INTELLIGENT REASONING SYSTEM

MASTER OF TECHNOLOGY

Coursera Course Recommender System

Author:	*ID No:*
Chen Zihao	A0261858R
Liu Haoran	A0261945X
Ku Maier	A0261822J
Shen Jingyu	A0261670H

October 30, 2022

Contents

List of Figures

List of Tables

1 Executive Summary

With the explosion of online information, modern people can easily access different fields of knowledge and develop an interest in areas they may not be familiar with. With the gradual maturation of online education platforms, there is a great deal of enthusiasm for people to try out new knowledge online. However, with the variety of courses available on the internet, it is easy to be overwhelmed and unable to choose the one that best suits their background and interests. To help people have an easy and happy start to their e-learning experience, this project developed a course recommender for the Coursera education platform. The objective is to select the best courses that match the user's interests from the vast educational resources and recommend them to the user, helping the user to make the proper choice more easily and quickly. After the user selects one of the courses, the system will calculate the similarity among the courses and keep making recommenders to our users.

Three kinds of reasoning systems—content-based reasoning, similarity-based reasoning, and knowledge-based reasoning—are used by the product. We carefully observed the courses' details and extracted a set of features from them to thoroughly describe Coursera's course. The system also has a user preference setting page for collecting and generalising user preferences into the user profile. Based on the user's profile, the system will recommend the top ten closest match courses against the course descriptions to the user. Besides, the similarity between every two courses is pre-calculated considering all the set of course features and is loaded into the system. Each time the user selects a course of interest, the system will recommend ten similar courses, ranked by their distance. Each element in the set of features disparately contributes to the similarity calculation. Our team assigned different weights to different features based on our research and knowledge, to perform the similarity calculation.

The knowledge base for this project is collected from open source Coursera Course Dataset published on Kaggle, combined with further detailed course information scraped from Coursera's official website. Data preparation and data transformation, such as data cleaning and feature engineering, are applied to the raw data. All processed course features are stored in an SQL database. According to the attribution of similarity-based reasoning, the similarity between every two classes is pre-calculated. The top 10 similar courses' id are then stored in the SQL database.

With the Coursera Course Recommender System, one can approach a suitable set of courses in a short time, saving a lot of unnecessary time in browsing through numerous web pages and picking a satisfying course. It allows you to focus more on the course's content and learning itself.

2 Problem Statement & Objectives

2.1 Market Search

2.1.1 Popularity of online learning

People are now spending more and more time surfing the web. In the field of education, online learning has also become a trend. More and more people choose online education over face-to-face learning due to the convenience, scalability and flexibility of online learning. According to the figure1 from BestColleges 2022 Online Education Trends report[1], the convenience of studying around existing work and family commitments was the biggest motivator of online enrollment. It helps people study anywhere and anytime at their pace: whether it is in daily life or work, whether it is a student, a professional, or a stay-at-home mother, all people can acquire skills or certificates through online learning to achieve better life goals.

In one word, online learning has many benefits: It can help people from all walks of life get promoted in their current field, move to a full-time career in a different industry, or pursue a new hobby without time and space constraints.

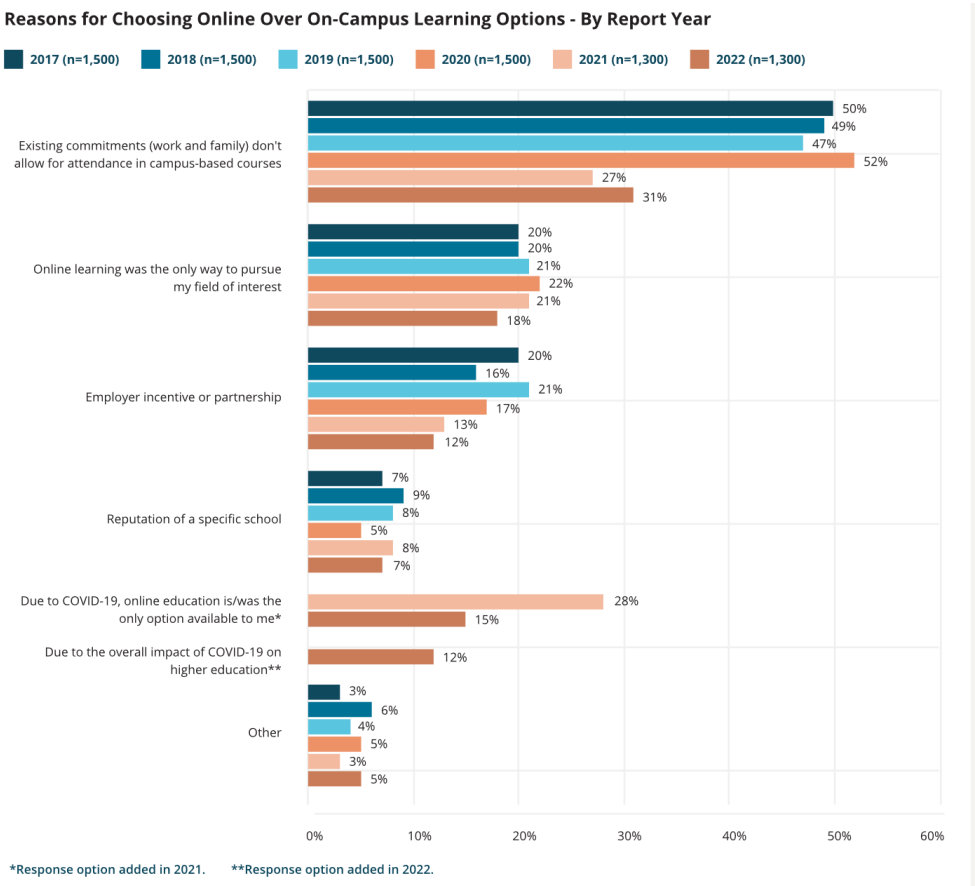

Figure 1: BestColleges 2022 Online Education Trends

2.1.2 Online learning industry market size and trends

In recent years, with the rapid development of the Internet, the online learning industry has begun to develop rapidly. It is estimated that the online learning industry will reach $325 billion by 2025[3]. Especially after COVID-19 raged around the world, many students and aspiring people have turned to online learning.

E-Learning Market Report Coverage

Report Coverage	Details
Base Year:	2021
Market Size in 2021:	315 billion (USD)
Forecast Period:	2022 to 2028
Forecast Period 2022 to 2028 CAGR:	20%
2028 Value Projection:	1 trillion (USD)
Historical Data for:	2018 to 2021
No. of Pages:	300
Tables, Charts & Figures:	388
Segments covered:	Technology, Provider, Application, and Region

Figure 2: E-Learning Market Report Coverage

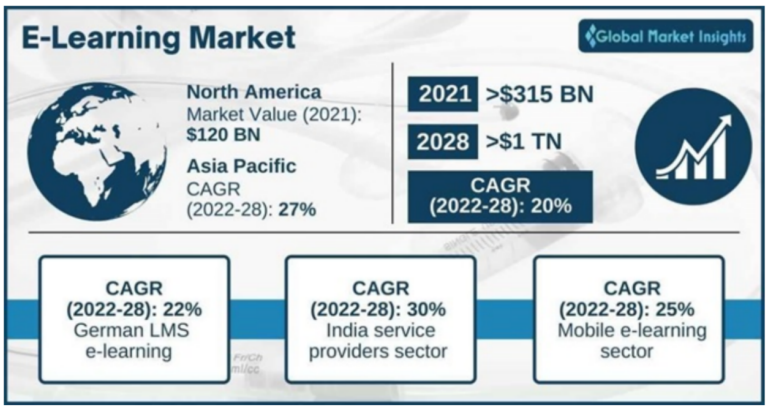

Figure 3: E-Learning Market

According to figure2 and figure3 from Global Market Insights, E-learning Market size surpassed USD 315 billion in 2021 and is projected to observe 20% CAGR (Compound annual growth rate) from 2022 to 2028.

2.1.3 Existing online courses recommendation systems, features and pricing

When online learning became a popular, common and easy learning mode, some large online course recommendation websites (Massive Open Online Courses) emerged. There are some very beneficial platforms like Coursera, Udemy, edX and LinkedIn learning that offer people productive tools for choosing and learning online courses. Such platforms prove to be extremely advantageous for skillful people from different fields like Architecture, Animation, Photography, Health and Wellness, Finance, Marketing, Designing, Music, Crafts, etc. In this section, we will focus on comparing and studying some well-known online course recommendation systems. The comparison is shown in the table below.[2]

Table 1: Comparison of some well-known online course recommendation systems

Websites	Feature	Field	Number of courses	Pricing
Udemy	1. Once purchased, lifetime access to purchased course materials 2. Many free courses with a wide range of topics 3. Classify and recommend courses according to subjects, level, language, pricing and video duration	Design, Development, Marketing, IT and Software	130,000+	1.Free courses 2.$19.99-$199.99
eDX	1. Courses are offered by famous universities or companies like Harvard, Berkeley and Microsoft 2. Many free courses but relative small scope of topics (specializes in higher education and science) 3. Classify and recommend courses according to subjects, programs, degrees, schools and partners	Data Science, Computer Science, Business, Engineering	3000+	1.Free courses 2.$50- $300/course $10,000–$25,000/degree
Coursera	1. Provide financial aid and scholarships 2. Plenty of free course 3. Classify and recommend courses according to subjects, degrees, schools	Arts and Humanities, Business, Computer Science, Data Science, IT	4600+	1.free courses 2.$0-$99/course $9,000–$45,000/degree
LinkedIn Learning	1.Provide financial aid and scholarships 2.Plenty of free courses 3. Classify and recommend courses according to subjects, degrees, schools	Project Management Institute (PMI), programming languages, illustration	16,000+	1.$29.99/month or $239.88/year
SkillShare	1.The best online course platform for creative people 2.Thousands of free courses with plenty of course resources 3.Students can share creative work with each other and get comments from others	Animation, design, illustration, lifestyle	29,000+	1. Free courses 2.$19.99/month or $99.99/year
Treehouse	1.Specialize in coding training 2.Many programs provide practical, hands-on material	Android, API, C#, Computer Science, CSS, Data Analysis	285+	1.$29.99/month or $199/month

From the comparison of the six well-known course recommendation systems in the figure below, It can be concluded to three points: 1.Most courses are charged for users. 2.Courses are classified and recommended according to subjects, degrees, schools, etc. without considering personal preferences of users. 3.Some recommendation websites are dedicated to recommending subjects in certain fields like IT, business, etc. while some are in general fields.

