Disynergy
Страница материала ↗
EXQT
Готовые ответы — на EXQT
Доступ к ответам приобретается отдельно.
Найти ответы ↗

Отчет о выполнении

ВВЕДЕНИЕ

Инструкция к шаблону	Текст в квадратных скобках замените своими данными. Блоки с инструкциями можно удалить из финального отчета. В местах для скриншотов вставьте реальные изображения работы приложения.

Напишите 5-8 предложений о цели практики, выбранной системе и стеке технологий.

Тема практики: разработка программного модуля на C# с применением ООП, архитектурных слоев, паттернов проектирования, JSON-хранилища, REST API и GUI/веб-интерфейса.

Таблица 1 – Паспорт практики

Параметр	**Значение**
ФИО студента	[ФИО]
Группа	[ГРУППА]
Название приложения	Система учета заявок пользователей
Язык программирования	C#
Хранение данных	JSON-файлы
API	ASP.NET Core Web API / REST API
Интерфейс	[Windows Forms / WPF / Web]

1. C# И ООП-МОДЕЛЬ

1.1 Описание предметной области

[Опишите систему учета заявок пользователей: кто создает заявки, какие данные содержит заявка, какие действия выполняет пользователь.]

1.2 ООП-модель заявки

Таблица 2 - ООП-модель заявки

Свойство	**Тип данных**	**Назначение**
Id	Guid	уникальный идентификатор заявки
Title	string	тема заявки
Description	string	описание проблемы
AuthorName	string	автор заявки
CreatedAt	DateTime	дата создания
Status	RequestStatus	статус заявки
Priority	string	приоритет заявки

1.3 Перечисление статусов

Код enum RequestStatus

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

1.4 Класс UserRequest

Код класса UserRequest

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

МЕСТО ДЛЯ СКРИНШОТА
Рисунок 1 — Запуск тестового кода создания заявки

Вывод по этапу 1

[Напишите, какая модель создана и какие принципы ООП применены.]

2. АРХИТЕКТУРА И ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ

2.1 Структура проекта

[Опишите структуру папок проекта: Models, Repositories, Services, Controllers, UI, Data.]

Структура проекта

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

2.2 Слои приложения

Таблица 3 – Слои приложения

Слой	Назначение	Классы/папки проекта
Presentation	интерфейс пользователя	[UI / Pages / Forms]
Business Logic	бизнес-логика	Services
Data Access	работа с данными	Repositories
Models	модели предметной области	Models

2.3 Используемые паттерны

Таблица 4 – Паттерны

Паттерн	Где используется	Зачем нужен
Repository	JsonRequestRepository	скрывает работу с JSON-файлом
Service Layer	RequestService	выносит бизнес-логику
MVC / MVVM	[интерфейс]	отделяет UI от логики
Singleton / DI	Program.cs	управляет зависимостями

2.4 Интерфейсы репозитория и сервиса

Код IRequestRepository

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

Код IRequestService

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

2.5 Реализация сервиса

Код RequestService

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

Вывод по этапу 2

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

[Опишите, как архитектура помогает разделить ответственность между классами.]

3. JSON-ХРАНИЛИЩЕ ДАННЫХ

3.1 Структура JSON

[Опишите, как заявка хранится в JSON: объект, массив объектов, поля.]

Пример файла requests.json

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

3.2 Сериализация и десериализация

[Кратко объясните, что такое сериализация и десериализация, какой пакет используется: System.Text.Json или Newtonsoft.Json.]

3.3 Репозиторий JSON

Код JsonRequestRepository

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

МЕСТО ДЛЯ СКРИНШОТА
Рисунок 2 — Файл requests.json после добавления заявок

Вывод по этапу 3

[Напишите, как реализовано сохранение данных и почему репозиторий отделен от сервиса.]

4. REST API

4.1 Описание REST API

[Опишите, что такое REST API и как интерфейс взаимодействует с сервером через JSON.]

4.2 Маршруты API

Таблица 5 - Маршруты

Метод	URL	Назначение
GET	/api/requests	получить все заявки
GET	/api/requests/{id}	получить заявку по ID
POST	/api/requests	создать заявку
PUT	/api/requests/{id}	обновить заявку
PUT	/api/requests/{id}/status	изменить статус
DELETE	/api/requests/{id}	удалить заявку

4.3 Настройка Program.cs

Код Program.cs

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

4.4 Контроллер RequestsController

Код RequestsController

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

МЕСТО ДЛЯ СКРИНШОТА

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Рисунок 3 — Проверка API в Swagger или Postman

Вывод по этапу 4

[Напишите, какие методы API реализованы и какие статус-коды используются.]

5. GUI ИЛИ ВЕБ-ИНТЕРФЕЙС

5.1 Выбор интерфейса

[Укажите выбранный вариант: Windows Forms, WPF или веб-интерфейс. Объясните выбор.]

5.2 Элементы интерфейса

Таблица 6 – Элементы интерфейса

Элемент	**Назначение**
Таблица заявок	показывает список заявок
Поля ввода	ввод темы, автора, описания
Кнопка Добавить	создает новую заявку
Кнопка Изменить	редактирует заявку
Кнопка Удалить	удаляет заявку
Выбор статуса	меняет статус заявки

5.3 Обращение к REST API из интерфейса

Код HttpClient или JavaScript fetch

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

МЕСТО ДЛЯ СКРИНШОТА
Рисунок 4 — Главное окно или веб-страница приложения

МЕСТО ДЛЯ СКРИНШОТА
Рисунок 5 — Добавление и отображение заявки через интерфейс

Вывод по этапу 5

[Напишите, как интерфейс связан с REST API и какие операции доступны пользователю.]

6. ИТОГОВАЯ СБОРКА ПРОЕКТА

6.1 Итоговая структура проекта

Итоговая структура папок проекта

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

6.2 Проверка работы приложения

Таблица 7 - Работоспособность

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru

Что проверить	Результат
Проект запускается без ошибок	[да / нет]
API возвращает список заявок	[да / нет]
Новая заявка добавляется	[да / нет]
Заявка редактируется	[да / нет]
Статус заявки меняется	[да / нет]
Заявка удаляется	[да / нет]
Данные сохраняются в JSON	[да / нет]
Интерфейс получает данные через API	[да / нет]

МЕСТО ДЛЯ СКРИНШОТА
Рисунок 6 — Итоговая проверка полного сценария работы приложения

6.3 README проекта

Содержание README.md

[ВСТАВЬТЕ ФРАГМЕНТ КОДА ЗДЕСЬ]

Вывод по этапу 6

[Напишите, что реализовано в итоговом приложении и какие технологии использованы.]

ЗАКЛЮЧЕНИЕ

[Напишите общий вывод: что было разработано, какие возможности есть у приложения, какие паттерны применены, как организовано хранение данных и обмен через REST API.]

ССЫЛКИ НА GITHUB-РЕПОЗИТОРИЙ

[Вставьте ссылку на репозиторий с проектом или укажите, где расположен архив проекта.]

Помогу с учебным заданием без предоплаты. Оплата после выполнения
Нажмите на мессенджер, чтобы написать
Telegram ↗ WhatsApp ↗ MAX ↗
+7 (924) 305-23-08 · Пишите в мессенджер
disynergy.ru