Personal Recipe Book

INF 551 Data Management Final Project Report

University of Southern California

Anmol Chawla & Nikhit Mago

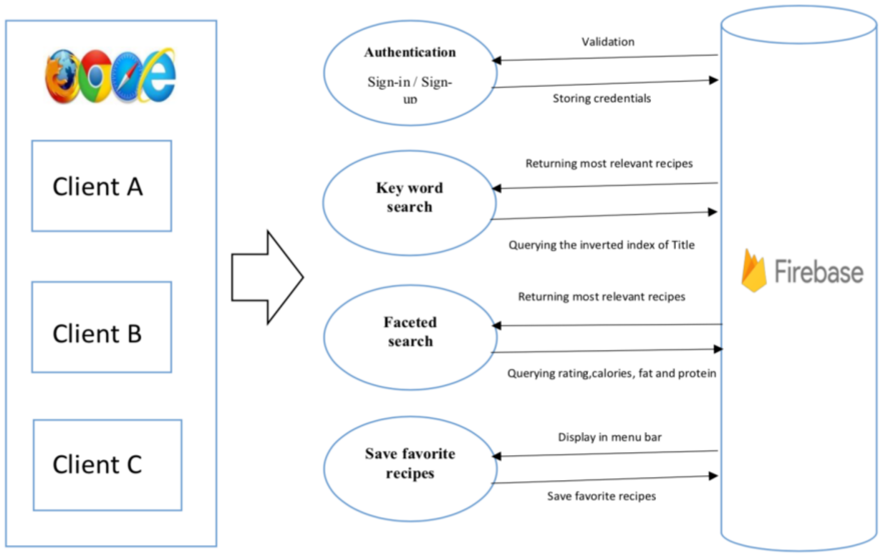
https://dmhw1-28e4c.firebaseapp.com/

Project motivation and goal

The motivation for the project was simple, to create an online web application where users can access data. The goal was to apply our classroom knowledge of databases and to learn a new skill of creating a web application. It was decided that a Kaggle Dataset which consisted of 20,000+ recipes will be used to create an online portal, where users can search for recipes based on keywords, calories, proteins and fat content. The user will also be able to save, execute and delete past searches.

Architecture of the application

1. **Detailed Architecture of the Application:**



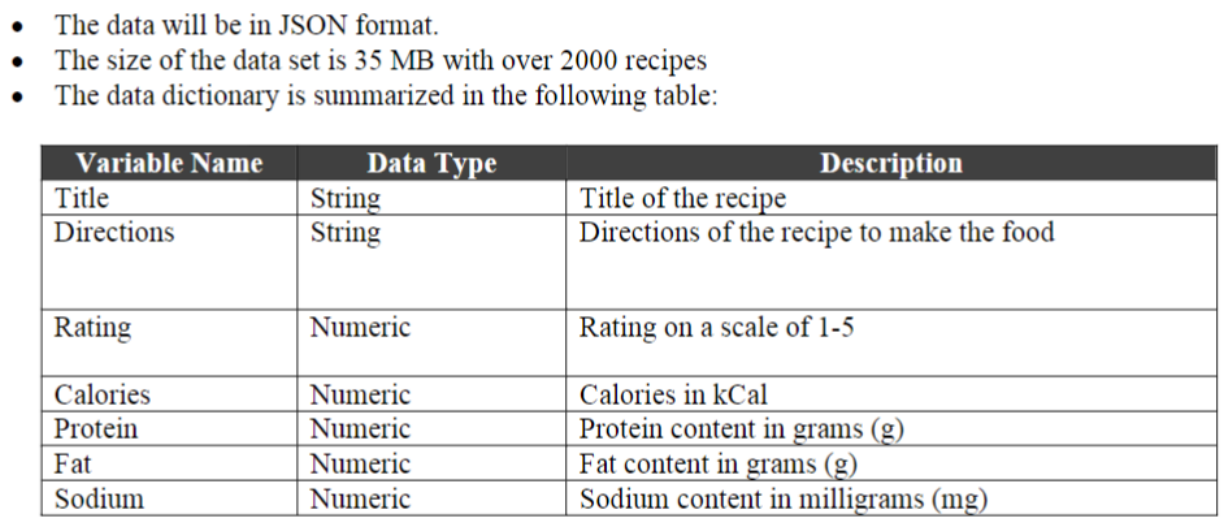
2. **Data Management Components:**

- **Data Structures:** Inverted index was the primary data structure that was used for querying the data uploaded to Google Firebase. An inverted index for each token in the title was created for querying the results. Each facet was also hashed into different bins and these bins were also converted into inverted indices.
- **Query Methods:** JavaScript was used to query the inverted index that was uploaded to Firebase. The querying was very similar to REST CURL commands like PUT, GET and UPDATE. Intersection of indices was used to display the results.

