

Final_project_proposal

Salma Elshahawy

10/13/2020

Contents

Introduction	1
Project Objectives	1
Dashboard Features	1
The data source used for that project	2
The Tech Stack	2

Introduction

Covid-19 had changed our personal lives dramatically. One of the significant changes within the lifestyle is the way of how we keep ourselves safe. The Centers for Disease Control and Prevention - CDC had released plenty of hygiene guidelines people should follow to be safe. It also declared that wearing a cloth mask helps decrease the infection chances in the closed areas. Additionally, it said that people with underlying medical complications or fall into specific age categories would be at high risk of contracting the virus. A lot of data was released per state per zip code areas, counting the number of cases per age category. To keep track of the cases, I decided to create an interactive dashboard tracking the number of cases in the US per zip code.

Project Objectives

It is hard to navigate to any public news website to know the latest number of confirmed cases in a specific state. For that purpose, an interactive dashboard would be a good alternative, where it is easy to read without reading in long articles.

Dashboard Features

To design a well communicative dashboard, we need to identify the major features we may include. Some of the preliminary features would be:

1. Header, footer, and body.
2. The number of confirmed cases, deaths, and tests done.

3. An interactive map showing the confirmed cases per state.
 4. A list of the states with their updated number of fatalities.
-

The data source used for that project

For this project, I would rely on the data that most news used from the Johns Hopkins University Center for Systems Science and Engineering. I intend to use the COVIDTracking repository, where it has the features I need to build the dashboard.

The Tech Stack

To accomplish the project's goals, I would prefer to use Python as a back-end language to clean up and prepare the visualization data. I would use the Flask/Dash-render combined with some react components to make the rendering quick. I would also make use of MongoDB to manage the back-end when hosted to Heroku.