

# Разработка и внедрение тестирующей системы на занятиях по программированию

<sup>1</sup>\*МУХАРСКИЙ Дмитрий Васильевич, лектор, [DMyharskii@shokan.edu.kz](mailto:DMyharskii@shokan.edu.kz),  
<sup>1</sup>САБИТОВ Аманбек, докторант, ассистент лектора, [asabitov@shokan.edu.kz](mailto:asabitov@shokan.edu.kz),  
<sup>1</sup>НАО «Кокшетауский университет имени Ш. Уалиханова», ул. Абая, 76, Кокшетау, Казахстан,

\*автор-корреспондент.

**Аннотация.** Основным и самым эффективным видом деятельности при обучении программированию является решение задач. Автоматизированные системы проверки задач по программированию могут облегчить обязанности преподавателя, позволят охватить за меньшее время большее количество учащихся, уделить каждому учащемуся больше времени. В статье представлена автоматическая тестирующая программа для обучения программированию на языке Python, разработанная в Кокшетауском университете им. Ш. Уалиханова, описан интерфейс системы и её возможности. Описывается работа с системой и пути дальнейшего улучшения и развития. Предварительно рассмотрены тестирующие платформы для проведения олимпиад по программированию и обучения программированию. Проанализированы их положительные стороны и недостатки, заставившие авторов разрабатывать собственную тестирующую программу.

**Ключевые слова:** тестирующая программа, преподавание программирования, автоматическая проверка, олимпиадное программирование, веб-интерфейс, Flask, система проверки решения задач по программированию, веб-разработка.

## Введение

На современном этапе развития общества подготовка квалифицированных кадров в области IT-специальностей остаётся актуальной проблемой. Базовым умением для практически любого IT-специалиста является освоение им языков программирования. Умение программировать развивает все основные навыки, что требуются хорошему IT-специалисту: алгоритмическое мышление, умение структурировать свои мысли и выражать их в чётких инструкциях программы, умение мыслить логически и переводить математически сформулированные алгоритмы в устойчиво работающие программы.

Программирование остаётся сложной для обучения областью компьютерных наук. Любой программист знает, как сложно отладить даже, казалось бы, простую программу, заставить её работать как того хочет разработчик. Программа может хорошо работать на одних данных, начать выдавать неверные результаты на других и оказаться совершенно нерабочей на третьих.

Преподаватель в классе вынужден оценивать много решений в условиях ограниченного времени. Решения разных учащихся могут сильно отличаться по уровню. Кто-то

может легко понять общую идею решения и самостоятельно её реализовать. Другому требуется объяснять буквально построчно. Всё это приводит к тому, что преподаватель не в состоянии в полном объёме за отведённое время адекватно оценить решение. В этой ситуации легко пропустить неверные решения, работающие для ограниченного множества данных.

Учебная программа пишется, сдаётся и о ней больше не вспоминают. При этом к функции из реального проекта требования гораздо более жёсткие. Функция должна работать быстро, чтобы не тормозить весь проект, она должна быть хорошо документирована, легко масштабироваться и читаться и т.д. Кроме того, популярным методом разработки программ является разработка на основе тестирования. Программный продукт при каждой модификации должен проходить ряд тестов, гарантирующих сохранение его работоспособности. Эти требования, в какой-то мере, напоминают требования к олимпиадным программам [1, 2]. А раз так, то хорошо бы было с самых азов приучать студентов и учеников к такому способу программирования.

Существуют тестирующие системы, вхо-

дящие в состав более расширенных обучающих систем. Однако эти системы обладают рядом недостатков. Это явилось причиной для разработки нами собственной тестирующей системы. В статье авторы дают краткое описание функционирования и опыт применения такой системы на базе обучения студентов программированию на Python.

### Методы и материалы

На просторах интернета можно найти достаточно много ресурсов для автоматизированной проверки программ. В число наиболее используемых и проверенных авторами входят: Contester [3], система для дистанционной подготовки по информатике informatics.msk.ru [4], Timus Online Judge [5], Stepik [6], School.sgu.ru [7]. Основное назначение представленных выше ресурсов – проведение онлайн-олимпиад. Они мало пригодны для ежедневных занятий по программированию.

Все рассмотренные платформы имеют зарубежное базирование. Их иногда бывает трудно адаптировать к казахстанским требованиям и стандартам. Это, в первую очередь, касается и языка платформы и многих других аспектов использования.

