Recommendation System Development Based on Intelligent Search, NLP and Machine Learning Methods

Illia Balush, Victoria Vysotska and Solomiia Albota

Lviv Polytechnic National University, S. Bandera street, 12, Lviv, 79013, Ukraine

Abstract
Intelligent search is a search with the possibility of linguistic analysis, modern algorithms for parsing and finding words, recommendations based on user preferences. Such a search is a necessity for Internet resources in the field of e-commerce because it is almost impossible to find the right product or data without some help, filtering or sorting. One can highlight the following features of the search: spell check (if the search query is misspelt, the search for relevance will try to find close matches based on what was entered); recognition of standard abbreviations and acronyms (this search will be able to recognize standard abbreviations or acronyms, such as MK (Michael Kors), NASA (National Aeronautics and Space Administration)); better understanding (a search for relevance will try to understand the query better, applying well-known knowledge, recognizing frequently used synonyms and limited knowledge of the natural language); filters and sorting (the use of filters by specific categories and sorting will allow to more comfortably and quickly find the necessary information). The latter will come in handy for almost any type of online resource, from online stores to entertainment sites. Moreover, the paper highlights the topic of recommendation systems. Over the past few decades, with the growth of YouTube, Amazon, Netflix and many other such web services, referral systems have become more and more important in our lives. Recommendation systems are critical in some areas because they can generate huge revenue or stand out significantly from competitors. The paper considers the basic methods and approaches used to build search engines and recommendation systems. The implementation of these approaches on examples and natural systems has been considered. A web application based on Java and Elasticsearch has been developed with the performance of a recommendation system based on a collaborative filtering algorithm. The research object is an intelligent search information system with the possibility of recommendations in e-commerce. The subject of research is the basic principles and requirements for the construction of recommendation systems and intelligent search systems. The study aims to develop a fast and reliable search engine in e-commerce with the possibility of recommendations for users.

Keywords 1
Search engine, Elasticsearch, referral systems, collaborative filtering, web application, search, fuzzy search, inverted index, search query, intelligent search system, entry point, recommendation system, Levenstein distance, smart search, posting list, collaborative filtering algorithm, recommendation algorithm, shared filtering, stop word, search result, user rating, Levenstein algorithm, java application, recommender system, fuzzy search algorithm, category product category, user purchase data

1. Introduction

Today, a considerable amount of information is stored on the Internet. In addition, every day its number is only growing. It applies to online resources that contain entertainment content, such as Netflix, Megogo, and online stores, such as Amazon, Rozetka and others.

MoMLeT+DS 2021: 3rd International Workshop on Modern Machine Learning Technologies and Data Science, June 5, 2021, Lviv-Shatsk, Ukraine
EMAIL illiabalush@gmail.com (I. Balush); Victoria.A.Vysotska@lpnu.ua (V. Vysotska); Solomiia.M.Albota@lpnu.ua (S. Albota)
ORCID: 0000-0002-0498-4719 (I. Balush); 0000-0001-6417-3689 (V. Vysotska); 0000-0002-0845-5304 (S. Albota)

All these systems have one thing in common –for users to be satisfied with the work of these resources, they must be able to quickly, easily and efficiently search for the information they need. Also, one of the basics that help keep customers on the web is recommendations. 60% of customers prefer to return to stores with requests, and 75% of the digital generation, who grew up in the age of social networks, believe that recommendation systems are an integral part of any store or entertainment platform. Therefore, today, fast and efficient search for information with the possibility of recommendations is one of the primary needs of people and businesses. The chosen topic is highly relevant today, because today the demand for goods offered by the company, especially online, is growing at an unprecedented rate. It applies to video and audio products, goods, books and much more. It is especially true for Ukrainian buyers and users because, according to statistics, 25% of Ukrainians regularly buy goods online. 33% of Ukrainians buy a specific product online at least once a year.

This work aims to develop an intelligent search system for goods with the possibility of recommendations for the user. To do this, you need to determine how to store and analyse product information. What algorithms of proposals can be applied to this system? How to ensure convenient use of this system? The object of study of this work will be an information system of intelligent search, with the possibility of recommendations for users in e-commerce. The subject of this work will be the basic principles and requirements for intelligent search systems and recommendation systems. In this system, compared to existing systems, a collective algorithm of product recommendations for users was used, which focuses on the general coincidences of the choice of similar users. In addition, an algorithm for parsing and analysing words for the search was used [1], which provides an accurate search result, with the possibility of entering an error in the investigation. Also, the ability to auto-complete words when searching in this system has been added. Applying the principles laid down in this paper, you can build an effective, profitable and stable search engine for almost any type of business that contains many information or goods and is in demand among users.

2. Related works

General provisions for the construction of intelligent search systems. Let's start with the main points for building a search engine. Any intelligent search system consists of a repository, where all the necessary data is stored. A program that uses specific algorithms analyses the input feed, accesses the repository and returns the relevant result. The difficulty is to properly store the data and adequately configure the search engine processing algorithms. In addition, given that it should also be a recommendation system, it is necessary to choose the correct recommendation algorithm, which will have enough data for accurate calculations. The server part of the program has the most significant impact. We will try to build a standard modern search engine model. For simplicity, suppose we only accept text queries - a set of words. The backend returns the results that best match the input words. There is no logic of conjunction or negation among the input words. When a backend receives a search query string, it firstly divides it into specific parts or other tokens. For each token, the server part requests the repository. Documents are stored in the warehouse. After calling the storage, the backend already has information about which documents and how many search words are contained. They are then combined into a final list, sorted by relevance (most relevant at the top) and returned to the user. It is logical that for fast search in storage, storing data correctly [1, 2].

Search engine data warehouse. Let's say you want to determine which of The Washington Post's news articles contain the words "environment" and "health" from the beginning. One approach is to start over and read the entire text, writing down each article that contains the words mentioned. Usually, this technique is considered a direct review of the text. And this process takes a significant amount of time. One of the most popular ways to avoid such a linear scan for each request is to index the documents in advance, for example, based on inverted indexes [3-6]. This way, we will be able to properly store our data in the repository, which can be found quickly.

Search engine implementations. Every user tends to make mistakes when searching. Therefore, there is a problem: how to determine whether the term stored in the repository matches the user's search term? Levenstein's distance will help us here [7]. Levenstein's distance can also be called an editing distance, although it can also denote a more prominent family of distance metrics. It is closely related to the pairing of rows [8-10]. Levenstein's algorithm calculates the least number of editing operations required to modify one line to obtain another line [11, 12]. Thus, these approaches allow building a system capable of searching for relevant results in the repository. It remains to consider the main points related to the choice of the recommendation algorithm.

The referral system is a machine learning program that provides users with recommendations on what they might like based on their historic preferences. It can then be defined as a system that issues individual recommendations as source data or, as a result, has a personalized referral of the user to intriguing objects in a larger space of possible options. Examples:

- Offer readers news articles based on the interests of the reader.
- Offer customers suggestions on what they can buy based on their history of purchases or searches.

There are three types of recommended systems:

Collaborative filtering. Shared recommendation systems summarize object ratings or recommendations, recognize commonalities between users based on their ratings, and generate new recommendations based on comparisons between users. They work well for complex objects, where taste variations are responsible for most preferences changes. Shared filtering is based on the assumption that people who have agreed in the past will decide in the future and that they will be like similar objects as they were before.

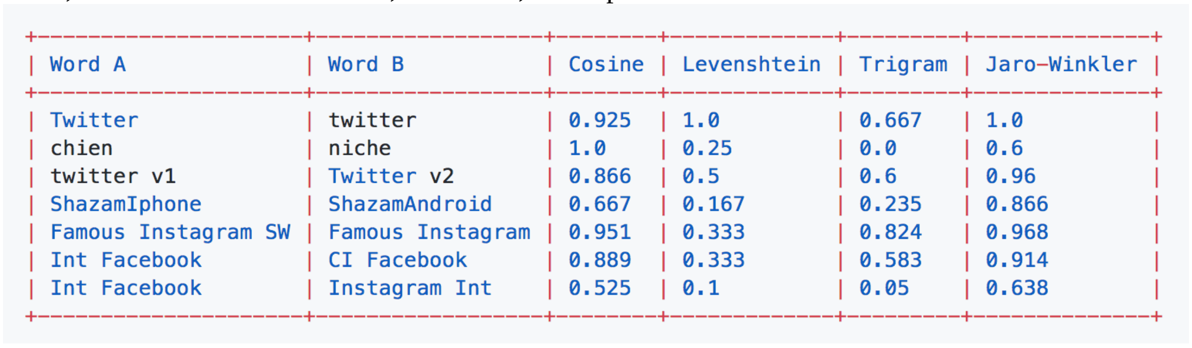
Content-based. It is recommended based on content. It examines the profile of the new user's interests based on the available functions in the objects that the user has evaluated. It is a system of recommendations for specific keywords. Here the keywords are used to describe the goods. Thus, the content-based recommendation system uses algorithms that recommend similar items that the user liked before or are currently testing.

Hybrid. Combining both systems of recommendations in a way that corresponds to a particular industry is known as a system of hybrid recommendations. Netflix is an excellent example of a hybrid approach. They provide recommendations by comparing the viewing and searching habits of similar users (shared filtering), as well as offering movies that share common characteristics with film that the user has rated highly (content-based filtering) [13-15]. Let's compare basic recommendation algorithms.

Table 1
Comparative table of recommendation algorithms

Name	Benefits	Disadvantages	Application
Collaborative filtering	User ratings and ratings are taken into account	Low performance for first users	Information portals
	Not tied to the subject of the service	A lot of information about user ratings is needed	Small online stores
Content-based filtering	Works instantly, even for first users	Tied to the content of the service	Blogs
	Works correctly, even with a small amount of data	Not based on the wishes of users	Music or movie platforms
Hybrid filtration	High productivity	Hard to maintain	Large online stores
	No problems with previous approaches	Complex development process	Complex systems with a large number of users

Considering the table above, we can conclude that the implementation of our system will be a practical algorithm for collaborative filtering because the advantages of this method are ideal for the search engine, and the disadvantages are not significant not to choose this method. Also, let's compare fuzzy search algorithms. The numbers that can be seen in Fig. 1 is the accuracy of the words between the first and second columns, from 0 to 1. Comparing all the results, we can conclude that Levenstein's algorithm is ideal for our system because, in this case, the accuracy of search results will be the highest. Levenstein's algorithm compares words, not phrases, taking into account the length of the change of words, or the number of deletions, insertions, and replacement of letters in the word.



```
+----------------------+------------------+--------+-------------+---------+--------------+
| Word A               | Word B           | Cosine | Levenshtein | Trigram | Jaro-Winkler |
+----------------------+------------------+--------+-------------+---------+--------------+
| Twitter              | twitter          | 0.925  | 1.0         | 0.667   | 1.0          |
| chien                | niche            | 1.0    | 0.25        | 0.0     | 0.6          |
| twitter v1           | Twitter v2       | 0.866  | 0.5         | 0.6     | 0.96         |
| ShazamIphone         | ShazamAndroid    | 0.667  | 0.167       | 0.235   | 0.866        |
| Famous Instagram SW  | Famous Instagram | 0.951  | 0.333       | 0.824   | 0.968        |
| Int Facebook         | CI Facebook      | 0.889  | 0.333       | 0.583   | 0.914        |
| Int Facebook         | Instagram Int    | 0.525  | 0.1         | 0.05    | 0.638        |
+----------------------+------------------+--------+-------------+---------+--------------+
```

Figure 1: Comparison of the accuracy of calculations of fuzzy search methods

3. Material and methods

The primary purpose of this system's goal tree is to enable the user to conveniently and quickly find the desired product. It includes several aspects, such as the ability to auto-complete search words, errors in terms when searching, and so on. In addition, the system must generate recommendations for the user, taking into account the purchases of similar users (Fig. 2).

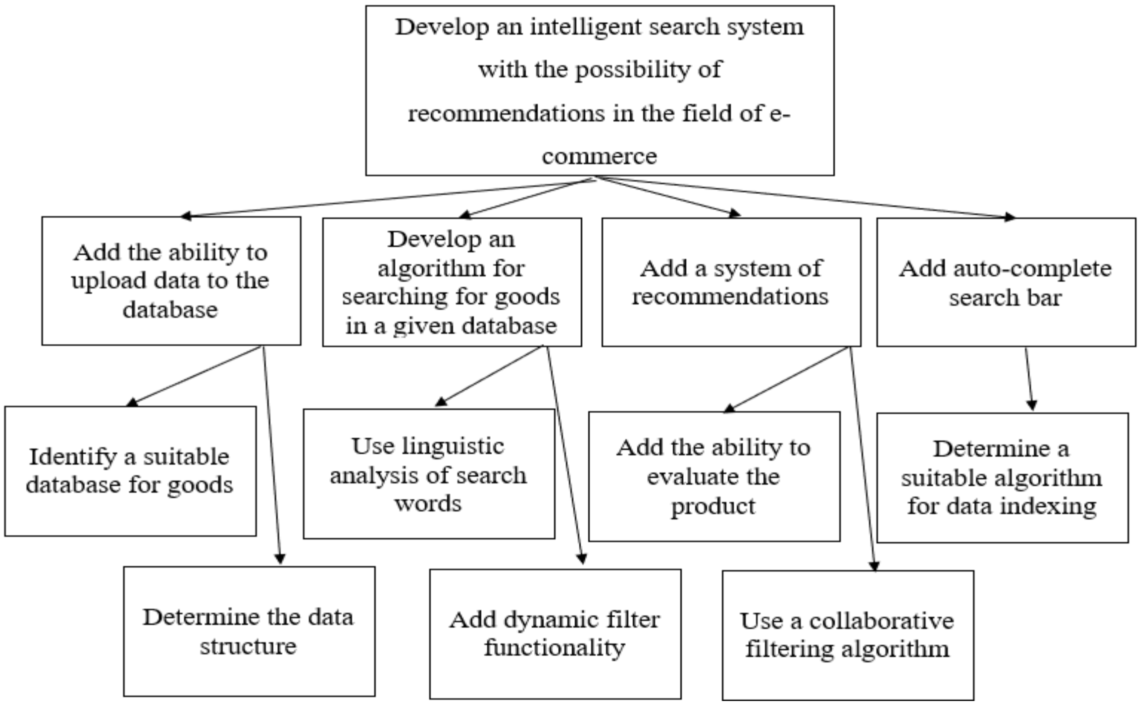

Develop an intelligent search system with the possibility of recommendations in the field of e-commerce
Add the ability to upload data to the database
Develop an algorithm for searching for goods in a given database
Add a system of recommendations
Add auto-complete search bar
Identify a suitable database for goods
Use linguistic analysis of search words
Add the ability to evaluate the product
Determine a suitable algorithm for data indexing
Determine the data structure
Add dynamic filter functionality
Use a collaborative filtering algorithm

Figure 2: Goal Tree

Concretization of system functioning. IDEF0 considers the logical relationship between works, not their temporal sequence (workflow) [16, 17]. The IDEF0 standard was developed in 1981 in the United States by the Air Force Department for Industrial Automation. In software development, developers were faced with the need to create new methods of business process analysis. As a result, the IDEF0 functional modelling methodology appeared, in which special IDEF0 notations are used for analysis. The most critical function is located in the upper left corner. And processes are connected among themselves using arrows and descriptions of functional blocks. In this case, each type of arrow or activity has its meaning. This model allows describing all the main types of processes, both administrative and organizational [18].

Since the primary purpose of our system is to provide search and recommendations for the user, the primary mechanism will be to requests a search for the user (Fig. 3). The main input parameters of this process will be the user ID and the actual search query. The mechanisms that ensure the proper operation of this process will be a database or repository, software libraries. The control mechanisms will be technical documentation, as well as collected user requirements. The result of this process will be relevant, as requested, and recommendations for the user.

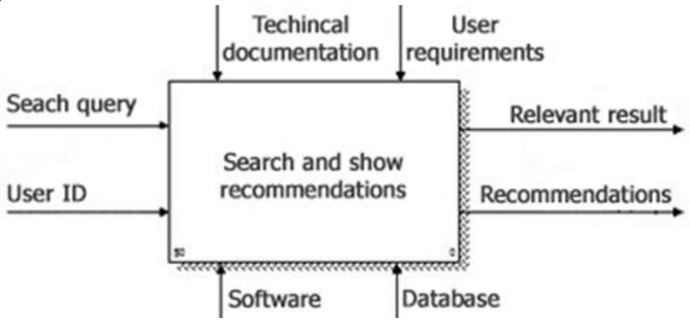


Figure 3: IDEF0 first level

Let's try to decompose the model described above. First of all, let's highlight a few basic, smaller blocks (Fig. 4).

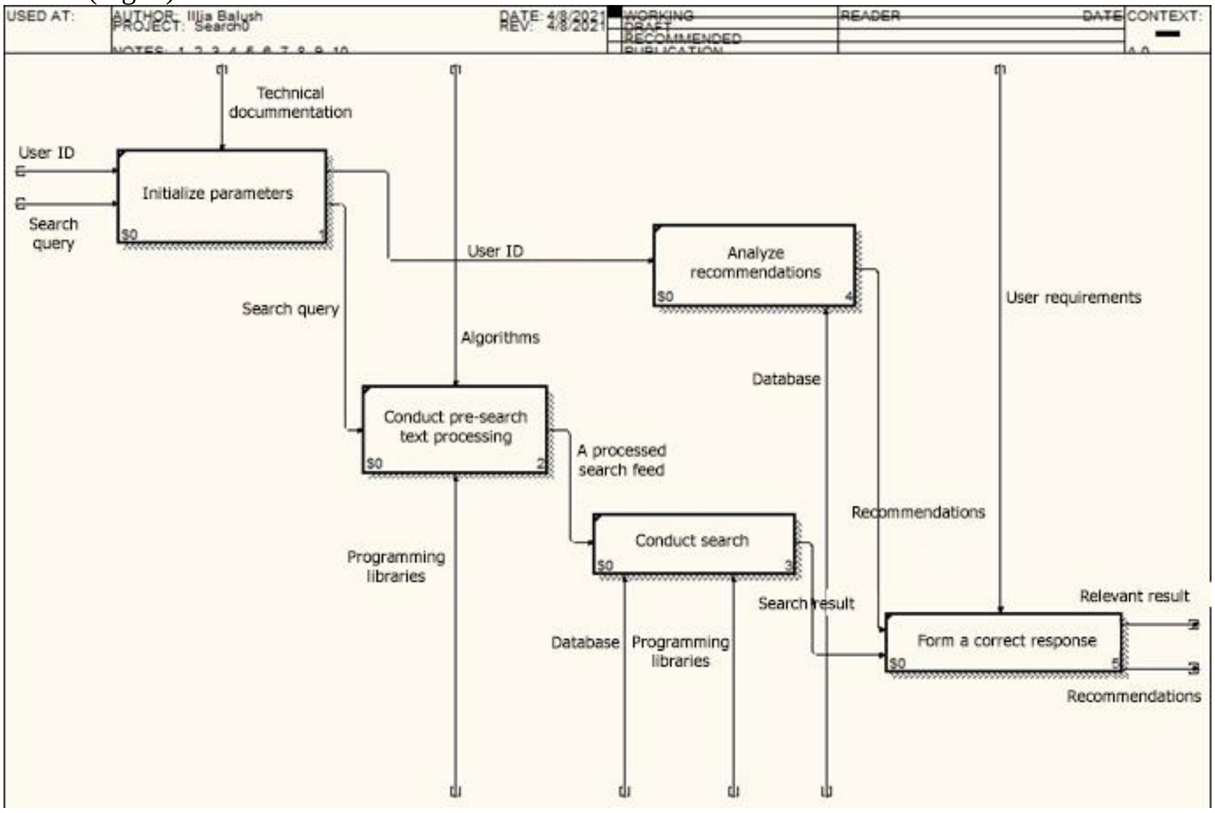


Figure 4: IDEF0 second level

At the initial address to the program, it is necessary to initialize input parameters. Therefore, at the entrance to the first process, there are the user ID and search query. After initialization, we can proceed to the following methods. For a successful search, you must first analyse the input text in terms of the program. Therefore, for the process "Pre-search text processing" (Fig. 4), the input parameter will be the search bar. Using linguistic analysis, we will be able to process the tape to the desired form. It will help us with some ready-made software modules that we can connect. Then, using the processed tape, we can find the necessary documents in the repository. Now, let's move on to another process, "Analyse the recommendations" (Fig. 4). Only the user ID is logged in to this process. Since user purchase data is stored in the repository, the mechanism that will help this process is the database. Using the collaborative filtering algorithm and knowing the user ID, we will analyse and issue recommendations for this user. After that, the recommendations and the search results move on to the following, final process of "Generate the correct answer". Since the definitive answer can be influenced by specific user requirements and the understanding that you need to show only a certain amount of information to the user, we need to form an answer that will contain only the best recommendations and the best result.

Let's try to decompose the process of pre-search text processing (Fig. 5). To begin with, to select some logical unit in the text, we will divide the search query into tokens. The token can be considered as one word. Next, all our operations and processes will use and modify the newly formed tokens. For a single letter format, convert the tokens into a single letter case (Fig. 5). Stop words are words in any language that do not attach much importance to a sentence. They can be safely ignored without sacrificing the meaning of the sentence. Therefore, the third process will remove all tokens that are stop words. To complete this process, you need to provide a dictionary of stop words.

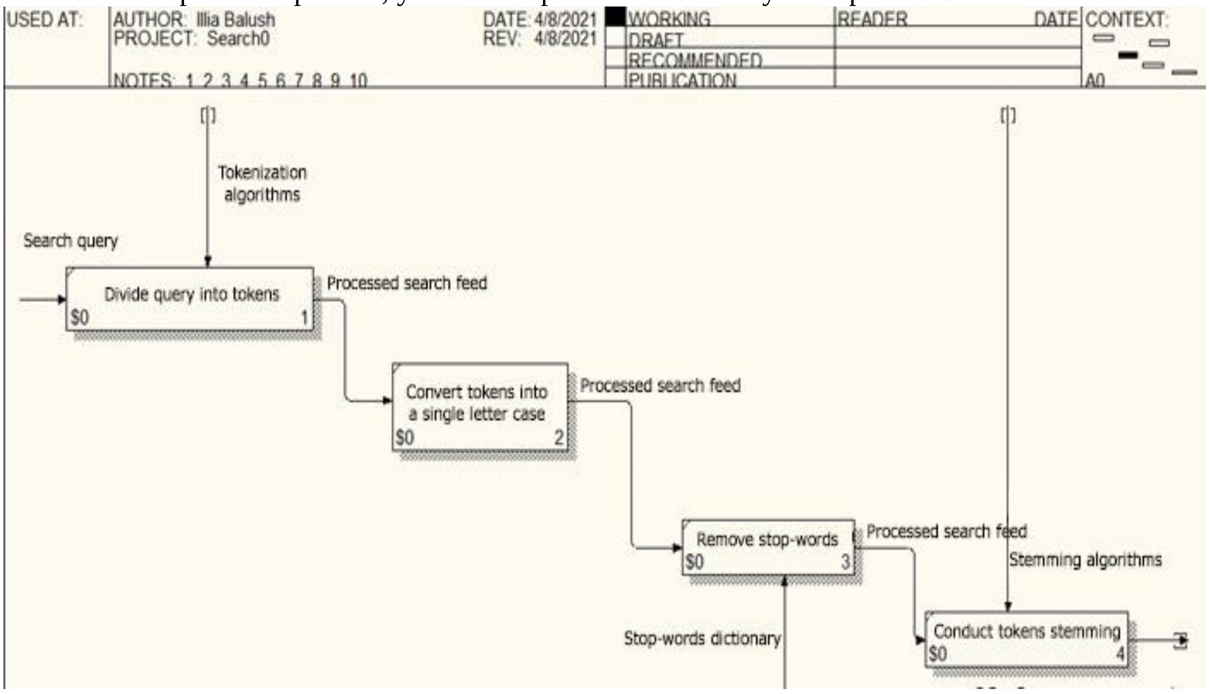


Figure 5: IDEF0 of the third level

After that, the filtered tokens go to the final process - stemming tokens. The stemming algorithm is a process of linguistic normalization in which variant word forms are reduced to a general form. That is, the central, root part of the word remains without suffixes and endings. After this process, our tokens, or processed text, is ready for the following search process.

Also, expand the process of recommendations analysis (Fig. 6). In the first process, we need to extract user purchase data. A repository or database will help us with this. Having data on purchases of buyers, we pass to the following process. Here we need to apply a collaborative filtering algorithm that can provide predictions about products that the user has not yet purchased, based on selecting products from similar users. The data that will be at the end of this process must be analysed, and the best recommendations are chosen. To choose the best offers, you need to be guided by user requirements. After this process, we can pass requests to the following procedure.

