

**Mid-Term Report**

# DEATH RATE FINDER

**Introduction:**

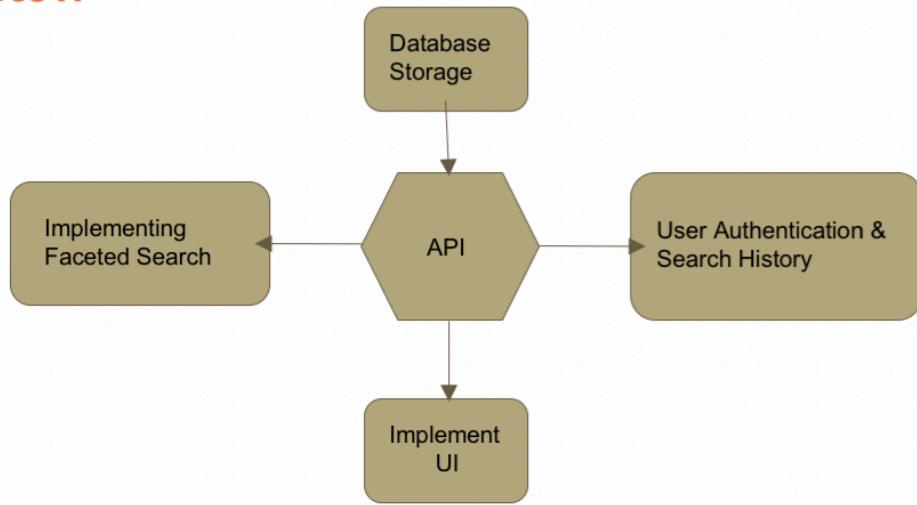
Death Rate Finder is a web application which will enable the users to search for the Age adjusted death rate, cause of death, year of death and the State of death in the United States of America. The project handles a real-world death rate dataset from the government website.

**Functionalities of the Application:**

1. User Login
2. Faceted Search
3. Save User Search history

**Project Workflow:**

## Workflow

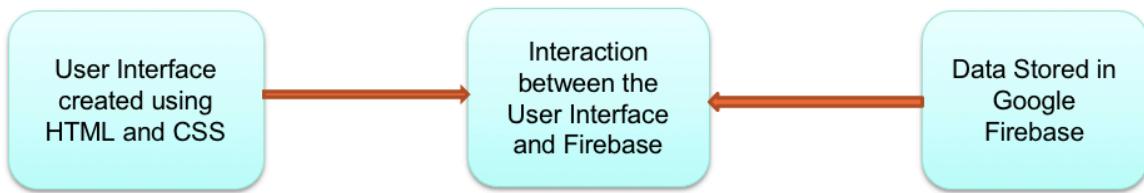


The workflow of our project consists of three major implementation layers:

1. Creating User Interface using HTML and CSS.
2. Data cleansing, preprocessing and storing in the database. We are using Firebase to store the data.
3. Interaction between the user interface and the backend database stored on firebase. We have planned to implement the interaction using JavaScript. Implementation of faceted search and save search for user functionalities.

