



**Министерство науки и высшего образования Российской  
Федерации  
Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

**ФАКУЛЬТЕТ**

**«Радиотехнический»**

**КАФЕДРА**

**ИУ-5 «Системы обработки информации и управления»**

**Отчет по домашнему заданию**

**Разработка интернет-приложений**

**Тема работы: "Создание прототипа веб-приложения на основе  
базы данных с использованием фреймворка Django"**

Выполнил: Мирсонов В. А.

Группа: РТ5-51Б

Дата

выполнения: «20» января 2021 г.

Подпись: \_\_\_\_\_

Проверил: Гапанюк Ю. Е.

Дата

проверки: «20» января 2021 г.

Подпись: \_\_\_\_\_

Москва, 2020 г.

## Содержание

<b>Общее описание задания .....</b>	<b>3</b>
<b>Выполнение домашнего задания .....</b>	<b>3</b>

**Цель домашнего задания** - изучение возможностей создания прототипа веб-приложения на основе базы данных с использованием фреймворка Django.

### **Общее описание задания**

Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

1. Две модели, связанные отношением один-ко-многим.
2. Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
3. Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
4. Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фреймворк по желанию студента.

## **Выполнение домашнего задания**

### **Urls.py**

```
from django.contrib import admin
from django.urls import path, include
from . import views
```

```
urlpatterns = [
    path("", views.index, name='home'),

    path('create-fold', views.FoldCreate.as_view(), name='fold_create_url'),
    path('create-item', views.ItemCreate.as_view(), name='item_create_url'),
    #path('folds/<int:pk>/update', views.FoldUpdate.as_view(), name='fold_update_url'),

    path('first', views.first, name='fir'),
    path('instruction', views.instruction, name='instruction'),
    path('second', views.second, name='sec'),
    path('js-diagram', views.js_diagram, name='js'),
    path('third', views.third, name='thi'),
    path('folds', views.folds_home, name='folds_home'),
    path('items', views.items_home, name='items_home'),
    path('items/full_list', views.item_list, name='items_list'),
    path('items/full_list_f', views.fold_list, name='folds_list'),

    path('folds/<int:pk>', views.FoldDetailView.as_view(), name='fold-detail'),
    path('folds/<int:pk>/update', views.FoldUpdateView.as_view(), name='fold-update'),
```

```

path('folds/<int:pk>/delete', views.FoldDeleteView.as_view(), name='fold-delete'),
path('items/<int:pk>', views.ItemDetailView.as_view(), name='item-detail'),
path('items/<int:pk>/update', views.ItemUpdateView.as_view(), name='item-update'),
path('items/<int:pk>/delete', views.ItemDeleteView.as_view(), name='item-delete'),

```

```

]
```

## Forms.py

```

from django import forms
from .models import *
from django.forms import ModelForm, TextInput, Textarea, Select
from django.core.exceptions import ValidationError
import datetime

class NewFoldCreateForm(forms.ModelForm):
    class Meta:
        model = FoldNewOne
        fields = ["name", "parknumber", "responsible", "adress", "square"]
        widgets = {
            "name": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите название склада'
            }),
            "parknumber": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите номер платформы'
            }),
            "responsible": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите ФИО ответственного (необязательно)'
            }),
            "adress": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите адрес склада (необязательно)'
            }),
            "square": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите площадь склада'
            }),
        }

    def clean_name(self):
        new_name = self.cleaned_data['name']

        if new_name == 'create':
            raise ValidationError('Запись о складе с таким именем не может быть создана!')
        if FoldNewOne.objects.filter(name__iexact=new_name).count():

```

```

        raise ValidationError(
            'Название склада должно быть уникальным. В записях уже существует склад
            "{} {}".format(new_name)

        return new_name

def clean_parknumber(self):
    new_parknumber = self.cleaned_data['parknumber']

    if new_parknumber <= 0:
        raise ValidationError('Запись о номере платформы разгрузки должна быть целым
        неотрицательным числом!')

    return new_parknumber

def clean_square(self):
    new_square = self.cleaned_data['square']

    if new_square <= 0:
        raise ValidationError('Запись о площади склада должна быть целым
        неотрицательным числом!')

    return new_square

```

```

class NewFoldUpdateForm(forms.ModelForm):
    class Meta:
        model = FoldNewOne
        fields = ["name", "parknumber", "responsible", "adress", "square"]
        widgets = {
            "name": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите название склада'
            }),
            "parknumber": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите номер платформы'
            }),
            "responsible": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите ФИО ответственного (необязательно)'
            }),
            "adress": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите адрес склада (необязательно)'
            }),
            "square": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите площадь склада'
            }),
        }

```

```

def clean_name(self):
    new_name = self.cleaned_data['name']

    if new_name == 'create':
        raise ValidationError('Запись о складе с таким именем не может быть создана!')

    return new_name

def clean_parknumber(self):
    new_parknumber = self.cleaned_data['parknumber']

    if new_parknumber <= 0:
        raise ValidationError('Запись о номере платформы разгрузки должна быть целым неотрицательным числом!')

    return new_parknumber

def clean_square(self):
    new_square = self.cleaned_data['square']

    if new_square <= 0:
        raise ValidationError('Запись о площади склада должна быть целым неотрицательным числом!')

    return new_square

classNewItemCreateForm(forms.ModelForm):
    class Meta:
        model = ItemNewOne
        folds = FoldNewOne.objects.all()
        category = (
            ('Мебель'),
            ('Стройматериалы'),
            ('Инструменты'),
            ('Техника'),
        )

        moduls = (
            ('А'),
            ('Б'),
            ('В'),
            ('Г'),
            ('Д'),
            ('Е'),
            ('Ж'),
            ('З'),
        )

        sections = (
            ('S-1'),

```

```
    ('S-2'),  
    ('S-3'),  
    ('S-4'),  
    ('S-5'),  
    ('S-6'),  
    ('S-7'),  
    ('S-8'),  
    ('S-9'),  
    ('S-10'),  
    ('S-11'),  
    ('S-12'),  
    ('S-13'),  
    ('S-14'),  
    ('S-15'),  
    ('S-16'),  
)
```

```
rows = (  
    ('1'),  
    ('2'),  
    ('3'),  
    ('4'),  
    ('5'),  
    ('6'),  
    ('7'),  
    ('8'),  
    ('9'),  
    ('10'),  
    ('11'),  
    ('12'),  
    ('13'),  
    ('14'),  
    ('15'),  
    ('16'),  
    ('17'),  
    ('18'),  
    ('19'),  
    ('20'),  
    ('21'),  
    ('22'),  
    ('23'),  
    ('24'),  
    ('25'),  
    ('26'),  
    ('27'),  
    ('28'),  
    ('29'),  
    ('30'),  
    ('31'),  
    ('32'),  
)
```

```
places = (  
    ('M0'),  
    ('M1'),  
    ('M2'),  
    ('M3'),  
    ('M4'),  
    ('M5'),  
    ('M6'),  
    ('M7'),  
    ('M8'),  
    ('M9'),  
    ('M10'),  
    ('M11'),  
    ('M12'),  
    ('M13'),  
    ('M14'),  
    ('M15'),  
    ('M16'),  
    ('M17'),  
    ('M18'),  
    ('M19'),  
    ('M20'),  
    ('M21'),  
    ('M22'),  
    ('M23'),  
    ('M24'),  
    ('M25'),  
    ('M26'),  
    ('M27'),  
    ('M28'),  
    ('M29'),  
    ('M30'),  
    ('M31'),  
    ('M32'),  
    ('M33'),  
    ('M34'),  
    ('M35'),  
    ('M36'),  
    ('M37'),  
    ('M38'),  
    ('M39'),  
    ('M40'),  
    ('M41'),  
    ('M42'),  
    ('M43'),  
    ('M44'),  
    ('M45'),  
    ('M46'),  
    ('M47'),  
    ('M48'),  
    ('M49'),  
    ('M50'),
```



