RESULTS OF DEVELOPING THE RECOMMENDATION SYSTEM FOR ELECTRONIC EDUCATIONAL RESOURCE SELECTION

Valentina Yunchyk
Department of General Mathematics and Computer Science Teaching Methods
Lesya Ukrainka Volyn National University
Lutsk, Ukraine
uynchik@gmail.com,

Yurii Fedoniuk
Department of Computer Science and Cybersecurity
Lesya Ukrainka Volyn National University
Lutsk, Ukraine
yura.fedonyuk@gmail.com

Abstract

This paper proposes developing a recommendation system based on fuzzy logic methods for expert evaluation of electronic educational resources (EERs) and decision-making regarding selecting the most effective resources for educational processes. The concepts of recommendation systems for selecting optimal EERs are examined and analyzed. Scientific publications on expert evaluation and recommendation system utilization are reviewed. The overall structure of the recommendation system is presented, along with descriptions of its subsystems. Fuzzy logic methodologies are used for the assessment of EERs, with a well-defined procedural framework and explicit algorithmic representation. Expert analysis results in a compilation of recommended EER alternatives that align with specified criteria. The EER selection recommendation system is further elucidated through the generation of UML diagrams delineating use cases, sequences, and activities. The initial phases of user engagement with the recommendation system are described in depth, facilitating the selection of electronic learning resources. The recommendation system is introduced with a three-tier architecture consisting of presentation, application, and data administration layers. EER collection, expert criterion-based evaluation input, recommendation rating computation, recommended resource list formation, data visualization, authorization implementation, access provisioning, and creation of the administrator interface are the main technological elements that are described in detail. For each of these elements, implementation strategies and tools are explained. Programming code for the development of each of the key recommendation system stages is provided. The most significant elements of the web application interface are demonstrated, including the recommendation system's criterion selection page, administrator panel, category modification and addition panel, and the recommendation system's output.

Keywords: recommendation system; electronic educational resources; fuzzy logic; expert evaluation; UML diagrams; recommendation system architecture

I. Introduction

Recent developments have accelerated the acceptance of electronic learning and its growing significance, resulting in the development of electronic educational resources (EERs) and high-quality instructional content. In the evolution of EER, the caliber of content presentation, delivery strategies, and visual representation have become crucial factors.

To ensure a control system for the evaluation of electronic educational resources, it is important to form a list of requirements for them, select competent experts to carry out a qualitative evaluation of the relevant resources and conduct an expert quality analysis [1].

The process of evaluating electronic learning resources is complex. It includes analysis of functionality, availability, architecture, and other criteria. It is essential to develop a specific algorithm that contributes to the effective evaluation of electronic educational resources. An effective variant of such an algorithm is a recommender system.

Literature Analysis. Pioneering research in the realm of recommendation system research, development, and enhancement is conducted by both foreign and domestic scholars. Notable contributions include studies by A. Esteban et al. [2], which explore the effectiveness of a hybrid multi-criteria recommendation system for guiding students in course selection. A. Cañas et al. [3] delve into recommendation systems for informal education, grounded in a semantic approach. A comprehensive overview of recommendation systems is provided by A. Choi et al. [4], presenting recommendation models, methods, and applications. Investigations by B. Ojokoh et al. [5] focus on recommendation systems based on fuzzy logic. V. Pasichnyk et al. [6] delve into recommendation systems within electronic tourism. In the realm of educational services, recommendation systems are utilized to tailor educational resources, factoring in learning styles and knowledge levels, thus enhancing the efficacy of the educational process [7]. In other words, the participants in the educational process can be provided with personalized educational content [8]. J. Shu et al. [9] demonstrate the use of a recommendation system employing neural networks to comprehend textual data within educational resources, offering personalized learning materials aligned with user preferences. Personalized recommendation algorithms for online educational resources are introduced by [10], grounded in knowledge association. Furthermore, [11] presents a recommendation system founded on Bayesian networks to present digital educational resources. A recommendation web service aiding individualized learning pathways in transportation systems programming is detailed by [12]. A. Morales et al. [13] explore a graph-oriented database-based recommendation system for an open educational resources repository. P. Dwivedi et al. [14] showcase a recommendation system structuring user groups, reflecting their individual preferences, and delivering tailored educational content. J. K. Tarus et al. [15] propose a recommendation system for online learning resource selection, built upon collaborative filtering based on knowledge models.

In our study, recommender systems are a class of intelligent systems that generate recommendations based on fuzzy logic methods by forming a ranked list of electronic educational resources that meet requirements and certain criteria [16].

The purpose of the article is to present the results of modeling and development of a recommendation system for expert evaluation of electronic educational resources, which allows you to choose the most effective electronic resources for use in the educational process based on the generated recommendations.

II. Research results

The purpose of a recommendation system for evaluating electronic learning resources (EER) is to provide users with personalized recommendations for choosing the best EER in specific educational situations. The system is aimed at providing users with high-quality and relevant resources that meet their needs and increase the effectiveness of learning [17].

The scope of the recommendation system is the educational process. The recommendation system is intended for expert environments of higher education, educational program support groups, pedagogical teams of schools or individual departments, scientific and methodical commissions of specialties and faculties, scientific and methodical councils of institutes and universities, in general, for all expert communities that need to make decisions about the choice of electronic educational resources.

The functional requirements of the recommender system are depicted in Figure 1 in the form of a UML diagram of use cases and the main actors in the process of working with the recommender system are highlighted.

In the recommender system, there are two actors "Expert" and "Administrator". "Expert" is a person who has authorization in the system and full access to all functions and capabilities.

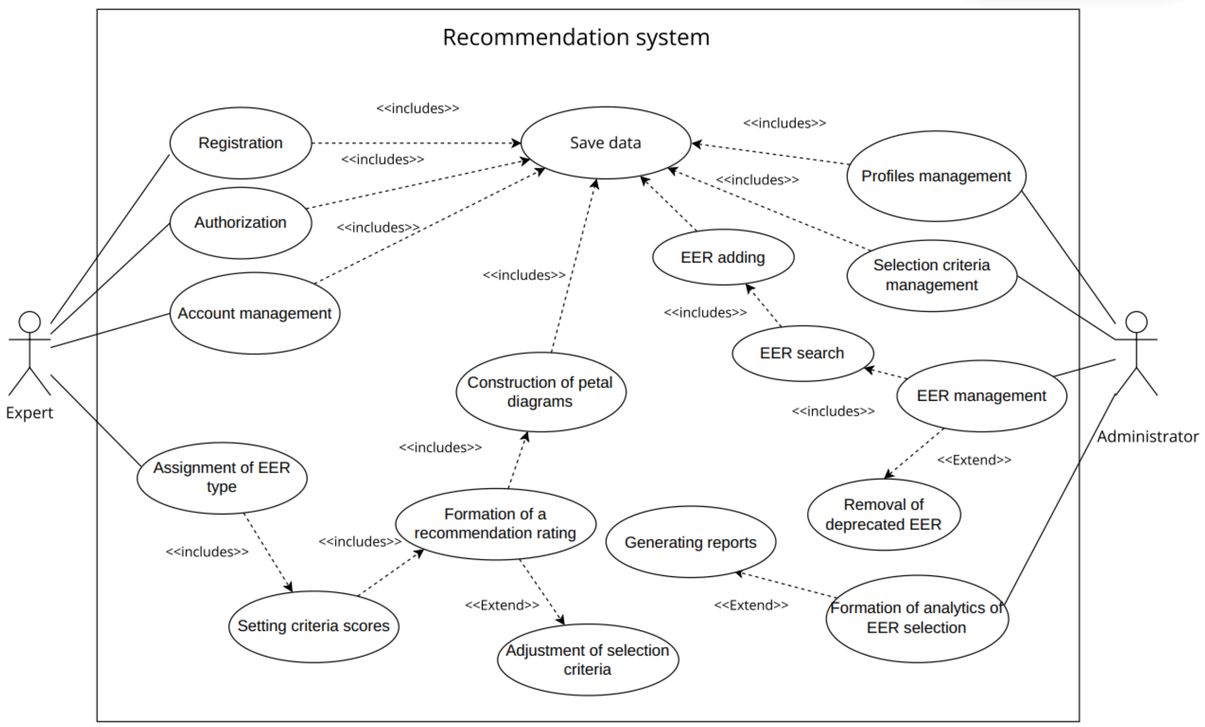


Fig. 1. Diagram of use cases for a recommendation system for selecting electronic educational resources

A registered "Expert" can create their profile, which will serve as the basis for initial recommendations. The "Expert" can generate a request for selecting electronic educational resources by specifying the type and rating for each criterion, and then receive a list of recommended alternatives. Additionally, the "Expert" can modify their profile parameters, effectively managing their account.

The "Administrator," on the other hand, is a user responsible for configuring the recommendation system, user management, and receiving analytics related to user interactions with the system. The set of use cases includes: "Account Management," "Selection Criteria Management," "Electronic Educational Resource Management," and "Formation of EER Selection Analytics."

