Проектный офис ИКТИБ

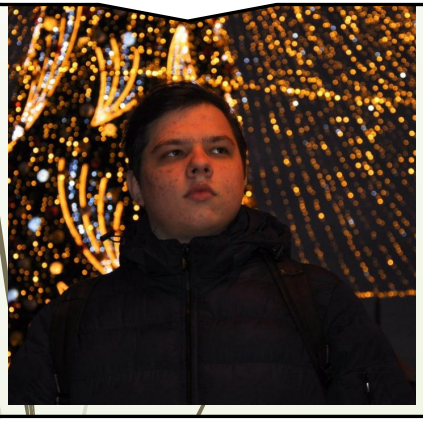
ИКТИБ ЮФУ
Введение в инженерную деятельность
Создание NFT-маркетплейса
Gel.ru

Доцент кафедры ИБТКС к.т.н., А.П. Плёнкин

Проектный офис ИКТИБ

Наша команда

Тришкин Игорь
Капитан, разработчик **backend**



Илья Башмашников
Разработчик **frontend**

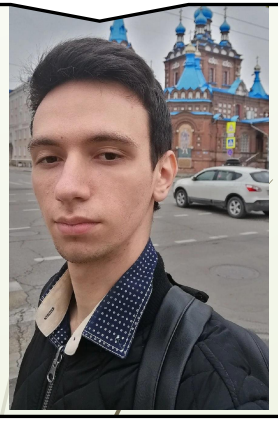
Сергей Гараев
Разработчик **frontend**

Авакимов Константин
Разработчик **frontend**

Проектный офис ИКТИБ

Наша команда

Шевченко Лев
Разработчик **backend**



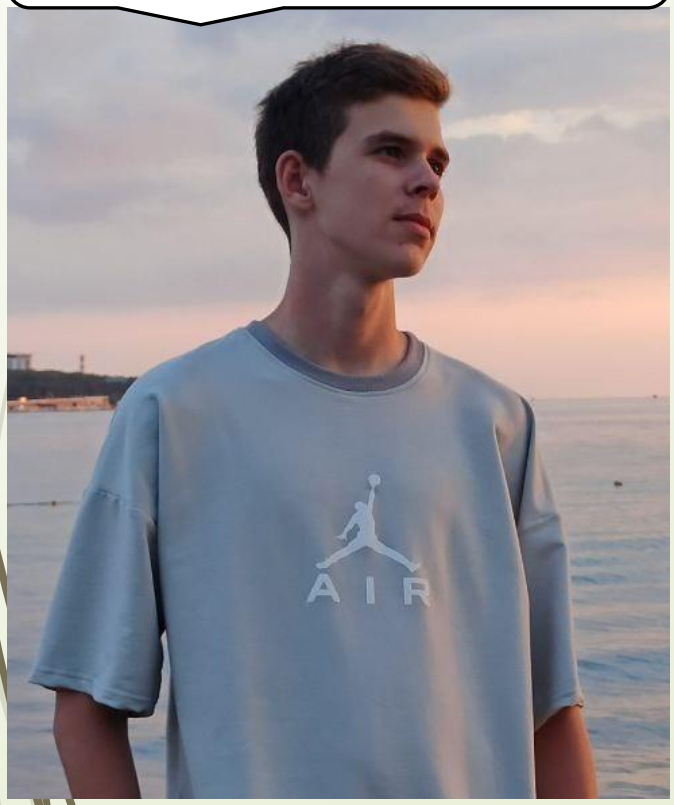
Бугримов Игорь
Разработчик **frontend**

Балабан Иван
Разработчик **frontend**

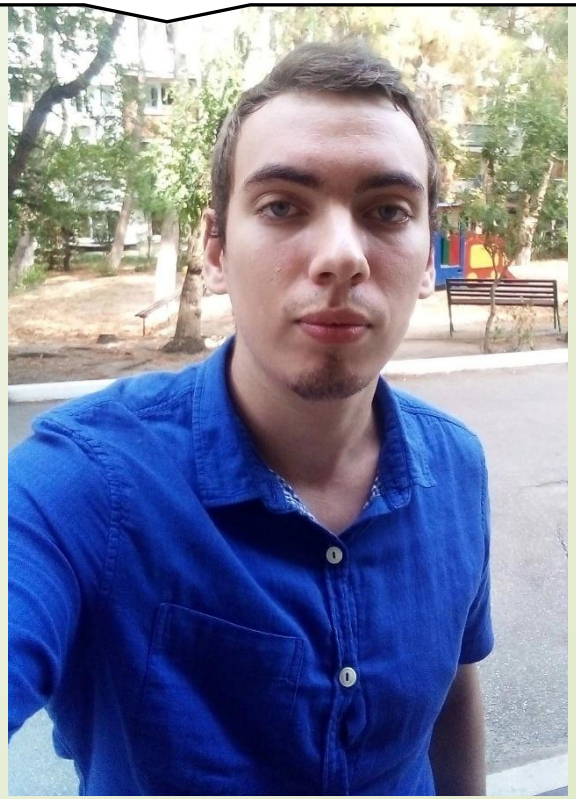
Ильяс Аппазов
Разработчик **frontend**

Наша команда

Данил Стина
Дизайнер



Садецкий Максим
Дизайнер



Техническое задание

- изучение материала по тематике проекта;
- распределение ролей участников проектной группы;
- составление плана выполнения работ;
- выбор технологии реализации сайта, разработка оригинального дизайна;
- определение основных функциональных особенностей сайта, его реализация и тестирование;
- подготовка документации проекта.

Проблема

Когда NFT начало набирать популярность, стали появляться маркетплейсы, в которые можно загружать свои NFT, создавать коллекции, продавать и покупать NFT у других авторов. К сожалению, у большинства маркетплейсов появилась одна общая проблема- это повторяющиеся NFT. Мы решили попытаться решить проблему повторного добавления уже имеющегося на сайте NFT и создать свой маркетплейс.

Предлагаемая технология решения проблемы

Было решено разработать алгоритм, который будет сравнивать новую картинку с другими, уже имеющимися на сайте. Немного о работе алгоритма: берутся две NFT, сравниваются по декомпозиции, сначала линейно потом по особому алгоритму, который основан на алгоритме нечёткого сходства строк Джаро-Винклера.

Расстояние Джаро d_j между двумя заданными строками s_1 и s_2 это:

$$d_j = \begin{cases} 0 & \text{когда } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{в остальных случаях} \end{cases}$$

Где:

