BOSTON COMMUNITY EVENT TRACKER DATABASE

Milestone: Project Report

Group 12

Rutuja Jadhav Gayatri Nair

857-390-1757 (Tel of Student 1)

857-421-9112 (Tel of Student 2)

jadhav.rutu@northeastern.edu
nair.gaya@northeastern.edu

Percentage of Effort Contributed by Student1 : 50

Percentage of Effort Contributed by Student2 : 50

Signature of Student 1: Rutuja J

Signature of Student 2: Gayatri N

Submission Date : 8 December 24

USE CASE STUDY REPORT

Group No.: Group 12

Student Names: Rutuja Jadhav & Gayatri Nair

Executive Summary:

Being a newcomer in Boston everyone was keen on exploring this beautiful city. However, it was challenging to find good event spots and events happening in and around the city. Moreover, since Boston is a huge educational hub, it has a larger student population. And since this city has so much to offer, people want to go about and spend their time experiencing the city. If you are into history, education, sports and love a bustling city life with a side of East Coast charm, then this city has it all. Except maybe a place to see and get notified about all the happening events in the city. This is the motivation behind taking it up as our Data Management for Analytics project case study.

I. Introduction

Current event tracking platforms often target global audiences or large-scale events, leaving local and community-focused needs underserved. The concept is aimed at localizing various events including flea markets, concerts, music festivals, museum events, art shows, exhibitions, sports events, restaurants and cafes within a single place. More like a Boston city calendar map for customers. In this project we capture various details about local events around the city.

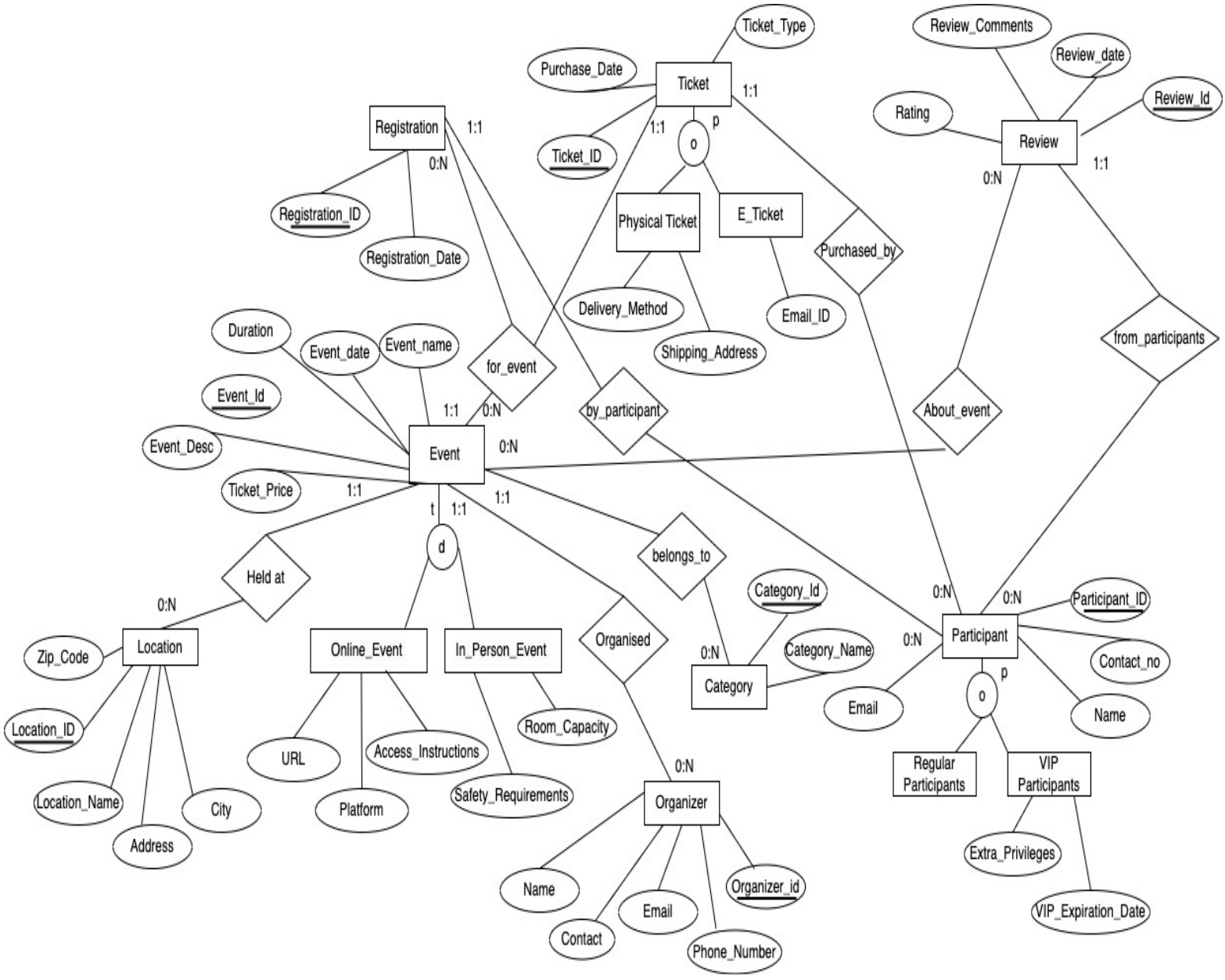
People can check the calendar and book various events according to their choice and even request the same event again. Company needs to record basic information about the events, location, organizers, and participants. For each event the company needs to record event name, event description, event date, duration, locations, category, organizer, ticket price, max participants. For location, location name, address, city, zip code will be needed. For organizers, we require their name, contact, email, Phone number. Company needs participants' basic information such as name, email and contact number.

