

УДК 004.65

РАЗРАБОТКА И ВНЕДРЕНИЕ ВЕБ-СИСТЕМЫ ИНВЕНТАРИЗАЦИИ БИБЛИОТЕЧНЫХ КНИГ С ИСПОЛЬЗОВАНИЕМ PYTHON И DJANGO

Го Цзя¹, Чжао Хэ¹, Омуралиев Т.С.¹, Халмуратов Б.О.¹

¹КГУ им. И. Арабаева

Библиотеки часто используют разрозненные и неэффективные методы работы. Для оптимизации процессов была разработана веб-система, включающая в себя необходимые новые модули.

Прототип системы был создан и протестирован в среде разработки с использованием следующих инструментов: Python 3.10, Django 2.2.10, PyCharm и Windows 10. Система была протестирована на всех ключевых этапах использования, включая вход в систему, навигацию по меню, заимствование книг, управление библиотеками и личными кабинетами.

Созданная система предназначена для библиотек с небольшим и средним числом посетителей. Она обладает базовыми функциями и требует минимальных вложений в установку и обслуживание.

Текущая версия системы имеет некоторые ограничения, которые можно устранить, улучшив ее масштабируемость, возможности интеллектуального поиска, интеграцию со считывателем штрих-кодов и возможность работы в облаке.

Ключевые слова. библиотечная информационная система, инвентаризация книг, Django, Python, SQLite, веб-приложения, автоматизация библиотек.

PYTHON ЖАНА DJANGO АРКЫЛУУ ВЕБ-КИТЕПКАНА КИТЕПТЕРИНИН ИНВЕНТАРИЗАЦИЯ СИСТЕМАСЫН ИШТЕП ЧЫГУУ ЖАНА ИШКЕ АШЫРУУ

Го Цзя¹, Чжао Хэ¹, Омуралиев Т.С.¹, Халмуратов Б.О.¹

¹И. Арабаев атындагы Кыргыз мамлекеттик университети

Китепканалар көбүнчө ар кандай жана натыйжасыз иштөө ыкмаларын колдонушат. Процесстерди оптималдаштыруу үчүн жаңы модулдарды камтыган веб-система иштелип чыккан. Системанын прототиби төмөнкү куралдарды колдонуу менен өнүктүрүү чөйрөсүндө түзүлгөн жана сыналган: Python 3.10, Django 2.2.10, PyCharm жана Windows 10.

Система колдонуунун бардык негизги баскычтарында, анын ичинде кирүү, Менюларды багыттоо, китептерди карызга алуу, китепканаларды жана жеке эсептерди башкаруу боюнча сыналган.

Түзүлгөн система аз жана орто сандагы коноктор менен китепканалар үчүн иштелип чыккан. Анын негизги функциялары бар жана орнотуу жана техникалык тейлөөгө минималдуу инвестиция талап кылынат.

Системанын азыркы версиясында анын масштабдуулугун, акылдуу издөө мүмкүнчүлүктөрүн, штрих-код окугуч менен интеграциялоону жана булутта иштөө мүмкүнчүлүгүн жакшыртуу менен жоюуга мүмкүн болгон айрым чектөөлөр бар.

Баштапкы сөздөр: китепкана маалымат системасы, китеп инвентаризациясы, Django, Python, SQLite, веб-тиркеме, китепкананы автоматташтыруу.

DESIGN AND IMPLEMENTATION OF A WEB-BASED LIBRARY BOOK INVENTORY SYSTEM USING PYTHON AND DJANGO

Guo Jia¹, Zhao He¹, Omuraliev T.S.¹, Khalmuratov B.O.¹

¹ Arabaev Kyrgyz State University

Libraries often use disparate and inefficient working methods. To optimize the processes, a web-based system has been developed that includes the new modules needed. The prototype of the system was created and tested in a development environment using the following tools: Python 3.10, Django 2.2.10, PyCharm and Windows 10.

The system has been tested at all key stages of use, including logging in, navigating menus, borrowing books, managing libraries and personal accounts. The system created is designed for libraries with a small and medium number of visitors. It has basic functions and requires minimal investment in installation and maintenance.

The current version of the system has some limitations that can be addressed by enhancing its scalability, intelligent search capabilities, integration with a barcode reader, and cloud compatibility.

Keywords: library information system, book inventory, Django, Python, SQLite, web application, library automation.