2.1.4 Areas of improvements

According to the previous chapter, the course recommendation system in this project has 8000+ courses that concern IT, business, language and other fields. Compared with most online course recommendation systems on the market, this system has the following three characteristics:

1. Completely free for users
2. Distinctive UI design for comparing similar courses
3. Recommend courses by courses

The most prominent feature and advantage of this system is that it can not only recommend courses according to user preferences, but also recommend courses according to courses. Specifically, after the system recommends 10 courses according to the user's preference, the user can click on a course of interest ran-

domly, then the system will recommend another 10 similar courses according to this course. Such design not only facilitates the user to discover courses of interest, but also facilitates the user to choose the best study course by comparing similar courses. In summary, such improvements increases the flexibility, scalability and convenience of the system.

2.2 Aims and Objectives

2.2.1 Aims

Course recommendation system in this project aims to use 6000+ available courses as the dataset. What' more, it will be designed with distinctive UI interface to help users compare similar courses so as to choose the most suitable course. The system aims to implement two functional modules: recommendation of courses according to users' preference and recommendation of courses by courses.

2.2.2 Objectives

The system consists of two parts: front-end and back-end. Front-end will provide smooth UI experience in the form of web-page. Back-end will be responsible for storing and manipulating data. Users interact with website pages and then will be recommended 10 courses of her his interests.

3 Solution

3.1 Knowledge Modelling and Acquisition

3.1.1 Knowledge Acquisition

Before knowledge modelling could be implemented, some information was necessary to be gathered and reviewed to build up the initial knowledge base for the project. Various sources of information were pursued, from existing state-of-art to domain knowledge from websites and subject matter experts. The original data was collected from Kaggle website to be used in our project. There were two different sources of data, the first one contained 3416 data and the second one contained 6565 data.

- https://www.kaggle.com/code/sagarbapodara/coursera-course-recommendation-system-webapp/data
- https://www.kaggle.com/datasets/trajput508/coursera-data

After got the original data, we needed to do data preprocessing because of the messy data. In machine learning tasks, the most important part of project is the quality of raw data. If data is good enough, a normal or simple model can also get a good performance in a particular problem. So, the data preprocessing process is significant. In these two datasets, there were some similarities and also some differences between each other.

In the first dataset, according to the course URL, we redirected to the respective website and found the courses which were not existed now and dropped this kind of data.

In the second one, analyzing the features of the data, we found the course type would be no use for the recommendation as well as the instructors of the particular courses. The duration data column was too messy and very hard to transform to a good format in recommender. So we dropped all of these three columns in the dataset. After that, the identical parts of the two data remained course_name, university_name, course_language, course_rating, course_level and course_detail. We still had another problem, what about the skills column in the first dataset? After our discussion, we decided that we would use two different methods during the recommendation part. The skills column was used to recommend based on users' preference. The course_detail column was for courses recommended according to users' click. Therefore, to combine these two datasets together to provide more data in recommendation, we needed to get some information about skills in the second one.

We crawled skills which we could get after learning these courses to the second one using Selinium python package. Selenium is an open-source tool that automates web browsers. It provides a single interface that lets users write test scripts in programming languages. Selenium could be used to automatically direct to the course link which we needed, and got the required information inside the website. After that, we were able to concat these two datasets together. Also, as there were different languages in the dataset, we needed to translate them into English. So the course_detail column data were all translated into English for further use using Google translation API.

For front-end part use and also for user convenience, we also crawled the course link and course image in order to make it more reasonable and user-friendly when we wanted to do a recommendation. Finally, the final dataset was formed.

3.1.2 Knowledge Representation

After acquiring the data required for the project, the next step was to represent the knowledge. Knowledge representation is the field of artificial intelligence (AI) dedicated to representing information about the world in a form that a computer system can use to solve complex tasks such as diagnosing a medical condition or having a dialog in a natural language. Knowledge representation incorporates findings from psychology about how humans solve problems and represent knowledge in order to design formalisms that will make complex systems easier to design and build. Knowledge representation also incorporates findings from logic to automate various kinds of reasoning, such as the application of rules or the relations of sets and subsets.

First, based on the knowledge acquired we formalized the categorical and the numerical data we would utilize in our recommender system as seen below in table2:

Table 2: Knowledge Data Type and Categories

Course Level	University Name	Course Language	Course Skills
Beginner	Coursera Project Network	English	Marketing
Intermediate	Peking University	Spanish	Strategy
Advanced	National Taiwan University	French	Artificial Intelligence
None	...	...	...

After defining the data to be used, formal logic rules were then created with the categorical data:

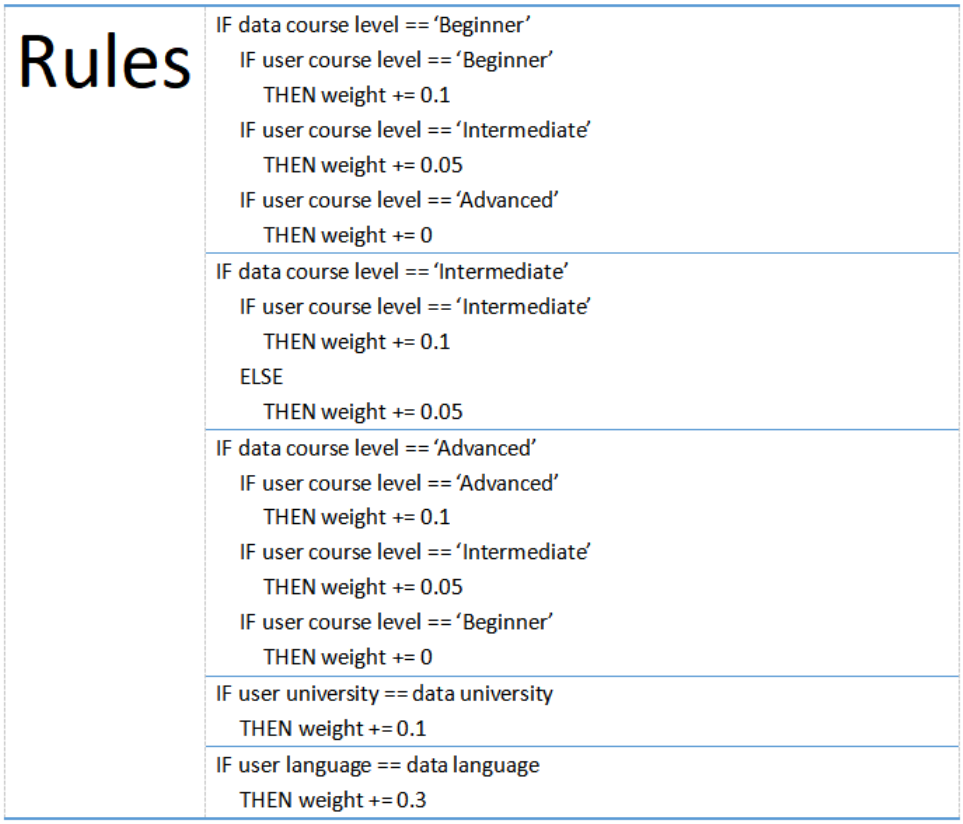

Figure 4: Knowledge Rules and Logical Inference

3.1.3 Knowledge Implementation (Model)

Combining the tacit knowledge acquired, the categorical data and associated rules, we developed the Flowchart to model the course recommender system. This guided the flow of how the knowledge and rules were utilized in a typical user workflow.

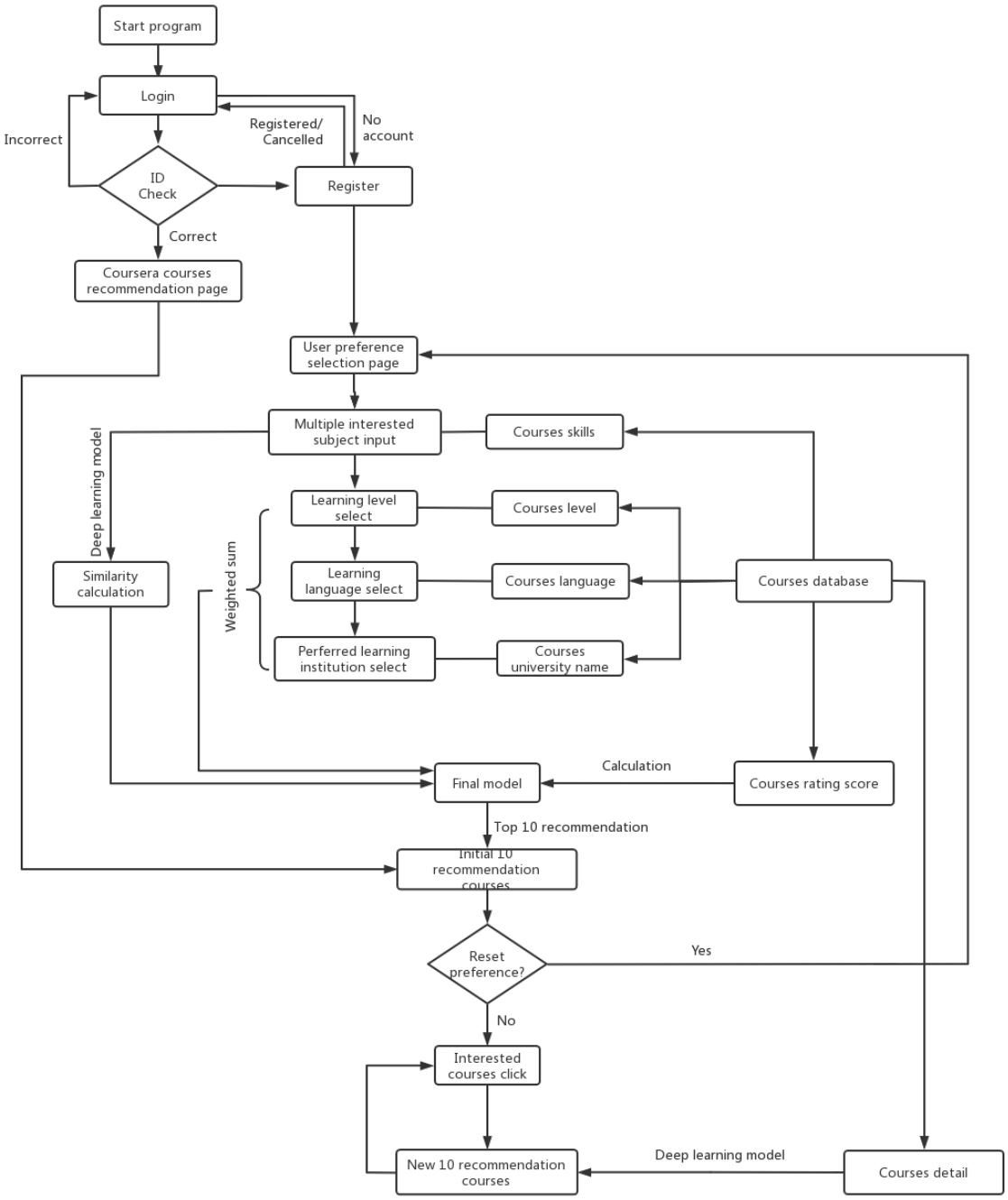

Figure 5: Knowledge Flowchart and Diagram

3.2 System architecture

In order to provide users with a better course recommendation experience, our course recommendation system consists of a front-end interaction page, a course recommendation model and a back-end web architecture. The system architecture is shown in the Fig.6. We implement the front-end page by using HTML+CSS+JavaScript technology stack to provide a beautiful and intuitive interface for users to interact with. The back-end implementation is built by using Django library, which is a high-level Python web framework. In our system, it is responsible for receiving requests from users to the back-end via interactive pages, distributing these requests to different views via routing, and returning the results to the front-end interactive pages for display.