To keep the record about the participants for multiple events we require registrations, tickets, categories of events and review from participants for future

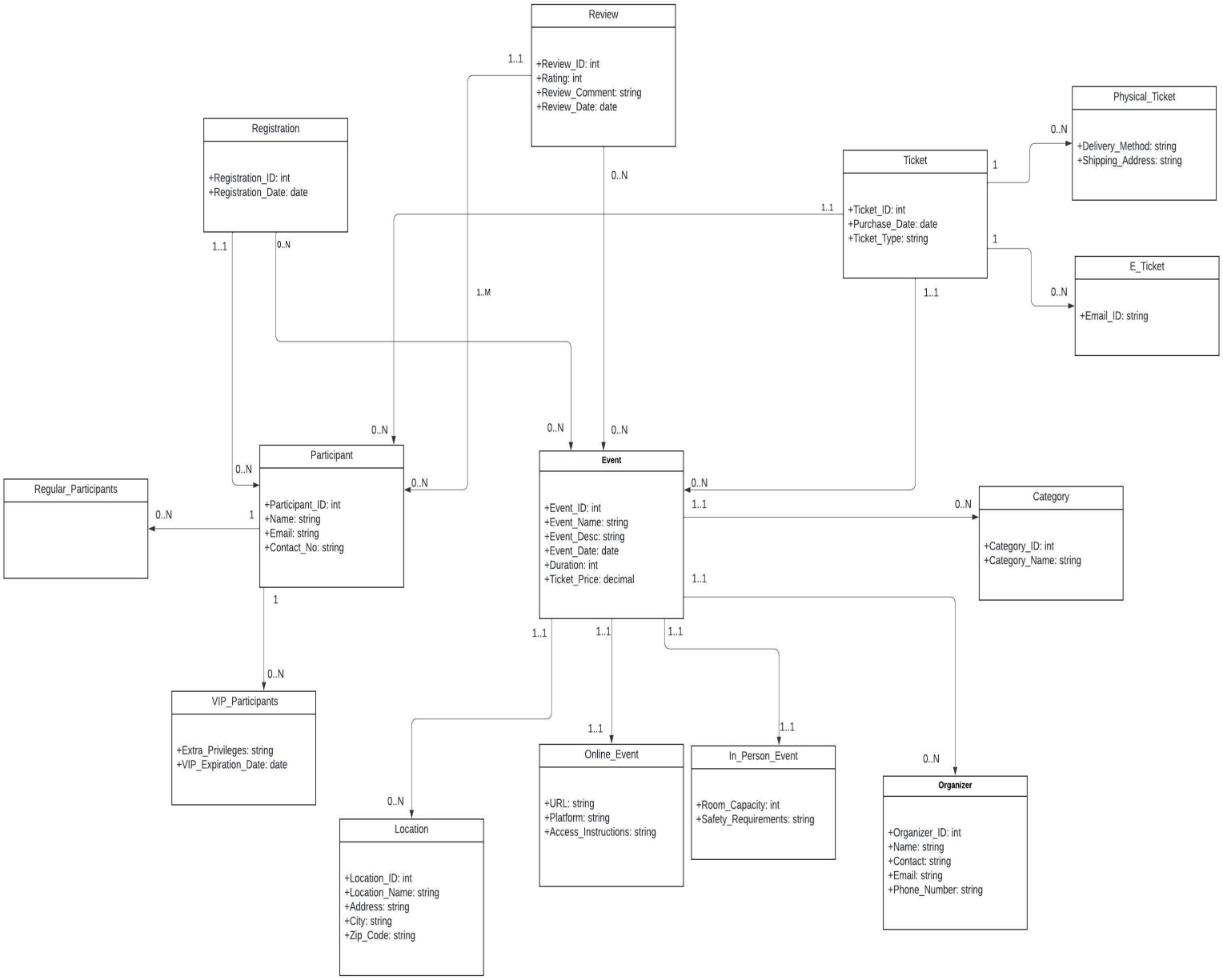
events. For registrations we need registration id, event id, participants id, registration date. For tickets, Ticket id, event id, Ticket type, ticket price, purchase date. For categories, the company requires information about category names such as music, sports, theater. Reviews from the participants would be useful for the company's future events and potential future participants. For review companies will need Event id, Participant id, Rating, Review comment, reviewdate

