

# **РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ В РАМКАХ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**КОЧЕТКОВА ЕЛЕНА АЛЕКСАНДРОВНА**

преподаватель факультета среднего профессионального образования,  
ФГБОУ ВО «Мордовский государственный педагогический  
университет имени М. Е. Евсевьева», г. Саранск, Россия

**ОГУРЦОВ НИКИТА ПАВЛОВИЧ**

студент факультета среднего профессионального образования,  
ФГБОУ ВО «Мордовский государственный педагогический  
университет имени М. Е. Евсевьева», г. Саранск, Россия

Ключевые слова: web-ресурс, разработка, дизайн, электронный образовательный ресурс, Figma, Python, PyCharm Professional, Django, MySQL.

## **DEVELOPMENT AND DESIGN OF ELECTRONIC EDUCATIONAL RESOURCES WITHIN THE FRAMEWORK OF SECONDARY VOCATIONAL EDUCATION**

**KOCHETKOVA ELENA ALEKSANDROVNA**

lecturer of the Faculty of Secondary Vocational Education,  
FGBOU VO «Mordovian State Pedagogical University  
named after M. E. Evsevyev», Saransk, Russia

**OGURTSOV NIKITA PAVLOVICH**

student of the Faculty of Secondary Vocational Education,  
FGBOU VO «Mordovian State Pedagogical University  
named after M. E. Evsevyev», Saransk, Russia

Keywords: web-resource, development, design, electronic educational resource, Figma, Python, PyCharm Professional, Django, MySQL.

В современном мире постоянно развивающихся технологий для каждой компании как никогда важна автоматизация бизнес-процессов, позволяющая выйти на более высокий уровень. Любой бизнес, который осуществляется посредством взаимодействия с широкой аудиторией, требует постоянной рекламы. Учитывая тот факт, что реклама в печатных изданиях не способна в полной мере осветить деятельность предприятия, а информация в рекламных буклетах и брошюрах быстро теряет актуальность, наиболее эффективным инструментом продвижения товаров или услуг являются интернет-ресурсы.

Любая компания или организация имеет собственный web-сайт, содержащий информацию о предоставляемых услугах. Выбор данных информационных систем достаточно велик – от нескольких персональных страниц до объемных бизнес – ресурсов.

Для многих предприятий создание собственного представительского сайта является актуальным. Несомненным плюсом готового сайта является возможность обновления информации, а также понятный и адаптивный интерфейс.

Информационные технологии широко используются во всех сферах современной жизни, в том числе в образовании. Собственный сайт становится неотъемлемым атрибутом любого образовательного учреждения. На первый взгляд, сайт должен соответствовать очевидным требованиям: полезный и понятный контент, профессиональный дизайн, удобная навигация и интерактивность, но зачастую они не выполняются.

Для уверенного и эффективного существования в сети web-сайт учебного заведения должен представлять собой полноценный информационно-коммуникационный ресурс.

Сегодня в создании сайта участвуют не только программисты, но и разработчики интерфейсов, которые понимают тонкости подачи информации и ее использования посетителями сайта. Разработка интерфейса сайта играет большое значение, так как помогает человеку взаимодействовать с устройством, решать задачи, достигать быстрее определенных целей.

Первая связь посетителя с web-сайтом основана на визуальном восприятии страницы, потому нужно следить за цветовым балансом и правильно подбирать пропорции в оформлении. Грамотный web-дизайн – это показатель имиджа, отношения к посетителям сайта.

Также необходимо предоставлять посетителям различные возможности: обратную связь, форум, голосование, различные web -тесты, поиск по сайту, счётчик посещений и многое другое вы не можете сделать это с помощью обычного HTML. Эта форма имеет все необходимые атрибуты: и поле для ввода текста, и кнопку отправки. Однако, при нажатии на кнопку в большинстве случаев не произойдет ровным счетом ничего – ведь не был задан сценарий действий, которые следует выполнить, чтобы получить результат. Поэтому создание интерактивных компонентов является сложной задачей для web-программистов.

Основными элементами web-программирования являются – скрипты. Данные программные средства делятся на два вида: серверные и клиентские. Серверные скрипты выполняются на стороне сервера, то есть на компьютере, на котором размещен сайт. Они выполняются еще до загрузки страницы сайта на компьютер пользователя.

Клиентские скрипты, в свою очередь, выполняются на компьютере клиента после загрузки страницы с сервера и не требует дополнительной перезагрузки.

Языки web-программирования, на которых выполняются и те, и иные скрипты различны. Некоторые из этих языков используются только для создания серверных скриптов,

другие используются только для клиентских скриптов, а многие используются и для того, и для другого.

