УДК 004

Тенгаева Айжан Абденовна
кандидат физико-математических наук, ассоц. профессор
Оразбек Куаныш
магистрант 2-го курса
Казахский национальный аграрный исследовательский университет, Казахстан

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ DJANGO

Tengayeva A.A.
candidate of physical and mathematical sciences, associate professor
Orazbek Kuanysh
2nd year master's student
Kazakh National Agrarian Research University, Kazakhstan

DEVELOPMENT OF WEB-APPLICATIONS USING DJANGO

Summary. Development of web-applications using Django is considered. Explains the core standards of the modern web framework Django, which has changed the definition of web development in the Python world. The Electronic Syllabus web application is considered as an example.

Аннотация. Рассматриваются разработка web-приложений с использованием Django. Поясняются основные стандарты современного веб-фреймворка Django, который изменил определение веб-разработки в мире Python. В качестве примера рассматривается создание веб-приложение «Электронный силлабус» .

Key words: development, web application, Django, Python, framework, MVC pattern, standards, technology, model, controller, syllabus

Ключевые слова: разработка, веб-приложение, Django, Python, фреймворк, шаблон MVC, стандарты, технология, модель, контроллер, силлабус

Введение. Способы разработки web-приложений могут быть разделены на три большие категории: подходы, основанные на программировании или скриптах: внешние программы или скрипты; расширения web-сервера; подходы, основанные на использовании шаблонов web-страниц, включающих вставки кода скриптов и специальных серверных тэгов; объектные среды (фреймворки).

Фреймворки представляют собой совершенствования разработки web-приложений вместо объединения разметки и логики в единый модуль. Фреймворки поддерживают принцип отделения содержания от представления [1].

Django - это веб-фреймворк Python высокого уровня, который поддерживает быструю разработку и элегантный дизайн. Полноценный подход, прагматичный дизайн и превосходная документация - вот некоторые из причин его успеха. Django работает для автоматизации производства общих веб-разработок задачи, чтобы разработчик мог вместо этого сосредоточиться на проблемах приложения более высокого уровня низкоуровневой обработки данных.

Django поддерживает модель-представление-контроллер (MVC)- разделение задач таким образом, чтобы сделать разработку проще и продуктивнее. Разработчики Django называют это «веб-фреймворком для перфекционистов с установленными сроками». которое означает, что чистый дизайн и соблюдение некоторых стандартных практик. В Django, как и в большинстве фреймворков, можно следовать модель того, как разрабатывать и структурировать код и пользоваться преимуществами фреймворка, или вы можете бороться с фреймворком и рисковать. Его функции охватывают все, от связи до баз данных, от обработки URL-адресов и создания шаблонов веб-страниц. Существуют полные книги, в которых подробно рассматривается Django [2].

Шаблон MVC - это архитектурный шаблон, изначально разработанный для Smalltalk, язык объектно-ориентированного программирования, которое позволяет четко разделить логику представления, логику управления и бизнес-объекты. Архитектуры и фреймворки, включающие аспекты концепции MVC, имеют множество приложений для решения задач, требующих любого нетривиального уровня представления. Наборы инструментов и фреймворки, которые используются для разработки большинства полнофункциональных клиентских приложений для Windows, Платформы Mac, Linux и UNIX каким-то образом включают концепцию MVC. Множество распространенных вариаций концепции были определены, и подходы к применению MVC для множества проблем.

Шаблон MVC определяет три основных компонента следующим образом:

- Просмотр: управляет выводом на дисплей (графическим или текстовым). Это состоит из необходимой логики для создания компонентов пользовательского интерфейса и использования модели для загрузки данных в компоненты.
- Контроллер: управляет вводом данных с клавиатуры и мыши от пользователя. Контроль осматривает и интерпретирует предоставленные

входные данные. На основе входных данных он затем пересылает команды модели или просмотр.

- Модель: управляет данными домена и их обработкой. Это включает в себя создание постоянного представления для данных домена и операций, которые могут быть выполнены с данными.

Шаблон также определяет следующие ограниченные отношения между каждым из объектов:

- Вид и контроллер должны иметь прямое однозначное соответствие. Каждый вид должен иметь ссылку на контроллер и наоборот.
- Отношения между представлением и моделью, а также между контроллером и моделью меньше непосредственный.

Следовательно, следует использовать протокол уведомления, посредством которого регистрируется представление или контроллер как интересующийся набор моделей. Когда модель меняет свое внутреннее состояние, она должна уведомить набор обо всех его заинтересованных контроллерах и представлениях. Модель никогда не должна напрямую взаимодействовать с контроллером или представлением. Надо обратить внимание, что когда адаптируем шаблон MVC к веб-приложению, отношения между представление и модель, а также между контроллером и моделью обычно не поддерживаются. Это не практичный подход, учитывая отключенный характер веб-приложений [2].

