

Содержание

1 Уровень 1: Основы шаблонов и Bootstrap	1
1.1 Задача 1.1: Визитная карточка	1
1.2 Задача 1.2: Прогноз погоды	1
1.3 Задача 1.3: Динамическое меню	2
2 Уровень 2: Формы и обработка данных	3
2.1 Задача 2.1: Контактная форма	3
2.2 Задача 2.2: Форма загрузки файлов	3
3 Уровень 3: Комплексные проекты	3
3.1 Задача 3.1: Интернет-магазин (с корзиной)	3
3.1.1 Страница каталога:	3
3.1.2 Корзина:	4
3.1.3 Форма оформления:	4

1 Уровень 1: Основы шаблонов и Bootstrap

1.1 Задача 1.1: Визитная карточка

Цель: Создать одностраничный сайт-визитку с использованием Bootstrap.

Требования:

- 1) Главная страница `index.html`
- 2) Используйте Bootstrap-сетку для размещения:
 - Имя (заголовок `h1`)
 - Фото или `placeholder` изображение
 - Краткое описание
 - Список навыков (используйте цикл `{% for %}`)

1.2 Задача 1.2: Прогноз погоды

Цель задачи

Создать простое веб-приложение на Flask, которое показывает прогноз погоды для разных городов.

Требования к выполнению

1. Создайте базовую структуру приложения

- Файл `app.py` с основным приложением Flask
- Папка `templates` для шаблонов HTML
- Папка `static` для статических файлов (CSS, изображения)

2. Реализуйте главную страницу (`index.html`)

- Заголовок "Прогноз погоды"
- Список городов: Москва, Санкт-Петербург, Казань, Екатеринбург, Новосибирск
- Каждый город должен быть ссылкой на страницу с прогнозом

3. Создайте маршруты

- `/` - главная страница со списком городов
- `/weather/<city>` - страница прогноза для конкретного города

4. Данные о погоде (хранить в словаре в `app.py`)

```
1 weather_data = {
2     'москва': {'temp': '+15 С', 'description': 'Облачно', 'humidity': '75%'},
3     'санкт-петербург': {'temp': '+12 С', 'description': 'Дождь', 'humidity': '85%'},
4     'казань': {'temp': '+18 С', 'description': 'Солнечно', 'humidity': '60%'},
5     'екатеринбург': {'temp': '+10 С', 'description': 'Пасмурно', 'humidity': '80%'},
6     'новосибирск': {'temp': '+8 С', 'description': 'Снег', 'humidity': '90%'},
7 }
```

1.3 Задача 1.3: Динамическое меню

Цель: Реализовать динамическое формирование навигационной панели на основе данных из Python.

Требования:

- 1) В `app.py` создайте список словарей для меню:

```
1 menu = [
2     {'url': '/', 'title': 'Main', 'active': False},
3     {'url': '/about', 'title': 'About me', 'active': False},
4     {'url': '/projects', 'title': 'Projects', 'active': True},
5     {'url': '/contact', 'title': 'Contacts', 'active': False},
6 ]
```

- 2) Передайте этот список в каждый шаблон через `render_template`

2 Уровень 2: Формы и обработка данных

2.1 Задача 2.1: Контактная форма

Цель: Создать страницу с контактной формой, используя Bootstrap стили.

Требования:

- 1) Создайте страницу `/contact` с методами GET и POST
- 2) Реализуйте шаблон `contact.html` с формой, содержащей:
 - Имя (текстовое поле, обязательное)
 - Email (поле типа email, обязательное)
 - Тема сообщения (выпадающий список)
 - Сообщение
 - Кнопку отправки
- 3) Используйте Bootstrap классы для стилизации формы

2.2 Задача 2.2: Форма загрузки файлов

Цель: Создать страницу для загрузки изображений.

Требования:

- 1) Форма должна поддерживать `multipart/form-data`
- 2) Поля:
 - Выбор файла (только изображения)
 - Описание изображения
 - Категория (чекбоксы: "Пейзаж" "Портрет" "Натюрморт")
- 3) Реализуйте предпросмотр изображения перед отправкой
- 4) Ограничьте размер файла (максимум 5MB)

3 Уровень 3: Комплексные проекты

3.1 Задача 3.1: Интернет-магазин (с корзиной)

Цель: Создать простой интернет-магазин с функцией корзины.

Требования:

3.1.1 Страница каталога:

- Сетка товаров (карточки Bootstrap)
- Фильтры по категориям (чекбоксы)
- Сортировка (выпадающий список)
- Кнопка "В корзину" на каждой карточке

3.1.2 Корзина:

- Отдельная страница
- Список товаров с количеством
- Возможность изменить количество или удалить товар
- Итоговая сумма
- Форма оформления заказа

3.1.3 Форма оформления:

- Данные покупателя
- Выбор способа доставки
- Выбор способа оплаты
- Подтверждение заказа