('M51'),  
('M52'),  
('M53'),  
('M54'),  
('M55'),  
('M56'),  
('M57'),  
('M58'),  
('M59'),  
('M60'),  
('M61'),  
('M62'),  
('M63'),  
('M64'),  
('M65'),  
('M66'),  
('M67'),  
('M68'),  
('M69'),  
('M70'),  
('M71'),  
('M72'),  
('M73'),  
('M74'),  
('M75'),  
('M76'),  
('M77'),  
('M78'),  
('M79'),  
('M80'),  
('M81'),  
('M82'),  
('M83'),  
('M84'),  
('M85'),  
('M86'),  
('M87'),  
('M88'),  
('M89'),  
('M90'),  
('M91'),  
('M92'),  
('M93'),  
('M94'),  
('M95'),  
('M96'),  
('M97'),  
('M98'),  
('M99'),  
('M100'),  
('M101'),  
('M102'),

```
('M103'),
('M104'),
('M105'),
('M106'),
('M107'),
('M108'),
('M109'),
('M110'),
('M111'),
('M112'),
('M113'),
('M114'),
('M115'),
('M116'),
('M117'),
('M118'),
('M119'),
('M120'),
('M121'),
('M122'),
('M123'),
('M124'),
('M125'),
('M126'),
('M127'),
('M128'),
)
```

```
fields = ['name', 'category', 'amount', 'shelflifeday', 'option', 'foldid', 'module', 'section', 'row',
'place']
```

```
widgets = {
    "name": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите наименование товара'
    }),
    "category": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите категорию'
    }, choices=category),
    "amount": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите количество (шт.)'
    }),
    "shelflifeday": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите срок хранения (дней)'
    }),
    "option": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите описание (необязательно)'
    }),
}
```

```

"foldid": Select(attrs={
    'class': 'form-control',
    'placeholder': 'Выберите склад'
}), choices=folds),
"module": Select(attrs={
    'class': 'form-control',
    'placeholder': 'Выберите модуль склада'
}), choices=moduls),
"section": Select(attrs={
    'class': 'form-control',
    'placeholder': 'Выберите модуль склада'
}), choices=sections),
"row": Select(attrs={
    'class': 'form-control',
    'placeholder': 'Выберите модуль склада'
}), choices=rows),
"place": Select(attrs={
    'class': 'form-control',
    'placeholder': 'Выберите модуль склада'
}), choices=places),
}

```

```

def clean_name(self):
    new_name = self.cleaned_data['name']

    if new_name == 'create':
        raise ValidationError("Запись о складе с таким именем не может быть создана!")
    if ItemNewOne.objects.filter(name__iexact=new_name).count():
        raise ValidationError(
            'Наименование товара должно быть уникальным. В записях уже существует
товар "{ }"'.format(
                new_name))

    return new_name

```

```

def clean_amount(self):
    new_amount = self.cleaned_data['amount']

    if new_amount <= 0:
        raise ValidationError("Запись о количестве товара должна быть целым
неотрицательным числом!")

    return new_amount

```

```

def clean_shelflifeday(self):
    new_days = self.cleaned_data['shelflifeday']

    if new_days <= 0:
        raise ValidationError("Запись о количестве дней хранения должна быть целым
неотрицательным числом!")

```

```
return new_days
```

```
classNewItemUpdateForm(forms.ModelForm):
```

```
class Meta:
```

```
    model = ItemNewOne
```

```
    folds = FoldNewOne.objects.all()
```

```
    category = (  
        ('Мебель'),  
        ('Стройматериалы'),  
        ('Инструменты'),  
        ('Техника'),  
    )
```

```
    moduls = (  
        ('А'),  
        ('Б'),  
        ('В'),  
        ('Г'),  
        ('Д'),  
        ('Е'),  
        ('Ж'),  
        ('З'),  
    )
```

```
    sections = (  
        ('S-1'),  
        ('S-2'),  
        ('S-3'),  
        ('S-4'),  
        ('S-5'),  
        ('S-6'),  
        ('S-7'),  
        ('S-8'),  
        ('S-9'),  
        ('S-10'),  
        ('S-11'),  
        ('S-12'),  
        ('S-13'),  
        ('S-14'),  
        ('S-15'),  
        ('S-16'),  
    )
```

```
    rows = (  
        ('1'),  
        ('2'),  
        ('3'),  
    )
```

```
('4'),  
('5'),  
('6'),  
('7'),  
('8'),  
('9'),  
('10'),  
('11'),  
('12'),  
('13'),  
('14'),  
('15'),  
('16'),  
('17'),  
('18'),  
('19'),  
('20'),  
('21'),  
('22'),  
('23'),  
('24'),  
('25'),  
('26'),  
('27'),  
('28'),  
('29'),  
('30'),  
('31'),  
('32'),  
)
```

```
places = (  
    ('M0'),  
    ('M1'),  
    ('M2'),  
    ('M3'),  
    ('M4'),  
    ('M5'),  
    ('M6'),  
    ('M7'),  
    ('M8'),  
    ('M9'),  
    ('M10'),  
    ('M11'),  
    ('M12'),  
    ('M13'),  
    ('M14'),  
    ('M15'),  
    ('M16'),  
    ('M17'),  
    ('M18'),  
    ('M19'),
```

('M20'),  
('M21'),  
('M22'),  
('M23'),  
('M24'),  
('M25'),  
('M26'),  
('M27'),  
('M28'),  
('M29'),  
('M30'),  
('M31'),  
('M32'),  
('M33'),  
('M34'),  
('M35'),  
('M36'),  
('M37'),  
('M38'),  
('M39'),  
('M40'),  
('M41'),  
('M42'),  
('M43'),  
('M44'),  
('M45'),  
('M46'),  
('M47'),  
('M48'),  
('M49'),  
('M50'),  
('M51'),  
('M52'),  
('M53'),  
('M54'),  
('M55'),  
('M56'),  
('M57'),  
('M58'),  
('M59'),  
('M60'),  
('M61'),  
('M62'),  
('M63'),  
('M64'),  
('M65'),  
('M66'),  
('M67'),  
('M68'),  
('M69'),  
('M70'),  
('M71'),

