

# Smart Transit Management System

## Using Service Now

### Project Description:

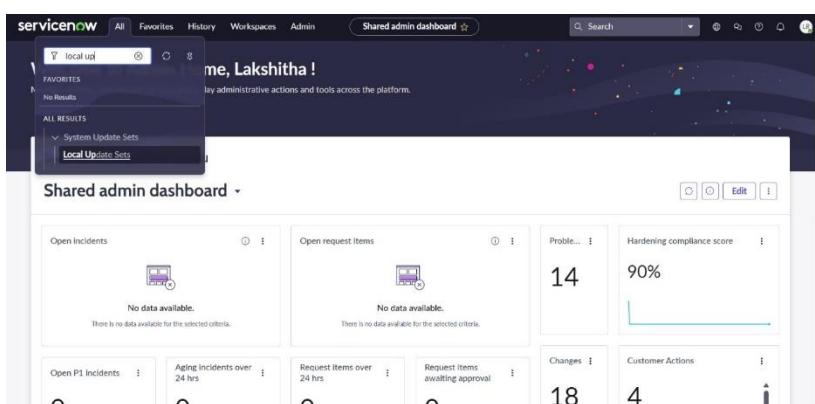
The Smart Transit Management System is built on the ServiceNow platform to streamline and automate bus service operations. It enables passengers to book and manage tickets, drivers to access schedules, and administrators to oversee buses, routes, and reports. The system leverages ServiceNow's catalog items, workflows, roles, and reporting features to create a smooth end-to-end solution.

### Skills:

- **ServiceNow:** Catalog Items, Record Producers, Tables, Workflows, ACLs, Notifications, Reporting.
- **Technical:** JavaScript, Glide API, Database Design.
- **Project Skills:** Problem-solving, Requirement Analysis, Documentation, Team Collaboration.

### Mile stone 1: Creating an Update Set

1. Click on All >> Local Update set



2. Click New

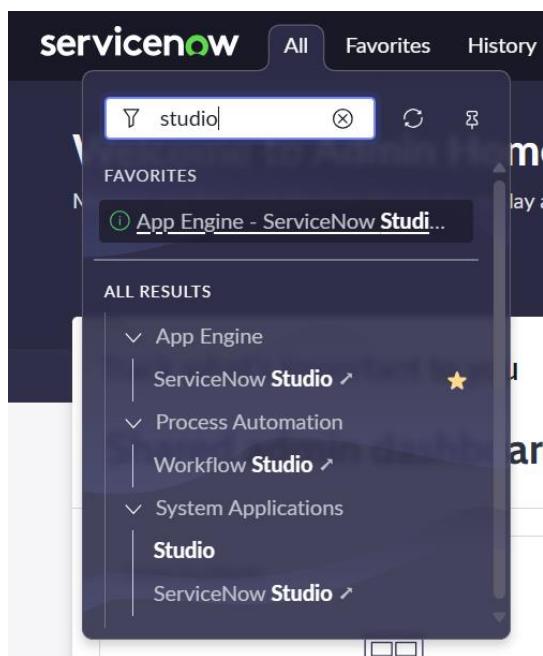
Name	Application	State	Installed from	Created	Created by	Parent	Batch Base
Search	Search	Search	Search	Search	Search	Search	Search
ADF-Y-syntax editor macros	Global	Complete		2025-08-18 18:09:57	system	(empty)	(empty)
App Store Registration 2.0	Global	Complete		2025-08-18 18:22:05	system	(empty)	(empty)
CB_Book a Ticket_2025-09-05 07:05:41	Smart Transit Management System	Complete		2025-09-05 07:05:41	Gangothri Madduru	(empty)	(empty)

3. Enter the Details Name: Smart Transit Management System >> Click on submit and make current.

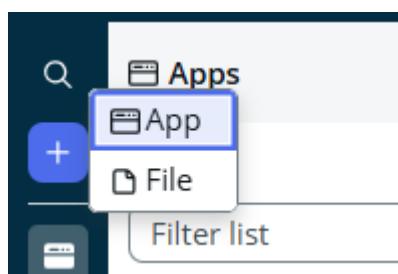
The screenshot shows the 'Update Set - Create New Update Set' page in ServiceNow. The 'Name' field is populated with 'Smart Transit Management System'. Other fields like 'State' (In progress), 'Parent', 'Release date', and 'Description' are present. At the bottom, there are 'Submit' and 'Submit and Make Current' buttons.

## Mile stone 2: Creating Application

- 1.All >> App Engine >> ServiceNow Studio



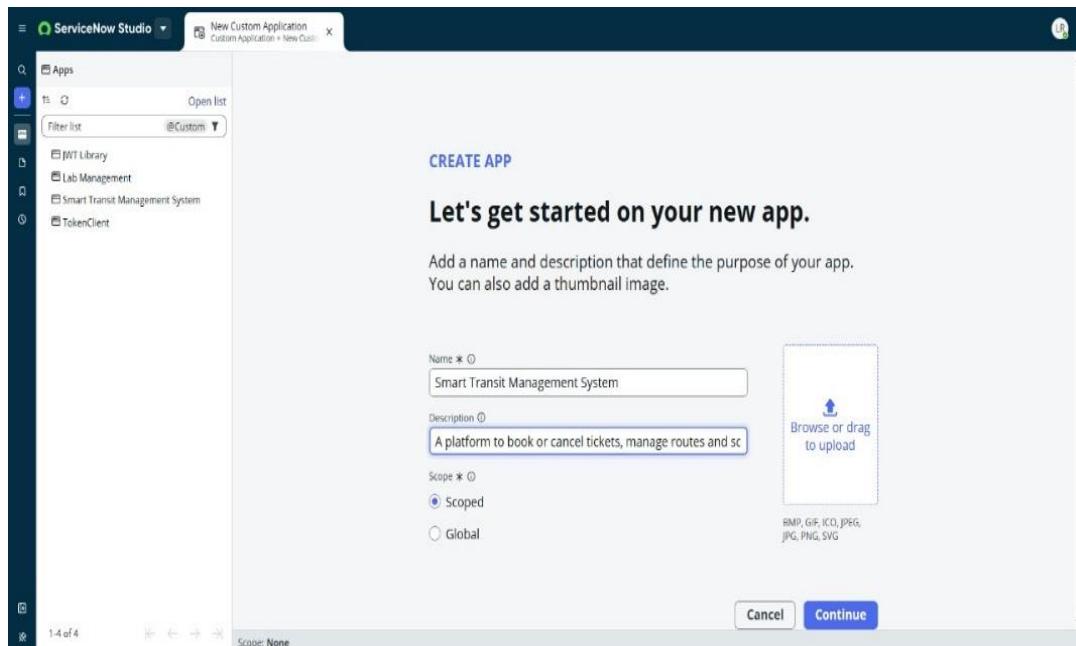
- 2.Select the Create icon [+] and then select App.



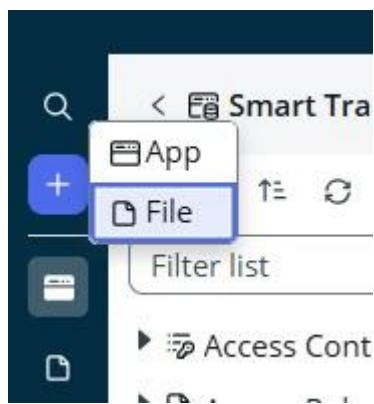
## Name: Smart Transit Management System

**Description:** A platform to book or cancel tickets, manage routes and schedules, and handle passenger complaints efficiently for smooth transit operations.

### 3. Select continue



### 4. Select the Create icon [+] and then select File.

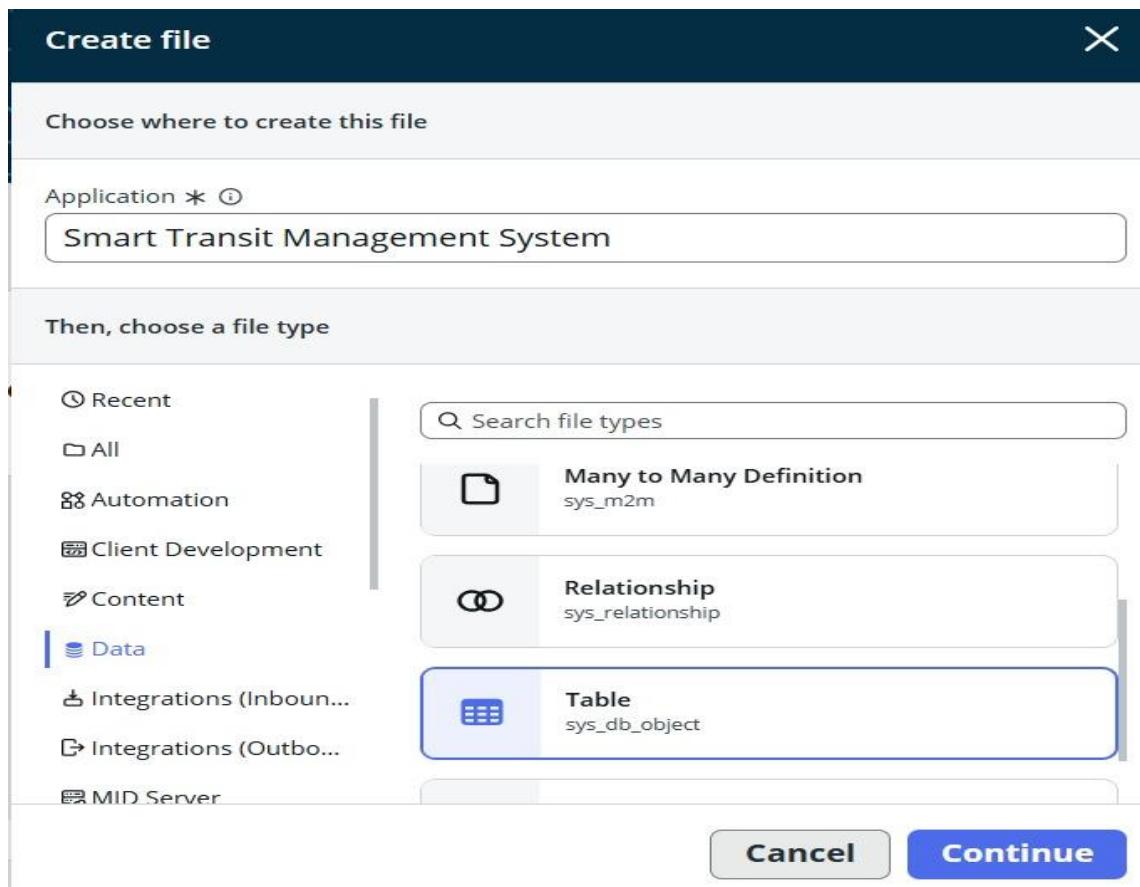


### 5. Specify the Application the file will belong to: *Smart Transit Management System*.

### 6. Select Data on the left-side panel.

7.Scroll down and select Table as the file type.

8.Select Continue.



### Mile stone 3: Creating Tables

#### Creation of various tables in ServiceNow

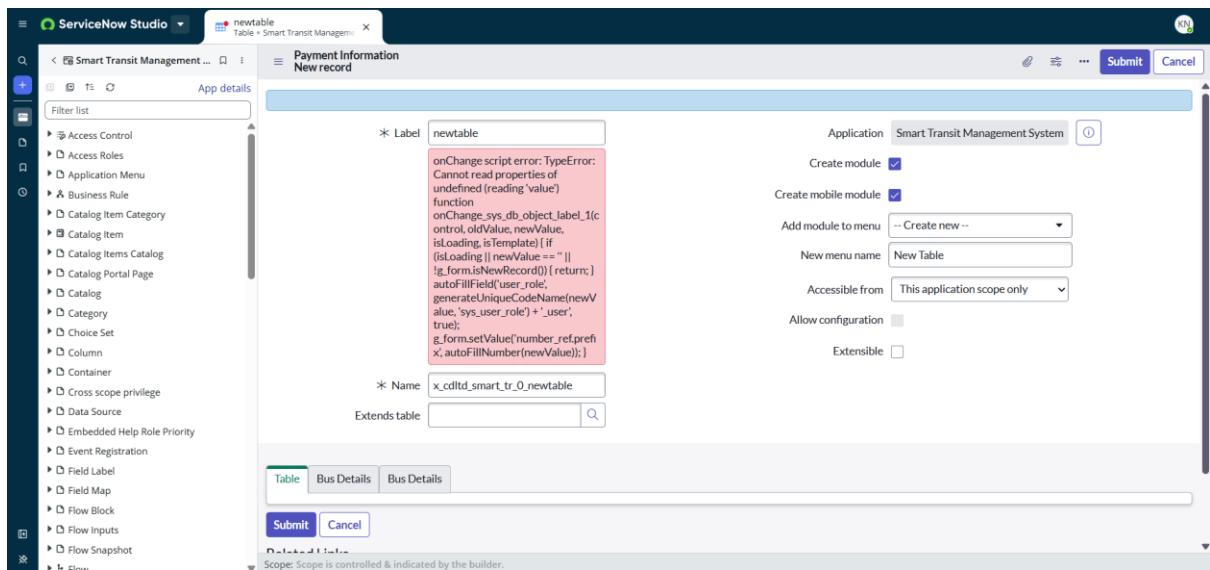
**Step 1: Open ServiceNow Studio (All -> System Applications -> ServiceNow Studio)**

The screenshot shows the ServiceNow Admin Home interface. On the left, there's a sidebar with 'ServiceNow Studio' selected under 'App Engine'. The main area displays the 'Shared admin dashboard' card with the title 'Welcome to Admin Home, Kothakonda !'. The card includes a message: 'Manage, monitor, and discover all your day to day administrative actions and tools across the platform.' Below the message, there's a section titled 'Track what's important to you' with the heading 'Shared admin dashboard'. This section contains several cards: 'Open Incidents' (No data available), 'Open request Items' (No data available), 'Hardening compliance...' (90%), 'Open P1 incidents' (1...), 'Aging incidents over 24 hrs' (2...), 'Request items over 24 hrs' (3...), 'Request items awaiting approval' (4...), and 'Customer Actions'.

**Step 2: Click on Create (File) and Select your application scope and table(sys\_db\_object) and click on create.**

The screenshot shows the ServiceNow Studio interface. On the left, there's a sidebar with various options like Access Control, Catalog Item Category, and Catalog Item. The main area shows a 'Welcome, Kothakonda' message and recently opened files and apps. A 'Create file' dialog box is open on the right. It has fields for 'Choose where to create this file' (Application: Smart Transit Management System) and 'Then, choose a file type' (Table: sys\_db\_object). There are also sections for Recent, All, Automation, Client Development, Content, Data, Integrations (Inbound), and Integrations (Outbound). At the bottom, there are 'Cancel' and 'Continue' buttons.

**Step 3: Enter the details of the table that you want to create and click on Submit**



#### Step 4: Add the fields in the table by double clicking on the table fields and click on Save

Column label *	Column name *	Type *	Reference	Max length	Default value	Display	
Booking ID	booking_id	String		40		<input type="checkbox"/>	20
Bus ID	bus_id	Reference	Bus			<input type="checkbox"/>	20
Created	sys_created_on	Date/Time				<input type="checkbox"/>	20
Created by	sys_created_by	String		40		<input type="checkbox"/>	20
Date	date	Date				<input type="checkbox"/>	20
Passenger Name	passenger_number	Reference	Passenger			<input type="checkbox"/>	20
Seat Number	seat_number	Integer				<input type="checkbox"/>	20
Status	status	Choice (3 Choices)				<input type="checkbox"/>	20
Time	time	Time				<input type="checkbox"/>	20

For this project, we have created multiple tables. The following gives you the details of each table

#### Table 1: Active Bookings

This table displays the data of passengers booking status as Pending, Booked or Cancelled.

**Bookings**

Booking ID	Bus ID	Passenger Name	Date	Time	Status	Seat Number
B003	BUS009	Priya Mehta	2025-09-08	10:00:00	Pending	5
B007	BUS009	Mounika	2025-09-07	11:30:00	Booked	7
B006	BUS008	Sneha	2025-09-09	08:50:00	Pending	6
B008	BUS006	Ganesh	2025-09-08	11:40:00	Booked	8
B011	BUS005	Navadeep	2025-09-24	15:00:00	Booked	18
B010	BUS005	Divya Kapoor	2025-09-09	07:20:00	Booked	6
B001	BUS003	Rohan Sharma	2025-09-06	09:00:00	Booked	1
B005	BUS003	Nikhil	2025-09-08	08:45:00	Cancelled	4
B004	BUS002	Arjun Reddy	2025-09-08	10:10:00	Cancelled	6
B002	BUS002	Navadeep	2025-09-07	00:05:00	Booked	3
B009	BUS001	Gangothri	2025-09-08	07:15:00	Pending	2

**Table 2: Active Buses**

This table displays the buses that are available for the service to passengers

**Bus**

Bus ID	Bus Name	Capacity	Driver ID	Route ID	Status
BUS009	V04	110	DRV0010	RTE0000029	Active
BUS004	XYZ	120	DRV0011	RTE0000033	Active
BUS005	ST	110	DRV0012	RTE0000027	Active
BUS008	F1g	120	DRV0008	RTE0000028	Active
BUS010	HFW	85	DRV0011	RTE0000033	Active
BUS001	AVM	100	DRV0006	RTE0000030	Active

**Table 3: GPS Tracking**

This table displays the GPS tracking of the buses with their BusID referenced to the Bus table

**GPS Trackings**

Bus ID	Latitude	Longitude	Speed	Status	Timestamp	Tracking ID
BUS001	17.37	78.48	50	In Transit	2025-09-04 08:00:00	GPS0001
BUS001	17.38	78.49	48	In Transit	2025-09-04 08:05:37	GPS0002
BUS001	17.39	78.5	55	In Transit	2025-09-04 08:10:54	GPS0003
BUS001	17.4	78.51	45	In Transit	2025-09-04 08:15:06	GPS0004
BUS004	8.52	76.88	45	In Transit	2025-09-04 09:00:17	GPS0005
BUS004	8.53	76.89	48	In Transit	2025-09-04 09:10:22	GPS0006
BUS004	8.54	76.9	47	In Transit	2025-09-04 09:15:37	GPS0007
BUS004	8.55	77.01	50	In Transit	2025-09-04 09:20:00	GPS0008
BUS005	17.37	78.48	55	In Transit	2025-09-04 07:35:24	GPS0009
Preview GPS0011	17.38	78.49	53	In Transit	2025-09-04 07:40:34	GPS0010
BUS005	17.39	78.5	45	In Transit	2025-09-04 07:45:48	GPS0011
BUS005	17.4	78.51	48	In Transit	2025-09-04 07:52:59	GPS0012
BUS008	10.85	76.21	49	In Transit	2025-09-04 10:05:14	GPS0013
BUS008	10.86	76.22	50	In Transit	2025-09-04 10:10:23	GPS0014
BUS008	10.87	76.23	52	In Transit	2025-09-04 10:20:42	GPS0015
BUS003	12.97	77.58	0	Maintenance	2025-09-04 04:06:56	GPS0016
BUS002	12.97	78.88	0	Inactive	2025-09-04 04:06:08	GPS0017

