INFORMATION TECHNOLOGY

DEVELOPMENT OF A SERVER-SIDE WEB APPLICATION FOR STUDENT ACADEMIC STATUS MANAGEMENT: DESIGN, IMPLEMENTATION AND OPTIMIZATION

Florentina-Anica PINTEA
Daniel PURDA
Kristijan CINCAR

Abstract: *The existing traditional system for managing students' academic status, based on outdated technologies, has many shortcomings. Difficult access and knowledge of Web Design required to use it, are just some of the problems faced by the current system. Through this work it is proposed to develop a Server-Side Web application that will significantly streamline the management of students' academic status. It will provide a centralized platform where students and teachers can easily access up-to-date information about grades, events and much more, the application will also facilitate effective communication between the two parties involved in the educational process. The app will play an important role in improving access to information, being an essential tool in academic management, providing quick and accurate solutions to administrative problems.*
Keywords: *academic situation, web design, platform, educational process, students, communication*

1. INTRODUCTION

Over the last decade, the world of education has undergone unprecedented transformations imposed by the rapid advance of digital technology. Developments in recent years have been driven not only by technological progress, but also by social and economic changes, which have heightened the need to adapt education systems to new realities.

A key component of this digitization process has been the introduction of digital technologies, such as computers, tablets, software and edu-

cation apps, in kindergartens, schools and universities. These developments in educational policies on digital technologies have had a significant impact on the current state of the art in the field (Zounek, Záleská, Juhaňák, Bárta, & Vlčková, 2018), facilitating access to information, student collaboration and personalization of the educational process (Silvestre & Claros, 2023).

Technology has led to an improvement in the quality of education, enabling innovative approaches to teaching and learning (Silvestre & Claros, 2023). However, this context of digitization also entails certain challenges, such as bridging the digital divide, data security and adequate teacher training (Silvestre & Claros, 2023), (Fernández-Batanero, Román-Graván, Montenegro-Rueda, López-Meneses, & Fernández-Cerero, 2021).

Sustainable management of this digital transformation in higher education is thus essential, both to enhance the quality of education and to provide an optimal learning experience for students (Abad-Segura, González-Zamar, Infante-Moro, & Ruipérez García, 2020). Finally, the effective integration of technology in education can help to make universities more attractive, improve the training process and reduce the risk of school dropouts (Abad-Segura et al., 2020).

1.1. Context and importance of managing students' academic situation

Managing students' academic situation is a complex and essential process for their individual success as well as for the overall quality of education offered by higher education institutions. The context in which this process takes place is influenced by factors such as curricular requirements, educational policies, digital technologies, but also by the individual and collective needs of students. The structure of study programs, the mode of assessment, the complexity of subjects, all influence the pace of learning and the academic performance of students. Digital learning tools, online platforms, digital resources have radically changed the way students learn and interact with learning material (Iordan, 2015).

Current context

Digitization in education has the potential to deliver considerable benefits as technologies can support a wide range of learning modalities and pedagogical approaches. Students thrive in these digital environments, where today's technologies are shaping their expectations and their ability to access, manage and use information. The effective integration of digital technologies in higher education makes it possible to monitor student

progress and identify obstacles in the learning process, thus reducing the risk of dropout (Abad-Segura, González-Zamar, Infante-Moro, & Ruipérez García, 2020).

Students come from diverse social, economic and cultural backgrounds with different expectations and needs in terms of academic support. Institutions need to adopt personalized strategies to respond to this diversity. Modern technologies enable the automation of many aspects of academic management, such as access to information about grades, attendance, syllabus and academic performance. Online platforms (such as Moodle, Blackboard or other educational management systems) are increasingly integrated into education (Ţălu, 2020).

In a globalized environment, institutions are evaluated based on student performance and graduation rates. Good academic management contributes to improving these indicators (Upadhyaya & Vrinda, 2021).

As a similar existing solution there is the current page where you can see the academic status and fees paid (https://studenti.tibiscus.ro/) but it does not offer all the functionalities that I want integrated in the application developed in this project and the assignment of grades/information is complicated by inserting HTML tables, due to this the user must have minimal knowledge of web programming. With the solution proposed here, everything is realized by simply operating a computer/laptop.

