PD.TVZ.HR
POLYTECHNIC & DESIGN Vol. 13, No. 3, 2025.
Mrežna inačica: ISSN 2459-6302
Tiskana inačica: ISSN 1849-1995
DOI: 10.19279/TVZ.PD.2025-13-3-18

APLIKACIJA ZA GENERIRANJE ISPITNIH ZADATAKA ZA AUTOMATSKO BODOVANJE U LMS SUSTAVU

APPLICATION FOR GENERATING EXAM TASKS WITH AUTOMATIC GRADING IN LEARNING MANAGEMENT SYSTEM

Danko Ivošević, Aleksandar Stojanović

Zagreb University of Applied Sciences, Vrbik 8, 10000 Zagreb, Croatia

SAŽETAK

Provođenje provjera znanja u obrazovnim i visokoškolskim okruženjima jedno je od temeljnih aktivnosti takvih okolina prilikom provođenja nastavnog procesa. S obzirom na postojanje digitalnih sustava za učenje (engl. *learning management system - LMS*) uobičajeno je da se neki broj provjera znanja provodi putem računala umjesto klasičnim pisanjem olovkom na papiru ili usmenim ispitivanjem te postoji stvarna potreba automatiziranja tog dijela nastavnog procesa. U ovom radu se opisuje implementacija prototipa aplikacije kojoj je svrha brzo nadopunjavanje baze pitanja prema prethodno definiranim tipskim skupinama pitanja s varijacijama u sadržaju ulaznih podataka koji služe pojedinim vrstama pitanja, i gdje se podrazumijeva da se pitanja i provjere znanja LMS sustava automatski boduju. Daje se kratki pregled vrsta i primjera generiranih pitanja s njihovim specifičnostima ovisno o sadržaju zadatka.

Ključne riječi: *Moodle, LMS, baza pitanja, automatsko bodovanje, vrste pitanja, provjera znanja, XML, Python.*

ABSTRACT

Conducting knowledge assessments in educational and higher education settings is a fundamental activity within the teaching process. With the advent of digital learning systems (learning management systems - LMS), it is now common for some knowledge assessments to be conducted via computer rather than traditional pen-and-paper or oral examinations, creating a genuine need to automate this aspect of teaching. This paper describes the implementation of a prototype application designed to quickly update the question database according to predefined question types with variations in the content of input data serving each type. It is assumed that the questions and knowledge assessments within the LMS are automatically scored. A brief overview of the types and examples of generated questions is provided, highlighting their specific features depending on the content of the task.

Keywords: *Moodle, LMS, question bank, automatic grading, question types, quiz, XML, Python.*

1. UVOD
1. INTRODUCTION

During the implementation of semester teaching activities, there is a constant need to update course content. In addition to regular reviews and updates of the syllabus, lecture, seminar, and laboratory content, there is also a continuous need to develop new materials for knowledge assessments. It is necessary to design questions and tasks for mid-terms and final exams.
When using a Moodle-based LMS, knowledge assessments can be conducted on the computer using questions that have previously been stored in the question bank, which is an integral part of such a system. Furthermore, there is a possibility to create questions that the system will automatically grade [1].

2. POVEZANI RADOVI
2. RELATED WORK

The topic of automating exam question generation in the context of digital learning systems (Learning Management Systems – LMS) appears in scientific and professional literature within the domain of educational technology and computer-supported learning. The recognition of the need for effective, repeatable, and scalable mechanisms for designing knowledge assessments results from the increasing number of LMS users in formal educational institutions as well as non-formal education settings. Moodle, as an open-source LMS, is widely accepted and used in educational institutions at various levels. Thanks to extensive documentation and extensibility support, Moodle enables automated grading through various types of questions, including multiple choice, short answer, numerical answer, and questions based on regular expressions and embedded (cloze) questions [2]. Each type has a specific question structure, and consequently, a particular export notation that allows for programmatic generation and the import of large numbers of questions from external sources.

