## horizontal line



Death Rate Finder!

INF551 Project Report

Developed by -

Ashwini Giri

USC ID: 5413882039

Isha Patil

USC ID: 6634971288

## 

## Instructor

Wensheng Wu

# Introduction

# Death Rate Finder is a search based web application which enable the users to search for the death rate, cause of death as well as year of death in the states of America. The project handles a real-world death rate dataset from the government website. By using this web application the user will be able to know about the death rate in particular state as well as the cause of the death. The application provides the users with login and signup authentication so that they can save their search history.

# Motivation

Scientists, researchers and medical practitioners use American Community Survey statistics to simulate the spread of disease, allowing decision-makers to prepare for the next potential outbreak. They have to collect stats, reports and surveys to get the data for their research. The research may include study of Epidemical disease, leading causes of deaths etc. The medical researchers, doctors and health sciences department then can analyze the statistics and streamline their research to find the remedy and drugs for the leading causes of death. After keeping all this in mind we thought of helping these people to speed up their research. We developed a web application that will help users to find the leading cause of death, death rate and analyze the data in a fraction of seconds.

# Goals

1. Analyzing the present data from government data, pre-processing and cleaning the dataset (like handling missing values, normalizing and featuring data). Converting the data in json format and storing in cloud storage like firebase for maximum availability and quick response times.
2. To build a web application that stores a real-world database downloaded from government website on firebase for the users to search through Death Rate based on many search criterions picked from the attributes of the data set like cause of death.
3. Implement keyword search interface which returns results based on the degree of relevancy with search.
4. Implementing faceted search with three facets.
5. To implement login functionality and save search which allows users to save past searches.

# Application architecture

The application contains 3 major components

1. The user interface
2. API
3. Database storage

## **Data structure**

**Data Source:**

The real-world data set used in the application is fetched from: [*https://catalog.data.gov/dataset/age-adjusted-death-rates-for-the-top-10-leading-causes- of-death-united-states-2013*](https://catalog.data.gov/dataset/age-adjusted-death-rates-for-the-top-10-leading-causes-)

**DataSet used: NCHS – Leading causes of death in united states.**

****

**Details:**

Number of columns: 6 Number of evidences: 15029

* The data set contains age-adjusted death rates for the 10 leading cases of death in USA beginning in 1999.
* The data comes from resident death certificates filed in 50 states and the district of Columbia using demographic and medical characteristics.
* The Column age adjusted death rates specifies death rates based on the 2000 US population(per 1 lac population)
* The column year specifies the year of death
* Column cause name specifies the cause of the death and column state specifies the state in which death was filed.
* The data is recorded from year 1999 to 2015. The data will be updated every census calculation.

**Data Storage:**

The data is stored on firebase since it gives a basic and unified platform with many Google features packed-in making the development process easier and more efficient by providing real-time database, cloud storage, authentication etc.

**Data Structure:**

Building a properly structured database requires quite a bit of forethought. Most importantly, planning for how data is going to be saved and later retrieved to make that process as easy as possible.

The data is stored in JSON format on firebase realtime database. Unlike a SQL database, there are no tables or records. When we add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key.

We have created an inverted index on the keyword search attribute ‘State of death’ for efficient and easy querying.

## **Implementation Details**

We have implemented the applications by breaking down the complete projects into below phases:

1. **Data preparation and Analysis:**

The data was available in CSV format, the cloud data store that we planned to use stores data in JSON format. Hence we converted the data from CSV to JSON format. Initial analysis of data like choosing the attributes to use for keyword search and faceted search was done in this face.

