

Тестовое веб приложение

EXAM_ASP

[Home](#)

[Privacy](#)

123

[Ваши результаты](#)

[Добавить нового пользователя](#)

[Редактировать тесты](#)

[Добавить новый тест](#)

[Выход](#)

Столица Росии?

Кто мяукает?

- ☐ Собака
- ☐ Кот
- ☐ Корова
- ☐ Кит

Млекопитающие?

- ☐ Тунец
- ☐ Жаба
- ☐ Собака
- ☐ Кот

В каком классе получают аттестат?

- ☐ 9
- ☐ 8
- ☐ 1
- ☐ 11
- ☐ 7

[Завершить тест](#)

Самойлов Андрей

Титульный лист

1. Титульный лист стр.2
2. Кратко о приложении стр.3
3. Диаграммы и схемы стр.4-6
4. Реализация входа, регистрации стр.7-10
5. Реализация добавления нового пользователя стр.11
6. Реализация основного меню приложения стр.12
7. Реализация тестирования стр.13-15
8. Реализация результатов стр.16
9. Реализация добавления тестов стр.17-19
10. Перспективы развития приложения стр.20

Кратко о программе

Программа для проведения тестирования. Тестовые задания и список пользователей хранятся в базе. Также ведется учет каждого прохождения. Программа должна поддерживать различные виды вопросов. Программа делится на две части: редактор тестов, тестовая система. В редакторе тестов создаются тесты, которые хранятся в базе. Каждый тест состоит из набора вопросов. Вопрос содержит произвольное количество вариантов ответов. Тестовая система воспроизводит тесты для пользователей. Пользователь заходит в систему под своим логином и паролем. Результаты заносятся в базу. Разработанное приложение подключается к удаленной базе (SQL Server 2019) и функционирует на основе информации записанной в ней (тесты, пользователи, прохождения).

Диаграмма вариантов использования (USE Case)

