

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра Информатики и программирования

**ОБРАЗОВАТЕЛЬНОЕ ПРИЛОЖЕНИЕ «ИСПОЛНИТЕЛЬ КУЗНЕЧИК»  
НА БАЗЕ LMS ПЛАТФОРМЫ MOODLE  
АВТОРЕФЕРАТ БАКАЛАВРСКАЯ РАБОТА**

студента 4 курса 441 группы  
направления 02.03.03 — Математическое обеспечение и  
администрирование информационных систем  
факультета КНиИТ  
Кузичкина Павла Александровича

Научный руководитель

старший преподаватель

\_\_\_\_\_

А. А. Казачкова

Заведующий кафедрой

к. ф.-м. н., доцент

\_\_\_\_\_

М. В. Огнева

Саратов 2023

## ВВЕДЕНИЕ

### **Актуальность темы.**

В настоящие дни стремительно развивается направление онлайн-образования. Появляется множество различных онлайн-школ, в которых можно получить знания в любой области, достаточно лишь найти понравившийся ресурс, на котором удобнее всего обучаться.

Большинство таких онлайн образовательных учреждений являются веб-приложениями, которые написаны с использованием совершенно разных стеков технологий, а также следуя множественным уже реализованным практикам проектирования веб-приложений. При таком многообразии зачастую встает вопрос выбора подходящих технологий.

Данная работа посвящена созданию образовательной платформы для обучения детей младших классов различных образовательных учреждений основам математического и алгоритмического мышления. На данной платформе реализована работа с исполнителем «Кузнечик».

Контроль, мониторинг и оценку качества работы учеников сложно представить без наличия некоторой системы, которая производит все эти действия сразу и в едином месте. В этом нам помогает система контроля образования (LMS) Moodle. Она предполагает взаимодействия двух основных групп пользователей – учителей и учеников. Учителя выдают задания ученикам, следят за их успеваемостью, рекомендуют и назначают целые образовательные курсы. В то время как ученики выполняют требования и рекомендации преподавателей и получают новые знания.

Цель работы: разработать и интегрировать приложение для исполнителя «Кузнечик» в LMS Moodle.

Задачи работы:

1. Изучить источники, необходимые для реализации веб-приложения, его интеграции в LMS и развертывания на сервере.
2. Изучить аналогичное ПО, которое предоставляет работу с различными исполнителями, в том числе с «Кузнечиком».

3. Изучить LMS-платформу Moodle.
4. Изучить путь ее соединения с основным web-сервером.
5. Изучить, как устроен Ubuntu-сервер и как его использовать для публикации приложения.
6. Интегрировать приложение в рабочий образовательный портал, развернутый при помощи платформы Moodle.
7. Проработать взаимодействие учителей и учеников на платформе Moodle.
8. Опубликовать приложение на внешний сервер, на котором будет работать одновременно web-сервер и Moodle.

#### **Методологические основы.**

Разработка web-приложений на основе Python Django, React, Docker (Docker-compose), ее примеры, пояснения и размышления представлены в работах следующих авторов: Д. Б. Аглямудинова, А. В. Готвальд, К. А. Мащенко, М. Поритта, Б. Кришнамурти, Дж. Рексфорда.

#### **Практическая значимость работы.**

Практическая значимость работы заключается в использовании данного приложения для осуществления образовательного процесса по информатическим дисциплинам у учеников младших классов.

#### **Структура и объем работы.**

Бакалаврская работа состоит из введения, 13 разделов, заключения, списка использованных источников и 10 приложений. Общий объем работы – 84 страницы, из них 42 страницы – основное содержание, включая 12 рисунков, цифровой носитель в качестве приложения, список источников информации – 22 наименования.

**В разделе 1 «Обзор источников»** были рассмотрены источники по технологиям, которые можно разделить на две категории: 1) источники с главной технической информацией, чаще всего на английском языке, и 2) источники с побочной и вспомогательной информацией, на русском и английском языках. Примеры источников: статьи о RESTful API with Python Django, технологии CI/CD, системе управления базами данных PostgreSQL, разработке собственного модуля для платформы Moodle и использовании платформы Moodle в обучении иностранным языкам.