It can be said that implementing automated support for the management of teaching processes based on the Moodle LMS platform, for populating the question bank, generally consists of several aspects or tasks: automatic generation with parameterization and random content generation, individualized generation of exam versions, storage and export of the question bank contents in different file formats, using various programming language technologies [3], [4], [5], [6], [7]. XML notation, commonly used for import and export, allows tools developed in programming languages like Python, with XML manipulation support (for example via the xml.etree.ElementTree module [8]) and graphical interfaces such as TkInter [9], to create accessible standalone solutions usable without an additional server or Moodle integration. The advantage of such approaches is that they give teachers complete control over content design, while simultaneously enabling flexibility and extensibility in accordance with the specific needs of their teaching process.

This paper presents an application aimed at generating sets of thematic or typical questions to populate the LMS question bank. Simple mathematical problems, which should be sufficiently illustrative for explaining the concept, were chosen as examples of question content. The properties of question types constituting the LMS question bank and the application implementation will be shown, along with a brief description of the Moodle XML format [10] used for exporting and importing questions from external sources into the LMS question bank and examples of generated questions. The use of *Calculated* question types, which utilize formulas and randomly generated variables to create unique simple mathematical problems at each test run, was deliberately omitted because the purpose of the developed prototype is not to address the domain of mathematical problems. The goal of the developed application is to create a starting point and demonstrate the extensibility of the implementation for applying to more complex problem tasks that are not necessarily based on simple calculations. Likewise, the existence of VPL support for creating questions with more complex tasks, which would be solved algorithmically, is a separate possibility to cover knowledge assessment requirements that include verifying programming skills of students or learners, which is not the focus of this work [3], [11].

Within this context, the proposed prototype from this paper positions itself as a lightweight, efficient, and easily extensible alternative whose functionality is aligned with the real needs of teachers in everyday educational work, and is especially suitable for courses that involve a large number of students and frequent knowledge assessments.

3. LMS BAZA PITANJA I MOODLE XML OBLIK
3. LMS QUESTION BANK AND MOODLE XML FORMAT

The LMS implementation at TVZ is located at http://lms-2020.tvz.hr/ and all shown examples will relate to the use of that platform. The platform supports maintaining and using a question bank for each course.

3.1. PREGLED VRSTA PITANJA
3.1. QUESTION TYPE OVERVIEW

According to Moodle documentation, there are as many as 17 different types and subtypes of questions that can be used to design a knowledge assessment. Among these, a few that have been used and are relevant to the implementation presented in this paper will be highlighted. These are:

- *Multiple Choice*, referred to in the Moodle XML description by the keyword *multichoice*,
- *Short Answer*, referred to in Moodle XML description by the keyword *shortanswer*,
- *Regular Expression Short-Answer*, referred to in Moodle XML description by the keyword *regexp*,
- *Numerical*, referred to in Moodle XML description by the keyword *numerical*,
- *Embedded Answers*, referred to in Moodle XML description by the keyword *cloze*.

All these question types imply the possibility of automatic grading, which significantly eases the teacher's work, especially for courses with a large number of students and insufficient teaching staff.

The key attributes or question settings used when entering new questions into the LMS question bank are:

- Question title or name,
- Question text,
- General feedback,
- Expected answers which can be partially or fully correct or incorrect depending on the question type.

The question author may optionally enter general feedback as information displayed after a student answers the question.

3.2. MOODLE XML OBLIK
3.2. MOODLE XML FORMAT

The description of the XML format is widely known as one of the standards for marking or structuring data stored in appropriate files, and the Moodle XML format has its predefined tags used in structuring the question descriptions.