Figure 2 shows the sequence diagram of the recommender system for selecting electronic educational resources. First, the user requests a recommendation according to the type and is given ratings for each ENR selection criterion. The controller checks the metadata and descriptions of the e-learning resources database, forms a matrix of pairwise comparisons, and transforms the data into a fuzzy matrix. Next, the system uses fuzzy logic to form a recommendation rating. Electronic learning resources are recommended to users if they have higher indicators according to the specified criteria, in particular, taking into account the importance of the criteria.

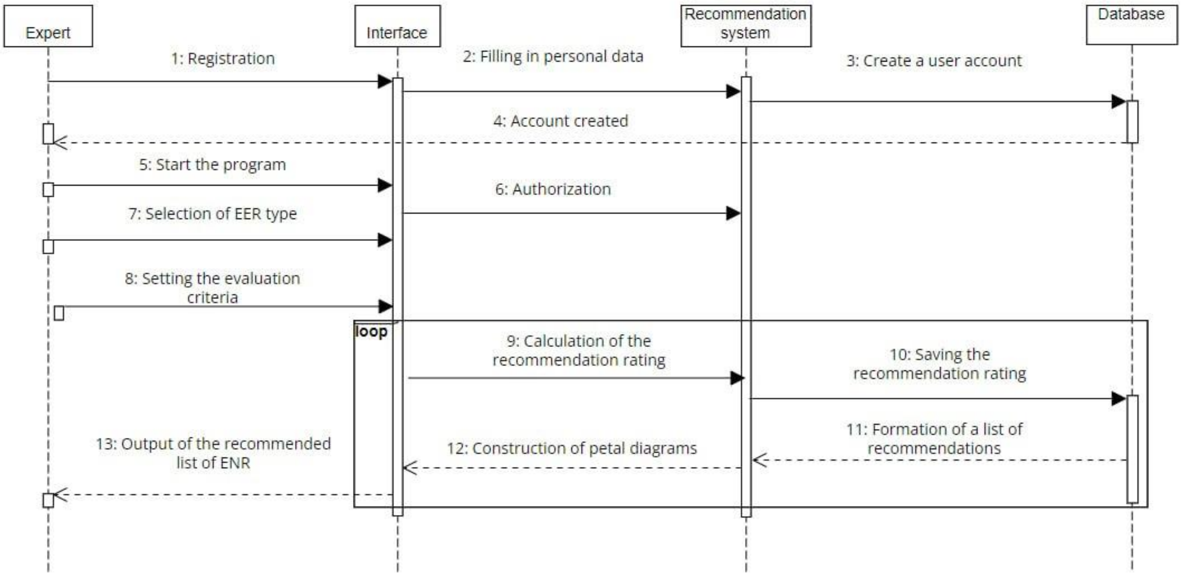

Expert
Interface
Recommendation system
Database
1: Registration
2: Filling in personal data
3: Create a user account
4: Account created
5: Start the program
6: Authorization
7: Selection of EER type
8: Setting the evaluation criteria
loop
9: Calculation of the recommendation rating
10: Saving the recommendation rating
11: Formation of a list of recommendations
12: Construction of petal diagrams
13: Output of the recommended list of ENR

Fig. 2. Sequence diagram of the recommendation system for selecting EERs

The recommendation system has been developed based on a three-tier architecture (Figure 3). This approach allows for breaking down the system into distinct interconnected components, distributing system functions, and separating the user interface from the data.

The three-tier architecture encompasses:

- *Presentation Tier*: This tier is where users perceive information.
- *Application Tier*: This tier houses the tools for managing the recommendation system, along with the following components: specifying the type of electronic educational resource, resource search, results display, and report presentation.
- *Data Management Tier*: This tier is where data is physically stored. It includes components such as the subsystem for determining the type of EER, the EER analysis subsystem, the results generation subsystem, and the user report generation subsystem.

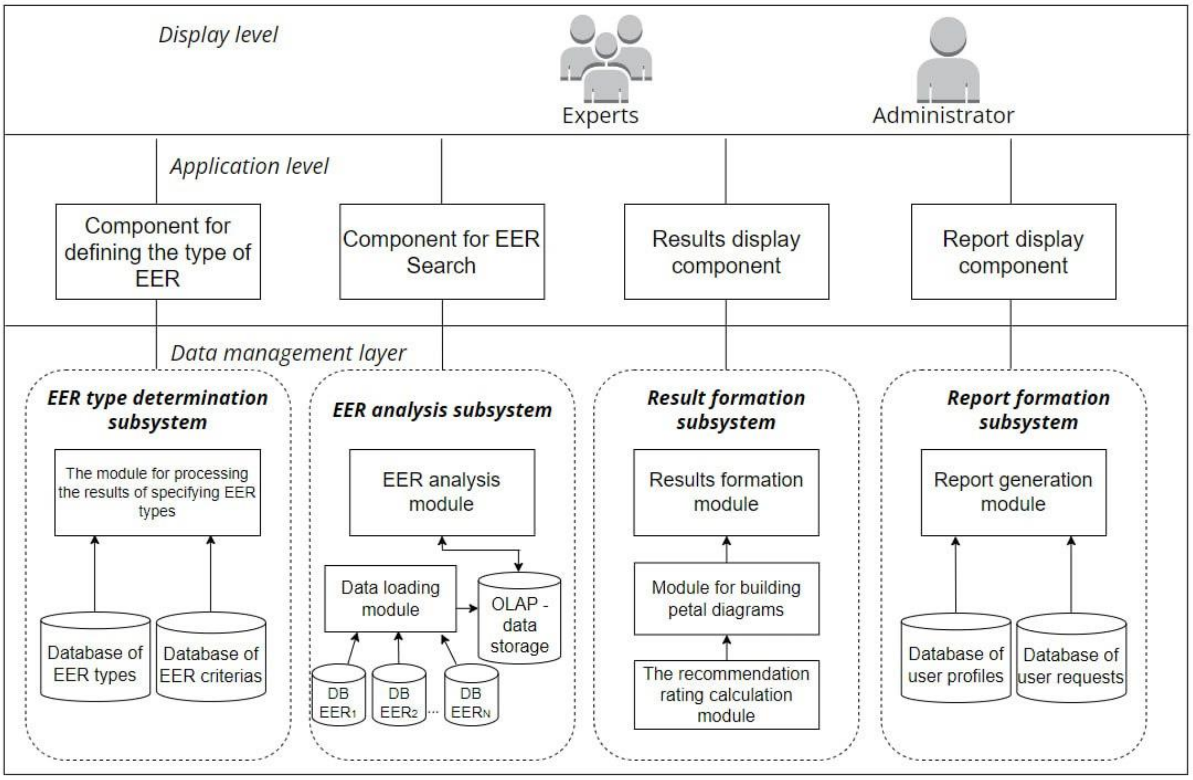

Display level
Experts
Administrator
Application level
Component for defining the type of EER
Component for EER Search
Results display component
Report display component
Data management layer
EER type determination subsystem
The module for processing the results of specifying EER types
Database of EER types
Database of EER criterias
EER analysis subsystem
EER analysis module
Data loading module
OLAP - data storage
DB EER1
DB EER2
DB EERN
Result formation subsystem
Results formation module
Module for building petal diagrams
The recommendation rating calculation module
Report formation subsystem
Report generation module
Database of user profiles
Database of user requests

Fig. 3. Structure of the recommendation system for selecting EERs

The subsystem for determining the type of ERR enables users to specify the EER type and select the criteria by which educational resources should be evaluated. This subsystem encompasses the processing module for handling the results of EER type specification, a database of EER types, and a database of EER criteria. The EER types database is designed to store a set of electronic educational resource types, while the EER criteria database stores a set of criteria used for evaluating educational resources. The results processing module handles the specified types and criteria.

The EER analysis subsystem comprises an OLAP repository, databases of electronic educational resources, a data loading module, and an EER analysis module. The OLAP repository holds the source data for analyzing EERs. The data structure is multidimensional and adapted for conducting OLAP analysis procedures. The primary function of the data loading module is to ensure current data in the repository, process queries to multidimensional data, and manage data storage. The EER analysis module processes the results of multidimensional data analysis concerning the functionality of electronic educational resources.

The results generation subsystem is responsible for creating the recommendation ranking of EERs and visualizing data. This subsystem includes a module for calculating the recommendation ranking, a module for constructing petal diagrams, and a module for generating results. The ranking calculation module processes data based on specified criteria using fuzzy logic methods. The petal diagrams construction module generates charts for each selected EER based on criterion values. The results generation module presents the recommended list of EERs from most to least recommended.

The report generation subsystem is intended for generating reports on EER query analytics. It consists of a database of user profiles, a database of user queries, and a report generation module.

Figure 4 depicts the activity diagram of the recommendation system for selecting electronic educational resources.