The course recommendation model plays a key role in our system. In order to calculate the similarity between courses, we extract the text features in the course descriptions and apply different weight values based on the importance and meaning of the different fields, and finally calculate the similarity between different courses to achieve the function of recommending courses based on courses information. In addition, a similar algorithm will be adopted for recommending courses based on user input. To speed up the course recommendation process, we put all the course recommendation results into the model first for calculation and store the returned results in a field in the database. This allows our system to respond faster to customer requests when the user clicks on the course page.

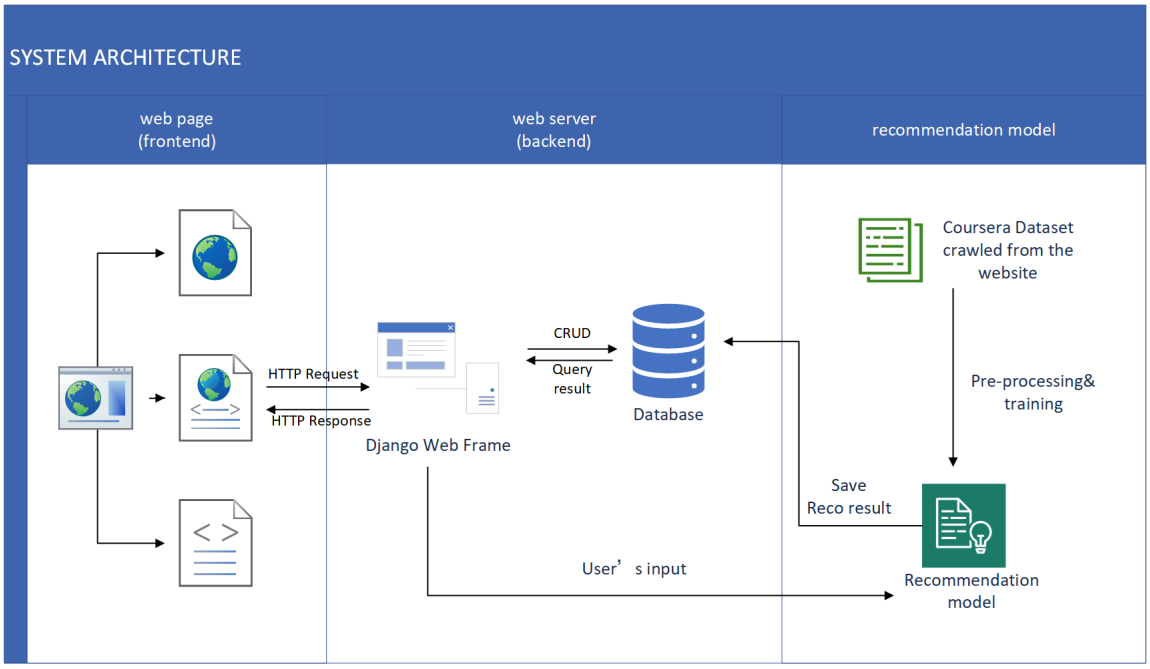

Figure 6: System Architecture

3.3 Recommender Logic

3.3.1 User-preference courses recommender

For the first recommendation of courses, in order to solve cold-start problems, a cosine similarity-based recommendation algorithm was implemented. The reason why cosine similarity was applied instead of collaborative filtering (CF) was due to the fact that when a new person signs up to the app, we would only have his/her basic information such as: university name, preferred course skills, preferred course level and preferred course language. With such limited information, we would be unable to apply either item-based or user-based CF effectively. As such we developed a recommendation algorithm using the specific data based on the similarity between user preference and the existing courses data. After signing up, users will be asked to fill up the information which is useful for recommendation. We presented top 10 courses based on the user-preference recommender.

Using the exercise selected as an indication of the user's preference, we apply cosine similarity plus giving a specific weight in different features to find other similar courses to recommend to the user as user-preference recommendation. The workflow for the user preference recommendation is as seen in the diagram 7.

As seen in the diagram, we utilized different columns (university_name, course_language, course_level) and gave them respective weights. The most important part was the similarity based recommender according to the user preference and existing courses skills. BERT algorithm would be used to calculate the similarity between each other using a string form. It was a deep learning pre-trained model with transformer. Using GPU to train, the result was concise and efficient. This method returned word vectors. Applying cosine similarity, we could get the similarity between two sentences or words. Cosine similarity is one metric used to measure how similar two items are. In math, it works by measuring the cosine angle between two vectors when projected in multi-dimensional space with an output from -1 to 1.

Figure 7: Basic information page where new users are required to fill up

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{1}$$

Supposed we inputted some preference information, using sentence transformer and cosine similarity calculation, we could get the similarity value between user's input and all of the existing data.

After applying cosine similarity to the word embedding vector, we got a cosine_sim numpy array, with courses as the column headers and the data represents the cosine similarity score. As seen in the snapshot of the array in the figure 8, the result showed the similarity between user selections and existing data.

	0	1	2	3	4	5	6	7	8	9	...	8600	8601	8602	8603	8604	8605	8606	8607	8608	8609
0	0.706	0.446	0.189	0.638	0.09	0.513	0.262	0.526	0.327	0.54	...	0.255	0.206	0.296	0.275	0.427	0.264	0.273	0.424	0.26	0.303

Figure 8: User cosine similarity list

After using cosine similarity measuring similarity between user preferred skills and existing courses skills, we were able to add features mentioned above to weight the similarity based on other information users provided. We then generated a list of similar courses using the initial preferred information (in figure 7) selected by the user. After generating the list, we then sorted the courses similarities in descending order to get the most similar courses on the top.

$$finalList = similarityList + levelList + universityList + languageList + ratingList \tag{2}$$

As can be seen in the image 9, the course_id "0" had the highest recommendation score of 1.056081 because it relatively matches user's preference. From here we could then select the top N number of courses to pass to the user as the initial recommendation. In this case, N equalled to 10. With the first set of courses recommended, the user can then choose the courses which they like, the recommended courses based on user click will be shown afterwards. This knowledge-based and similarity-based reasoning will serve as inputs for the subsequent content-based user courses recommender.

3.3.2 Subsequent content-based user-courses recommender

For the second part of recommendation of courses, actually, the principle of recommendation was similar to the first user preference recommender. The difference between these two parts was, we needed to calculate the similarity between user chosen and the existing information in real-time. For the second one, as the similarity between each courses could be calculated in advance, we could save the similarity matrix before recommendation. The advantage of this process is that when user choose the courses they like, server doesn't need to calculate the similarity, just simply call a function to get the value then return the recommender results to the user. As we mentioned above, we used sentence transformer to calculate the similarity between course detail

	id	Recommendation score
0	0	1.056081
1	6018	1.024709
2	1358	1.011425
3	703	0.991654
4	3904	0.988153
...	...	...
8605	5137	0.017857
8606	2761	0.015176
8607	5165	-0.025613
8608	5065	-0.062770
8609	4852	-0.094941

Figure 9: User preferred courses ID

```
User preferred insititution name: Coursera Project Network
User preferred course level: Beginner
User preferred course language: English
User preferred course related skills: Marketing, Strategy, Content, Machine Learning
Recommended courses based on user's preference:  Approve Social Media Posts with Zapier and Trello
Recommended courses based on user's preference:  Building a Business Presence With Facebook Marketing
Recommended courses based on user's preference:  Custom Attribution Modeling with Google Analytics
Recommended courses based on user's preference:  Building Digital Media using Graphic Design in Google Slides
Recommended courses based on user's preference:  Product Development: Customer Journey Mapping with Miro
Recommended courses based on user's preference:  Set up Google Analytics for a single page website
Recommended courses based on user's preference:  Introduction to Customer Segmentation in Python
Recommended courses based on user's preference:  Create a Storyboard using Canva
Recommended courses based on user's preference:  Build Data Analysis and Transformation Skills in R using DPLYR
Recommended courses based on user's preference:  Advertise YouTube Videos with Zapier
```

Figure 10: User preferred courses recommendation

of user clicked course and detail of existing courses data, which is shown in figure 11

	0	1	2	3	4	5	6	7	8	9	...	8600	8601	8602	8603	8604	8605	8606	8607	8608	8609
0	1.000	0.206	0.055	0.423	0.050	0.393	0.101	0.098	0.251	0.342	...	0.165	0.077	0.030	-0.003	0.177	0.169	-0.035	0.329	0.148	0.170
1	0.206	1.000	0.041	0.201	0.538	0.131	0.124	0.132	0.258	0.289	...	0.175	0.269	0.013	0.243	0.414	0.259	0.022	0.283	0.246	0.337
2	0.055	0.041	1.000	0.065	-0.050	-0.042	0.116	0.139	0.091	0.006	...	0.210	0.108	-0.057	0.089	0.174	0.013	-0.006	0.123	0.012	0.027
3	0.423	0.201	0.065	1.000	0.087	0.241	0.099	0.066	0.125	0.100	...	0.111	0.027	-0.045	0.051	0.177	0.186	-0.000	0.125	0.130	0.159
4	0.050	0.538	-0.050	0.087	1.000	0.086	0.145	0.160	0.145	0.193	...	0.119	0.221	0.076	0.264	0.239	0.215	-0.068	0.256	0.121	0.299
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8605	0.169	0.259	0.013	0.186	0.215	0.111	0.203	0.265	0.332	0.172	...	0.293	0.326	0.032	0.098	0.293	1.000	0.045	0.290	0.318	0.165
8606	-0.035	0.022	-0.006	-0.000	-0.068	0.050	-0.152	0.041	-0.029	-0.030	...	0.091	0.024	0.122	-0.016	0.034	0.045	1.000	0.046	-0.000	-0.014
8607	0.329	0.283	0.123	0.125	0.256	0.196	0.257	0.380	0.336	0.183	...	0.253	0.266	0.102	0.076	0.375	0.290	0.046	1.000	0.245	0.267
8608	0.148	0.246	0.012	0.130	0.121	0.180	0.307	0.268	0.214	0.183	...	0.379	0.432	0.049	0.082	0.286	0.317	-0.000	0.245	1.000	0.124
8609	0.170	0.338	0.027	0.159	0.299	0.056	0.169	0.118	0.139	0.104	...	0.084	0.161	0.036	0.336	0.278	0.165	-0.014	0.267	0.125	1.000