For example, for a *short answer* question involving a simple arithmetic calculation, the export functionality to an appropriate file provides the source code of the record for that question shown in Listing 1.

```
<question type=»shortanswer»>
  <name>
   <text>Zbrajanje 2+2 - Kratki odgovor</text>
  </name>
  <questiontext format=»html»>
   <text><![CDATA[<p>Koliko je 2+2?</p>]]></text>
  </questiontext>
  <generalfeedback format=»html»>
   <text><![CDATA[<p>2+2 = 4</p>]]></text>
  </generalfeedback>
  <defaultgrade>1.0000000</defaultgrade>
  <penalty>0.3333333</penalty>
  <hidden>0</hidden>
  <idnumber></idnumber>
  <usecase>0</usecase>
  <answer fraction=»100» format=»moodle_auto_format»>
   <text>4</text>
   <feedback format=»html»>
    <text></text>
   </feedback>
  </answer>
</question>
```

Ispis 1 *Primjer Moodle XML opisa pitanja s kratkim odgovorom.*

Listing 1 *Example of Short-Answer question Moodle XML description.*

The *type* attribute of the root element question with the keyword *shortanswer* defines the question type. The section marked with name contains the question title; the section marked *questiontext* contains the question text; and the section marked *generalfeedback* contains the general feedback.

The section marked *answer* represents the expected correct answer. Correct answers are generally described with multiple attributes:

- The *fraction* attribute as the percentage correctness of the answer,
- The *text* tag as the text or content of the offered answer,
- The *feedback* section as feedback for the particular offered answer.

The Moodle system in the dropdown menu offers a range of predefined percentage values based on standard divisions of the whole from 100% to 10 equal parts, which can be used in defining the correctness percentage of each answer.

3.3. POSEBNOSTI POJEDINIH VRSTA PITANJA
3.3. SPECIFICITIES OF INDIVIDUAL QUESTION TYPES

Listing 1 shows a *short answer* question asking for the result of a simple operation (e.g., 2+2), and the system automatically grades the correct input ("4"). In *multiple answer* questions, the user chooses from several possible answers, where one or more may be correct. Options exist to shuffle the order of answers and distribute points among correct and incorrect answers. A *short answer question based on regular expression* allows accepting different answer variants, e.g., with spaces, increasing tolerance to typing errors. *Numeric questions* allow setting an error tolerance, useful for decimal calculations. *Embedded answer questions* (cloze) combine multiple question types in one. Syntax like {1:SHORTANSWER_C:=6} defines correct answers, grading method, and answer positions in the displayed question text.

4. OPIS APLIKACIJE
4. APPLICATION DESCRIPTION

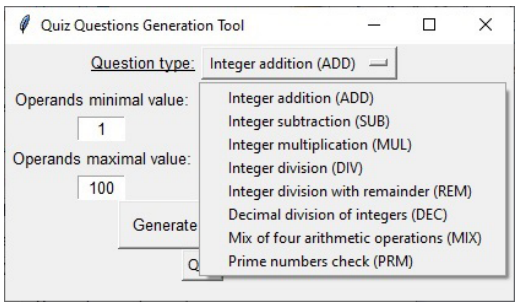
The basic idea of the developed application is to have the ability to quickly generate an arbitrarily large set of questions for the Moodle LMS question bank. The sequence of actions or processes that occur during the use of the application is as follows:

- On the user side of the application, the type of question and accompanying parameters are selected, which are loaded by the application.
- In the background part of the application, a set of questions is generated according to the defined requirements shaped by the preset parameters.
- The generated set of questions is saved into a Moodle XML format file ready for import into the Moodle LMS course system.

4.1. KORISNIČKO SUČELJE
4.1. USER INTERFACE

The user interface of the application, which consists of options for selecting important question characteristics (Figure 2):

- The lowest and highest allowed value of input operands represented by the labels *Operands minimal value* and *Operands maximal value* and the corresponding text fields below them.
- Selection of the type of tasks that will be the content of the generated questions.



