

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**  
**Кафедра информационных технологий**

Допустить к защите  
Заведующий кафедрой  
канд. физ.-мат. наук, доц.  
\_\_\_\_\_ В.В. Подколзин  
(подпись)  
\_\_\_\_\_ 2024 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**  
**(БАКАЛАВРСКАЯ РАБОТА)**

**ПРОГРАММНОЕ ПРИЛОЖЕНИЕ РАЗРАБОТКИ ТЕСТОВ**

Работу выполнил \_\_\_\_\_ И.В. Бойко  
(подпись)

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) Программирование и информационные технологии

Научный руководитель, доц.  
канд. физ.-мат. наук, зав. каф. \_\_\_\_\_ В.В. Подколзин  
(подпись)

Нормоконтролер  
канд. пед. наук, доц. \_\_\_\_\_ А.В. Харченко  
(подпись)

Краснодар  
2024

## СОДЕРЖАНИЕ

Введение.....	3
1 Методы проведения тестирования.....	4
2 Обзор существующих программных решений.....	10
3 стек технологий для разработки приложения.....	15
3.1 стек технологий веб-приложения.....	15
3.2 стек технологий мобильного приложения.....	21
4 Постановка задачи.....	26
4.1 Постановка задачи разработки веб-приложения.....	26
4.2 Постановка задачи разработки мобильного приложения.....	27
5 Описание приложения.....	28
5.1 Описание веб-приложения.....	33
5.2 Описание мобильного приложения.....	37
Заключение.....	40
Список использованных источников.....	41

## ВВЕДЕНИЕ

В образовательном процессе важную роль играет контроль знаний. Существует множество различных форм контроля знаний. Одной из наиболее распространённых форм контроля знаний является тестирование.

В настоящее время в эпоху информационных технологий в образовании применяется обучение с использованием компьютерных технологий, и тестирование также проводится при помощи электронных образовательных технологий. В связи с этим актуальна задача создания программного приложения разработки тестов, пригодного для использования как преподавателями, так и обучающимися в учреждениях школьного и высшего образования.

Данная работа преследует собой две основные цели.

Первой целью является написание веб-приложения, которое позволяет преподавателям создавать и изменять тесты, а студентам проходить тесты и просматривать статистику выполнения заданий и прохождения тем.

Вторая цель - разработать мобильное приложение, предназначенное для применения студентами для прохождения тестов.

Также данная работа ставит перед собой следующие задачи: изучить методы проведения тестирования с использованием различных источников информации, изучить существующие программные решения для проведения тестирования с использованием официальных информационных ресурсов разработчиков данных продуктов. изучить стек технологий, которые предлагается использовать при разработке приложения при помощи как ресурсов в сети Интернет, так и бумажных изданий.

Созданные программные приложения должны быть пригодны для применения в реальном образовательном процессе в школах и университетах и удобными для использования преподавателями и обучающимися.

## **1 Методы проведения тестирования**

В настоящее время как средство интеллектуального развития человека выступает усвоение и обобщение готовых знаний. Главным является умение самостоятельно искать информацию, находить проблемы и методы их решения, проводить анализ и использовать приобретаемые знания в процессе решения новых задач[1].

Одним из методов усвоения и обобщения знаний являются тесты, которые представляют собой систему заданий, выполнение которых позволяет измерить уровень владения учебным материалом по тем или иным предметам при помощи специфической системы оценивания результатов[20]. Изначально в исторически сложившейся системе образования в России тестированию не отводилось место. Поэтому попытка использовать тестирования как одно из основных средств контроля столкнулись с серьёзным сопротивлением со стороны преподавателей и обучающихся.

Тестирование выполняет следующие три основные функции: контроль, управление и обучение. При этом контролирующая функция является главной. Тестирование может контролировать результат обучения, но не позволяет производить оценку самого учебного процесса. Именно в этом и состоит главная проблема технологии тестирования, связанная с расширением её функциональных возможностей[1].

Помимо функций у тестирования также есть и свои задачи[20]: развитие интереса к изучаемому предмету у участников образовательного процесса, развитие у обучающихся самостоятельной подготовки к прохождению тестирования, формирование ответственности у студентов за результаты контроля знаний, замер уровня усвоения учебного материала по дисциплинам, по которым проходит тестирование, помощь обучающимся в определении результатов прохождения курса и планирование подготовки к будущему контролю усвоения учебного материала и помощь преподавателям в оценке индивидуальных достижений обучающихся и изменении методики обучения.

Помимо этого, существуют следующие критерии содержания тестового материала: значимость, научная достоверность, разнообразность и комплексность содержания[20]. Ниже представлены их определения.

Значимость – указание необходимости включения только тех материалов, относящихся к наиболее важным, в отсутствие которых знания были бы недостаточно полными.

Научная достоверность – включаться в тест может только тот материал учебной дисциплины, истинность которого легко объективно подтвердить и аргументировать.

Разнообразность содержания – означает, что содержание тестовых материалов должно изменяться вместе с учебной программой.

Комплексность содержания – данный термин относится к тем тестам, которые предназначены для проведения итогового контроля усвоения учебного материала и означает, что он должен состоять из заданий по всем темам учебной программы[20].

Тестовые технологии применяются по следующей схеме: обучение – формирование – контроль. При попытке отхода от этой схемы возможно возникновение противоречий. Обучение происходит по традиционной схеме, а контроль – при помощи тестов. Применение тестовых технологий непосредственно для обучения не представляется возможным.

По своему назначению тесты делятся на следующие три основные категории: обучающие, формирующие и контролирующие. Ниже представлено их описание.

Обучающие тесты представляют собой набор заданий, ориентированных на усвоение небольших объёмов материала с последующей проверкой и работой над ошибками. Целью обучающих тестов является применение наводящих вопросов и подсказок, совместный разбор результатов и форма работы в группах. Подобные тесты определяют уровень возможностей, сущность изучаемого материала и глубину его восприятия.

Формирующие тесты помогают формировать качественные знания. Также подобные тесты выявляют и устраняют пробелы, если такие имеются, в усвоении учебного материала. Обучающие модули с формулами, правилами, алгоритмами, примерами и рекомендациями – вот что включают в себя формирующие тесты.

Контролирующие тесты способствуют выявлению уровня знаний и владения отдельными видами деятельности: познавательной, научной и практической[1].

По своей форме тесты бывают открытыми и закрытыми[20]. Открытая форма подразумевает собой развёрнутый ответ, который обучающийся должен сформулировать самостоятельно. Закрытая форма предполагает выбор одного или нескольких из определённого числа предложенных вариантов ответа. В свою очередь, закрытая форма тестирования делится на соответствующие подвиды: с выбором одного варианта ответа; с выбором нескольких вариантов ответа; с установлением соответствия между условиями задания и предложенными вариантами ответа; с определением порядка следования друг за другом вариантов ответа.

У тестирования как средства контроля знаний есть свои преимущества и недостатки. Далее описываются три основных преимущества тестирования. Первое преимущество заключается в том, что тестирование подразумевает собой стандартизированную процедуру сбора и обработки данных, а также последующую их интерпретацию. Второе преимущество заключается в возможности проверки степени усвоения учебного материала обучающимися по большому спектру модулей. И третье преимущество - тестирование исключает субъективность контроля и оценки со стороны преподавателя и, таким образом, являясь наиболее объективным средством контроля знаний[20].

С другой стороны, тестирование также имеет ряд существенных и в некоторых случаях критических недостатков. Перечислим четыре основных недостатка тестирования. Первый заключается в том, что если тестирование

проводится в закрытой форме, то невозможно полностью исключить вероятность угадывания ответа на вопрос. Но возможно сильно уменьшить эту вероятность путем изменения формулировок заданий, вариантов ответа и их порядка. Вторым недостатком - тестирование не позволяет проверить расширенные индивидуальные знания по предмету. Третьим недостатком тестирования заключается в узнавании как основного фактора при составлении тестовых материалов. Это связано с тем, что большинство школьных задач имеют шаблонный характер и решаются по строго детерминированному алгоритму, который в процессе обучения доводится до автоматизма. Как следствие, студенты обучаются работе со стандартизированными формулировками, что приводит к формированию шаблонного типа мышления. Любое даже незначительное изменение формулировки задания приводит к невозможности его выполнения. Более того, развитие стереотипности мышления начинается уже в начальной школе, что приводит к потере инициативы уже на данном этапе обучения. Поэтому тестовые задания не должны быть направлены только на решение типичных задач, но также содержать в себе задания познавательного и логического характера. Они должны побуждать обучающихся к сравнениям, наблюдениям, анализу заданий, выявлению закономерностей и причинно-следственных связей. И четвертым недостатком - тестирование предполагает наличие инструкции по работе с материалами для контроля знаний. Это приводит к вероятности непонимания того, как нужно выполнять задания, и, следовательно, к ошибкам в ответах[20].

На данный момент существует множество вариантов обучающих тестов по разным предметам в системе начального, общего, среднего и высшего образования. Начиная со второй половины 1990-х годов создаются тесты, предназначенные для обучения по разнообразным дисциплинам. Помимо этого проводились различные эксперименты по применению обучающих тестов на практике, по результатам которых были сделаны выводы, суть которых заключается в том, что использование обучающих тестов в

образовательном процессе приводит к более полному усвоению учебного материала[1]. В век информационных технологий широкое распространение получило тестирование в онлайн режиме. Электронная форма проведения тестирования имеет следующие преимущества[21]: снижаются затраты на изготовление материалов в печатной форме, что особенно актуально в случае проведения тестирования в университетах в больших учебных группах либо на всём учебном потоке; сокращается время проверки и обработки результатов тестирования, что также является преимуществом при работе с большим количеством обучающихся; обучающимся предоставляется удобный и понятный формат выполнения заданий, что уменьшает вероятность неправильного понимания процедуры тестирования и совершения ошибок; наличие единого банка тестовых заданий, который можно при необходимости быстро изменить.