Figure 11: Cosine similarity matrix

In order to made our recommendation more precise and reasonable, we added valuable features, including course_level, course_language, university_name. The course_rating information was also taken into account and we would add it in the similarity matrix. Because the rating score was between 0-5, we set 4 was an average score, this rating score was then transfer based on formula 3

$$course\_score = (course\_rating - 4) * 0.1 \tag{3}$$

Combining all of the calculated matrixes together, the final matrix didn't just indicate similarity, the meaning of the result was the recommender value. The higher it was, the more possibility it would be recommended.

$$finalMatrix = similarityMatrix + levelMatrix + universityMatrix + languageMatrix + ratingMatrix \tag{4}$$

	0	1	2	3	4	5	6	7	8	9	...	8600	8601	8602	8603	8604	8605	8606	8607	8608	8609
0	1.350	0.436	0.195	0.693	0.220	0.563	0.371	0.288	0.631	0.492	...	0.365	0.267	0.190	-0.003	0.317	0.429	0.205	0.489	0.408	0.390
1	0.406	1.380	0.331	0.421	0.758	0.251	0.344	0.372	0.488	0.489	...	0.225	0.409	0.123	0.193	0.404	0.369	0.212	0.393	0.356	0.507
2	0.155	0.321	1.390	0.185	0.070	0.078	0.236	0.379	0.221	0.106	...	0.260	0.248	0.053	0.039	0.164	0.123	0.184	0.233	0.122	0.197
3	0.723	0.481	0.255	1.320	0.207	0.461	0.419	0.206	0.455	0.300	...	0.261	0.267	0.165	0.101	0.267	0.396	0.290	0.335	0.340	0.429
4	0.300	0.868	0.190	0.257	1.270	0.156	0.315	0.450	0.425	0.343	...	0.219	0.311	0.136	0.164	0.279	0.375	0.072	0.316	0.281	0.419
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8605	0.419	0.389	0.153	0.356	0.285	0.281	0.373	0.455	0.612	0.222	...	0.493	0.516	0.192	0.098	0.433	1.360	0.285	0.450	0.578	0.385
8606	0.165	0.202	0.184	0.220	-0.048	0.270	0.068	0.181	0.201	0.070	...	0.241	0.264	0.332	0.034	0.124	0.255	1.390	0.256	0.210	0.256
8607	0.479	0.413	0.263	0.295	0.226	0.366	0.427	0.470	0.516	0.233	...	0.353	0.456	0.362	0.176	0.415	0.450	0.286	1.360	0.405	0.487
8608	0.398	0.376	0.152	0.300	0.191	0.350	0.477	0.458	0.494	0.233	...	0.579	0.622	0.209	0.082	0.426	0.577	0.240	0.405	1.360	0.344
8609	0.370	0.518	0.217	0.379	0.319	0.276	0.389	0.258	0.369	0.204	...	0.234	0.401	0.246	0.386	0.368	0.375	0.276	0.477	0.335	1.370

Figure 12: Final courses recommendation matrix

The weight of each feature was determined by experience. After a large number of times attempting the recommender results, finally we got a relatively good weight in the recommender.

```
User clicked course:  AI Workflow: Enterprise Model Deployment
Top 10 recommendation based on user's interest: AI Workflow: Data Analysis and Hypothesis Testing
Top 10 recommendation based on user's interest: AI Workflow: AI in Production
Top 10 recommendation based on user's interest: AI Workflow: Business Priorities and Data Ingestion
Top 10 recommendation based on user's interest: Machine Learning Rapid Prototyping with IBM Watson Studio
Top 10 recommendation based on user's interest: AI Workflow: Enterprise Model Deployment
Top 10 recommendation based on user's interest: Logistic Regression Tutorial with Python and Numpy
Top 10 recommendation based on user's interest: Tutorial: Exploratory Data Analysis with Python and Pandas
Top 10 recommendation based on user's interest: Basic Data Analysis and Model Building using Python
Top 10 recommendation based on user's interest: First Steps in Linear Algebra for Machine Learning
Top 10 recommendation based on user's interest: Introduction to Artificial Intelligence (AI)
```

Figure 13: Courses recommendation based on user's interest

For efficient use and easy to call the result, the top 10 recommender result was stored in the database. The format of it is shown in figure 14:

reco_course_id
3377 8389 2975 2593 1203 1715 1358 6883 727 6028
3283 2785 399 1494 488 454 476 1119 1130 619
162 1900 446 5019 4712 3731 1899 318 1896 3545
1290 3659 727 1139 222 3377 28 2593 223 1177
349 705 2568 4969 2758 477 2785 488 1658 476
4679 7430 6529 7764 4824 4990 4989 5456 4209 1230
1638 4377 2550 936 7810 304 8567 8462 366 4280
2796 1735 892 4708 3479 2790 5008 1763 175 1391
1204 1286 637 1188 1212 1195 1211 1455 7012 1225
955 224 341 1251 1338 342 1126 1133 5107 1234
3287 3266 3451 1936 3284 2756 474 1944 1119 3294

Figure 14: Top 10 recommendation of each course

3.4 System Back-end

3.4.1 Web server framework

The back-end of the system consists of three parts, namely the web service architecture, the database and the recommendation model. web service is the core part of the back-end and is responsible for linking the database, the model and the web pages. The web architecture used in this system is Django. Django is independent and a complete set in itself. It means that it does not require any other external solution. So, it fits in well with the rest of the system. When Django receives a request from a front-end page, it distributes the request to different views based on the routes set. The views process the request received at the URL, manipulate the database or recommendation model as needed, and implement the rendering of the front-end page. For example, in a user login scenario, the front-end page would send the request to the '/login/' URL, which includes key information such as username and password, and Django would receive an HTTP POST request from the front-end page and find the login view corresponding to that URL based on the route settings.

The login view compares the information in the request with the information in the database and returns the results of the query.

The database stores the user information and course information for Django to query. As the course recommendations are offline, i.e. the data in the database is the result of the model's calculation of the course recommendations, only the recommendation model based on user input is running in the back-end of the system. When a request is received from the '/profile/' URL, Django calls the corresponding view. This view passes the user input, such as skills, language and level, to the recommendation model. The model calculates the similarity between the skill and course information entered by the user, while adding different weights to the other inputs to calculate the overall similarity and return a recommendation.

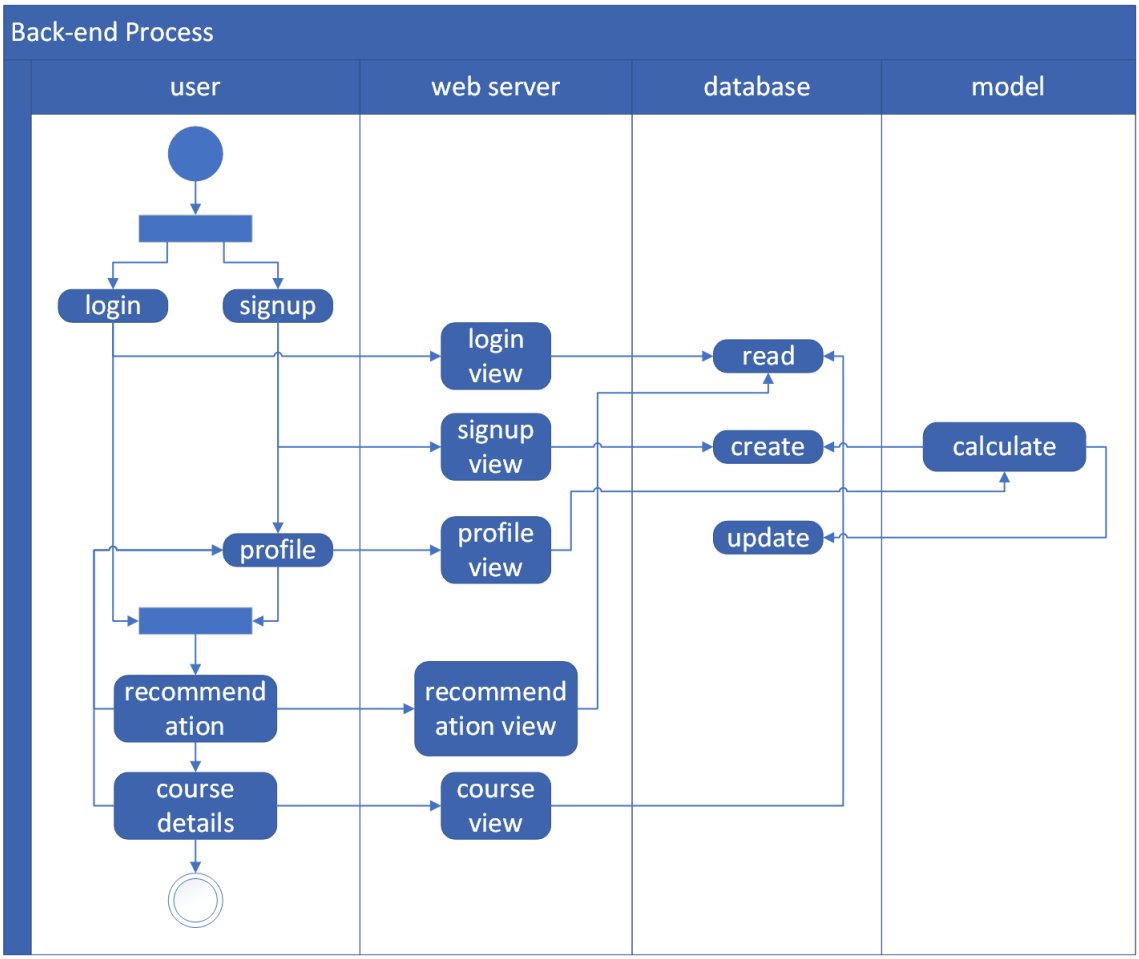

Figure 15: System back-end flow chart

3.4.2 Database

As the Coursera dataset used in this system has approximately 8,000 pieces of data, storing this data in files to read would incur a very large memory overhead. In addition, it would not be safe to store user information in a file. At the same time, Django uses the MVC design pattern, which makes it simple to implement operations on databases. So this system uses SQLite as the database to store the relevant data. The data from the model pre-processing forms the course information data table. The database contains not only course information, but also personal information about the user in order to provide the user with personalized recommendation results. The database consists of the following two tables.

Course Table The course table stores information on more than 8,000 courses crawled from coursera, as well as the recommendations calculated by the recommendation model. The table has eleven fields in total, including course_id, course_name, university_name, course_language, course_rating, course_level, course_detail, course_link, course_image, reco_course_id, where course_id is used as the primary key and foreign key of this data table which allow other tables to reference and connect to. And the content stored in reco_course_id is the

corresponding course_id of the recommended courses calculated by the model.

