

Smart Transport Management System

Using Service Now

Project Description:

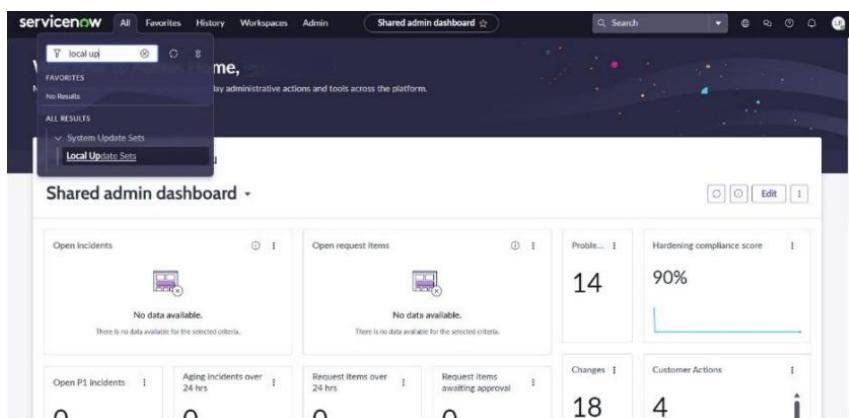
The Smart Transport Management System is built on the ServiceNow platform to streamline and automate bus service operations. It enables passengers to book and manage tickets, drivers to access schedules, and administrators to oversee buses, routes, and reports. The system leverages ServiceNow's catalog 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
ADF-Y-syntax editor macros	Global	Complete	Search	2025-08-18 18:09:57	system	(empty)	(empty)
App Store Registration 2.0	Global	Complete	Search	2025-08-18 18:22:05	system	(empty)	(empty)
CB_Book a Ticket_2025-09-05 07:05:41	Smart Transit Management System	Complete	Search	2025-09-05 07:05:41	Gangothri Madduru	(empty)	(empty)

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

* Name: Smart Transit Management System

State: In progress

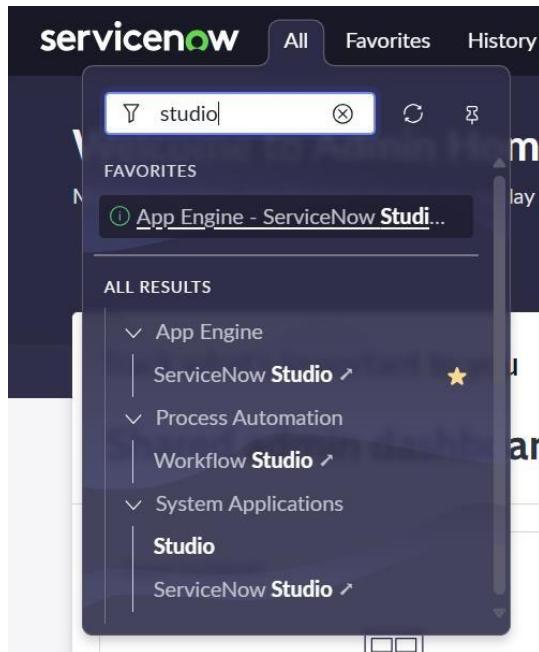
Parent: (empty)

Description: (empty)

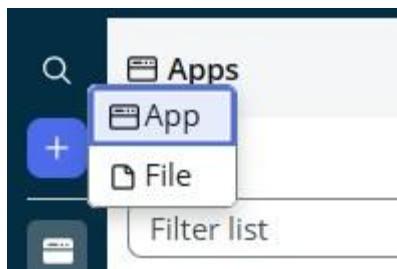
Submit | Submit and Make Current

Mile stone 2: Creating Application

1.All >> App Engine >> ServiceNow Studio



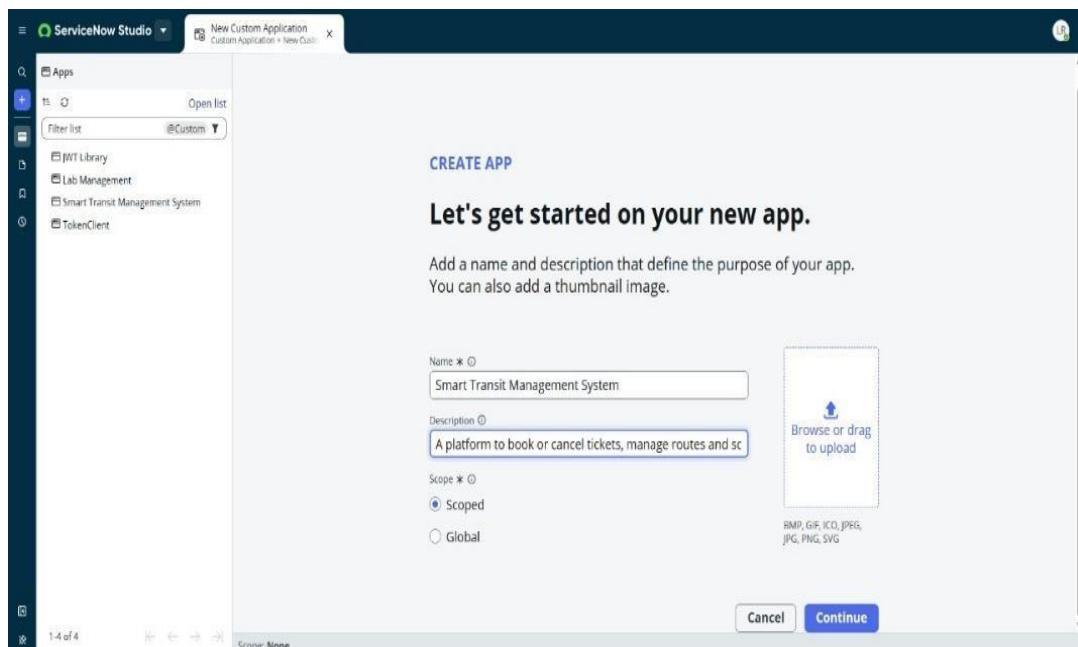
2.Select the Create icon [+] and then select App.



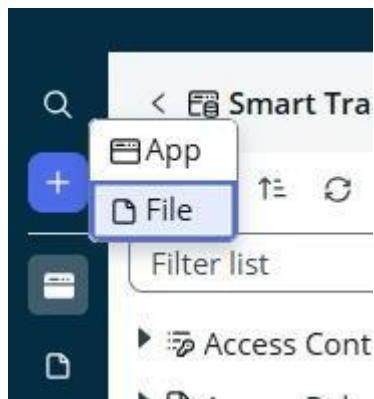
Name: Smart Transport Management System

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

3. Select continue



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

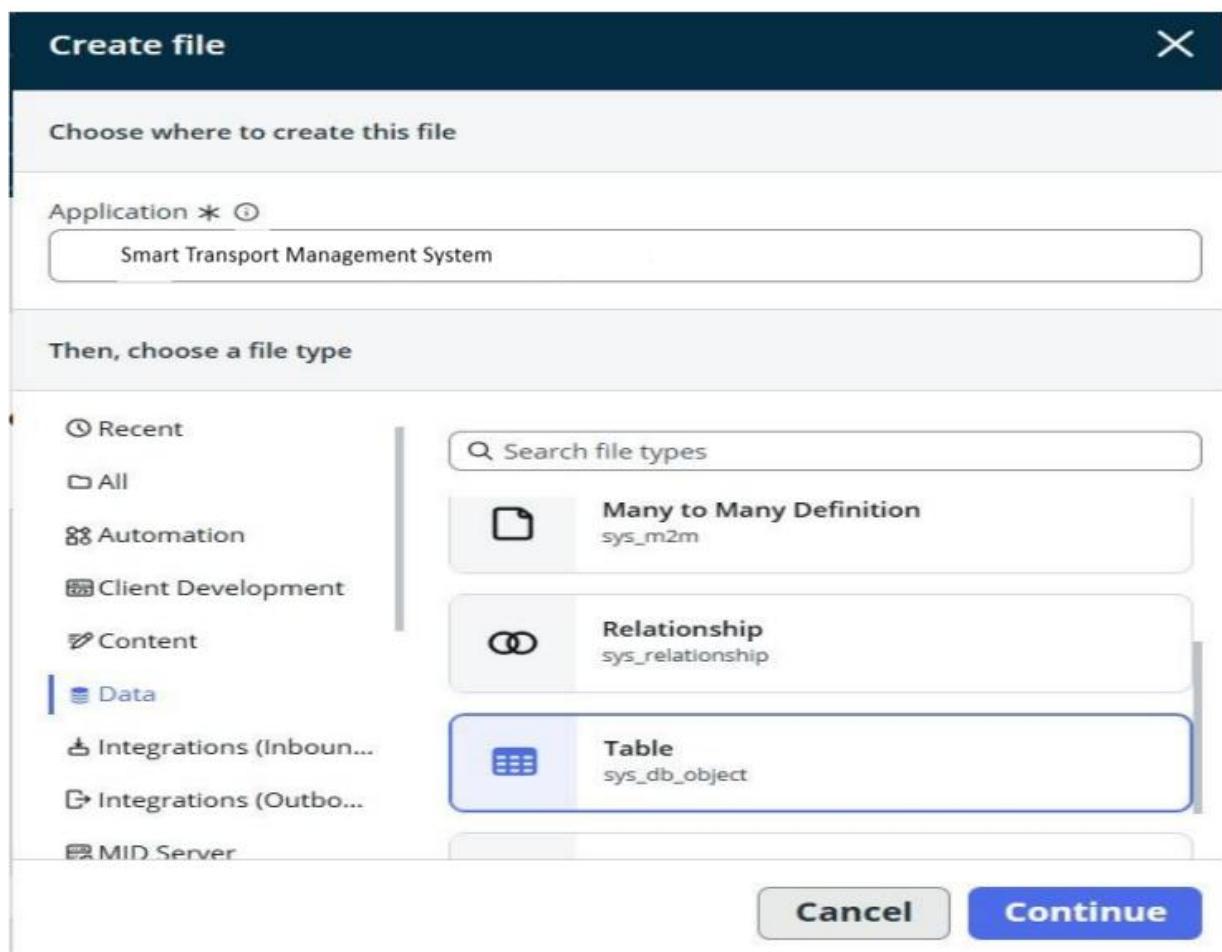


5.Specify the Application the file will belong to: *Smart Transport Management System*.

6.Select Data on the left-side panel.

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

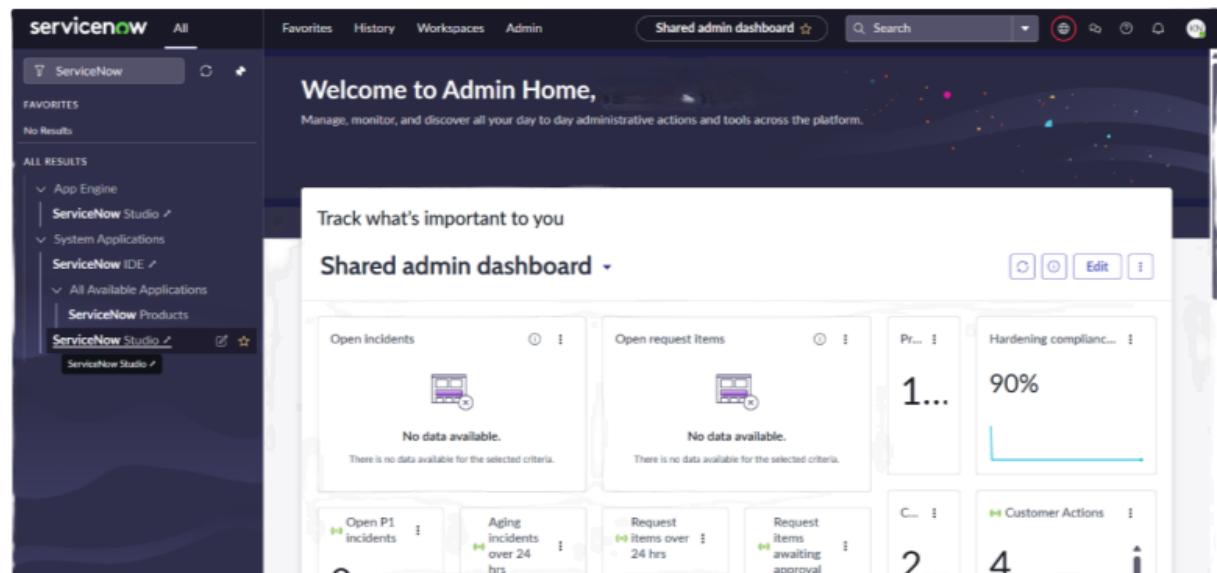
8.Select Continue.



Mile stone 3: Creating Tables

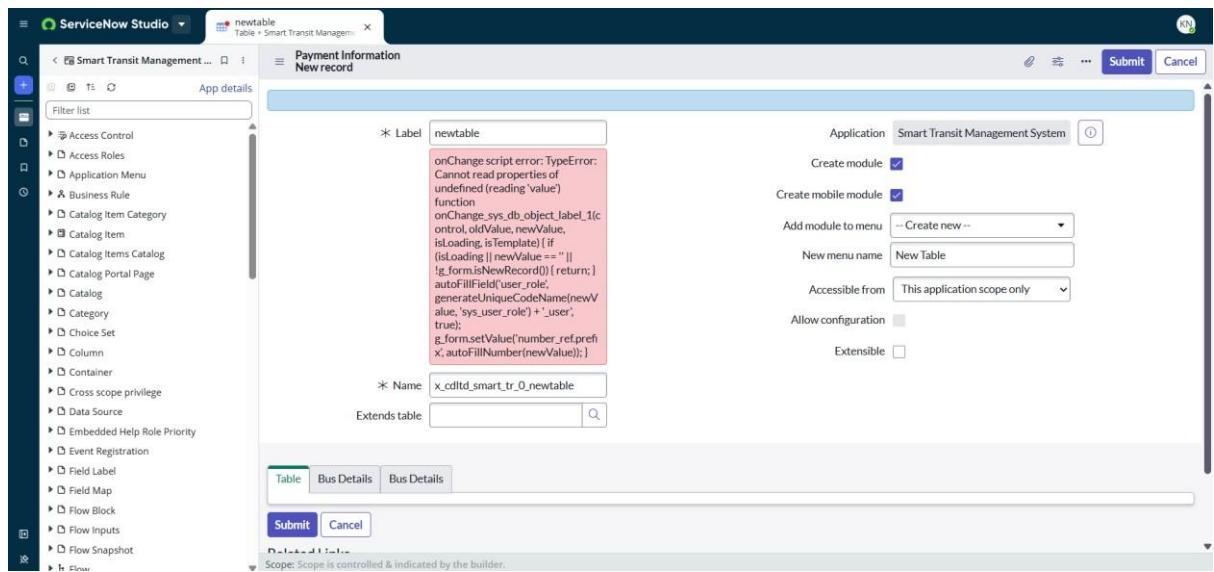
Creation of various tables in ServiceNow

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



Step 2: Click on Create (File) and Select your application scope and table(sys_db_object) and click on create.

