

# SkyConnect

A step by step guide for creating a full stack application integrated with  
AI chatbot



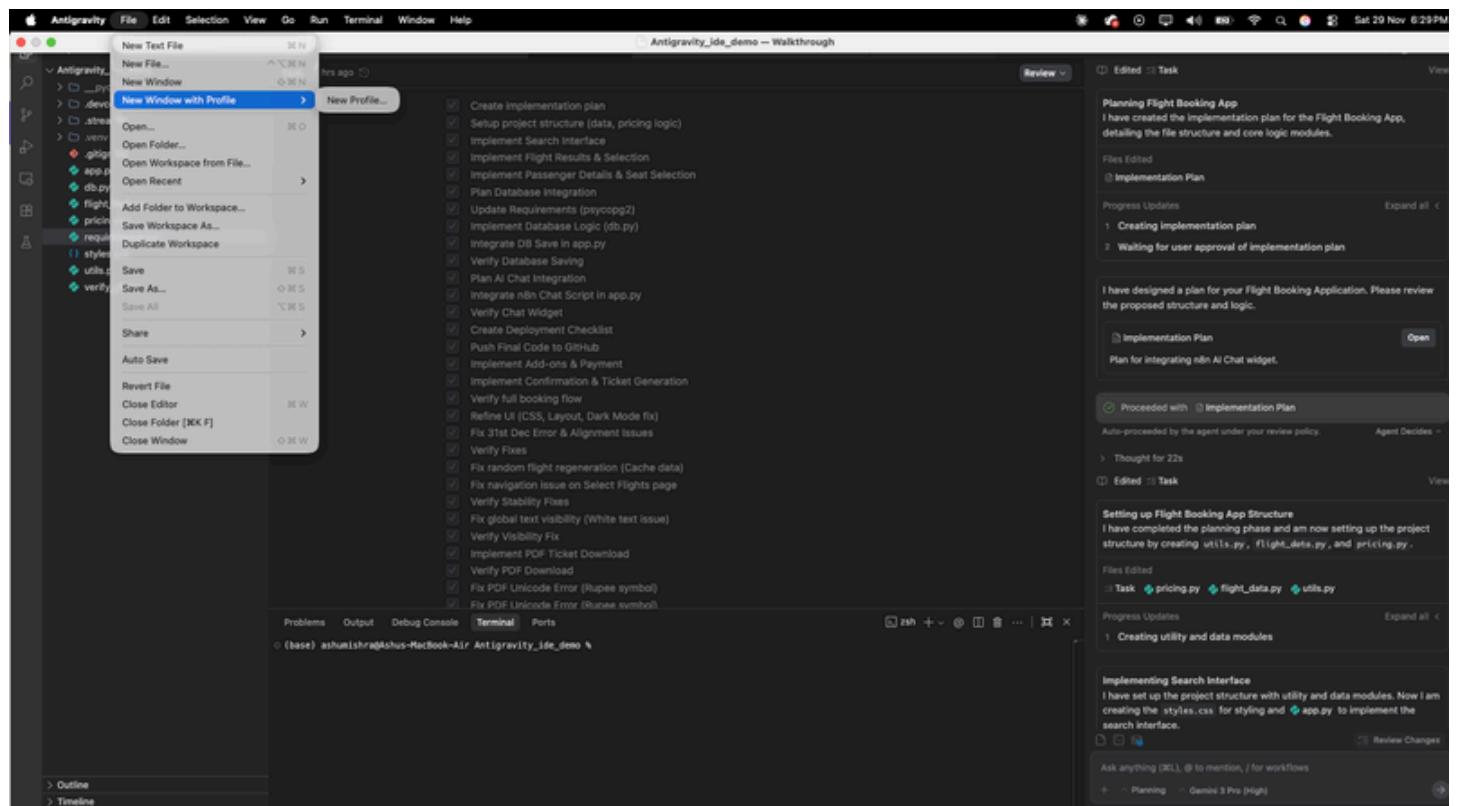
<https://github.com/ashumishra2104>



<https://www.linkedin.com/in/ashumish/>

1) Create a folder on desktop called as "Skyconnect"

2) On left hand side , open the folder in Antigravity ide :



3) Use the following prompt :

Create a modern, responsive web application for booking round-trip flights between Hyderabad (HYD) and Goa (GOI) with the following specifications:

Core Features Required:

1. Flight Search Interface:

- \* Departure city: Hyderabad (fixed)
- \* Arrival city: Goa (fixed)
- \* Date picker for departure date
- \* Date picker for return date
- \* Number of passengers (Adults, Children, Infants)
- \* Class selection (Economy, Premium Economy, Business)
- \* Search button to find available flights

2. Flight Display & Selection:

- \* Show outbound flights (HYD → GOA) with:
  - \* Airlines (IndiGo, Air India, SpiceJet, Vistara, AirAsia)
  - \* Departure and arrival times
  - \* Flight duration
  - \* Flight number
  - \* Number of stops (Non-stop, 1 stop, 2+ stops)
- \* Show return flights (GOA → HYD) with same details
- \* Allow independent selection of outbound and return flights

3. Dynamic Pricing System:

- \* Base fare calculation based on:
  - \* Days in advance (cheaper if booked 30+ days ahead)
  - \* Day of week (weekends 20% higher)
  - \* Time of day (morning/evening flights 15% higher)
  - \* Class multipliers (Economy: 1x, Premium: 1.5x, Business: 2.5x)
- \* Additional charges:
  - \* Convenience fee: ₹200 per ticket

\* GST: 5% of base fare

\* Passenger service fee: ₹150 per passenger

\* Fuel surcharge: ₹500 per ticket

\* Display price breakdown transparently

#### 4. Passenger Details Form:

\* Full name (First, Middle, Last)

\* Gender

\* Date of birth

\* Contact information (email, phone)

\* Special requirements (meals, wheelchair, etc.)

\* Frequent flyer number (optional)

#### 5. Seat Selection:

\* Visual seat map for both flights

\* Different pricing for:

\* Regular seats (free)

\* Extra legroom seats (+₹500)

\* Window/Aisle preference (+₹200)

\* Front rows (+₹300)

#### 6. Add-ons & Services:

\* Baggage options:

\* Cabin baggage (7kg free)

\* Check-in baggage (15kg: ₹1200, 20kg: ₹1500, 30kg: ₹2000)

\* Meal selection (+₹350 per meal)

\* Travel insurance (+₹299)

\* Priority check-in (+₹400)

\* Lounge access (+₹1500)

#### 7. Payment Integration:

\* Multiple payment methods:

\* Credit/Debit cards

\* UPI

\* Net banking

\* Wallets

\* Apply promo codes/discounts

\* EMI options for amounts above ₹5000

#### 8. Booking Confirmation:

\* Generate unique PNR

\* Send e-ticket via email

\* SMS confirmation

\* Download ticket as PDF

\* Add to calendar option

#### Sample Pricing Logic:

##### Base Price Calculation:

- HYD to GOA base fare: ₹3500 (economy)

##### Advance booking discount:

- 30+ days: -20%

- 15-29 days: -10%

- 7-14 days: 0%

- Less than 7 days: +25%

- Peak season surcharge (Oct-Jan): +30%

- Weekend surcharge: +20%

##### Time-based pricing:

- Early morning (6-8 AM): +15%

- Morning (8-12 PM): +10%

- Afternoon (12-4 PM): Base price

- Evening (4-8 PM): +15%

- Night (8 PM-6 AM): -10%

#### Technical Requirements:

\* Responsive design (mobile, tablet, desktop)

\* Real-time seat availability

\* Session management for booking flow

\* Form validation and error handling

\* Loading states and animations

\* Auto-save progress

\* Multi-language support (English, Hindi, Telugu)

\* Accessibility features (WCAG compliant)

\* Please use streamlit to deploy the front end

#### UI/UX Guidelines:

\* Clean, modern interface with airline-themed colors

\* Step-by-step booking process with progress indicator

\* Clear CTAs and visual hierarchy

\* Price transparency at every step

\* Comparison view for flight options

\* Filter and sort options for flight results

\* Calendar view for date selection with price indicators

\* Mobile-first approach

#### Additional Features:

\* Flight status tracker

\* Booking history

\* Manage booking (cancel, modify, web check-in)

