

Заключение

В ходе выполнения курсовой работы была проведена формализация задачи и данных, рассмотрены типы пользователей и требуемые функционалы. Также был проведен анализ существующих моделей баз данных и было решено использовать в данной работе реляционную СУБД. Спроектирована база данных и приложение для доступа к ней. Был спроектирован триггер, осуществляющий автоматический пересчет количества товаров на складе при добавления заказов. Представлены средства разработки программного обеспечения, выбор языка программирования и описан интерфейс программы. Также рассмотрены примеры работы программы

В результате исследования было выяснено, что при использовании индексирования ускоряет выполнение запросов SELECT больше чем в 5 раз.

Данная программа может иметь следующие перспективы развития:

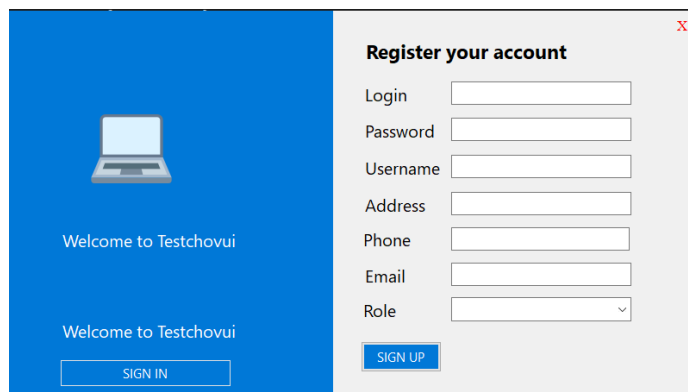
- анализ продаж: внедрение инструментов для анализа данных о продажах, тенденциях и предпочтениях клиентов;
- аналитические панели и отчеты: создание удобных интерфейсов для управления данными и генерации отчетов для аналитики;

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Зачем нужен интернет-магазин? <https://integrion.biz/articles/zachem-nuzhen-internet-magazin>.
2. Бизнес-план магазина компьютерной техники. <https://plan-pro.ru/torgovlya/raznye-magaziny/biznes-plan-magazina-kompyuternoj-tehniki/>.
3. Информационная система компьютерного магазина. <https://cyberleninka.ru/article/n/informatsionnaya-sistema-kompyuternogo-magazina/viewer>.
4. КаРа-Ушанов В. Ю. SQL — Язык реляционных баз данных. 2016. с. 6.
5. Документация по C#. <https://learn.microsoft.com/ru-ru/dotnet/csharp/>.
6. Что такое реляционная база данных. <https://help.reg.ru/support/server/vps/oblastnyye-bazy-dannykh/zakaz-i-upravleniye-uslugoy-oblastnyye-bazy-dannykh/relyatsionnyye-bazy-dannykh#0>.
7. Для чего используется триггер? <https://blog.skillfactory.ru/triggery-v-bazah-dannyh/>.
8. Сравнение современных СУБД. <https://drach.pro/blog/hi-tech/item/145-db-comparison>.

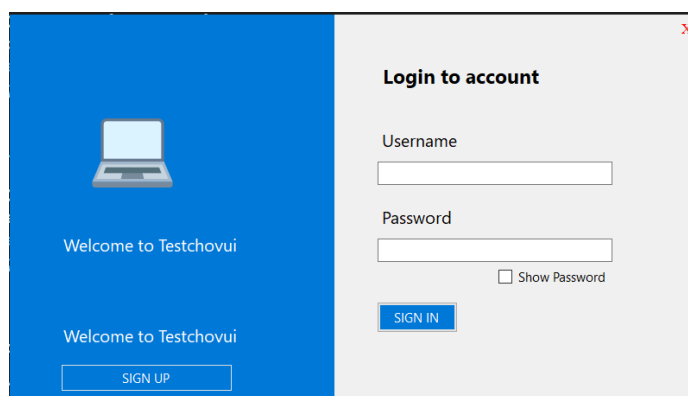
ПРИЛОЖЕНИЕ А

Описание интерфейсов



The screenshot displays the registration interface of the Testchovui application. It is divided into two main sections. The left section has a blue background and features a laptop icon, the text "Welcome to Testchovui" repeated twice, and a "SIGN IN" button. The right section, titled "Register your account" with a red close button (X) in the top right corner, contains a form with the following fields: "Login", "Password", "Username", "Address", "Phone", "Email", and "Role" (a dropdown menu). A "SIGN UP" button is located at the bottom of the form.