II. Conceptual Data Modeling

1. EER Diagram



2. UML Diagram



III. Mapping Conceptual Model to Relational Model

Primary Key- Underlined **Foreign Key- *Italicized***

Event (Event_Id, Event_name, Event_Desc, Event_date, Ticket_Price, Duration, Organizer_Id, Category_Id, Location_ID)
• **Primary Key**: Event_Id
• **Foreign Keys:**
o Organizer_Id references Organizer (Organizer_Id) Not Null
o Category_Id references Category (Category_Id) Not Null
o Location_ID references Location (Location_ID) (Null Allowed)
• **Constraints:**
o Event_name, Event_Desc, Event_date, Ticket_Price, Organizer_Id, Category_Id should be NOT NULL

Category (Category_Id, Category_Name)
• **Primary Key**: Category_Id
• **Constraints:**
o Category_Name should be NOT NULL

Location (Location_ID, Location_Name, Address, City, Zip_Code)
• **Primary Key**: Location_ID
• **Constraints**:
o Location_Name, Address, City, Zip_Code should be NOT NULL

Online_Event (Event_Id, URL, Platform, Access_Instructions)
• **Primary Key**: Event_Id
• **Foreign Key**:
o Event_Id references Event (Event_Id) NOT NULL
• **Constraints**:
o URL, Platform should be NOT NULL

In_Person_Event (Event_Id, Room_Capacity, Safety_Requirements)
• **Primary Key**: Event_Id
• **Foreign Key**:
o Event_Id references Event (Event_Id) NOT NULL
• **Constraints**:
o Room_Capacity, Safety_Requirements can be NULL

Organizer (Organizer_Id, Name, Email, Contact, Phone_Number)
• **Primary Key**: Organizer_Id
• **Constraints**:
o Name, Email should be NOT NULL

Registration (Registration_ID, Registration_Date, Event_Id)
• **Primary Key**: Registration_ID
• **Foreign Key**:
o Event_Id references Event (Event_Id) NOT NULL

• **Constraints**:
o Registration_Date, Event_Id should be NOT NULL

Ticket (Ticket_ID, Purchase_Date, Delivery_Method, Ticket_Type, Event_Id, Participant_ID)
• **Primary Key**: Ticket_ID
• **Foreign Keys**:
o Event_Id references Event (Event_Id) NOT NULL
o Participant_Id references Participant (Participant_id)
• **Constraints**:Purchase_Date, Ticket_Type, Event_Id, Participant_ID should be NOT NULL