По мнению авторов, имеется ещё один существенный недостаток в рассмотренных платформах. Даже зарегистрированные пользователи не имеют возможности напрямую просматривать базы данных, сохраняющие информацию о прохождении курсов. Информация доступна только через веб-интерфейсы. Нет возможности автоматизировать и анализировать эту информацию для построения, например, собственной рекомендательной системы на базе ИИ.

Учитывая всё вышесказанное, авторами было принято решение разработать собственную тестовую платформу по программированию.

Тестирующая система написана на языке программирования Python и построена на базе фреймворка Flask [8, 9]. Данный фреймворк был выбран в качестве рабочего благодаря его простоте и популярности для разработки маленьких и средних проектов [10].

Тестирующая система построена по принципам ООП. Ядром системы является класс, принимающий на вход текст программы, подающий в неё тестовые входные данные и сравнивающий выдаваемые результаты с ожидаемыми. Остальные модули предназначены для обслуживания веб-интерфейса. В частности, имеется класс для работы с деревом папок, в котором хранятся тестовые данные, класс для работы с базой данных зарегистрированных пользователей и ре-

зультатов решений, несколько классов для работы с элементами шаблонов веб-страниц и т.д.

В качестве рабочей базы данных используется библиотека SQLAlchemy [11, 12] и дополнительный модуль Flask-sqlalchemy. Встроенный модуль хорошо интегрируется со всеми остальными компонентами фреймворка и позволяет сосредоточиться на логике программы, беря многие мелкие детали реализации на себя [13].

Кроме вышеописанного модуля, в программе использованы следующие расширения: Flask-WTF для работы с формами в шаблонах веб-страниц, Flask-login – для авторизации и регистрации пользователей и интеграции с базой данных [14], Flask-migrate – для комфортной работы с изменениями базы данных и миграции данных из одной базы в другую, Flask-bootstrap – для оформления внешнего вида шаблонов в стиле Bootstrap [15].

Для хранения сведений о зарегистрированных пользователях и прогрессе обучения используются две таблицы базы данных. Первая из таблиц содержит сведения о пользователях: имя, фамилию и пароль. Вторая таблица хранит сведения о решаемых пользователями задачах.

Веб-интерфейс программы построен на основе динамических шаблонов Jinja2 [10]. Веб-ресурс состоит из нескольких страниц. Для использования ресурса в первый раз студент должен пройти простую процедуру регистрации. После этого открывается страница с задачами. Для начала работы нужно перейти на страницу с иерархическим списком тем. Если выбрать только название темы, то будет предложена случайная задача по выбранной теме. Так же можно выбрать конкретную задачу (рисунок 1).

Дерево строится из иерархии папок с темами, в которых хранятся файлы с наборами тестов. Для каждой задачи имеется свой текстовый файл с набором тестов только для этой задачи. Структура тестового файла подразумевает различные формы ввода и вывода. Программа может вообще не получать никакого ввода и только выводить результат. Возможен ввод одного значения или ряда значений (например, списка), ввод значений из нескольких операторов input(). Вывод также подразумевает либо одного значения, либо нескольких (в том числе и в столбик). Формат ввода и вывода оговаривается при формулировке задачи. В особо сложных случаях в формулировку задачи добавлен пример форматирования ввода и вывода.

Условие выбранной задачи отображается на странице задач. Учащийся может либо

набирать задачу напрямую в специализированном поле или скопировать её из другого редактора (рисунок 2).

После того как учащийся решил задачу, её можно отправить на проверку на сервер. По задаче на экран выдаётся краткий отчёт, сколько тестов пройдено и какие из тестов пройдены неправильно (рисунок 3). При этом в базу данных записывается, когда задача была открыта впервые, когда от-

правлено последнее решение, сколько было попыток отправки и другая информация, позволяющая отслеживать прогресс.

Доступен полный отчёт по всем решаемым данным учащимся задачам на странице Отчёт. На экран выдаётся часть информации из базы данных, позволяющая отследить результаты каждой из когда-либо решаемых учащимся задач (рисунок 4). Выставляемая учащемуся оценка зависит от количества