Slika 2 *Izbor vrste pitanja.*

Figure 2 *Question type selection.*

The mathematical tasks that can be the content of the question set are as follows:

- basic arithmetic operations between two integers under the first three items: addition represented by the item "Integer addition (ADD)", subtraction represented by the item "Integer subtraction (SUB)", multiplication represented by the item "Integer multiplication (MUL)", as *short answer questions*,
- integer division of two numbers where there is no remainder – numbers are generated to be multiples of each other represented by the item "Integer division (DIV)" as a *short answer question*,
- integer division of two numbers with remainder represented by the item "Integer division with remainder (REM)" as a *short answer question based on regular expression*,
- decimal division of two integers represented by the item "Decimal division of integers (DEC)" as a *numerical answer question*,
- a series of basic arithmetic operations which are concatenated represented by the item "Mix of four arithmetic operations (MIX)" as a *cloze question with embedded answers*,
- marking prime numbers within the set of integer values represented by the item "Prime numbers check (PRM)" as a *multiple-choice question*.

Assuming the selection of options as in Figure 2, clicking the button to start generating questions will create a set of questions with tasks of the chosen type.

4.2. PRIMJERI GENERIRANIH PITANJA
4.2. GENERATED QUESTIONS EXAMPLES

Starting the generation of the desired set of task types will create an XML file according to the notation expected for import into the Moodle system. The file will consist of a set of questions containing the generated tasks.

Question with regular expression and variations of the correct answer

For example, one task with division of numbers with a remainder will be generated in the form of a question as in Listing 3.

Under the comment and in the question title, the question index is indicated, which in this case is 4. The question is of the *short answer type based on a regular expression*, with the attribute *type* set to the value "regexp." In this regard, there are two sections with the correct answer. Since it is an operation of dividing the numbers 534 and 383, the correct answer is that the quotient of the division is 1, and the remainder is 151, so the first correct answer option is the pair of these numbers separated by a comma, i.e., "1, 151." This is expected as the most accurate answer considering that the question text specifies that "answers must be separated by a comma." Considering that different variations of the correct pair of answer values are possible with spaces inserted before, between, and after, the second option for a fully or 100% correct answer specifies a regular expression allowing these variations, namely "( )?1( )?, ( )?151( )?."

```
<!-- question: 4 -->
 <question type=»regexp»>
  …
  <questiontext format=»html»>
   <text>&lt;p&gt;Koliki su količnik i ostatak dijeljenja
534/383? (odgovore je potrebno odvojiti zarezom)&lt;/
p&gt;</text>
  </questiontext>
  …
  <answer fraction=»100»>
   <text>1, 151</text>
   <feedback format=»html»></feedback>
  </answer>
  <answer fraction=»100»>
   <text>( )?1( )?,( )?151( )?</text>
   <feedback format=»html»></feedback>
  </answer>
</question>
```

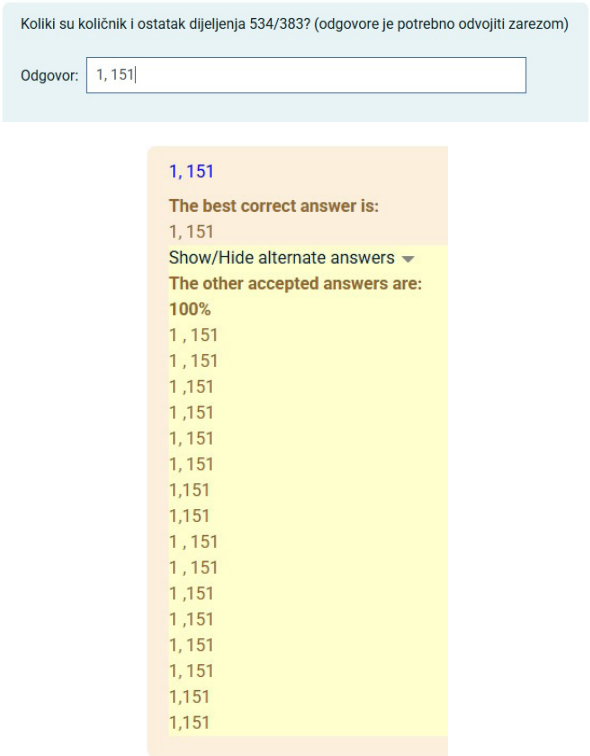
Ispis 3 *Primjer Moodle XML opisa pitanja s kratkim odgovorom temeljenom na regularnom izrazu.*

Listing 3 *Example of Short-Answer Regular Expression questionMoodle XML description.*

Appearance of the generated question in the Moodle system after importing the XML file with the question record from Listing 3 into the LMS system, viewed from the user's or the question taker's perspective, is shown in Figure 4.