1. Введение

Цифровизация образовательных и информационных услуг резко повысила спрос на эффективное управление библиотечными ресурсами. По мере того как библиотека превращалась из преимущественно печатного ресурса в информационный центр, все большее значение приобретало ведение точного учета книг и услуг, а также отслеживание пользователей и тиражей. Основанные на бумажных носителях подходы к управлению библиотечными фондами, как правило, подвержены ошибкам, затрудняют заполнение и создание отчетов и отнимают много времени.

Аналогичным образом, электронные подходы, которые распределены по нескольким программам или управляются отдельными библиотеками, могут создавать аналогичные проблемы. В этой статье кратко излагаются результаты магистерской диссертации, в которой были выявлены проблемы, связанные с традиционными подходами к управлению библиотечными запасами, и описывается автоматизированное веб-решение этих проблем, разработанное с использованием языка программирования Python и платформы приложений Django. [1], [4]

Кроме того, автоматизированные библиотечные системы более подробно описаны в разделе "Информация о необходимости автоматизированных систем в библиотеках". Хотя небольшие библиотеки могут хранить свои каталогизационные данные, данные о членах и истории заимствований в формате Excel, им удобнее иметь систему, которая может постоянно отслеживать список доступных книг, сведения о заимствованных экземплярах, сведения о просроченных экземплярах, а также статистику книг в разных изданиях. категории. Этот тип системы может быть создан за короткое время с использованием таких фреймворков, как Django. Такая структура обеспечивает возможность разработки на базе Django наилучшим образом, что позволяет быстро создавать прототипы, повторно использовать код и разделять задачи. Поскольку информационная система в библиотеках работает в соответствии со стандартным бизнес-процессом, разделение задач

является критическим фактором в обслуживании информационной системы.

Этот исследовательский проект направлен на разработку работоспособной, удобной в использовании и ремонтпригодной системы управления книжными запасами для библиотеки. Исследование преследует четыре цели. Эти цели заключаются в определении потребностей пользователей и систем, настройке архитектуры систем, разработке модели данных и схемы базы данных, разработке и внедрении программы для управления бухгалтерскими книгами и участниками, а также для отслеживания операций заимствования. Исследование имеет значительную научную ценность, поскольку демонстрирует, что даже при наличии небольшого набора технологий можно создать работающую модульную систему автоматизации библиотек, которую легко понять и внедрить.

2. Обзор литературы

Основой разработанной системы являются три основных компонента: язык Python, фреймворк Django и использование реляционной базы данных. Python - один из наиболее часто используемых языков программирования для веб-разработки, благодаря легко читаемому коду, огромному количеству готовых модулей и наборов инструментов, а также эффективным возможностям быстрого создания веб-приложений. Python не ограничивается веб-разработкой. Он используется для реализации различных скриптов для автоматизации различных процессов, для статистических и аналитических задач, для задач машинного обучения и искусственного интеллекта, для создания тестовых наборов и многих других целей. Упомянутые особенности делают язык чрезвычайно популярным среди ученых и студентов.

Django — это бесплатное программное обеспечение с открытым исходным кодом, выпущенное под разрешительной лицензией New BSD. Написанный на Python, он способствует быстрой разработке и часто упоминается как веб-фреймворк высокого уровня на Python, который способствует быстрой разработке и соответствует шаблону Model-View-Template (M-V-C), при этом приложение разделено на четыре раздела: Модели, представления, шаблоны и URL-адреса. Кроме того, Django поддерживает систему объектно-реляционного отображения (ORM), позволяющую хранить данные в реляционной базе данных, включает мощную систему маршрутизации URL-адресов, разработан со встроенной защитой и имеет множество инструментов, которые упрощают и ускоряют разработку форм и интерфейса администрирования. [1] , [5]

В своей работе мы использовали SQLite, облегченную реляционную базу данных, на уровне хранения данных. SQLite - это автономный, бессерверный файл базы данных SQL с нулевой конфигурацией, который обеспечивает быструю и простую в использовании реляционную базу данных для веб-сайтов и веб-приложений, которым требуется доступ к небольшому набору данных. Хотя утверждается, что SQLite не подходит для высоконагруженных многопользовательских корпоративных систем, его более чем достаточно для академической проверки концепции, мелкомасштабных внедрений и реализации очень маленьких библиотек из-за его простоты развертывания. Наконец, общие принципы проектирования баз данных, используемые в этой работе, схожи с принципами типичного моделирования реляционных данных и разработки приложений на основе SQL. [2], [3], [11], [12].