Рисунок 4.2 – Демонстрация работы программы при регистрации



The screenshot displays the login interface of the Testchovui application. It is divided into two main sections. The left section has a blue background and features a laptop icon, the text "Welcome to Testchovui" repeated twice, and a "SIGN UP" button. The right section, titled "Login to account" with a red close button (X) in the top right corner, contains a form with the following fields: "Username" and "Password". Below the password field is a checkbox labeled "Show Password". A "SIGN IN" button is located at the bottom of the form.

Рисунок 4.3 – Демонстрация работы программы при входе в систему

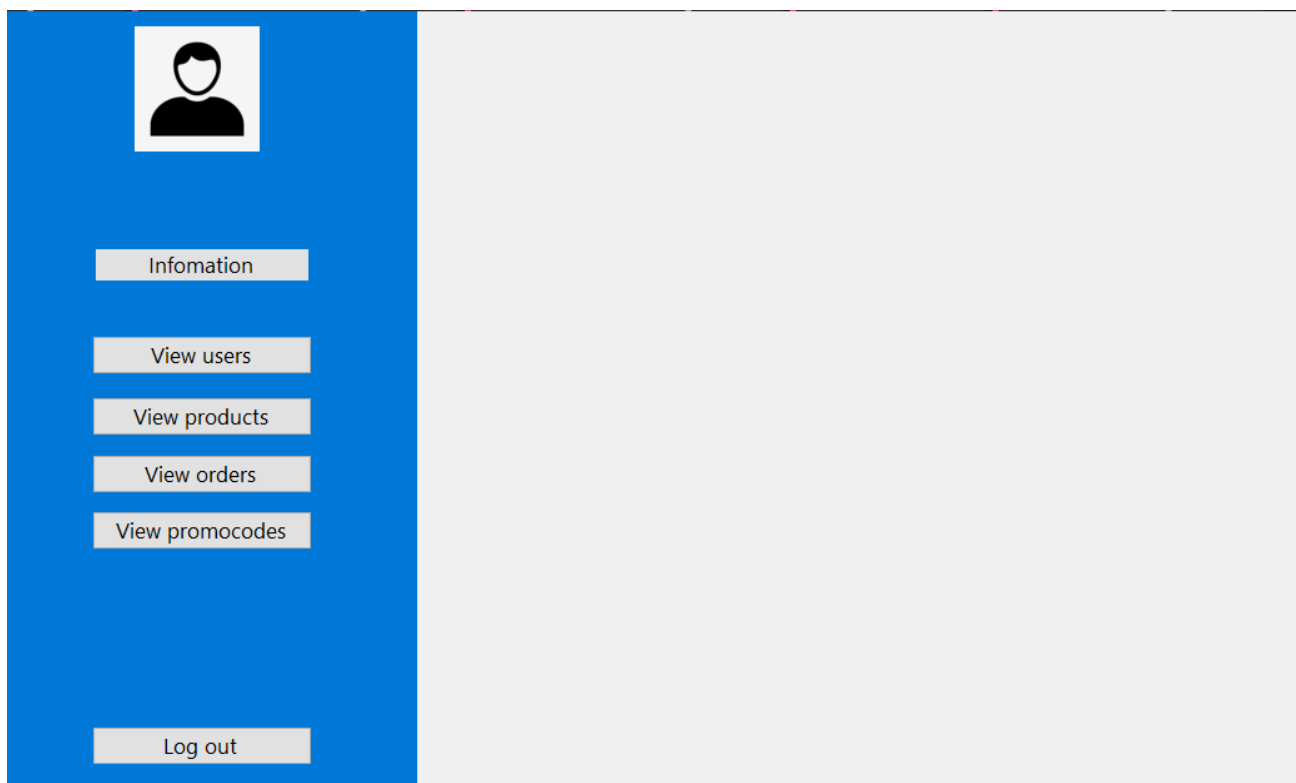


Рисунок 4.4 – Демонстрация главного экрана админстратора

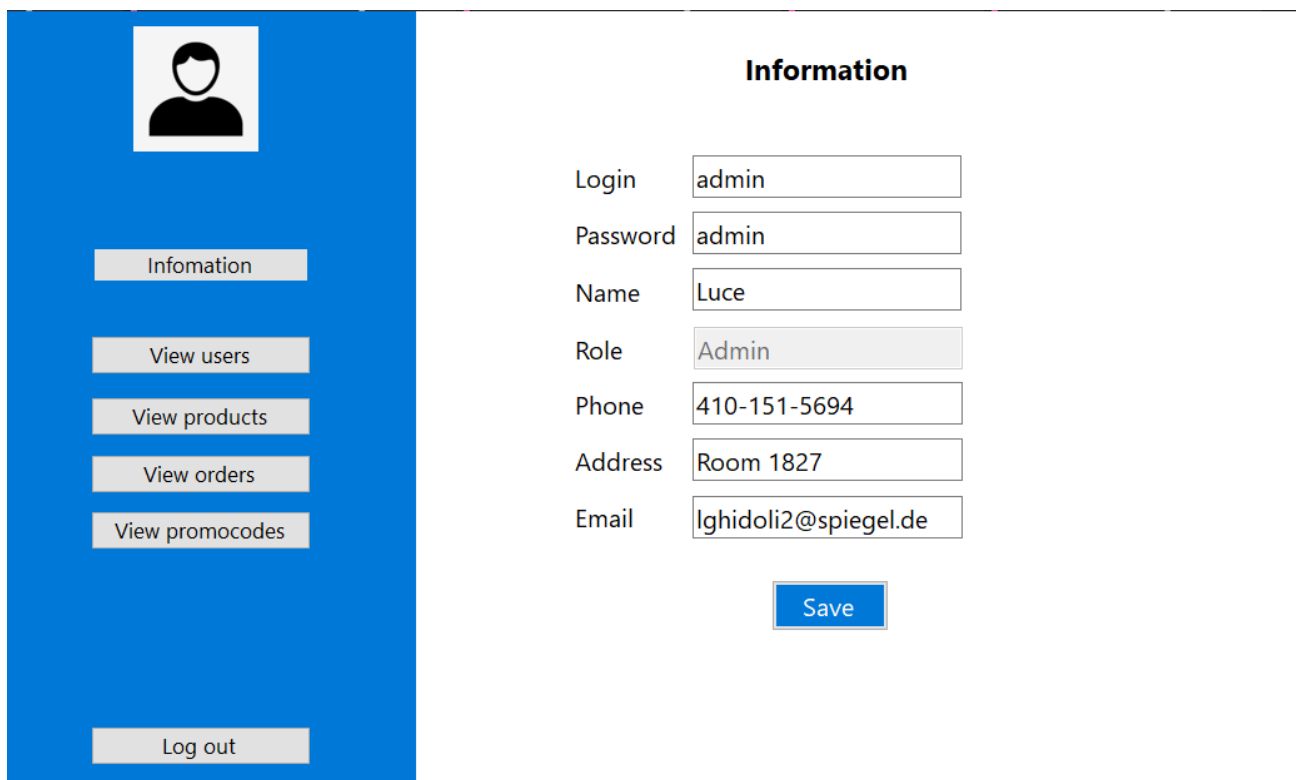



Рисунок 4.5 – Демонстрация экрана админстратора при просмотре информации



Information
 View users
 View products
 View orders
 View promocodes
 Log out


ID	Login	Password	Name	Address	Phone
