PES UNIVERSITY

B.TECH. (CSE)

V SEMESTER

UE19CS301 – DATABASE MANAGEMENT SYSTEM

ASSIGNMENT3

TEAM NO: 13 TOPIC: AIRLINE RESERVATION SYSTEM

SUBMITTED BY

NAME	SRN
SAI JAYANTH IMMADISETTY	PES2UG19CS152
AMIT JITTA	PES2UG19CS169
KUNTAL GORAI	PES2UG19CS198

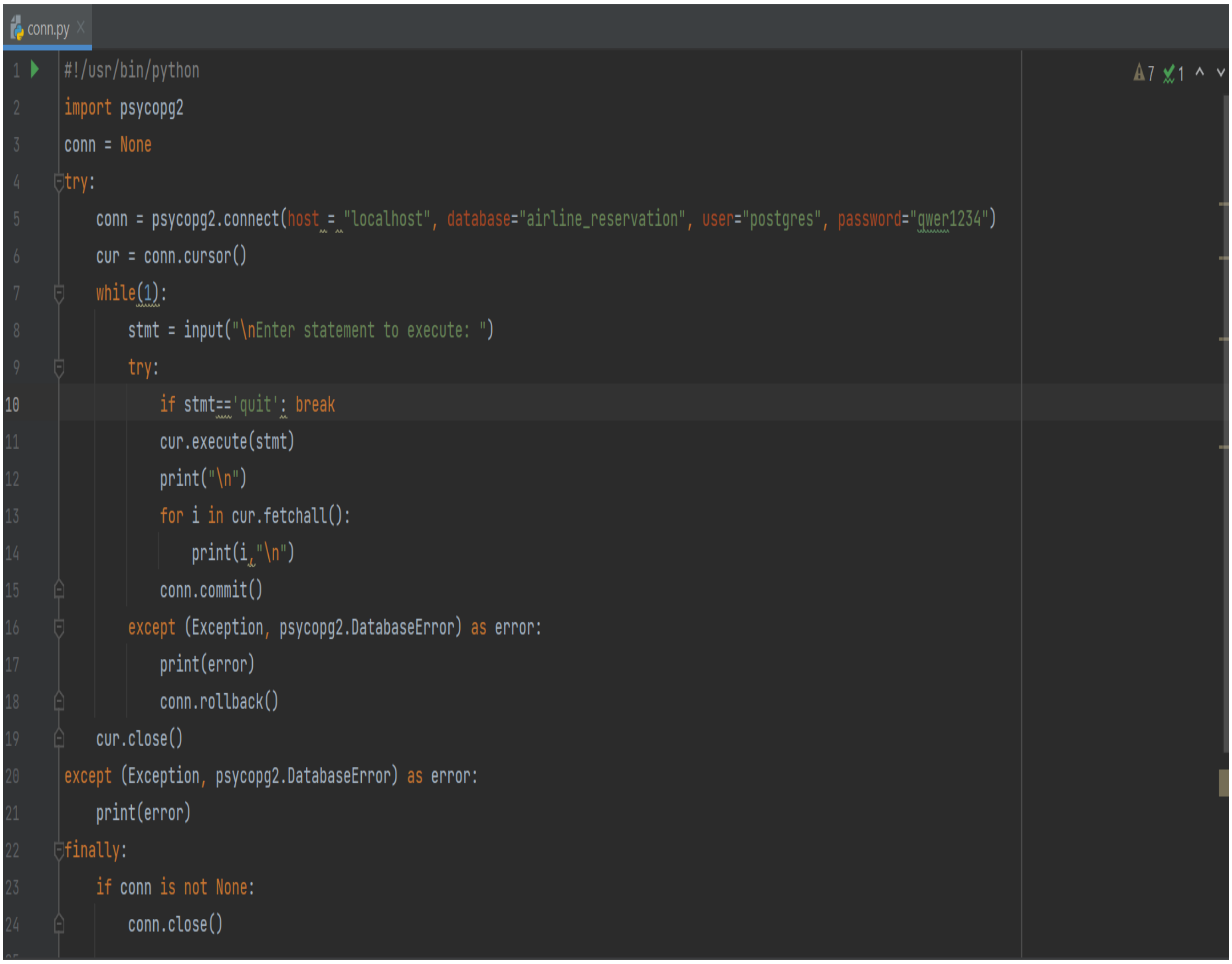
AUGUST– DECEMBER 2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ELECTRONIC CITY CAMPUS,

