

НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ УНИВЕРСИТЕТ “СИНЕРГИЯ”»  
Факультет Информационных технологий  
Кафедра Цифровой экономики

**ОТЧЕТ**  
**о прохождении производственной практики**  
**по профессиональному модулю**  
**ПМ.02 Осуществление интеграции программных модулей**  
**в период с «08» июня 2025 г. по «21» июня 2025 г.**  
**Специальность 09.02.07 Информационные системы и**  
**программирование**

**ФИО обучающегося:** Рыжова Елизавета Романовна  
**Группа:** ДКИП – 205 прог  
**ФИО Руководителя:** Сибирев Иван Валерьевич



# Содержание

1. Инструктаж по соблюдению правил противопожарной безопасности, правил охраны труда, техники безопасности, санитарно-эпидемиологических правил и гигиенических нормативов
2. Ознакомление с инструментальными средствами
3. Сбор информации об объекте практики и анализ содержания источников
4. Экспериментально-практическая работа. Приобретение необходимых умений и практического опыта работы по специальности в рамках освоения вида деятельности ВД 2. Осуществление интеграции программных модулей
5. Обработка и систематизация полученного фактического материала



# Организационный этап

Пример заполнения

Правила внутреннего распорядка, правила и нормы охраны труда, техники безопасности при работе с вычислительной техникой

Я, Рыжова Елизавета Романовна, проходила производственную практику на базе ООО СИМУЛТЕХ

При выполнении индивидуального задания по практике решал кейс: КЛАСС КОТОРЫЙ ВЫПОЛНЯЕТ ФУНКЦИЮ МАРШРУТИЗАТОРА.

Перед началом практики:

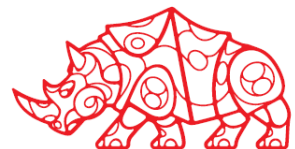
- Принял(а) участие в организационном собрании по практике.
- Ознакомил(а)сь с комплектом шаблонов отчетной документации по практике.
- Уточнил(а) контакты руководителя практики от Образовательной/ Профильной организации, а также правила в отношении субординации, внешнего вида, графика работы, техники безопасности:

Требования к внешнему виду: Официально-деловой стиль

График работы: не более 8 часов в день, 5 дней

Круг обязанностей: Практикант

Доступ к данным: Ограниченный

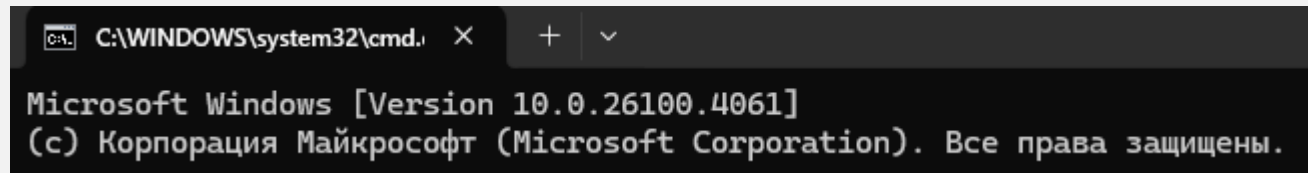
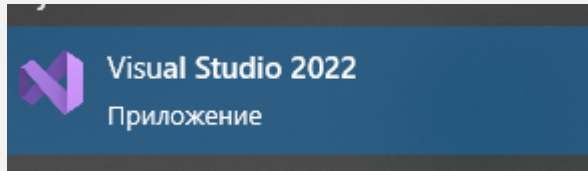


# Организационный этап

Пример заполнения

## Ознакомление с ПО

Ознакомиться с инструментальными средствами для выполнения производственной практики и осуществить предустановку программного обеспечения.