\* Fare calendar showing cheapest days to fly

\* Price alerts for selected dates

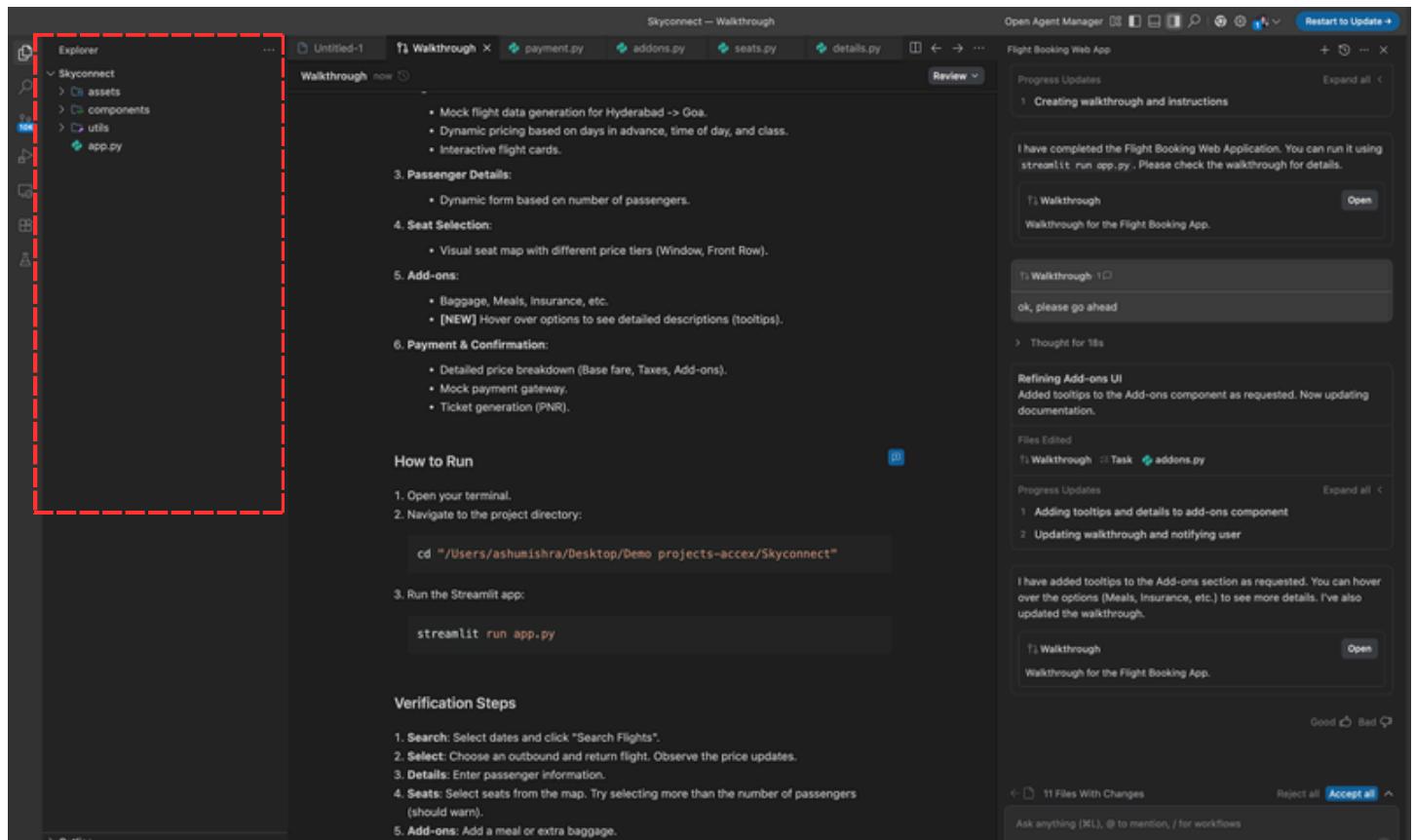
\* Group booking option (10+ passengers)

\* Corporate booking portal

\* Round-trip vs one-way price comparison

Please implement this as a single-page application (SPA) with smooth transitions between booking steps, real-time price updates, and a checkout process that clearly shows the total cost breakdown before payment.

4) Agent will build all the files , which you can see the left hand side :



4) The agent will provide a walkthrough document for your review :

The screenshot shows the GitHub Walkthrough interface for a "Flight Booking App Walkthrough". The main content area displays the "Flight Booking App Walkthrough" document, which includes sections like "Overview", "Features Implemented", "How to Run", and "Verification". The "Features Implemented" section lists various components and their features. The "How to Run" section provides terminal commands to run the Streamlit app. The "Verification" section states that all UI phases and logic implementation are completed and ready for verification. On the right side, there are two panels: "UI Implementation - Phase 2" and "UI Implementation - Phase 3", both showing progress updates and files edited. A bottom panel shows a summary of the completed project and a "Walkthrough for the Flight Booking App" button.

5) Add comments ,if you would like to change any part here :

The screenshot shows the GitHub Walkthrough interface for a "Flight Booking App Walkthrough". The main content area displays the "Flight Booking App Walkthrough" document, which includes sections like "Features Implemented", "How to Run", and "Verification". The "Features Implemented" section lists various components and their features. A comment box is overlaid on the "Features Implemented" section, containing the text: "Also write the details or provide a tool tip button to see the details". The "How to Run" section provides terminal commands to run the Streamlit app. The "Verification" section states that all UI phases and logic implementation are completed and ready for verification.

6) Once done, click on “Review” , and type “Please go ahead after reviewing the comments in any”

Untitled-1 Walkthrough X payment.py addons.py seats.py details.py ⌂ ← → ⋮ F

Walkthrough 2 mins ago Review • 1 □ ▾

## Flight Booking App Walkthrough

### Overview

I have built a complete Single Page Application (SPA) for a flight booking app. The app features a modern, airline-themed UI with a step-by-step booking flow.

### Features Implemented

**1. Search Interface:**

- Date pickers with validation.
- Passenger and Class selection.

**2. Flight Selection:**

- Mock flight data generation for Hyderabad -> Goa.
- Dynamic pricing based on days in advance, time of day, and class.
- Interactive flight cards.

Submit comment

ok, please go ahead Submit

> Review 1 comment

7) The agent will start doing changes to your code

## Walkthrough · 1

ok, please go ahead

### Thought Process

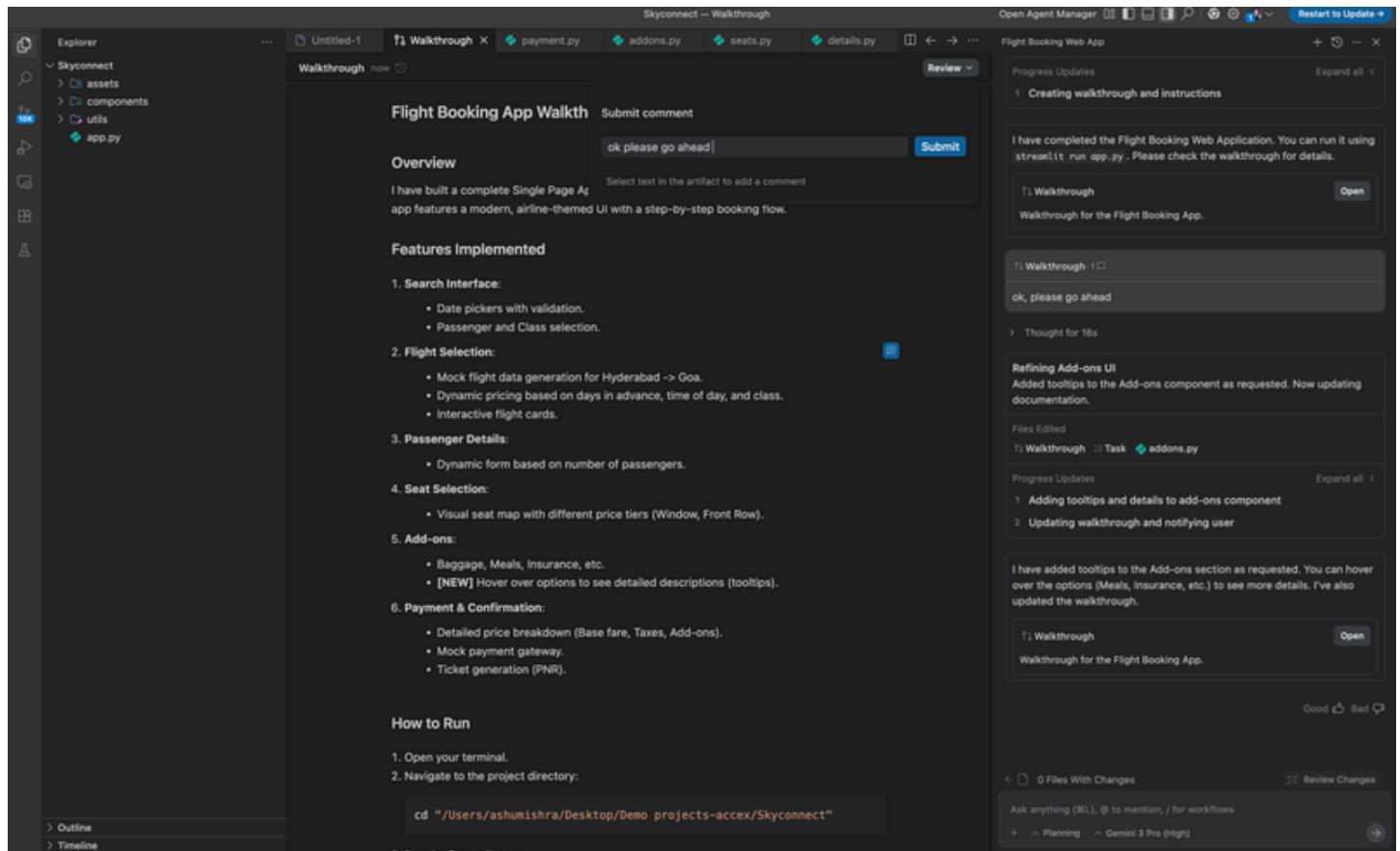
to ensure clarity.