BENGALURU – 560100, KARNATAKA, INDIA

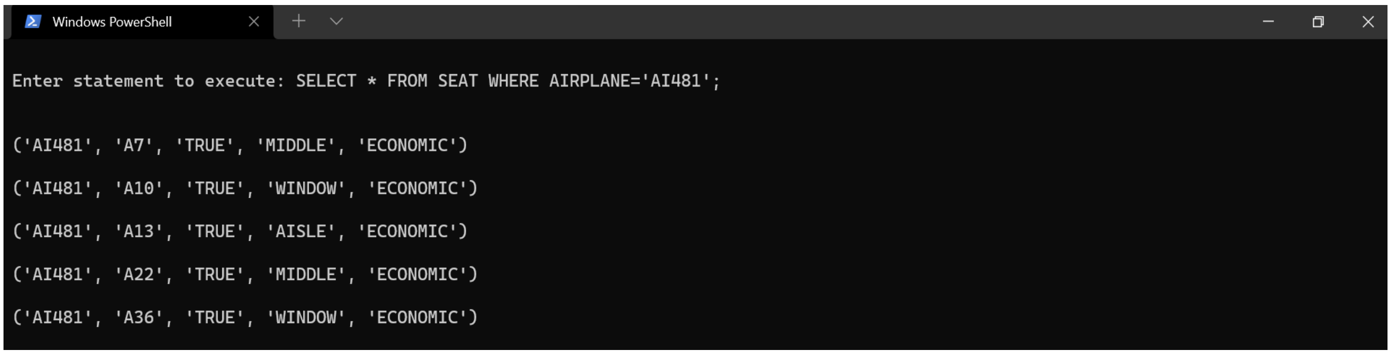
FRONT END PYTHON FLLE



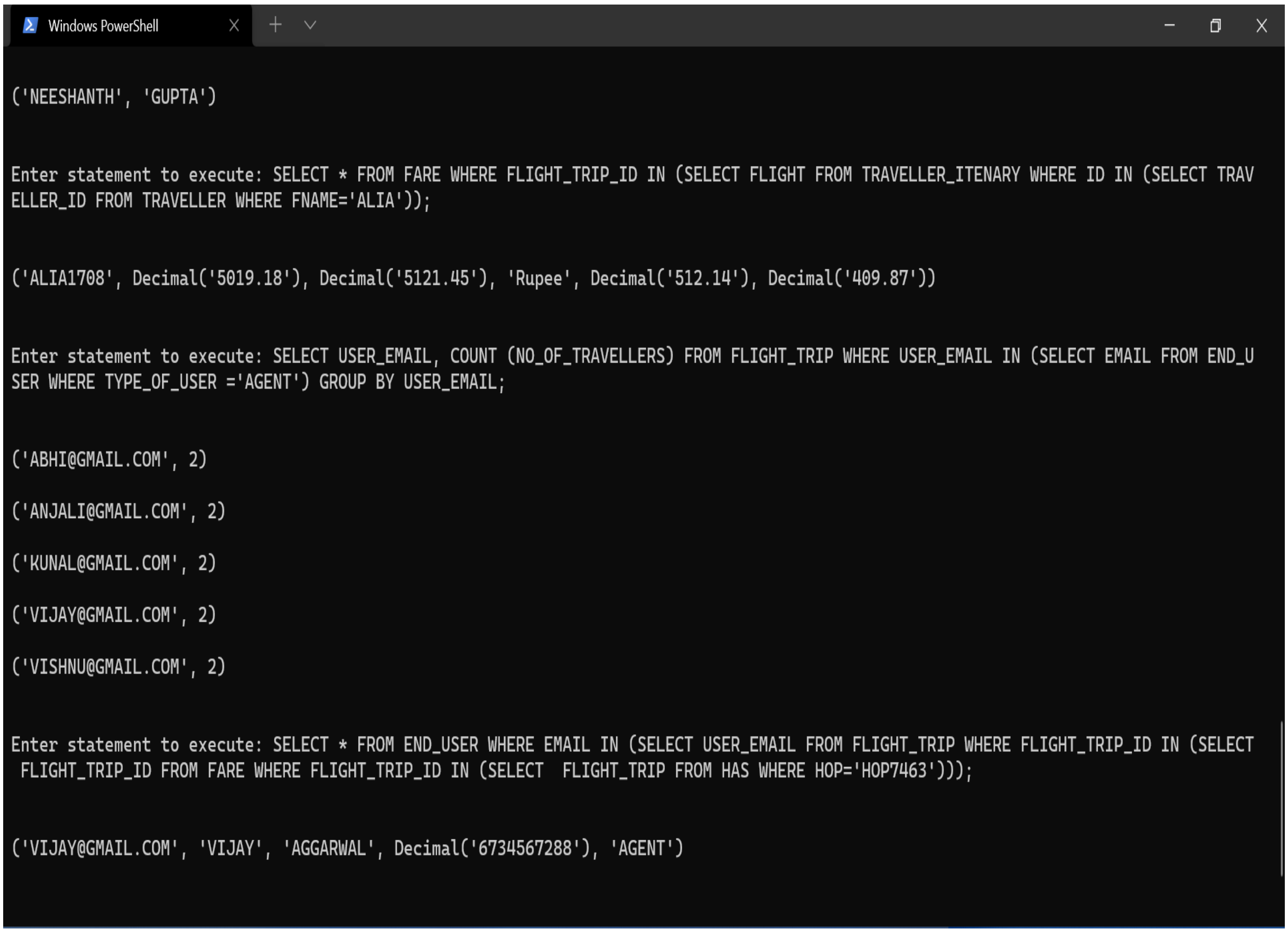
```python
#!/usr/bin/python
import psycopg2
conn = None
try:
    conn = psycopg2.connect(host = "localhost", database="airline_reservation", user="postgres", password="qwer1234")
    cur = conn.cursor()
    while(1):
        stmt = input("\nEnter statement to execute: ")
        try:
            if stmt=='quit': break
            cur.execute(stmt)
            print("\n")
            for i in cur.fetchall():
                print(i,"\n")
            conn.commit()
        except (Exception, psycopg2.DatabaseError) as error:
            print(error)
            conn.rollback()
    cur.close()
except (Exception, psycopg2.DatabaseError) as error:
    print(error)
finally:
    if conn is not None:
        conn.close()
```

The Dependencies that needs to be installed is psycopg2. Psycopg2 is a DB API 2.0 compliant PostgreSQL driver that is actively developed. It is designed for multi-threaded applications and manages its own connection pool. Other interesting features of the adapter are that if you are using the PostgreSQL array data type, Psycopg will automatically convert a result using that data type to a Python list. Psycopg2 is a DB API 2.0 compliant PostgreSQL driver that is actively developed. It is designed for multi-threaded applications and manages its own connection pool. Other interesting features of the adapter are that if you are using the PostgreSQL array data type, Psycopg will automatically convert a result using that data type to a Python list.

