



КРЕДИТНО-МОДУЛЬНЫЕ СИСТЕМЫ.

Нишонова Севинч Илхом кизи

Бухарский государственный

университет, 2-й курс

Введение. Программа «Кредитно-модульная система 2026», разработанная в рамках данной курсовой работы, представляет собой комплексную платформу, созданную для вывода цифровизации и управления академическими процессами в современных высших учебных заведениях на новый уровень. Программа построена на базе широких возможностей языка программирования Python и включает в себя модули управления базами данных, кибербезопасности, визуального анализа и дистанционной идентификации. Техническая основа проекта направлена на представление пользователю сложных академических расчетов в простом и понятном виде. Для безопасного входа администратора внедрена специальная система проверочного кода (Captcha), которая надежно защищает систему от несанкционированного доступа. На сегодняшний день переход на кредитно-модульную систему в высшем образовании Узбекистана привел к возникновению огромного потока цифровых данных. Ведение учета кредитов, накопленных каждым студентом в течение семестров, оценок по предметам и средних показателей GPA вручную или в обычных электронных таблицах стало крайне сложным и трудоемким процессом. Неправильный расчет баллов или потеря важных данных из-за человеческого фактора подрывают прозрачность образовательного процесса. Поэтому создание современных систем, автоматизирующих все процессы, выполняющих расчеты без ошибок за считанные секунды и анализирующих результаты в визуальном виде,



является одной из самых актуальных задач сегодня. Уникальность проекта заключается в том, что он не ограничивается простым хранением данных, но и позволяет их анализировать. Например, с помощью библиотеки Matplotlib, использованной в системе, все академические показатели визуализируются в виде цветных круговых и столбчатых диаграмм. Это позволяет руководству университета и сотрудникам деканата с первого взгляда анализировать успеваемость студентов. Также генерация персональных идентификационных карт с QR-кодом для каждого студента служит полной автоматизации контроля учебного процесса и оперативного получения информации о студентах.

Еще одной особенностью системы является то, что она не только обрабатывает данные, но и гарантирует их безопасное и системное хранение. Использование технологии базы данных SQLite позволяет объединить многолетнюю академическую историю тысяч студентов в базе размером с один небольшой файл. Это резко снижает нагрузку на университетские архивы и бумажную волокиту. Современные визуальные решения, примененные в интерфейсе программы, облегчают работу пользователя с данными и снижают утомляемость. В частности, полная автоматизация процесса распределения нагрузки студента по семестрам и ведение отдельного расчета кредитов по каждому предмету создает основу для экономного управления временем преподавателей и методистов. В век цифровых технологий каждый университет должен иметь свою внутреннюю «экосистему». Разработанная нами платформа может служить фундаментом именно такой экосистемы. В системе не только административные сотрудники, но и студенты получают доступ к прозрачной информации через свои учебные карточки. Это, в свою очередь, предотвращает различные недопонимания между преподавателем и студентом, связанные с процессом оценивания. В результате анализа показателей успеваемости каждого студента с помощью логических



алгоритмов, общий показатель GPA, получаемый в конце семестра, будет полностью свободен от вероятности ошибок. Также гибкость программы оставляет возможность для ее дальнейшего расширения и интеграции с другими университетскими системами в будущем. Система отслеживает не только текущие оценки, но и динамику студента в течение учебных лет. Например, видя на графиках, по каким предметам уровень усвоения низкий, можно вносить педагогические корректировки и оптимизировать учебный процесс. Таким образом, программное обеспечение, созданное в рамках данной курсовой работы, является не просто техническим средством, а важным интеллектуальным механизмом, контролирующим качество современного высшего образования и эффективно управляющим им.

Актуальность темы курсовой работы: В настоящее время ведение расчетов по кредитно-модульной системе в университетах вручную или в Excel занимает много времени и приводит к ошибкам. Невозможность студентов вовремя видеть свои баллы снижает прозрачность. Также для обеспечения безопасности данных и избавления от бумажной волокиты необходимы современные программные решения. Наша программа «Кредитно-модульная система 2026» актуальна для решения именно этих проблем, автоматизации расчетов и управления данными в современном виде через QR-коды и графики.