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		20
Bus ID	bus_id	Reference	Bus			20
Created	sys_created_on	Date/Time				20
Created by	sys_created_by	String		40		20
Date	date	Date				20
Passenger Name	passenger_number	Reference	Passenger			20
Seat Number	seat_number	Integer				20
Status	status	Choice <small>3 Choices</small>				20
Time	time	Time				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.

The screenshot shows the ServiceNow interface for the Bookings table. The left sidebar has a 'Smart' search bar and a 'Favorites' section with 'No Results'. Under 'ALL RESULTS', there's a 'Smart Transit Management' category 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 of 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

The screenshot shows the ServiceNow interface for the Bus table. The left sidebar has a 'Smart' search bar and a 'Favorites' section with 'No Results'. Under 'ALL RESULTS', there's a 'Smart Transit Management' category 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 of buses:

All > Status = Active	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

The screenshot shows the GPS Trackings table in ServiceNow. The table has columns: Bus ID, Latitude, Longitude, Speed, Status, Timestamp, and Tracking ID. The data shows multiple buses (BUS001 to BUS005) with their current coordinates, speed, and status (In Transit or Maintenance). The timestamp indicates the data was recorded between September 4, 2025, and September 5, 2025.

	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
	BUS004	8.52	76.88	45	In Transit	2025-09-04 08:15:06	GPS0004
	BUS004	8.53	76.89	48	In Transit	2025-09-04 09:00:17	GPS0005
	BUS004	8.54	76.9	47	In Transit	2025-09-04 09:10:22	GPS0006
	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	GP0016
	BUS002	12.97	78.88	0	Inactive	2025-09-04 04:06:08	GP0017

Table 4: Payments

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

The screenshot shows the Payments table in ServiceNow. The table has columns: Amount, Booking Passenger, Passengers, Payment Date, Payment ID, Payment Method, Payment Status, and Transaction ID. The data shows various transactions for passengers like Rohan Sharma, Nikhil, Gangothri, Divya Kapoor, Ganesh, Sneha, and Nikhil, involving different payment methods like Cash, UPI, Net Banking, and Card, with statuses ranging from Pending to Completed.

	Amount	Booking Passenger	Passenger	Payment Date	Payment ID	Payment Method	Payment Status	Transac
	\$89.2226	9edc5c5e837722107f443dfb0fd1e46	Nikhil	2025-09-24 13:18:54	PAYID01048	Cash	Pending	PAYIDC
	\$13.9092	bb7c585e837722107f443dfb0fd1e57	Rohan Sharma	2025-09-24 07:26:33	PAYID01036	Cash	Completed	PAYIDC
	\$5.0887	baacd85e837722107f443dfb0fd1e31	Nikhil	2025-09-02 14:28:10	PAYID01034	UPI	Pending	PAYIDC
	\$1.357	bb7c585e837722107f443dfb0fd1e57	Gangothri	2025-09-25 07:25:29	PAYID01032	Net Banking	Refunded	PAYIDC
	\$101.4356	977d989e837722107f443dfb0fd1ecb	Mounika	2025-10-03 07:24:18	PAYID01030	Card	Failed	PAYIDC
	\$5.1566	a2db941e837722107f443dfb0fd1ed4	Divya Kapoor	2025-10-07 07:23:42	PAYID01028	Cash	Failed	PAYIDC
	\$64.1182	865d989e837722107f443dfb0fd1ec2	Gangothri	2025-10-01 07:23:00	PAYID01026	UPI	Failed	PAYIDC
	\$152.0969	8ffc509e837722107f443dfb0fd1e57	Ganesh	2025-09-27 07:22:31	PAYID01024	Net Banking	Pending	PAYIDC
	\$89.2226	baacd85e837722107f443dfb0fd1e31	Sneha	2025-09-02 14:28:10	PAYID01022	Cash	Failed	PAYIDC
	\$64.1182	bb7c585e837722107f443dfb0fd1e57	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 the ServiceNow interface with the 'Prices' table selected. The table has columns: Distance, End Point, Price, and Start Point. The data includes:

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 the ServiceNow interface with the 'Route Tables' table selected. The table has columns: Route Id, Active, Distance, End Point, Start Point, and Stops. The data includes:

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, Visaka
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.

This screenshot shows the ServiceNow interface for the 'Feedback_News' table. The left sidebar has a 'smart' search bar and a list of bus-related navigation links. The main area displays a table of feedback records.

Feedback_News Table Data:

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 13:39:52	Lost Item	4 - Average	I've lost my airpods in the bus.

Table 10: Passenger

This table displays the details of the passenger

This screenshot shows the ServiceNow interface for the 'Passengers' table. The left sidebar has a 'smart' search bar and a list of passenger-related navigation links. The main area displays a table of passenger records.

Passenger Table Data:

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	naveeddeep61@gmail.com	Navadeep	Male	Passenger Navadeep 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	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 40000

Table 11: Drivers

This table deals with the contact details of the bus driver

The screenshot shows the ServiceNow interface with a sidebar on the left containing various navigation links such as Bus Assignment, Create New Bus, 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 content area is titled 'Drivers' and displays a table with columns: Driver ID, Name, Assigned Bus, and Contact. The table contains 10 rows of data.

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 78765 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

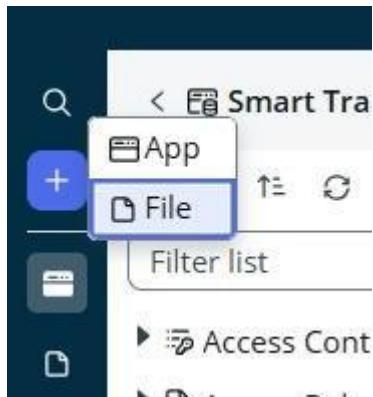
This table deals with booking of bus for a particular bus

The screenshot shows the ServiceNow interface with a sidebar on the left containing various navigation links such as Bus Assignment, Create New Bus, 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 content area is titled 'Drivers' and displays a table with columns: Driver ID, Name, Assigned Bus, and Contact. The table contains 10 rows of data.

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 78765 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 Transport Management System.*

3.Select Data on the left-side panel.

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

5.Select Continue.

Select table name: Route Table

6.Select Open

Open a table to start building a new form.

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

What is Form Builder?

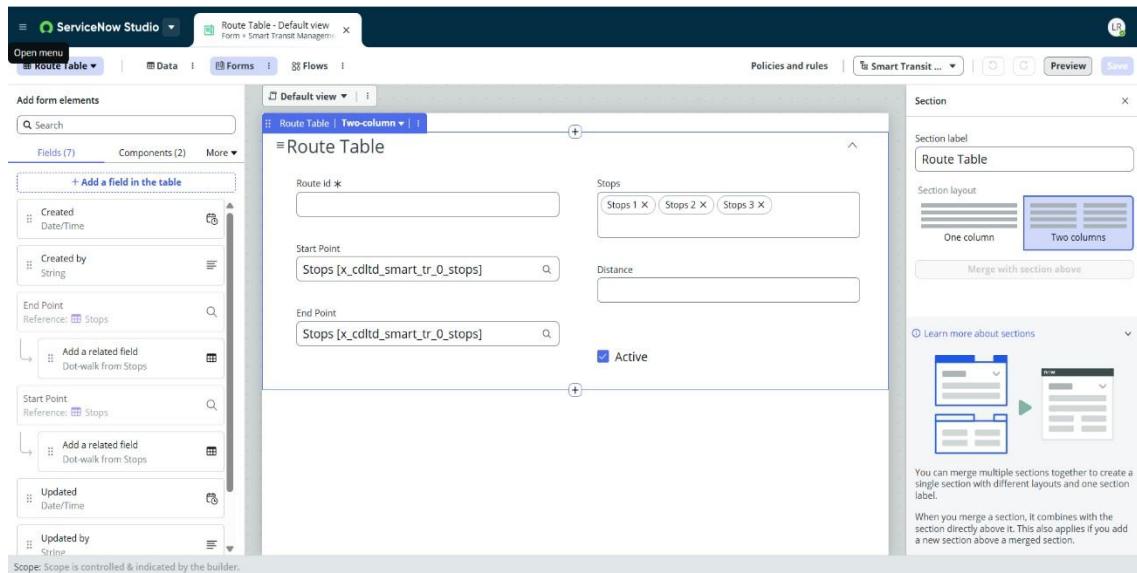
Form Builder is a drag-and-drop builder tool that helps you create custom forms for showing current data and collecting information.

- Create and customize form views**
Form views determine how your form looks. You can customize each form view's appearance by adding different form elements and choosing section layouts.
- Configure field properties**
Each field on a form view has specific field properties that you can modify, including the label, choices that appear, and the state of a field.
- Set up form logic using UI Policies**
Fields can also have UI Policies, which change whether a field is read-only or mandatory based on specific conditions.

These are only a few things you can do with Form Builder. To learn more, click the link below.

7.Add the field required fields in the form.

8.Select save.



Similarly, create form for other tables:

1.Bookings[x_cdltd_smart_tr_0_bookings]

2.Bus[x_cdltd_smart_tr_0_bus]

3.Driver[x_cdltd_smart_tr_0_driver]

4.Feedback_new[x_cdltd_smart_tr_0_feedback_new]

5.GPS Tracking[x_cdltd_smart_tr_0_gps_tracking]

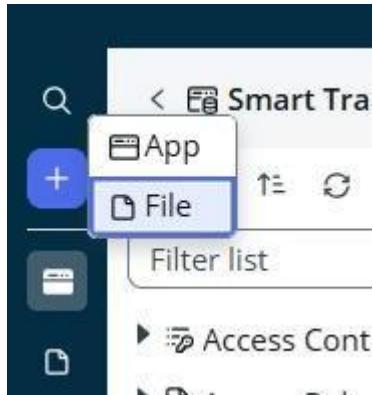
6.Passenger[x_cdltd_smart_tr_0_passenger]

7.Payment[x_cdltd_smart_tr_0_payment]

8.Price[x_cdltd_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 Transport Management System*.

3. Select Data on the left-side panel.

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

5. Select Continue.

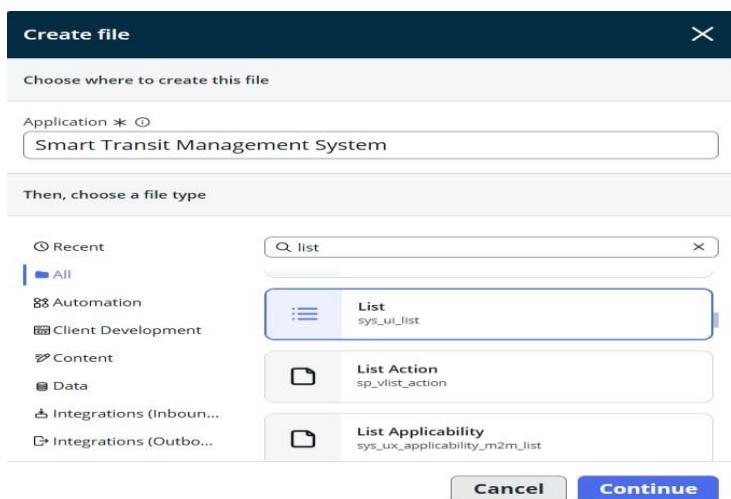


Table: Route Table[x_cdl_d_smart_tr_0_route_table]

Application: Smart Transport Management System .

6.Select Submit.

The screenshot shows the ServiceNow Studio interface with the following details:

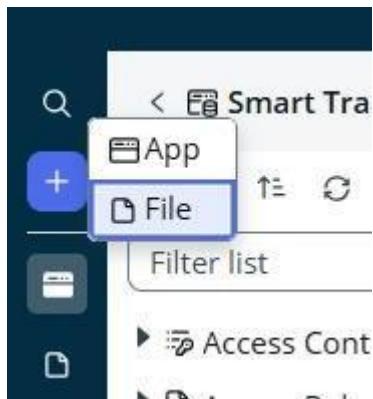
- Top navigation bar: ServiceNow Studio, Route Table - Default view, x_cdlt_d_smart_tr_0_route
- Left sidebar: List, New record
- Table selection: Route Table [x_cdlt_d_smart_tr_0_route]
- Parent field: A dropdown menu currently showing a single item.
- View field: Default view
- Application: Smart Transit Management System
- User: A dropdown menu with a search icon.
- Relationship: A dropdown menu with a search icon.
- Bottom right: Submit button

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.