В то время как исследования, посвященные автоматизации библиотек, в основном сосредоточены на информационных системах для хранения каталогов, а также для решения оперативных задач, таких как распространение, регистрация заемщиков, регистрация событий и создание отчетов для руководства, исследование, проведенное в рамках платформы Django, касалось системы управления запасами библиотеки. Это исследование показало положительное влияние модульной системы, ролевой ориентации и общих книжных данных на повседневную работу библиотеки.

В статье представлено практическое решение для управления публичной библиотекой, основанное на структурно понятной реализации веб-интерфейса с информационной панелью, событиями за последние несколько дней и сводной статистикой.

3. Материалы и методы

Исследование проводилось в рамках практического проекта по разработке программного обеспечения. Процесс включал в себя сбор требований, разбивку функций, разработку баз данных и интерфейсов, кодирование и проведение тестирования в режиме "черного ящика". Анализ требований был сосредоточен на потребностях пользователей с административными полномочиями, которые отвечают за ведение каталога, регистрацию участников, выдачу книг, обработку возвратов и мониторинг последних действий. Поскольку не все целевые пользователи могут обладать глубокими техническими знаниями, при разработке интерфейса приоритетное внимание уделялось простоте, удобству использования и минимальной когнитивной нагрузке.

С системной точки зрения приложение было разработано как веб-информационная система с централизованной базой данных и отдельными функциональными компонентами. Интерфейс основан на шаблонах Django, в то время как внутренние операции выполняются с помощью представлений и моделей на основе Python. Хранение данных организовано с использованием реляционных таблиц. Этот подход подходит для доменов, где транзакции и четко определенные сущности доминируют в рабочем процессе.

Среда разработки, указанная в дипломной работе, включала PyCharm в качестве инструмента программирования, Python 3.10 и Django 2.2.10 в качестве программной платформы и Windows 10 в качестве основной операционной системы. Проект был инициализирован и выполнен с использованием стандартного рабочего процесса командной строки Django, включая команды миграции и локальный сервер разработки. Такая конфигурация позволила тиражировать и тестировать прототип без необходимости в дополнительной серверной инфраструктуре.

Таблица 1. Основные функциональные модули предлагаемой системы

Модули	Основные функции	Цели
Модуль входа	Регистрация, аутентификация, контроль доступа	Обеспечивает безопасный вход в среду управления
Модуль меню	Панель показателей, диаграммы, последние события	Предоставляет быстрый обзор состояния системы и визуализацию данных
Модуль записей о выдаче	Выдача книг, возврат книг, просмотр записей, удаление записей	Поддерживает учет книговыдачи и историю транзакций
Модуль управления библиотекой	Книги, категории, издатели, читатели	Поддерживает основной каталог и справочные сущности
Модуль личного кабинета	Просмотр и обновление профиля	Позволяет администратору управлять сведениями учетной записи

3. Архитектура системы и модель данных

Архитектуру системы можно описать как трехуровневое решение. Интерфейсный уровень включает в себя веб-интерфейс, который создается с использованием шаблонов Django и объектов навигации. Средний уровень содержит весь код на Python, который обрабатывает запросы, применяет бизнес-логику, проверяет наличие ошибок и получает доступ к данным в серверной части через представления и модели Django. Внутренний уровень включает в себя все структурированные данные, такие как книги, категории, издатели, участники, действия по заимствованию, административные пользователи и статистику использования. Благодаря разделению архитектуры на эти три уровня система стала более управляемой, понятной, тестируемой и расширяемой.

Мы разработали функцию, позволяющую создать домашнюю страницу в стиле информационной панели, на которой отображается информация о количестве книг, подписчиках, категориях и издателях, а также обо всех возвращенных книгах и всех административных событиях. Это преобразование превратило систему, изначально предназначенную для выдачи книг напрокат, в более простую систему поддержки принятия решений. Мы также внедрили диаграммы для визуализации определенных данных, таких как наиболее популярные товары на складе, книги, которые чаще всего берут напрокат (уже взяты или нет), и недавно зарегистрированные пользователи. Аналитические возможности по-прежнему ограничены, но они должны быть полезны для повседневного управления публичной библиотекой.