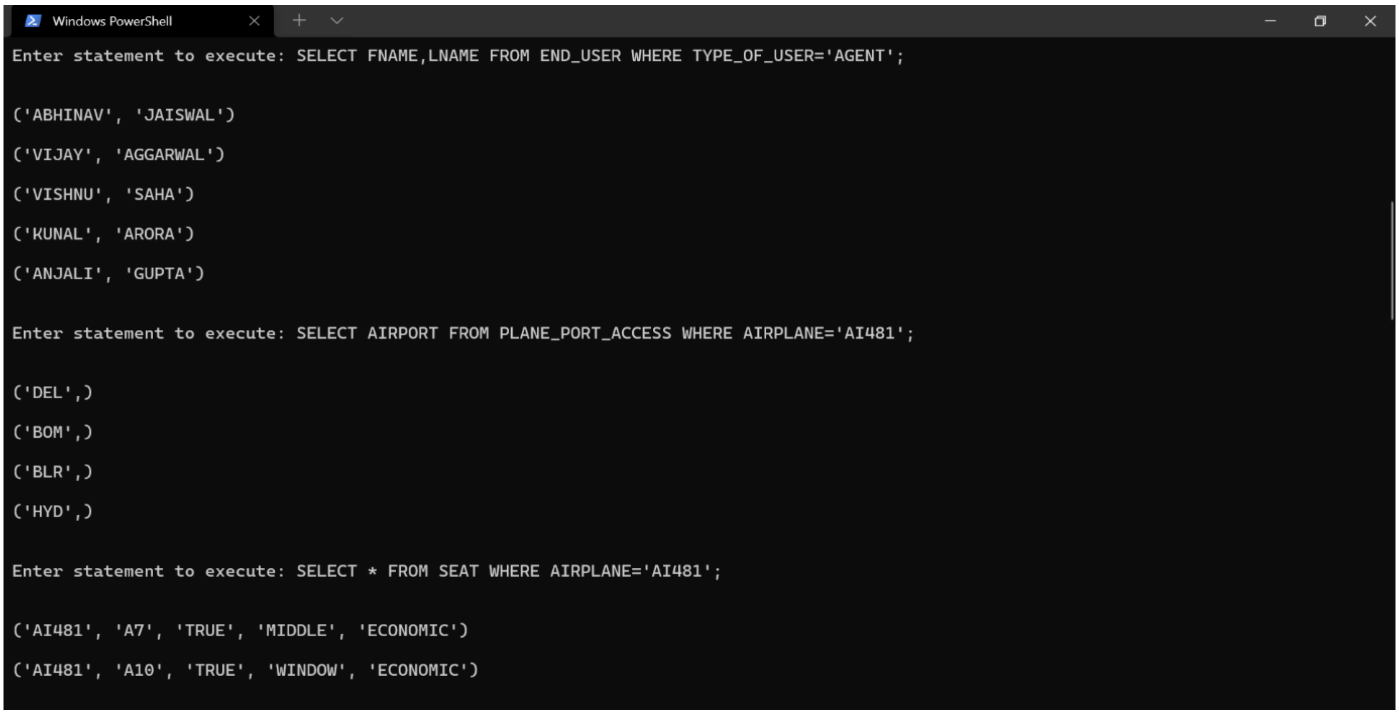
SCREENSHOTS OF STATEMENT EXECUTED IN THE FRONT END



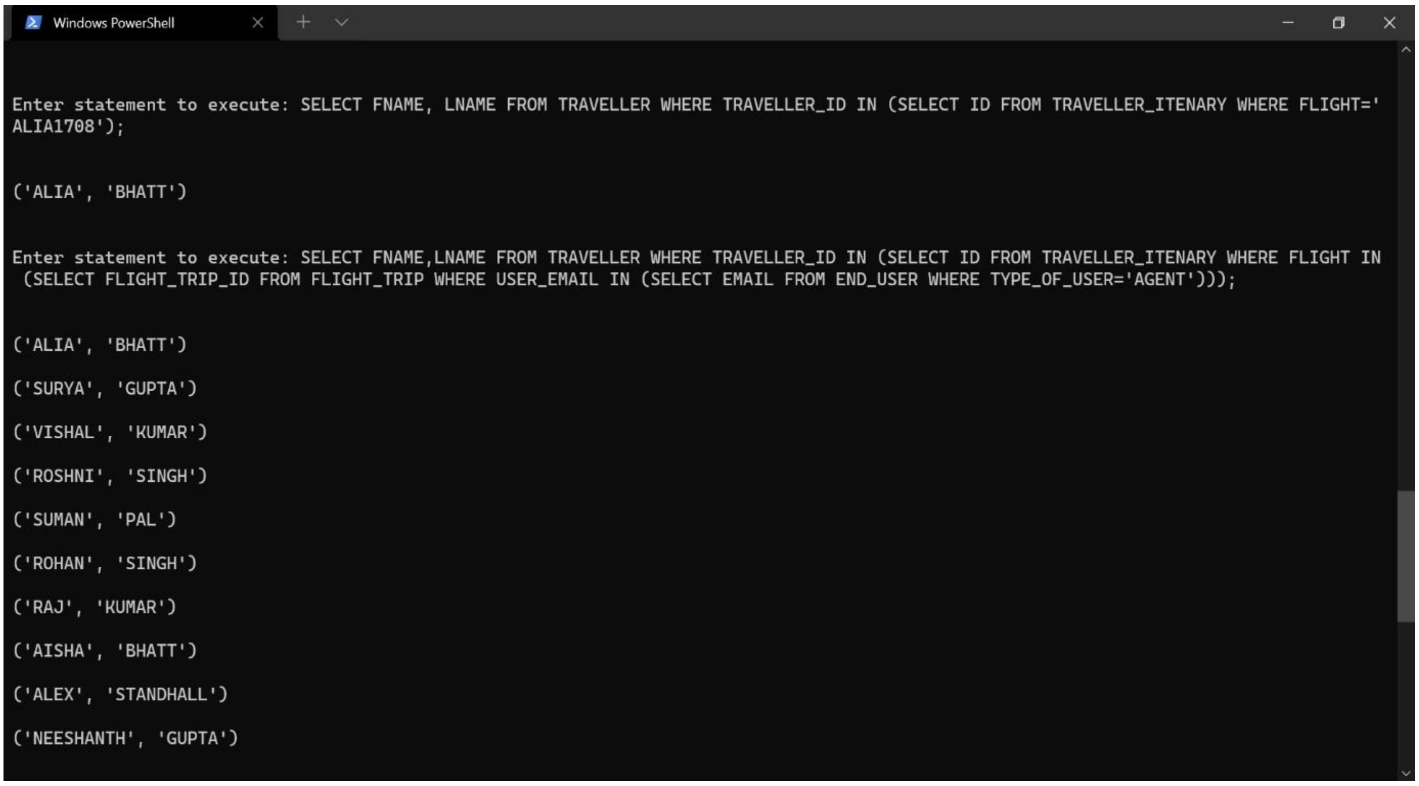
```
Enter statement to execute: SELECT * FROM SEAT WHERE AIRPLANE='AI481';

('AI481', 'A7', 'TRUE', 'MIDDLE', 'ECONOMIC')

('AI481', 'A10', 'TRUE', 'WINDOW', 'ECONOMIC')

('AI481', 'A13', 'TRUE', 'AISLE', 'ECONOMIC')

('AI481', 'A22', 'TRUE', 'MIDDLE', 'ECONOMIC')

('AI481', 'A36', 'TRUE', 'WINDOW', 'ECONOMIC')
```



```
('NEESHANTH', 'GUPTA')


Enter statement to execute: SELECT * FROM FARE WHERE FLIGHT_TRIP_ID IN (SELECT FLIGHT FROM TRAVELLER_ITENARY WHERE ID IN (SELECT TRAV
ELLER_ID FROM TRAVELLER WHERE FNAME='ALIA'));


('ALIA1708', Decimal('5019.18'), Decimal('5121.45'), 'Rupee', Decimal('512.14'), Decimal('409.87'))


Enter statement to execute: SELECT USER_EMAIL, COUNT (NO_OF_TRAVELLERS) FROM FLIGHT_TRIP WHERE USER_EMAIL IN (SELECT EMAIL FROM END_U
SER WHERE TYPE_OF_USER ='AGENT') GROUP BY USER_EMAIL;


('ABHI@GMAIL.COM', 2)

('ANJALI@GMAIL.COM', 2)

('KUNAL@GMAIL.COM', 2)

('VIJAY@GMAIL.COM', 2)

('VISHNU@GMAIL.COM', 2)


Enter statement to execute: SELECT * FROM END_USER WHERE EMAIL IN (SELECT USER_EMAIL FROM FLIGHT_TRIP WHERE FLIGHT_TRIP_ID IN (SELECT
 FLIGHT_TRIP_ID FROM FARE WHERE FLIGHT_TRIP_ID IN (SELECT  FLIGHT_TRIP FROM HAS WHERE HOP='HOP7463')));