User Table The user form stores the user's authentication information and personal preferences. Because Django includes a user authentication module that works better with other features, the user form inherits the django.contrib.auth.models.AbstractUser class. So in addition to the fields that come with AbstractUser, the user form also includes level, skills, language, university and reco_course_id. Similarly, the reco_course_id field is the course recommendation result calculated by the user recommendation model.

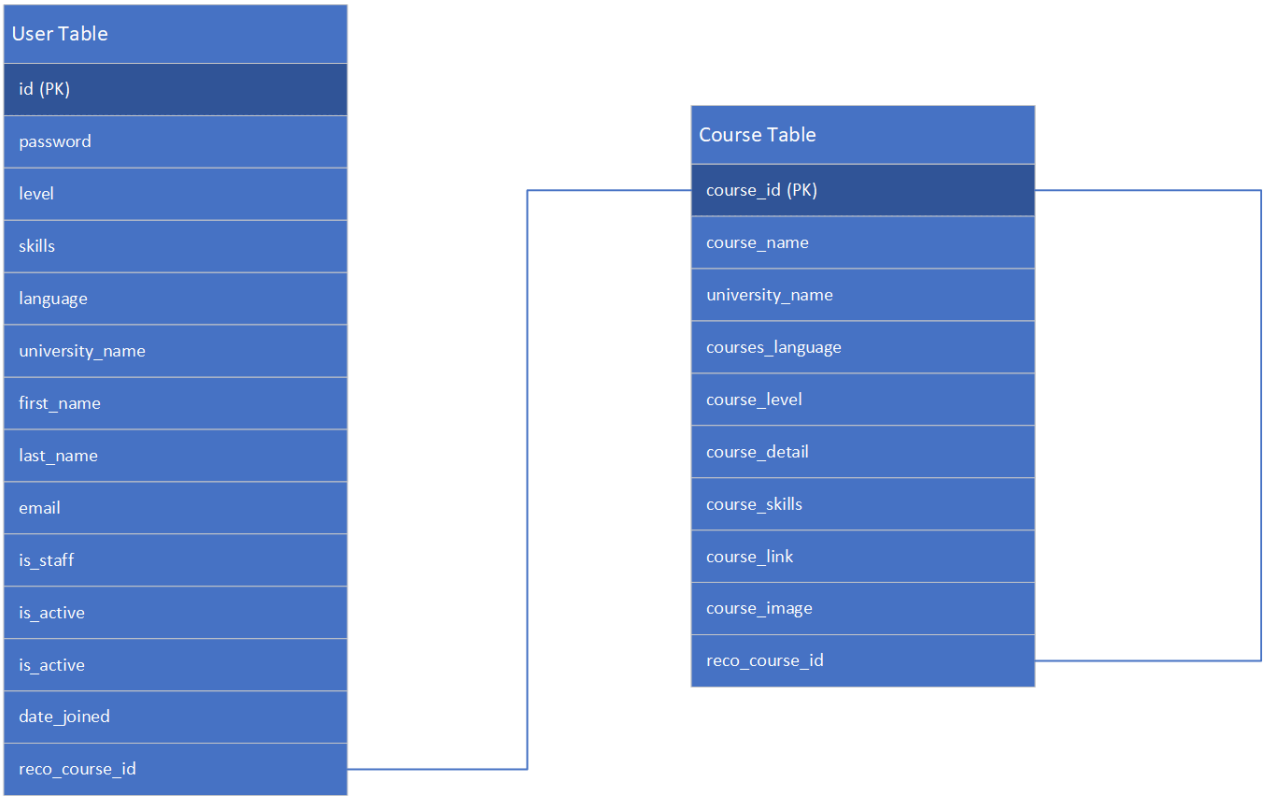

Figure 16: Database relationship diagram

3.5 System Front-end (User interface)

The UI is the bridge between our product and users, since users can only interact with the product through the UI. It determines the user's impression and experience of the product. Our user interface is designed to present our product clearly and unambiguously, with all icons and labels being concise and convincing, intending to enhance our user experience.

An HTML, CSS, and JavaScript-based web application serves as the front end of the product. The front-end documents are integrated into the Django framework to provide quick and effective request and response transmission between the front-end and back-end.

An administrative component and a recommender component make up the web application. The administrative part contains a login page and a sign-up page. The recommendation section consists of one user preference setting page and one dynamic Coursera course recommender page for a specific user, which presents the particularly recommended Coursera courses for a specific customer. The figure. 17 displays the user flow diagram.

The key pages that highlight the system's goals and functions will be detailed and presented in Project Implementation.

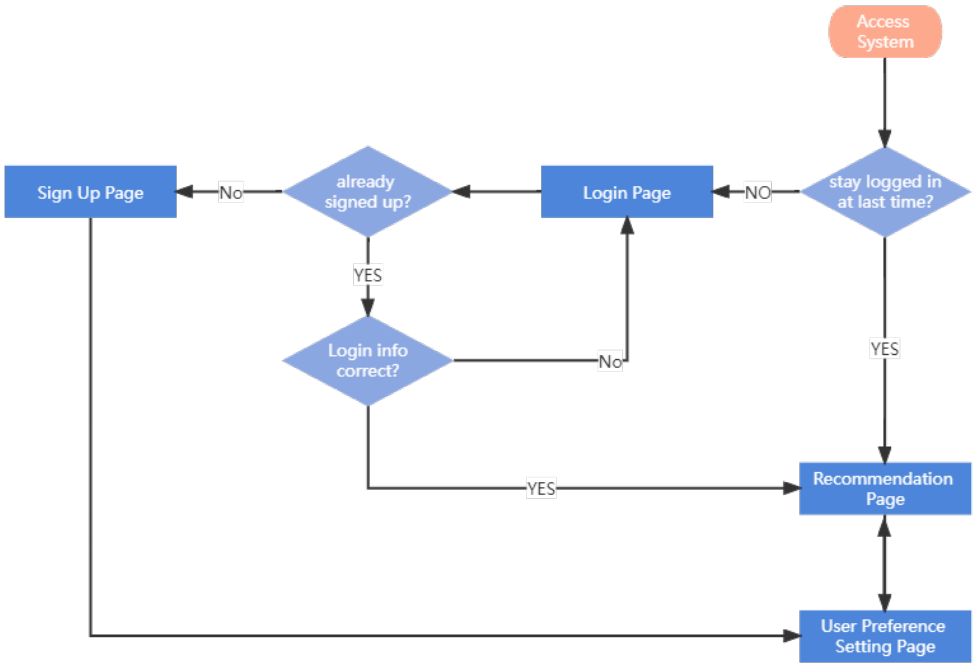

Figure 17: The user flow diagram.

4 Project Implementation: A Case Study

Sascha is a finance industry employee from the UK. With the concept of digital finance being introduced and becoming famous, Sascha wanted to learn about it online to enhance her competitiveness. Sascha had never been exposed to the concept of digital or AI and was overwhelmed by the number of courses available, so she chose to try out Coursera Course Recommendation System to help her get started with e-learning.

Sascha first created an account using her email (fig.18).

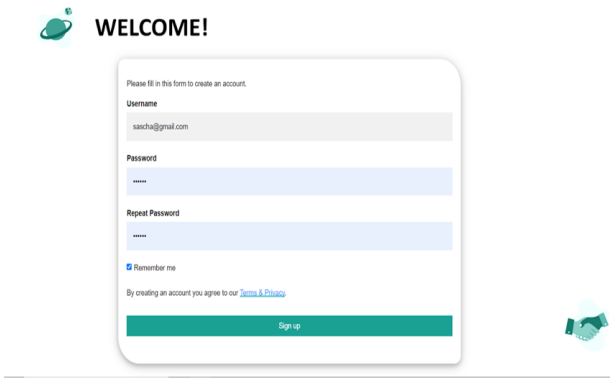

Figure 18: The sign up page.

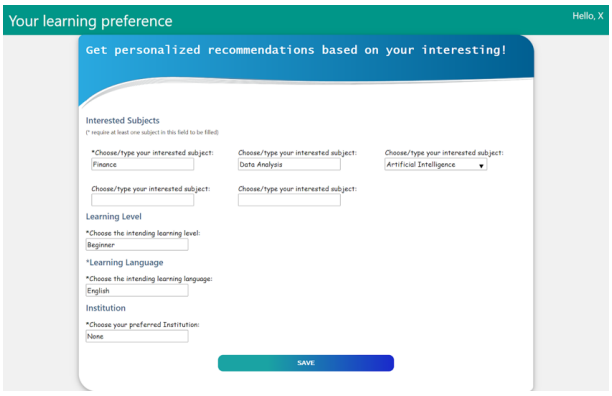

Figure 19: The user preference setting page.

After Sascha has registered an account, the system automatically redirects this new user to the user preference setting page (fig.19). Her preference input enriches her profile to let the system calculate the recommended courses.

In subsequence, the user recommendation page (fig.20) will present 10 recommended Coursera courses based on similar-based reasoning across the course database and user profile. In the user recommendation page, *Building Data Analysis and Transformation Skills in R using DPLYR* caught Sascha's attention, so she clicked the course's title. She was then directed to the course recommendation page(fig.21), which has details about the course she selected along with ten others that are comparable and would be of interest to her.

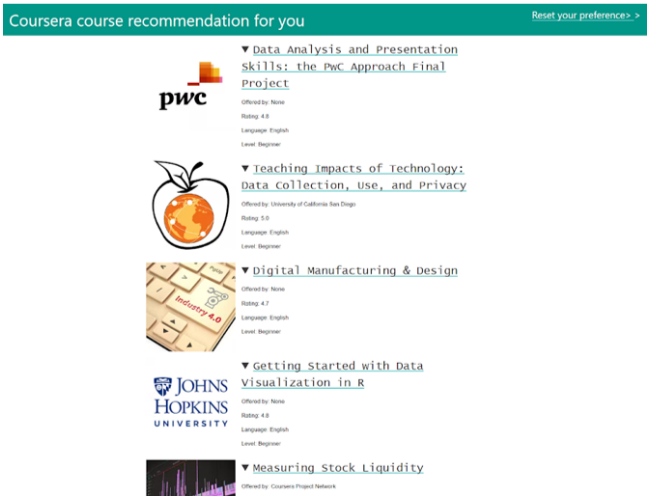

Figure 20: The first course recommendation page.

Sascha noticed that one of the ten courses in the right row, which lists similar courses, accurately reflects her expectations. She clicked the course title and discovered that the right row had been refreshed and offered a new set of ten recommended courses. In addition, the course description on the left had been changed to reflect the latest course she had chosen (fig.22).

