TABLE OF CONTENTS

CONTRIBUTION

Person	Role	Contribution
Shahryar Abbas Mirza	Front-end, client developer	Mobile friendly Web app design. Animation, Layout and user based visual Analytics using nvd3 angular variant
Yamin Huzaifa	Back-end, server developer and front-end	Smooth back-end integration and deployment with multiple servers on Google app engine and database on heruku-Aws Storage. API implementation, resolving bugs and security issues like CORS policy

Equal effort and contribution by both team members

LINKS

Live Url	https://nodetest-229207.appspot.com
Github	https://github.com/Huzi1/MovieRecommend.git
Movie database (API)	https://www.themoviedb.org/
User guide Video	https://drive.google.com/file/d/11ECgLsmRltw5AFVhGEPeuf_V59vgOpyj/view

SUMMARY:

The main objective of this project was to create a web application where user interacts, and that user behaviour is analysed. In order to do this, we have added different features.

INTRODUCTION:

Truly understanding our customers is the most important role of a product manager. After we ship a feature, my most pressing question is, "are people using it and how?!" I want to know whether we made the right assumptions and built the thing people wanted. This is a question that vanity metrics like Daily Active Users and Sessions cannot answer. Really understanding the impact of my decisions on my product's users requires a deeper layer of analysis –a behavioural analysis.

Apart from this we were also interested in the recommendation systems. Movie streaming services like Netflix, Hulu, Amazon Prime, and others are increasingly used by consumers to enjoy video content. For example, in 2017 Netflix subscribers collectively watched more than 140 million hours per day and Netflix surpassed $11 billion in revenue in 2017. In fact, roughly 80% of hours streamed at Netflix were influenced by their proprietary recommendation system. Undoubtedly, movie streaming services have become an integral part of how we consume video content today, and the importance of movie recommendation systems cannot be understated-they are an integral part of how we consume video content today. With this in mind, we wanted added a feature called recommendations. Where the user are suggested with a list of 5 movies based on the movie he selected.

TOOLS AND LANGUAGES:

Google App engine

- Main Project on Node.js environment
- Mini-Project running Microservice on node.js environment

Database

- PostgresSQL running on Heruku(aws) database

Languages

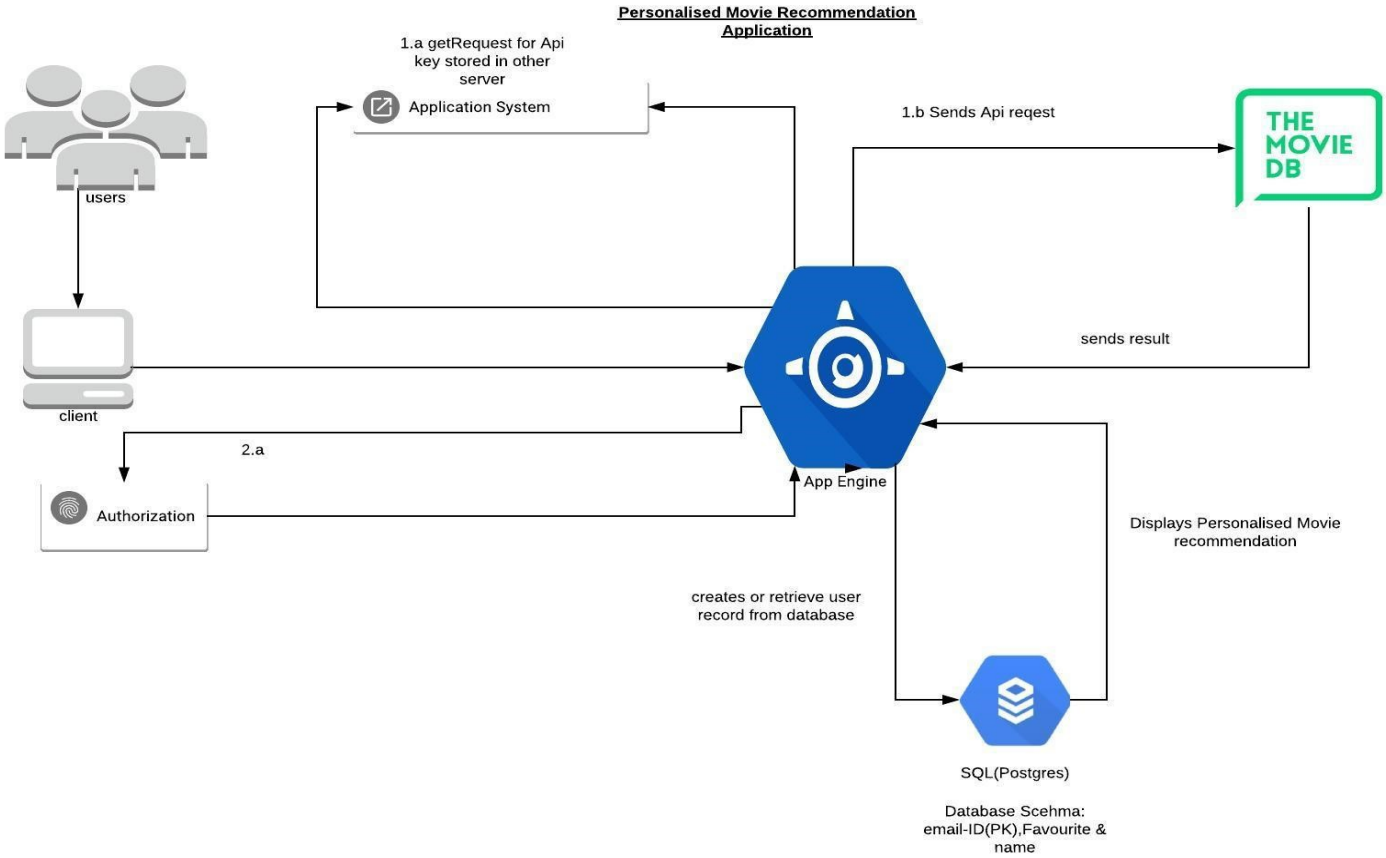
- Html,Css, Javascript and Js framework (Angular)
- Node.js at server