Используя новейшие технологии, можно создать удобный и красиво оформленный интерактивный web-ресурс, понятный и доступный пользователю, в рамках программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование. В процессе обучения рассматриваются основные виды web-ресурсов, языки программирования и на основании выбранного средства реализации строится подход к формированию структуры курсового проекта.

В последние годы очень сильно возросла роль интернета как в оффлайн-бизнесе, так и для бизнеса в самой всемирной паутине, то появилась нужда в web-мастерах, кто сможет организовать всё это. Можно выделить три группы web-разработчиков:

- Front-end – разработчики занимаются визуальной составляющей сайта. Часть сайта, которую видит его посетитель. Он делает дизайн, каркас, стили, меню используя в качестве своего вооружения HTML5, CSS, JavaScript.

- Back-end разработчик – это тип программиста, который создает логическую серверную часть и основную вычислительную логику web-сайта, программного обеспечения или информационной системы. Разработчик создает компоненты и функции, к которым пользователь косвенно обращается через внешнее приложение или систему.

- Full-stack – программисты совмещают обязанности Front-end и Back-end разработчиков. В основном такие разработчики используют язык программирования напрямую связанной с HTML, такие как PHP. Это сокращает время необходимо на разработку.

Существует широкий диапазон классификаций web-сайтов, каждый из которых удовлетворяет разные потребности, использует альтернативный план и сопровождает меняющиеся затраты на структуру. Рассмотрим несколько популярных разновидностей web-сайтов:

- Бизнес-сайт – это любой сайт, посвященный определенному бизнесу. Он должен быть брендирован как компания (похожий логотип и позиционирование) и отражать виды товаров и услуг, которые предлагает компания/бизнес. На данный момент каждая компания, большая или маленькая, имеет сайт. Каждый потенциальный клиент, с которым вы столкнетесь, просто будет ожидать, что, если он погуглит ваш бизнес в поисках дополнительных данных, он обнаружит сайт. Иначе это вызовет недоверие клиента и уход к потенциальным конкурентам.

- Web-сайты магазинов. Web-сайты, на которых пользователи могут делать покупки и совершать покупки. Все они использовали бы эти сайты множество раз, и они стали

обязательными для каждого продающего бизнеса. С развитием технологий разработки веб - сайтов стало относительно легко создать сайт электронной коммерции и начать продавать.

– Сайты образовательных организаций относятся к классу образовательных сайтов. Эти сайты имеют основную цель либо предоставить учебные материалы посетителям, либо предоставить им данные об образовательной организации. На некоторых образовательных сайтах будет реклама, как на развлекательных и медиа-сайтах. Более того, некоторые из них используются в качестве цифрового присутствия для образовательной организации. В связи с растущим интересом к этим сайтам эту функцию предлагают многочисленные разумные устройства.

– Web-сайты социальных сетей. Социальные сети существуют в различных формах. Эти web-сайты обычно создаются для того, чтобы люди могли делиться размышлениями, фотографиями или мыслями или просто общаться с другими людьми по определенной теме. Web-сайты социальных сетей постепенно превращаются в место, где люди могут узнавать новости, делать покупки и общаться с людьми. Социальные сети оказывают существенное влияние во всех областях.

Для реализации курсового проекта были выбраны следующие средства разработки:

- Figma - инструмент для совместной разработки интерфейсов;
- Python – язык программирования;
- PyCharm Professional – IDE среда web-разработки для языка программирования Python;
- Django – фреймворк для языка Python.

Рассмотрим каждое средство отдельно.

Figma – это инструмент для web-дизайна, полностью основанный на облачных вычислениях. Это позволяет пользователю работать с членами своей команды одновременно над одним и тем же файлом без каких-либо хлопот.

Figma была создана для преобразования дизайна в код наилучшим образом: web-разработчики могут проверять, копировать и экспортировать CSS и ресурсы дизайна непосредственно из файлов дизайна Figma, устраняя догадки и упрощая передачу. Figma объединяет многие основные функции других инструментов в одном продукте для более эффективного и мощного рабочего процесса.

PyCharm Professional – интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает web-разработку.

Современная web-разработка на Python очень проста благодаря прямой поддержке PyCharm популярных сред web-разработки, таких как Django, Flask, Google App Engine, Pyramid и web2py.

Django – это высокоуровневая web-инфраструктура Python, которая способствует быстрой разработке и чистому, прагматичному дизайну. Созданный опытными разработчиками, он берет на себя большую часть хлопот web-разработки.