The importance of managing the academic situation

Monitoring grades, absences and engagement helps teachers to identify struggling students early. Support measures, such as personalized tutoring or academic counseling, can then be implemented. It is important to identify the underlying causes of academic difficulties. These may be academic - difficulties in understanding certain concepts, personal - emotional, health or social - financial, family (Stăiculescu & Richiteanu Nastase, 2019).

By using centralized systems, academic information is easier to manage, which reduces the time and resources allocated to bureaucratic activities. Access to clear information and resources relevant to their academic progress enhances satisfaction and confidence in their educational institution.

Good academic management enables interventions that reduce performance disparities between students and promote a more equitable educational environment.

1.2. Common challenges

In the development of a server-side web application for student academic status management, several common challenges arise that need to be considered to ensure the success of the project. Developing a mobile-friendly web application can increase the accessibility and usability of the application. The digitization of education has been an increasingly important topic in recent years as technological advances have revolutionized the way we approach teaching and learning. A key aspect of this transformation is the massive collection and storage of various educational data, from student performance to administrative records (Danubianu, 2016). This raw data hold immense potential for generating valuable insights that can benefit both students and educational institutions (Danubianu, 2016).

To effectively harness the power of this data, researchers have proposed several solutions. One approach is the optimization of database management systems, which can significantly improve the speed and efficiency of data processing (Zheng & Hong, 2019). This includes techniques such as schema optimization, index optimization, and query performance optimization (Zheng & Hong, 2019).

Another crucial aspect is the normalization and structuring of data to eliminate redundancies and facilitate updates (Amo, Gómez, Hernández-Ibáñez, & Fonseca, 2021). By adopting modular and scalable data system architecture, educational institutions can collect and manage a wide range of data generated in both the teaching-learning and administrative processes (Amo et al., 2021). Alongside these technical solutions, the integration of advanced data analysis tools and reporting modules can provide valuable insights for decision-making (Danubianu, 2016) (Gan & Jie, 2024).

By recognizing that the process of implementing change within groups and individuals mirrors the strategic innovation processes, organizations can create a more cohesive and efficient approach to developing technologies. This can lead to a more agile, adaptable organization with a higher likelihood of successful technology adoption and application (Carlopio, 2014).

In (Nevo, Nevo, & Pinsonneault, 2015) presents a nuanced perspective on post-implementation change behaviors in relation to information technology (IT). While most research has classified these behaviors under the term “adaptation,” the authors argue that this term has become overloaded, which may limit our understanding and theorizing of such behav-

iors. This paper proposes a shift in focus towards a behavior called IT re-invention, distinguishing it from adaptation and calling for new theoretical frameworks to understand it.

1.3. Motivation for choosing the topic

The importance of this theme is evident in the context of the need for modernization and optimization in educational processes. The reason for the choice of this theme is the significant impact it will have on efficiency. The methodology used involved a combination of theoretical and practical research, analysis of existing solutions and prototype testing. Among the many difficulties encountered were managing the diversity of requirements and data integration. Compared to the existing solution, StudentApp (the proposed theme) is distinguished by its modularity and flexibility, being able to adapt to specific needs.

The stages taken in the research were:

- The existing application was analyzed, identifying its functionalities, strengths and weaknesses.
- Making the detailed design of the application, including the database, the graphical interface, the navigation flows and the main functionalities.
- The application was developed using the Ruby on Rails Framework.

2. TECHNOLOGIES USED

The technologies used for the development of a Student App dedicated to grade management can be described in more detail in terms of main components and functionalities. Such an application involves the use of a diverse set of technologies and tools.

2.1. Ruby on Rails (Bächle & Kirchberg, 2007) (Ruby, Thomas, & Hansson, 2023)**.** Also known as Rails, it is a popular, open source, server-side web framework built on the Ruby programming language. Released in 2004 by David Heinemeier Hansson (DHH), Rails gained popularity quickly because it is a simple and efficient approach to rapid web application development. It offers a wide range of features that make it an excellent choice for project management.

2.2. HotWire - HTML Over The Wire (Bruun , 2023) - is a technology developed by the team behind the ROR Framework, it is a modern

approach to web application development, helping ROR to re-enter the list of favorite frameworks, reducing the dependency on traditional frontend frameworks based on the JavaScript programming language, such as React or Angular.

2.3. Tailwind CSS (Gerchev, 2022) is a utility-oriented CSS framework, providing a set of predefined utility classes that facilitate web design development. These classes improve consistency and capability, allowing quick styling without writing custom CSS code. Because this framework focuses on functional classes rather than traditional CSS elements, web application developers can create fast and visually appealing interfaces with much less work. It is a very popular framework because of its innovative method of styling web interfaces, and it is more performant and easier to load than Bootstrap.