Charts

- D3,Nvd3

SOFTWARE DESIGN/ARCHITECTURE

High level diagram



LIST OF API USED

	API
1	Google Authorization API
2	TMDB database API
3	Movie Search API
4	TV Series API
5	Movie Recommend API
6	TV series Recommend API

DEVELOPERS MANUAL.

Follow these steps to set up the project and deploy on Google app engine

CLONE THE FILES .

1. Clone the Movie Recommend files from GitHub repository, *Alternatively you can directly download the .zip file*

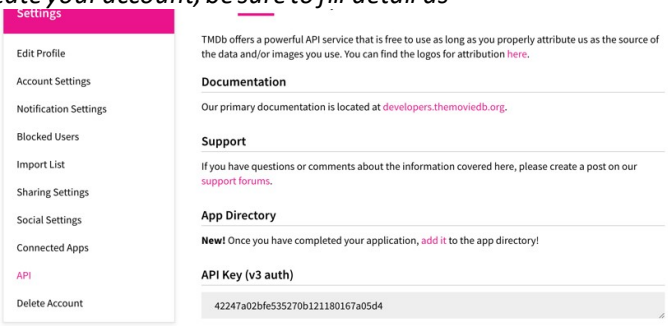
   Git clone https://github.com/Huzi1/MovieRecommend.git

GETTING API KEY FROM TMDB

For this application to work you need movie database . We are getting database from TMDB through API. You need to register and request for API from TDB

2. *Go to https://www.themoviedb.org and create your account, be sure to fill detail as developer not any business organisation. The developer version is free but it has some limitation such as number of api call in a minute. An api key from TMDB will look something like this :*

   42247a02bfe535270b121180167a05d4



   You can also view it later from https://www.themoviedb.org/settings/api

SETTING UP **NODE.JS ENVIRONMENT AND DEPLOYING NODE.JS SERVER ON GCLOUD**

3. *Create a free gmail account and follow googles tutorial to deploy your first node.js application. Name the project microservice*
   https://cloud.google.com/nodejs/getting-started/hello-world