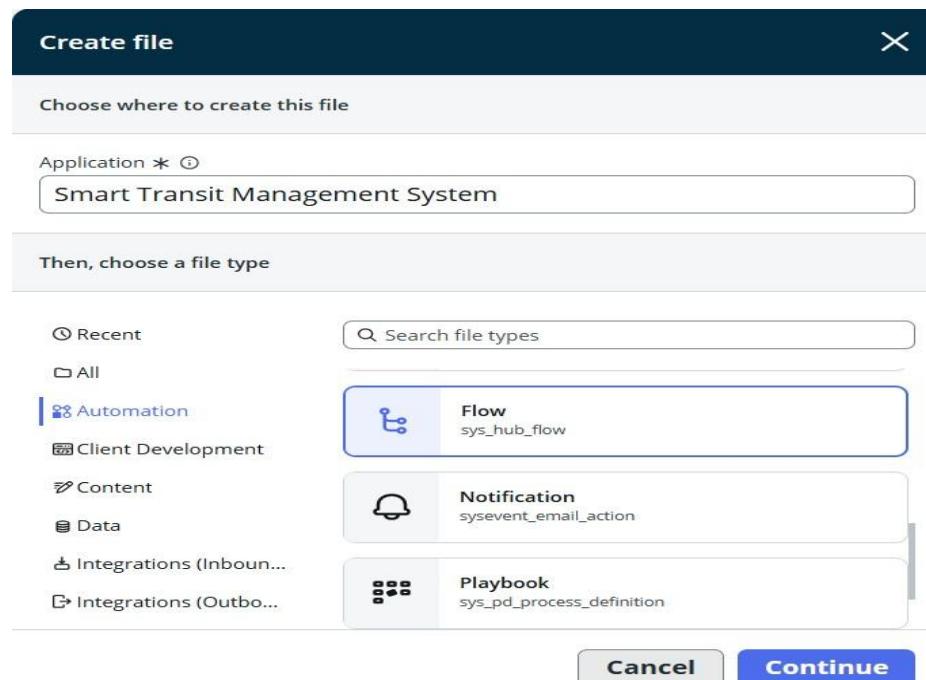


2. Specify the Application the file will belong to: *Smart Transport Management System*.

3. Select Automation on the left-side panel.

4. Scroll down and select Flow(sys_hub_flow) as the file type.

5. Select Continue.



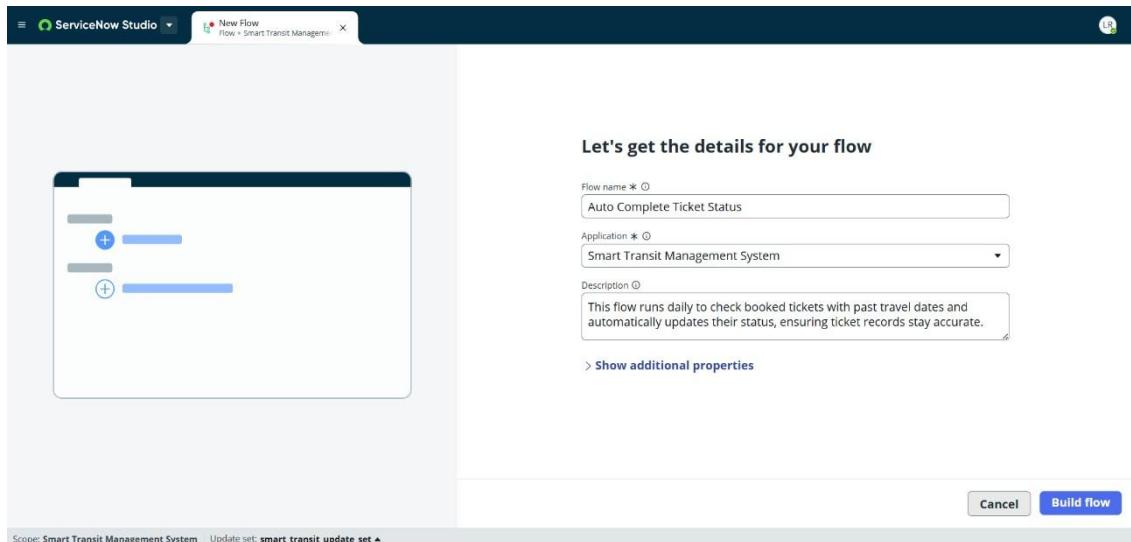
Activity 1:

Flow Name: Auto Complete Ticket Status.

Application: Smart Transport Management System.

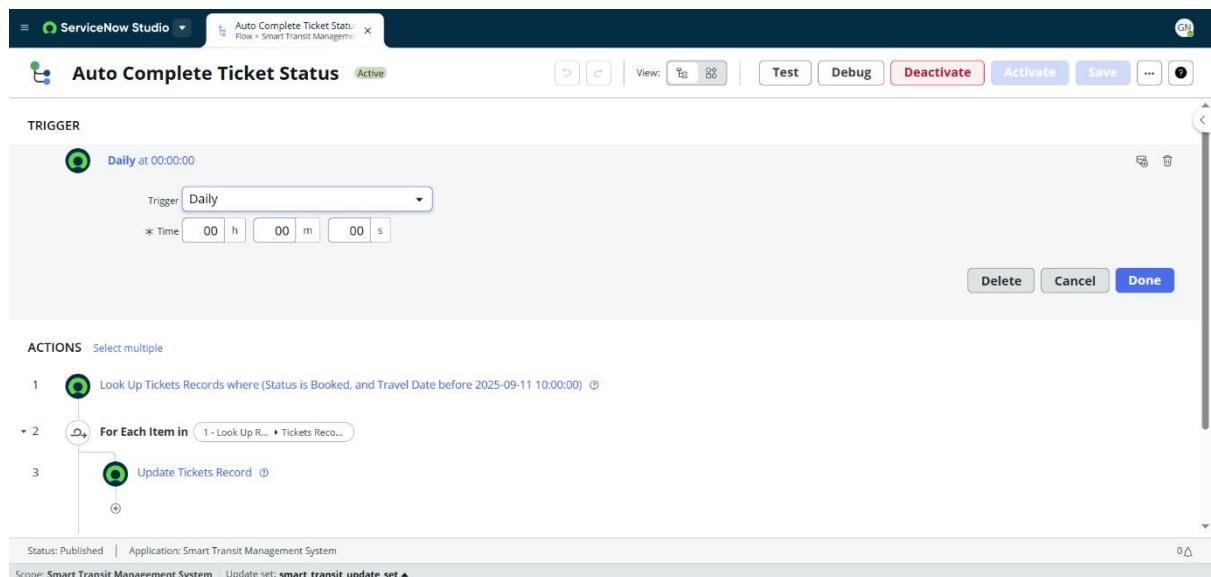
Description: This flow runs daily to check booked tickets with past travel dates and automatically updates their status, ensuring ticket records stay accurate.

Select Build Flow.



Step 1: Add Trigger

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



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 ServiceNow Studio interface with the flow titled "Auto Complete Ticket Status". Step 1 is a "Look Up Tickets Records" action. Step 2 is a "For Each Item in 1 - Look Up R... (Tickets Record)" loop. Step 3 is an "Update Tickets Record" action. The "Action Properties" for the update step show "Action: Update Record", "Record: 2 - For Each > Tickets Record", "Table: Tickets [x_cldtd_smart_tr_0_ticket]", and "Fields: Status (Completed)". The "Conditions" for the loop are set to "Status is Booked AND Travel Date is before Current hour".

Step 4: Add Action – Update Record

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

The screenshot shows the ServiceNow Studio interface with the flow titled "Auto Complete Ticket Status". Step 1 is a "Look Up Tickets Records" action. Step 2 is a "For Each Item in 1 - Look Up R... (Tickets Record)" loop. Step 3 is an "Update Tickets Record" action. The "Action Properties" for the update step show "Action: Update Record", "Record: 2 - For Each > Tickets Record", "Table: Tickets [x_cldtd_smart_tr_0_ticket]", and "Fields: Status (Completed)".

Step 5: Save and Activate Flow

- Click Save → then Activate.

Activity 2: Auto fill Ticket Amount

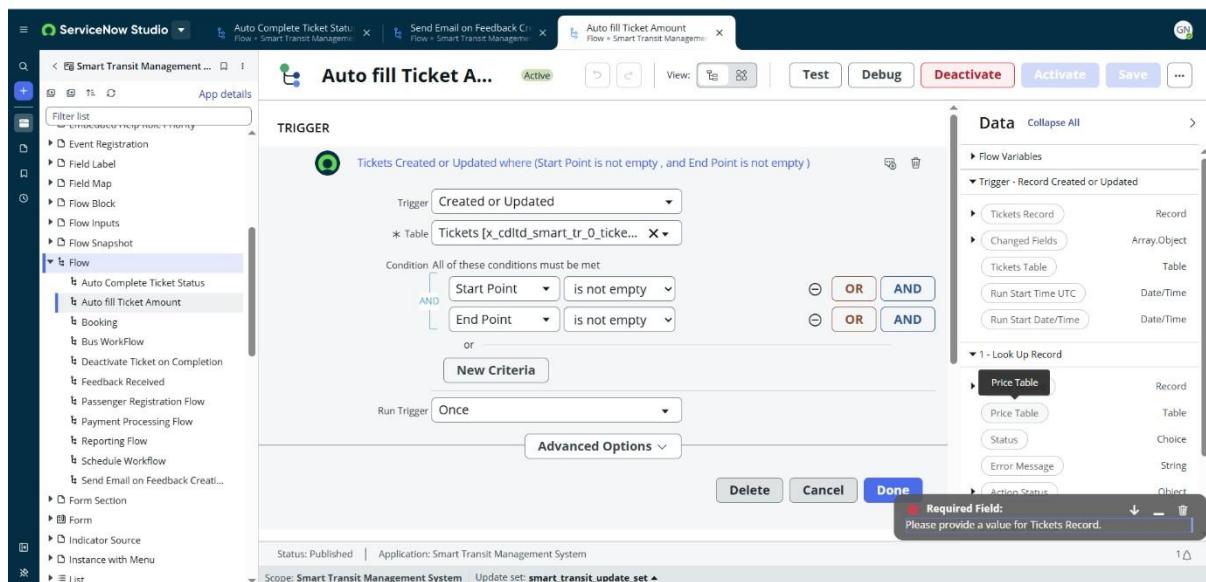
Flow Name: Auto fill Ticket Amount

Application: Smart Transport 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 Transport Management System

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

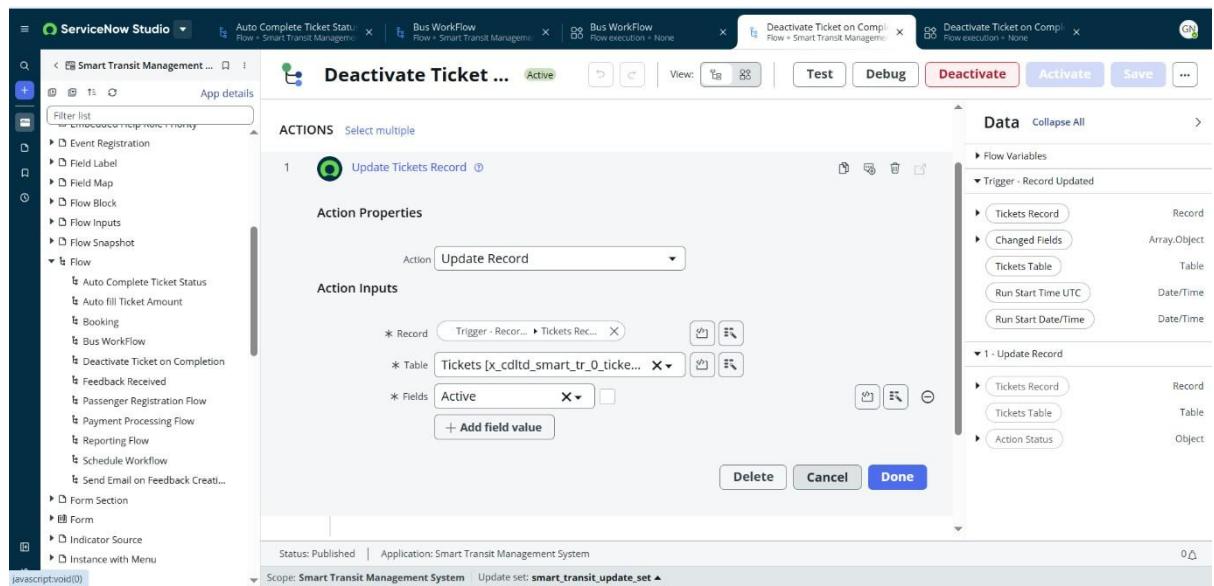
Step 1: Add Trigger

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

The screenshot shows the ServiceNow Studio interface with the 'Deactivate Ticket ...' trigger configuration. The trigger is set to 'Tickets Updated where (Status is Completed)'. The condition is 'Status is Completed'. The run trigger is set to 'Once'. A required field error message is displayed: 'Please provide a value for Tickets Record.' The right sidebar shows available actions: '1 - Update Record' (Ticket Record, Changed Fields, Tickets Table, Run Start Time UTC, Run Start Date/Time), 'Tickets Record' (Record), 'Tickets Table' (Table), and 'Action Status' (Object). The bottom status bar indicates 'Status: Published | Application: 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.



Step 3: Save and Activate Flow

- Click Save → then Activate.

Activity 4: Passenger Registration Flow

Flow Name: Passenger Registration Flow

Application: Smart Transport 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.*

The screenshot shows the ServiceNow Studio interface for configuring a flow. The flow is titled "Passenger Registration Flow". The "TRIGGER" section shows a "Passenger Created" trigger with a "Created" condition on the "Passenger" table. The "ACTIONS" section contains two steps: "Update Passenger Record" and "Send Email". The "Update Passenger Record" step is currently selected, showing its "Action Properties" and "Action Inputs". The "Action Properties" dropdown is set to "Update Record". The "Action Inputs" section shows the "Record" input set to "Trigger - Record" and the "Table" input set to "Passenger". The "Fields" input is set to "Active". The right sidebar displays various data objects and their types, such as "Passenger Record" (Record), "Passenger Table" (Table), "Run Start Time UTC" (Date/Time), and "Run Start Date/Time" (Date/Time).

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 flow is titled "Passenger Registration Flow". The "ACTIONS" section shows the "Update Passenger Record" step selected. The "Action Properties" dropdown is set to "Update Record". The "Action Inputs" section shows the "Record" input set to "Trigger - Record" and the "Table" input set to "Passenger". The "Fields" input is set to "Active". The right sidebar displays various data objects and their types, such as "Passenger Record" (Record), "Passenger Table" (Table), "Run Start Time UTC" (Date/Time), and "Run Start Date/Time" (Date/Time).