## Mid-Term Report

### Project Design and Architecture:



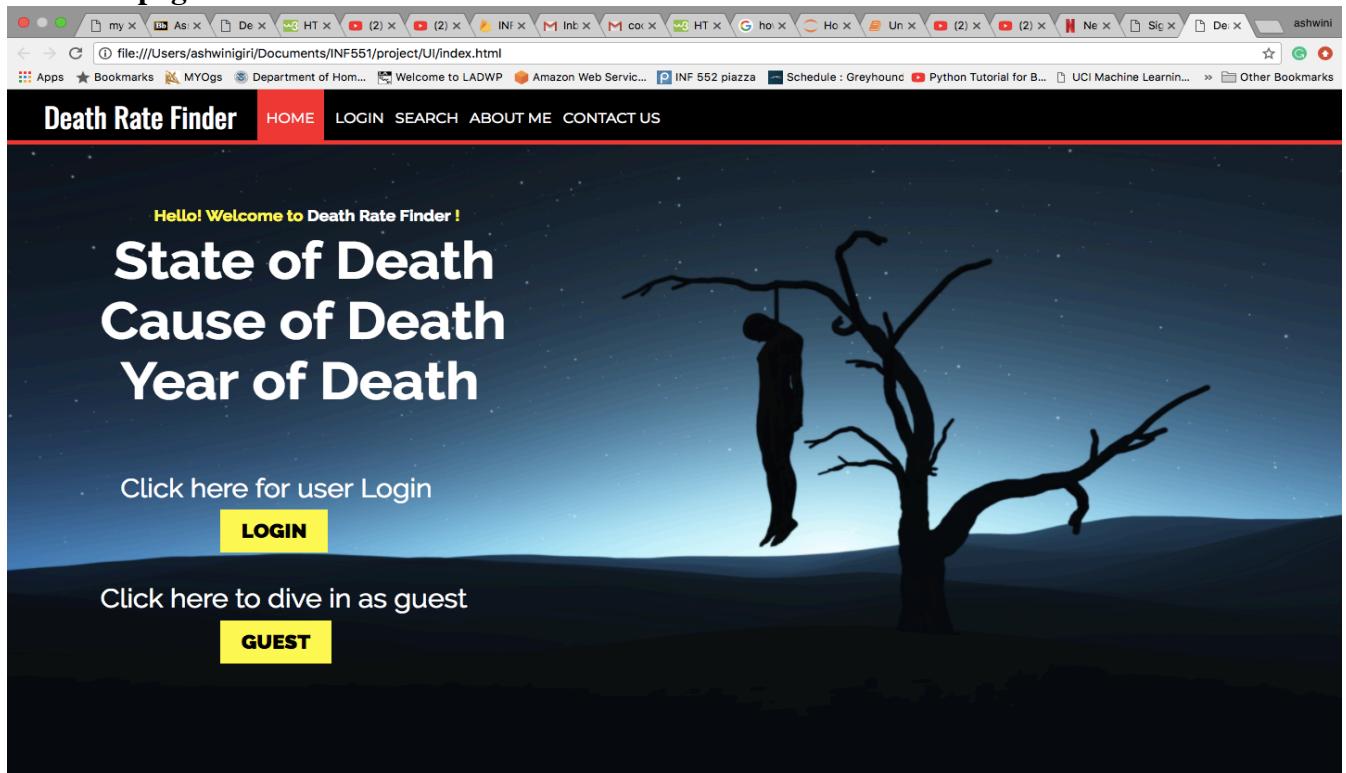
We decided to complete the project in three major steps which involves data storage, User Interface Implementation and building interaction between the database and User Interface. The first phase that we completed is data preparation and storing it on Firebase. For this we learned about firebase datastore, cloud storage, Json data.

In the second phase we started with the implementation of User Interface. For this we learned working with HTML and CSS.

We have created four web pages namely Homepage, Search page, Login page, and Contact us page for the User Interface.