The basis for making a process hierarchy is the DFD hierarchy. DFD is a notation designed to model information systems in the storage, processing and transmission of data [19]. In our system, the central external entity is the user (Fig. 7). In the data stream, it can pass its ID as well as the request. The primary process or system is the Search Engine. At the output, the system gives the relevant result and recommendations to the user. Let's move on to the DFD chart of the second level. In part, it is similar to IDEF0, so that some processes can be omitted. In this diagram, we have 3 data warehouses, namely: a store of information about customer purchases, a dictionary of stop words needed for linguistic analysis, and a store of information about the company's products. Decomposing the process of pre-search text processing, we obtain the following DFD diagram (Fig. 9). At once, it is possible to allocate that at this level to remove stop words. We use a data warehouse - the dictionary of stop words. In general, the pre-search process consists of 4 main processes that are interconnected, namely: the division of the search bar into tokens or words, modification of tokens into a single format, filtering redundant words, as well as stemming words or highlighting only the root part of the word. The same can be done with the process of analysing the recommendations. There is information about customers' purchases as a repository of data, including the user who makes the request. Next, we can assume that a particular product will be appropriate for our user using the recommendation algorithms. After that, we will be able to form recommendations and pass them on to the following process.

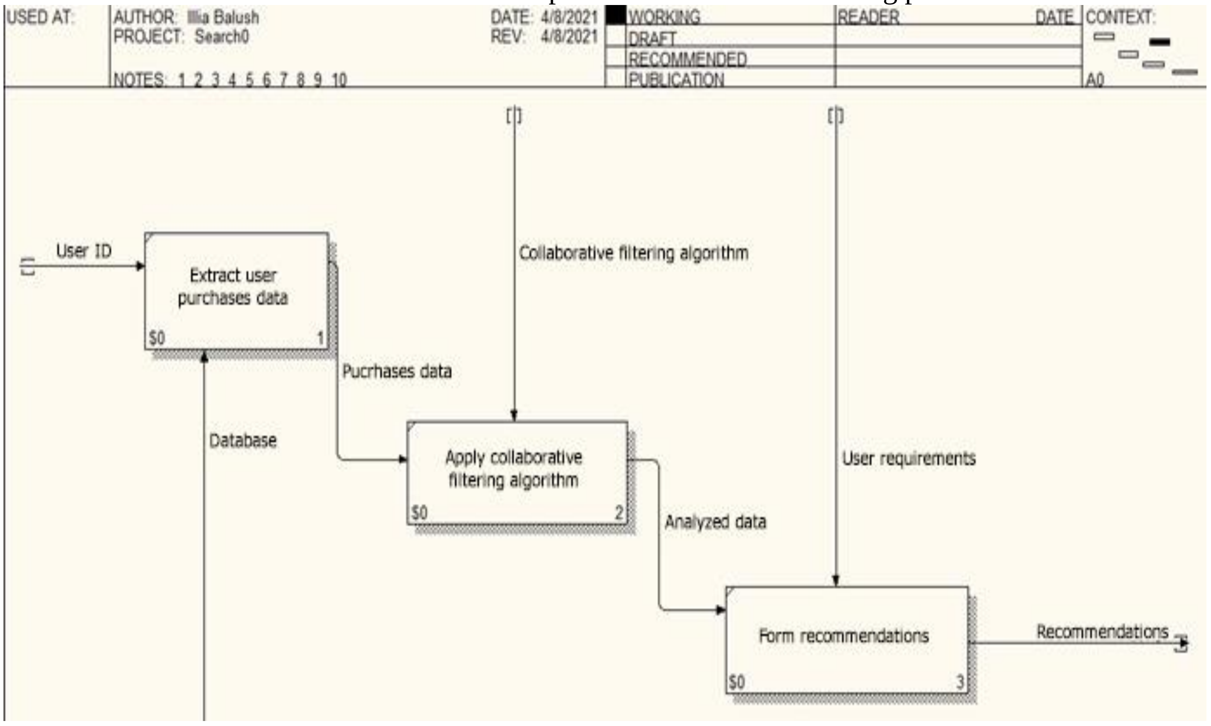


Figure 6: IDEF0 of the third level

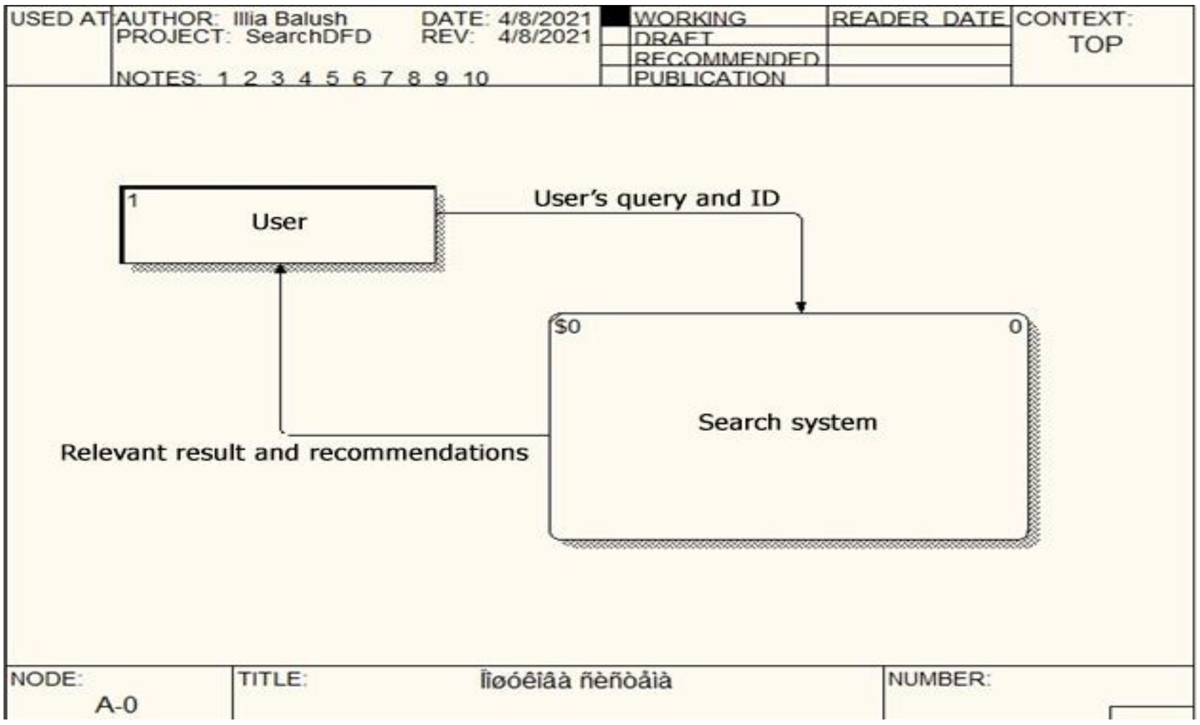


Figure 7: DFD first level

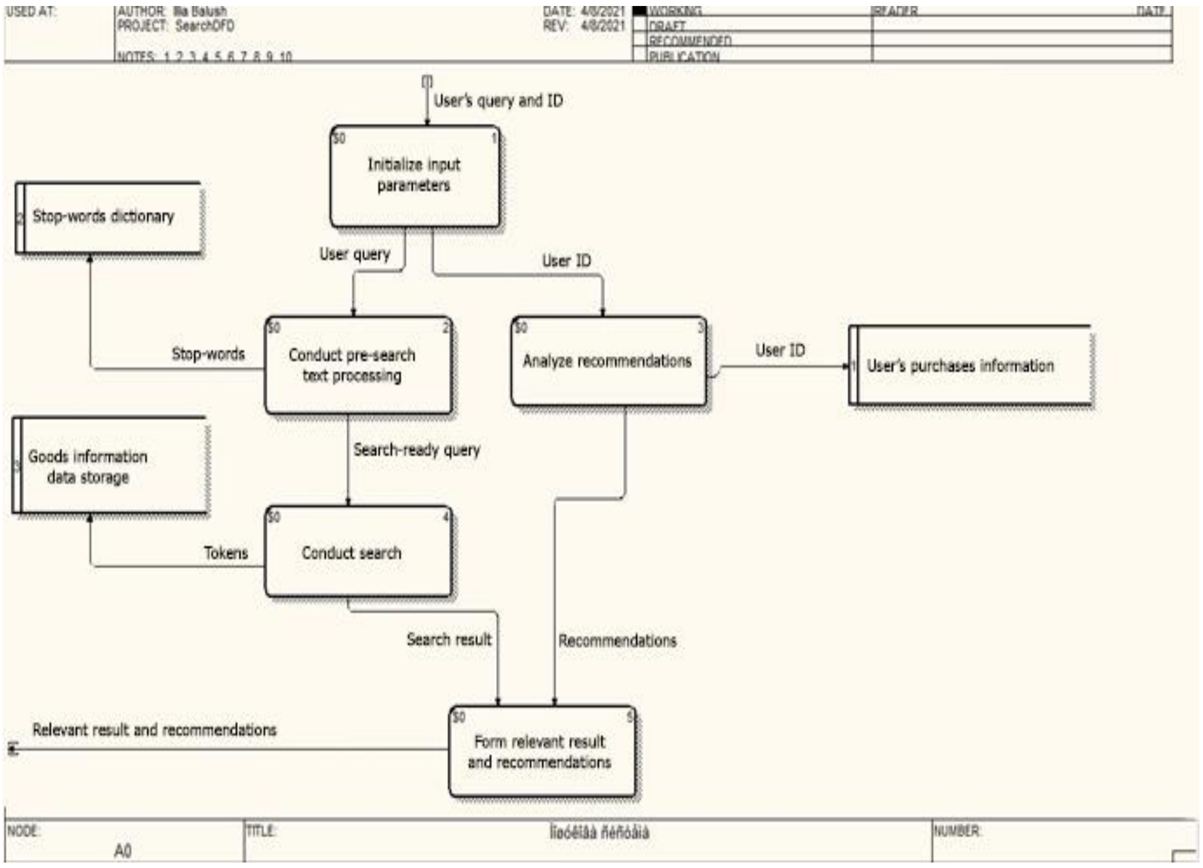


Figure 8: DFD of the second level

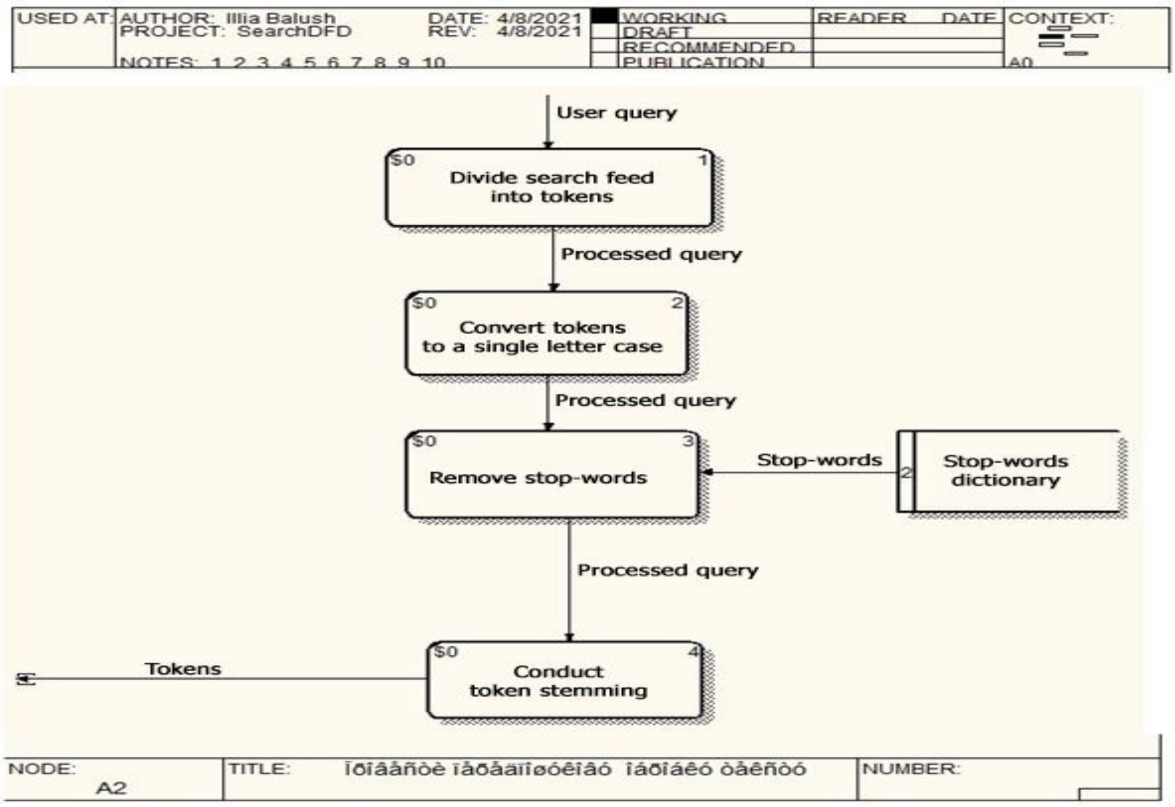


Figure 9: DFD of the third level

4. Software solving problems

Selection and justification of means of solving the problem.

- *Elasticsearch*. First of all, let's start with the data warehouse, which will store information about all products. Elasticsearch was chosen as this repository. Elasticsearch is a high-scale open-source full-text search and analytics engine [20].
- *MySQL* was chosen to store user purchase data as well as product evaluations. MySQL is a relational database management system based on SQL - a structured query language and used for a wide range of purposes, including storage, e-commerce and registration applications [21].
- *Java*. Let's move on to the software implementation of our system. Java was chosen as the primary programming language [22-23]. Why was Java chosen?
    a. Price – free.
    b. Productivity - due to the HotSpot JIT compiler, the code is executed very quickly.
    c. Efficiency is a wide range of libraries that contain optimized code that is easy to write.
    d. Portability - programs written in Java can run on almost any device.

Language Support – the developer company, continues to improve language capabilities by correcting past bugs, improving performance, and simplifying the complexity of writing code.

Algorithms. Let's move on to the selection of algorithms. The main algorithm for the recommendations was the Slope One algorithm. The Slope One algorithm is a shared filtering system based on element evaluation [24]. It means that it is based entirely on the ranking of user positions. When we calculate the similarity between objects, we only know the history of ratings, not the content itself. This similarity is then used to predict the rating of potential users for user-element pairs that are not in the dataset. Initially, users evaluate various elements in the system. Next, the algorithm calculates the similarity. After that, the system predicts the rating of goods that the user has not yet rated. Co-filtration (CF) is a technique used by recommended systems [25].

Technical characteristics of selected software development tools.

Java version 8 was chosen for this system. This version is considered the company's most significant release and remains a stable version for many companies worldwide. You can highlight the following advantages of this version:

- Better type output. The Java 8 compiler has significantly improved inference type. In many cases, explicit type parameters can be output by the compiler, supporting cleaner code.
- Lambda and Functional Interfaces. Lambda (also known as secure or anonymous methods) is the most significant and most anticipated language change in the entire Java 8 release programming. Many languages on the JVM platform had lambda from day one, but Java developers had no choice but to submit lambda through anonymous classes.
- Default interfaces and static methods. In Java 8, the definition of interfaces has been expanded with two new concepts: the default method and the static method. They allow adding new methods to existing interfaces without compromising backward compatibility for previously written versions of these interfaces. The difference between default and abstract methods is that abstract methods must be implemented, and there are no default methods. Instead, each interface must provide a so-called default implementation, and all successors will receive it by default (with the ability to override this default implementation if necessary).
- Repeated annotations. Since annotation support was introduced in Java 5, this feature has become very popular and widely used. However, one of the limitations of using annotations was the fact that the same instruction could not be announced more than once in one place. Java 8 violates this rule and contains repetitive annotations. It allows the same annotations to be repeated several times in the place where they are announced. Duplicate annotations should annotate themselves using the @Repeatable annotation [26, 27].

Maven. Maven is an assembly management tool. It determines how your .java files are compiled into .class, packaged into .jar files (or .war or .ear), (pre / after) processed by tools, managing your CLASSPATH and all other types of tasks needed to build your project. It's similar to Apache Ant or Gradle or Makefiles in C / C ++. Still, it tries to be utterly standalone because one doesn't need additional tools or scripts, including other everyday tasks such as downloading and installing the necessary libraries. It is also designed with the idea of portability building so that you don't get problems like having the same code with the same build script on one computer but not on another (this is a known issue, we have Windows 98 virtual machines because we could not build some of our Delphi programs anywhere). Because of this, it's also the best way to work on a project between people using different IDEs because the Ant IDs created are difficult to import into other IDEs, but all IDEs today understand and support Maven (IntelliJ, Eclipse, and NetBeans). Even if you don't like Maven, it will eventually become a starting point for all other modern build tools. Why was Maven used? Maven will download all the libraries you use and the libraries they use for you automatically. It avoids the "hell of dependence" of libraries. It uses the "Configuration Convention", so you do not need to specify the tasks you want to perform by default. You don't need to write step "compile", "test", "package", or "clean" as you would in Ant or Makefile. Just put the files where Maven is waiting for them. Maven also has many excellent plug-ins that you can install that will handle many routine tasks: from creating Java classes from an XSD schema using JAXB to measuring the amount of code covered by tests using Cobertura. Just add them to pom.xml, and they will integrate with everything one wants to do.

Liquibase. Liquibase is an open-source schema change management solution that makes it easy to manage changes to your database. Liquibase uses changeSets to represent a single change in your database [28]. ChangeSet is what you use to group databases and is a unit of transition that Liquibase makes in a database. A changeLog is a list of changes created by multiple sets of changes. Three elements define the set of changes: "ID" and "Author" and the path to the ChangeLog file name.

Liquibase allows to perform the following tasks:

- Rollback Support: one can use the liquibase.bat rollback to undo an update, as it allows for rolling back changeSets based on the number of changeSets, to a specified date, or to a specified tag stored in the database.
- Automatic update: instead of performing updates or rollbacks directly against the database, one can create SQL that will be run for validation and/or execution manually.
- Exiting a future rollback: before applying the update to the database, one can generate the SQL that one will need to run to return the database to its current state for validation.
- ChangeSet Contexts: changeSet can be assigned the "contexts" in which they will run. Contexts are selected at runtime and can be used to modify sets that work only in test instances or other unique circumstances.
- Prerequisites of ChangeLog and ChangeSet: conditions can be added to changeLog or individual changeSets to check the status of the database before attempting to execute them.
- ChangeSet checksums: when a changeSet is executed, Liquibase retains the checksum and may not run or change the execution if it detects a change between the original changeSet definition when it was started and the current report.
- Difference support: although Liquibase is designed to use database comparisons to manage change, Liquibase has support for this, which is helpful in many cases, such as validation between databases.

Spring Framework provides a comprehensive model of development and configuration for modern business applications in Java - on any platform. A key element of Spring is program-level infrastructure support: the focus is on the "plumbing" of business applications, so developers can focus on business logic without unnecessary tweaks depending on the runtime environment [29].

5. Experiments, Results and Discussions

Description of the created software. *General Information*. Program name: "Search Engine". The program can be run on almost any platform: Linux, Windows, macOS, etc. Basically, the program is written in Java using the Spring framework.

Two databases are used:

- MySQL – to save data on purchases and recommendations.
- Elasticsearch – to save search documents and index data.

To simplify the management of database schemas, the Liquibase framework is used, which allows changing the database structure quickly. Maven technology is used to manage the dependency between Java libraries.

Functionalities. The program's primary functions are intelligent search for documents uploaded to the repository and the possibility of recommendations for users [30-36]. In addition, to fill the warehouse with data, a module for loading and indexing data into the storage was created. To simulate the purchase of goods, a unique entry point was built in the program.

Description of the logical structure. The program consists of three main modules:

- Data loader.
- Engine API.
- Engine-Server.

Fig. 10 shows the primary interaction between the modules.

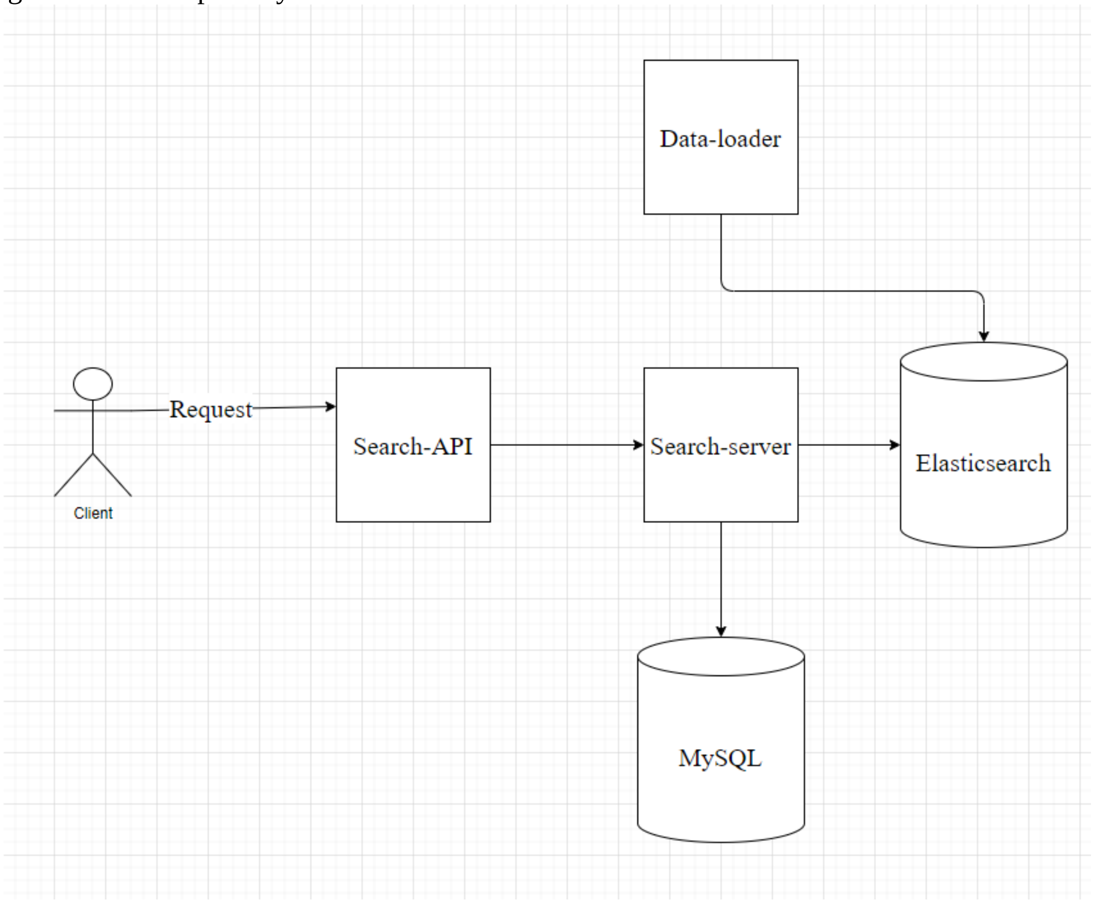
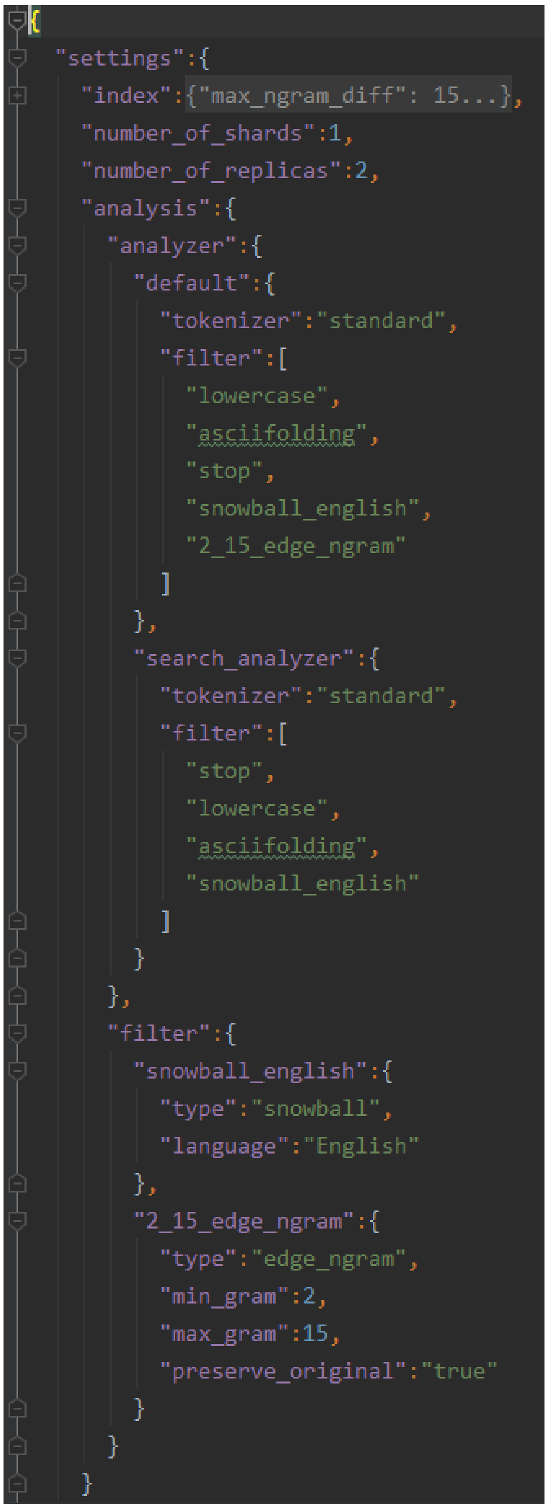


Figure 10: Interaction between software modules

Data-loader is a module that is required to load data into the repository. It contains the structure of the document. The document can consist of different fields. Therefore, it is necessary to create a general design considering all documents with their types. An example of such a structure shows in Fig. 11. The Search-API is an input module that accepts all incoming requests. It specifies what type each input request has, along with all input parameters. Search-server is the main module, where all the basic logic of the program is executed. To make it easier to control the processing of each request, a concept such as a Pipeline was created. The Pipeline is a specific sequence of program code execution. It is a high-level code concept so that it can be used universally for each type of request. The Pipeline consists of stages or Stage-s. Each step, or phase, processes the input parameters and passes the processed parameters further to the next step. For the convenience of describing these Pipeline and their stages, a unique structured data format was created. An example of such a Pipeline shows in Fig. 12.

