УДК 621.350.11

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ДИСТАНЦИОННОГО ОБУЧЕНИЯ

Шавель Я. С.
Научный руководитель – ст. преподаватель Русак Л. В.
БНТУ

Аннотация: Основная цель процесса автоматизации обучения – повышение качества образовательных услуг, предоставляя быстрый доступ к актуальным ресурсам максимальному количеству обучающихся в реальном времени.

Ключевые слова: автоматизация, проектирование, разработка, база данных, курс, модуль, преподаватель, студент.

В современном обществе, увеличивается автоматизация любых производственных процессов, а также эффективность работы с большими объемами информации. Система образования, как и другие жизни сферы общества, обязательно проходит процесс автоматизации своей деятельности.

Основная цель процесса автоматизации обучения – повышение качества образовательных услуг, предоставляя быстрый доступ к актуальным ресурсаммаксимальному количеству обучающихся в реальном времени.

К разрабатываемой системе был выставлен ряд требований:

– возможность хранения и обработки информации о дисциплинах (курсах), студентах, преподавателях, учебныхматериалах;

– составление структуры дисциплины (курса);

– возможность внесения изменений в структуру дисциплины (курса);

– разграничение прав пользователей по группам (преподаватель, студент);

– расширяемость системы (возможность ее доработки в случаеповышения требований к автоматизированной системе).

Задача организации решения проекта – разработать автоматизированную систему дистанционного обучения, которая позволит улучшить предоставление информации студентам и преподавателям.

Проектирование системы

При проектировании системы была разработана диаграмма деятельности.

Входными потоками модели являются: учебные материалы и заявка на обучение.

Управляющими потоками являются: ГОСТы, правила и процедуры обучения, а также учебный план.

Выходным потоком является подтверждение о прохождении курса. Исполняющими потоками являются студенты и преподаватели.

Детализация диаграммы включает в себя 4 процесса:

– набор студентов;

– создание курса;

– создание модулей;

– прохождение курса.

Также был разработан ряд алгоритмов для разработки системы, основным из которых является алгоритм управления курсом.

После авторизации преподавателя ему открывается меню управления курсом, в котором предоставляется выбор: создание нового курса, редактирование или удаление уже созданного ранее курса. В данном алгоритме после создания и редактирования выполняется проверка на сохранение изменений, а при удалении – подтверждение действия.

Построение базы данных

База данных построена на основе инфологического моделирования и в последующем даталогического проектирования. По итогам проектирования создана даталогическая схема данных, представляющая собой логические связи между элементами независимо от их содержания и среды хранения, а также база данных, содержащая 12 связанных нормализированных таблиц.

Реализация автоматизированной системы дистанционного обучения

Структура проекта Django включает в себя несколько компонентов, включая приложения, которые являются основным способом организации кода в Django.

Приложения Django – это независимые модули, которые могут быть повторно использованы в разных проектах. Они содержат код для обработки запросов, моделей базы данных, шаблонов и статических файлов. Каждое приложение может иметь свои собственные URL-адреса, представления и шаблоны [1].

Структура проекта состоит из 3 приложений, каждое из которых выполняет свой независимый функционал:

– приложение «Курсы» (courses) содержит функции для работы с курсами, модулями и содержимым модулей;

– приложение «Студенты» (students) содержит функции для работы со студентами (зачисление на курс, списки доступных курсов и т. д.);

– приложение «Пользователи» (users) содержит функции для работы с пользователями (регистрация, авторизация и т. д.).

В корневой директории study_platform находится файл manage.py, который используется для управления проектом, а также директория study_platform, которая содержит основные файлы проекта:

– файл, который содержит настройки проекта, такие как база данных, статические файлы, шаблоны и другие параметры;

– файл, который содержит маршруты URL для проекта.

Архитектурный паттерн Model-View-Template (MVT) является вариантом паттерна Model-View-Controller (MVC) и используется во многих веб-фреймворках, включая Django.

MVT разделяет приложение на три основных компонента: model (отвечает за хранение данных и их обработку), view (отвечает за отображение данных и взаимодействие с пользователем) и template (отвечает за отображение данных на странице).

