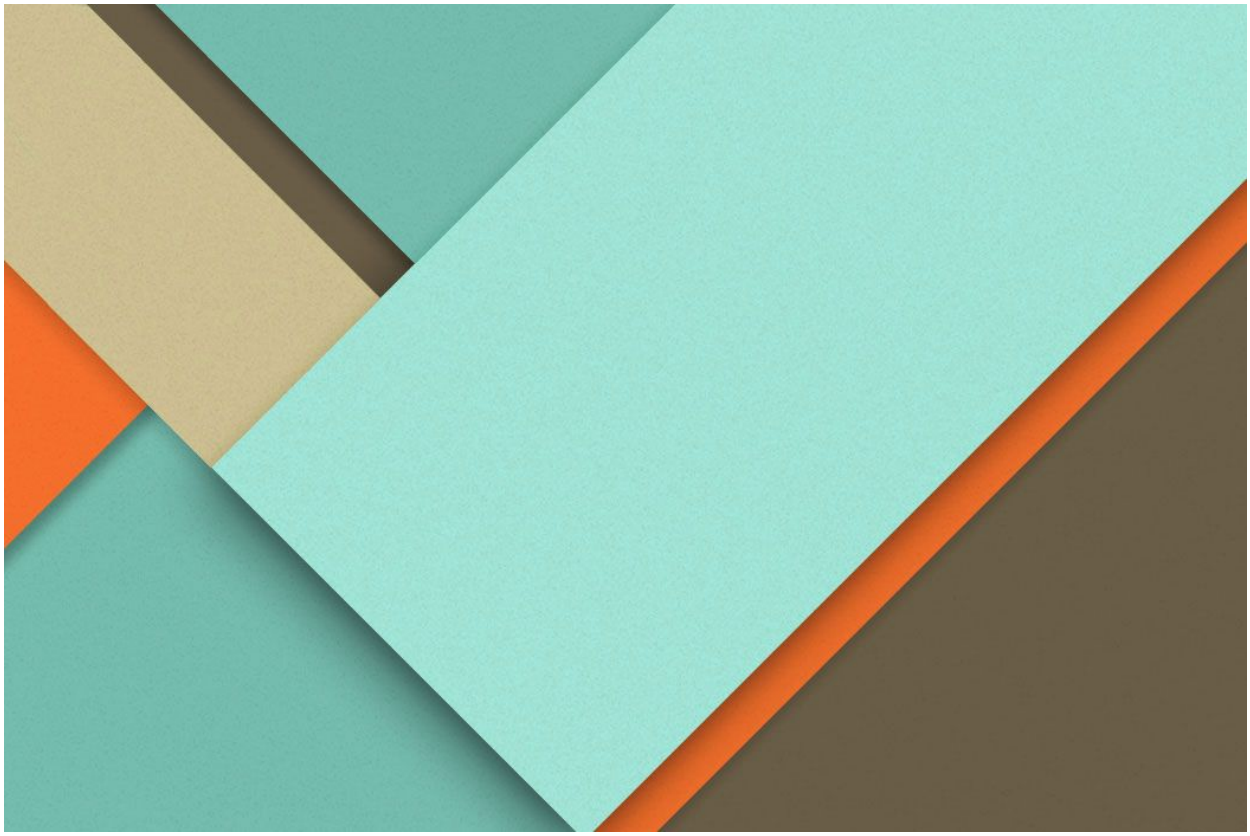


Death Rate Finder!

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 Epidemics, 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.

