**EXPERIMENT-2**

**AIM**

To design and deploy a simple **Flight Reservation System** as a **cloud-based software application** using a Cloud Service Provider (CSP), and to demonstrate **Software as a Service (SaaS)**, where users can access the application through the internet.

**PROCEDURE**

## **Step 1: Create App**

1. Login to Zoho Creator.
2. Click **Create New Application → From Scratch**.
3. Name it: **Flight Reservation System**.

## **Step 2: Create Forms (Database Tables)**

### (a) Flights Form

Fields:

1. Flight ID (Auto Number → Prefix: FL, Start: 1001 → e.g., FL1001)
2. Flight Name (Single Line)
3. Source (Dropdown → e.g., Chennai, Delhi, Mumbai, Bangalore)
4. Destination (Dropdown)
5. Departure Date/Time (Date-Time)
6. Total Seats (Number)
7. Available Seats (Number → default = Total Seats)
8. Status (Dropdown → Active / Cancelled)

### (b) Reservations Form

Fields:

1. Reservation ID (Auto Number → Prefix: R, Start: 5001)
2. Passenger Name (Single Line)
3. Email (Email)
4. Phone (Phone)
5. Flight (Lookup → Link to **Flights Form**)
6. Seats Booked (Number)
7. Booking Status (Dropdown → Confirmed / Waitlisted / Cancelled)

## **Step 3: Add Business Logic (Deluge Script)**

**Update Available Seats when booking is made**

1. Go to **Reservations Form → Workflows → On Success**.
2. Add **Deluge Script**:

flightRec = Flights[ID == input.Flight]; // Fetch selected flight

if(flightRec.Available\_Seats >= input.Seats\_Booked)

{

flightRec.Available\_Seats = flightRec.Available\_Seats - input.Seats\_Booked;

input.Booking\_Status = "Confirmed";

}

else

{

input.Booking\_Status = "Waitlisted";

}

## **Step 4: Add Cancellation Logic**

When a reservation is cancelled, restore the seats.

1. Go to **Reservations Form → Workflows → On Update → If Booking\_Status = Cancelled**
2. Add Script:

if(input.Booking\_Status == "Cancelled")

{

flightRec = Flights[ID == input.Flight];

flightRec.Available\_Seats = flightRec.Available\_Seats + input.Seats\_Booked;

}

## **Step 5: Create Reports (UI Views)**

**Available Flights** → Show flights where Status = Active & Available Seats > 0.

**My Reservations** → Shows bookings made by the logged-in user.

**Admin Dashboard** → All flights + reservations.

## **Step 6: Manage Users (SaaS Feature)**

Go to **Settings → Users & Permissions** → Create roles:

**Admin** → Add flights, view all bookings.

**Passenger** → Only book flights, view own bookings.

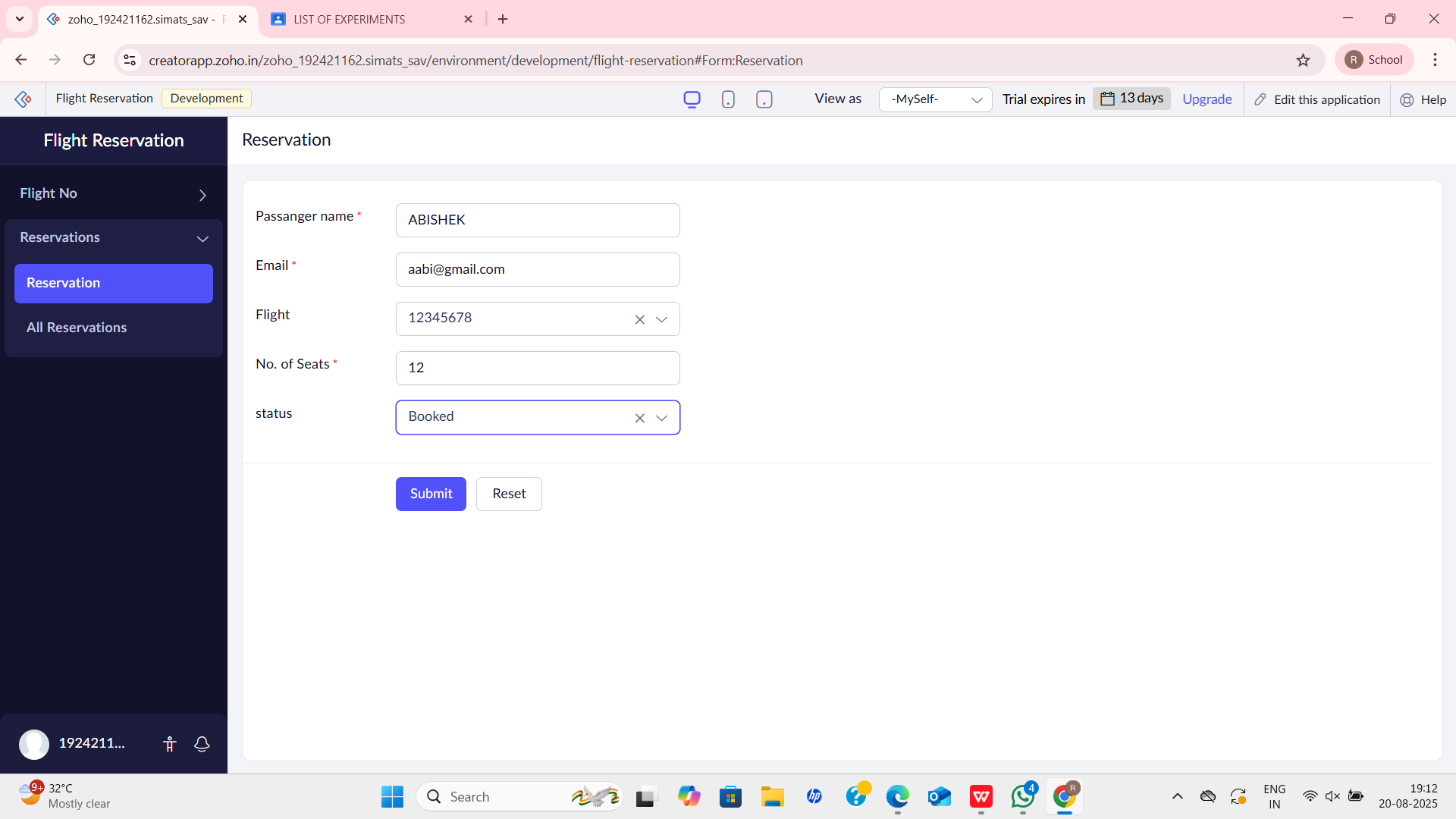
Assign users to roles when inviting them.