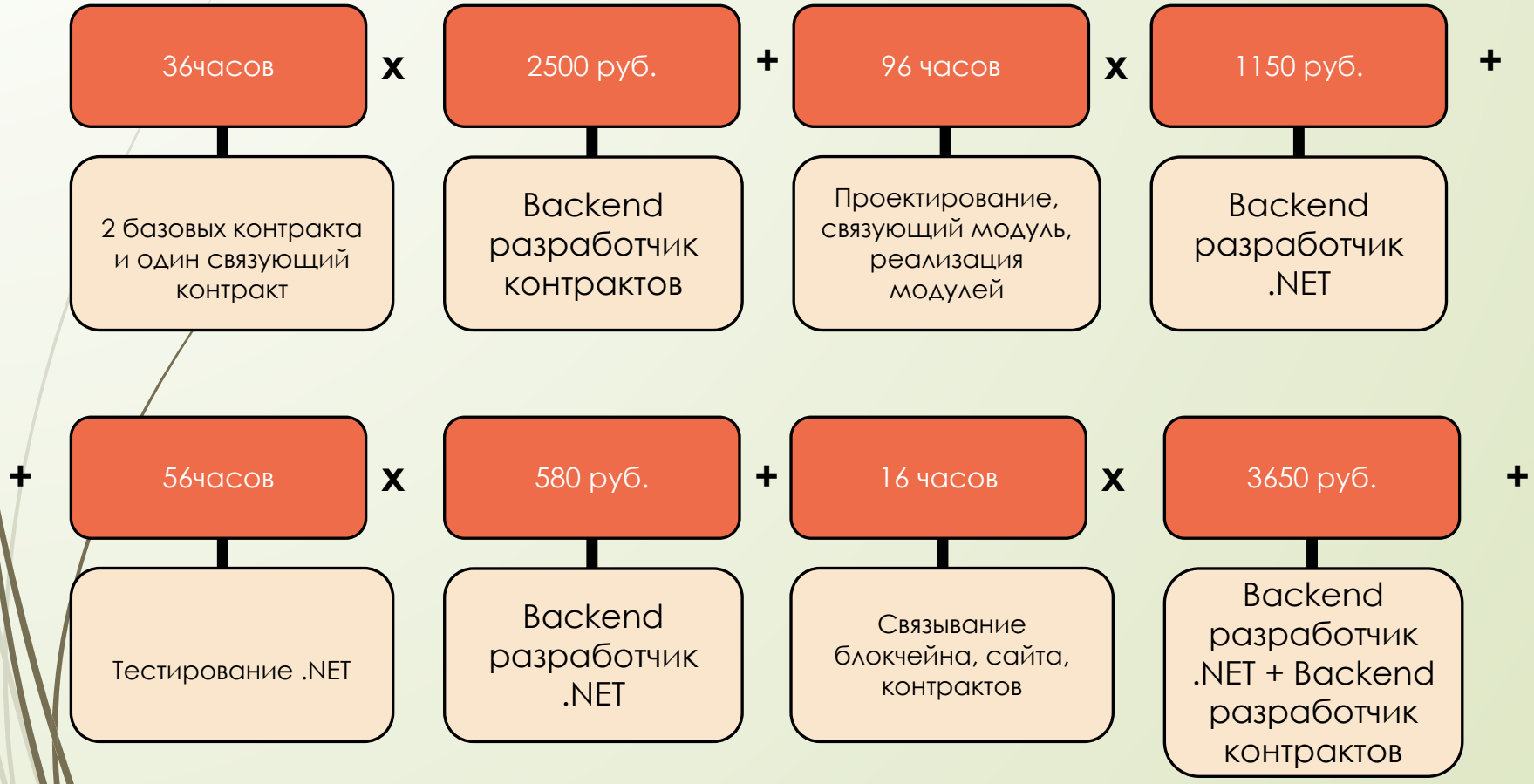
- $|s_i|$ — длина строки s_i;
- m — число *совпадающих символов* (см. ниже);
- t — половина числа *транспозиций* (см. ниже).

Продукт проекта

Продуктом нашего проекта будет являться NFT-маркетплейс. Пользователи смогут загружать на него свои NFT картинки, продавать их или при желании отключить возможность продажи. Кроме того, у пользователей сайта есть возможность покупать NFT картинки авторов за цену, которую выставил владелец NFT.