It is visible that the title and text of the question correspond to what is stated in Listing 3, and the question is ready for solving or answering like any other question.

After entering the answer and submitting the completion of the knowledge test, feedback is received on the accuracy of the given answer. It can be seen that in cases of "messy" entry of the correct answer with many spaces entered in the answer field, the system lists different options of fully acceptable answers in the correct answer comment.



Slika 4 *Opcije točnog odgovora kod pitanja s kratkim odgovorom temeljenom na regularnom izrazom.*

Figure 4 *Correct answers option in Short-Answer Regular Expression question.*

The possibility to specify multiple variations of the correct answer is also available in s*hort answer questions* without the use of regular expressions, as well as in *numerical answer question* where the specification interface explicitly lists options for acceptable answers with different percentages of

grading. Additionally, this possibility is inherently built into *multiple-choice questions* where there can be more than one correct answer, and here it is shown on a *short answer question based on a regular expression*, which may be the most complex for question creators and at the same time the form with the widest possibilities for entering variations of the correct answer.

Question with embedded answers and general feedback

The question consists of four given and connected different arithmetic operations in sequence where the result of the previous operation represents one of the inputs or operands for the next and so on until the last operation.

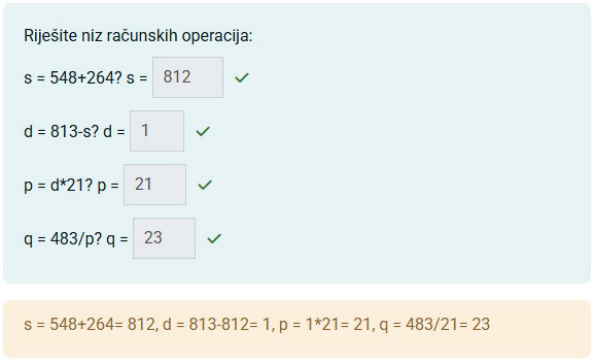
Listing 5 shows the generated XML record that defines such a question where primarily it can be noticed that the attribute *type* is set to the keyword "cloze", meaning it is a type of question with missing words to be filled-out or *embedded answers*, of which there are four. The embedded answers relate to the *short answer question type*, and the novelty in the question record compared to the previous example is that the section marked *generalfeedback* contains more detailed feedback on how the sequence of expected correct answers developed. This shows an example of useful use of the feedback section.

```
<!-- question: 13 -->
<question type=»cloze»>
  …
  <questiontext format=»html»>
   <text>&lt;p dir=»ltr» style=»text-align:
left;»&gt;&lt;/p&gt;&lt;p&gt;Riješite niz računskih
operacija:&lt;/p&gt;&lt;p&gt;s = 548+264? s =
{1:SHORTANSWER_C:=812}&lt;/p&gt;&lt;p&gt;&lt;/
p&gt;&lt;p&gt;d = 813-s? d = {1:SHORTANSWER_C:=1}&lt;/
p&gt;&lt;p&gt;&lt;/p&gt;&lt;p&gt;p = d*21? p =
{1:SHORTANSWER_C:=21}&lt;/p&gt;&lt;p&gt;&lt;/
p&gt;&lt;p&gt;q = 483/p? q = {1:SHORTANSWER_C:=23}&lt;/
p&gt;&lt;p&gt;&lt;/p&gt;</text>
  </questiontext>
  <generalfeedback format=»html»>
   <text>s = 548+264= 812, d = 813-812= 1, p = 1*21= 21, q
= 483/21= 23</text>
  …
</question>
```

Ispis 5 *Primjer Moodle XML opisa pitanja s „uklopljenim“ odgovorima.*

Listing 5 *Example of Embedded Answers question Moodle XML description.*

The display of the question in the user interface after entering the answer can be seen in Figure 6.