**Table 4: Payments**

This Payments table deals with the payments status and mode of the payment made by the passenger

**Payments**

Amount	Booking Passenger	Passenger	Payment Date	Payment ID	Payment Method	Payment Status	Transac
\$89.2226	9edc5c5e837722107f443dfb0fda1e46	Nikhil	2025-09-24 13:18:54	PAYID01048	Cash	Pending	PAYIDC
\$13.9092	bb7c585e837722107f443dfb0fda1e57	Rohan Sharma	2025-09-24 07:26:33	PAYID01036	Cash	Completed	PAYIDC
\$5.0887	baacd85e837722107f443dfb0fda1e31	Nikhil	2025-09-02 14:28:10	PAYID01034	UPI	Pending	PAYIDC
\$1.357	bb7c585e837722107f443dfb0fda1e57	Gangothri	2025-09-25 07:25:29	PAYID01032	Net Banking	Refunded	PAYIDC
\$101.4356	977d989e837722107f443dfb0fda1ecb	Mounika	2025-10-03 07:24:18	PAYID01030	Card	Failed	PAYIDC
\$5.1566	a2db941e837722107f443dfb0fda1ed4	Divya Kapoor	2025-10-07 07:23:42	PAYID01028	Cash	Failed	PAYIDC
\$64.1182	865d989e837722107f443dfb0fda1ec2	Gangothri	2025-10-01 07:23:00	PAYID01026	UPI	Failed	PAYIDC
\$152.0969	8ffc509e837722107f443dfb0fda1e57	Ganesh	2025-09-27 07:22:31	PAYID01024	Net Banking	Pending	PAYIDC
\$89.2226	baacd85e837722107f443dfb0fda1e31	Sneha	2025-09-02 14:28:10	PAYID01022	Cash	Failed	PAYIDC
\$64.1182	bb7c585e837722107f443dfb0fda1e57	Nikhil	2025-09-30 07:21:05	PAYID01020	Card	Completed	PAYIDC

**Table 5: Prices**

This table shows the bus charges from one point to another point

The screenshot shows a ServiceNow application window titled "Prices". The left sidebar has a "Smart" search bar and a "Favorites" section. Under "ALL RESULTS", there is a category "Smart Transit Management" containing links like Active Bookings, Active Buses, Booking Summary Page, Bookings, Bus Assignment, Create New Bus, GPS Tracking, Payments, Prices, Route, Schedule, Stops, Tickets, Update Price List, and My Bookings. The main content area displays a table with the following data:

Distance	End Point	Price	Start Point
750	Hyderabad	1,500	Chennai
950	Kozhikode	1,900	Chennai
450	Hubballi	900	Coimbatore
700	Hyderabad	1,400	Coimbatore
500	Vijayawada	1,000	Madurai
300	Mangaluru	600	Trichy
150	Trichy	300	Salem
350	Chennai	700	Bengaluru
150	Mysuru	300	Bengaluru
300	Madurai	600	Mysuru
750	Thiruvananthapuram	1,500	Mysuru
550	Kochi	1,100	Mangaluru
350	Hyderabad	700	Hubballi
150	Vijayawada	300	Visakapatnam
300	Mysuru	600	Vijayawada
650	Chennai	1,300	Tirupati
400	Kozhikode	800	Tirupati

**Table 6: Route**

This table displays in detail stops from start point to end point.

The screenshot shows a ServiceNow application window titled "Route Tables". The left sidebar has a "Smart" search bar and a "Favorites" section. Under "ALL RESULTS", there is a category "Smart Transit Management" containing links like Active Bookings, Active Buses, Booking Summary Page, Bookings, Bus Assignment, Create New Bus, GPS Tracking, Payments, Prices, Route, Schedule, Stops, Tickets, Update Price List, and My Bookings. The main content area displays a table with the following data:

Route Id	Active	Distance	End Point	Start Point	Stops
RTE0000026	true	515.3	Bengaluru	Chennai	Chennai, Salem, Bengaluru
RTE0000027	true	679.8	Hyderabad	Chennai	Chennai, Tirupati, Kurnool, Hyderabad
RTE0000028	true	698.6	Kochi	Chennai	Chennai, Salem, Coimbatore, Thrissur, Ko
RTE0000029	true	758.3	Thiruvananthapuram	Bengaluru	Bengaluru, Salem, Coimbatore, Thrissur,
RTE0000030	true	748.8	Visakapatnam	Hyderabad	Hyderabad, Warangal, Vijayawada, Visakapa
RTE0000031	true	666.4	Mysuru	Chennai	Chennai, Salem, Bengaluru, Mysuru
RTE0000032	true	1029.7	Hyderabad	Coimbatore	Coimbatore, Salem, Tirupati, Kurnool, Hy
RTE0000033	true	641.0	Thiruvananthapuram	Mangaluru	Mangaluru, Kozhikode, Thrissur, Kochi, T
RTE0000034	true	1053.9	Bengaluru	Vijayawada	Vijayawada, Tirupati, Chennai, Salem, Be
RTE0000035	true	998.6	Kochi	Hubballi	Hubballi, Bengaluru, Salem, Coimbatore,
RTE0000036	false		Chennai	Chennai	Madurai, Hyderabad

**Table 7: Schedule**

This table deals with scheduling the bus with arrival time to the start point and depart time at the end point

Schedule ID	Bus	Driver Name	Arrival Time	Departure Time	Route	Status
1	BUS003	DRV0012	2025-09-08 07:18:21	2025-09-07 07:18:18	RTE0000026	
4	BUS004	DRV0011	2025-09-09 07:20:13	2025-09-07 07:20:10	RTE0000029	
6	BUS005	DRV0015	2025-09-11 07:21:27	2025-09-10 07:21:23	RTE0000031	
SCH00004	BUS003	DRV0008	2025-09-19 13:54:07	2025-09-12 13:54:03	RTE0000026	Rescheduled
5	BUS007	DRV0013	2025-09-16 07:20:47	2025-09-15 07:20:43	RTE0000030	
10	BUS006	DRV0010	2025-09-12 07:24:08	2025-09-11 07:24:05	RTE0000035	
3	BUS009	DRV0007	2025-09-10 07:19:38	2025-09-08 07:19:34	RTE0000028	
7	BUS008	DRV0008	2025-09-07 07:22:22	2025-09-07 07:22:19	RTE0000032	
9	BUS001	DRV0009	2025-09-07 07:23:33	2025-09-13 07:23:29	RTE0000034	
2	BUS002	DRV0014	2025-09-10 07:19:05	2025-09-09 07:19:01	RTE0000027	
8	BUS010	DRV0006	2025-09-11 07:22:55	2025-09-11 07:22:50	RTE0000033	
SCH00005	BUS003	DRV0006	2025-09-17 13:57:43	2025-09-26 13:57:28	RTE0000029	on Hold

**Table 8: Ticket**

This table displays the TicketID and activeness of the ticket as Booked, Completed or Cancelled

Number	Active	Passenger	Route	Status	Travel Date
TKT0000007	false	Rohan Sharma	RTE0000026	Completed	2025-09-05 20:30:00
TKT0000008	false	Navadeep	RTE0000029	Completed	2025-09-03 14:41:12
TKT0000009	false	Arjun Reddy	RTE0000035	Cancelled	2025-09-06 09:38:49
TKT0000010	false	Priya Mehta	RTE0000033	Completed	2025-09-04 17:00:00
TKT0000012	false	Gangothri	RTE0000029	Completed	2025-09-07 10:00:00
TKT0000013	true	Mounika	RTE0000030	Cancelled	2025-09-05 18:00:00
TKT0000014	false	Sneha	RTE0000032	Completed	2025-09-02 13:00:00
TKT0000015	true	Divya Kapoor	RTE0000034	Booked	2025-09-13 15:30:00
TKT0000016	true	Ganesh	RTE0000033	Booked	2025-09-15 17:00:00
TKT0000017	true	Ganesh	RTE0000029	Booked	2025-09-20 07:00:00
TKT0000019	true	Gangothri	RTE0000032	Booked	2025-09-15 17:45:55
TKT0000020	false	Divya Kapoor	RTE0000026	Completed	2025-09-08 17:58:00
TKT0000021	false	Mounika	RTE0000027	Completed	2025-09-08 18:10:00
TKT0000022	true	Rohan Sharma	RTE0000026	Booked	2025-09-18 11:56:38
TKT0000023	true	Mounika	RTE0000029	Booked	2025-09-19 13:24:12
TKT0000024	false	(empty)	(empty)	(empty)	(empty)
TKT0000025	true	Mounika	RTE0000029	Booked	2025-09-19 13:24:12

**Table 9: Feedback**

This table displays the feedback given by each passenger after completion of their journey in their scheduled bus.

The screenshot shows the ServiceNow interface for the 'Feedback\_News' table. The left sidebar contains navigation links for bus assignment, GPS tracking, payments, prices, route, schedule, stops, tickets, update price list, my bookings, all bookings, bus, create new booking, driver, feedback, passenger, price table, and price table new. The main area displays a list of feedback entries with columns: Feedback ID, Passenger Name, Bus, Feedback Date, Feedback Type, Rating, and Comments. The comments column contains truncated text such as 'Dirty bus', 'Excellent driving skills', and 'The Behaviour of driver is not upto the ...'. A search bar at the top right is set to 'Bus'.

Feedback ID	Passenger Name	Bus	Feedback Date	Feedback Type	Rating	Comments
FB0000037	Rohan Sharma	BUS010	2025-09-04 14:10:49	Cleanliness	4 -- Average	Dirty bus
FB0000033	Nikhil	BUS010	2025-09-04 14:08:20	Compliment	1 -- Extraordinary	Excellent driving skills
FB0000030	Nikhil	BUS010	2025-09-04 13:39:57	Driver Behaviour	3 -- Good	The Behaviour of driver is not upto the ...
FB0000043	Rohan Sharma	BUS009	2025-09-11 13:36:46	Complaint	3 -- Good	i lost my bag
FB0000034	Mounika	BUS009	2025-09-04 14:08:52	Compliment	3 -- Good	Ok but need to work more
FB0000040	Gangothri	BUS008	2025-09-07 07:00:35	Compliment	2 -- Very Good	Very good
FB0000041	Navadeep	BUS007	2025-09-07 16:19:18	Driver Behaviour	5 -- Worst	Dis-respectful driver
FB0000032	Rohan Sharma	BUS006	2025-09-04 13:45:23	Compliment	2 -- Very Good	Excellent skills of driver. Very smooth...
FB0000038	Gangothri	BUS005	2025-09-04 14:11:21	Lost Item	4 -- Average	I lost my bag
FB0000035	Divya Kapoor	BUS005	2025-09-04 14:09:23	Driver Behaviour	3 -- Good	Very rude and irresponsible
FB0000027	Ganesh	BUS004	2025-09-04 13:21:24	Lost Item	4 -- Average	I've lost my airpods in the bus. Can't find...

**Table 10: Passenger**

This table displays the details of the passenger

The screenshot shows the ServiceNow interface for the 'Passengers' table. The left sidebar contains the same navigation links as the previous screenshot. The main area displays a list of passenger entries with columns: Number, Active, Date of Birth, Email, Full Name, Gender, Notes, and Phone. The notes column contains truncated text such as 'Passenger Rohan Sharma Created on 2025-09-0...' and 'Passenger Navdeep Kothakonda Created on 20...'. A search bar at the top right is set to 'Number'.

Number	Active	Date of Birth	Email	Full Name	Gender	Notes	Phone
PASS1001	true	1998-04-15	rohan.sharma@gmail.com	Rohan Sharma	Male	Passenger Rohan Sharma Created on 2025-09-0...	+91 95782 36771
PASS1002	true	2004-08-14	naveedee61@gmail.com	Navadeep	Male	Passenger Navdeep Kothakonda Created on 20...	+91 89197 66746
PASS1003	true	2000-09-28	mailto:priya.mehta@gmail.com	Priya Mehta	Female		+91 63013 60645
PASS1004	true	1997-11-10	arjun.reddy@gmail.com	Arjun Reddy	Male	Passenger Arjun Reddy Created on 2025-09-04...	+91 95501 60654
PASS1005	true	1996-12-03	mailto:nikhil.verma@gmail.com	Nikhil	Male		+91 80086 78713
PASS1006	true	2001-03-18	sneha.iyer@gmail.com	Sneha	Female	Passenger Sneha Created on 2025-09-04 14:18:12	+91 81060 62232
PASS1007	true	2005-05-18	mailto:mounikajaya12@gmail.com	Mounika	Female	Passenger Mounika Created on 2025-09-04 14:...	+91 93810 33762
PASS1008	true	2004-07-26	ganesh16@gmail.com	Ganesh	Male	Passenger Ganesh Created on 2025-09-04 14:2...	+91 98480 16664
PASS1009	true	2004-01-04	gangothri41@gmail.com	Gangothri	Female	Passenger Gangothri Created on 2025-09-04 1...	+91 89197 54024
PASS1010	true	2000-01-29	mailto:divya.kapoor@gmail.com	Divya Kapoor	Female		+91 76654 04562
PASS1014	true	2025-09-09	mounikajayaveerapalli91@gmail.com	V.Mounika	Female	Passenger V.Mounika Jaya Created on 2025-09-09...	+91 98765 47000

**Table 11: Drivers**

This table deals with the contact details of the bus driver

Driver ID	Name	Assigned Bus	Contact
DRV0006	Ramesh Kumar	BUS003	+91 98765 43210
DRV0007	David Johnson	BUS004	+1 (202) 555-0147
DRV0008	Oliver Smith	BUS002	+44 (0)7700 900123
DRV0009	Sunil Nair	BUS001	+91 98765 43216
DRV0010	Vikram Yadav	BUS010	+91 98765 43217
DRV0011	Fatima Zahra	BUS004	+212 612-345678
DRV0012	Ahmed Hassan	BUS006	+20 111 234 5678
DRV0013	Luca Rossi	BUS001	+39 345 678 9012
DRV0014	Arun Verma	BUS008	+7 916 123-45-67
DRV0015	Oliver Smith	BUS009	+44 (0)7700 900123

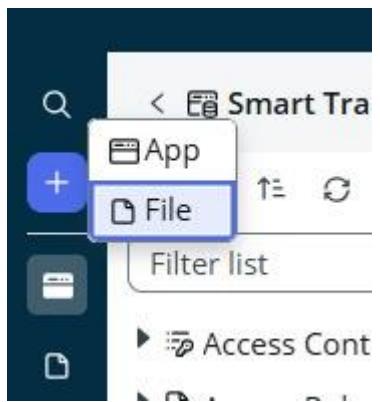
**Table 12: Bookings**

This table deals with booking of bus for a particular bus

Driver ID	Name	Assigned Bus	Contact
DRV0006	Ramesh Kumar	BUS003	+91 98765 43210
DRV0007	David Johnson	BUS004	+1 (202) 555-0147
DRV0008	Oliver Smith	BUS002	+44 (0)7700 900123
DRV0009	Sunil Nair	BUS001	+91 98765 43216
DRV0010	Vikram Yadav	BUS010	+91 98765 43217
DRV0011	Fatima Zahra	BUS004	+212 612-345678
DRV0012	Ahmed Hassan	BUS006	+20 111 234 5678
DRV0013	Luca Rossi	BUS001	+39 345 678 9012
DRV0014	Arun Verma	BUS008	+7 916 123-45-67
DRV0015	Oliver Smith	BUS009	+44 (0)7700 900123

## Mile stone 4: Creating Forms

1. Select the Create icon [+] and then select File.



2.Specify the Application the file will belong to: *Smart Transit Management System*.

3.Select Data on the left-side panel.

4.Scroll down and select Forms as the file type.

5.Select Continue.

**Create file**

Choose where to create this file

Application \* ⓘ  
Smart Transit Management System

Then, choose a file type

Recent  
All  
Automation  
Client Development  
Content  
**Data**  
Integrations (Inbound...)  
Integrations (Outbound...)

Search file types

**Form**  
sys\_ui\_form

Form Section  
sys\_ui\_section

Many to Many Definition  
sys\_m2m

**Cancel** **Continue**

## Select table name: Route Table

### 6. Select Open

The screenshot shows the ServiceNow Studio interface with a search bar at the top containing 'Route Table'. Below it is a button labeled 'Open'. A table titled 'Default view' lists various form views and their details:

View	Table	Scope	Updated
Default view	Tickets	Smart Transit Manage...	2025-09-08 17:18:30
Default view	Price	Smart Transit Manage...	2025-09-06 08:26:04
Default view	Stops	Smart Transit Manage...	2025-09-05 14:08:18
Default view	PAR Dashboard Permission	Global	2025-09-04 12:29:03
Default view	Route Table	Smart Transit Manage...	2025-09-02 12:48:54
Default view	Bus Assignment	Smart Transit Manage...	2025-08-30 02:48:25
Default view	Table	Global	2025-08-28 15:15:39

To the right, there is a sidebar titled 'What is Form Builder?' with sections on creating and customizing form views, configuring field properties, and setting up form logic using UI Policies.

### 7. Add the field required fields in the form.

### 8. Select save.

The screenshot shows the ServiceNow Studio interface for the 'Route Table - Default view'. The left sidebar lists available fields: Created (Date/Time), Created by (String), End Point (Reference: Stops), Start Point (Reference: Stops), Updated (Date/Time), and Updated by (String). The main area displays the 'Route Table' form with fields: 'Route Id \*' (text input), 'Start Point' (dropdown with value 'Stops [x\_cdlt\_d\_smart\_tr\_0\_stops]'), 'End Point' (dropdown with value 'Stops [x\_cdlt\_d\_smart\_tr\_0\_stops]'), 'Stops' (text input with placeholder 'Stops 1 X, Stops 2 X, Stops 3 X'), 'Distance' (text input), and 'Active' (checkbox checked). To the right, a 'Section' panel is open, showing a section labeled 'Route Table' with a 'Two columns' layout selected. A note at the bottom right explains merging sections.

