

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)

КУРСОВОЙ ПРОЕКТ

НА ТЕМУ

РАЗРАБОТКА ПОДСИСТЕМЫ ПОДАЧИ ЗАЯВОК

НА МЕРОПРИЯТИЯ СЛУЖБЫ ЗАНЯТОСТИ

Л109. 25КП01. 025 ПЗ

(Обозначение документа)

МДК.02.01 Технология разработки

программного обеспечения

Студент	ИСПП-21	08.12.2025	А.Н. Поршнев
	(Группа)	(Подпись)	(И.О. Фамилия)
Преподаватель		09.12.2025	Ю.С. Маломан
		(Подпись)	(И.О. Фамилия)

Архангельск 2025

СОДЕРЖАНИЕ

Перечень сокращений и обозначений.....	3
Введение.....	4
1 Анализ и разработка требований.....	6
1.1 Назначение и область применения.....	6
1.2 Постановка задачи	6
1.3 Выбор состава программных и технических средств	8
2 Проектирование программного обеспечения	9
2.1 Проектирование интерфейса пользователя.....	9
2.2 Разработка архитектуры программного обеспечения.....	11
2.3 Проектирование базы данных	11
3 Разработка и интеграция модулей программного обеспечения.....	13
3.1 Разработка программных модулей.....	13
3.2 Реализация интерфейса пользователя.....	13
3.3 Разграничение прав доступа пользователей	15
3.4 Экспорт и импорт данных.....	16
4 Тестирование и отладка программного обеспечения.....	19
4.1 Структурное тестирование.....	19
4.2 Функциональное тестирование	20
5 Инструкция по эксплуатации программного обеспечения	22
5.1 Установка программного обеспечения.....	22
5.2 Инструкция по работе	22
Заключение	29
Список использованных источников	30

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем курсовом проекте применяются следующие сокращения и обозначения:

БД – база данных

ОС – операционная система

СУБД – система управления базами данных

ASP.NET – платформа для разработки веб-приложений

MVC – паттерн программирования Модель-Представление-Контроллер

CSS – язык стилей

EF Core – технология объектно-реляционного отображения для доступа к данным в приложениях .NET

ERD – диаграмма «сущность-связь»

HTML – язык разметки для создания и структурирования веб-страниц

IDE – интегрированная среда разработки

SDK – набор инструментов для разработки программного обеспечения

ВВЕДЕНИЕ

В эпоху цифровизации и стремительного развития интернет-технологий веб-приложения становятся неотъемлемой частью повседневной жизни – от онлайн-шопинга и банковских сервисов до образовательных платформ и систем управления бизнесом. Ключевой задачей современных разработчиков является создание надёжных, безопасных и удобных в использовании веб-приложений с продуманной архитектурой. Кроме того, в условиях постоянной эволюции технологий критически важно проектировать архитектуру с запасом гибкости — чтобы приложение могло безболезненно интегрировать новые стандарты, API и пользовательские сценарии в будущем.

В современных социально-экономических условиях кадровые центры играют ключевую роль в поддержке трудоустройства населения, организации профессионального обучения и информировании граждан о вакансиях и мероприятиях, направленных на повышение их конкурентоспособности на рынке труда. Их деятельность сегодня приобретает ещё большую значимость в связи с динамичными изменениями на рынке труда — автоматизацией процессов, появлением новых профессий, трансформацией требований к компетенциям. Для повышения эффективности работы кадровые центры активно внедряют цифровые инструменты. Одним из важнейших инструментов взаимодействия службы занятости с гражданами является организация и проведение мероприятий: ярмарок вакансий, тренингов, семинаров, курсов повышения квалификации.

Целью данного курсового проекта является разработка веб-приложения для подачи заявок на мероприятия службы занятости, предназначенного для получения актуальных сведений о проводимых мероприятиях и доступных вакансиях. Сайт предоставляет пользователю возможность просмотра списка мероприятий, списка вакансий, подачи заявок на мероприятия и просмотра поданных заявок в личном кабинете пользователя.

Для достижения поставленной цели требуется решить следующие задачи:

проанализировать предметную область;

определить требования к разрабатываемому программному средству;

проектировать структуру БД для хранения информации необходимой для функционирования веб-приложения;

обеспечить удобство взаимодействия пользователя с приложением за счёт оптимизации навигации и логики интерфейса;

реализовать серверную часть, обеспечивающую выполнение бизнес-логики приложения и разграничение прав доступ;

провести тестирование разработанного программного продукта.

Выполнение вышеперечисленных задач позволит создать полноценное, работоспособное веб-приложение для подачи заявок на мероприятия службы занятости – с продуманной архитектурой, удобным интерфейсом и надёжной логикой.

1 Сбор и анализ требований

1.1 Назначение и область применения

Подсистема разрабатывается для центра занятости населения. Оно упростит процесс подачи заявок на мероприятия, предоставит возможность просмотра списка доступных вакансий и активных мероприятий.

Разрабатываемая подсистема будет актуальна для учебных заведений, проводящих профориентационные работы для своих студентов, людей, находящихся в поиске работы, а также для самого кадрового центра, который проводит мероприятия.

1.2 Постановка задачи

Необходимо разработать веб-сайт для подачи заявок на мероприятия службы занятости. В подсистеме будет реализована следующая функциональность:

- авторизация и регистрация пользователя;
- подача заявки на мероприятие службы занятости;
- просмотр списка вакансий;
- просмотр списка мероприятий;
- сортировка списка мероприятий по названию, дате публикации, дате проведения;
- фильтрация мероприятий по статусу проведения;
- публикация информации о новых мероприятиях;
- просмотр списка заявок;
- формирование отчёта о проведении мероприятий;
- обработка заявок (принятие или отклонение).

Интерфейс должен быть интуитивно понятен для пользователя. В целях безопасности на сайте необходимо пройти авторизацию перед использованием основной функциональности.

Гость может зарегистрироваться и авторизоваться, просмотреть список мероприятий и доступных вакансий.

Пользователь может выполнять все действия, доступные гостю, а также подать заявку на мероприятие, просмотреть поданную заявку, удалить её и изменять её содержимое.

Менеджер может выполнять все действия, доступные пользователю, а также редактировать информацию о мероприятиях и вакансиях.

Администратор может просматривать список заявок, обрабатывать заявки и выполнять все действия, доступные менеджеру.

На рисунке 1 представлена диаграмма вариантов использования подсистемы различными категориями пользователей.