Многие интернет ресурсы, такие как Moodle, 1С:Электронное обучение и другие, позволяют создавать обучающие тесты по различным предметам. Они должны предоставлять возможность создавать задания разного типа, добавлять новые задания в уже созданный тест, использовать технологии мультимедиа в проведении тестирования, автоматически представлять результаты тестирования, в том числе статистики по группам и по заданиям.

На сегодняшний день тестирование применяется в различных образовательных технологиях и методиках и выполняют множество функций. При использовании тестов в образовательном процессе необходимо обращать внимание на их качество. Важной деталью является то, что создание заданий для проведения тестирования является методическим процессом, который требует многолетнего опыта и больших практических навыков у преподавателя.

В условиях подхода к обучению, основанном на компетенциях, системной деятельности и электронном образовании, тесты не могут применяться в их традиционном понимании. Причина заключается в невозможности формирования и измерения компетенций путем подхода к

тестированию как ответ верно или неверно. Формирование компетенций делится на этапы и имеет свои особенности. Поэтому при проведении тестирования должны определяться данные этапы и измеряться в соответствии с компетенциями уровень владения ими. Также необходимо определить возможности тестирования помимо контролирующей функции.[1]

## 2 Обзор существующих программных решений

В качестве примера рассмотрим программное обеспечение Moodle – бесплатная онлайн система управления обучением, которая даёт возможность преподавателям генерировать собственный веб-сайт, наполненный динамическими курсами, которые независимо от места и времени расширяют возможности обучения[2]. Moodle удобен для большого количества участников образовательного процесса, таких как преподаватели, студенты и др. Moodle содержит в себе настраиваемое ядро с широким спектром стандартных функций.

Общие характеристики[2]: современный интерфейс, который прост в использовании на различных устройствах; персонализированная страница «Мой курс», отображающая набор курсов: текущие, прошлые и предстоящие; краткий обзор; инструменты и мероприятия для совместной работы; удобное управление файлами, в том числе возможность загружать файлы из облачных хранилищ; простые и понятные текстовые редакторы Atto и TinyMCE; уведомления; отслеживание прогресса.

Административные функции[2]: настраиваемые дизайн и макет; безопасная авторизация и регистрация; многоязычность; простое создание курсов с резервным копированием, а также возможность создавать курсы оптом, создавать резервные копии и восстанавливать курсы из созданных резервных копий; управление ролями и разрешениями пользователей; поддержка открытых стандартов электронного обучения; высокая совместимость; возможность легко управлять дополнениями; регулярные обновления и исправления безопасности; подробные отчёты и журналы.

Moodle поддерживает разработку курсов и управление ими в соответствии с запросами пользователей, способствует сотрудничеству посредством встроенных функций совместной публикации, поддерживает включение различных материалов и упражнений со сторонних сайтов, а также подключение к сведениям успеваемости, помогает в интеграции

мультимедийных ресурсов, например, видео и аудио, позволяет объединять обучающихся в группы, чтобы делать совместную работу легкой, разграничивать виды деятельности и др., способствует маркировке заданий, модерации отметок и производит текущий и итоговый контроль, имеет удобную встроенную функцию комментариев и отзывов, поддерживает функцию интегрированных значков. Семинары и опросы, созданные на платформе Moodle, позволяют обучающимся анализировать и давать оценку как своей работе, так и работе остальных студентов. На платформе Moodle можно работать и с методом оценивания в широком спектре, включающего критерии итогового контроля. Ещё одной полезной функцией Moodle является настройка компетенций. Moodle – безопасная и конфиденциальная платформа, которая разрешает в личном пространстве, доступном только преподавателям и студентам данного курса, делиться информацией.

Системные требования: 160 мегабайт свободного места на диске; 256 мегабайт оперативной памяти; программное обеспечение для веб-сервера (Apache, Nginx, IIS и другие); язык PHP; сервер баз данных (MySQL, PostgreSQL, Microsoft SQL Server, Oracle и другие)[2].

Moodle можно загрузить как в виде архива, так и через CVS.

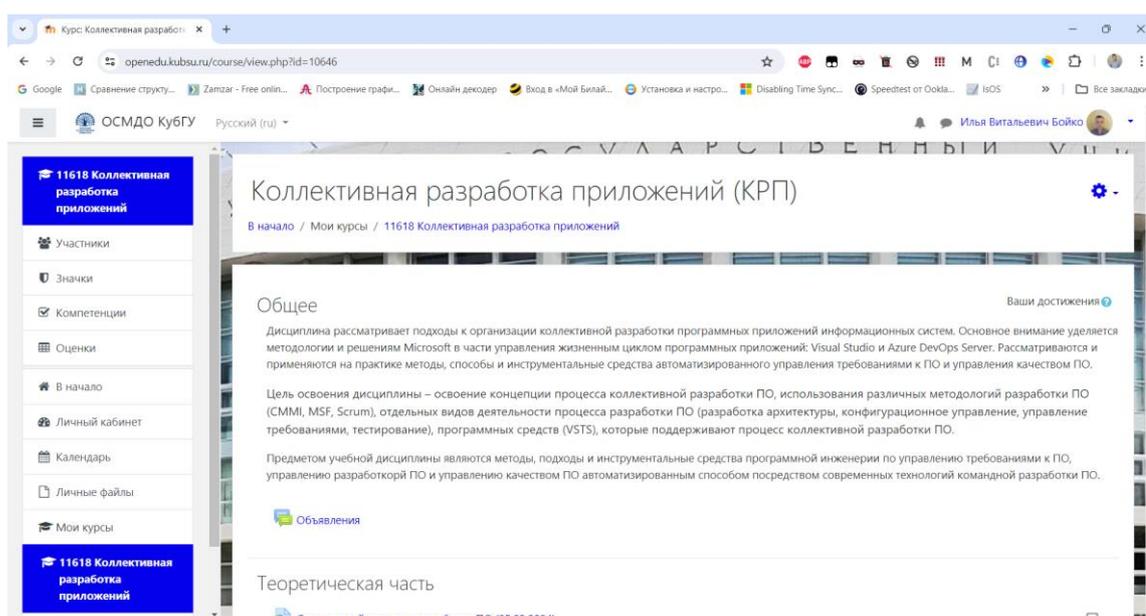


Рисунок 1 – Скриншот страницы Moodle

В Moodle каждому пользователю назначается соответствующая роль. Роль – набор разрешений, определённых для всей системы, назначаемых конкретным пользователем в определённых контекстах. В стандартной комплектации Moodle доступны следующие роли[2]: администратор, менеджер, создатель курса, преподаватель, преподаватель без редактирования, студент, гость и аутентифицированный пользователь. Ниже представлено описание каждой из ролей.

Администратор – имеет неограниченные полномочия.

Менеджер – имеет доступ к курсам и их изменению, к частичному выполнению функций администратора, непосредственно связанные с курсами, обучающимися, настройками отметок и т.п.

Создатель курса. Позволяет создавать курсы, а также просматривать скрытые курсы.

Преподаватель. Может делать практически всё в рамках курса, включая добавление и изменение видов деятельности и выставление оценок студентам.

Преподаватель без редактирования. Имеет право просмотра и оценивания работы обучающихся, но не имеет права на изменение и удаление любых действий или ресурсов.

Студент. Имеет доступ к курсам и может выбирать их.

Гость. Имеет доступ к курсам, но не может их выбирать.

Аутентифицированный пользователь. Назначается автоматически всем пользователям, впервые вошедших в систему. Им разрешено изменять свой профиль, отправлять комментарии, вести личный дневник, а также регулировать другие действия, не входящие в курсы.

Рассмотрим другой продукт под названием 1С:Электронное обучение, разработанный российской компанией 1С. Это система дистанционного обучения, разработанная на платформе «1С:Предприятие 8», которая используется для развития персонала предприятий, корпоративных учебных центров и образовательных учреждений. Далее перечислены особенности данного программного продукта: в нём сочетаются система дистанционного

обучения и средства разработки электронных учебных материалов, присутствует возможность проведения тестирования при помощи сети Интернет, в том числе, с мобильных устройств, а также допустимо использовать информацию ограниченного доступа и распространения при эксплуатации 1С:Электронное обучение в среде «1С:Предприятие 8»[22].

1С:Электронное обучение обладает следующей функциональностью[22]: разработка электронных курсов и тестов, единая среда общения, управление пользователями и встроенная интеграция. Ниже представлено описание каждой из функций.

Разработка электронных курсов и тестов. В 1С:Электронное обучение присутствует встроенный конструктор образовательных ресурсов, позволяющий разрабатывать и дорабатывать курсы и тесты.

Единая среда общения. В 1С:Электронное обучение существуют учебные форумы, при помощи которых преподаватель занимается организацией дискуссии по учебным заданиям в подгруппе обучающихся. Также присутствует система личных сообщений, автоматическая рассылка персональных сообщений на электронную почту и новости.