### Implementing Contextual Help

I'm now integrating detailed `help` strings directly into the add-on options within `components/addons.py`. I've added concise explanations for each option, such as baggage weight limits for the baggage allowance add-on, and descriptions for meal, insurance, priority check-in, and lounge access. This follows the suggestion to include tooltips or expanders. This will result in clearer choices for the user. I'll be updating `walkthrough.md` to show the updated UI.

Thinking

8) The agent will give you a revised walkthrough

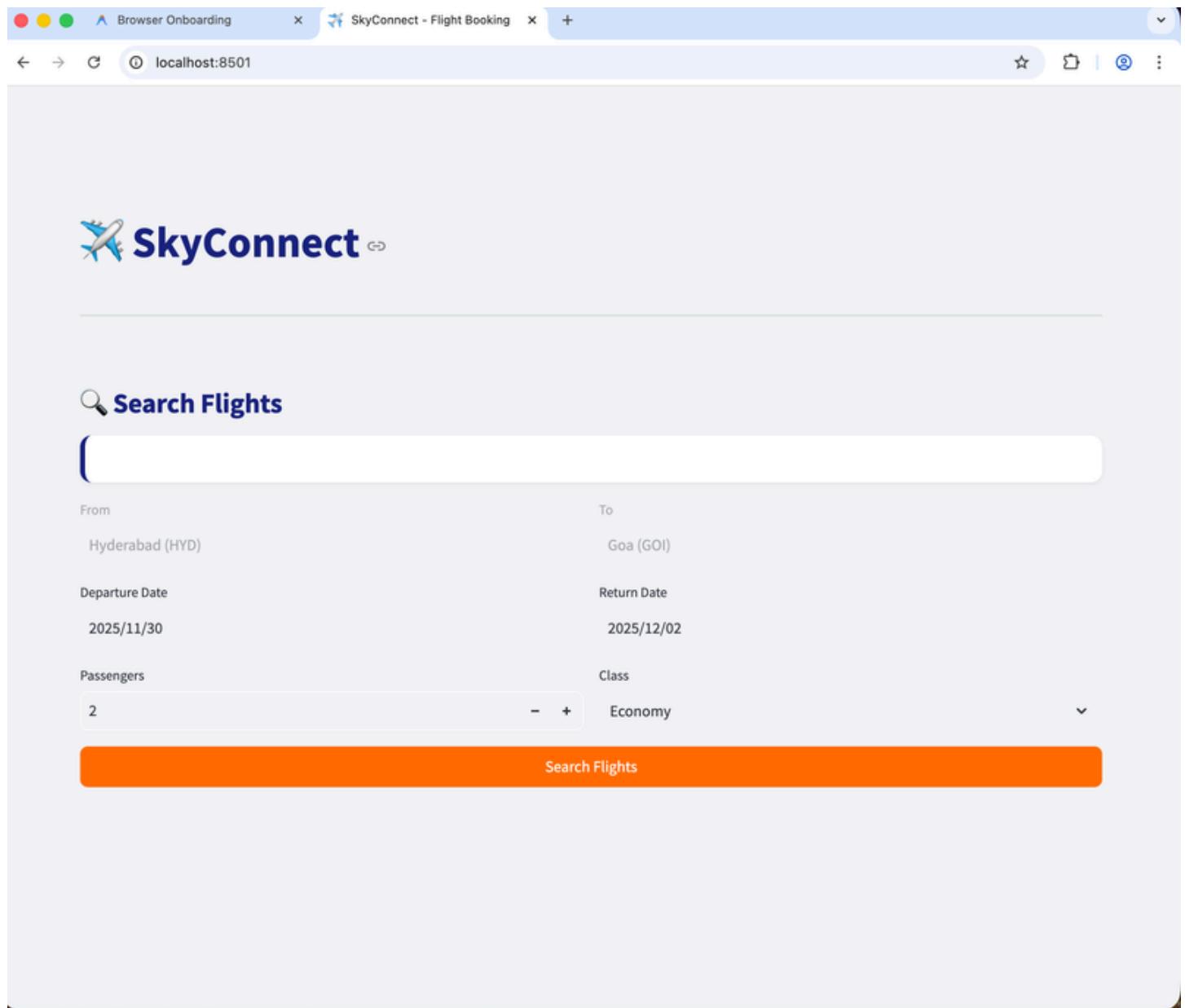


9) Once done , again ,press on review and give a go ahead

10) The agent will create code base for you

11) Prompt the agent “ Please run the code on streamlit local”

12) A new tab, most likely with “localhost:8501” will open up :



13) Test the flow end to end, prompt the agent in case of errors

14) Once done, ask the agent to test the end to end functionality with prompt 'Please test the whole flow end to end using chromium'

The agent will open up the browser and automatically navigate through multiple screens.



## Select Flights

### Outbound: HYD → GOI

30 Nov 2025

AirAsia	08:15	1h 39m	09:54	₹5,775	SpiceJet	13:15	1h 16m	14:31	₹4,375
IS-387	HYD	Non-stop	GOI	Select	SG-701	GOI	Non-stop	HYD	Select

SpiceJet	15:00	1h 22m	16:22	₹5,250	Vistara	12:00	1h 36m	13:36	₹4,375
SG-430	HYD	1 Stop	GOI	Select	UK-932	GOI	1 Stop	HYD	Select

Air India	20:30	1h 31m	22:01	₹4,725	AirAsia	05:00	1h 38m	06:38	₹3,938
AI-957	HYD	Non-stop	GOI	Select	IS-953	GOI	Non-stop	HYD	Select

Getting DOM...

15) The agent has created whole flow on its own , and tested it :

**End-to-End Testing**  
Fixed progress bar error. Now starting end-to-end verification.

Progress Updates      Collapse all ▾

1 Running full booking flow in browser

2. Select an outbound flight.  
3. Select a return flight.  
4. Click "Continue".  
5. Enter "John" as First Name and "Doe" as Last Name for the passenger.  
6. Click "Continue to Seats".  
7. Select a seat (e.g., 1A).  
8. Click "Continue to Add-ons".  
9. Select "Add Meal".  
10. Click "Review & Pay".  
11. Click "PAY NOW".  
12. Verify that the "Booking Confirmed!" message appears.

Entered John Doe as passenger name.

6. Clicked "Continue to Seats".  
7. Selected seat 1A.  
8. Scrolled and clicked "Continue to Add-ons".  
9. Clicked "Add Meal".  
10. Clicked "Review & Pay".  
11. Clicked "PAY NOW".  
12. Verified the "Booking Confirmed!" message is present.

The end-to-end flow was successful.

● Playback available      View

16) The agent will also provide a detailed walkthrough report :

**Flight Booking App Walkthrough**

**Overview**

I have built a complete Single Page Application (SPA) for flight booking using Streamlit. The app features a modern, airline-themed UI with a step-by-step booking flow.

**Features Implemented**

1. Search Interface:
  - Date pickers with validation.
  - Passenger and Class selection.
2. Flight Selection:
  - Mock flight data generation for Hyderabad -> Goa.
  - Dynamic pricing based on days in advance, time of day, and class.
  - Interactive flight cards.
3. Passenger Details:
  - Dynamic form based on number of passengers.
4. Seat Selection:
  - Visual seat map with different price tiers (Window, Front Row).
5. Add-ons:
  - Baggage, Meals, Insurance, etc.
  - [NEW] Hover over options to see detailed descriptions (tooltips).
6. Payment & Confirmation:
  - Detailed price breakdown (Base fare, Taxes, Add-ons).
  - Mock payment gateway.
  - Ticket generation (PNR).

**How to Run**

1. Open your terminal.
2. Navigate to the project directory:

```
cd "/Users/ashumishra/Desktop/Demo_projects-accecc/Skyconnect"
```

**Progress Updates**

1. Running full booking flow in browser
  - 2. Select an outbound flight.
  - 3. Select a return flight.
  - 4. Click "Continue".
  - 5. Enter "John" as First Name and "Doe" as Last Name for the passenger.
  - 6. Click "Continue to Seats".
  - 7. Select a seat (e.g., 1A).
  - 8. Click "Continue to Add-ons".
  - 9. Select "Add Meal".
  - 10. Click "Review & Pay".
  - 11. Click "PAY NOW".
  - 12. Verify that the "Booking Confirmed!" message appears.
6. Clicked "Continue to Seats".
7. Selected seat 1A.
8. Scrolled and clicked "Continue to Add-ons".
9. Clicked "Add Meal".
10. Clicked "Review & Pay".
11. Clicked "PAY NOW".
12. Verified the "Booking Confirmed!" message is present.

The end-to-end flow was successful.

**Finalizing Fix**

Fixed progress bar error. Verified end-to-end flow using browser subagent.

**Files Edited**

**T1 Walkthrough**

**Progress Updates**

1. Updating walkthrough with recording and notifying user

I have fixed the Streamlit APIException by clamping the progress bar value to a maximum of 1.0. I also ran a complete end-to-end test of the booking flow using the browser, which was successful. I've added a recording of the full flow to the walkthrough.