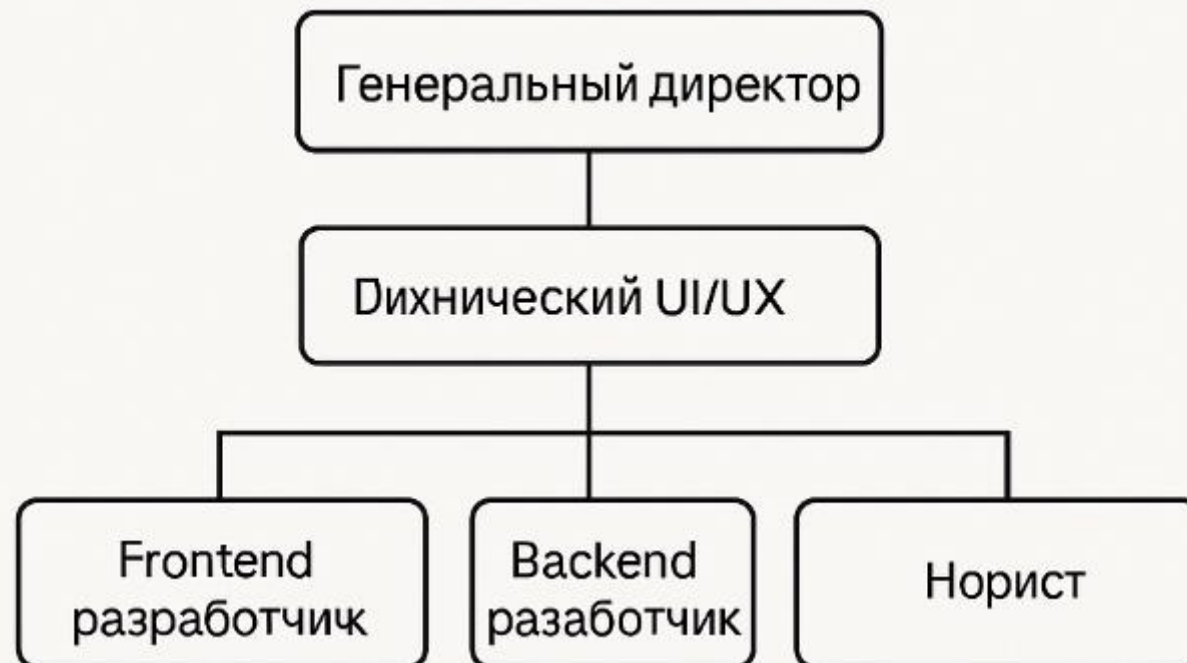


# Организационный этап

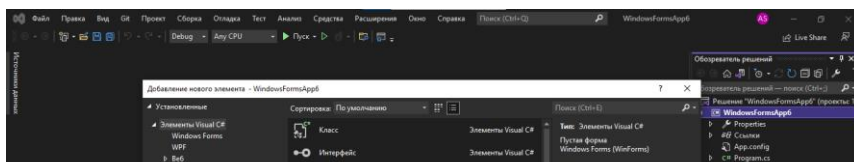
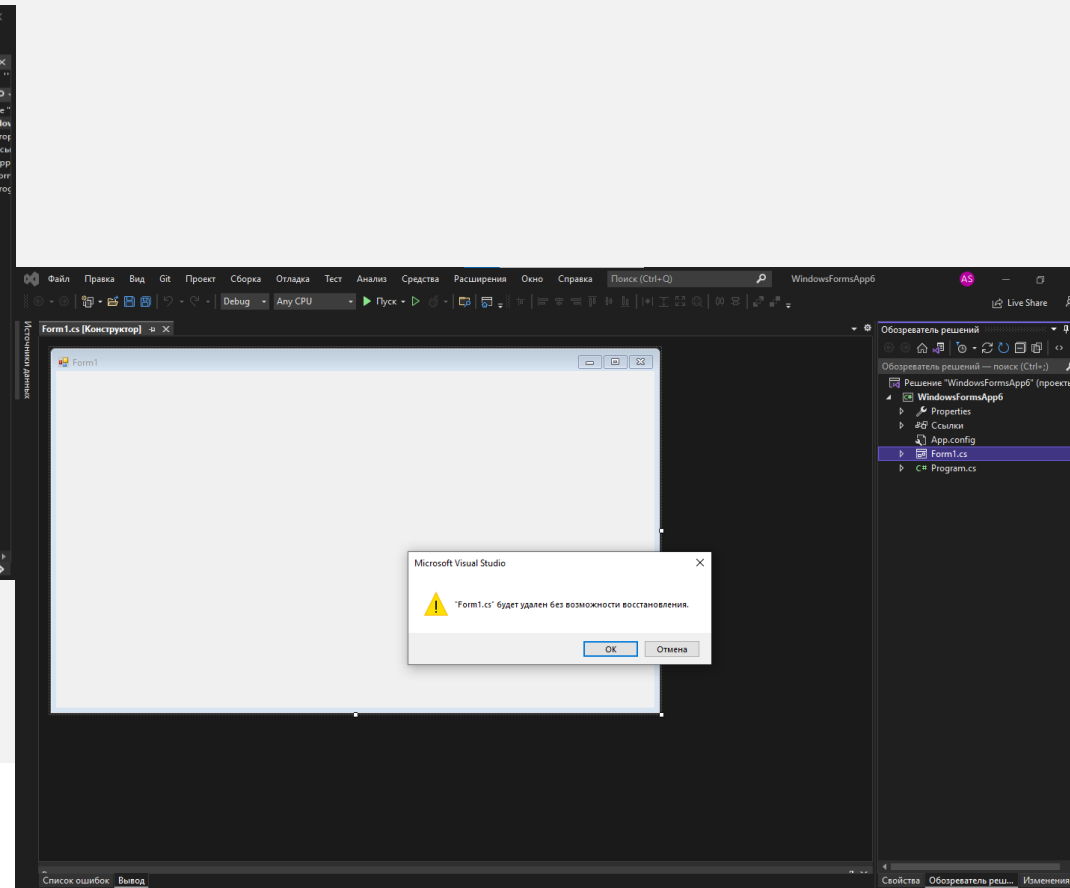
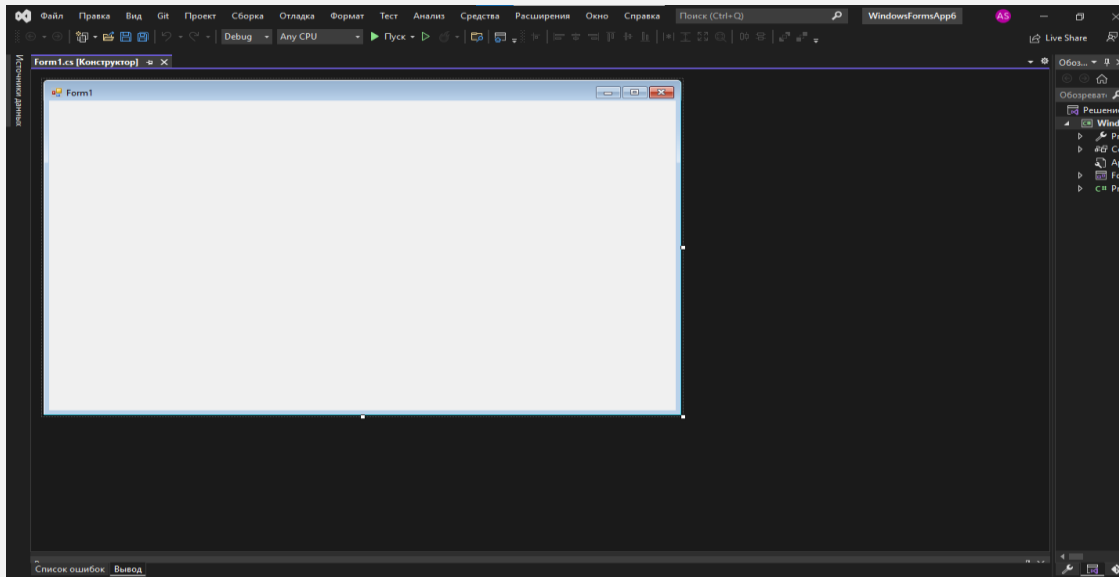
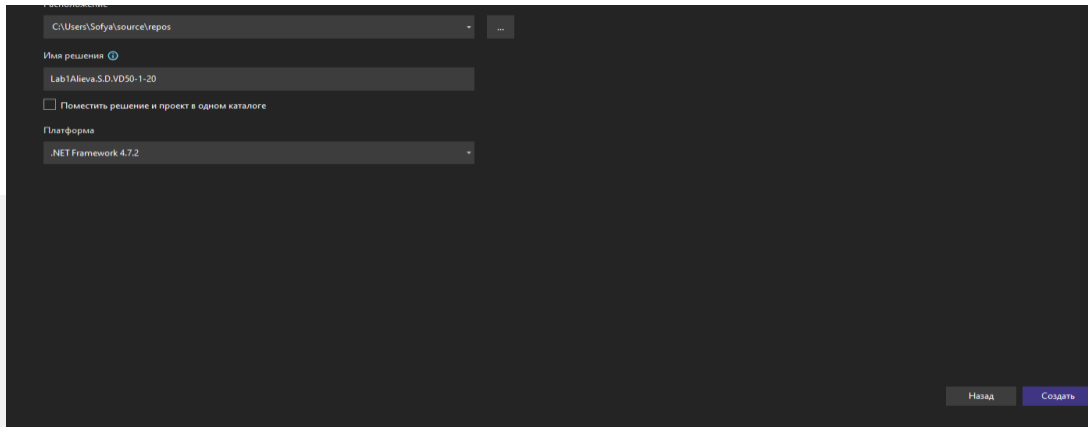
Пример заполнения

Сбор информации об объекте практики и анализ содержания источников

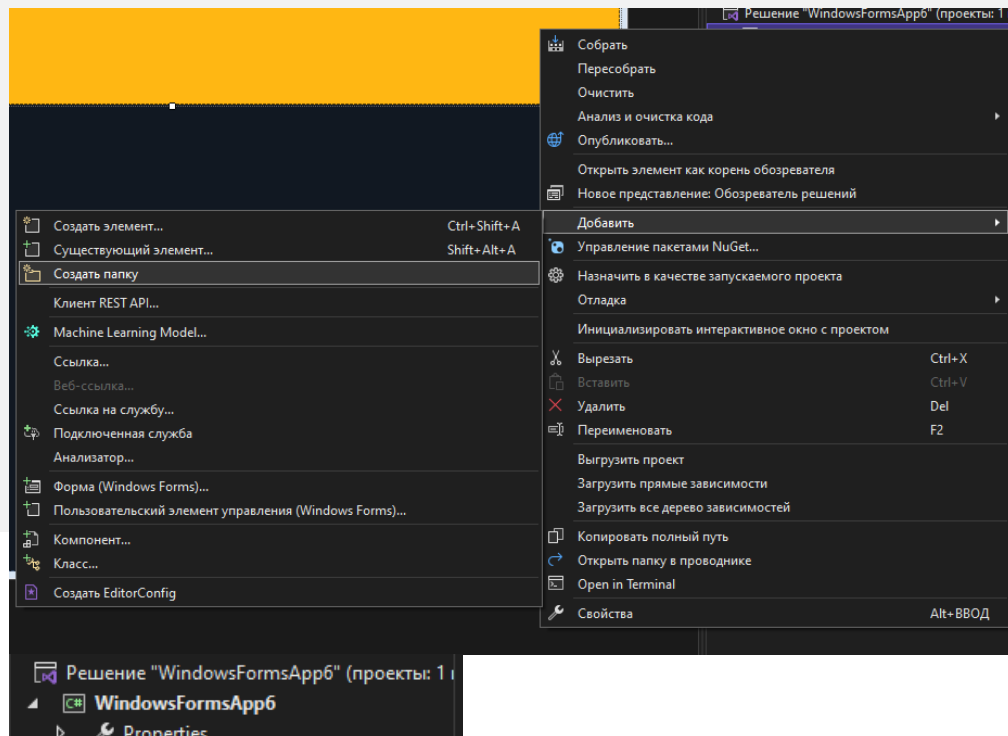
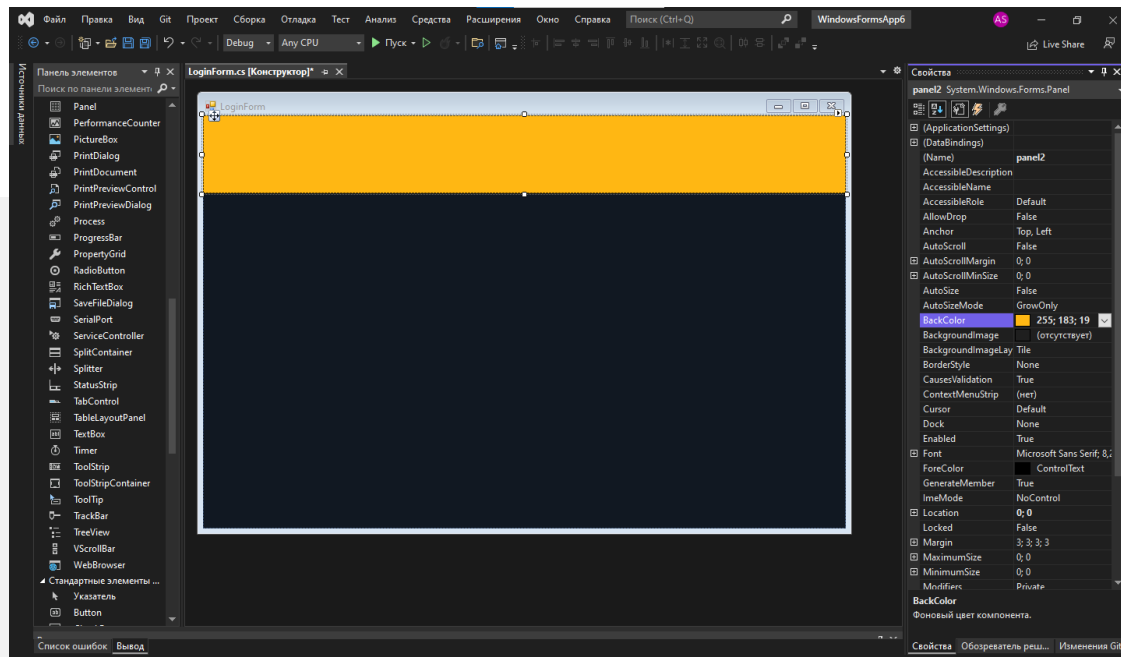
Спроектировать организационную структуру и описать выбранную предметную область.