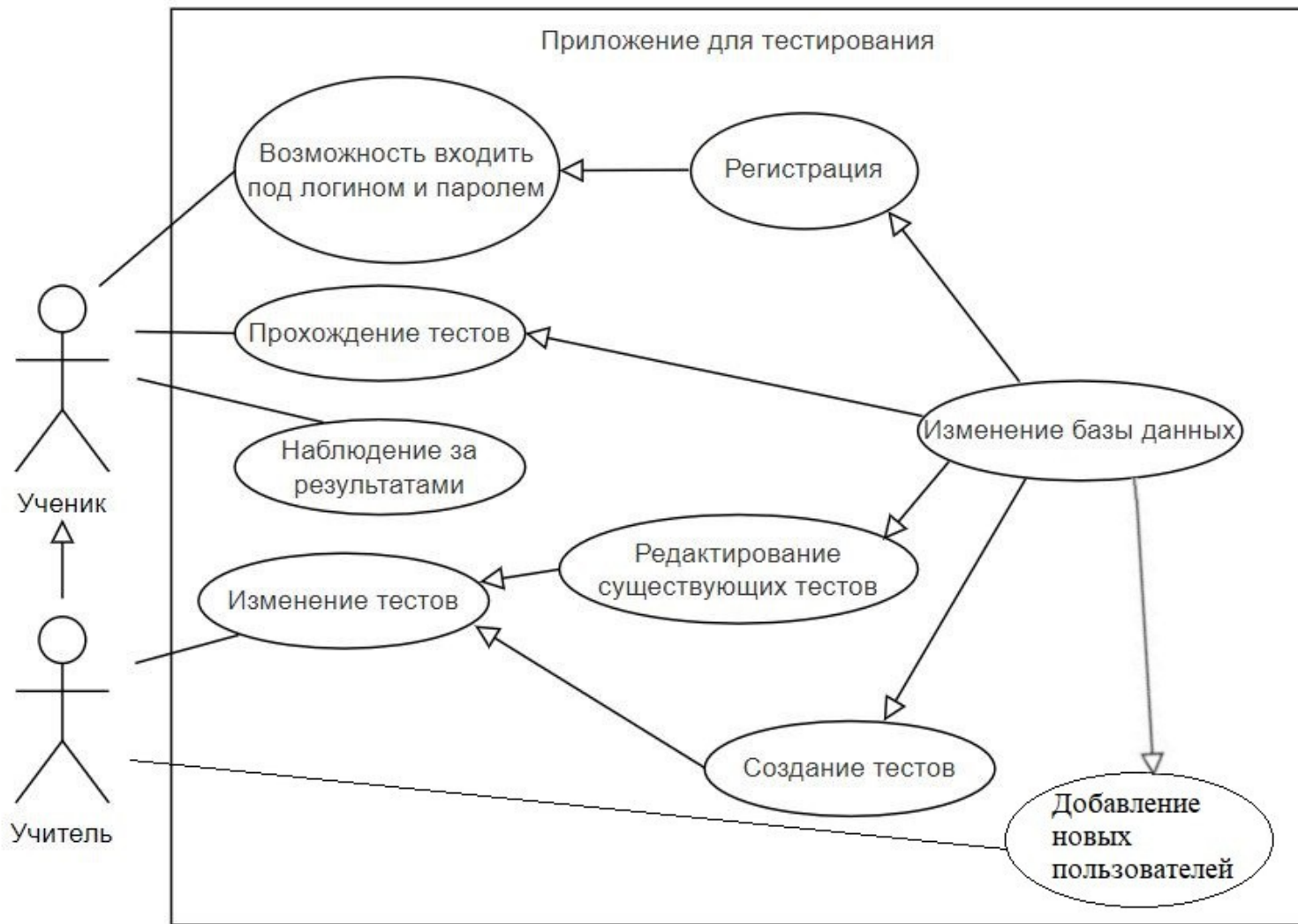


Схема базы данных

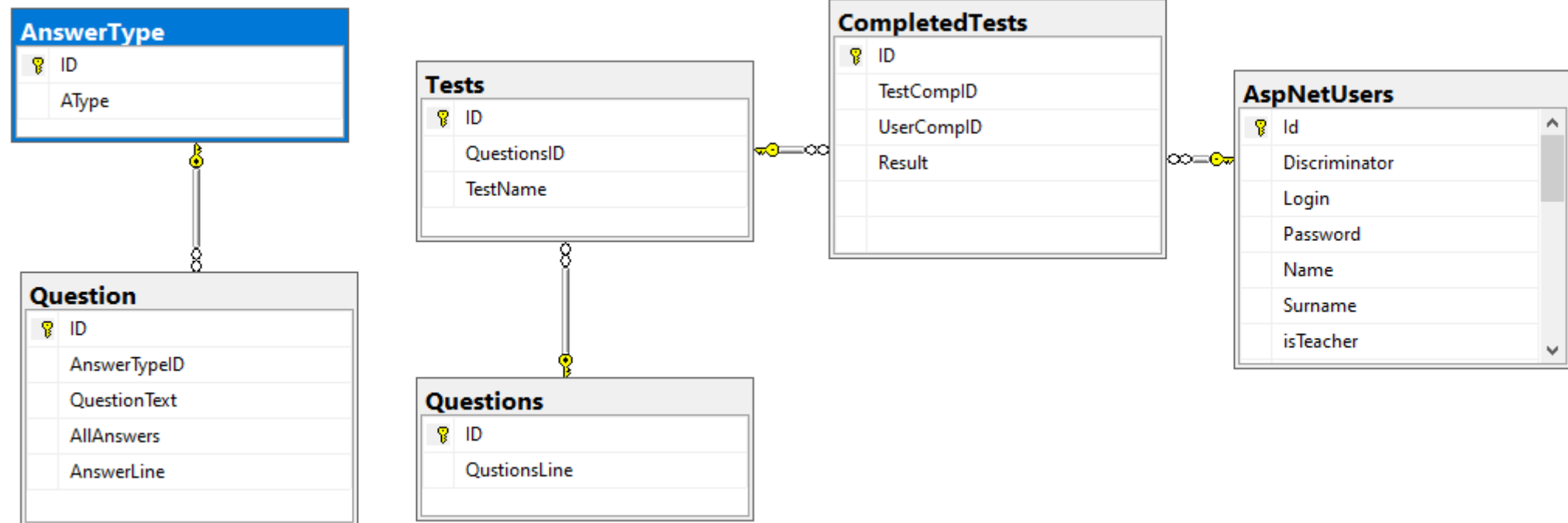
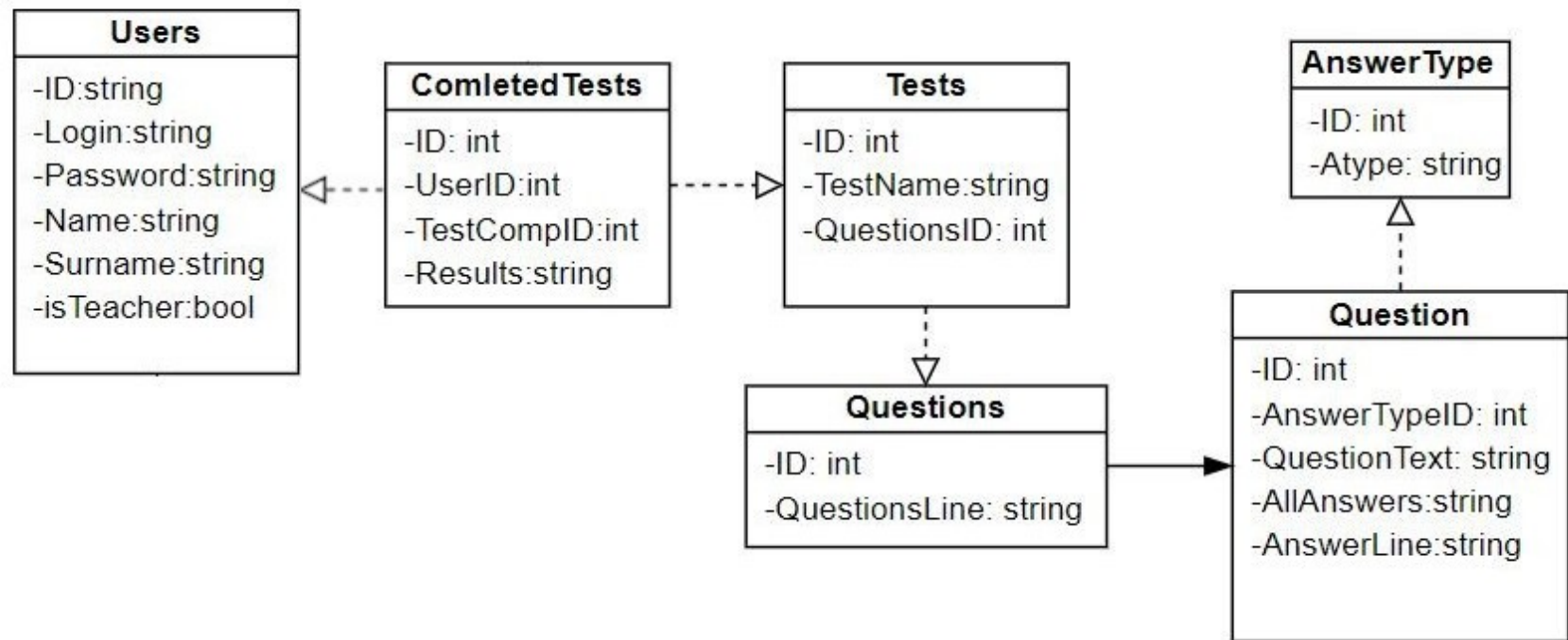


Диаграмма классов



Вход в приложение

EXAM_ASP [Home](#) [Privacy](#) [Вход](#) [Регистрация](#)

Вход в приложение

Login

123

Пароль

.....

Запомнить?

