DEVELOPMENT OF A WEB PLATFORM APPLICATION FOR THE ADMINISTRATION OF BACHELOR'S DEGREE FINAL PROJECTS

CHIRIAC Nicu – Manuel[1], TARBĂ Ioan – Cristian[2]
[1]Faculty of Industrial Engineering and Robotics, Study program: Applied Informatics in Industrial Engineering, Academic year: 4, e-mail nicu.chiriac99@yahoo.com
[2]Faculty of Industrial Engineering and Robotics, Manufacturing Engineering Department, University POLITEHNICA of Bucharest

ABSTRACT: The purpose of the developed application is to facilitate the way in which the projects of the students of final years are managed by the department management. The application must keep track of the students, the coordinating teachers, the assigned topics and the files that contribute to the final grade. Files can be text, 2D drawings, tables, etc. The platform administrator can monitor and approve the whole process of assigning homework, uploading files on time, approving students for the final exam, etc. The application uses web-specific technologies to ensure a consistent process from assigning the diploma topic to taking the bachelor's thesis final exam.

KEY WORDS: client, server, database, communication, authentication

1. Introduction

Efficient data management is a key point in the conduct of any process involving a data flow. A web application that integrates the data management part becomes necessary when multiple users from different locations need to access and / or modify certain records stored in a database. The application aims to achieve this goal, giving teachers the opportunity to view, insert, assign topics or extract information / files about the students in the years III and IV from Manufacturing Engineering Department. Within the platform, the administrator / director of the department / or the approved staff of the faculty can register students, teachers, homework assignments, assign them, creating the relationship between student - bachelor's thesis - coordinating teacher, upload various documents / files necessary for the student to be able to participate in the final exam. Also, the Admin of the platform (user with full access and rights) will be able to confirm the correctness of the data, give students access to the final verification after all their main data / files are validated, will be able to edit fields or delete records. The application will integrate all the necessary data for the bachelor's degree administration process.

2. Application structure and technologies

The project is divided into two distinct parts: Front-end and Back-end.

The Front-end is the user interface, also known as the client-side. Here are important the design elements (graphics display), their intuitiveness in use, adaptability to different types of displays, fast response. It is the interactive area of the platform, as this is where the requests are made - the exchange of data between the client and the server.

The Back-end (server) is the area where the logic needed to run the application is performed. Various endpoints (routes) are called here, which are in fact often asynchronous functions, which are built to serve, as the case may be, the needs of the user, which he can trigger (make requests) at any time he wishes to get data from the server or even transmit it. The back-end in this case is complemented by another extremely important component - the database.

The database serves to maintain the coherence of all the data that is saved / stored. With a strong emphasis on the idea of management, the platform uses a relational model database, separating the data

into tables specific to the type of records: students, teachers, topics, files, users. Using specific SQL methods, the attributes can be combined, forming new entities necessary for the case type.

The technologies used will be presented and defined as follows:

- **HTML** (HyperText Markup Language) is a descriptive language that creates the basic structure of a site, consisting of tags, elements and attributes.
- **CSS** (Cascading Style Sheets) represents the language used for editing / styling the layout and for formatting all the elements we see on the site.
- **JavaScript** is a client-side scripting language that is interpreted by the web browser. JavaScript scripts are used to increase the interactivity of web pages, giving users the power to take action within the web page. In the early stages of the Web, most sites were static, only in presentation format, meaning they displayed static elements, and there was no interaction between the page and the user.
- **React** is one of the open source JavaScript libraries. It is used to build interactive user interfaces. It is an efficient, declarative and flexible library. It deals with the component V in the Model-View-Controller Architecture (MVC). It's not a whole framework, it's just a front-end library. It allows the creation of complex user interfaces using isolated and small pieces of code known as components. The major advantage of the components is that the change to any other component does not affect the entire application. It was developed by the software engineer Jordan Walke, who works at Facebook. Facebook deployed it in their newsfeed and used it to improve its user interface. It was made public in May 2013. [1]
- **Node.js** is an open-source JavaScript execution environment, that runs on the Google Chrome V8 engine and can run JavaScript code outside of a web browser [2]. It is mainly used on the back-end (server) programming side.
- **Express.js** is one of the most used frameworks for Node.js. It makes the creation of HTTP routes and content parsing very easy for the application.
- **PostgreSQL** is a relational database management system (RDBMS) that uses Structured Query Language (SQL) as the main query language. [3]

