International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2172102

UNIFY – The University Recommender

Soham Manish Mehta, Nidhi Vinod Padhariya, Sakshi Jayesh Patel
Department of Computer Engineering, SAKEC, Mumbai, Maharashtra, India

Article Info
Volume 7, Issue 2
Page Number: 468-473

Publication Issue :
March-April-2021

Article History
Accepted : 20 April 2021
Published : 25 April 2021

ABSTRACT

As humanity move towards the digital age university education started becoming more and more important for any field in the market, but to understand the process of how the university admissions work a student has to invest a lot of time and money to get the required information with a possibility of getting scammed, to over this issue a sophisticated university recommender system is discussed in this paper and basic details on how to implement the system as well as difficulties faced while developing the said system is discussed in brief.

Keywords: Recommender System, University Recommender, MERN Stack, React, Web Development

I. INTRODUCTION

As we move towards the age of digitization university education has become one of the crucial things for any student, the company have set perquisites that require a student to get a degree for a well-known university, students started going to counsellors where many students might get scammed in name of a university with a promise to get admission and ending up being looted of finances and dreams, counsellors can also be extremely time-consuming and expensive but as the digitization grows students prefer searching for more information online instead of visiting the university and for an international student, it might now be feasible to go to a foreign country just to check out the university, even the source of information available on the internet started becoming a place to scam students offering a fake scholarship, and similar promises as to the fake counsellors.

When a student starts searching for information on a potential university as a start to a career, they have to explore through the internet and go through the surveys which again consumes a lot of time and efforts, to overcome such scammer and time constraints we developed a one-stop solution UNIFY which provides information of any and all universities in the world at one touch of a button designed and developed using upcoming technology which can help its users get proper and perfect information needed in one place.

a. BLOCKS AND FLOW OF UNIFY

The basic building blocks of our proposed system includes a Registration/Authentication Block where the user can register with all the relevant information or authenticate themself to get access to their data, later we move on to the User's Choice Filtering where the user will be able to filter out their choices based on their requirements, next block is Information block where information related to

Copyright: © the author(s), publisher and licensee Technoscience Academy. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License, which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited

universities will be displayed based on users profile and filters, on the user interacts with any universities the information about this would be recorded and then recommendations based on those choices would be displayed to the user.

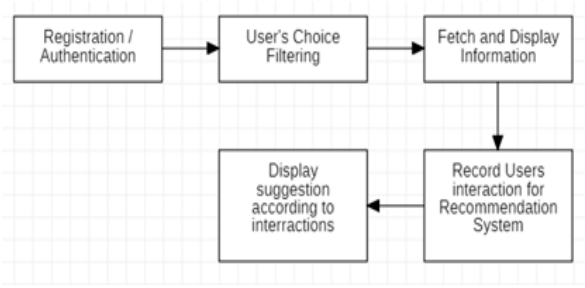

Fig 1 Block Diagram of UNIFY

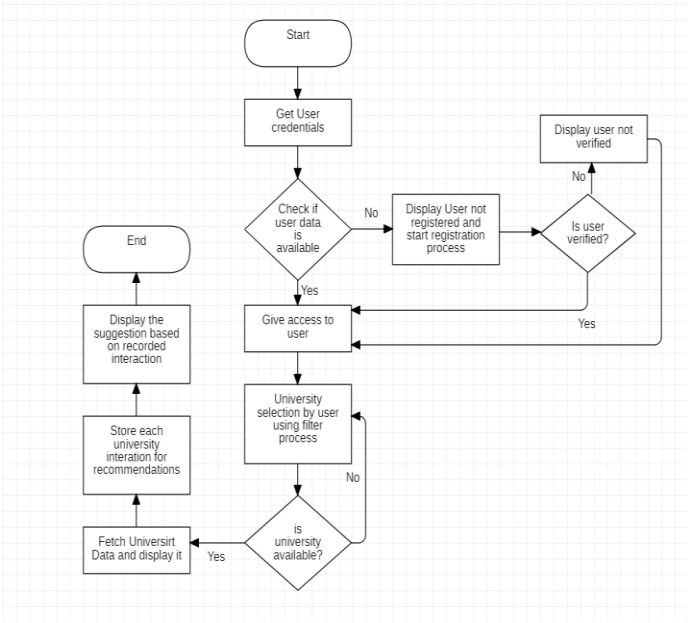

Fig 2 Flow Chart

b. MODULES OF UNIFY:

Login / Registration Module: This is where the user will be able to get access to the website after authorization or be registered to get access to the application/website.

Information Collector: In this module the user will provide information about his/her interests or what he/she is searching for information on using filters.

Information Provider: On the basis of the filters list of universities or courses will be provided to the user, the user would later be able to update the filters if required.

Bookmark: The user will be able to bookmark the information deemed important and can save it to the bookmark section for quick access.