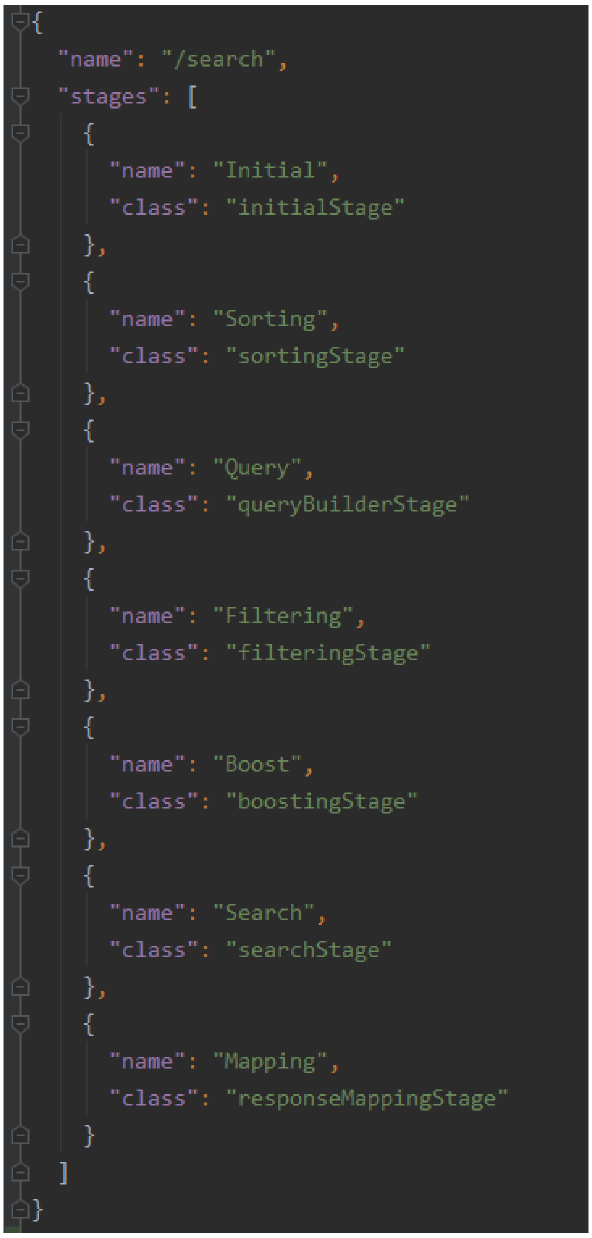
In the beginning, in the field "name", the name of this Pipeline is set. The following are the main stages of this Pipeline. The project contains three main Pipeline:

- / search - describes the process of performing the intelligent search.
- /search/recommendation - describes the process of executing the algorithm of recommendations.
- /search/suggestion - describes the process of generating auto-search words.



```
{
  "settings":{
    "index":{"max_ngram_diff": 15...},
    "number_of_shards":1,
    "number_of_replicas":2,
    "analysis":{
      "analyzer":{
        "default":{
          "tokenizer":"standard",
          "filter":[
            "lowercase",
            "asciifolding",
            "stop",
            "snowball_english",
            "2_15_edge_ngram"
          ]
        },
        "search_analyzer":{
          "tokenizer":"standard",
          "filter":[
            "stop",
            "lowercase",
            "asciifolding",
            "snowball_english"
          ]
        }
      },
      "filter":{
        "snowball_english":{
          "type":"snowball",
          "language":"English"
        },
        "2_15_edge_ngram":{
          "type":"edge_ngram",
          "min_gram":2,
          "max_gram":15,
          "preserve_original":"true"
        }
      }
    }
```

Figure 11: The structure of documents in the repository

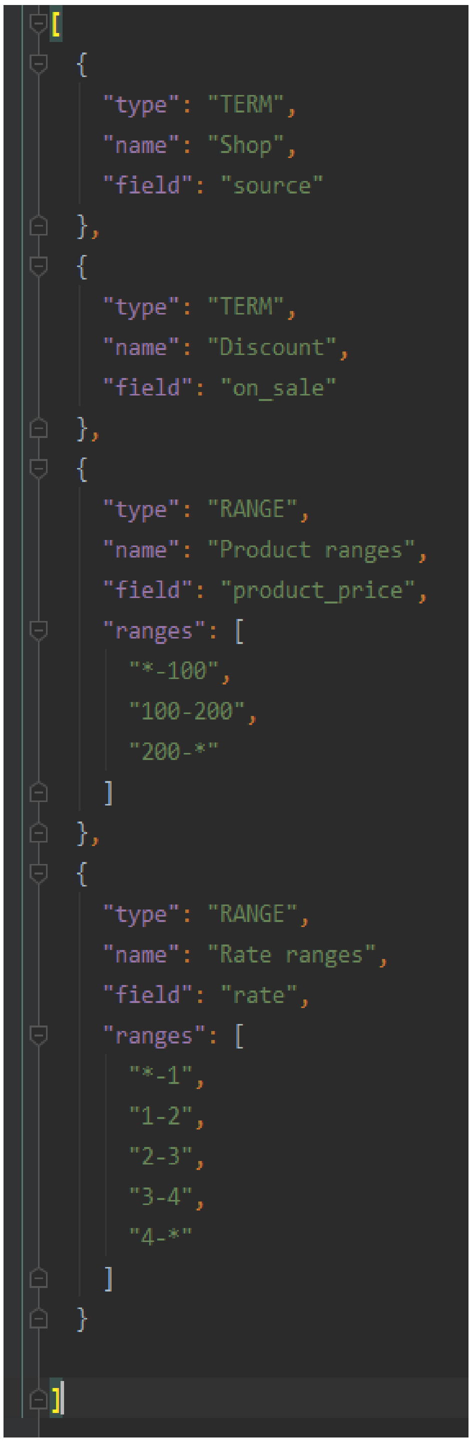


```
{
  "name": "/search",
  "stages": [
    {
      "name": "Initial",
      "class": "initialStage"
    },
    {
      "name": "Sorting",
      "class": "sortingStage"
    },
    {
      "name": "Query",
      "class": "queryBuilderStage"
    },
    {
      "name": "Filtering",
      "class": "filteringStage"
    },
    {
      "name": "Boost",
      "class": "boostingStage"
    },
    {
      "name": "Search",
      "class": "searchStage"
    },
    {
      "name": "Mapping",
      "class": "responseMappingStage"
    }
  ]
}
```

Figure 12: Example Pipeline

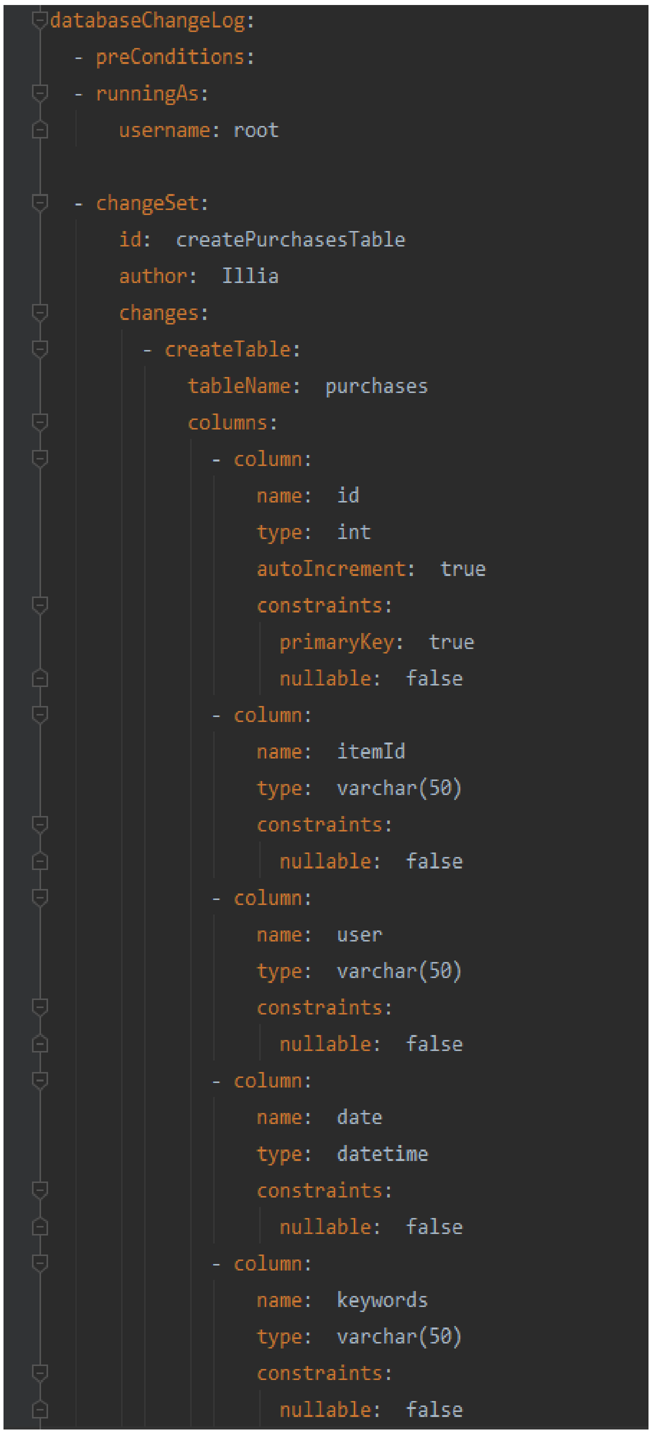
In addition, filters are used to ease the search. Filters are based on the concept of aggregation. For the convenient use of aggregations, a particular document with a strictly defined structure was created, which describes the main aggregations and their capabilities. Fig. 13 shows an example of such a document. This example shows four filters:

- Shop - a filter that allows filtering products by the manufacturer.
- Discount - a filter that filters goods at a deal.
- Price - a filter that filters goods at the selected price.
- Rate - a filter that filters products by rating.



```
[
  {
    "type": "TERM",
    "name": "Shop",
    "field": "source"
  },
  {
    "type": "TERM",
    "name": "Discount",
    "field": "on_sale"
  },
  {
    "type": "RANGE",
    "name": "Product ranges",
    "field": "product_price",
    "ranges": [
      "*-100",
      "100-200",
      "200-*"
    ]
  },
  {
    "type": "RANGE",
    "name": "Rate ranges",
    "field": "rate",
    "ranges": [
      "*-1",
      "1-2",
      "2-3",
      "3-4",
      "4-*"
    ]
  }

]
```

Figure 13: File with defined aggregations



```
databaseChangeLog:
  - preConditions:
  - runningAs:
      username: root

  - changeSet:
      id:  createPurchasesTable
      author:  Illia
      changes:
        - createTable:
            tableName:  purchases
            columns:
              - column:
                  name:  id
                  type:  int
                  autoIncrement:  true
                  constraints:
                    primaryKey:  true
                    nullable:  false
              - column:
                  name:  itemId
                  type:  varchar(50)
                  constraints:
                    nullable:  false
              - column:
                  name:  user
                  type:  varchar(50)
                  constraints:
                    nullable:  false
              - column:
                  name:  date
                  type:  datetime
                  constraints:
                    nullable:  false
              - column:
                  name:  keywords
                  type:  varchar(50)
                  constraints:
                    nullable:  false
```

Figure 14: Example file with table structure

To sort the results found by specific criteria, the concept of Sort is used. For convenient and fast use of new sorting options, a document was created to describe the desired sorting by specifying the required field. Fig. 13 shows an example of such a file. This example shows two sorting options:

- Default - which sorts the "_score" field in descending order.
- Cheap is which sorts the "product_price" field in ascending order.

Liquibase technology is used to determine the structure of the database, as was mentioned earlier. To use this technology, one needs to create an appropriate file called "liquibase-changeslog.yml". An example of such a file shows in Fig. 14. It defines the main tables in the database, as well as the fields and their type. The following are the main tables (Table 2-4) used in the program.

Table 2
Description of tables in the database

Table	Fields	Description
purchases	id - int itemId - varchar (50) user - varchar (50) date - datetime keywords - varchar (50)	Contains basic data on purchases of goods
rates	id - int itemId - varchar (50) keywords - varchar (50) user - varchar (50) date - datetime rate – int	Contains information about the evaluation of goods
recommendations	id - int itemId - varchar (50) user - varchar (50)	Contains products that should be recommended to users

Call and download the program. The source code is contained on the GitHub service. After downloading the source code, from the root point of the code, one needs to call the command: mvn clean install. After that, .jar files of each file will be generated. To run the program, go to the root directory of engine-api and run the command java -jar engine-api.jar.

Incoming data. The program contains 6 main entry points after starting the server.

Table 3
Description of entry points

Entry point	Parameters	Description
/ search	query - search phrase category - product category count - the desired number of returned results sort - sort order	The main entry point for finding products.
/ search / suggestion	query - part of a word or phrase count - the desired number of returned results	An entry point that offers auto-completion of a word or phrase if the query contains unfinished words
/ search / recommendation	userId - user ID	Provides a list of recommendations for the user with the specified ID
/ search / inner / recommendation	userId - user ID	Launches the collaborative filtering algorithm
/ search / shop	itemId - product identifier userId - user ID category - product category	Records the purchase of goods with the selected ID for the specified user

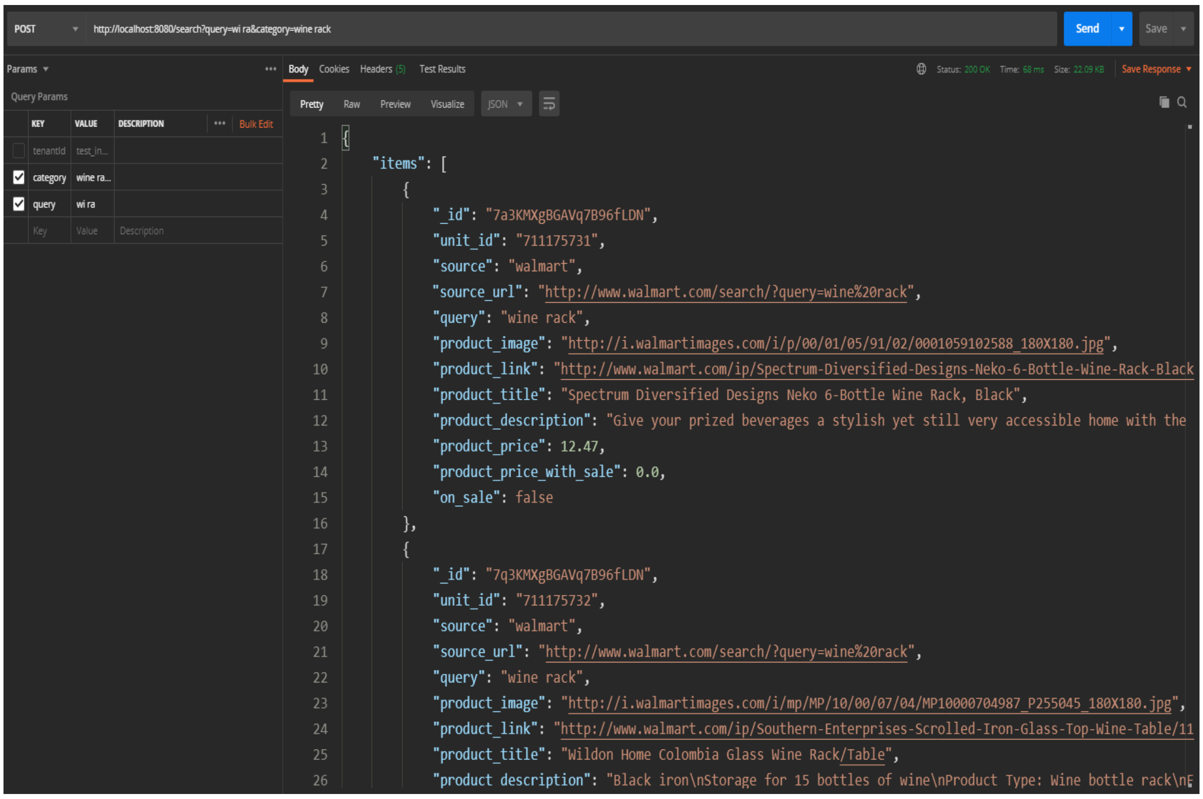
/ search / rate	itemId - product identifier userId - user ID category - product category rate - product evaluation	Entry point for evaluating the product with the selected ID for the specified user

Output data. In table 6 the source data and their format for each entry point are described.

Table 4
Description of the source data

Entry point	Format
/ search	items - array of goods with the following structure: id - product identifier source - manufacturer of goods sourceUrl - manufacturer link query - product keywords productImage - photo of the goods productTitle - product title productDescription - description of the goods productPrice - the price of the product productPriceWithSale - discounted product price onSale - shows whether the product contains a discount
/ search / suggestion	suggestion - auto-completed word
/ search / recommendation	topItems - an array of goods that are in the top 10 for sale recommendations - an array of products recommended by the system for the specified user bucketModels - an array of available filters, with the following structure: name - selection option in the filter count - the number of products with such a filter
/ search / inner / recommendation	-
/ search / shop	-
/ search / rate	-

User manual. Full program name: "Search-engine". Abbreviated name of the program: "SrchNgn". Primary programming language: Java. Development environment: IntelliJ IDEA Community edition. The program's primary purpose is to provide the user with the ability to search in e-commerce and recommendations intelligently [37-44]. The program can be used for online stores for small and medium businesses. This system can find the relevant product according to the search query. In addition, if the search bar is incomplete or contains errors, the system can find the most appropriate product. Fig. 15 shows an example of a search in a system where the search bar includes two words, two of which are incomplete. The system can publish the top 10 products purchased by other users and recommend products that the user may like based on the preferences and ratings of similar users. Fig. 16 shows an example of user recommendations, along with the top 10 products and search filters.



POST http://localhost:8080/search?query=wi ra&category=wine rack

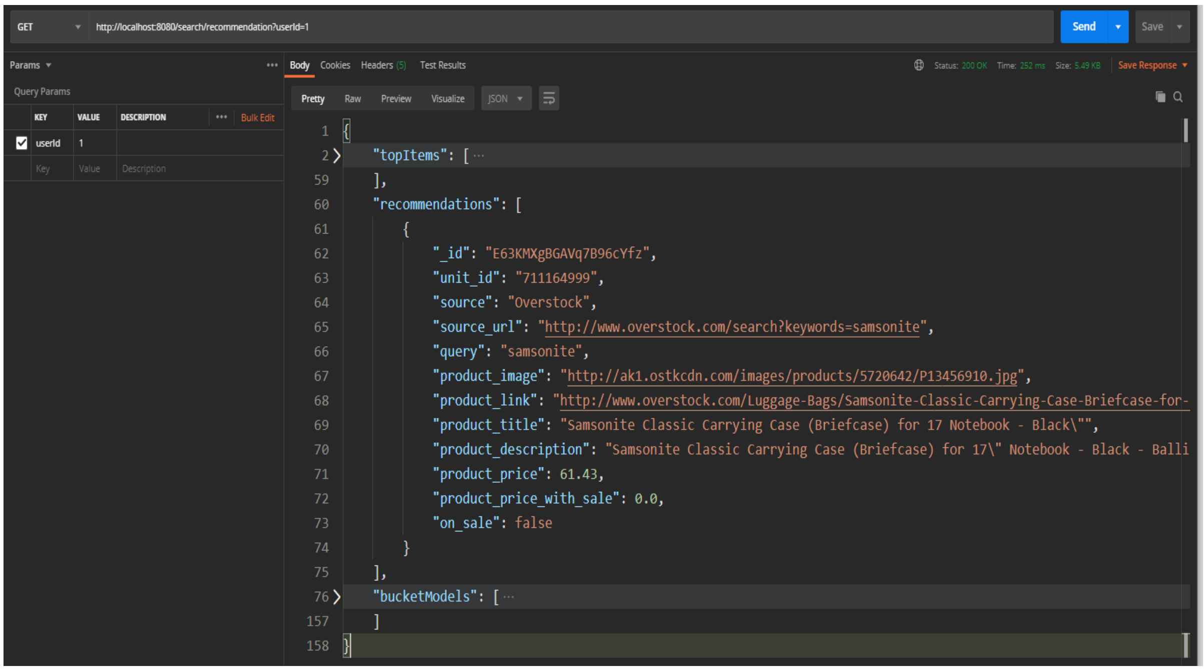
Params · Query Params

	KEY	VALUE	DESCRIPTION
☐	tenantId	test_in...	
✓	category	wine ra...	
✓	query	wi ra	
	Key	Value	Description

Body · Status: 200 OK Time: 68 ms Size: 22.09 KB

```
{
    "items": [
        {
            "_id": "7a3KMXgBGAVq7B96fLDN",
            "unit_id": "711175731",
            "source": "walmart",
            "source_url": "http://www.walmart.com/search/?query=wine%20rack",
            "query": "wine rack",
            "product_image": "http://i.walmartimages.com/i/p/00/01/05/91/02/0001059102588_180X180.jpg",
            "product_link": "http://www.walmart.com/ip/Spectrum-Diversified-Designs-Neko-6-Bottle-Wine-Rack-Black
            "product_title": "Spectrum Diversified Designs Neko 6-Bottle Wine Rack, Black",
            "product_description": "Give your prized beverages a stylish yet still very accessible home with the
            "product_price": 12.47,
            "product_price_with_sale": 0.0,
            "on_sale": false
        },
        {
            "_id": "7q3KMXgBGAVq7B96fLDN",
            "unit_id": "711175732",
            "source": "walmart",
            "source_url": "http://www.walmart.com/search/?query=wine%20rack",
            "query": "wine rack",
            "product_image": "http://i.walmartimages.com/i/mp/MP/10/00/07/04/MP10000704987_P255045_180X180.jpg",
            "product_link": "http://www.walmart.com/ip/Southern-Enterprises-Scrolled-Iron-Glass-Top-Wine-Table/11
            "product_title": "Wildon Home Colombia Glass Wine Rack/Table",
            "product description": "Black iron\nStorage for 15 bottles of wine\nProduct Type: Wine bottle rack\nE
```

Figure 15: Search functionality



GET http://localhost:8080/search/recommendation?userId=1

Params · Query Params

	KEY	VALUE	DESCRIPTION
✓	userId	1	
	Key	Value	Description

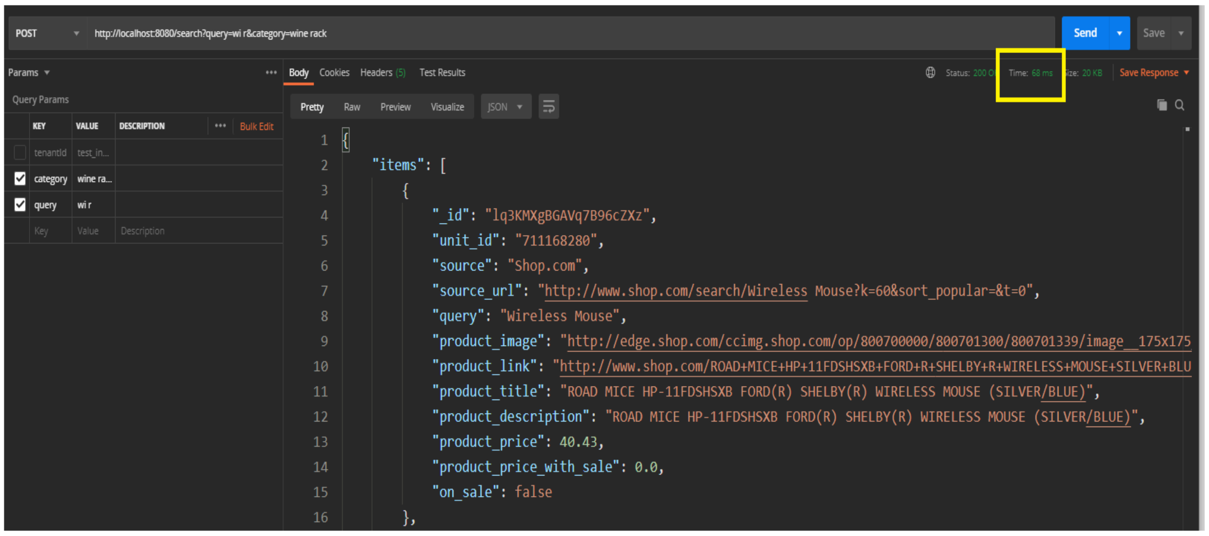
Body · Status: 200 OK Time: 252 ms Size: 5.49 KB

```
{
    "topItems": [ ···
    ],
    "recommendations": [
        {
            "_id": "E63KMXgBGAVq7B96cYfz",
            "unit_id": "711164999",
            "source": "Overstock",
            "source_url": "http://www.overstock.com/search?keywords=samsonite",
            "query": "samsonite",
            "product_image": "http://ak1.ostkcdn.com/images/products/5720642/P13456910.jpg",
            "product_link": "http://www.overstock.com/Luggage-Bags/Samsonite-Classic-Carrying-Case-Briefcase-for-
            "product_title": "Samsonite Classic Carrying Case (Briefcase) for 17 Notebook - Black\"",
            "product_description": "Samsonite Classic Carrying Case (Briefcase) for 17\" Notebook - Black - Balli
            "product_price": 61.43,
            "product_price_with_sale": 0.0,
            "on_sale": false
        }
    ],
    "bucketModels": [ ···
    ]
}
```

Figure 16: Recommendations

Levenstein's algorithm was used to implement fuzzy search [45-55], which counts the number of letter substitutions in a word and calculates the required word length based on this. For the possibility of product recommendations, a collaborative filtering algorithm was used, which does not depend on the specific topic of the site. NLP algorithms were used to index the data [56-60]. For example, edge-gram, which divides one word by the selected number of parts.