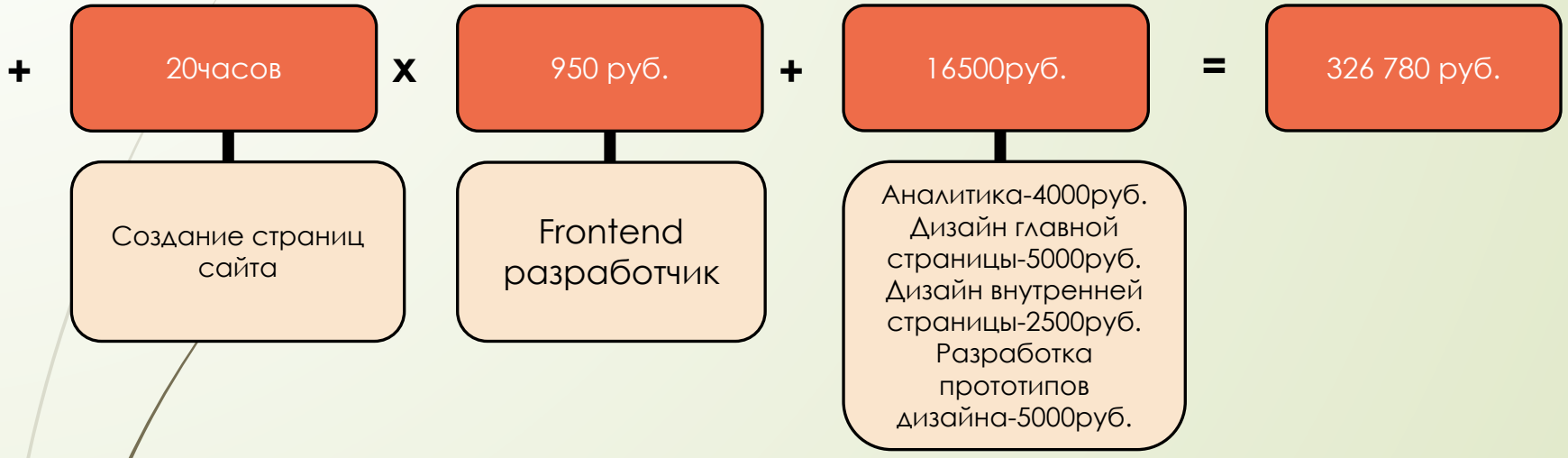
Проектный офис ИКТИБ

Оценка рынка



Проектный офис ИКТИБ

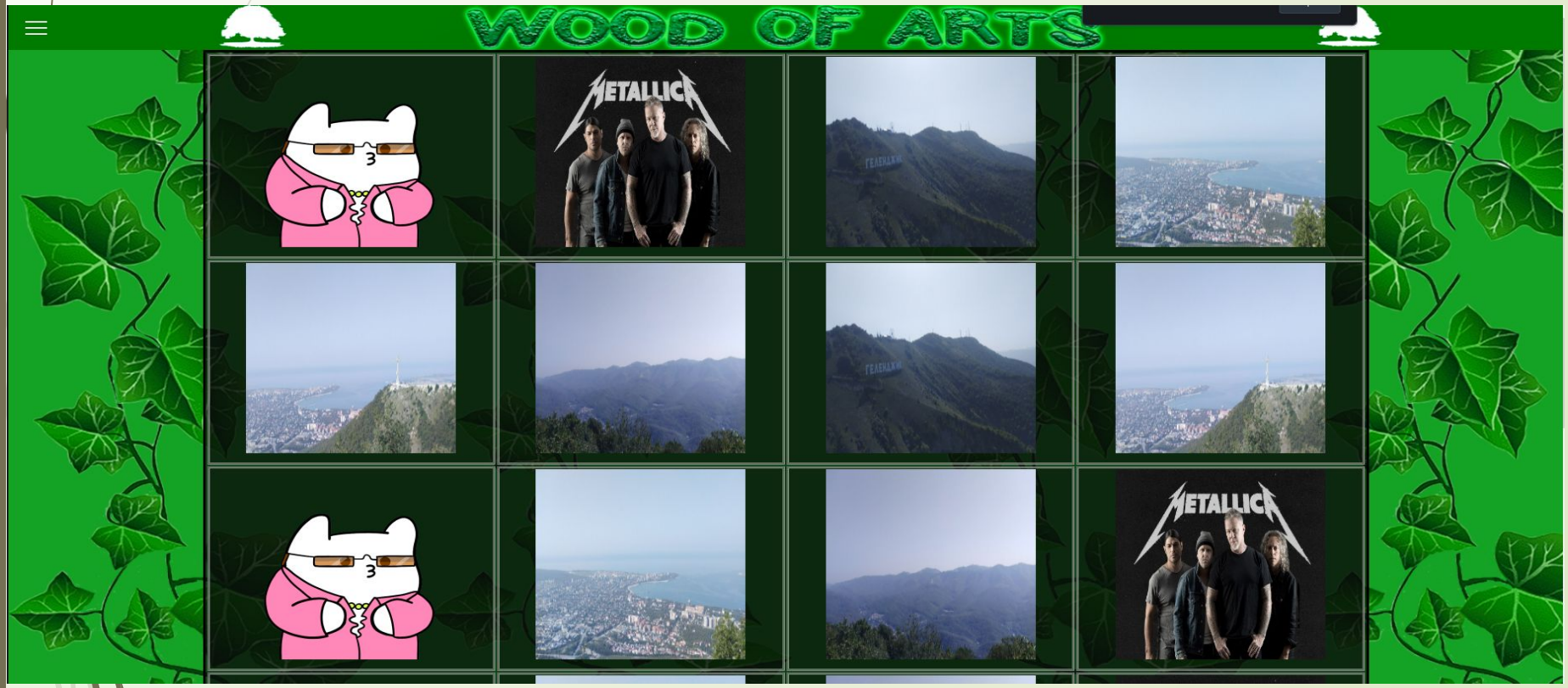
Оценка рынка



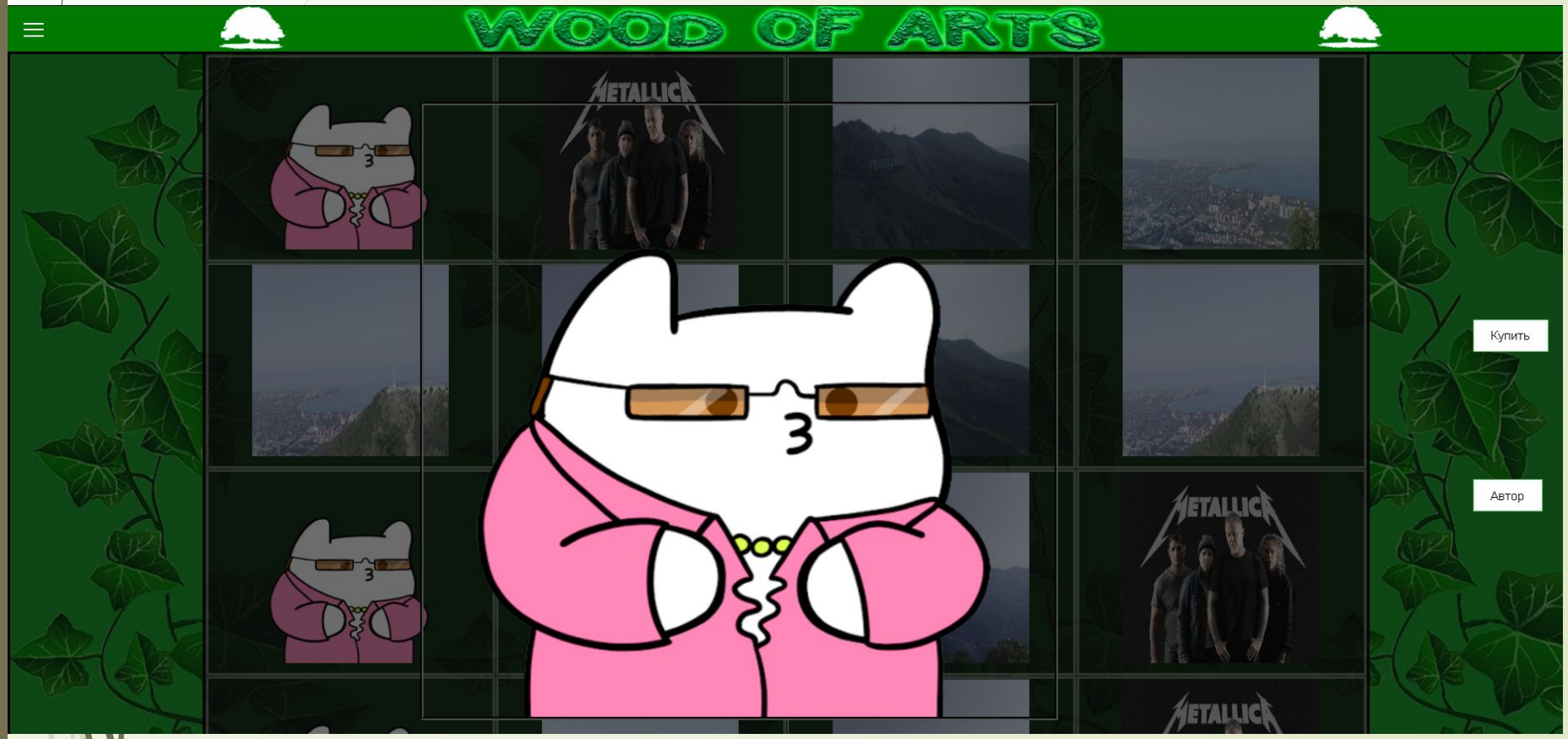
Проектный офис
ИКТИБ

Задачи нашей команды

Главная страница

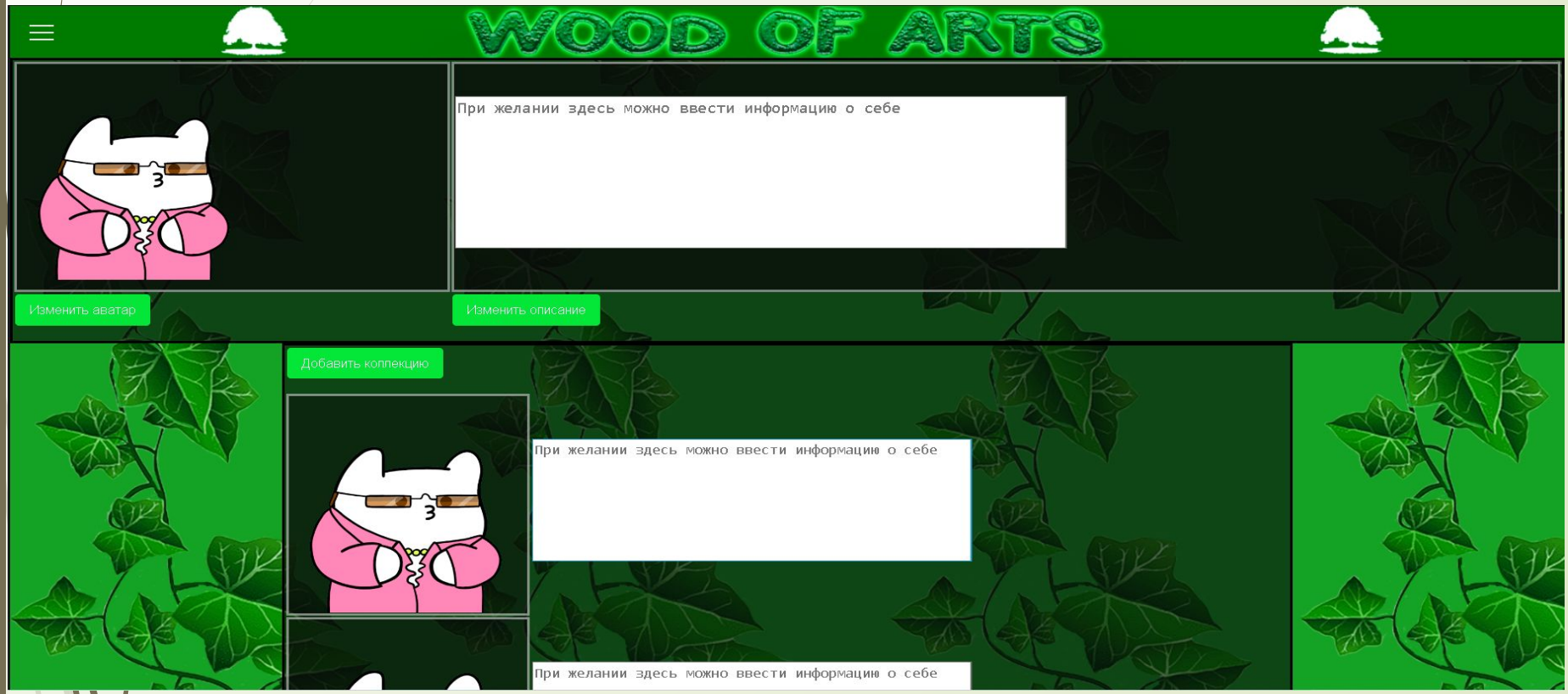


Главная страница

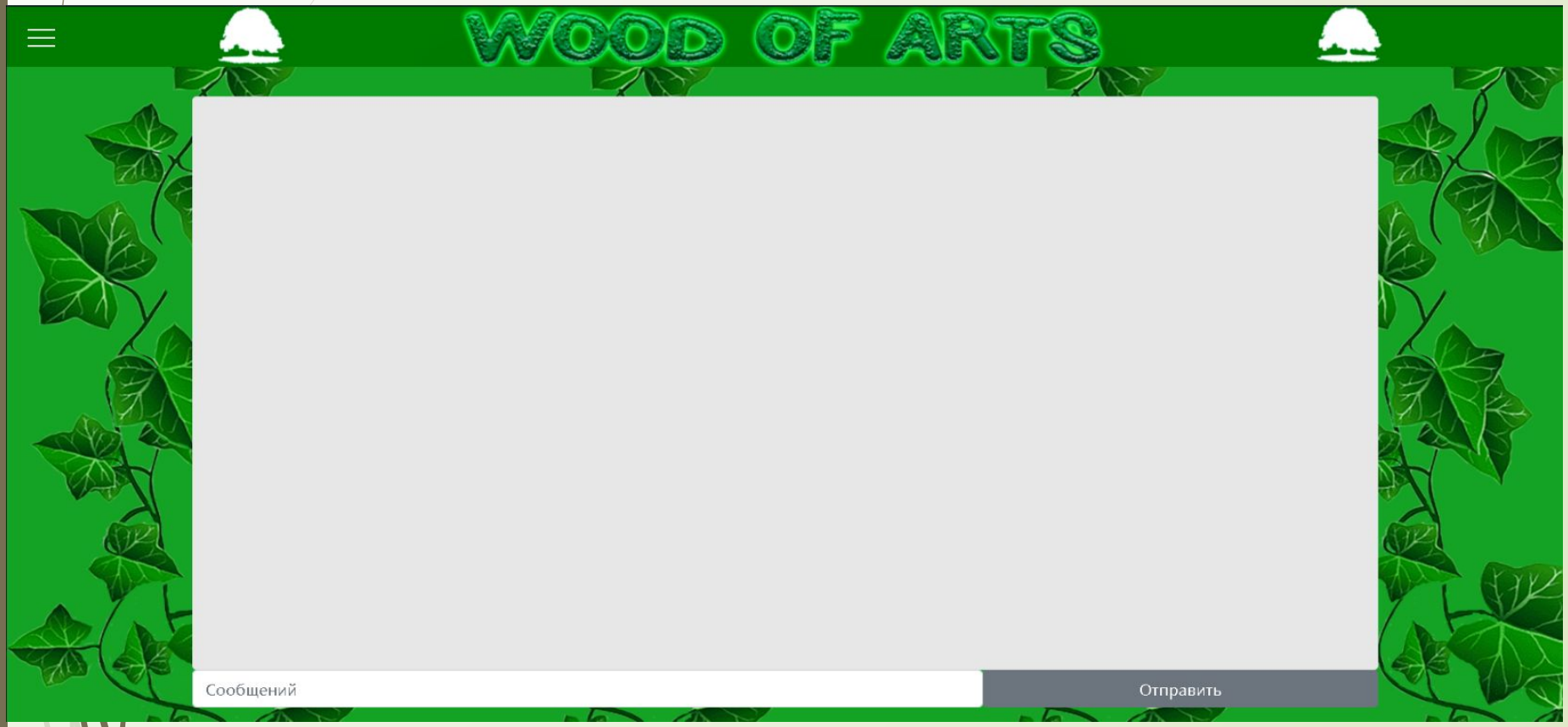


Проектный офис ИКТИБ

Страница личного кабинета