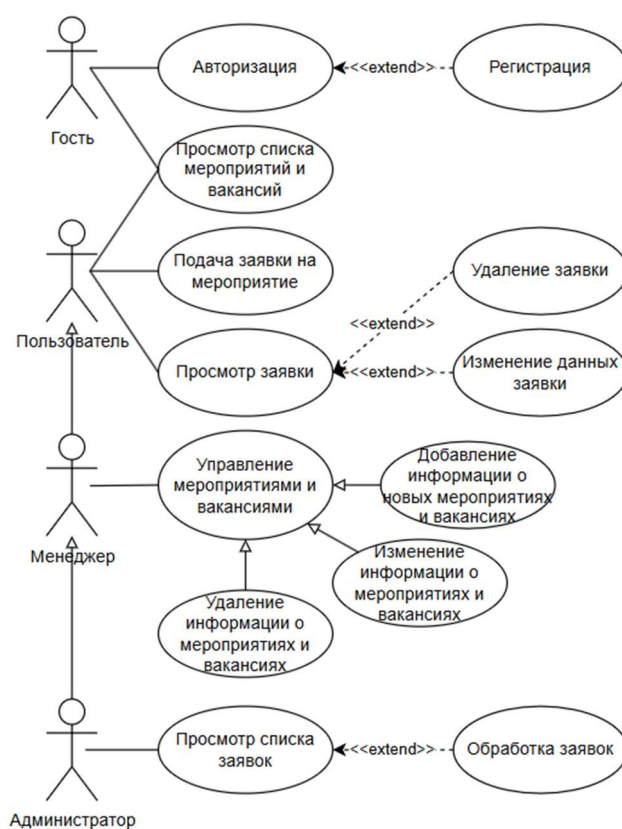


Рисунок 1 – Диаграмма вариантов использования

1.3 Выбор состава программных и технических средств

Согласно цели проекта требуется разработать веб-сайт для подачи заявок и БД для хранения данных.

В качестве СУБД выбрана MySQL 8.0, так как она имеет высокую производительность, проста в освоении и легко масштабируется.

Сайт будет написан на языке программирования C# с использованием фреймворка ASP.NET MVC, так как он обеспечивает высокую производительность, гибкую архитектуру и простоту тестирования.

Для разработки будет использоваться интегрированная среда разработки программ Visual Studio 2022, так как это производительная и простая в освоении IDE, обеспечивающая быструю и комфортную разработку.

Для функционирования системы на стороне сервера необходимы следующие программные и технические средства:

- ОС Windows 10 или выше, Windows Server 2019 или выше, Ubuntu 22.04 или выше;
- 64-х разрядная архитектура;
- MySQL Server не ниже 8.0;
- .NET SDK не ниже 8.0 и ASP.NET MVC 5;
- 3 ГБ доступной оперативной памяти;
- процессор с частотой не менее 1 ГГц и не менее 2 ядер;
- 10 ГБ доступного дискового пространства.

Доступ к системе осуществляется с помощью браузеров, таких как Chrome (версии 90 и выше), FireFox (версии 88 и выше) и других, поддерживающих HTML5 и CSS3, необходимых для функционирования системы на стороне клиента. Помимо программных, необходимы также следующие технические средства:

- процессор с частотой не менее 1 ГГц;
- 2 ГБ доступной оперативной памяти.

2 Проектирование ПО

2.1 Проектирование интерфейса пользователя

В рамках разработки веб-сайта подачи заявок на мероприятия службы занятости спроектирован интерфейс пользователя в виде wireframe при помощи сервиса draw.io. Эти визуальные представления позволяют наглядно понять структуру приложения, его основные элементы и функциональность.

Для интерфейса веб-сайта выбрана следующая цветовая палитра:

- #ffffff – цвет фона и подписей кнопок
- #ff5124 – цвет заголовков
- #000000 – цвет текста
- #014898 – вторичный цвет текста и цвет кнопок
- #007bff – вторичный цвет кнопок

Интерфейс страниц «Мероприятия», «Вакансии», «Личный кабинет» представлены на рисунках 2, 3 и 4 соответственно.

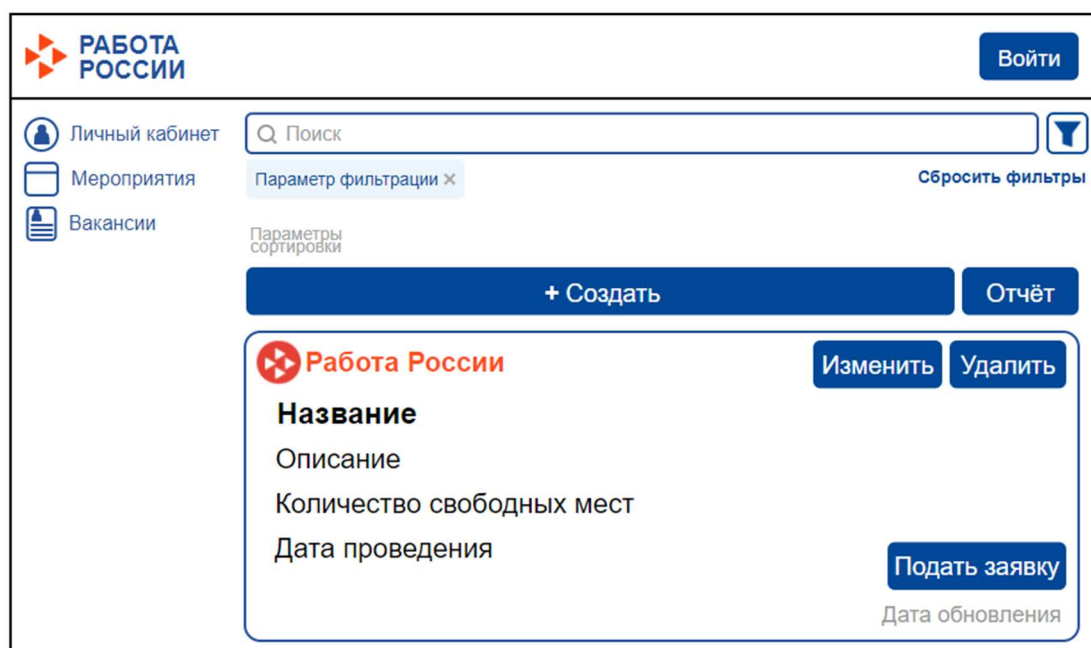


Рисунок 2 – Draw.io. Макет страницы «Мероприятия»