## Similarly, create form for other tables:

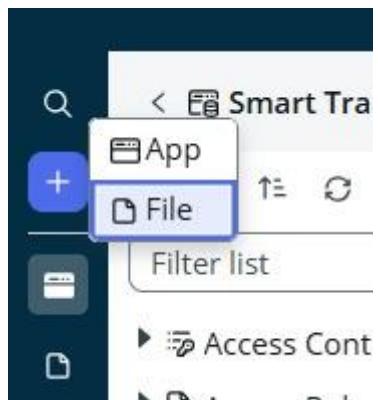
1. Bookings[x\_cdlt\_d\_smart\_tr\_0\_bookings]

2. Bus[x\_cdlt\_d\_smart\_tr\_0\_bus]

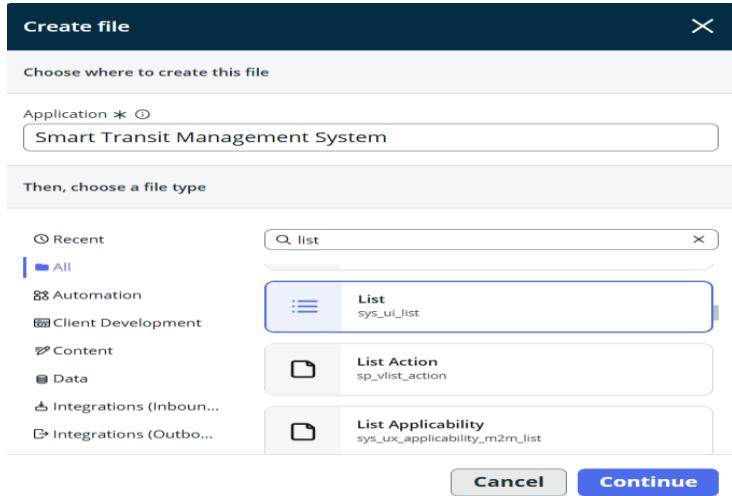
- 3.Driver[x\_cdlt\_d\_smart\_tr\_0\_driver]
- 4.Feedback\_new[x\_cdlt\_d\_smart\_tr\_0\_feedback\_new]
- 5.GPS Tracking[x\_cdlt\_d\_smart\_tr\_0\_gps\_tracking]
- 6.Passenger[x\_cdlt\_d\_smart\_tr\_0\_passenger]
- 7.Payment[x\_cdlt\_d\_smart\_tr\_0\_payment]
- 8.Price[x\_cdlt\_d\_smart\_tr\_0\_price]

## Mile stone 5: Creating Lists

- 1.Select the Create icon [+] and then select File.



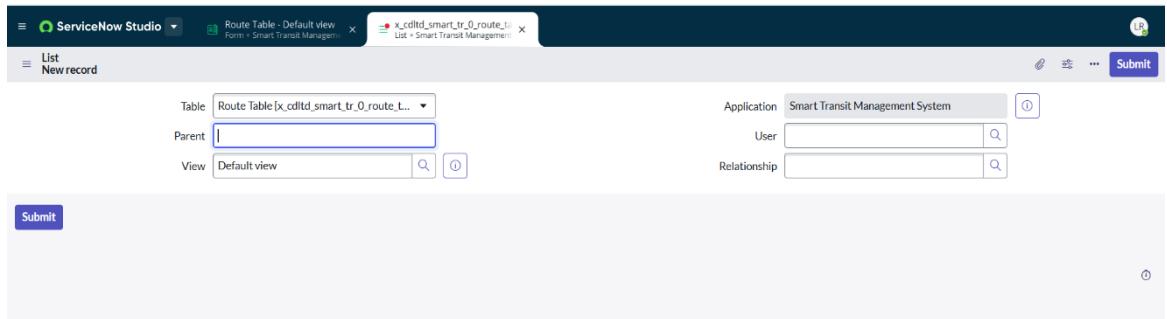
- 2.Specify the Application the file will belong to: *Smart Transit Management System*.
- 3.Select Data on the left-side panel.
- 4.Scroll down and select List as the file type.
- 5.Select Continue.



**Table:** Route Table[x\_cdlt\_d\_smart\_tr\_0\_route\_table]

**Application:** Smart Transit Management System.

6. Select Submit.



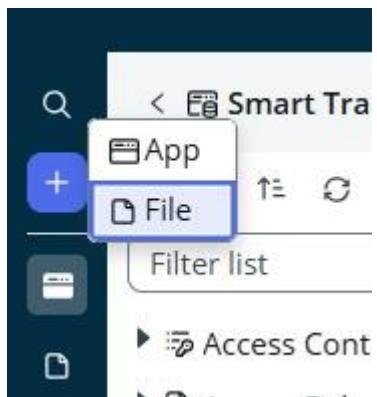
Similarly, create list for other tables:

1. Bookings[x\_cdlt\_d\_smart\_tr\_0\_bookings]
2. Bus[x\_cdlt\_d\_smart\_tr\_0\_bus]
3. Driver[x\_cdlt\_d\_smart\_tr\_0\_driver]
4. Feedback\_new[x\_cdlt\_d\_smart\_tr\_0\_feedback\_new]
5. GPS Tracking[x\_cdlt\_d\_smart\_tr\_0\_gps\_tracking]
6. Passenger[x\_cdlt\_d\_smart\_tr\_0\_passenger]
7. Payment[x\_cdlt\_d\_smart\_tr\_0\_payment]
8. Price[x\_cdlt\_d\_smart\_tr\_0\_price]
9. Schedule\_table[x\_cdlt\_d\_smart\_tr\_0\_schedule\_table]

10. Stops[x\_cdlt\_d\_smart\_tr\_0\_stops]
11. Tickets[x\_cdlt\_d\_smart\_tr\_0\_tickets]

## Mile stone 6: Creating Flows

1. Select the Create icon [+] and then select File.



2. Specify the Application the file will belong to: *Smart Transit Management System*.

3. Select Automation on the left-side panel.

4. Scroll down and select Flow(sys\_hub\_flow) as the file type.

5. Select Continue.

**Create file**

Choose where to create this file

Application \* ⓘ  
Smart Transit Management System

Then, choose a file type

Recent All Automation Client Development Content Data Integrations (Inbound...) Integrations (Outbound...)

Search file types

**Flow**  
sys\_hub\_flow

**Notification**  
sysevent\_email\_action

**Playbook**  
sys\_pd\_process\_definition

**Cancel** **Continue**

## Activity 1:

**Flow Name:** Auto Complete Ticket Status.

**Application:** Smart Transit Management System.

**Description:** This flow runs daily to check booked tickets with past travel dates and automatically updates their status, ensuring ticket records stay accurate.

Select Build Flow.

ServiceNow Studio New Flow Flow + Smart Transit Management

Let's get the details for your flow

Flow name \* ⓘ Auto Complete Ticket Status

Application \* ⓘ Smart Transit Management System

Description ⓘ This flow runs daily to check booked tickets with past travel dates and automatically updates their status, ensuring ticket records stay accurate.

Show additional properties

Cancel Build flow

Scope: Smart Transit Management System Update set: smart\_transit\_update\_set

## Step 1: Add Trigger

- Click on Add Trigger.
- Select Daily at 00:00:00.

The screenshot shows the ServiceNow Studio interface for creating a trigger. The title bar says "Auto Complete Ticket Status" and "Active". The main area is titled "TRIGGER" and contains a "Daily at 00:00:00" trigger configuration. Below it, the "ACTIONS" section lists three steps:

1. Look Up Tickets Records where (Status is Booked, and Travel Date before 2025-09-11 10:00:00)
2. For Each Item in [1 - Look Up R... • Tickets Reco...]
3. Update Tickets Record

At the bottom, status bars indicate "Status: Published" and "Application: Smart Transit Management System".

## Step 2: Add Action – Look Up Records

- Click Add Action → select Look Up Records.
- Configure conditions: Status = Booked and Travel Date < Current Date.

## Step 3: Add Action – For Each

- Add a For Each action.
- Choose the collection from the previous Look Up Records step.

The screenshot shows the configuration of a "Look Up Tickets Records" action. The "Action Properties" section shows "Action: Look Up Records". The "Action Inputs" section includes a "Table" dropdown set to "Tickets [x\_cldtd\_smart\_tr\_0\_ticket]" and a "Conditions" section with the following criteria:

All of these conditions must be met

AND

- Status is Booked
- Travel Date before Current hour

or

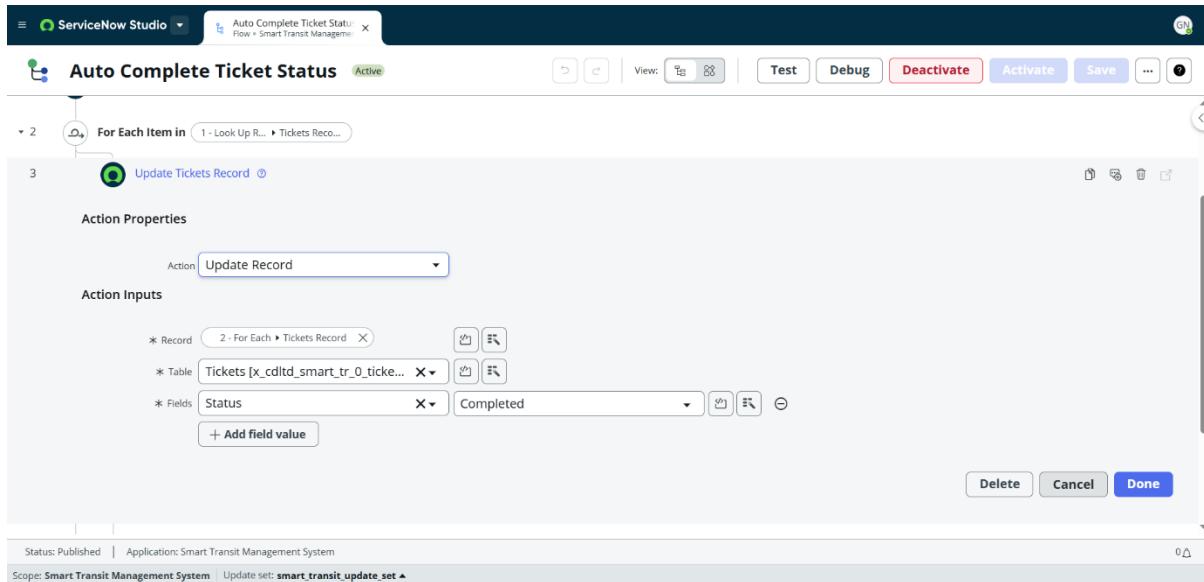
New Criteria

Order by: Select a field, a to z

At the bottom, status bars indicate "Status: Published" and "Application: Smart Transit Management System".

## Step 4: Add Action – Update Record

- Inside the loop, add Update Record.
- Update Ticket Status = Completed (or your required value).



## Step 5: Save and Activate Flow

- Click Save → then Activate.

## Activity 2: Auto fill Ticket Amount

**Flow Name:** Auto fill Ticket Amount

**Application:** Smart Transit Management System

**Description:** This flow automatically calculates and updates the ticket amount whenever a new ticket is created or updated, based on the selected Start Point and End Point.

## Step 1: Add Trigger

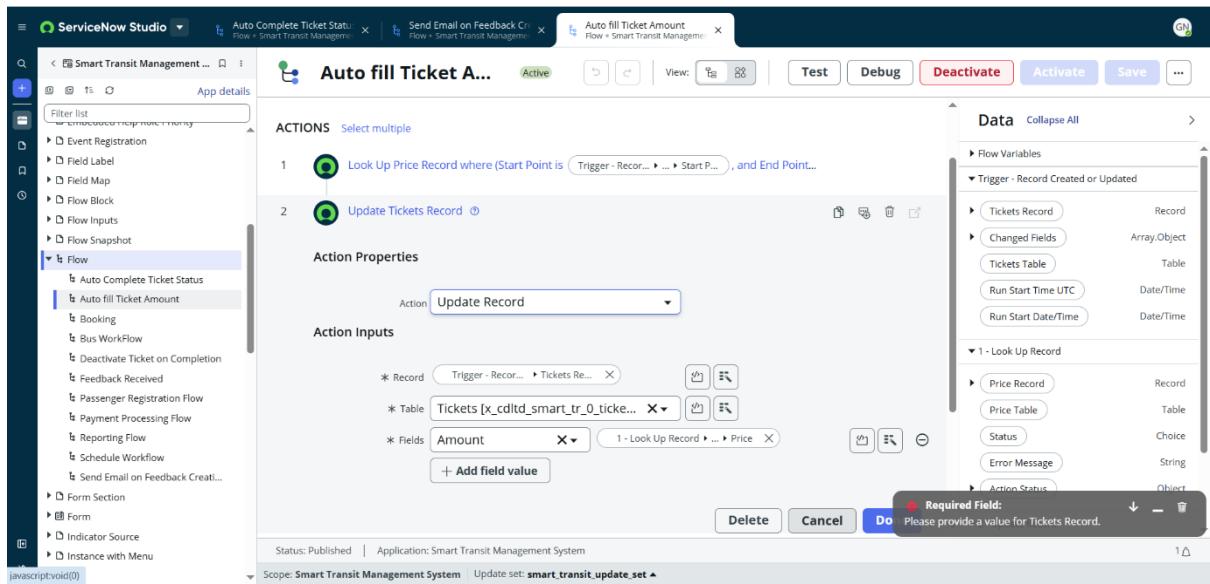
- Click on Add Trigger.
- Select Record Created or Updated.
- Condition: *Tickets Created or Updated where (Start Point is not empty, and End Point is not empty)*.

## Step 2: Add Action – Look Up Record

- Click Add Action → select Look Up Records.
- Configure: *Look Up Price Record where (Start Point = Ticket Start Point, End Point = Ticket End Point).*

## Step 3: Add Action – Update Record

- Add an Update Record action.
- Configure it to update the Ticket record with the calculated price amount.



## Step 4: Save and Activate Flow

- Click Save → then Activate to enable the flow.

## Activity 3: Deactivate Ticket on Completion

**Flow Name:** Deactivate Ticket on Completion

**Application:** Smart Transit Management System

**Description:** This flow automatically deactivates a ticket once its status is updated to *Completed*.

### Step 1: Add Trigger

- Click on Add Trigger.
- Select Record Updated.
- Condition: Tickets Updated where (Status = Completed).

The screenshot shows the ServiceNow Studio interface for configuring a flow named 'Deactivate Ticket'. In the 'TRIGGER' section, there is one trigger: 'Tickets Updated where (Status is Completed)'. This trigger is set to run 'Once'. In the 'ACTIONS' section, there is one action: 'Update Tickets Record'. The action properties show the 'Action' as 'Update Record' and the 'Record' as 'Trigger - Record... > Tickets Rec...'. The 'Table' is set to 'Tickets [x\_cdtd\_smart\_tr\_0\_ticket]' and the 'Fields' are set to 'Active' with a value of 'false'. A tooltip indicates that 'Active' is a required field. The status bar at the bottom shows 'Status: Published' and 'Scope: Smart Transit Management System'.

## Step 2: Add Action – Update Record

- Add an Update Record action.
- Configure it to update the Ticket record so that it becomes deactivated once the status is completed.

This screenshot shows the 'Deactivate Ticket' flow configuration after adding the 'Update Tickets Record' action. The 'ACTIONS' section now contains one action: 'Update Tickets Record'. The 'Action Properties' show the 'Action' as 'Update Record'. The 'Action Inputs' section shows the 'Record' as 'Trigger - Record... > Tickets Rec...', the 'Table' as 'Tickets [x\_cdtd\_smart\_tr\_0\_ticket]', and the 'Fields' as 'Active' with a value of 'false'. The 'Data' panel on the right provides detailed configuration for the 'Update Record' step, including 'Tickets Record' (Record), 'Changed Fields' (Array.Object), 'Tickets Table' (Table), 'Run Start Time UTC' (Date/Time), 'Run Start Date/Time' (Date/Time), and 'Action Status' (Object). The status bar at the bottom shows 'Status: Published' and 'Scope: Smart Transit Management System'.

## Step 3: Save and Activate Flow

- Click Save → then Activate.

## Activity 4: Passenger Registration Flow

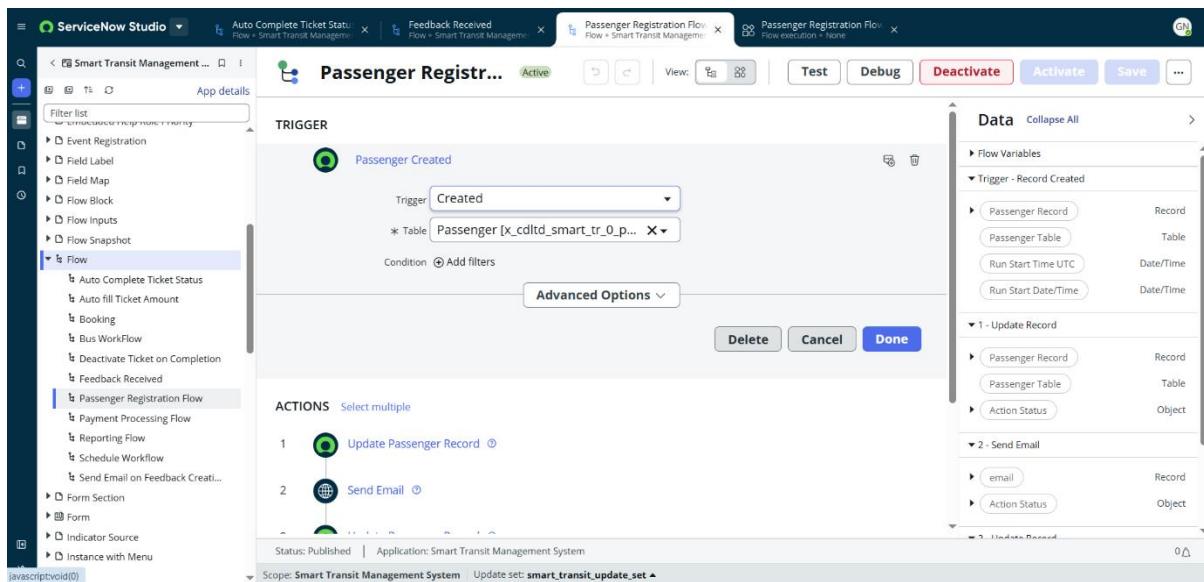
**Flow Name:** Passenger Registration Flow

**Application:** Smart Transit Management System

**Description:** This flow manages passenger registration. When a new passenger is created, it updates the record, sends a confirmation email, and ensures data consistency.

### Step 1: Add Trigger

- Click on Add Trigger.
- Select Record Created.
- Condition: *Passenger Created*.



### Step 2: Add Action – Update Passenger Record

- Add an Update Record action.
- Configure it to update the passenger details after registration.

The screenshot shows the ServiceNow Studio interface for configuring a flow. The main window displays the 'Passenger Registration Flow' with one action: 'Update Passenger Record'. The 'Action Properties' section shows the 'Action' dropdown set to 'Update Record', and the 'Record' dropdown set to 'Trigger - Re...'. The 'Table' dropdown is set to 'Passenger [x\_cdlt\_smart\_tr\_0\_p...]' and the 'Fields' dropdown is set to 'Active'. The right panel shows the flow's variables and steps. The flow consists of three steps: 1. Update Record (Passenger Record, Passenger Table, Action Status), 2. Send Email (email, Action Status), and 3. Update Record (Passenger Record, Passenger Table, Action Status). The status bar at the bottom indicates the flow is published.