2.4. Playwright (Microsoft, 2024) is an open-source automation library for testing web pages developed by Microsoft and released on January 31, 2020. It quickly became popular among programmers and web application developers.

It has been created specifically to meet the needs of end-to-end testing, supports all modern rendering engines, including Chromium, WebKit and Firefox, offers the ability to automate tasks in browsers.

The tests are written in the JavaScript programming language, as it is the native language for Node.js on which Playwright is based. Although JavaScript is the recommended primary language, Playwright can also be used with other languages through third-party libraries. For example, there are libraries for Python, TypeScript(in my application I used the TypeScript variant) and C#.

2.5. Rspec și Rubocop (Rappin, 2021)

Rspec is a testing tool for Ruby, designed for behavior-based programming (a method of software development that focuses on the desired behavior of the application instead of the raw functionality). It is the most widely used testing library for Ruby in production applications. It provides an expressive and easy-to-understand DSL domain-specific language that allows tests to be written in a natural language-like way. The tests written in Rspec are organized in a hierarchical structure, using blocks and contexts which allows for better organization and a clearer understanding of the relationships between the different parts of the code under test.

Rubocop is a Ruby style and code format checker based on the Ruby

style guide developed by the Ruby community. It is extremely flexible, most aspects of its behavior can be adjusted through various configuration options. At the beginning Rubocop supported a safe style and individual checks could not be disabled, but as it was very difficult to find a set of values that would satisfy everyone, new features were added.

2.6. Git și Github (Tsitoara, 2020)

Git is a free, open source, distributed, open source version control system designed to manage any kind of project, large or small, whether in Ruby or JavaScript. It is easy to learn, has extremely fast performance, and is used to coordinate work across software development teams. It was created in 2005 by Linus Torvalds originally to develop the Linux kernel.

GitHub is an online web platform that facilitates real-time collaboration. It is ideal for teams of programmers writing code, building web applications or any other project. It's useful in the development stage whether it's software, programming, content or research, you can easily track changes to the project as well as navigate through existing versions.

3. REQUIREMENTS ANALYSIS AND APPLICATION DESIGN

3.1. Charts and patterns used

A normalized database has been implemented, as atomic as possible, each table to be with a specific purpose and well-defined relationships, without redundant columns for better efficiency. The database has been designed to manage the academic information of the students of a university educational institution, its structure includes several interconnected tables with standard attributes but which can be modified or scaled according to requirements and needs.

To realize the schema of the database, and to think about its functionality, the classic pen and sheet combo was used, and later after being satisfied with the result we used "dbdiagram.io" to make a digital representation of the database.

In order to establish meaningful connections between entities in the database, three fundamental types of relationships have been implemented: A record in the user table is associated with a single record in the student table, teacher, or admin. A record in the student table can be associated with multiple records in the grades table.

In order to establish meaningful connections between entities in the database, three fundamental types of relationships have been implemented:

A record in the user table is associated with a single record in the student table, teacher, or admin. A record in the student table can be associated with multiple records in the grades table. A record in the teacher table can be associated with multiple records in the subject table, and vice versa.