Step 3: Add Action – Send Email

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

Action Properties

Action: Send Email

Action Inputs

- Target Record: Trigger - Record Created → Passenger Record
- Table: Passenger [x_cdlt_smart_tr_0_p...]
- Include Watermark: checked
- * To: Trigger - Record Created → Email
- CC:
- BCC:
- * Subject: Trigger - Record Created → Full Name
- Body: Hello, Trigger - Record Created → Full Name.

Data

- Trigger - Record Created
- 1 - Update Record
- 2 - Send Email

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.

Action Properties

Action: Update Record

Action Inputs

- * Record: Trigger - Record Created → Passenger Record
- * Table: Passenger [x_cdlt_smart_tr_0_p...]
- * Fields: Notes, Passenger
- + Add field value

Step 5: Save and Activate Flow

- Click Save → then Activate.

Activity 5: Reporting Flow

Flow Name: Reporting Flow

Application: Smart Transport 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.

The screenshot shows the ServiceNow Studio interface for creating a 'Reporting Flow'. The left sidebar lists various flow types, and the main area shows the configuration for a 'Reporting Flow' named 'Reporting Flow'. The 'TRIGGER' section contains a single entry: 'Daily at 06:00:00'. The 'ACTIONS' section contains one action: 'Look Up Route Table Records'. The 'Action Properties' panel shows the action is set to 'Look Up Records' and the 'Table' is 'Route Table [x_cdtd_smart_tr_0...]'.

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

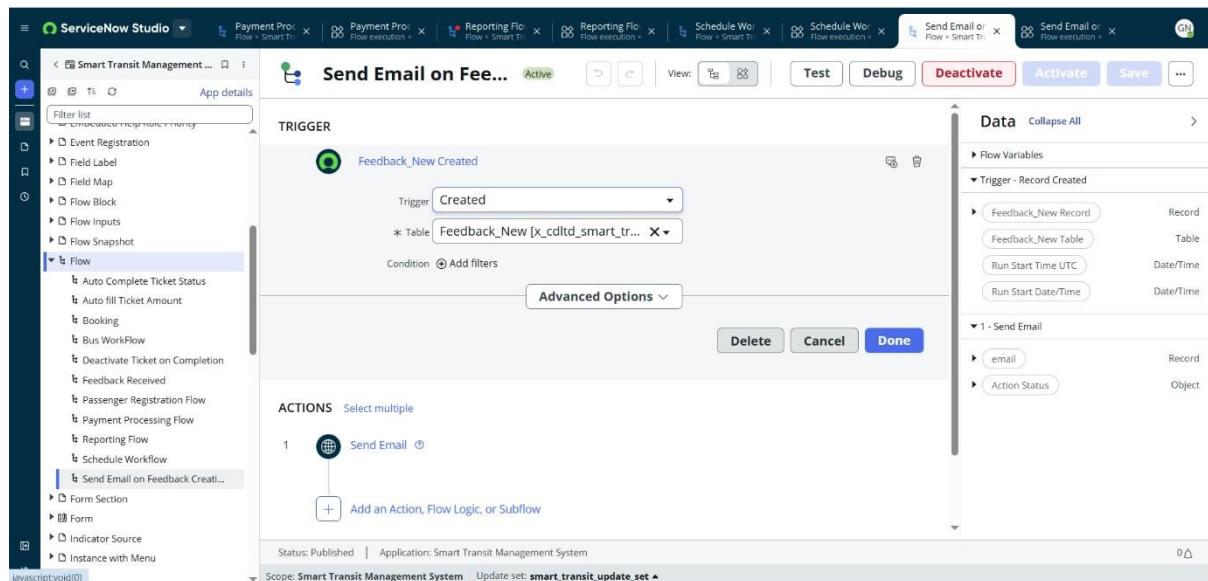
Application: Smart Transport Management System

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.

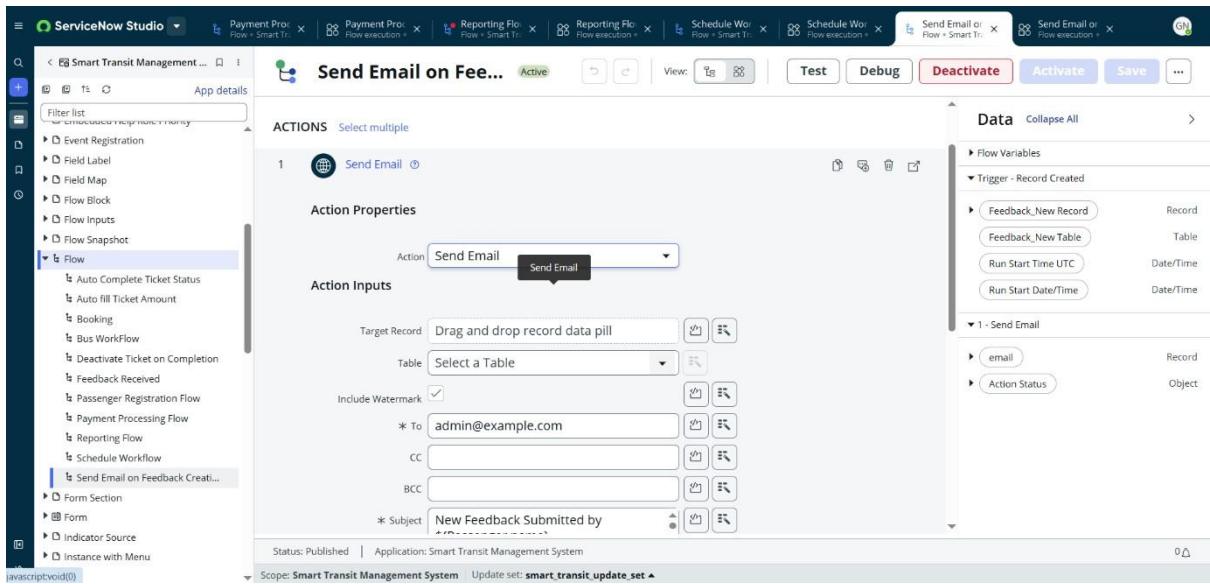
- Select Record Created.
- Condition: *Feedback Created*.



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.



Step 3: Save and Activate Flow

- Click Save → then Activate.

Activity 7: Payment Processing Flow

Flow Name: Payment Processing Flow

Application: Smart Transport 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.

The screenshot shows the ServiceNow Studio interface for a 'Payment Processing Flow'. The flow is active and has one step: 'Update Payment Record'. The 'Action Properties' section shows the action is set to 'Update Record'. The 'Action Inputs' section shows the record is triggered by 'Trigger - Reco...' and the table is 'Payment [x_cdlt_smart_tr_0_pa...]'.

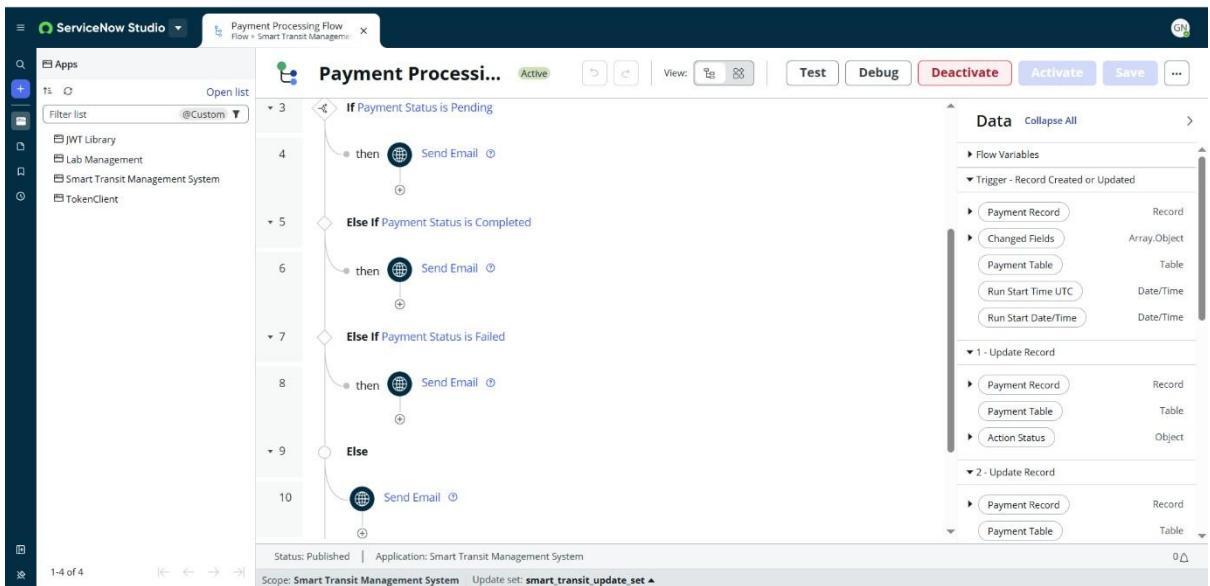
Step 3: Add Action – Update Payment Record

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

The screenshot shows the ServiceNow Studio interface for the 'Payment Processing Flow'. The flow now contains two actions: '1 - Update Record' and '2 - Update Record'. The second action is currently selected. Both actions are configured as 'Update Record' with the same trigger and table settings as the first one.

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

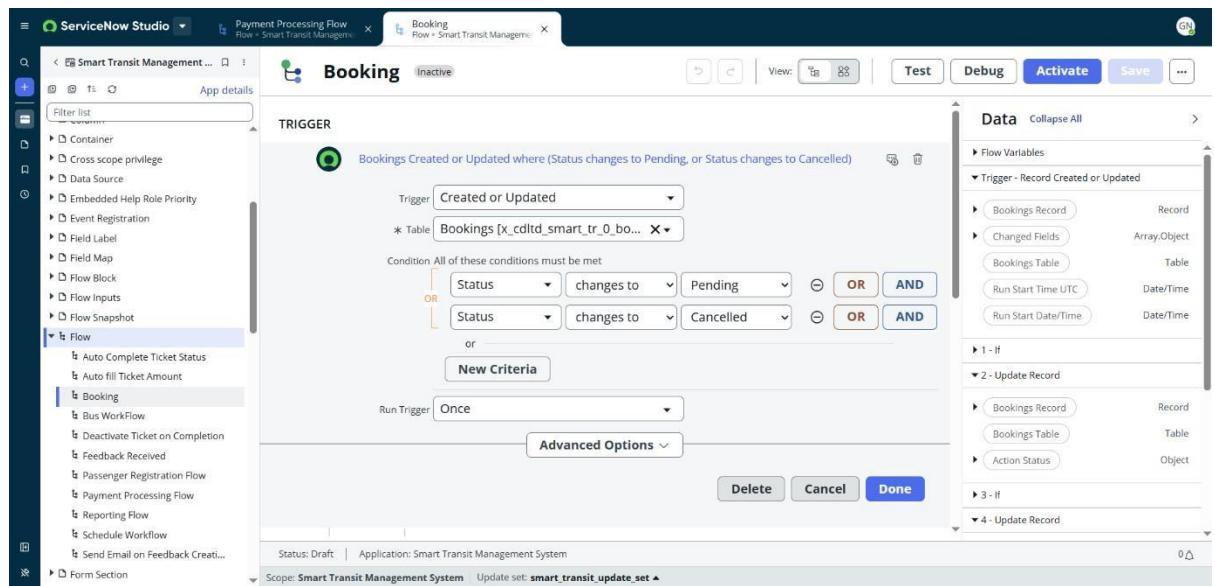
Application: Smart Transport Management System

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

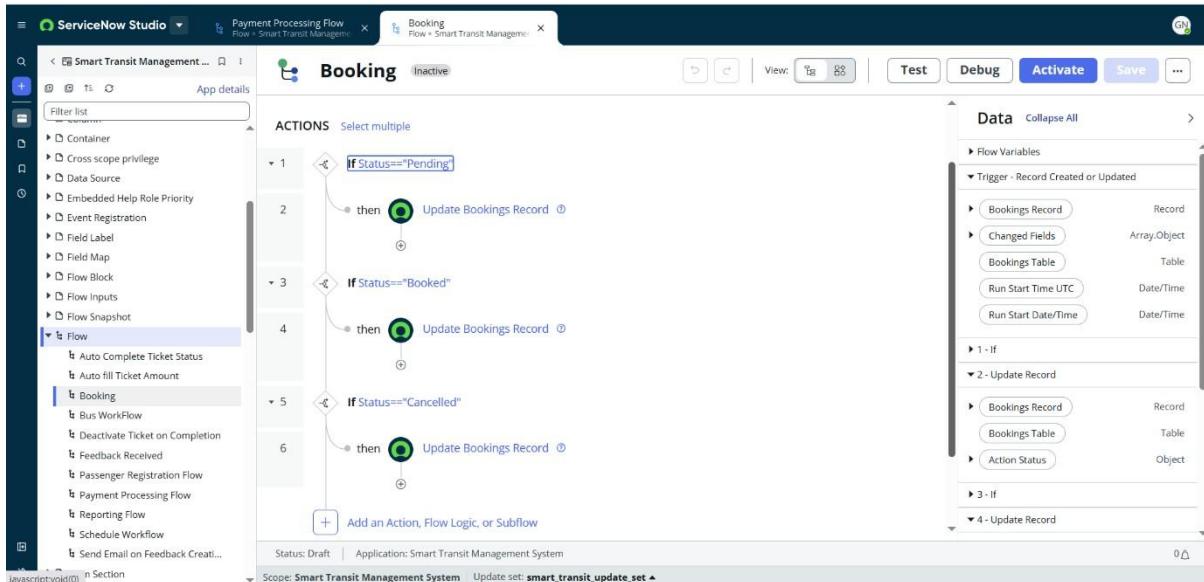
- Add a condition If Status == Booked.
- Then add Update Booking Record to confirm the booking.