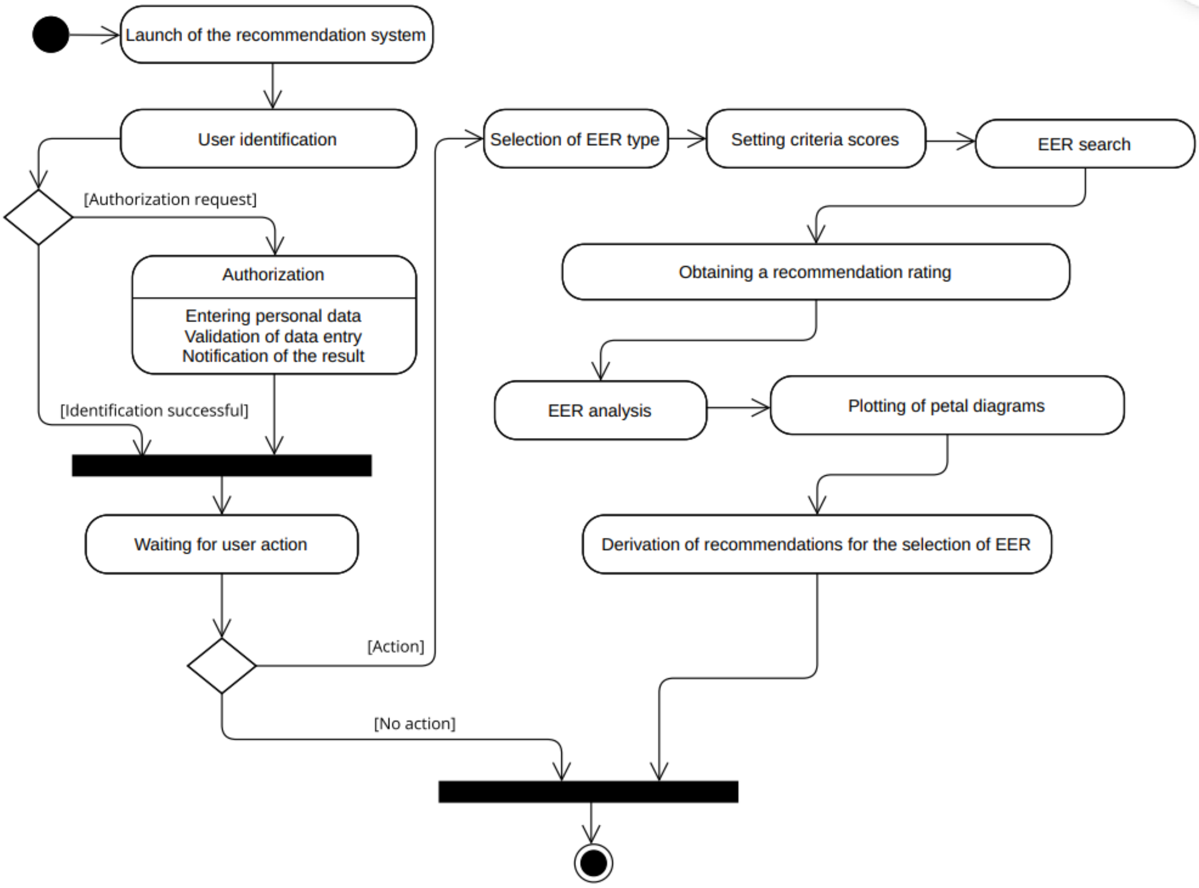


Fig. 4. Activity diagram of the recommendation system

The information technology of the functioning of the recommendation system for evaluating electronic educational resources includes the use of various technologies, algorithms, and methods for collecting, processing, and providing personalized recommendations to users.

The main idea of the recommender system is to gather expert ratings for educational resources based on specific criteria. Using these ratings, the system constructs a recommendation ranking of resources and presents them in order from the most recommended to the least recommended.

Additionally, the system provides data visualization by creating petal charts for each resource, where each petal represents the value of a criterion, considering its importance. This helps users evaluate which aspects each recommended resource corresponds to and make informed choices.

The main components of the educational resources evaluation recommender system's functioning technology include:

1. Collection of electronic educational resources
2. Input of experts' evaluations according to criteria
3. Calculation of the recommendation rating
4. Formation of a list of recommended resources
5. Data visualization
6. Implementation of authorization and organization of access provision
7. Formation of the administrator interface

I. Collecting educational resources: The search for educational resources based on specified criteria is conducted on the Internet. The system utilizes criteria such as interactivity, multimedia, modifiability, cross-platform compatibility, open distribution, architecture, functionality, and more. The search results are filtered to ensure the most accurate and relevant resources based on different types of educational resources (e.g., resources for learning mathematics, resources for demonstrating educational materials, etc.).

After the search and selection of educational resources, the system stores general information about each resource. Essential attributes like title, description, link, and other metadata associated with each resource are stored in the database. The database accommodates an unlimited number of educational resources.

System administrators can update or remove existing resource records from the database. This enables the system to maintain the currency and accuracy of resource information.

For collecting educational resources from websites, web scraping is employed. The web scraping process is automated using web scrapers.

The main steps of web scraping are shown in Figure 5:

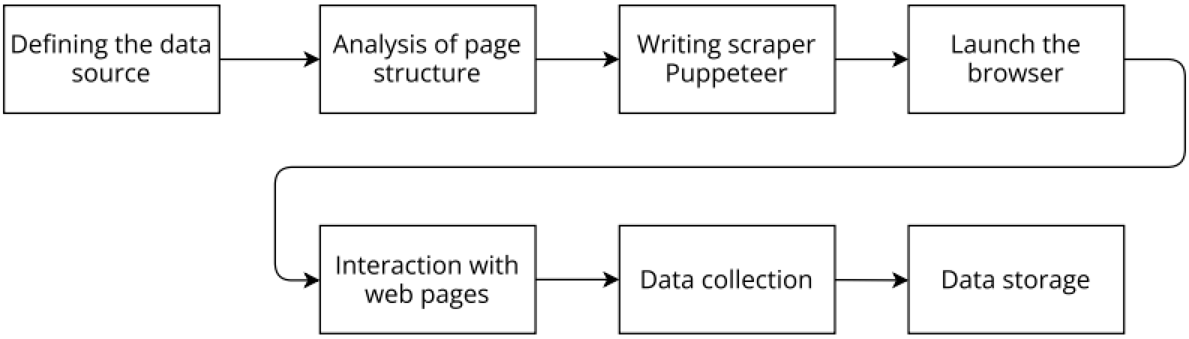

Defining the data source
Analysis of page structure
Writing scraper Puppeteer
Launch the browser
Interaction with web pages
Data collection
Data storage

Fig. 5. Workflow diagram of web scraping using Puppeteer

Step 1. Defining the data source involves accurately determining from which website or source data needs to be collected. It's important to consider the legality and ethical aspects of web scraping, as not all websites allow automated data collection.

Algorithm for defining the data source:

1. Determine the topic or domain from which data is needed (electronic learning resources for mathematics).
2. Choose websites that contain information about electronic learning resources related to the chosen topic (educational platforms, university pages, publishers, etc.).
3. Analyze data accessibility. Some websites might restrict access or require authorization.
4. Assess the size of the data source. A large volume of data may require additional resources.
5. Check the quality and reliability of the data.
6. Verify ethical considerations. Ensure actions do not infringe on copyright or other ethical aspects.

Step 2. Analyzing the structure of pages and locating the data to be gathered. This helps identify which elements of the page need to be found and how to access that data.

Algorithm for analyzing page structure:

1. Define target data (textual data, images, and other types of content).
2. Review the data source. Open the data source in a web browser and explore the page structure. Use the browser's inspector mode to view HTML code and CSS styles.
3. Study the DOM structure. Examine the DOM structure of pages to find elements containing the desired data.
4. Use selectors. Employ CSS selectors or XPath to choose specific elements on the page containing the necessary data. Selectors help locate elements by their attributes, classes, or other characteristics.
5. Interact with the page. Test how the page responds to interaction. Keep in mind that some data may be dynamically loaded through JavaScript after the page is displayed.
6. Process the data. Develop a scraper that collects and processes data from the pages.

Step 3. Writing a Puppeteer scraper, which controls the Chrome or Chromium browser, automatically browses web pages, and finds, and extracts the required data.

To write a scraper using Puppeteer, you need to install Node.js and the Puppeteer library itself.