2.1. Database

The database is a central landmark of the application. Most of the functionality is around it, as the platform exists for this purpose - to store and process data. The design of the database followed a number of well-established steps: organizing the necessary information, dividing the information into tables, transforming the information elements into columns (each element becomes a field in a table column), specifying the primary keys (unique way of identifying each row - ID), configuration of the relationships in the table, testing and application of normalization rules (checking the coherent structure to reduce redundant data).

Thus, the database was structured in 4 linked tables: „studenti” (students), „teme” (topics or homeworks), „profesori” (teachers) and „fisiere” (files). An additional table called "users" was created to store user authentication data (id, email, password), but that is standalone, having no direct connection to the rest of the tables. Fig. 1 shows the Entity Relationship Diagram (ERD), where the tables are presented with: table header - table name, primary key (PK), foreign key (FK) and the relationships (associations) between entities. In order to be able to make the diagram, it is necessary to know the cardinality between the tables; these were shown separately in Fig.2.

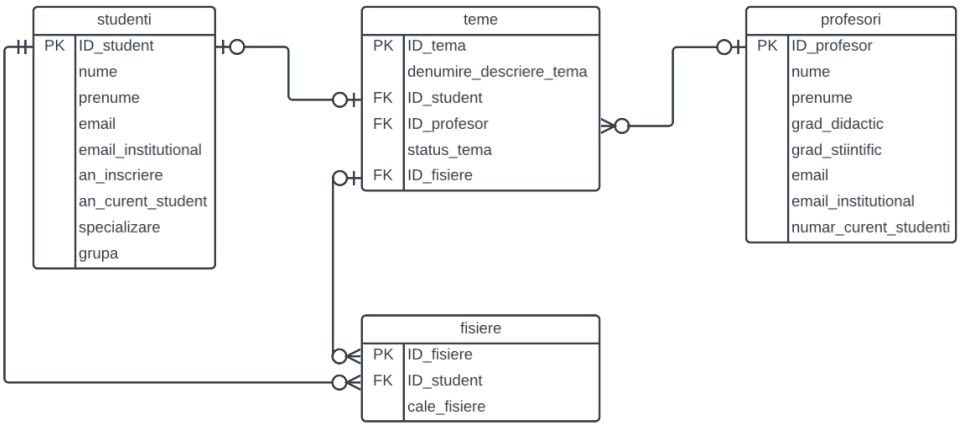

Fig. 1 – Entity Relationship Diagram (ERD)

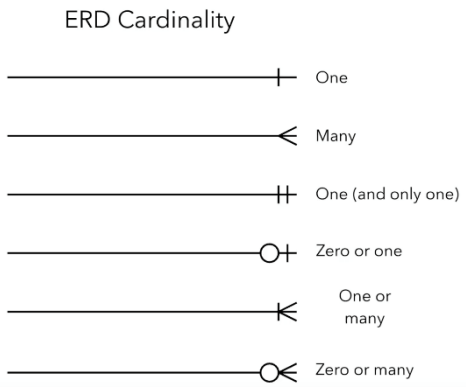

Fig. 2 – ERD Cardinality[4]

2.2. Server and routes

The construction of the server is facilitated by the use of two technologies: Node.js and Express.js. The first of these became very popular in web development, as it unified the use of a single language for both the frontend and the backend - JavaScript. Previously, it was necessary to know at least two languages in order to work in both areas of an application structure. The second (Express.js) is a framework (a library from which you can directly extract predefined / improved functionality) that complements the utility needs that a programmer / web developer may encounter. Among the advantages of using Express.js are: fast definition of Hypertext Transfer Protocol (HTTP) method-based routes; it includes various middleware modules that can perform additional tasks on client-server requests and responses; allows for quick setup when connecting to a database. A simplified case of creating a server in Node.js using Express.js and making a request between the client and the database will be presented (Fig.3).