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 Transport 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.



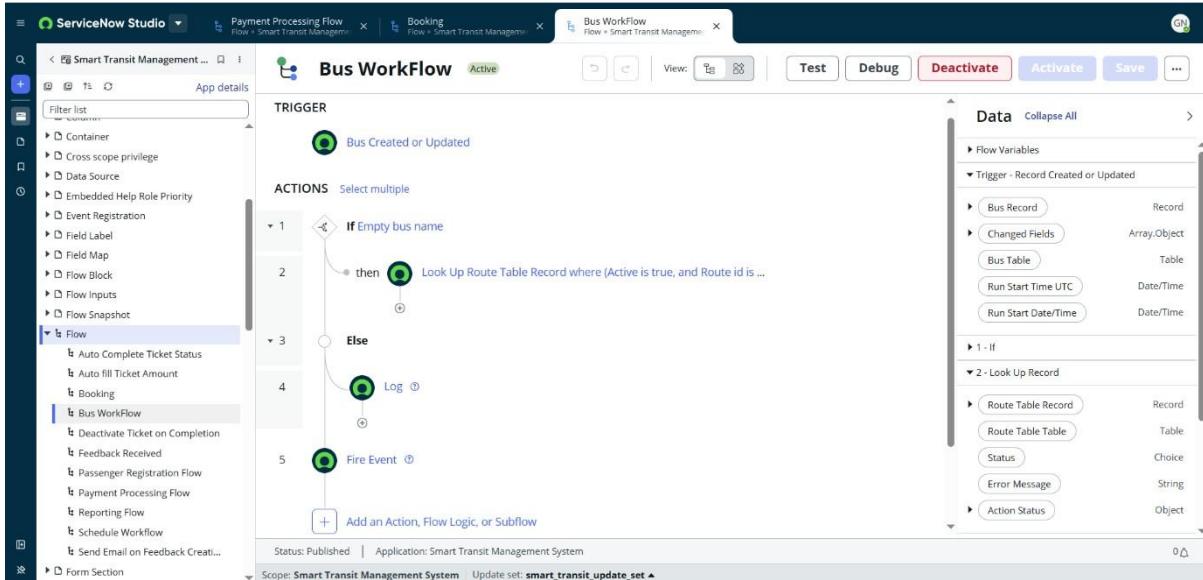
Step 3: Add Action – Look Up Route Table Record

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 Transport 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

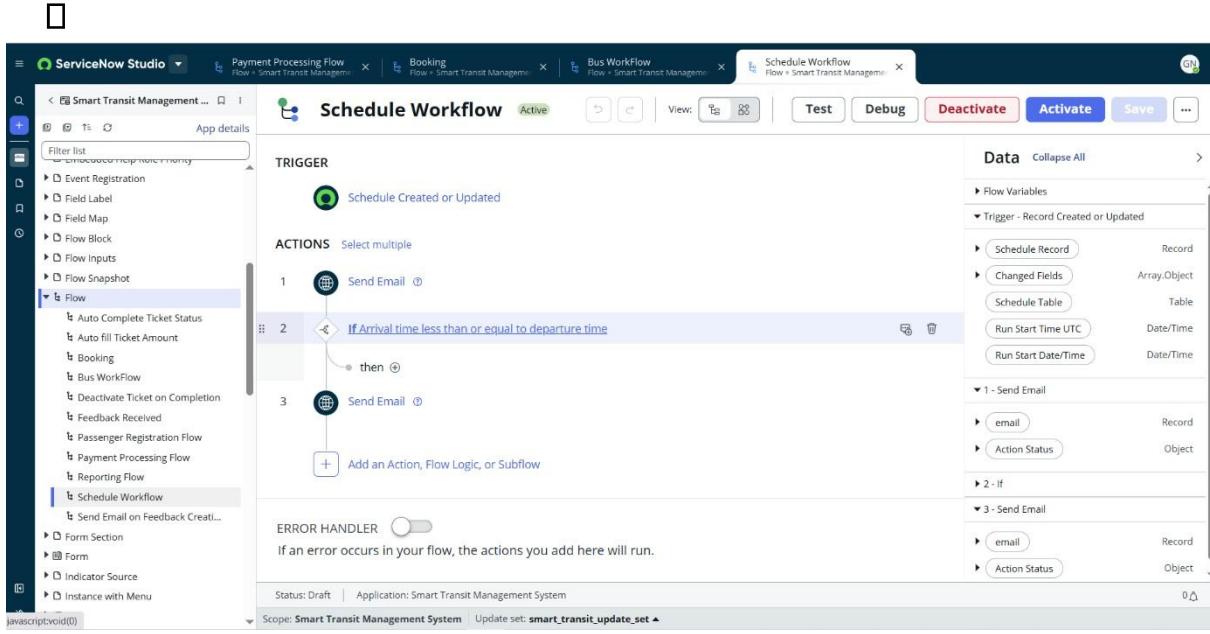
- Click on Add Trigger.
Select Record Created or Updated.
- Condition: *Schedule Record Created or Updated* **Step 2: Add Action – Send Email (Initial Notification)** □ Add a Send Email action.
- Configure it to notify admins/drivers when a new schedule is created or updated.

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.

The screenshot shows the ServiceNow Shared admin dashboard. In the top navigation bar, 'Reports' is selected under the 'Data Sources' category. The main content area displays various dashboards and reports. On the left, there's a sidebar with sections like 'Platform Analytics Administration', 'Data Sources', 'Report Sources', 'Usage and governance', and 'Reports'. Below this is a 'List of Report definition records' section with categories: 'Schedules', 'Scheduled Reports', 'Color Settings', and 'Report Ranges'. The main dashboard area has several cards: 'Open request items' (No data available), 'Problem...' (14), 'Hardening compliance score' (90%), 'Changes' (18), and 'Customer Actions' (4). At the bottom, there are links for 'Open P1 incidents', 'Aging incidents over 24 hrs', 'Request items over 24 hrs', and 'Request items awaiting approval'.

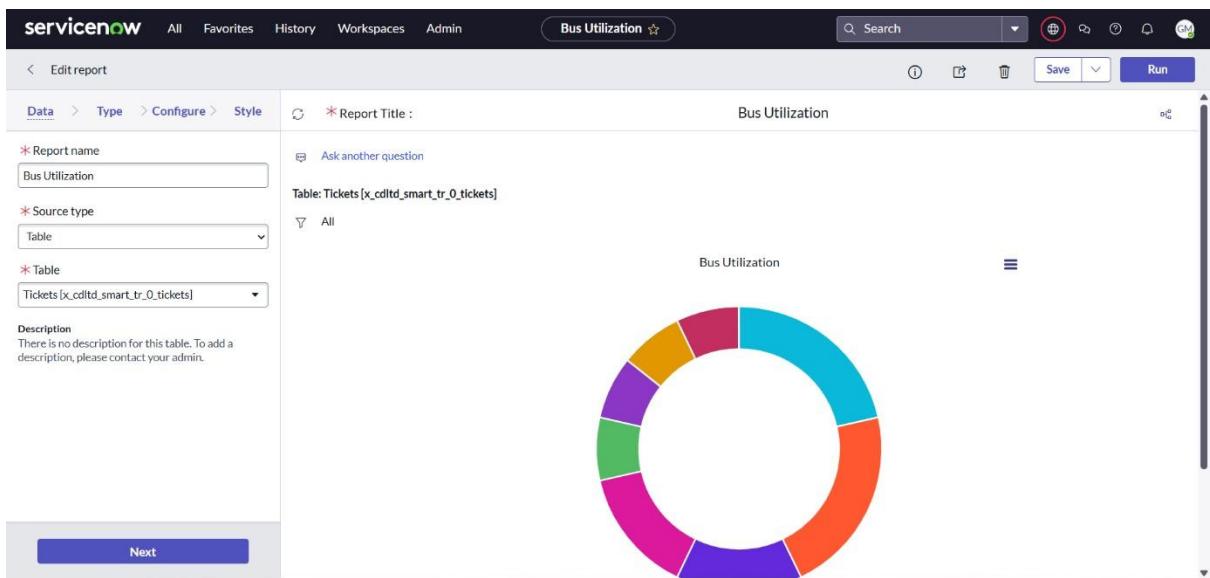
2. Click on new

The screenshot shows the 'Reports' list view in ServiceNow. The top navigation bar has 'Reports' selected. The main content area displays a table of report definitions. The columns include Title, Table, Type, Field Name, Created by, and Updated. The table contains 16 rows of data, each representing a different report definition. The last row visible in the screenshot is 'Problems By State'.

Title	Table	Type	Field Name	Created by	Updated
Active Changes > 7 days	Change Request [change_request]	Single Score		admin	2020-11-11 19:43:52
Business Services by Location	Service [cmdb_ci_service]	Pie	location	admin	2013-11-18 19:48:56
Requestor API Usage (Monthly)	API Transactions Requestor Monthly Stats [sys_api_stats_requestor_monthly]	Trend	api_name	admin	2016-07-14 18:18:43
KPI - Average Work Effort for Resolving	Incident Time Worked [incident_time_worked]	Pivot Table	inc_category	glide.maint	2011-11-30 22:35:43
Servers in an application service	label [servers_svc_assoc]	Single Score	s_sys_class_name	admin	2020-03-09 21:06:39
My Groups Work	Task [task]	List		admin	2021-07-26 22:52:15
Service View - Completeness Trend	CMDB Service Health Scorecard [cmdb_health_scorecard_service]	Line		admin	2016-05-16 18:24:29
Highest Change Activity (3m)	Change Request [change_request]	Bar	cmdb_ci	admin	2020-09-15 22:52:55
Open Incidents by Assignment	Incident [incident]	Bar	assigned_to	admin	2013-11-19 13:59:17
Problems By State	Problem [problem]	Horizontal bar	state	admin	2015-11-13 14:52:47

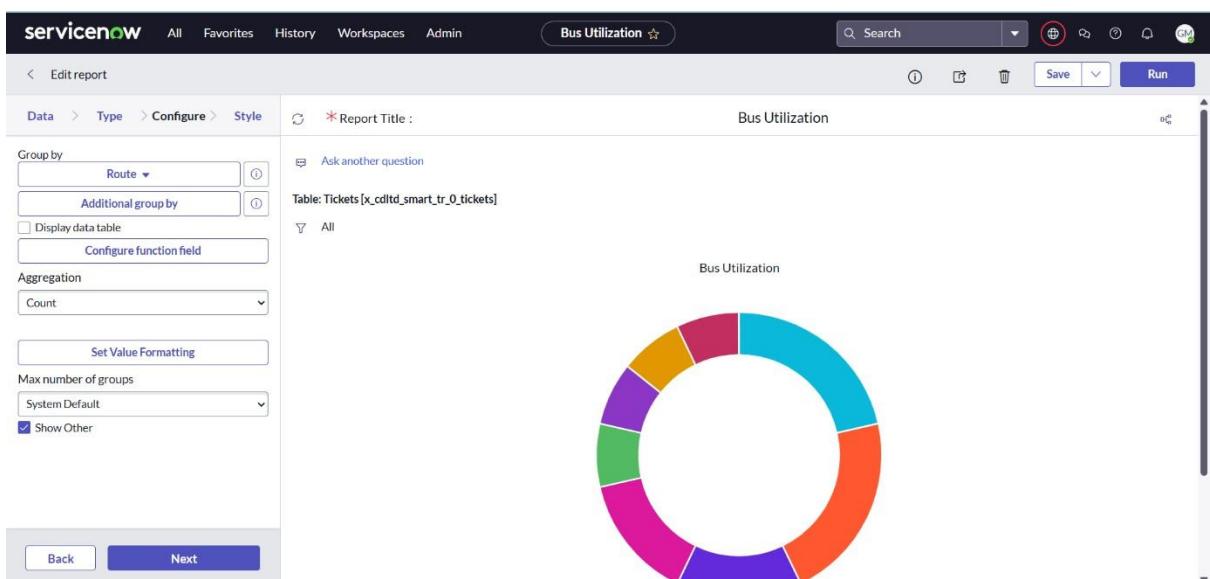
3. On Data pane, Enter following details.

