**ЛР 1 HTML**

**Главная**: Краткая информация о последней опубликованной статье, реклама в виде баннера, список компаний партнеров (логотипы со ссылками на сайты компаний – выбрать произвольно)

**О компании**: информация о компании, видео, логотип, история по годам, реквизиты, сертификат (пока только текст без оформления стилями)

Пример сертификата:



**Новости**: список статей в соответствии с тематикой сайта с заголовком, кратким содержанием (одно предложение), картинкой и кнопкой «Читать далее» при нажатии на которую открывается вся статья

**Словарь терминов и понятий**: список часто-задаваемых вопросов с датой добавления на сайт, при нажатии на которые открывается развернутый ответ

**Контакты**: Фото сотрудников с описанием выполняемых работ, телефонами, почтой и т.д.

**Политика конфиденциальности**: в соответствии с тематикой разработанного сайта

Вакансии: список вакансий с описанием

**Отзывы**: список отзывов с указанием имени, оценки, текста, даты. Кнопка добавить отзыв с переходом к окну регистрации или в личный кабинет залогиненного пользователя. При нажатии кнопки открытие формы с полем для текста отзыва и выбора оценки, кнопкой «Отправить», которая сохраняет отзыв в базе.

**Промокоды и купоны**: список промокодов действующих и в архиве.

* микро и мета данные (для использования в поисковых системах) <https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML> ,
* иконку сайта,
* сематическую верстку <https://developer.mozilla.org/ru/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure> ,

<https://www.w3schools.com/html/html5_semantic_elements.asp>

* семантическое выделение текста,
* глобальные атрибуты,
* пример листинга, сокращений, определений, цитат, дата-время, текст в виде четверостишия с фиксированным и потенциальным переводом строки, ссылки (разного вида, в том числе и для скачивания документа), якоря, контейнерные элементы, списки, фигуры, навигация горизонтальная и вертикальная, также должна быть и таблица с заголовочными ячейками, заполненная цифровыми и текстовыми данными, объединенными ячейками, атрибутом headers(не использовать для верстки, только для данных), формы, элементы управления (включая разные виды input, выпадающие списки, контейнеры, кнопки и т.д., а также их валидацию), iframe,изображения (адаптивные), элементы для отображения видео и прослушивания аудио информации.

Предоставить при защите результаты проверки на валидацию сайта и семантической верстки.

Для заполнения использовать генераторы текста и картинок.

Для обеспечения единообразного стиля сайта использовать базовый шаблон ( для справки <https://metanit.com/python/django/2.6.php> , <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Home_page> )

**Варианты заданий:**

| **№** | **Web сайт** | **Ориентировочные модели** | **Описание**  **(что должно быть обеспечено)** | **Отображается для Админа (Superuser)** | **Отображается для User с регистрацией – личный кабинет + все что видит User без регистрации** | **Отображается для User без регистрации** |
| --- | --- | --- | --- | --- | --- | --- |
|  | Автосалон | Сотрудники, Товары, Виды товаров, Изготовители, Заказы, Клиенты | Вы являетесь сотрудником коммерческого отдела компании, продающей автомобили. Вашей задачей является отслеживание финансовой составляющей работы компании. Работа Вашей компании организована следующим образом: на Интернет-сайте компании представлены (выставлены на продажу) автомобили. Каждый из них имеет некоторое название, цену характеристики. Для проведения исследований и оптимизации работы магазина Вы пытаетесь собирать данные с Ваших клиентов. При этом для Вас определяющее значение имеют стандартные анкетные данные, а также телефон и адрес электронной почты для связи. По каждому факту продажи Вы автоматически фиксируете клиента, товары, количество, дату продажи, дату доставки. | Прайс-лист на различные типы товаров;  Список заказчиков с группировкой по городам;  Товар, пользующийся наибольшим спросом;  Товар, не пользующийся спросом;  Ежемесячный объем продаж товаров каждого вида;  Годовой отчет поступлений от продаж;  Анализ данных должен включать следующие элементы:  Сводные таблицы и диаграммы по клиентам, видам продукции, объемам и поступлениям от продаж;  Прогноз продаж;  Построение линейного тренда продаж | User с регистрацией – сотрудник – видит инфо о заказах, продажах и клиентах с которыми работает | инфо о категориях оказываемых услуг и самих услугах с фильтрацией по цене/категории и т.д. |