The average execution time of a search query is 10-250ms. The initial request can take up to 350ms. It is since the system has not yet optimized the critical path of the program. After optimization, execution time is reduced to 100ms. Fig. 17 shows the execution time of the request:



```
http://localhost:8080/search?query=wi r&category=wine rack
```

Query Params

	KEY	VALUE	DESCRIPTION
	tenantId	test_in...	
✓	category	wine ra...	
✓	query	wi r	
	Key	Value	Description

```
{
    "items": [
        {
            "_id": "1q3KMXgBGAVq7B96cZXz",
            "unit_id": "711168280",
            "source": "Shop.com",
            "source_url": "http://www.shop.com/search/Wireless Mouse?k=60&sort_popular=&t=0",
            "query": "Wireless Mouse",
            "product_image": "http://edge.shop.com/ccimg.shop.com/op/800700000/800701300/800701339/image__175x175
            "product_link": "http://www.shop.com/ROAD+MICE+HP+11FDSHSXB+FORD+R+SHELBY+R+WIRELESS+MOUSE+SILVER+BLU
            "product_title": "ROAD MICE HP-11FDSHSXB FORD(R) SHELBY(R) WIRELESS MOUSE (SILVER/BLUE)",
            "product_description": "ROAD MICE HP-11FDSHSXB FORD(R) SHELBY(R) WIRELESS MOUSE (SILVER/BLUE)",
            "product_price": 40.43,
            "product_price_with_sale": 0.0,
            "on_sale": false
        },
```

Figure 17: Query execution time

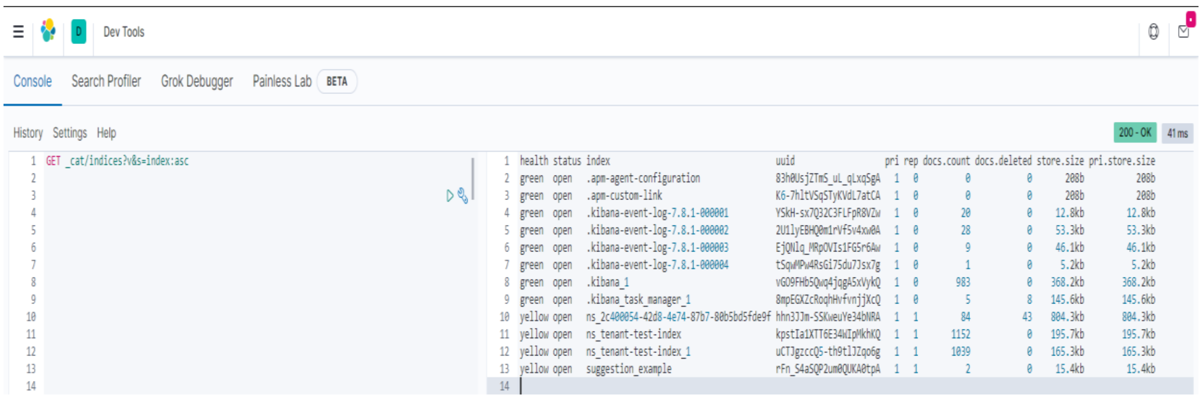
The program can run smoothly. To update the data, one must index the data on the new index and then restart the program by specifying a new index with the data.

To run the program, you must meet the following conditions:

- Install JDK 8 and prescribe the necessary paths to the JDK in Windows environment variables
- Install Maven and define the required paths to Maven in Windows environment variables
- Install Elasticsearch 7.12
- Install MySQL 15.1

After downloading the source code from the Git repository, you must run the command mvn clean install in the root directory of the downloaded code. This command will generate .jar files for each of the modules. After that, it is necessary to load the data. For this purpose, it is required to pass to the folder with the data-loader module and further to the target folder. There will be a generated .jar file. To run it, run the java -jar data-loader.jar command. After downloading the data, you need to create the necessary database structure in MySQL. To do this, go to the project's root directory and run the command mvn liquibase: update **-Denv = dev**. After creating the database schema, you need to go to the folder with the search-API module and then to the target folder. Then run the java-jar command search-api.jar. After that, we will perform search queries at http: // localhost: 8080 / search? query =. The MySQL and Elasticsearch databases must be running and have the necessary tables and indexes for the program to work correctly.

Analysis of the control example. In the beginning, check the data in the repository. To do this, we will use a tool such as Kibana, which usually comes with Elasticsearch. Fig. 18 shows all existing indexes in the warehouse. We are interested in the e-commerce index. As one can see in the figure below, we do not have such an index.

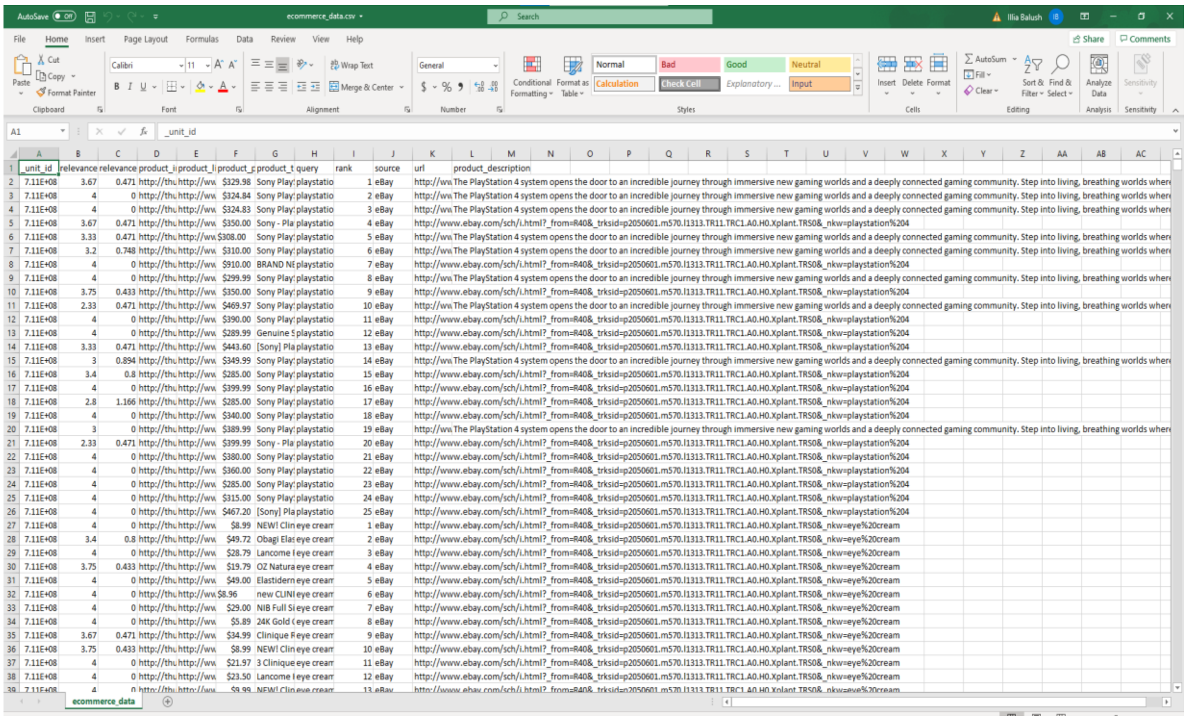


```
GET _cat/indices?v&s=index:asc
```

```
health status index                                    uuid                   pri rep docs.count docs.deleted store.size pri.store.size
green  open   .apm-agent-configuration                 83h0UsjZTmS_uL_qLxqSgA   1   0          0            0       208b           208b
green  open   .apm-custom-link                         K6-7hltVSqSTyKVdL7atCA   1   0          0            0       208b           208b
green  open   .kibana-event-log-7.8.1-000001           YSkH-sx7Q32C3FLFpR8VZw   1   0         20            0     12.8kb         12.8kb
green  open   .kibana-event-log-7.8.1-000002           2U1lyEBHQ0m1rVf5v4xw0A   1   0         28            0     53.3kb         53.3kb
green  open   .kibana-event-log-7.8.1-000003           EjQNlq_MRpOVIs1FG5r6Aw   1   0          9            0     46.1kb         46.1kb
green  open   .kibana-event-log-7.8.1-000004           tSqwMPw4RsGi75du7Jsx7g   1   0          1            0      5.2kb          5.2kb
green  open   .kibana_1                                vGO9FHb5Qwq4jqgA5xVykQ   1   0        983            0    368.2kb        368.2kb
green  open   .kibana_task_manager_1                   8mpEGXZcRoqhHvfvnjjXcQ   1   0          5            8    145.6kb        145.6kb
yellow open   ns_2c400054-42d8-4e74-87b7-80b5bd5fde9f hhn3JJm-SSKweuYe34bNRA   1   1         84           43    804.3kb        804.3kb
yellow open   ns_tenant-test-index                     kpstIa1XTT6E34WIpMkhKQ   1   1       1152            0    195.7kb        195.7kb
yellow open   ns_tenant-test-index_1                   uCTJgzccQ5-th9tlJZqo6g   1   1       1039            0    165.3kb        165.3kb
yellow open   suggestion_example                       rFn_S4aSQP2um0QUKA0tpA   1   1          2            0     15.4kb         15.4kb
```

Figure 18: Indexes in storage

To create an e-commerce index, you need to run the data-loader module and specify the path to the data. The data file is shown in Fig. 19.

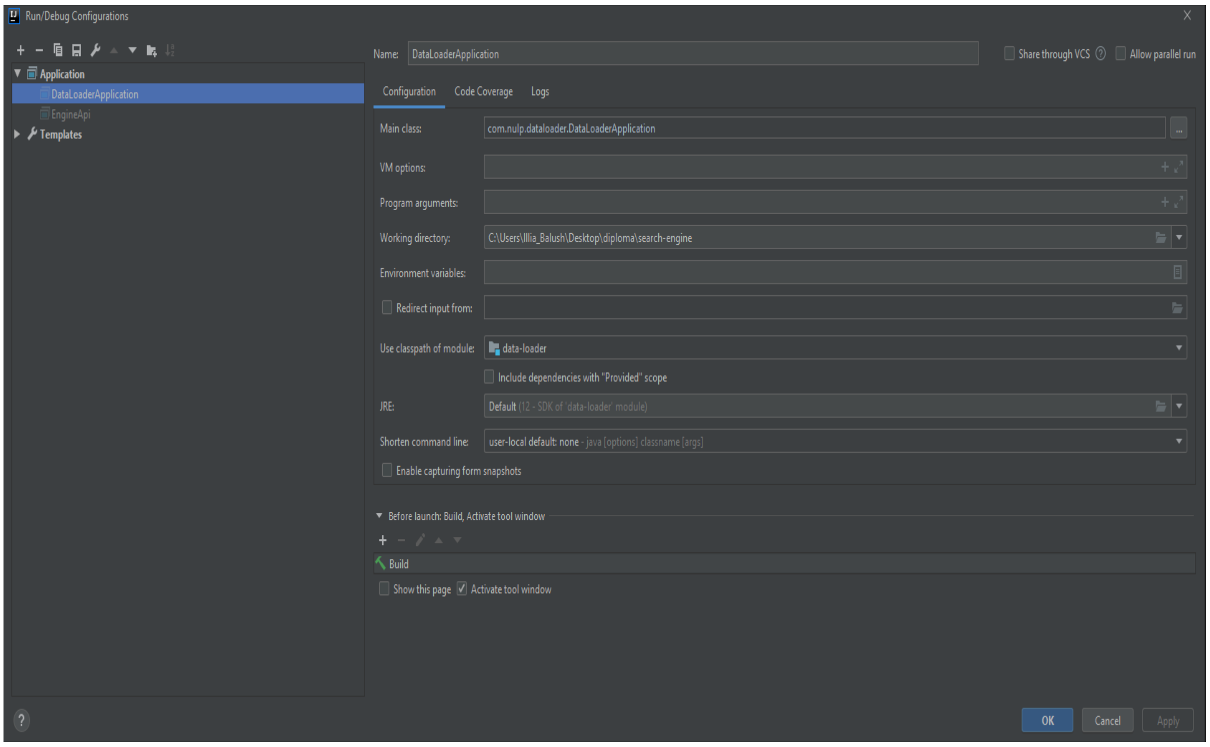


A1 _unit_id

	A	B	C	D	E	F	G	H	I	J	K	L
1	_unit_id	relevance	relevance	product_i	product_li	product_p	product_t	query	rank	source	url	product_description
2	7.11E+08	3.67	0.471	http://thu	http://ww	$329.98	Sony Play	playstatio	1	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
3	7.11E+08	4	0	http://thu	http://ww	$324.84	Sony Play	playstatio	2	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
4	7.11E+08	4	0	http://thu	http://ww	$324.83	Sony Play	playstatio	3	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
5	7.11E+08	3.67	0.471	http://thu	http://ww	$350.00	Sony - Pla	playstatio	4	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
6	7.11E+08	3.33	0.471	http://thu	http://ww	$308.00	Sony Play	playstatio	5	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
7	7.11E+08	3.2	0.748	http://thu	http://ww	$310.00	Sony Play	playstatio	6	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
8	7.11E+08	4	0	http://thu	http://ww	$910.00	BRAND NE	playstatio	7	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
9	7.11E+08	4	0	http://thu	http://ww	$299.99	Sony Play	playstatio	8	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
10	7.11E+08	3.75	0.433	http://thu	http://ww	$350.00	Sony Play	playstatio	9	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
11	7.11E+08	2.33	0.471	http://thu	http://ww	$469.97	Sony Play	playstatio	10	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
12	7.11E+08	4	0	http://thu	http://ww	$390.00	Sony Play	playstatio	11	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
13	7.11E+08	4	0	http://thu	http://ww	$289.99	Genuine S	playstatio	12	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
14	7.11E+08	3.33	0.471	http://thu	http://ww	$443.60	[Sony] Pla	playstatio	13	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
15	7.11E+08	3	0.894	http://thu	http://ww	$349.99	Sony Play	playstatio	14	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
16	7.11E+08	3.4	0.8	http://thu	http://ww	$285.00	Sony Play	playstatio	15	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
17	7.11E+08	4	0	http://thu	http://ww	$399.99	Sony Play	playstatio	16	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
18	7.11E+08	2.8	1.166	http://thu	http://ww	$285.00	Sony Play	playstatio	17	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
19	7.11E+08	4	0	http://thu	http://ww	$340.00	Sony Play	playstatio	18	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
20	7.11E+08	3	0	http://thu	http://ww	$389.99	Sony Play	playstatio	19	eBay	http://ww	The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deeply connected gaming community. Step into living, breathing worlds where
21	7.11E+08	2.33	0.471	http://thu	http://ww	$399.99	Sony - Pla	playstatio	20	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
22	7.11E+08	4	0	http://thu	http://ww	$380.00	Sony Play	playstatio	21	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
23	7.11E+08	4	0	http://thu	http://ww	$360.00	Sony Play	playstatio	22	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
24	7.11E+08	4	0	http://thu	http://ww	$285.00	Sony Play	playstatio	23	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
25	7.11E+08	4	0	http://thu	http://ww	$315.00	Sony Play	playstatio	24	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
26	7.11E+08	4	0	http://thu	http://ww	$467.20	[Sony] Pla	playstatio	25	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204	
27	7.11E+08	4	0	http://thu	http://ww	$8.99	NEW! Clin	eye cream	1	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
28	7.11E+08	3.4	0.8	http://thu	http://ww	$49.72	Obagi Elas	eye cream	2	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
29	7.11E+08	4	0	http://thu	http://ww	$28.79	Lancome I	eye cream	3	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
30	7.11E+08	3.75	0.433	http://thu	http://ww	$19.79	OZ Natura	eye cream	4	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
31	7.11E+08	4	0	http://thu	http://ww	$49.00	Elastidern	eye cream	5	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
32	7.11E+08	4	0	http://thu	http://ww	$8.96	new CLINI	eye cream	6	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
33	7.11E+08	4	0	http://thu	http://ww	$29.00	NIB Full Si	eye cream	7	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
34	7.11E+08	4	0	http://thu	http://ww	$5.89	24K Gold (	eye cream	8	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
35	7.11E+08	3.67	0.471	http://thu	http://ww	$34.99	Clinique F	eye cream	9	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
36	7.11E+08	3.75	0.433	http://thu	http://ww	$8.99	NEW! Clin	eye cream	10	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
37	7.11E+08	4	0	http://thu	http://ww	$21.97	3 Clinique	eye cream	11	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
38	7.11E+08	4	0	http://thu	http://ww	$23.50	Lancome I	eye cream	12	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	
39	7.11E+08	4	0	http://thu	http://ww	$9.99	NEW! Clin	eye cream	13	eBay	http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=eye%20cream	

Figure 19: Data file

This file contains all the necessary information about the goods. To update product information, one needs to prepare a file of the same structure, delete the previous index and run the data-loader module. Upload the data to the repository. Using IntelliJ IDEA, one needs to create the next entry point into the program and run it.



Name: DataLoaderApplication

Main class: com.nulp.dataloader.DataLoaderApplication

VM options:

Program arguments:

Working directory: C:\Users\Illia_Balush\Desktop\diploma\search-engine

Environment variables:

Redirect input from:

Use classpath of module: data-loader

Include dependencies with "Provided" scope

JRE: Default (12 - SDK of 'data-loader' module)

Shorten command line: user-local default: none - java [options] classname [args]

Enable capturing form snapshots

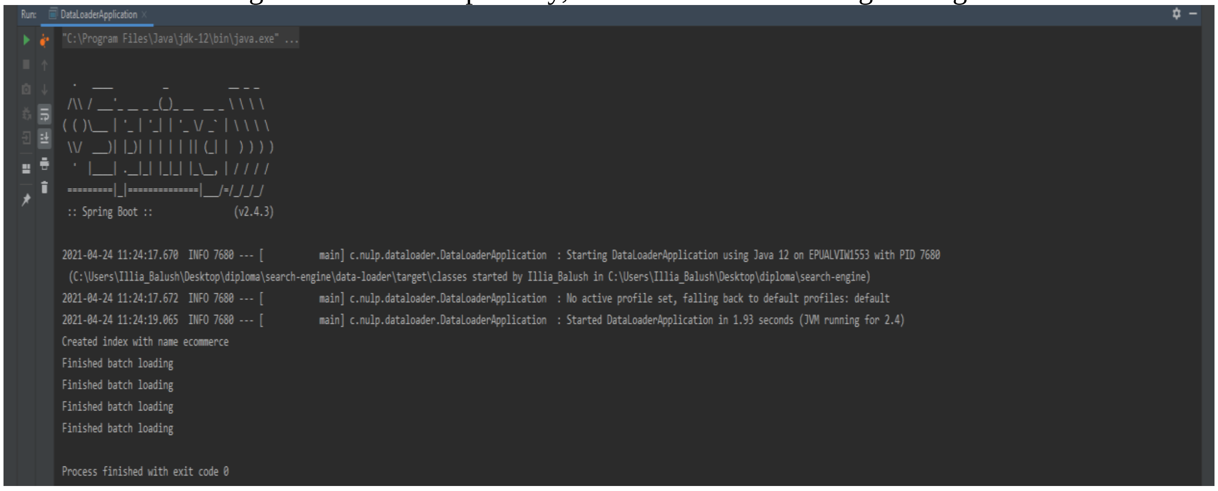
Before launch: Build, Activate tool window

Build

Show this page [✓ Activate tool window]

Figure 20: Data-loader settings

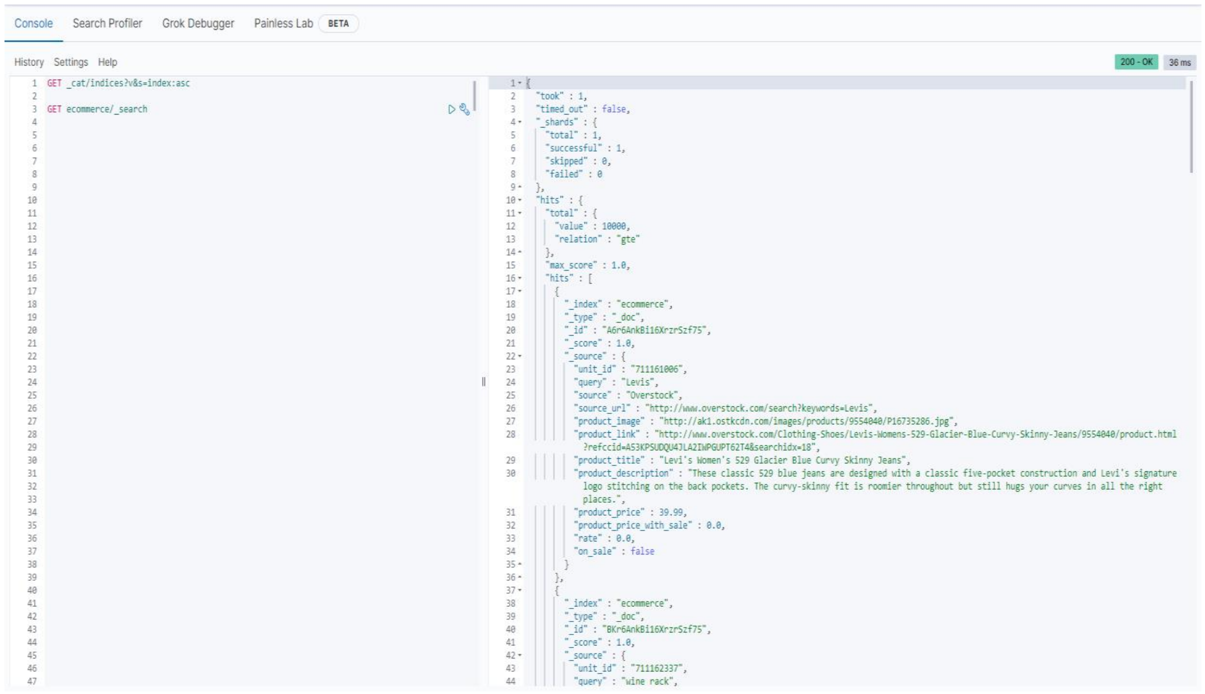
After downloading the data to the repository, we will see the following messages:



```
"C:\Program Files\Java\jdk-12\bin\java.exe" ...

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.4.3)

2021-04-24 11:24:17.670  INFO 7680 --- [           main] c.nulp.dataloader.DataLoaderApplication  : Starting DataLoaderApplication using Java 12 on EPUALVIW1553 with PID 7680
 (C:\Users\Illia_Balush\Desktop\diploma\search-engine\data-loader\target\classes started by Illia_Balush in C:\Users\Illia_Balush\Desktop\diploma\search-engine)
2021-04-24 11:24:17.672  INFO 7680 --- [           main] c.nulp.dataloader.DataLoaderApplication  : No active profile set, falling back to default profiles: default
2021-04-24 11:24:19.065  INFO 7680 --- [           main] c.nulp.dataloader.DataLoaderApplication  : Started DataLoaderApplication in 1.93 seconds (JVM running for 2.4)
Created index with name ecommerce
Finished batch loading
Finished batch loading
Finished batch loading
Finished batch loading

Process finished with exit code 0
```

Figure 21: Messages after downloading data

Now let's check the data in our repository. To do this, again, use Kibana. We will perform a search query on the index named e-commerce.



```
GET _cat/indices?v&s=index:asc

GET ecommerce/_search
```

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 10000,
      "relation" : "gte"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "ecommerce",
        "_type" : "_doc",
        "_id" : "A6r6AnkBi16XrzrSzf75",
        "_score" : 1.0,
        "_source" : {
          "unit_id" : "711161006",
          "query" : "Levis",
          "source" : "Overstock",
          "source_url" : "http://www.overstock.com/search?keywords=Levis",
          "product_image" : "http://ak1.ostkcdn.com/images/products/9554040/P16735286.jpg",
          "product_link" : "http://www.overstock.com/Clothing-Shoes/Levis-Womens-529-Glacier-Blue-Curvy-Skinny-Jeans/9554040/product.html
            ?refccid=A53KPSUDQU4JLA2IWPGUPT62T4&searchidx=18",
          "product_title" : "Levi's Women's 529 Glacier Blue Curvy Skinny Jeans",
          "product_description" : "These classic 529 blue jeans are designed with a classic five-pocket construction and Levi's signature
            logo stitching on the back pockets. The curvy-skinny fit is roomier throughout but still hugs your curves in all the right
            places.",
          "product_price" : 39.99,
          "product_price_with_sale" : 0.0,
          "rate" : 0.0,
          "on_sale" : false
        }
      },
      {
        "_index" : "ecommerce",
        "_type" : "_doc",
        "_id" : "BKr6AnkBi16XrzrSzf75",
        "_score" : 1.0,
        "_source" : {
          "unit_id" : "711162337",
          "query" : "wine rack",
```

Figure 22: Data on goods in storage

Having prepared the data to be searched, we will create tables in MySQL to monitor purchases and recommendations among users. Fig. 23 shows that so far, we do not have any database.

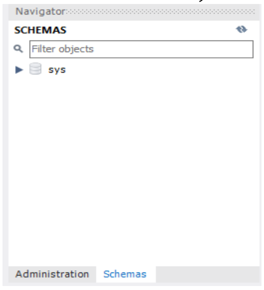
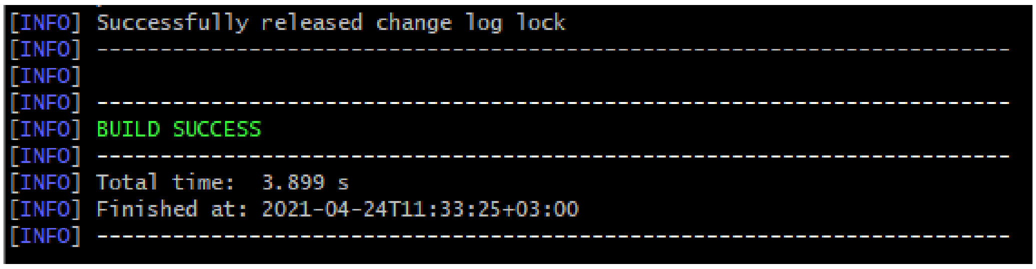


Figure 23: MySQL database

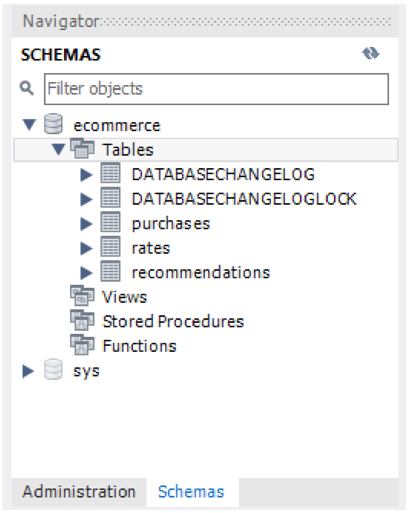
We will receive the following message by executing the command mvn liquibase: update -Denv = dev in the engine-server module.