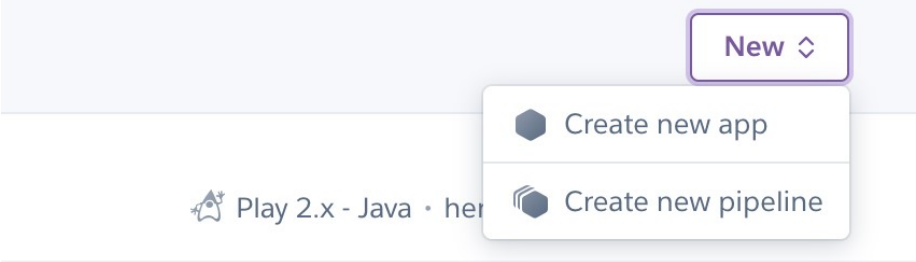
   after running it successfully locally and on cloud you need to edit the app.js file

   This server will only store you TMDB key and respond to your primary server's request for key

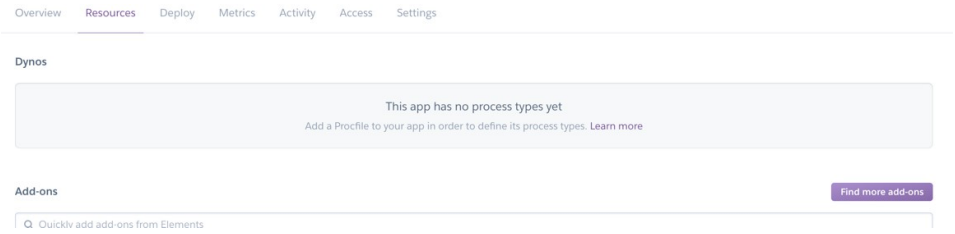
SETTING UP YOUR DATABASE AND BUILDING **THE CONNECTION**

4. Go heruku cloud service website and create your account *https://www.heroku.com/*

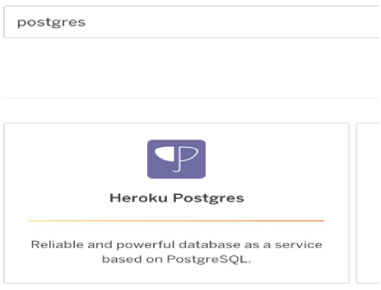
   Once you successfully create an account click on create new app

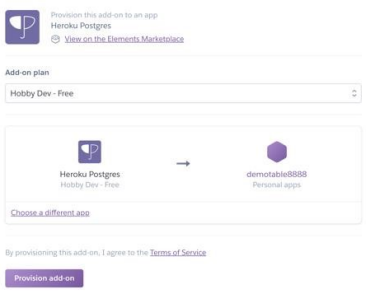


- Write your *heroku-appname* choose appropriate *region*
- *Go to resources and click add on in your app*



- Search Postgres and install the extension. Use Hobby Dev-Free Plan and write your *heroku-appname*






```
const express = require('express');

const app = express();

app.use(function (req, res, next) {
    res.header("Access-Control-Allow-Origin", "*");
    res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
    next();
});

app.get('/getTmdb', (req, res) => {
  res
    .status(200)
    .send('42247a02bfe535270b121180167a05d4')
    .end();
});

// Start the server
const PORT = process.env.PORT || 8080;
app.listen(PORT, () => {
  console.log(`App listening on port ${PORT}`);
  console.log('Press Ctrl+C to quit.');
});
// [END gae_node_request_example]
```

- Now go to Data and you will see your database link. Go to setting and then credentials. You would need these credentials to link Postgress database to your local machine and in your script as well to establish connection with your webapp

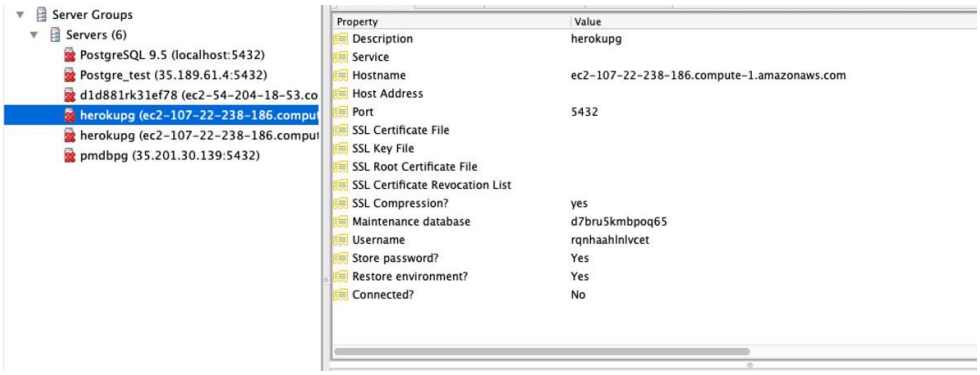
Database Credentials

Get credentials for manual connections to this database.