Управление пользователями. В стандартной комплектации доступны следующие роли пользователей: «Обучающийся», «Преподаватель», «Администратор обучения», «Старший менеджер по обучению», «Младший менеджер по обучению», «Методист» и «Администратор».

Встроенная интеграция 1С:Электронно обучение поддерживает обмен данными с другими продуктами компании 1С. Также реализованы HTTP-сервисы для передачи в 1С:Колледж и 1С:Управление учебным центром оценок, которые были получены в процессе обучения, обмен данными со свободным ПО для настройки вебинаром, обмен курсами между разными модулями 1С:Электронное обучение, экспорт и импорт курсов с использованием стандарта SCORM 2004 и органичная интеграция с программным обеспечением «1С:Электронное обучение. Веб-кабинет преподавателя и обучающегося»[22].

1С:Электронное обучение обладает следующими преимуществами[22]: интеграция с приложениями кадрового учёта, развития и оценки персонала и управления образовательными организациями, разработанными 1С, свобода конкурентного выбора организации и специалистов для внедрения и поддержки программных продуктов, открытый, структурированный и документированный исходный код приложений, позволяющий быстро дорабатывать проекты, созданные при помощи 1С:Электронное обучение.

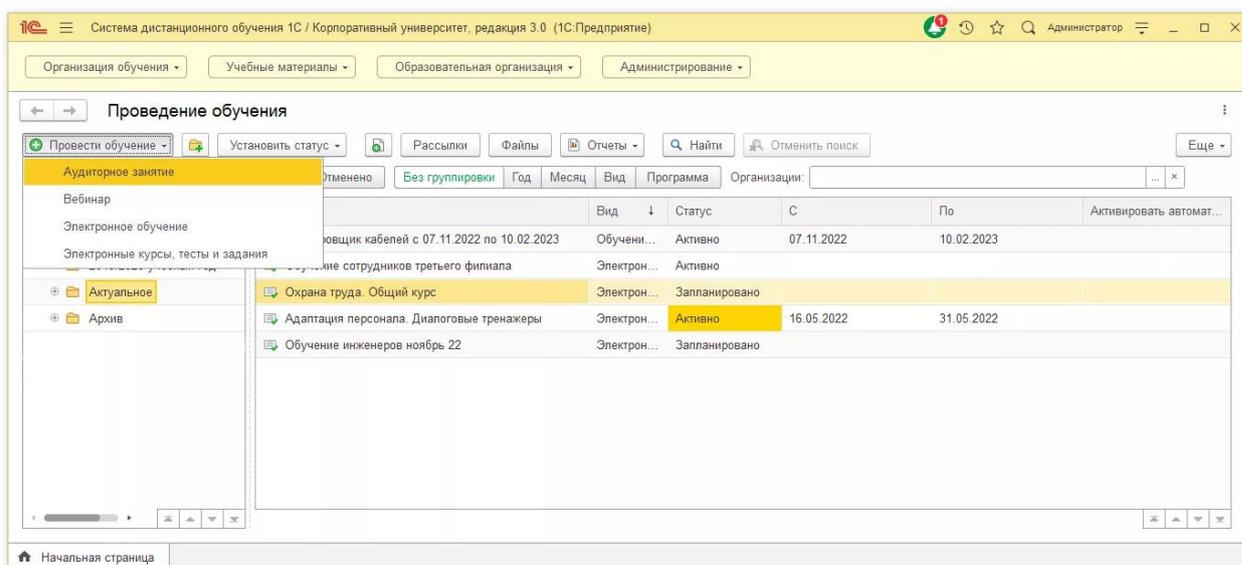


Рисунок 2 – Скриншот 1С:Электронное обучение

## 3 Стек технологий для разработки приложения

### 3.1 Стек технологий веб-приложения

При разработке веб-приложений предлагается использовать следующий стек технологий:

PostgreSQL — свободная объектно-реляционная СУБД.

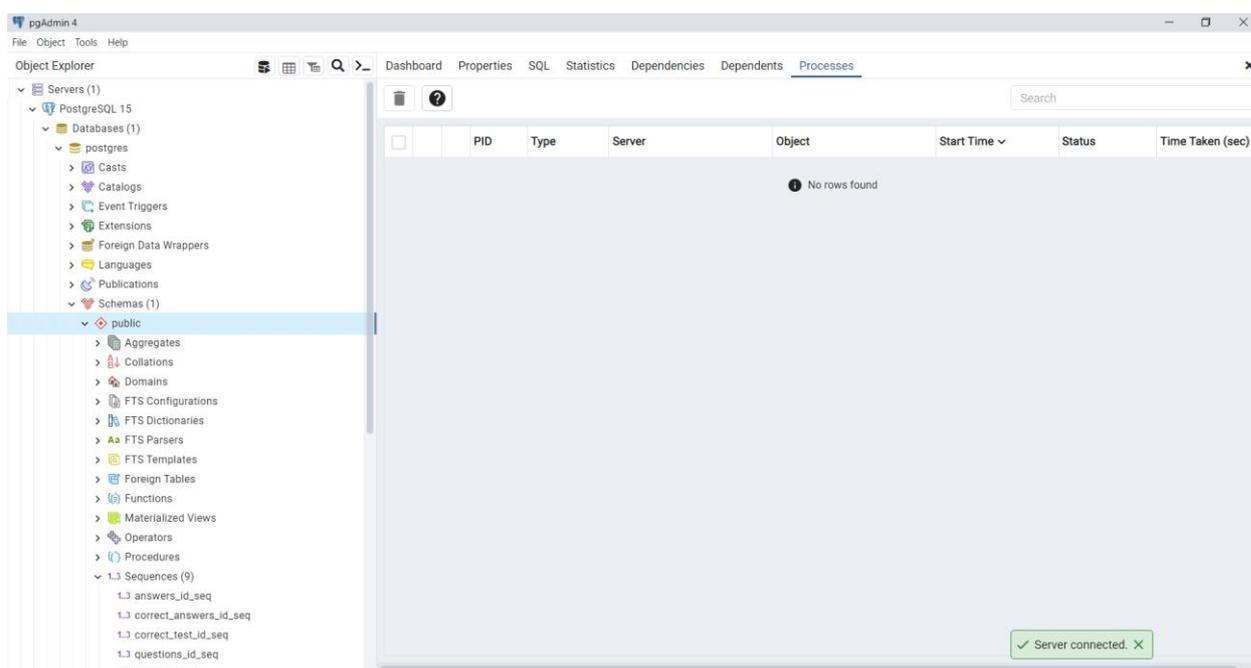


Рисунок 3 – Скриншот окна pgAdmin

Основными возможностями являются функции, триггеры и индексы[3]. Далее представлено их описание.

Функции – группа произвольных операторов на языке SQL, предназначенных для выполнения некоторой задачи. Могут выполнять различные операции (SELECT, INSERT, UPDATE, DELETE). Функции могут писаться на разных языках, по умолчанию, на встроенном языке PL/pgSQL.

Триггеры – это особый вид процедуры обработки события, который вызывается автоматически при выполнении определённых действий.[19]

Индексы – традиционное средство увеличения производительности базы данных.

По умолчанию PostgreSQL поддерживает следующие типы данных[3]: числовые, денежные, символьные, двоичные, дата и время, логический, типы перечислений, геометрические, сетевые адреса, битовые строки, типы, предназначенные для текстового поиска, UUID, XML, типы JSON, массивы, составные типы, диапазонные типы, типы доменов, идентификаторы объектов БД, тип `pg_lsn` и псевдотипы.

ngrok — это глобально распределённый обратный прокси-сервер, защищающий и ускоряющий приложения и сетевые службы, независимо от того, где они запускаются[17]. Также позволяет получить URL адрес для публичного доступа других пользователей к сайту. Для использования ngrok необходимо выполнить следующие шаги:

- 1) зарегистрироваться на сайте [ngrok.com](https://ngrok.com)
- 2) скачать ngrok с сайта [ngrok.com](https://ngrok.com)
- 3) в командной строке выполнить команду:  
`ngrok config add-authtoken <authtoken>`  
где `authtoken` – ключ, полученный при регистрации
- 4) запустить ngrok при помощи команды:  
`ngrok http http://localhost:80`

Результат:

```
C:\Windows\system32\cmd.exe x + v
ngrok (Ctrl+C to quit)
K8s Gateway API support available now: https://ngrok.com/r/k8sgb

Session Status      online
Account             iliaboyko1@gmail.com (Plan: Free)
Update              update available (version 3.9.0, Ctrl-U to update)
Version              3.4.0
Region              Europe (eu)
Latency              -
Web Interface        http://127.0.0.1:4040
Forwarding           https://75b3-212-192-144-177.ngrok-free.app -> http://localhost:80

Connections
  ttl   opn   rt1   rt5   p50   p90
   0     0    0.00  0.00  0.00  0.00
```

Рисунок 4 – Запуск ngrok

PHP - это встраиваемый в HTML-документы язык сценариев, выполняемый на стороне сервера.[18] PHP встраивается в код HTML следующим образом:

```
<?php ... ?>
```

В стандартной комплектации PHP предоставляет большой набор встроенных средств веб-разработки[4]: HTTP-аутентификация, постоянные соединения с базами данных, cookies, сессии, управление файлами и каталогами, работа с соединениями, работа из командной строки и сборка мусора. Для использования PostgreSQL в PHP необходимо добавить следующие строки в файл php.ini:

```
extension=pdo_pgsql
extension=pgsql
```

JavaScript — сценарный язык программирования. Является одним из самых популярных языков для разработки веб-приложений.

