# РУКОВОДСТВО ПРОГРАММИСТА

## 1. Введение

Настоящий документ содержит техническое описание системы MSE-MOOC, включая:

1. Архитектуру приложения.
2. Модели данных.
3. Контроллеры и API-эндпоинты.
4. Описание представлений (Views).
5. Инструкции по развертыванию.

## 2. Архитектура системы

Система разработана на основе ASP.NET Core MVC с использованием:

1. Backend: C#, Entity Framework Core, MS SQL Server.
2. Frontend: Razor Pages, Bootstrap 5, JavaScript
3. API: RESTful JSON API.

## 3. Модели данных

3.1. Основные сущности

1. Course (Курс)

public class Course

{

public int Id { get; set; }

[Required]

[MaxLength(200)]

public string Title { get; set; }

[Url]

[MaxLength(500)]

public string? Link { get; set; }

[MaxLength(2000)]

public string? Description { get; set; }

public bool IsSelfPassed { get; set; }

[Column(TypeName = "date")]

public DateOnly StartDate { get; set; }

[Column(TypeName = "date")]

public DateOnly EndDate { get; set; }

[Range(0, 1000000)]

public int Price { get; set; }

[MaxLength(50)]

public string? Language { get; set; }

public bool Certificated { get; set; }

public bool IsCertificatePaid { get; set; }

[Range(0, 1000)]

public int Credits { get; set; }

[Range(0, 5)]

public float Reviews { get; set; }

public virtual ICollection<User> Users { get; set; } = new List<User>();

public virtual ICollection<Discipline> Disciplines { get; set; } = new List<Discipline>();

public int? UniversityId { get; set; }

public virtual University? University { get; set; }

public virtual ICollection<Lecturer> Lecturers { get; set; } = new List<Lecturer>();

}

2. Discipline (Дисциплина)

public class Discipline

{

public int Id { get; set; }

[Required]

[MaxLength(200)]

public string Title { get; set; }

[Column(TypeName = "date")]

public DateOnly StartDate { get; set; }

[Column(TypeName = "date")]

public DateOnly EndDate { get; set; }

[Range(0, 100)]

public int Credits { get; set; }

public virtual ICollection<Course> Courses { get; set; } = new List<Course>();

}

3. Lecturer (Лектор)

public class Lecturer

{

public int Id { get; set; }

[Required]

[MaxLength(50)]

public string FirstName { get; set; }

[Required]

[MaxLength(50)]

public string LastName { get; set; }

public virtual ICollection<Course> Courses { get; set; } = new List<Course>();

}

4. User (Пользователь)

public class User

{

public int Id { get; set; }

[Required]

[MaxLength(50)]

[Column("login")]

public string Login { get; set; }

[Required]

[MaxLength(100)]

public string Password { get; set; }

[EmailAddress]

[MaxLength(100)]

public string? Email { get; set; }

[MaxLength(50)]

public string? FirstName { get; set; }

[MaxLength(50)]

public string? LastName { get; set; }

[Column(TypeName = "datetime2")]

public DateTime RegistrationDate { get; set; } = DateTime.UtcNow;

public bool IsActive { get; set; } = true;

[MaxLength(20)]

public string? PhoneNumber { get; set; }

public virtual ICollection<Course> Courses { get; set; } = new List<Course>();

}

5. University (университет)

public class University

{

public int Id { get; set; }

[Required]

[MaxLength(200)]

public string Name { get; set; }

public virtual ICollection<Course> Courses { get; set; } = new List<Course>();

}

## 4. Контроллеры и API

4.1. CourseController

|  |  |  |
| --- | --- | --- |
| **Метод** | **Эндпоинт** | **Описание** |
| GetAllCourses | GET /api/courses | Получить список всех курсов. |
| GetCourseById | GET /api/courses/{id} | Получить курс по ID. |
| CreateCourse | POST /api/courses | Добавить новый курс. |
| UpdateCourse | PUT /api/courses/{id} | Обновить данные курса. |
| DeleteCourse | DELETE /api/courses/{id} | Удалить курс. |
| FilterCourses | GET /api/courses/filter | Фильтрация курсов. |

4.2. UserController

|  |  |  |
| --- | --- | --- |
| **Метод** | **Эндпоинт** | **Описание** |
| Register | POST /api/users/register | Регистрация нового пользователя. |
| Login | POST /api/users/login | Аутентификация. |
| GetUserCourses | GET /api/users/{id}/courses | Получить курсы пользователя. |
| EnrollToCourse | POST /api/users/enroll | Записаться на курс. |

4.3. AdminController

|  |  |  |
| --- | --- | --- |
| Метод | Эндпоинт | Описание |
| GetAllUsers | GET /api/admin/users | Получить список пользователей. |
| BlockUser | PUT /api/admin/users/block | Заблокировать пользователя. |
| ImportCourses | POST /api/admin/import | Импорт курсов из внешнего API. |

## 5. Представления (Razor Pages)

5.1. Основные страницы

|  |  |  |
| --- | --- | --- |
| **Страница** | **Путь** | **Описание** |
| Index.cshtml | / | Главная страница с поиском курсов. |
| CourseDetails.cshtml | /course/{id} | Детальная информация о курсе. |
| MyCourses.cshtml | /mycourses | Личный кабинет пользователя. |
| AdminPanel.cshtml | /admin | Панель администратора. |

## 6. Развертывание

6.1. Требования

Сервер: Windows/Linux с .NET 6.

База данных: MS SQL Server 2019+.

6.2. Инструкция

Клонировать репозиторий:

<https://github.com/EdIInorojka/MOOC.git>

Настроить appsettings.json:

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=MOOCDB;Trusted\_Connection=True;"

}

}

Применить миграции:

dotnet ef database update

Запустить приложение:

dotnet run