('M72'),  
('M73'),  
('M74'),  
('M75'),  
('M76'),  
('M77'),  
('M78'),  
('M79'),  
('M80'),  
('M81'),  
('M82'),  
('M83'),  
('M84'),  
('M85'),  
('M86'),  
('M87'),  
('M88'),  
('M89'),  
('M90'),  
('M91'),  
('M92'),  
('M93'),  
('M94'),  
('M95'),  
('M96'),  
('M97'),  
('M98'),  
('M99'),  
('M100'),  
('M101'),  
('M102'),  
('M103'),  
('M104'),  
('M105'),  
('M106'),  
('M107'),  
('M108'),  
('M109'),  
('M110'),  
('M111'),  
('M112'),  
('M113'),  
('M114'),  
('M115'),  
('M116'),  
('M117'),  
('M118'),  
('M119'),  
('M120'),  
('M121'),  
('M122'),  
('M123'),

```

('M124'),
('M125'),
('M126'),
('M127'),
('M128'),
)

fields = ['name', 'category', 'amount', 'shelflifeday', 'option', 'foldid', 'module', 'section',
          'row', 'place']
widgets = {
    "name": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите наименование товара'
    }),
    "category": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите категорию'
    }, choices=category),
    "amount": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите количество (шт.)'
    }),
    "shelflifeday": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите срок хранения (дней)'
    }),
    "option": TextInput(attrs={
        'class': 'form-control',
        'placeholder': 'Введите описание (необязательно)'
    }),
    "foldid": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите склад'
    }, choices=folds),
    "module": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите модуль склада'
    }, choices=moduls),
    "section": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите модуль склада'
    }, choices=sections),
    "row": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите модуль склада'
    }, choices=rows),
    "place": Select(attrs={
        'class': 'form-control',
        'placeholder': 'Выберите модуль склада'
    }, choices=places),
}

```



```

def clean_amount(self):
    new_amount = self.cleaned_data['amount']

    if new_amount <= 0:
        raise ValidationError('Запись о количестве товара должна быть целым
неотрицательным числом!')

    return new_amount

def clean_shelflifeday(self):
    new_days = self.cleaned_data['shelflifeday']

    if new_days <= 0:
        raise ValidationError('Запись о количестве дней хранения должна быть целым
неотрицательным числом!')

    return new_days

```

## Models.py

```

from django.db import models
from django.shortcuts import reverse

class FoldNewOne(models.Model):
    name = models.CharField('Название склада', max_length=50)
    parknumber = models.IntegerField('Номер платформы')
    responsible = models.CharField('Ответственный', max_length=50, blank=True)
    adress = models.CharField('Адрес', max_length=200, blank=True)
    square = models.IntegerField('Площадь склада')

    def __str__(self):
        return str(self.name)

    def get_item_count(self):
        count = 0
        items = ItemNewOne.objects.all()
        for i in items:
            if i.foldid.id == self.id:
                count += 1
        return count

    def get_id(self):
        return self.id

    def get_absolute_url(self):
        return f'/folds/{self.id}'

```

```

"""def get_update_url(self):
    return reverse('fold_update_url',kwargs={'id': self.id})
"""

class Meta:
    verbose_name = 'Склад'
    verbose_name_plural = 'Склады'

class ItemNewOne(models.Model):
    ITEM_CATEGORY = (
        ('Мебель', 'Мебель'),
        ('Стройматериалы', 'Строительные материалы'),
        ('Инструменты', 'Инструменты для выполнения строительных,отделочных работ'),
        ('Техника', 'Технические средства, машины для строительных, отделочных работ'),
    )

    ITEM_MODULE = (
        ('А', 'А'),
        ('Б', 'Б'),
        ('В', 'В'),
        ('Г', 'Г'),
        ('Д', 'Д'),
        ('Е', 'Е'),
        ('Ж', 'Ж'),
        ('З', 'З'),
    )

    ITEM_SECTION =(
        ('S-1', 'S-1'),
        ('S-2', 'S-2'),
        ('S-3', 'S-3'),
        ('S-4', 'S-4'),
        ('S-5', 'S-5'),
        ('S-6', 'S-6'),
        ('S-7', 'S-7'),
        ('S-8', 'S-8'),
        ('S-9', 'S-9'),
        ('S-10', 'S-10'),
        ('S-11', 'S-11'),
        ('S-12', 'S-12'),
        ('S-13', 'S-13'),
        ('S-14', 'S-14'),
        ('S-15', 'S-15'),
        ('S-16', 'S-16'),
    )

    ITEM_ROW=(
        ('1', '1'),
        ('2', '2'),
        ('3', '3'),

```

```
( '4', '4'),  
( '5', '5'),  
( '6', '6'),  
( '7', '7'),  
( '8', '8'),  
( '9', '9'),  
( '10', '10'),  
( '11', '11'),  
( '12', '12'),  
( '13', '13'),  
( '14', '14'),  
( '15', '15'),  
( '16', '16'),  
( '17', '17'),  
( '18', '18'),  
( '19', '19'),  
( '20', '20'),  
( '21', '21'),  
( '22', '22'),  
( '23', '23'),  
( '24', '24'),  
( '25', '25'),  
( '26', '26'),  
( '27', '27'),  
( '28', '28'),  
( '29', '29'),  
( '30', '30'),  
( '31', '31'),  
( '32', '32'),  
  
)  
  
ITEM_PLACE = (  
    ('M1', 'M1'),  
    ('M2', 'M2'),  
    ('M3', 'M3'),  
    ('M4', 'M4'),  
    ('M5', 'M5'),  
    ('M6', 'M6'),  
    ('M7', 'M7'),  
    ('M8', 'M8'),  
    ('M9', 'M9'),  
    ('M10', 'M10'),  
    ('M11', 'M11'),  
    ('M12', 'M12'),  
    ('M13', 'M13'),  
    ('M14', 'M14'),  
    ('M15', 'M15'),  
    ('M16', 'M16'),  
    ('M17', 'M17'),  
    ('M18', 'M18'),  
    ('M19', 'M19'),
```

('M20', 'M20'),  
('M21', 'M21'),  
('M22', 'M22'),  
('M23', 'M23'),  
('M24', 'M24'),  
('M25', 'M25'),  
('M26', 'M26'),  
('M27', 'M27'),  
('M28', 'M28'),  
('M29', 'M29'),  
('M30', 'M30'),  
('M31', 'M31'),  
('M32', 'M32'),  
('M33', 'M33'),  
('M34', 'M34'),  
('M35', 'M35'),  
('M36', 'M36'),  
('M37', 'M37'),  
('M38', 'M38'),  
('M39', 'M39'),  
('M40', 'M40'),  
('M41', 'M41'),  
('M42', 'M42'),  
('M43', 'M43'),  
('M44', 'M44'),  
('M45', 'M45'),  
('M46', 'M46'),  
('M47', 'M47'),  
('M48', 'M48'),  
('M49', 'M49'),  
('M50', 'M50'),  
('M51', 'M51'),  
('M52', 'M52'),  
('M53', 'M53'),  
('M54', 'M54'),  
('M55', 'M55'),  
('M56', 'M56'),  
('M57', 'M57'),  
('M58', 'M58'),  
('M59', 'M59'),  
('M60', 'M60'),  
('M61', 'M61'),  
('M62', 'M62'),  
('M63', 'M63'),  
('M64', 'M64'),  
('M65', 'M65'),  
('M66', 'M66'),  
('M67', 'M67'),  
('M68', 'M68'),  
('M69', 'M69'),  
('M70', 'M70'),  
('M71', 'M71'),