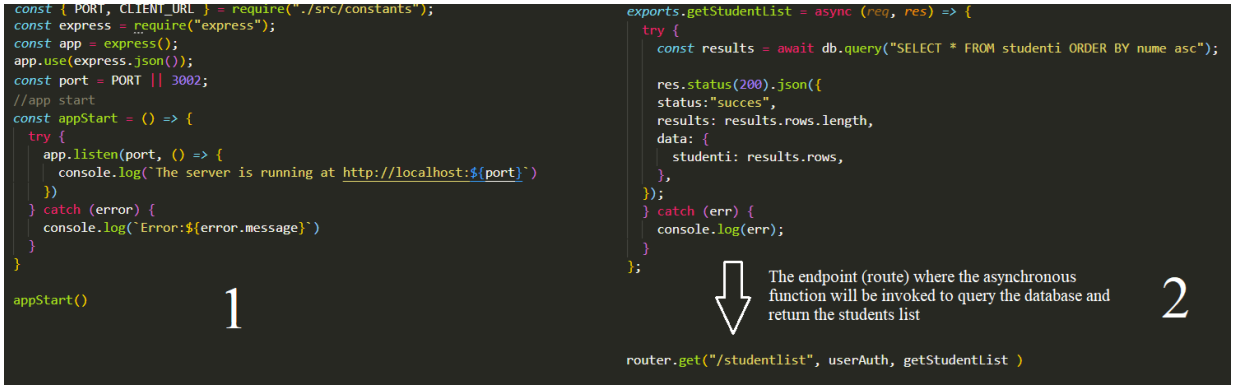

Fig. 3 - 1)creating the simplest server structure using Express.js;
2)the syntax of a server request to the database.

In Fig.3, in the area marked with 1, it can be seen that in order to create and start a server the following parameters were required: importing the express library, defining a constant running express, defining a port and using the "listen" method, integrated in a function, respectively in a "try-catch" block (a structure that knows how to manage the occurrence of errors). The "listen" method has in this case set 2 arguments: the port and a callback function (a function that runs / is "called" only after completing the function whose argument it is in). In the area numbered 2 there is an example of an asynchronous function (which allows the execution of the following lines of code, without blocking / delaying other procedures in case the response can be long-lasting), which queries the database by a query of type SELECT * FROM which returns all records in a table. The response is converted by the .json () method (which is based on the JSON.stringify () method) to a JavaScript Object Notation (JSON) object, because the server needs a data type that it can interpret. . Further processing is done on the object to return only the data of interest - the columns with students. Also, in zone 2 is presented the form of a route (endpoint) that uses the HTTP method of GET type, "/ studentlist", which can be called later by the front-end to get the desired set of results. Why is it necessary to use asynchronous functions in these cases? Because it does not block the server if it doesn't return the answer (which is known as promise) - the application continues to execute other requests even if a function fails to return.

2.3. Client and interface

The client (front-end) and the interface are the bridge between a user and the rest of the application. The key features that define the success of an application, beyond the result it offers, is the way in which it manages to mediate the communication between man and machine. In making this component, details are needed in advance about the operating points that want to be interfaced. Traditionally, websites were based on HTML, CSS and JavaScript (in its unaltered form, without the addition of other frameworks or libraries). Due to performance reasons, coding time optimization, and structuring of complex projects, different libraries (based on JavaScript) have appeared to fill the gaps of the classic model.

The React library was chosen for this project. The reasons for this were chosen as follows: 1) the structuring of the project into individual parts called components, which can exist independently of each other, and can be reused, for example: a button is built, stylized and with the functionality implemented; it can be imported into any other component without having to rebuild it. This results in a high degree of application development speed and flexibility; 2) high performance, as the core of the React lies in the existence of a Virtual Document Object Model (Virtual DOM).