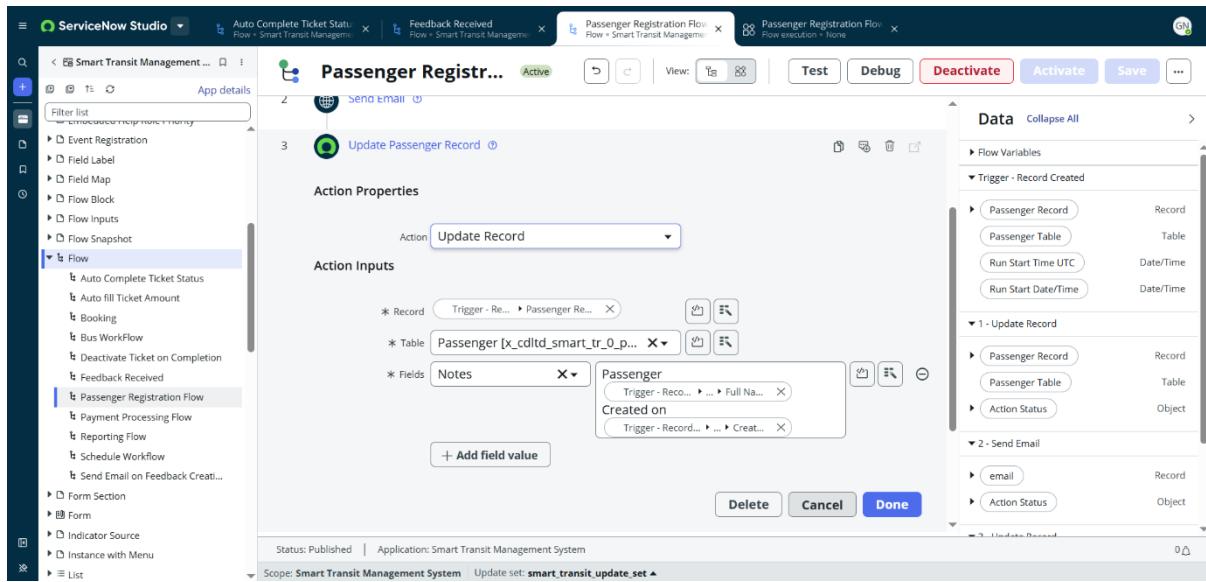
## Step 3: Add Action – Send Email

- Add a Send Email action.
- Configure it to send a confirmation email to the registered passenger.

The screenshot shows the ServiceNow Studio interface for configuring a flow. The main window displays the 'Passenger Registration Flow' with one action: 'Send Email'. The 'Action Properties' section shows the 'Action' dropdown set to 'Send Email', and the 'Target Record' dropdown set to 'Trigger - Re...'. The 'Table' dropdown is set to 'Passenger [x\_cdlt\_smart\_tr\_0\_p...]' and the 'Subject' dropdown is set to 'Trigger - Reco... > ... > Full Na...'. The right panel shows the flow's variables and steps. The flow consists of three steps: 1. Update Record (Passenger Record, Passenger Table, Action Status), 2. Send Email (email, Action Status), and 3. Update Record (Passenger Record, Passenger Table, Action Status). The status bar at the bottom indicates the flow is published.

## Step 4: Add Action – Update Passenger Record (Final)

- Add another Update Record action.
- Configure it to finalize or adjust the passenger's status after email confirmation.



## Step 5: Save and Activate Flow

- Click Save → then Activate.

## Activity 5: Reporting Flow

**Flow Name:** Reporting Flow

**Application:** Smart Transit Management System

**Description:** This flow generates automated reports daily by retrieving active routes and bus assignments, then sends them via email to administrators.

## Step 1: Add Trigger

- Click on Add Trigger.
- Select Run Daily.
- Condition: *Daily at 06:00:00*.

## Step 2: Add Action – Look Up Route Table Records

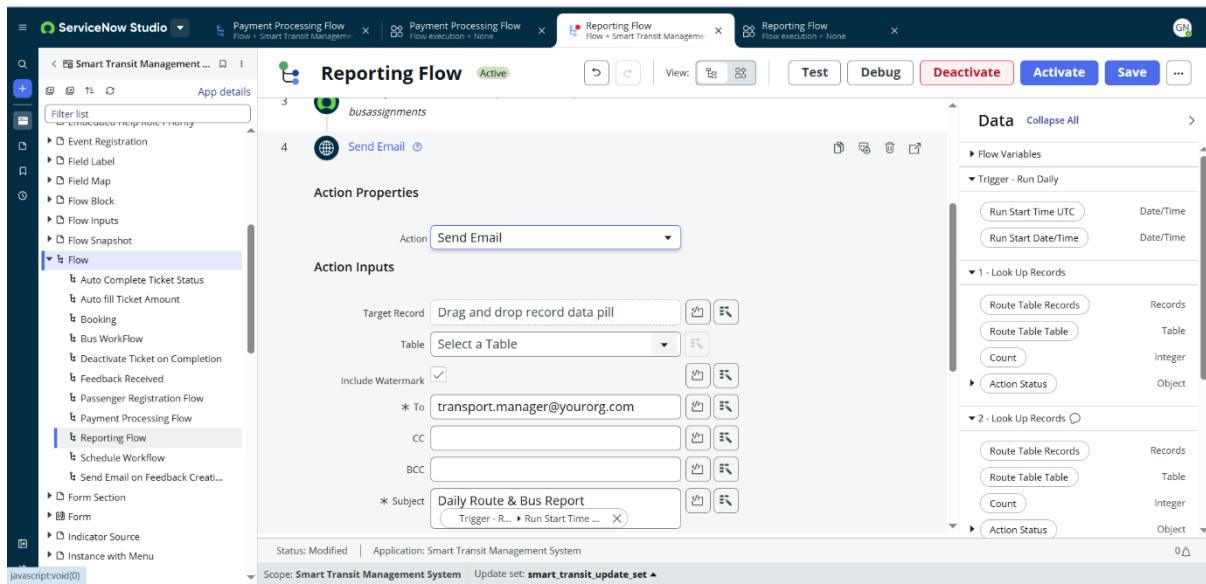
- Add a Look Up Records action.
- Configure it to look up all Route Table Records.

## Step 3: Add Action – Look Up Active Routes

- Add another Look Up Records action.
- Configure it to look up all Route Table Records where Active = true.

## Step 4: Add Action – Send Email

- Add a Send Email action.
- Configure it to send the daily report to the relevant recipients.



## Step 6: Save and Activate Flow

- Click Save → then Activate.

## Activity 6: Send Email on Feedback Creation

**Flow Name:** Send Email on Feedback Creation

**Application:** Smart Transit Management System

**Description:** This flow ensures that whenever feedback is created, an email notification is automatically sent to the admin or support team for quick action.

## Step 1: Add Trigger

- Click on Add Trigger.
- Select Record Created.
- Condition: *Feedback Created.*

The screenshot shows the ServiceNow Studio interface for creating a flow named "Send Email on Fee...".

- Trigger:** Feedback\_New Created (Created, \* Table: Feedback\_New)
- Action:** Send Email (email: admin@example.com, Subject: New Feedback Submitted by)
- Data:** Trigger - Record Created (Feedback\_New Record, Feedback\_New Table, Run Start Time UTC, Run Start Date/Time)
- Advanced Options:** None selected.

## Step 2: Add Action – Send Email

- Add a Send Email action.
- Configure it to send an acknowledgment or notification email to the admin/support team.

The screenshot shows the ServiceNow Studio interface for the "Send Email on Fee..." flow, with the "Action Properties" section visible.

- Action:** Send Email (Send Email button)
- Action Inputs:**
  - Target Record: Drag and drop record data pill
  - Table: Select a Table
  - Include Watermark: checked
  - To: admin@example.com
  - CC: (empty)
  - BCC: (empty)
  - Subject: New Feedback Submitted by

## Step 3: Save and Activate Flow

- Click Save → then Activate.

## Activity 7: Payment Processing Flow

**Flow Name:** Payment Processing Flow

**Application:** Smart Transit Management System

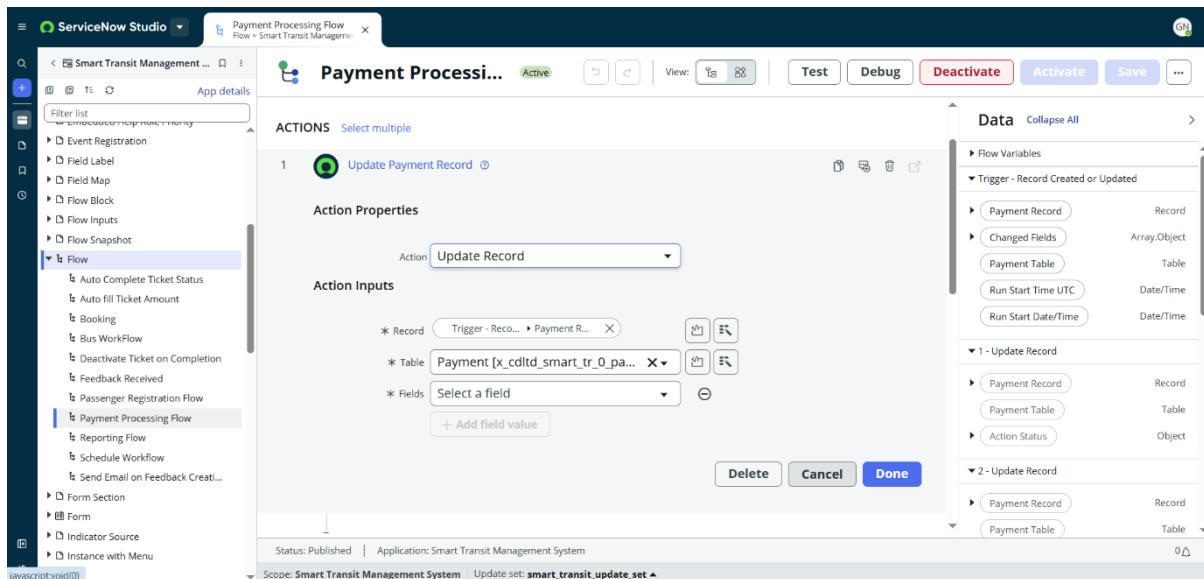
**Description:** This flow ensures that whenever a passenger makes a payment, the payment record is updated, and related actions such as sending a confirmation email or updating ticket status are performed automatically.

### Step 1: Add Trigger

- Click on Add Trigger.
- Select Record Created or Updated.
- Condition: *Payment Record Created or Updated*.

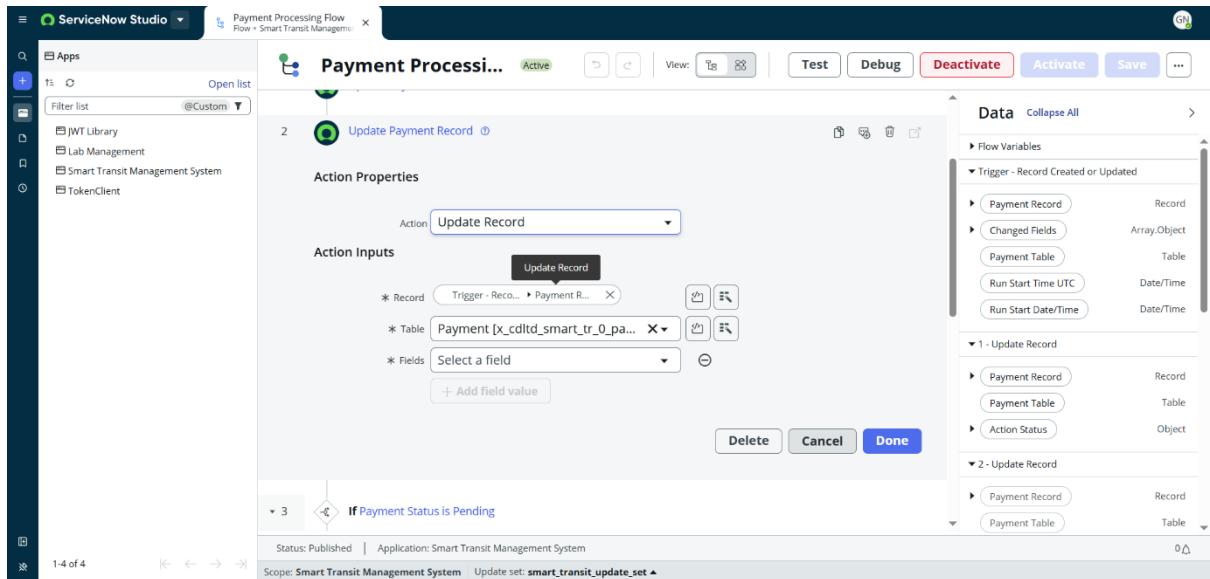
### Step 2: Add Action – Update Payment Record

- Add an Update Record action.
- Configure it to mark the payment as Processed or Completed once validated.



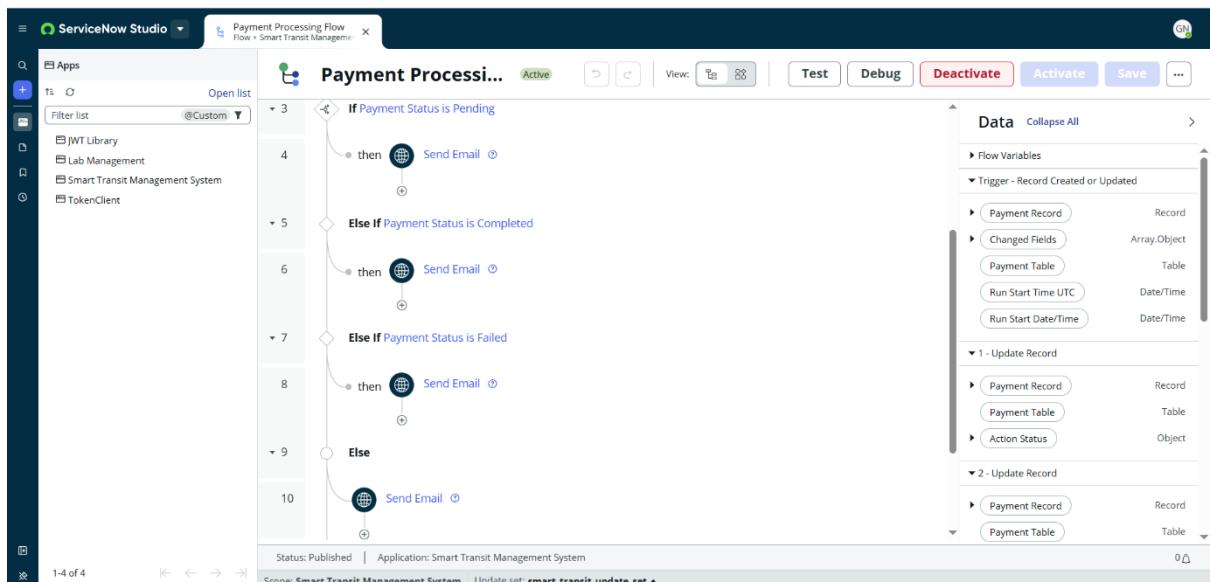
### Step 3: Add Action – Update Payment Record

- Add another Update Record action.
- Configure it to update the Ticket Status (e.g., Paid, Confirmed) once the payment is successful.



## Step 4: Add Action – Send Confirmation Email

- Add a Send Email action.
- Configure it to send a Payment Confirmation Email to the passenger



## Step 5: Save and Activate Flow

- Click Save → then Activate.

## Activity 8: Booking Flow

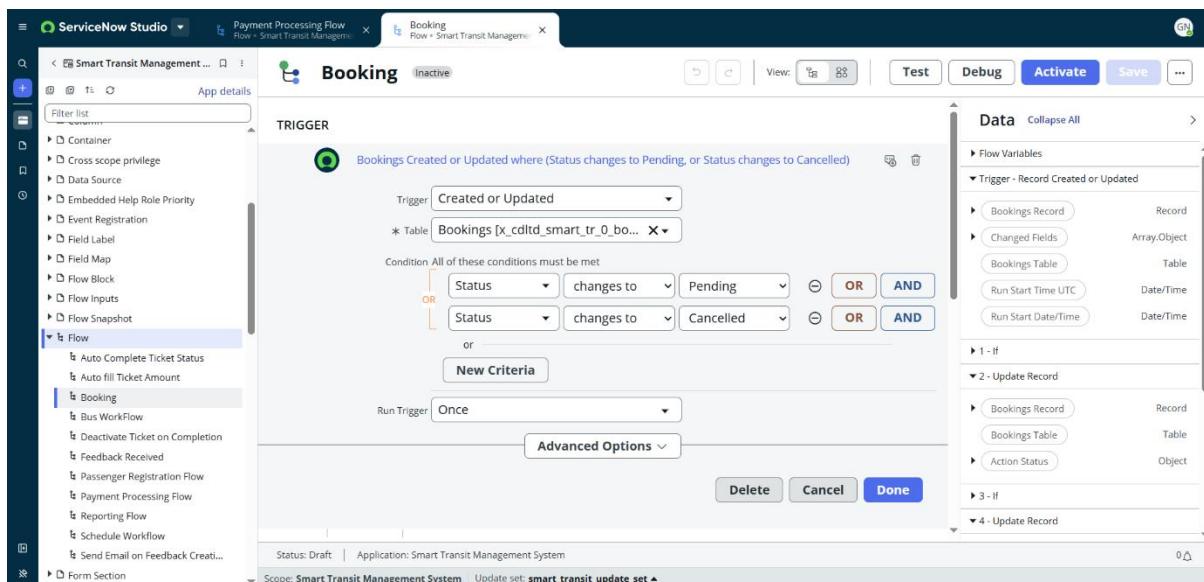
**Flow Name:** Booking Flow

**Application:** Smart Transit Management System

**Description:** This flow automates the booking process. When a passenger creates a new booking, it updates the booking record, assigns a seat, and sends a booking confirmation email.

### Step 1: Add Trigger

- Click on Add Trigger.
- Select Record Created.
- Condition: *Booking Created*.



### Step 2: Add Action – If Status = Pending

- Add a condition If Status == Pending.
- Then add Update Booking Record to set status or details for pending bookings.