Tester Задачи Дерево Отчёт Logout

Дерево тем

Name	Modified Time	Size
Parent Directory		
Tasks(empty).txt	2024-02-01 20:15:32	38.6KB
Tasks.txt	2024-03-24 20:13:24	87.3KB
Tasks_o.txt	2024-03-13 10:40:50	4.1KB
Вложенные циклы	2024-04-02 08:44:48	0.0B
Выбор из нескольких действий	2024-04-02 08:44:48	0.0B
Вычисления по формулам	2023-12-17 21:57:48	0.0B
Использование словарей	2024-04-02 08:44:49	0.0B
Использование списков	2024-04-02 08:44:49	0.0B
Использование функций	2024-04-02 08:44:53	0.0B
Многомерные списки	2024-04-02 08:44:53	0.0B
Обработка строк символов	2024-04-02 08:44:53	0.0B
Олимпиада78	2024-04-02 08:44:53	0.0B
Повторение действий по условию	2024-04-02 08:44:53	0.0B
Повторение известного количества действий	2024-04-02 08:44:53	0.0B
Рубежный контроль 1	2024-04-02 08:44:54	0.0B
Целочисленная арифметика	2023-12-17 21:58:20	0.0B

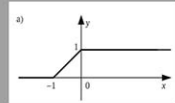
Рисунок 1 – Иерархический список тем

Tester Задачи Дерево Отчёт Logout

Hi, Admin!

Загружена задача: 4.120\_a

Для функции, заданной графически, определить значение  $y$  при заданном значении  $x$ .



Поле для ввода кода

Поле для ввода кода

```
num1 = float(input())
if x <= -1:
    print(0.0)
elif -1 < x <= 0:
    print(-(-1-x))
else:
    print(1)
```