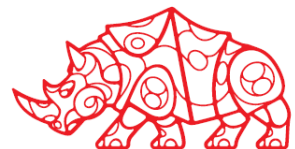
- Создание проекта



# Меняем значения свойства BackColor элемента Panel на 255;183;19



Создаем папку с изображениями под названием image, копируем изображения и вставляем в эту папку.

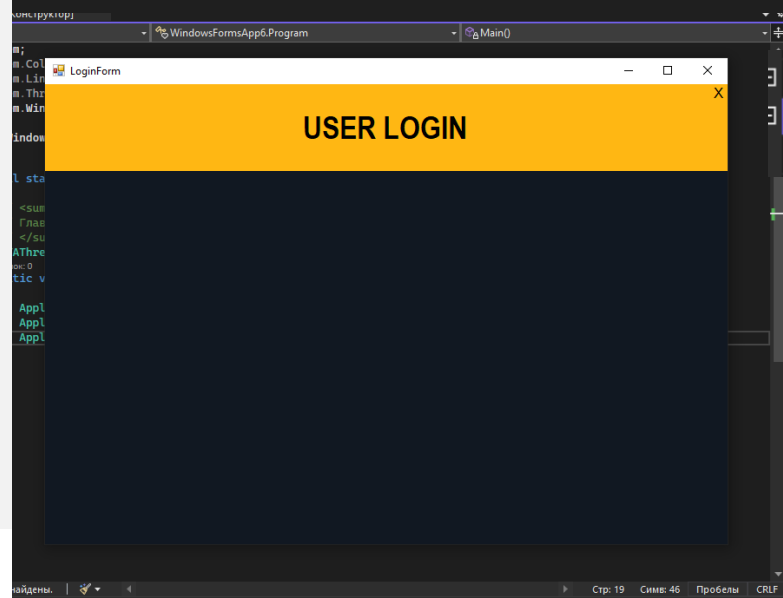
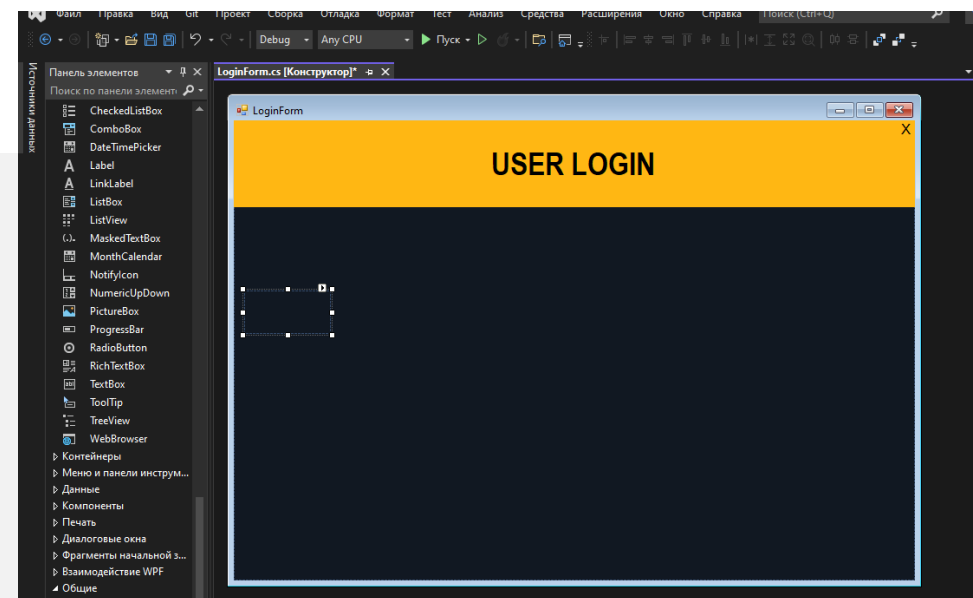


## Заходим в Program.cs и меняем Form1 на LoginForm, чтобы при включении приложения загружалась LoginForm

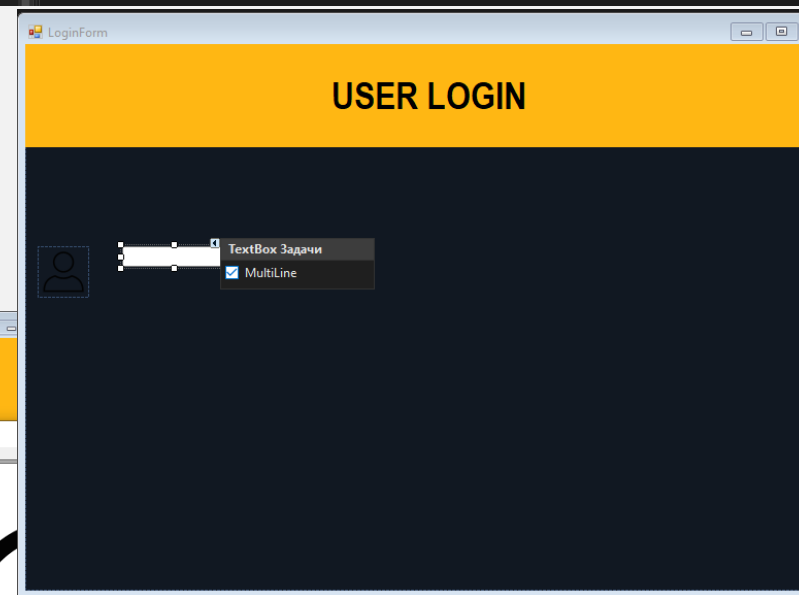
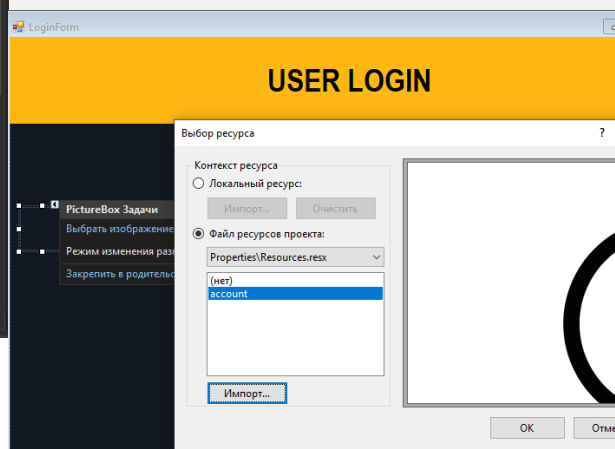
```
Project  Build  Debug  Test  Analysis  Tools  Extensions  Window  Help  WindowsFormsApp6
Debug  Any CPU  Start  Stop  Break  Run  Run and Debug  Live Share

Program.cs  LoginForm.cs [Конструктор]
WindowsFormsApp6  WindowsFormsApp6.Program  Main0

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace WindowsFormsApp6
8  {
9      internal static class Program
10     {
11         /// <summary>
12         /// Главная точка входа для приложения.
13         /// </summary>
14         [STAThread]
15         static void Main()
16         {
17             Application.EnableVisualStyles();
18             Application.SetCompatibleTextRenderingDefault(false);
19             Application.Run(new LoginForm());
20         }
21     }
22
23
```



Padding	0; 0; 0; 0
Size	50; 50
SizeMode	Normal





- Копируем сделанный pictureBox и textBox и меняем изображение в pictureBox на замок

LoginForm

USER LOGIN

Do

lock

button1

Выбор ресурса

Контекст ресурса

Локальный ресурс

Импортировать

Очистить

Файл ресурсов проекта

Properties/Resources.resx

(нет)

account

account1

Импортировать

Открыть

WindowsFormsApp6

images

Поиск в: images

Упорядочить

Новая папка

Быстрый доступ

Рабочий стол

Загрузки

Документы

Изображения

Лайбы

BD50-1-20 Acces

Db aviastrouteinc

repos

ОАиП Лаборато

OneDrive

account

lock

Имя файла: lock

Файлы изображений (\*.gif;\*.jpg)

Открыть

Отмена

LoginForm

USER LOGIN

Do

lock

button1

BackColor

242; 103; 46

FlatAppearance

Flat

Font

Microsoft Sans Serif

Шрифт

Шрифт

Microsoft Sans Serif

обычный

20

OK

Отмена

Видоизменение

Зачеркнутый

Подчеркнутый

Образец

AaBbBbФф

Набор символов

Кириллица

ForeColor

White

General

Другой

Интернет

Система

Image

Transparent

Black

DimGray

Gray

DarkGray

Silver

LightGray

Gainsboro

WhiteSmoke

White

RosyBrown

IndianRed

FlatAppearance

BorderColor

BorderSize

0

MouseDownBackC

MouseOverBackCo

LoginForm

USER LOGIN

Do

lock

button1

FlatAppearance

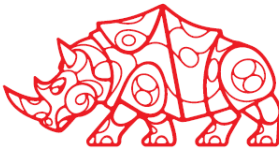
BorderSize

0

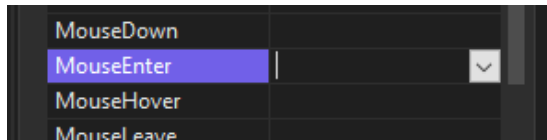
FlatAppearance

BorderSize

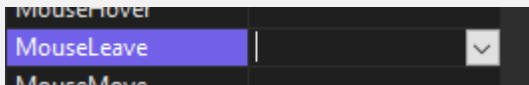
0



Заходим в события labelClose и находим MouseEnter и MouseLeave и задаем цвета, чтобы label был белым, а при наведении курсора черным