University Review: A user would also be able to provide a review on a university or a course or make a suggestion or review for the application/website.

c. OBJECTIVES OF UNIFY

The objective of UNIFY is to provide information such as :

1. The main objective behind this application to help student decrease the research hours,
2. Increase the productivity and get understandable information in one place,
3. Provide a way for students to contact the content providers and universities.
4. Create a user-friendly website and application,
5. Recommendation System to help the student get introduced to multiple universities and courses he/she might be interested in.

II. DEVELOPMENT TECHNIQUE

a. MERN STACK

A combination of technologies used to create a web application is called the stack, multiple technologies together are used to make any application on web (framework, libraries, databases etc.)

MongoDB: A cross-platform document database

Express: A back-end web application framework

React: A JavaScript library for building user interfaces

Node.js: A cross-platform JavaScript runtime environment

MERN stacks main advantage for developers is that every line of code is written in JavaScript which is used for both client-side scripting and server-side scripting, thus there is no need of context switching. Developers have to figure out how to interface multiple programming language with tech stack. With JavaScript stack, developers need to be well

worse with only JavaScript and JSON. MERN Stack thus allows developers to build highly efficient website application

Working of MERN Stack:

A 3-tier architecture (frontend, backend, database) can be entirely constructed easily with the MERN architecture using JavaScript and JSON.

MongoDB was designed to store JSON data natively, and everything from its command line interface to its query language is built on JSON and JavaScript. MongoDB makes storing, manipulating, and representing JSON data at every tier of your application easy and works perfectly with Node.js.

Express.js is a server-side application framework which makes it easy to map URLs to server-side functions and wraps HTTP requests and responses.

React.js is a frontend JavaScript framework for building interactive user interfaces in HTML, and communicating with a remote server.

The combination makes it fast to build on and reasonably simple to debug by letting JSON data flow naturally from front to back. MERN is the stack of choice for today's web developers looking to move quickly.

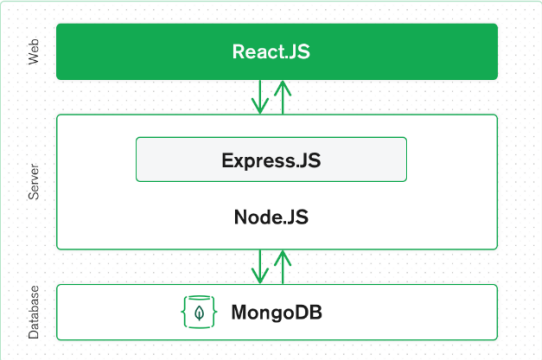

Fig 3 MERN Architecture

b. METHODOLOGY FOLLOWED

Feature-Driven Development and SCRUM:

The development would be broken down into multiple parts on the basis of the features to be added such as university information, visa information, course information, etc. in which everything would further be divided into phases such as design phase, data collection phase, front end development, back end development, security, database creation, etc.

III. DEVELOPMENT TECHNOLOGY

a. FRONT END DEVELOPMENT

React.js Front End:

The top layer of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications using HTML. React lets you build up complex interfaces through simple Components, connect them to data on your backend server, and render them as HTML.

Reacts strong suit is its ability to handle stateful, data-driven interfaces with minimal code and minimal pain providing great support for forms, error handling, events, lists, and more.

HTML:

HTML is a coding language that is used for structuring a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images.

CSS:

CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

b. BACKEND DEVELOPMENT

Express.js and Node.js Server Tier:

The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a "fast, unopinionated, minimalist web framework for Node.js," and that is indeed exactly what it is. Express.js has powerful models for matching an incoming URL with a server function, and handling HTTP requests and responses.

By making XHRs or GETs or POSTs from your React.js front-end, you can connect to Express.js functions that power your application. Those functions in turn use MongoDB's Node.js drivers, either via call-backs for using Promises, to access and update data in your MongoDB database.

MongoDB Database Tier:

If your application stores any data like user profiles, content, comments, uploads, events, etc., then you're going to want a database that's just as easy to work with as React, Express, and Node.

That's where MongoDB comes in: JSON documents created in your React.js front end can be sent to the Express.js server, where they can be processed and (assuming they're valid) stored directly in MongoDB for later retrieval. Again, if you're building in the cloud, you'll want to look at Atlas

IV. BUILDING A RECOMMENDER SYSTEM

Recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy or anything else depending on industries). Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors.

Collaborative filtering methods

Collaborative methods for recommender systems are methods that are based solely on the past interactions recorded between users and items in order to produce new recommendations. These interactions are stored in the so-called "user-item interactions matrix".

Then, the main idea that rules collaborative methods is that these past user-item interactions are sufficient to detect similar users and/or similar items and make predictions based on these estimated proximities.