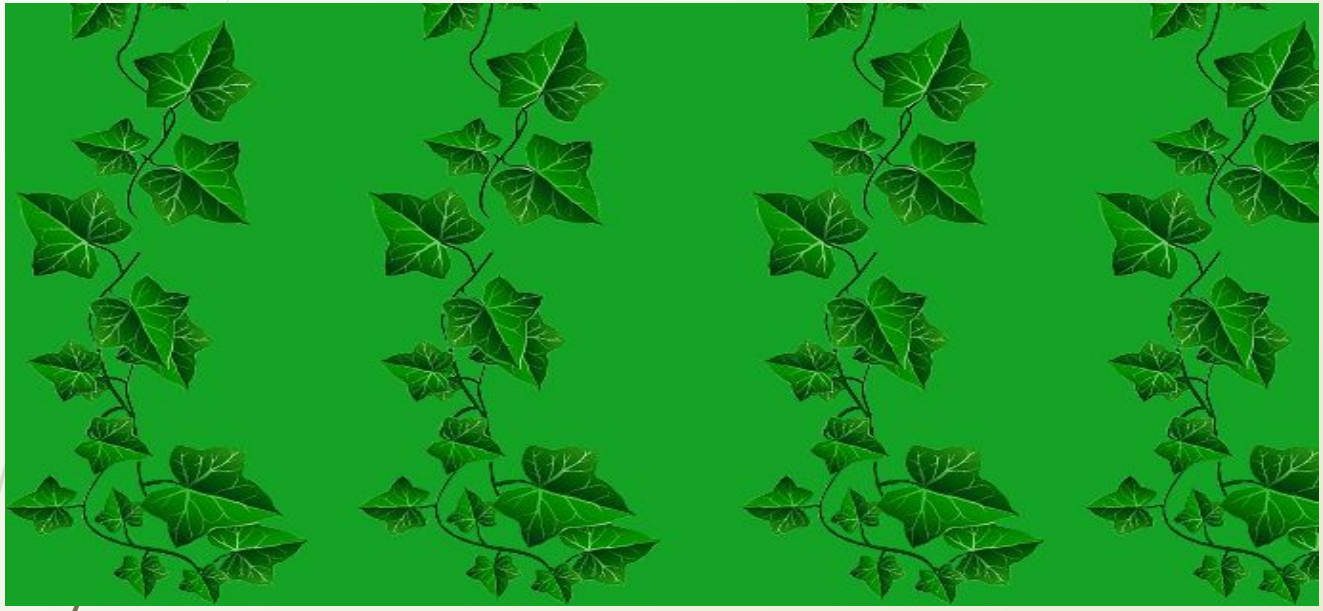
Страница обратной связи



Проектный офис
ИКТИБ

Меню для нашего сайта

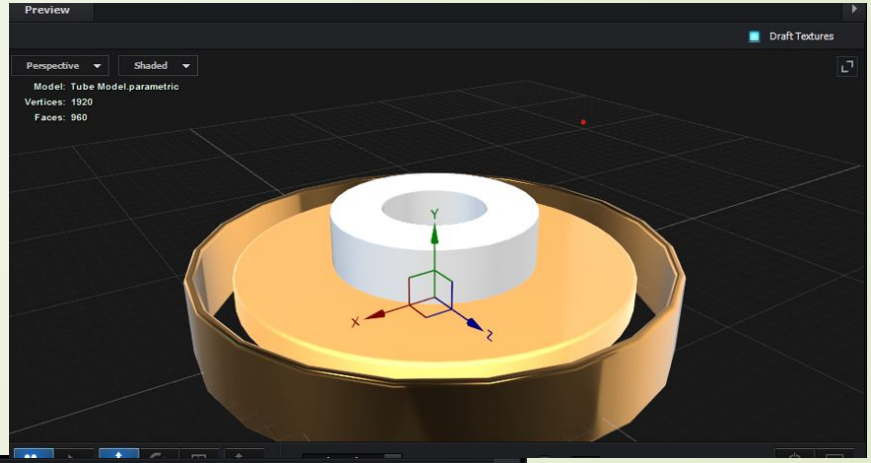
Фон, шапка и фавикон нашего сайта



Создание монеты

Проектный офис
ИКТИБ

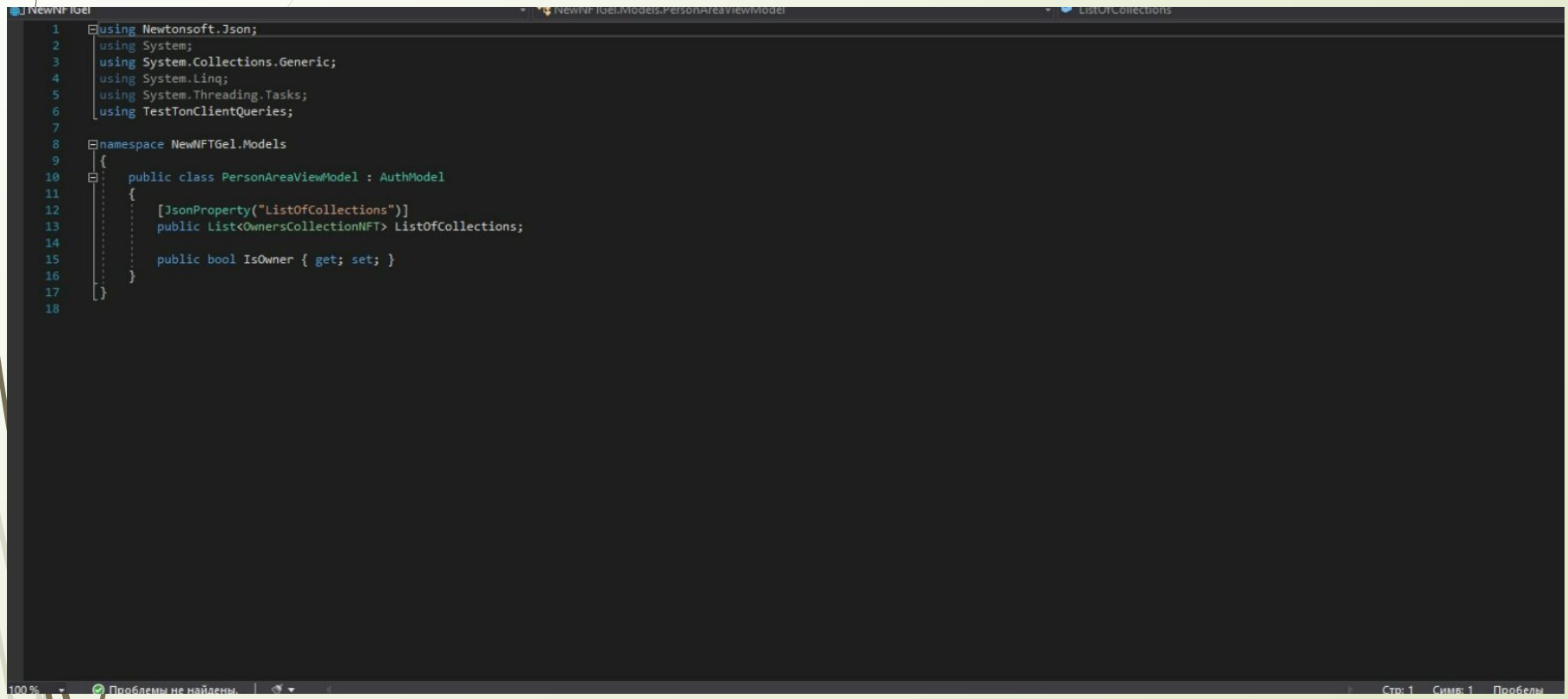
Создание монеты



Создание монеты

Проектный офис ИКТИБ

Backend



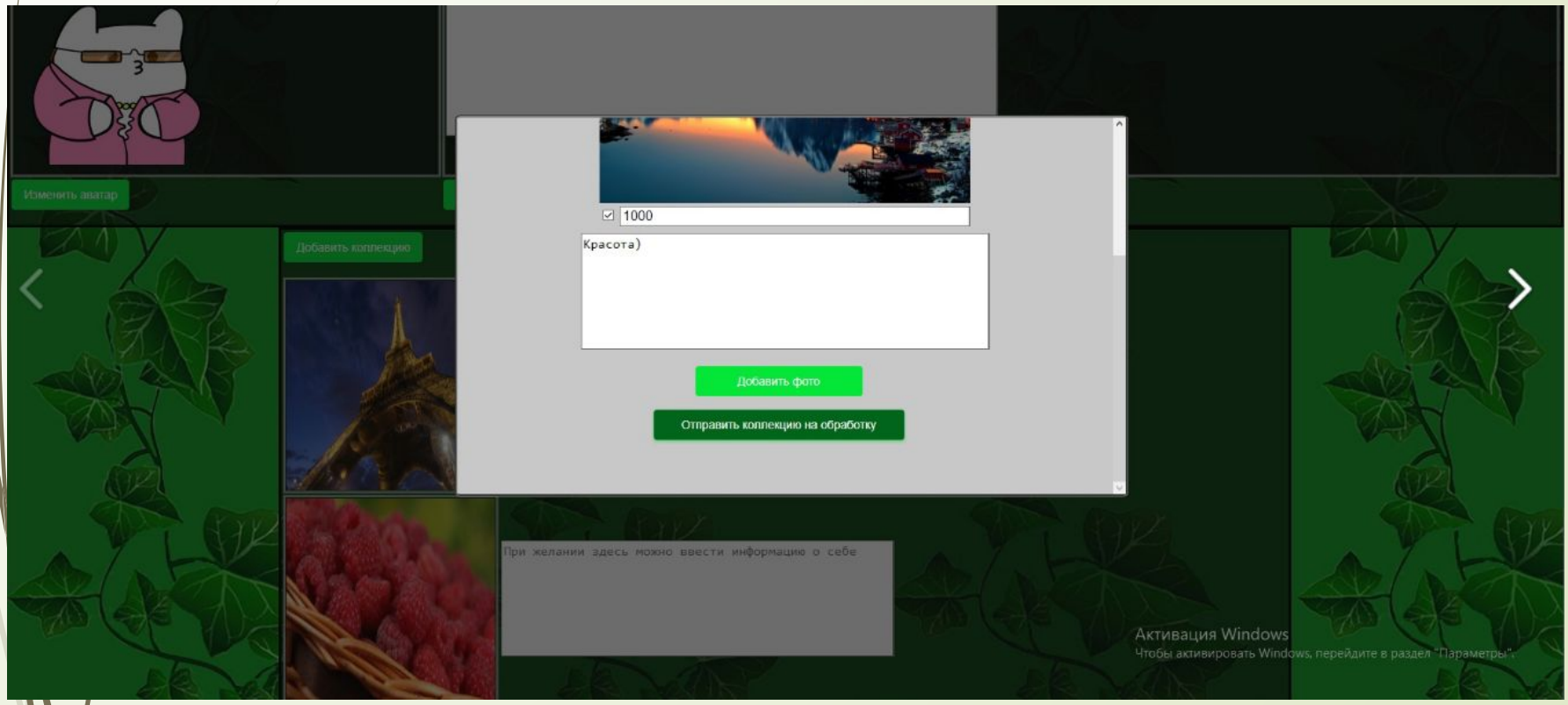
```
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using TestTonClientQueries;