**Attributes of data:**

**Facets:**

1. Year
2. Cause name
3. Number of deaths

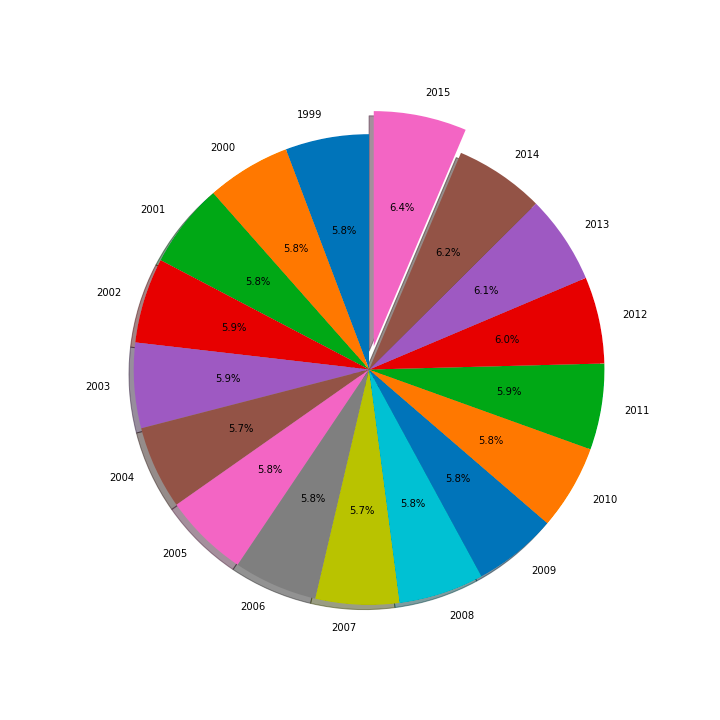
**Keyword search:**

1. State of death occurrence

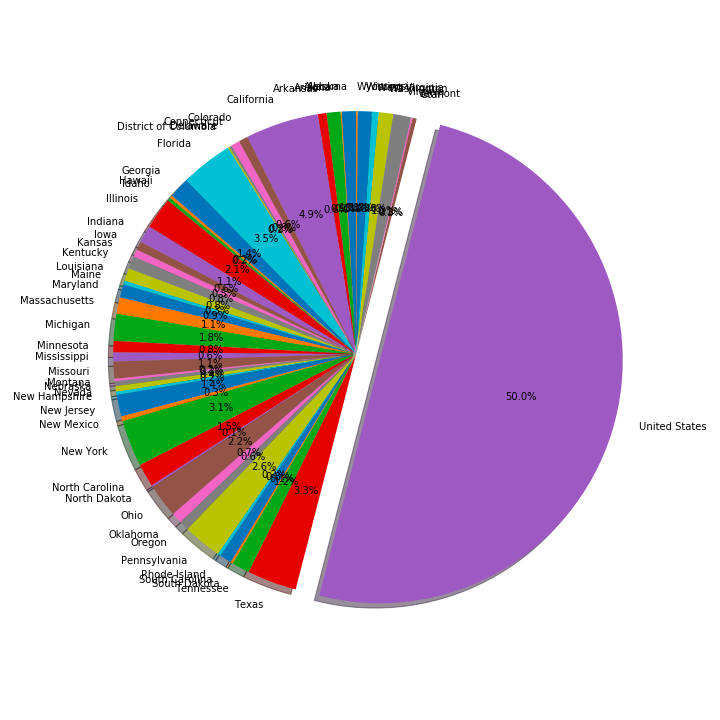
**Remaining attributes:**

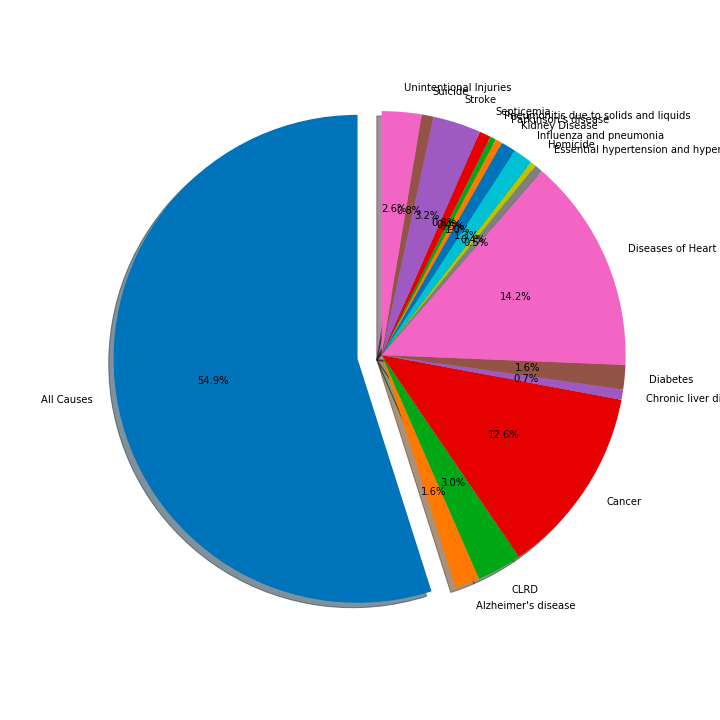
1. Age adjusted death rate
2. Scientific cause name