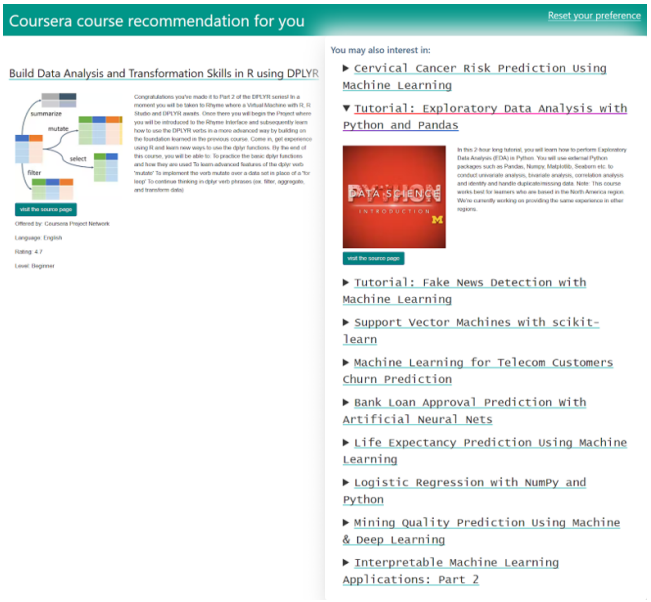

Figure 21: The second course recommendation page.

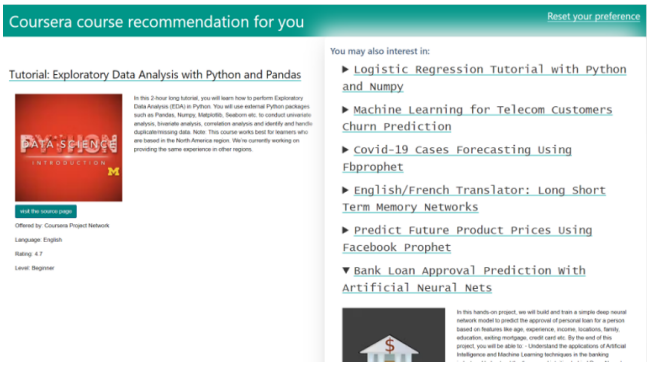

Figure 22: The third course recommendation page.

This case study of Sascha concretely describes the convenient and efficient operational processes of the system. It also specifies the function of each part of the user interface. It presents the strong ability of the system to reason and act based on Sascha's given information and the memorized course data. It also visualises the functionality and usage scenarios of different reasoning systems. It promises an improvement in the user experience and system reliability.

5 System Performance

To evaluate the performance of the recommendation system, we used mean average precision (MAP). Mean average precision (MAP) for a set of queries is the mean of the average precision scores for each query.

As we have two types of recommendations: recommendations based on user input and recommendations based on course information, the following evaluation will also be divided into two parts.

5.1 Recommendation For User

Ten use cases were used to simulate scenarios that might be encountered in real life use. In each use case, the user enters different information and we evaluate the model based on the top5 results and top10 results it returns. In the table, 0 means that the recommended result is not or not fully relevant to the input, and 1 means that the result is relevant to the input.

Table 3: Evaluation table for recommendation for user

Case No.	User Inputs	R1	R2	R3	R4	R5	MAP (Top5)	R6	R7	R8	R9	R10	MAP (Top10)
1	Artificial Intelligence, Machine Learning/ Advanced/English	0	1	1	1	1	0.8	1	0	0	1	1	0.7
2	Deep Learning, Data Analysis, Machine Learning/ Beginner/English	1	0	0	1	1	0.6	0	1	1	0	1	0.6
3	Computer Science, Python /Intermediate/English	1	1	0	1	1	0.8	1	0	1	0	0	0.6
4	Finance,Stock, Data Analysis/ Beginner/English	1	1	0	1	1	0.8	1	0	1	1	1	0.8
5	Accounting, Management/ Beginner/English	1	1	0	0	1	0.6	1	0	1	0	0	0.5
6	History/ Beginner/English	1	1	1	0	1	0.8	0	0	1	1	1	0.7
7	International Relationship/ Intermediate/English	1	1	1	1	0	0.8	0	1	0	1	0	0.6
8	Music/Art/ Advanced/English	1	0	1	1	1	0.8	0	1	0	0	0	0.5
9	Education/ Beginner/English	1	1	0	0	1	0.6	1	1	0	0	1	0.6
10	Psychology/ Beginner/English	0	1	1	1	1	0.8	1	1	1	1	0	0.8
					Avg		0.74				Avg		0.64

The data in the table 3 shows that the model performs reasonably well on the Top 5, with a MAP of 0.74. Also, the first result returned by the model is correct in most cases. It is worth noting that, due to the limitations of the data source, the model's performance is affected when the user enters fewer keywords or when the keywords correspond to colder topics.

5.2 Recommendation For Course

We used python's random.randint function to generate ten random numbers, each corresponding to a course id. We then evaluated the model in the same way as in the previous section, based on the recommendations from visiting "/course/course_id".

Table 4: Evaluation table for recommendation for course

Course ID	Course Name	R1	R2	R3	R4	MAP (Top5)	R6	R7	R8	R9	R10	MAP (Top10)
4134	Response Surfaces, Mixtures, and Model Building	1	0	1	1	0.8	0	0	1	0	0	0.5
41	Strategically Build and Engage Your Network on LinkedIn	1	1	1	0	0.6	1	0	1	0	0	0.5
732	Building a Hiring Plan by Analyzing Past Data in Sheets	1	1	1	1	0.8	1	1	0	0	1	0.7
93	A Start Guide: Product Marketing Using G Suite	1	0	1	1	0.8	1	1	1	0	0	0.7
6976	AI Workflow: Feature Engineering and Bias Detection	1	1	1	1	1	1	1	1	0	0	0.8
4852	Uso de bases de datos con Python	1	1	0	1	0.6	1	0	0	0	0	0.4
7891	Draw and Style Custom Letters with Inkscape	1	1	1	0	0.6	1	0	0	0	0	0.4
6367	FinTech Security and Regulation (RegTech)	1	1	1	1	0.8	1	0	0	1	1	0.7
2129	Fundamentals of Leadership, with Goldman Sachs 10,000 Women	1	1	0	0	0.6	0	0	1	1	0	0.5
6035	Songwriting: Writing the Lyrics	1	1	0	0	0.4	1	0	1	0	0	0.4
			Avg MAP (Top5)			0.7			Avg MAP (Top10)			0.56

As we can see from the table 4, the model continues to perform well in the top5, especially when the subject of the course is a popular one. However, due to the limitation of the number of entries in the data source, the recommendation results for minor language courses and cold direction courses do not perform well.

6 Conclusion

We presented a pilot version of the course recommender system that reminds students of their duties, warns them against difficult courses and recommends them potentially beneficial and preferred courses in Coursera. Therefore, the system helps students with their decisions, taylors their needs when they want to attain some new information for their continuous study.

More specifically, we designed our own particular algorithm suitable for the course recommendation. The algorithm incorporates different kinds of domain knowledge together, including content-based, similarity-based, knowledge-based, machine translation and deep learning to make a final decision of the recommendation. In the modelling part of this system, the most important step is data collection and alignment. There are two modules in the recommendation, one is user preference recommendation, the other is user preferred course based recommendation. Through calculating the similarity of skills and course detail between user's preference and existing data, plus other valuable information in the database, we are able to get a final result to recommend.

By obtaining the data and information of the courses, this system has addressed the difficulties to choose courses of their personal need. It has successfully extracted the domain knowledge and translated them into a course reasoning system. Our system can assist learners in crafting courses that meet their learning goals and match user preferences through a combination of knowledge-based, similarity-based and content-based reasoning techniques. In addition, we provided a user-friendly interface and page for learners to easily get the courses they want.

Certainly, there are still further improvements our team would like to include, such as implementing real user ratings of the courses to make a collaborative filtering reasoning recommendation and also, using Hadoop structure to make a real-time update of the data, performing cloud implementations. Nonetheless, this system is unarguably an impactful solution in the course recommendation as everyone could own their personal courses recommender which provides affordable, smarter and more intuitive recommendations.

7 References

[1] Ph.D. Melissa A. Venable. *2022 Online Education Trends Report*. https://www.bestcolleges.com/app/uploads/2022/03/2022-Online-Education-Trends-Report-Edited.pdf. Accessed 27, Oct.2022.

[2] Online. *8 recommended foreign online learning online course platforms*. https://haiwaisky.com/best-online-learning-platforms/. Accessed 27, Oct.2022.

[3] Online. *E-Learning Market. Global Market Insights*. https://www.gminsights.com/industry-analysis/elearning-market-size. Accessed 27, Oct.2022.

8 Appendix

8.1 Appendix A: Project Proposal

GRADUATE CERTIFICATE: Intelligent Reasoning Systems (IRS)

PRACTICE MODULE: Project Proposal

Date of proposal:
30 Oct 2022

Project Title:
Course recommendation system

Sponsor/Client: *(Name, Address, Telephone No, email)*

Institute of System Science (ISS) at 25 Heng Mui Keng Terrace, Singapore
NATIONAL UNIVERSITY OF SINGAPORE(NUS)
Contact: Mr. GU ZHAN / Lecture & Consultant
Telephone No.: 65-6516 8021
Email: zhan.gu@nus.edu.sg

Background/Aims/Objectives:

Background

With the rapid development of internet, more and more people tend to learn online to acquire knowledge and skills systematically. Many online recommendation websites are designed to provide people with massive open online courses (MOOCs). These course recommendation systems provide an affordable and flexible way to learn new skills, advance your career and deliver quality educational experiences at scale.

Aim

Course recommendation system in this project aims to use 6000+ available courses as the dataset. What' more, it will be designed with distinctive UI interface to help users compare similar courses so as to choose the most suitable course. The system aims to implement two functional modules: recommendation of courses according to users' preference and recommendation of courses by courses.

Objective

The system consists of two parts: front-end and back-end. Front-end will provide smooth UI experience in the form of web-page. Back-end will be responsible for storing and manipulating data. Users interact with website pages and then will be recommended 10 courses of her \ his interests.