**Во разделе 2 «Обзор аналогов приложения»** для создания интерактивного исполнителя «Кузнечика» в веб-среде нужно изучить рынок готовых решений и приложений, которые соответствуют необходимым критериям. Одним из наиболее подходящих аналогов является приложение «Кумир», которое предоставляет простой и расширяемый набор инструкций для работы с исполнителем «Кузнечиком». В «Кумире» есть некоторые проблемы, такие как ограниченное количество команд и проблемы с указанием некоторых чисел. При создании веб-приложения будут учтены все эти особенности.

Для создания веб-приложения, аналогичного исполнителю «Кузнечик» в «Кумире», необходимо учитывать не только его преимущества и недостатки, но и возможности, которые могут быть доступны в веб-среде. Например, в веб-приложении можно предоставить более удобный и интуитивно понятный интерфейс для работы с исполнителем «Кузнечиком», а также использовать более широкий набор инструкций и функций.

Одним из возможных подходов к созданию веб-приложения является использование современных технологий, таких как HTML5, CSS и JavaScript, чтобы обеспечить более динамическое и интерактивное взаимодействие пользователя с исполнителем. Также можно использовать различные библиотеки и фреймворки, такие как React, Angular или Vue.js, для упрощения разработки веб-приложения.

Таким образом, создание веб-приложения, аналогичного исполнителю «Кузнечик» в «Кумире», может быть достигнуто за счет использования современных технологий и библиотек, учета особенностей и недостатков существующих решений, а также обеспечения хорошей производительности и скорости работы приложения.

**В разделе 3 «Формальный язык для исполнителя Кузнечик»** поясняется теоретическая основа того, как может быть разработан статически анализируемый язык для исполнителя. Грамматика языка задает его синтаксические закономерности и представляет собой конечное описание бесконечного множества цепочек языка. Существуют разные синтаксические теории и формализмы для задания грамматик. Грамматики могут быть распознающими, порождающими и перечисляющими, причем порождающие и перечисляющие грамматики можно преобразовывать друг в друга, но распознающие грамматики могут быть менее мощными.

**В разделе 4 «Описание платформы Moodle»** рассказывается о преимуществах и недостатках платформы Moodle. Moodle - бесплатная система электронного обучения, которая позволяет дистанционно развивать и тестировать учеников со всего мира. Система имеет много плагинов, которые меняют дизайн и расширяют функциональность. Она доступна на более чем 100 языках и используется крупными университетами во всем мире. В Moodle можно создавать онлайн-курсы и тесты, ограничивать время и число попыток. Система автоматически проверяет ответы и показывает допущенные ошибки и набранный балл. Есть платный аналог - iSpring Learn, но функционала Moodle достаточно для достижения целей данной работы.

**В разделе 5 «Описание связи со сторонними сервисами на примере LMS Moodle»** Интеграции систем могут связываться по протоколу, который позволяет передавать сообщения унифицированного формата. Один из популярных вариантов - протокол HTTP (HTTPS), который поддерживает различные гипертекстовые форматы. Приложение "Исполнитель Кузнечик" может интегрироваться с LMS Moodle, используя модуль на языке PHP,

который будет перенаправлять пользователя на выполнение задания и записывать результаты выполнения в качестве попытки ученика. Модуль также будет считать баллы за выполнение каждого задания отдельно и средний балл за все задания

**В разделе 6 «Развертывание приложения на Ubuntu Linux сервере»** рассказывается об основах администрирования Linux серверов, о правилах разграничения прав пользователей сервером, генерация SSH-подключения и доступа пользователя.

**В разделе 7 «Создание веб-интерфейса для управления Кузнечиком»** рассказывается о том, как был создан интерфейс для приложения, как он был реализован и какие концепты были использованы.

Для создания веб-интерфейса для управления исполнителем «Кузнечик» на Python с использованием фреймворка Django, была взята за основу архитектура REST API, которая предполагает собой сохранение и передачу данных с сервера в определенном виде, представляя собой текущее состояние данных в базе. Клиентская часть приложения была выполнена при помощи фреймворка front-end разработки React, разработанного компанией Facebook.