The DOM is a cross-platform, language-independent application programming interface that treats an HTML, XHTML, or XML document as a tree structure in which each node is an object that is a part of

the document. Objects can be handled on a scheduled basis, and any visible changes may be reflected in the document display. [5]

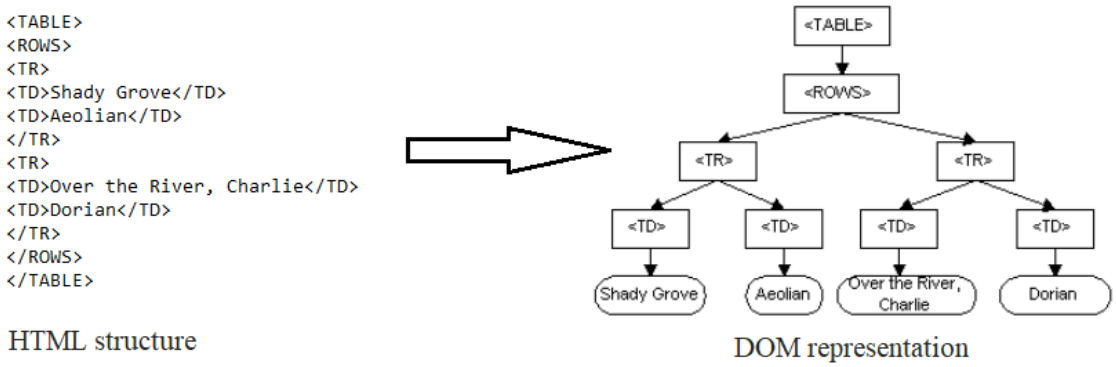

Fig. 4 – Representation of the tree structure in DOM[6]

In essence, the DOM interprets and graphically renders / displays the structure of a document, in this case HTML. The important idea behind the Virtual DOM that React has as a basis for its functionality is that it acts as a comparator with the original DOM, making the minimum number of changes (down to the lowest level), respectively the re-rendering of a component, only if it has been modified. In Fig. 5 is the differentiated rendering of components, a major factor in improving the performance of an application.

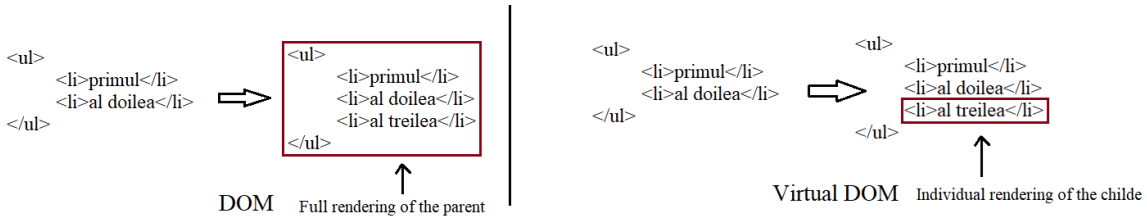

Fig. 5 – DOM vs. Virtual DOM

2.4. Authentication

The authentication process provides the security of the platform. Existing and established methods of authorizing user access were used in its creation. First, in the registration stage, an encryption method called bcrypt was used, which uses the Blowfish block cipher cryptomatic algorithm. The use of this algorithm ensures the storage of passwords in the database in an encrypted form, its reinterpretation being possible only by its decryption by the server. At the same time, JSON Web Token (JWT) was used to validate the user's identity. It consists of a header that contains the encryption method, a payload that contains information about the user trying to authenticate and a signature that contains the secret key of the server. The token will be stored in the Cookie Storage of the platform and only based on its existence the users can perform actions within the platform. A JWT is shaped like:

eyJhbGciOiJIUzI1NiJ9.(HEADER)eyJuYW1lIjoiSm9lIENvZGVyIn0.(PAYLOAD)5dlp7GmziL2QS06sZgK4mtaqv0_xX4oFUuTDh1zHK4U(SIGNATURE)