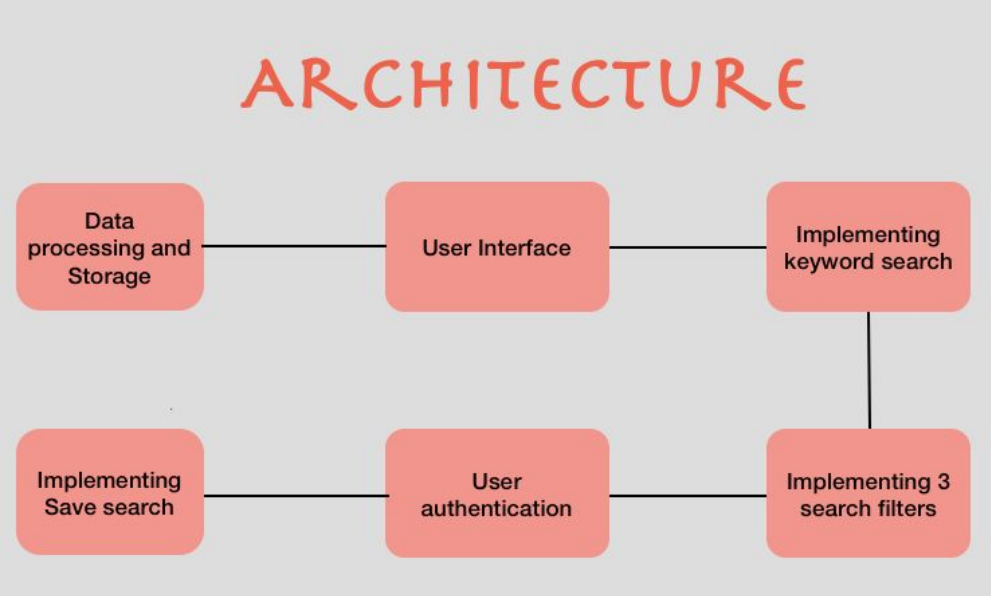
Goals

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 four major components

1. Data processing and Storage on firebase database
2. User interface
3. Implementing the functionalities.
    a. Keyword Search
    b. Faceted Search
    c. Save Search
4. User Authentication



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

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

Year	113 Cause Name	Cause Name	State	Deaths	Age-adjusted Death Rate
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Alabama	2313	52.2
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Alaska	294	55.9
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Arizona	2214	44.8
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Arkansas	1287	47.6
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	California	9198	28.7
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Colorado	1519	39
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Connecticut	1034	29.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Delaware	267	35.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	District of Columbia	161	28.4
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Florida	5961	35.7
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Georgia	3078	41.5
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Hawaii	293	24.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Idaho	597	48.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Illinois	4125	33.7
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Indiana	2309	38.4
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Iowa	1123	35.2
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Kansas	1126	40.7
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Kentucky	1730	43.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Louisiana	1940	44.7
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Maine	458	34.9
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Maryland	1296	25.9
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Massachusetts	1303	19.6
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Michigan	3188	32.8
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Minnesota	1772	35.8
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Mississippi	1642	58.9
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Missouri	2465	43.2
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Montana	461	50.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Nebraska	668	37.3
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	Nevada	710	38.5
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	New Hampshire	329	27.5
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	New Jersey	2227	26.2
1999	Accidents (unintentional injuries) (V01-X59,Y85-Y86)	Unintentional Injuries	New Mexico	969	55.9

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:

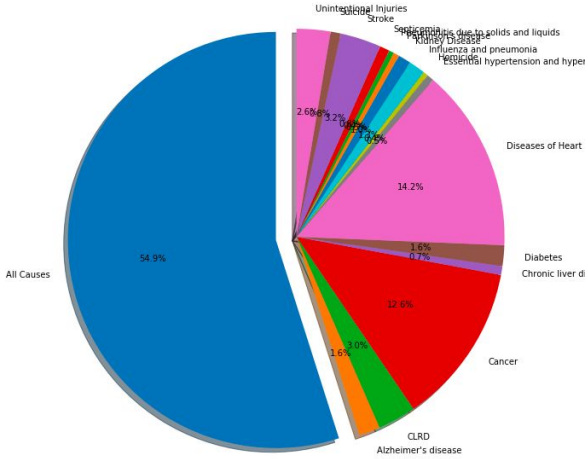
4. State of death occurrence

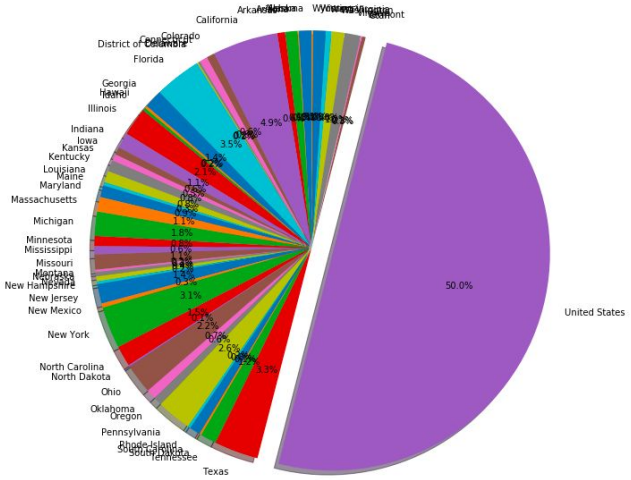
Other attributes:

5. Age adjusted death rate
6. 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 the 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.