**Тема:**

1. Веб-программирование;
2. Веб-фреймворк Django

**Цель работы:**

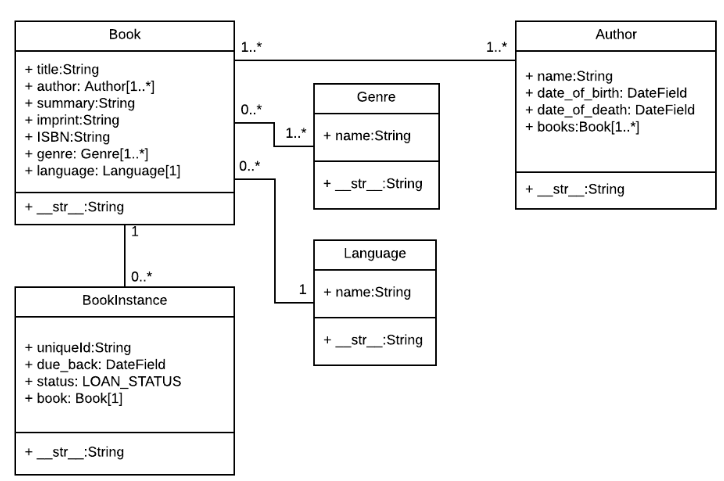
Закрепить знания и навыки по работе со стандартными библиотеками, фреймворком Django.

**Задание.**

Реализовать веб-сайт с использованием Веб-фреймворк Django по индивидуальному заданию (выбрать вариант по номеру в журнале или реализовать свой проект - согласовать с преподавателем до начала разработки)

**Постановка задачи**:

* Определить необходимые сущности для описания предметной области. Реализовать их в терминах моделей, используя подходящие типы данных и связи объектов <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Models>. Представить на диаграмме в виде, как на примере ниже. В качестве базы данных рекомендована sqlite, но можно другую по согласованию с преподавателем.
* Реализовать связи один к одному (OneToOneField), один ко многим (ForeignKey) и многие ко многим (ManyToManyField).



* Реализовать CRUD (create, read, update, delete) операции (<https://metanit.com/python/django/5.4.php> );
* Добавить все модели в админ панель, создать суперюзера, обеспечить работу с данными, фильтрацию, встроенное редактирование связанных записей (<https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Admin_site>).
* Реализовать механизмы авторизации/аутентификации;
* Разграничить доступ в зависимости от прав: например, владелец магазина(superuser), User с регистрацией, User без регистрации(см. инд. задание)
* Обеспечить наполнение данными для демонстрации не менее 10 записей в списке товаров/услуг/объектов/клиентов
* Подключить и использовать как минимум 2 сторонние API (базовый список прилагается), какие именно - выбрать на свой вкус.
* Использовать регулярные выражения для связи URL-адреса с функцией отображения (<https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Generic_views> )
* отображать статистические показатели для своего сайта (использовать данные, соответствующие предметной области). Например:
  + список клиентов/товаров в алфавитном порядке и общую сумму продаж;
  + статистические показатели (среднее, мода и медиана) по сумме продаж;
  + статистические показатели (среднее и медиана) по возрасту клиентов;
  + какой тип товаров наиболее популярен?
  + какой тип товаров приносит наибольшую прибыль?
* отображать тайм зону пользователя, текущую дату, дату добавления/изменения данных в таблицы для тайм зоны пользователя и для UTC, календарь в текстовом виде. Дату указывать в формате (DD/MM/YYYY)
* Номер телефона клиента указывается в формате +375 (29) XXX-XX-XX;
* Клиенты и сотрудники должны иметь возрастное ограничение 18+;
* Реализовать визуализацию в виде диаграммы или графика для отображения распределения показателей по группам и/или изменения показателей по датам/группам.
* Реализовать поиск по любому из параметров и сортировку отображаемых данных
* Добавить тесты на pytest. Использовать fixtures для создания юзеров и соединения к базе данных. Использовать parametrize(<https://pytest-docs-ru.readthedocs.io/ru/latest/parametrize.html> );
* Добавить logging(<https://habr.com/ru/companies/wunderfund/articles/683880/> );
* Добавить тесты; Покрытие тестами кода на 80% и выше;
* Приемлемый внешний вид. Использование css (например <https://www.bootstrapcdn.com/>), при желании реализация своих стилей (найти и скачать шаблоны в интернете);
* Валидация форм как на стороне сервера, так и на стороне клиента;
* Поддержка разного уровня логирования (уровень логирования брать из конфигурации приложения);
* Ограничить использование API проекта для неавторизованных запросов.

**Примечание**: если проект будет не только в вашем гит репозитории, то результат лабораторных в этом семестре обнуляется.

Доработка html, css, js будет выполнена на первых трех ЛР в курсе «Современные технологии разработки web-приложений», поэтому обеспечьте сохранность проекта до следующего семестра.

Для справки cм. также инфо в СЭО:

Список бесплатных API: <https://lms.bsuir.by/mod/resource/view.php?id=214284>

Как создать проект в Django и использовать стороннее API <https://lms.bsuir.by/mod/resource/view.php?id=214285>