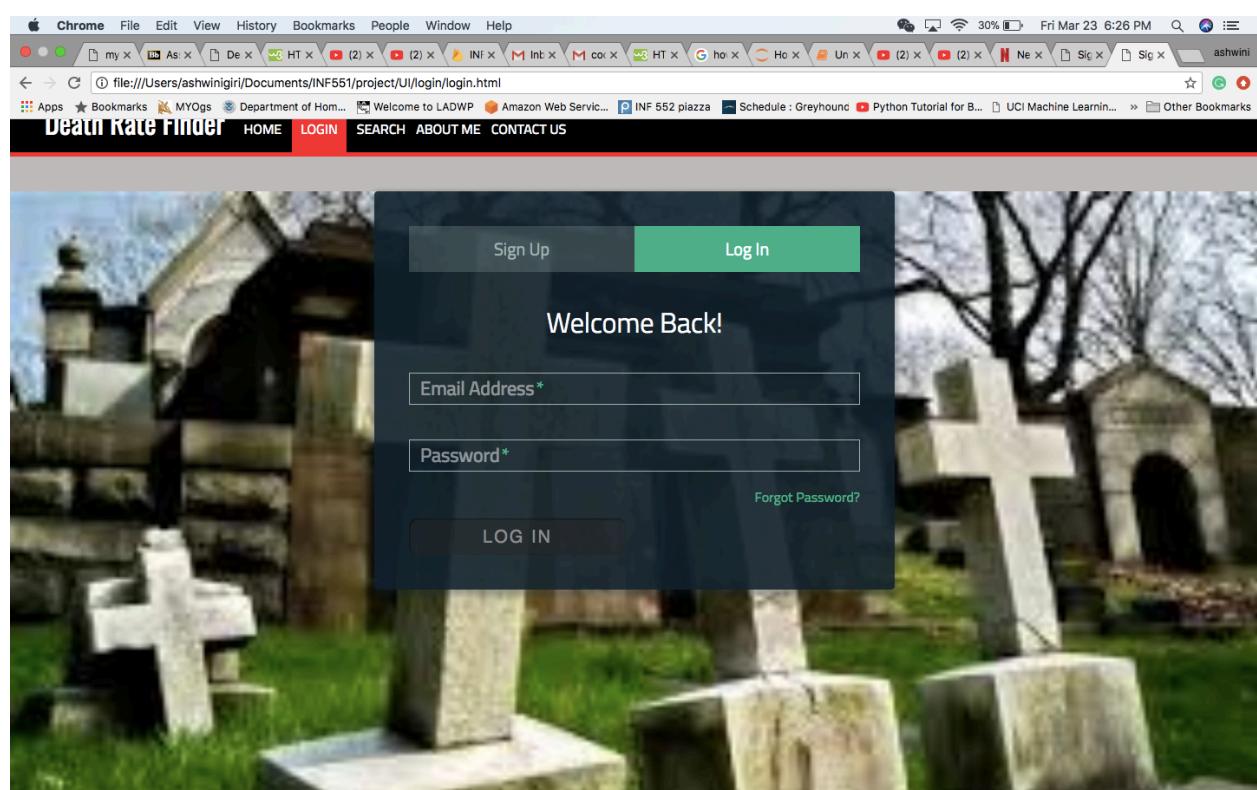
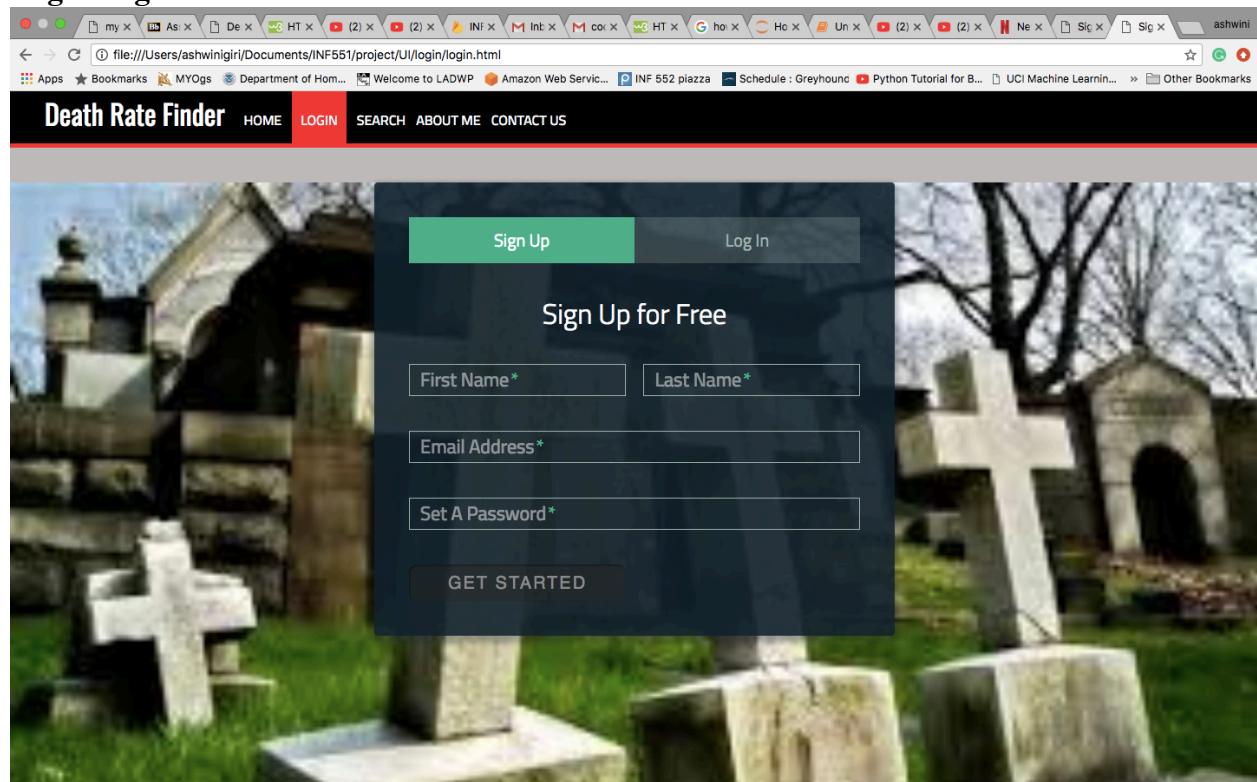
### User Interface