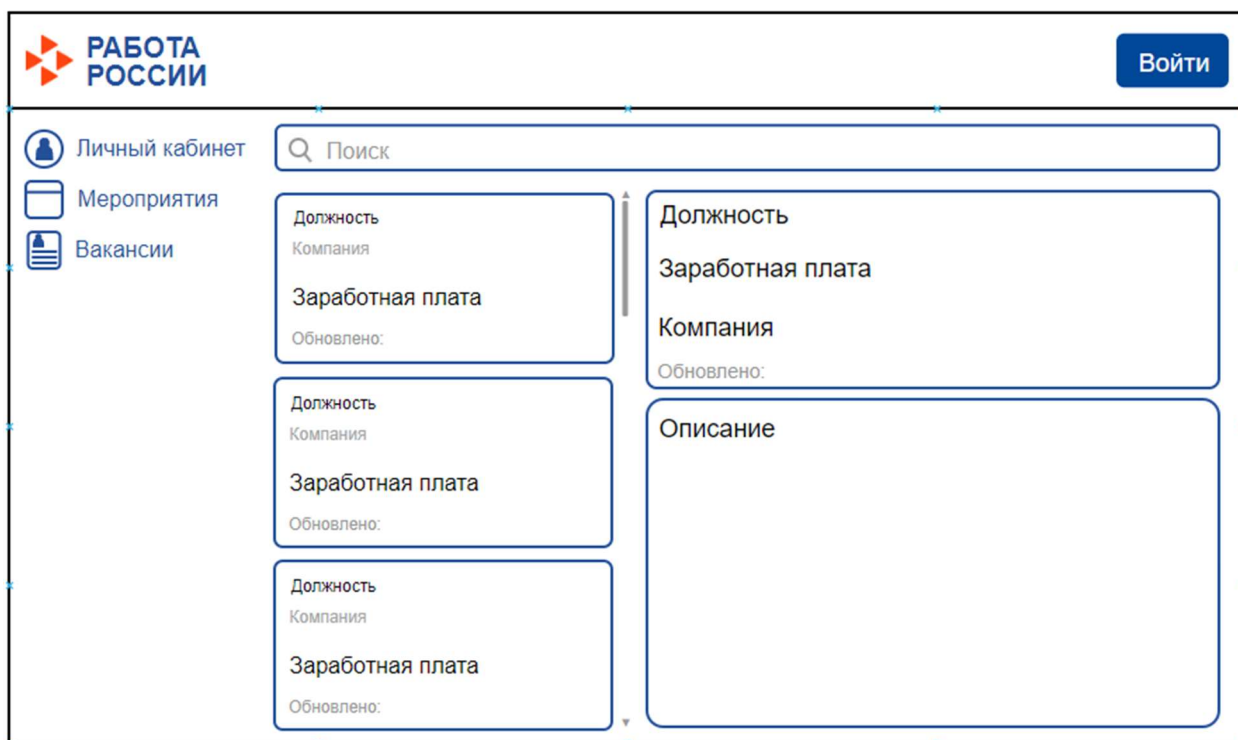


Рисунок 3 – Draw.io. Макет страницы «Вакансии»

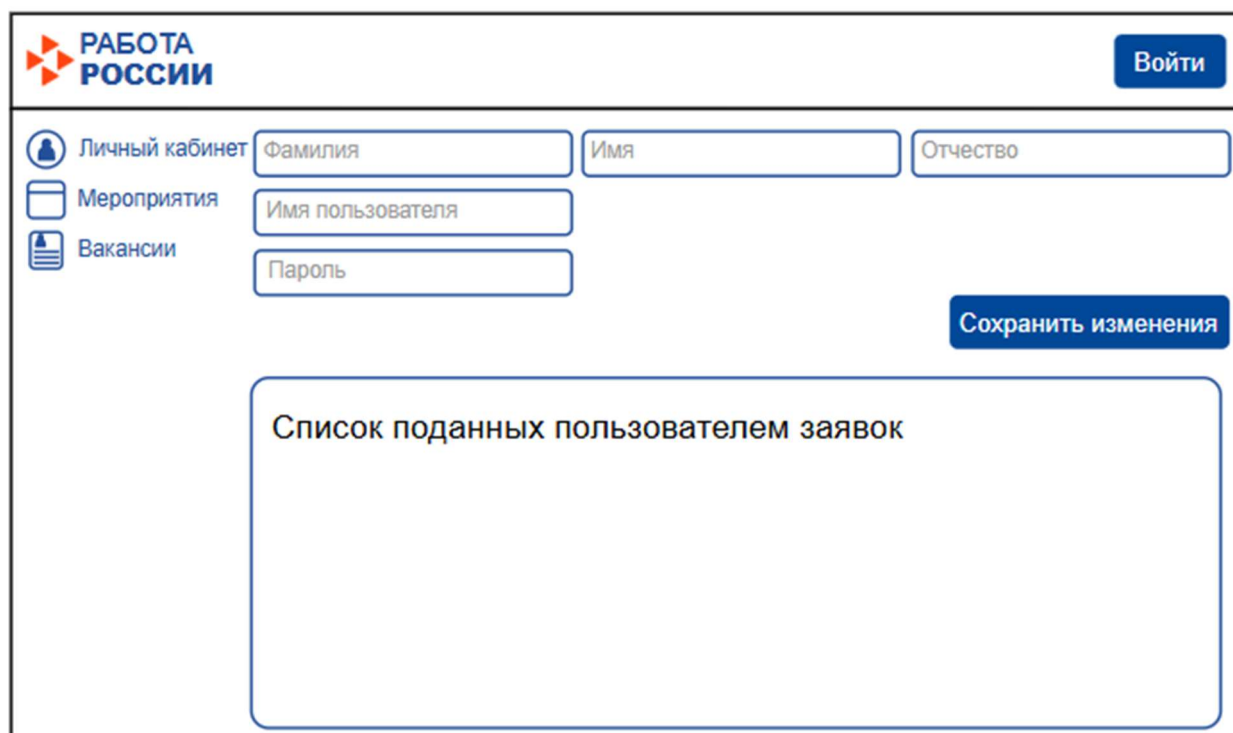


Рисунок 4 – Draw.io. Макет страницы «Личный кабинет»

2.2 Разработка архитектуры программного обеспечения

Приложение предназначено для подачи заявок на мероприятия службы занятости. Архитектура приложения построена на основе клиент-серверной модели и включает в себя несколько ключевых компонентов: серверная часть приложения, клиентская часть приложения и БД. Диаграмма развёртывания компонентов представлена на рисунке 5.