- **Report Name:** Bus Utilization
- **Source Type:** Table
- **Table:** Tickets[c_cdltd_smart_tr_0_tickets]

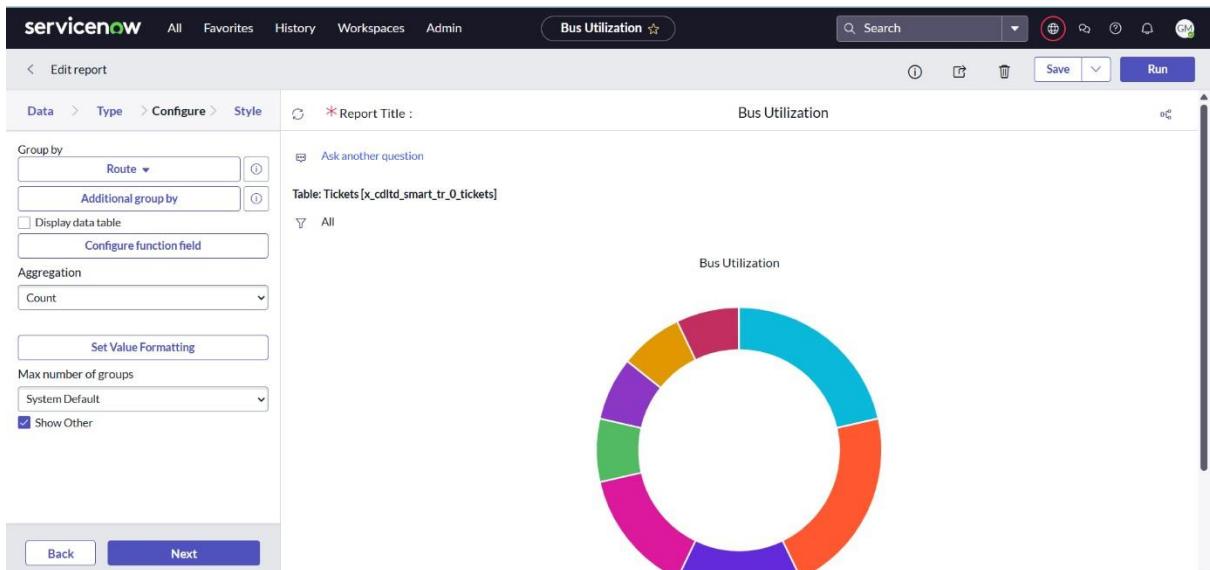


4. Click next and fill details as below in Type and configure pane.

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



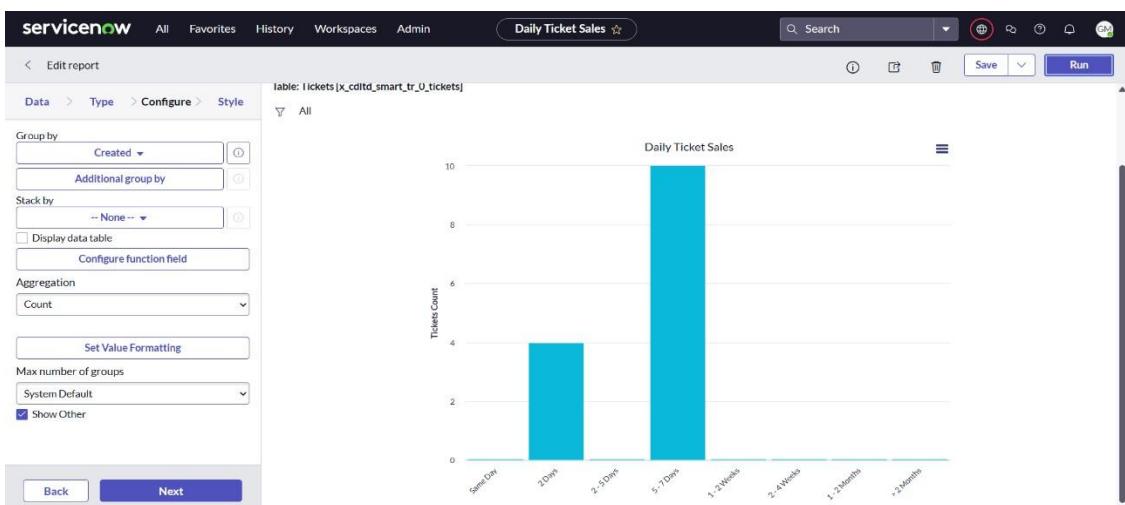
5. Save and Run it.



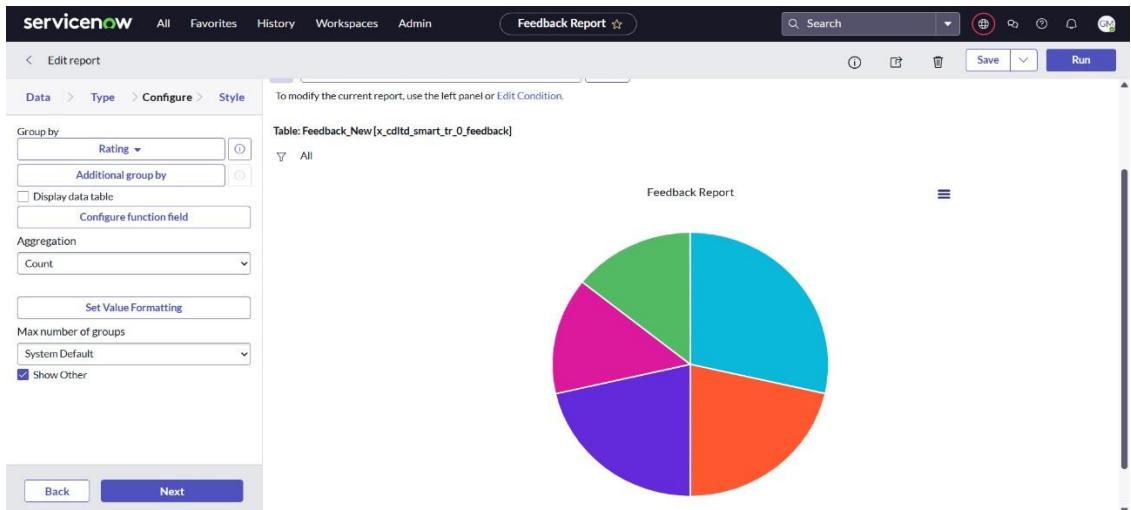
Activity 2: Create Reports for remaining

1. Repeat above steps for Daily Ticket Sales.

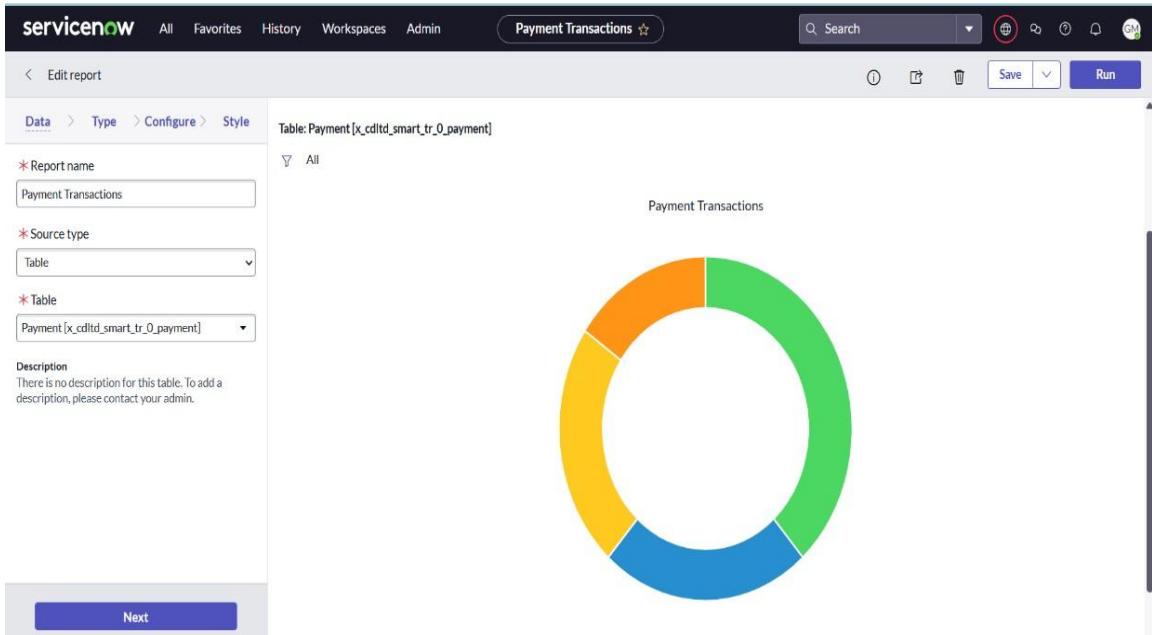
- **Name:** Daily Ticket Sales
- **Table:** Tickets
- **Type:** Bar



2. Feedback Report



3. Payment Transactions



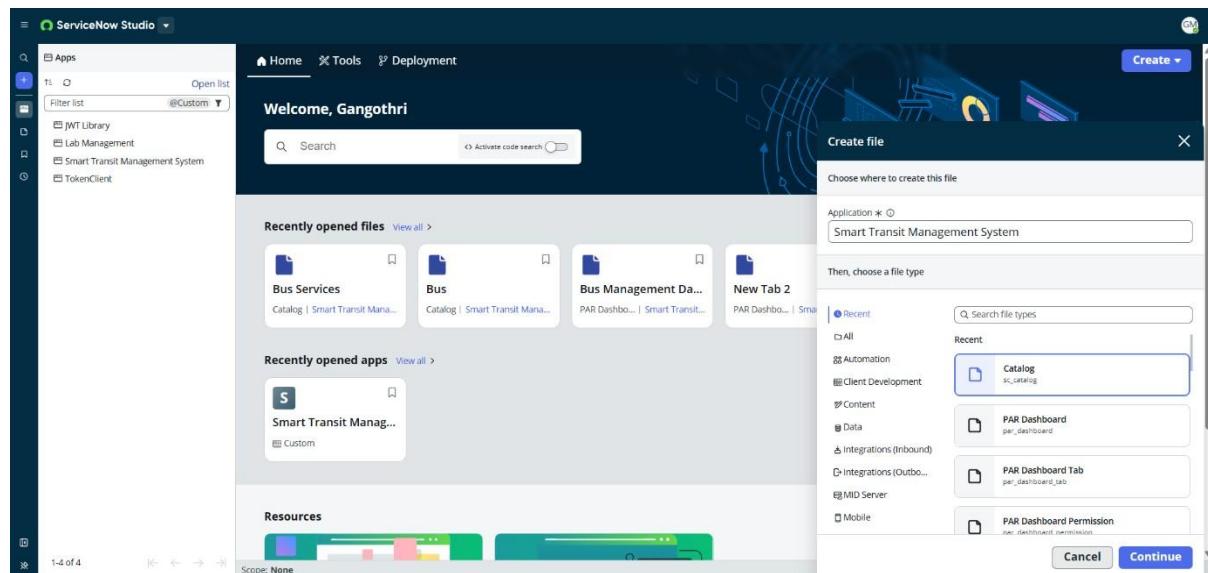
4. Top Routes by Bookings



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 Transport 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:



Activity-2: Create Categories

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

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

The screenshot shows a software interface for managing categories. At the top, there's a toolbar with icons for edit, copy, delete, and other actions. Below the toolbar, the main area has several input fields:

- Title:** Financial Services
- Catalog:** Bus Services
- Location:** (empty)
- Description:** Services related to payments and refunds, including ticket fare payments, transaction support, and refund requests for canceled bookings.
- Application:** Smart Transit Management System
- Active:** checked
- Parent:** (empty)

Below the input fields, there are links for "Desktop image" and "Header icon", each with a "Click to add..." button. There are also "Icon" and "Icon" buttons. At the bottom left, there are "Update" and "Delete" buttons. A sidebar on the left contains "Related Links" with options like "Force to Update Set", "Make available for Item Designer", and "Run Point Scan".

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

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 Transport Management System
- **Category:** General
- **Description:** Custom page to show booking summary.
- **Endpoint:** x_cdltd_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 Transport Management → Booking Summary

The screenshot shows the ServiceNow UI Page editor interface. At the top, the title bar reads "UI Page - booking_summary". The left sidebar shows the navigation path: "All" > "UI Page" > "booking_summary". The main configuration area includes fields for "Name" (booking_summary), "Category" (General), "Application" (Smart Transit Management System), "Endpoint" (x_cdltd_smart_tr_0_booking_summary.do), and a "Description" box containing "Custom page to show booking summary". Below this is an "HTML" tab with the following code:

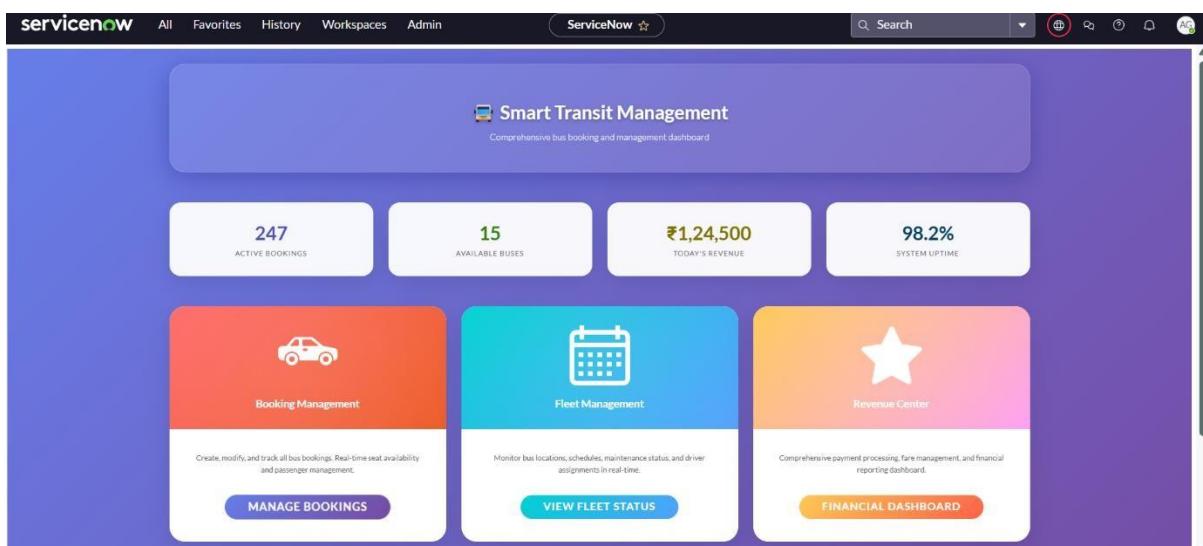
```
<?xml version="1.0" encoding="utf-8"?>
<j:jelly xmlns:j="jelly:core" xmlns:g="glide">
<g:ui_form>
<style>
.dashboard-container {
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
min-height: 100vh;
padding: 20px 0;
}

.dashboard-header {
background: rgba(255, 255, 255, 0.1);
backdrop-filter: blur(10px);
border-radius: 20px;
padding: 30px;
margin-bottom: 40px;
border: 1px solid rgba(255, 255, 255, 0.2);
box-shadow: 0 8px 32px rgba(31, 38, 135, 0.37);
}
```

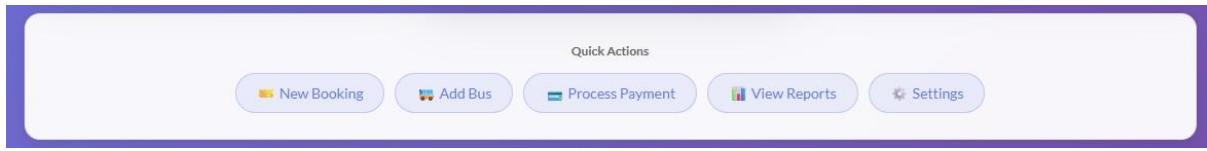
Step 4: Page Layout

The page is structured with the following components:

1. **Dashboard Header** ○ Title: *Smart Transport 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



4. **Quick Actions (Bottom Section)** ○ New Booking ○ Add Bus ○ Process Payment ○ View Reports ○ Settings



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

- **Transport_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							Actions on selected rows...	New
All	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	Gangothri	2025-09-08	07:15:00	Pending	2	
	B007	BUS009	Mounika	2025-09-07	11:30:00	Booked	7	
	B010	BUS005	Divya 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	06:05:00	Booked	3	
	B011	BUS005	Navadeep	2025-09-24	15:00:00	Booked	18	
	B005	BUS003	Nikhil	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							Actions on selected rows...	New
All	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	AVM	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									Actions on selected rows...	New
All	Amount	Booking Passenger	Passengers	Payment Date	Payment ID	Payment Method	Payment Status	Transaction ID		
	£1.0062	4e0cd81e837722107443dfb0fd0da1e0d	Priya Mehta	2025-09-02 14:28:10	PAYID01016	Net Banking	Pending	PAYID01015		
	£47.542	865998e837722107443dfb0fd0da1ec2	Gangothri	2025-10-01 07:23:00	PAYID01026	UPI	Failed	PAYID01025		
	£112.776	0fffc509e837722107443dfb0fd0da1e57	Ganesh	2025-09-27 07:23:31	PAYID01024	Net Banking	Pending	PAYID01023		
	£75.2119	977d998e837722107443dfb0fd0da1ecb	Mounika	2025-10-03 07:24:18	PAYID01030	Card	Failed	PAYID01029		
	£28.9277	7edc5ce837722107443dfb0fd0da1e46	Arjun Reddy	2025-09-02 04:55:42	PAYID01018	UPI	Refunded	PAYID01017		
	£3.8235	a2db941e837722107443dfb0fd0da1ed4	Divya Kapoor	2025-10-07 07:23:42	PAYID01028	Cash	Failed	PAYID01027		
	£66.1563	bacd85e837722107443dfb0fd0da1e31	Sneha	2025-09-02 14:28:10	PAYID01022	Cash	Failed	PAYID01021		
	£1.0062	bb7c585e837722107443dfb0fd0da1e57	Gangothri	2025-09-25 07:25:29	PAYID01032	Net Banking	Refunded	PAYID01031		
	£10.3133	bb7c585e837722107443dfb0fd0da1e57	Rohan Sharma	2025-09-24 07:26:33	PAYID01036	Cash	Completed	PAYID01035		
	£9.2233	cc4a1c9a837722107443dfb0fd0da1e7b	Rohan Sharma	2025-08-20 10:59:50	PAYID01012	Card	Completed	PAYID01011		
	£81.2762	cc3a949e837722107443dfb0fd0da1ee2	Navadeep	2025-09-02 04:55:42	PAYID01014	Cash	Failed	PAYID01013		
	£47.542	bb7c585e837722107443dfb0fd0da1e57	Nikhil	2025-09-30 07:21:05	PAYID01020	Card	Completed	PAYID01019		
	£3.7732	bacd85e837722107443dfb0fd0da1e31	Nikhil	2025-09-02 14:28:10	PAYID01034	UPI	Pending	PAYID01033		

Step 9: Testing & Validation

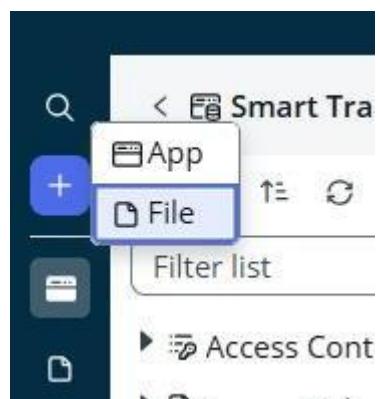
- Verified that booking count updates when new booking is added.
- Confirmed available buses update when a bus is marked inactive.
- Tested that revenue value changes after payment processing.
- Validated quick action buttons redirect correctly to respective modules.

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**.

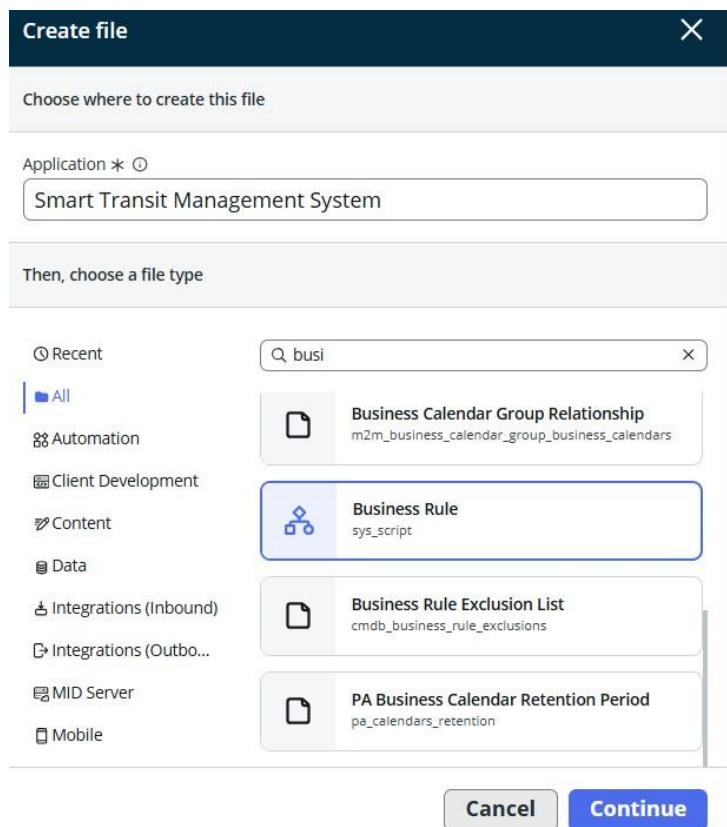


2. Specify the **Application** the file will belong to: *Smart Transport Management System*.

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

4. Scroll down and select **Business Rule(sys_script)** as the file type.

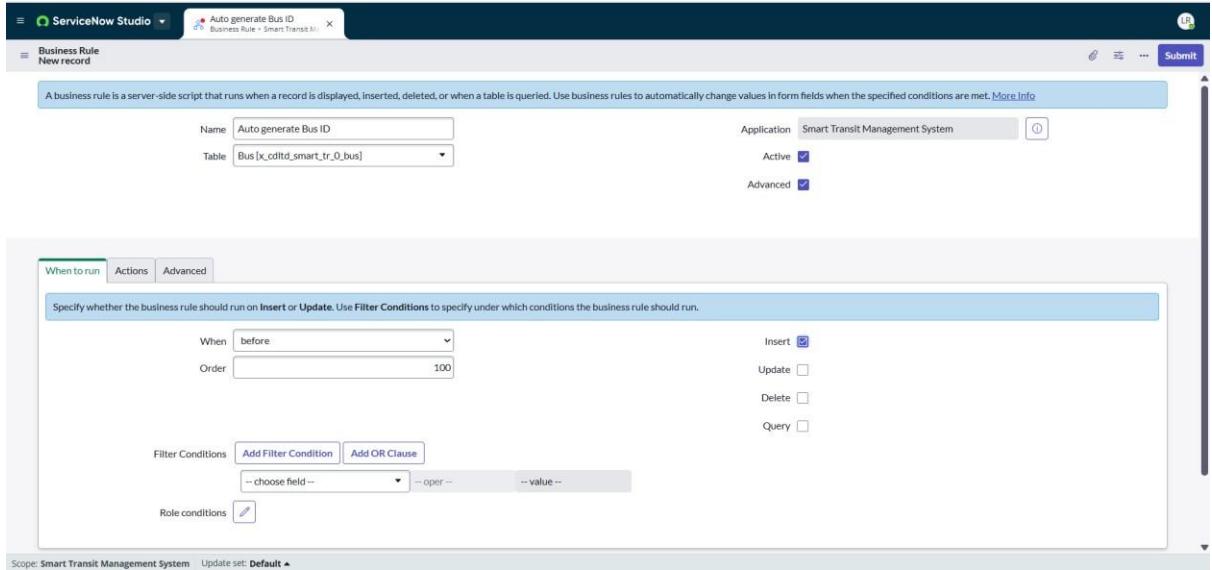
5.Select Continue.



Activity 1:

- Name: Auto generate Bus ID Table:
Bus[x_cdlt_d_smart_tr_0_bus]
- Select **Advanced**.
- In when to run section When:
Beforre Order: 100. Select

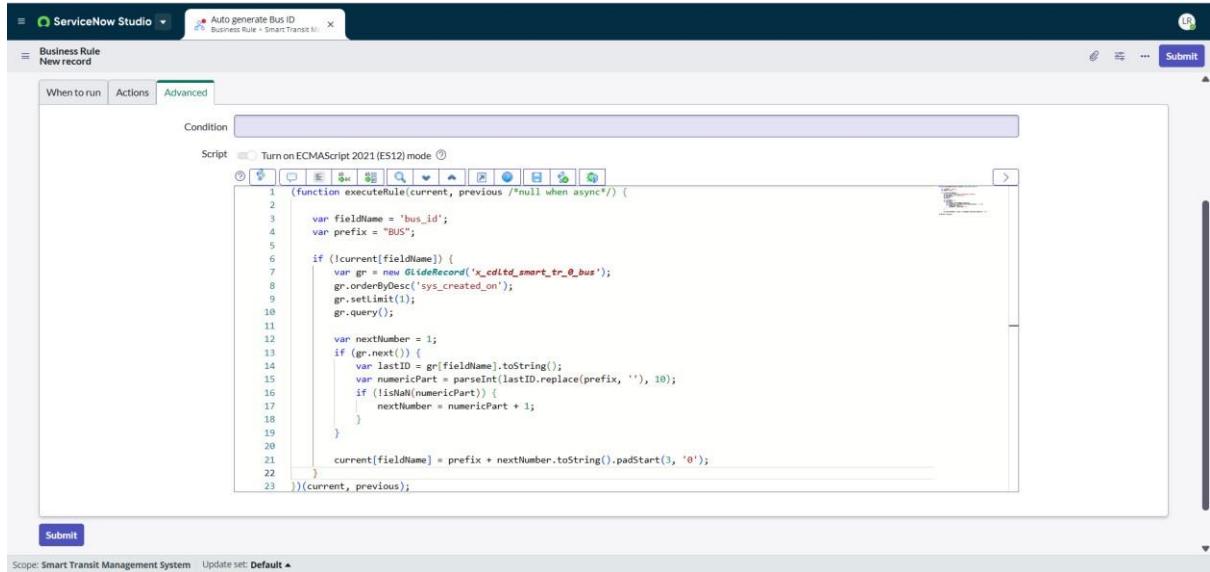
Insert.



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.



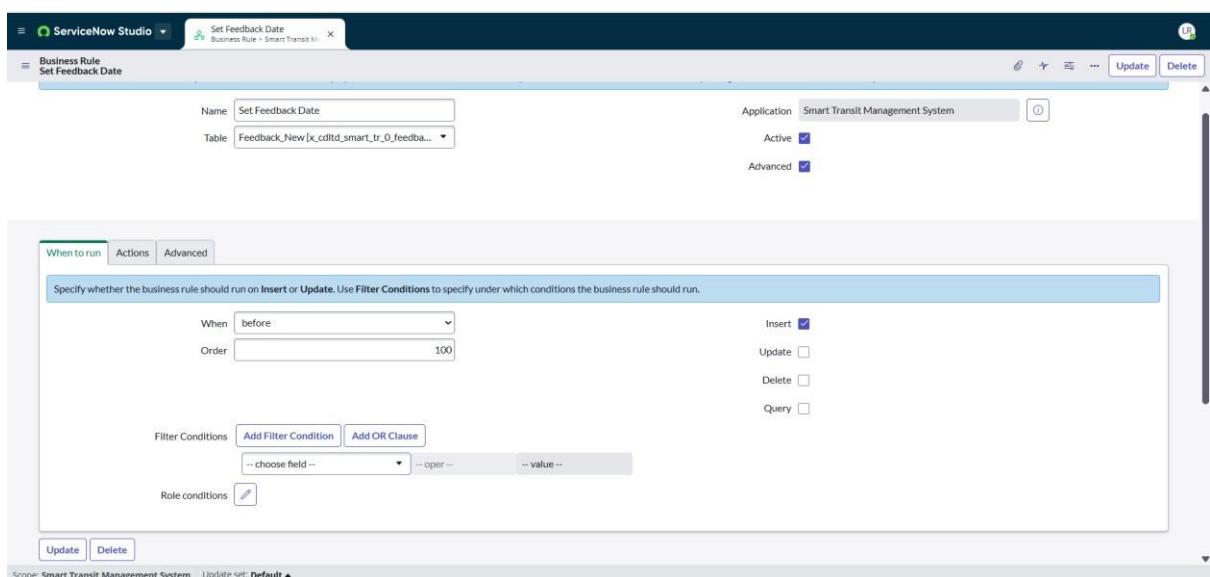
The screenshot shows the ServiceNow Studio interface for creating a business rule. The title bar says "ServiceNow Studio" and the tab is "Business Rule". The sub-tab is "New record". The main area has three tabs: "When to run", "Actions", and "Advanced". The "Actions" tab is selected. It contains a "Condition" section with a "Script" editor. The script is written in ECMAScript 2021 (ES12) mode and defines a function `executeRule` that generates bus IDs. The code uses `gr` (Global Record) to query the database and `nextNumber` to determine the next ID. The "Submit" button is visible at the bottom left.

```
1 (function executeRule(current, previous /*null when async*/){  
2     var fieldName = 'bus_id';  
3     var prefix = "BUS";  
4  
5     if (!current[fieldName]) {  
6         var gr = new GlideRecord('x_cdltd_smart_tr_0_bus');  
7         gr.orderByDesc('sys_created_on');  
8         gr.setLimit(1);  
9         gr.query();  
10  
11         var nextNumber = 1;  
12         if (gr.next()) {  
13             var lastID = gr[fieldName].toString();  
14             var numericPart = parseInt(lastID.replace(prefix, ''), 10);  
15             if (!isNaN(numericPart)) {  
16                 nextNumber = numericPart + 1;  
17             }  
18         }  
19         current[fieldName] = prefix + nextNumber.toString().padStart(3, '0');  
20     }  
21 })(current, previous);
```

Activity 2:

- Name: Set Feedback date
Table: Feedback_new[x_cdltd_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 Transport 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.

