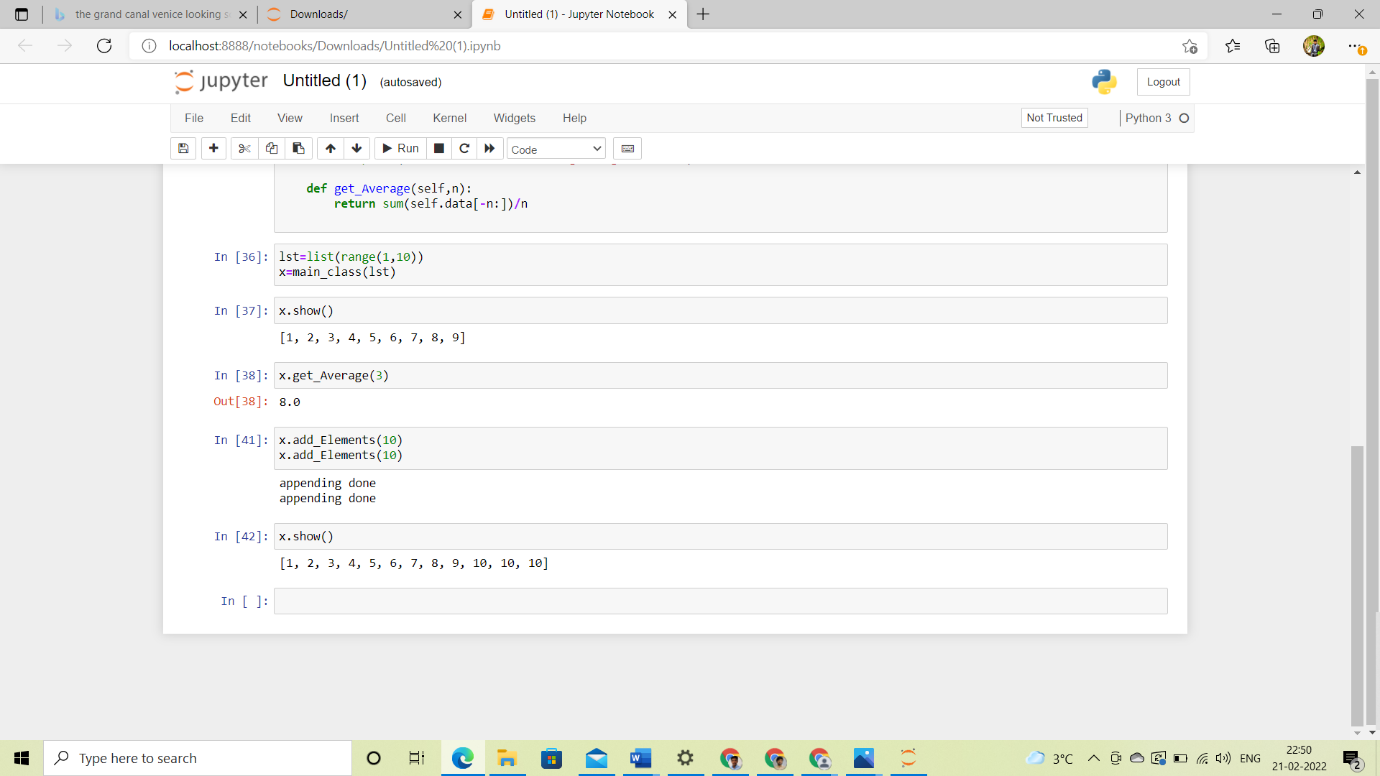
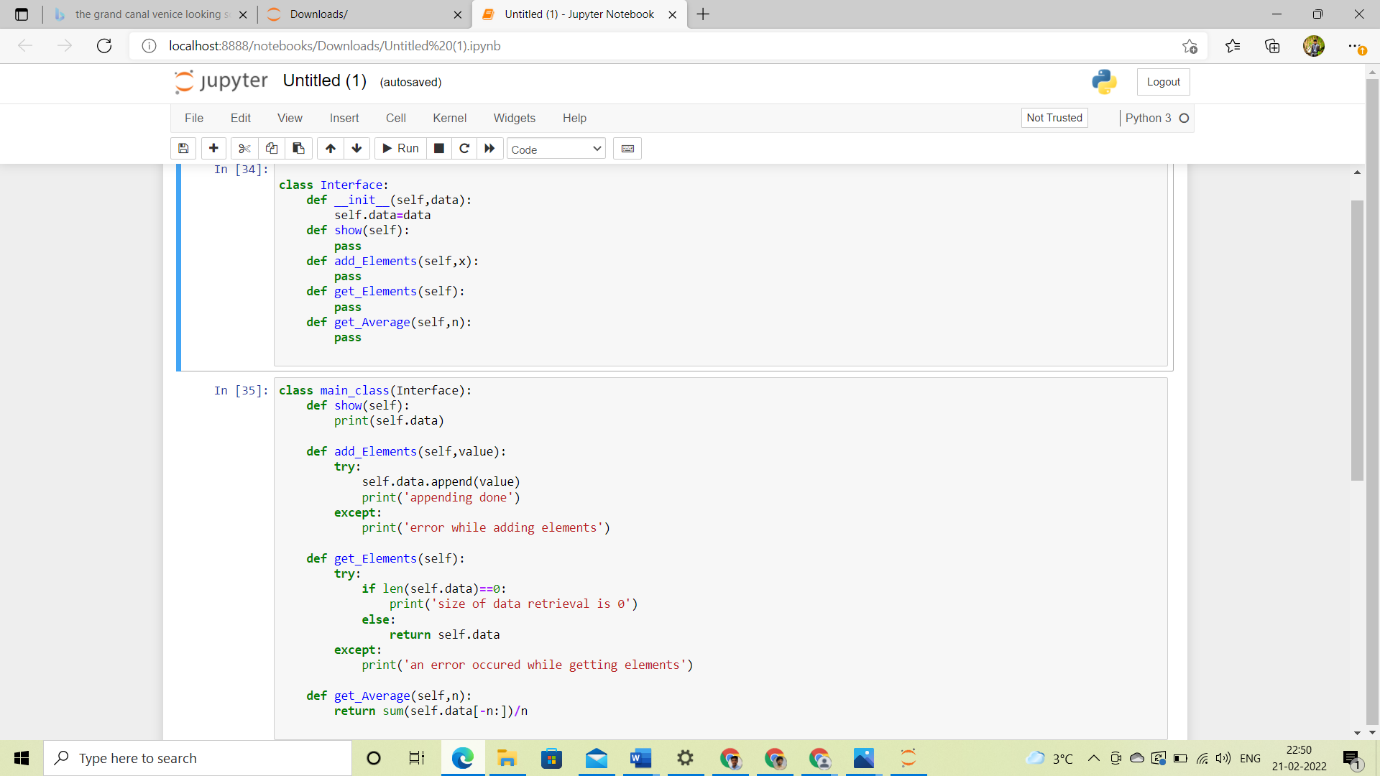
