COVID-19 saver

A monitoring system which will alert users if they were exposed to any infected person and let them know to take precautions :)

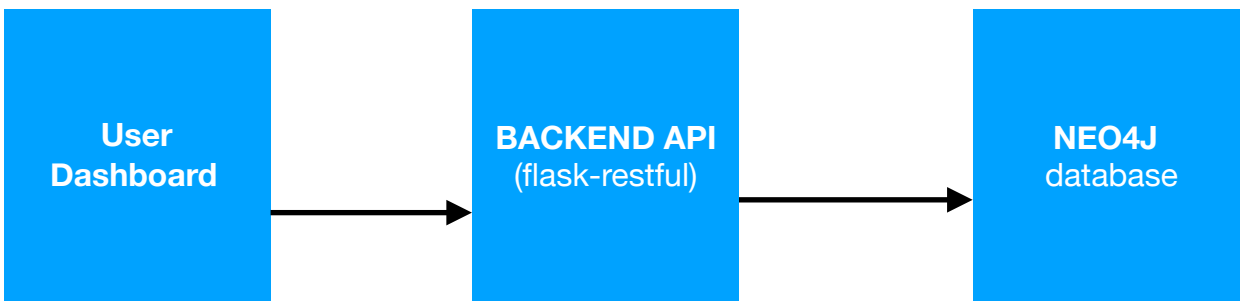
Technologies need to be used:-

1. *Flask*(Python framework) for building APIs
2. *Neo4j*(Database)
3. *Vue.js(for admin dashboard)*

Now let's try to understand the **dashboard** a bit:-

- *User can register using only his email address(email address should be validated) and country of residence.*
  - *Once user registration is successful an API token would be generated(randomly generated 64 character string) which would be stored in database.*
- *Whenever user logs into the portal get the Geo Location using browser location API of the user and store it in Neo4j as a visited place by that user.*
  - *After Login is successful user gets his API token which was generated after registration step.*
  - *It gets stored in front end and user passes this API token to authenticate himself with every request.*
- *A user can mark himself/herself as infected anytime.*
- *Users can see their own reports by logging into the dashboard(for simplicity let's say for now password is hard-coded to **tribes-covid-fighter**), Below reports would be required:-*
  - *People infected around his current location with the distance*
  - *List of the users he got exposed to who were infected with covid.*
  - *User can enter a geo lat/long code and check If the place he is planning to visit has been visited by any corona infected person in last 4 hours.*
  - ***OPTIONAL**:- Show current count of infected people in the country user is registered in(you can use https://www.bing.com/covid/graphdata api)*
- *User can logout himself.*

Below is the architecture design of the system in my mind, feel free to improvise on it:-



NOTE:-

1. All data transfer happens only through backend api and dashboard can't directly talk to NEO4J database.
2. For front-end if needed you are welcome to use any dashboard template based on vue.js.

After user logs in an API token would be returned which would be passed to user dashboard back and for any further communication dashboard should pass back this token to the backend API. Without access token backend API should return an error and return data only if token is valid.

Evaluation Criteria:-

1. Best coding practices(naming conventions, comments etc)
2. Completeness
3. Time taken to complete the assignment
4. Effective use of git(frequent commits and proper commit messages)
5. Proactiveness in communication during the entire assignment(if want to add anything you are welcome)