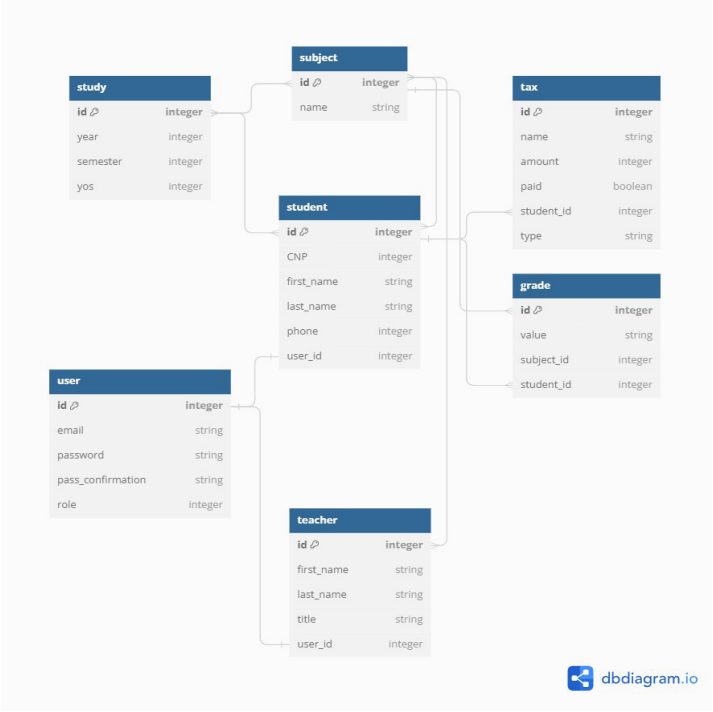

Fig. 1 Database diagram

The benefits of using relationships have been ensuring data consistency and accuracy by restricting how entities can be associated, optimizing queries and reducing data redundancy, as well as allowing complex modeling of relationships between entities.

3.2. App deployment

After finalizing the database model and establishing it, the actual implementation of the application was achieved. This stage involved installing all of the Ruby on Rails dependencies listed above, generating code for the models, controllers, view pages, and paths for each action.

3.3. Graphical Interface

Subsequently, attention was focused on the development of the user interface (UI). A harmonious color palette was chosen, representa-

tive logos were created and an intuitive and visually pleasing interface was designed, easy to navigate. The result was the first version of the application, shown in the following figures.

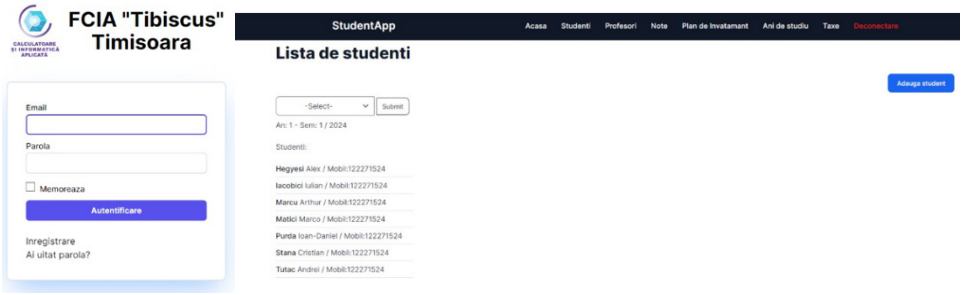

Fig. 2 Login and view the student list

After a detailed analysis of the literature, a thorough research and testing of various color palettes and combinations, we have developed the final version of the application presented in this paper.

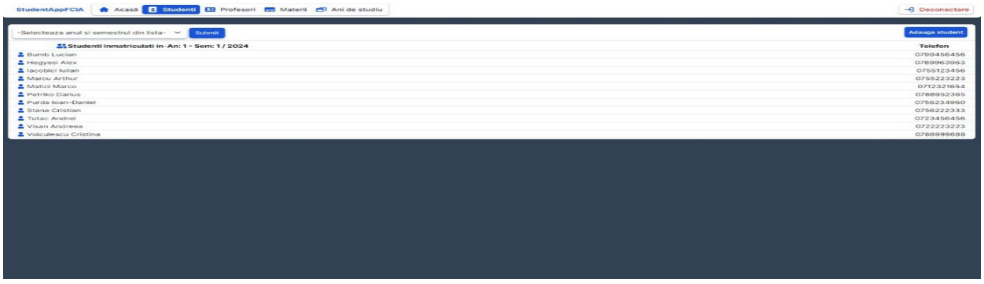

Fig. 3 New student list page

The new design of the graphical interface demonstrates a significant improvement in terms of attractiveness, structure and presentation of information, a more modern and harmonious color palette is observed, replacing the previously more monotonous appearance, the presence of additional graphic elements, such as icons and decorative elements.

3.4. Development of backend components

To increase the speed and fluidity of the management interface, we implemented HotWire technology, using Turbo Frames. This approach allows you to update the details of a selected student from the displayed list, replacing only a specific frame on the page, instead of reloading the page entirely.

In addition to optimizing student detail pages, we've extended the benefits of Turbo Frames to other sections of the admin interface as well:

- Teacher and subject details pages: Similar to students, teacher and subject details are displayed by refreshing specific frames, eliminating the need to completely reload the page.
- Student grade add page: Regardless of the action performed (add, edit or delete), student grade operations are executed via Turbo Frames, providing a smooth and fast experience.