#### Homepage



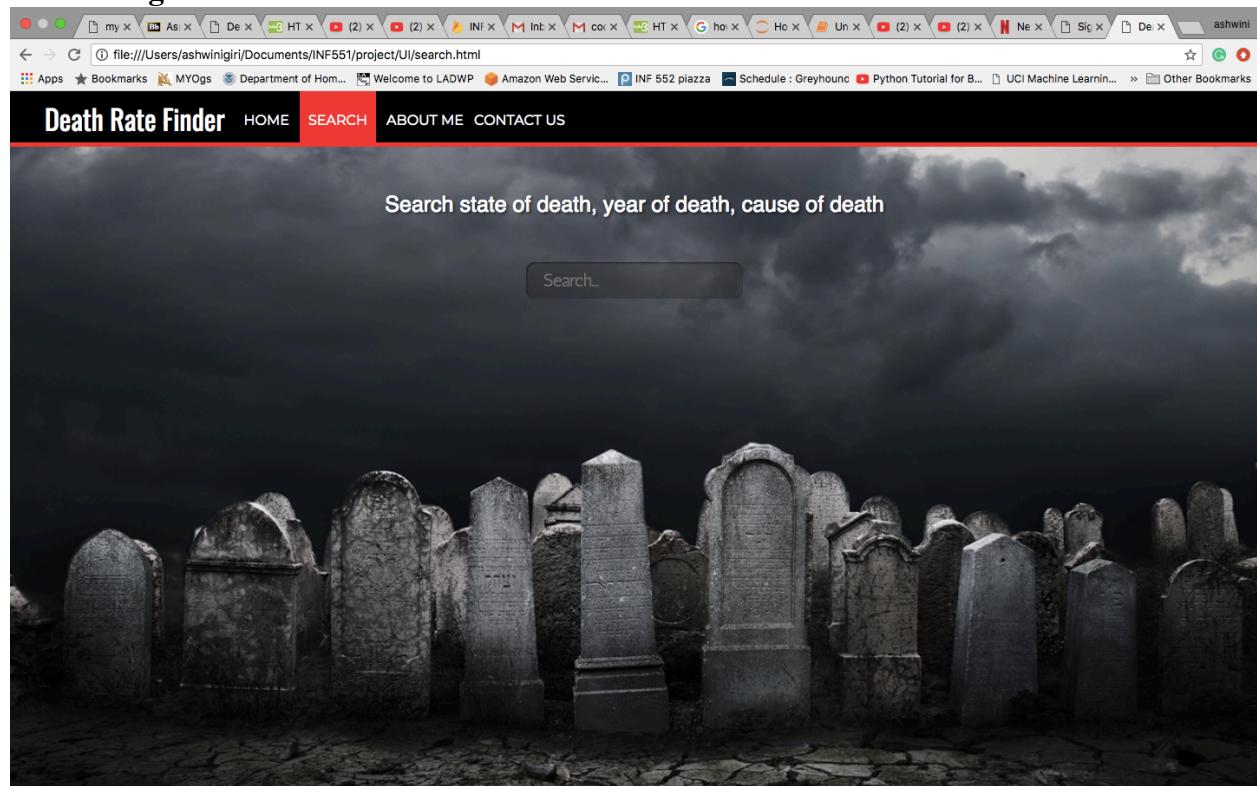
# Mid-Term Report

## Login Page

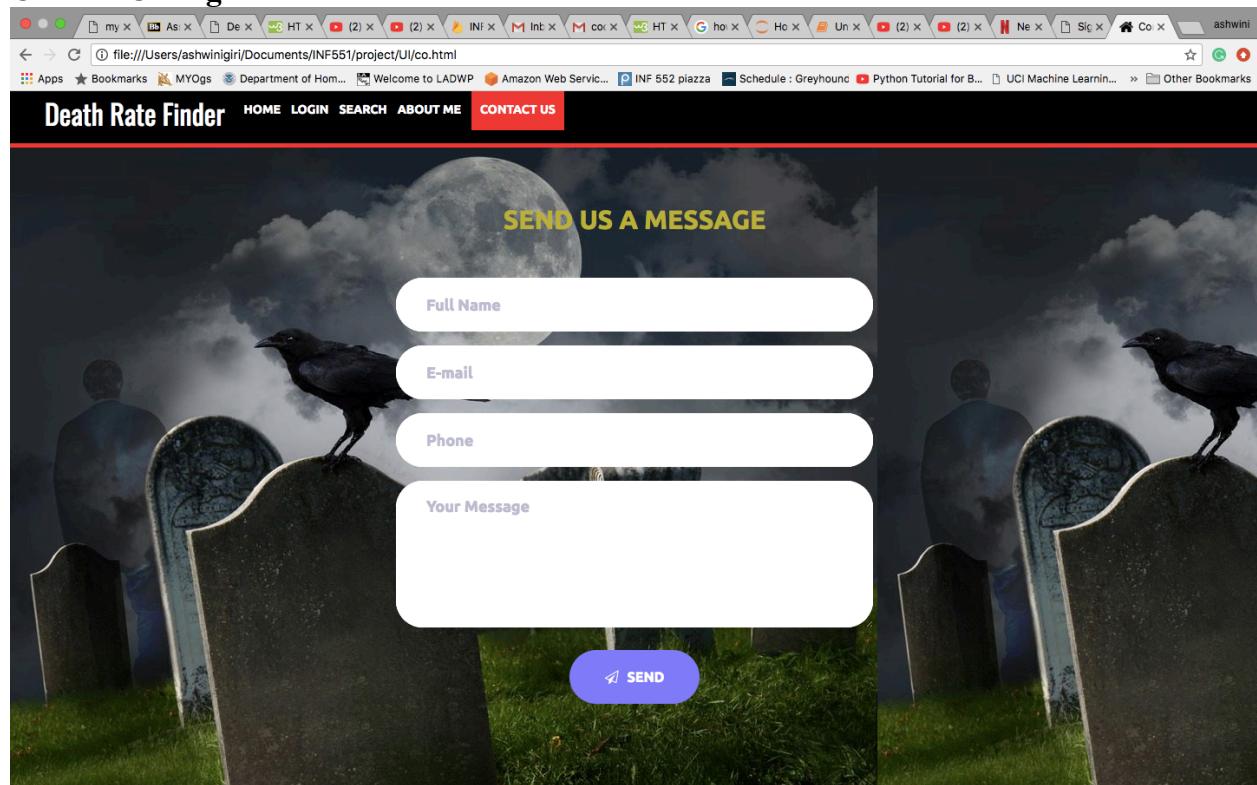


# Mid-Term Report

## Search Page



## Contact Us Page



# Mid-Term Report

## Milestones:

### Completed:

#### 1. Data preparation and Storage

To start creating a database, the crucial step is analyzing and preprocessing the data. As per the analysis, we found that the data has no missing values, hence we did not have to handle the missing values using imputation techniques.

The dataset is available in a csv file, and we are using firebase as a storage. Firebase database works on Json data, hence we have converted the csv data to Json file. We have uploaded the data on firebase, we have implemented the code using python programming language and python REST API to connect with firebase.

### Firebase Realtime database

The screenshot shows the Firebase Realtime Database console. The left sidebar includes sections for DEVELOP (Authentication, Database, Storage, Hosting, Functions), STABILITY (Crashlytics, Performance, Test Lab), ANALYTICS (Dashboard, Events, Audiences, Attrib...), and GROW (Predictions, Notifications, Remote C...). The main area displays the database structure under the project 'inf551'. The 'Database' tab is selected, showing the 'Realtime Database' dropdown. Below, tabs for DATA, RULES, BACKUPS, and USAGE are visible. A message at the top states: 'Read-only & non-realtime mode activated to improve browser performance. Select a key with fewer records to edit or view in realtime.' The database structure is as follows:

```

inf551-8d68c
  cause
    full
      9099
        113 Cause Name: "Influenza and pneumonia (J09-J18)"
        Age-adjusted Death Rate: 25.3
        Cause Name: "Influenza and pneumonia"
        Deaths: 536
        State: "West Virginia"
        Year: 2003
      91