1	admin	admin	Luce	Room 1827	410
2	client	client	Lacey	PO Box 90...	37
4	Spriggs	ed	Lacy	Apt 708	98
5	Elden	123	Kaylyn	PO Box 61...	66
6	Longland	33	Krystal	8th Floor	59
7	Downing	fP5#4//D6	Trstram	PO Box 18...	79
9	Daughton	iD9~0Luqll...	Madelene	19th Floor	23
10	McKinna	rB9/\$W50j...	Borq	PO Box 85...	28
8	seller	seller	Michael	Apt 1109	84

Login Daughton
 Password iD9~0Luqll\$j#K67
 Name Madelene
 Role Client

Phone 231-252-1575
 Address 19th Floor
 Email mdaughton8@vkontakte

Add
 Delete
 Update

Рисунок 4.6 – Демонстрация экрана администратора при просмотре пользователей




Information
 View users
 View products
 View orders
 View promocodes
 Log out

ID	Name	Price	Quantity	Manufacturer
1	Galliano	12	75	Coqilith
2	Fireball Wh...	96	49	Browseblab
3	Cookies - A...	100	72	Jabbertype
4	Mix Pina C...	79	51	Myworks
5	Beef Cheek ...	99	28	Aqivu
6	Jolt Cola - ...	51	8	Roexo
7	Beer - Laba...	68	37	Zooveo
8	Tomatoes - ...	24	71	Thoughttwo...
9	Bacardi Bre...	26	29	Fiveclub
10	Pepper - Re...	88	74	Coqidoo