('M72', 'M72'),  
('M73', 'M73'),  
('M74', 'M74'),  
('M75', 'M75'),  
('M76', 'M76'),  
('M77', 'M77'),  
('M78', 'M78'),  
('M79', 'M79'),  
('M80', 'M80'),  
('M81', 'M81'),  
('M82', 'M82'),  
('M83', 'M83'),  
('M84', 'M84'),  
('M85', 'M85'),  
('M86', 'M86'),  
('M87', 'M87'),  
('M88', 'M88'),  
('M89', 'M89'),  
('M90', 'M90'),  
('M91', 'M91'),  
('M92', 'M92'),  
('M93', 'M93'),  
('M94', 'M94'),  
('M95', 'M95'),  
('M96', 'M96'),  
('M97', 'M97'),  
('M98', 'M98'),  
('M99', 'M99'),  
('M100', 'M100'),  
('M101', 'M101'),  
('M102', 'M102'),  
('M103', 'M103'),  
('M104', 'M104'),  
('M105', 'M105'),  
('M106', 'M106'),  
('M107', 'M107'),  
('M108', 'M108'),  
('M109', 'M109'),  
('M110', 'M110'),  
('M111', 'M111'),  
('M112', 'M112'),  
('M113', 'M113'),  
('M114', 'M114'),  
('M115', 'M115'),  
('M116', 'M116'),  
('M117', 'M117'),  
('M118', 'M118'),  
('M119', 'M119'),  
('M120', 'M120'),  
('M121', 'M121'),  
('M122', 'M122'),  
('M123', 'M123'),

```

        ('M124', 'M124'),
        ('M125', 'M125'),
        ('M126', 'M126'),
        ('M127', 'M127'),
        ('M128', 'M128'),
    )

    name = models.CharField('Наименование товара', max_length=100)
    category = models.CharField('Категория', max_length=150, choices=ITEM_CATEGORY)
    amount = models.IntegerField('Количество')
    shelflifeday = models.IntegerField('Дней хранения')
    option = models.TextField('Описание', blank=True)
    foldid = models.ForeignKey(FoldNewOne, on_delete=models.CASCADE,
verbose_name="Склад", blank=True")
    camedate = models.DateField('Дата поставки', blank=True, null=True)
    module = models.CharField('Модуль склада', max_length=5, blank=True,
choices=ITEM_MODULE)
    section = models.CharField('Секция модуля склада', max_length=5, blank=True,
choices=ITEM_SECTION)
    row = models.CharField('Ряд секции', blank=True, max_length=5, choices=ITEM_ROW)
    place = models.CharField('Место в ряду', blank=True,
max_length=5, choices=ITEM_PLACE)

    def __str__(self):
        return 'Наименование товара: ' + str(self.name) + '; Категория: ' + str(self.category)

    def get_id(self):
        return self.foldid.id

    def get_name(self):
        return self.name

    def get_absolute_url(self):
        return f'/items/{self.id}'

    class Meta:
        verbose_name = 'Товар'
        verbose_name_plural = 'Товары'

```

## Views.py

```

from django.shortcuts import render, redirect
from .models import *
from .forms import *
from django.views.generic import DetailView, UpdateView, DeleteView, View
from django.http import Http404, HttpResponse, HttpResponseRedirect
from django.shortcuts import render
import os

```

```
def index(request):
    return render(request, 'main/master.html')
```

```
def instruction(request):
    return render(request, 'main/instruction.html')
```

```
def first(request):
    get_items = ItemNewOne.objects.all()
    get_folds = FoldNewOne.objects.all()

    return render(request, 'main/first.html', {'title':"Сводный отчет",
                                                'get_item': get_items,'get_fold': get_folds})
```

```
def second(request):
    items = ItemNewOne.objects.all()
    return render(request, 'main/second.html', {'title':"Вторая страница сайта", 'item': items})
```

```
def item_list(request):
    items = ItemNewOne.objects.all()
    return render(request, 'main/item_full_list.html', {'title':"Информация о товарах", 'item':
items})
```

```
def fold_list(request):
    folds = FoldNewOne.objects.all()
    return render(request, 'main/fold_full_list.html', {'title':"Информация о складах", 'fold':
folds})
```

```
def js_diagram(request):
    str1 = "
    mebcoun = 0
    stroicoun = 0
    instrcoun = 0
    techcoun = 0
    dict1 = {}
    tmpdict = {}
    lst1 = []
    get_items = ItemNewOne.objects.all()
    get_folds = FoldNewOne.objects.all()
    for i in get_folds:
        if (i.get_item_count()) > 0:
            dict1.update({"label":i.name})
            print("Для склада:", i.name, i.id)
            for e in get_items:
```

```

print(e.get_id(), i.id)
if (e.get_id()) == i.id:
    if e.category == "Мебель":
        mebcoun+= e.amount
    elif e.category == "Стройматериалы":
        stroicount += e.amount
    elif e.category == "Инструменты":
        instrcount += e.amount
    elif e.category == "Техника":
        techcount += e.amount
    else:
        pass

dict1.update({"valueMeb": mebcoun})
dict1.update({"valueStroi": stroicount})
dict1.update({"valueInstr": instrcount})
dict1.update({"valueTech": techcount})
lst1.append(get_dict(dict1))
mebcoun = 0
stroicount = 0
instrcount = 0
techcount = 0
print("\nМассив - ", lst1, "\n")
return render(request, 'main/js-diagramm.html',{ 'title':"Сводный отчет",
                                                    'get_item': get_items,'get_fold': get_folds, 'datas':lst1 })

```

```

def get_dict(dict2):
    new_dict = { }
    for key,val in dict2.items():
        new_dict.update({key:val})
    print("\nВ функции по получению нового словаря - ", new_dict, "\n")
    return new_dict

```

```

""" print(tmpdict)
    for key, value in dict1.items():
        tmpdict.update({key:value})
    print(tmpdict)
    lst1.append(tmpdict)"""
"""module_dir = os.path.dirname(__file__) # get current directory
file_path = os.path.join(module_dir, 'datas.txt')
with open(file_path, 'w') as out:
    for key,val in dict1.items():
        out.write("{}:{}".format(key,val))

with open(file_path) as inp:
    for i in inp.readlines():
        str1 = '{'+i+'}'
        print("Строка", str1)
        lst1.append(str1)"""

```



```

def third(request):
    #warehouses = Warehouse.objects.find() - найти объект по его айди order - сортировка
    warehouses = FoldNewOne.objects.all()
    return render(request, 'main/third.html', {'title': "Третья страница сайта", 'warehouses':
warehouses})

class FoldCreate(View):
    def get(self,request):
        form = NewFoldCreateForm()
        return render(request, 'main/new_fold_create.html',{'form' : form})

    def post(self, request):
        post_form = NewFoldCreateForm(request.POST)

        if post_form.is_valid():
            post_form.save()
            return redirect('/folds')
        else:
            return render(request, 'main/new_fold_create.html',{'form' : post_form})

class ItemCreate(View):
    def get(self,request):
        form =NewItemCreateForm()
        return render(request, 'main/new_item_create.html',{'form' : form})

    def post(self, request):
        post_form =NewItemCreateForm(request.POST)

        if post_form.is_valid():
            post_form.save()
            return redirect('/items')
        else:
            return render(request, 'main/new_item_create.html',{'form' : post_form})

def folds_home(request):
    warehouses = FoldNewOne.objects.all()
    return render(request, 'main/folds_home.html', {'title': "Склады", 'warehouses': warehouses})

def items_home(request):
    items = ItemNewOne.objects.all()
    return render(request, 'main/items_home.html', {'title': "Товары", 'item': items})

"""def createitem(request):
    error = "
    if request.method == 'POST':
        form = ItemCreateTestForm(request.POST)
        if form.is_valid():
            form.save()