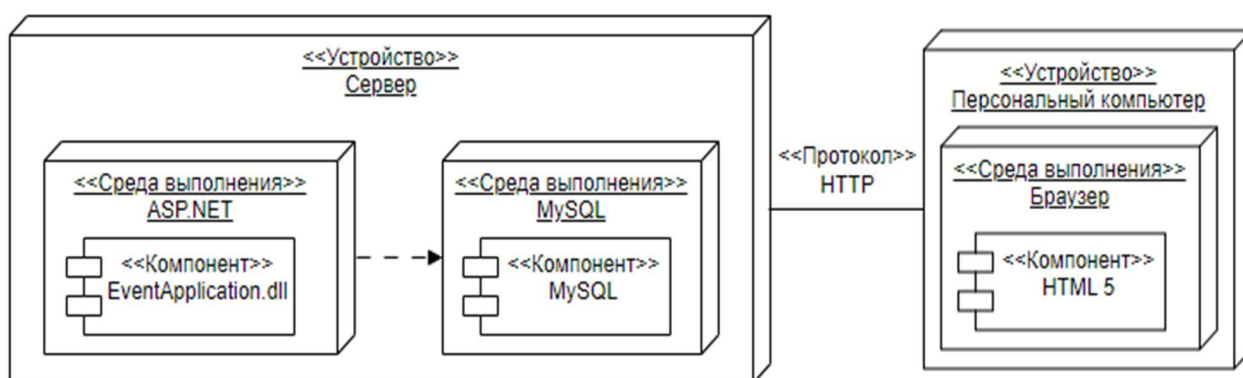


Рисунок 5 – Диаграмма развёртывания компонентов

2.3 Проектирование БД

В рамках курсового проектирования требуется разработать БД для хранения информации о мероприятиях, заявках и пользователях [3]. Модели БД созданы с использованием MySQL Workbench. Обеспечена ссылочная целостность через внешние ключи, что минимизирует риски некорректных данных. Физическая модель БД соответствует третьей нормальной форме и разработана с использованием нотации Crow's Foot для наглядного отображения связей.

На рисунке 6 в виде ERD представлена физическая модель БД.

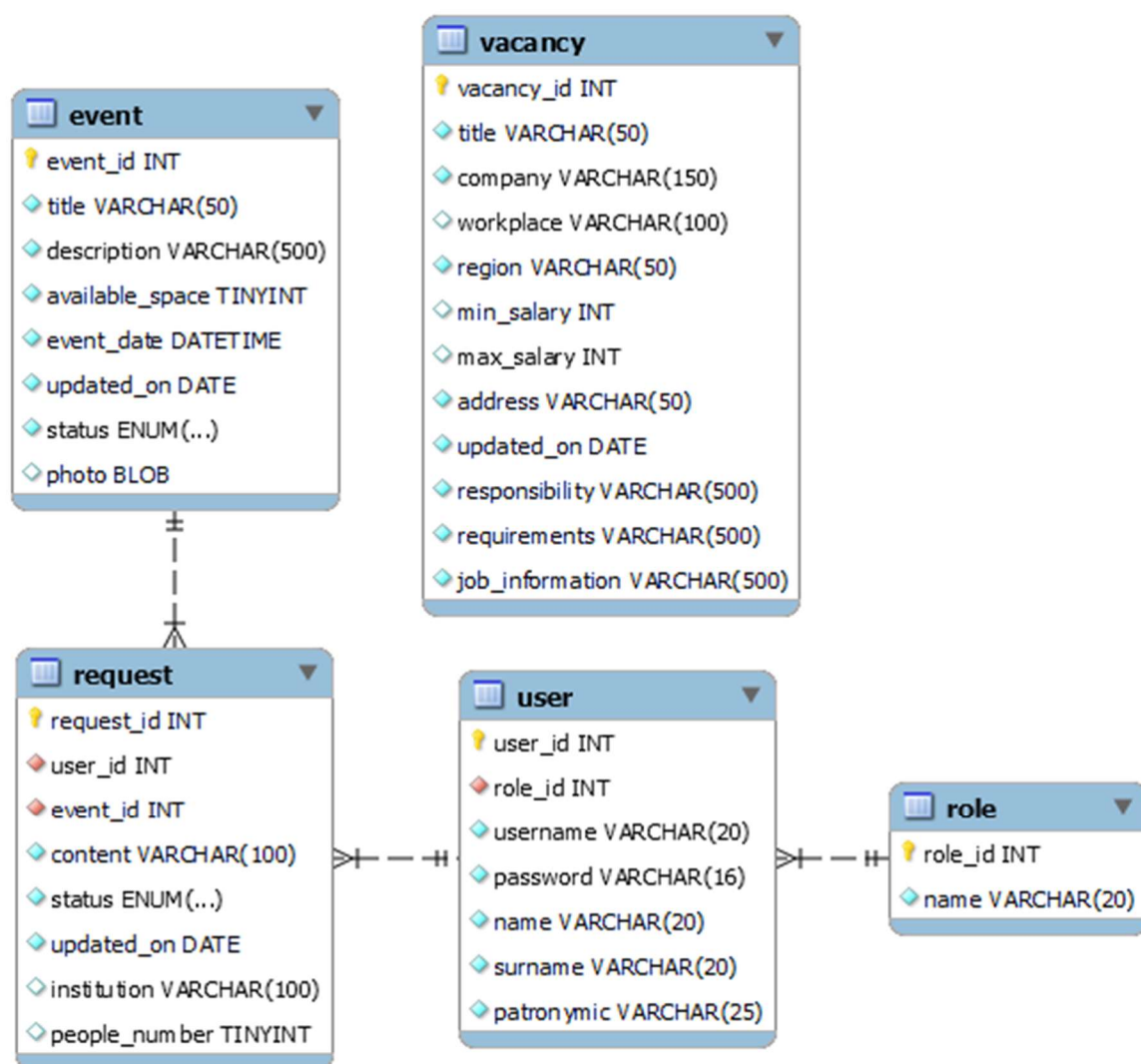


Рисунок 6 – MySQL Workbench. Физическая модель БД

3 Разработка и интеграция модулей ПО

3.1 Разработка программных модулей

Веб-сайт разработан на C# с использованием фреймворка ASP.NET MVC в IDE Visual Studio 2022 [2].

Сайт взаимодействует с БД посредством EF Core [5].

Для работы с данными для сайта разработаны следующие контроллеры: AuthController (для управления авторизацией), EventsController (для работы с мероприятиями) и UsersController (для изменения данных пользователя). Код метода добавления нового мероприятия представлен листингом 1.

Листинг 1 – Код метода Create

```
public async Task<IActionResult>
Create([Bind("EventId, Title, Description, AvailableSpace, UpdatedOn
, EventDate, Status")] Event @event)
{
    if (ModelState.IsValid)
    {
        @event.UpdatedOn = DateOnly.FromDateTime(DateTime.Now);
        _context.Add(@event);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(@event);
}
```

3.2 Реализация интерфейса пользователя

Интерфейс приложения реализован с использованием CSS для оформления страниц [4]. Для вставки кода на C# на страницах использована технология MVC. Для отображения списка мероприятий разработана страница Events, код верстки которой представлен листингом 2.