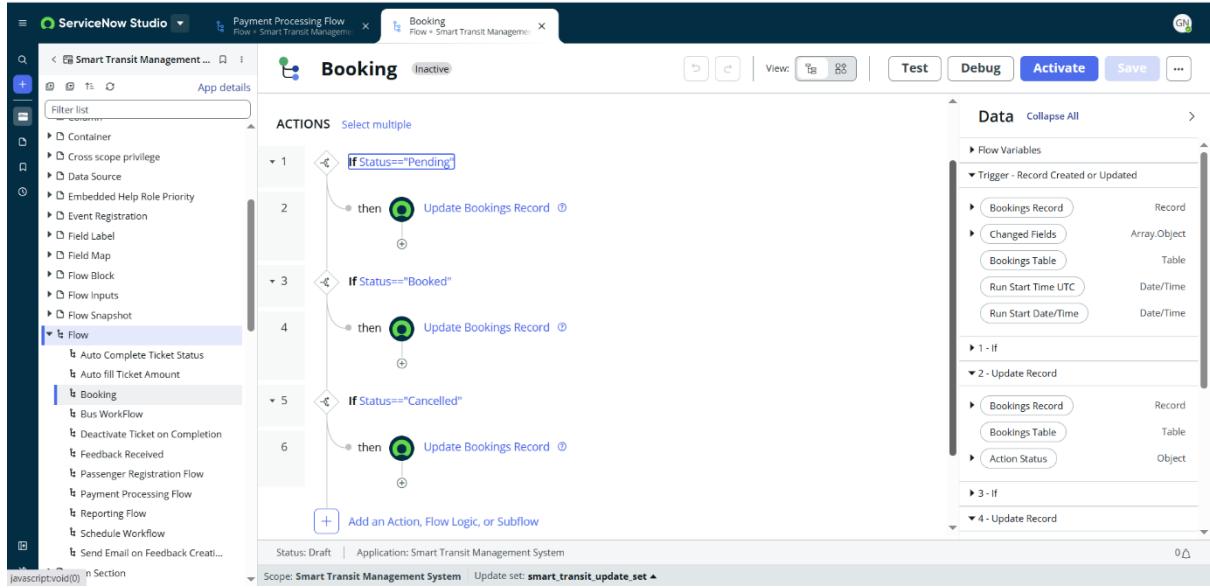
### Step 3: Add Action – If Status = Booked

- Add a condition If Status == Booked.
- Then add Update Booking Record to confirm the booking.

### Step 4: Add Action – If Status = Cancelled

- Add a condition If Status == Cancelled.

- Then add Update Booking Record to mark the booking as cancelled



## Step 5: Save and Activate Flow

- Click Save → then Activate.

## Activity 9: Bus Workflow

**Flow Name:** Bus Workflow

**Application:** Smart Transit Management System

**Description:** This flow manages bus records. When a bus record is created or updated, the system updates related details such as status, schedule assignments, and sends notifications if needed.

### Step 1: Add Trigger

- Click on Add Trigger.
- Select Record Created or Updated.
- Condition: *Bus Record Created or Updated*.

### Step 2: Add Condition – If Empty Bus Name

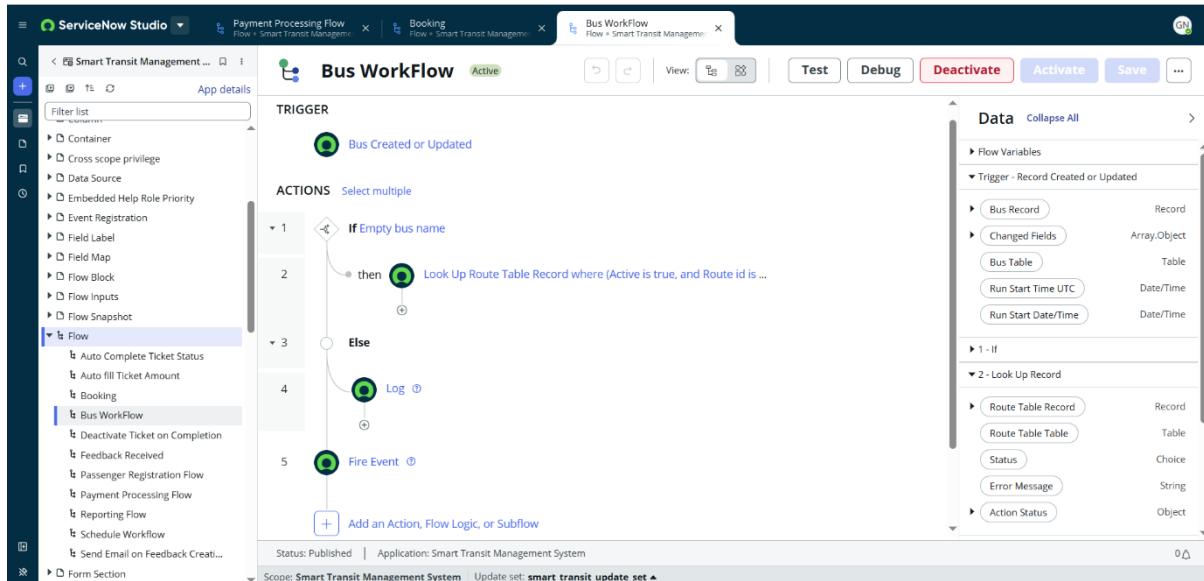
- Add a condition: If Bus Name is Empty.
- Then perform further actions.

### Step 3: Add Action – Look Up Route Table Record

- If the bus name is empty, add a Look Up Records action.
- Configure it to Look Up Route Table Record where Active = true and Route ID matches.

## Step 4: Add Else Condition – Log Action

- Under the Else branch, add a Log Action to record the issue or error.



## Step 5: Add Action – Fire Event

- Add a Fire Event action to trigger subsequent flows or notifications when needed.

## Step 6: Save and Activate Flow

- Click Save → then Activate.

## Activity 10: Schedule Workflow

**Flow Name:** Schedule Workflow

**Application:** Smart Transit Management System

**Description:** This flow manages bus schedules. When a new schedule is created or updated, it ensures driver assignment, bus assignment, and notifies the respective parties.

## Step 1: Add Trigger

- Click on Add Trigger.

- Select Record Created or Updated.
- Condition: *Schedule Record Created or Updated*

## Step 2: Add Action – Send Email (Initial Notification)

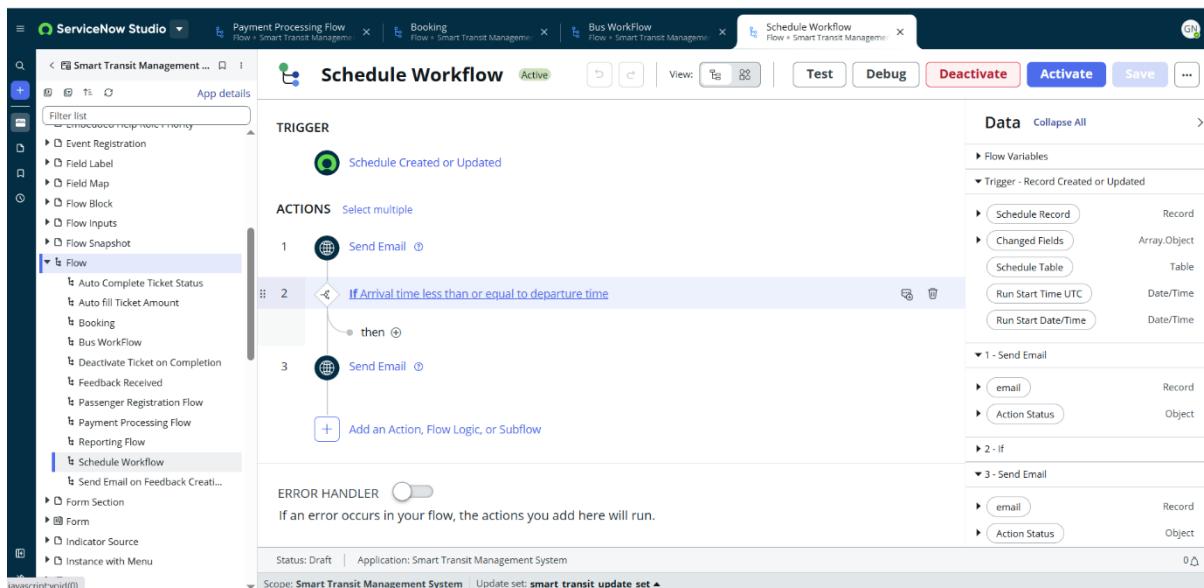
- Add a Send Email action.
- Configure it to notify admins/drivers when a new schedule is created or updated.

## Step 3: Add Condition – Validate Arrival and Departure Time

- Add a condition: If Arrival Time  $\leq$  Departure Time.
- This ensures schedules are logically valid.

## Step 4: Add Action – Send Email (Validation Notification)

- If the condition is met, add another Send Email action.
- Configure it to notify about incorrect or overlapping schedule times.



## Step 5: Save and Activate Flow

- Click Save → then Activate.

# MileStone 7: Creating Reports

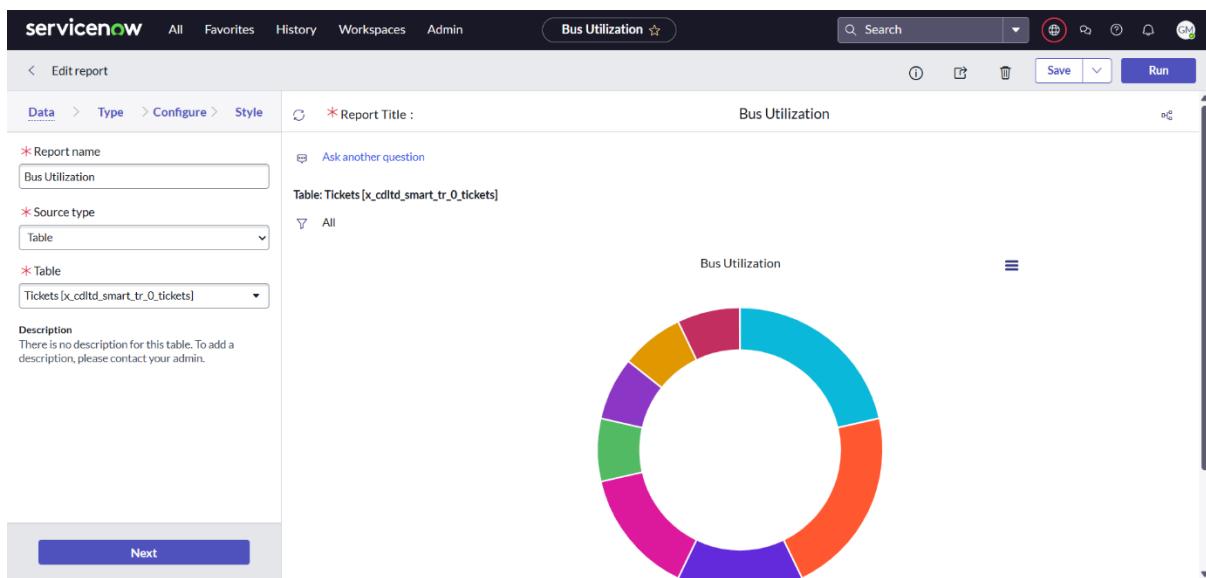
## Activity – 1: Bus Utilization Report

1. Navigate to All -> Reports and click on it.

2. Click on new

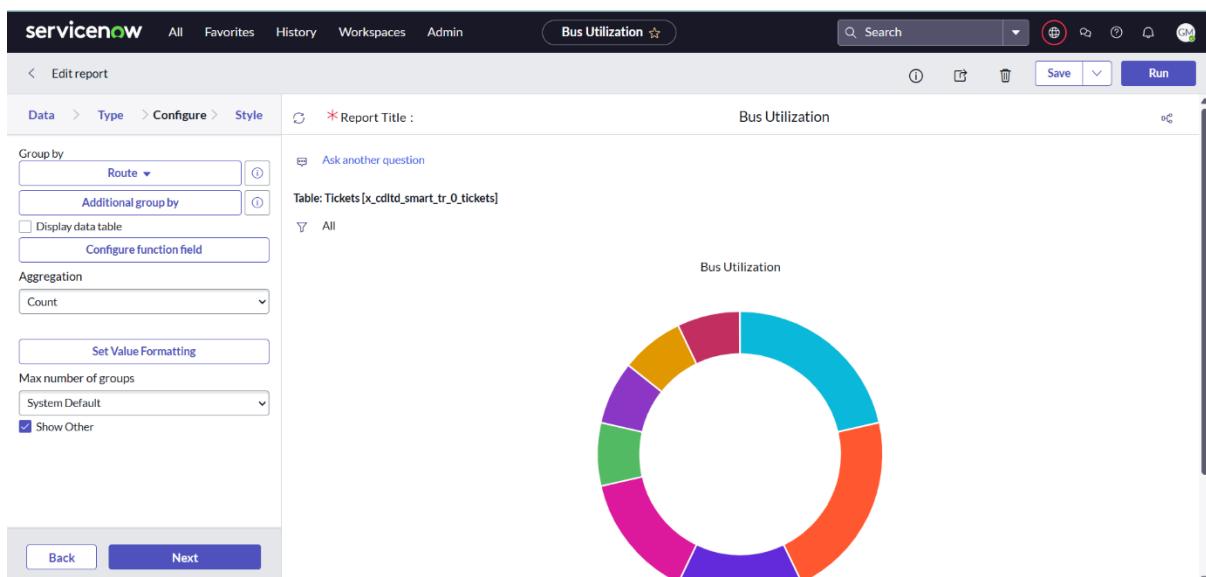
3. On Data pane, Enter following details.

- **Report Name:** Bus Utilization
- **Source Type:** Table
- **Table:** Tickets[c\_cdltd\_smart\_tr\_0\_tickets]

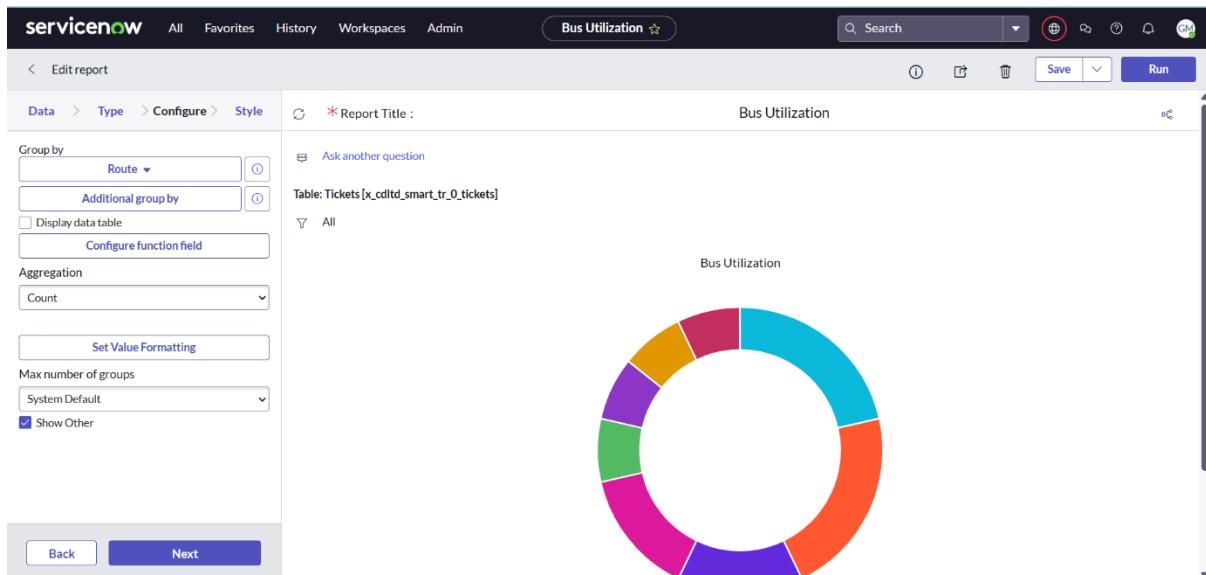


4. Click next and fill details as below in Type and configure pane.

- **Type:** Donut
- **Group by:** Route
- **Aggregation:** Count



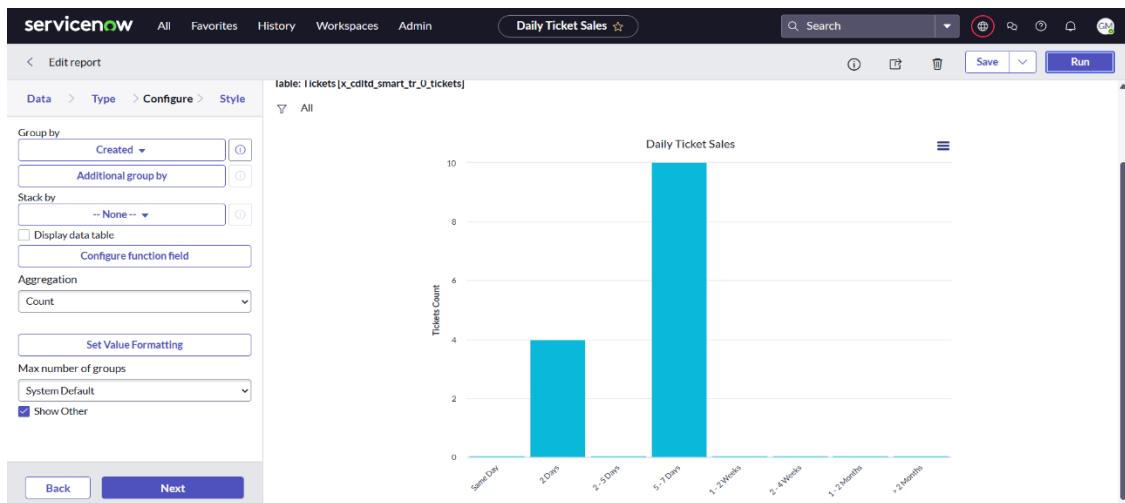
5. Save and Run it.