Django серьезно относится к безопасности и помогает разработчикам избежать многих распространенных ошибок безопасности, таких как внедрение SQL, межсайтовый скриптинг, подделка межсайтовых запросов и кликджекинг. Его система аутентификации пользователей обеспечивает безопасный способ управления учетными записями пользователей и паролями.

Для разработки сайта были выведены основные этапы разработки:

1. *Аналитика*. Начать стоит с определения целевой аудитории и анализа конкурентов. Важно понимать свою целевую аудиторию и её потребности. Тогда можно адаптировать сайт под их интересы и получить из этого собственную выгоду. Стоит найти ответы на вопросы:

- Для кого предназначен web-ресурс?
- Каковы предпочтения ваших потенциальных клиентов?
- Какая информация будет для них полезной и интересной?
- Какими коммуникационными каналами они пользуются?

2. *Создание технического задания (ТЗ)*. Погрешности здесь недопустимы, так как они могут проявиться на любом этапе и в итоге отбросить проект далеко назад. В ТЗ прописываются основные требования проекта, его возможности и способ работы, учитывая анализ целевой аудитории и конкурентов, цели и стратегию.

3. *Создание макетов дизайна*. Для дальнейшей работы дизайнером создается макет с подробным расположением элементов и навигацией. Последняя требует особенного внимания. Пользователь должен при первом контакте с продуктом понять, как им пользоваться. Дизайнер в первую очередь обеспечивает удобство интерфейса. Все элементы на странице должны быть расположены таким образом, чтобы пользователь без труда мог найти и воспользоваться ими. При помощи расположения информационных блоков дизайнер управляет вниманием посетителя, демонстрирует ему то, что он должен или хочет увидеть в данный момент. При формировании дизайна используется единый стиль, вписывается логотип, что делает сайт полностью узнаваемым и соответствующим организации для которой формируется.

4. *Верстка макета*. Это расположение всех визуальных элементов на web-странице. Благодаря преднамеренному позиционированию элементов страницы можно контролировать взаимосвязь между ними, чтобы лучше направлять пользовательский опыт. После

завершения работы макет отправляется на утверждение и только потом передается на адаптивную верстку. Она подразумевает перевод макета в HTML, адаптацию изображения сайта под любой браузер и устройство.

5. *Программирование.* Back-end разработка охватывает логику и интеграцию web-приложений на стороне сервера, а также действия, такие как написание API, создание библиотек и работа с системными компонентами. Back-end разработчики создают код, который позволяет базе данных и приложению взаимодействовать друг с другом. Back-end разработчики заботятся и поддерживают внутреннюю часть web-сайта, включая базы данных, серверы и приложения.

6. *Базовое наполнение.* Этап заполнения предполагает перевод контента в необходимый формат и размещение на страницах ресурса. Материалы готовятся заранее. Но на этом работа не прекращается. Контент требует стабильного обновления. Иначе поисковые системы при ранжировании будут обходить сайт. Немаловажный фактор – заинтересованность пользователей.

Разработка структуры сайта – важный этап проектирования, от которого зависит не только удобство создаваемого ресурса для пользователя, но и эффективность его поискового продвижения.

Структура является каркасом создаваемого информационного портала, это своего рода скелет, который отображает логику взаимосвязи между его страницами. Она разделяет сайт на систему разделов и подразделов, что может значительно упростить пользователю навигацию и поиск нужного контента.

Web-структура – это то, как информация организована и взаимосвязана на web-сайте. Эффективная структура сайта повышает удобство использования и взаимодействие с пользователем, что делает web-структурирование важным шагом в процессе web-дизайна.

Дизайн сайта играет немаловажную роль в создании самого сайта. Проанализируем возможные цветовые решения для web-ресурса. Первое решение можно назвать «деловым». Оно отличается холодными цветами – темно-зеленый, синий различных оттенков, а также коричневый. Как правило, здесь можно встретить черный текст на белом фоне, синие ссылки и строгое оформление. Такое решение обычно используют для своих сайтов крупные компании, специализирующиеся на IT-технологиях, промышленные, строительные предприятия. Такое сочетание цветов говорит потенциальному клиенту или заказчику о надежности и серьезности предприятия.

Второе решение характеризуется яркостью, броскостью цветов, сочетаниями розового, красного, салатного, желтого цветов. Такой выбор делают как правило создатели развлекательных сайтов, в том числе некоторых интернет-магазинов. Их целью является

необычное оформление и привлечение людей, как правило молодых, для общения, обсуждения общих тем разной направленности.

Третье решение – «спокойное». Его характеризует преобладание неярких, постельных тонов. Его часто можно наблюдать на сайтах учебной направленности, так как оно не отвлекает учащегося от процесса усвоения знаний; также такое решение можно встретить на информационных сайтах, особенно созданных для людей разных возрастов.