Листинг 2 – Код верстки страницы просмотра списка мероприятий

```
// Контейнер списка мероприятий
<div class="post-wall" style="display: flex; flex-wrap: wrap;">
  // Валидация роли пользователя для доступа к функциональности
  @if (Context.Session.GetString("UserRole") == "Администратор"
  || Context.Session.GetString("UserRole") == "Менеджер")
  {
    <div class="create-event-container">
      // Переход на форму добавления мероприятия
      <a class="create-event-btn btn" style="margin: 0 2.5px"
asp-controller="Events" asp-action="Create"><i class="fa fa-
plus"></i> Создать</a>
      // Формирование отчёта о проведении мероприятий
      <form class="action-form" asp-action="ExportToXlsx">
        <button type="submit" class="create-report-btn btn"
style="margin: 0 2.5px" title="Сформировать отчёт"><i class="fa
fa-file"></i></button>
      </form>
    </div>
  }
  // Перебор элементов для отображения списка мероприятий
  @foreach (var item in Model)
  {
    // Контейнер мероприятия
    <div class="post-container">
      <div class="post-header">
        <div class="profile-container">
          
          <span style="font-size: 22px; color: #FF5124; font-
weight: bold">Работа России</span>
        </div>
        @if (Context.Session.GetString("UserRole") ==
"Администратор" || Context.Session.GetString("UserRole") ==
"Менеджер")
        {
          <div class="post-actions">
            // Переход на форму редактирования мероприятия
            <a class="action-form btn" asp-action="Edit" asp-
route-id="@item.EventId" title="Изменить"><i class="fa fa-
edit"></i></a>
            // Удаление мероприятия из БД
            <form class="action-form" asp-action="Delete" asp-
route-id="@item.EventId">
              <button type="submit" class="btn delete-btn"
title="Удалить"/><i class="fa fa-trash"></i>
            </form>
          </div>
        }
      </div>
    </div>
  }
}
```

```

        // Блок содержимого мероприятия
        <div class="post-content">
            // Блок фотографии
            <div class="photo-container">
                // Использование заглушки в качестве фотографии
                мероприятия
                
            </div>
            // Блок текста мероприятия
            <div class="post-text-container">
                <h4 style="font-weight:
bold;">@Html.DisplayFor(modelItem => item.Title)</h4>
                <h5 style="text-align:
justify">@Html.DisplayFor(modelItem => item.Description)</h5>
                <h5>Свободных мест: @Html.DisplayFor(modelItem =>
item.AvailableSpace)</h5>
                <div style="display: flex; justify-content: space-
between; align-items: center;">
                    <h5>Дата проведения: @Html.DisplayFor(modelItem =>
item.EventDate)</h5>
                    @if
(!string.IsNullOrEmpty(Context.Session.GetString("UserRole")))
                    {
                        // Переход на форму подачи заявки
                        <a style="text-align: right;" class="btn" asp-
controller="Requests" asp-action="Create" asp-route-
eventId="@item.EventId" asp-route-
userId="@Context.Session.GetString("UserId")"><i class="fa fa-
share"></i> Подать заявку</a>
                    }
                </div>
            </div>
            <div class="post-date-updated">
                @Html.DisplayFor(modelItem => item.UpdatedOn)
            </div>
        </div>
    }
</div>

```

3.3 Разграничение прав доступа

В веб-приложении реализовано разграничение прав доступа пользователей с использованием технологии Session. Данные и роли пользователей хранятся в БД в таблицах user и role соответственно.

Код метода авторизации представлен листингом 3.

Листинг 3 – Код метода Login

```
public ActionResult Login([Bind("Username,Password")] User
@user)
{
    // Проверка наличия пользователя в БД по имени пользователя
    var currentUser = _context.Users.FirstOrDefault(u =>
u.Username == @user.Username);
    // Валидация данных
    if (currentUser != null && currentUser.Password ==
user.Password)
    {
        // Занесение данных роли пользователя в контекст сессии
        HttpContext.Session.SetString("UserRole",
_context.Roles.Where(r => r.RoleId ==
currentUser.RoleId).Select(r => r.Name).FirstOrDefault());
        // Занесение данных имени пользователя в контекст сессии
        HttpContext.Session.SetString("UserFullname",
${currentUser.Surname} {currentUser.Name}
{currentUser.Patronymic}");
        // Переход на страницу мероприятий
        return RedirectToAction("Index", "Events");
    }
    else
    {
        ModelState.AddModelError("", "Неверный логин или пароль");
        return View();
    }
}
```

3.4 Экспорт и импорт данных

На сайте реализован экспорт мероприятий за текущий месяц для создания отчётов о проведении мероприятий. Экспорт реализован с использованием библиотеки EPPlus в формате xlsx.

Код генерации отчёта представлен листингом 5.

Листинг 5 – Код метода ExportToXlsxAsync

```
public async Task<IActionResult> ExportToXlsxAsync()
{
    // Установка лицензии для личного пользования
    ExcelPackage.License.SetNonCommercialPersonal("CProj");
}
```



```

// Получение данных из БД
List<Event> events = await _context.Events.Where(e =>
e.EventDate.Month == DateTime.Now.Month).ToListAsync();
string filePath = @"C:\Temp\{DateTime.Now.ToString("MMMM yyyy
HH:mm:ss")}.xlsx";
using (var package = new ExcelPackage(new FileInfo(filePath)))
{
    // Создание рабочего листа
    var worksheet =
package.Workbook.Worksheets.Add($"{DateTime.Now.ToString("MMMM
yyyy")}");
    worksheet.Cells[1, 1].Value = "№ п/п";
    SetStyle(worksheet, 1, 1);
    worksheet.Cells[1, 2].Value = "Мероприятие";
    SetStyle(worksheet, 1, 2);
    worksheet.Cells[1, 3].Value = "Описание";
    SetStyle(worksheet, 1, 3);
    worksheet.Cells[1, 4].Value = "Количество мест";
    SetStyle(worksheet, 1, 4);
    worksheet.Cells[1, 5].Value = "Статус";
    SetStyle(worksheet, 1, 5);
    worksheet.Cells[1, 6].Value = "Дата проведения";
    SetStyle(worksheet, 1, 6);
    // Заполнение ячеек таблицы данными мероприятий
    for (int i = 0; i < events.Count; i++)
    {
        worksheet.Cells[i + 2, 1].Value = i + 1;
        SetStyle(worksheet, i + 2, 1);
        worksheet.Cells[i + 2, 2].Value = events[i].Title;
        SetStyle(worksheet, i + 2, 2);
        worksheet.Cells[i + 2, 3].Value = events[i].Description;
        worksheet.Cells[i + 2, 3].Style.WrapText = true;
        SetStyle(worksheet, i + 2, 3);
        worksheet.Cells[i + 2, 4].Value =
events[i].AvailableSpace;
        SetStyle(worksheet, i + 2, 4);
        worksheet.Cells[i + 2, 5].Value = events[i].Status;
        SetStyle(worksheet, i + 2, 5);
        worksheet.Cells[i + 2, 6].Value = events[i].EventDate;
        SetStyle(worksheet, i + 2, 6);
    }
    // Настройка стилей
    worksheet.Cells.AutoFitColumns();
    worksheet.Column(3).Width = 60;
    worksheet.Column(6).Style.Numberformat.Format = "dd.MM.yyyy
HH:mm:ss";
    // Сохранение документа
    package.Save();
}
return RedirectToAction(nameof(Index));
}

```

На рисунке 7 представлен пример сгенерированного отчёта.