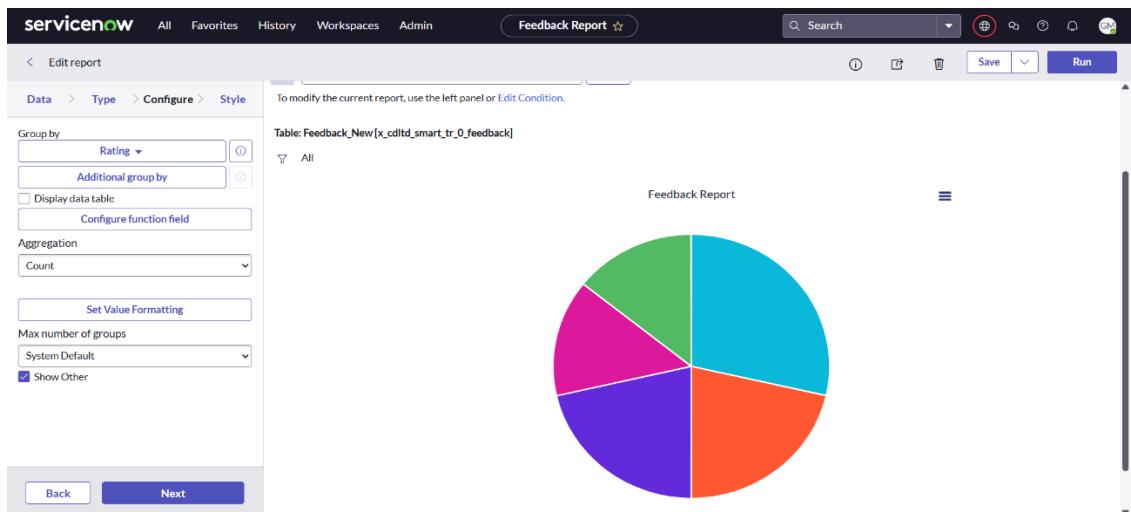
## Activity 2: Create Reports for remaining

1. Repeat above steps for Daily Ticket Sales.

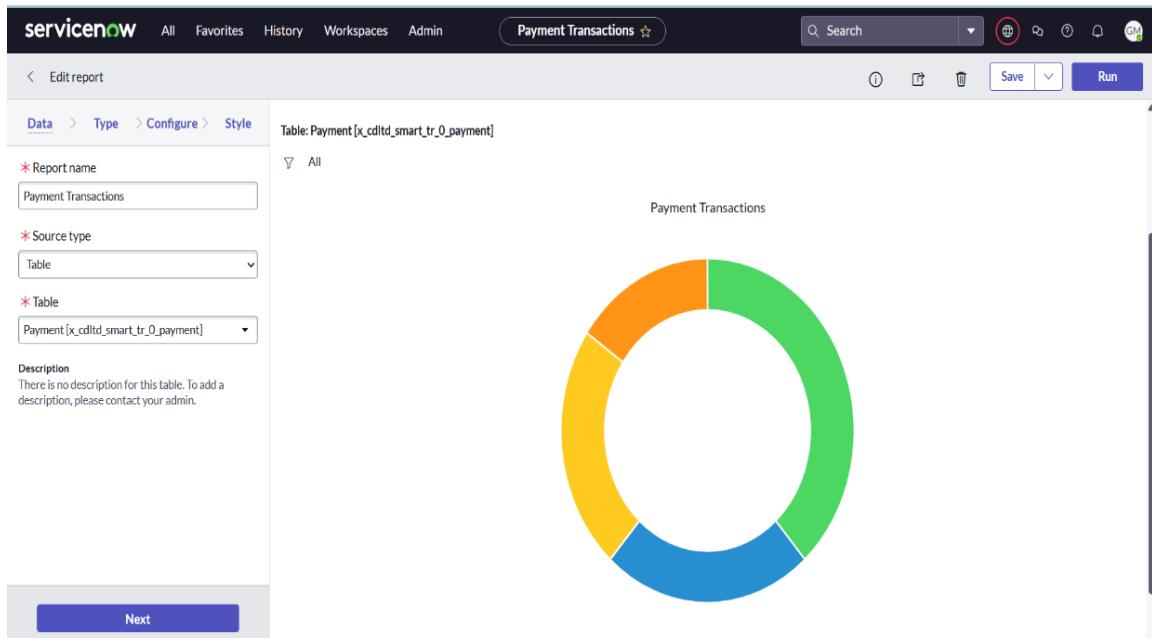
- **Name:** Daily Ticket Sales
- **Table:** Tickets
- **Type:** Bar



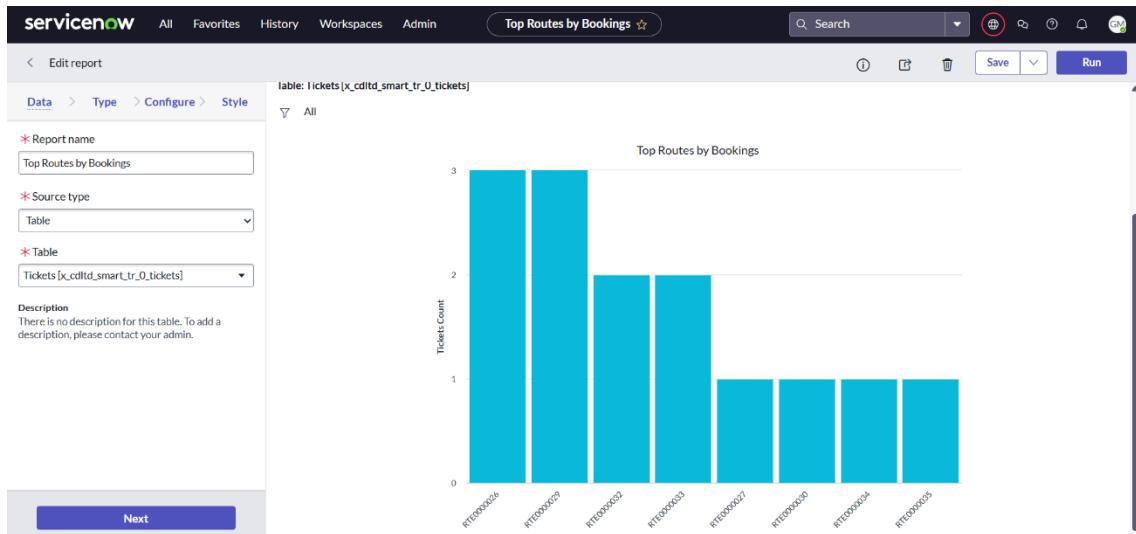
## 2. Feedback Report



### 3. Payment Transactions



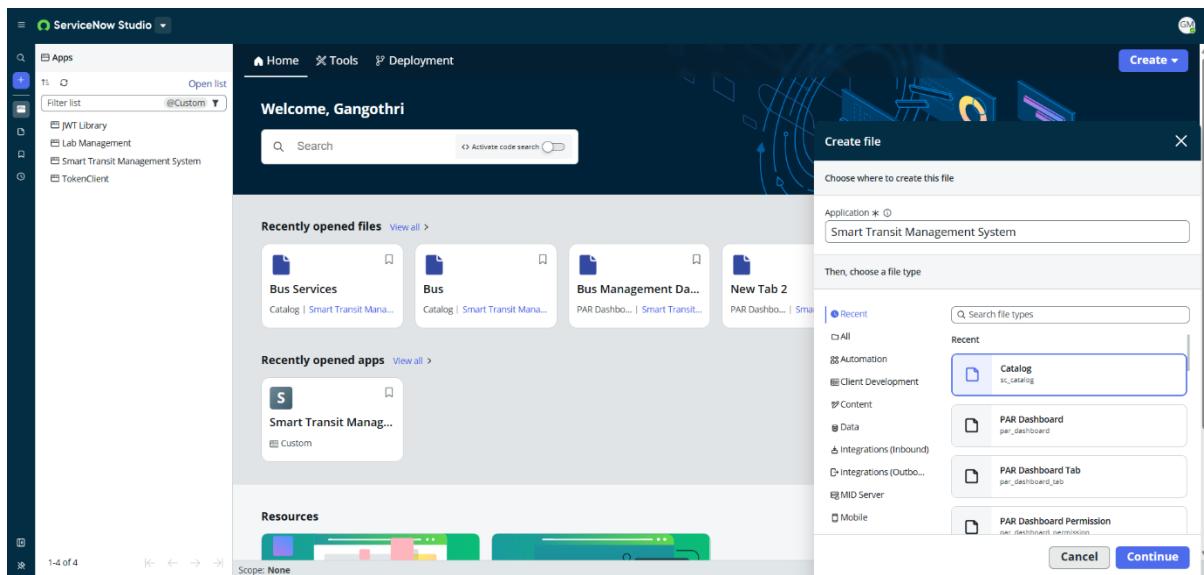
### 4. Top Routes by Bookings



## Milestone 8: Creating Service Catalog

### Activity 1: Create Catalog

**Step-1:** Navigate to ServiceNow Studio and click on create a file.



**Step-2:** Choose Application as Smart Transit Management System and file type as Catalog. Enter details as below and save it.

- **Title:** Bus Services
- **Active:** True
- **Enable Wishlist:** True

- **Description:** The Bus Booking, Reporting, Requesting and Cancellation services allow customers to conveniently reserve seats on available buses.
- Desktop image:



ServiceNow Studio - Bus Services Catalog - Smart Transit Management System

**Catalog Bus Services**

Title: Bus Services  
Manager: [ ]  
Editors: [ ]  
Application: Smart Transit Management System  
Active:   
Enable Wish List:

Description: The Bus Booking, Reporting, Requesting and Cancellation services allow customers to conveniently reserve seats on available buses.

Desktop Image: [Update][Delete]

'Catalog Home' Page: [ ]

'Continue Shopping' page: [ ]

Related Links: Force to Update Set

## Activity-2: Create Categories

**Step-1:** Ensure you are in catalog bus services form. Scroll down to categories pane and click on new.

Catalog Bus Services

'Continue Shopping' page: [ ]

Update Delete

Catalog Items Categories Catalog Portal Pages (1) Sites

for text Search New

Catalog = Bus Services

Search	Title	Description	Active	Parent
[ ]	[ ]	[ ]	[ ]	[ ]

No records to display

**Step-2:** Populate the form as below for Financial Services and save it.

The screenshot shows the 'Smart Transit Management System' interface for creating a new category. The 'Title' field is set to 'Financial Services'. The 'Catalog' field is set to 'Bus Services'. The 'Description' field contains the text: 'Services related to payments and refunds, including ticket fare payments, transaction support, and refund requests for canceled bookings.' The 'Active' checkbox is checked. The 'Application' dropdown is set to 'Smart Transit Management System'. There are buttons for 'Update' and 'Delete' at the top right.

**Step-3:** Repeat Step-2 for Admin Services and Passenger Services Categories.

**Step-4:** Create Catalog items as below:

**1.Name:** Book a Ticket

**Catalogs:** Bus Services

**Category:** Passenger Services

**State:** Published

**Active:** True

**Short description:** Reserve a seat on your preferred bus.

**Description:** Submit details such as bus, date, and passenger information to book your travel ticket.

**Step-5:** Scroll and create variables for Book a Ticket Catalog item as below.

**1.Type:** Single line text

**Catalog item:** Book a Ticket

**Active:** True

**Question:** Please enter your name(customer)...

**Name:** customer\_name

**Label:** your name

**2.Type:** Date/Time

**Catalog item:** Book a Ticket

**Active:** True

**Question:** When do you want to travel?

**Name:** travel\_date

**Label:** Select travel date

**3.Type:** Lookup Select Box

**Catalog item:** Book a Ticket

**Active:** True

**Question:** Select your bus

**Name:** bus\_id

**Label:** Choose bus

**4.Type:** Numeric Scale

**Catalog item:** Book a Ticket

**Active:** True

**Question:** How many seats do you want to book?

**Name:** seats\_required

**Label:** Enter seat count

The screenshot shows the ServiceNow Studio interface for configuring a Catalog Item named "Book a Ticket". The left sidebar shows the navigation tree under "Bus Services". The main area displays the "Catalog Item" configuration page for "Book a Ticket".

**Meta:** [Empty text input field]

**Related Links:**

- Force to Update Set
- Item Diagnostic
- Show VA render type
- Run Point Scan

**Variables (4):**

Type	Question	Order
Single Line Text	Please enter your name(customer)...	100
Date/Time	When do you want to travel?	200
Lookup Select Box	Select your bus	300

**Catalog Item = Book a Ticket:**

Type	Question	Order
Single Line Text	Please enter your name(customer)...	100
Date/Time	When do you want to travel?	200
Lookup Select Box	Select your bus	300
Numeric Scale	How many seats do you want to book?	500

**Actions:** Copy, Update, Edit in Catalog Builder, Delete

**Scope:** Smart Transit Management System **Update set:** smart\_transit\_update\_set

**Step-6:** Create Catalog item: Bus Maintainace Request and Category Admin Services.

**Step-7:** Create variables listed below

1.**Question:** Describe the issue

**Type:** Multiline Text

2.**Question:** Select bus ID

**Type:** Lookup Select Box

3.**Question:** Reported by

**Type:** Single line Text

4.**Question:** Urgency Level

**Type:** Multiple Choice

**Step-8:** Create Catalog item: Cancel item and Category: Passenger Services

Create Variables.

1.**Question:** Enter your Ticket number

**Type:** Single line text

2.**Question:** Reason for cancellation

**Type:** Multi line text

**Step-9:** Create Catalog item: Report Lost Item in Bus and Category: Passenger services

Create Variables.

1. Enter Your name(Single Line Text)
2. Contact details(Single Line Text)
3. Select the bus you have lost item(Lookup Select Box)

4. When did you Travel(Date)
5. Describe the lost item(Multi line text)

### **Step-10: Create Catalog item: Request Refund and Category: Financial Services**

Create Variables as below.

Type	Question	Order
Single Line Text	Enter your name	100
Date	Date of travel	200
Lookup Multiple Choice	Select bus number	300
Single Line Text	Enter your ticket/booking ID	400
Multi Line Text	Why are you requesting a refund?	500
Single Line Text	Refund amount you are requesting	600
Single Line Text	Provide your contact details for updates	700
Multiple Choice	Select payment method used	800

### **Step-11: Update and Click on Activate.**

### **Step-12: On Catalog Portal Pages pane, Add portal page: Catalog\_bus\_services.**

## **Milestone 9: Booking Summary UI Page**

### **Step 1: Page Overview**

- **Page Name:** booking\_summary
- **Application:** Smart Transit Management System
- **Category:** General
- **Description:** Custom page to show booking summary.
- **Endpoint:** x\_cdlted\_smart\_tr\_0\_booking\_summary.do

## Step 2: Purpose

The **Booking Summary UI Page** is designed to provide a visual dashboard that displays real-time booking statistics, available buses, revenue details, and system uptime for administrators and managers.

## Step 3: Navigation Path

Navigate to:

**Smart Transit Management → Booking Summary**

The screenshot shows the ServiceNow UI Page editor interface. At the top, there's a header with 'servicenow' and navigation links for All, Favorites, History, Workspaces, and Admin. The title bar says 'UI Page - booking\_summary'. Below the header, there are fields for Name ('booking\_summary'), Category ('General'), Application ('Smart Transit Management System'), Endpoint ('x\_cldtd\_smart\_tr\_0\_booking\_summary.do'), and Description ('Custom page to show booking summary'). A 'Try It' button is also present. On the left, there's an 'HTML' tab with code editor tools and some CSS code. The main right panel shows a preview of the dashboard with various cards and data.

## Step 4: Page Layout

The page is structured with the following components:

### 1. Dashboard Header

- Title: *Smart Transit Management*
- Subtitle: *Comprehensive bus booking and management dashboard*

### 2. Key Metrics (Top Counters)

- Active Bookings → Shows total current active bookings
- Available Buses → Displays count of buses available
- Today's Revenue → Shows total revenue for the day

- System Uptime → Displays uptime percentage

### 3. Functional Cards (Middle Section)

- **Booking Management** → Manage bookings, seat availability, and passenger details
- **Fleet Management** → Monitor bus locations, schedules, and driver assignments
- **Revenue Center** → Handle payment processing, fare management, and financial reporting

The screenshot shows the ServiceNow Smart Transit Management dashboard. At the top, there's a header bar with 'servicenow' and various navigation links like 'All', 'Favorites', 'History', 'Workspaces', and 'Admin'. A search bar is also present. Below the header is a purple header section with the title 'Smart Transit Management' and a subtitle 'Comprehensive bus booking and management dashboard'. The main area features four key performance indicators (KPIs) in white boxes: '247 ACTIVE BOOKINGS', '15 AVAILABLE BUSES', '₹1,24,500 TODAY'S REVENUE', and '98.2% SYSTEM UPTIME'. Below these KPIs are three main functional cards: 'Booking Management' (orange), 'Fleet Management' (blue), and 'Revenue Center' (yellow). Each card has a small icon, a title, a brief description, and a call-to-action button.

### 4. Quick Actions (Bottom Section)

- New Booking
- Add Bus
- Process Payment
- View Reports
- Settings

The screenshot shows a 'Quick Actions' bar at the bottom of the interface. It has a purple header with the text 'Quick Actions' and five rounded rectangular buttons below it, each with an icon and a label: 'New Booking' (yellow), 'Add Bus' (blue), 'Process Payment' (green), 'View Reports' (orange), and 'Settings' (purple).

## Step 5: Functionality

- Displays real-time booking summary data.

- Provides quick navigation to other modules via functional cards.
- Allows users to perform frequent operations directly using quick action buttons.

## Step 6: Data Flow

1. UI Page is loaded via endpoint.
2. Data such as active bookings, buses, revenue, and uptime are fetched from related tables (Booking, Bus, Payment, System Logs).
3. The data is displayed in summary counters.
4. Clicking on functional cards or quick actions redirects the user to respective modules.

## Step 7: Roles & Permissions

- **Transit\_Admin** → Full access to dashboard.
- **Booking\_Manager** → Can view booking summary and manage bookings.
- **Finance\_Manager** → Can access revenue and payment actions.

## Step 8: Dependencies

- Booking table (to fetch active bookings).

Bookings							
Booking ID	Bus ID	Passenger Name	Date	Time	Status	Seat Number	
B003	BUS009	Priya Mehta	2025-09-08	10:00:00	Pending	5	
B009	BUS001	Gangotri	2025-09-08	07:15:00	Pending	2	
B007	BUS009	Mounika	2025-09-07	11:30:00	Booked	7	
B010	BUS005	Diluya Kapoor	2025-09-09	07:20:00	Booked	6	
B006	BUS008	Sneha	2025-09-09	08:50:00	Pending	6	
B002	BUS002	Navadeep	2025-09-07	00:05:00	Booked	3	
B011	BUS005	Navadeep	2025-09-24	15:00:00	Booked	18	
B005	BUS003	Nithil	2025-09-08	08:45:00	Cancelled	4	
B004	BUS002	Arjun Reddy	2025-09-08	10:10:00	Cancelled	6	
B008	BUS006	Ganesh	2025-09-08	11:40:00	Booked	8	
B001	BUS003	Rohan Sharma	2025-09-06	09:00:00	Booked	1	

- Bus table (to fetch available buses).

Bus						
	Bus ID	Bus Name	Capacity	Driver ID	Route id	Status
	BUS005	ST	110	DRV0012	RTE0000027	Active
	BUS008	F1g	120	DRV0008	RTE0000028	Active
	BUS003	PQ	100	DRV0008	RTE0000029	Maintenance
	BUS009	V04	110	DRV0010	RTE0000029	Active
	BUS011	avb	120	DRV0010	RTE0000029	Maintenance Due
	BUS007	B2c	170	DRV0013	RTE0000030	Maintenance
	BUS001	A/M	100	DRV0006	RTE0000030	Active
	BUS004	XYZ	120	DRV0011	RTE0000033	Active
	BUS010	HFW	85	DRV0011	RTE0000033	Active
	BUS002	bbb	75	DRV0007	RTE0000034	Inactive
	BUS006	A1b	100	DRV0009	RTE0000034	Inactive

- Payment/Transaction table (for daily revenue).

