



# С# ASP.NET Core разработчик

# Меня хорошо видно & слышно?



# Защита проекта

## Тема: Платформа управления киберспортивными соревнованиями



Группа № 2  
CyberTech Team

# План защиты



Цели проекта

Что планировалось

Используемые  
технологии

Что получилось

Схемы/архитектура

Выводы

# Цели проекта

Разработка платформы для учета киберспортивных соревнований. Которая, включает в себя следующие задачи:

1. регистрация и авторизация пользователя
2. панель админа (для ведения справочников и ввода расписания матчей)
3. новостная лента
4. сервис генерации результатов прошедших матчей
5. хранение файлов (картинки/фото) в NoSQL БД
6. расчет рейтинга команд на основе очков, начисленных за прошедшие матчи
7. просмотр статистики каждой команды (сколько проведено матчей/ побед/ поражений, дата последнего и ближайшего матча)

# Что планировалось

Группа с разным уровнем знаний.

Выполнение проекта велось на протяжении всего курса с разной интенсивностью и различной активностью участников группы 😊



# Используемые технологии

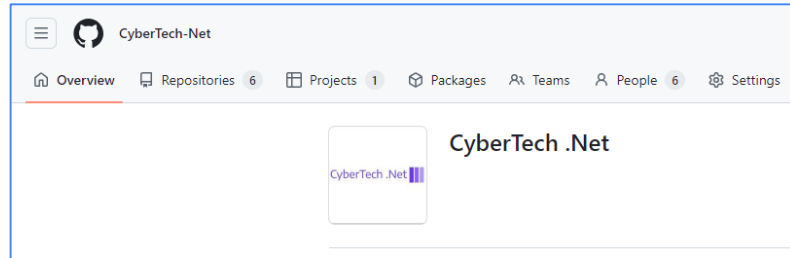
## Back-end (ASP .NET)

- MongoDB
- RabbitMQ
- PgSql
- Docker
- GitHub
- Redis
- Ocelot
- Prometheus
- Masstransit
- EF Core
- ASP.NET Core Identity
- Serilog
- xUnit
- FluentValidation
- SignalR

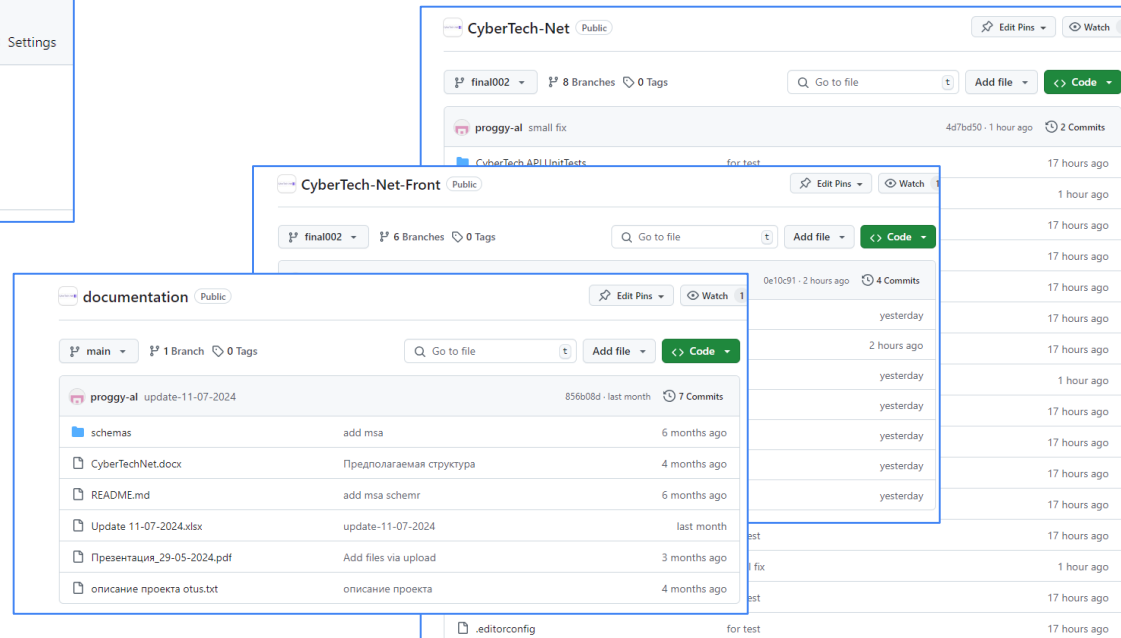
## Front-end (React)

- интерфейсный инструментариий antd и bootstrap
- пакет Redux Toolkit
- библиотеки dayjs, React Toastify, signalR и др.