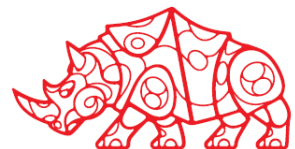


```
Ссылка: 1
private void labelClose_MouseEnter(object sender, EventArgs e)
{
    labelClose.ForeColor = Color.Black; labelClose.Location = new Point(
}
```

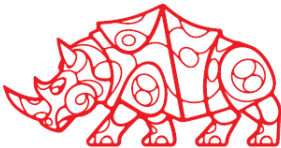
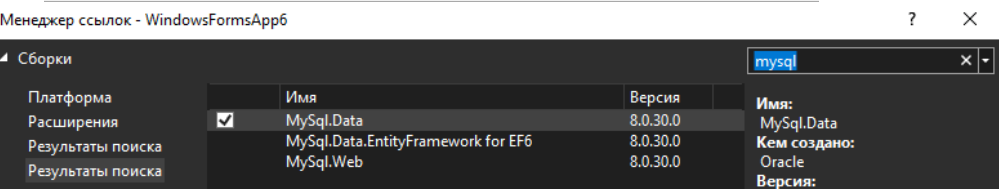
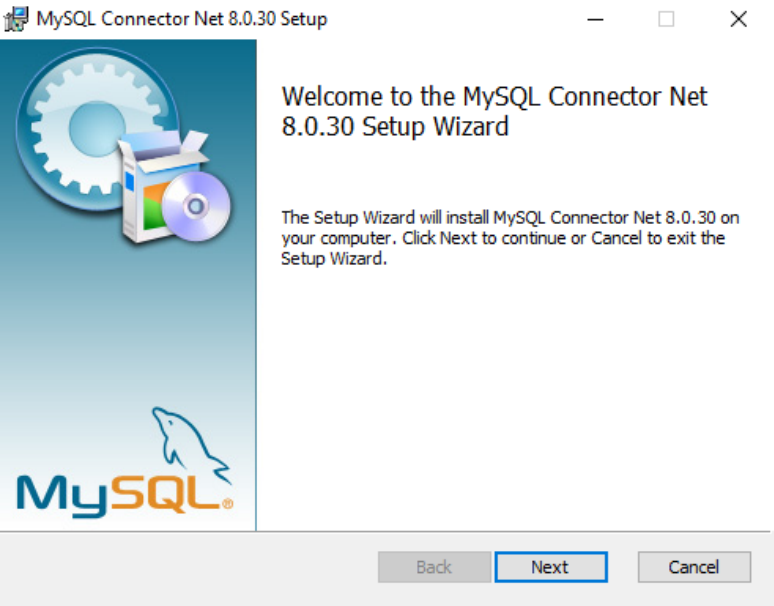
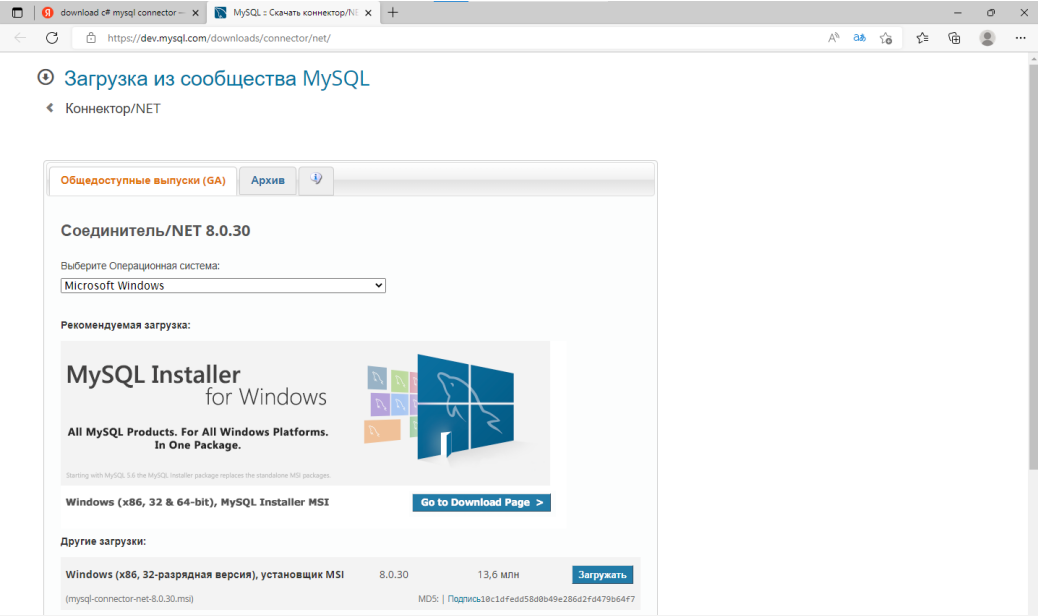


```
Ссылка: 1
private void labelClose_MouseLeave(object sender, EventArgs e)
{
    labelClose.ForeColor = Color.White;
}
```

Готовая форма

A screenshot of a web form titled 'USER LOGIN' in a yellow header bar. The form has a dark blue background. It contains two input fields: the first is preceded by a person icon, and the second by a padlock icon. Below the input fields is an orange button with the text 'LOGIN'. A close button (X) is in the top right corner of the header.

# скачиваем mysql connector



# Заполняем в таблице атрибуты

Имя таблицы:     Добавить  поле(я)

Структура ⓘ

Имя	Тип ⓘ	Длина/Значения ⓘ	По умолчанию ⓘ	Сравнение	Атрибуты	Null	Ис...
<input type="text" value="id"/>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="Нет"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	Pl
<input type="text" value="username"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="100"/>	<input type="text" value="Нет"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--
<input type="text" value="password"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="100"/>	<input type="text" value="Нет"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	--

```
private MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=;database=csharp_users_db");
```

Добавление нового элемента - WindowsFormsApp6

Установленные

Элементы Visual C#

- Windows Forms
- WPF
- Веб
  - Данные
  - Код
  - Общие
- AspNetCore

В сети

Сортировка: По умолчанию

Класс

Интерфейс

Форма (Windows Forms)

Пользовательский элемент управления (Wi...

Класс компонента

Пользовательский элемент управления (W...

JSX-файл TypeScript

Machine Learning Model (ML.NET)

XML-файл

XSLT-файл

База данных, основанная на службах

Визуализатор отладчика

Генератор EF 5.x DbContext

Генератор EF 6.x DbContext

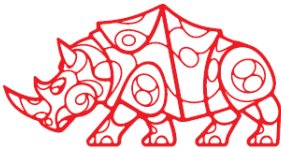
Поиск (Ctrl+E)