Payments								
	Amount	Booking Passenger	Passengers	Payment Date	Payment ID	Payment Method	Payment Status	Transaction ID
	E1.0062	4e0cd81e037722107443dfb0fda1e0d	Priya Mehta	2025-09-02 14:28:10	PAYID01016	Net Banking	Pending	PAYID01015
	E47.542	865d989e037722107443dfb0fda1ec2	Gangothri	2025-10-01 07:23:00	PAYID01026	UPI	Failed	PAYID01025
	E112.776	8ff5c09e037722107443dfb0fda1e57	Ganesh	2025-09-27 07:22:31	PAYID01024	Net Banking	Pending	PAYID01023
	E75.2119	977969e037722107443dfb0fda1ecb	Mounika	2025-10-03 07:24:18	PAYID01000	Card	Failed	PAYID01029
	E26.9277	9edc5c5e037722107443dfb0fda1e46	Arjun Reddy	2025-09-02 04:55:42	PAYID01018	UPI	Refunded	PAYID01017
	E3.8235	a2db941e037722107443dfb0fda1e4	Divya Kapoor	2025-10-07 07:23:42	PAYID01028	Cash	Failed	PAYID01027
	E64.1563	bacd05e037722107443dfb0fda1e31	Sneha	2025-09-02 14:28:10	PAYID01022	Cash	Failed	PAYID01021
	E1.0062	bb7c585e037722107443dfb0fda1e57	Gangothri	2025-09-25 07:25:29	PAYID01032	Net Banking	Refunded	PAYID01031
	E10.3133	bb7c585e037722107443dfb0fda1e57	Rohan Sharma	2025-09-24 07:26:33	PAYID01036	Cash	Completed	PAYID01035
	E9.2233	cc4a1c9a037722107443dfb0fda1e7b	Rohan Sharma	2025-08-20 10:59:50	PAYID01012	Card	Completed	PAYID01011
	E812762	cc3d949e037722107443dfb0fda1ee2	Navadeep	2025-09-02 04:55:42	PAYID01014	Cash	Failed	PAYID01013
	E47.542	bb7c585e037722107443dfb0fda1e57	Nikhil	2025-09-30 07:21:05	PAYID01020	Card	Completed	PAYID01019
	E3.7732	bacd05e037722107443dfb0fda1e31	Nikhil	2025-09-02 14:28:10	PAYID01034	UPI	Pending	PAYID01033

## Step 9: Testing & Validation

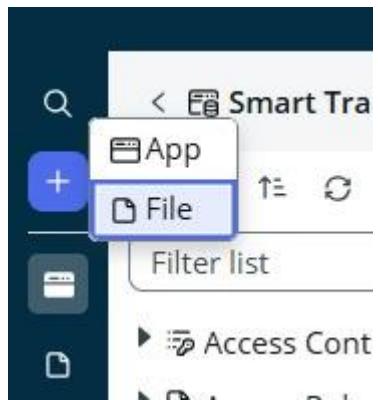
- Verified that booking count updates when new booking is added.
- Confirmed available buses update when a bus is marked inactive.
- Tested that revenue value changes after payment processing.
- Validated quick action buttons redirect correctly to respective modules.

## Step 10: Future Enhancements

- Add trend charts (e.g., daily/weekly bookings, revenue trends).
- Enable role-based widgets (e.g., Finance view vs Operations view).
- Integrate alerts/notifications for critical events.

## Mile stone 10: Creating Business Rules

1. Select the **Create** icon [+] and then select **File**.

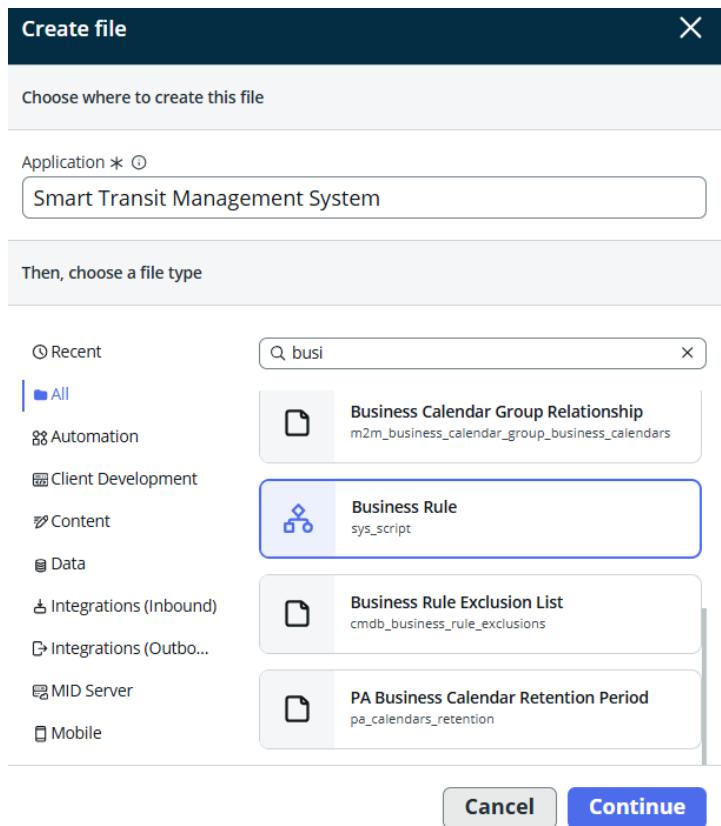


2. Specify the **Application** the file will belong to: *Smart Transit Management System*.

3. Select **Automation** on the left-side panel.

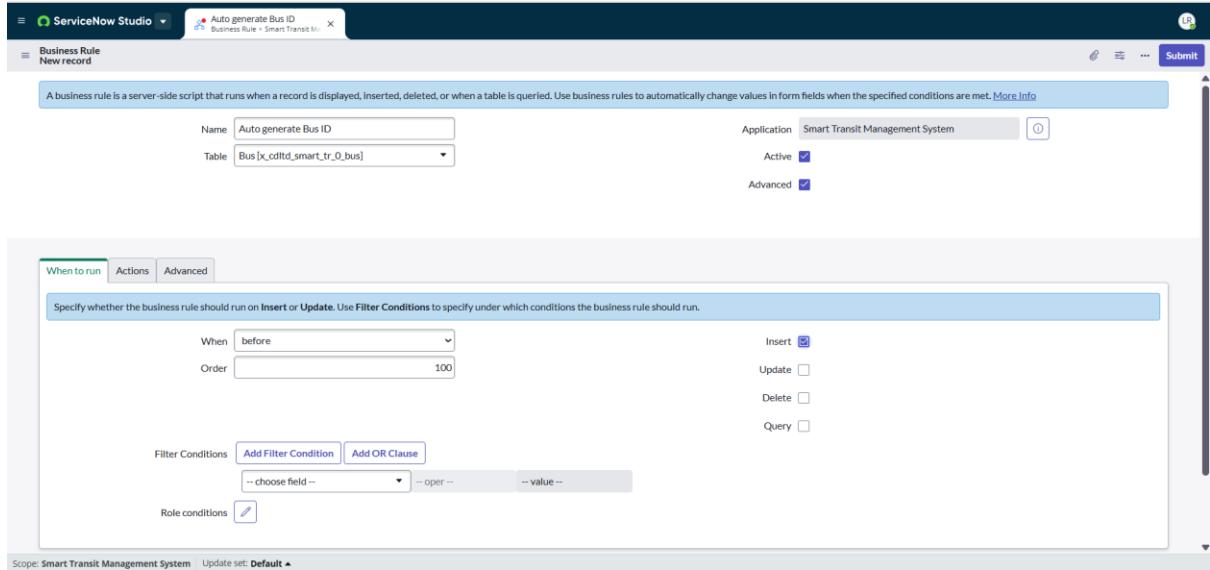
4. Scroll down and select **Business Rule(sys\_script)** as the file type.

5. Select **Continue**.



## Activity 1:

- Name: Auto generate Bus ID  
Table: Bus[x\_cdlt\_d\_smart\_tr\_0\_bus]
- Select **Advanced**.
- In when to run section
  - When: Before
  - Order: 100.
  - Select **Insert**.



## 1.In Advanced Section Add script.

```
(function executeRule(current, previous /*null when async*/) {
    var fieldName = 'bus_id';
    var prefix = "BUS";
    if (!current[fieldName]) {
        var gr = new GlideRecord('x_cdltd_smart_tr_0_bus');
        gr.orderByDesc('sys_created_on');
        gr.setLimit(1);
        gr.query();
        var nextNumber = 1;
        if (gr.next()) {
            var lastID = gr[fieldName].toString();
            var numericPart = parseInt(lastID.replace(prefix, ""), 10);
            if (!isNaN(numericPart)) {
                nextNumber = numericPart + 1;
            }
        }
        current[fieldName] = prefix + nextNumber.toString().padStart(3, '0');
    }
})
```

```

        }
})(current, previous);

```

## 2. Select Submit.

```

Business Rule
New record

When to run Actions Advanced

Condition
Script Turn on ECMAScript 2021 (ES12) mode ⓘ
1 (function executeRule(current, previous /*null when async*/){
2
3     var fieldName = 'bus_id';
4     var prefix = "BUS";
5
6     if (!current[fieldName]) {
7         var gr = new GlideRecord('x_cdlt_d_smart_tr_0_bus');
8         gr.orderByDesc('sys_created_on');
9         gr.setLimit(1);
10        gr.query();
11
12        var nextNumber = 1;
13        if (gr.next()) {
14            var lastID = gr[fieldName].toString();
15            var numericPart = parseInt(lastID.replace(prefix, ''), 10);
16            if (!isNaN(numericPart)) {
17                nextNumber = numericPart + 1;
18            }
19        }
20
21        current[fieldName] = prefix + nextNumber.toString().padStart(3, '0');
22    }
23 })(current, previous);

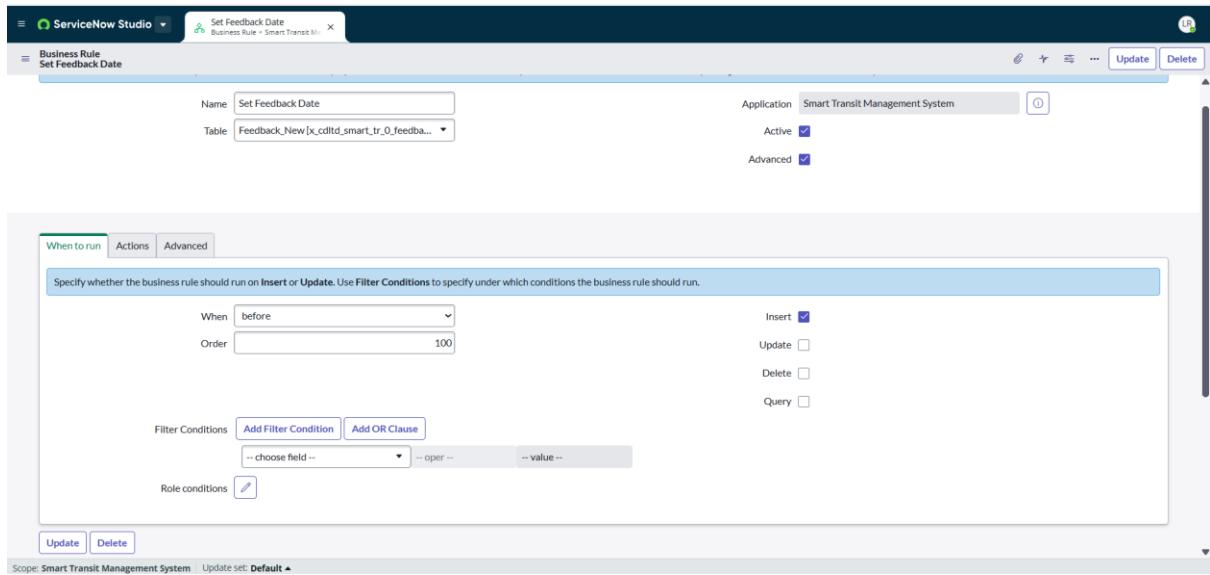
```

Submit

Scope: Smart Transit Management System Update set: Default ▾

## Activity 2:

- Name: Set Feedback date  
Table: Feedback\_new[x\_cdlt\_d\_smart\_tr\_0\_bus]
- Select **Advanced**.
- In when to run section
  - When: Before
  - Order: 100.
  - Select **Insert**.



## 1.In Advanced Section Add script.

```
(function executeRule(current, gSNC, gs) {
    if (!current.feedback_date) {
        current.feedback_date = new GlideDateTime();
    }
})(current, gSNC, gs);
```

## 2.Select Submit.

**Similarly create business rule for other tables to auto generate id,**

- 1.Bookings[x\_cdltd\_smart\_tr\_0\_bookings]
- 2.Route Table[x\_cdltd\_smart\_tr\_0\_route\_table]
- 3.Driver[x\_cdltd\_smart\_tr\_0\_driver]
- 5.Passenger[x\_cdltd\_smart\_tr\_0\_passenger]
- 6.Payment[x\_cdltd\_smart\_tr\_0\_payment]
- 7.Schedule\_table[x\_cdltd\_smart\_tr\_0\_schedule\_table]
- 9.Tickets[x\_cdltd\_smart\_tr\_0\_tickets]

## Milestone 11: Knowledge Base

### 1. Accessing the Knowledge Base

**Step 1:** Log in to the **Smart Transit Management System Portal**.

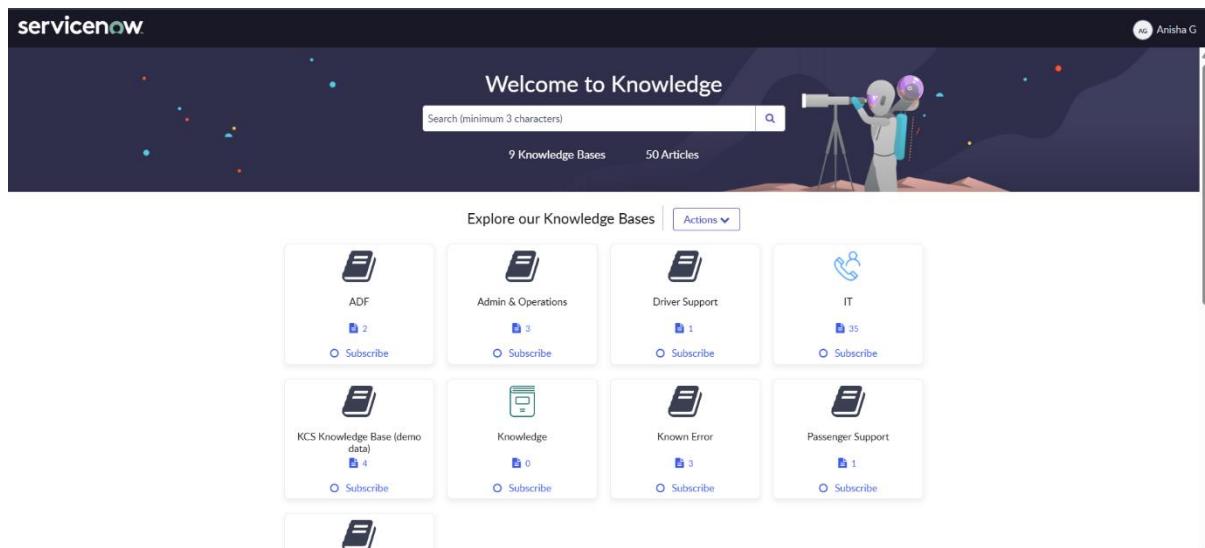
**Step 2:** From the homepage, click on **Knowledge Search**.

**Step 3:** In the search bar, enter at least **3 characters** of your query.

**Step 4:** Click on **Filters** and select the required category:

- Admin & Operations
- Driver Support
- Passenger Support

**Step 5:** Click on the desired article title to open and view full details.



### 2. Admin & Operations Knowledge Articles

#### (a) How to Add, Remove, or Update a Driver

**Step 1:** Log in to the **Admin Portal**.

**Step 2:** Navigate to **Admin Dashboard → Driver Management → Add Driver**.

**Step 3:** Enter driver details (Name, Driver ID, Contact, Vehicle).

**Step 4:** Save the record to complete driver addition.

**Step 5:** To update or remove, select the driver record and perform the required action.

### **(b) How to Generate Weekly or Monthly Reports**

**Step 1:** Log in to the **Admin Portal**.

**Step 2:** Navigate to **Admin Dashboard → Reports**.

**Step 3:** Choose **Weekly Report or Monthly Report**.

**Step 4:** Generate the report and download/export if needed.

### **(c) How to Schedule and Manage Routes**

**Step 1:** Log in to the **Operations Portal**.

**Step 2:** Navigate to **Dashboard → Routes → Manage Routes**.

**Step 3:** Click **Add Route**.

**Step 4:** Enter route details (Start Point, End Point, Time, Driver).

**Step 5:** Save changes.

**Step 6:** Use **Edit/Delete** to manage existing routes.

The screenshot shows a ServiceNow Knowledge Search results page. At the top, there is a search bar with placeholder text "Search (minimum 3 characters)" and a magnifying glass icon. Below the search bar are filter options: "Filters" and "Applied Filters: Admin & Operations". The results section displays three articles:

- How to Add, Remove, or Update a Driver**: Admin & Operations | Driver Management. This article explains how Admins can manage drivers in the Smart Transit Management System. Add a Driver Log In to the Admin Portal. Navigate to Admin Dashboard → Driver Management → Add Driver. Enter driver details: Name, Driver ID (Auto-populated), Contact Assign... Monalisa Saha • 5 Views • 17d ago • ⓘ ⓘ ⓘ ⓘ ⓘ
- How to Generate Weekly or Monthly Reports**: Admin & Operations | Reports. This article explains how Admins can generate weekly and monthly reports in the Smart Transit Management System for performance monitoring and record-keeping. Generate a Weekly Report Log In to the Admin Portal. Navigate to Admin Dashboard → Reports... Monalisa Saha • 1 View • 12d ago • ⓘ ⓘ ⓘ ⓘ ⓘ
- How to Schedule and Manage Routes**: Admin & Operations | Scheduling. This article explains how Operations staff can create and manage routes in the Smart Transit Management System. Add or Create a Route Log In to the Operations Portal. Go to Dashboard → Routes → Manage Routes. Click Add Route and enter route details (Start/End points,... Monalisa Saha • 1 View • 12d ago • ⓘ ⓘ ⓘ ⓘ ⓘ

Below the results, there are sorting and filtering options: "Sort by: Views Newest Alphabetical" and "Clear All".