Requirements Overview:

- Research ability
- Programming ability
- System integration ability
- Basic front end developing ability
- Excellent understanding of recommendation algorithm

Resource Requirements (please list Hardware, Software and any other resources)

Hardware proposed for consideration:

- CPU
- GPU

Software proposed for consideration:

- Python, javascript, css
- Python packages listed in the requirements.
- Web browser (Google Chrome / Microsoft Edge)

Methods and Standards:

Procedures	Objective	Key Activities
Market research and needs	• Define market needs and wants • Prioritizing key points to be designed • Aimed at targeted market section	1. Market research and identify market trend 2. Compare existing mature market product 3. Define system designed goal and scope
Technical workflow draft	• Define modules to be designed • Define detailed linking information about end-end • Define end in mind product • Set up platforms	1. Define, divide and link modules to be designed 2. Overview whole workflow and perfect technical details
Front-end development (web-page interface)	• Define functions of front-end • Conceive and draw up UI design of web-pages • Build websites	1. Plan for Acceptance Testing 2. Choose programming language and packages 3. Development 4. Evaluate acceptance testing
Back-end development (recommendation algorithm)	• Define best algorithm • Develop algorithm as per requirement	1. Plan for Acceptance Testing 2. Choose programming language and packages 3. Development 4. Evaluate acceptance testing
Front-end and Back-end integration	• Integrate front-end and back-end	1. Add front-end logic to back-end 2. Test features

Deployment	• Package application • Create user guide	1. Tidy up working environment in the project folder 2. Create step-step installation and user guide 3. Test user guide

8.2 Appendix B: Functionalities & Intelligent Reasoning System Mapping

The CCRS web application aims to utilize the topic introduced in the IRS (Intelligent Reasoning System). In this section every feature is mapped to topic studied under one of the category: Machine Reasoning (MR) , Reasoning System (RS) and Cognitive System (CS).

Section Number Section Title	IRS topics
3.1.1 Knowledge Acquisition	Machine Reasoning – Knowledge Discovery through data mining / Knowledge Elicitation using Knowledge Models / Knowledge Acquisition
3.1.2 Knowledge Representation	Machine Reasoning – Knowledge Representation
3.3.1 user-preference courses recommender	Machine Reasoning – Form of knowledge representation (rules) Reasoning System – Knowledge based recommendation, Cosine similarity, Word embedding
3.3.2 Subsequent content-based user courses recommender	Reasoning System – Matrix factorization for recommendation system, Content-based recommendation, Cosine similarity, Word embedding, Knowledge based recommendation
3.4 System Back-end	Machine Reasoning – Database architecture and implementation / Knowledge Discovery through data mining
3.5 System Front-end (User interface)	Cognitive System – Cognitive System Architecture Cognitive System – Knowledge Representation and Reasoning

8.3 Appendix C: User Manual Guide

Installation Guide

As the development and debugging of this project is based on the Windows operating system, we strongly recommend that you use Windows to run our system. All the following guides are based on Windows.

Step1: Download and install Anaconda

You can skip this step if you have downloaded and installed anaconda.
Anaconda is an open source distribution of Python and R that simplifies package management systems and deployment. You can download Anaconda from their website and install it by following official instruction.

Step2: Create a new conda environment

As a large number of third party Python packages will be used in this project, you should create a new conda environment to avoid conflicts between packages.
Type in conda prompt: conda create -n your_env_name python=3.8

Step3: Clone or download our system from Github

You can use git command to clone our project from Github.
Type in git prompt: git clone https://github.com/Theno-Chan/CourseraCourseRecommender.git

Or you can directly download system code from Github.

Step4: Install the Required Python Packages

You need to change directory to the Clone Repository folder or download folder.
Type in cmd: cd /to/the/right/folder
And run this command in cmd: pip install -r requirements.txt

Step5: Run web server

Run this command in cmd: python manage.py runserver
After running, you can open your web browser and type 127.0.0.1:8000 in the address bar. Then you will be directed to the index page!

User Manual

Sign up for a new account

Click on the ”signup” button on the home page and you will be redirected to the registration page. You should enter your username as well as your password in the corresponding fields on the new page and click on the ”Sign up” button. (Fig. 23)

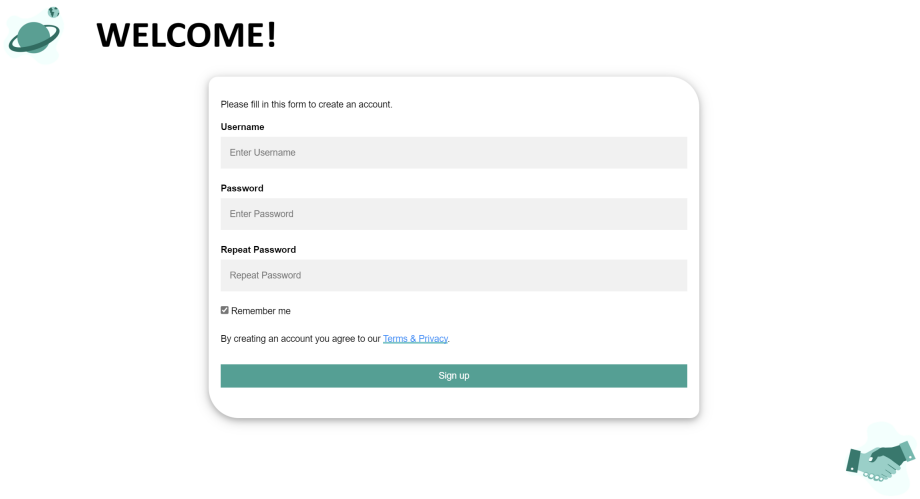

Figure 23: Sign up for a new account

Log in

You should enter your username as well as your password in the corresponding fields on the home page and click on the ”Log in” button. (Fig. 24)

Set up your preference

You will access this part in two ways: when you first register your account or when you click the ”Reset your preferences” button. You should enter at least 1 subject and up to five subjects. And other fields marked with an asterisk are required. (Fig. 25)

Personalised recommendations

Once you have finished logging in or completed your preferences, you will be able to access the personalised recommendation results. This page will display the top ten recommendations calculated by the model and you can view basic information about the course on this page. You can also click on the course link to access the course details. (Fig. 26)

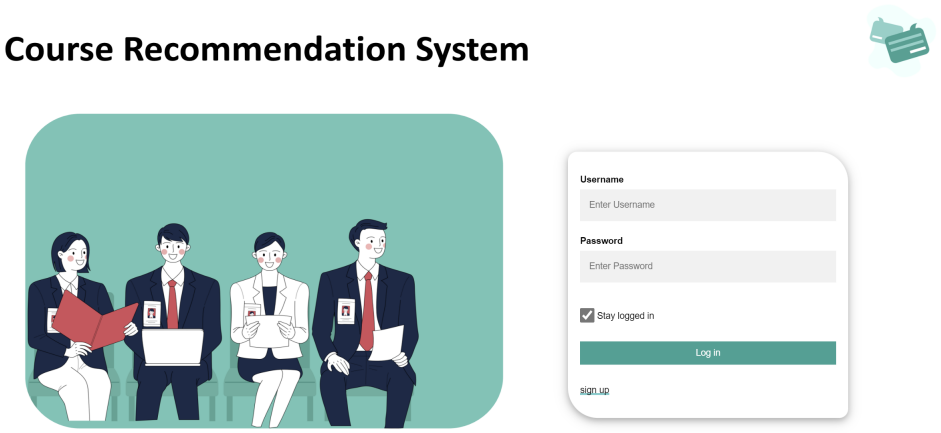

Figure 24: Log in

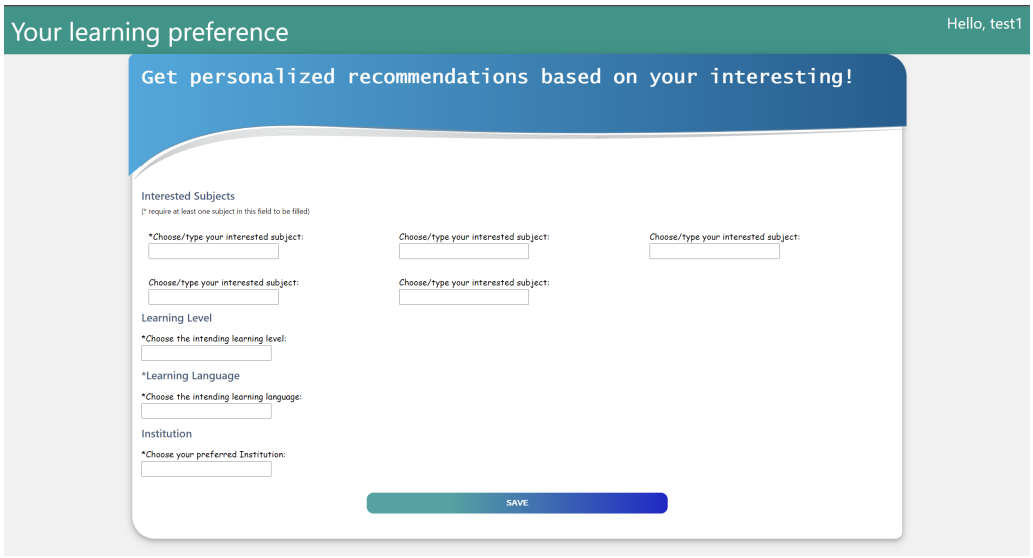

Figure 25: Set up your preference

Courses details and recommendations

This page will show you details of the course you clicked on, as well as ten similar courses recommended by the system. (Fig. 27)