Основная часть.

Основные страницы приложения "Электронный силлабус" состоят из 4 страниц. Это: страница админа, страница авторизации, страница выбора и готовая страница силлабуса.

Первая страница - страница авторизации. То есть преподаватели должны написать свой логин и пароль. Страница авторизации веб-приложения "Электронный силлабус" представлена на рисунке 1.

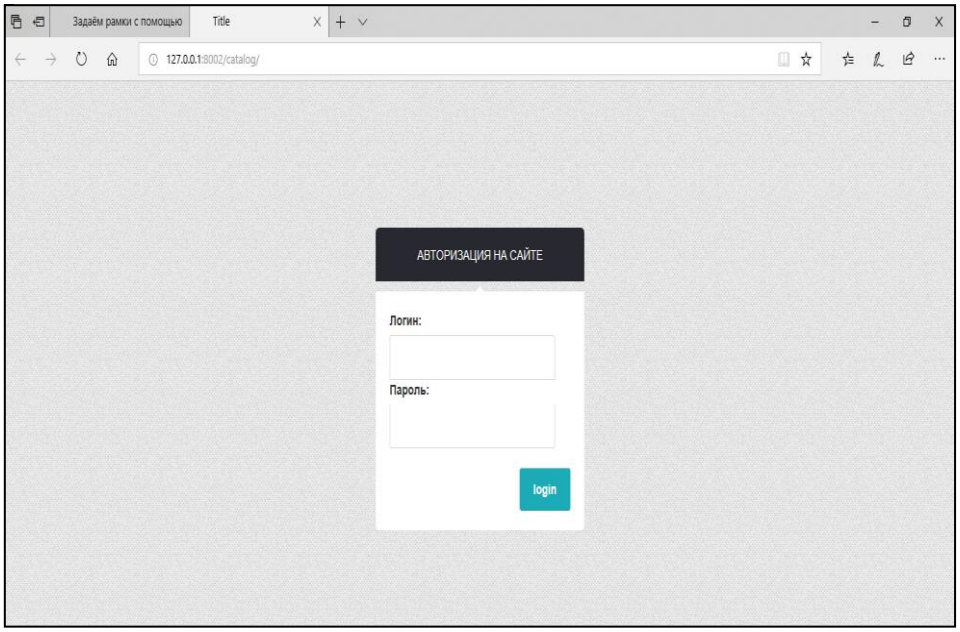

Рисунок 1. Страница авторизации веб-приложения "Электронный силлабус"

Страница выбора веб-приложения «Электронный силлабус» показано на рисунке 2. Функции, используемые на главной странице:

– создать стол
– выбор преподавателя
– учитель практики
– учитель лаборатории
– выбор уроков
– добавить урок
– выбор постреквизитов
– выбор предпосылок
– выберите список литературы
– добавить литературу
– сохранить в формате PDF

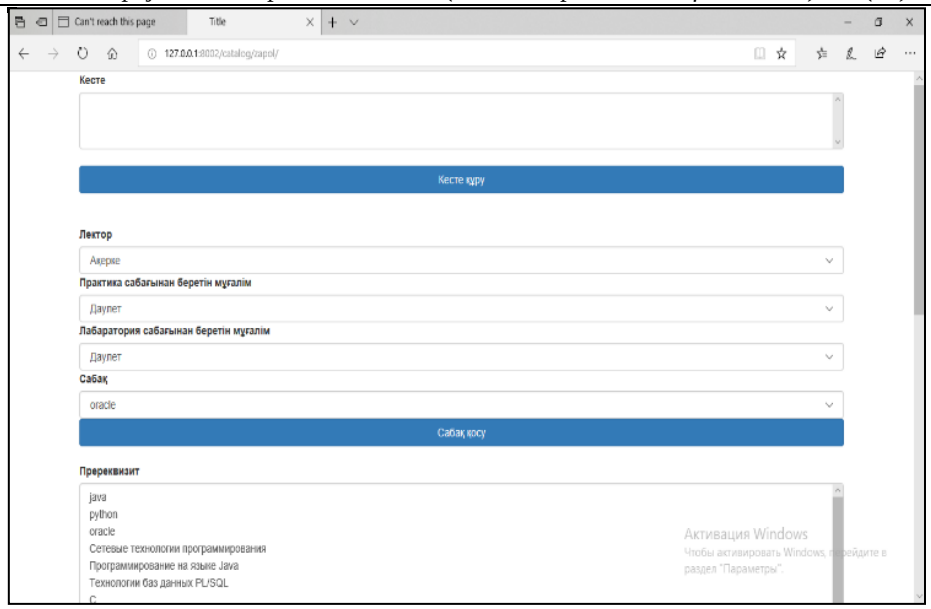

Рисунок 2. Страница выбора веб-приложения "Электронный силлабус"