You can write a scraper with the following code:

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

(async () => {
  try {
    // Open the browser
    const browser = await puppeteer.launch({ headless: true });
    const page = await browser.newPage();

    // Navigate to the data source page
    await page.goto('https://example.com');

    // Perform actions on the page (click a button, fill out a form, etc.)
    await page.click('#button-id');
    await page.type('#input-id', 'your search query');

    // Wait for data to load
    await page.waitForTimeout(3000);

    // Gather data
    const data = await page.evaluate(() => {
      // Execute JavaScript code in the page context to gather data
      // For instance, you can locate the desired elements
      const elements = document.querySelectorAll('.data-element');
      const dataArray = Array.from(elements).map(element => element.textContent);
      return dataArray;
    });

    console.log(data);

    // Close the browser
    await browser.close();
  } catch (err) {
    console.error('Error:', err);
  }
})();
```

In this case, Puppeteer was used to opening a browser, go to `https://example.com`, perform some actions on the page (such as clicking a button and filling in a search field), wait for the data to be loaded, and collect data from a page using `page.evaluate()`.

After running the scraper, you can see the collected data results in the console.

Step 4. Launching the browser using the scraper to interact with web pages.

One of the main components of a web scraper is the browser used to interact with web pages. Puppeteer provides control over the Chrome or Chromium browser, allowing you to perform actions on the page and retrieve data.

The working algorithm is as follows:

1. Opening the browser. The first step is to launch the browser using Puppeteer. The Puppeteer library provides the `puppeteer.launch()` method, which allows you to open a new instance of the browser.

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

(async () => {
  const browser = await puppeteer.launch();
  // Next, you can use the browser variable to control the browser
})();
```

2. Creating a new page. After launching the browser, you can create a new page instance using the `browser.newPage()` method. A new page allows you to navigate to different URLs, perform actions on the page, and retrieve data.

```
const page = await browser.newPage();
```

3. Interacting with the page. After creating a page, you can interact with it. For example, you can navigate to a URL, click buttons, fill in forms, scroll the page, and perform other actions.

```
await page.goto('https://example.com');
await page.click('#button-id');
await page.type('#input-id', 'your search query');
```

4. Data collection. Puppeteer retrieves data from the page using the `page.evaluate()` method. You can execute JavaScript code in the context of the page to find the desired elements or data.

```
  const data = await page.evaluate(() => {
  const elements = document.querySelectorAll('.data-element');
  const dataArray = Array.from(elements).map(element => element.textContent);
  return dataArray;
});
```

5. Closing the browser. After finishing your work with the browser, it's important to close it to free up resources. Use the `browser.close()` method to close the browser.

```
await browser.close();
```

Thus, Puppeteer allows you to control the browser, interact with web pages, and read data, which allows you to implement web scraping and other automated tasks.

Step 5. Interacting the scraper with web pages by performing actions that a user can perform in the browser, including clicking buttons, filling in forms, navigating through links, and more.

Overall, Puppeteer performs actions similar to those of a user in a browser. This allows the scraper to read and collect data from various web pages, making it a powerful tool for web scraping and automating actions on web pages.

Step 6. Collecting necessary data from pages.

After data collection, the scraper can output data to the console, save it to a file, upload it to a server, or process it further.

Step 7. Saving data in the appropriate format in a MongoDB database.

MongoDB is a document-oriented NoSQL database that stores data in JSON-like documents. This accommodates unstructured or semi-structured data, which is often encountered in web scraping.

Below is an example of how to save collected data in MongoDB using Node.js and Mongoose (a library for working with MongoDB):

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

// Connect to the MongoDB database
mongoose.connect('mongodb://localhost:27017/my_database', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

// Define the document schema
const dataSchema = new mongoose.Schema({
  title: String,
  description: String,
  url: String,
  // Other fields as needed
});

// Create a model based on the schema
const DataModel = mongoose.model('Data', dataSchema);

// Assuming data is an object of collected data
const data = {
  title: 'Sample Title',
  description: 'Sample Description',
  url: 'https://example.com',
};