Participant (Participant_ID, Name, Contact no, Email)
• **Primary Key**: Participant_ID
• **Constraints**:
o Name, Contact_no, Email should be NOT NULL

VIP_Participant (Participant_ID, Extra_Privileges, VIP_Expiration_Date)
• **Primary Key**: Participant_ID
• **Foreign Key**:
o Participant_ID references Participant (Participant_ID) NOT NULL
• **Constraints**:
o Extra_Privileges and VIP_Expiration_Date can be NULL

Review (Review_Id, Review_Comments, Review_Date, Rating, Event_Id, Participant_ID)
• **Primary Key: Review_Id**
• **Foreign Keys:**
o Event_Id references Event (Event_Id) NOT NULL
o Participant_ID references Participant (Participant_ID) NOT NULL

Event_Review (Event_Id, Review_Id)
• **Foreign Keys**
o Event_Id references Event (Event_Id) NOT NULL
o Review_Id references Review (Review_Id) NOT NULL

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:

The database was created in MySQL and the following queries were performed:

Query 1: What is the SQL command to retrieve all data from the Category table?

Select * From Category;

category_id	category_name
1	Entertainment
2	Sports
3	History & Culture
4	Food & Drink
5	Art & Exhibitions
6	Festivals & Celebrations
7	Education
8	Technology
NULL	NULL

Query 2: Which events have a ticket price greater than $40, and how can you fetch their names and prices?

```
SELECT Event_name, Ticket_Price
FROM Event
WHERE Ticket_Price > 40;
```

Event_name	Ticket_Price
Boston Jazz Festival	50.00
Boston Film Festival	45.00
Boston Foodie Tour	60.00
Symphony Hall Concert	80.00
North End Food Tour	45.00
Boston Comic Con	45.00
Boston Wine Expo	50.00
Boston Ballet Nutcracker	75.00

Query 3: How can you calculate the average and maximum rating from the Review table?

```
SELECT AVG(Rating) AS
Average_Rating, MAX(Rating) AS
Max_Rating
FROM Review;
```

Average_Rating	Max_Rating
3.95000	5.0

Query 4: How can you use an INNER JOIN to link event names to their respective categories?

```
SELECT E.Event_name, C.Category_Name
FROM Event E
INNER JOIN Category C ON E.Category_Id = C.Category_Id;
```

Event_name	Category_Name
Boston Harbor Boat Tour	Entertainment
Beacon Hill Holiday Stroll	Entertainment
Boston Marathon	Sports
Fenway Park Open Day	Sports
Head of the Charles Reg…	Sports
Boston Dragon Boat Fest…	Sports
St. Patrick's Day Parade	Sports
Freedom Trail Walking Tour	History & Culture
Boston Harborfest	History & Culture
New Year's Eve on the H…	History & Culture
Boston Pride Parade	History & Culture

Query 5: How can a LEFT OUTER JOIN help fetch event names even if their location is missing?

```
SELECT E.Event_name, L.Location_Name
FROM Event E
LEFT OUTER JOIN Location L ON E.Location_ID = L.Location_ID;
```

Event_name	Location_Name
Boston Chocolate Tour	Boston Chocolate Fact…
First Night Boston	Virtual Event
Freedom Trail Lantern Tour	Freedom Trail
Boston Marathon Expo	Virtual Event
Seaport Beer Festival	Virtual Event
Boston Urban Gardens Tour	South Boston
Italian Heritage Parade	South Boston
Boston Harbor Boat Tour	Boston Harbor
Beacon Hill Holiday Stroll	Beacon Hill
Cambridge Science Festival	Harvard University

Query 6: How can you retrieve the names of organizers who manage events where the ticket price exceeds $30?

```
SELECT E.Event_name, L.Location_Name
FROM Event E
LEFT OUTER JOIN Location L
ON E.Location_ID = L.Location_ID;
```

Organizer_Name
John Doe
Liam Brown
Ethan White
Charlotte King
John Doe
Oliver Brown
Elijah Black
Mia Blue
Archer Black
Madison White
Dylan Green
David Green

Query 7: How can you find participants who have more than two reviews?

```
SELECT P.Participant_ID, P.Name
FROM Participant P
WHERE 2 < (
   SELECT COUNT(*)
   FROM Review R
   WHERE P.Participant_ID = R.Participant_ID
);
```

Participant_ID	Name
15	Alexander Lee
39	Lucas Ramirez
NULL	NULL

Query 8: How can you find events with ticket prices greater than or equal to the ticket price of all events?

