Assessment Task – Cities of the World

You are tasked with building an HTTP API for a website for geography enthusiasts. The goal is to allow users to query historic population data for various cities of the world.

We have prepared three tasks that build upon each other. They are meant to be solved in order.

What we look for in your solution:

- Your solution should include tests.
- Your solution should consider errors and edge cases.
- The code must be under version control (ideally git). Please commit at least after every task so it is possible for us to follow the evolution of your code between the tasks.
- If you want to deviate from any specifications feel free to do so, if you can explain the reasons for those decisions.

Available Datasets

You have three datasets available:

- populations: contains population numbers by year for various cities across the world.
- geonames_coordinates: links Geoname IDs to their coordinates
- cities_geonames: relational table between cities and their Geoname IDs

For tasks 1 and 2 you will only need the populations dataset.

Task 1 – Query Population by City and Year

The first requirement is that our API must handle requests containing a city name and optionally a year and respond the population of that city for the given year. If no year is provided, the latest data must be returned.

Request	Expected Response
`{ "cityName": "München" }`	`{ "population": 1388308 }`
`{ "cityName": "München", "year": 1999 }`	`{ "population": 1194560 }`

Task 2 – Exclude Estimates

After testing the results of task 1, some of our key users requested to not see any provisional data. As you can see for example in the 2011 data for Cairns there are rows where the "Reliability" is not "Final", but "Provisional".
Add a flag to the API that allows the client to ask for latest data only.

Request	Expected Response
{ "cityName": "Cairns", "year": 2011, "includeProvisional": true }	{ "population": 139693 }
{ "cityName": "Cairns", "year": 2011, "includeProvisional": false }	{ "population": 133911 }

Task 3 – Include Geographic Coordinates

We're adding a new feature to the website that displays the city on a world map. For this, our frontend team needs the geographic coordinates as latitude and longitude.

Tap into the datasets "cities_geonames" and "geonames" to get the latitude and longitude values.

Request	Expected Response
{ "cityName": "Cairns", "year": 2011, "includeProvisional": true }	{ "population": 139693, "latitude": -16.92366, "longitude": 145.76613 }

Finally, think about how to make the application as performant as possible. You do not need to implement all your ideas. It is sufficient to put some notes in a text file outlining possible optimizations.

Dataset Sources

geonames: https://public.opendatasoft.com/explore/dataset/geonames-all-cities-with-a-population-1000/information/?disjunctive.cou_name_en&sort=name

populations: https://datahub.io/core/population-city

cities: https://datahub.io/core/world-cities