When selecting a teacher, their data will be displayed in the right frame without the rest of the page changing.

Integration of the prawn plugin

One of the desired functionalities in the application was the possibility to download a student's situation electronically. The implementation of this feature was achieved by using the Prawn gem, which we identified as an optimal solution after a detailed analysis of the resources available online.

Prawn facilitates the generation of PDF documents, allowing in this case the export of students' academic status. The implementation of this gem as a solution has eliminated the need to manually create electronic files with this data, significantly reducing the amount of repetitive manual work. The code below shows the primary method of generating the PDF file.

```
def generate_pdf(student)
  @student = student
  pdf = Prawn::Document.new
  add_header(pdf)
  add_student_information(pdf)
  add_grades_table(pdf)
  add_footer(pdf)
  pdf.render
end
```

Fig. 4 The main method generate_pdf

As you can see, within the main generate_pdf method, two variables are initialized: @student and pdf. The @student variable stores the object containing the student data, while pdf is used to create a new PDF document using the Prawn library.

Subsequently it is observed that several methods are called which receive the above initialized pdf variable as parameter, these methods add elements to the PDF file. Finally, after including all the elements, a final pdf object is created and finalizes the generation of the document, preparing it for saving or viewing.

In conclusion the above method is a centralized function that coordinates the creation of the PDF document with the student's information.

3.5. Testing application functionality

The application uses an authentication system based on user accounts with well-defined roles. Each role has access to a specific set of information and functionality. Within this application, we have created three main roles: admin, teacher and student.

The code shown below defines an "enum" called "role" with three possible values (admin, teacher and student). An enum is a data type that restricts the possible values of an attribute to a predefined list.

Finally, the role enum simplifies the code maintenance process. If roles need to be added or removed in the future, the modification is done centrally by updating the enum definition. This centralization eliminates the need to search and modify code in multiple locations, reducing the risk of errors and simplifying the application maintenance process.

Gaining access to the application is conditional on the prior creation of a user account. This account can be generated by navigating to the dedicated 'Register' page. Users will be guided through a simple data-filling process, similar to most online platforms.

The user will be asked to enter a valid e-mail address, password and password confirmation. Once entered, by pressing the "Register" button, the system will check the validity of the e-mail address and the absence of an existing account associated with it. In case of a valid e-mail address and no account, the account will be created and the user will be redirected to the login page.

In addition to the registration and authentication functionalities, we have also implemented a password reset functionality, which is essential in situations where users forget their passwords. On the login page, under the "register" link, there is a "Forgot your password?" link. Clicking on this link redirects the user to a page with a form asking the user to enter the e-mail address for which they want to reset their password.

Once the e-mail address is entered and the "Reset my password!" button is clicked, the application will check whether the e-mail address is associated with an existing account in the database. If so, an e-mail will be sent to that address, containing a unique link to a dedicated password reset page.

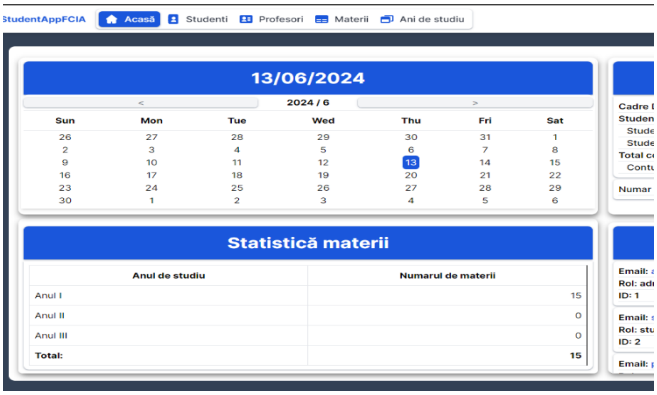

Fig. 5 User admin homepage

The application contains:

Functionality for the user with the role "teacher, having the possibility to View contact information, View assigned subjects, View assigned subject details page, Assign grades to students for assigned subjects, Modify personal data.

User functionalities for the "student" role, with the possibility to View contact information, View marks obtained in exams/exams, Download the student's school situation in PDF format, Modify personal data.

Fig. 6 User authenticated with the role "student"

4. CASE STUDY AND EVALUATION OF THE APPLICATION

4.1. Application of the solution in a real/simulated academic environment

To evaluate the efficiency and functionality of the developed application, it has been implemented and tested in the Faculty of Computer Science

and Applied Informatics of Tibiscus University of Timisoara.