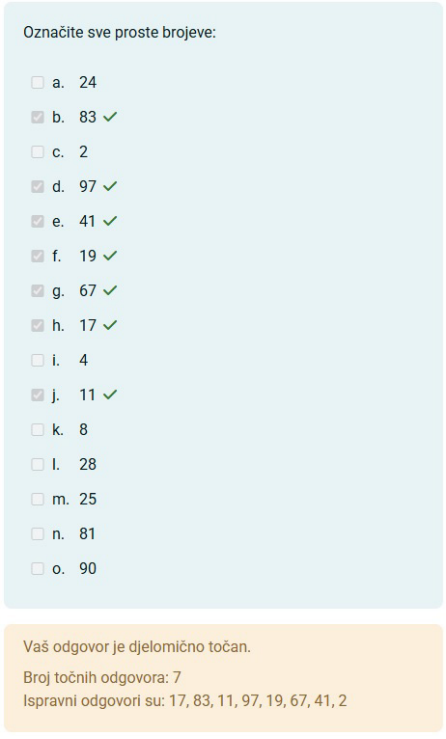
Slika 6 *Primjer opće povratne informacije kod pitanja s „uklopljenim“ odgovorima.*

Figure 6 *Example of Embedded Answers question general feedback.*

At the bottom of the screen window, there is also visible and highlighted general feedback as an explanation of the task solution, which can be particularly significant for more complex tasks and questions.

Multiple-choice question and grading shares of different answers

Figure 7 shows an example of a *multiple-choice question* with multiple correct answers. It is an example of a task where prime numbers are sought in a sequence of integer values.



Slika 7 *Primjer bodovanja pitanja s višestrukim odgovorima.*

Figure 7 *Example of Mulitple Choice question grading.*

It is possible to select multiple checkboxes next to a number considered to be a prime number. Among the offered answers, eight are correct: 2, 83, 97, 19, 41, 67, 17, and 11. The other options are incorrect answers, and there are seven of them. Therefore, for each correct answer, 100/8 or 12.5% of points is awarded, and for each incorrect answer, 100/7 or 14.28571% of points is subtracted. Picture 7 shows an example of answers and scoring for this question. For instance, if all correct answers except the answer "2" are marked and no incorrect answer is marked, the system will score it as 7×12.5 or 87.5% of the total points.

If an incorrect answer is selected, 14.28571% of the points share will be deducted.

The questions generated by the described application are always constructed so that the shares of correct and incorrect answer options are distributed evenly, depending on their number.

5. ZAKLJUČAK
5. CONCLUSION

In this paper, we presented a prototype application for generating Moodle questions for automatic grading using the Moodle XML format, which is easily exported and imported into the Moodle LMS system. The characteristics of several types of Moodle questions were presented, according to which sets of questions can be generated based on user preferences for different mathematical problems.

The implementation is done in Python programming language version 3.10.11 with an interface made in TkInter tool version 8.6. XML handling is achieved via the ElementTree API module for the Python programming language. The implemented prototype application is developed as a starting point for further upgrades in terms of expanding the set of questions and topics they cover.

The content of the prototype is currently limited to simple mathematical tasks as proof of concept, and more specific guidelines for upgrading the application could be divided into two aspects considering practicality and long-term application.

In the narrower sense and near future, upgrade steps would consist of the following interventions:

- Evaluation of the application's use in a specific teaching course where typical questions specific to the course material would be implied and expected to be more complex compared to simple arithmetic presented in this work,
- Introduction of new types of Moodle questions where tasks with arithmetic calculations according to formulas given by the course material would use the calculated question type, and for other tasks, for example, matching or ordering questions,
- Functionalities for designing the structure and content of written exams based on the generated question database, generating general or personalized exam copies for knowledge testing on paper, which would be exported to PDF before printing. This could be approached by extending the functionality within the current application or developing and using independent modules with varying degrees of integration with the current application.