Event_name	Ticket_Price
Symphony Hall Concert	80.00

```
SELECT Event_name, Ticket_Price
FROM Event
WHERE Ticket_Price >= ALL (
   SELECT Ticket_Price
   FROM Event
);
```

Query 9: How can you fetch events with ticket prices higher than any event priced below $30?

Event_name	Ticket_Price
North End Food Tour	45.00
Boston Arts Festival	10.00
Boston Comic Con	45.00
Boston Fashion Week	25.00
Boston Wine Expo	50.00
Boston Ballet Nutcracker	75.00
Fenway Halloween Bash	25.00
Summer Concert Series	20.00
Boston Chocolate Tour	35.00
Freedom Trail Lantern Tour	20.00
Boston Marathon Expo	10.00
Seaport Beer Festival	30.00
Boston Urban Gardens Tour	15.00
Boston Harbor Boat Tour	40.00
Cambridge Science Festival	5.00

```
SELECT Event_name, Ticket_Price
FROM Event
WHERE Ticket_Price > ANY (
   SELECT Ticket_Price
   FROM Event
   WHERE Ticket_Price < 30
);
```

Query 10: How can you fetch the event IDs and names for events that have reviews with ratings of 4 or more?

Event_Id	Event_name
12	Boston Foodie Tour
22	Boston Comic Con
9	Boston Film Festival
3	Freedom Trail Walking Tour
18	North End Food Tour
27	Boston Ballet Nutcracker
1	Boston Jazz Festival
35	Seaport Beer Festival
10	Duck Boat Tour
29	St. Patrick's Day Parade
8	Museum of Fine Arts Night
26	Boston Wine Expo
4	Shakespeare in the Park
NULL	NULL

```
  SELECT Event_Id, Event_name
FROM Event E
WHERE EXISTS (
   SELECT 1
   FROM Review R
   WHERE R.Event_Id = E.Event_Id AND R.Rating >= 4
                );
```

Query 11: How can you find events that do not have any associated reviews?

Event_Id	Event_name
2	Boston Marathon
6	Fenway Park Open Day
11	Boston Harborfest
13	Oktoberfest Boston
15	Boston Tech Expo
17	New Year's Eve on the Harbor
19	Boston Pride Parade
21	Head of the Charles Regatta
23	Boston Fashion Week
24	Boston Dragon Boat Festival
25	Patriots' Day Celebration
28	Fenway Halloween Bash
30	Summer Concert Series
33	Freedom Trail Lantern Tour
34	Boston Marathon Expo

```
SELECT Event_Id,Event_name
FROM Event E
WHERE NOT EXISTS (
   SELECT 1
   FROM Review R
   WHERE R.Event_Id = E.Event_Id
                );
```

Query 12: How can you use the UNION operation to retrieve event names with ticket prices above $40 and organizer names that start with the letter 'A'?

```
SELECT Event_name AS Name
FROM Event
WHERE Ticket_Price > 40
UNION
SELECT Name AS Name
FROM Organizer
WHERE Name LIKE 'A%';
```

Name
Boston Jazz Festival
Boston Film Festival
Boston Foodie Tour
Symphony Hall Concert
North End Food Tour
Boston Comic Con
Boston Wine Expo
Boston Ballet Nutcracker
Ava Blue
Archer Black
Adeline King
Alexander Blue

Query 13: How can you calculate and display the average ticket price for each event using a subquery?

```
SELECT Event_name,
    (SELECT AVG(Ticket_Price)
     FROM Event E2
     WHERE E2.Event_Id = E.Event_Id) AS
Average_Ticket_Price
FROM Event E;
```