// Save data in the database
const newData = new DataModel(data);
newData.save((err, savedData) => {
  if (err) {
    console.error('Error while saving data:', err);
  } else {
    console.log('Data successfully saved:', savedData);
  }
});
```

In this case, when connecting to the MongoDB database, a schema for the documents is defined, a model is created based on the schema, and the collected data is stored in the database.

Tools such as Node.js and the Puppeteer library itself were used to build the Puppeteer scraper. Below is a list of the tools that were used:

1. Node.js: A JavaScript runtime that allows executing JavaScript code on the server. Node.js is used to leverage Puppeteer for browser control and web scraping.
2. Puppeteer: A Node.js library that provides an interface to control the Chrome or Chromium browser. It allows automating interactions with web pages, performing actions on pages, and collecting data for further use.
3. MongoDB: The database used for storing collected data.
4. Mongoose: A Node.js library that simplifies working with MongoDB by allowing the definition of document schemas and creating models for storing and retrieving data.

II. Introduction of expert evaluation based on criteria. Experts can authenticate in the system to access the resource evaluation functionality. Dedicated roles for experts enable them to rate resources and save their evaluations.

Experts can also select predefined evaluation criteria for each resource and rate each resource according to the chosen criteria. The system stores the ratings provided by each expert for further use in calculating the recommendation rating. The system ensures transparency and reliability in the evaluation process by maintaining a history of ratings provided by each expert. This allows for analysis and verification of the accuracy of the ratings and ensures high-quality recommendations.

System administrators can manage experts' access rights and provide adequate control over the evaluation process. Expert evaluations are a crucial stage in building a quality recommendation rating that meets users' needs.

An interactive web form is used as a means for experts to input ratings for each criterion of each electronic educational resource (EER).

The algorithm for the interactive web form to collect expert ratings based on criteria for each EER includes the following steps:

1. Display Form: The user (expert) opens the web form for inputting ratings.
2. Select EER: The user selects the type of electronic educational resource.
3. Input Ratings: The user enters ratings for each criterion from 1 to 9, expressing the degree of correspondence of that criterion to the resource.
4. Summary Step: The ability to review and confirm entered data or edit the ratings.
5. Data Saving: The data is stored in the database for further processing and calculation of recommendations for users.

This is a general algorithm for the interactive web form. The ability to input ratings for each criterion and for each EER helps provide personalized and accurate recommendations for users based on their preferences and requirements for educational resources.

The following tools were used to organize the interactive web form for expert ratings input:

1. Express.js: A web framework that allows creating web servers and handling HTTP requests.
2. CSS: Used for styling the web form.
3. Bootstrap: Used to create a responsive and user-friendly interface.
4. MongoDB: Used to store expert ratings based on criteria.

III. Calculation of recommendation rating. The calculation of the recommendation rating for electronic educational resources is based on the ratings provided by experts for criteria and their weights.

The main calculation steps are as follows:

1. Use of expert ratings: The system utilizes the collected ratings from experts for each criterion of each EER. Expert ratings are numerical values ranging from 9 to 1, where 9 represents absolute conformity, and 1 represents no conformity. These values are further processed using fuzzy logic methods.
2. Determination of criterion weights: The process of determining the weight of each criterion relative to others is carried out using the pairwise comparison method. Experts compare criteria to each other and indicate which criteria are more important than others.
3. Normalization of ratings: Expert ratings are normalized to a single value range. This allows equal consideration of the values of different criteria in the rating calculation.

4. Rating calculation: The recommendation rating is calculated using fuzzy logic methods to process expert ratings and determine the degree of recommendation for each resource. Fuzzy logic methods enable the consideration not only of precise numerical values but also variable degrees of conformity. This is essential for evaluating EERs based on multiple criteria. By using fuzzy rules and interpreting fuzzy terms, the system assigns a rating to each resource based on aggregated expert ratings.

5. Sorting of Ratings: The obtained ratings are sorted in descending order, from the most recommended resource to the least recommended one. This way, users receive a list of E-learning resources (ENRs) in the order of the recommendation rating.

This algorithm takes into account the weight of each criterion and the expert ratings. It enables the creation of personalized recommendations for users based on their preferences and needs. If there are changes in expert ratings or criterion weights, the recommendation rating can also change. This allows the recommendations to adapt to changing conditions and user requirements.

To calculate the recommendation rating, the method of fuzzy logic, the method of pairwise comparisons, and the weighted average method were used.

The fuzzy logic method is employed for calculating the recommendation rating. It facilitates the evaluation of E-learning resources while considering vague and ambiguous aspects that often arise during assessment. The fuzzy logic method employs fuzzy sets and rules to describe fuzzy relationships among data.

The algorithm for utilizing the fuzzy logic method to calculate the recommendation rating is as follows (Figure 6):

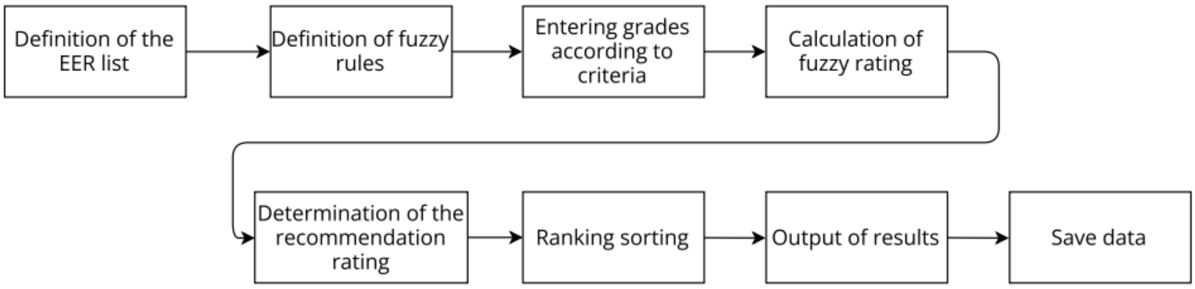

Definition of the EER list
Definition of fuzzy rules
Entering grades according to criteria
Calculation of fuzzy rating
Determination of the recommendation rating
Ranking sorting
Output of results
Save data

Fig. 6. Algorithm for using the fuzzy logic method to calculate recommendation rating

The application of fuzzy logic allows for a precise evaluation of resources while considering the ambiguity and uncertainty in expert ratings, thus improving the quality of recommendations.

Using the method of paired comparisons to determine the importance of criteria involves the process of comparing each criterion with each other to determine the relative importance of the criteria. This method helps to understand which criteria are more important or acceptable compared to others [18].

The algorithm for using the pairwise comparison method to determine the importance of criteria is as follows (Figure 7):

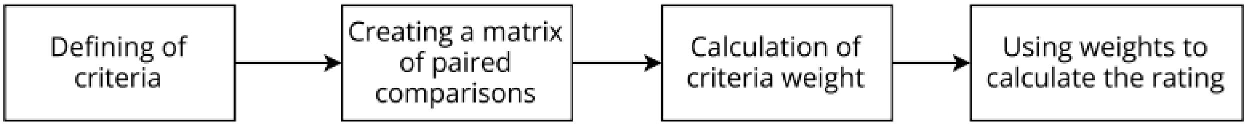

Defining of criteria
Creating a matrix of paired comparisons
Calculation of criteria weight
Using weights to calculate the rating

Fig. 7. Algorithm for using the method of pairwise comparisons

1. Definition of criteria: You need to define a list of criteria by which resources are evaluated.
2. Creation of pairwise comparison matrix: Experts compare each criterion with every other criterion and establish which criterion they consider more important. The result is a matrix where each element represents the relative weight of one criterion compared to others.
3. Calculation of criterion weights: Based on the pairwise comparison matrix, the weights of criteria are calculated. This is done using methods such as the eigenvalue analysis method, and the Saaty method.
4. Use of weights for rating calculation: After obtaining the weights of criteria, they are used to calculate the rating of each resource based on the evaluations provided by experts for each criterion.

The pairwise comparison method helps to reconcile the opinions of different experts and consider their relative importance for each criterion. This allows for the creation of more objective and balanced recommendations for users.

The weighted average method is a simple and widely used approach for calculating the recommendation rating based on criterion evaluations and their weights. It enables the consideration of each criterion's importance in calculating the rating for each electronic learning resource.

To apply the weighted average method, you need to follow these steps (Figure 8):

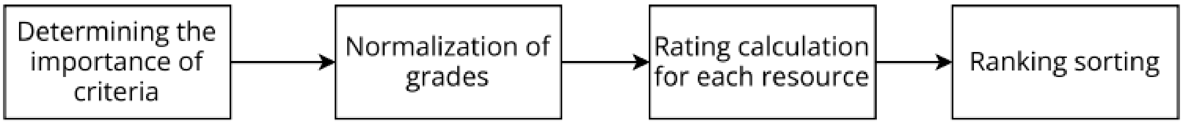

Determining the importance of criteria
Normalization of grades
Rating calculation for each resource
Ranking sorting

Fig. 8. Algorithm for using the weighted average method

1. Determining the weight of criteria: Experts determine the weight of each criterion relative to others.
2. Normalization of ratings: Since expert ratings may have different value ranges, they need to be normalized to a single range. This allows for equal consideration of the values of different criteria when calculating the rating.

3. Calculation of ratings for each resource: A weighted average rating is calculated for each electronic learning resource. For each criterion, the normalized rating is multiplied by its corresponding weight, and then the scores for all criteria are summed up.