Name Beer - Labatt Blue
 Price 68
 Manufacturer Zooveo
 Quantity 37
 Description Insect, stick

Add
 Delete
 Update

Рисунок 4.7 – Демонстрация экрана администратора при просмотре товаров



Information

View users

View products

View orders

View promocodes

Log out

ID	Data	Id_User	Id_promo	Status
1	6/10/24 12...	1	1	Init
2	10/23/23 1...	2	2	Init
4	2/9/24 12:...	4	4	Init
5	12/31/23 1...	5	5	Init
6	8/18/23 12...	6	6	Delivered
7	3/21/24 12...	7	7	Delivered
8	1/11/24 12...	8	8	Delivered
9	12/27/23 1...	10	9	Delivered
10	2/23/24 12...	10	10	Delivered
11	8/29/24 12...	2	1	Init
12	8/29/24 12...	2	2	Init

Status: Delivered


Data: 8/18/23 12:00:00 AM

User Login: Longland

PromoCode: CDU

Delete
Update

Рисунок 4.8 – Демонстрация экрана администратора при просмотре заказов



Information

View users

View products

View orders

View promocodes

Log out

ID	Code	Discount	Start	End
1	0	0	12/31/23 1...	11/2/23 12...
2	HSV	41	4/19/24 12...	7/10/24 12...
3	CIE	8	8/20/23 12...	7/1/24 12:...
4	OZC	78	2/28/24 12...	4/13/24 12...
5	YLV	35	9/29/23 12...	8/14/23 12...
6	CDU	30	7/25/24 12...	4/14/24 12...
7	MXK	99	12/8/23 12...	5/29/24 12...
8	AVB	89	9/8/23 12:...	11/23/23 1...
9	SUG	74	11/25/23 1...	11/1/23 12...
10	IKK	32	1/3/24 12:...	4/16/24 12...

Code:


Discount:

Start:

End:

Add
Delete
Update

Рисунок 4.9 – Демонстрация экрана администратора при просмотре промокодов



Information

View products

View orders


Log out

Information

Login
 Password
 Name
 Role
 Phone
 Address
 Email

Save

Рисунок 4.10 – Демонстрация главного экрана поставщика



Information

View users

View products

View orders

View promocodes

Log out

ID	Code	Discount	Start	End
1	0	0	12/31/23 1...	11/2/23 12...
2	HSV	41	4/19/24 12...	7/10/24 12...
3	CIE	8	8/20/23 12...	7/1/24 12:...
4	OZC	78	2/28/24 12...	4/13/24 12...
5	YLV	35	9/29/23 12...	8/14/23 12...
6	CDU	30	7/25/24 12...	4/14/24 12...
7	MXK	99	12/8/23 12...	5/29/24 12...
8	AVB	89	9/8/23 12:...	11/23/23 1...
9	SUG	74	11/25/23 1...	11/1/23 12...
10	IKK	32	1/3/24 12:...	4/16/24 12...
*				

Code

Discount

Start


End

Add

Delete

Update

Рисунок 4.11 – Демонстрация экрана клиентов при просмотре корзины



Information

View products

View carts

View orders

View promos

Log out

ID	Name	Price	Quantity	Manufacturer
1	Galliano	12	75	Cogilith
2	Fireball Whisky	96	49	Browseblab
3	Cookies - Assor...	100	72	Jabbertype
4	Mix Pina Colada	79	51	Myworks
5	Beef Cheek Fresh	99	28	Agivu
6	Jolt Cola - Elect...	51	8	Rooxo
7	Beer - Labatt Bl...	68	37	Zooveo
8	Tomatoes - Vin...	24	71	Thoughtworks
9	Bacardi Breezer...	26	29	Fiveclub
10	Pepper - Red C...	88	74	Cogidoo