**В разделе 8 «Изоляция приложения в Docker-контейнеры»** описывается процесс контейнеризации приложений с использованием Docker для упрощения развертывания и масштабирования. Для создания Docker-контейнера необходимо создать Dockerfile, который будет содержать инструкции для сборки Docker-образа. В случае приложения Django, в Dockerfile необходимо определить базовый образ Docker, рабочую директорию контейнера, установить зависимости приложения, скопировать файлы проекта, указать порт и определить команду для запуска сервера разработки.

Для приложения React необходимо создать Dockerfile, определить базовый образ Docker, рабочую директорию контейнера, скопировать файлы приложения, установить зависимости и определить команду для запуска сервера разработки. После создания Dockerfile необходимо собрать Docker-образ и запустить контейнер. Контейнеризация с использованием Docker

позволяет упростить развертывание и масштабирование приложения, обеспечивает его изолированность и портативность.

**Раздел 9 «Разработка собственного дистрибутива Moodle».** Moodle - это LMS, используемая в образовательных учреждениях. Создание собственного дистрибутива Moodle может быть необходимо для адаптации системы под специфические потребности конкретных учебных заведений. Для этого нужно определить требования, исследовать существующие модули и плагины, разработать план и концепцию дистрибутива, установить и настроить стандартную версию Moodle, определить и настроить необходимые модули и плагины, а также разработать и настроить собственные модули и плагины. Затем нужно подготовить пакет дистрибутива и распространить его среди целевой аудитории. Создание собственного дистрибутива позволяет адаптировать систему под конкретные потребности пользователей и образовательных организаций.

**В разделе 10 «Порядок разработки плагина для Moodle»** рассказывается о разработке модуля (плагина) на платформе Moodle предоставляет возможность добавить новую функциональность и расширить возможности системы управления обучением. В данном тексте для диплома будет представлена инструкция по разработке модуля (плагина) на Moodle.

Moodle предоставляет гибкую архитектуру и API для разработки модулей и плагинов. Разработка собственного модуля или плагина позволяет добавить новые возможности и настроить систему в соответствии с требованиями конкретного образовательного учреждения или организации.

**В разделе 11 «Соединение Moodle и web-сервера по протоколу HTTP/1.1»** подробно рассматривается соединение Moodle и Django-приложения при помощи API, что позволяет обмениваться данными между системами. Для этого необходимо подготовить Moodle для работы с API и создать клиентское приложение в Moodle. Затем нужно настроить Django для работы с API, установив необходимые библиотеки и добавив настройки в файл settings.py. Далее следует реализовать функциональность API в Django-

приложении, используя библиотеку requests для отправки HTTP-запросов к API Moodle и определив функции и классы для взаимодействия с Moodle.

Наконец, необходимо создать документацию и подготовить приложение для деплоя на сервере или хостинге. Создание соединения позволяет взаимодействовать между системами и обмениваться данными.

**В разделе 12 «Описание внутренней работы приложения»** показана полная реализация приложения и демонстрация его работы. Было показано, какой интерфейс имеет приложение, каков порядок пользования и как между собой могут взаимодействовать разные группы пользователей. В том числе – ученики и учителя на портале Moodle, ученики с исполнителем на портале Кузнечика.

**Раздел 13 «Публикация приложения на сервер»** рассказывает о том, как опубликовать приложение в общий доступ, для того, чтобы каждый желающий пользователь мог им воспользоваться. Причем на одном сервере существуют сразу 2 приложения:

- Портал Кузнечика;
- Портал Moodle;



## ЗАКЛЮЧЕНИЕ

По итогам данной работы было создано веб-приложение «Исполнитель Кузнечик» для обучения основам программирования учеников младших классов общеобразовательных учебных заведений, было рассмотрено понятие RESTful API архитектуры построения веб-приложений, способы общения клиента и сервера в рамках этой архитектуры.

Была рассмотрена, изучена и применена технология Docker (Docker-compose) для изоляции отдельных частей приложения в своих собственных контейнерах, была произведена композиция Docker-контейнеров в один единый контейнер при помощи Docker-compose. При помощи этого приложение запускается распределенно в нескольких потоках.