The tests were performed for the Bachelor of Computer Science degree program, providing a real and controlled environment for observing the solution performance.

The application was populated with data relevant to approximately 80% of students enrolled in the undergraduate program (Fig. 9). This included information about Personal data (name, surname, CNP, address, telephone), Academic data (grades, credit standing), Administrative data (payment statuses, administrative requests).

The data was entered manually and, for the final implementation, it was proposed to import the data from existing sources via CSV or Excel files.

4.2. Evaluation of the application by users (teachers, students)

The system was tested to evaluate the response time to search, update and report generation operations using data from over 80% of the students. The behavior of the application was tracked as the database was expanded by adding new records (Fig. 10).

The application was evaluated by the secretariat and teachers to check the user-friendliness and clarity of functionalities. Users of the application (administrative staff, teachers, students) had differentiated access to the functionalities based on the defined access level. Protection measures against unauthorized access were evaluated, such as user and password authentication and encryption of sensitive data.

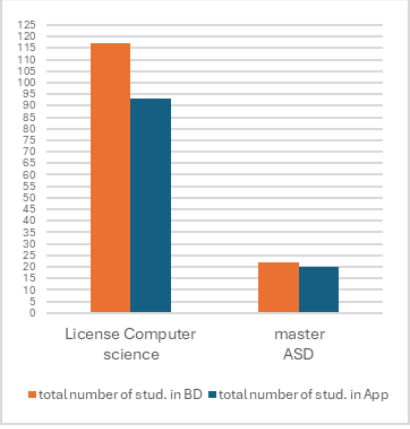

Fig. 7 Total number of stud. in App

Fig. 8 Evaluation

4.3. Analysis of feedback and suggestions for improvement

Administrative staff appreciated the intuitive interface and the speed of data processing.

Teachers noted the ease of grade management and easy access to students' academic history. It was suggested to optimize the interface for mobile devices to improve accessibility. Integration of an additional module for student feedback on courses.

Testing the application in real academic environment demonstrated its viability and usefulness in managing academic processes. The implementation revealed both strengths and areas for improvement. The results obtained support the use of the application as an effective solution for academic situation management, with potential for extension and customization for other faculties or academic institutions.

5. CONCLUSIONS AND PERSPECTIVES

StudentApp revolutionizes the management of academics, that in our faculty by offering a centralized and intuitive platform dedicated to students and teachers. By integrating modern technology, the app significantly enhances the educational experience, contributing to faster, more transparent and efficient administration of academic processes.

The key benefits of StudentApp are for students

- Grades, credits and academic standing are available in one place.
- Students can submit requests to administration and communicate directly with professors through the platform.
- Schedules, deadlines and events are centralized and easily accessible.

The key benefits of StudentApp are For teachers

- Teachers can enter and update grades quickly, with the ability to generate customized reports.
- Access to students' academic history for more informed assessment.
- Ability to send messages and notifications directly through the app.

StudentApp not only simplifies academic management, but also contributes to a more interactive and personalized educational experience. By using this solution, academic institutions can improve operational efficiency, while students and teachers benefit from a modern platform that makes their daily activities easier. In this way, **StudentApp** becomes a catalyst for

the modernization of educational processes, facilitating better collaboration between all stakeholders.

REFERENCES

Abad-Segura, E., González-Zamar, M.-D., Infante-Moro, J. C., & Ruipérez García, G. (2020). Sustainable Management of Digital Transformation in Higher Education: Global Research Trends. Sustainability, 12(5), 2107. doi:https://doi.org/10.3390/su12052107

Amo, D., Gómez, P., Hernández-Ibáñez, L., & Fonseca, D. (2021). Educational Warehouse: Modular, Private and Secure Cloudable Architecture System for Educational Data Storage, Analysis and Access. Applied Sciences, 11(2), 806. doi:https://doi.org/10.3390/app11020806

Bächle, M., & Kirchberg, P. (2007). Ruby on Rails. IEEE Software, vol. 24, no. 6, pp. 105-108. doi:https://doi.org/10.1109/MS.2007.176

Bruun , H. (2023). Hot-wire Anemometry: Principles and Signal Analysis. Oxford University Press, ISBN 1383029083.