JavaScript используется прежде всего для создания клиентских веб-приложений, но может применяться и с серверной стороны[5].

JavaScript поддерживает следующие типы данных[5]: String, Number, BigInt, Boolean (true или false), Undefined (указывает, что значение не

определено), Null (указывает, что значение отсутствует), Symbol (встроенный объект, конструктор которого возвращает примитив symbol, часто использующийся для добавления к объекту уникальных ключевых свойств) и Object (представляет собой сложный объект).

JavaScript поддерживает анонимные функции, слабую и динамическую типизацию, встроенные объекты (Date, Math, Number, Proxy), работу с JSON и XML, автоматическую сборку мусора и другие возможности[5].

Apache HTTP-сервер — свободное кроссплатформенное программное обеспечение для размещения и поддержки веб-сервера.[6] Apache позволяет открывать динамические PHP-документы, распределять нагрузку на сервер, обеспечивать его отказоустойчивость, тренироваться в запуске PHP-скриптов и настройке сервера[6].

Основными преимуществами Apache являются гибкость, кроссплатформенность, доступность, функциональность, масштабируемость, поддержка множества различных языков программирования и техническая поддержка. Сервер состоит из ядра и модулей. Ядро включает в себя основные возможности, такие как обработку файлов конфигурации, HTTP и загрузку модулей. Модули представляют собой программные блоки, которые подключаются к ядру веб-сервера и расширяют его функционал. На сегодняшний день доступны более 500 модулей для веб-сервера Apache[6].

Apache позволяет ограничивать доступ к файлам и каталогам, в том числе по их типам и по IP-адресам пользователей. Также Apache поддерживает авторизацию пользователей на основе HTTP-аутентификации и digest-аутентификации.

Nginx – это свободный HTTP-сервер и обратный прокси-сервер, а также TCP/UDP сервер общего назначения.[7] В основе архитектуры Nginx лежат основной и различные рабочие процессы, гибкая конфигурация, настройки, которые можно изменять, не тревожа обслуживание пользователей, расход в среднем 2,5 мегабайт памяти на 10000 неактивных keep-alive соединений и минимальное количество операций копирования[7].

Nginx поддерживает следующую основную функциональность, которая позволяет обслуживать статические запросы, создание списка файлов и индексных файлов, обратно проксировать, распределять нагрузку и отказоустойчивость, поддерживать FastCGI и SCGI сервера, модули, SSL и TLS SNI, HTTP/2 и HTTP/3[7].

К прочим возможностям сервера относятся[7]: виртуальные серверы, поддержка keep-alive соединений, настройка форматов файлов журнала и буферизированная запись в файлы журнала, страницы для ошибок 3xx, 4xx и 5xx, блокировка IP-адресов и аутентификация по имени пользователя и паролю, проверка заголовка запроса HTTP referrer, ограничение скорости ответа и числа одновременных соединений с одного IP-адреса, определение местоположения по IP-адресу и зеркалирование запросов.

Django — это высокоуровневый фреймворк на языке Python, созданный для разработки современных и безопасных веб-приложений.[8]

С использованием Django сайт состоит из одного или нескольких приложений. Веб-приложения, созданные с использованием Django обладают следующими свойствами[8]: полнота, разносторонность, безопасность, масштабируемость, удобство в сопровождении и переносимость.

Django также предоставляет следующие дополнительные возможности[8]: HTML-формы, аутентификация пользователя и разрешения, кэширование, административная панель, обслуживание данных в форматах XML и JSON, возможность создания RESTful API с помощью Django REST Framework, интеграция с облачными хранилищами данных и интегрированная защита от наиболее распространённых дыр в безопасности.

CGI является технологией, которая способствует приложениям, написанными с использованием множества различных языков программирования, работать на веб-сервере.[9] Последний вариант описан в RFC 3875. Шлюзом является веб-сервер, который получает запрос от клиента, преобразует в CGI-форму, вызывает обработчик и конвертирует его ответ из CGI-формы в форму HTTP-ответа клиенту[23].

CGI позволяет запускать приложения, написанные на любом языке программирования, который поддерживает операции со стандартными устройствами ввода-вывода. Скрипты помещаются в каталог cgi-bin.

HTML - это язык разметки гипертекста, используемый преимущественно для создания документов в сети Интернет.[10]

HTML-документ состоит из блоков, которые называются элементами. В свою очередь, каждый элемент состоит из тегов. Как правило, HTML-документ имеет следующую структуру[10]:

```
<!DOCTYPE html>
<html>
<head>
...
</head>
<body>
...
</body>
</html>
```

Элемент doctype сообщает браузеру тип документа. Элемент html содержит всё содержимое документа. Элемент head является заголовком веб-страницы, а элемент body определяет содержимое самой веб-страницы.

В HTML можно встроить язык программирования JavaScript с помощью тега. Также при помощи тега <style> можно определить стили элементов веб-страницы[10].

CSS — язык, определяющий представление HTML-документа и его внешний вид. Также может применяться к любым XML-документам, например, к SVG или XUL.

Стилем в CSS является правило, указывающее браузеру способ форматирования элемента.[10] CSS разрабатывался с целью отделения описания структуры веб-страницы от её внешнего вида, что увеличивает гибкость HTML-документа и уменьшает его сложность. Вынесение стилей в отдельный файл с расширением .css делая HTML-код чище и компактнее.

## 3.2 Стек технологий мобильного приложения

При разработке мобильных приложений предлагается использоваться следующий стек технологий:

Kotlin – это современный, статически типизированный язык программирования с открытым исходным кодом, который был создан и развивается компанией JetBrains.[11] Kotlin применяется в первую очередь для разработки мобильных приложений под операционной системой Android. Данный язык программирования обладает следующими особенностями: блоки кода ограничены открывающейся и закрывающейся фигурными скобками,

Java является языком программирования, который был разработан корпорацией Sun Microsystems. Ключевой особенностью данного языка программирования является трансляция кода приложения перед его исполнением в байт-код, не зависящий от архитектуры и платформы. После этого он выполняется в виртуальной машине JVM. Кроме того, в языке Java присутствует система сборки мусора (неиспользуемые переменные, память и т.п.), очищающая память от более не используемых объектов[12].

В настоящее время Java – один из самых используемых языков программирования для разработки разнообразных приложений: веб-сайты и веб-сервисы, программы для настольных компьютеров, мобильные приложения для операционной системы Android.[12]

Gradle является системой автоматической сборки для написания приложений на языках Java и Kotlin. Gradle автоматизирует сборку, тестирование и развёртывание программного обеспечения на основе информации в сценариях сборки.[13] Сценарии сборки подробно описывают в Gradle, какие шаги необходимо предпринять для сборки проекта. Каждый проект может включать в себя один или несколько сценариев сборки. Также Gradle включает в себя управление зависимостями – автоматизированный метод объявления и разрешения внешних ресурсов, необходимых проекту. Каждый проект обычно включает в себя ряд внешних зависимостей,

разрешаемых Gradle во время сборки. В Gradle присутствуют задачи – базовые единицы работы, такие как компиляция кода или запуск теста. Каждый проект содержит одну или несколько задач, определённых внутри сценария сборки или плагина.[13]

Apache Maven является фреймворком, применяемым для автоматической сборки проектов. Наиболее распространено его применение при разработке приложений на языке программирования Java[14].

Во время выполнения приложения Apache Maven выполняет компиляцию и управление зависимостями и ресурсами . Maven обладает следующими преимуществами[14]: он используется в средах разработки, таких как Eclipse, IntelliJ IDEA, NetBeans и многих других, может работать на любой операционной системе, для его использования не требуется установка, параметры имеют оптимальные настройки по умолчанию, он упрощает командную работу и документирование приложения, запускает библиотеки для модульного тестирования, обеспечивает соблюдение основных стандартов и имеет большое количество различных расширений.

Android Studio — среда разработки для написания мобильных приложений для операционной системы Android. Является официальным средством разработки Android-приложений[15].

Основным ограничением является отсутствие возможности полностью отключить автосохранение файлов.

Android Studio поддерживает сборку при помощи Gradle[15].