```
[INFO] Successfully released change log lock
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.899 s
[INFO] Finished at: 2021-04-24T11:33:25+03:00
[INFO] ------------------------------------------------------------------------
```

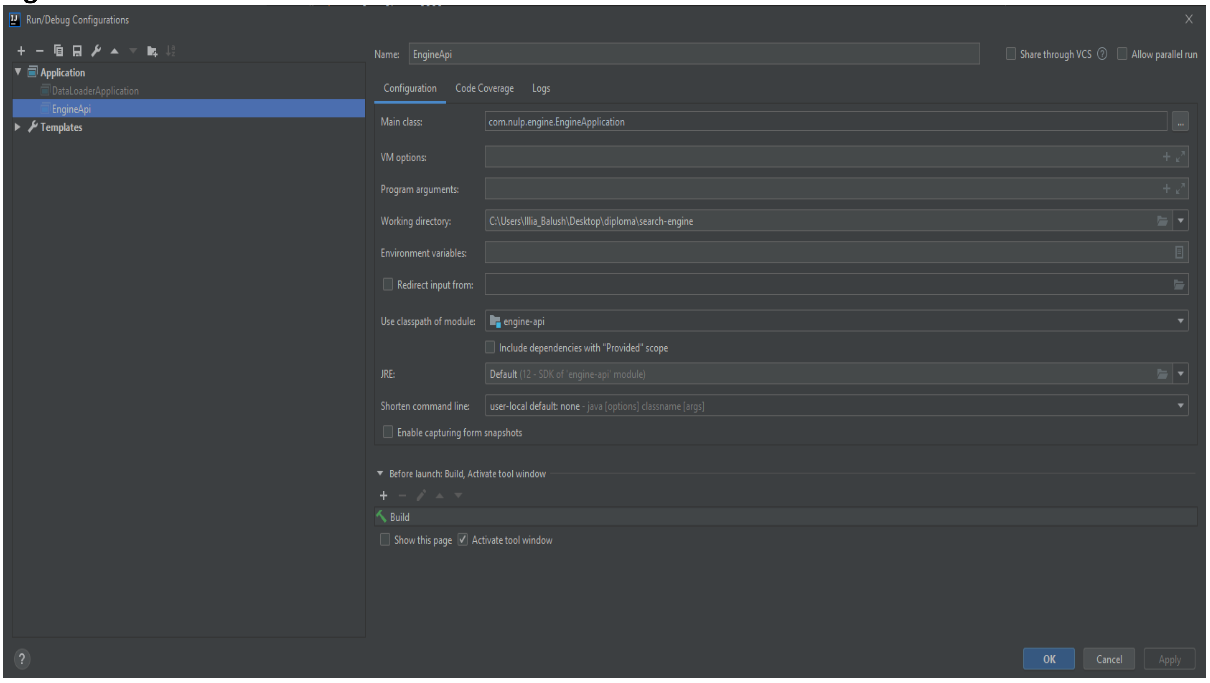
Figure 24: Notification of successful database creation

And let's recheck the database. Fig. 25 shows the new e-commerce database and tables: purchases, rates, and recommendations. Now that all the necessary conditions are met for the program to run successfully let's try to run it. Using IntelliJ IDEA, we will create an entry point, as is shown in Fig. 26.



```
Navigator
SCHEMAS
Filter objects
▼ ecommerce
  ▼ Tables
    ▶ DATABASECHANGELOG
    ▶ DATABASECHANGELOGLOCK
    ▶ purchases
    ▶ rates
    ▶ recommendations
    Views
    Stored Procedures
    Functions
▶ sys
Administration   Schemas
```

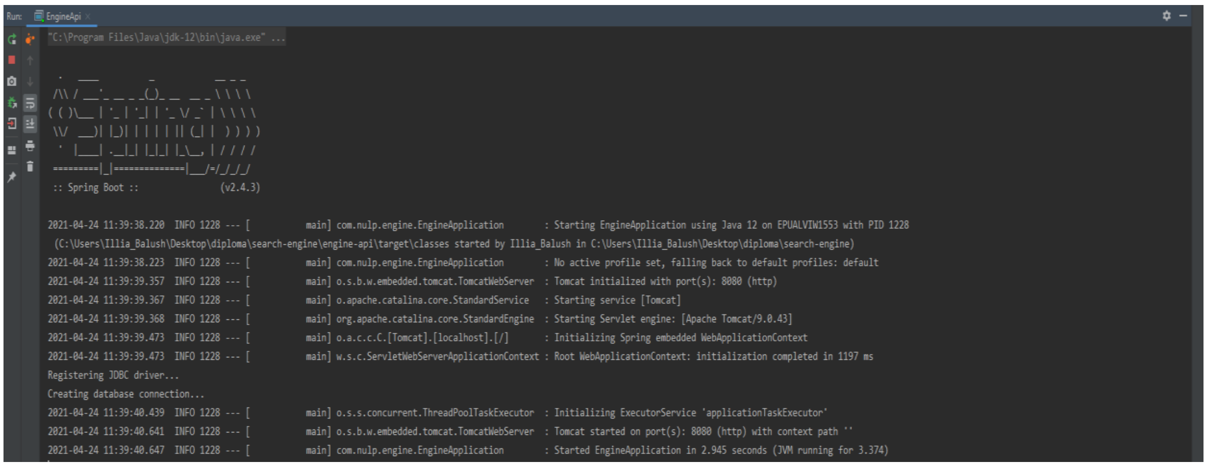
Figure 25: Ecommerce database



```
Run/Debug Configurations
▼ Application
    DataLoaderApplication
    EngineApi
▶ Templates

Name: EngineApi                          ☐ Share through VCS   ☐ Allow parallel run
Configuration   Code Coverage   Logs
Main class:                com.nulp.engine.EngineApplication
VM options:
Program arguments:
Working directory:         C:\Users\Illia_Balush\Desktop\diploma\search-engine
Environment variables:
☐ Redirect input from:
Use classpath of module:   engine-api
                           ☐ Include dependencies with "Provided" scope
JRE:                       Default (12 - SDK of 'engine-api' module)
Shorten command line:      user-local default: none - java [options] classname [args]
☐ Enable capturing form snapshots

▼ Before launch: Build, Activate tool window
Build
☐ Show this page  ☑ Activate tool window

OK   Cancel   Apply
```

Figure 26: Engine-API settings

After a successful launch, we should see the following messages:



```
"C:\Program Files\Java\jdk-12\bin\java.exe" ...

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.4.3)

2021-04-24 11:39:38.220  INFO 1228 --- [           main] com.nulp.engine.EngineApplication        : Starting EngineApplication using Java 12 on EPUALVIW1553 with PID 1228
 (C:\Users\Illia_Balush\Desktop\diploma\search-engine\engine-api\target\classes started by Illia_Balush in C:\Users\Illia_Balush\Desktop\diploma\search-engine)
2021-04-24 11:39:38.223  INFO 1228 --- [           main] com.nulp.engine.EngineApplication        : No active profile set, falling back to default profiles: default
2021-04-24 11:39:39.357  INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 8080 (http)
2021-04-24 11:39:39.367  INFO 1228 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2021-04-24 11:39:39.368  INFO 1228 --- [           main] org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache Tomcat/9.0.43]
2021-04-24 11:39:39.473  INFO 1228 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2021-04-24 11:39:39.473  INFO 1228 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1197 ms
Registering JDBC driver...
Creating database connection...
2021-04-24 11:39:40.439  INFO 1228 --- [           main] o.s.s.concurrent.ThreadPoolTaskExecutor  : Initializing ExecutorService 'applicationTaskExecutor'
2021-04-24 11:39:40.641  INFO 1228 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080 (http) with context path ''
2021-04-24 11:39:40.647  INFO 1228 --- [           main] com.nulp.engine.EngineApplication        : Started EngineApplication in 2.945 seconds (JVM running for 3.374)
```

Figure 27: Messages after successful startup

We will perform several search queries. We use Postman to send search queries. Let's try to look at what products the program will offer us, looking for the phrase "wi r".

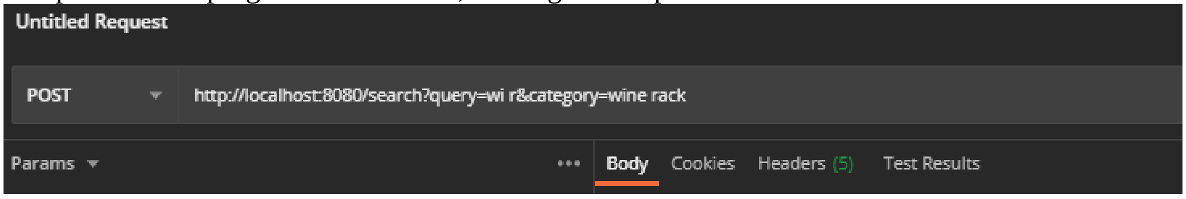
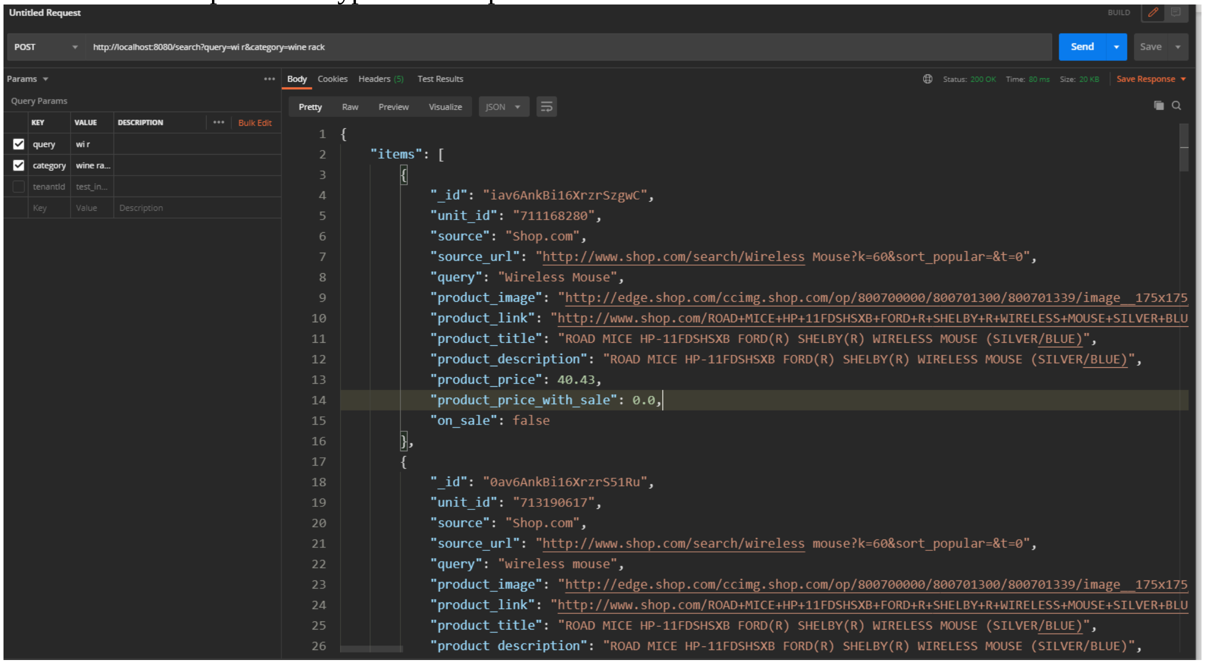


Figure 28: Search for the phrase "wi r"

Fig. 29 shows that the program has found several products. The top 5 products that meet the search query are:

- Wireless mouse is two types - wireless mouse.
- WII (R) - 5 types - game console.
- wii microphone - 3 types – microphone.



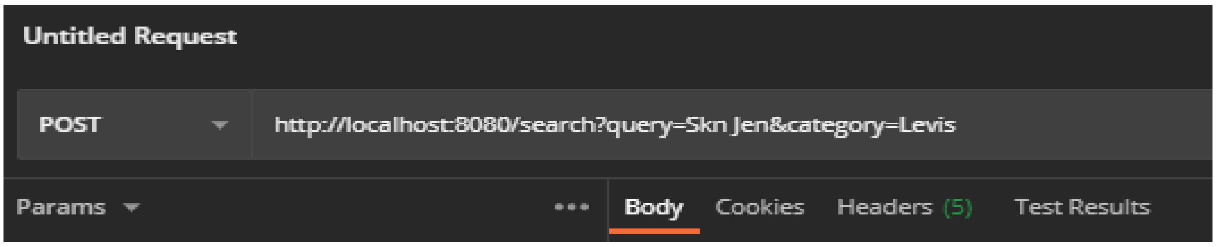
Query Params

	KEY	VALUE	DESCRIPTION
✓	query	wi r	
✓	category	wine ra...	
	tenantId	test_in...	
	Key	Value	Description

```
{
    "items": [
        {
            "_id": "iav6AnkBi16XrzrSzgwC",
            "unit_id": "711168280",
            "source": "Shop.com",
            "source_url": "http://www.shop.com/search/Wireless Mouse?k=60&sort_popular=&t=0",
            "query": "Wireless Mouse",
            "product_image": "http://edge.shop.com/ccimg.shop.com/op/800700000/800701300/800701339/image__175x175
            "product_link": "http://www.shop.com/ROAD+MICE+HP+11FDSHSXB+FORD+R+SHELBY+R+WIRELESS+MOUSE+SILVER+BLU
            "product_title": "ROAD MICE HP-11FDSHSXB FORD(R) SHELBY(R) WIRELESS MOUSE (SILVER/BLUE)",
            "product_description": "ROAD MICE HP-11FDSHSXB FORD(R) SHELBY(R) WIRELESS MOUSE (SILVER/BLUE)",
            "product_price": 40.43,
            "product_price_with_sale": 0.0,
            "on_sale": false
        },
        {
            "_id": "0av6AnkBi16XrzrS51Ru",
            "unit_id": "713190617",
            "source": "Shop.com",
            "source_url": "http://www.shop.com/search/wireless mouse?k=60&sort_popular=&t=0",
            "query": "wireless mouse",
            "product_image": "http://edge.shop.com/ccimg.shop.com/op/800700000/800701300/800701339/image__175x175
            "product_link": "http://www.shop.com/ROAD+MICE+HP+11FDSHSXB+FORD+R+SHELBY+R+WIRELESS+MOUSE+SILVER+BLU
            "product_title": "ROAD MICE HP-11FDSHSXB FORD(R) SHELBY(R) WIRELESS MOUSE (SILVER/BLUE)",
            "product description": "ROAD MICE HP-11FDSHSXB FORD(R) SHELBY(R) WIRELESS MOUSE (SILVER/BLUE)",
```

Figure 29: Search results for the phrase "wi r"

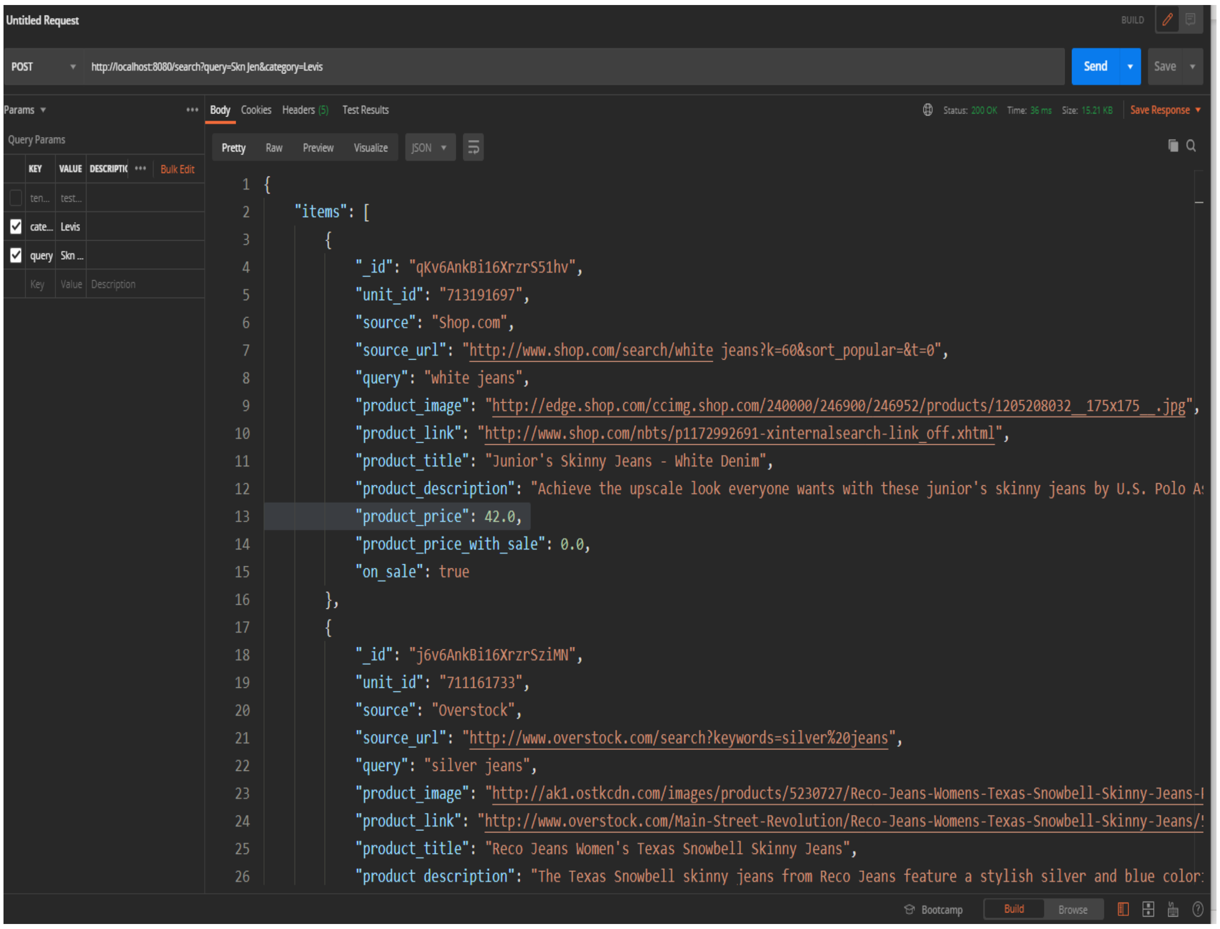
Let's try to find "Skinny Jeans", but making several mistakes in the search query. Instead of "Skinny Jeans" we will look for "Skn Jen".



```
POST  http://localhost:8080/search?query=Skn Jen&category=Levis
```

Figure 30: Search by the phrase "Skn Jen"

Fig. 31 shows the search result. The program found the goods despite a vague request. Levenstein's fuzzy search algorithm helped us in this. In addition, the result in Fig. 32 is sorted by relevance. Let's try to sort these results in descending order of price. Fig. 33 shows the first product has a price of 6135.0, and the second after it - 4260.99. Thus, the goods are sorted in descending order of price.