```

```

        return redirect('/items')
    else:
        error = 'Данные на форме неверны!'

    form = ItemCreateTestForm()
    context = {
        'form': form,
        'error': error
    }
    return render(request, 'main/create_item.html', context)

def createfold(request):
    error = ""
    if request.method == 'POST':
        form = WarehouseCreateForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('/folds')
        else:
            error = 'Данные на форме неверны!'

    form = WarehouseCreateForm()
    context = {
        'form': form,
        'error': error
    }
    return render(request, 'main/create_fold.html', context)

"""
"""class FoldUpdate(View):
    def get(self,request):
        fold = FoldNew.objects.get(id__iexact=self.id)
        bound_form=NewFoldCreateForm(instance=fold)
        return render(request, 'main/update_new_fold.html', context={'form': bound_form, 'folds':
fold})
"""

class FoldDetailView(DetailView):
    model = FoldNewOne
    template_name = 'main/fold_detail.html'
    context_object_name = 'folds'

class FoldUpdateView(UpdateView):
    model = FoldNewOne
    template_name = 'main/update_new_fold.html'

# form_class = NewFoldCreateForm
form_class = NewFoldUpdateForm

```

```

class FoldDeleteView(DeleteView):
    model = FoldNewOne
    #form_class = NewFoldUpdateForm
    success_url = '/folds'
    template_name = 'main/delete_fold.html'

class ItemDetailView(DetailView):
    model = ItemNewOne
    template_name = 'main/item_detail.html'
    context_object_name = 'items'

class ItemUpdateView(UpdateView):
    model = ItemNewOne
    template_name = 'main/update_new_item.html'

    form_class =NewItemUpdateForm

class ItemDeleteView(DeleteView):
    model = ItemNewOne
    success_url = '/items'
    template_name = 'main/delete_new_item.html'

# form_class = WarehouseCreateForm

```

## Текст HTML

### master.html

```

{ % load static % }

<!DOCTYPE html>

<body lang="en">
<head>
    <meta charset="UTF-8">
    <title>Главная</title>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
    <link href="/docs/5.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"

```

```

crossorigin="anonymous">
  <link rel="stylesheet" href="{ % static 'main/css/main.css' % }">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
  <meta name="generator" content="Hugo 0.79.0">

  <link rel="canonical" href="https://getbootstrap.com/docs/5.0/examples/product/">

  <!-- Bootstrap core CSS -->
  <link href="/docs/5.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHYTfAAsrekwKmP1"
crossorigin="anonymous">

  <!-- Favicons -->
  <link rel="apple-touch-icon" href="/docs/5.0/assets/img/favicons/apple-touch-icon.png"
sizes="180x180">
  <link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-32x32.png" sizes="32x32"
type="image/png">
  <link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-16x16.png" sizes="16x16"
type="image/png">
  <link rel="manifest" href="/docs/5.0/assets/img/favicons/manifest.json">
  <link rel="mask-icon" href="/docs/5.0/assets/img/favicons/safari-pinned-tab.svg"
color="#7952b3">
  <link rel="icon" href="/docs/5.0/assets/img/favicons/favicon.ico">
  <meta name="theme-color" content="#7952b3">

</head>

<body>
<header class="d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg-dark
border-bottom shadow-sm">
  <nav class="my-3 my-md-0 me-md-3">
    <a class="p-3 text-white" href="{ % url 'home' % }">Главная</a>
    <a class="p-3 text-white" href="{ % url 'folds_home' % }">Склады</a>
    <a class="p-3 text-white" href="{ % url 'items_home' % }">Товары</a>
    <a class="p-3 text-white" href="{ % url 'instruction' % }">Инструкция</a>
    <a class="p-3 text-white" href="{ % url 'fir' % }">Сводный отчет</a>
    <a class="p-3 text-white" href="{ % url 'js' % }">Диаграмма</a>

  </nav>
</header>

<main>

<div class="position-relative overflow-hidden p-3 p-md-5 m-md-3 text-center bg-light">
  <div class="col-md-10 p-lg-5 mx-auto my-5">
    <h1 class="display-4 fw-normal">Если вы пользуетесь впервые,</h1>
    <h1 class="display-4 fw-normal"> прочитайте инструкцию</h1>
    <a class="btn btn-outline-success" href="{ % url 'instruction' % }">Читать

```

```

инструкцию</a>
</div>
<div class="product-device shadow-sm d-none d-md-block"></div>
<div class="product-device product-device-2 shadow-sm d-none d-md-block"></div>
</div>

<div class="d-md-flex flex-md-equal w-500 my-md-3 ps-md-3">
  <div style="width:50%" class="bg-dark me-md-3 pt-3 px-3 pt-md-5 px-md-5 text-center text-white overflow-hidden">
    <div class="my-3 py-3">
      <h2 class="display-5">Склады</h2>
      <p class="lead">Управление данными и администрирование записей складов.</p>
      <a href="{ % url 'folds_home' % }" class="btn btn-info">Перейти</a>
    </div>
  </div>
  <div style="width:50%" class="bg-light me-md-3 pt-3 px-3 pt-md-5 px-md-5 text-center overflow-hidden">
    <div class="my-3 p-3">
      <h2 class="display-5">Товары</h2>
      <p class="lead">Создание, редактирование и другая работа с данными о товарах.</p>
      <a href="{ % url 'items_home' % }" class="btn btn-outline-info">Перейти</a>
    </div>
  </div>
</div>
<div class="d-md-flex flex-md-equal w-500 my-md-3 ps-md-3">
  <div style="width:100%" class="bg-light-dark me-md-3 pt-3 px-3 pt-md-5 px-md-5 text-center text-dark overflow-hidden">
    <div class="my-3 py-3">
      <h2 class="display-5">Сводный отчет о складах и товарах</h2>
      <p class="lead">Выводит HTML-страницу, содержащую связанные данные из двух моделей (склады и товары).</p>
      <a href="{ % url 'fir' % }" class="btn btn-primary">Перейти</a>
    </div>
  </div>
</div>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</main>
</body>