# Что получилось

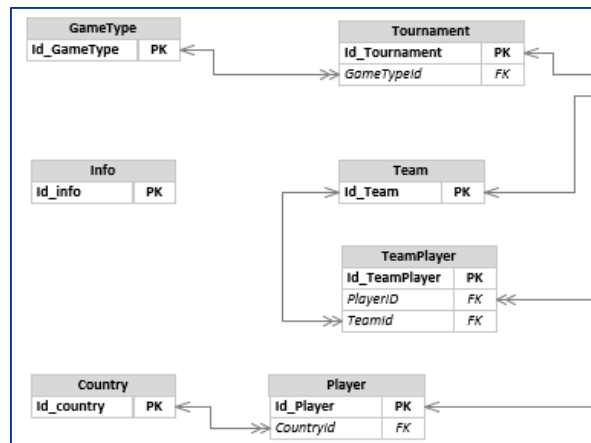


<https://github.com/CyberTech-Net>





# Схемы (БД)



IdInfo	Guid	PK
TitleInfo	string (50)	3a
TextInfo	string (500)	Te
DataInfo	date	Da

Tournament (Турнир)			
Id_Tournament	Guid	Primary Key	
GameTypeId	Guid	Foreign Key	
TitleTournament	string(150)	Название	
TypeTournament	string(20)	Тип турнира	местный, региональный, государственный
DataTournamentInit	date	Дата начала турнира	
DataTournamentEnd	date	Дата окончания турнира	
PlaceName	string(150)	Место проведения	
EarndTournament	double	Призовой фонд	DEFAULT = 0

Match (Игровые встречи турнира)			
Id_TournamentMeet	Guid	Primary Key	
TournamentId	Guid	Foreign Key	
FirstTeamId	Guid	Foreign Key	
SecondTeamId	Guid	Foreign Key	
startDateTime	datetime	Дата и время матча	

MatchResult (Результаты матча)			
Id_MatchResult	Guid	Primary Key	
MatchId	Guid	Foreign Key	
TeamId	Guid	Foreign Key	
Score	int	Набранные командой очки	DEFAULT = 0
isWin	bool	Победитель или проигравший	DEFAULT = false

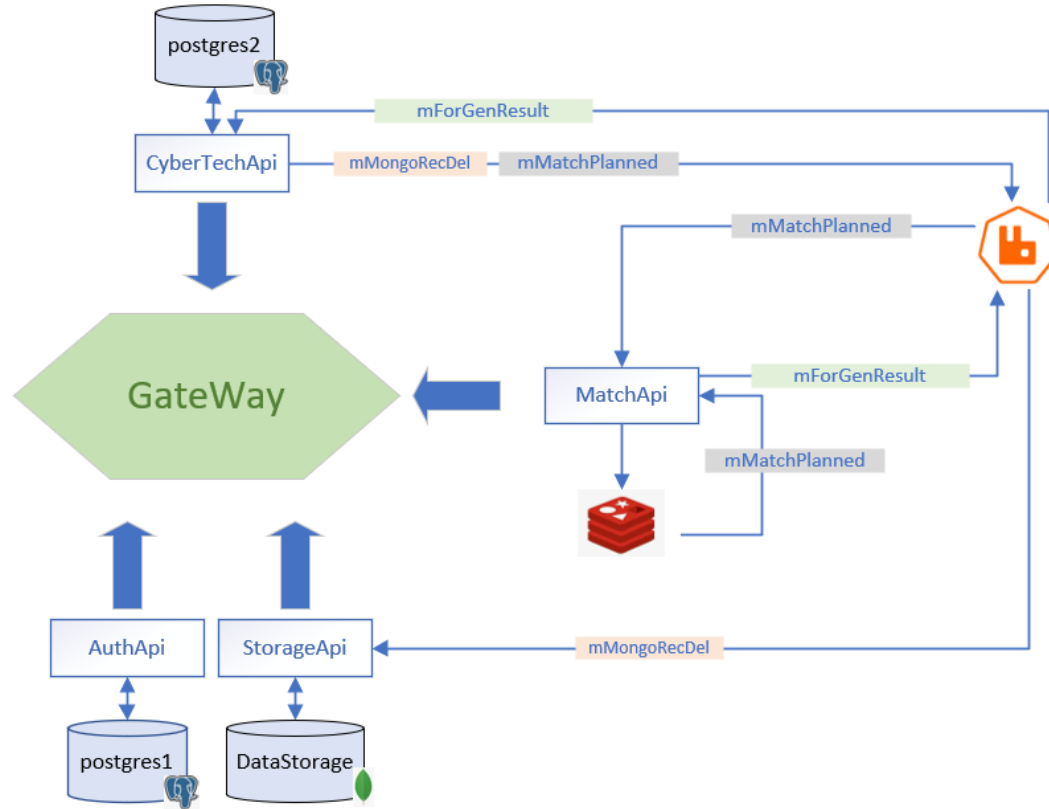
Картинка из MongoDB

Team (Команды)			
Id	Guid	Primary Key	
Name	string(50)	Название	
Picture	string(500)	Картинка из MongoDB	
RegistrationDate	date	Дата регистрации	DEFAULT = DateTime.Today