Name

Mix Pina Colada

Price

79

Manufacturer

Myworks

Quantity

51

Description

Smith's bush squirrel

Choose Cart

Add to Cart

Рисунок 4.12 – Демонстрация экрана клиентов при добавления товаров в корзину

Создание базы данных

Листинг 4.1 – Создание всех таблиц

```
drop table userdb CASCADE;
drop table promodb CASCADE;
drop table productdb CASCADE;
drop table userpromodb CASCADE;
drop table cartdb CASCADE;
drop table itemcartdb CASCADE;
drop table orderdb CASCADE;
drop table itemorderdb CASCADE;

create table UserDB(
    Id serial primary key,
    Name varchar(50) not null,
    Phone varchar(50) not null,
    Address varchar(50) not null,
    Email varchar(50) not null,
    Login varchar(50) not null,
    Password varchar(50) not null,
    Role varchar(50) not null
);

create table PromoDB(
    Id serial primary key,
    code varchar(20) not null,
    discount int not null,
    data_start date not null,
    data_end date not null
);

create table ProductDB (
    Id serial primary key,
    Name varchar(50) not null,
    Price int not null,
    Quantity int not null,
    Manufacturer varchar(50) not null,
    Description varchar(50) not null
);

create table UserPromoDB(
    ID serial primary key,
    id_user int not null,
    id_promo int not null,
    foreign key (id_user) references UserDB(Id) ON DELETE CASCADE,
    foreign key (id_promo) references PromoDB(Id) ON DELETE CASCADE
);
```

```

create table CartDB(
    Id serial primary key,
    data_created date not null,
    id_user int references UserDB(Id) ON DELETE CASCADE
);

create table ItemCartDB(
    Id serial primary key,
    id_product int not null,
    id_cart int not null,
    quantity int not null,
    foreign key (id_cart) references CartDB(Id) ON DELETE CASCADE,
    foreign key (id_product) references ProductDB(Id) ON DELETE CASCADE
);

create table orderDB(
    Id serial primary key,
    status varchar(20) not null,
    data_created date not null,
    id_user int not null,
    id_promo int references PromoDB(Id) ON DELETE CASCADE,
    foreign key (id_user) references UserDB(Id) ON DELETE CASCADE
);

create table ItemOrderDB(
    Id serial primary key,
    id_product int not null,
    id_order int not null,
    quantity int not null,
    foreign key (id_order) references OrderDB(Id) ON DELETE CASCADE,
    foreign key (id_product) references ProductDB(Id) ON DELETE CASCADE
);

```

Листинг 4.2 – Реализация триггера

```

drop trigger after_itemorder_insert on itemorderdb;

create or replace function process_itemorder()
returns trigger
as
$$
begin
    update productdb set quantity = (select quantity - new.quantity from
        productdb where id = new.id_product) where id = new.id_product;
    RETURN NEW;

end;
$$ LANGUAGE plpgsql;

```

```
CREATE TRIGGER after_itemorder_insert
AFTER INSERT ON itemorderdb
for each row
EXECUTE FUNCTION process_itemorder();
```

Листинг 4.3 – Создание роли администратора и выдача права

```
create role Radmin with
    connection limit -1
    login
    password 'admin';

grant all privileges
on all tables in schema public
to Radmin;
```

Листинг 4.4 – Создание роли поставщика и выдача права

```
create role Rseller with
    connection limit 2
    login
    password 'seller';

grant select on
    public."productdb",
    public."orderdb",
    public."userdb"
to Rseller;

grant insert on
    public."productdb"
to Rseller;

grant update on
    public."productdb",
    public."orderdb",
    public."itemorderdb",
    public."userdb"
to Rseller;

grant delete on
    public."productdb",
    public."orderdb",
    public."itemorderdb"
to Rseller;
```

Листинг 4.5 – Создание роли клиента и выдача права

```
create role Rclient with
```

```

connection limit 100
login
password 'client';

grant select on
    public."productdb",
    public."cartdb",
    public."itemcartdb",
    public."orderdb",
    public."itemorderdb",
    public."userdb"
to Rclient;

grant insert on
    public."cartdb",
    public."itemcartdb",
    public."orderdb",
    public."itemorderdb"
to Rclient;

grant update on
    public."cartdb",
    public."itemcartdb",
    public."orderdb",
    public."itemorderdb",
    public."userdb"
to Rclient;

grant delete on
    public."cartdb",
    public."itemcartdb"
to Rclient;

```

Листинг 4.6 – Создание роли гостя и выдача права

```

create role Rguest with
    connection limit 100
    login
    password 'guest';

grant insert on
    public."userdb"
to Rguest;

```

Листинг 4.7 – Реализация тестирования для триггера after_itemorder_insert

```

CREATE OR REPLACE FUNCTION test_trigger(product_id int, order_id int,
    quantityI int)
RETURNS void AS $$