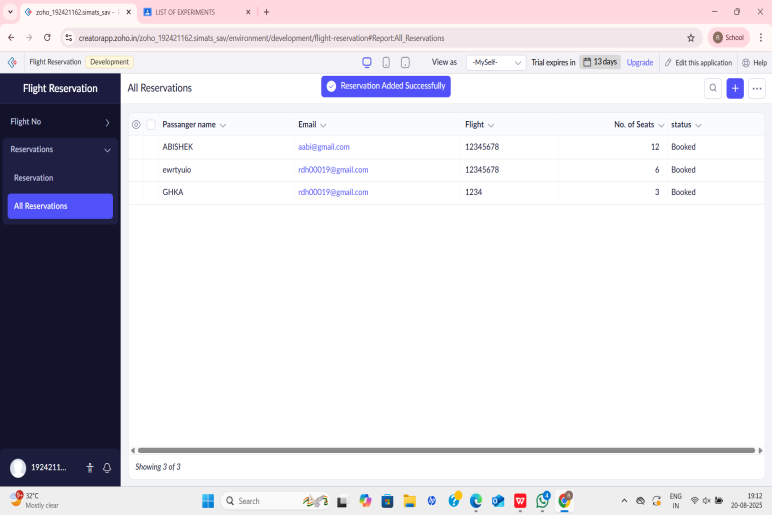
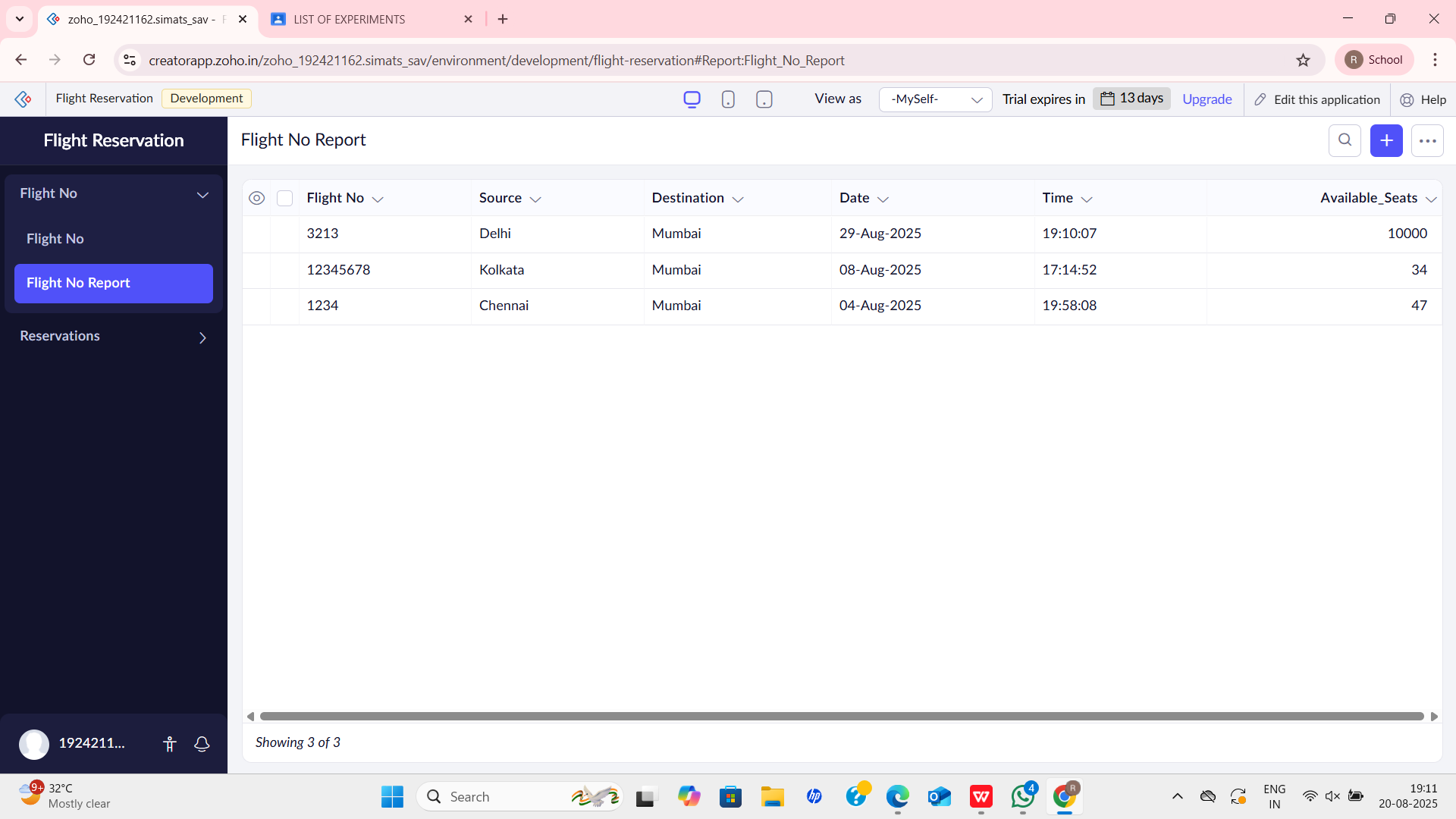
## **Step 7: Publish the App (SaaS Demo)**