Изучена возможность публикации приложения на открытый хостинг, путем развертывания всего приложения на сервере с операционной системой Ubuntu Server 20.04 LTS. Для успешного развертывания приложения необходима настройка удаленного подключения к такому серверу, настройка доступов пользователей, смена конфигурации сервера с «личного пользования» на «публичный доступ».

Была реализована авторизация пользователя в приложении, данная авторизация в будущем будет применена для связки с образовательной платформой Moodle для более тщательного и планомерного мониторинга за образовательным процессом учеников в формате онлайн. Был описан возможный сценарий того, как можно связать «исполнителя Кузнечика» с образовательной платформой Moodle.

Настройка Docker приложения на Ubuntu-сервере включает несколько шагов, включающих установку Docker, настройку контейнеров и запуск приложения в Docker-контейнере. Были описаны необходимые шаги и проделана соответствующая работа для публикации приложения

Часть выпускной квалификационной работы была представлена на конференции «ИТО-Саратов 2022» в октябре 2022 года. Были показаны

промежуточные этапы, которые были продолжены и завершены в рамках этой работы.

## **Основные источники информации.**

1. Графические исполнители в системах пиктограммного, блочного и текстового программирования / Д. Б. Аглямудинова, А. В. Готвальд, К. А. Мащенко [и др.] // Труды научно-исследовательского института системных исследований Российской академии наук. – 2019. – Т. 9, № 6. – С. 6-20. – DOI 10.25682/NIISI.2019.6.0001. – EDN XVUTNQ. (дата обращения: 03.05.2023)
2. EMA-4-Moodle: The European project of studying foreign languages with the help of Moodle / В. Krouk, N. Chupakhina, K. Lomakin // Proceedings - 2010 IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering, SIBIRCON-2010 : 2010 IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering, SIBIRCON-2010, 11–15 июля 2010 года / sponsors: IEEE Region 8, Russian Foundation for Basic Research. – Irkutsk Listvyanka, 2010. – P. 390-392. – DOI 10.1109/SIBIRCON.2010.5555113. – EDN MXPDLZ. (дата обращения: 03.05.2023)
3. What is RESTful API? [Электронный ресурс]: интернет-статья. - URL: <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (дата обращения: 15.10.2022)
4. What is CI/CD? [Электронный ресурс]: интернет-статья. - URL: <https://www.redhat.com/en/topics/devops/what-is-ci-cd#:~:text=CI%2FCD%20is%20a%20method,continuous%20delivery%2C%20and%20continuous%20deployment.> (дата обращения: 10.11.2022)
5. Разработка собственного модуля Moodle. [Электронный ресурс]: интернет-статья. - URL: <https://Moodle.org/mod/forum/discuss.php?d=349964#p1412105> (дата обращения: 19.03.2023)
6. RESTful API with Python Django [Электронный ресурс]: интернет-статья. - URL: <https://www.django-rest-framework.org/> (дата обращения: 27.10.2022)

7. Советов П.Н. РАЗРАБОТКА КОМПИЛЯТОРОВ ПРЕДМЕТНО-ОРИЕНТИРОВАННЫХ ЯЗЫКОВ ДЛЯ СПЕЦПРОЦЕССОРОВ // Труды ИСП РАН. 2020. №5. URL: <https://cyberleninka.ru/article/n/razrabotka-kompilyatorov-predmetno-orientirovannyh-yazykov-dlya-spetsprotsessorov> (дата обращения: 16.04.2023)
8. Use Docker-compose [Электронный ресурс]: интернет-статья. - URL: [https://docs.docker.com/get-started/08\\_using\\_compose/#:~:text=Docker%20Compose%20is%20a%20tool,or%20tear%20it%20all%20down](https://docs.docker.com/get-started/08_using_compose/#:~:text=Docker%20Compose%20is%20a%20tool,or%20tear%20it%20all%20down). (дата обращения: 25.12.2022)
9. A docker environment for Moodle developers. [Электронный ресурс]: интернет-статья. – URL: <https://github.com/moodlehq/moodle-docker> (дата обращения: 01.05.2023)