Тип: Элементы Visual C#  
Пустое определение класса

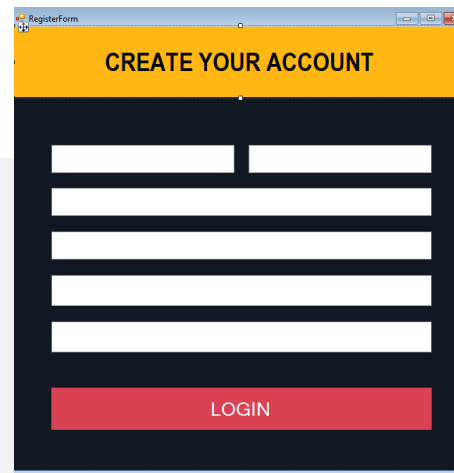
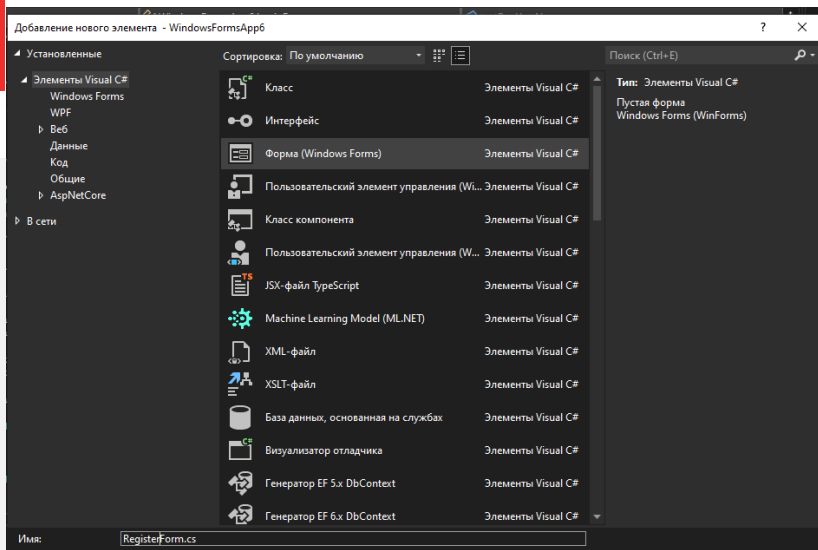
Имя:

```
using MySql.Data.MySqlClient;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

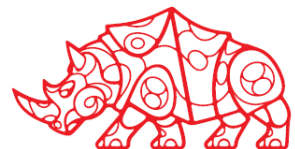
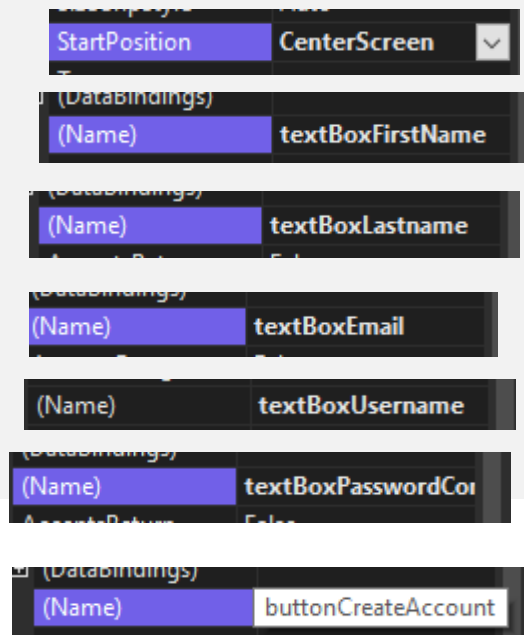
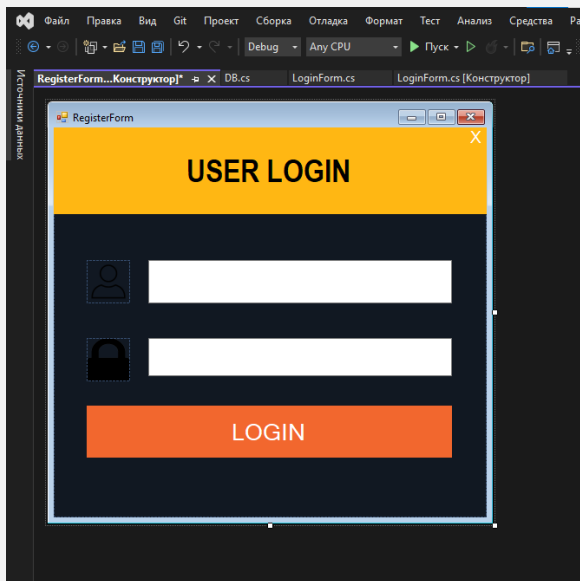
namespace WindowsFormsApp6
{
    Ссылка: 2
    internal class DB
    {
        private MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=;database=csharp_users_db");
    }
}
```



# Добавляем новую форму в проект



```
Ссылка: 0
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new RegisterForm());
}
```



Для textBoxFirstName пишем функции, чтобы при нажатии на данное поле надпись скрывалась, если поле остается пустым, то надпись опять появляется

```
Ссылка: 1
private void textBoxFirstName_Enter(object sender, EventArgs e)
{
    String fname = textBoxFirstName.Text;
    if(fname.ToLower().Trim().Equals("first name"))
    {
        textBoxFirstName.Text = "";
        textBoxFirstName.ForeColor = Color.Black;
    }
}
```

```
Ссылка: 1
private void textBoxFirstName_Leave(object sender, EventArgs e)
{
    String fname = textBoxFirstName.Text;
    if (fname.ToLower().Trim().Equals("first name") || fname.Trim().Equals(""))
    {
        textBoxFirstName.Text = "first name";
        textBoxFirstName.ForeColor = Color.Gray;
    }
}
```

```
Ссылка: 1
public RegisterForm()
{
    InitializeComponent();
    this.ActiveControl = label1;
}
```

```
Ссылка: 1
private void textBoxLastName_Enter(object sender, EventArgs e)
{
    String lname = textBoxLastName.Text;
    if (lname.ToLower().Trim().Equals("last name"))
    {
        textBoxLastName.Text = "";
        textBoxLastName.ForeColor = Color.Black;
    }
}
```

```
Ссылка: 1
private void textBoxLastName_Leave(object sender, EventArgs e)
{
    String lname = textBoxLastName.Text;
    if (lname.ToLower().Trim().Equals("last name") || lname.Trim().Equals(""))
    {
        textBoxLastName.Text = "last name";
        textBoxLastName.ForeColor = Color.Gray;
    }
}
```

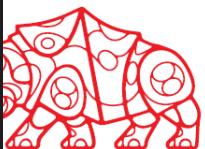
```
Ссылка: 1
private void textBoxEmail_Enter(object sender, EventArgs e)
{
    String email = textBoxEmail.Text;
    if (email.ToLower().Trim().Equals("email address"))
    {
        textBoxEmail.Text = "";
        textBoxEmail.ForeColor = Color.Black;
    }
}
```

```
Ссылка: 1
private void textBoxEmail_Leave(object sender, EventArgs e)
{
    String email = textBoxEmail.Text;
    if (email.ToLower().Trim().Equals("email address") || email.Trim().Equals(""))
    {
        textBoxEmail.Text = "email address";
        textBoxEmail.ForeColor = Color.Gray;
    }
}
```

```
Ссылка: 1
private void textBoxUsername_Enter(object sender, EventArgs e)
{
    String username = textBoxUsername.Text;
    if (username.ToLower().Trim().Equals("username"))
    {
        textBoxUsername.Text = "";
        textBoxUsername.ForeColor = Color.Black;
    }
}
```

```
Ссылка: 1
private void textBoxUsername_Leave(object sender, EventArgs e)
{
    String username = textBoxUsername.Text;
    if (username.ToLower().Trim().Equals("username") || username.Trim().Equals(""))
    {
        textBoxUsername.Text = "username";
        textBoxUsername.ForeColor = Color.Gray;
    }
}
```

```
Ссылка: 1
private void textBoxPassword_Enter(object sender, EventArgs e)
{
    String password = textBoxPassword.Text;
    if (password.ToLower().Trim().Equals("password"))
    {
        textBoxPassword.Text = "";
        textBoxPassword.UseSystemPasswordChar = true;
        textBoxPassword.ForeColor = Color.Black;
    }
}
```



Для label, который выполняет функцию закрытия приложения пишем следующие события, чтобы при нажатии на него оно закрывалось, а при наведении label менял цвет на черный

```
Оссылка: 1
private void labelClose_Click(object sender, EventArgs e)
{
    this.Close();
}

Оссылка: 1
private void labelClose_MouseEnter(object sender, EventArgs e)
{
    labelClose.ForeColor = Color.Black;
}

Оссылка: 1
private void labelClose_MouseLeave(object sender, EventArgs e)
{
    labelClose.ForeColor = Color.White;
}
```

FormBorderStyle None  
HelpButton False

CREATE YOUR ACCOUNT

first name

last name

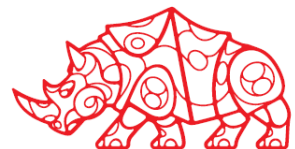
email address

username

••

confirm password

CREATE ACCOUNT



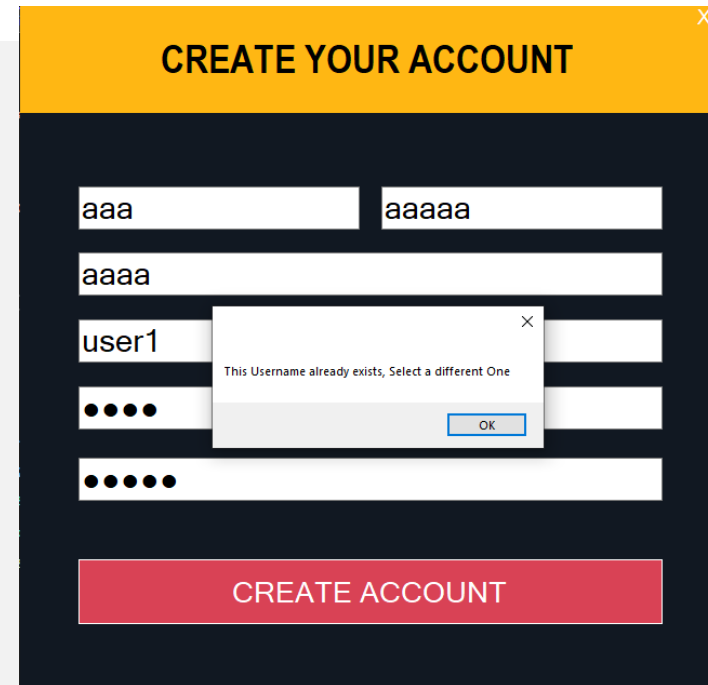
Пишем условие, в котором определяется создан такой пользователь уже или нет, если не создан то выводим месседжбокс, с надписью «account created»

```
if(checkUsername())
{
    MessageBox.Show("This Username already exists, Select a different One");
}
else
{
    if (command.ExecuteNonQuery() == 1)
    {
        MessageBox.Show("ACCOUNT CREATED");
    }
    else
    {
        MessageBox.Show("ERROR");
    }
}
```