☐

Войти

```
public class LoginViewModel
{
    [Required]
    [Display(Name = "Login")]
    Ссылка: 4
    public string Login { get; set; }

    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Пароль")]
    Ссылка: 4
    public string Password { get; set; }

    [Display(Name = "Запомнить?")]
    Ссылка: 3
    public bool RememberMe { get; set; }

    Ссылка: 5
    public string ReturnUrl { get; set; }
}
```

Вход в приложение: контроллер

```
[HttpPost]
[ValidateAntiForgeryToken]
Ссылка: 0
public async Task<IActionResult> Login(LoginViewModel model)
{
    var result =
        await _signInManager.PasswordSignInAsync(model.Login, model.Password, model.RememberMe
    if (result.Succeeded)
    {
        if (!string.IsNullOrEmpty(model.ReturnUrl) && Url.IsLocalUrl(model.ReturnUrl))
        {
            return Redirect(model.ReturnUrl);
        }
        else
        {
            return RedirectToAction("Index", "Home");
        }
    }
    else
    {
        ModelState.AddModelError("", "Неправильный логин и (или) пароль");
    }

    return View(model);
}
```


Регистрация

EXAM_ASP [Home](#) [Privacy](#) [Вход](#) [Регистрация](#)

Регистрация нового пользователя

Login

Name

Surname

Пароль

Подтвердить пароль

```
public class RegisterViewModel
{
    [Required]
    [Display(Name = "Login")]
    Ссылка: 5
    public string Login { get; set; }

    [Required]
    [Display(Name = "Name")]
    Ссылка: 4
    public string Name { get; set; }

    [Required]
    [Display(Name = "Surname")]
    Ссылка: 4
    public string Surname { get; set; }

    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Пароль")]
    Ссылка: 5
    public string Password { get; set; }

    [Required]
    [Compare("Password", ErrorMessage = "Пароли не совпадают")]
    [DataType(DataType.Password)]
    [Display(Name = "Подтвердить пароль")]
    Ссылка: 3
    public string PasswordConfirm { get; set; }
}
```

Регистрация: контроллер

[HttpPost]

Ссылка: 0

```
public async Task<IActionResult> Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        Users user = new Users { Name = model.Name, Surname = model.Surname, Login = model.Login, Password = model.Password,
        // добавляем пользователя
        var result = await _userManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            // установка куки
            await _signInManager.SignInAsync(user, false);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            foreach (var error in result.Errors)
            {
                ModelState.AddModelError(string.Empty, error.Description);
            }
        }
    }
    return View(model);
}
```

Добавление нового пользователя

EXAM_ASP

[Home](#)

[Privacy](#)

123

[Ваши результаты](#)

[Добавить нового пользователя](#)

[Редактировать тесты](#)

[Добавить новый тест](#)

[Выход](#)

Добавление нового пользователя

Login

Name

Surname

Пароль

Подтвердить пароль

isTeacher

☐

[Регистрация](#)

Основное меню

EXAM_ASP

[Home](#)

[Privacy](#)

123

[Ваши результаты](#)

[Добавить нового пользователя](#)

[Редактировать тесты](#)

[Добавить новый тест](#)

[Выход](#)

EXAM_ASP

[Home](#)

[Privacy](#)

1

[Ваши результаты](#)

[Выход](#)

Test

TESTSTS

Новый Тест

Финальный тест

роро

FT

Прохождение теста

EXAM_ASP

[Home](#)

[Privacy](#)

1

[Ваши результаты](#)

[Выход](#)

Столица России?

Москва

Кто мяукает?

- ☐ Собака
- ☒ Кот
- ☐ Корова
- ☐ Кит

Млекопитающие?

- ☐ Щука
- ☐ Жаба
- ☒ Собака
- ☒ Кот

В каком классе получают аттестат?