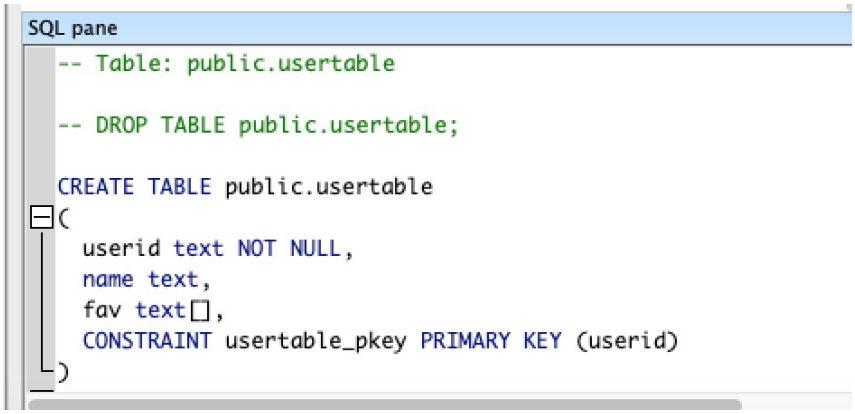
Please note that **these credentials are not permanent**.
Heroku rotates credentials periodically and updates applications where this database is attached.

Host	ec2-54-243-128-95.compute-1.amazonaws.com
Database	d6bl4drm56dh1a
User	abiefwnirpistf
Port	5432
Password	cfb329f613482558a7b137c50fa95be4b4eb8c317ae7548e2bcd7153819fe646
URI	postgres://abiefwnirpistf:cfb329f613482558a7b137c50fa95be4b4eb8c317ae7548e2bcd7153819fe646@ec2-54-243-128-9
Heroku CLI	heroku pg:psql postgresql-spherical-45364 --app demotable8888

- Download Postgres pgadmin from https://www.pgadmin.org/download
- Once installed, go to Connection and establish the link using the credentials before mentioned. It should be filled as following



- Once the link is established go to your database and create new table with following columns and data types



- Now go to *./pmdb-final-heruku./server.js* *and edit the script according to your credentials*
- After all these changes save it and deploy your app on app engine
- When filled your table will look like this

```
const client = new Client({       //database connection on heruku

host:'ec2-107-22-238-186.compute-1.amazonaws.com',
user:'rqnhaahlnlvcet',
database:'d7bru5kmbpoq65',
password:'5dae6b0e27e09998074b8376ab75e121c129672d5ecbb71fc4f42397072f331b',
port:'5432',
ssl:true


})
```

	userid [PK] text	name text	fav text[]
1	huzaifayamin11@gmail.com	huzaifa yamin	{505954_T-34_M,"480530_Creed II_M",4528
2	m.s.abbas16@gmail.com	Mirza Abbas	{450465_Glass_M,297802_Aquaman_M,"42469
3	s3667340@student.rmit.edu.au	Yamin Huzaifa	{496076_Zero_M,"438_Cube Zero_M"}
*			

SETTING UP GOOGLE AUTHENTICATION API

5. *You need to request for The Api key from google follow this step by step guide from google,but first deploy Movie recommend app on different project ID on Gcloud after your edit the after mentioned script in your app.js file .*

   You also need to edit file pmdb-final-heruku/assets/js/app.js before you deploy the app on gcloud

```
1  var app = angular.module('pmdbApp', ['nvd3']);
2
3  //getting key from other server pmdbmicro
4
5  app.factory("MyService", function ($http) {
6      return {
7          fighters: function () {
8              return $http.get("https://pmdbmicro.appspot.com/getTMDB").then(function (response) {
9                  return response.data;
10             });
11         }
12     }
13 })
14
```

edit the url on line 7 to your 1st project(microservice)