2. 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.

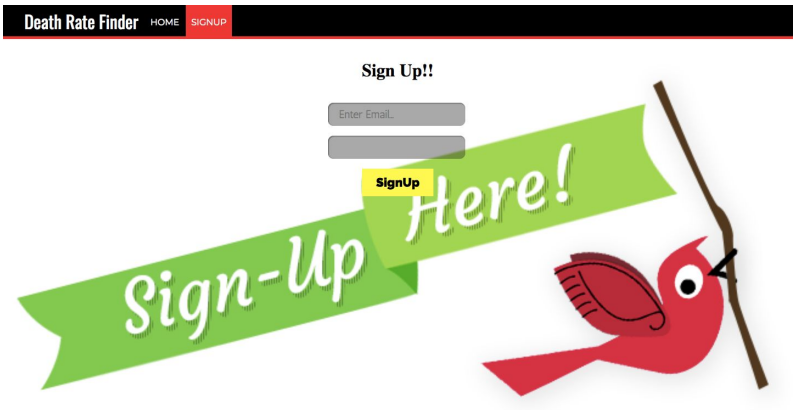
3. Creating User Interface:

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

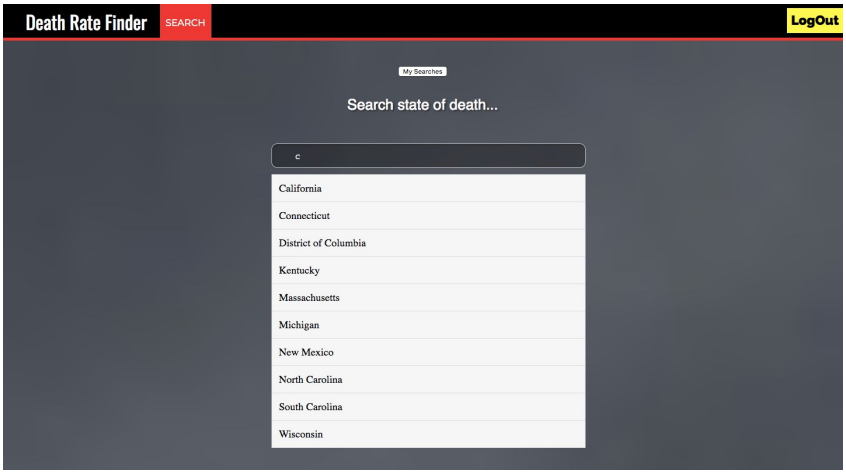
- **Home Page**

- **Login Page**

- **Sign Up Page**



- **Search Page**



- **Contact Us Page**

4. **Implementing Keyword search:**

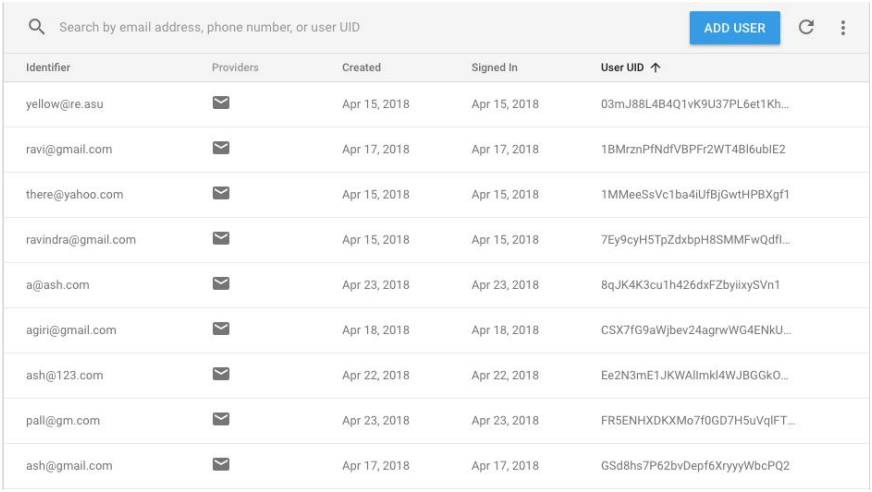
The keyword search functionality is implemented on the attribute 'state of death'. To implement the functionality we need to query the database efficiently and quick. Hence we thought of creating an inverted index on the state attribute. The functionality is implemented in JavaScript using firebase querying and coding constructs.