A	B	C	D	E	F
№ п/п	Мероприятие	Описание	Количество мест	Статус	Дата проведения
1	Профориентационный семинар для молодежи	Приглашаем молодых людей на наш профориентационный семинар, где вы сможете определить свои профессиональные предпочтения, получить консультации и составить план карьерного развития. Сделайте первый шаг к своей мечте!	40	В планах	17.12.2025 16:00:00
2	Мастер-класс по созданию бизнес-плана	Приглашаем вас на мастер-класс, где вы научитесь разрабатывать эффективные бизнес-планы, узнаете о ключевых аспектах предпринимательства и получите практические навыки для запуска собственного дела.	30	В планах	10.12.2025 15:00:00
3	Семинар по развитию лидерских качеств	Приглашаем вас на семинар, который поможет раскрыть ваши лидерские способности, научит управлять командой и достигать поставленных целей. Сделайте шаг к профессиональному росту!	20	В планах	08.12.2025 15:30:00

Рисунок 7 – Excel. Пример отчёта о проведении мероприятий

4 Тестирование и отладка программного обеспечения

4.1 Структурное тестирование

В ходе курсового проектирования проведено структурное тестирование метода `Create` контроллера `Events` с помощью библиотеки `xUnit`. Код для тестирования метода представлен листингом 7.

Листинг 5 – Код модульного теста добавления мероприятия

```
[Fact]
public async Task
Create_ValidModel_SavesToDb_And_RedirectsToIndex()
{
    // Подготовка тестовых данных
    var eventToCreate = new Event
    {
        Title = "Test Event",
        Description = "Test description",
        AvailableSpace = 25,
        EventDate = DateTime.Now.AddDays(7),
        UpdatedOn = DateOnly.FromDateTime(DateTime.Now),
        Status = "В планах"
    };

    // Вызов метод контроллера
    var result = await _controller.Create(eventToCreate);
    // Проверка перехода на страницу мероприятий
    var redirectResult =
Assert.IsType<RedirectToActionResult>(result);
    Assert.Equal("Index", redirectResult.ActionName);

    // Проверка сохранения в БД
    var savedEvent = await _context.Events.FirstOrDefaultAsync(e
=> e.Title == "Test Event");
    Assert.NotNull(savedEvent);
    Assert.Equal(eventToCreate.Title, savedEvent.Title);
    Assert.Equal(eventToCreate.AvailableSpace,
savedEvent.AvailableSpace);
    Assert.Equal(DateOnly.FromDateTime(DateTime.Now),
savedEvent.UpdatedOn);
}
```

Результат тестирования представлен на рисунке 8.

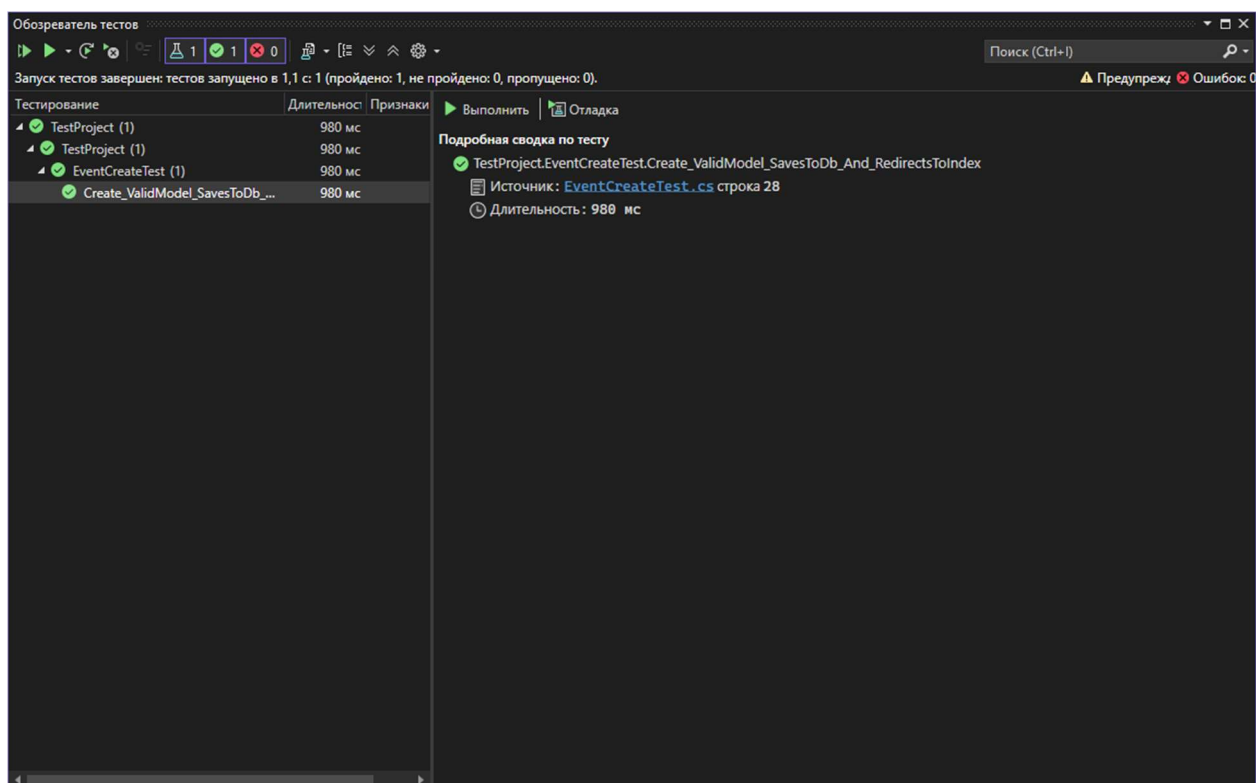


Рисунок 8 – Visual Studio. Обозреватель тестов с результатами тестирования

4.2 Функциональное тестирование

Во время курсового проектирования проведено функциональное тестирование страницы мероприятий по методологии «чёрного ящика», результаты тестирования представлены в таблице 1.