Fig.6 shows the flow chart of the user authentication process

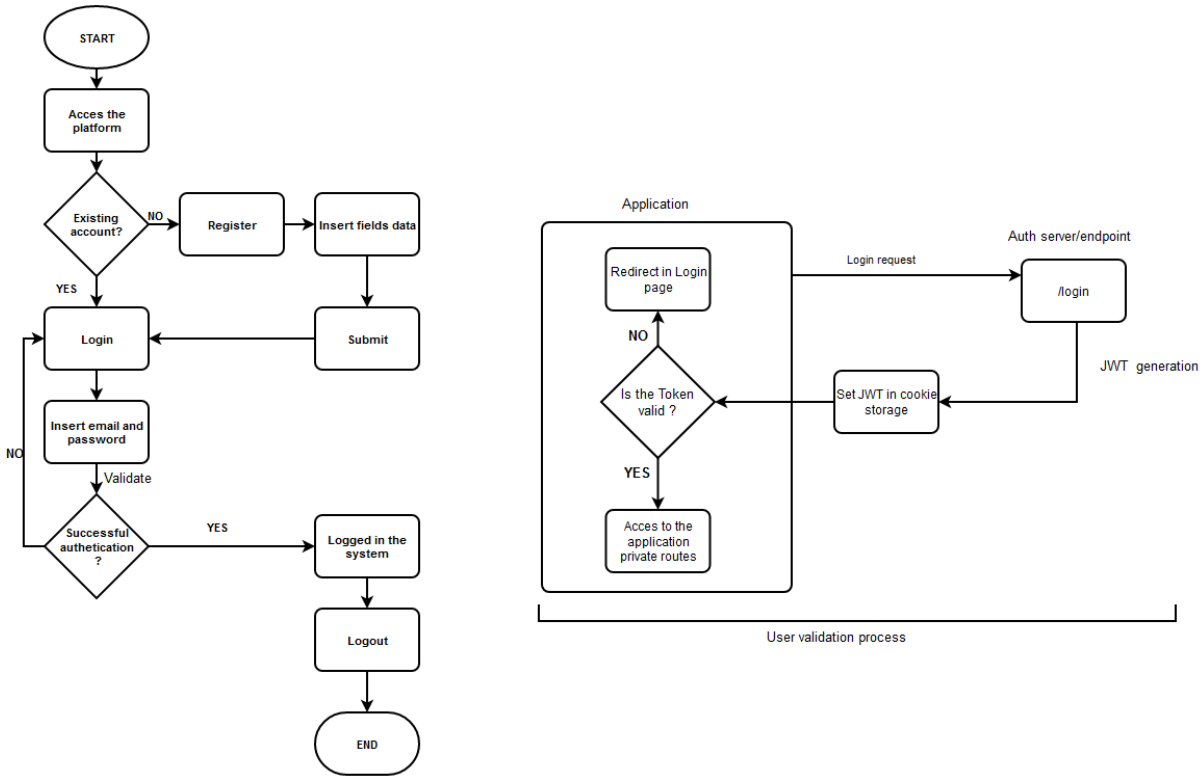

Fig. 6 - Authentication process flowchart

3. Future developments

The future directions for the application development will include methods for storing files on the server, adding rights for users (improvements to the current form of authentication) and scalability of the platform on any device, either desktop browser or mobile.

4. Conclusions

The Internet is an indispensable tool now, every field that operates on a large scale requires a means of transition / migration in the online area through one or more applications, which expose the data and processes to a wider audience, users or customers. The application aims to make the work of teachers easier, to eliminate working with multi-files (.txt, .csv, Google Docs, etc.) to keep track of data, unifying sources into a monolithic structure from which data can be entered or extracted at any time from the diploma work process.

5. Bibliography

[1]. https://ro.education-wiki.com/9050114-what-is-react [accessed on : 15.04.2022]
[2]. https://ro.wikipedia.org/wiki/Node.js [accessed on : 29.03.2022]
[3]. https://ro.education-wiki.com/5154595-what-is-postgresql [accessed on : 3.04.2022]
[4].https://stackoverflow.com/questions/54544859/er-diagram-are-the-relations-and-cardinalities-correct [accesed on : 25.04.2022]
[5] https://ro.wikipedia.org/wiki/Document_Object_Model [accessed on : 5.05.2022]
[6] https://www.w3.org/TR/WD-DOM/introduction.html [accessed on : 5.05.2022]