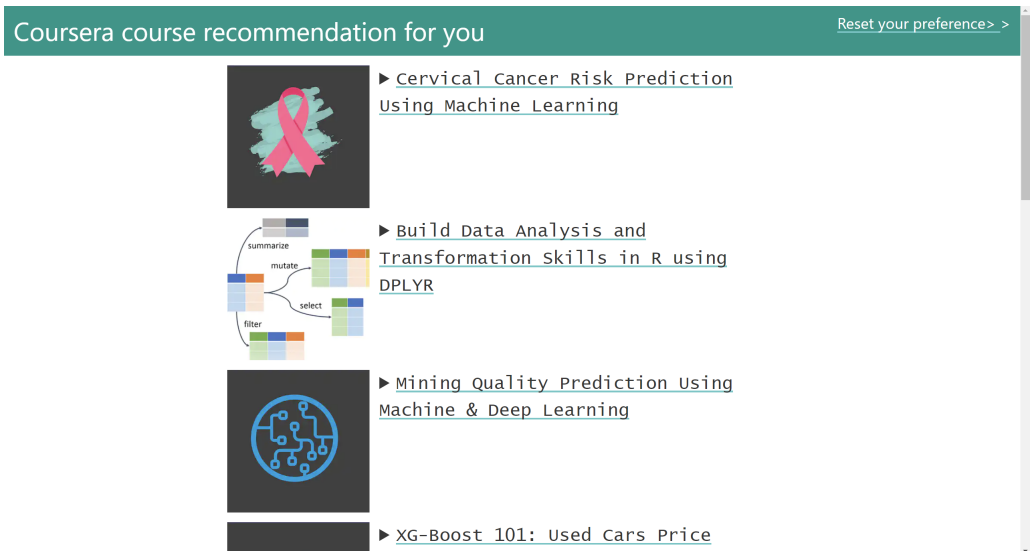

Figure 26: Personalised recommendations

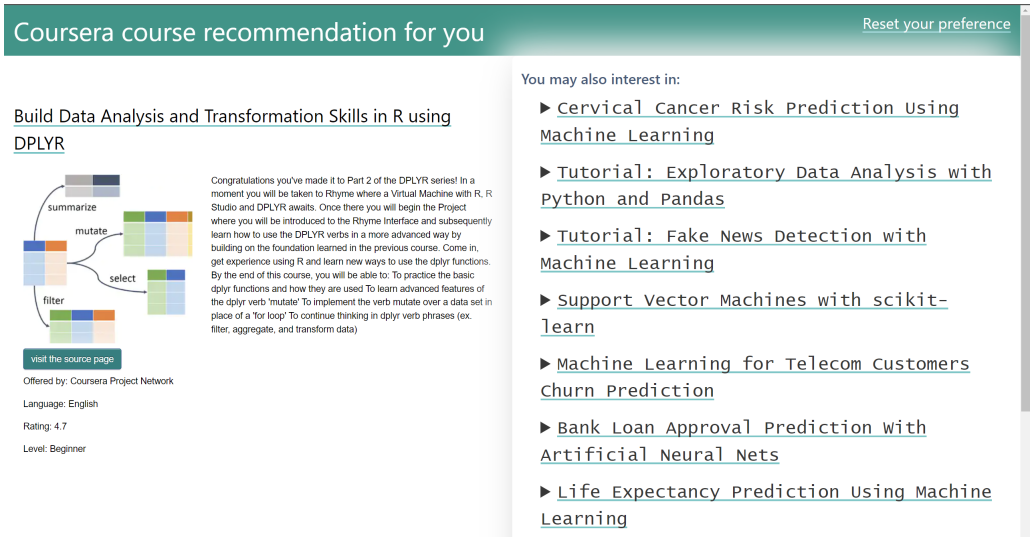

Figure 27: Courses details and recommendations

8.4 Appendix D: Individual Project report

Your Name:	Liu Haoran
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

Personal Contribution:
I participated in the following modules in our project:
1. System architecture design
As our project uses Django + HTML to provide web services, this can lead to a strong coupling between the front-end and back-end. So before development, my teammates and I discussed the architecture of the whole system, including the content to be displayed on the front-end pages, the user operation logic, the data flow between the front and back-end, the interface design between the model and the back-end, etc.
2. Back-end development
After defining the architecture and functionality of the system, I started developing the Django web service part of the back-end. First, I wrote view functions based on the system functionality to return different results depending on the HTTP request. Then I configured URL routing to bind the URLs to the view functions. At the same time, I worked on the database fields and the relationships between the various data tables.
3. System integration and test

I was responsible for putting the modules together after my teammates had completed the other parts. For the front-end part, I wrote ajax requests that can send data from the front-end pages to the back-end in an asynchronous manner. And I also incorporated Django's template language into the front-end to allow the pages to be displayed dynamically. For the model part, I imported the processed data into the database and deployed the model on the back-end.

Learning Outcome:
Through this project, I have gained:
1. Experience in several common recommendation algorithms (e.g. similarity-based, content-based, etc.);
2. Large improvement in Web programming ability, especially in Django Web Framework;
3. Knowledge about how to design and build intelligent reasoning system

Knowledge and Skill Application
In the future I would like to work as a software development engineer. I believe the knowledge I have learnt in this project about system architecture design, web framework building and database design will be of great help to me in my future work. In addition, as artificial intelligence technology is also playing a very important role in advancing industry, I think the knowledge about recommendation systems in the project will also help me in my future development.

Your Name:	Chen Zihao
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

Personal Contribution:
I participated in the following modules in our project:
1. Data collection and preprocessing:
For the Intelligent Reasoning Systems Practice module, I lead the development of the solution architecture via Knowledge Modelling and Acquisition to design the overall flow of our Coursera Course Recommender. The design of the solution architecture involved linking what data we could obtain to what our solution can feasibly do with our limited time and finally how the solution would be presented to our users. From the raw data we have which collected from Kaggle, I reviewed the relevant data columns, chose the useful information to be used in the dataset and also did the data preprocessing to make the data easy to analyze. And also, because of the lack of information in two respective datasets, I crawled the lacking course detail and course skills from Coursera website to make a more reliable recommendation using Selenium. These columns were important as they will serve as inputs to the recommendation algorithms that were to be developed at the later stage. From there, I developed the formal logic rules to guide the flow of our solution based on the selected data. For front-end part use and also for user convenience, I also crawled the course link and course image in order to make it more reasonable and user-friendly when we wanted to do a recommendation.
2. System models and algorithms design:
I was also tasked to design the recommender algorithm for the system. This was an interesting problem as we would have very limited and basic information on new users signing up for our app and yet we had to be able to provide them a usable recommendation. That was what we called a cold-start. To figure out this problem, we decided to make a page get the initial information of user preference for further use in the recommendation. By comparing user preference and existing courses data, I used sentence transformer to get the similarity score, added other calculation metric to evaluate the recommender score of each courses. After user logined in, based on the initial preference, the system could automatically recommend top 10 most similar courses to the user. But we still needed other recommendation algorithm to recommend based on courses. As we would not have much user data on the new customer, I realized that the content-based simliarity method would work for our project and what we learnt in class is directly applicable. Based on the learnings from the state of art research and what was taught in class, I decided to use Cosine similarity as the similarity metric also using sentence transformer.

Learning outcome
Through this project, I have learned:
1. Experience to use Selenium to crawl the useful information automatically online.
2. A deeper understanding of how data relates to real business problems and what it represents behind the scenes
3. Knowledge about how to design situable algorithms based on different situations
4. Knowledge about how to link the front and back ends of system and the model

Knowledge and Skill Application

I have learned different knowledge range from data collection, data preprocessing and data modeling which are all related to machine learning. In the future I would like to work as a machine learning engineer so from this project, I literally learnt a lot domain knowledge. I have a deeper understanding about recommender algorithms like how to use BERT to calculate similarity in this kind of problems. Also, As we used Gitub to manage our project, I feel much more comfortable with Git, it's such a handy project management tool. I believe what I have learnt in this project will contribute greatly to my future work.

Your Name:	Ku Maier
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

Personal Contribution
At different stages of this project, I was actively involved and worked hard to complete my part of the project and contribute to the group work.
In the pre-project stage, my group and I brainstormed together to analyse the topic requirements and determine the project theme and system architecture. After determining the project topic, I actively researched and collected many open-source datasets to share in the group meeting. After discussing with the group members, two of the datasets were used in the project as raw data.
After completing the pre-discussions, Jing Yu and I worked together to design the front-end user interaction part. After the UI design was roughly finalized, I was responsible for implementing the framework, functionality, and some of the style renderings for each page using HTML, CSS and JavaScript. After my team members had finished rendering the page styles, I completed the final system layout adjustments and corrections.
During the model selection process, I was involved in testing the performance of the doc2vec model and comparing it with the models tested by other team members to help finalize the model used in our system. The final model used was the SentenceTransformers model and its performance was filtered and improved as the project progressed, thanks to the efforts of the whole team.
Later in the project's progress, I was responsible for writing the Abstract, Project Implementation, and front-end-related sections, as well as producing business videos with Jingyu to enrich our project presentations.

Learning outcome
1. During this project, I practised my front-end skills.
2. With the help of my group members, I gained a deeper understanding of the design of the overall system architecture, as well as the integration of the front-end and back-end, and became familiar with the Django framework.
3. During our group discussions and project practice, I gained a deeper understanding of the advantages and disadvantages of different reasoning systems and their selection and application in different scenarios. Designing solutions to specific real-world problems not only exercised the skills I learnt but also made my knowledge practical and applicable, which is more valuable.

Knowledge and Skill Application
I learnt a lot of knowledge and skills from this project, all of which will help me get a quick start in the design and implementation of future industrial systems.

During the project, I learnt about the process of recommender systems and learned a lot about recommender reasoning systems, which will help me to get started quickly in designing and implementing recommender systems in the future and to design suitable solutions. I have also learnt how to select and apply different reasoning systems so that I can quickly and correctly select and apply reasoning systems and embed them into the solution system for other given real-world problems and project objectives in the future.

I became familiar with a range of models that can transform the text into vectors and calculate the similarity between texts, which can be applied in any future system that requires relevant functionality.

My front-end skills were practised. When I need to design a UI in the future, I can quickly get up to speed and achieve a clean and beautiful UI to improve the user experience. I also learnt the back-end implementation logic and how to integrate the database, front-end part and back-end part of the system, which are all core to the system design and implementation of the project.

Finally, I enjoyed my teamwork and I learned a lot of teamwork skills from it. Our whole team is very enthusiastic and engaged in the project. We had weekly group meetings to discuss the further development of our project. We worked together using overleaf and shared documents, and the results of the group meetings and everyone's tasks were recorded in the meeting notes and assigned. We use GitHub for version control, which

makes our team management and project progress very clear. This will also contribute to my future teamwork.

Your Name:	SHEN JINGYU
Certificate:	Graduate Certificate in Intelligent Reasoning Systems

Personal Contribution:
I participated in the following modules in our project:
1. Project research:
Together with my three team members, I contributed to brainstorming of possible project ideas. I was mainly responsible for market research. I read and studied the report 2022 Online Education Trends Report from U.S. news (in module of best colleges). From that, I learned about 2022-2023 e-learning trends and owned a basic E-learning market outlook. To more detailed information, I browsed through the website Global Market Insights to study market size like CAGR and so on. Then I compared six famous course recommendation systems about their pricing, feature and field. Considering the advantages and shortcomings of these recommendation systems, we defined our aim and objectives of our project.

2. Front-end design:
I was mainly responsible for the design of front-end with my teammate Ku Maier. We defined the basic functions of front-end. Then I started the design of UI about interactive interface. I studied the UI design of some well-known websites about their overall layout and color matching, and finally conceived the UI interface design of this system. I set the base color of the page to green, and then added the appropriate icon and illustration vector as the background image of the page. Regarding technical details about web-page, I designed, modified and adjusted the layout of the functional modules to match the UI interface based on Html, Css, Javascript.

Learning outcome
Through this project, I have learned:
1. Knowledge about market research:
I have learned a series of systematic knowledge about project topic selection, background investigation and architecture-designing about workflow chart. I know how to design a truly complete intelligent system using Machine Learning.
2. Knowledge about the design of interactive interface:
I learned that when we design a intelligent interactive system, we have to consider every detailed aspects according to users' experience. What's more, I understood the importance of team work from this project.
3. Knowledge about semantic transformation, ensemble learning, etc. that helps me to have a deeper understanding of intelligent reasoning.

Knowledge and Skill Application
In future work, I will use the knowledge learned from the project to conduct market research and topic selection. Throughout the duration of the project, I will maintain effective communication with users and team members to improve the system gradually. Also, in the future design of intelligent systems especially about NLP, I can apply the knowledge about semantic transformation, intelligent reasoning to build intelligent systems.