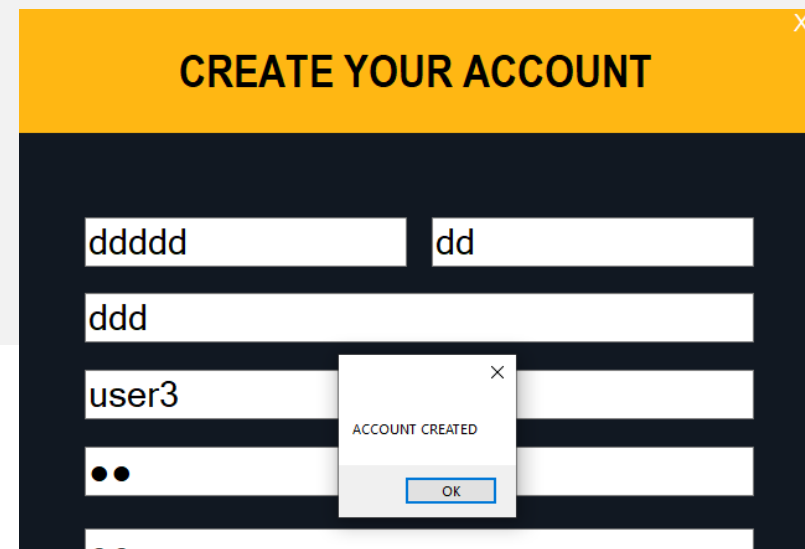
ссылка: 0

```
public Boolean checkTextBoxesValues()
{
    string fname = textBoxFirstName.Text;
    string lname = textBoxLastName.Text;
    string email = textBoxEmail.Text;
    string uname = textBoxUsername.Text;
    string pass = textBoxPassword.Text;

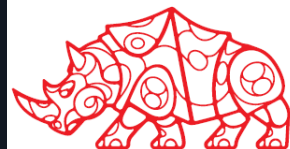
    if(fname.Equals("first name") || lname.Equals("last name") ||
        email.Equals("email address") || uname.Equals("username")
        || pass.Equals("password"))
    {
        return true;
    }
    else
    {
        return false;
    }
}
```



A screenshot of a web application titled "CREATE YOUR ACCOUNT". The form has several input fields: "aaa" for first name, "aaaaa" for last name, "aaaa" for email, "user1" for username, and a password field with four dots. A modal dialog box is displayed over the form with the text "This Username already exists, Select a different One" and an "OK" button. At the bottom of the form is a red button labeled "CREATE ACCOUNT".

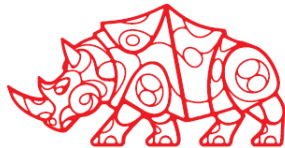
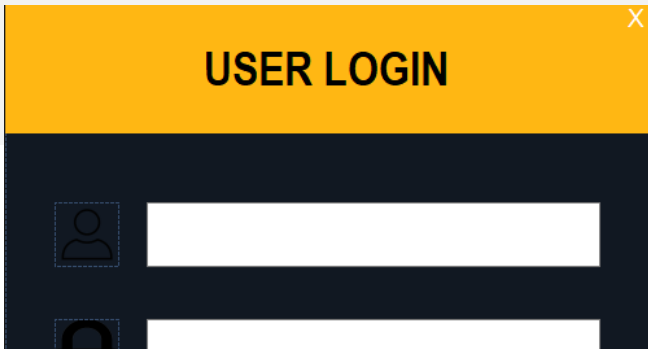
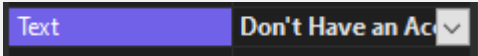
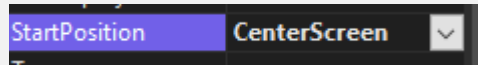
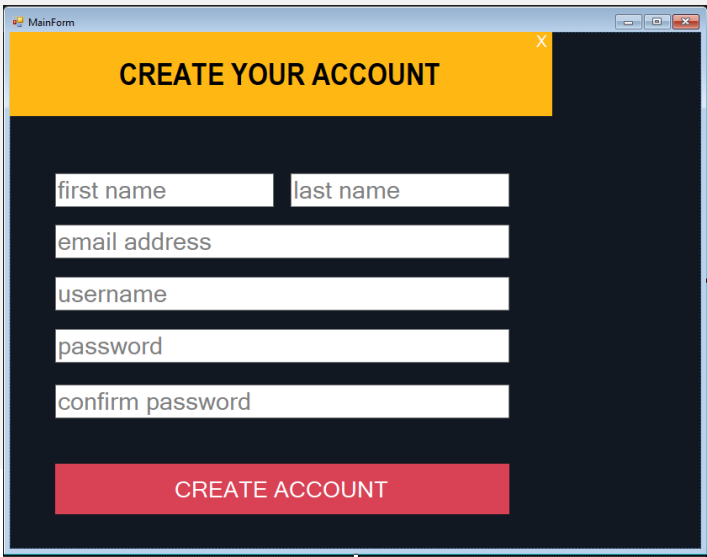
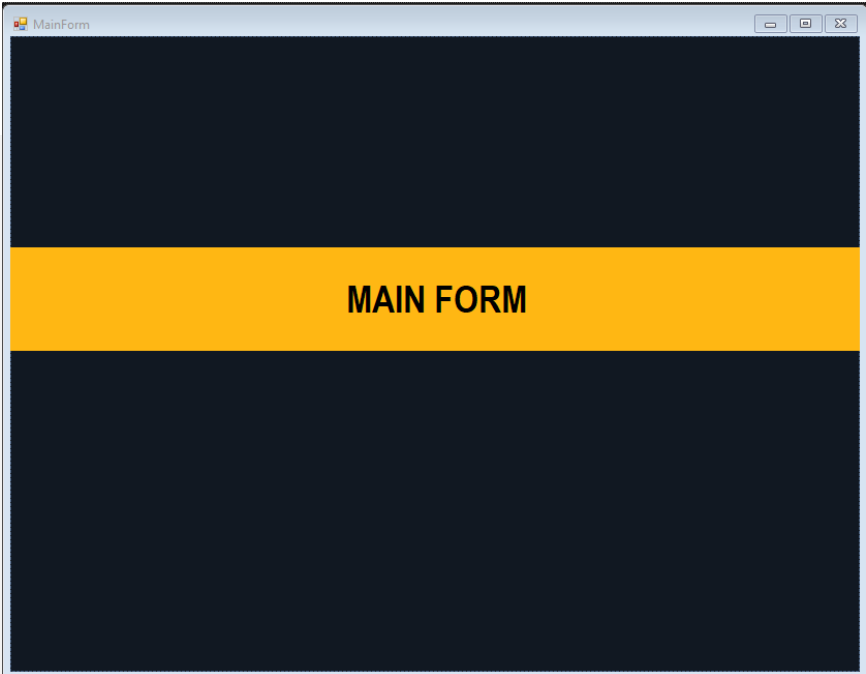
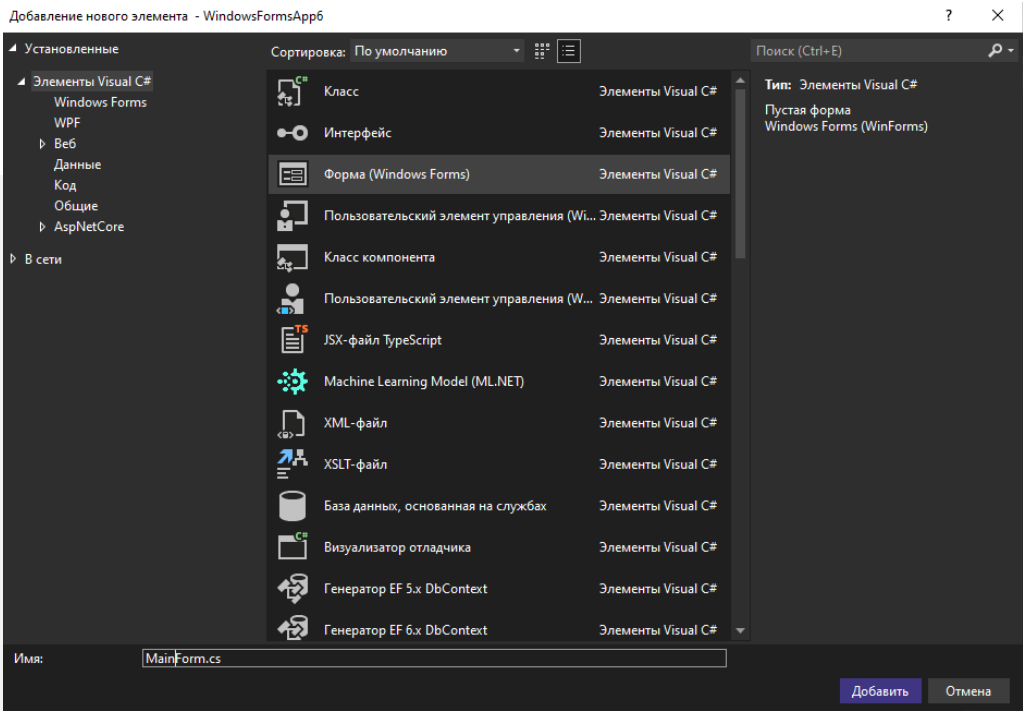


A screenshot of the same "CREATE YOUR ACCOUNT" form. The input fields now contain "dddd" for first name, "dd" for last name, "ddd" for email, "user3" for username, and a password field with two dots. A modal dialog box is displayed with the text "ACCOUNT CREATED" and an "OK" button. The "CREATE ACCOUNT" button is still visible at the bottom.