Для оформления сайта выбран «спокойный» дизайн, так как оно больше всего подходит для проекта учебной направленности. Основной фон страниц сайта сделан белым, текст окрашен в тёмно-серый, цветом для ссылок стал синий. Для выделения части текста или проведения разделительной полосы используется светло-серый цвет. Такое сочетание не раздражает человека, цвета не смазываются, текст и ссылки хорошо видны на белом фоне.

Web-сайт – совокупность логически связанных между собой web-страниц также место расположения контента сервера. Обычно сайт в интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователем как единое целое. Web-сайты так называются потому, что доступ к ним происходит по протоколу HTTP.

Создана совокупность логически связанных между собой web-страниц. В структуре сайта нет ничего необычного. Есть главная страница, на которой создано меню. При нажатии на каждый из пунктов меню открывается новая страница или прокручивается на нужное место.

Верстка – это расположение всех визуальных элементов на web-странице. Благодаря преднамеренному позиционированию элементов страницы можно контролировать взаимосвязь между ними, чтобы лучше направлять пользовательский опыт.

Главное меню сайта содержит восемь пунктов: «Главная», «О факультете», «Учебный процесс», «Новости», «Сотрудники», «Задать вопрос», «Контакты».

Для стилизации меню использован язык таблиц стилей Cascading Style Sheets (CSS), стили используются для создания и изменения стиля элементов web-страниц и пользовательских интерфейсов, написанных на языках HTML.

В дальнейшем будет использоваться web-фреймворк Django то все наши получившиеся HTML страницы необходимо оптимизировать для работы с данным фреймворком.

Все повторяющиеся элементы HTML страниц необходимо перенести в отдельный файл. Эти файлы используются как шаблон при создании страницы сайта, при этом на сайте имеется главный, базовый шаблон в который в последующем подключаются остальные шаблоны.

Для реализации шаблона используется функция `extends` и название этой функции. Шаблон повторяющегося элемента помещается в тело функции `extends`, а в базовом шаблоне остаётся ссылка на функцию.

Django основан на архитектуре MTV (Model – Templates – View) – это шаблон проектирования программного обеспечения для разработки web-приложения (рисунок 1).

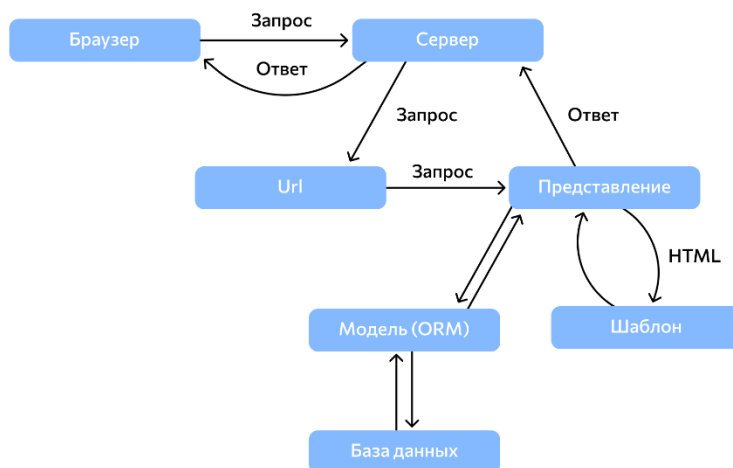


Рисунок 1 MTV архитектура web-фреймворка Django

Структура MTV состоит из модели, шаблона и представления, рассмотрим эти части подробнее:

- Модель будет действовать как интерфейс ваших данных. Он отвечает за сохранение данных. Это логическая структура данных, лежащая в основе всего приложения и представленная базой данных (обычно это реляционные базы данных, такие как MySQL, Postgres).

- Шаблон состоит из статических частей желаемого вывода HTML, а также некоторого специального синтаксиса, описывающего, как будет вставляться динамический контент.

- Представление – это пользовательский интерфейс, при отображении web-сайта в браузере. Он представлен файлами HTML/CSS/Javascript и Jinja.

Структура проекта и файловая структура. Django использует структуру каталогов для организации различных частей web-приложения. Он создает проект и папку приложения для этого. Создание надлежащего проекта и его организация помогают сохранить проект чистым. При создании проекта, сам Django создает корневой каталог проекта с именем проекта.

Логическая модель включает построение структуры базы данных, которая содержит описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных.