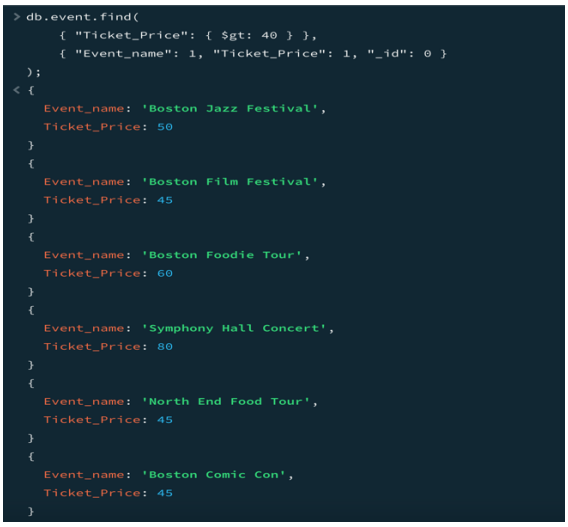
Event_name	Average_Ticket_Pri...
Boston Jazz Festival	50.000000
Boston Marathon	0.000000
Freedom Trail Walking Tour	25.000000
Shakespeare in the Park	15.000000
Boston Seafood Festival	20.000000
Fenway Park Open Day	10.000000
Christmas Tree Lighting	0.000000
Museum of Fine Arts Night	30.000000
Boston Film Festival	45.000000
Duck Boat Tour	35.000000
Boston Harborfest	0.000000
Boston Foodie Tour	60.000000
Oktoberfest Boston	40.000000
Symphony Hall Concert	80.000000
Boston Tech Expo	25.000000

NoSQL Implementation:

The following Mongodb queries were done:

Query 1: Find the names and ticket prices of events where the ticket price is greater than $40.

```
db.event.find(
   { "Ticket_Price": { $gt: 40 } },
   { "Event_name": 1, "Ticket_Price": 1, "_id": 0 }
);
```



Query 2: Find ticket details along with the names and emails of participants who purchased them.

```
db.ticket.aggregate([
   {
      $lookup: {
         from: "participant",
         localField: "Participant_ID",
         foreignField: "Participant_ID",
         as: "participantDetails"
      }
   },
   {
      $lookup: {
         from: "event",
         localField: "Event_Id",
         foreignField: "Event_Id",
         as: "eventDetails"
      }
   },
   {
      $project: {
         Ticket_ID: 1,
         Purchase_Date: 1,
         Ticket_Type: 1,
```

```
            "participantDetails.Name": 1,
            "participantDetails.Email": 1,
            "eventDetails.Event_name": 1
        }
    },
    {
        $limit: 6 // Limit the result to 6 documents
    }
]);
```

```
  _id: ObjectId('674d0366d4c5991706ef6a96'),
  Ticket_ID: 1,
  Purchase_Date: 2024-01-01T00:00:00.000Z,
  Ticket_Type: 'General Admission',
  participantDetails: [
    {
      Name: 'John Smith',
      Email: 'john.smith@example.com'
    }
  ],
  eventDetails: [
    {
      Event_name: 'Boston Jazz Festival'
    }
  ]
}
{
  _id: ObjectId('674d0366d4c5991706ef6a97'),
  Ticket_ID: 2,
  Purchase_Date: 2024-01-02T00:00:00.000Z,
  Ticket_Type: 'VIP',
  participantDetails: [
    {
      Name: 'Emily Johnson',
      Email: 'emily.johnson@example.com'
    }
  ],
  eventDetails: [
```

Query 3: Find the total number of online events grouped by their platform.