Carlopio, J. (2014). Implementing Innovation and Technical Change. Innovation: Organization & Management, p. 17-19. doi:https://doi.org/10.5172/impp.1998.1.1.17

Danubianu, M. (2016). First steps towards curriculum development decisions from raw educational data. International Conference on Development and Application Systems (DAS), pp. 281-286. doi:https://doi.org/10.1109/DAAS.2016.7492587

Fernández-Batanero, J. M., Román-Graván, P., Montenegro-Rueda, M., López-Meneses, E., & Fernández-Cerero, J. (2021). Digital Teaching Competence in Higher Education: A Systematic Review. Education Sciences, 11(11), 689. doi:https://doi.org/10.3390/educsci11110689

Gan, Y., & Jie, B. (2024). Research on Optimization of Student Management Evaluation System under the Background of Big Data. doi:https://doi.org/10.1155/2022/2917450

Gerchev, I. (2022). Tailwind CSS: Craft Beatiful, Flexible, and Responsive Design. eBook ISBN 978-1-925836-51-6.

Iordan, M. (2015). Îmbunătățirea managementului universitar. Preluat de pe https://uefiscdi.gov.ro/resource-824527-14-imbunatatirea-managementului-universitar-2015-ro.pdf

Microsoft. (2024). Playwright enables reliable end-to-end testing for modern web apps. Preluat de pe https://playwright.dev/: https://playwright.dev/docs/intro

Nevo, S., Nevo, D., & Pinsonneault, A. (2015). Differentiating Post-implementation Change Behaviors towards Information Technology. 48th Hawaii International Conference on System Sciences (pg. 4928-4937). Kauai, HI, USA: IEEE. doi:https://doi.org/10.1109/HICSS.2015.584

Rappin, N. (2021). Modern Front-End Development for Rails. Pragmatic Bookshelf. Preluat de pe https://books.google.ro/books?id=mYFGEAAAQBAJ&newbks=1&newbks_redir=0&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

Ruby, S., Thomas, D., & Hansson, D. (2023). Agile Web Development with Rails 4. Preluat de pe The pragmatic programmers.

Silvestre, R. E., & Claros, M. P. (2023). El papel de la tecnología para la mejora de la calidad educativa. Ciencia Latina Revista Científica Multidisciplinar, 7(3), 1018-

1027. doi:https://doi.org/10.37811/cl_rcm.v7i3.6245

Stăiculescu, C., & Richiteanu Nastase, E. R. (2019). University dropout. Causes and solution. Mental Health: Global Challenges Journal, 1(1), 71–75. doi:https://doi.org/10.32437/mhgcj.v1i1.29

Tsitoara, M. (2020). Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management. Apress. doi:https://doi.org/10.1007/978-1-4842-5313-7

Ţălu, Ş. (2020). New Perspectives in the Implementation of Smart-Technologies in Higher Education. Proceedings of the 2nd International Scientific and Practical Conference "Modern Management Trends and the Digital Economy: from Regional Development to Global Economic Growth" (MTDE), 253-257. doi:https://doi.org/10.2991/aebmr.k.200502.042

Upadhyaya, P., & Vrinda. (2021). Impact of technostress on academic productivity of university students. Education and Information Technologies,26, 1647–1664. doi:https://doi.org/10.1007/s10639-020-10319-9

Zheng, H., & Hong, G. (2019). Design and Implementation of Campus Service System Platform Based on Big Data. IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), New York, NY, USA, pp. 299-302. doi:https://doi.org/10.1109/CSE/EUC.2019.00064

Zounek, J., Záleská, K., Juhaňák, L., Bárta, O., & Vlčková, K. (2018). Studia paedagogica: Digital Youth and Their Ways of Learning, Vol.23,No.4. Czech Republic and Norway on Their Path to Digital Education. doi:https://doi.org/10.5817/sp2018-4-2

Notes on the authors

Florentina-Anica PINTEA, PhD, Associate Professor at the Faculty of Computers and Applied Informatics, Tibiscus University of Timisoara, Head of Multidisciplinary Informatics Research Center UTT, Institute of Psychotherapy, Psychological Counseling, and Clinical Supervision, email: florentina.pintea@tibiscus.ro

Daniel PURDA, Master's Student at the Faculty of Computers and Applied Informatics, Tibiscus University of Timisoara

Kristijan CINCAR, PhD is University assistant at the Faculty of Computers and Applied Informatics, Tibiscus University in Timisoara