```

```
{ % load static % }
```

```
<!DOCTYPE html>
```

```
<body lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <title>Диаграмма</title>
```

```
    <link href="{ % static 'main/css/c3.css' % }" rel="stylesheet">
```

```
    <script src="{ % static 'main/js/d3-5.8.2.min.js' % }" charset="utf-8"></script>
```

```
    <script src="{ % static 'main/js/c3.min.js' % }"></script>
```

```
    <script src="{ % static 'main/js/c3.js' % }"></script>
```

```
    <script src="//ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
```

```
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script>
```

```
    <link href="/docs/5.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
```

```
    <link rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
```

```
integrity="sha384-
```

```
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
```

```
crossorigin="anonymous">
```

```
    <link rel="stylesheet" href="{ % static 'main/css/main.css' % }">
```

```
    <meta charset="utf-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <meta name="description" content="">
```

```
    <meta name="author" content="Mark Otto, Jacob Thornton, and Bootstrap contributors">
```

```
    <meta name="generator" content="Hugo 0.79.0">
```

```
    <link rel="canonical" href="https://getbootstrap.com/docs/5.0/examples/product/">
```

```
    <!-- Bootstrap core CSS -->
```

```
    <link href="/docs/5.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
```

```
    <!-- Favicons -->
```

```
    <link rel="apple-touch-icon" href="/docs/5.0/assets/img/favicons/apple-touch-icon.png"
sizes="180x180">
```

```
    <link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-32x32.png" sizes="32x32"
type="image/png">
```

```
    <link rel="icon" href="/docs/5.0/assets/img/favicons/favicon-16x16.png" sizes="16x16"
type="image/png">
```

```
    <link rel="manifest" href="/docs/5.0/assets/img/favicons/manifest.json">
```

```
    <link rel="mask-icon" href="/docs/5.0/assets/img/favicons/safari-pinned-tab.svg"
color="#7952b3">
```

```
    <link rel="icon" href="/docs/5.0/assets/img/favicons/favicon.ico">
```

```
    <meta name="theme-color" content="#7952b3">
```

```
</head>
```

```
<body>
```

```
<header class="d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg-dark
```

```
border-bottom shadow-sm">
```

```
<nav class="my-3 my-md-0 me-md-3">
```

```
<a class="p-3 text-white" href="{ % url 'home' % }">Главная</a>
```

```
<a class="p-3 text-white" href="{ % url 'folds_home' % }">Склады</a>
```

```
<a class="p-3 text-white" href="{ % url 'items_home' % }">Товары</a>
```

```
<a class="p-3 text-white" href="{ % url 'instruction' % }">Инструкция</a>
```

```
<a class="p-3 text-white" href="{ % url 'fir' % }">Сводный отчет</a>
```

```
<a class="p-3 text-white" href="{ % url 'js' % }">Диаграмма</a>
```

```
</nav>
```

```
</header>
```

```
<fieldset>
```

```
<div id="chart">
```

```
<script >
```

```
var data = { { datas|safe } }
```

```
var convertedData = [];
```

```
data.forEach(function(item){
```

```
    convertedData.push([item.label, item.valueMeb, item.valueStroi, item.valueInstr,  
item.valueTech]);
```

```
});
```

```
var chart = c3.generate({
```

```
bindto: '#chart',
```

```
data: {
```

```
    columns: convertedData,
```

```
    type: 'bar'
```

```
},
```

```
zoom: {
```

```
    enabled: true
```

```
},
```

```
axis: {
```

```
    x: {
```

```
        label: 'Категории товаров',
```

```
        type: 'category',
```

```
        categories: ['Мебель', 'Стройматериалы', 'Инструменты', 'Техника']
```

```
    },
```

```
    y: {
```

```
        label: 'Количество товаров'
```

```
    }
```

```
},
```

```
legend: {
```

```
    show:true
```

```
}
```

```
});
```

```

        setTimeout(function () {
            chart.transform('line');
        }, 3000);

        setTimeout(function () {
            chart.transform('bar');
        }, 5000);

    </script>
</div>
</fieldset>

<footer class="footer" style="margin-bottom:10px">
    <p align = "middle" class = "my-0 me-md-auto fw-normal p-2 text-white bg-dark align-items-center"><a href="{ % url 'home' % }"><font color="white"><big>Главная</big></font></a></p>
</footer>

</body>

```

## Base\_template.html

```

{ % load static % }

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{ % block title % } { % endblock % } </title>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
    <link href="/docs/5.0/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJf6kkoqNq00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet" href="{ % static 'main/css/main.css' % }">
</head>

<body>

<header class="d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg-dark border-bottom shadow-sm">
    <nav class="my-3 my-md-0 me-md-3">
        <a class="p-3 text-white" href="{ % url 'home' % }">Главная</a>
        <a class="p-3 text-white" href="{ % url 'folds_home' % }">Склады</a>
        <a class="p-3 text-white" href="{ % url 'items_home' % }">Товары</a>
        <a class="p-3 text-white" href="{ % url 'instruction' % }">Инструкция</a>
        <a class="p-3 text-white" href="{ % url 'fir' % }">Сводный отчет</a>
        <a class="p-3 text-white" href="{ % url 'js' % }">Диаграмма</a>
    
```



```

</nav>
</header>

<div class="content">
  <div class="content-inside">
    {% block content %}{% endblock %}
  </div>
</div>
{% block tomain %}{% endblock %}

</body>
</html>

```

## Item\_detail.html

```

{% extends 'main/base_template.html' %}
<!--получим данные из словаря по файлу views-->
{% block title %}
Подробнее о товаре {{ items.name }}
{% endblock %}

{% block content %}

  <h1>{{ items.name }}</h1>
  <h2>Дополнительная информация:</h2>
  <p>Категория: {{ items.category }}</p>
  <p>Количество: {{ items.amount }}</p>
  <p>Дней хранения: {% if items.shelflifeday != "" %} {{ items.shelflifeday }} {% else %}
Данные не указаны {% endif %}</p>
  <p>Описание: {% if items.option != "" %} {{ items.option }} {% else %} Данные не указаны
{% endif %}</p>
  <p>Склад: {{ items.foldid }}</p>
  <p>Модуль склада: {% if items.module != "" %} {{ items.module }} {% else %} Данные не
указаны {% endif %}</p>
  <p>Секция модуля склада: {% if items.section != "" %} {{ items.section }} {% else %}
Данные не указаны {% endif %}</p>
  <p>Ряд: {% if items.row != "" %} {{ items.row }} {% else %} Данные не указаны {% endif
%}</p>
  <p>Место: {% if items.place != "" %} {{ items.place }} {% else %} Данные не указаны {%
endif %}</p>

  <a href="{% url 'item-delete' items.id %}" class="btn btn-danger">Удалить</a>
  <a href="{% url 'item-update' items.id %}" class="btn btn-info">Редактировать</a>

  <a href="{% url 'items_home'%}" class="btn btn-warning">Назад</a>

{% endblock %}

{% block tomain %}
<footer class="footer">
  <p align="middle" class="my-0 me-md-auto fw-normal p-2 text-white bg-dark align-items-
center"><a href="{% url 'home' %}"><font

```

```

color="white"><big>Главная</big></font></a></p>
</footer>
{% endblock %}