- ☒ 9
- ☐ 8
- ☐ 1
- ☒ 11
- ☐ 7

[Завершить тест](#)

```
[Authorize]
[HttpGet]
Ссылка: 0
public IActionResult Testing(int tID)
{
    var optionsBuilder = new DbContextOptionsBuilder<TestSystemEntities>();

    var options = optionsBuilder
        .UseSqlServer(@"Server=DESKTOP-HNAF04G;Database=TestSystemNew;T
        .Options;

    using (TestSystemEntities db = new TestSystemEntities(options))
    {
        Tests test = new Tests();

        foreach (var k in from s in db.Tests select s)
        {
            if (tID == k.ID)
            {
                test = k;
                TestViewModel model = new() { Test = test };
                return View(model);
            }
        }
        return View();
    }
}
```

Прохождение теста: представление и модель

```
public class TestViewModel
{
    public Tests Test = new Tests();

    Ссылка 15
    public string[] AnswersStr { get; set; }

    Ссылка 5
    public bool[][] AnswersCB { get; set; }
}
```

```
Model.AnswersCB = new bool[questionList.Where(q => q.AnswerTypeID == 3).Count()][];
int cb = 0;
@for (int i = 0; i < questionList.Count; i++)
{
    if (questionList[i].AnswerTypeID == 3)
    {
        string[] CAIAnswers = questionList[i].AllAnswers
        .Split(new string[] { ";;;" }, StringSplitOptions.None);
        Model.AnswersCB[cb] = new bool[CAIAnswers.Length];
        cb++;
    }
}
```

```
@for (int i = 0; i < questionList.Count; i++)
{
    <div style="margin:15px">
        <h5>@questionList[i].QuestionText</h5>
        @if (questionList[i].AnswerTypeID == 1)
        {
            @Html.EditorFor(m => m.AnswersStr[i])
        }
        @if (questionList[i].AnswerTypeID == 2)
        {
            string[] CAIAnswers = questionList[i].AllAnswers
            .Split(new string[] { ";;;" }, StringSplitOptions.None);
            @for (int j = 0; j < CAIAnswers.Length; j++)
            {
                @Html.RadioButtonFor(m => Model.AnswersStr[i], j)
                <span>@CAIAnswers[j]</span> <br />
            }
        }
        @if (questionList[i].AnswerTypeID == 3)
        {
            string[] CAIAnswers = questionList[i].AllAnswers
            .Split(new string[] { ";;;" }, StringSplitOptions.None);
            for (int j = 0; j < CAIAnswers.Length; j++)
            {
                @Html.CheckBoxFor(m => m.AnswersCB[cbI][j], false)
                <span>@CAIAnswers[j]</span> <br />
            }
            cbI++;
        }
    </div>
}
```

Прохождение теста: контроллер

```
int cbI = 0;
int res = 0;
List<string> CorrectAnswers = new List<string>();
List<string> Answers = new List<string>();
for (int i = 0; i < qList.Count; i++)
{
    if (qList[i].AnswerTypeID == 2)
    {
        if (model.AnswersStr[i] != "" || model.AnswersStr[i] != null)
            model.AnswersStr[i] = (Convert.ToInt32(model.AnswersStr[i]) + 1).ToString();
        else
        {
            model.AnswersStr[i] = "0";
        }
    }
    if (qList[i].AnswerTypeID == 3)
    {
        string ans = "";
        for (int j = 0; j < model.AnswersCB[cbI].Length; j++)
        {
            if (model.AnswersCB[cbI][j])
                ans += (j+1).ToString() + ";;";
        }
        ans = ans.Remove(ans.Length - 2);
        cbI++;
        Answers.Add(ans);
    }
    else
    {
        Answers.Add(model.AnswersStr[i]);
    }
}
```

Результаты

EXAM_ASP

[Home](#)

[Privacy](#)

123

[Ваши результаты](#)

[Добавить нового пользователя](#)

[Редактировать тесты](#)

[Добавить новый тест](#)

[Выход](#)

Test RES: 2 / 4

Test RES: 1 / 4

Test RES: 4 / 4

Новый Тест RES: 1 / 1

Финальный тест RES: 4 / 4

Тест для презентации

Создать тест

Добавление тестов

Введите текст вопроса Как звали Поттера?