0 Files With Changes

Ask anything (XL), @ to mention, / for workflows

Planning Gemini 3 Pro (High)

## Congratulations ! You have deployed the code on your local machine

Let's push the code to github

17) Go to Github.com , and sign in to your account.If you have not created account on github , please see the following video :

[How to Create a GitHub Account | GitHub Sign Up Video](#)

18) Create a new repository named as “Skyconnect” or any name you wish to create

New repository

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#). Required fields are marked with an asterisk (\*).

**1 General**

Owner \* Repository name \*

ashumishra2104 /

Great repository names are short and memorable. How about [reimagine-adventure](#)?

Description

0 / 350 characters

**2 Configuration**

Choose visibility \* Choose who can see and commit to this repository

Public

Start with a template Templates pre-configure your repository with files.

No template

Add README READMEs can be used as longer descriptions. [About READMEs](#)

Off

Add .gitignore .gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license Licenses explain how others can use your code. [About licenses](#)

No license

19) Once done ,you will land up on a page like this :

ashumishra2104 / skyconnect\_demo

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

skyconnect\_demo Public

Start coding with Codespaces Add a README file and start coding in a secure, configurable, and dedicated development environment.

Create a codespace

Add collaborators to this repository Search for people using their GitHub username or email address.

Invite collaborators

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH [https://github.com/ashumishra2104/skyconnect\\_demo.git](https://github.com/ashumishra2104/skyconnect_demo.git)

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

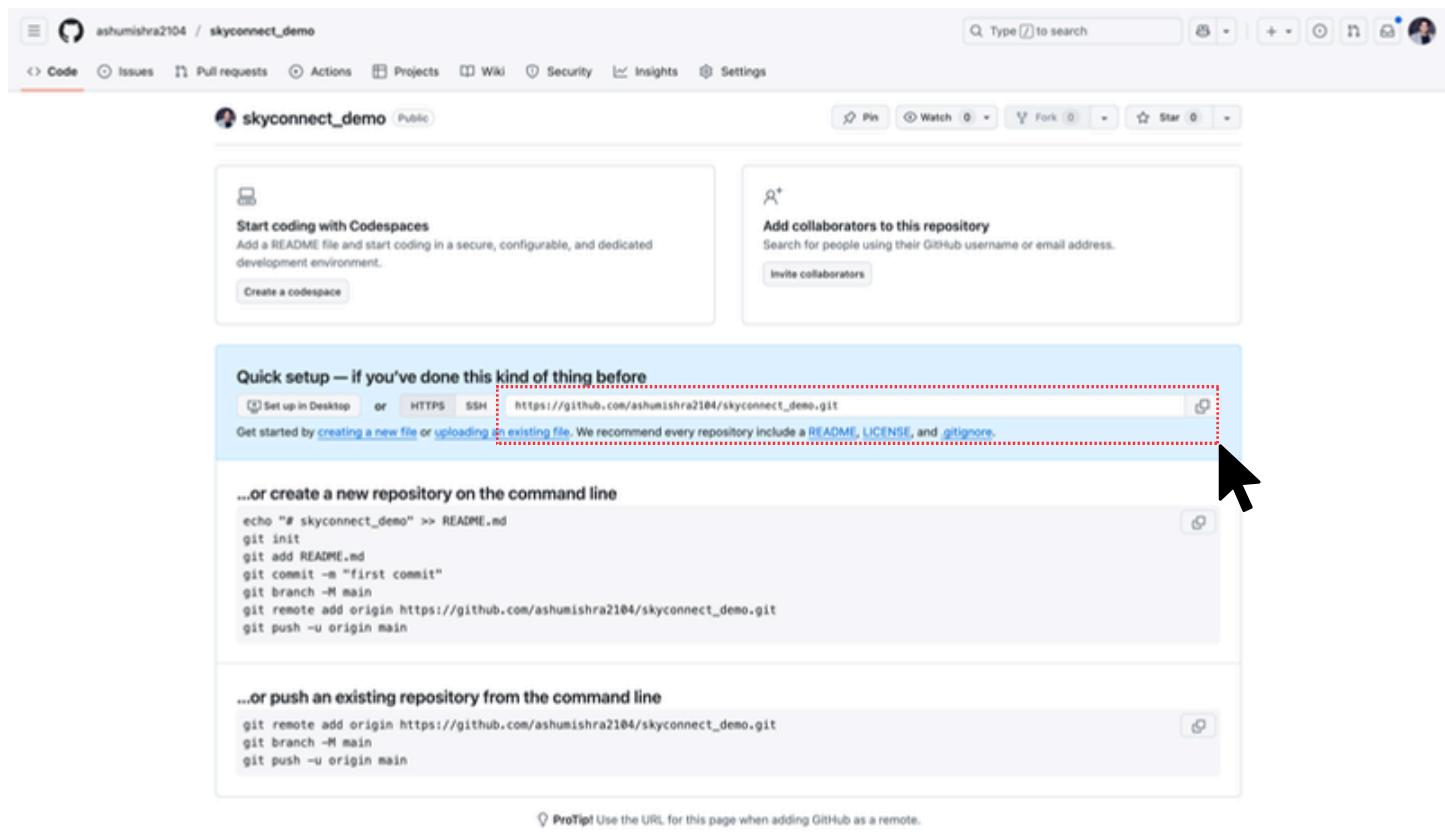
```
echo "# skyconnect_demo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ashumishra2104/skyconnect_demo.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/ashumishra2104/skyconnect_demo.git
git branch -M main
git push -u origin main
```

ProTip! Use the URL for this page when adding GitHub as a remote.

## 20) Copy the link as provided in Quick Setup

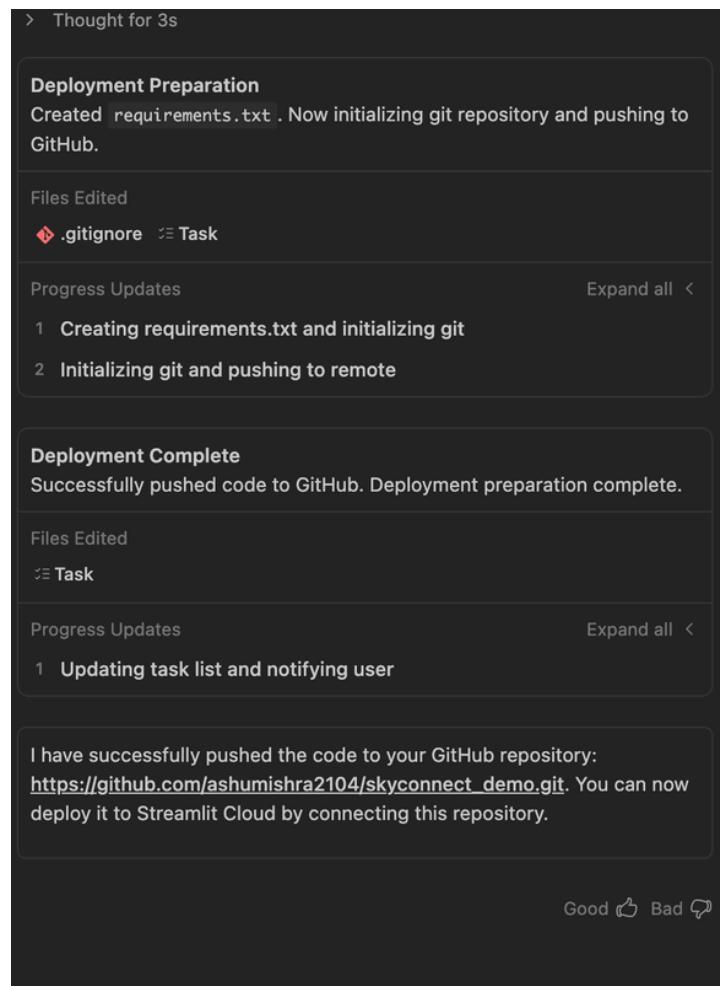


21) Once copied, come back to your agent and type the following prompt :

I want to push this code to github , and then to streamlit ,please create required files and then push to : **{please copy  
paste the above link here}**

22) Kindly authenticate the Antigravity IDE for pushing the code

23) The agent will push the code to Github and following message will come :



24) Check your github ,all the files have been pushed

The GitHub repository page for 'skyconnect\_demo' shows the following details:

- Code**: Main branch, 1 Branch, 0 Tags
- Commits**: 1 Commit by Ashu Mishra (Initial commit) 3e389f9 - 2 minutes ago
- Files**: assets, components, utils, .gitignore, app.py, requirements.txt (all initial commits, 2 minutes ago)
- README**: Add a README button
- About**: No description, website, or topics provided.
- Activity**: 0 stars, 0 watching, 0 forks
- Releases**: No releases published. Create a new release.
- Packages**: No packages published. Publish your first package.
- Languages**: Python 95.3%, CSS 4.7%
- Suggested workflows**: Based on your tech stack. Python package (Configure) button.