```

## Item\_full\_list.html

```

{% extends 'main/base_template.html' %}
<!--получим данные из словаря по файлу views-->
{% block title %}
{{ title }}
{% endblock %}

{% block content %}
<input style="width:250px" class="btn btn-warning" value="Вернуться к главной странице
товаров" onClick="javascript:history.go(-1);"></input>

<hr>
<h3>Доступная информация о товарах:</h3>
<fieldset>
{% if item %}
<table class="table table-striped">
<thead>
<tr>
<th scope="col">Наименование товара</th>
<th scope="col">Категория</th>
<th scope="col">Количество(шт.)</th>
<th scope="col">Дней хранения</th>
<th scope="col">Описание</th>
<th scope="col">Склад</th>
<th scope="col">Модуль склада</th>
<th scope="col">Секция модуля склада</th>
<th scope="col">Ряд</th>
<th scope="col">Место</th>
</tr>
</thead>
<tbody>
{% for element in item %}
<tr>
<th scope="row">{{ element.name }}</th>
<td>{{ element.category }}</td>
<td>{{ element.amount }}</td>
<td>{{ element.shelflifeday }}</td>
<td>{% if element.option != "" %} {{ element.option }} {% else %} Данные не
указаны {% endif %}</td>
<td>{{ element.foldid }}</td>
<td>{% if element.module != "" %} {{ element.module }} {% else %} Данные
не указаны {% endif %}</td>
<td>{% if element.section != "" %} {{ element.section }} {% else %} Данные
не указаны {% endif %}</td>
<td>{% if element.row != "" %} {{ element.row }} {% else %} Данные не
указаны {% endif %}</td>
<td>{% if element.place != "" %} {{ element.place }} {% else %} Данные не

```

```

указаны {% endif % }</td>
        <!--<td><a href="{ % url 'item-detail' element.id % }" class="btn btn-
warning">Подробнее</a></td>-->
    </tr>
    {% endfor % }

</tbody>
</table>
{% else % }
    <p>Данные отсутствуют!</p>
    {% endif % }
</fieldset>
{% endblock % }

{% block tomain % }
<footer class="footer">
    <p align = "middle" class = "my-0 me-md-auto fw-normal p-2 text-white bg-dark align-items-
center"><a href="{ % url 'home' % }"><font
color="white"><big>Главная</big></font></a></p>
</footer>
{% endblock % }

```

## New\_item\_create.html

```

{% extends 'main/base_template.html' % }

{% block title % }
Создание записи о товаре
{% endblock % }

{% block content % }
    <h1>Создать запись о складе</h1>
    <form action="{ % url 'item_create_url' % }" method="post">
        {% csrf_token % }
        {% for field in form % }
            <div class="form-group">
                <p>{{ field.label }}: {{ field }}</p>
                {% if field.errors % }
                    <div class="alert alert-danger">
                        {{ field.errors }}
                    </div>
                {% endif % }
            </div>
        {% endfor % }

        <button type="submit" class="btn btn-success" >Добавить</button>
    <!-- {% if field.errors % }
        <div class="alert alert-danger">
            <span>{{ error }}</span>
        </div>
    {% endif % }

```

```

-->
    <input class="btn btn-warning" value="Назад" onClick="javascript:history.go(-
1);"></input>
    <p style="color: white">

    </p>
</form>
{% endblock %}

{% block tomain %}
<footer class="footer">
    <p align = "middle" class = "my-0 me-md-auto fw-normal p-2 text-white bg-dark align-items-
center"><a href="{% url 'home' %}"><font
color="white"><big>Главная</big></font></a></p>
</footer>
{% endblock %}

```

### Update\_new\_item.html

```

{% extends 'main/base_template.html' %}

{% block title %}
ОБНОВЛЕНИЕ записи о товаре
{% endblock %}

{% block content %}
<h1>Обновить запись о складе</h1>
<!--action="{% url 'item_create_url' %}"-->
<form method="post">
    {% csrf_token %}
    {% for field in form %}
        <div class="form-group">
            <p>{{ field.label }}: {{ field }}</p>
            {% if field.errors %}
                <div class="alert alert-danger">
                    {{ field.errors }}
                </div>
            {% endif %}
        </div>
    {% endfor %}

    <button type="submit" class="btn btn-success">Изменить</button>
<!-- {% if field.errors %}
    <div class="alert alert-danger">
        <span>{{ error }}</span>
    </div>
    {% endif %}

-->
    <input class="btn btn-warning" value="Назад" onClick="javascript:history.go(-
1);"></input>

```

```

<p style="color: white">

    </p>
</form>
{% endblock % }

{% block tomain % }
<footer class="footer">
    <p align = "middle" class = "my-0 me-md-auto fw-normal p-2 text-white bg-dark align-items-center"><a href="{ % url 'home' % }"><font color="white"><big>Главная</big></font></a></p>
</footer>
{% endblock % }

```

## First.html

```

{% extends 'main/base_template.html' % }

{% block title % }
{{ title }}
{% endblock % }

{% block content % }
{% if get_fold % }
    {% if get_item % }
        {% for fold_elem in get_fold % }
            <h2>Склад: {{ fold_elem.name }}</h2>
            {% if fold_elem.get_item_count > 0 % }
                <br>
                <table class="table table-striped table-bordered">
                    <thead>
                        <tr>
                            <th scope="col"><p class = "colwi1">Наименование товара</p></th>
                            <th scope="col"><p class = "colwi1">Категория</p></th>
                            <th scope="col"><p class = "colwi1">Количество (шт.)</p></th>
                        </tr>
                    </thead>
                    <tbody>
                        {% for item_elem in get_item % }
                            {% if item_elem.foldid.id == fold_elem.id % }
                                <tr>
                                    <td><p class = "colwi">{{ item_elem.name }}</p></td>
                                    <td><p class = "colwi">{{ item_elem.category }}</p></td>
                                    <td><p class = "colwi">{{ item_elem.amount }}</p></td>
                                </tr>
                            {% endif % }
                        {% endfor % }
                    </tbody>
                </table>
            {% else % }
                <p style="margin-left:100px; font-size:150%; font-color:#930"><b>Нет данных о

```

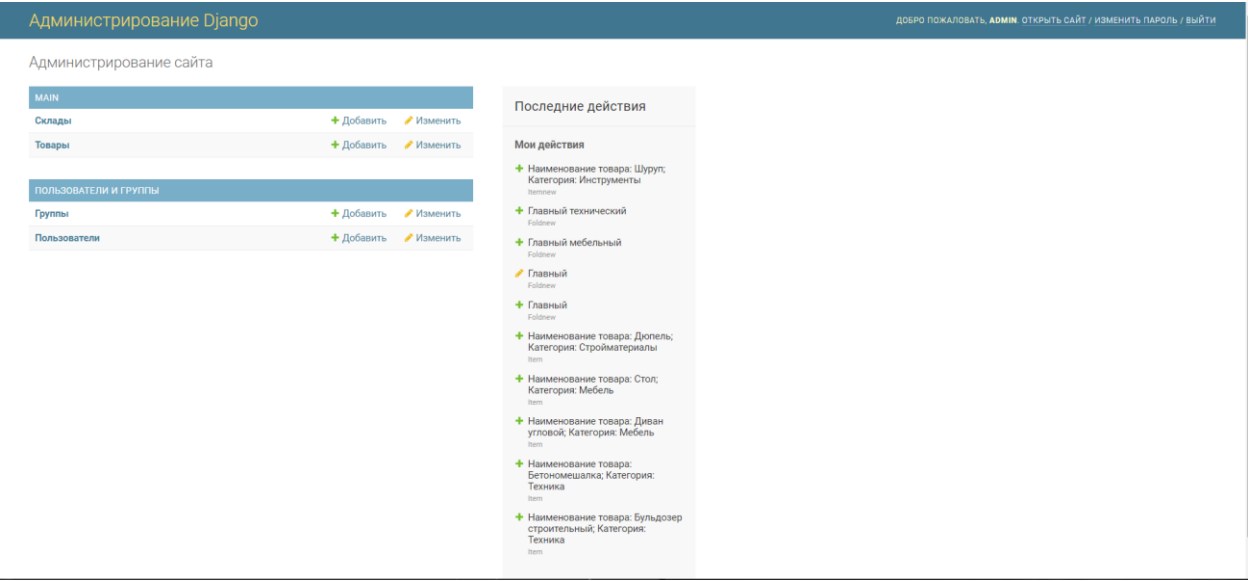
```
товарах.</b></p>
    { % endif % }
</br>
<hr>
    { % endfor % }
    { % endif % }
{ % endif % }

{ % endblock % }

{ % block tomain % }
<div style = "margin-top: -10px;height: 10px;">
    <p align = "middle" class = "my-0 me-md-auto fw-normal p-2 text-white bg-dark align-items-
center"><a href="{ % url 'home' % }"><font
color="white"><big>Главная</big></font></a></p>
    </div>
{ % endblock % }
```