```
POST  http://localhost:8080/search?query=Skn Jen&category=Levis
```

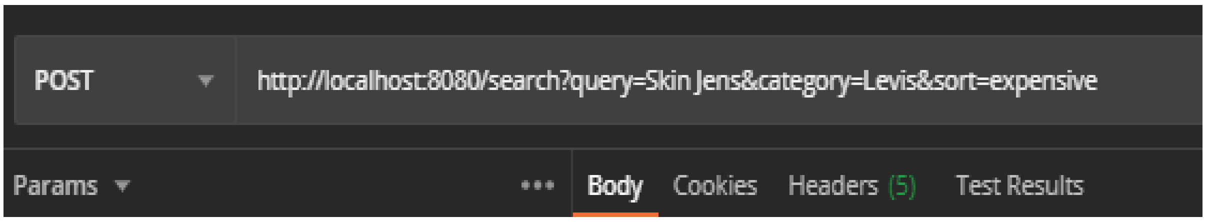
Status: 200 OK Time: 36 ms Size: 15.21 KB

Query Params

	KEY	VALUE	DESCRIPTIC
	ten...	test...	
✓	cate...	Levis	
✓	query	Skn ...	
	Key	Value	Description

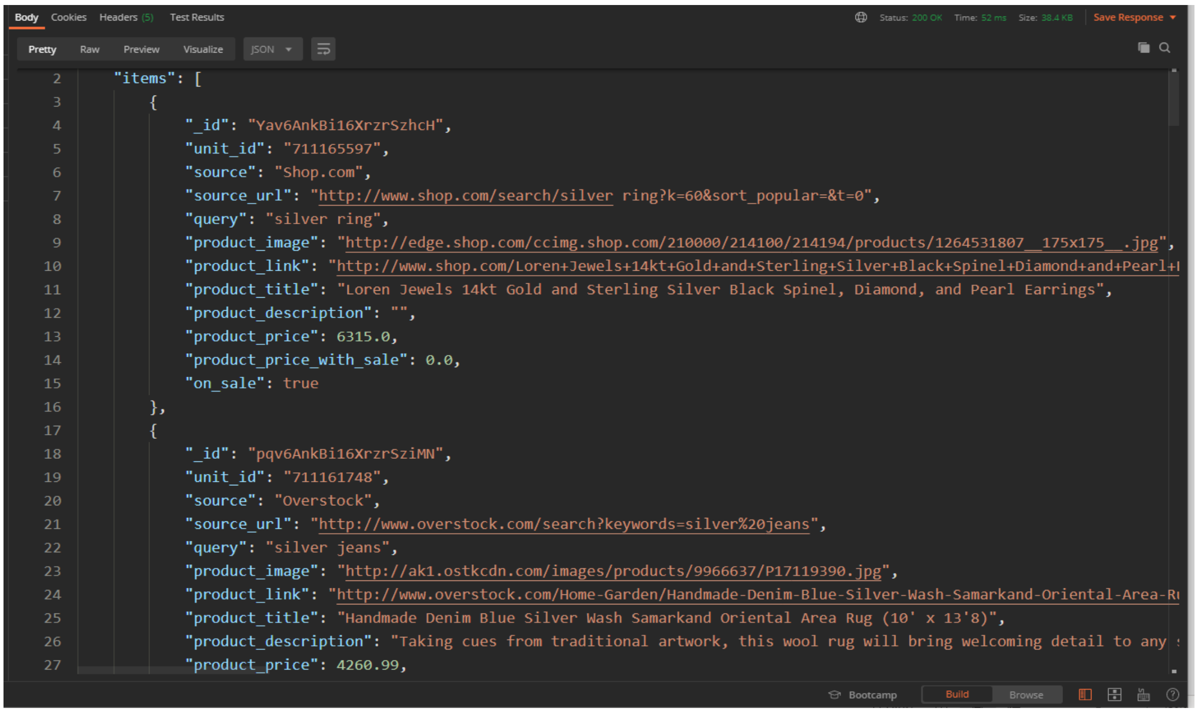
```
{
    "items": [
        {
            "_id": "qKv6AnkBi16XrzrS51hv",
            "unit_id": "713191697",
            "source": "Shop.com",
            "source_url": "http://www.shop.com/search/white jeans?k=60&sort_popular=&t=0",
            "query": "white jeans",
            "product_image": "http://edge.shop.com/ccimg.shop.com/240000/246900/246952/products/1205208032__175x175__.jpg",
            "product_link": "http://www.shop.com/nbts/p1172992691-xinternalsearch-link_off.xhtml",
            "product_title": "Junior's Skinny Jeans - White Denim",
            "product_description": "Achieve the upscale look everyone wants with these junior's skinny jeans by U.S. Polo A
            "product_price": 42.0,
            "product_price_with_sale": 0.0,
            "on_sale": true
        },
        {
            "_id": "j6v6AnkBi16XrzrSziMN",
            "unit_id": "711161733",
            "source": "Overstock",
            "source_url": "http://www.overstock.com/search?keywords=silver%20jeans",
            "query": "silver jeans",
            "product_image": "http://ak1.ostkcdn.com/images/products/5230727/Reco-Jeans-Womens-Texas-Snowbell-Skinny-Jeans-
            "product_link": "http://www.overstock.com/Main-Street-Revolution/Reco-Jeans-Womens-Texas-Snowbell-Skinny-Jeans/
            "product_title": "Reco Jeans Women's Texas Snowbell Skinny Jeans",
            "product description": "The Texas Snowbell skinny jeans from Reco Jeans feature a stylish silver and blue color
```

Figure 31: Search Results for "Skn Jen"



```
POST  http://localhost:8080/search?query=Skin Jens&category=Levis&sort=expensive
```

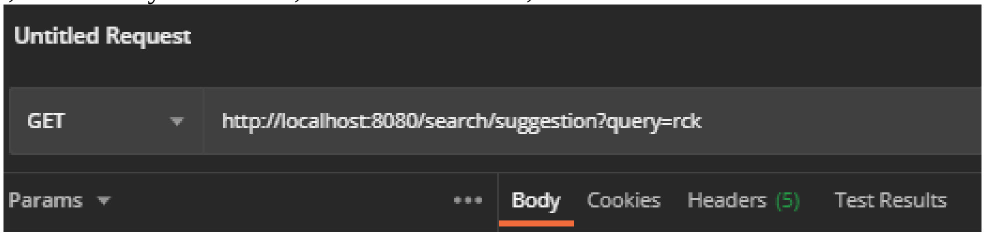
Figure 32: Sorting in descending order



```
    "items": [
        {
            "_id": "Yav6AnkBi16XrzrSzhcH",
            "unit_id": "711165597",
            "source": "Shop.com",
            "source_url": "http://www.shop.com/search/silver ring?k=60&sort_popular=&t=0",
            "query": "silver ring",
            "product_image": "http://edge.shop.com/ccimg.shop.com/210000/214100/214194/products/1264531807__175x175__.jpg",
            "product_link": "http://www.shop.com/Loren+Jewels+14kt+Gold+and+Sterling+Silver+Black+Spinel+Diamond+and+Pearl+E
            "product_title": "Loren Jewels 14kt Gold and Sterling Silver Black Spinel, Diamond, and Pearl Earrings",
            "product_description": "",
            "product_price": 6315.0,
            "product_price_with_sale": 0.0,
            "on_sale": true
        },
        {
            "_id": "pqv6AnkBi16XrzrSziMN",
            "unit_id": "711161748",
            "source": "Overstock",
            "source_url": "http://www.overstock.com/search?keywords=silver%20jeans",
            "query": "silver jeans",
            "product_image": "http://ak1.ostkcdn.com/images/products/9966637/P17119390.jpg",
            "product_link": "http://www.overstock.com/Home-Garden/Handmade-Denim-Blue-Silver-Wash-Samarkand-Oriental-Area-Ru
            "product_title": "Handmade Denim Blue Silver Wash Samarkand Oriental Area Rug (10' x 13'8)",
            "product_description": "Taking cues from traditional artwork, this wool rug will bring welcoming detail to any s
            "product_price": 4260.99,
```

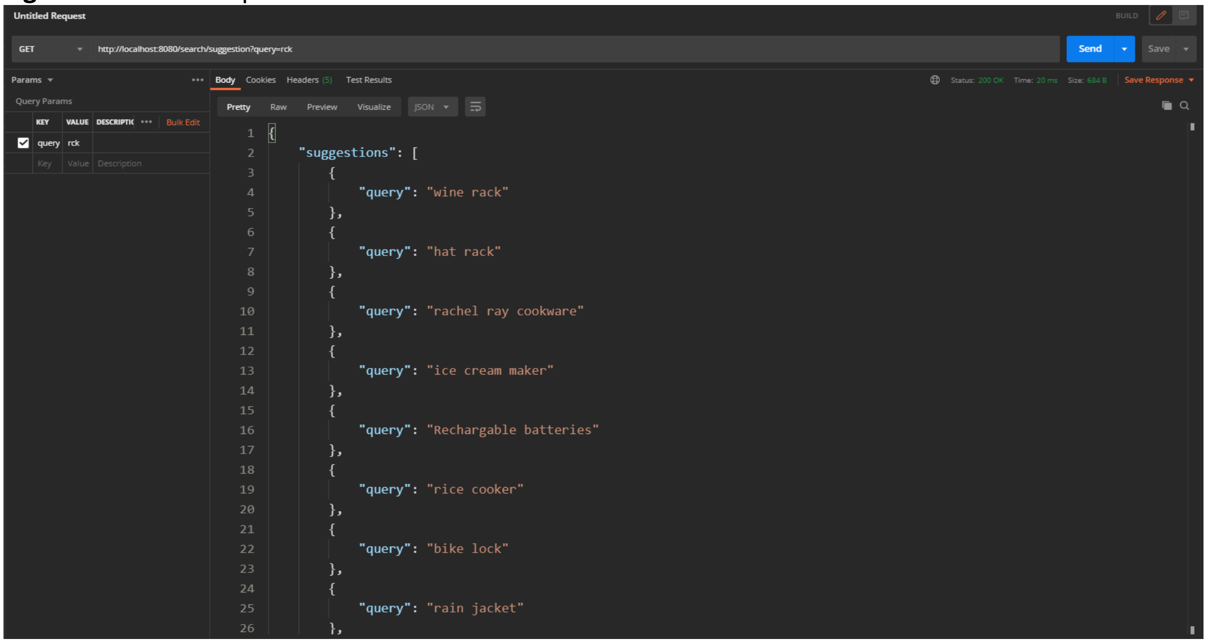
Figure 33: Sort result

Let's see how auto-completion of search words works. To do this, imagine that the user typed the word "rck" in the search bar. Fig. 35 shows the options for completing the word "rck" => "wine rack", "hat rack", "Rachel ray cookware", "ice cream maker", etc.



```
http://localhost:8080/search/suggestion?query=rck
```

Figure 34: AutoComplete the word "rck"



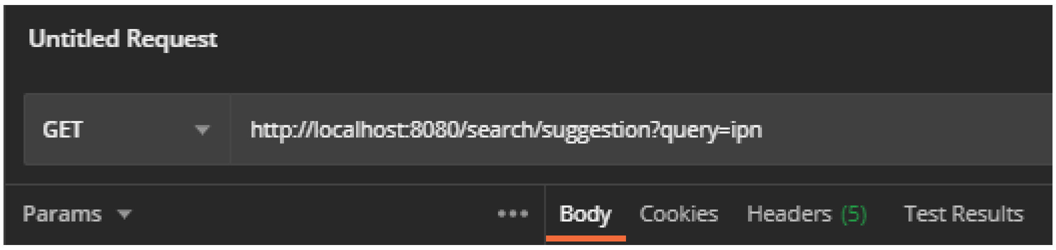
```
http://localhost:8080/search/suggestion?query=rck
```

KEY	VALUE	DESCRIPTION
✓ query	rck	

```
{
    "suggestions": [
        {
            "query": "wine rack"
        },
        {
            "query": "hat rack"
        },
        {
            "query": "rachel ray cookware"
        },
        {
            "query": "ice cream maker"
        },
        {
            "query": "Rechargable batteries"
        },
        {
            "query": "rice cooker"
        },
        {
            "query": "bike lock"
        },
        {
            "query": "rain jacket"
        },
```

Figure 35: Auto-completion results for the word "rck"

Let's try this functionality for the word "ipn", meaning the word "iphone".

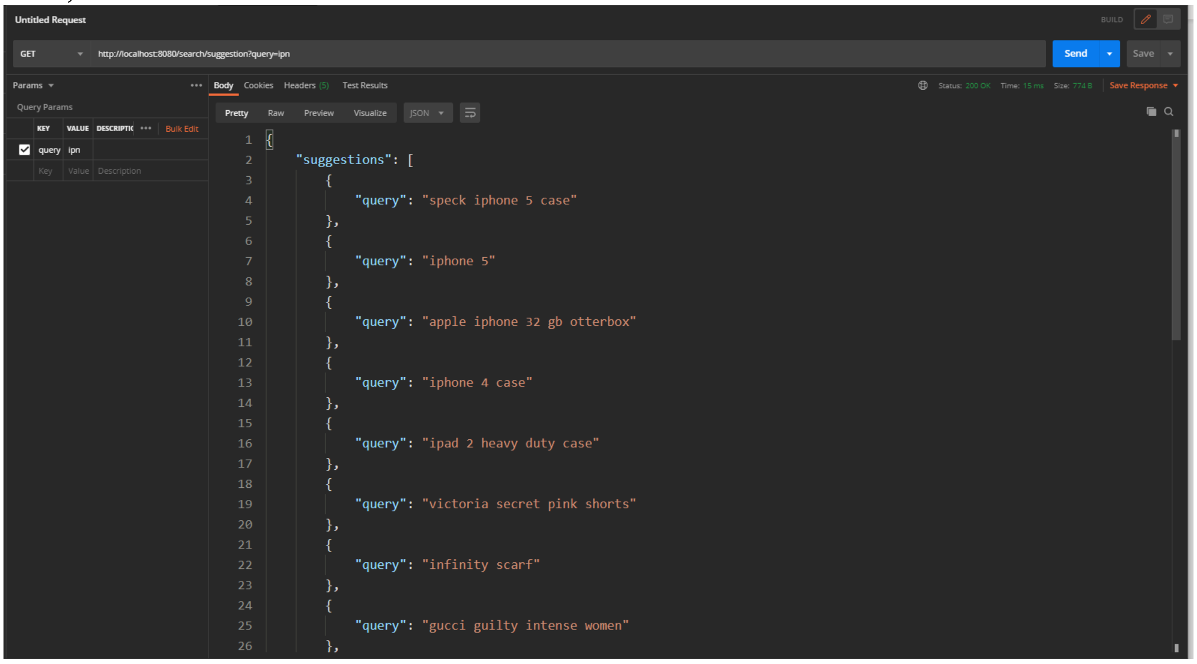


Untitled Request

GET http://localhost:8080/search/suggestion?query=ipn

Figure 36: Auto-completion of the word "Ipn"

According to Fig. 37 the program offered us the following options for completing this word: "speck iPhone 5 case", "iphone 5", "apple iPhone 32 GB", etc., which in principle is close to the desired. Now fill in the shopping table using the entry point/search/shop. Imagine that we have three users User1, User2, User3.



Untitled Request

GET http://localhost:8080/search/suggestion?query=ipn

Status: 200 OK Time: 15 ms Size: 774 B

KEY	VALUE	DESCRIPTION
query	ipn	

```
{
    "suggestions": [
        {
            "query": "speck iphone 5 case"
        },
        {
            "query": "iphone 5"
        },
        {
            "query": "apple iphone 32 gb otterbox"
        },
        {
            "query": "iphone 4 case"
        },
        {
            "query": "ipad 2 heavy duty case"
        },
        {
            "query": "victoria secret pink shorts"
        },
        {
            "query": "infinity scarf"
        },
        {
            "query": "gucci guilty intense women"
        },
```

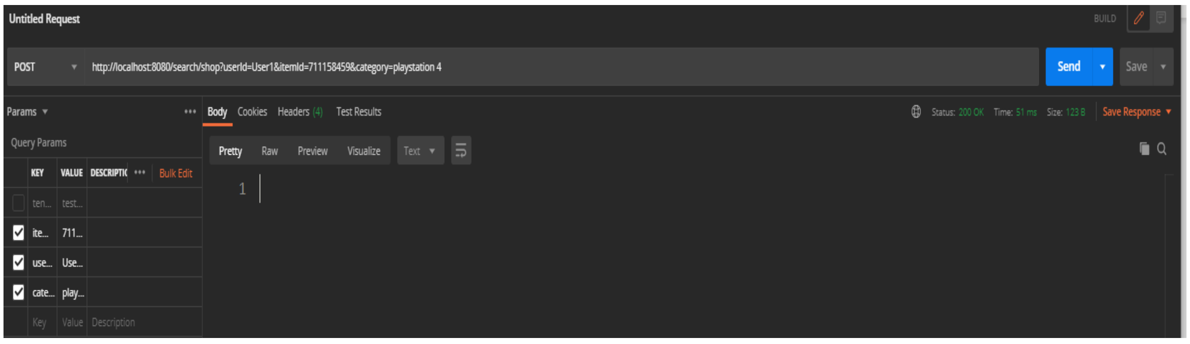
Figure 37: Auto-completion of the word "ipn"

Table 5
User purchases

User	Purchased goods
User1	Sony PlayStation 4 (PS4) (Latest Model) - 500 GB Jet Black Console (711158459)
	Porter-Cable 3-1 / 4 HP Five-Speed Router 7518 - Power Tools Routers (711158511)
	Cosco Slim Fold High Chair, Kontiki, Free Shipping, New (711158585)
	2.4GHz Red Wireless Optical Mouse Mice (711158784)
	Anti-Venom IN HAND Marvel Legends (711158812)
User2	Sony PlayStation 4 (PS4) (Latest Model) - 500 GB Jet Black Console (711158459)
	2.4GHz Red Wireless Optical Mouse Mice (711158784)
	Porter-Cable 3-1 / 4 HP Five-Speed Router 7518 - Power Tools Routers (711158511)
	iPhone 5 Ease Fit Arm Band (711168736)
	Nintendo Wii U Pro Controller (711175233)
	Oster 14 Speed Blender (713196549)
User3	iPhone 5 Ease Fit Arm Band (711168736)
	Sony PlayStation 4 (PS4) (Latest Model) - 500 GB Jet Black Console (711158459)

Porter-Cable 3-1 / 4 HP Five-Speed Router 7518 - Power Tools Routers (711158511)
Batman Joker Graffiti (713196402)
samsung 60in 1080p 120hz smart led tv (713196223)

Fig. 38 shows an example of a request to purchase goods.

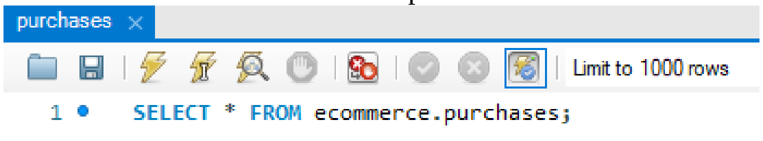


```
POST  http://localhost:8080/search/shop?userId=User1&itemId=711158459&category=playstation 4
```

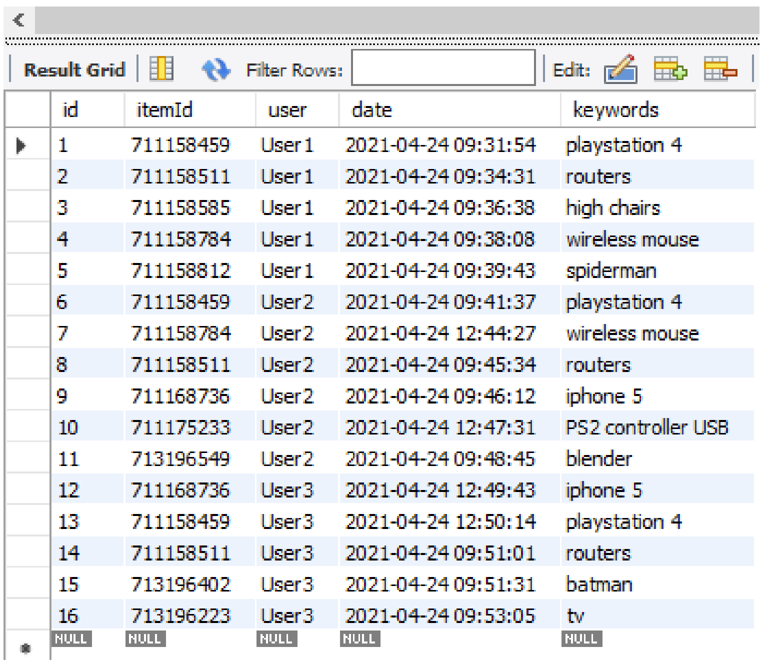
	KEY	VALUE	DESCRIPTION
☐	ten...	test...	
✓	ite...	711...	
✓	use...	Use...	
✓	cate...	play...	
	Key	Value	Description

Figure 38: Example of a request to purchase goods.

We will query the database and check whether all purchases are saved.



```sql
SELECT * FROM ecommerce.purchases;
```



id	itemId	user	date	keywords
1	711158459	User1	2021-04-24 09:31:54	playstation 4
2	711158511	User1	2021-04-24 09:34:31	routers
3	711158585	User1	2021-04-24 09:36:38	high chairs
4	711158784	User1	2021-04-24 09:38:08	wireless mouse
5	711158812	User1	2021-04-24 09:39:43	spiderman
6	711158459	User2	2021-04-24 09:41:37	playstation 4
7	711158784	User2	2021-04-24 12:44:27	wireless mouse
8	711158511	User2	2021-04-24 09:45:34	routers
9	711168736	User2	2021-04-24 09:46:12	iphone 5
10	711175233	User2	2021-04-24 12:47:31	PS2 controller USB
11	713196549	User2	2021-04-24 09:48:45	blender
12	711168736	User3	2021-04-24 12:49:43	iphone 5
13	711158459	User3	2021-04-24 12:50:14	playstation 4
14	711158511	User3	2021-04-24 09:51:01	routers
15	713196402	User3	2021-04-24 09:51:31	batman
16	713196223	User3	2021-04-24 09:53:05	tv
NULL	NULL	NULL	NULL	NULL

Figure 39: Goods of buyers

Now imagine a situation where buyers begin to appreciate the product. The system provides for the case when the buyer cannot evaluate the product if the system does not have a record of this user's purchase of this product. To evaluate the product, we will use the entry point/search/rate. In Table 6 you can conveniently view the ratings of all users.

Table 6
User ratings

User	PS 4	routers	High chair	Wireless mouse	Spiderman	Iphone5	PS 2	Blender	Batman	TV
1	5	4	3	4	3	?	?	?	?	?
2	4	4	?	4	?	3	5	4	?	?
3	2	3	?	?	?	5	?	?	4	3

Fig. 40 shows an example of a query for a user to rate a product.

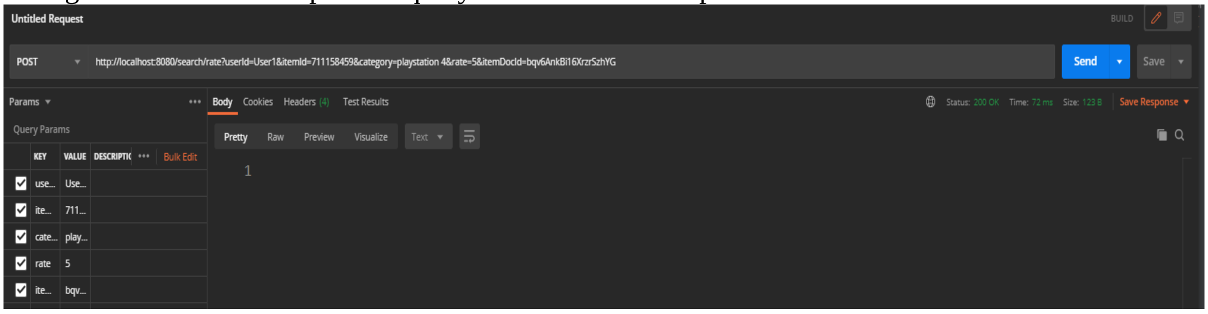
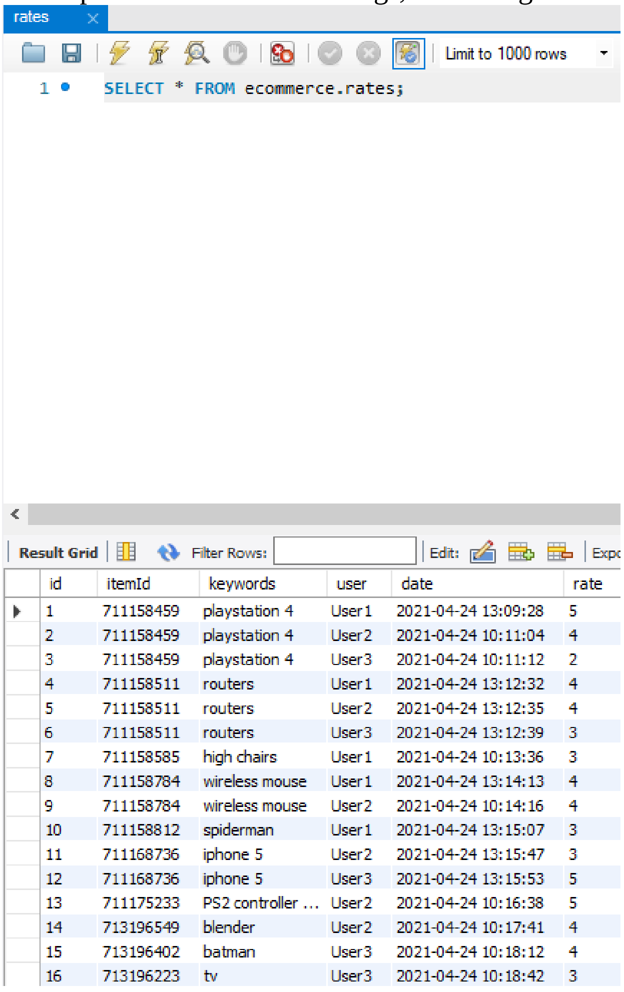

```
Untitled Request
POST  http://localhost:8080/search/rate?userId=User1&itemId=711158459&category=playstation 4&rate=5&itemDocId=bqv6AnkBi16XrzrSzhYG
Status: 200 OK   Time: 72 ms   Size: 123 B

Query Params
KEY    VALUE
use…   Use…
ite…   711…
cate…  play…
rate   5
ite…   bqv…
```

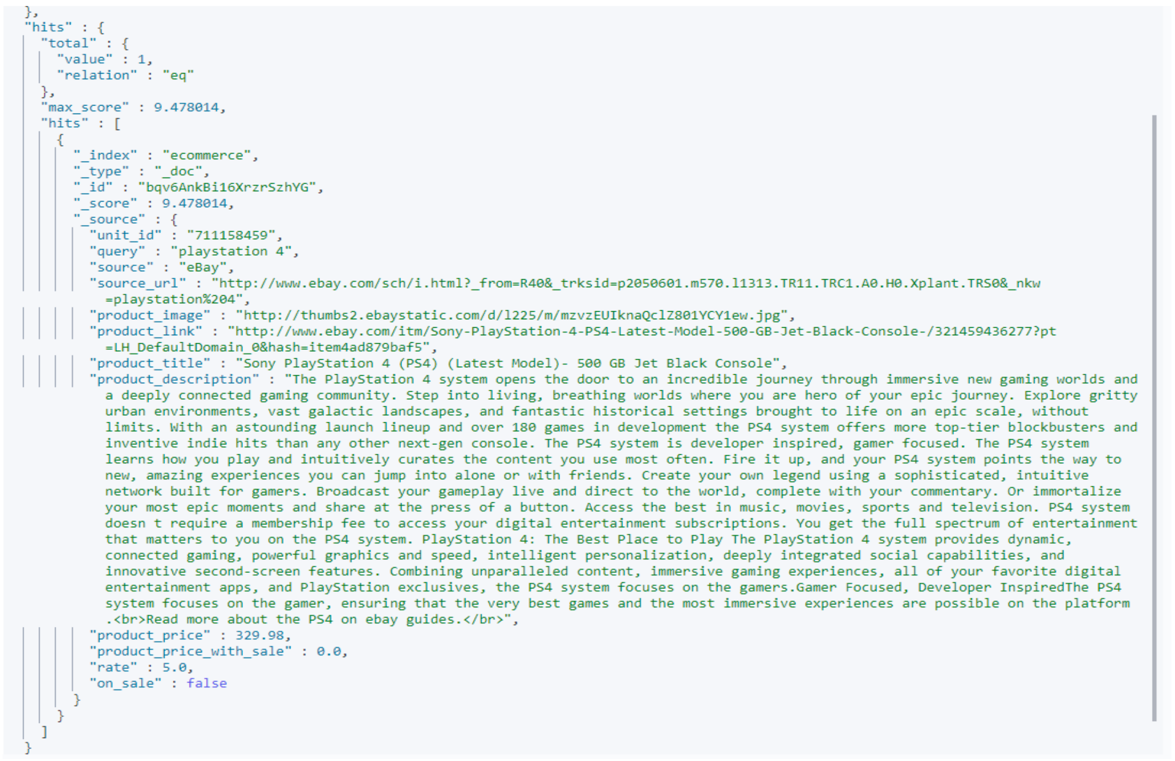
Figure 40: Example of a request to evaluate the product

After evaluating the products, we will review the rating data in the database. In addition, after each evaluation of the product, the field "rate" will change depending on the arithmetic mean of all assessments. Fig. 41 shows the product data to user ratings, and in Fig. 42 after evaluation.