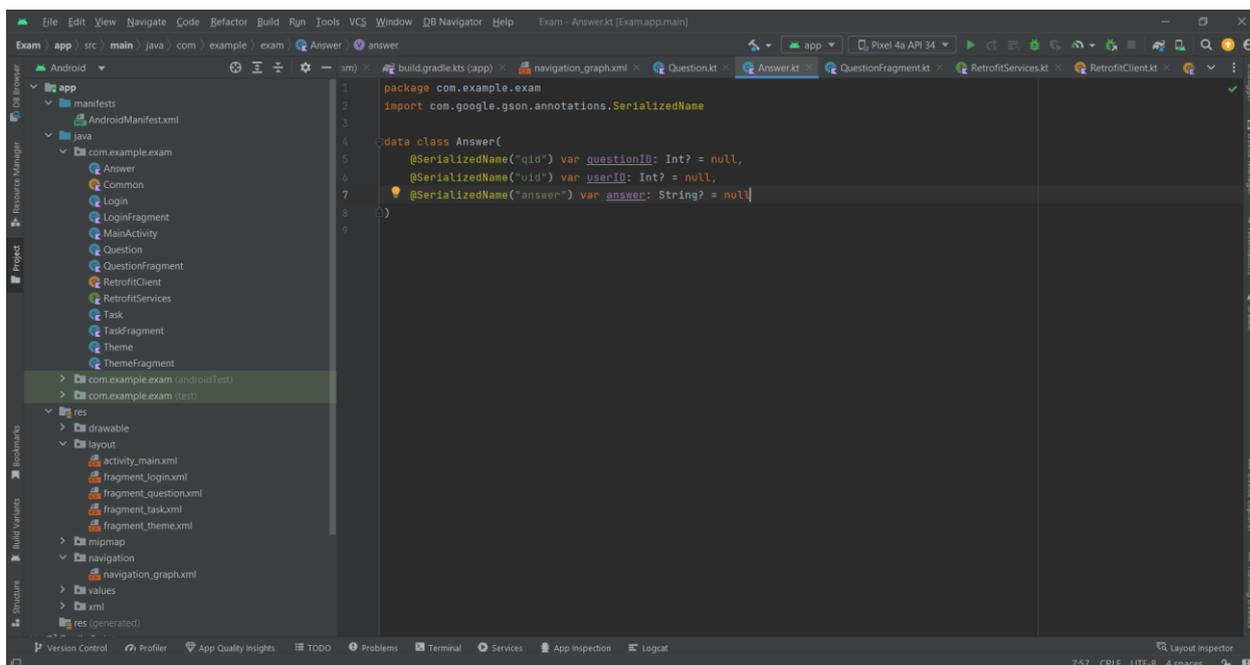


Рисунок 5 – Скриншот окна Android Studio

Retrofit — типобезопасный HTTP-клиент для Android и Java[16]. Для использования библиотеки retrofit2 необходимо добавить следующие строки в файл build.gradle в раздел dependencies:

```
implementation("com.squareup.retrofit2:retrofit:2.9.0")
implementation("com.squareup.retrofit2:converter-gson:2.9.0")
```

Также добавляем раздел repositories:

```
repositories {
    google()
    maven {
        url = uri("https://sdk.squareup.com/android")
    }
}
```

Реализация интерфейса RetrofitServices:

```
package com.example.exam
import com.example.exam.Theme
import com.example.exam.Task
import com.example.exam.Question
import retrofit2.Call
import retrofit2.http.*

interface RetrofitServices {
    @GET("android.php?get=themes")
```

```

fun getThemes(): Call<MutableList<Theme>>

@GET("android.php?get=tasks")
fun getTasks(@Query("id") themeId: Int?): Call<MutableList<Task>>

@GET("android.php?get=question")
fun getQuestions(@Query("id") taskId: Int?, @Query("number")
questionNumber: Int?,
                @Query("uid") userID: Int?): Call<Question>

@GET("android.php?get=login")
fun getLogin(@Query("user") user: String?, @Query("pass")
password: String?): Call<Login>

@Headers("Content-Type: application/json")
@POST("android.php")
fun postAnswer(@Body data: Answer): Call<Answer>
}

```

Реализация объекта RetrofitClient:

```

package com.example.exam
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory

object RetrofitClient {
    private var retrofit: Retrofit? = null

    fun getClient(baseUrl: String): Retrofit {
        if (retrofit == null) {
            retrofit = Retrofit.Builder()
                .baseUrl(baseUrl)
                .addConverterFactory(GsonConverterFactory.create())
                .build()
        }
        return retrofit!!
    }
}
}

```

Реализация объекта Common:

```

package com.example.exam
import com.example.exam.RetrofitServices
import com.example.exam.RetrofitClient

object Common {
    private val BASE_URL = "http://10.0.2.2/"
}

```

```
    val retrofitService: RetrofitServices
        get() =
RetrofitClient.getClient(BASE_URL).create(RetrofitServices::class.java
)
}
```

## **4 Постановка задачи**

Целью данной работы является написание веб-приложения, позволяющего преподавателям создавать и изменять тесты, а студентам проходить тесты. Помимо этого, стоит задача разработки мобильного приложения, позволяющего студентам проходить тесты. Для реализации в данном приложении выбраны обучающие тесты в открытой и закрытой формах с выбором одного варианта ответа, поскольку их проще всего реализовать.

### **4.1 Постановка задачи разработки веб-приложения**

Разрабатываемое веб-приложение должно обеспечивать возможность преподавателям создавать и изменять тесты, а студентам регистрироваться, проходить тесты и просматривать статистику выполнения заданий.

Веб-приложение состоит из серверной и клиентской части. Серверная часть обращается к базе данных, получает из неё данные, отправляет их в клиентскую часть, от неё получает данные и записывает их в базу данных. Клиентская часть получает данные от серверной части, обрабатывает их и выводит на экран, затем ожидает ввода данных от пользователя, и после того, как данные получены, отправляет их в серверную часть. В веб-приложении используется следующий стек технологий: Apache версии 2.4.53 (обеспечивает работу веб-сервера), PHP версии 8.2.8 (используется для написания серверной части веб-приложения), PostgreSQL версии 15.3 (используется для разработки базы данных), nginx версии 3.4.0 (делает веб-сервер общедоступным), HTML (используется для написания компонента клиентской части веб-приложения, отвечающего за вывод информации на экран), CSS (используется для описания внешнего вида клиентской части веб-приложения) и JavaScript (используется для изменения содержимого клиентской части веб-приложения).

## **4.2 Постановка задачи разработки мобильного приложения**

Разрабатываемое мобильное приложение должно обеспечивать возможность студентам проходить тесты.

Мобильное приложение состоит из клиентской части, которая по сети обращается к серверной части разработанного веб-приложения: получает от него данные и отправляет введенные пользователем данные.

В приложении используется следующий стек технологий: Android Studio Giraffe версии 2022.3 (используется непосредственно для создания мобильного приложения), Kotlin (является составной частью среды разработки и используется для написания кода приложения), Gradle (является составной частью среды разработки и используется для определения зависимостей и импорта библиотек) и библиотека retrofit2 версии 2.9.0 (обеспечивает сетевое взаимодействие между мобильным приложением и серверной частью веб-приложения).

## 5 Описание приложения

На рисунке ниже представлена диаграмма базы данных, созданная при помощи инструмента ERD в pgAdmin:

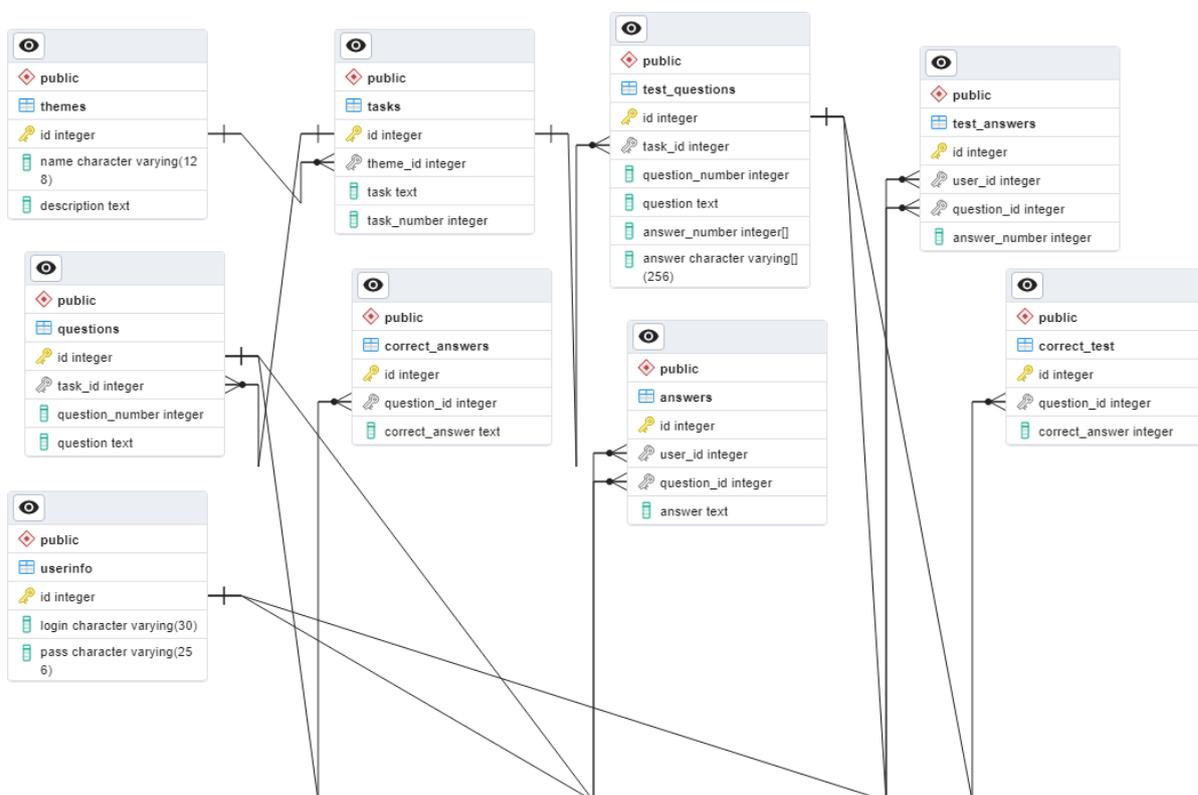


Рисунок 6 – Структура базы данных

База данных состоит из следующих таблиц: userinfo, themes, tasks, test\_questions, questions, correct\_test, correct\_answers, test\_answers и answers. Далее представлено описание таблиц.