Implementation Details

The Implementation can be split into the following parts

1. **Data Acquisition:**
   Data was downloaded from a Kaggle Dataset
   https://www.kaggle.com/hugodarwood/epirecipes



- The data will be in JSON format.
- The size of the data set is 35 MB with over 2000 recipes
- The data dictionary is summarized in the following table:

Variable Name	Data Type	Description
Title	String	Title of the recipe
Directions	String	Directions of the recipe to make the food
Rating	Numeric	Rating on a scale of 1-5
Calories	Numeric	Calories in kCal
Protein	Numeric	Protein content in grams (g)
Fat	Numeric	Fat content in grams (g)
Sodium	Numeric	Sodium content in milligrams (mg)

2. **Data Cleanup and Upload:**

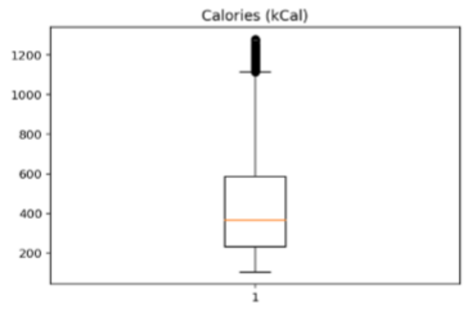
.

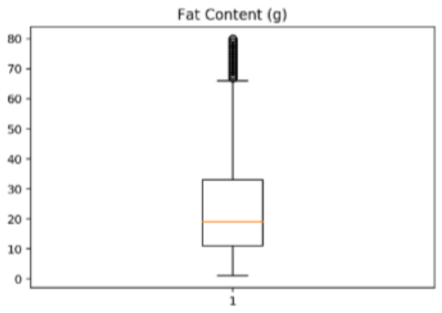
   a. The following attributes were present in each id : 'directions', 'fat', 'date', 'categories', 'calories', 'desc', 'protein', 'rating', 'title', 'ingredients', 'sodium'.
   b. There were some fields with missing values and some with non-existent fields all together.
   c. The pre-processing phase required us to determine our attributes for key - word search and faceted search.
   d. After a visual representation and some research paper references the following attributes were chosen.

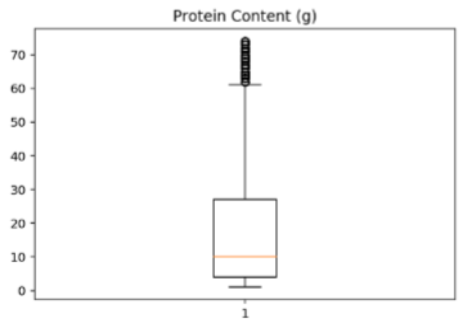
Type	Attribute	Low	Medium	High	Heavy
Key-Word Search	Title	Null	Null	Null	Null
Faceted Search 1	Rating	[0 – 1.6)	[1.6 – 3.2)	[3.2 – 5]	Null
Faceted Search 2	Protein	0 -10	10 -20	20 – 30	30 – 50
Faceted Search 3	Fats	0 - 12	12-30	30 – 40	40 - 60
Faceted Search 4	Calories	[0 – 350]	[350 - 625]	[625 - 1000]	[1000 -2500]

Note: Box plot, design flow

- The data to be displayed would be the Title, ingredients and recipe.
- The pre - processing involved finding the quartiles of each attribute and then comparing it with scientifically researched values for categorization.
- Rows of missing data were removed as imputation in such cases was not feasible due to presence of string variables.
- Imputation for other aspects was out of the scope of the project.