```
SELECT * FROM ecommerce.rates;
```

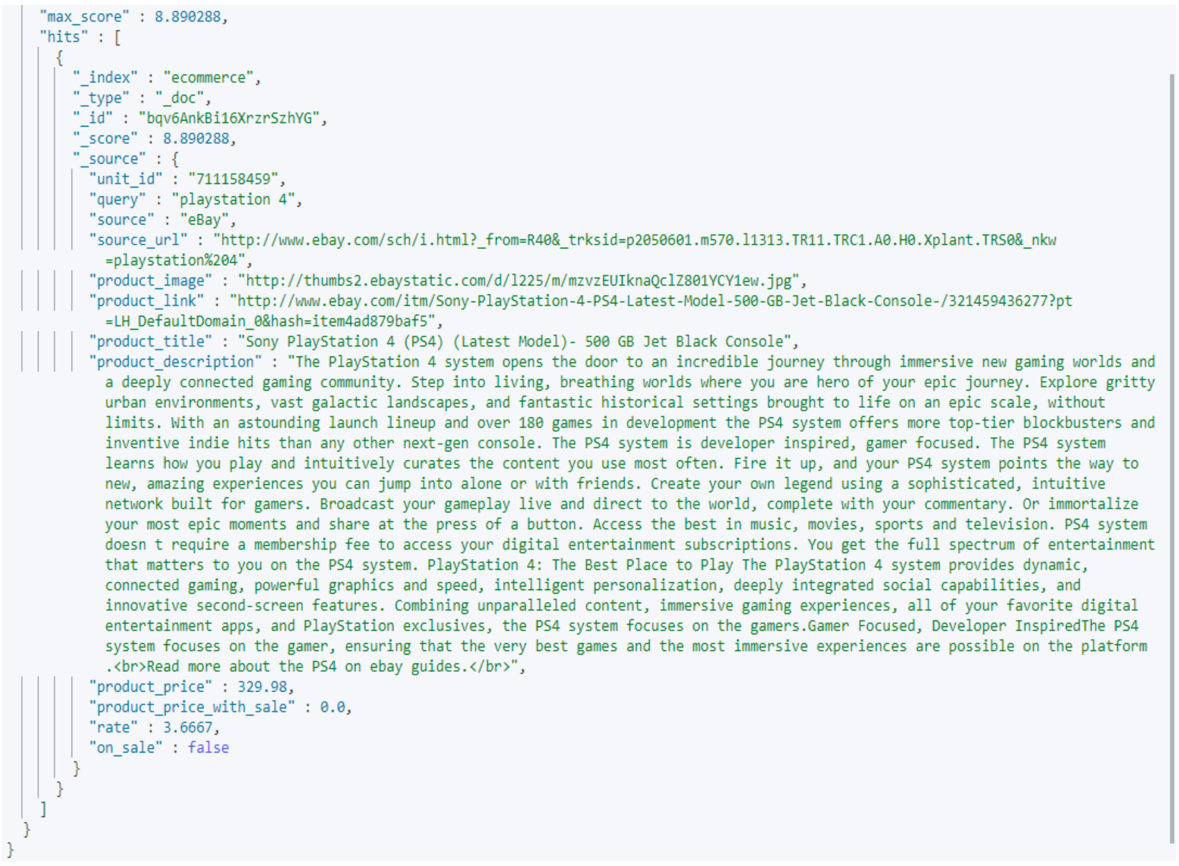
id	itemId	keywords	user	date	rate
1	711158459	playstation 4	User1	2021-04-24 13:09:28	5
2	711158459	playstation 4	User2	2021-04-24 10:11:04	4
3	711158459	playstation 4	User3	2021-04-24 10:11:12	2
4	711158511	routers	User1	2021-04-24 13:12:32	4
5	711158511	routers	User2	2021-04-24 13:12:35	4
6	711158511	routers	User3	2021-04-24 13:12:39	3
7	711158585	high chairs	User1	2021-04-24 10:13:36	3
8	711158784	wireless mouse	User1	2021-04-24 13:14:13	4
9	711158784	wireless mouse	User2	2021-04-24 10:14:16	4
10	711158812	spiderman	User1	2021-04-24 13:15:07	3
11	711168736	iphone 5	User2	2021-04-24 13:15:47	3
12	711168736	iphone 5	User3	2021-04-24 13:15:53	5
13	711175233	PS2 controller ...	User2	2021-04-24 10:16:38	5
14	713196549	blender	User2	2021-04-24 10:17:41	4
15	713196402	batman	User3	2021-04-24 10:18:12	4
16	713196223	tv	User3	2021-04-24 10:18:42	3

Figure 41: User rating data



```
},
"hits" : {
  "total" : {
    "value" : 1,
    "relation" : "eq"
  },
  "max_score" : 9.478014,
  "hits" : [
    {
      "_index" : "ecommerce",
      "_type" : "_doc",
      "_id" : "bqv6AnkBi16XrzrSzhYG",
      "_score" : 9.478014,
      "_source" : {
        "unit_id" : "711158459",
        "query" : "playstation 4",
        "source" : "eBay",
        "source_url" : "http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw
          =playstation%204",
        "product_image" : "http://thumbs2.ebaystatic.com/d/l225/m/mzvzEUIknaQclZ801YCY1ew.jpg",
        "product_link" : "http://www.ebay.com/itm/Sony-PlayStation-4-PS4-Latest-Model-500-GB-Jet-Black-Console-/321459436277?pt
          =LH_DefaultDomain_0&hash=item4ad879baf5",
        "product_title" : "Sony PlayStation 4 (PS4) (Latest Model)- 500 GB Jet Black Console",
        "product_description" : "The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and
          a deeply connected gaming community. Step into living, breathing worlds where you are hero of your epic journey. Explore gritty
          urban environments, vast galactic landscapes, and fantastic historical settings brought to life on an epic scale, without
          limits. With an astounding launch lineup and over 180 games in development the PS4 system offers more top-tier blockbusters and
          inventive indie hits than any other next-gen console. The PS4 system is developer inspired, gamer focused. The PS4 system
          learns how you play and intuitively curates the content you use most often. Fire it up, and your PS4 system points the way to
          new, amazing experiences you can jump into alone or with friends. Create your own legend using a sophisticated, intuitive
          network built for gamers. Broadcast your gameplay live and direct to the world, complete with your commentary. Or immortalize
          your most epic moments and share at the press of a button. Access the best in music, movies, sports and television. PS4 system
          doesn t require a membership fee to access your digital entertainment subscriptions. You get the full spectrum of entertainment
          that matters to you on the PS4 system. PlayStation 4: The Best Place to Play The PlayStation 4 system provides dynamic,
          connected gaming, powerful graphics and speed, intelligent personalization, deeply integrated social capabilities, and
          innovative second-screen features. Combining unparalleled content, immersive gaming experiences, all of your favorite digital
          entertainment apps, and PlayStation exclusives, the PS4 system focuses on the gamers.Gamer Focused, Developer InspiredThe PS4
          system focuses on the gamer, ensuring that the very best games and the most immersive experiences are possible on the platform
          .<br>Read more about the PS4 on ebay guides.</br>",
        "product_price" : 329.98,
        "product_price_with_sale" : 0.0,
        "rate" : 5.0,
        "on_sale" : false
      }
    }
  ]
}
```

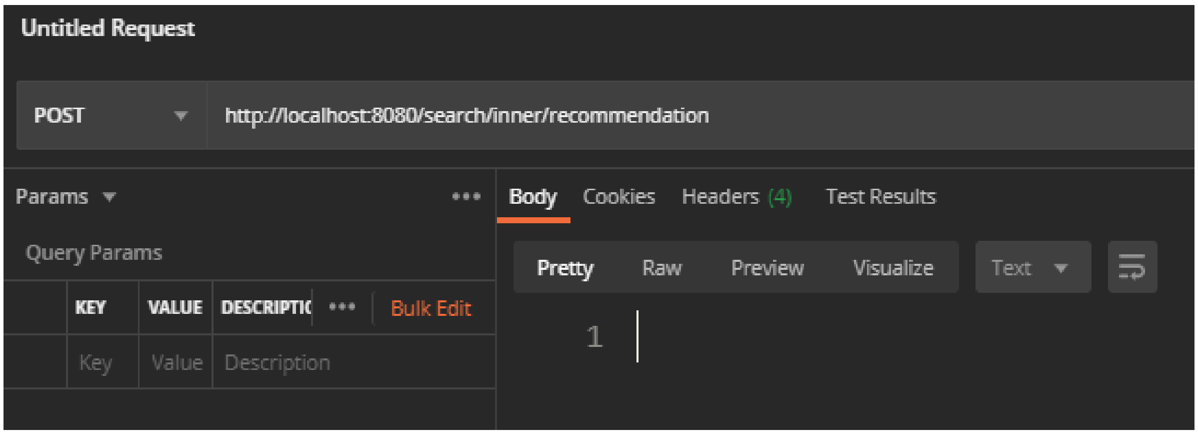
Figure 42: Product data before user reviews



```
  "max_score" : 8.890288,
  "hits" : [
    {
      "_index" : "ecommerce",
      "_type" : "_doc",
      "_id" : "bqv6AnkBi16XrzrSzhYG",
      "_score" : 8.890288,
      "_source" : {
        "unit_id" : "711158459",
        "query" : "playstation 4",
        "source" : "eBay",
        "source_url" : "http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw
          =playstation%204",
        "product_image" : "http://thumbs2.ebaystatic.com/d/l225/m/mzvzEUIknaQclZ801YCY1ew.jpg",
        "product_link" : "http://www.ebay.com/itm/Sony-PlayStation-4-PS4-Latest-Model-500-GB-Jet-Black-Console-/321459436277?pt
          =LH_DefaultDomain_0&hash=item4ad879baf5",
        "product_title" : "Sony PlayStation 4 (PS4) (Latest Model)- 500 GB Jet Black Console",
        "product_description" : "The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and
          a deeply connected gaming community. Step into living, breathing worlds where you are hero of your epic journey. Explore gritty
          urban environments, vast galactic landscapes, and fantastic historical settings brought to life on an epic scale, without
          limits. With an astounding launch lineup and over 180 games in development the PS4 system offers more top-tier blockbusters and
          inventive indie hits than any other next-gen console. The PS4 system is developer inspired, gamer focused. The PS4 system
          learns how you play and intuitively curates the content you use most often. Fire it up, and your PS4 system points the way to
          new, amazing experiences you can jump into alone or with friends. Create your own legend using a sophisticated, intuitive
          network built for gamers. Broadcast your gameplay live and direct to the world, complete with your commentary. Or immortalize
          your most epic moments and share at the press of a button. Access the best in music, movies, sports and television. PS4 system
          doesn t require a membership fee to access your digital entertainment subscriptions. You get the full spectrum of entertainment
          that matters to you on the PS4 system. PlayStation 4: The Best Place to Play The PlayStation 4 system provides dynamic,
          connected gaming, powerful graphics and speed, intelligent personalization, deeply integrated social capabilities, and
          innovative second-screen features. Combining unparalleled content, immersive gaming experiences, all of your favorite digital
          entertainment apps, and PlayStation exclusives, the PS4 system focuses on the gamers.Gamer Focused, Developer InspiredThe PS4
          system focuses on the gamer, ensuring that the very best games and the most immersive experiences are possible on the platform
          .<br>Read more about the PS4 on ebay guides.</br>",
        "product_price" : 329.98,
        "product_price_with_sale" : 0.0,
        "rate" : 3.6667,
        "on_sale" : false
      }
    }
  ]
 }
}
```

Figure 43: Product data after user reviews

In Table 6, some cells contain?. It means that these buyers were not able to purchase the specified product. Therefore, it can be a potentially recommended product for the user if its rating is higher than 4. Let's start the recommendation algorithm and review the results of the recommendations.


Untitled Request

POST http://localhost:8080/search/inner/recommendation

Params | Query Params

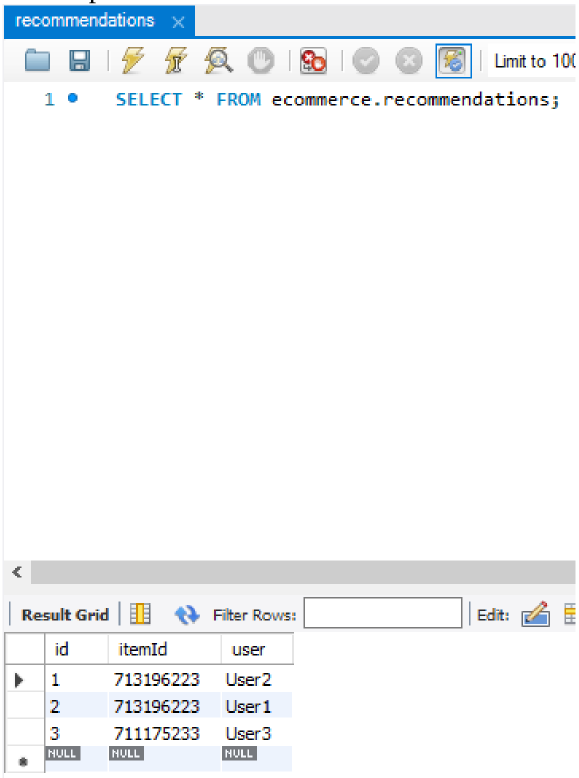
KEY	VALUE	DESCRIPTIC
Key	Value	Description

Body | Cookies | Headers (4) | Test Results

Pretty | Raw | Preview | Visualize | Text

Figure 44: The starting point for launching recommendations

Fig. 45 shows after executing the recommendation request, new recommendations about recommendations appeared in the recommendations table, namely: for users User2 and User1 to recommend a product with ID 713196223, and for user User3 to recommend a product with ID 711175233. In Table 7 green highlighted products will be recommended to users. Looking at User3's ratings, one can see a trend of lower ratings compared to User1 and User2. Therefore, it is logical to assume that if user User3 gives a rating of 3 to a particular product, users User1 and User2 may have a 4 or even 5. It is the case with the product tv.


recommendations

```
SELECT * FROM ecommerce.recommendations;
```

Result Grid | Filter Rows: | Edit:

id	itemId	user
1	713196223	User2
2	713196223	User1
3	711175233	User3
NULL	NULL	NULL

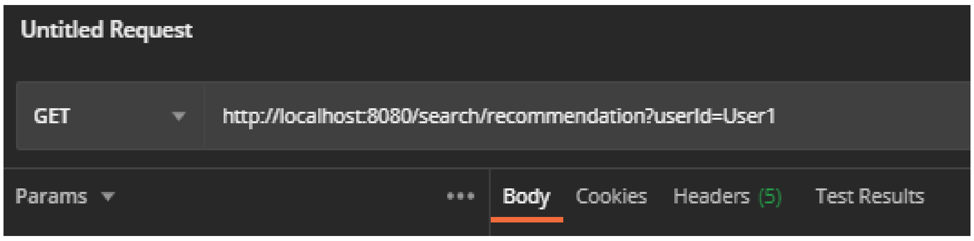
Figure 45: Records of recommendations in databases

Table 7

Recommendations for users

User	PS 4	routers	High chair	Wireless mouse	Spiderman	Iphone5	PS 2	Blender	Batman	TV
1	5	4	3	4	3	?	?	?	?	+
2	4	4	?	4	?	3	5	4	?	+
3	2	3	?	?	?	5	+	?	4	3

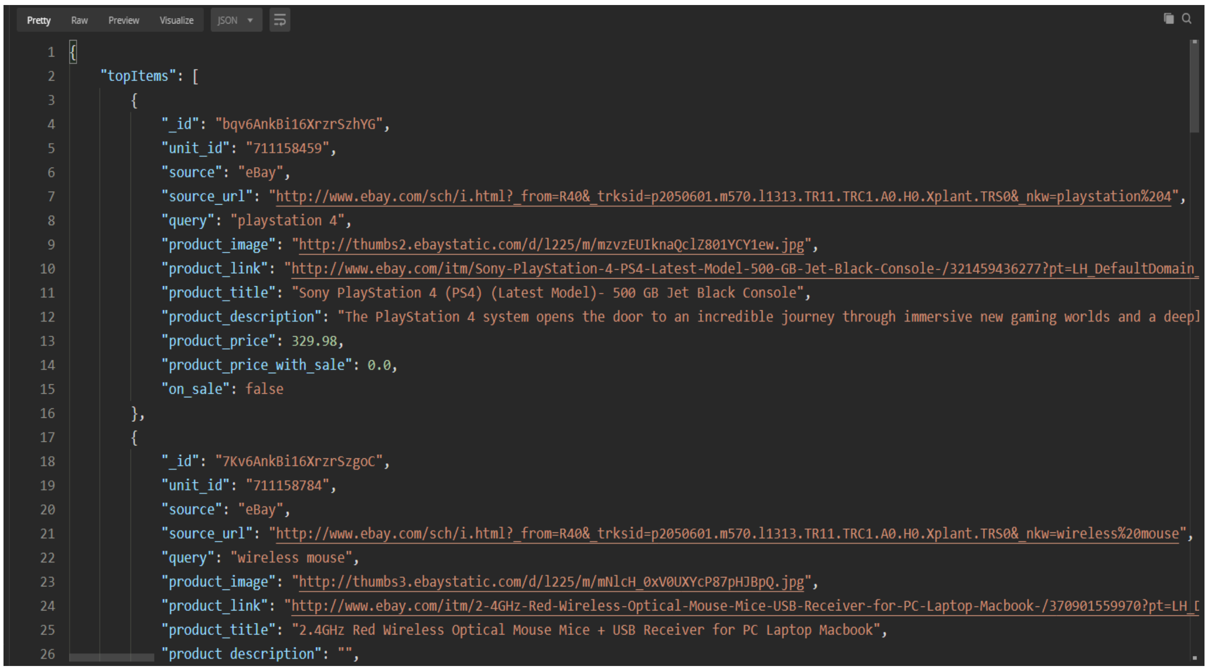
Execute the following query for user User1.



```
GET    http://localhost:8080/search/recommendation?userId=User1
```

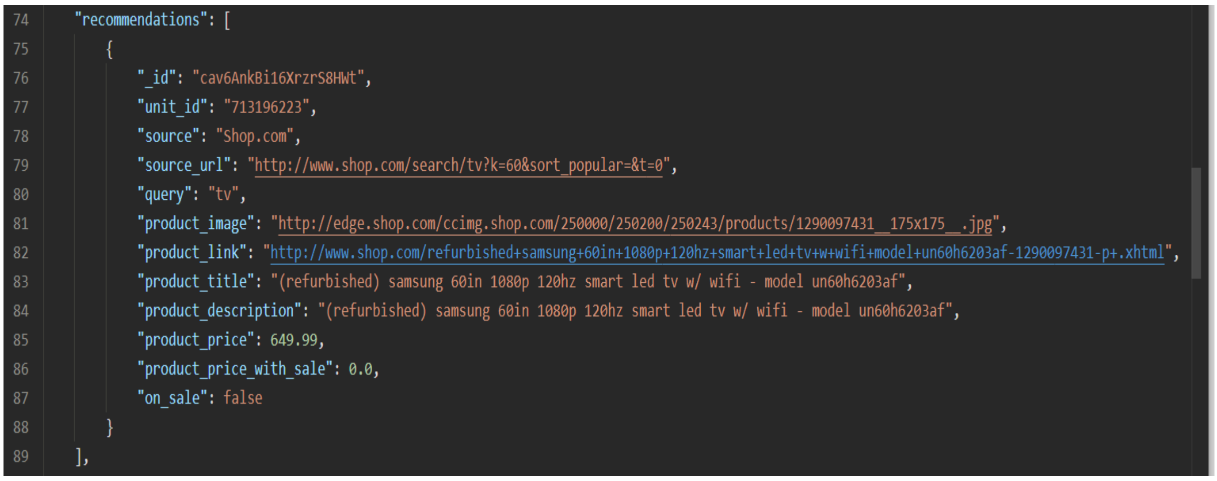
Figure 46: Request for recommendations for user User1

This query will return us the Top 5 products by several purchases among all users, products recommended for the specified user, and several filters for ease of search. In Fig. 47 - Fig. 49, one can see the result of this query. Fig. 47 shows the most popular products are PlayStation 4 and a wireless mouse. Goods purchased by all users according to Table 10. Fig. 48 shows the recommended product generated by the system in Fig. 45.



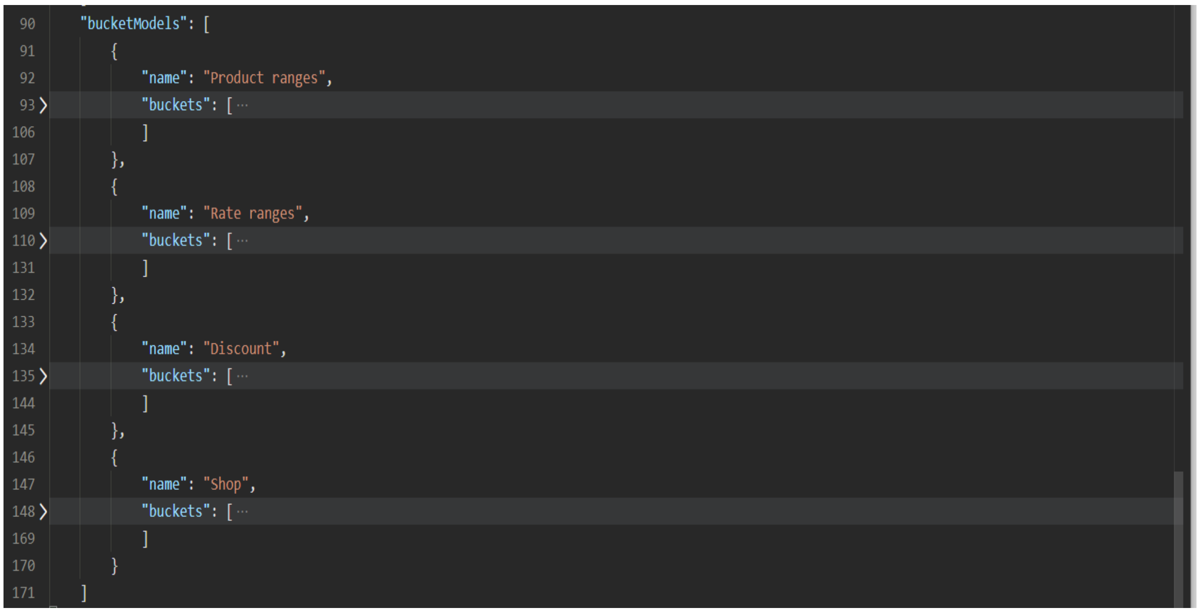
```
{
    "topItems": [
        {
            "_id": "bqv6AnkBi16XrzrSzhYG",
            "unit_id": "711158459",
            "source": "eBay",
            "source_url": "http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=playstation%204",
            "query": "playstation 4",
            "product_image": "http://thumbs2.ebaystatic.com/d/l225/m/mzvzEUIknaQclZ801YCY1ew.jpg",
            "product_link": "http://www.ebay.com/itm/Sony-PlayStation-4-PS4-Latest-Model-500-GB-Jet-Black-Console-/321459436277?pt=LH_DefaultDomain_
            "product_title": "Sony PlayStation 4 (PS4) (Latest Model)- 500 GB Jet Black Console",
            "product_description": "The PlayStation 4 system opens the door to an incredible journey through immersive new gaming worlds and a deepl
            "product_price": 329.98,
            "product_price_with_sale": 0.0,
            "on_sale": false
        },
        {
            "_id": "7Kv6AnkBi16XrzrSzgoC",
            "unit_id": "711158784",
            "source": "eBay",
            "source_url": "http://www.ebay.com/sch/i.html?_from=R40&_trksid=p2050601.m570.l1313.TR11.TRC1.A0.H0.Xplant.TRS0&_nkw=wireless%20mouse",
            "query": "wireless mouse",
            "product_image": "http://thumbs3.ebaystatic.com/d/l225/m/mNlcH_0xV0UXYcP87pHJBpQ.jpg",
            "product_link": "http://www.ebay.com/itm/2-4GHz-Red-Wireless-Optical-Mouse-Mice-USB-Receiver-for-PC-Laptop-Macbook-/370901559970?pt=LH_[
            "product_title": "2.4GHz Red Wireless Optical Mouse Mice + USB Receiver for PC Laptop Macbook",
            "product_description": "",
```

Figure 47: Top 5 products among users