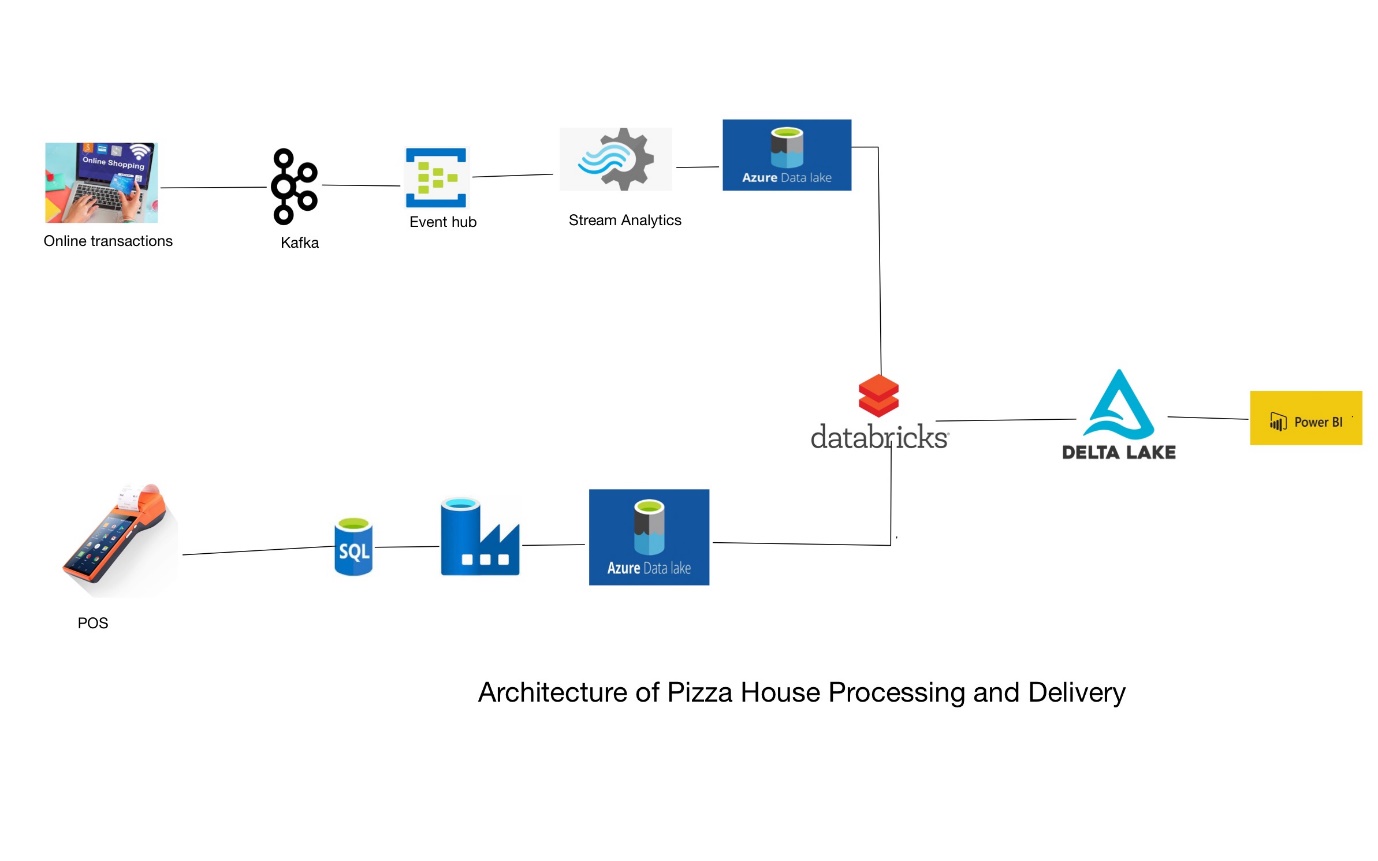
**Coding Question:**