ПРОВЕРИТЬ

Рисунок 2 – Решение задачи

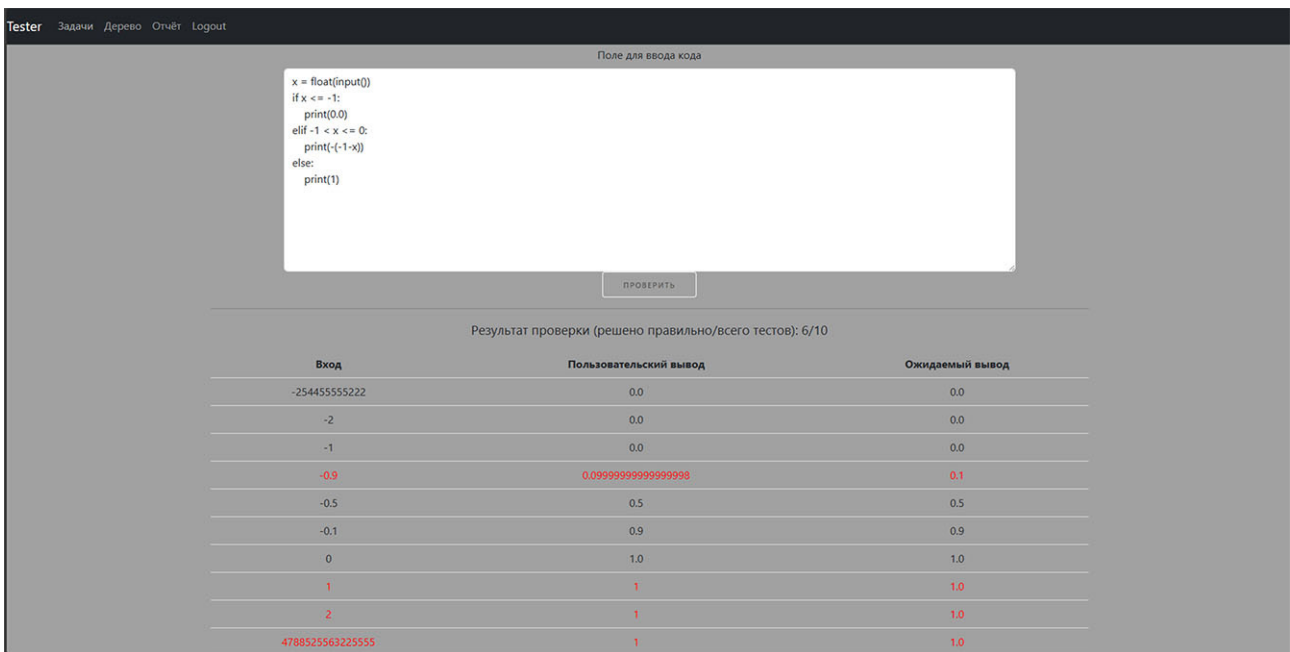


Рисунок 3 – Результаты отправки задачи на проверку

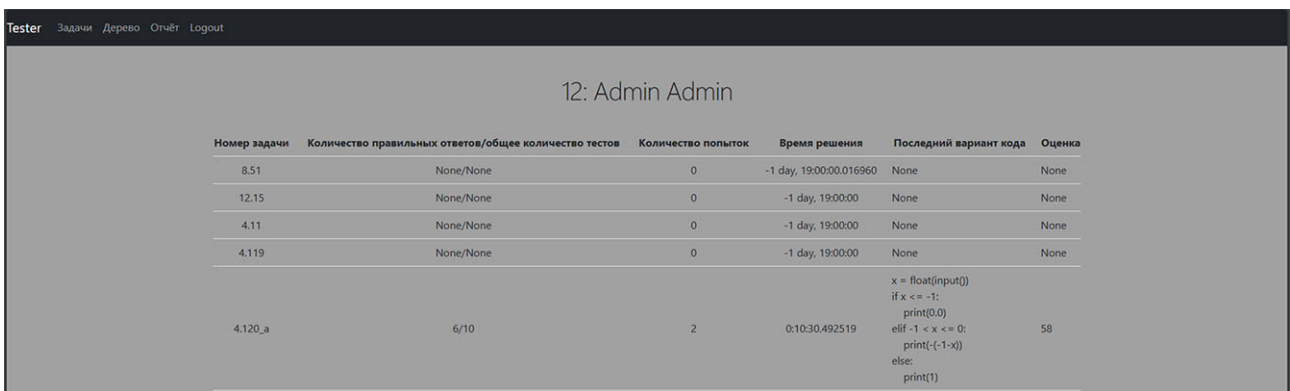


Рисунок 4 – Отчёт о решаемых задачах

успешно пройденных тестов и от количества попыток решения. Оценка выставляется по результатам последней проверки.

На момент написания статьи ресурс имеет простой интерфейс и ограниченную функциональность. Функции добавляются по мере возникновения необходимости в них и по мере отслеживания работы учащихся с ресурсом.

Разработанная система позволяет уделять больше времени каждому из учащихся и во время решения ими задач уделять больше времени проработке алгоритма решения, оптимизации кода, что особенно важно для языка Python. Отпала необходимость вспоминать, как была решена та или иная задача из предложенных на лабораторной, прогонять вручную много тестов и затрачивать на это время. Просмотр отчёта помогает препода-

вателю сразу вспомнить, как тот или иной учащийся решал и посмотреть приблизительную оценку, рекомендуемую системой.

Главным побудительным мотивом для авторов начать работу над проектом по созданию своей тестирующей программы, была возможность иметь полный контроль над системой, возможность оперативно исправлять ошибки и вносить новые функции, не дожидаясь обновлений от сторонних разработчиков.

### Результаты и обсуждения

Использование автоматизированных систем тестирования имеет ряд отрицательных сторон. Во-первых, при использовании автоматизированной проверки задач входные и выходные данные должны иметь строго заданные форматы. Учащиеся должны быть

заранее предупреждены о необходимости форматировать данные входа и выхода. Хорошей практикой будет прописывать в условиях задачи, в каком виде надо подавать данные и в каком формате выводить, как это делается в олимпиадном программировании. Однако новички в программировании всё равно зачастую делают ненужные добавки при вводе и выводе, вставляя комментарии в операторы `input()` и `print()`.

Во-вторых, зачастую желательно, чтобы задача была решена с использованием заданных конструкций и подмена конструкций нежелательна. Например, необходимо решить задачу с использованием цикла с условием, и замена его на более простой в использовании цикла с параметром не допустима, или необходимо решить задачу через функцию. В этом случае прогон тестовых случаев также можно дополнить поиском в исходных тестах программы некоторых ключевых слов (`while`, `def` и т.д.)

В-третьих, желательно научить не только просто решить задачу, а и решить её как можно более оптимально или компактно. Это требование может касаться, например, программирования на Python. Желательно научить учащихся пользоваться списковыми включениями, встроенными методами, спецфункциями и писать однострочники. Тогда можно наряду с прогоном тестов так-

же учитывать длину написанной программы и начислять или отнимать дополнительные баллы за компактность.

Ни один из рассмотренных ресурсов не позволял вместе с тестами проводить дополнительные операции с исходными кодами, а тем более добавлять какие-либо пользовательские операции с кодом. Наличие собственного ресурса позволяет полностью контролировать процесс тестирования. В будущем авторы предполагают включить в ресурс элементы ИИ для построения рекомендательной системы по подбору задач по уровню подготовки.