Consequently, in a broader context of application, the direction of development and improvement of the application would consist of the following interventions:

- expansion of integration capabilities with various systems by shaping external content in other known file formats such as JSON or CSV,
- parameterization and generation of more complex content through the development of specialized templates and language syntax,
- evaluation in a real teaching environment regarding applicability and teacher satisfaction.

In conclusion, such work, besides representing a solid foundation for assumed expansions and applications, could also serve as additional motivation for implementations of the same or expanded functionalities in other technologies, all with the aim of supporting the teaching process in educational and higher education institutions.

6. REFERENCE

6. REFERENCES

[1.] T. Alajbeg, D. Cika, and T. Razov, Using Moodle to automatically grade laboratory exercises," Proceedings of the 34th International Convention MIPRO 2011, ed. Buljanović P., pp. 1343–1347, 978-953-233-067-0, Rijeka 2011

[2.] Moodle question types, https://docs.moodle.org/405/en/Question_types, Accessed on 16/10/2025

[3.] F. Zampirolli et al., Parameterized and automated assessment on an introductory programming course, Simpósio brasileiro de informática na educação (SBIE), 2020, 1573-1582 DOI: 10.5753/cbie.sbie.2020.1573

[4.] M. Gangur, Automatic Generation of Cloze Questions, in Proceedings of the 10th International Conference on Computer Supported Education (CSEDU), 2018, pp. 264–269. DOI: 10.5220/0003339102640269

[5.] W. Rolke, R Package moodlequizR: Fully Randomized Moodle Tests, 2024, arXiv preprint arXiv:2411.08032

[6.] A. Zeileis, N. Umlauf, and F. Leisch, Flexible Generation of E-Learning Exams in R: Moodle Quizzes, OLAT Assessments, and Beyond, Journal of Statistical Software, vol. 58, 2014, pp. 1–36, DOI: 10.18637/JSS.V058.I01

[7.] I. Mintii, S. Shokaliuk, T. A. Vakaliuk, M. Mintii, and V. Soloviev, Import Test Questions Into Moodle LMS, Educational Dimension 1, 2019, pp. 111-121, DOI: 10.31812/EDUCDIM.V53I1.3836

[8.] xml.etree.ElementTree — The ElementTree XML API, https://docs.python.org/3/library/xml.etree.elementtree.html, Accessed on 16/10/2025

[9.] tkinter — Python interface to Tcl/Tk, https://docs.python.org/3/library/tkinter.html, Accessed on 16/10/2025

[10.] Moodle XML format, https://docs.moodle.org/405/en/Moodle_XML_format, Accessed on 16/10/2025

[11.] S. Šalković, N. Sikirica, Automatsko vrednovanje algoritamskih zadataka, Polytechnic & Design, Vol. 7, No. 3, 2019., pp. 163-168, DOI: 10.19279/TVZ.PD.2019-7-3-02

AUTORI · *AUTHORS*

• Danko Ivošević - je viši predavač na Tehničkom veleučilištu u Zagrebu i izvodi nastavu na predmetima iz područja programskog inženjerstva i algoritama i struktura podataka na prijediplomskom i diplomskom studiju.

Diplomirao je 1999. i doktorirao 2013. godine na Fakultetu elektrotehnike i računarstva. Od 2019. godine zaposlen je na Tehničkom veleučilištu u Zagrebu, a prije toga je nakon završetka studija bio zaposlen na više radnih mjesta na poslovima programskog inženjera te na mjestu znanstvenog novaka i poslijedoktoranda na Fakultetu elektrotehnike i računarstva. Kao član strukovne udruge IEEE je u razdoblju od 2015. do 2022. obnašao službe predsjednika i dopredsjednika Odjela za računarstvo Hrvatske sekcije.

Korespondencija · *Correspondence*

divosevic@tvz.hr

• Aleksandar Stojanović - The unchanged biography can be found in Polytechnic & Design Vol. 12, No. 3, 2024.

Korespondencija · *Correspondence*

astojanov@tvz.hr