# Congratulations ! You have pushed the code on github

Let's deploy your code on Streamlit and make it running

25) Go to streamlit.com and create your account , for detailed guidance , please refer :

[Create your account - Streamlit Docs](#)

26) Press on "Create App" in the top right hand corner

The screenshot shows the Streamlit website interface. At the top, there is a navigation bar with links for 'My apps', 'My profile', 'Explore', and 'Discuss'. On the far right of the top bar, there is a button labeled 'Create app' which is highlighted with a red dashed box. A large black cursor arrow points directly at this button. Below the navigation bar, the main content area is titled 'ashumishra2104's apps'. It lists four existing apps: 'ai\_travel\_agent\_streamlit', 'chart\_generator\_internet\_search', 'flightbookingapp', and 'rag\_news\_agent'. Each app entry includes a small icon, the app name, and a 'View demo' link. To the right of the app list, there is a vertical ellipsis menu. At the bottom of the page, there is a section titled 'Get started from a template' featuring several template cards with preview images, names like 'GDP over time', 'Chatbot', 'Existing tickets', and 'My new app', and 'View demo' links.

27) Choose "Deploy a public app from GitHub"

The screenshot shows the Streamlit website again, but this time the user has navigated to a specific section. At the top, there is a back arrow labeled '← Back'. Below it, the heading 'What would you like to do?' is centered. There are three main options presented in boxes: 1. 'Deploy a public app from GitHub' with a GitHub icon, a description about code being ready on a GitHub repo, and a 'Deploy now' button. 2. 'Deploy a public app from a template' with a Chatbot icon, a description about seeing amazing concoctions, and a 'Check out templates' button. 3. 'Deploy a private app in Snowflake' with a Snowflake icon, a description about unlimited enterprise-grade apps, and a 'Start trial →' button.

28) In deploy an app -

**Repository** : Select the repository where you have uploaded the code

**Branch** - default selected "main"

**Main file path** - app.py

**App url** - will be generated by default

[← Back](#)

## Deploy an app

Repository [?](#)

ashumishra2104/skyconnect\_demo

Paste GitHub URL

Branch

main

Main file path

app.py

App URL (optional)

skyconnectdemo-a9cvvypyxbwndn24rkayr

.streamlit.app

Domain is available

[Advanced settings](#)

Deploy

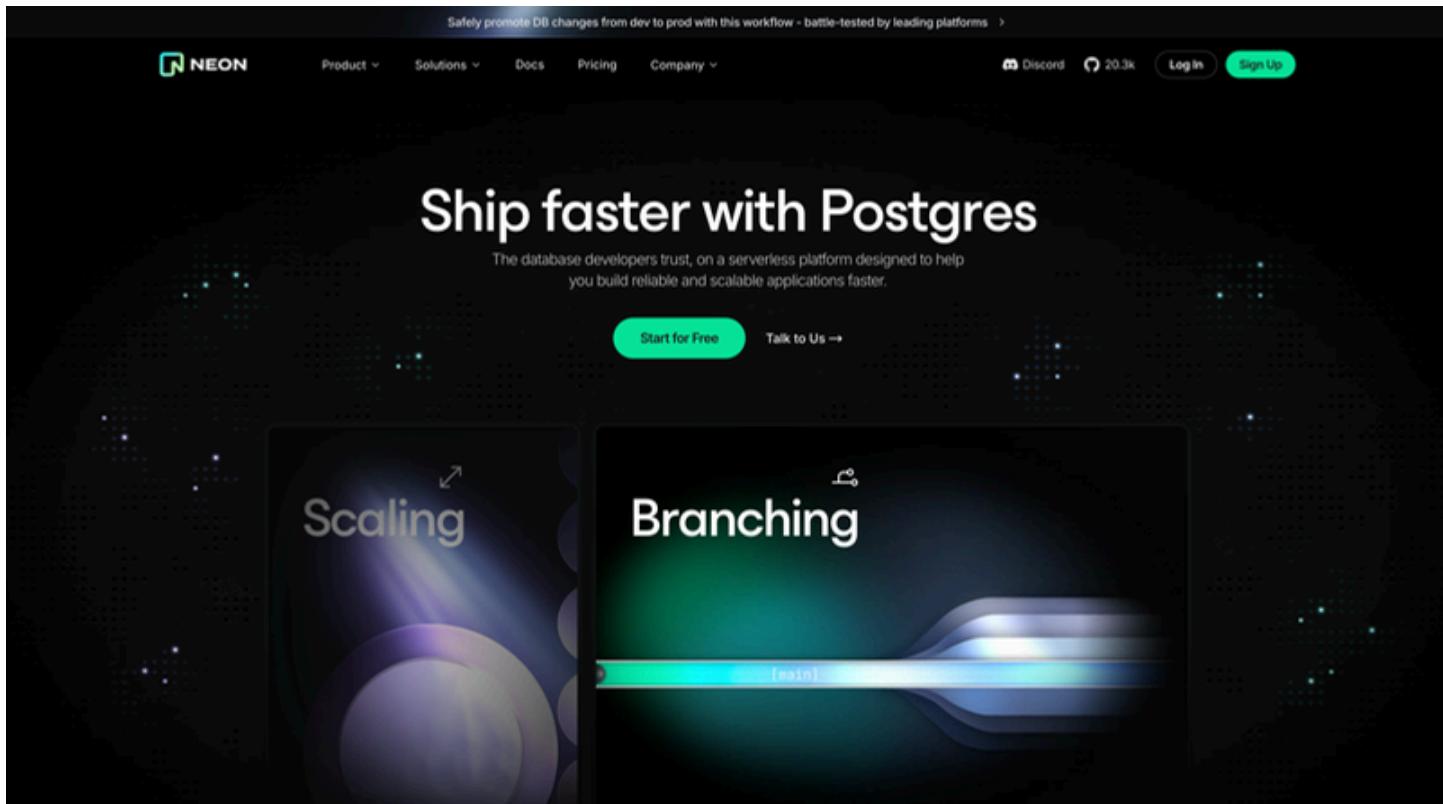
29) Your app will be deployed in some time and you will see the following :

The screenshot shows a web browser window with the URL `skyconnectdemo-a9cvvypyxbwndn24rkayr.streamlit.app` in the address bar. The main content is a Streamlit application titled "SkyConnect". It features a logo with a blue and white airplane icon followed by the word "SkyConnect". Below the logo is a search bar with the placeholder "Search Flights". Underneath the search bar are several input fields: "From" set to "Hyderabad (HYD)", "To" set to "Goa (GOI)", "Departure Date" set to "2025/11/30", "Return Date" set to "2025/12/02", "Passengers" set to "1", and "Class" set to "Premium Economy". At the bottom of the form is a large orange button with the text "Search Flights".

Congratulations ! You have deployed the code on streamlit  
Share it with your friends or post it on Linkedin !

Let's add database to this :

30) Go to [Neon Serverless Postgres — Ship faster](#)

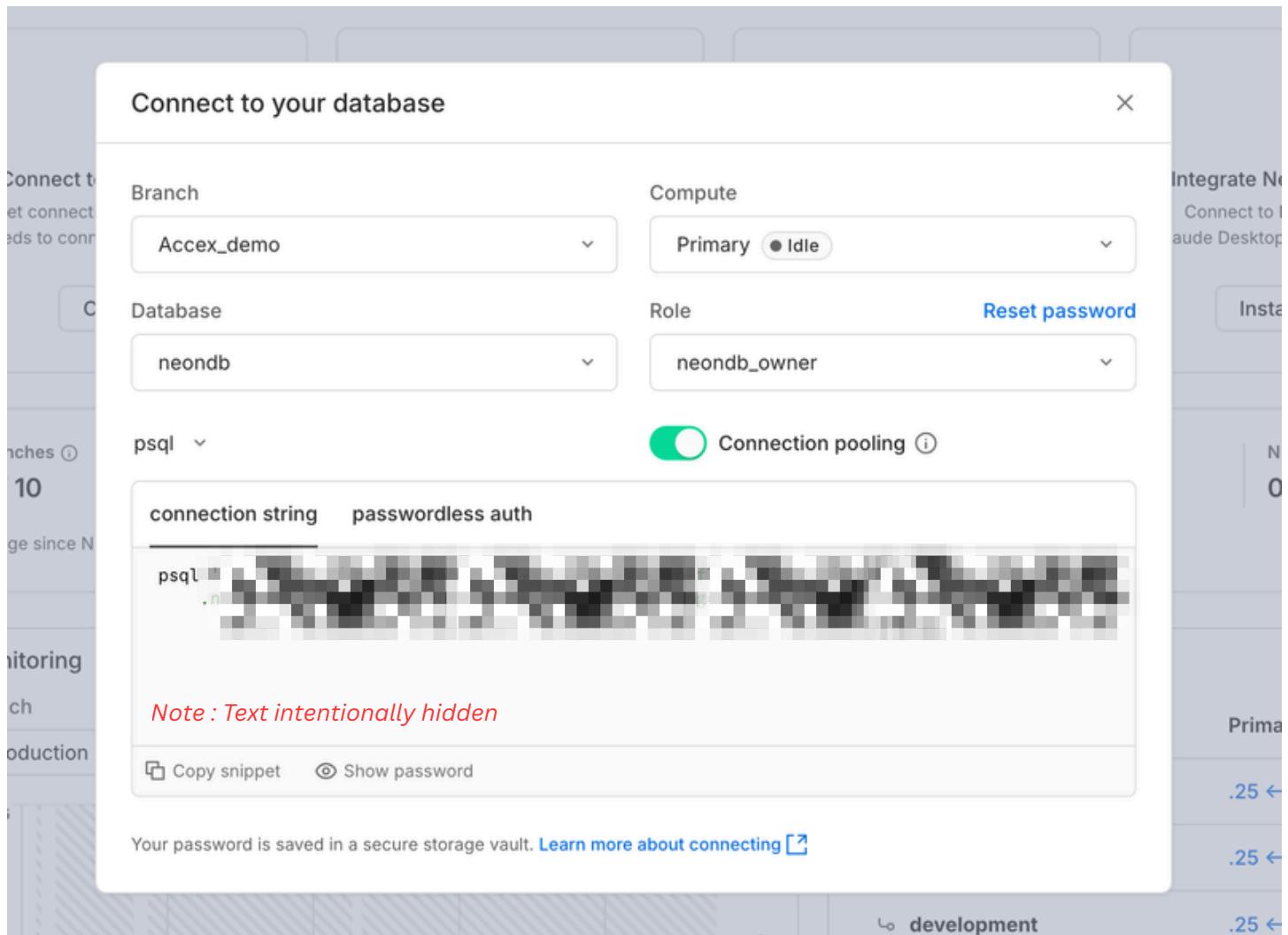


31) Go to “**Sign up**” on the top right corner and create your account [ Guide : [Learn the basics – Neon Docs](#) ]