4. Rating sorting: The obtained ratings are sorted in descending order, from the most recommended to the least recommended resource.

The weighted average method allows experts to consider individual needs and the importance of different criteria when recommending electronic learning resources.

IV. Formation of recommended resource list: The formation of the recommended resource list is based on the calculated ratings for each resource. After processing expert ratings and determining the importance of criteria, approaches are applied to create the list of recommended resources.

Below is the algorithm for forming this list:

1. Calculation of ratings for each electronic learning resource.
2. Sorting of ratings.
3. Resource filtering.
4. Personalization of recommendations.
5. Displaying results.

Interactive filters are used for resource filtering in the recommendation system. Users can choose filters on the web page to limit the list of recommended resources.

The following libraries are used for organizing interactive filters:

1. Express.js: Express.js is used to create a web application on Node.js.
2. Formidable: Formidable is a library for handling forms in Node.js. It allows for receiving and processing data from user-filled forms.
3. Natural Language Processing (NLP) Libraries: For processing textual content and language analysis, libraries like Natural, NLP.js, and Tokenizer are used. These libraries help identify keywords, exclude unnecessary words, and understand the semantics of the text.

The functionality of the recommended resource list helps provide a convenient and informative way for users to select the best electronic learning resources according to their needs and interests.

V. Data Visualization: For each recommended resource, the system automatically constructs a petal diagram that visualizes the values of each evaluation criterion. Each spoke of the chart represents an individual criterion and its score[19]. These petal diagrams are interactive, allowing users to hover over a spoke to view the specific criterion value. This enables users to quickly assess the resource's strong and weak aspects, aiding them in making informed decisions about resource selection based on specific requirements.

Petal diagrams are an effective method for visualizing recommendation system data, as they present multidimensional resource data and properties. Each axis of the chart corresponds to a separate criterion, and the values on the axes show the degree of compliance of each recommended resource with that criterion.

For constructing petal diagrams in the Node.js environment, the following tools are used:

1. Chart.js: A library for creating various types of charts, including petal charts. It provides a wide range of customization options and styling capabilities for charts.
2. Echarts: A library that has built-in support for petal charts, along with many settings for customization and interactivity.
3. Plotly: A library for creating interactive plots and charts, including petal charts. It offers a convenient way to create interactive visualizations for data.

VI. Implementation of authorization and access control: Organizing authorization and ensuring access to the recommendation system is an important aspect of ensuring security and control over the system. The main methods used for this purpose are as follows:

1. User registration and authentication: Users need to register in the system and create an account. After that, they can use their account details for authentication and login.
2. Password protection: User passwords are protected and stored in encrypted form using libraries like bcrypt.js in Node.js.
3. Assignment of roles and permissions: The system can have different user roles, such as administrator, expert, etc. Each role has its permissions for accessing resources and system functions.

Utilizing these methods helps ensure a high level of security and protection of the recommendation system, ensuring compliance with regulatory requirements and the protection of user personal data.

For organizing user registration and authentication, the following tools are used:

1. Passport.js: A library for user authentication in Node.js. It supports various authentication strategies, including local authentication using username and password, authentication through social media (Facebook, Google, Twitter, etc.), JWT (JSON Web Tokens), and many more.
2. bcrypt.js: A library for hashing passwords in Node.js. It allows the protection of user passwords by generating a salt and applying a hash function.
3. express-session: Middleware for storing user session information. It allows for storing session data in a MongoDB repository.
4. jsonwebtoken: A library for creating and verifying JSON Web Tokens (JWT). JWT is used to store information about an authenticated user, such as their identifier or role.

5. MongoDB: Used for storing user account data. It provides fast data access and the ability to structure user information.
6. RBAC (Role-Based Access Control) Libraries: Some specialized libraries, such as 'rbac' and 'accesscontrol', allow organizing role assignments and permissions according to the RBAC concept. With the help of such libraries, roles are defined, permissions are granted, and user rights are checked before performing certain actions.

Libraries help effectively manage roles and permissions in the recommendation system and ensure security and resource access control.

VII. Formation of Administrative Interface: Only users with special administrator privileges have access to the administrative interface. The administrator needs to authenticate to gain access to the control panel.

An administrator can perform the following actions:

- Add, edit, and delete evaluation criteria used by experts to assess resources.
- Manage the list of experts, add new experts, block, or delete expert accounts as needed.
- Add new electronic learning resources to the system or remove outdated ones that are no longer relevant, enabling the administrator to maintain the relevance and pertinence of the recommendation system.
- Manage access rights for different users, ensuring data security and confidentiality in the system.

The administrative interface can:

- Record administrator events and actions for debugging and tracking actions in the system.
- Ensure the efficiency of the administrator's work by being convenient and intuitive.
- Assist administrators in effectively managing the recommendation system, allowing them to configure and maintain the system in an up-to-date state while ensuring data security.

The creation of administrative interfaces in Node.js is accomplished using various approaches and tools. The primary goal of the administrative interface is to provide administrators with the ability to control and oversee the system. Administration is an important part of the recommendation system, as it enables effective management of users, system configuration, and monitoring of operations.

The recommendation system is implemented as a multi-page web application, offering several advantages:

- Rich interface functionality.
- Rapid interface response due to reduced server requests for each action.
- Significant reduction in server load.

– Personalization and data transfer speed.

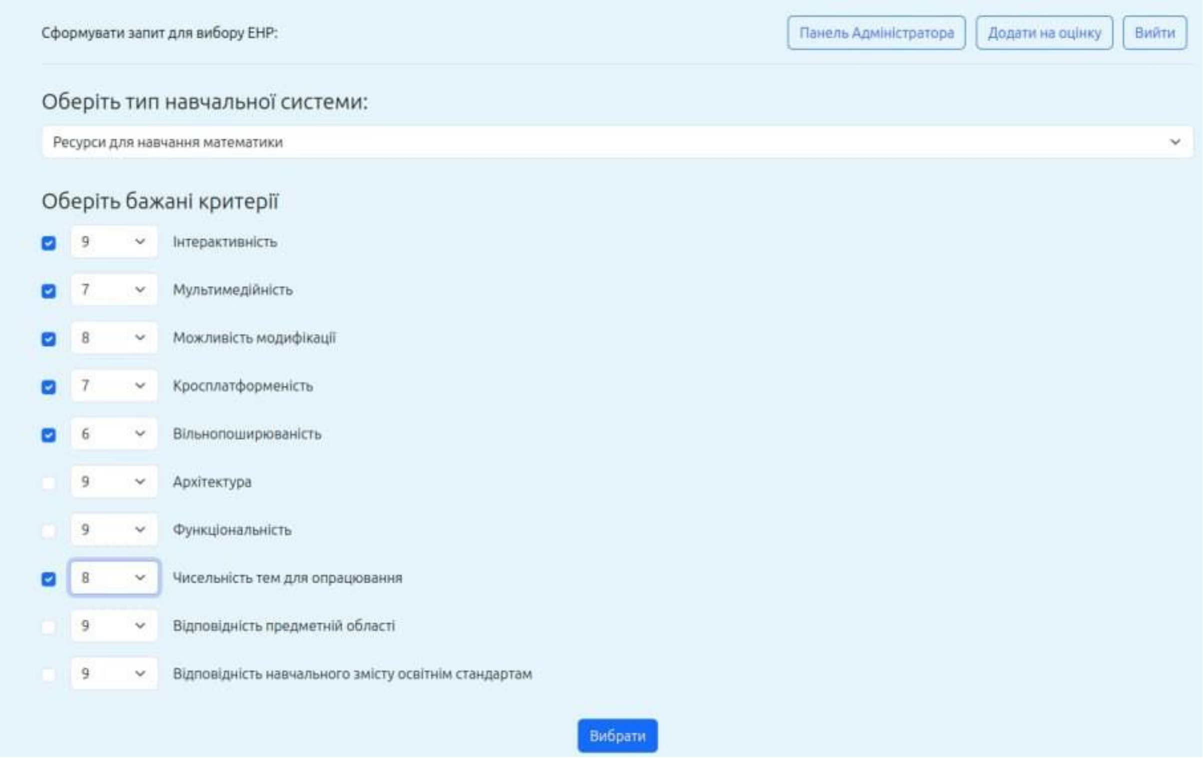
The software implementation was carried out using the following software tools and programming languages:

1. Node.js: An open-source platform for executing high-performance network applications written in JavaScript. It is used for creating web application scripts.
2. Express: A widely used Node.js-based framework for web application and API development. It provides a minimal set of functions that allows web developers to create fast and efficient web applications with less code compared to manual development using pure Node.js.
3. Twig Template Engine (for HTML): Twig is a template engine used for developing HTML templates in PHP applications.
4. CSS (Cascading Style Sheets): A style language used for formatting and presenting the appearance of web pages written using markup languages such as HTML or XML.
5. MongoDB: A document-oriented database categorized as a NoSQL database.
6. Npm (Node Package Manager): One of the most popular package managers in the Node.js and JavaScript environment. It allows developers to manage and use third-party libraries and modules developed by other programmers. It also enables publishing their packages for other users to utilize.

The web application's interface is one of the most significant elements, comprising a set of visual elements for convenient perception and understanding of information by users of the recommendation system.

User access to working with the recommendation system involves performing authorization or registration procedures.

On the main page of the recommendation system (Fig. 9), users can choose the type of learning resources and criteria by which to explore resources and assign weights to each selected criterion.



Сформувати запит для вибору ЕНР:

Оберіть тип навчальної системи:

Ресурси для навчання математики

Оберіть бажані критерії

☑ 9 Інтерактивність
☑ 7 Мультимедійність
☑ 8 Можливість модифікації
☑ 7 Кросплатформеність
☑ 6 Вільнопоширюваність
☐ 9 Архітектура
☐ 9 Функціональність
☑ 8 Чисельність тем для опрацювання
☐ 9 Відповідність предметній області
☐ 9 Відповідність навчального змісту освітнім стандартам

Fig. 9. Criteria selection page of the recommendation system

The process of selecting the type of electronic resources and setting the criteria is described by the following code:

```
app.post("/apply-resource", async (req, res) => {
    try {
        // Get the selected resource types from the request body
        let bulkData = req.body
        let chosenType_of_resource = bulkData.type_of_resource
        let chosenImportance = bulkData.importance
        let selectedCheck = bulkData.check

        //put numbers only into chosenImportance
        for ( let i = 0; i < chosenImportance.length; i++) {
            chosenImportance[i] = parseInt(chosenImportance[i])
        }

        //put numbers only into selectedCheck
        let name_of_each_vertex = []
        for (let i = 0; i < chosenCheck.length; i++) {
            name_of_each_vertex[i] = chosenCheck[i].slice(0, -2)
            chosenCheck[i] = parseInt(chosenCheck[i].slice(-1))
        }
```

The query to the database is carried out by the command:

```
const LOST = await Category.find({type: chosenType_of_resource});

        //working names
        // LOST = list of Systems That Fit the criteria
```

To build matrices of pairwise comparisons, the following commands were used:

```
// STEP1 - COMPUTING THE PULL OF MATRIX
        // this is 3-dimensional matrix in format: matrix3d_forAGn[pick AGn from 0 to 9][pick a row for selected matrix][pick index]
        // matrix3d_forAGn[AGn][h][i]
        const matrix3d_forAGn = []
        for (let AGn = 0; AGn < 10; AGn++) {
            let matrix = []
            for (let i = 0; i < LOST.length; i++) {
                let line = []
                for (let h = 0; h < LOST.length; h++){
                    let result
                    let h_rating = LOST[h].params[AGn].rating
                    let i_rating = LOST[i].params[AGn].rating
                    if (i_rating === h_rating) {
                        result = 1
                    } else if (i_rating > h_rating) {
                        result = i_rating - h_rating
                    } else {
                        result = 1/(h_rating - i_rating)
                    }
                    line.push(result)
                }
                matrix.push(line)
            }
            matrix3d_forAGn.push(matrix)
        }
```

You can get fuzzy sets using the following commands:

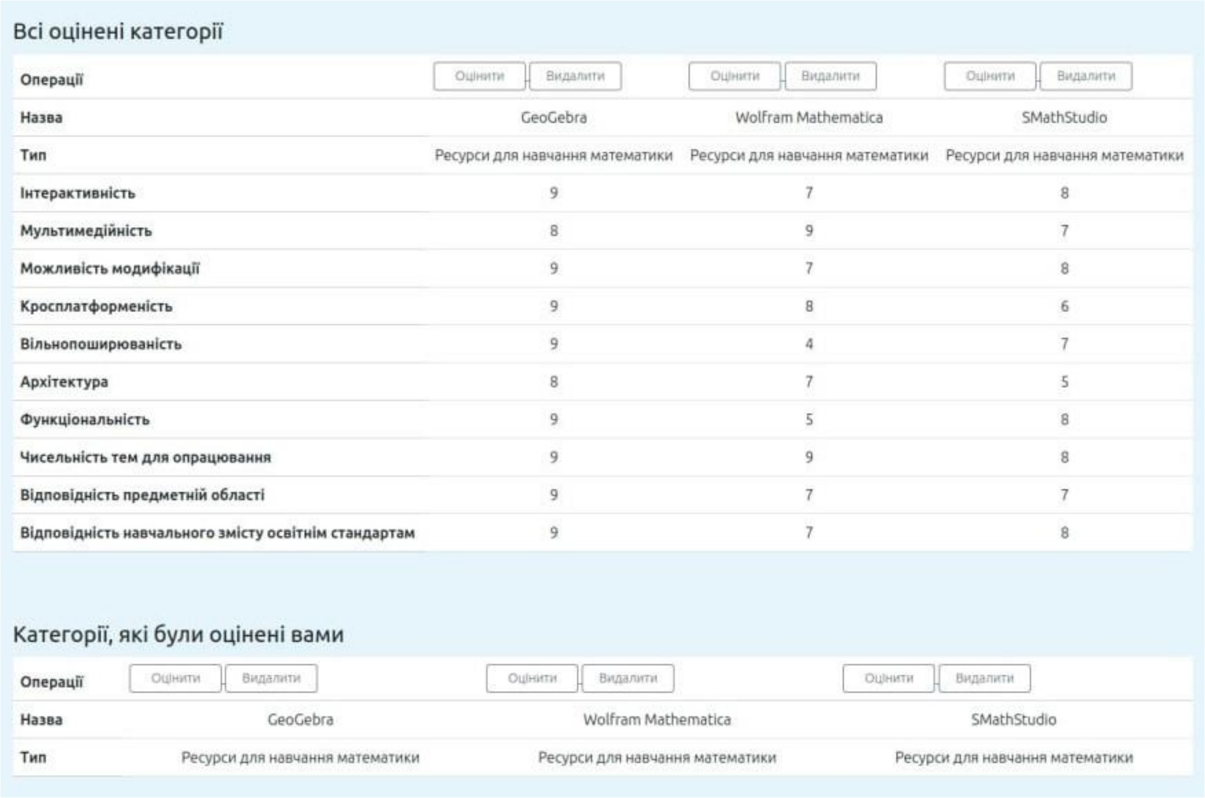
```
// STEP - FINDING AN INVERTED SUM OF COLUMNS AND PUTTING IT INTO THE ARRAY
        let Gn_matrix = []
        for (let Gn = 0; Gn < matrix3d_forAGn_filtered.length; Gn++) {
            let Gn_line = []
            for (let i = 0; i < LOST.length; i++) {
                let result = 0
                for(let h = 0; h < LOST.length; h++){
                    result += matrix3d_forAGn_filtered[Gn][h][i]
                }
                let inverted_result = 1/result
                Gn_line.push(inverted_result)
            }
            Gn_matrix.push(Gn_line)
        }
```

The construction of the matrix used to calculate the weight of the criteria is carried out as follows:

```
// STEP - PICK ONLY CHOSEN PARAMS FOR WEIGHTS TO BUILD A-MATRIX
    let chosenImportance_filtered = []
    for (let i = 0; i < chosenCheck.length; i++) {
        let element = chosenImportance[chosenCheck[i]]
        chosenImportance_filtered.push(element)
    }

    // STEP - EVALUATE THE WEIGHT OF EACH PARAMETER
    let weight_matrix = []
    for (let i = 0; i < chosenCheck.length; i++) {
        let line = []
        for (let h = 0; h < chosenCheck.length; h++){
            let result
            if (chosenImportance_filtered[i] === chosenImportance_filtered[h]) {
                result = 1
            } else if (chosenImportance_filtered[i] > chosenImportance_filtered[h]) {
                result = chosenImportance_filtered[i] - chosenImportance_filtered[h]
            } else {
                result = 1/(chosenImportance_filtered[h] - chosenImportance_filtered[i])
            }
            line.push(result)
        }
        weight_matrix.push(line)
    }
```

Each expert can track the categories that were recently evaluated, the categories that have been evaluated by the expert overall, and the categories that have been evaluated by other experts (Fig. 10).



Всі оцінені категорії

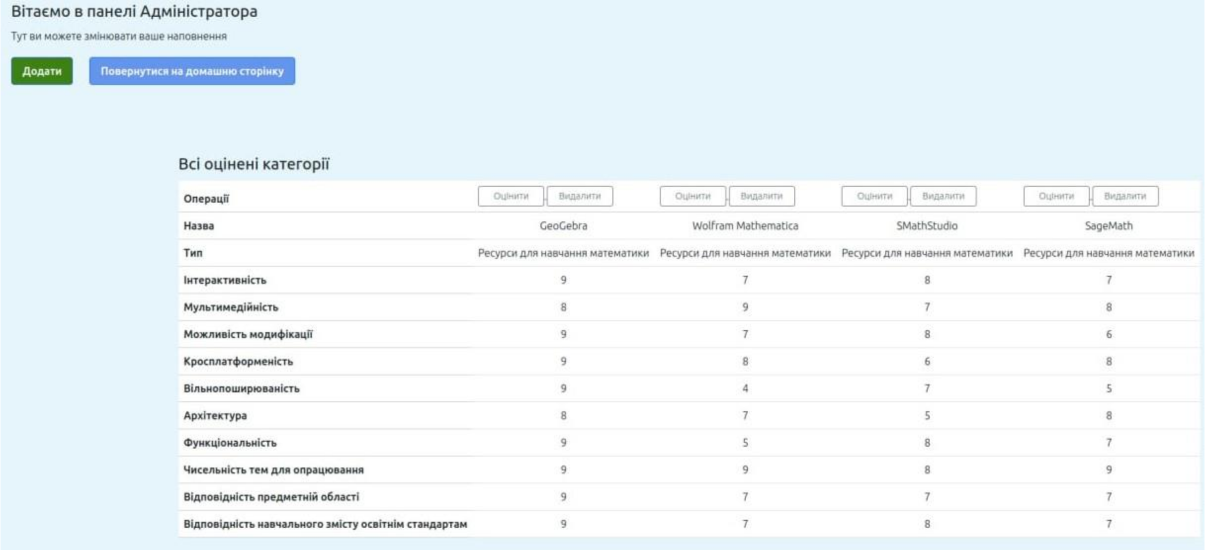
Операції	Оцінити Видалити	Оцінити Видалити	Оцінити Видалити
Назва	GeoGebra	Wolfram Mathematica	SMathStudio
Тип	Ресурси для навчання математики	Ресурси для навчання математики	Ресурси для навчання математики
Інтерактивність	9	7	8
Мультимедійність	8	9	7
Можливість модифікації	9	7	8
Кросплатформеність	9	8	6
Вільнопоширюваність	9	4	7
Архітектура	8	7	5
Функціональність	9	5	8
Чисельність тем для опрацювання	9	9	8
Відповідність предметній області	9	7	7
Відповідність навчального змісту освітнім стандартам	9	7	8

Категорії, які були оцінені вами

Операції	Оцінити Видалити	Оцінити Видалити	Оцінити Видалити
Назва	GeoGebra	Wolfram Mathematica	SMathStudio
Тип	Ресурси для навчання математики	Ресурси для навчання математики	Ресурси для навчання математики

Fig. 10. Expert Panel

The administrator panel, where you can manage resources, their types, and criteria, looks as follows (Fig. 11)



Вітаємо в панелі Адміністратора

Тут ви можете змінювати ваше наповнення

Всі оцінені категорії

Операції	Оцінити Видалити	Оцінити Видалити	Оцінити Видалити	Оцінити Видалити
Назва	GeoGebra	Wolfram Mathematica	SMathStudio	SageMath
Тип	Ресурси для навчання математики	Ресурси для навчання математики	Ресурси для навчання математики	Ресурси для навчання математики
Інтерактивність	9	7	8	7
Мультимедійність	8	9	7	8
Можливість модифікації	9	7	8	6
Кросплатформеність	9	8	6	8
Вільнопоширюваність	9	4	7	5
Архітектура	8	7	5	8
Функціональність	9	5	8	7
Чисельність тем для опрацювання	9	9	8	9
Відповідність предметній області	9	7	7	7
Відповідність навчального змісту освітнім стандартам	9	7	8	7

Fig. 11. Administrator panel

The result of the recommendation system's work appears as follows (Fig. 12): resources are arranged in order of the most relevant to the specified criteria, and radar charts are constructed for them, allowing tracking of criteria compliance [19].

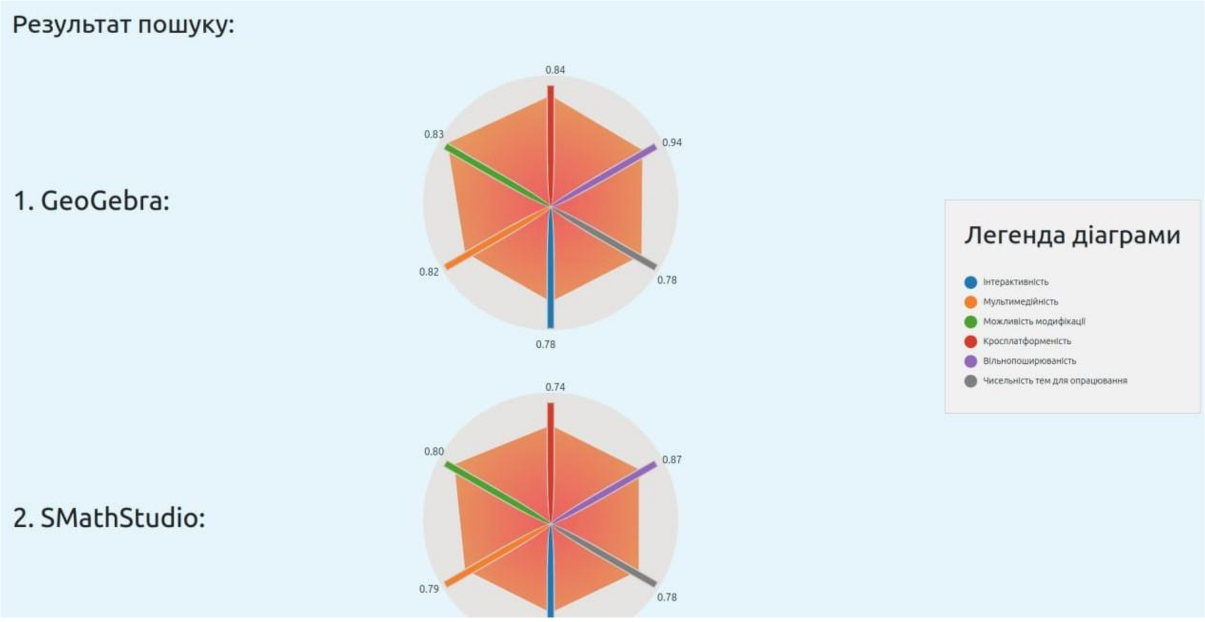


Fig. 12. Result of the recommendation system's work.

V. Conclusion

As a result of the research, a recommendation system has been developed. It generates recommendations based on fuzzy logic methods for expert evaluation of electronic educational resources. This system serves to select alternatives in the process of choosing the most effective EERs for use in the educational process. The process of EER selection is based on fuzzy logic theory using the Analytic Hierarchy Process (AHP) method. A methodology has been created for working with qualitative and quantitative criteria under conditions of uncertainty. The applicability of these methods is illustrated using the example of the problem of selecting electronic educational resources. An analysis of recommendation system concepts has led to the introduction of the concept of a recommendation system that generates recommendations for selecting the most effective electronic educational resources. UML diagrams of use cases, sequences, and activities have been provided for the recommendation system for choosing electronic educational resources. A three-tier architecture of the recommendation system has been presented, including the presentation layer, the application layer (components: specifying the type of electronic educational resource, resource search, displaying results, displaying reports), and the data management layer (subsystem for defining the type of EER, subsystem for analyzing EER, subsystem for generating results, subsystem for generating user reports). The main components of the functionality of the recommendation system for evaluating electronic educational resources are described in detail, including collecting electronic educational resources, inputting expert ratings by criteria, calculating recommendation ratings, generating a list of recommended resources, data visualization, implementing authorization and access control, and forming the administrative interface. Methods and tools of implementation are described for each of these components. The program code for development has been provided for each of the key stages of the recommendation system. The most significant elements of the web application interface are shown, including the criteria selection page of the recommendation system, the administrator panel, the panel for changing and adding categories, and the output of the recommendation system's results. The recommendation system for evaluating and selecting electronic educational resources, which provides users with recommendations according to their requirements and selected criteria, represents an important and relevant tool in the field of education. This system allows users to be provided with a selection of the most suitable electronic educational resources that contribute to enhancing the quality of learning. The use of fuzzy logic and the expert pairwise comparison method in the calculation of recommendation ratings reflects a scientific approach and practical value of the system. These methods enable reconciling individual user differences and considering objective criteria in the evaluation of the quality of educational resources. The implementation of an interactive web form and content filter enables a convenient and efficient process of inputting expert ratings and eliminates unnecessary resources in the recommendation process. The use of visualizations, such as petal diagrams, adds clarity and understanding to the selection process. The organization of the administrative interface, password protection, and access control demonstrate a high level of security and manageability of the system. The use of databases for storing and organizing resources ensures convenient and efficient access to necessary data. Using these components, a comprehensive and balanced recommendation system has been created that meets the needs of users and contributes to enhancing the quality of learning through proper selection and recommendation of electronic educational resources.

References

[1] Pasichnyk, V. et al. Using fuzzy logic in the process of expert evaluation of learning resources. *Scientific Bulletin of UNFU*, vol. 32(4), 2022, pp. 66-76. https://doi.org/10.36930/40320411

[2] Esteban, A. et al. Helping university students choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization. *Knowl.-Based Syst.*, vol. 194, p. 105385, Apr. 2020, doi: 10.1016/j.knosys.2019.105385

[3] Cañas, A. et al. A Recommender System for Non-traditional Educational Resources: A Semantic Approach. J. *Univers. Comput. Sci.,* vol. 21, no. 2, pp. 306–325, 2015

[4] Ko, H. et al. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*, vol. 11, no. 1, p. 141, Jan. 2022, doi: 10.3390/electronics11010141

[5] Ojokoh, B. et al. A fuzzy logic-based personalized recommender system. *Int. J. Comput. Sci. Inf. Technol. Secure.*, vol. 2, no. 5, pp. 1008–1015, 2012

[6] Artemenko, O. et al. E-tourism recommender systems: a survey and development perspectives. *Econtechmod*, vol. 6, no. 2, pp. 91–95, 2017

[7] Wang, F. Personalized Recommendation System of College Students' Employment Education Resources Based on Cloud Platform. in *e-Learning, e-Education, and Online Training, W. Fu and G. Sun, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics, and Telecommunications Engineering,* vol. 454. Cham: Springer Nature Switzerland, 2022, pp. 318–333. doi: 10.1007/978-3-031-21164-5_25

[8] Lin, J. et al. Intelligent Recommendation System for Course Selection in Smart Education. *Procedia Comput.* Sci., vol. 129, pp. 449–453, 2018, doi: 10.1016/j.procs.2018.03.023

[9] Shu, J. et al. A content-based recommendation algorithm for learning resources. *Multimedia. Syst.*, vol. 24, no. 2, pp. 163–173, Mar. 2018, doi: 10.1007/s00530-017-0539-8

[10] Xu Z. et al. Study on Personalized Recommendation Algorithm of Online Educational Resources Based on Knowledge Association. *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Sep. 2022, doi: 10.1155/2022/2192459

[11] Slimani, H. et al. Semantic recommendation system of digital educational resources. in *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, Rabat Morocco: ACM, Oct. 2018, pp. 1–6. doi: 10.1145/3289402.3289513.

[12] Kotsyuba, I. et al. Recommendation web service for choosing an individual educational path in the field of transportation systems. *programming Transp. Res. Procedia*, vol. 63, pp. 591–599, 2022, doi: 10.1016/j.trpro.2022.06.052

[13] Morales, A. et al. Recommendation system with graph-oriented databases for a repository of open educational resources. *IOP Conf. Ser. Mater. Sci. Eng.,* vol. 1154, no. 1, p. 012021, Jun. 2021, doi: 10.1088/1757-899X/1154/1/012021

[14] Dwivedi, P. et al. e-Learning recommender system for a group of learners based on the unified learner profile approach. *Expert Syst.*, vol. 32, no. 2, pp. 264–276, Apr. 2015, doi: 10.1111/exsy.12061

[15] Tarus, J. K. et al. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artif. Intell. Rev.*, vol. 50, no. 1, pp. 21–48, Jun. 2018, doi: 10.1007/s10462-017-9539-5

[16] Pasichnyk, V. et al. Selection of electronic educational resources using a recommendation system. in *Theses of the reports of the 12th International Scientific and Practical Conference 'Mathematics. Information Technologies. Education in Information Technologies,* vol. XII. Lutsk-Svitiaz: Lesya Ukrainka Volyn National University, Jun. 2023, pp. 129–131

[17] Pasichnyk, V. et al. Model of the Recommender System for the Selection of Electronic Learning Resources. *CEUR Workshop Proc. 5 Rd Int. Workshop Mod. Mach. Learn. Technol. Data Sci.,* no. 3426, pp. 344–355, 2023

[18] Yunchyk, V. et al. Application of the hierarchy analysis method for the choice of the computer mathematics system for the IT-sphere specialist's preparation. *J. Phys. Conf. Ser.,* vol. 1840, no. 1, p. 012065, Mar. 2021, doi: 10.1088/1742-6596/1840/1/012065

[19] Pasichnyk, V. et al. Comparative characteristics of the functionality of the system of computer mathematics in the process of task solving. *Journal of Lviv Polytechnic National University "Information Systems and Networks*", no. 11, pp. 87–102, Jun. 2022, doi: 10.23939/sisn2022.11.087

[20] Liu, Y. Personalized Recommendation Service of Educational Media Resources Based on Multi-dimensional Feature Fusion, Int. *J. Emerg. Technol. Learn.,* vol. 18, no. 07, pp. 131–146, Apr. 2023, doi: 10.3991/ijet.v18i07.39233.

[21] Thongchotchat, V. et al. Educational Recommendation System Utilizing Learning Styles: A Systematic Literature Review. *IEEE Access*, vol. 11, pp. 8988–8999, 2023, doi: 10.1109/ACCESS.2023.3238417.

[22] Felfernig, A. et al. Constraint-Based Recommender Systems. in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds., Boston, MA: Springer US, 2015, pp. 161–190. doi: 10.1007/978-1-4899-7637-6_5.

[23] Jain, R. A true multimedia client. *IEEE MultiMedia*, vol. 12, no. 2, p. 104, Apr. 2005, doi: 10.1109/MMUL.2005.20.

[24] Hazar, M. J. et al. A Recommendation System Involving a Hybrid Approach of Student Review and Rating for an Educational Video. In *Review, preprint*, Feb. 2023. doi: 10.21203/rs.3.rs-2375194/v1.