```
    "recommendations": [
        {
            "_id": "cav6AnkBi16XrzrS8HWt",
            "unit_id": "713196223",
            "source": "Shop.com",
            "source_url": "http://www.shop.com/search/tv?k=60&sort_popular=&t=0",
            "query": "tv",
            "product_image": "http://edge.shop.com/ccimg.shop.com/250000/250200/250243/products/1290097431__175x175__.jpg",
            "product_link": "http://www.shop.com/refurbished+samsung+60in+1080p+120hz+smart+led+tv+w+wifi+model+un60h6203af-1290097431-p+.xhtml",
            "product_title": "(refurbished) samsung 60in 1080p 120hz smart led tv w/ wifi - model un60h6203af",
            "product_description": "(refurbished) samsung 60in 1080p 120hz smart led tv w/ wifi - model un60h6203af",
            "product_price": 649.99,
            "product_price_with_sale": 0.0,
            "on_sale": false
        }
    ],
```

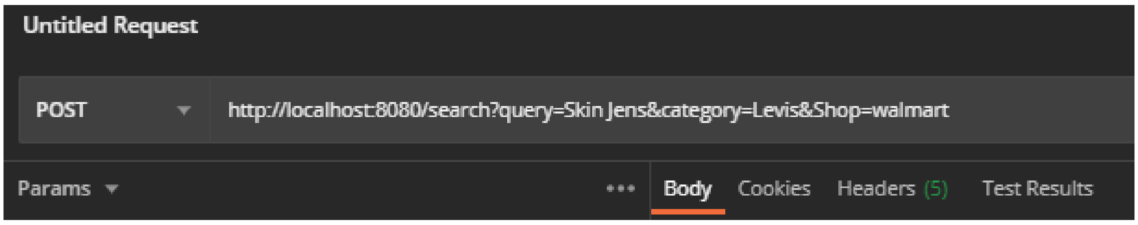
Figure 48: Recommended products for user User1



```
"bucketModels": [
    {
        "name": "Product ranges",
        "buckets": [···
        ]
    },
    {
        "name": "Rate ranges",
        "buckets": [···
        ]
    },
    {
        "name": "Discount",
        "buckets": [···
        ]
    },
    {
        "name": "Shop",
        "buckets": [···
        ]
    }
]
```

Figure 49: Filters

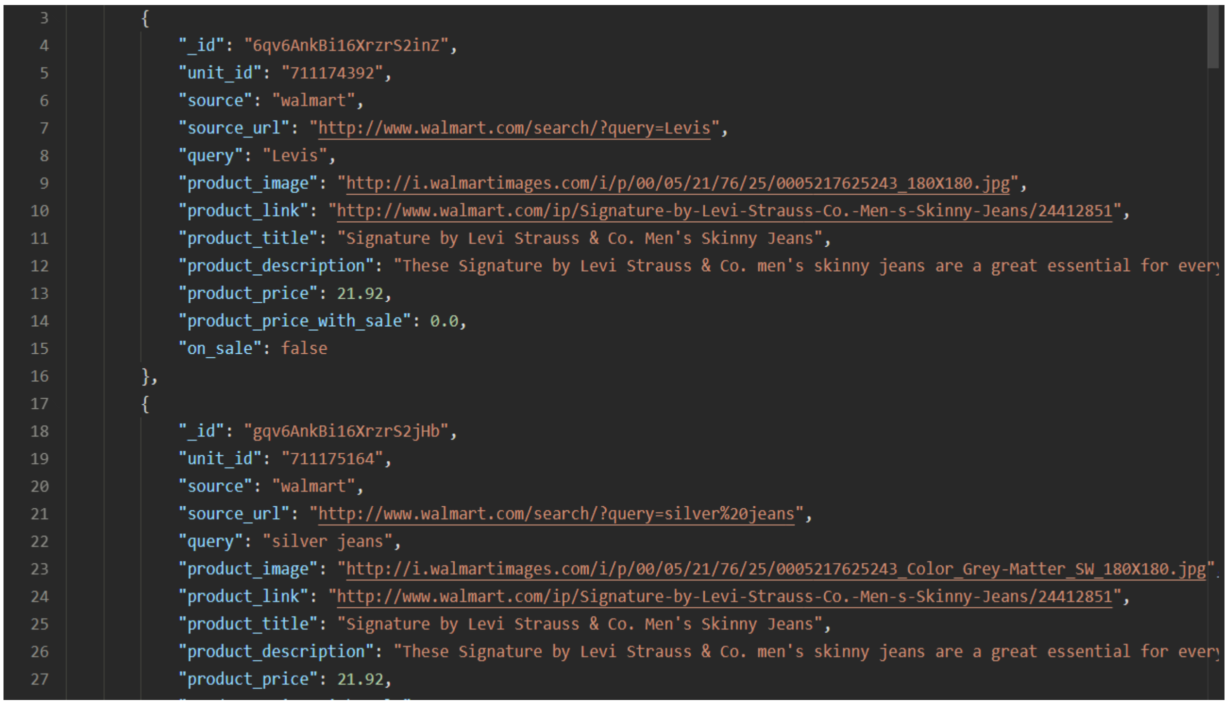
Let's try to use one of the filters. To do this, perform the query in Fig. 50. In this figure, we use the Shop filter, which has Walmart's value, i.e. all found products must be from the Walmart store. Fig. 51 shows the Walmart store publishes all the results.



Untitled Request

POST http://localhost:8080/search?query=Skin Jens&category=Levis&Shop=walmart

Figure 50: Query using a filter



```
    {
        "_id": "6qv6AnkBi16XrzrS2inZ",
        "unit_id": "711174392",
        "source": "walmart",
        "source_url": "http://www.walmart.com/search/?query=Levis",
        "query": "Levis",
        "product_image": "http://i.walmartimages.com/i/p/00/05/21/76/25/0005217625243_180X180.jpg",
        "product_link": "http://www.walmart.com/ip/Signature-by-Levi-Strauss-Co.-Men-s-Skinny-Jeans/24412851",
        "product_title": "Signature by Levi Strauss & Co. Men's Skinny Jeans",
        "product_description": "These Signature by Levi Strauss & Co. men's skinny jeans are a great essential for every
        "product_price": 21.92,
        "product_price_with_sale": 0.0,
        "on_sale": false
    },
    {
        "_id": "gqv6AnkBi16XrzrS2jHb",
        "unit_id": "711175164",
        "source": "walmart",
        "source_url": "http://www.walmart.com/search/?query=silver%20jeans",
        "query": "silver jeans",
        "product_image": "http://i.walmartimages.com/i/p/00/05/21/76/25/0005217625243_Color_Grey-Matter_SW_180X180.jpg".
        "product_link": "http://www.walmart.com/ip/Signature-by-Levi-Strauss-Co.-Men-s-Skinny-Jeans/24412851",
        "product_title": "Signature by Levi Strauss & Co. Men's Skinny Jeans",
        "product_description": "These Signature by Levi Strauss & Co. men's skinny jeans are a great essential for every
        "product_price": 21.92,
```

Figure 51: The result of the query with the filter

Also, a number of search queries with and without the NLP algorithm were performed. Fig. 52 shows the number of matches when searching with the NLP algorithm and without it.

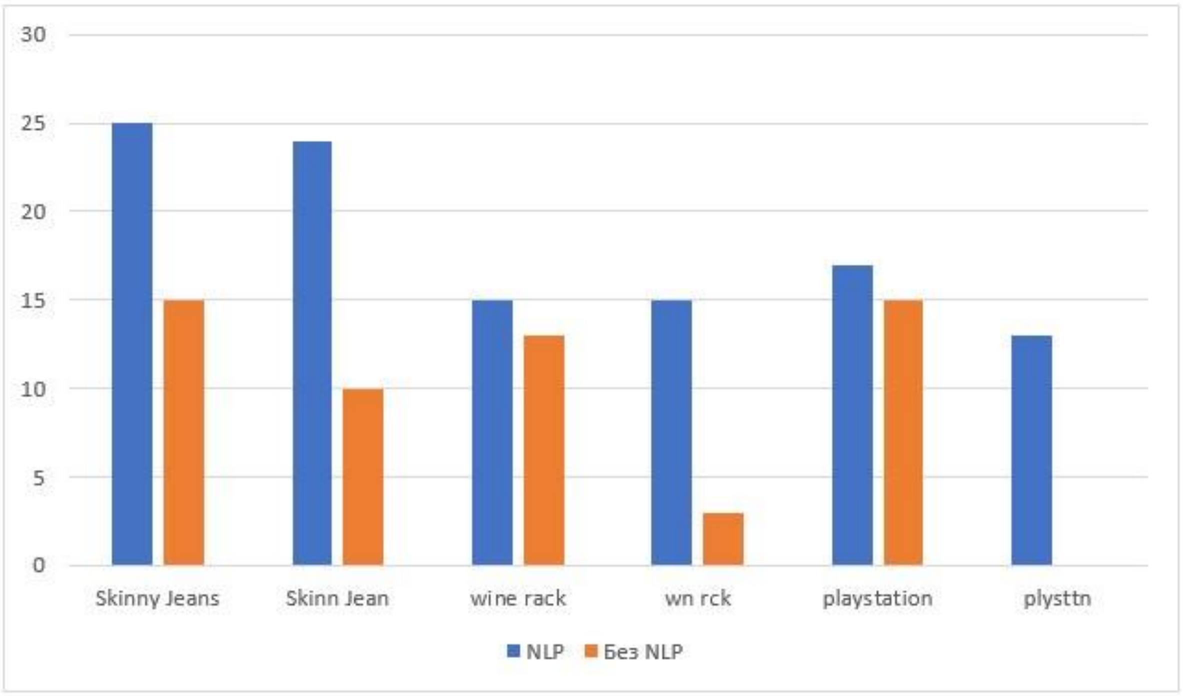


Figure 52: Comparison of search results with and without NLP-algorithm

The query execution speed was also compared with existing systems. This way, the amount of data in the repository may vary. That is why 60-70ms may be an error when comparing.

Fig. 53 shows that a query consisting of 1 or 2 words will be found much faster than analogues. But for 3 or more, the results are about the same.

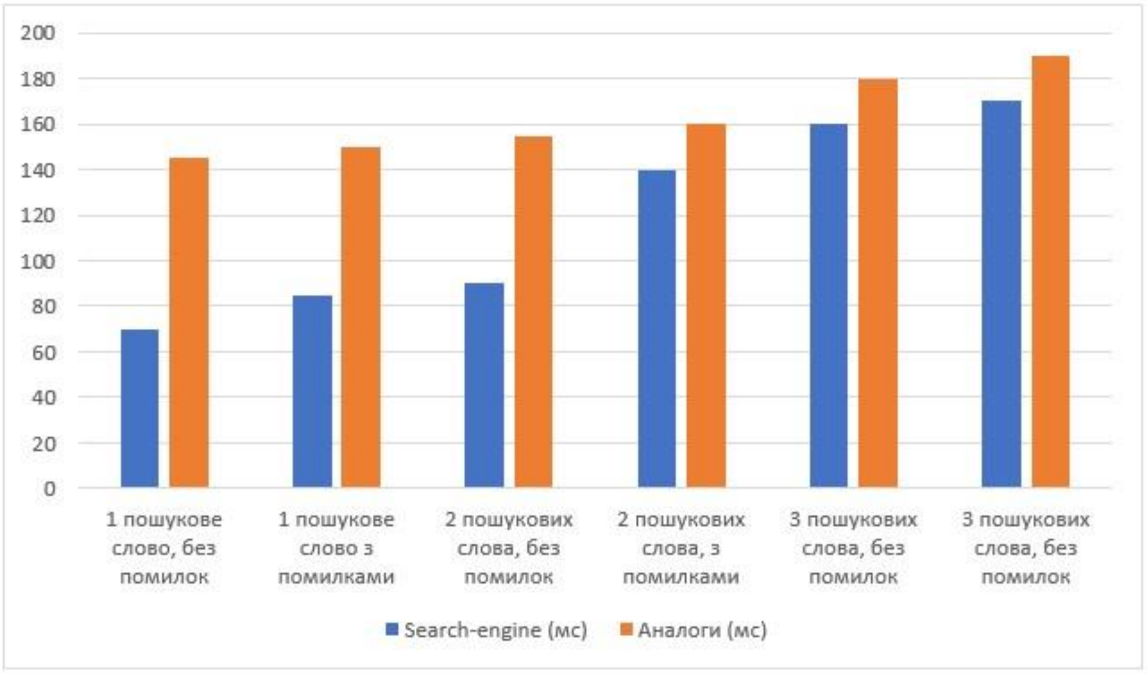


Figure 53: Comparison of query execution speed with analogues

The created software product was described in this section. The structure of the database, the mechanisms of inference, the structure and functions of the software used were considered. The software was described in accordance with the standard GOST 19.402-78 "Description of the program" or ISO / IEC 26514: 2008 "Systems and software engineering". Also, a user manual was added to help use the developed information system. After that, the control example was analysed. The control example confirmed the efficiency of the information system and the compliance of the results of the system with the task.

6. Conclusion

In this work, an intelligent product search system in e-commerce with the possibility of recommendations for the user was developed. It required several studies. It was determined exactly how to store and analyse product information. Different types of storage, their advantages and disadvantages were taken into account. It was shown how the repository could index the data. It was proved that it is necessary to use the recommended algorithm of collaborative filtering for this system because collaborative filtering takes into account the ratings and ratings of users, and there is no attachment to the subject of the service. Thus, the plan was able to show the recommendations compared to existing services more accurately. The main fuzzy search algorithms were compared, and their accuracy is also demonstrated in a table. Based on these studies, Levenstein's algorithm was chosen because it corresponds to words, taking into account the number of deletions, insertions, replacement of letters in the word. A tree of goals has been created in which the main goals and points that need to be taken into account when developing this system have been described. Also, three-level IDEF0 and DFD diagrams were created, showing the interaction of processes within the system.

Technical means and technologies were chosen to solve the above goals and justified the feasibility of their use. An intelligent search system has been developed to auto-complete search words, use filters for search speed, sort results, and use NLP algorithms. In essence, an efficient, profitable and stable search engine has been built for almost any type of business, which contains many information or goods and is in demand among users. Also, a control example was shown, which confirms the functionality of the program and the main functionality discussed above.

7. References

[1] S. Albota, Requirements for the Linguistic Quality Control of Wikipedia Article, in: Proceedings of the 14th. International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2019, pp. 16-19, doi: 10.1109/STC-CSIT.2019.8929771.

[2] S. Albota, Linguistic and Psychological Features of the Reddit News Post, in: Proceedings of the 15th. International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020, pp. 295-299, doi: 10.1109/CSIT49958.2020.9321991.

[3] The Search Backend, 2021. URL: https://eileen-code4fun.medium.com/system-design-interview-mini-google-search-6fd319cd66ca

[4] Traditional Database (Forward Indexes) vs Search Engines (Inverted Index), 2021. URL: https://dev.to/im_bhatman/introductionto-inverted-indexes-l04

[5] A. Franz, Th. Brants. All Our N-gram are Belong to You, 2006. URL: https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html

[6] D. Guthrie. A Closer Look at Skip-gram. NLP Research group, Computer Science, 2017. URL:https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.598&rep=rep1&type=pdf

[7] V. I. Levenstein, Binary codes with correction of drops, insertions and replacements, USSR, 1986.

[8] N. Gonzalo, A guided tour to approximate string matching, Department of Computer Science, University of Chile, 2006. URL: https://www.dcc.uchile.cl/TR/1999/TR_DCC-1999-005.pdf

[9] Definition, Example, 2021. URL: https://www.cuel.ogic.com/blog/the-levenshtein-algorithmhttps://www.cuel.ogic.com/blog/the-levenshtein-algorithm

[10] The Levenshtein distance, 2021. URL: https://habr.com/ru/post/114997/

[11] M.A. Ghazanfar, M.A. Prügel-Bennett, S. Szedmák, Kernel-Mapping Recommender system algorithms, Inf. Sci., 208 (2012). doi.org/10.1016/j.ins.2012.04.012

[12] F. Ricci, L. Rokach, B. Shapira, Introduction to Recommender Systems Handbook. In: Recommender Systems Handbook. Springer, Boston, MA (2011).

[13] S. Zare, Automated Detection and Exposure of Behavor-Based Relationships between Browsable Itemis (2017).
[14] L. Terveen, W. Hill, Beyond Recommender Systems: Helping People Help Each Other (2001).
[15] Collaborative Filtering Recommender System, 2021. URL: https://medium.com/@chaitanyarb619/recommendationsystems-a-walk-trough-33587fecc195
[16] Introducing intelligence in relevance search, 2021. URL: https://powerapps.microsoft.com/ru-ru/blog/intelligence-in-relevancesearch/
[17] The IDEF Family of Methods, 2021. URL: http://www.sba.oakland.edu/faculty/mathieson/mis524/resources/readings/idef/idef.ht ml
[18] How search algorithms work, 2021. URL: https://www.google.com/intl/uk/search/howsearchworks/algorithms/
[19] What is DFD (data flow charts), 2021. URL: https://habr.com/ru/company/trinion/blog/340064
[20] What is an Elasticsearch index? 2021. URL: https://www.elastic.co/what-is/elasticsearch
[21] 5 reasons to choose MySQL. URL: https://dataconomy.com/2017/04/5-reasons-challenges-mysql/
[22] Java, 2021. URL: https://techterms.com/definition/java
[23] Answer, 2021. URL: https://stackoverflow.com/questions/209555/why-would-you-choose-the-javaprogramming-language-over-others
[24] The Slope One Algorithm, 2021. URL: https://www.baeldung.com/java-collaborative-filtering-recommendations
[25] Sh.Yue, M. Larson, A. Hanjalic, Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges, ACM Comput. Surv. 47, 1, Article 3 (2014). doi.org/10.1145/2556270
[26] A. Figueroa, J. Atkinson, Contextual Language Models for Ranking Answers to Natural Language Definition Questions, Computational Intelligence, 28 (2012).
[27] New features in the language Java 8, 2021. URL: https://javarush.ru/groups/posts/1037-osobennosti-java-8--maksimaljhnoerukovodstvo-chastjh-1
[28] Liquibase allows you to perform the following, 2021. URL: https://medium.com/podiihq/getting-started-with-liquibase8965897092aa
[29] Answer, 2021. URL: https://stackoverflow.com/questions/18193253/what-exactly-is-an-n-gram
[30] V. Lytvyn, V. Vysotska, V. Shatskykh, I. Kohut, O. Petruchenko, L. Dzyubyk, V. Bobrivetc, V. Panasyuk, S. Sachenko, M. Komar, Design of a recommendation system based on Collaborative Filtering and machine learning considering personal needs of the user, volume 4(2-100) of Eastern-European Journal of Enterprise Technologies, 2019, pp. 6-28.
[31] V. Husak, O. Lozynska, I. Karpov, I. Peleshchak, S. Chyrun, A. Vysotskyi, Information System for Recommendation List Formation of Clothes Style Image Selection According to User’s Needs Based on NLP and Chatbots, volume 2604 of CEUR workshop proceedings, 2020, pp. 788-818.
[32] O. Artemenko, V. Pasichnyk, N. Kunanets, K. Shunevych, Using sentiment text analysis of user reviews in social media for e-tourism mobile recommender systems, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 259-271.
[33] S. Makara, L. Chyrun, Y. Burov, Z. Rybchak, I. Peleshchak, R., Peleshchak, R. Holoshchuk, S. Kubinska, A. Dmytriv, An Intelligent System for Generating End-User Symptom Recommendations Based on Machine Learning Technology, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 844-883.
[34] N. Shakhovska, K. Shakhovska, S. Fedushko, Some Aspects of the Method for Tourist Route Creation, volume 902 of Advances in Artificial Systems for Medicine and Education II, 2019, pp. 527-537.
[35] N. Shakhovska, S. Fedushko, M. Greguš, I. Shvorob, Y. Syerov, Development of Mobile System for Medical Recommendations, in: The 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), 2019, pp. 43-50.
[36] A. Chiche, Hybrid Decision Support System Framework for Crop Yield Prediction and Recommendation, volume 18(2) of International Journal of Computing, 2019, pp. 181-190.
[37] N. Antonyuk, M. Medykovskyy, L. Chyrun, M. Dverii, O. Oborska, M. Krylyshyn, A. Vysotsky, N. Tsiura, O. Naum, Online Tourism System Development for Searching and Planning Trips with User’s Requirements, volume 1080 of Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG, 2020, pp. 831-863.

[38] O. Pavlenko, I. Tymofieieva, Search Query Data Analysis: Challenges and Opportunities, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 452-461.
[39] O. Kliuiev, N. Vnukova, S. Hlibko, N. Brynza, D. Davydenko, Estimation of the Level of Interest and Modeling of the Topic of Innovation Through Search in Google, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 523-535.
[40] P. Radiuk, N. Hrypynska, A Framework for Exploring and Modelling Neural Architecture Search Methods, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 1060-1074.
[41] O. Cherednichenko, M. Vovk, O. Kanishcheva, M. Godlevskyi, Towards Improving the Search Quality on the Trading Platforms, in: 11th SIGSAND/PLAIS, LNBIP 333, 2018, pp. 21-30.
[42] O. Veres, B. Rusyn, A. Sachenko, I. Rishnyak, Choosing the Method of Finding Similar Images in the Reverse Search System, volume Vol-2136 of CEUR Workshop Proceedings, 2018, 99-107.
[43] T. Basyuk, A. Vasyliuk, V. Lytvyn, Mathematical Model of Semantic Search and Search Optimization, volume Vol-2362 of CEUR Workshop Proceedings, 2019, pp. 96-105.
[44] A. Adamuthe, T. Nitave, Adaptive Harmony Search for Optimizing Constrained Resource Allocation Problem, volume 17(4) of International Journal of Computing, 2018, pp. 260-269.
[45] N. Vasylkiv, L. Dubchak, A. Sachenko, Estimation Method of Information System Functioning Quality Based on the Fuzzy Logic, volume 2631 of CEUR Workshop Proceedings, 2020, 40-56.
[46] A. Bakurova, M. Pasichnyk, E. Tereschenko, Y. Filei, Formalization of Ukrainian-Language Content for Fuzzy Product in Court, vol. 2604 of CEUR workshop proceedings, 2020, 428-441.
[47] M. Bublyk, O. Rybytska, A. Karpiak, Y. Matseliukh, Structuring the fuzzy knowledge base of the IT industry impact factors, in: Computer sciences and information technologies (CSIT), 2018.
[48] A. Gozhyj, I. Kalinina, V. Gozhyj, Fuzzy cognitive analysis and modeling of water quality, in: International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2017, pp. 289-294.
[49] S. Babichev, V. Lytvynenko, A. Gozhyj, at., A fuzzy model for gene expression profiles reducing based on the complex use of statistical criteria and Shannon entropy, volume 754 of Advances in Intelligent Systems and Computing, 2018, pp. 545-554.
[50] S. Sachenko, T. Lendyuk, S. Rippa, G. Sapojnyk, Fuzzy Rules for Tests Complexity Changing for Individual Learning Path Construction. Svitlana Sachenko, in: Int. Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2015, pp. 945-948.
[51] S. Sachenko, O. Chereshnyuk, V. Panasyuk, A. Banasik, I. Golyash, Fuzzy-multiple Approach in Choosing the Optimal Term for Implementing the Innovative Project, in: Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2017, pp. 533-537.
[52] P. Kravets, The control agent with fuzzy logic, in: Perspective Technologies and Methods in MEMS Design, MEMSTECH, 2010, pp. 40-41.
[53] P. Kravets, R. Kyrkalo, Fuzzy logic controller for embedded systems, in: International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH, 2009, pp. 58-59.
[54] M. Pasieka, N. Grzesik, K. Kuźma, Simulation Modeling of Fuzzy Logic Controller for Aircraft Engines, volume 16(1) of International Journal of Computing, 2017, pp. 27-33.
[55] I. Perova, Y. Bodyanskiy, Fast Medical Diagnostics Using Autoassociative Neuro-Fuzzy Memory, volume 16(1) of International Journal of Computing, 2017, pp. 34-40.
[56] O. Bisikalo, V. Vysotska, Y. Burov, P. Kravets, Conceptual Model of Process Formation for the Semantics of Sentence in Natural Language, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 151-177.
[57] O. Iosifova, I. Iosifov, O. Rolik, V. Sokolov, Techniques Comparison for Natural Language Processing, volume Vol-2631 of CEUR Workshop Proceedings, 2020, pp. 57-67.
[58] E. Fedorov, O. Nechyporenko, T. Utkina, Forecast Method for Natural Language Constructions Based on a Modified Gated Recursive Block, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 199-214.
[59] V. Lytvyn, S. Kubinska, A. Berko, T. Shestakevych, L. Demkiv, Y. Shcherbyna, Peculiarities of Generation of Semantics of Natural Language Speech by Helping Unlimited and Context-Dependent Grammar, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 536-551.
[60] O. Bisikalo, Y. Ivanov, V. Sholota, Modeling the Phenomenological Concepts for Figurative Processing of Natural-Language Constructions, volume Vol-2362 of CEUR Workshop Proceedings, 2019, pp. 1-11.