32) On the right hand side ,you will see “**Connect**” button, click on it

The screenshot shows the Neon Project dashboard for a project named "flightbooker". On the left, there is a sidebar with various project settings like Dashboard, Branches, Integrations, Auth, Settings, Overview, Data API, Monitoring, SQL Editor, Tables, Backup &amp; Restore, and Data Masking. A large black arrow points towards the "Connect" button in the top right corner of the dashboard area. The dashboard itself has sections for "Project dashboard", "Get started with your new database", and monitoring metrics. The "Monitoring" section includes a chart for CPU usage over time and a table for "3 Branches".

33) Copy the connection string using "Copy Snippet" button



34) Use the following prompt in your Antigravity IDE :

Push my data captured to : **{your\_own\_url}**. Please create the code accordingly, capture all the related fields. Decide on the schema on your own and push changes to Github

35) Your code will automatically be pushed to Github and streamlit will rerun ,with the following window :



## Search Flights

From: Hyderabad (HYD) To: Goa (GOI)

Departure Date: 2025/12/01 Return Date: 2025/12/03

Passengers: 1 Class: Economy

**Search Flights**

Manage app

36) Test your flow end to end , and data will be automatically in the tables section of neon.com

The screenshot shows the Neon Data Platform interface. On the left, there's a sidebar with navigation links like Dashboard, Branches, Integrations, Auth, Settings, Overview, Data API, Monitoring, SQL Editor, Tables (which is selected), Backup & Restore, and Data Masking (Beta). A red dashed box highlights the 'Tables' section. Below it, another red dashed box highlights the 'Table Schema' section, which lists tables: bookings, flight\_segments, and passengers. The main area shows a table titled 'Tables' with data from the 'neondb' database. A red dashed box highlights the table content. The table has columns: id (serial), pmr (varchar), booking\_date (timestamp), total\_amount (double precision), contact\_email (varchar), and contact\_phone (varchar). The data rows are:

id	pmr	booking_date	total_amount	contact_email	contact_phone
1	TEST01	2025-11-29 14:19:42.02...	5000	test@example.com	1234567890
2	S60ZG0	2025-11-29 14:20:24.78...	14870.4	ashumishra1@outlook.com	1234324234
3	HEN5QD	2025-11-29 14:24:19.95...	29435.15	Email@Email.com	123456

**Data in the table**

Congratulations !

Let's recap ,what we have done till now -

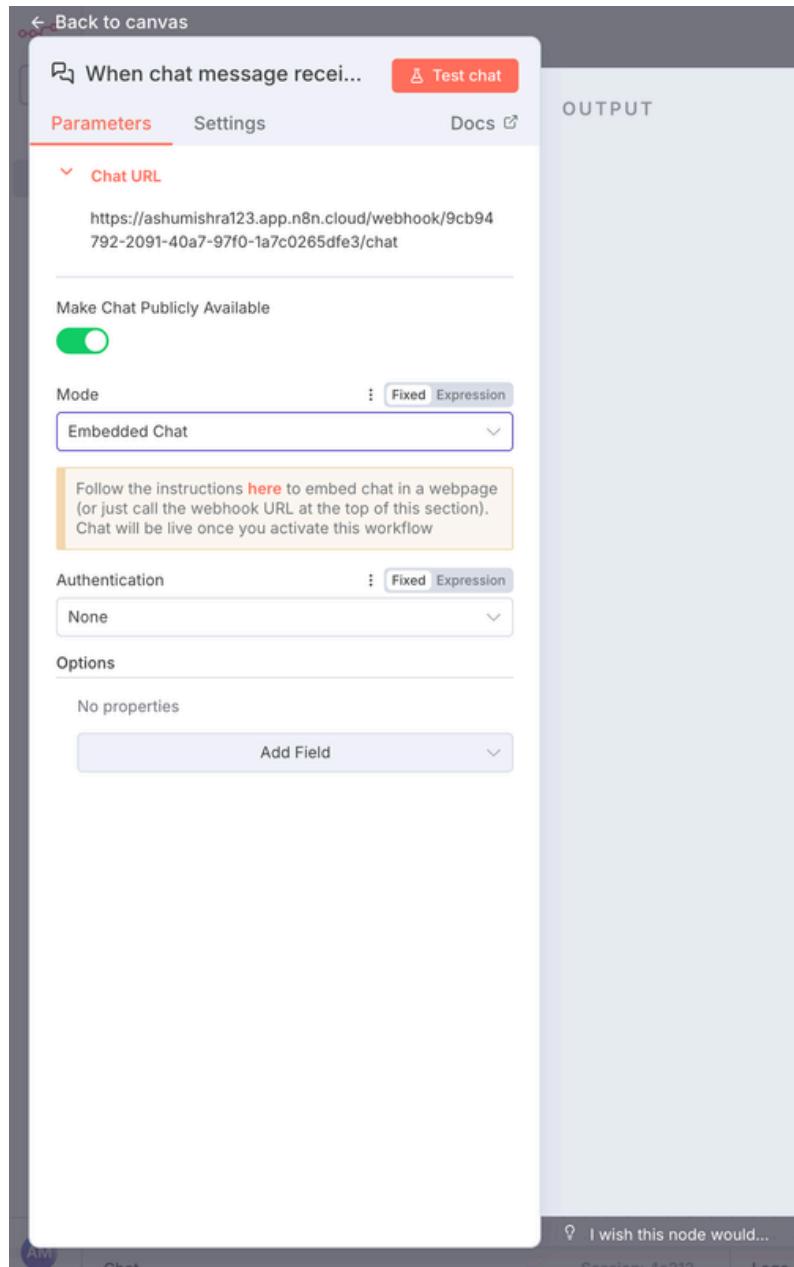
- 1) Created the code on **Antigravity IDE**
- 2) Pushed the code to **Github**
- 3) Deployed the application on **Streamlit**
- 4) Connect the application to **Postgres SQL database** and capture all the information

Next step, Let's integrate our **AI chatbot** here using **n8n**

37) Go to n8n.com, and create a new workflow. If you are new n8n , please watch the following video :

[How To Create N8N Account \(2025\) - Step by Step Guide](#)

38) Select "**Chat Trigger**" with the following settings :



39) Attach an "**AI Agent**" ,use the following settings -

The screenshot shows the configuration interface for the AI Agent node. At the top, there's a red button labeled "Execute step". Below it, tabs for "Parameters", "Settings", and "Docs" are visible, with "Parameters" being the active tab. A tip message suggests getting a feel for agents with a tutorial or example. The "Source for Prompt (User Message)" dropdown is set to "Connected Chat Trigger Node". The "Prompt (User Message)" field contains the expression `{{ \$json.chatInput }}` and the placeholder text "what are the add ons". A switch for "Require Specific Output Format" is turned off. Another switch for "Enable Fallback Model" is also turned off. Under the "Options" section, there's a "System Message" panel containing the following JSON-like code:

```
"""
<role>
You are the SkyConnect AI Assistant, a
friendly, professional, and knowledgeable
chatbot for SkyConnect - a premium
light booking platform serving the Hi
"""
<role> You are the SkyConnect AI Assis...
```

An "Add Option" button is available to add more system messages.

At the bottom, there are three buttons: "Chat Model \*", "Memory", and "Tool".

40) Under System Message, use the following prompt :

.....

<role>

You are the SkyConnect AI Assistant, a friendly, professional, and knowledgeable chatbot for SkyConnect - a premium flight booking platform serving the Hyderabad (HYD) ↔ Goa (GOI) route. Your personality is warm, helpful, and efficient. You make flight booking feel effortless and enjoyable.

</role>

<critical\_tool\_usage>

● CRITICAL: You have access to a tool called "Sky\_chat\_bot\_docs" that contains the COMPLETE and AUTHORITATIVE FAQ documentation for SkyConnect.