В процессе разработки этого тезиса важным шагом стало создание модели взаимосвязи сущностей. Модель включает в себя семь сущностей: книги, записи о заимствованиях, категории, издатели, профили участников, профили администраторов и, что не менее важно, операции администратора. Каждая из этих основных сущностей обладает набором атрибутов, достаточных для эффективного управления ими. Например, сущность "Книга" включает в себя такие атрибуты, как автор, название, описание, количество экземпляров в наличии, статус, идентификатор категории, идентификатор издателя и статистику выдачи. Организация "Учет заимствований" связывает книги с участниками и регистрирует выпуски и сроки их получения. Мероприятия для издателей и категорий облегчают организацию каталога для этих организаций. Контактная информация и идентификационные данные участников хранятся для обеспечения надлежащего контроля за распространением книг.

Таблица 2. Основная суть в проектировании базы данных

Суть	Характерные атрибуты	Роль в системе
Книга	Автор, название, описание, количество, статус, категория, идентификационный номер читателя	Stores core inventory data
Запись о выдаче	читатель, электронная почта, телефон	Отслеживание выдачи
Категория	Название	Классификация книг по группам
Издатель	Название, город, контакты	Справочная информация об издательстве
Членство	Имя, возраст, город, email, телефон, номер карты, пол	Представляет зарегистрированных пользователей библиотеки
Профайл	биография, фото профиля, номер телефона, email, admin	Хранит данные учетной записи администратора
Пользовательская активность	целевая модель, создание, детали, тип операции	Регистрирует административные действия

5. Реализация функциональных модулей

Система начинается с процесса входа в систему, в ходе которого система проверяет наличие содержимого в полях имени пользователя и пароля, гарантируя, что предоставленные учетные данные соответствуют зарегистрированному пользователю. После успешного входа в систему пользователи перенаправляются на основную платформу. Интерфейс системы прост: слева расположена панель навигации для доступа к различным категориям, а справа - область содержимого. Такая компоновка обеспечивает легкий доступ к различным разделам системы, что делает ее удобной для пользователя.

Модуль управления меню предоставляет полный обзор текущего состояния библиотеки, доступный с домашней страницы модуля. Страница диаграммы предлагает визуальное представление этой информации, представляя ее в более доступном формате. Страница сводной статистики содержит подробный анализ текущей статистики с более обширным списком параметров по сравнению с домашней страницей.

Используя информацию, доступную на главной странице и странице диаграмм, администраторы могут без особых усилий определять наиболее популярные книги, отслеживать распределение книг на полках и отслеживать текущую активность пользователей.

Модуль для управления записями о заимствованиях поддерживает процесс выдачи и возврата бухгалтерских книг. При инициировании транзакции создается новая запись о заимствовании, содержащая все необходимые данные. Любую запись, хранящуюся в базе данных, можно просмотреть полностью или удалить по желанию. Модуль учета заимствований служит центральным узлом для системы циркуляции в рамках более широкого пакета программного обеспечения.

Система библиотечного администрирования состоит из нескольких подсистем, каждая из которых отвечает за определенный аспект библиотечной деятельности. К этим подсистемам относятся управление книгами, управление категориями, управление издателями, управление пользователями и персональное администрирование.

Управление книгами позволяет пользователям добавлять, изменять и удалять книги, а также просматривать историю их заимствований. Управление категориями позволяет распределять книги по различным категориям, а управление издателями предоставляет подробную информацию о каждом издателе, например, его имя и местоположение. Управление участниками облегчает создание и обновление списков участников. Персональное администрирование используется администраторами для просмотра и обновления личной информации, такой как имена пользователей, адреса электронной почты, номера телефонов и фотографии профиля.

6. Функциональное тестирование и результаты

Для оценки корректности реализованных функций был использован метод тестирования по методу "черного ящика". Цель такого тестирования - определить, приведут ли видимые операции к ожидаемым результатам для типичных сценариев использования без знания пути выполнения кода.

Для тестирования и реализации функций использовалась одна и та же среда.

Я протестировал модуль входа в систему, ничего не вводя, вводя неверные данные для входа и вводя правильные данные для входа. Он функционировал должным образом. Меню всегда отображало правильную информацию на главной странице и визуальных диаграммах. На странице "заимствование" корректно загружаются все записи, отображаются сведения об отдельных записях, удаляются выбранные записи и добавляются новые записи. На странице "Управление библиотекой" корректно просматриваются, добавляются, обновляются и удаляются книги, категории, издатели и участники. На странице "персональное управление" корректно отображались и обновлялись профили.

