some time to proccess their transaction.

## Video URL

* <https://drive.google.com/drive/folders/1SSW_fOKGbTSKrRhVd3JoOU1LCLge1APb>