Deploy your app and copy the url of the application

You will need the URL address of your main project for google login setup.

https://developers.google.com/identity/sign-in/web/sign-in

once you have the key you need to edit your home.html file in *./pmdb-final-heruku /assets/html/home.html on line 7(write your key in quotes)*

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6
7      <meta name="google-signin-client_id" content="45154372968-0u3rkhb3g2hptlaq9o1sfmkh13v7s97q.apps.googleusercontent.com">
8      <script src="/js/platform.js" async defer></script>
9
10     <link rel="stylesheet" href="/css/nv.d3.min.css" />
11     <script src="/js/angular.min.js"></script>
12     <script src="/js/d3.min.js" charset="utf-8"></script>
```

LEARNING AND CHALLENGES

Through this assignment we got hand on experience on different cloud platforms, handling different Api and integrating them with our Web app smoothly. We also polished our programming skills on JS by using popular Angularjs framework for coding and d3 libraries for visualisation. We wanted to further optimize the user-based recommendation by implementing collaborative functions, but due to time constraint we decided to explore this venue later on. This could be a possible extension of this project including implementing a cross platform app on Js iconic framework. One of the major challenges was implementing Google authentication api without any CORS policy issue

ANALYTICS:

As mentioned in the introduction about the behaviour analysis of user, it is the data that tells you how your customers behave in mobile applications or on websites. It goes beyond basic metrics like monthly active users or pageviews. Behavioural data reveals how engagement with your product impacts retention, conversion, revenue, and the outcomes you care about.

Understanding user behaviour is necessary to increase engagement, retention, lifetime value, conversion rates, and ultimately, revenue. And the building blocks of behavioural analysis is EVENTS. Events represent any action a user can perform in your product (like opening the app, creating an account, viewing a video) or any activity associated with the user (like making a purchase).

Sending optimal event data to your analytics platform is the single most important step toward understanding how your users are engaging with your product. If you're too hasty in instrumenting your analytics you may never get the full value of your data.

As we have built a movie application. The onboarding process or the series of events which are done by the user are: 'Open the website', 'Registration using Gmail', 'search for any movies or Tv series', 'Bookmark favourite movies'.

Behind every great user behaviour analytics is a great event taxonomy – the way you organize this collection of events and properties you're using to define the actions people can take within your product. Think of the event taxonomy as the foundation for all future analysis you'll do with your analytics platform. It's crucial to get right. Here we are keeping track of the users by saving data in one of the popular cloud services called Heroku.

Most analytics platforms require you to configure some kind of identifier–a username or email, for example–in their mobile SDKs or HTTP API for keeping track of unique users. This lets you match data from multiple devices and sessions to one user. Because of this, it's important to make sure the user ID is set to something that will not change.

As we want to understand user behaviour as a whole across the entire user's journey, so the solution is we can do cross-platform instrumentation that is by using google sign in and their Gmail as the user ID.

Assigning user properties and event properties can give you deeper insight into the behaviours your customers are exhibiting as they engage with the application. A user property describes attributes of an individual person using your app (e.g. age, gender, location). An event property describes a

property of an event (e.g. how someone performs the different events). In our application the user and event properties that we can set up are shown below.

Instrumenting your user behaviour events is a huge accomplishment. It's time to start putting all this data to use, now let's sort data which gives meaningful insights of the users. In order to do this we will split the movies/ Tv series which are bookmarked by the user based on the genres and represent them using pie chart. This representation gives that which genres user mostly interested in. We can also visualize another thing that is popularity of that particular Movie/ Tv series using Bar Graph. By this we can understand that whether the user is interested in the most popular or trending Movies/Tv – series. So by doing all this in future we could apply to recommendation system and improve our application.

RESOURCES AND REFERENCES:

- /css/nv.d3.min.css
- /js/angular.min.js
- /js/d3.min.js
- /js/nv.d3.min.js
- /js/angular-nvd3.min.js
- StackOverflow
- TutorialsPoint
- Lynda
- Angularjs [https://angularjs.org]
- D3 libraries [https://d3js.org/]
- Movie Database[https://www.themoviedb.org]
- GoogleAppengine
- Heroku app services
- Demo slides lucidCharts