The screenshot shows the ServiceNow Knowledge Base interface. At the top, there's a search bar with placeholder text "Search (minimum 3 characters)" and a magnifying glass icon. To the right of the search bar are two buttons: "9 Knowledge Bases" and "50 Articles". On the far right, there's a user profile icon for "Anisha G". Below the header, there's a section titled "Explore our Knowledge Bases" with a "Actions" dropdown menu. The main area displays a grid of knowledge base cards. Each card has a title, a small icon, a count of articles, and a "Subscribe" button. The cards visible include:

Knowledge Base	Count	Action
ADF	2	Subscribe
Admin & Operations	3	Subscribe
Driver Support	1	Subscribe
IT	35	Subscribe
KCS Knowledge Base (demo data)	4	Subscribe
Knowledge	0	Subscribe
Known Error	3	Subscribe
Passenger Support	1	Subscribe

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 interface. 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. To the right, there are buttons for "Sort by" (set to "Newest"), "Views" (set to "Alphabetical"), and "Clear All". The results section displays three articles under the heading "3 Results".
1. "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. It includes steps like logging in to the Admin Portal, navigating to the Driver Management module, adding a driver, and entering contact information.
2. "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. It includes steps like logging in to the Admin Portal, navigating to the Reports module, and generating reports.
3. "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. It includes steps like logging in to the Operations Portal, navigating to the Routes module, clicking "Add Route", and entering route details (start/end points).
Each article has a small thumbnail icon, a title, a brief description, and a "View" button. The "How to Add, Remove, or Update a Driver" article also has a "Monolithic" status indicator.

3. Driver Support Knowledge Articles

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

Step 1: Open the Smart Transport 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 the 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 the search bar is a "Filters" button. The main area displays "1 Results" for the query "Driver Support". The result card has a title "Guides and resources for drivers." and a subtitle "Driver Support | Route Assignment". It includes a brief description: "How to Check Assigned Routes [Drivers] Audience: Drivers Category: Route Assignment Status: 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...)" followed by a truncated URL. At the bottom of the result card, it says "Anisha G • 4 Views • 12d ago • 12 12 12 12". The top right corner of the screen shows the user profile "Anisha G".

4. Passenger Support Knowledge Articles

(a) How to Book a Bus Ticket

Step 1: Open the **Smart Transport 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 base search result. The search term 'How to book a bus ticket?' has been entered into the search bar. One result is displayed, titled 'How to book a bus ticket?'. The result is categorized under 'Passenger Support | Booking Tickets' and is owned by 'Anisha G'. It has 0 views and was created 12d ago. The content of the article is a step-by-step guide on booking a bus ticket via the web portal, including instructions on opening the portal, signing in, and entering travel details.

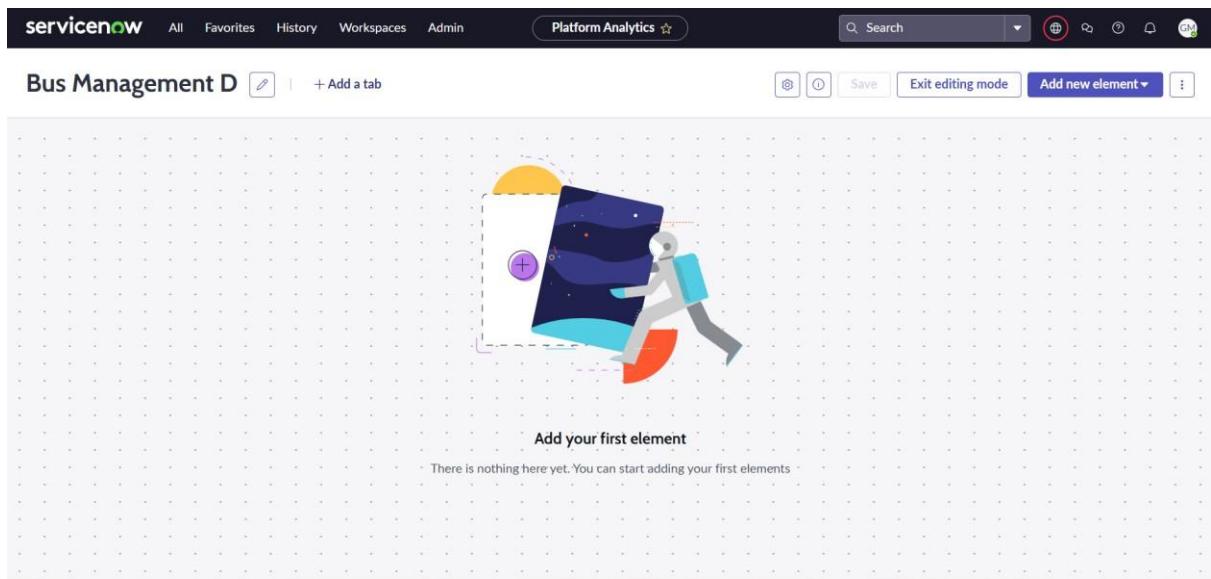
Milestone 12: Dashboards

Activity 1: Bus Management Dashboard

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

The screenshot shows the ServiceNow Dashboards page. On the left, there is a sidebar with filters: 'Recent', 'Bookmarked', 'Certified', 'Owned by me', 'Shared with me', and 'All' (which is selected). The main area displays a grid of 12 dashboard cards. 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). A blue button labeled 'Create new dashboard' is located in the top right corner of the dashboard grid.

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

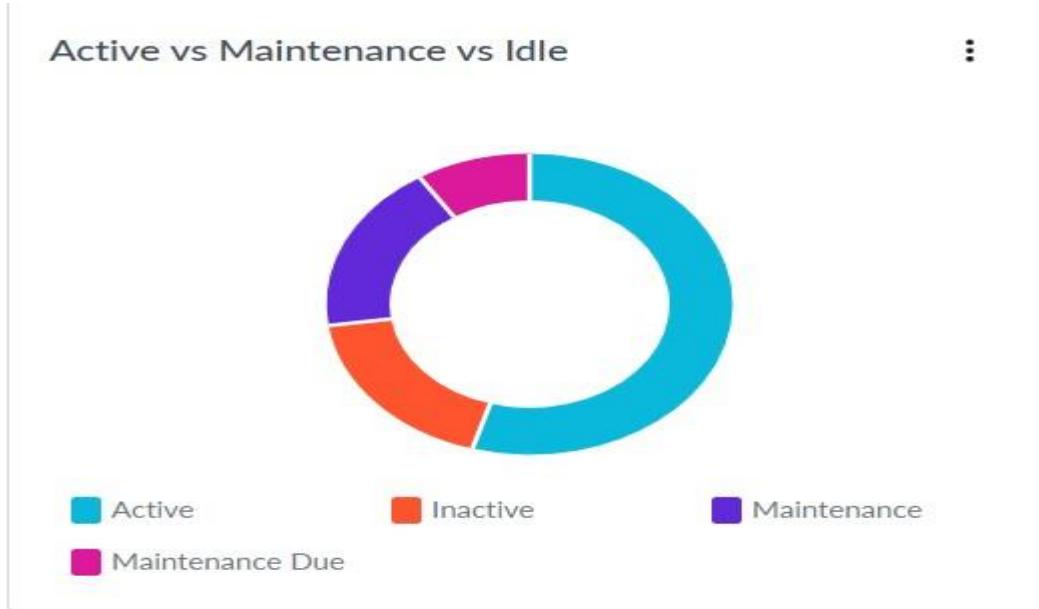


Step-3: Click on Add new element and select data visualization then new visualization.

On configuration pane, Add data source as bus table.

Visualization type: Donut

Chart title: Active vs Maintenance vs idle



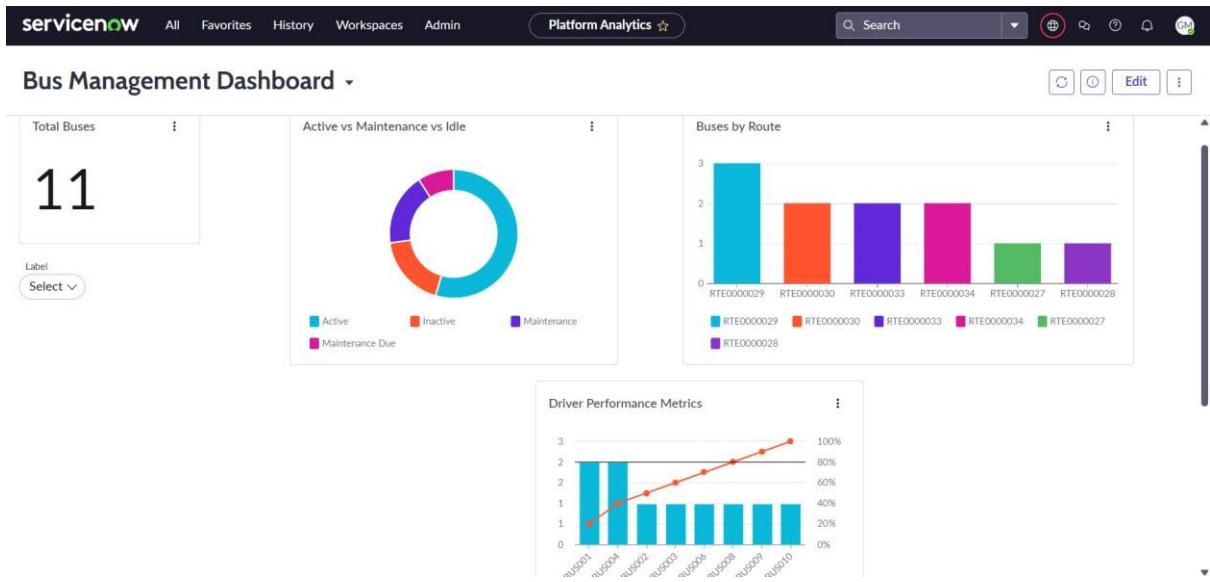
Step-4: Click on Add new element to add Buses by route Barchart. Proceed same as above step. Group by : Route.



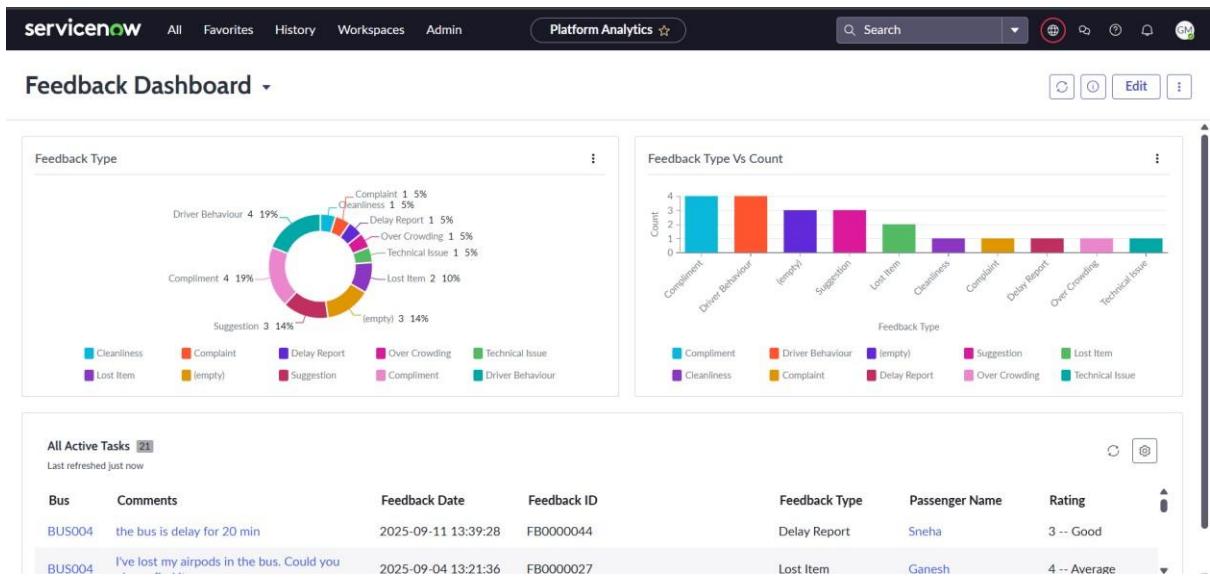
Step-5: Again click on add new element for Driver performance metrics using table: Drivers.



Step-6: Save changes.



Activity-2: Feedback Dashboard



Activity-3: Payments Dashboard

Payments Dashboard ▾



All Active Tasks [15]

Last refreshed just now

Amount	Booking Passenger	Passengers	Payment Date	Payment ID	Payment Method	Payment Status	Transaction ID
£66.0025	9edc5c5e837722107f443dfb0fda1e46	Nikhil	2025-09-24 13:18:54	PAYID01048	Cash	Pending	PAYID01047
£5.0192	865d989e837722107f443dfb0fda1ec2	Mounika	2025-09-13 16:07:42	PAYID01064	Card	Completed	PAYID01063
£1.0038	4e0cd81e837722107f443dfb0fda1e0d	Priya Mehta	2025-09-02 14:28:10	PAYID01016	Net Banking	Pending	PAYID01015

[View all](#)

Activity-4: Routes Dashboard

Route Dashboard ▾

Route id ▲	Active ▼	Distance	End Point	Start Point
RTE00000026	true	515.3	Bengaluru	Chennai
RTE00000027	true	679.8	Hyderabad	Chennai