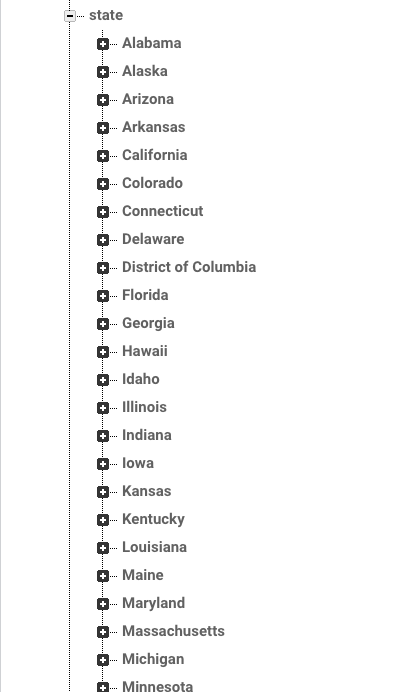
We have also analysed our data using python and plotted some inferences from the data using python’s library matplotlib.



The plots are based on year of death, cause of death and the state of death. The highlighted portion of the pie chart indicated the highest number of deaths for the respective attribute used to plot the graph.





1. **Data storage:**

This was the part where we needed lot of thoughts. We have to think how we can make the search quick and easy to implement. Hence, we have created and inverted index on the keyword search attribute ‘state of death’ and stored the data on firebase realtime database.