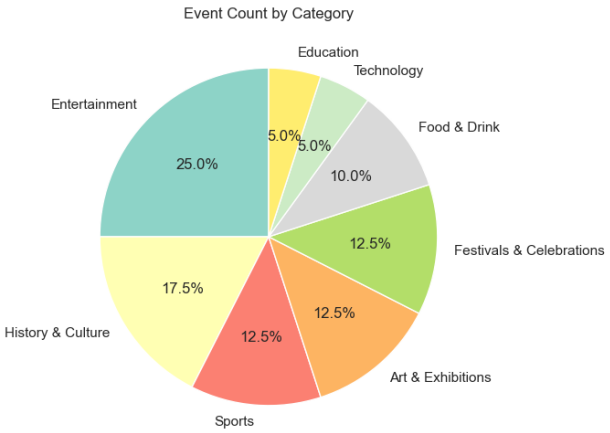
```
]);
{
  _id: 'Zoom',
  totalEvents: 4
}
{
  _id: 'YouTube',
  totalEvents: 2
}
{
  _id: 'WebEx',
  totalEvents: 2
}
{
  _id: 'Facebook Live',
  totalEvents: 1
}
{
  _id: 'Google Meet',
  totalEvents: 1
}
bostontracker>
```

V. Database Access via Python

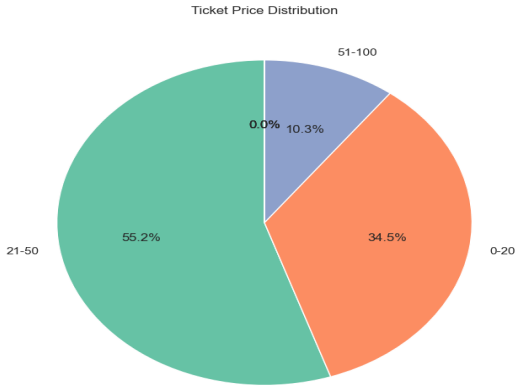
The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using mysql.connector, followed by cursor.excecute to run and fetchall from query, followed by converting the list into a

dataframe using pandas library and using matplotlib to plot the graphs for the analytics.

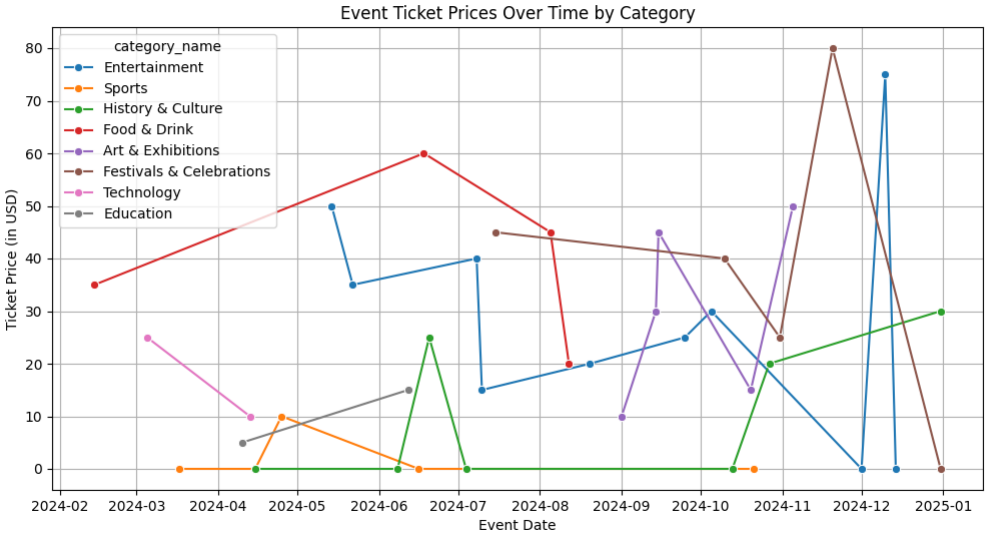
Graph1: Event Count by Category



Graph2: Total Price Distribution



Graph3: Event Ticket Prices Over Time by Category



VII. Summary and Recommendation

A centralized platform to discover, track, and get notified about events in the city is lacking which is where this project comes into picture. Utilizing frameworks such as Scikit-learn and TensorFlow for predictive modeling, alongside sentiment analysis tools like VADER, enables a more comprehensive understanding of the customer's behavior. Interactive dashboards, created with platforms like Tableau or Python libraries such as Dash, combined with audience segmentation through clustering algorithms clearly enhances the precision of marketing strategies. These innovations drive more targeted and effective approaches to audience engagement which in turn can help data driven industries flourish.