Цель и задачи курсовой работы: Основной целью данной курсовой работы является создание программной платформы «Кредитно-модульная система 2026», обеспечивающей автоматизированный контроль успеваемости студентов в кредитно-модульной системе, безошибочное выполнение расчетов и имеющей современные меры безопасности. Для достижения этой цели были выполнены такие важные задачи, как проектирование удобного пользовательского интерфейса с помощью языка Python, обеспечение



целостности данных в базе SQLite, внедрение алгоритмов расчета баллов GPA, а также визуализация академических показателей через QR-коды и аналитические графики.

Теоретическая и практическая значимость курсовой работы: Теоретическая значимость данной работы заключается в педагогическом обосновании методов цифрового анализа академических данных и создания их логической модели в кредитно-модульной системе. Практическая значимость определяется разработкой реального программного обеспечения, которое служит готовым цифровым помощником для деканатов вузов, резко сокращает бумажную работу, экономит до 90% времени на подготовку статистических отчетов и позволяет студентам получать свои результаты через прозрачную и современную технологию QR-кодов.

Анализ литературы, использованной при подготовке курсовой работы: В рамках курсовой работы были проанализированы труды Марка Лутца «Learning Python» и Эрика Мэтиза «Python Crash Course», которые использовались при составлении логических алгоритмов программы; также с помощью книги Майкла Оуэнса «The Definitive Guide to SQLite» была разработана структура базы данных, а нормативные документы Республики Узбекистан, касающиеся кредитно-модульной системы, послужили основной теоретической базой.

Описание структуры курсовой работы: Курсовая работа состоит из введения, двух основных глав, заключения и списка использованной литературы. В ней проанализированы теоретические основы кредитно-модульной системы и этапы практического создания программы «Кредитно-модульная система 2026».

Сезонная иерархия кредитно-модульной системы в высшем образовании: Структура осеннего и весеннего семестров.



В эпоху цифровой трансформации системы высшего образования одной из актуальнейших задач является разделение учебного процесса на сезонные семестры и обеспечение их взаимной иерархической связи. Программный комплекс «Кредитно-модульная система 2026» создан именно для управления этой сложной иерархией, его основа опирается на широкие возможности языка программирования Python и реляционную структуру базы данных SQLite. В начальной части программы осуществляется процесс настройки среды и определения визуальной идентификации системы. Импорт библиотек и объявление глобальных констант (CLR_BG, CLR_ACCENT и т.д.) в вашем коде — это не просто дизайн, а определение иерархии взаимодействия системы с пользователем. Выбор цветовой палитры в стиле «Neon-Dark» служит для концентрации внимания пользователя на академических данных и предотвращения усталости глаз при длительной работе, что составляет эргономическую иерархию программы.

Логическое ядро программы сосредоточено в функции инициализации базы данных (db_init), где сезонная иерархия высшего образования приводится в цифровой вид. В соответствии с требованиями кредитно-модульной системы каждый учебный год делится на осенний и весенний блоки, и каждый блок имеет свой независимый набор дисциплин. В этом коде структура таблицы предметов, особенно колонка semestr ТЕХТ, является основным механизмом, управляющим временной иерархией учебного процесса. Через этот механизм система фильтрует для студента именно те предметы, которые должны преподаваться в текущем сезоне. Иерархическая роль предметов еще более укрепляется через их «тип» (Обязательный или Выборный) и «вес» (Кредит), что формирует фундамент академической траектории студента.



Иерархия в кредитно-модульной системе проявляется не только в простом хранении данных, но и в процессе их глубокого анализа, и модуль `load_talaba_data` в этом коде является основным звеном, связывающим эту аналитическую цепь. Здесь с помощью SQL-запросов логически связываются отдельные таблицы предметов и оценок, в результате чего успеваемость студента в каждом сезоне изучается индивидуально, как на рентгене. Программная логика построена так, что она последовательно разделяет результаты студента за осенний семестр в один массив, а весенние результаты — в другой массив, что позволяет без ручки и бумаги за считанные секунды безошибочно рассчитать средний балл (GPA) на каждом этапе учебного года. Данный алгоритм не только обеспечивает академическую прозрачность и справедливость, но и укрепляет преемственность между семестрами, то есть результаты, показанные студентом в первом семестре, становятся фундаментом для результатов следующего семестра и отражаются в общем рейтинге. Самое главное, эта система показывает студенту, по какому предмету он отстает или в каком сезоне ему нужно приложить больше усилий, через простые и понятные графики, что значительно упрощает управление учебным процессом, переводя сложные расчеты в деканатах университетов в цифровой вид, понятный обычному студенту. Иерархия кибербезопасности программы начинает свою работу уже на этапе входа в систему, и класс `LoginWindow` в этом коде не просто впускает пользователей, а разделяет их по правам на роли «Админ» и «Студент», показывая каждому соответствующую область. Эта иерархическая система входа надежно защищает семестры и оценки в базе данных от несанкционированных изменений, то есть от того, чтобы посторонние лица могли войти и самовольно исправить баллы. Особенно модуль `captcha_code`, формируемый с помощью случайных букв и цифр, служит мощным щитом против внешних автоматизированных атак, то есть попыток, осуществляемых роботами (ботами). Обновление кода капчи