Таблица 3. Сводка результатов функционального тестирования

Модуль	Основной фокус тестирования	Результат
Вход	Пустые поля, неверные учетные данные, корректные учетные данные	Пройдено
Меню	Показатели на главной странице, визуализация диаграмм	Пройдено
Записи о выдаче	Просмотр, удаление, добавление новой записи о выдаче	Пройдено
Управление библиотекой	Операции с книгами, категориями, издателями и читателями	Пройдено
Личный кабинет	Отображение и обновление данных профиля	Пройдено

Общие результаты системных тестов, проведенных на основных модулях прототипа дипломной работы, являются положительными. Система работала так, как предполагалось, без каких-либо функциональных сбоев на протяжении различных сценариев

тестирования, корректно обрабатывала переходы между интерфейсами и передавала соответствующие данные. Хотя эти тесты не гарантируют полной оценки производительности, нагрузки или безопасности системы, было отмечено, что поведение системы соответствует целям проекта. Результатом использования этой системы стало то, что фрагментация множества выполняемых вручную и дублирующих друг друга процедур, которые в настоящее время выполняются, была преобразована в единый рабочий процесс.

7. Обсуждение

Базовая платформа для инвентаризации библиотек может быть создана с помощью удивительно малого количества строк кода. Платформа обладает несколькими ключевыми функциями, которые упрощают ее обслуживание и расширение. Он разделен на несколько разделов: один для входа в систему, другой для отображения панели мониторинга, один для управления тиражом, один для управления каталогом и один для управления / изменения отдельных учетных записей. Одной из его самых сильных сторон является простота настройки рабочей версии этого кода. Поскольку он был построен поверх Python, Django и SQLite, его могут настроить люди с минимальными техническими ресурсами или использовать в качестве отправной точки для разработки более надежной системы.

На данный момент система является прототипом, который мы разработали в рамках академического проекта. Поскольку в коде для хранения данных используется SQLite, он не так эффективен в сценариях с высокой степенью параллелизма или когда речь идет о больших наборах данных или распределенных установках. В дипломной работе эта система тестируется на работоспособность. Количественный тест производительности, стресс-тест, тест на безопасность (тест на проникновение), тест на удобство использования в работе не проводятся. Поэтому мы не можем с уверенностью утверждать о возможности масштабирования и стабильности этой системы в будущем.

Есть несколько потенциальных улучшений для системы в целом. Во-первых, хранение данных в базе данных PostgreSQL или другой серверной СУБД повысило бы параллелизм и масштабируемость для многопользовательской работы. Во-вторых, использование дополнительных возможностей фильтрации, полнотекстового поиска и рекомендательной системы могло бы улучшить интерфейс поиска. В-третьих, интеграция системы со штрих-кодом или RFID-системой, используемой для распространения книг, повысила бы эффективность

процесса проверки и возврата книг. Наконец, возможность облачного развертывания и удобный для мобильных устройств веб-интерфейс сделали бы систему более подходящей для географически распределенной или децентрализованной публичной библиотеки, где администрирование может осуществляться удаленно.

8. Практическое значение для библиотечной деятельности

Система особенно актуальна для библиотек со среднесрочной стратегией. Эти учреждения стремятся перейти от ведения бухгалтерии вручную к структурированной системе цифровой обработки книг и других носителей, но не располагают ресурсами для внедрения крупной интегрированной корпоративной системы для работы с несколькими типами носителей. Наиболее важным требованием к таким учреждениям является не обязательно наличие передовых технологий, а скорее наличие системы, которая была бы проста в освоении, использовании и обслуживании. Разработанный прототип для регистрации средств массовой информации и займов предлагает простую навигацию, четкое разделение объектов и унифицированную модель данных. Веб-панель мониторинга позволяет администраторам легко контролировать события, связанные с запасами и займами, заменяя страницы бумажного журнала или электронной таблицы. Такие администраторы часто отвечают за несколько частично совпадающих задач в образовательных учреждениях, таких как составление бюджета, управление персоналом, сбор средств и обработка займов. Панель мониторинга предлагает им множество повторяющихся действий, которые можно выполнить несколькими щелчками мыши.