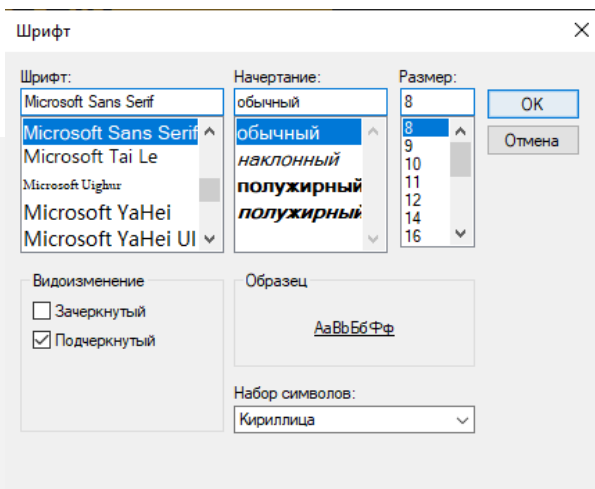




# Создаем новую форму – MainForm и копируем в нее все элементы с RegisterForm.



В настройках шрифта у labelGoSingUp ставим галочку напротив «Подчеркнутый»



(Name)	labelGoToLogin
--------	----------------

```
private void labelGoToLogin_MouseEnter(object sender, EventArgs e)
{
    labelGoToLogin.ForeColor = Color.Yellow;
}

Ссылка: 1
private void labelGoToLogin_MouseLeave(object sender, EventArgs e)
{
    labelGoToLogin.ForeColor = Color.White;
}
```

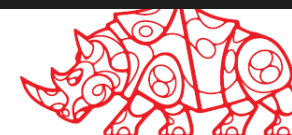
```
private void labelGoToLogin_Click(object sender, EventArgs e)
{
    this.Hide();
    LoginForm loginform = new LoginForm();
    loginform.ShowDialog();
}
```

```
Ссылка: 1
private void labelClose_Click(object sender, EventArgs e)
{
    //this.Close();
    Application.Exit();
}
```

```
if(table.Rows.Count > 0)
{
    this.Hide();
    MainForm mainform = new MainForm();
    mainform.Show();
}
else
```

```
Ссылка: 1
private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    Application.Exit();
}
```

```
else if (password.Trim().Equals(""))
{
    MessageBox.Show("Enter Your Password To Login", "Empty Password", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

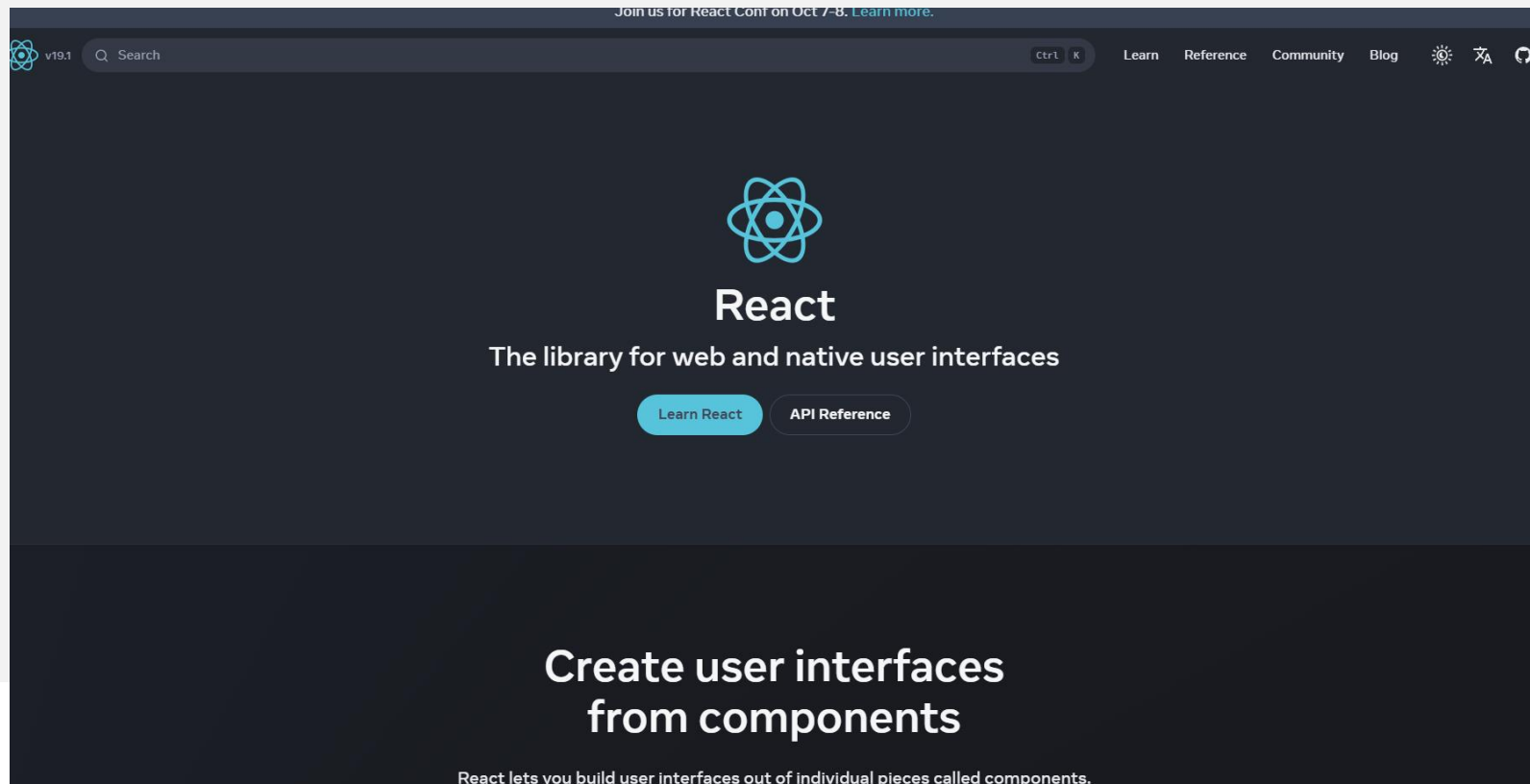


# Этап проектирования

Пример заполнения

## Интеграция модулей в программное обеспечение

На данном слайде необходимо продемонстрировать скриншоты разработанных форм авторизации и регистрации. Разработка дизайна осуществляется самостоятельно в рамках выбранной обучающимся предметной области.



# Этап проектирования

## Пример заполнения

### Отладка программных модулей

#### Создание React-проекта

```
C:\Users\002>npx create-react-app summer-practice
Need to install the following packages:
create-react-app@5.1.0
Ok to proceed? (y) yes

npm warn deprecated uid-number@0.0.6: This package is no longer supported.
npm warn deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported
npm warn deprecated fstream-ignore@1.0.5: This package is no longer supported.
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much
more comprehensive and powerful.
npm warn deprecated fstream@1.0.12: This package is no longer supported.
npm warn deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security updates. Please upgrade asap.
create-react-app is deprecated.

You can find a list of up-to-date React frameworks on react.dev
For more info see:https://react.dev/link/cra

This error message will only be shown once per install.

Creating a new React app in C:\Users\002\summer-practice.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1324 packages in 2m

269 packages are looking for funding
  run 'npm fund' for details

Initialized a git repository.

Installing template dependencies using npm...

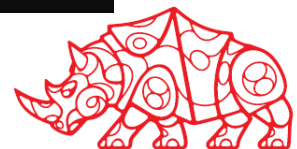
added 18 packages, and changed 1 package in 9s

269 packages are looking for funding
  run 'npm fund' for details
Removing template package using npm...

removed 1 package, and audited 1342 packages in 5s

269 packages are looking for funding
  run 'npm fund' for details

9 vulnerabilities (3 moderate, 6 high)
```



# Этап проектирования

## Отладка программных модулей

```
C:\WINDOWS\system32\cmd.exe
To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.

Created git commit.

Success! Created summer-practice at C:\Users\002\summer-practice
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd summer-practice
  npm start

Happy hacking!

C:\Users\002>cd summer-practice

C:\Users\002\summer-practice>npm start

> summer-practice@0.1.0 start
> react-scripts start

(node:9880) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:9880) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view summer-practice in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.0.19:3000

Note that the development build is not optimized.
To create a production build, use npm run build.
```



# Этап проектирования

## Отладка программных модулей

```
C:\WINDOWS\system32\cmd.  X  +  v

Compiled successfully!

You can now view summer-practice in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.0.19:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
Compiling...
Compiled successfully!