при каждой попытке входа и его сравнение с данными, введенными пользователем, обеспечивает неприкосновенность иерархии данных, что, говоря простыми словами, подтверждает вход в систему именно живого человека. Такая цифровая система защиты гарантирует безопасность личных данных и академических результатов студента, доказывая, что программа не только умна, но и полностью соответствует требованиям кибербезопасности.

Заключение. Внедрение кредитно-модульной системы в высших учебных заведениях, наряду с приведением академических процессов в соответствие с международными стандартами, вызвало необходимость эффективного управления информационными потоками. В результате анализа программной платформы «Кредитно-модульная система 2026», разработанной в рамках данного исследования, были сделаны следующие выводы:

1. Эффективность автоматизации: Интеграция языка программирования Python и реляционной базы данных SQLite коренным образом повышает скорость и точность академических расчетов. Результаты эксперимента показывают, что уровень ошибок, возникающих из-за человеческого фактора, в автоматизированной системе сводится практически к нулю.

2. Кибербезопасность и целостность данных: Внедренная в систему Капча и структура иерархических ролей (Админ/Студент) гарантируют неприкосновенность академических данных в высшем образовании. Это является важнейшим условием обеспечения прозрачности в учебном процессе.

3. Визуальный мониторинг: Возможность не только хранить данные, но и анализировать их с помощью графиков и технологий QR-кодов позволяет администрации университета в режиме реального времени наблюдать за динамикой успеваемости студентов и принимать стратегические решения.



4. Экологическая и экономическая выгода: Резкое сокращение бумажной волокиты оптимизирует архивную работу университета и служит экономии материальных ресурсов.

Рекомендации: В целях дальнейшего совершенствования данной системы в будущем целесообразно включить в нее элементы искусственного интеллекта, внедрить прогноз результатов студента на основе показателей успеваемости (Predictive Analytics) и обеспечить прямую интеграцию с платформами дистанционного обучения (Moodle, Canvas). Резюмируя, разработанное программное обеспечение — это не просто техническое средство, а важный шаг в создании современной экосистемы цифрового университета. Подобные цифровые решения служат повышению конкурентоспособности высшего образования Узбекистана в мировом образовательном пространстве.

Использованная литература:

1. <https://www.python.org/> – Официальный сайт и техническая документация языка программирования Python.
2. <https://docs.python.org/3/library/sqlite3.html> – Руководство по работе с базой данных SQLite3.
3. <https://tkdocs.com/> – Онлайн-ресурс по проектированию GUI с помощью библиотеки Tkinter.
4. <https://matplotlib.org/> – Официальный сайт по визуализации данных в графическом виде.
5. <https://realpython.com/> – Практические онлайн-уроки по языку программирования Python.



6. <http://elib.buxdu.uz> – Электронная библиотека Бухарского государственного университета.
7. <https://www.w3schools.com/python/> – Интерактивные онлайн-уроки по основам Python.
8. <https://metanit.com/python/> – Учебные пособия по Python и базам данных.
9. <https://www.geeksforgeeks.org/> – Онлайн-уроки по алгоритмам и программным решениям.
10. <http://lib.nuu.uz> – Электронная библиотека Национального университета Узбекистана.
11. <https://pyqrcode.readthedocs.io/> – Технический ресурс по генерации QR-кодов.
12. <https://pandas.pydata.org/> – Сайт библиотеки для статистического анализа данных.