Таблица 1 – Набор тестов

Действие	Ожидаемый результат	Фактический результат
Нажать на кнопку «Создать».	Переход на страницу добавления мероприятия.	Соответствует ожидаемому
Нажать на кнопку «Редактировать» у первого мероприятия в списке	Переход на страницу редактирования выбранного мероприятия.	Соответствует ожидаемому

Продолжение таблицы 1

Действие	Ожидаемый результат	Фактический результат
Нажать на кнопку «Создать». Ввести «Ярмарка вакансий», «Пример», «30» и выбрать дату проведения «22.12.2025 16:00» в графе «Мероприятие», «Описание», «Количество мест» и «Дата проведения» соответственно. Нажать на кнопку «Опубликовать»	Переход на страницу добавления мероприятия. После ввода данных и нажатия на кнопку «Опубликовать» данные о мероприятии заносятся в БД и осуществляется переход на страницу мероприятий.	Соответствует ожидаемому
Нажать на кнопку «Создать». Нажать на кнопку «Опубликовать»	Под графами «Мероприятие», «Описание», «Количество мест» и «Дата проведения» высвечивается сообщение об обязательности заполнения полей ввода.	Соответствует ожидаемому
Нажать на кнопку «Редактировать» у первого мероприятия в списке. Выбрать статус «Отменено» из выпадающего списка «Статус». Нажать на кнопку «Сохранить изменения».	Статус мероприятия сохранится в БД и осуществится переход на страницу мероприятий.	Соответствует ожидаемому

По результатам тестирования можно сделать вывод, что созданное приложение работает корректно и согласно ожиданиям.

5 Инструкция по эксплуатации программного обеспечения

5.1 Установка программного обеспечения

Для функционирования системы на стороне сервера необходимы следующие программные и технические средства:

- ОС Windows 10 или выше, Windows Server 2019 или выше, Ubuntu 22.04 или выше;
- 64-х разрядная архитектура;
- MySQL Server не ниже 8.0;
- .NET SDK не ниже 8.0 и ASP.NET MVC 5;
- 3 Гб доступной оперативной памяти;
- процессор с частотой не менее 1 ГГц и не менее 2 ядер;
- 10 Гб доступного дискового пространства.

Для функционирования веб-приложения необходимы браузеры, такие как Chrome (версии 90 и выше), FireFox (версии 88 и выше) и другие, поддерживающие HTML5 и CSS3, необходимые для функционирования системы на стороне клиента. Помимо программных, необходимы также следующие технические средства:

- процессор с частотой не менее 1 ГГц;
- 2 Гб доступной оперативной памяти.

В качестве учётных данных используются:

- имя пользователя: admin;
- пароль: 12345678.

5.2 Инструкция по работе

После запуска веб-приложения на экране отобразится страница просмотра мероприятий, представленная рисунком 9.

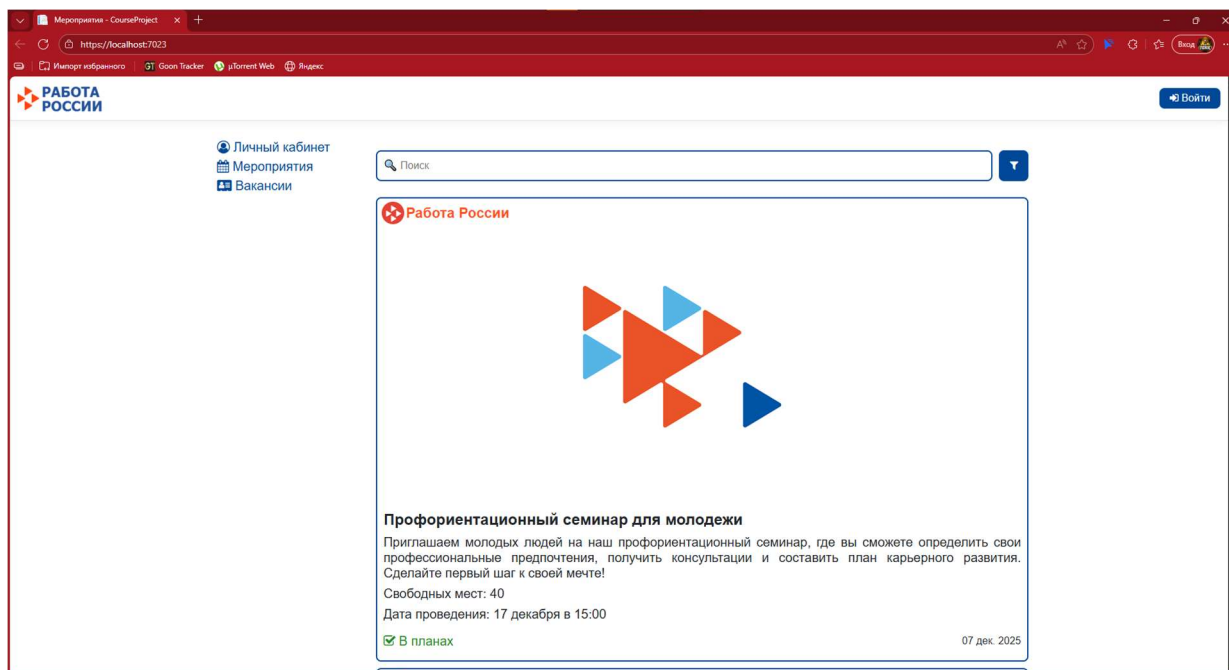


Рисунок 9 – Работа России. Вид страницы мероприятий

Помимо просмотра списка мероприятий, гостю доступны поиск и фильтрация списка мероприятий, а также просмотр списка доступных вакансий (рисунок 10).