namespace NewNFTGel.Models
{
    public class PersonAreaViewModel : AuthModel
    {
        [JsonProperty("ListOfCollections")]
        public List<OwnersCollectionNFT> ListOfCollections;

        public bool IsOwner { get; set; }
    }
}
```

Backend

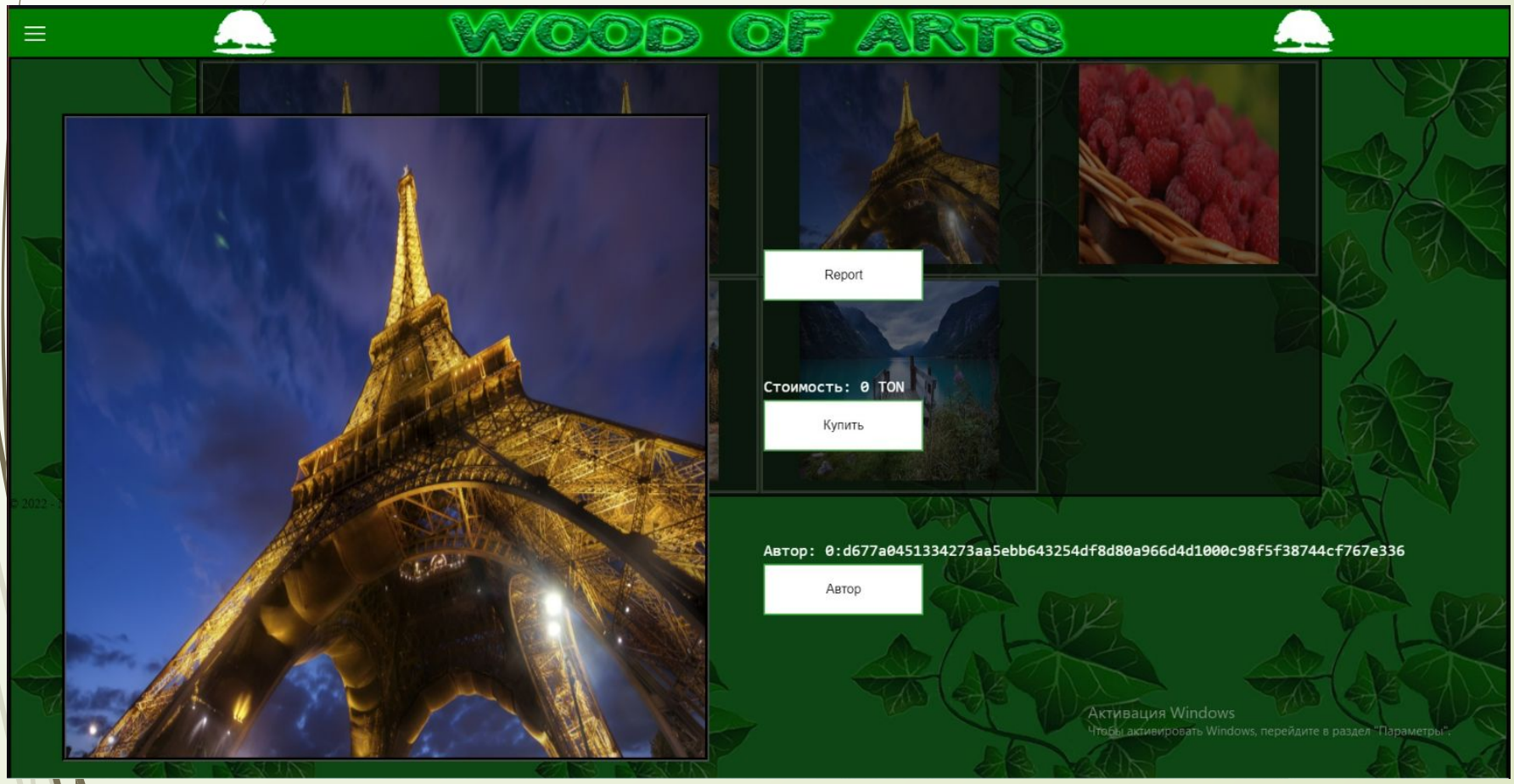
```
using Newtonsoft.Json;

namespace NewNFTGel.Models
{
    public abstract class AuthModel
    {
        [JsonProperty("secret_key")]
        public string secret_key { get; init; }

        [JsonProperty("public_key")]
        public string public_key { get; init; }

        [JsonProperty("address")]
        public string address { get; init; }

        [JsonProperty("IsFinaleAuth")]
        public bool IsFinaleAuth { get; set; }

        [JsonProperty("User")]
        public static string User => "PersonLog";
    }
}
```