На закладке каждую неделю отображается проект семестра с использованием операции выбора темы и операции выбора лабораторных работ. На этой странице каждую неделю визуально отображается небольшая таблица, когда вы нажимаете кнопку выбора темы и лабораторной работы. По прошествии всех 15 недель нужно нажать кнопку возврата. Когда вы нажимаете эту кнопку, он возвращается на главную страницу. Визуальный график также включает недели, лекции и лабораторные работы. На рисунке 3 показана страница создания таблицы веб-приложения «Электронный силлабус».

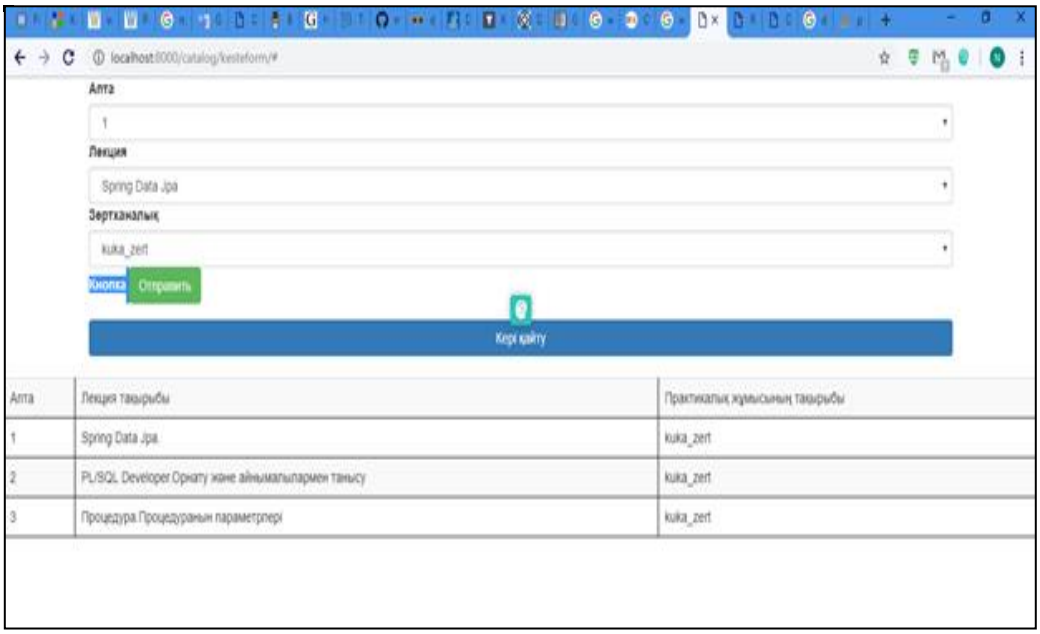

Рисунок 3. Страница создание таблицы веб-приложения "Электронный силлабус"

Страница администратора должна позволять суперпользователям редактировать, удалять и создавать данные. Чтобы перейти на страницу администратора, вам нужно добавить ссылку / admin в url. На странице "Модель" в базе данных создается таблица путем создания класса. И здесь следует указать взаимосвязь таблиц. После того, как вы создали класс на странице моделирования, вам нужно будет зарегистрировать эти классы на Admin.py.

Только после этого этапа, наши таблицы будут отображаться на странице администратора. И вам нужно произвести необходимые настройки на странице settings.py. Страница models.py веб-приложения «Электронный силлабус» показано на рисунке 4.