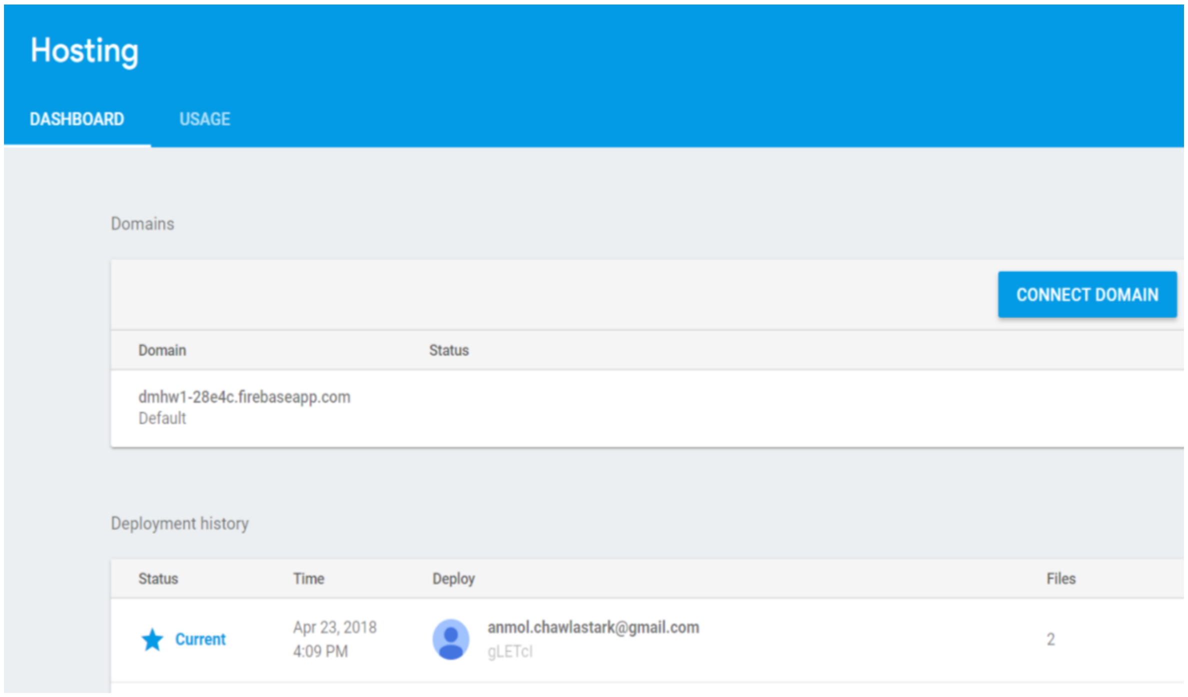




- The preprocessing of ‘Title’ for keyword search involved removal of special characters, using regular expressions to fetch out proper words and handling of different type of file format.
- An inverted index was created for each word in the titles for querying the keywords
- The pre-processing involved finding the quartiles of each attribute and then comparing it with scientifically researched values for categorization.
- Rows of missing data were removed as imputation in such cases was not feasible due to presence of string variables.
- Imputation for other aspects was out of the scope of the project.
- After the pre-processing step, there were a total of 8000+ recipes in the database.
- The other aspects like calorie, fat, protein was put into buckets. The aim was to assign a user input to one of the buckets.

3. Firebase Hosting

The website was hosted on google firebase with the standard hosting rules, at: https://dmhw1-28e4c.firebaseapp.com/

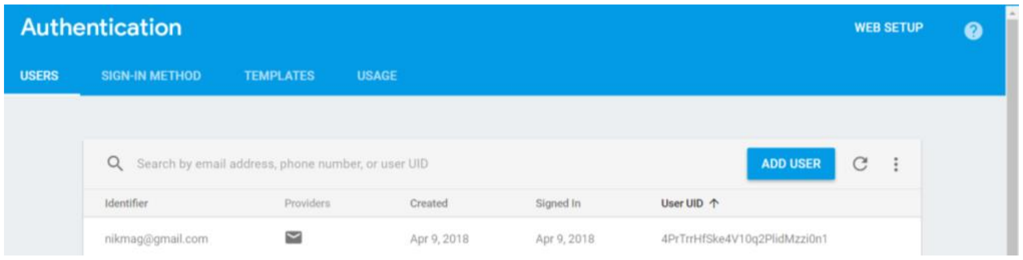


Domain	Status
dmhw1-28e4c.firebaseapp.com Default	

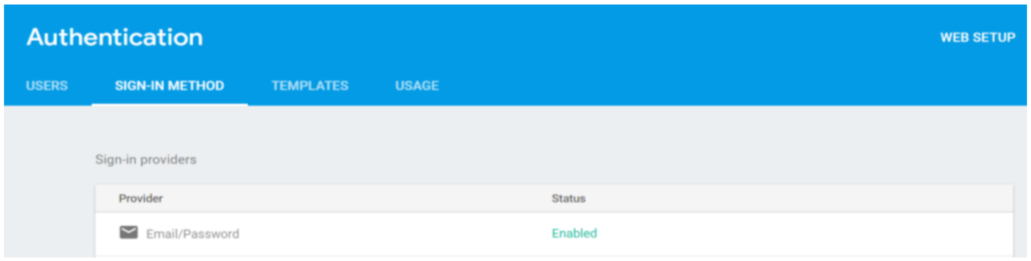
Status	Time	Deploy	Files
Current	Apr 23, 2018 4:09 PM	anmol.chawlastark@gmail.com gLETcl	2

4. **Firebase User Management System**

- Google's standard user management system was used, which was based on the user's email Id and password. Each user was allocated a unique ID which was later used for further feature implementation.