('VIJAY@GMAIL.COM', 'VIJAY', 'AGGARWAL', Decimal('6734567288'), 'AGENT')
```



```
Enter statement to execute: SELECT FNAME,LNAME FROM END_USER WHERE TYPE_OF_USER='AGENT';

('ABHINAV', 'JAISWAL')
('VIJAY', 'AGGARWAL')
('VISHNU', 'SAHA')
('KUNAL', 'ARORA')
('ANJALI', 'GUPTA')

Enter statement to execute: SELECT AIRPORT FROM PLANE_PORT_ACCESS WHERE AIRPLANE='AI481';

('DEL',)
('BOM',)
('BLR',)
('HYD',)

Enter statement to execute: SELECT * FROM SEAT WHERE AIRPLANE='AI481';

('AI481', 'A7', 'TRUE', 'MIDDLE', 'ECONOMIC')
('AI481', 'A10', 'TRUE', 'WINDOW', 'ECONOMIC')
```



```
Enter statement to execute: SELECT FNAME, LNAME FROM TRAVELLER WHERE TRAVELLER_ID IN (SELECT ID FROM TRAVELLER_ITENARY WHERE FLIGHT='
ALIA1708');

('ALIA', 'BHATT')

Enter statement to execute: SELECT FNAME,LNAME FROM TRAVELLER WHERE TRAVELLER_ID IN (SELECT ID FROM TRAVELLER_ITENARY WHERE FLIGHT IN
 (SELECT FLIGHT_TRIP_ID FROM FLIGHT_TRIP WHERE USER_EMAIL IN (SELECT EMAIL FROM END_USER WHERE TYPE_OF_USER='AGENT')));

('ALIA', 'BHATT')
('SURYA', 'GUPTA')
('VISHAL', 'KUMAR')
('ROSHNI', 'SINGH')
('SUMAN', 'PAL')
('ROHAN', 'SINGH')
('RAJ', 'KUMAR')
('AISHA', 'BHATT')
('ALEX', 'STANDHALL')
('NEESHANTH', 'GUPTA')
```

Database Migration

Our airline database can have large amount of data that needs to be processed (data can sometimes contain an unstructured data). To ensure the scalability and flexibility we may choose to migrate your data from an older, monolithic relational database such as PostgreSQL, to modern, general-purpose database such as MongoDB

Advantages of MongoDB over PostgreSQL:

- MongoDB will shine because developers can reshape the data on their own when they need to. MongoDB enables you to manage data of any structure, not just tabular structures defined in advance.
- For supporting an application that will have to be scaled in terms of volume of traffic or size of data (or both) and that needs to be distributed across regions for data locality or data sovereignty, MongoDB's scale-out architecture will meet those needs automatically.

Using JSON for data migration is preferable if your PostgreSQL schema is complex and you want to nest arrays of related records inside of a MongoDB document.

To return the results of a PostgreSQL query as JSON, you will need three functions:

1. **row_to_json**: Returns a row as a JSON object with column names as keys and the values from the row
2. **array_to_json**: Returns an array of data as a JSON array
3. **array_agg**: Accepts a set of values and returns an array where each value in the set becomes an element in the array

TOPIC	CONTRIBUTION BY	TIME TAKEN
Front end connectivity	Kuntal Gorai	1 hour
Database Migration	Kuntal Gorai	20 minutes