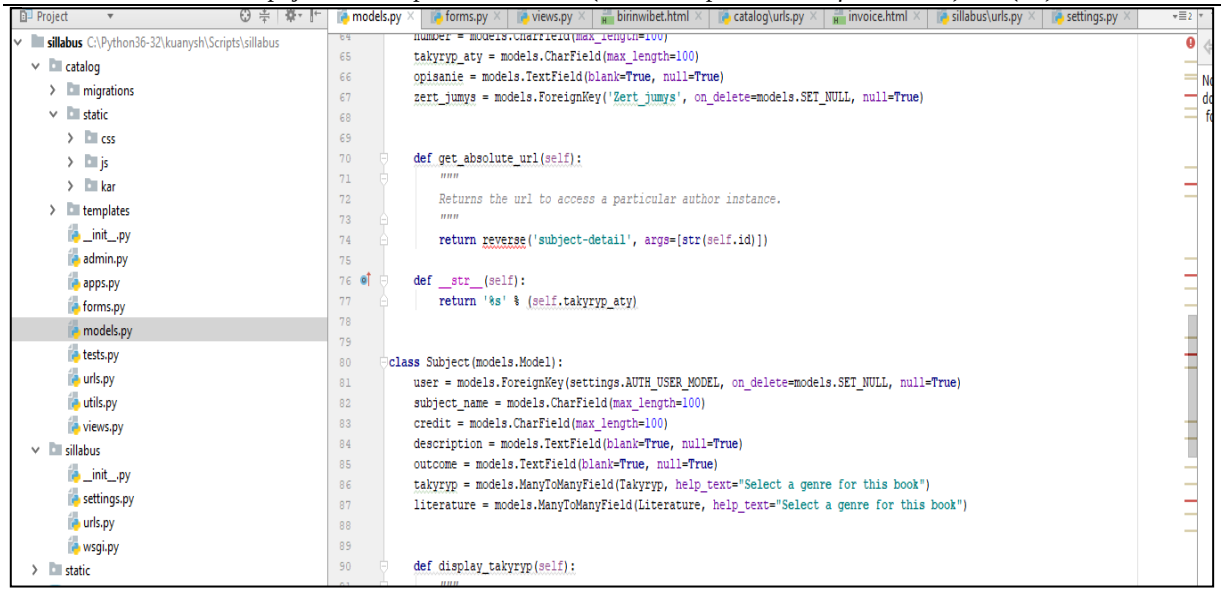

Рисунок 4. Страница models.py веб-приложения "Электронный силлабус"

На странице View создается информация, которая необходима пользователю для отображения на странице просмотра и обнаруживает, отвечает на запрос информации. Есть два типа спроса GET и POST. Страница views.py веб-приложения «Электронный силлабус» показано на рисунке 5.

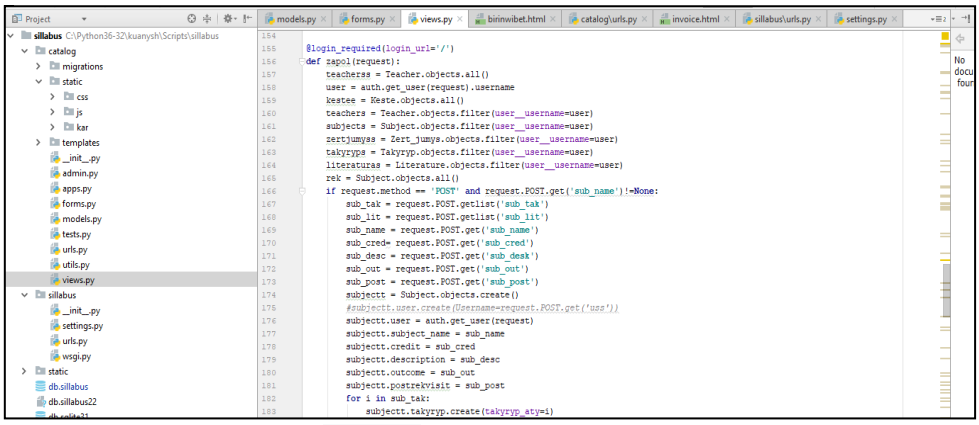

Рисунок 5. Страница views.py веб-приложения "Электронный силлабус"

Страница контроллера отвечает на ссылки во всех наших приложениях. Функция просмотра страницы отображается для каждой ссылки. Страница urls.py веб-приложения электронный силлабус показано на рисунке 6.

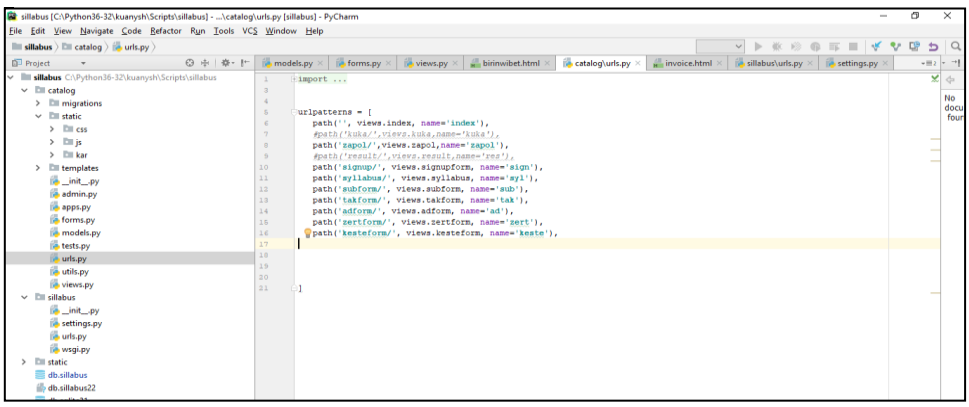

Рисунок 5. Страница urls.py веб-приложения "Электронный силлабус"