## **3. Driver Support Knowledge Articles**

### **(a) Guides and Resources for Drivers – How to Check Assigned Routes**

**Step 1:** Open the **Smart Transit Mobile App**.

**Step 2:** Log in and go to **Driver Dashboard**.

**Step 3:** Click on **My Routes**.

**Step 4:** Select the date.

**Step 5:** View assigned route details:

- Start & Destination Points
- Scheduled Time
- Vehicle Number
- Assigned Passengers

The screenshot shows a ServiceNow Knowledge Search results page. At the top, there is a search bar with placeholder text 'Search (minimum 3 characters)' and a magnifying glass icon. Below the search bar is a 'Filters' button. The main area displays '1 Results' for the query 'Driver Support'. The result card for 'Guides and resources for drivers.' includes the following details:

- Category: Driver Support | Route Assignment
- Description: How to Check Assigned Routes (Drivers) Audience: Drivers Category: Route Assignment Steps Log in to the Smart Transit mobile app. Go to Driver Dashboard → My Routes. Select the Date. Your assigned route details will display: Starting point & destination Scheduled time Vehicle number Assigned passengers (if...)
- Views: 4
- Last updated: 12d ago
- Rating: ★★★★☆

## 4. Passenger Support Knowledge Articles

### (a) How to Book a Bus Ticket

**Step 1:** Open the Smart Transit Web Portal.

**Step 2:** Sign in or continue as Guest.

**Step 3:** Enter trip details – From, To, Date, Time, Number of Passengers.

**Step 4:** Click Search.

**Step 5:** Select the desired bus from results.

**Step 6:** Confirm booking details.

**Step 7:** Proceed to payment and confirm ticket.

The screenshot shows a ServiceNow Knowledge Search interface. At the top, there's a search bar with the placeholder "Search (minimum 3 characters)" and a magnifying glass icon. Below it is a "Filters" button. The main area displays a single result titled "How to book a bus ticket?". The result includes a small icon of a document, the title, a brief description, and some metadata: "Passenger Support | Booking Tickets", "Book on the Web Portal Open the portal Go to: https://your-portal-url [Screenshot: Portal home with 'Plan your trip'] Sign in / Continue as guest Click Sign in and enter your credentials. [Screenshot: Sign-in page] Plan your trip From & To Travel Date & Time Window Number of Passengers Click Search. [Screenshot...]", "Anisha G • 0 Views • 12d ago • ⓘ ⓘ ⓘ ⓘ", and sorting options "Sort by" (Views, Newest, Alphabetical) and a "Clear All" button.

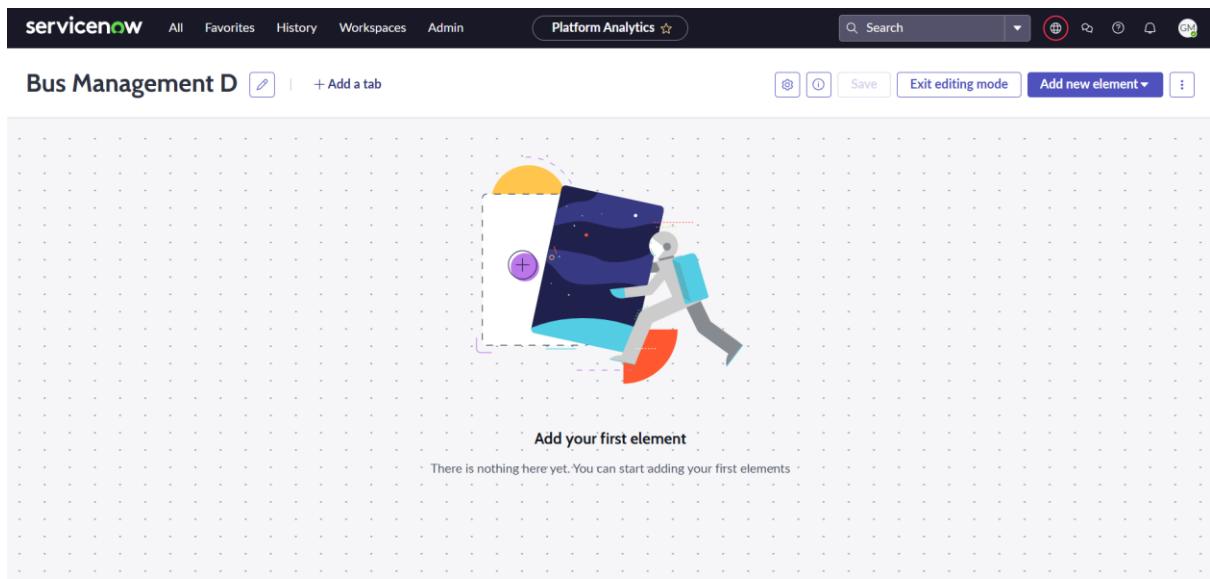
## Milestone 12: Dashboards

### Activity 1: Bus Management Dashboard

**Step-1:** Navigate to All > Dashboards and Click on Create dashboard.

The screenshot shows the ServiceNow Dashboards page. At the top, there are navigation links for "All", "Favorites", "History", "Workspaces", and "Admin", followed by a "Platform Analytics" button. A search bar and a "Create new dashboard" button are also at the top. On the left, a sidebar has a "Filter by name or owner" search bar and a list of filters: "Recent", "Bookmarked", "Certified", "Owned by me", "Shared with me", and "All" (which is selected and highlighted in green). The main area, titled "Dashboards", contains a grid of 15 dashboard cards. Each card has a title, a preview image, and ownership information. The cards include: "Analytics Usage Overview" (Owner: System Administrator), "Application Services Dashboard" (Owner: System Administrator), "Asset Overview" (Owner: System Administrator), "Bus Management Dashboard" (Owner: Anisha G), "Change Request" (Owner: System Administrator), "Data Classification" (Owner: System Administrator), "Data Usage Visualization Console" (Owner: System Administrator), "Feedback Overview" (Owner: Kothakonda Navadeep), "Feedback Dashboard" (Owner: Kothakonda Navadeep), "Gen AI Actions Dashboard" (Owner: System Administrator), "Guided Tours – Operational Reporting" (Owner: System Administrator), "Incident Management" (Owner: System Administrator), "Instance Scan Results Next Experience Dashboard" (Owner: System Administrator), "Interaction" (Owner: System Administrator), "IT Agent Dashboard" (Owner: System Administrator), and "Knowledge Management Overview" (Owner: System Administrator).

**Step-2:** Enter name : Bus Management Dashboard and click Create new dashboard.

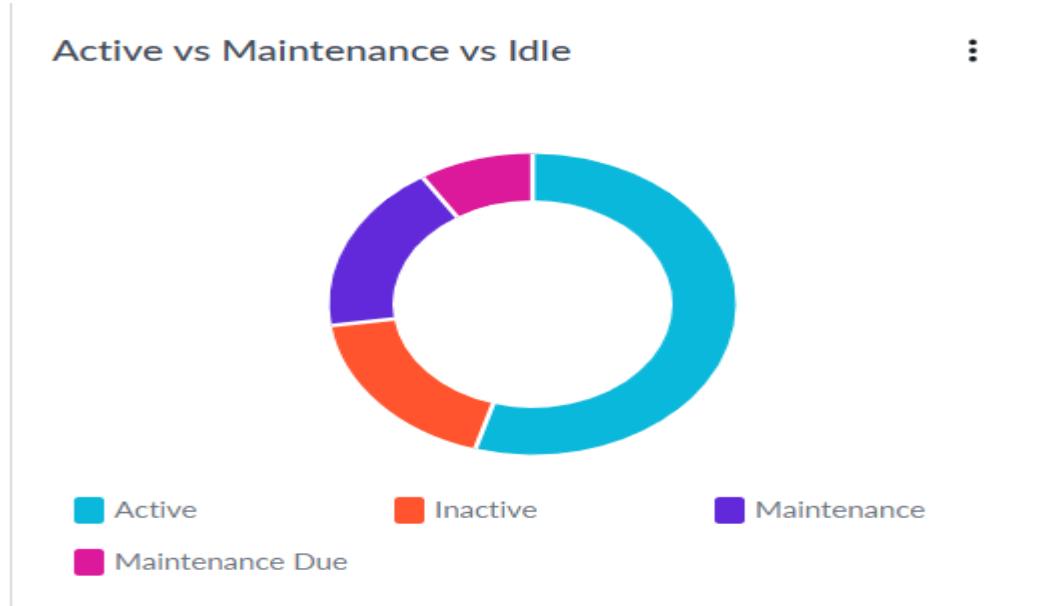


**Step-3:** Click on Add new element and select data visualization then new visualization.

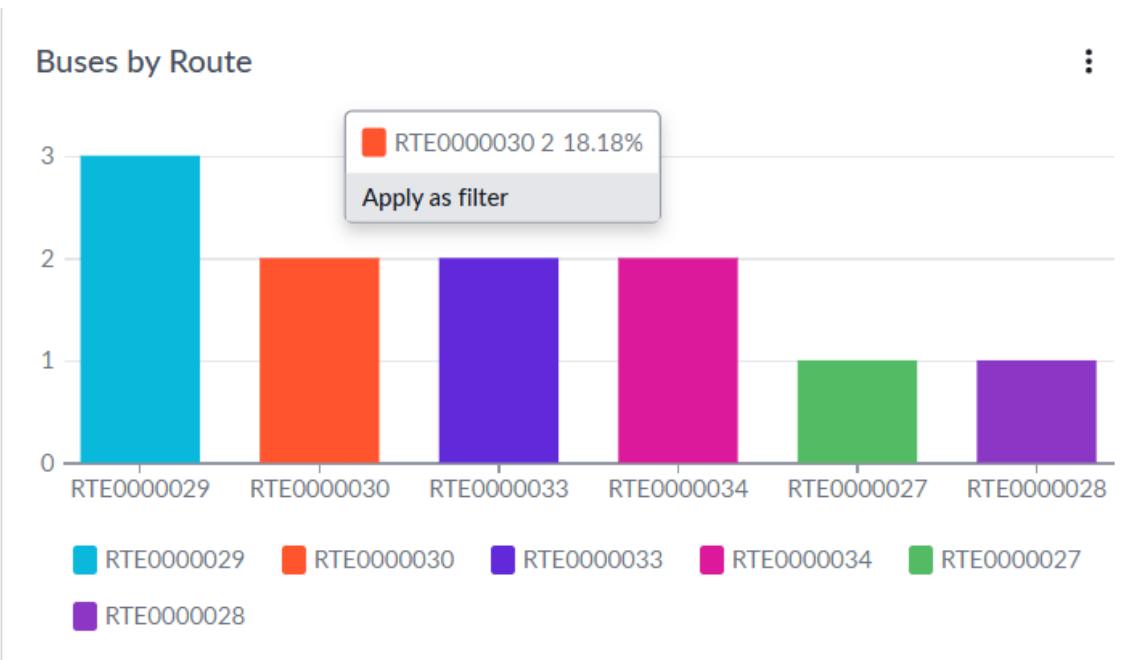
On configuration pane, Add data source as bus table.

Visualization type: Donut

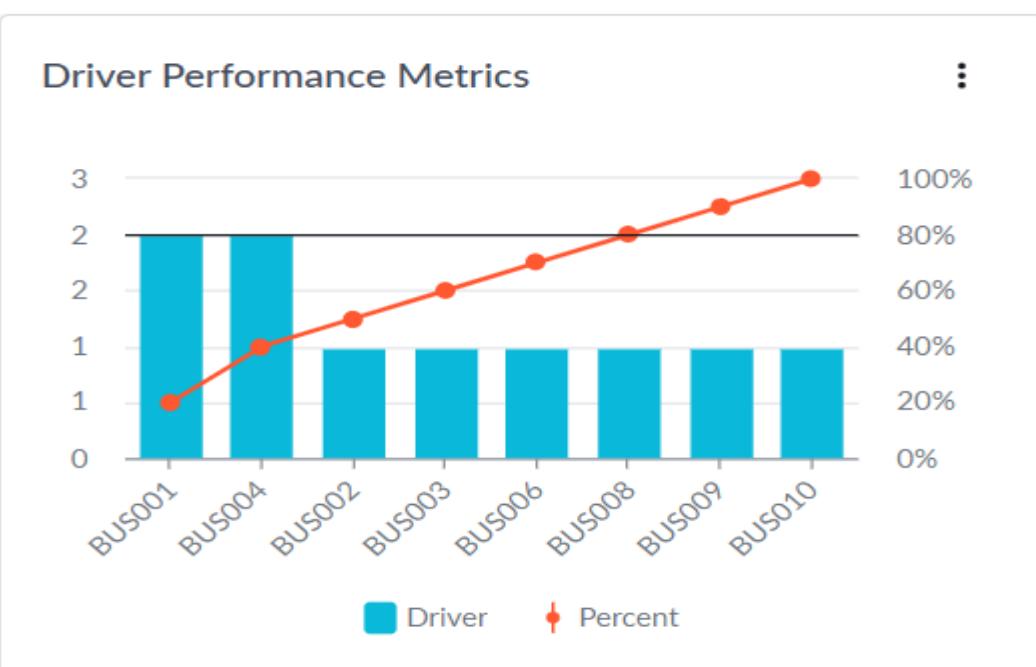
Chart title: Active vs Maintenance vs Idle



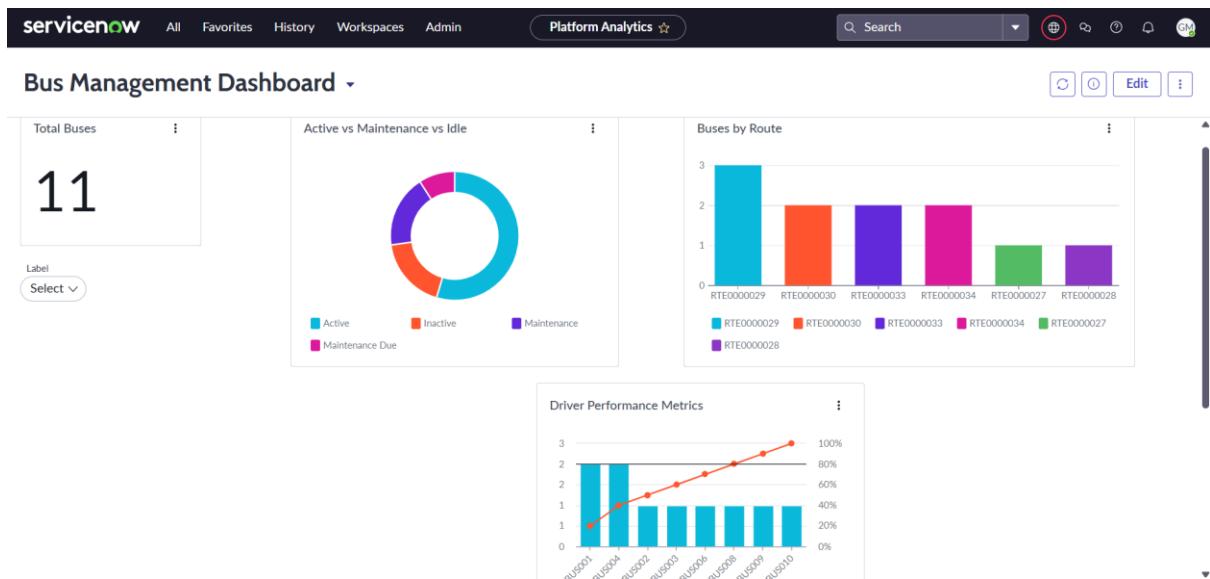
**Step-4:** Click on Add new element to add Buses by route Barchart. Proceed same as above step. Group by : Route.



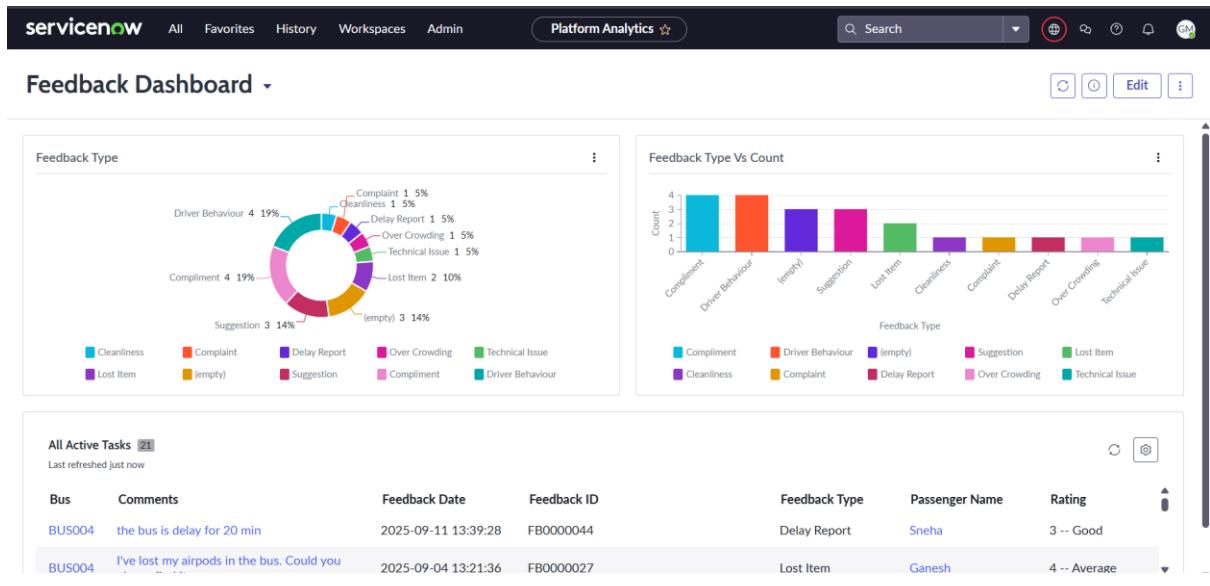
**Step-5:** Again click on add new element for Driver performance metrics using table: Drivers.



**Step-6:** Save changes.



## Activity-2: Feedback Dashboard



## Activity-3: Payments Dashboard

**servicenow** All Favorites History Workspaces Admin Platform Analytics ⚡

Search

Payments Dashboard ▾

Amount Vs Payment Status

Payment Status	Amount
Failed	£200.00
Pending	£100.00
Completed	£50.00
Refunded	£20.00

Payments Type

Payment Method	Count	Percentage
Cash	5	33%
Card	4	27%
Net Banking	3	20%
UPI	3	20%

All Active Tasks 15

Last refreshed just now

View all

Amount	Booking Passenger	Passenger	Payment Date	Payment ID	Payment Method	Payment Status	Transaction ID
£66.0025	9edc5c5e837722107f443dfb0fd1e46	Nikhil	2025-09-24 13:18:54	PAYID01048	Cash	Pending	PAYID01047
£5.0192	865d989e837722107f443dfb0fd1ec2	Mounika	2025-09-13 16:07:42	PAYID01064	Card	Completed	PAYID01063
£1.0038	4e0cd81e837722107f443dfb0fd1e0d	Priya Mehta	2025-09-02 14:28:10	PAYID01016	Net Banking	Pending	PAYID01015

## Activity-4: Routes Dashboard

Route Dashboard ▾

All Active Tasks 10

Last refreshed just now

View all

Route id	Active	Distance	End Point	Start Point
RTE0000026	true	515.3	Bengaluru	Chennai
RTE0000027	true	679.8	Hyderabad	Chennai

Destination	Count	Percentage
Bengaluru	2	18%
Chennai	1	9%
Hyderabad	2	18%
Mysuru	1	9%
Thiruvananthapuram	2	18%
Kochi	2	18%