Одно из практических применений заключается в том, как такая система может быть использована в образовательных целях с точки зрения ее реализации и организационной структуры в самой библиотеке. Библиотечная платформа — это нечто большее, чем просто инвентарный учет и регистрация операций с книгами. Она позволяет генерировать обширный массив оперативных исторических данных. Просматривая историю заимствований, разбивку по категориям запасов и активность участников, можно определить наиболее активные области коллекции, а также области, требующие дополнительной закупки запасов. Просматривая последние события, генерируемые системой, можно выявить узкие места в работе и проблемы в процессах обращения и возврата. Для университетов и колледжей с обширными функциями преподавания и поддержки обучения такая небольшая информационная система может предоставить множество данных для принятия

управленческих решений. Кроме того, система служит примером для студентов, изучающих разработку программного обеспечения, проектирование баз данных и разработку информационных систем. Технологии и методология, используемые для создания приложения, широко используются во всем мире, а код написан на языке и фреймворке, широко используемых профессиональными веб-разработчиками в различных контекстах.

Представленный здесь код, облегчающий печать сведений о заемщике, служит наглядным примером подхода, использованного при разработке базового программного обеспечения для оцифровки данных учреждений. Вместо того чтобы начинать с нуля с создания комплексного решения, поэтапный подход позволяет создать надежное ядро прототипированных функций (каталогизация, регистрация заемщиков, контроль привилегий при заимствовании), прежде чем переходить к аналитике, интерфейсам устройств идентификации, средствам удаленного доступа или совместимости с системами других образовательных учреждений. Следовательно, такой инструмент, как представленный здесь, следует рассматривать не просто как отдельное программное обеспечение, а скорее как строительный блок и инструмент перехода, который позволяет учреждениям методично продвигаться к следующему поколению цифровых услуг для библиотек.

9. Заключение

В этой статье описывается разработка и внедрение веб-системы для простой инвентаризации библиотечных книг. Работа, описанная в этой статье, была основана на попытке облегчить эффективные, точные и прозрачные процессы инвентаризации с использованием Python и Django. В статье описываются результаты анализа требований, результирующая архитектура модульной системы и ее подробная реализация, включая аутентификацию пользователей, мониторинг статуса книг, отслеживание записей о заимствованиях, управление каталогом книг, регистрацию участников и административные детали.

Результаты функциональных тестов показали, что разработанная система поддерживает базовые пользовательские сценарии, настроенные на этапе проектирования. Хотя созданная система может быть использована в небольших и средних библиотеках, в ее нынешнем виде это скорее рабочий прототип системы автоматизации библиотеки и рабочая модель для внедрения образовательных информационных систем с использованием фреймворка Django.

Дальнейшая работа может быть направлена на увеличение емкости базы данных, повышение аналитического потенциала, повышение безопасности и интеграцию с современными технологиями библиотечной идентификации. Хотя системе все еще может не хватать некоторых из этих улучшений, она представляет собой важный шаг вперед в компьютеризации библиотечных рабочих процессов за счет автоматизации всех этапов в единой системе.

ЛИТЕРАТУРА

1. Ван Ян, Цзян Синьтун. Исследование проектирования MVC-фреймворка на основе Python и Django. — 2021.
2. Ли Цуньбинь. Технология применения баз данных. — Пекин: China Water Power Press, 2001.
3. Лю Ичэн. Практические упражнения по SQL. — Пекин: Изд-во Сианьского университета электронной науки и технологий, 2004.
4. Муссиано Ч., Кеннеди Б. HTML и XHTML: подробное руководство. — 2011.
5. Мэн Лэй. Проектирование и реализация Python в задачах автоматизированной эксплуатации и технического обслуживания // China Cable Television. — 2021.
6. Пуривал С. Основы разработки веб-приложений. — Санкт-Петербург: Питер, 2015.
7. Ржеутская С. Ю. База данных. Язык SQL. — Вологда: ВоГТУ, 2010.
8. Роббинс Дж. Н. Руководство по веб-дизайну. — 2014.
9. Син Гуанбинь, Ван Циньчжу. Развертывание проекта Django на основе Python 2.7 в системе CentOS 7. — 2021.
10. Сун Хунмэй, Хань Ляньфу, Люй Сюли, Чжоу Вэй, Фу Чанфэн, Хань Цзянь. Обсуждение преподавания Python, ориентированного на практическое применение // Computer Knowledge and Technology. — 2021.
11. У Чуньмэй, Цзян Линьли. Проектирование и реализация системы управления книжным фондом библиотеки колледжа на основе Python и Django. — 2020.
12. Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных. — Санкт-Петербург: КОРОНА-Век, 2009.
13. Чжан Цзюнь. Проектирование программного обеспечения пакетной обработки послепродажных деталей на основе Python // Automotive Practical Technology. — 2021.