# Лабораторная работа: Создание анимации в Django приложении

## Цель работы:

Научиться добавлять анимации в Django-приложение с использованием HTML, CSS, JavaScript и Django-шаблонов. Создать анимационные элементы, которые улучшают пользовательский интерфейс.

## Этапы выполнения работы

### 1. Подготовка окружения

1. Установите и настройте Django-проект:
 ```bash
 django-admin startproject animation\_project
 cd animation\_project
 python manage.py startapp animation\_app
 ```
2. Добавьте приложение `animation\_app` в `INSTALLED\_APPS` в файле `settings.py`.

3. Создайте папки для статики и шаблонов:
 ```bash
 mkdir -p animation\_app/static/animation\_app/css
 mkdir -p animation\_app/static/animation\_app/js
 mkdir -p animation\_app/templates/animation\_app
 ```

### 2. Настройка статики

1. Включите поддержку статики в `settings.py`:
 ```python
 STATIC\_URL = '/static/'
 STATICFILES\_DIRS = [BASE\_DIR / 'animation\_app/static']
 ```

2. Создайте базовый CSS-файл для анимаций:
 ```css
 body {
 font-family: Arial, sans-serif;
 background-color: #f0f0f0;
 margin: 0;
 padding: 0;
 display: flex;
 justify-content: center;
 align-items: center;
 height: 100vh;
 }

 .animated-box {
 width: 100px;
 height: 100px;
 background-color: #4caf50;
 animation: bounce 2s infinite;
 }

 @keyframes bounce {
 0%, 100% {
 transform: translateY(0);
 }
 50% {
 transform: translateY(-50px);
 }
 }
 ```

3. Подключите JavaScript-файл (опционально):
 ```javascript
 document.addEventListener('DOMContentLoaded', () => {
 const box = document.querySelector('.animated-box');
 box.addEventListener('click', () => {
 alert('Анимация работает!');
 });
 });
 ```

### 3. Создание HTML-шаблона

1. В папке `animation\_app/templates/animation\_app/` создайте файл `index.html`:
 ```html
 <!DOCTYPE html>
 <html lang="en">
 <head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Анимация в Django</title>
 <link rel="stylesheet" href="{% static 'animation\_app/css/styles.css' %}">
 </head>
 <body>
 <div class="animated-box"></div>
 <script src="{% static 'animation\_app/js/scripts.js' %}"></script>
 </body>
 </html>
 ```

### 4. Настройка маршрутов

1. Настройте `views.py` в приложении `animation\_app`:
 ```python
 from django.shortcuts import render

 def index(request):
 return render(request, 'animation\_app/index.html')
 ```

2. Создайте файл `urls.py` в `animation\_app` и добавьте маршруты:
 ```python
 from django.urls import path
 from .views import index

 urlpatterns = [
 path('', index, name='index'),
 ]
 ```

3. Подключите маршруты в `animation\_project/urls.py`:
 ```python
 from django.contrib import admin
 from django.urls import path, include

 urlpatterns = [
 path('admin/', admin.site.urls),
 path('', include('animation\_app.urls')),
 ]
 ```

### 5. Запуск и тестирование

1. Запустите сервер разработки:
 ```bash
 python manage.py runserver
 ```

2. Перейдите по адресу [http://127.0.0.1:8000/](http://127.0.0.1:8000/) и проверьте, отображается ли анимация.

### 6. Дополнительные функции

1. Добавьте больше анимаций. Например, создайте кнопки с анимацией при наведении:
 ```css
 .animated-button {
 padding: 10px 20px;
 background-color: #008cba;
 color: white;
 border: none;
 cursor: pointer;
 transition: transform 0.3s;
 }

 .animated-button:hover {
 transform: scale(1.1);
 }
 ```

2. Используйте JavaScript для динамического изменения стилей:
 ```javascript
 const newBox = document.createElement('div');
 newBox.className = 'animated-box';
 document.body.appendChild(newBox);
 ```

3. Добавьте Bootstrap или Tailwind CSS для более сложных анимаций.

### 7. Итог

После выполнения работы вы:
- Освоили настройку статики в Django.
- Добавили CSS-анимации.
- Настроили JavaScript для работы с анимациями.
- Создали динамичный интерфейс для улучшения пользовательского опыта.

### Дополнительное задание

1. Реализуйте анимацию с использованием библиотеки [Animate.css](https://animate.style/).
2. Добавьте AJAX для динамической загрузки анимаций.