### Заключение

В статье авторы на основе рассмотрения пяти ресурсов для автоматизированной проверки программ представляют собственную автоматизированную проверяющую систему и обосновывают необходимость её разработки.

Авторы описывают собственную разработанную в Кокшетауском Университете им. Ш. Уалиханова платформу для проведения занятий по программированию на языке Python. Разработанный тестовый вариант платформы, несмотря на простоту, успешно используется уже в течение полугода. Исправлено множество ошибок и заменены некоторые первоначальные решения.

## СПИСОК ЛИТЕРАТУРЫ

1. Кирюхин В.М. Методические рекомендации по проведению школьного и муниципального этапов всероссийской олимпиады по информатике в 2012/2013 учебном году.
2. Московские олимпиады по информатике. 2002-2009 // Под ред. Андреевой Е.В., Гуровца В.М., Матюхина В.А. М.: МЦНМО. – 2009.
3. Долгушин Н.А., Оленькова М.Н. Использование системы `contester` для проведения олимпиад по программированию // Международный студенческий научный вестник. – 2016. – № 3-2; URL: <https://eduherald.ru/ru/article/view?id=14960> (дата обращения: 29.03.2024).
4. Дистанционная подготовка по информатике. URL: <https://informatics.msk.ru> (дата обращения 02.1.2024).
5. Timus Online Judge, архив задач с проверяющей системой. URL: <http://acm.timus.ru> (дата обращения 02.1.2024).
6. Stepic онлайн-курсы от ведущих вузов и компаний страны URL: <https://welcome.stepik.org/ru> (дата обращения 02.1.2024).
7. Портал обучения информатике и программированию URL: <http://school.sgu.ru> (дата обращения 02.1.2024).
8. Dwyer G. Flask By Example // Packt Publishing Ltd. – 2016.
9. Relan K. Beginning with flask // Building REST APIs with Flask: Create Python Web Services with MySQL. – 2019. – С. 1-26.
10. RYBÁROVÁ N. Python-FLASK for creating simple web applications.
11. Myers J., Copeland R. Essential SQLAlchemy: Mapping Python to Databases // O'Reilly Media, Inc – 2015.

12. Hans-Jürgen Schönig Mastering PostgreSQL 13 – Fourth Edition: Build, administer, and maintain database applications efficiently with PostgreSQL 13 // Packt Publishing. – 2020. – 476 p.
13. Катлер Дж., Дикенсон М. Баз данных. В кн.: Вычислительные фреймворки для политических и социальных исследований с помощью Python. Учебники по политическому анализу. // Springer, Cham. – 2020. URL: [https://doi.org/10.1007/978-3-030-36826-5\\_8](https://doi.org/10.1007/978-3-030-36826-5_8)
14. Ghimire D. Comparative study on Python web frameworks: Flask and Django. – 2020.
15. Lathkar M. Building Web Apps with Python and Flask: Learn to Develop and Deploy Responsive RESTful Web Applications Using Flask Framework (English Edition). – BPB Publications. – 2021.

### **Бағдарламалау сабақтарында тестілеу жүйесін әзірлеу және енгізу**

<sup>1</sup>\***МУХАРСКИЙ Дмитрий Васильевич**, оқытушы, [DMyharskii@shokan.edu.kz](mailto:DMyharskii@shokan.edu.kz),  
<sup>1</sup>**САБИТОВ Аманбек**, докторант, оқытушы көмекшісі, [asabitov@shokan.edu.kz](mailto:asabitov@shokan.edu.kz),  
<sup>1</sup>«Ш. Уәлиханов атындағы Көкшетау университеті» КеАҚ, Абай көшесі, 76, Көкшетау, Қазақстан,  
\*автор-корреспондент.

**Аңдатпа.** Бағдарламалауды оқытудағы негізгі және ең тиімді әрекет есептерді шешу болып қала береді. Бағдарламалау тапсырмаларын тексеруге арналған автоматтандырылған жүйелер оқытушының осы жауапкершілігін жеңілдетеді, аз уақытта көбірек студенттерге қол жеткізуге және әрбір студентке көбірек уақыт бөлуге мүмкіндік береді. Мақалада Ш.Уәлиханов атындағы Көкшетау университетінде әзірленген Python тілінде бағдарламалауды оқытуға арналған автоматты тестілеу бағдарламасы берілген, жүйе интерфейсі және оның мүмкіндіктерін сипаттайды. Жүйемен жұмыс істеу және одан әрі жетілдіру және дамыту жолдарын сипаттайды. Бағдарламалау олимпиадаларын өткізу және бағдарламалауды оқыту үшін тестілеу алаңдары алдын ала қаралды. Олардың оң жақтары мен кемшіліктері талданады, бұл авторларды өздерінің тестілеу бағдарламасын жасауға мәжбүр етті.