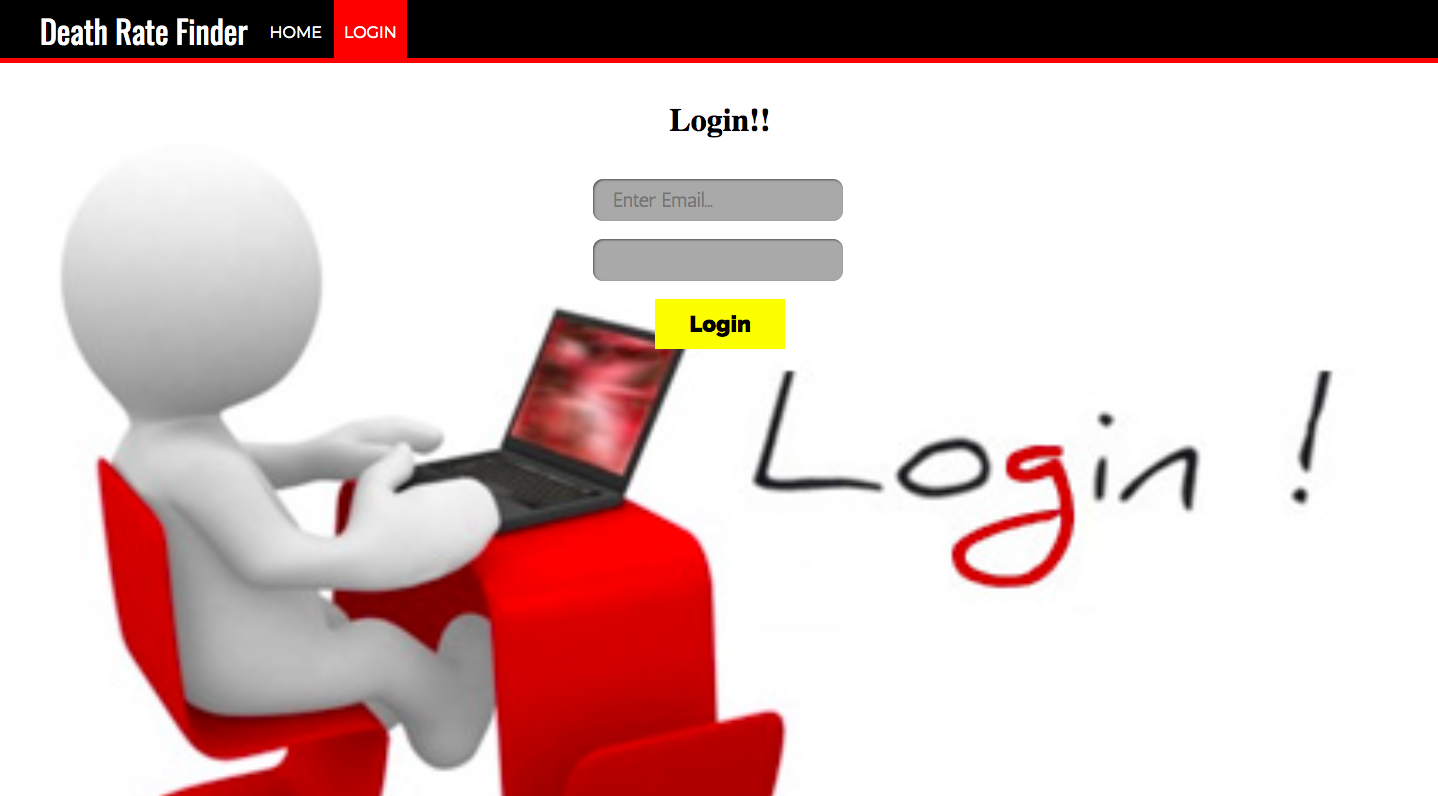
1. **Creating User Interface:**

We created the user interface using HTML and CSS. We have created five web pages for our project as below:

1. Home Page



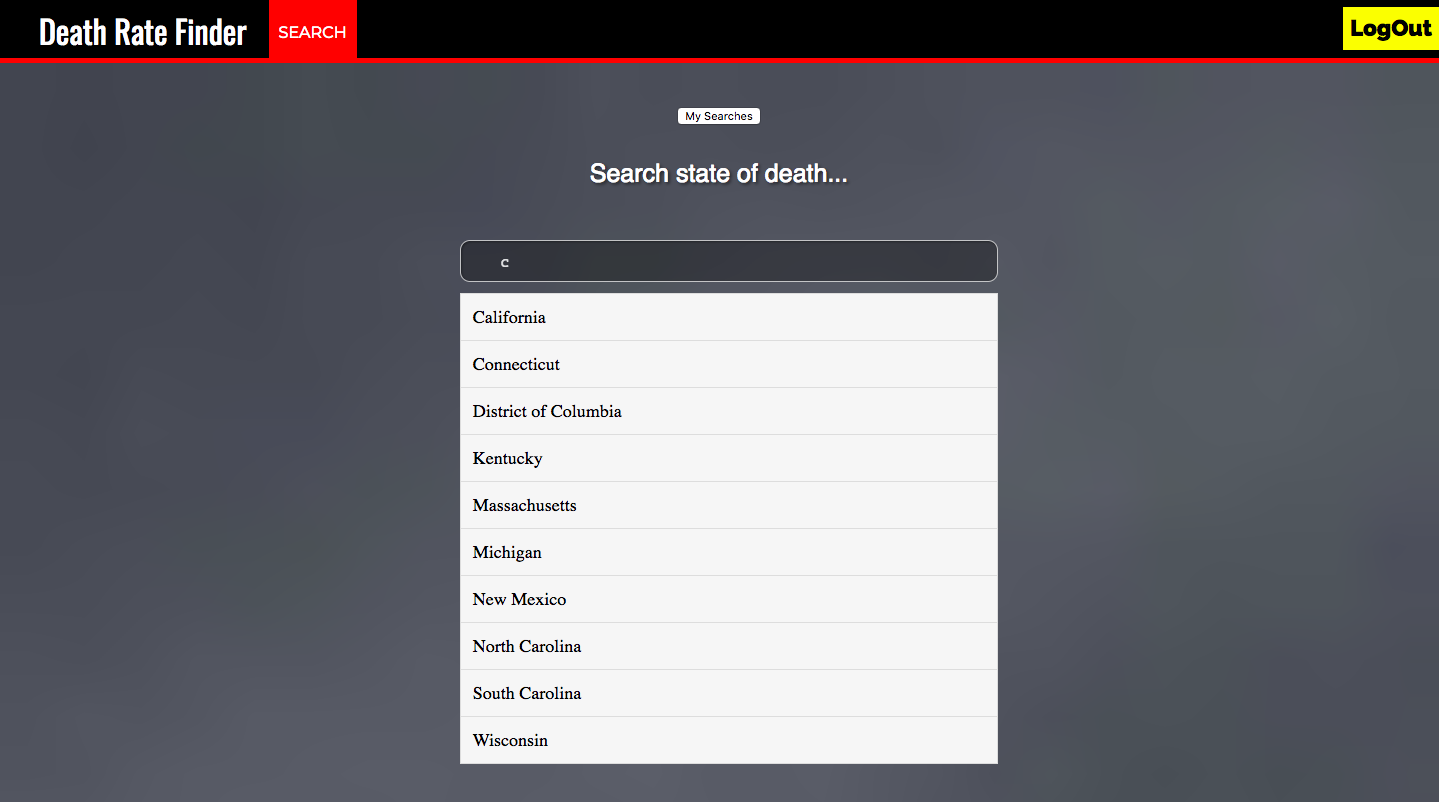
1. Login Page



1. Sign Up Page



1. Search Page



1. Contact Us Page



1. **Implementing Keyword search:**
2. **Implementing Faceted search:**
3. **Implementing User authentication:**
4. **Implementing Save search:**
5. **Implementing Contact Us:**

**Documentation of the code**

Programming Languages used:

1. Python: data transformation from csv to json. Creating inverted index on state attribute for easy keyword search. Restful API to upload the data to firebase. Analysis of data and plotting graphs.

We have used python libraries like pandas for reading the CSV file and converting it to JSON data. Matplotlib for plotting the pie charts which helped in easy visualization of data analysis.

1. HTML: The static web pages were created using HTML.
2. CSS: The styling of the web pages was done using the CSS properties.
3. JavaScript: Interaction of firebase and the web pages was implemented using JavaScript. Also the keyword search, database querying,

## **Advantages and Disadvantages of Firebase**

## **Advantages:**

* Since it is a cloud database, it is readily available.
* Login and authentication is easily implemented since it is already available.
* Firebase uses JSON storage ; no barrier between data and objects.
* Minimal setup.
* Massive storage potential.
* Highly secure since it is covered by Google.

## **Disadvantages:**

* Very limited querying and indexing.
* No aggregation
* No map-reduce
* Cannot query or list users or stored files.

# Work of each group member

Each member of the group have implemented the below functionalities:

**Ashwini:**

1. **Data preparation:** converting the data from csv to json format. Analyzing the data in python and plotting pie charts for visualization using python.
2. **Data Storage:** Creating inverted index on the keyword attribute for easy search.
3. **Creating web pages:** The home page, contact us page and search page with filters using HTML and CSS
4. Keyword Search: Implemented keyword search using JavaScript and firebase database querying
5. Faceted Search: Implemented faceted search using the filters and the search data in Javascript. This required a lot of querying to the firebase database.
6. Save Search: Implemented the save search functionality using JavaScript and stored the searches on firebase.
7. Contact Us: stored the data of contact us functionality on firebase and

**Isha:**

1. **Login:** Created the web page for login using HTML and CSS. Implemented the login function using firebase email authentication.
2. **Sign Up:** Created the web page for sign-up using HTML and CSS. Implemented the sign-up function using firebase email authentication.
3. **Project proposal report:** Created the project proposal draft.
4. **Mid-term Presentation slides:** Created the mid term presentation.
5. **Final project presentation slides:** Created the end term presentation.
6. **Final Report:**