        113 Cause Name: "Accidents (unintentional injuries) (V01-X59,Y85...)"
        Age-adjusted Death Rate: 35.8
        Cause Name: "Unintentional Injuries"
        Deaths: 1248
        State: "Oregon"
  
```

#### 2. Designing User Interface

We completed the User interface static pages designing. The pages are created using HTML and CSS. We have created a homepage through which the user can either login or dive in as guest. The second page that we have created is the search page. The user can access any page from the current page. The user will be directed to login page on clicking the LOGIN button on the home page. The login page provides functionality to login or sign up. The contact us page provides functionality to send the user query to the developers.

## Mid-Term Report

### To be Completed:

#### 1. Faceted Search

We have planned to implement the functionality of the search and interaction between the database and the web application using JavaScript.

#### 2. Save search history

If the user has logged-in then the user should be able to save the search history. The history will be saved in the firebase database. We will implement this functionality using the Login Authentication feature provided by the firebase.

We have implemented all the static design, and database of the project. The remaining task of implementing search and saving the search history will be done in coming days.

As per the scope of the project we are on the track towards completion of the project.

### Challenges:

The learning part of the project was challenging as we are novice in web application development. Hence, we faced many challenges while designing the user interface, but with few more learning we were able to complete the design.

### Group Information:

Ashwini Giri- USC ID (5413882039); agiri@usc.edu

Isha Patil- USC ID (6634971288); ishapati@usc.edu