```

```

DECLARE
    quantity_after int;
    quantity_before int;
BEGIN
    select quantity into quantity_before from productdb where id = product_id;

    INSERT INTO itemorderdb(id_product, id_order, quantity) VALUES
        (product_id, order_id, quantityI);

    SELECT quantity INTO quantity_after FROM productdb WHERE id =
        product_id;

    if (quantity_after = quantity_before - quantityI) then
        RAISE NOTICE 'This test passed';
    else
        RAISE EXCEPTION 'Test failed';
    end if;
END;
$$ LANGUAGE plpgsql;

--
SELECT test_trigger(1, 1, 5);
SELECT test_trigger(2, 1, 5);
SELECT test_trigger(3, 1, 5);

```

Листинг 4.8 – Функция входа в систему

```

private void btnSignIn_Click(object sender, EventArgs e)
{
    try
    {
        string login = tbUsername.Text;
        string password = tbPassword.Text;
        if (login == "" || password == "")
        {
            throw new Exception("Input error");
        }
        User user = _userService.Login(login, password);
        switch (user.Role)
        {
            case Role.Admin:
                FormAdmin frm_admin = new FormAdmin(user.Id, _userService,
                    _productService, _promoService, _orderService,
                    _cartService, _itemOrderService, _itemCartService);
                frm_admin.ShowDialog(this);
                break;
            case Role.Seller:
                FormSeller frm_seller = new FormSeller(user.Id,
                    _userService, _productService, _promoService,

```

```

        _orderService, _cartService, _itemOrderService,
        _itemCartService);
    frm_seller.ShowDialog(this);
    break;
case Role.Client:
    FormClient frm_client = new FormClient(user.Id,
        _userService, _productService, _promoService,
        _orderService, _cartService, _itemOrderService,
        _itemCartService);
    frm_client.ShowDialog(this);
    break;
}
this.Close();
}
catch (Exception ex) { MessageBox.Show(ex.Message, "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error); }
}

```

Листинг 4.9 – Функция работы пользователей

```

public User GetUser(int id)
{
    CheckConnection.checkConnection(Connector);
    string query = queryGetUser(id);

    NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);

    User user = null;

    NpgsqlDataReader reader = cmd.ExecuteReader();
    if (reader.Read())
    {
        user = new User(reader.GetInt32(0), reader.GetString(1),
            reader.GetString(2),
            reader.GetString(3), reader.GetString(4),
            reader.GetString(5), reader.GetString(6),
            (Role)Enum.Parse(typeof(Role), reader.GetString(7)));
    }
    reader.Close();
    return user;
}

public User GetUser(string login)
{
    CheckConnection.checkConnection(Connector);
    string query = queryGetUser(login);

    NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);

```

```

        User user = null;
        NpgsqlDataReader reader = cmd.ExecuteReader();

        if (reader.Read())
        {
            user = new User(reader.GetInt32(0), reader.GetString(1),
                reader.GetString(2),
                reader.GetString(3), reader.GetString(4), reader.GetString(5),
                reader.GetString(6), (Role)Enum.Parse(typeof(Role),
                    reader.GetString(7)));
        }
        reader.Close();

        return user;
    }

    public void AddUser(User user)
    {
        CheckConnection.checkConnection(Connector);
        string query = queryAddUser(user);
        NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);
        cmd.ExecuteNonQuery();
    }

    public void DelUser(User user)
    {
        CheckConnection.checkConnection(Connector);
        string query = queryDelUser(user);
        NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);
        cmd.ExecuteNonQuery();
    }

    public void UpdateUser(User user)
    {
        CheckConnection.checkConnection(Connector);
        string query = queryUpdateUser(user);
        NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);
        cmd.ExecuteNonQuery();
    }

    public List<User> GetAll()
    {
        CheckConnection.checkConnection(Connector);
        string query = queryGetAll();
        List<User> allUser = new List<User>();
        NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);
        using (NpgsqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {

```

```

        allUser.Add(new User(reader.GetInt32(0), reader.GetString(1),
            reader.GetString(2),
            reader.GetString(3), reader.GetString(4),
            reader.GetString(5), reader.GetString(6),
            (Role)Enum.Parse(typeof(Role), reader.GetString(7))));
    }
    reader.Close();
}
return allUser;
}

public int CountAllUsers()
{
    CheckConnection.checkConnection(Connector);
    string query = queryCountAllUsers();

    NpgsqlCommand cmd = new NpgsqlCommand(query, Connector.Connect);

    return Convert.ToInt32(cmd.ExecuteScalar());
}

```


ПРИЛОЖЕНИЕ Б

Презентация к курсовой работе