Дизайн веб-приложения «Электронный силлабус» разработан для мобильных телефонов с использованием технологий CSS и Bootstrap. Для университета использовались простые белые и

голубые цвета. Базовый дизайн веб-приложения «Электронный силлабус» показано на рисунке 7.

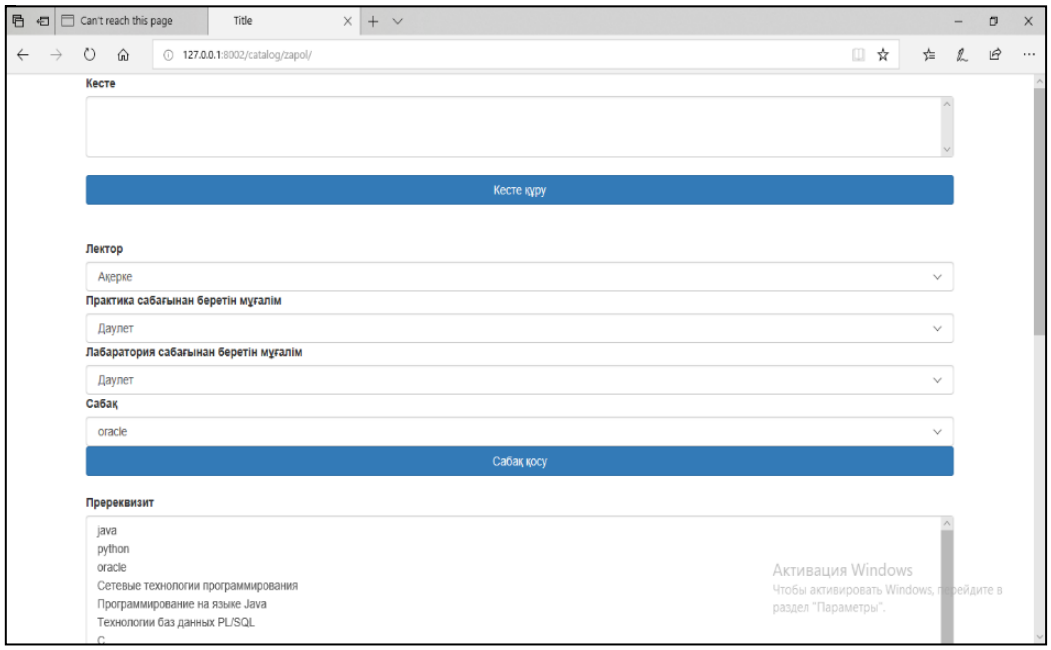

Рисунок 7. Основной дизайн веб-приложения "Электронный силлабус"

Заключение. Все популярные платформы, используемые для создания веб-приложений, работают в шаблоне MVC. Основная идея паттерна MVC - Модель, Представление, Контроллер. Фреймворк Django, поддерживающий этот шаблон, реализует его основные идеи. Веб-приложение электронного силлабуса было создано с использованием Python-фреймворка Django, которое полностью раскрывает шаблон MVC.

Список литературы:

1. Лобода Ю.Г., Орлова О.Ю. Технологии разработки web-приложений. Научные труды Одесской национальной академии пищевых технологий. Одесса , 2014. Выпуск 46. Том 1. стр. 239-244.

2. Juneau J., Baker J., Ng V., Soto L., Wierzbicki F. Web Applications With Django. In: Anglin S. et al. (eds) The Definitive Guide To Jython. Apress. 2010. pp.281-325.

3. Рябова К.М. Фреймворк Django: архитектура и возможности. Сборник статей XIII международной научно-практической конференции: Современные технологии: актуальные вопросы, достижения и инновации Пенза, 2018 г. стр. 118-120.

4. https://docs.python- guide.org/scenarios/web.

5. https://uwsgi-docs.readthedocs.io/en/latest/tutorials/Django_and_nginx.html.

6. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django.