# Prompt for Creating a Fully Interactive Take-Home Pay Comparison Streamlit App

Create a Streamlit-based interactive Python application that compares take-home pay for workers across U.S. cities, based on key user-adjustable financial parameters.  
  
Core Functionalities:  
1. Load a CSV of U.S. cities including columns: City, Latitude, Longitude.  
2. Ask the user for:  
 - Annual Salary (Slider)  
 - Gas Price per Gallon (Slider)  
 - Vehicle MPG (Slider)  
 - Annual Car Payment (Number)  
 - Annual Car Insurance (Number)  
 - Annual Healthcare Premiums (Number)  
 - Federal and State Tax Sliders  
3. Include commute logic:  
 - Sidebar checkbox controls for each weekday (Monday through Friday)  
 - Commute Miles and Gas Costs are proportional to the number of days checked (e.g., 3 days = 3/5 of cost).  
4. Allow user to:  
 - Enter a home address (geocoded to coordinates)  
 - Add and remove cities dynamically  
 - Add custom labeled locations in "Label, Address" format  
5. Recalculate and display:  
 - Distance from home to each city  
 - Annual Commute Miles  
 - Annual Gas Cost  
 - Federal Tax, State Tax, FICA, Car Payment, Car Insurance, Healthcare  
 - Final Net Pay  
6. Use the `geopy` package for geocoding and distance calculations.  
7. Visualize results:  
 - A Leaflet map with city markers using `folium`  
 - Popups displaying City name and Net Pay  
8. Present an interactive DataFrame showing all calculated columns.  
9. Use caching and geocode resilience (retry if timeout).  
10. Ensure the `selected\_days` from checkboxes dynamically updates calculations.  
  
Implementation Requirements:  
- Use lambda or closure functions to pass dynamic commute-day values into the financial calculation function.  
- Ensure the function `calculate\_financials(row, selected\_days)` accepts dynamic commute schedules.  
- Filter cities by max commute distance using a slider.  
- Color-code map markers (e.g., green for "Remote").  
  
Bonus:  
- Enable download of the final comparison table as CSV.  
- Include support for script execution via .command file on Mac.

# LinkedIn Article: Building a Geographic Take-Home Pay Analyzer with Streamlit

Today’s workforce is more mobile than ever. With hybrid and remote work models redefining how we think about jobs and geography, I wanted a tool that could answer a very practical question:  
  
💡 “Where can I work and still keep the most of my paycheck?”  
  
To answer this, I built an interactive \*\*Take-Home Pay Comparison App\*\* using \*\*Python\*\*, \*\*Streamlit\*\*, and \*\*Folium\*\*.  
  
Here’s what it does:  
📍 Accepts any U.S. address as a home base   
📊 Lets users enter salary, taxes, MPG, car costs, and health premiums   
📅 Uses commute checkboxes to reflect real hybrid schedules   
🗺️ Maps every city in a 200-mile radius with distance and pay calculations   
🧮 Shows net pay for each city after accounting for taxes, gas, car, and healthcare   
✍️ Allows adding/removing cities or even custom-labeled addresses like “My Office”   
📌 Visualizes everything on an interactive map + table  
  
What makes it powerful is how realistic it is:  
- It factors in commute days and cost scaling  
- It's geocode-resilient — even handling failed address lookups gracefully  
- It supports remote work scenarios by comparing against a “Remote” baseline  
  
For job seekers, relocators, or employers evaluating talent markets — this tool is a game changer.  
  
Built using:  
- Streamlit  
- Geopy  
- Folium  
- Pandas  
  
Want to try it? DM me for access or see the GitHub project [Link Optional].  
  
#python #streamlit #dataanalysis #remotework #careerplanning #geospatial #opensource #financialtools