In your own words, explain the problem and how this application solves it.

Kathy is underwriter and her job is decide the dynamic price of policies based on different parameters. But parameters are different for different geo location and due to these factors pricing of the policy can vary from place to place.

To decide the pricing policy Kathy wants a tool where she can enter the address/zip code and on the based on that address our platform will show all the possible risks for that particular location and a intuitive map from where she can confirm the place and nearby of that place to re confirm the place and and risk as this data is very crucial.

How long did it take you to complete this exercise?

Roughly 8-10 hours which includes problem understanding, requirement gathering, development and basic level of sanity and UT check.

Did you run into any problems or challenges while working on it? If so, elaborate.

Yes, few of the are:

1. Initial level of understanding related to mock Api endpoint was not clear to me.
2. Learning of Open street Map(leaflet). It is new to me.
3. How to make this structure scalable as I was not known about the nuances of Api and end user usage.

* **Why did you choose to use these libraries or frameworks (if any)?**

I have chosen not to choose any framework due to these points:

1. It's a small application so including framework will over kill this.
2. No state management was required.
3. Don't want to make the code size on a higher side which is not required in this scenario.
4. Do not want make any dependency on any third party library as it is not required.

*** What state do you feel the application is in? Are you ready to ship it or are there additional changes you'd want to make?**

Application is in good shape as per the requirement given but not the ideal state as we are dealing with mocks and complete use cases were not known for me. I want to ship it after catering below given feature and requirements:

1. Actual Api integration
2. If error occurred via Api (both).
3. If address not found by Kathy.
4. If lat and long was wrong.
5. Kathy can choose address from map also or play with map also as sometimes lat and long are not correct and might be Kathy wants to reverify the region by moving marker on map.
6. We can provide different color on map based on risks to be more clear.
7. Label on map also for better visibility.
8. Cache few data, so that we can minimize the api call.

9. May be a button to copy the data to clipboard from the risk list.
10. Responsive UI for mobile.
11. Maintain a table in the web page which shows address an risk and Kathy can simply copy all the location with risk in one go instead of copy again and again.
12. User manual of some labels on user input.
13. Authorization of this page with SSO/Login in as the data is very sensitive and we want to keep it secure.
14. Feedback form over there to submit any data discrepancy from Kathy.
15. Telemetry to log user events and log if any error occurred.
16. Unit test cases and end to end testing if required.