# **20 Платформа разработки Веб-приложений ASP.NET**

Задание 1. ViewModel: TaskViewModel с описанием, сроком выполнения, статусом. Форма + валидация: форма добавления задачи, валидировать дату ([DataType(DataType.Date)]) и обязательность описания. Сервис: ITaskService с методами добавления, завершения и получения задач. DI: внедрение сервиса в контроллер. TempData/ViewBag: уведомление о завершении задачи через ViewBag.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

public class TaskService : ITaskService

{ private static List<TaskDto> \_tasks = new();

public void AddTask(TaskDto task)

{ task.Id = \_tasks.Count + 1;

task.IsCompleted = false;

\_tasks.Add(task); }

public void CompleteTask(int id)

{ var task = \_tasks.FirstOrDefault(t => t.Id == id);

if (task != null)

{ task.IsCompleted = true; } }

public void UncompleteTask(int id)

{ var task = \_tasks.FirstOrDefault(t => t.Id == id);

if (task != null)

{ task.IsCompleted = false; } }

public void DeleteTask(int id)

{ var task = \_tasks.FirstOrDefault(t => t.Id == id);

if (task != null)

{ \_tasks.Remove(task); } }

public List<TaskDto> GetTasks(string filter = "all")

{ return filter.ToLower() switch

{

"completed" => \_tasks.Where(t => t.IsCompleted).OrderBy(t => t.DueDate).ToList(),

"incomplete" => \_tasks.Where(t => !t.IsCompleted).OrderBy(t => t.DueDate).ToList(),

\_ => \_tasks.OrderBy(t => t.DueDate).ToList()

};

}

}

Анализ результатов:

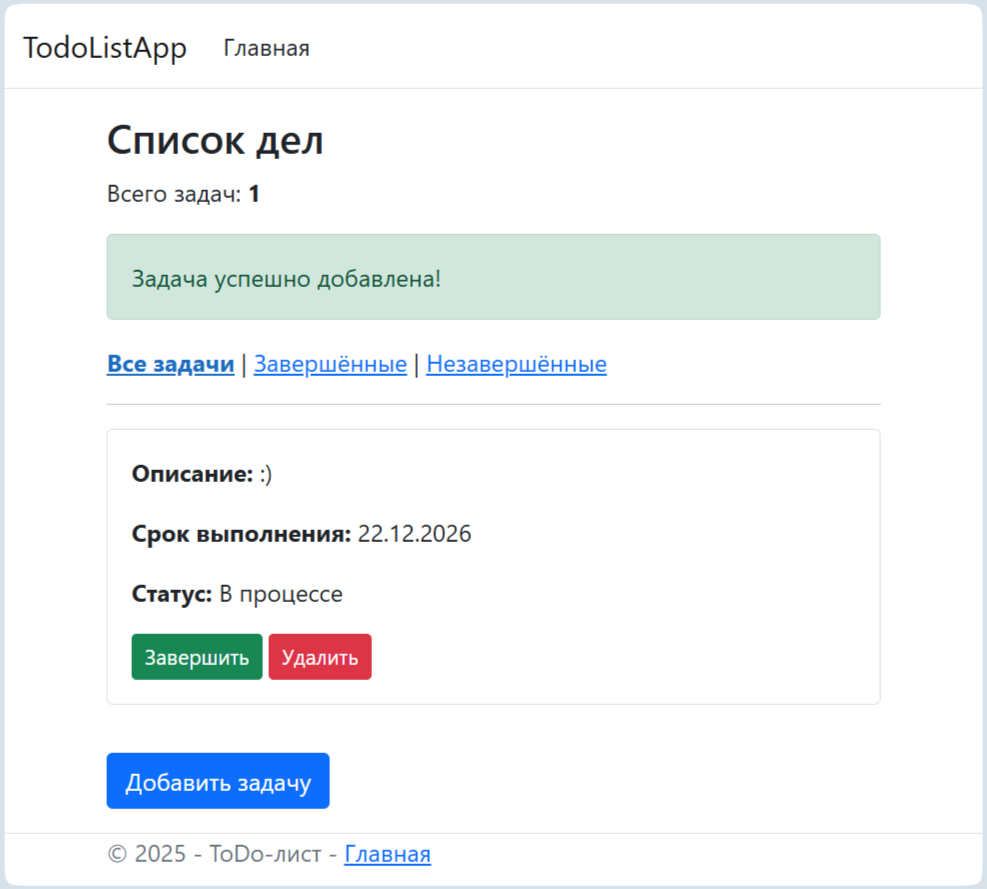


Рисунок 20.1 – Результат работы программы