You can now view summer-practice in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.0.19:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

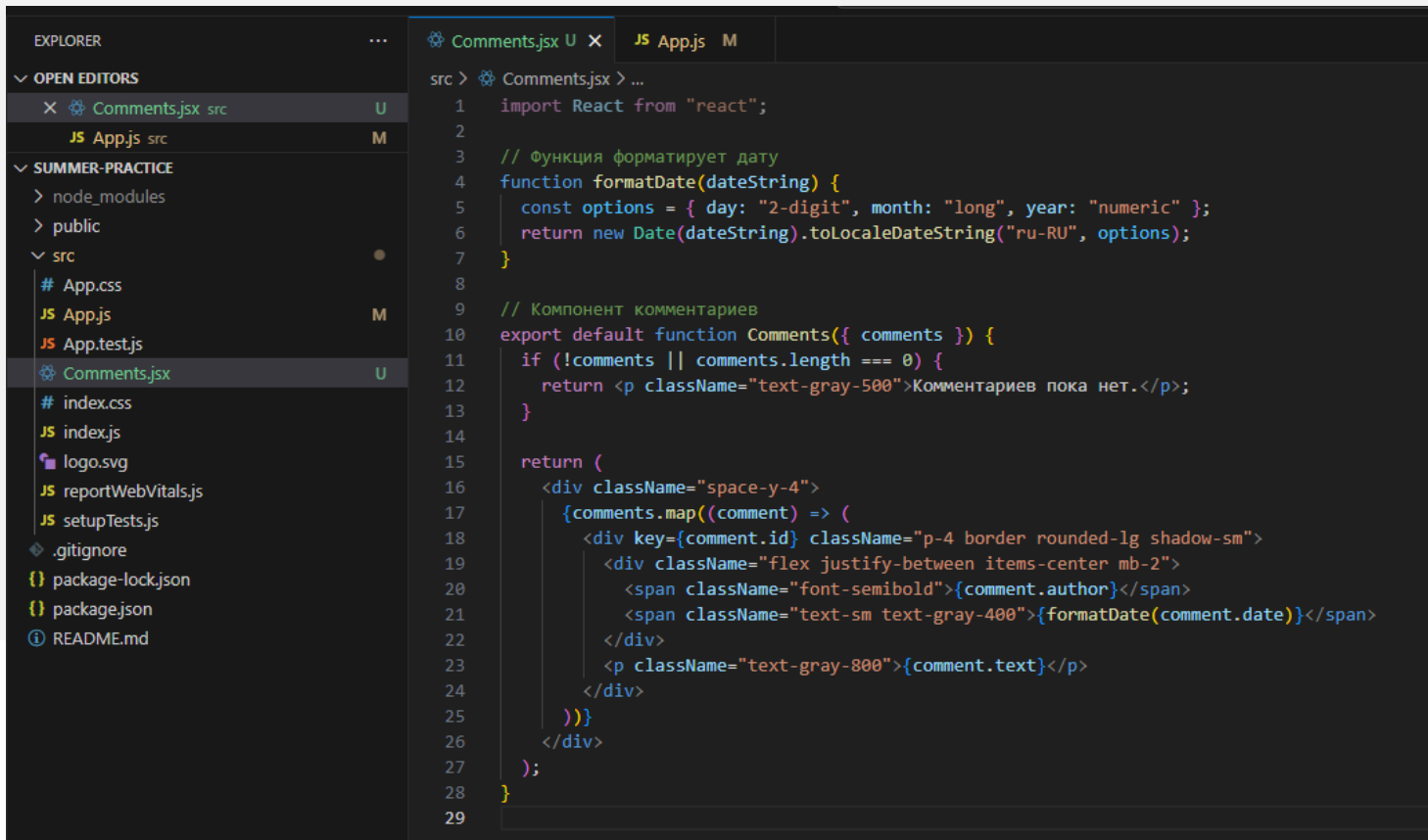
webpack compiled successfully
```



# Этап проектирования

Отладка программных модулей

Создание файла Comments.jsx в VS code



The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor on the right. The Explorer sidebar shows the project structure with the following files and folders:

- OPEN EDITORS
  - Comments.jsx src U
  - JS App.js src M
- SUMMER-PRACTICE
  - node\_modules
  - public
  - src
    - App.css
    - App.js M
    - App.test.js
    - Comments.jsx U
    - index.css
    - index.js
    - logo.svg
    - reportWebVitals.js
    - setupTests.js
  - .gitignore
  - package-lock.json
  - package.json
  - README.md

The Editor shows the content of Comments.jsx:

```
src > Comments.jsx > ...
1  import React from "react";
2
3  // Функция форматирует дату
4  function formatDate(dateString) {
5    const options = { day: "2-digit", month: "long", year: "numeric" };
6    return new Date(dateString).toLocaleDateString("ru-RU", options);
7  }
8
9  // Компонент комментариев
10 export default function Comments({ comments }) {
11   if (!comments || comments.length === 0) {
12     return <p className="text-gray-500">Комментариев пока нет.</p>;
13   }
14
15   return (
16     <div className="space-y-4">
17       {comments.map((comment) => (
18         <div key={comment.id} className="p-4 border rounded-lg shadow-sm">
19           <div className="flex justify-between items-center mb-2">
20             <span className="font-semibold">{comment.author}</span>
21             <span className="text-sm text-gray-400">{formatDate(comment.date)}</span>
22           </div>
23           <p className="text-gray-800">{comment.text}</p>
24         </div>
25       ))}
26     </div>
27   );
28 }
29
```



# Этап проектирования

Отладка программных модулей

Использование компонента в App.js

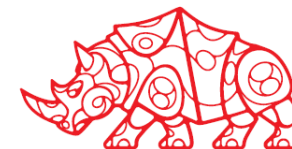




### Формирование отчетной документации по результатам работ

#### При оформлении отчетных материалов следует придерживаться действующих стандартов.

- В соответствии с ГОСТ 2.105-79 «Общие требования к текстовым документам» иллюстрации (графики, схемы, диаграммы) могут быть приведены как в основном тексте, так и в приложении. Все иллюстрации именуют рисунками. Все рисунки, таблицы и формулы нумеруют арабскими цифрами последовательно (сквозная нумерация) или в пределах раздела (относительная нумерация). В приложении - в пределах приложения. Каждый рисунок должен иметь подрисуночную подпись - название, помещаемую под рисунком.
- Рисунки следует размещать так, чтобы их можно было рассматривать без поворота страницы. Если такое размещение невозможно, рисунки следует располагать так, чтобы для просмотра надо было повернуть страницу по часовой стрелке. В этом случае верхним краем является левый край страницы. Расположение и размеры полей сохраняются.
- Номер таблицы размещают в правом верхнем углу или перед заголовком таблицы, если он есть. Заголовок, кроме первой буквы, выполняют строчными буквами. Ссылки на таблицы в тексте пояснительной записки указывают в виде слова «табл.» и номера таблицы. *Например: Результаты тестов приведены в табл. 4.*



### Формирование отчетной документации по результатам работ

#### **При оформлении отчетных материалов следует придерживаться действующих стандартов.**

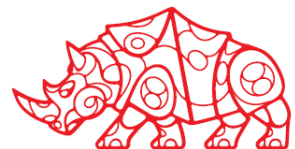
- Список литературы должен включать все использованные источники. Сведения о книгах (монографиях, учебниках, пособиях, справочниках и т.д.) должны содержать: фамилию и инициалы автора, заглавие книги, место издания, издательство, год издания. При наличии трех и более авторов допускается указывать фамилию и инициалы только первого из них со словами «и др.». Издательство надо приводить полностью в именительном падеже: допускается сокращение названия только двух городов: Москва (М.) и Санкт-Петербург (СПб.).
- Сведения о статье из периодического издания должны включать: фамилию и инициалы автора, наименование статьи, издания (журнала), серии (если она есть), год выпуска, том (если есть), номер издания (журнала) и номера страниц, на которых помещена статья.
- При ссылке на источник из списка литературы (особенно при обзоре аналогов) надо указывать порядковый номер по списку литературы, заключенный в квадратные скобки; например: [5].



Выводы о результатах прохождения производственной практики:  
выполняемая работа, приобретенные умения и навыки

### Подведите итоги прохождения производственной практики:

В рамках производственной практики мной был реализован компонент `<comments />` для отображения списка комментариев. Компонент отображает автора, текст и дату каждого комментария. Данные передаются через props и обрабатываются внутри компонента. Также была реализована функция форматирования даты в читаемый вид. Проект позволяет отработать навыки работы с React-компонентами, передачей данных, форматированием и базовой стилизацией интерфейса.



### Список используемой литературы

1. Советов, Б. Я. Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 420 с. — (Профессиональное образование). — ISBN 978-5-534-09324-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/514585>
2. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2023. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518499>
3. Нагаева, И.А. Основы алгоритмизации и программирования: практикум : [12+] / И.А. Нагаева, И.А. Кузнецов. — Москва ; Берлин : Директ-Медиа, 2021. — 169 с. : схем. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/>
4. Зыков, С. В. Архитектура информационных систем. Основы проектирования : учебник для среднего профессионального образования / С. В. Зыков. — Москва : Издательство Юрайт, 2025. — 260 с. — (Профессиональное образование). — ISBN 978-5-534-21539-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/575501>
5. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 258 с. — (Высшее образование). — ISBN 978-5-534-18107-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/536687>



# Отчетный этап

## Приложения

Пример заполнения

1.1. Создание форм авторизации и регистрации.docx

1.2. Валидация кода.docx