****

**Bond Challenge design question**



Event producers: here we have two types of event producers

* online transactions
* POS (point of sale)

Here streaming data is managed by Apache Kafka.

Sending events to event hubs or receiving events from event hubs is managed by the DOT net console

**Let’s discuss sending data from event hubs to Azure data lake:**

Event hubs reside in **namespaces.** An event hub can have one or more partitions, a partition is an ordered sequence of events that are held in an event hub. As new events arrive, they are added to the end of the sequence, a partition can be thought as of a commit log.

Event producers can provide a PARTITION KEY to target a partition or events can be inserted into partitions in a round-robin manner. Then we will have consumer groups, A CONSUMER GROUP is a view of an entire event, consumer groups enable multiple consuming applications to have a separate view of the event stream.

An application that reads or subscribes to the event data is called an event receiver, each receiver will receive data from each consumer group

AZURE EVENT HUBS ENABLE US AUTOMATICALLY CAPTURE THE STREAMING DATA TO AN AZURE BLOB STORAGE OR DATA LAKE USING STREAM ANALYTICS.

So, data from event hubs will be stored in Azure data lake by using azure stream analytics.

**Sending POS data to azure data lake:**

we can send batch data from SQL server to azure data lake by using azure data factory activities with the help of triggers.

**Reading data from Azure data bricks and saving it into delta table:**

After the two data sources are stored in the data lake, we have to read them from data bricks by creating mounting both by using key vaults.

Initially, we need to define schemas & initialize **widgets** with respect to the storage locations, so that we can define data frames for streaming data & batch data.

Then we can start analyzing the data and have to do all the required transformations and joins for streaming data. And then for batch data.

so, that we can finally create two data frames one for streaming data and the other for batch data.

from here we will use **SPARK STRUCTURED STREAMING** to create a join between streaming data and batch data by using micro-batch processing by triggering the streaming job to load for every 15 min. so that we can perform a join between those two data frames and create a final data frame for our analyzing or visualizing purpose.

Then we have to register the data frame with hive meta store as a temporary view so that we can directly query the data using spark SQL.

Then by performing an appropriate SQL query we have to build an aggregated table.

Then, we have to save that aggregated table as a **delta table** using overwrite mode.

connect delta lake to power bi to build visualizations and configure them to refresh as scheduled.