Таблица userinfo хранит информацию о пользователях.

Таблица 1 – Структура таблицы userinfo

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор пользователя
login	character varying(30)	нет	Имя пользователя
pass	character varying(256)	нет	Хеш пароля пользователя

Таблица themes хранит информацию о темах тестов.

Таблица 2 – Структура таблицы themes

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор темы
name	character varying(128)	нет	Название темы
description	text	нет	Описание темы

Таблица tasks хранит информацию о заданиях тестов.

Таблица 3 – Структура таблицы tasks

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор задания
theme_id	integer	нет	Идентификатор темы, к которой относится задание
task	text	нет	Текст задания
task_number	integer	нет	Номер задания в теме

Таблица test\_questions хранит информацию о вопросах с выбором варианта ответа.

Таблица 4 – Структура таблицы test\_questions

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор вопроса с выбором варианта ответа
task_id	integer	нет	Идентификатор задания, к которому относится вопрос с выбором варианта ответа
question_number	integer	нет	Номер вопроса с выбором варианта ответа в задании
question	text	нет	Текст вопроса с выбором варианта ответа
answer_number	integer[]	нет	Номера вариантов ответа на вопрос
answer	character varying[](256)	нет	Варианты ответа на вопрос

Таблица questions хранит информацию о вопросах с развернутым ответом.

Таблица 5 – Структура таблицы questions

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор вопроса с развёрнутым ответом
task_id	integer	нет	Идентификатор задания, к которому относится вопрос с развёрнутым ответом
question_number	integer	нет	Номер вопроса с развёрнутым ответом в задании
question	text	нет	Текст вопроса с развёрнутым ответом

Таблица correct\_test хранит информацию о правильных ответах на вопросы с выбором варианта ответа.

Таблица 6 – Структура таблицы correct\_test

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор связи
question_id	integer	нет	Идентификатор вопроса с выбором варианта ответа
correct_answer	integer	нет	Номер правильного варианта ответа на вопрос

Таблица correct\_answers хранит информацию о правильных ответах на вопросы с развернутым ответом.

Таблица 7 – Структура таблицы correct\_answers

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор связи
question_id	integer	нет	Идентификатор вопроса с развёрнутым ответом
correct_answer	text	нет	Правильный ответ на вопрос с развёрнутым ответом

Таблица test\_answers хранит информацию об ответах пользователей на вопросы с выбором варианта ответа.

Таблица 8 – Структура таблицы test\_answers

Столбец	Тип данных	Первичный ключ	Описание
id	serial	да	Уникальный идентификатор записи в таблице
user_id	integer	нет	Идентификатор пользователя
question_id	integer	нет	Идентификатор вопроса с выбором варианта ответа
answer_number	integer	нет	Вариант ответа на вопрос, выбранный пользователем

Таблица `answers` хранит информацию об ответах пользователей на вопросы с развёрнутым ответом.

Таблица 9 – Структура таблицы `answers`

Столбец	Тип данных	Первичный ключ	Описание
<code>id</code>	<code>serial</code>	да	Уникальный идентификатор записи в таблице
<code>user_id</code>	<code>integer</code>	нет	Идентификатор пользователя
<code>question_id</code>	<code>integer</code>	нет	Идентификатор вопроса с развёрнутым ответом
<code>answer</code>	<code>text</code>	нет	Ответ на вопрос, отправленный пользователем

## 5.1 Описание веб-приложения

Веб-приложение состоит из следующих модулей: `index.php`, `login.php`, `register.php`, `test.php`, `create.php`, `edit.php` и `stats.php`. Ниже представлено подробное описание каждого из модулей.

`index.php` – главная страница. Если выполняется запрос методом GET, то в случае, если не выполнен вход в систему, происходит перенаправление на страницу входа в систему `login.php`. Если же вход выполнен, то открывается главная страница со ссылками на прохождение теста (`test.php`) и статистику (`stats.php`). В случае, когда вход выполнен под администратором, доступны также ссылки на создание теста (`create.php`) и изменение теста (`edit.php`). Также на странице присутствует кнопка «Выйти», которая отправляет запрос методом POST на страницу `index.php`, где происходит выход из системы. Если отправлен запрос методом POST, то происходит добавление или обновление ответов в базе данных.

login.php – страница входа в систему. На ней присутствуют поля для ввода логина и пароля и кнопка «Войти», при нажатии на которую происходит проверка логина и пароля, и если они совпадают с хешем в базе данных, то происходит запись логина и пароля в переменную сессии и перенаправление на главную страницу index.php

register.php – страница регистрации нового пользователя. На ней присутствуют поля для ввода логина и пароля и кнопка «Зарегистрироваться», при нажатии на которую происходит запись в базу данных логина и хеша пароля и перенаправление на страницу login.php

test.php – страница прохождения теста. При её открытии происходит запрос из базы данных заданий и вопросов с выбором варианта ответов и с развёрнутым ответом. После того, как пользователь выберет варианты ответа и введёт в текстовые поля ответы на вопросы с развёрнутым ответом и нажмёт на кнопку «Отправить», методом POST будет выполнен запрос к главной странице index.php, где ответы будут занесены в базу данных или обновлены в ней.

create.php – страница создания теста. При её открытии появляются поля с темой, заданиями, вопросами и ответами на них, которые вводятся пользователем и могут добавляться в любом количестве и удаляться при помощи кнопок «Добавить задание», «Удалить задание», «Добавить вопрос», «Добавить развёрнутый вопрос», «Удалить вопрос», «Добавить ответ» и «Удалить ответ». При нажатии на кнопку «Отправить» происходит вставка темы, заданий, вопросов и ответов на них в базу данных.

edit.php – страница редактирования теста. Реализует ту же функциональность, что и страница создания теста create.php, но при нажатии на кнопку «Отправить» происходит изменение уже существующих темы, заданий, вопросов и ответов на них в базе данных и добавление новых заданий, вопросов и ответов на них.

stats.php – страница статистики прохождения тестов. При её открытии происходит запрос из базы данных всех тем, затем по каждой теме всех

вопросов с выбором варианта ответа и с развёрнутым ответом и сравнение ответом, данных пользователями, с правильными ответами. В таблице в каждой строке напротив колонки с номером вопроса будет в зависимости от правильности ответа написано «Да» или «Нет». Если на все вопросы в теме будет дан правильный ответ, то после таблицы будет написано «Тема пройдена», в противном случае «Тема не пройдена».

На рисунке ниже представлен граф связи между компонентами приложения:

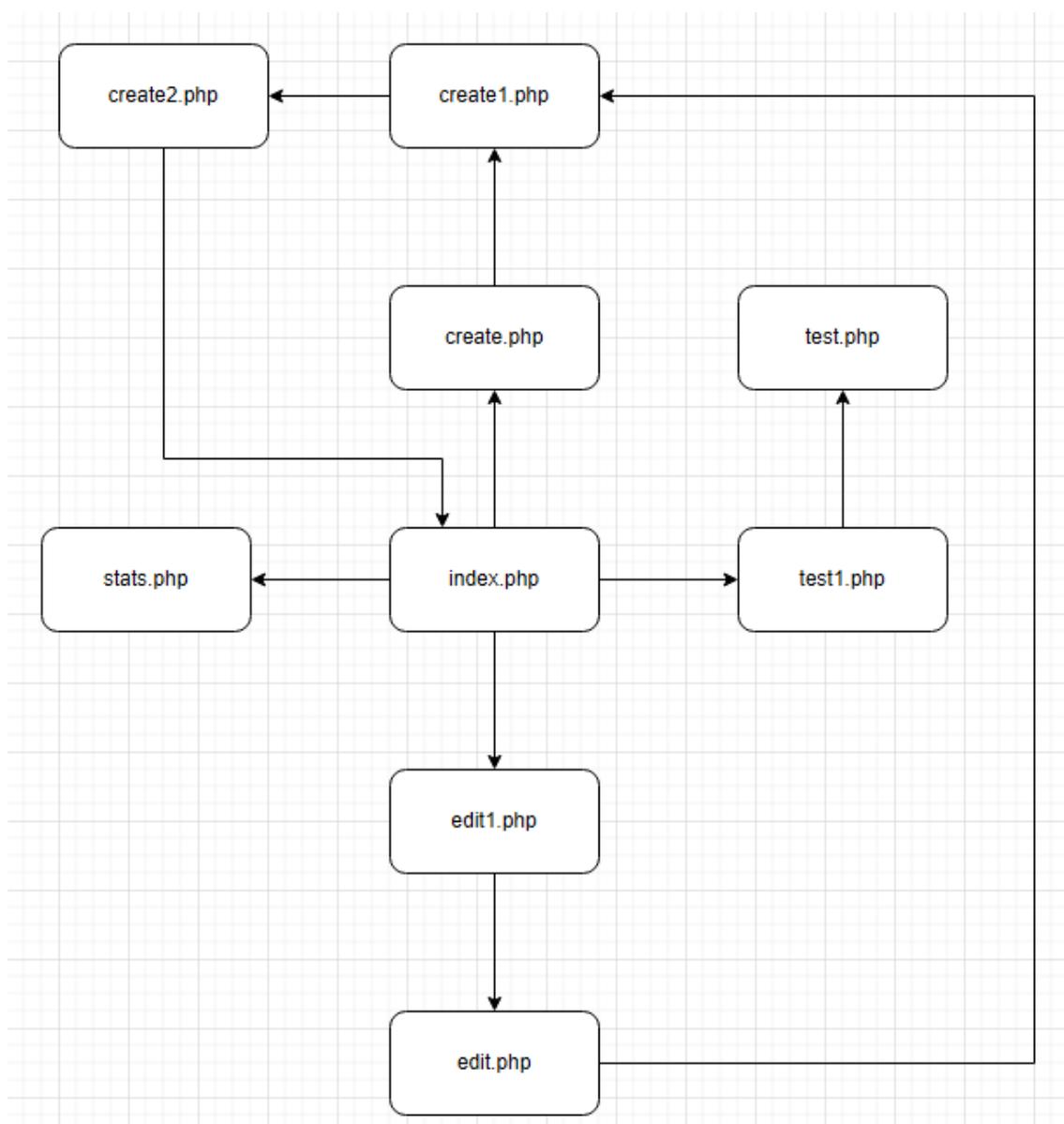


Рисунок 7 – Граф компонентов веб-приложения

Ниже представлены скриншоты главной страницы и страницы прохождения теста:

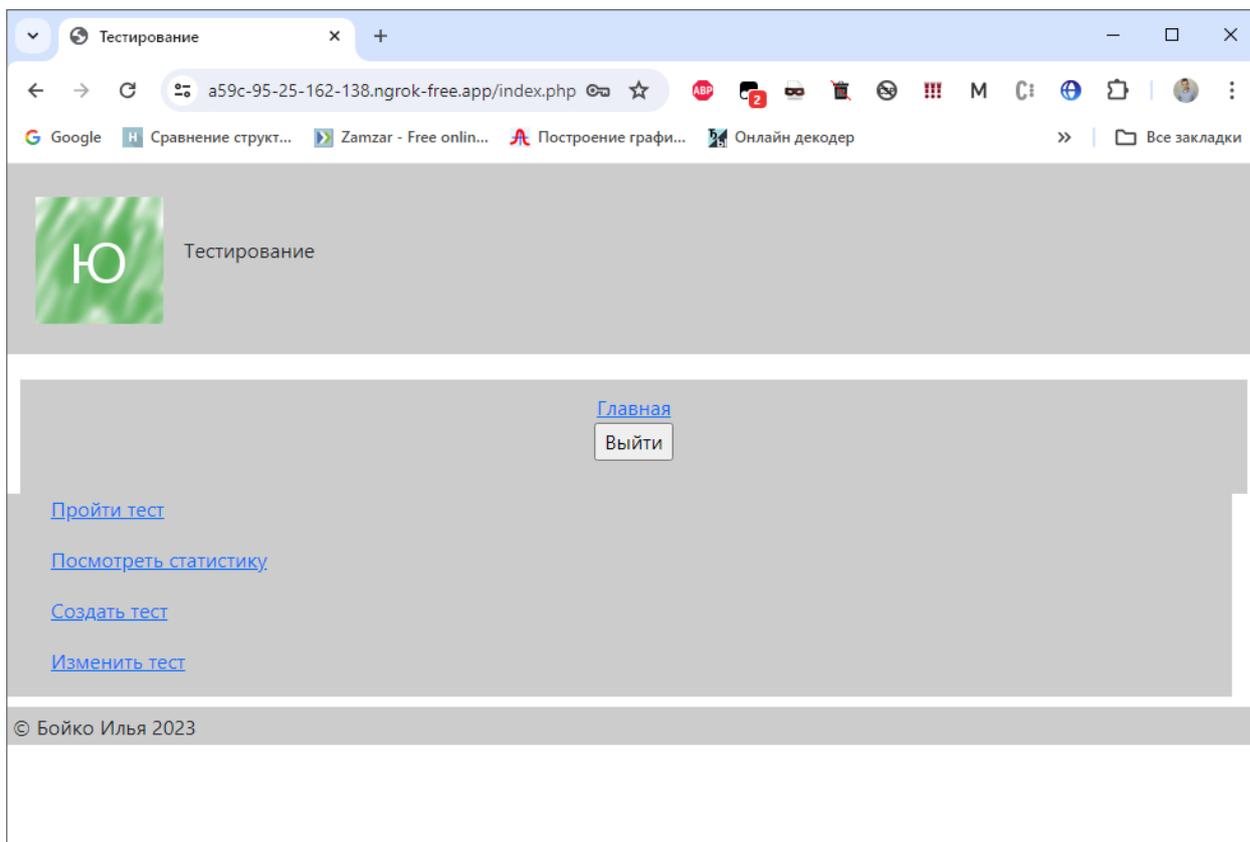


Рисунок 8 – Скриншот главной страницы веб-приложения

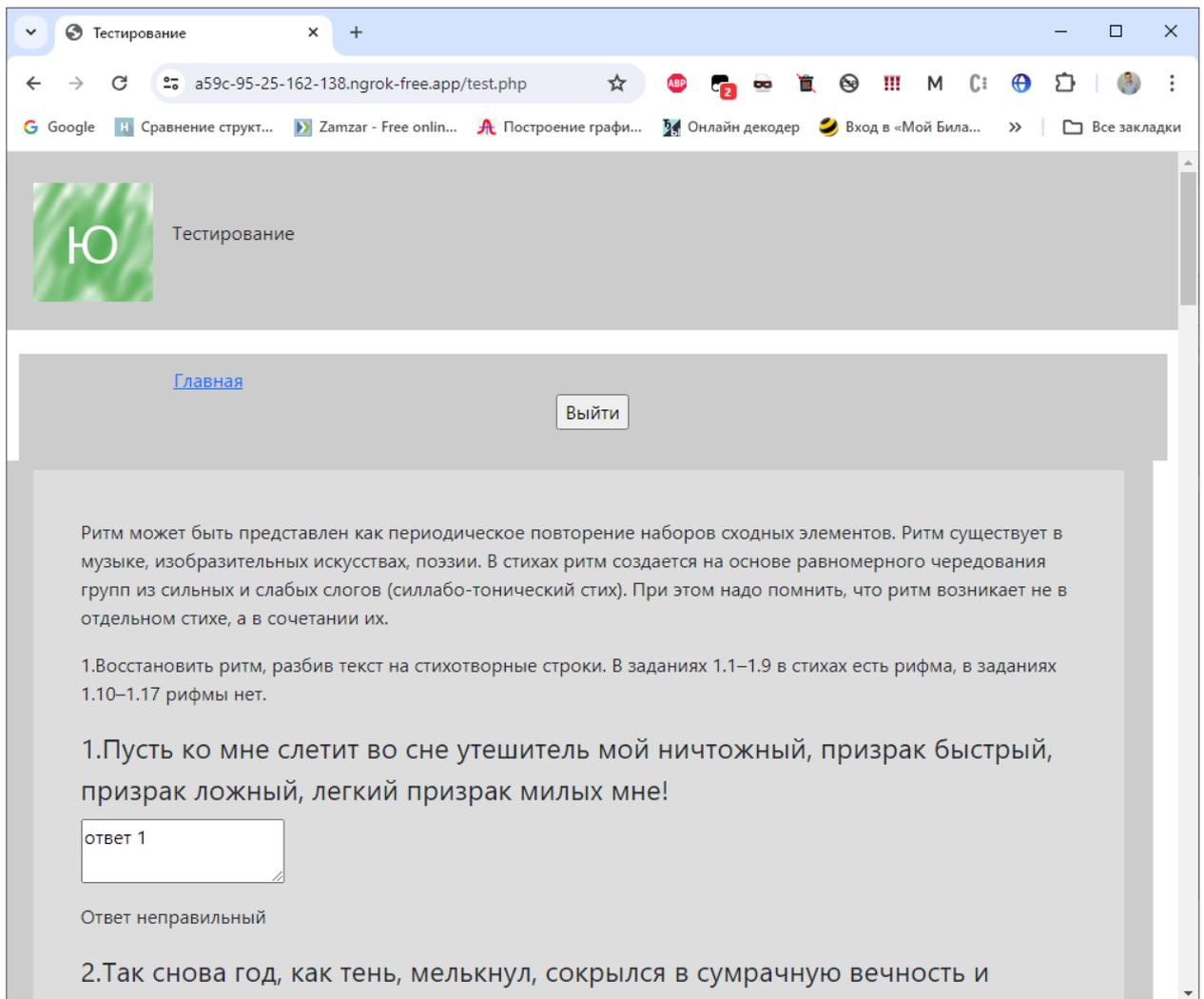


Рисунок 9 – Скриншот страницы прохождения теста

## 5.2 Описание мобильного приложения

На рисунке ниже представлен граф навигации мобильного приложения:

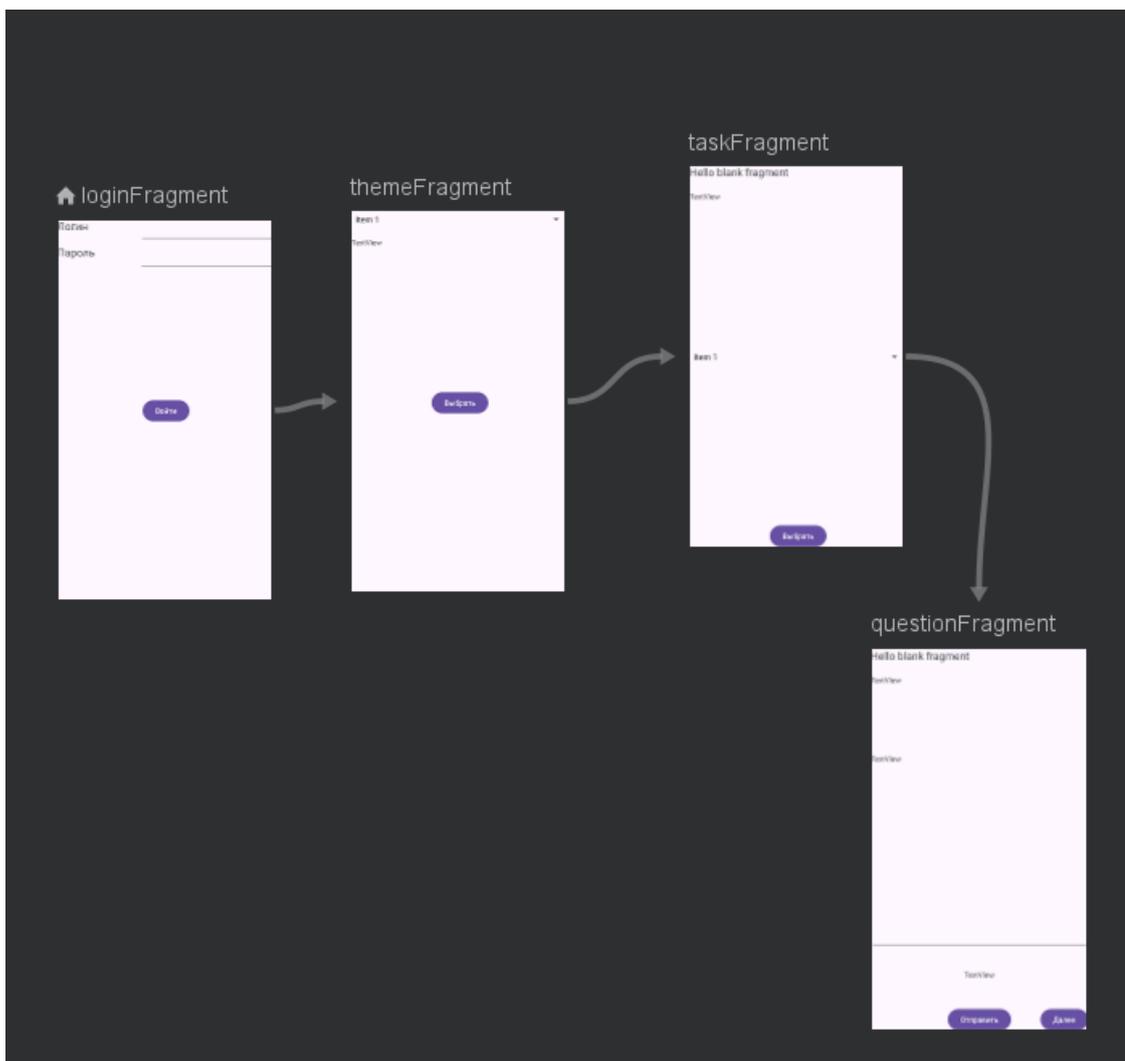


Рисунок 10 – Граф навигации мобильного приложения

На странице входа loginFragment присутствуют поля для ввода логина и пароля и кнопка «Войти», при нажатии на которую происходит обращение к серверу, на котором происходит проверка правильности логина и пароля, и если они верны, то происходит запись в память id пользователя и переход на страницу выбора темы themeFragment. На странице выбора темы themeFragment присутствуют выпадающий список тем, получаемый с сервера, описание тем и кнопка «Выбрать», при нажатии на которую происходит выбор темы и переход на страницу выбора задания taskFragment. На странице taskFragment присутствуют выпадающий список заданий, получаемый с сервера, текст заданий и кнопка «Выбрать», при нажатии на которую происходит выбор задания и переход на страницу прохождения теста

questionFragment. На странице questionFragment присутствуют текст задания, текст вопроса, получаемый с сервера, кнопка «Далее», при нажатии на которую происходит получение следующего вопроса с сервера, кнопка «Назад», при нажатии на которую происходит получение предыдущего вопроса с сервера, текст, принимающий значения «Ответ правильный» и «Ответ неправильный», и кнопка «Отправить», при нажатии на которую происходит отправка на сервер ответа на вопрос.

## **ЗАКЛЮЧЕНИЕ**

В ходе проделанной работы были изучены методы проведения тестирования, существующие на рынке программные продукты для проведения тестирования и стек технологий, при помощи которых можно разрабатывать веб-приложения и мобильные приложения.

Была разработана и создана база данных, содержащая таблицы с информацией о пользователях, темах, заданиях, тестовых вопросах и вопросах с развёрнутым ответом, правильных ответах на тестовые вопросы и вопросах с развёрнутым ответом и ответах пользователей на тестовые вопросы и вопросы с развёрнутым ответом.

Были достигнуты все цели работы. Разработано веб-приложение, позволяющее преподавателям создавать и изменять тесты, а студентам регистрироваться, проходить тесты и просматривать статистику выполнения заданий. Также разработано мобильное приложение, позволяющее студентам проходить тесты. Разработанные приложения пригодны для использования в образовательном процессе в школах и университетах.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рушингина, О. Использование обучающих тестов в учебном процессе / О. Рушингина, Ю. Правдин. – Cyberleninka: сайт. – URL: <https://cyberleninka.ru/article/n/ispolzovanie-obuchayuschih-testov-v-uchebnom-protssesse> (дата обращения: 16.05.2024).
2. MoodleDocs: сайт. – URL: [https://docs.moodle.org/402/en/Main\\_page](https://docs.moodle.org/402/en/Main_page) (дата обращения: 19.05.2024).
3. Документация к PostgreSQL 16.2. – PostgresPro: сайт. – URL: <https://postgrespro.ru/docs/postgresql/16/index> (дата обращения: 19.05.2024).
4. Руководство по PHP. – PHP: сайт. – URL: <https://www.php.net/manual/ru/index.php> (дата обращения: 19.05.2024).
5. Руководство по JavaScript. – Metanit: сайт. – URL: <https://metanit.com/web/javascript/> (дата обращения: 19.05.2024).
6. Apache HTTP-сервер: что это такое и как работает. – Skillfactory Media: сайт. – URL: <https://blog.skillfactory.ru/glossary/apache/> (дата обращения: 19.05.2024).
7. nginx - nginx.org: сайт. – URL: <https://nginx.org/ru/> (дата обращения: 19.05.2024).
8. Веб-фреймворк Django (Python) – Изучение веб-разработки. – MDN: сайт. – URL: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django> (дата обращения: 19.05.2024).
9. Что значит CGI и как его включить? – Handyhost: сайт. – URL: <https://handyhost.ru/help/term/cgi-skriptyi.html> (дата обращения: 19.05.2024).
10. Руководство по HTML5 и CSS3 – Metanit: сайт. – URL: <https://metanit.com/web/html5/> (дата обращения: 19.05.2024).
11. Руководство по языку Kotlin – Metanit: сайт. – URL: <https://metanit.com/kotlin/tutorial/> (дата обращения: 19.05.2024).
12. Java – Metanit: сайт. – URL: <https://metanit.com/java/> (дата обращения: 19.05.2024).

13. Gradle Basics. – Gradle Docs: сайт. – URL: [https://docs.gradle.org/current/userguide/gradle\\_basics.html](https://docs.gradle.org/current/userguide/gradle_basics.html) (дата обращения: 19.05.2024).
14. Apache Maven: что за фреймворк – программирование на Java. – Skillfactory Media: сайт. – URL: <https://blog.skillfactory.ru/glossary/maven/> (дата обращения: 19.05.2024).
15. Meet Android Studio. – Android Developers: сайт. – URL: <https://developer.android.com/studio/intro> (дата обращения: 19.05.2024).
16. Retrofit. – square.github.io: сайт. – URL: <https://square.github.io/retrofit/> (дата обращения: 19.05.2024).
17. Overview. – ngrok documentation: сайт. – URL: <https://ngrok.com/docs> (дата обращения: 19.05.2024).
18. Стасышин, В. М. Базы данных: технологии доступа : учебное пособие для вузов / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2024. — 164 с. — (Высшее образование). — ISBN 978-5-534-08687-4.
19. Маркин, А. В. Программирование на SQL : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 805 с. — (Высшее образование). — ISBN 978-5-534-18371-9.
20. Шуман, Е. А. Тестирование как форма контроля знаний в процессе обучения / Е. А. Шуман. — Текст : непосредственный // Молодой ученый. — 2022. — № 12 (407). — С. 183-186. — URL: <https://moluch.ru/archive/407/89633/> (дата обращения: 02.06.2024).
21. Агальцова, Д. В., Тестирование в образовательном процессе вуза: особенности, задачи и форматы применения. / Д. В. Агальцова, Н. А. Путилина // Гуманитарные науки. Вестник Финансового университета. – 2023. – Т. 13, № 1 – С. 6-9.
22. 1С:Электронное обучение. – 1С:Предприятие 8: сайт. – URL: <https://v8.1c.ru/elo/> (дата обращения: 02.06.2024).

23. RFC 3875 – The Common Gateway Interface (CGI) Version 1.1. – IETF Datatracker: сайт. – URL: <https://datatracker.ietf.org/> (дата обращения: 02.06.2024).