5. **Implementing Faceted search:**

The faceted search functionality is implemented on the attributes 'cause of death', 'year of death' and 'number of deaths'. To implement the functionality we created these 3 filters on the HTML page and retrieved the data based on user filters and search from firebase database by appropriately querying the database. The functionality is implemented in JavaScript using firebase querying and coding constructs.

6. Implementing User authentication:

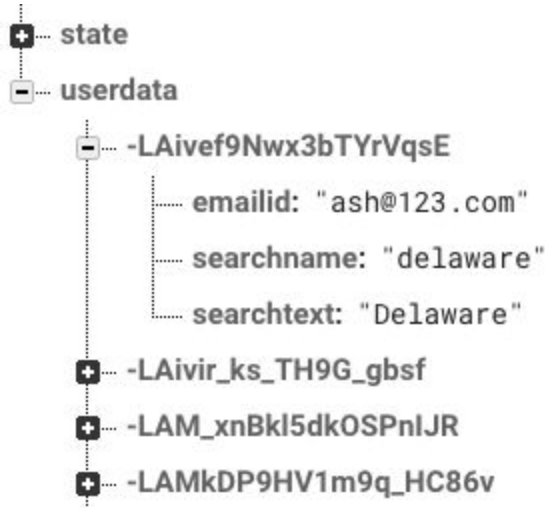
Firebase email user authentication is used for implementing the login and sign-up functionality. JavaScript is used to integrate the web page to firebase authentication and also for the implementation.



Identifier	Providers	Created	Signed In	User UID ↑
yellow@re.asu	✉	Apr 15, 2018	Apr 15, 2018	03mJ88L4B4Q1vK9U37PL6et1Kh...
ravi@gmail.com	✉	Apr 17, 2018	Apr 17, 2018	1BMrznPfNdfVBPFr2WT4Bl6ubIE2
there@yahoo.com	✉	Apr 15, 2018	Apr 15, 2018	1MMeeSsVc1ba4iUfBjGwtHPBXgf1
ravindra@gmail.com	✉	Apr 15, 2018	Apr 15, 2018	7Ey9cyH5TpZdxbpH8SMMFwQdfI...
a@ash.com	✉	Apr 23, 2018	Apr 23, 2018	8qJK4K3cu1h426dxFZbyiixySVn1
agiri@gmail.com	✉	Apr 18, 2018	Apr 18, 2018	CSX7fG9aWjbev24agrwWG4ENkU...
ash@123.com	✉	Apr 22, 2018	Apr 22, 2018	Ee2N3mE1JKWAIlmkl4WJBGGkO...
pall@gm.com	✉	Apr 23, 2018	Apr 23, 2018	FR5ENHXDKXMo7f0GD7H5uVqIFT...
ash@gmail.com	✉	Apr 17, 2018	Apr 17, 2018	GSd8hs7P62bvDepf6XryyyWbcPQ2

7. Implementing Save search:

The save search functionality is implemented in JavaScript and the data for user search is populated on firebase using firebase database querying. We have created a node ‘userdata’ to store the save search data of the user.



8. Implementing Contact Us:

The contact us data is fetched from the web page using dom attributes of the JavaScript and the data is pushed to firebase database and saved to a seperate contact us node created in the database.

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.
2. **HTML:** The static web pages were created using HTML.
3. **CSS:** The styling of the web pages was done using the CSS properties.
4. **JavaScript:** Interaction of firebase and the web pages was implemented using JavaScript. Also the keyword search, database querying, faceted and save search functionalities are implemented using JavaScript coding sonstructs.

Advantages and Disadvantages of Firebase Cloud

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

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(in assistance with Isha).
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 created the web page for it.
8. **Mid-term report:** Created the mid-term draft.
9. **Final project presentation slides:** Created half of the end term presentation.
10. **Final Report:** Drafted 80% of the final project report.

Isha:

1. **Data Preparation:** converting the data from csv to json format. Analyzing the data in python and plotting pie charts for visualization using python(in assistance with ashwini).
2. **Login:** Created the web page for login using HTML and CSS. Implemented the login function using firebase email authentication.
3. **Sign Up:** Created the web page for sign-up using HTML and CSS. Implemented the sign-up function using firebase email authentication.
4. **Project proposal report:** Created the project proposal draft.
5. **Mid-term Presentation slides:** Created the mid term presentation.
6. **Final project presentation slides:** Created half of the end term presentation.
7. **Final Report:** Drafted 20% of the final project report.