Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Уфимский университет науки и технологий»

Кафедра ТК

Лабораторная работа № 2

**Тема: Создание RESTful-сервисов**

**Вариант 3. Работа с сервисной службой интернет-провайдера**

Выполнил:

студент группы ИВТ-ИПОАС-103М

Бикметов Р.Р.

Проверил: доцент

Насыров Р. В.

Уфа 2025

**Назначение программы:**

Программа представляет собой серверное приложение, реализующее REST API для системы заявок **сервисной службой интернет-провайдера**.

Сервис предназначен для управления пользователя, заявками и разграничения доступа между пользователями и администраторами.

Система поддерживает простую аутентификацию через логин\пароль и манипуляции с заявка (создание, просмотр, удаление, редактирование).

**Назначение модулей и структура программы:**

Основные файлы проекта:

* program.cs — основной файл подготовки сервиса с REST API.
* SupportInternetProveder.cs — файл с описанием методов API.

**Используемые технологии:**

|  |  |
| --- | --- |
| **Компонент** | **Назначение** |
| C# | язык программирования |
| ASP Net | создание REST API и маршрутизация запросов |
| WebAPI | асинхронный веб-сервер |
| Swagger UI | встроенный интерфейс тестирования API |

**Алгоритм работы программы:**

* 1. При запуске приложения создаются csv баз данных для пользователей и заявок, если их не было раньше
  2. При просмотре пользователя проверяется логин и пароль, а также роль этого пользователя. Роль задается вручную (для админа указывается ADMIN)
  3. При создании заявки требуется указать id пользователя, для которого создается заявка, а также комментарий к ней.
  4. Для удаления и редактирования требуется указывать уникальный id заявки в формате Guid.
  5. Для просмотра заявок требуется указывать id пользователя, чьи заявки просматриваются.
  6. При создании записи проверяется доступность врача на выбранные даты.
  7. Данные хранятся в соответствующих csv файлах

**Описание моделей базы данных:**

Модель user

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Описание** |
| id | int | Уникальный идентификатор |
| login | str | логин |
| password | str | Пароль |
| Role | str | Роль пользователя |

Модель Request (Заявка)

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип** | **Описание** |
| date | DateTime | Время создания заявки |
| idRequest | Guid | ID заявки |
| userId | int | ID пользователя |
| Comment | date | Комментарий к заявке |

**Основные функции программы:**

|  |  |
| --- | --- |
| **Функция** | **Краткое описание** |
| GetAllUsers() | Получение всех пользователей (только admin) |
| CreateUser () | Создание пользователя |
| DeleteUserRequest () | Удаление заявки |
| CreateUserRequest () | Создание заявки |
| ChangeCommentUserRequest () | Изменение заявки |
| GetUserRequest () | Просмотр заявки |

**Пользовательские роли:**

|  |  |
| --- | --- |
| **Роль** | **Возможности** |
| user | Создание пользователя, полный доступ к заявкам (создание, удаление, просмотр, изменение) |
| admin | Полный доступ: создание, просмотр всех записей пользователей, полный доступ к заявкам (создание, удаление, просмотр, изменение) |

**Результаты тестирования:**

* Все эндпоинты успешно протестированы через Swagger UI и консольный клиент.
* Реализованы проверки прав доступа и валидация дат.
* При попытке доступа без прав возвращается 403 Forbidden.
* Проверка пересечения дат записи к врачу работает корректно.

**Вывод:**

Разработана и протестирована системазаписи пациентов к врачу. Программа реализует REST API с разграничением прав доступа и может быть использована как основа для более сложной системы управления отелем.