**Кілт сөздер:** тестілеу бағдарламасы, бағдарламалауды оқыту, автоматты тестілеу, бағдарламалау олимпиадалары, веб-интерфейс, Flask, бағдарламалау есептерінің шешімдерін тестілеу жүйесі, веб-әзірлеу.

### **A Testing System in Programming Classes Development and Implementation**

<sup>1</sup>\***MUKHARSKY Dmitry**, Lecturer, [DMyharskii@shokan.edu.kz](mailto:DMyharskii@shokan.edu.kz),  
<sup>1</sup>**SABITOV Amanbek**, Doctoral Student, Lecturer Assistant, [asabitov@shokan.edu.kz](mailto:asabitov@shokan.edu.kz),  
<sup>1</sup>NCJSC «Sh. Ualikhanov Kokshetau University, Abay Street, 76, Kokshetau, Kazakhstan»,  
\*corresponding author.

**Abstract.** The main, and most effective, activity when teaching programming is problem solving. Automated systems for checking programming problems can ease the teacher's responsibilities. They will allow you to reach a larger number of students in less time and devote more time to each student. The article presents an automatic testing program for teaching programming in Python, which was developed at Kokshetau University named after Sh. Ualikhanov. The program interface and its capabilities are described. Describes how to use the program and ways to further improve and develop it. Testing platforms for conducting programming Olympiads and teaching programming have been preliminarily reviewed. Their positive aspects and disadvantages are analyzed, which forced the authors to develop their own testing program.

**Keywords:** testing program, programming teaching, automatic testing, programming Olympiads, web-interface, Flask, system for testing solutions to programming problems, web-development.

## REFERENCES

1. Kiryukhin V.M. Metodicheskie rekomendacii po provedeniu Shkolnogo i municipalnogo etapov vsrossiyskoy olimpiadi po informatike v 2012/2013 uchebnom godu [in Russian].
2. Moskovskie olimpiadi po informatike. 2002-2009 // Ed. Andreeva E.V., Gurovets V.M., Matyukhina V.A. Moscow: MTsNMO. – 2009 [in Russian].
3. Dolgushin N.A., Olenkova M.N. Ispolzovanie systemi contestov dlya provedeniya olimpiad po programmirovaniyu // Mezdunarodniy studencheskiy nauchniy vestnik. – 2016. – No. 3-2. URL: <https://eduherald.ru/ru/article/view?id=14960> (date of access: 02/1/2024) [in Russian].
4. Distantsionnaya podgotovka po informatike. URL: <https://informatics.msk.ru> (access date 02/1/2024)
5. Timus Online Judge, archive of problems with a checking system. URL: <http://acm.timus.ru> (access date 02/1/2024)
6. Stepic online courses from leading universities and companies in the country URL: <https://welcome.stepik.org/ru> (access date 02/1/2024)
7. Portal for teaching computer science and programming URL: <http://school.sgu.ru> (access date 02/1/2024)
8. Dwyer G. Flask By Example // Packt Publishing Ltd. – 2016.
9. Relan K. Beginning with flask // Building REST APIs with Flask: Create Python Web Services with MySQL. – 2019. – Pp. 1-26.
10. RYBÁROVÁ, N. Python-FLASK for creating simple web applications.
11. Myers J., Copeland R. Essential SQLAlchemy: Mapping Python to Databases // O'Reilly Media, Inc – 2015.
12. Hans-Jürgen Schönig Mastering PostgreSQL 13 – Fourth Edition: Build, administer, and maintain database applications efficiently with PostgreSQL 13 // Packt Publishing. – 2020. – 476 p.
13. Cutler J., Dickenson M. Databases. In: Computational frameworks for political and social research using Python. Textbooks on Political Analysis // Springer, Cham. – 2020. URL: [https://doi.org/10.1007/978-3-030-36826-5\\_8](https://doi.org/10.1007/978-3-030-36826-5_8)
14. Ghimire D. Comparative study on Python web frameworks: Flask and Django. – 2020.
15. Lathkar M. Building Web Apps with Python and Flask: Learn to Develop and Deploy Responsive RESTful Web Applications Using Flask Framework (English Edition). – BPB Publications. – 2021.