Модели данных используются организациями для визуальной передачи отношений и связей между данными. Логические модели данных предоставляют подробную информацию



о данных и их объектах, обеспечивая визуальный контекст данных, атрибутов и взаимосвязей внутри них. Они служат образцом для описания того, как система должна быть реализована, не встраиваясь в конкретную базу данных или систему управления базами данных.

Для начала проектирования серверной части web-сайта, необходимо создать проект с виртуальным окружением, чтобы в дальнейшем не было проблем с версиями языка программирования и web-сайт работал корректно при добавлении его на хостинг.

Для этого в среде разработки PyCharm необходимо создать проект с указанием версии Python Venv (виртуальное окружение).

После создания проекта с встроенным виртуальным окружением необходимо установить библиотеку фреймворка Django для этого открываем терминал и указываем команду `pip install django`. После установки фреймворка в терминале указывается команда `Django admin startproject [название проекта]`. Django выстроит структуру файлов проекта и всё необходимое для его работы.

Чтобы созданные нами ранее HTML страницы открывались на сайте необходимо их указать в файле представлений с помощью класса `ListViews`, а также прописать для этих страниц URL адрес, по которому будет открываться каждая HTML страница на сайте.

Для подключения к web-сайту базы данных в фреймворке Django необходимо создать модель данных и с помощью миграции добавить эти модели в выбранную нами базу данных MySQL.

Для оптимизации сайтов с большим количеством пользователей в современном мире используется кэш-браузера.

Кэш – это специальное место для хранения временных файлов, благодаря которому устройство, браузер или приложение работают быстрее и эффективнее.

Запоминается и уменьшается количество SQL запросов на сайте с помощью их кэширования, это позволяет снизить нагрузки на базу данных и ускоряет загрузку страниц сайта, кэшированные страницы некоторое время хранятся в файловом каталоге web-сайта после чего удаляются и перестаёт использоваться на сайте, до того момента пока запрос снова не попадёт в кэш.

На стадии разработки web-сайта для отслеживания какие шаблоны в данный момент применяются на странице web-сайта, кэширования страниц, времени их загрузки и так далее, на сайте используется `debugtoolbar` он доступен только на стадии разработки, когда в файле параметров Django флаг `Debug` установлен в значении `True`.

Для того чтобы передавать параметры из базы данных в HTML страницы при их вызове через класс `Listview` используется оператор `return A` в шаблоне HTML с помощью

команд python циклов for и оператора выбора if эти данные перебираются и отображаются на web-странице.

Web-административный сервер – стандартная функция современных web-сайтов. Интерфейс администратора, или сокращенно admin, позволяет доверенным администраторам сайта создавать, редактировать и публиковать контент, управлять пользователями сайта и выполнять другие административные задачи.

Одной из самых мощных частей Django является автоматический интерфейс администратора. Он считывает метаданные из моделей данных, чтобы предоставить быстрый, ориентированный на модели интерфейс, в котором доверенные пользователи могут управлять контентом на web-сайте.

Регистрация и настройка модели для администратора осуществляется путем добавления моделей в файл events приложения admin.py.

После добавления моделей в файл admin.py, в панели администратора отображаются все базы данных web-сайта. С помощью этой панели можно добавлять удалять редактировать и просматривать данные из СУБД.

Во вкладке категории можно добавлять категории сотрудников. Во вкладке «новости» и «сотрудники» можно добавлять и редактировать записи, а также управлять их публикацией с помощью чекбоксов в панели администратора. Также в каждой вкладке присутствует фильтр записей по публикации, времени создания и категории.

Любой сайт, web-система или web-приложение имеют админ-панель, доступ к которой осуществляется через авторизацию пользователя. Лицо, ответственное за редактирование и наполнение сайта информацией, вводит имя пользователя и пароль по соответствующей web-ссылке и получает доступ к Content Management System (CMS) – системе управления контентом. Большинство админ-панелей имеют типовой функционал, а для ускорения разработки цифровых решений удобно использовать гибкие многофункциональные админ-панели.

Современный подход к созданию сайтов и web-приложений позволил преодолеть вышеуказанные ограничения. Сейчас принято использовать готовые шаблоны админ-панели на основе фреймворков. Они позволяют реализовать весь функционал, необходимый для управления контентом, с учетом необходимого качества UX/UI дизайна.

Результатом курсового проекта является web-сайт для студентов и преподавателей факультета среднего профессионального образования разработанный на языке Python. Для реализации проекта использован язык программирования Python, подключен и настроен web-фреймворк Django. Макет web-сайта создан в инструмент для совместной разработки

интерфейсов Figma. Также для работы с данными была построена структура базы данных и спроектирована в системе управления базами данных MySQL.