Проектный офис
ИКТИБ

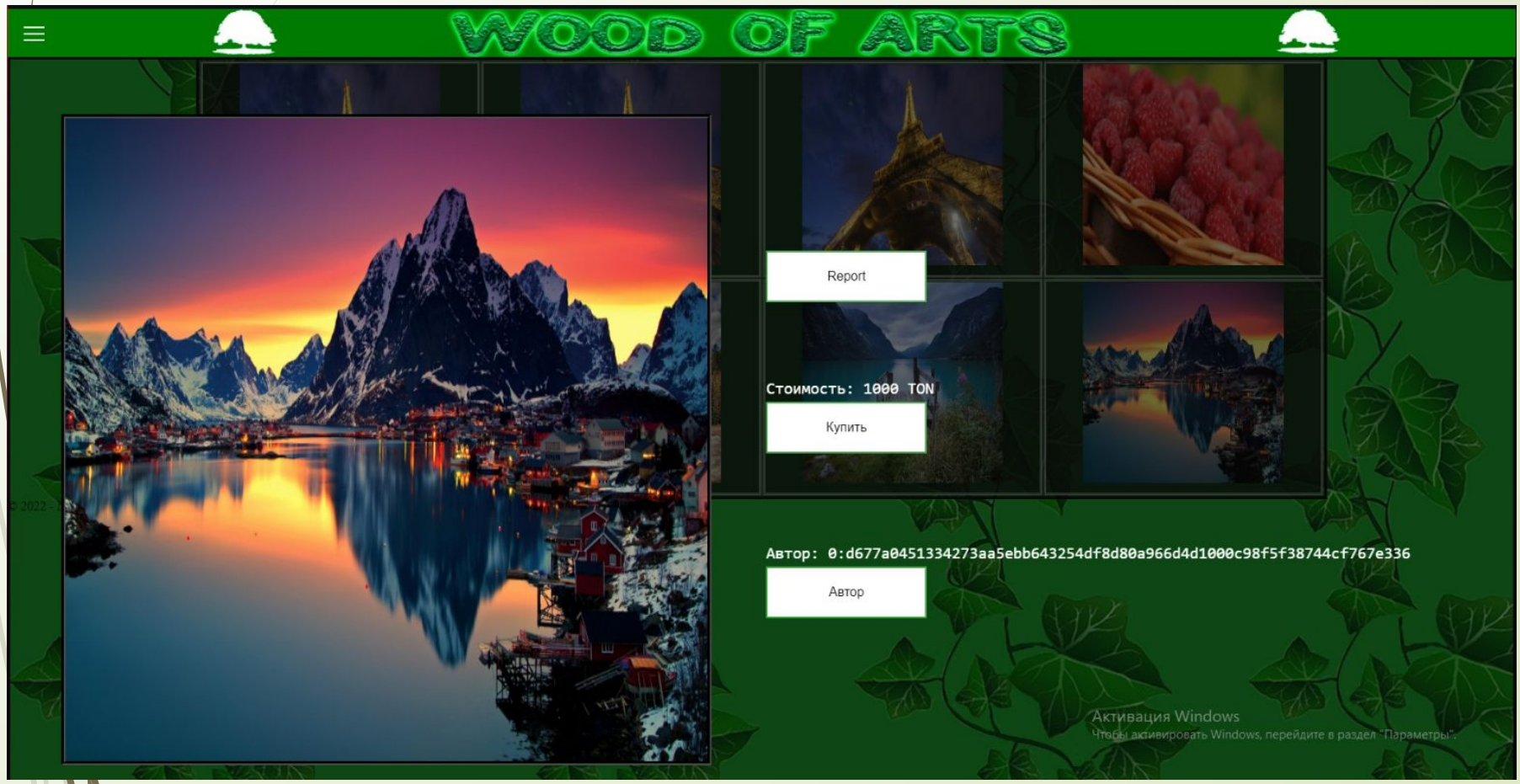
Backend



Backend


WOOD OF ARTS
Report
Стоимость: 0 TON
Купить
Автор: 0:d677a0451334273aa5ebb643254df8d80a966d4d1000c98f5f38744cf767e336
Автор
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Backend


WOOD OF ARTS
Report
Стоимость: 1000 TON
Купить
Автор: 0:d677a0451334273aa5ebb643254df8d80a966d4d1000c98f5f38744cf767e336
Автор
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Backend


WOOD OF ARTS
METALLICA
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Backend


WOOD OF ARTS
При желании здесь можно ввести информацию о себе
При желании здесь можно ввести информацию о себе
При желании здесь можно ввести информацию о себе
Активация Windows

Проектный офис ИКТИБ

Backend

```
    }

    uint128 lastID = 0;
    mapping(address => OwnerNFT) clientDB;
    mapping(uint128 => OwnersCollectionNFT) CollectionNFT;

    /// @dev Contract constructor.
    constructor() public{
        // check that contract's public key is set
        require(tvm.pubkey() != 0, 101);
        // Check that message has signature (msg.pubkey() is not zero) and message is signed with the owner's private key
        require(msg.pubkey() == tvm.pubkey(), 102);
       // clientDB[own.OwnAddrWallet] = own;
        tvm.accept();
    }

    // Modifier that allows function to accept external call only if it was signed
    // with contract owner's public key.
    modifier checkOwnerAndAccept {
        // Check that inbound message was signed with owner's public key.
        // Runtime function that obtains sender's public key.
        require(msg.pubkey() == tvm.pubkey(), 100);

        // Runtime function that allows contract to process inbound messages spending
        // its own resources (it's necessary if contract should process all inbound messages,
        // not only those that carry value with them).
        tvm.accept();
        _;
    }

    modifier alwaysAccept {
        tvm.accept();
        _;
    }

    function IsAuth(OwnerNFT own) public view checkOwnerAndAccept returns(bool){
        own.bal = 0;
        return clientDB.exists(own.OwnAddrWallet);
    }
```

Backend

```
12
13  struct OwnerNFT{
14      address OwnAddrWallet;
15      uint128 bal;
16      string LikesNFTbyID;
17  }
18
19  OwnerNFT _client;
20
21  /// @dev Contract constructor.
22  constructor() public{
23      // check that contract's public key is set
24      require(tvm.pubkey() != 0, 101);
25      // Check that message has signature (msg.pubkey() is not zero) and message is signed with the owner's private key
26      require(msg.pubkey() == tvm.pubkey(), 102);
27      _client.OwnAddrWallet = address(this);
28      tvm.accept();
29  }
30
31  function CreateClient(address addr, string likesNFT) public checkOwnerAndAccept returns(OwnerNFT client){
32      client.bal = address(this).balance;
33      client.LikesNFTbyID = likesNFT;
34      client.OwnAddrWallet = addr;
35      _client = client;
36      return _client;
37  }
38
39  function GetClient() public checkOwnerAndAccept returns(OwnerNFT client){
40      _client.bal = address(this).balance;
41      return _client;
42  }
43
44  function SetClient(OwnerNFT client) public checkOwnerAndAccept{
45      _client = client;
46  }
47
48  function GetBalance() public pure checkOwnerAndAccept returns(uint128 bal){
49      return address(this).balance;
50  }
```

Backend

```
39
40      /// @dev Contract constructor.
41      constructor(NFT nft) public{
42          // check that contract's public key is set
43          require(tvm.pubkey() != 0, 101);
44          // Check that message has signature (msg.pubkey() is not zero) and message is signed with the owner's private key
45          require(msg.pubkey() == tvm.pubkey(), 102);
46          _nft = nft;
47          tvm.accept();
48      }
49
50      function GetNFT() view public checkOwnerAndAccept returns(NFT nft){
51          return _nft;
52      }
53
54      // Modifier that allows function to accept external call only if it was signed
55      // with contract owner's public key.
56      modifier checkOwnerAndAccept {
57          // Check that inbound message was signed with owner's public key.
58          // Runtime function that obtains sender's public key.
59          require(msg.pubkey() == tvm.pubkey(), 100);
60
61          // Runtime function that allows contract to process inbound messages spending
62          // its own resources (it's necessary if contract should process all inbound messages,
63          // not only those that carry value with them).
64          tvm.accept();
65          _;
66      }
67
68      /// @dev Allows to transfer tons to the destination account.
69      /// @param dest Transfer target address.
70      /// @param value Nanotons value to transfer.
71      /// @param bounce Flag that enables bounce message in case of target contract error.
72      function sendTransaction(address dest, uint128 value, bool bounce) public pure checkOwnerAndAccept {
73          // Runtime function that allows to make a transfer with arbitrary settings.
74          dest.transfer(value, bounce, 0);
75      }
76
77  }
```