Identifier	Providers	Created	Signed In	User UID ↑
nikmag@gmail.com	✉	Apr 9, 2018	Apr 9, 2018	4PrTrrHfSke4V10q2PlidMzzi0n1



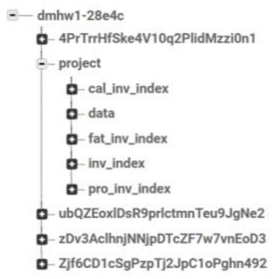
Provider	Status
✉ Email/Password	Enabled

5. **Material Lite UI/UX**

- The web elements of the UI were designed using material Lite

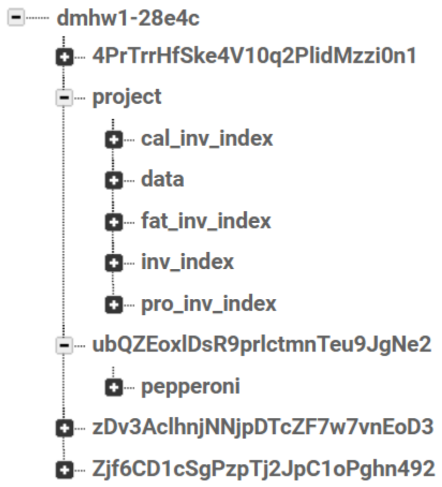
6. **Keyword Search feature and Faceted search feature**

Keyword search was implemented by creating an inverted index of every word and querying the index. A combinatorial algorithm was designed to display the most appropriate results first. This algorithm works by taking intersection of different combinations of indices and ordering them according to relevance. The faceted search feature works in a similar way where for every bin/category, an inverted index was created, queried and intersected with the keyword search. The inverted indices can be observed in the figure below.



```
dmhw1-28e4c
  4PrTrrHfSke4V10q2PlidMzzi0n1
  project
    cal_inv_index
    data
    fat_inv_index
    inv_index
    pro_inv_index
  ubQZEoxlDsR9prlctmnTeu9JgNe2
  zDv3AclhnjNNjpDTcZF7w7vnEoD3
  Zjf6CD1cSgPzpTj2JpC1oPghn492
```

7. **Save and remove search feature**



```
dmhw1-28e4c
  4PrTrrHfSke4V10q2PlidMzzi0n1
  project
    cal_inv_index
    data
    fat_inv_index
    inv_index
    pro_inv_index
  ubQZEoxlDsR9prlctmnTeu9JgNe2
    pepperoni
  zDv3AclhnjNNjpDTcZF7w7vnEoD3
  Zjf6CD1cSgPzpTj2JpC1oPghn492
```

Documentation of Codes

The code was compiled into one single HTML page. The code can be split into the following three components

1. The CSS style sheet
2. HTML Code and Material Lite design
3. JQuery, Javascript , Node Js
4. The single HTML page was named as Index.html
5. The code can directly be viewed on the following website https://dmhw1-28e4c.firebaseapp.com/

Advantages and Disadvantages of Google Firebase

Advantages:

(a) Simple, fast and easy to use NoSQL cloud based database
(b) Allows you to host your website without worrying about other servers
(c) Allows you to have Google Authentication
(d) Allows you to write simple and effective queries

(e) Does not require expert knowledge of SQL
(f) Indexing works very well with Google Firebase
(g) Good for Real-Time applications
(h) Complete package with hosting, domain name, database, image storage, web, iOS, Android, API's

Disadvantages:

(a) Not very robust for advanced queries like GroupBy, Join etc.
(b) Can query only based on the index
(c) Not easy for migration if your application is highly dependent on SQL
(d) Cost only justified if you have a large customer base

Responsibility of each group member

Anmol:

(a) Complete user interface and user experience design using HTML, CSS, Javascript
(b) Implementation of user authentication using Google's API
(c) Implementation of web hosting on Firebase
(d) Implementing save and display
(e) User session management and code integration for search feature.
(f) Implementing remove feature for saved search
(g) Maintaining documentation

Nikhit:

(a) Data cleaning, data visualization, inverted index creation and uploading data to Firebase
(b) Querying data in Javascript for keyword and faceted search
(c) Designing combinatorial algorithm for displaying most appropriate results first
(d) Implementing save and display
(e) Code integration with UI
(f) Maintaining documentation