Введите ответ вопроса Гарри

Введите текст вопроса Выберите вариант 3

☐ B1

☐ B2

☒ B3

☐ B4

Ответ строкой



Сохранить тип вопроса

+

-

Добавление тестов: представление

```
if (document.getElementById("questionType").value == 2) {
    document.getElementById('AllBlock').innerHTML += ('<div id = "ansBlock_" + ansC + "></div>' +
        '<br>');
    document.getElementById("questionType").remove();
    document.getElementById("saveQuestionType").remove();
    var optC = 0;
    document.getElementById('ansBlock_' + ansC).innerHTML += ('<br><label>Введите текст вопроса</label>' +
        '<input id = "QuestionText_" + ansC + '" type="text"/><br>');
    var pAnsOpt = document.createElement("button");
    pAnsOpt.id = "plusAnswerOption";
    document.getElementById('ansBlock_' + ansC).innerHTML += ('<div id = "opts" + ansC + "></div>' +
        '<br>');
    pAnsOpt.textContent = "+";
    pAnsOpt.onclick = function () {

        document.getElementById('opts' + (ansC - 1)).innerHTML += ('<label id = "lOpt">Введите текст варианта ответа</label>' +
            '<input id = "OptionText" type="text"/><br>');
        var pAnsOptText = document.createElement("button");
        pAnsOptText.id = "plusAnswerOptionText";
        pAnsOptText.textContent = "Сохранить текст варианта ответа";
        pAnsOptText.onclick = function () {

            document.getElementById('opts' + (ansC - 1)).innerHTML += ('<input type = "radio" name = "Option_" + (ansC - 1) + '"' +
                'value = "" + optC + '"><label id = "Option_" + (ansC - 1) + "' +
                '_' + optC + '">' + document.getElementById("OptionText").value + '</label>');
            optC += 1;
            document.getElementById("OptionText").remove();
            document.getElementById("lOpt").remove();
            document.getElementById("plusAnswerOptionText").remove();
        }
        document.getElementById('opts' + (ansC - 1)).appendChild(pAnsOptText);
    }
    document.getElementById('ansBlock_' + ansC).appendChild(pAnsOpt);
    ansC += 1;
    document.getElementById('myForm').innerHTML += ('<input name = "FieldForQuestion_" + (ansC - 1) +
        '" id = "FieldForQuestion_" + (ansC - 1) + '" type="hidden"/>');
    document.getElementById('myForm').innerHTML += ('<input name = "FieldForAnswer_" + (ansC - 1) +
        '" id = "FieldForAnswer_" + (ansC - 1) + '" type="hidden"/>');
    return true;
}
```

Добавление тестов: представление

```
if (ifex1 != null && ifex1 != undefined) {  
  
    document.getElementById('FieldForQuestion_' + i).value = document.getElementById('QuestionText_' + i).value;  
    document.getElementById('FieldForAnswer_' + i).value = document.getElementById('AnswerText_' + i).value;  
}
```

```
if (ifex2 != null && ifex2 != undefined) {  
  
    document.getElementById('FieldForQuestion_' + i).value = document.getElementById('QuestionText_' + i).value;  
    document.getElementById('FieldForAnswer_' + i).value = document.querySelector("input[type='radio']" + "[name=Option_" + i +  
    "]:checked").value;  
    document.getElementById('myForm').innerHTML += ('<input name = "FieldForAnswerTexts_' + i +  
    '" id = "FieldForAnswerTexts_' + i + '" type="hidden"/>');  
    for (optC = 0; optC < 100; optC++) {  
        var ifex4 = document.getElementById('Option_' + i + '_' + optC);  
        if (ifex4 != null && ifex4 != undefined) {  
            document.getElementById('FieldForAnswerTexts_' + i).value += document.getElementById('Option_' + i  
            + '_' + optC).textContent + ';;';  
        }  
        else {  
            break;  
        }  
    }  
}
```

Перспективы

1. Улучшение разметки
2. Добавление графики
3. Добавление редактирования тестов
4. Улучшить удобство в использовании
5. Улучшение базы данных

Спасибо за внимание