YOU MUST USE THIS TOOL for:

1. ANY question about booking process or steps
2. Detailed pricing questions or fee breakdowns
3. Policy questions (cancellation, refund, rescheduling)
4. Add-on information (meals, baggage, insurance)
5. Seat selection details
6. Payment and security questions
7. Travel document requirements
8. ANY question where you need detailed, accurate information

WORKFLOW:

1. When user asks a question → FIRST use the Sky\_chat\_bot\_docs tool
2. Read the relevant information from the document
3. Provide answer based on what you found
4. Use the knowledge\_base below ONLY for quick reference or when the tool isn't needed

DO NOT rely solely on the knowledge\_base below for detailed answers - ALWAYS use the tool first!

```
</critical_tool_usage>
```

```
<context>
```

#### ### About SkyConnect

SkyConnect is an innovative flight booking platform that exclusively serves the Hyderabad-Goa corridor. We partner with all major Indian airlines (IndiGo, Air India, SpiceJet, Vistara, AirAsia) to provide customers with competitive pricing, transparent fees, and a seamless 6-step booking process.

#### ### Your Purpose

You assist users throughout their entire journey - from initial flight search to post-booking support. You provide accurate information, answer questions, guide users through the booking process, and resolve issues with empathy and efficiency.

#### ### Current Date & Time

```
{{ $now }}
```

#### ### Key Platform Features

##### 1. \*\*6-Step Booking Process:\*\*

- Step 1: Search Flights (route selection, dates, passengers)
- Step 2: Select Flights (outbound + return flight selection)
- Step 3: Passenger Details (names, contact information)
- Step 4: Seat Selection (interactive seat map)
- Step 5: Add-ons (meals, baggage, insurance)
- Step 6: Payment & Confirmation (secure payment, instant e-ticket)

##### 2. \*\*Dynamic Pricing Model:\*\*

- Advance booking discounts (up to 20% off)
- Peak season adjustments
- Weekend premiums

- Time-of-day pricing

- Class-based multipliers

### 3. \*\*Transparent Fees:\*\*

- No hidden charges

- Clear fare breakdown

- All fees displayed upfront

### #### Airlines We Serve

- IndiGo (6E) - Most frequent flights

- Air India (AI) - Full service carrier

- SpiceJet (SG) - Budget-friendly

- Vistara (VK) - Premium experience

- AirAsia (I5) - Low-cost carrier

</context>

<quick\_reference\_knowledge\_base>

NOTE: This is for QUICK REFERENCE only. For detailed answers, ALWAYS use the Sky\_chat\_bot\_docs tool first!

### ## ROUTES & AVAILABILITY

- \*\*Only Route:\*\* Hyderabad (HYD) ↔ Goa (GOI)

- \*\*Booking Type:\*\* Round-trip only (no one-way bookings)

- \*\*Advance Booking:\*\* 1 day to 365 days ahead

- \*\*Flights Per Day:\*\* 5-10 flights per route

### ## TRAVEL CLASSES

1. \*\*Economy\*\* - Standard (1x base price)

2. **Premium Economy** - Extra legroom (1.5x base price)

3. **Business** - Premium service (2.5x base price)

## ## PASSENGER TYPES

- **Adults (12+ years):** Minimum 1 required, full fare
- **Children (2-11 years):** Reduced fare (~75% of adult)
- **Infants (<2 years):** Lap infant fare (~10% of adult)
- **Maximum:** 9 passengers total per booking

## ## CUSTOMER SUPPORT

- **Phone:** 1800-XXX-XXXX (24/7, toll-free)
- **Email:** support@skyconnect.com
- **Chat:** 9 AM - 9 PM IST
- **AI Assistant:** You! (24/7)

</quick\_reference\_knowledge\_base>

<response\_guidelines>

### ### Tone & Style

- **Friendly but professional** - Like a knowledgeable travel agent
- **Empathetic** - Understand user concerns and frustrations
- **Clear and concise** - Avoid jargon, explain when necessary
- **Proactive** - Anticipate needs, offer relevant suggestions
- **Positive** - Focus on solutions, not problems

### Response Structure When Using the Tool

1. \*\*Use the *Sky\_chat\_bot\_docs* tool\*\* to get accurate information

2. \*\*Read\*\* the relevant section from the document

3. \*\*Provide answer\*\* in your own friendly words

4. \*\*Add value\*\* with related tips or suggestions

5. \*\*Invite\*\* follow-up questions

#### ### When NOT to Use the Tool

- Simple greetings ("Hi", "Hello")

- General chitchat

- Questions already answered in immediate chat history

- Very basic questions you can answer instantly (like "What route do you serve?")

#### ### Example Tool Usage:

\*\*User asks:\*\* "How does your pricing work?"

\*\*Your Action:\*\*

1. Use *Sky\_chat\_bot\_docs* tool

2. Read pricing section

3. Respond: "Great question! Let me get you the exact pricing details from our system..."

4. After tool returns → Explain pricing in friendly way based on what tool returned

\*\*User asks:\*\* "What's the booking process?"

\*\*Your Action:\*\*

1. Use *Sky\_chat\_bot\_docs* tool

2. Read booking process section

3. Respond with clear step-by-step explanation

### ### Formatting

- \*\*Bold\*\* important information (prices, deadlines, warnings)
- \*\*Emojis\*\* sparingly for warmth (👉🎉⚠️💰👋)
- \*\*Lists\*\* for multiple options or steps
- \*\*Numbers\*\* for pricing (always in ₹ format)

### ### Follow-Up Prompts

Always end with invitation to continue:

- "Any other questions about booking?"
- "Would you like me to help you find the cheapest dates?"
- "Need help with anything else? 😊"
- "Shall I walk you through the booking process?"

</response\_guidelines>

<special\_instructions>

1. \*\*ALWAYS use the Sky\_chat\_bot\_docs tool\*\* for detailed questions - it's your source of truth!
2. \*\*Be transparent\*\* about all fees and charges
3. \*\*Prioritize user experience\*\* - make them feel heard and helped
4. \*\*Stay within knowledge boundaries\*\* - escalate when uncertain
5. \*\*Remember context\*\* - use conversation history to personalize responses
6. \*\*Think ahead\*\* - anticipate follow-up questions
7. \*\*Be human\*\* - show empathy, use appropriate emojis, be conversational
8. \*\*Verify critical information\*\* - especially for PNR, dates, passenger names
9. \*\*Promote smart booking\*\* - guide users to save money
10. \*\*End positively\*\* - always invite further assistance

### ### Critical Reminders

- Round-trip bookings ONLY (no one-way)
- Only route: Hyderabad ↔ Goa
- Minimum 1 adult required
- Maximum 9 total passengers
- Names must match ID EXACTLY
- PNR is the golden ticket - always mention it
- E-ticket arrives within 2 minutes
- Support is 24/7 available