Player (Игроки)			
Id	Guid	Primary Key	
TeamId	Guid	Foreign Key	
Nick	string(10)	Ник	
FirstName	string(50)	Фамилия	
SecondName	string(50)	Имя	
BirthDate	date	Дата рождения	
MongoPlayerPic	int	Фото из MongoDB	

TeamPlayer (Состав команды)			
Id_TeamPlayer	Guid	Primary Key	
PlayerId	Guid	Foreign Key	
TeamId	Guid	Foreign Key	
Year1	int	Год вступления в команду	
Year2	int	Год выхода из команды	NULL

# Схемы (архитектура)



# Демонстрация работы приложения

RabbitMQ 3.13.7 Erlang 26.2.5.2

Overview **Connections** Channels Exchanges **Queues and Streams**

## Queues

All queues (3)

Pagination

Page 1 of 1 - Filter:  ☐ Regex ?

Virtual host	Name	Type	Features	State
/	masstransit_check_match_data	classic	D	running
/	masstransit_delete_mongo_record	classic	D	running
/	masstransit_gen_match_result	classic	D	running

Redis Desktop Manager 0.9.3.817

Подключиться к серверу Redis

localhost

db0 (1)

c342dda2-d385-4a92-b730-427f4a91c54a

STRING: 30-427f4a91c54a TTL: -1 Переименовать Удалить Перезагрузить значение Установить TTL

Value: size: 83.00 bytes View as: Plain Text

{"Id": "c342dda2-d385-4a92-b730-427f4a91c54a", "StartDateTime": "2024-08-31T14:00:00"}

Test Explorer

Ready

Test	Duration	Trace
CyberTech.API.UnitTests (4)	690 ms	
CyberTech.API.UnitTests.Controllers ..	690 ms	
TournamentControllerTests (4)	690 ms	
IfCreatingTournamentIsValid_Re...	12 ms	

My Queries Performance Databases

Search

DataStorage

fs.chunks

fs.files

admin

config

local

Documents 55 Aggregations Schema Indexes (2) Validation

Type a query: { field: 'value' }

ADD DATA EXPORT DATA UPDATE DELETE

\_id: ObjectId('66d055d7fb126c948f0c2097')  
files\_id: ObjectId('66d055d7fb126c948f0c2096')  
n: 0  
data: Binary.createFromBase64('iVBORw0KGgoAAAANSUHuEugAAABgAAAAA')

\_id: ObjectId('66d055d7fb126c948f0c2097')  
files\_id: ObjectId('66d055d7fb126c948f0c2098')  
n: 0  
data: Binary.createFromBase64('iVBORw0KGgoAAAANSUHuEugAAABgAAAAA')

postgres2

Databases (2)

postgres

postgres2

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (3)

handbooks

public

tournaments

http\_requests\_received\_total

Table Graph

Evaluation time

http\_requests\_received\_total(action="GetAllAsync", code="200", controller="GameType", endpoint="api/GameType", instance="cybertech-administration-api", method="GET")

http\_requests\_received\_total(action="GetAllAsync", code="200", controller="Info", endpoint="api/info", instance="cybertech-administration-api", method="GET")

http\_requests\_received\_total(code="200", endpoint="/messageHub/negotiate", instance="cybertech-administration-api", method="POST")

http\_requests\_received\_total(code="101", endpoint="/messageHub", instance="cybertech-administration-api", method="POST")

http\_requests\_received\_total(action="GetAllAsync", code="200", controller="Team", endpoint="api/Team", instance="cybertech-administration-api", method="GET")

http\_requests\_received\_total(action="GetAllAsync", code="200", controller="MatchCard", endpoint="api/MatchCard", instance="cybertech-administration-api", method="GET")

http\_requests\_received\_total(action="GetAllAsync", code="200", controller="Tournament", endpoint="api/Tournament", instance="cybertech-administration-api", method="GET")

http\_requests\_received\_total(action="GetAllAsync", code="200", controller="Match", endpoint="api/Match", instance="cybertech-administration-api8080", job="cybertech-administration-api", method="GET")

http\_requests\_received\_total(action="AddAsync", code="200", controller="Match", endpoint="api/Match", instance="cybertech-administration-api8080", job="cybertech-administration-api", method="POST")

http\_requests\_received\_total(action="DeleteAsync", code="200", controller="Match", endpoint="api/Match/{idguid}", instance="cybertech-administration-api8080", job="cybertech-administration-api", method="DELETE")

# Выводы и планы по развитию

1. Закрепить полученные знания, занимаясь разработкой приложений (Back-end и Front-end)
2. Изучать новые инструменты, используя доступные каналы информации Интернета
- 3.

# Спасибо за внимание!