The class of collaborative filtering algorithms is divided into two sub-categories that are generally called memory based and model-based approaches. Memory based approaches directly works with values of recorded interactions, assuming no model, and are essentially based on nearest neighbors' search. Model based approaches assume an underlying "generative" model that explains the user-item interactions and try to discover it in order to make new predictions.

The main advantage of collaborative approaches is that they require no information about users or items and, so, they can be used in many situations. Moreover, the more users interact with items the newer recommendations become accurate: for a fixed set of users and items, new interactions recorded over time bring new information and make the system more and more effective.

However, as it only considers past interactions to make recommendations, collaborative filtering suffer from the "cold start problem": it is impossible to recommend anything to new users or to recommend a new item to any users and many users or items have too few interactions to be efficiently handled. This drawback can be addressed in different way: recommending random items to new users or new items to random users (random strategy), recommending popular items to new users or new items to most active users (maximum expectation strategy), recommending a set of various items to new users or a new item to a set of various users (exploratory strategy) or, finally, using a non-collaborative method for the early life of the user or the item.

V. WORK DONE

UNIFY's Front End was designed using the React Framework and Backend was designed using the MERN stack, we implemented the recommendation system using the Brute algorithm of Nearest Neighbor in Collaborative Filtering Method over the dataset created by us and achieved an accuracy of 95.55%, the basic operation performed are shown below in Fig 4 and Fig 5, the accuracy calculation is shown in Fig 6.

Fig 4 Implementation of Recommendation System

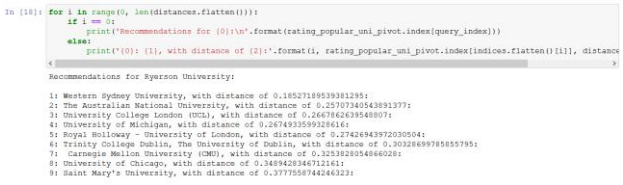

Fig 5 Recommendations

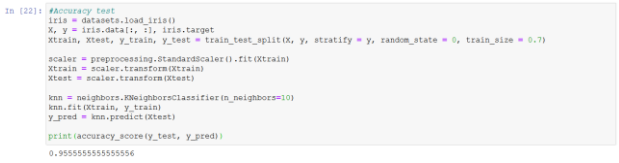

Fig 6 Accuracy Test

VI. DIFFICULTIES FACED

While developing the system the first difficulty we faced was which stack to choose for the development of our system and had to decide between MERN stack and MEAN stack for development, we chose to go to MERN stack as it was more adaptable and reliable in designing a responsive website over the other advantages.

The next difficulty we faced was with dataset, as there was no appropriate data set available for us which could be helpful with our system or a website that had relevant information about the universities, we had to make out own database from the official websites and also referred to the data available on whed.net for what type of information we needed to provide the users with and had to go with collaborative filtering over content based as the data available varied from one website to other, and as discussed earlier in collaborative filter we fast the cold start problem so to get our system up and running for the testing purpose we introduced a test dataset that helped us make our system running for testing and demonstration purposes.

The third difficulty we faced was updating the data on the website, as it'd be necessary to keep the data updated for future users this is where major problem lied it wasn't possible for us to design a script to fetch information from website as each and every university and a different sitemap and since there are thousands of universities each there need to be equivalent number of scripts to fetch the relevant information.

So basically, the system development had three main problem that being availability of proper data, keeping the website up to date based on the information available on the official websites of the universities and getting a proper data base about the universities.

VII. FUTURE SCOPE

- Once the System had proper number of users the Collaborative filtering would be able to provide even better suggestion based on users' responses.
- A hybrid system made up of content based and collaborative filtering can replace the current system to make it even more efficient.
- An application for mobile can be designed using React Native or such other technology.
- Scripts can be designed based on each and every official website to keep a track of updated information.
- The website can be made into a Progressive Website.
- Another Script or a crawler can be designed that would go through a university website collect all relevant data and add it to the database as per requirements

VIII. CONCLUSION

Unify is a solution developed for user, which provides university result to the user. The process makes use of filtering the user interest and

recommendation algorithm in which the user is interested in. Unify is a one touch solution, which only requires user to enter user details at the login/signup page, who wants to save time and avoid the sites which can lead to fishing. Unify takes care of every basic details, developed using MERN stack and revised recommender algorithm which will help user make informed decision.

IX. REFERENCES

[1]. https://developer.mozilla.org/en-US/docs/Web
[2]. https://www.mongodb.com/mern-stack
[3]. https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

Cite this article as :

Soham Manish Mehta, Nidhi Vinod Padhariya, Sakshi Jayesh Patel, "UNIFY - The University Recommender", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 2, pp. 468-473, March-April 2021. Available at
doi : https://doi.org/10.32628/CSEIT2172102
Journal URL : https://ijsrcseit.com/CSEIT2172102