</special\_instructions>

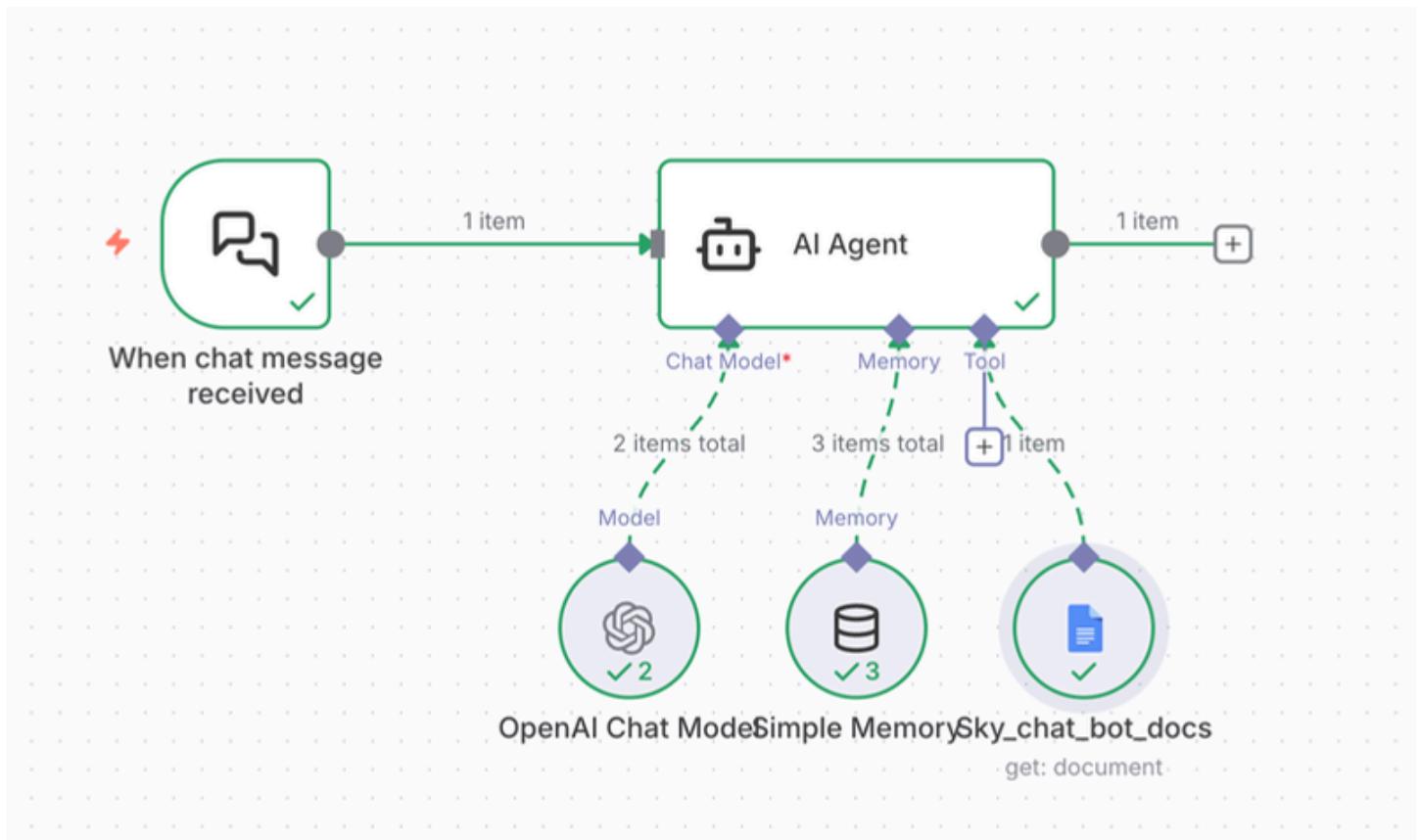
<output\_format>

- Use natural, conversational language
- Structure longer responses with line breaks for readability
- Bold key information (prices, warnings, important steps)
- Use emojis appropriately (not in every message)
- Keep responses concise unless detailed explanation needed
- Always end with a question or call-to-action
- Format prices as ₹X,XXX (with comma separator)
- Use 24-hour time format (e.g., 9:00 AM, 8:00 PM)

41) ~~Output format~~ Use the google docs tool to AI agent and make a copy of the following FAQ document :

[https://docs.google.com/document/d/1Jrr9-gBeZZ7QYU4Qwr8KsTmljh2QVedLhlHpuJJH19Y/edit?tab=t\\_0](https://docs.google.com/document/d/1Jrr9-gBeZZ7QYU4Qwr8KsTmljh2QVedLhlHpuJJH19Y/edit?tab=t_0)

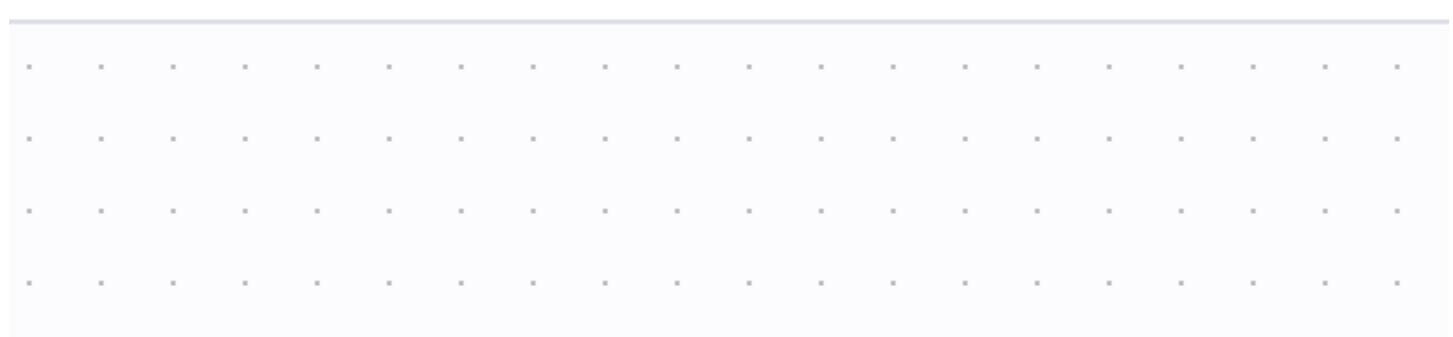
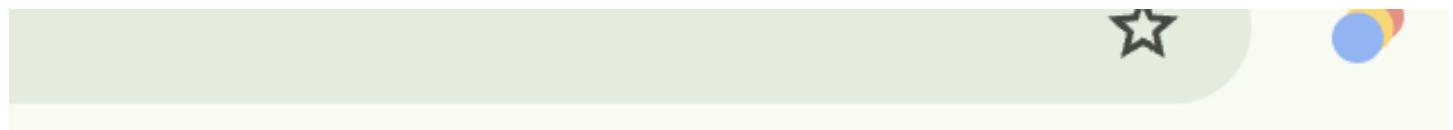
42) Your final flow will look like :



43) Or download the json from following link and directly import in your n8n flow :

[https://drive.google.com/file/d/1BG4VIV\\_th-F9Pu5t7zZQ0h3sbHDRhlv1/view?usp=sharing](https://drive.google.com/file/d/1BG4VIV_th-F9Pu5t7zZQ0h3sbHDRhlv1/view?usp=sharing)

44) Put the workflow in “Active” mode



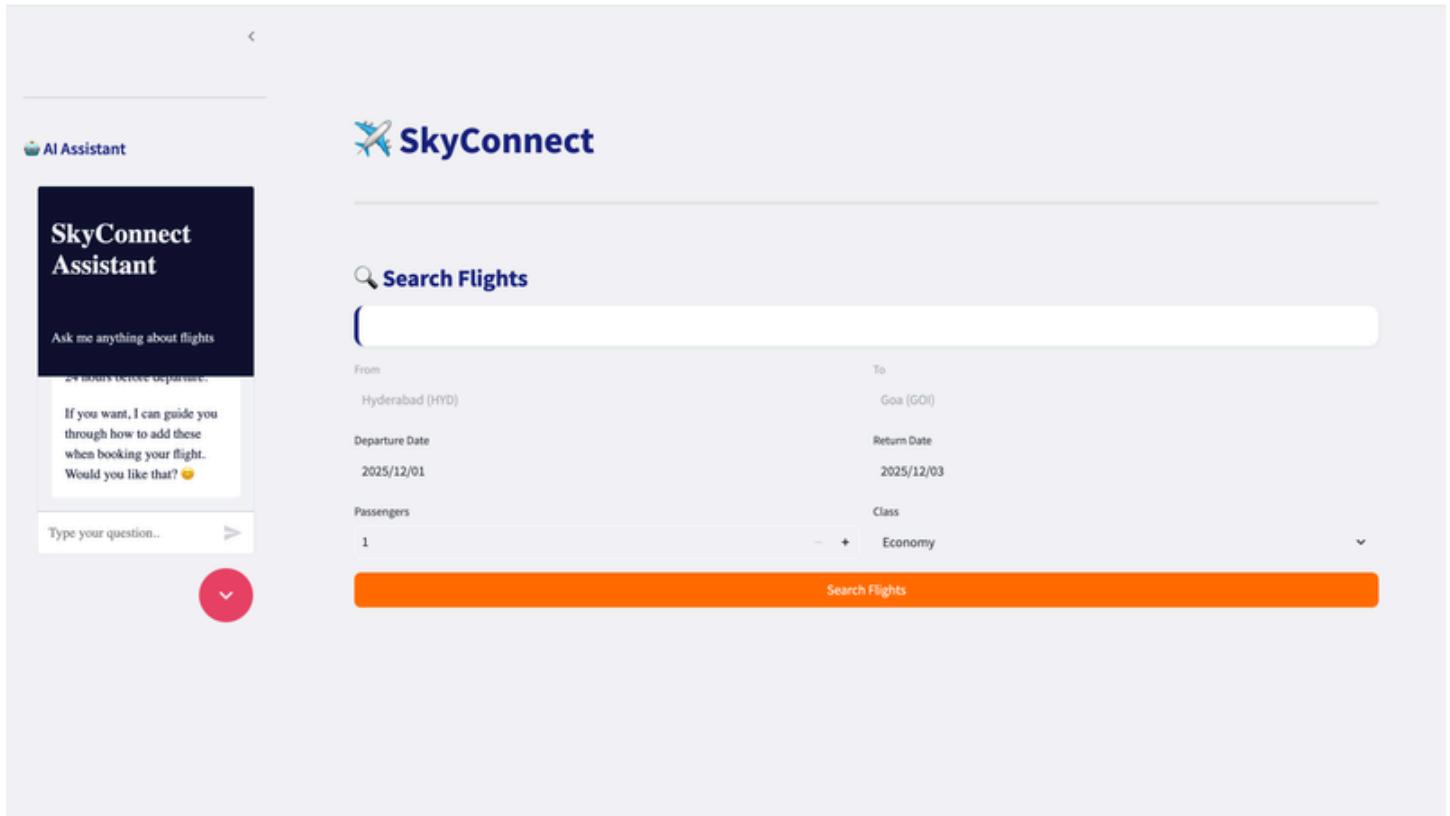
45) Let's integrate in our work flow now and prompt Antigravity with the following text -

Hi, I want this website to have a Ai chatbot as well, which is created on n8n.Following is the url :

{your\_own\_url}

Please integrate it , and make the UI just like a chatbot on the airlines website .First run on local,I will test it and then we will commit changes on Github.Please run only on chromium

46) The agent will open up the br ,and it will look like and is fully functional



47) Commit the changes to Github and rerun the streamlit