Экранные формы

Админ-страница



Главная страница

[Главная](#)[Склады](#)[Товары](#)[Инструкция](#)[Сводный отчет](#)[Диаграмма](#)

Если вы пользуетесь впервые,  
прочитайте инструкцию

Читайте инструкцию

**Склады**  
Управление данными и администрирование записей складов.  
Перейти

**Товары**  
Создание, редактирование и другая работа с данными о товарах.  
Перейти

**Сводный отчет о складах и товарах**  
Выводит HTML-страницу, содержащую связанные данные из двух моделей (склады и товары).  
Перейти

## Склады

[Главная](#)[Склады](#)[Товары](#)[Инструкция](#)[Сводный отчет](#)[Диаграмма](#)

Создать запись о новом складе

Создать

Отчет с полной информацией о складах доступен по кнопке:

Открыть отчет

Доступная информация о складах:

Название склада	
Мебельный	Подробнее
Главный	Подробнее
Инструментный	Подробнее
Распределенный	Подробнее

## Товары

[Главная](#)[Склады](#)[Товары](#)[Инструкция](#)[Сводный отчет](#)[Диаграмма](#)

Создать запись о новом товаре

Создать

Отчет с полной информацией о товарах доступен по кнопке:

Открыть отчет

Доступная информация о товарах:

Наименование товара	Категория	
Диван	Мебель	Подробнее
Шуруп	Стройматериалы	Подробнее
Гвоздь	Стройматериалы	Подробнее
Кровать 2-х спальная	Мебель	Подробнее
Шуруповерт	Инструменты	Подробнее
Молот строительный	Инструменты	Подробнее
Обои бежевые	Стройматериалы	Подробнее
Обои белые	Стройматериалы	Подробнее
Стул домашний	Мебель	Подробнее
Журнальный столик	Мебель	Подробнее

## Инструкция

Следуя простым шагам, вы сможете быстро освоится в данном веб-приложении

Инструкция содержит описание полей и требований, предъявляемых к корректности вводимых данных, описание возможностей, схем и механизмов переходов по веб-страницам.

Главная страница

На странице реализованы возможности:

- Переход к главной странице модели "Склады";
- Переход к главной странице модели "Товары";
- Вывод сводного отчета, содержащего связанные данные из двух моделей в формате: склад и ниже - список товаров и их некоторых характеристик (для каждого склада).

Склады

На главной странице реализованы возможности:

- Создание записи о новом складе;
- Переход к отчету, который содержит полные данные о складах;
- Просмотр списка складов с возможностью перехода к дополнительной информации о выбранном элементе.

Правила ввода данных

Поле	Пояснение (назначение)	Тип данных	Правила ввода
Название склада	Предназначено для указания названия склада	Текст	Значение должно быть уникальным
Номер платформы	Предназначено для указания номера платформы разгрузки товаров на склад	Целое число	Значение должно быть неотрицательным числом
СМО (сводный отчет)	Относносвязан с таблицей СМО (сводный отчет)	Текст	Наибольшее значение

Сводный отчет

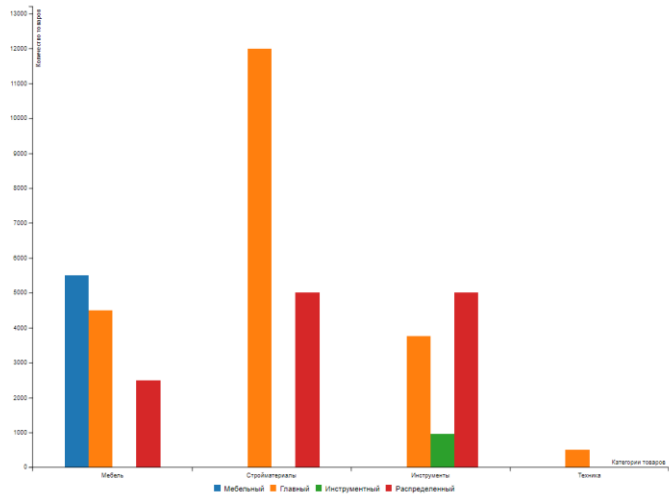
Склад: Мебельный

Наименование товара	Категория	Количество (шт.)
Диван	Мебель	1000
Кровать 2-х спальная	Мебель	4500

Склад: Главный

Наименование товара	Категория	Количество (шт.)
Шуруп	Стройматериалы	1000
Гвоздь	Стройматериалы	5000
Обои белые	Стройматериалы	6000
Журнальный столик	Мебель	4500
Каток строительный	Техника	500
Отвертка крестовая 5x100 мм	Инструменты	3750

Диаграмма





# Страница детализированного просмотра

ГлавнаяСкладыТоварыИнструкцияСводный отчетДиаграмма

Диван

Дополнительная информация:

Категория: Мебель

Количество: 1000

Дней хранения: 1000

Описание: Данные не указаны

Склад: Мебельный

Модуль склада: А

Секция модуля склада: S-1

Ряд: 1

Место: М1

Удалить

Редактировать

Назад

# Страница удаления объекта

ГлавнаяСкладыТоварыИнструкцияСводный отчетДиаграмма

Удаление записи о товаре.

☒ Я подтверждаю, что хочу удалить запись.

Удалить

Назад

# Страница редактирования объекта

## Обновить запись о складе

Наименование товара:	<input type="text" value="Диван"/>
Категория:	<input type="text" value="Мебель"/>
Количество:	<input type="text" value="1000"/>
Дней хранения:	<input type="text" value="1000"/>
Описание:	<input type="text" value="Введите описание (необязательно)"/>
Склад:	<input type="text" value="Мебельный"/>
Модуль склада:	<input type="text" value="А"/>
Секция модуля склада:	<input type="text" value="S-1"/>
Ряд секции:	<input type="text" value="1"/>
Место в ряду:	<input type="text" value="M1"/>
<div><div>Изменить</div><div>Назад</div></div>	