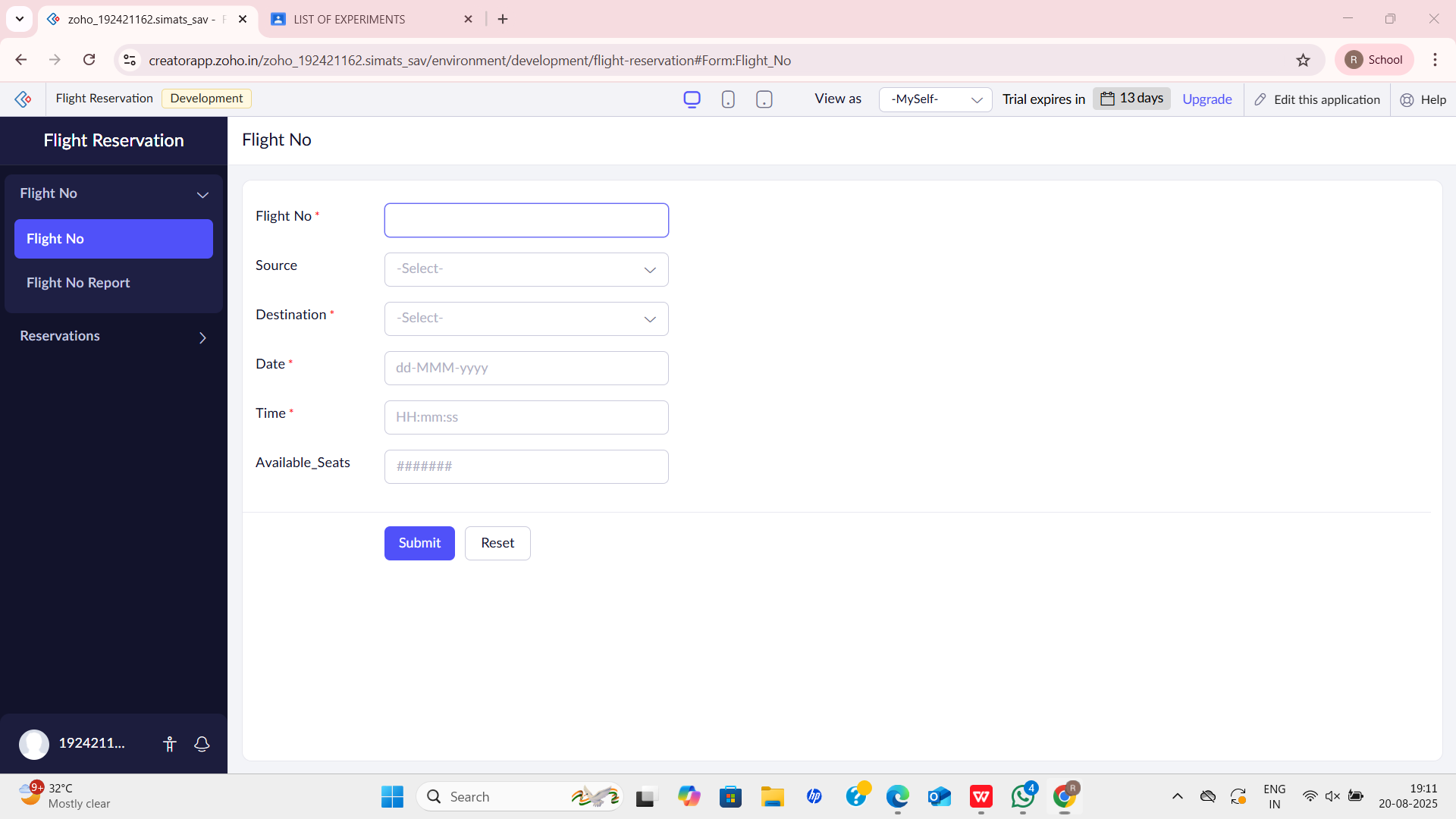
Go to **Settings → Publish → Enable Access**.

Share link with users.

Passengers can book, Admin can manage → SaaS style

**OUTPUT**





**RESULT**

A Flight Reservation System was successfully created and deployed on the cloud. The application can be accessed by multiple users via a web browser, without installation, thus proving the concept of Software as a Service (SaaS).