Когда к приложению приходит запрос, то URL dispatcher определяет, с каким ресурсом сопоставляется данный запрос и передает этот запрос выбранному ресурсу. Ресурс фактически представляет функцию или View, который получает запрос и определенным образом обрабатывает его. В процессе обработки View может обращаться к моделям и базе данных, получать из нее данные, или, наоборот, сохранять в нее данные. Результат обработки запроса отправляется обратно, и этот результат пользователь видит в своем браузере. Как правило, результат обработки запроса представляет сгенерированный html-код, для генерации которого применяются шаблоны (Template).

Реализация моделей

Модели в MVT используются для определения структуры данных и правил их валидации. Они представляют собой классы Python, которые наследуются от базового класса models. Model в Django.

Модели определяют поля, которые будут храниться в базе данных, и их типы данных. Каждое поле модели соответствует столбцу в таблице базы данных. Django автоматически создает таблицы в базе данных на основе определений моделей.

При создании таблицы Django также автоматически создает первичный ключ (primary key) для каждой записи в таблице. По умолчанию это поле id, которое является целочисленным полем, автоматически увеличивающимся при добавлении новых записей.

В ходе разработки системы было реализовано 9 моделей для работы с курсом и его материалами.

Рассмотрим пример реализации некоторых моделей:

Модель Course (Курс). Она создает таблицу для хранения в себе основной информацию о курсе: предмет курса, наименование, дата создания, дата редактирования, список студентов курса.

Модель ItemBase. Это абстрактная модель. Ее особенность в том, что при установке параметра abstract = True, отдельная таблица в базе данных не создается. Эта модель содержит набор идентичных полей для создания моделей видов контента.

Пример реализации одного из видов контента – модель Text (Текст). Она наследует в себя модель ItemBase. Что добавляет в таблицу не только поля данной модели, но еще и абстрактной.

Разработка представлений

В MVT представления (View) являются компонентом, который обрабатывает запросы и возвращает ответы. Они получают данные из модели

(Model), обрабатывают их и передают в шаблон (Template), который отображает данные в виде HTML-страницы [2].

Представления в MVT могут быть написаны как функции или классы, имогут использовать различные методы HTTP, такие как GET, POST, PUT и DELETE. Они могут также принимать параметры из URL, обрабатывать данные форм, выполнять операции с базой данных и многое другое.

Рассмотрим несколько примеров реализации представлений.

Представления OwnerMixin и OwnerEditMixin созданы для переопределения методов фильтрации объектов по атрибуту owner, чтобы получить объекты, принадлежащие текущему пользователю (request.user), и форм проекта, чтобы автоматически установить текущего пользователя владельцем объекта

Данные представления будут наследоваться в других, для уменьшения дупликации кода.

Рассмотрим следующую группу представлений, реализованных в проекте.

Данная группа представлений создана для управления курсом:

– просмотр курса;
– создание курса;
– редактирование курса;
– удаление курса.

В каждом классе указан атрибут permission_required, который будет регулировать доступ только пользователям с соответствующими разрешениями.

Следующее представление StudentCourseDetailView реализовано для студентов.

В нем переопределен метод get_queryset (), чтобы ограничить базовый набор запросов курсами, на которые зачислен студент. Также переопределен метод get_context_data (), чтобы установить модуль курса в контексте, если указан URL-параметр module_id. В противном случае устанавливается первый модуль курса. Таким образом, студенты смогут перемещаться по модулям внутри курса.

Результаты проектирования

Итогом разработки стала готовая система дистанционного обучения. Во время выполнения были пройдены следующие этапы:

– сформирован объект автоматизации;
– поставлены основные цели и задача проекта;
– выполнено проектирование программного обеспечения и базы данных;
– выполнено создание базы данных;
– выполнено кодирование, отладка и тестирование программного обеспечения.

Список использованных источников:

1. Дэн Бейдер «Чистый Python. Тонкости программирования для профи» [Электронный ресурс]. – Режим доступа: https://flibusta.club/b/546314/read. – Дата доступа: 15.04.2023.

2. Django документация [Электронный ресурс]. – Режим доступа: https://djbook.ru/rel1.9/intro/overview.html. – Дата доступа: 15.04.2023.

3. Дронов Владимир Django: практика создания Web-сайтов на Python – М.: БХВ-Петербург, 2016. – 865 с.

4. Проектирование программного обеспечения с использованием стандартов uml 2. 0 и SysML [Электронный ресурс]. – Режим доступа: https://cyberleninka.ru/article/n/proektirovanie-programmnogo-obespecheniya-s-ispolzovaniem-standartov-uml-2-0-i- sysml-1-0/viewer. – Дата доступа: 20.04.2023.