Проектный офис ИКТИБ

Backend

PictureComposition.cs | Program.cs

ModuleCheckPicture | ModuleCheckPicture.PictureComposition | PictureComposition(string path, bool IsFileComposition)

```
namespace ModuleCheckPicture
{
    abstract class Composition
    {
        public string CodeComposition;
        public int Height;
        public int Width;
        public Tuple<byte, byte, byte, byte> NormColor;
    }

    class PictureComposition : Composition
    {
        public string GetCodeComposition { get { return CodeComposition; } }

        public bool SaveCodeCompositionPicture(string path)
        {
            try
            {
                short count = 15;
                uint h = (uint)Height;
                uint w = (uint)Width;
                byte[] hb = new byte[4];
                byte[] wb = new byte[4];

                for (; count >= 0;)
                {
                    if (h - (int)Math.Pow(2, count) >= 0)
                    {
                        h -= (uint)Math.Pow(2, count);
                        hb[count / 8] += (byte)(Math.Pow(2, count)/Math.Pow(2, 8 * (count / 8)));
                    }
                    if (w - (int)Math.Pow(2, count) >= 0)
                    {
                        w -= (uint)Math.Pow(2, count);
                        wb[count / 8] += (byte)(Math.Pow(2, count) / Math.Pow(2, 8 * (count / 8)));
                    }
                    count--;
                }

                hb = hb.Concat(wb).ToArray();
                hb = hb.Concat(Encoding.Default.GetBytes(CodeComposition)).ToArray();

                using (FileStream fstream = new FileStream(path, FileMode.OpenOrCreate))
```

100 % | Проблемы не найдены. | Стр: 127 | Симв: 119 | Пробелы | CRLF

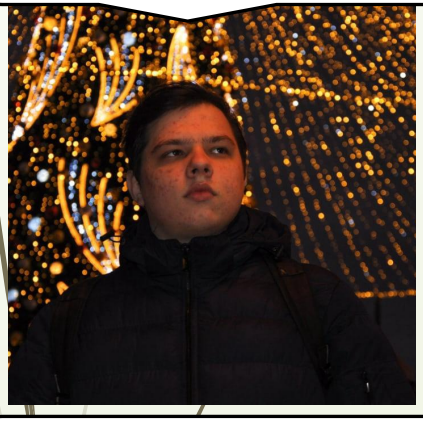
Список ошибок

Backend

Проектный офис ИКТИБ

Заключение

Тришкин Игорь
Капитан, разработчик **backend**



Илья Башмашников
Разработчик **frontend**

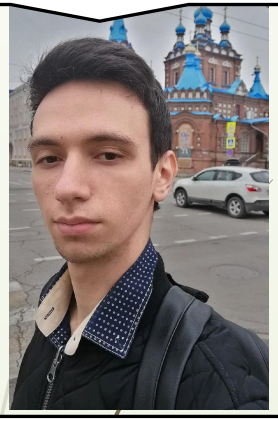
Сергей Гараев
Разработчик **frontend**

Авакимов Константин
Разработчик **frontend**

Проектный офис ИКТИБ

Заключение

Шевченко Лев
Разработчик **backend**



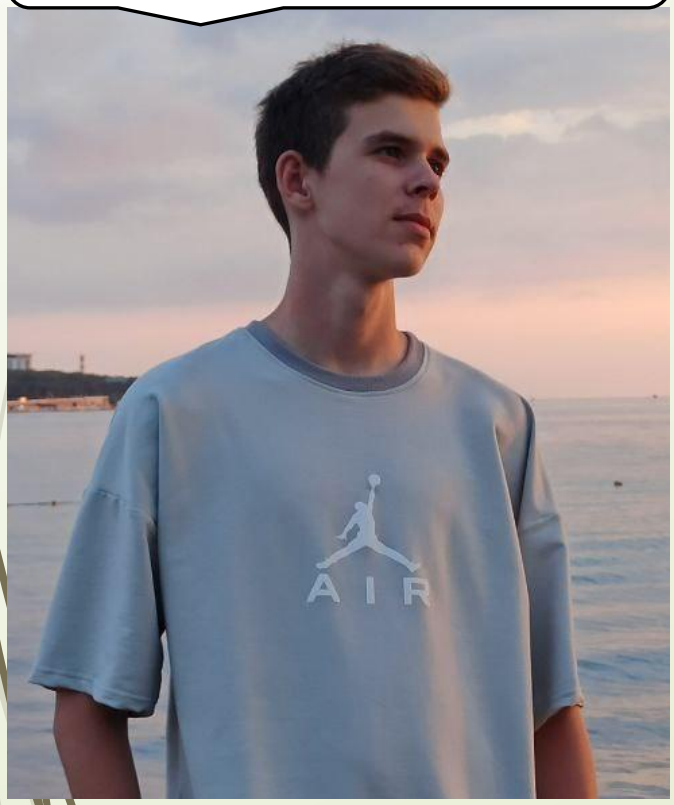
Бугримов Игорь
Разработчик **frontend**

Балабан Иван
Разработчик **frontend**

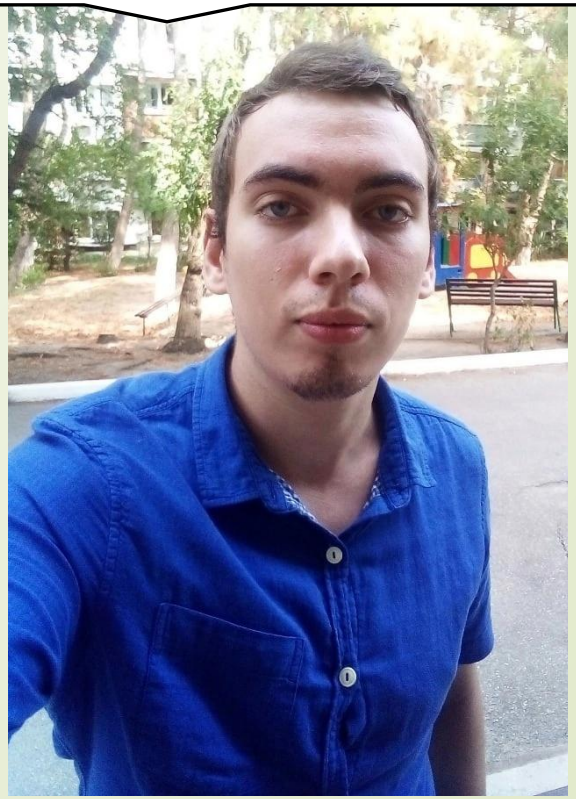
Ильяс Аппазов
Разработчик **frontend**

Заключение

Данил Стина
Дизайнер



Садецкий Максим
Дизайнер



Спасибо за внимание!