**Тема: «Платформа разработки Веб-приложений ASP.NET»**

**Практическая часть:** Взаимодействие с данными и бизнес-логикой. Продолжаем разработку заданий из прошлого занятия.

**1. Каталог фильмов**

* **ViewModel**: создать MovieViewModel с названием, жанром, годом, рейтингом.
* **Форма + валидация**: форма добавления фильма, поля [Required], [Range], [StringLength].
* **Сервис**: реализовать IMovieService с методами добавления и получения фильмов.
* **DI**: внедрить MovieService в MovieController.
* **TempData/ViewBag**: сообщение об успешном добавлении фильма через TempData.

**2. Список дел (ToDo-лист)**

* **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.

**3. Библиотека книг**

* **ViewModel**: BookViewModel с названием, автором, ISBN, жанром.
* **Форма + валидация**: все поля обязательны, ISBN — [RegularExpression].
* **Сервис**: IBookService с методами получения списка, добавления.
* **DI**: внедрить сервис в BookController.
* **TempData/ViewBag**: сообщение о новой книге через TempData.

**4. Учёт личных расходов**

* **ViewModel**: ExpenseViewModel с категорией, суммой, датой.
* **Форма + валидация**: сумма с [Range(0.01, 1000000)], дата — обязательна.
* **Сервис**: IExpenseService с логикой подсчёта суммы за период.
* **DI**: внедрение через конструктор.
* **TempData/ViewBag**: всплывающее уведомление о добавлении расхода.

**5. Телефонный справочник**

* **ViewModel**: ContactViewModel с ФИО, телефоном, email.
* **Форма + валидация**: телефон и email — обязательные поля.
* **Сервис**: IContactService с методами поиска и добавления контактов.
* **DI**: внедрение в ContactController.
* **TempData/ViewBag**: вывод сообщения через ViewBag.

**6. Тренировочный журнал**

* **ViewModel**: WorkoutViewModel с типом упражнения, временем, датой.
* **Форма + валидация**: обязательные поля, проверка даты.
* **Сервис**: IWorkoutService с логикой фильтра по типу тренировки.
* **DI**: внедрение через WorkoutController.
* **TempData/ViewBag**: сообщение об успешном сохранении.

**7. Коллекция фильмов/сериалов для просмотра**

* **ViewModel**: WatchlistItemViewModel с названием, жанром, типом (фильм/сериал).
* **Форма + валидация**: [Required], валидация поля жанра.
* **Сервис**: IWatchlistService с добавлением и удалением записей.
* **DI**: внедрение в WatchlistController.
* **TempData/ViewBag**: TempData для уведомления "Добавлено в список".

**8. Список покупок**

* **ViewModel**: ShoppingItemViewModel с названием и количеством.
* **Форма + валидация**: [Required], [Range(1, 100)] для количества.
* **Сервис**: IShoppingService с методами добавления, удаления.
* **DI**: внедрение в ShoppingController.
* **TempData/ViewBag**: использовать ViewBag для сообщения "Товар удалён".

**9. Список рецептов**

* **ViewModel**: RecipeViewModel с названием, описанием, списком ингредиентов.
* **Форма + валидация**: все поля обязательны, ограничение по длине описания.
* **Сервис**: IRecipeService с логикой поиска по ингредиентам.
* **DI**: внедрение сервиса в контроллер.
* **TempData/ViewBag**: сообщение через TempData.

**10. Учёт домашних животных**

* **ViewModel**: PetViewModel с кличкой, типом животного, датой рождения.
* **Форма + валидация**: [Required], [DataType(DataType.Date)].
* **Сервис**: IPetService с логикой добавления и отображения по типу.
* **DI**: внедрение сервиса.
* **TempData/ViewBag**: ViewBag — сообщение "Питомец добавлен".

**11. Мини-блог**

* **ViewModel**: PostViewModel с заголовком, текстом, автором.
* **Форма + валидация**: [Required], [StringLength], [MinLength].
* **Сервис**: IPostService с логикой создания, удаления постов.
* **DI**: внедрение сервиса.
* **TempData/ViewBag**: TempData для показа "Пост опубликован".

**12. Список фильмов на вечер**

* **ViewModel**: EveningMovieViewModel с фильмом, жанром, временем начала.
* **Форма + валидация**: [Required], [DataType(DataType.Time)].
* **Сервис**: IEveningMovieService — логика фильтрации по жанру.
* **DI**: внедрение через EveningMovieController.
* **TempData/ViewBag**: сообщение через ViewBag.

**13. Галерея изображений**

* **ViewModel**: ImageViewModel с названием, URL, описанием.
* **Форма + валидация**: [Url], [Required], [StringLength].
* **Сервис**: IImageService — логика загрузки и отображения.
* **DI**: внедрение в ImageController.
* **TempData/ViewBag**: уведомление "Изображение добавлено".

**14. Планировщик учёбы**

* **ViewModel**: StudyTaskViewModel с темой, сроком сдачи, приоритетом.
* **Форма + валидация**: [Required], [DataType(DataType.Date)].
* **Сервис**: IStudyTaskService — логика фильтрации и приоритезации.
* **DI**: внедрение в StudyTaskController.
* **TempData/ViewBag**: TempData для сообщения.

**15. Книга отзывов**

* **ViewModel**: FeedbackViewModel с именем, текстом отзыва, рейтингом.
* **Форма + валидация**: [Required], [Range(1, 5)], [StringLength(500)].
* **Сервис**: IFeedbackService — сохранение и отображение отзывов.
* **DI**: внедрение в FeedbackController.
* **TempData/ViewBag**: использовать TempData для сообщения "Спасибо за отзыв".