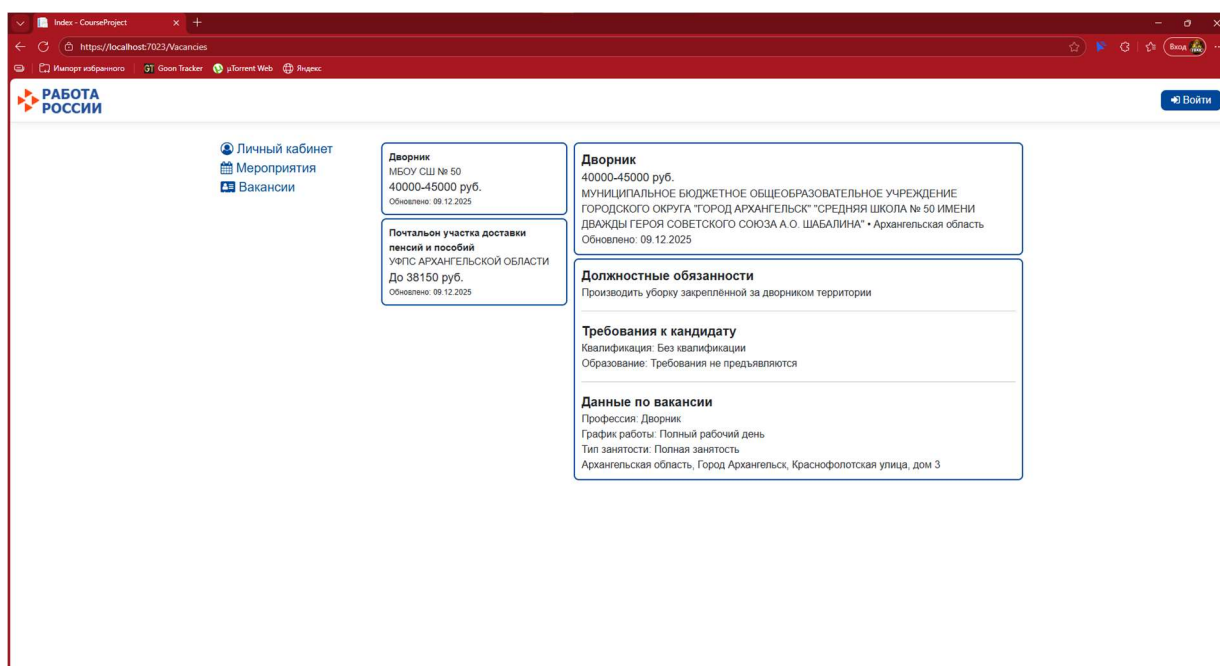


Рисунок 10 – Работа России. Вид страницы вакансий

Для использования основной функциональности приложения пользователю необходимо пройти авторизацию. При нажатии на кнопку «Войти» или на раздел «Личный кабинет», если пользователь не авторизован, открывается страница авторизации (представлена рисунком 11), на которой пользователю предлагается войти в систему с уже имеющимися данными или пройти регистрацию (рисунок 12) в случае отсутствия аккаунта.

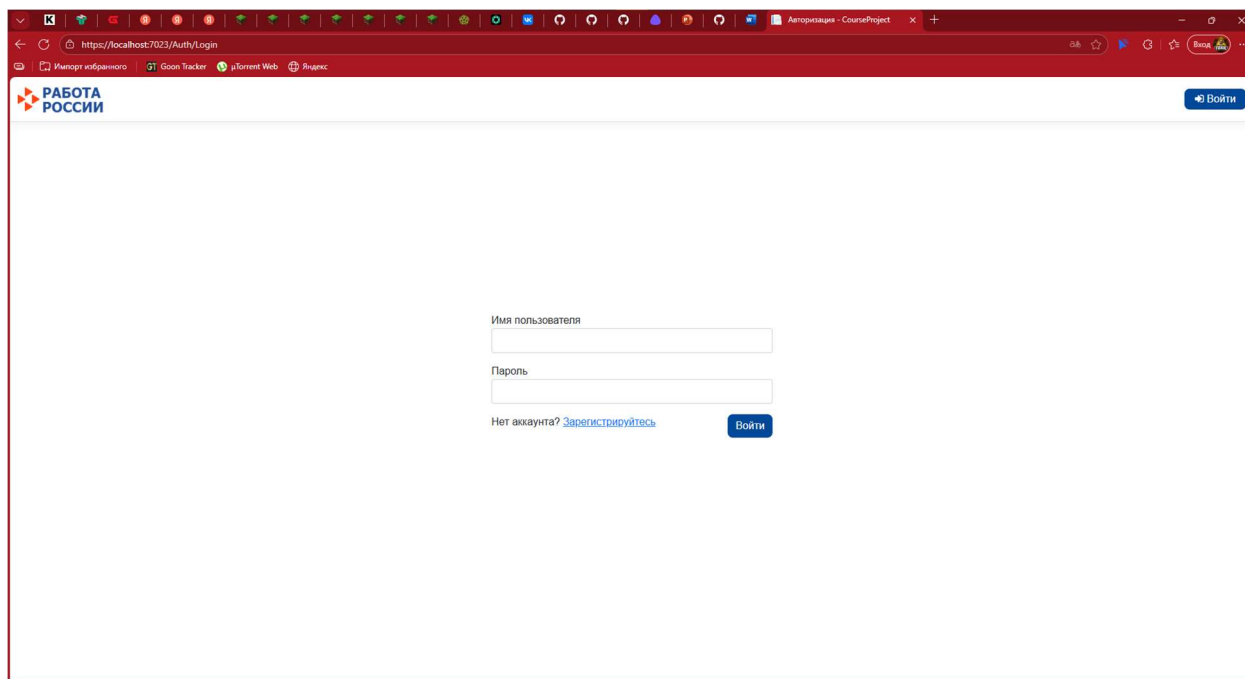
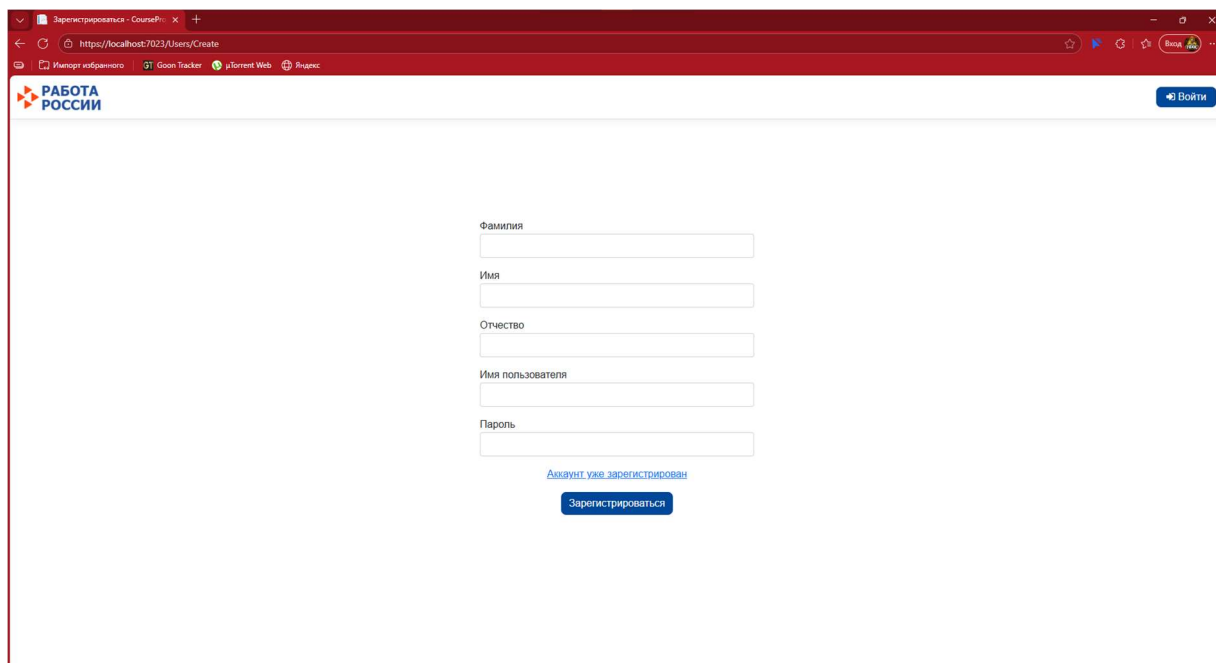


Рисунок 11 – Работа России. Вид страницы авторизации

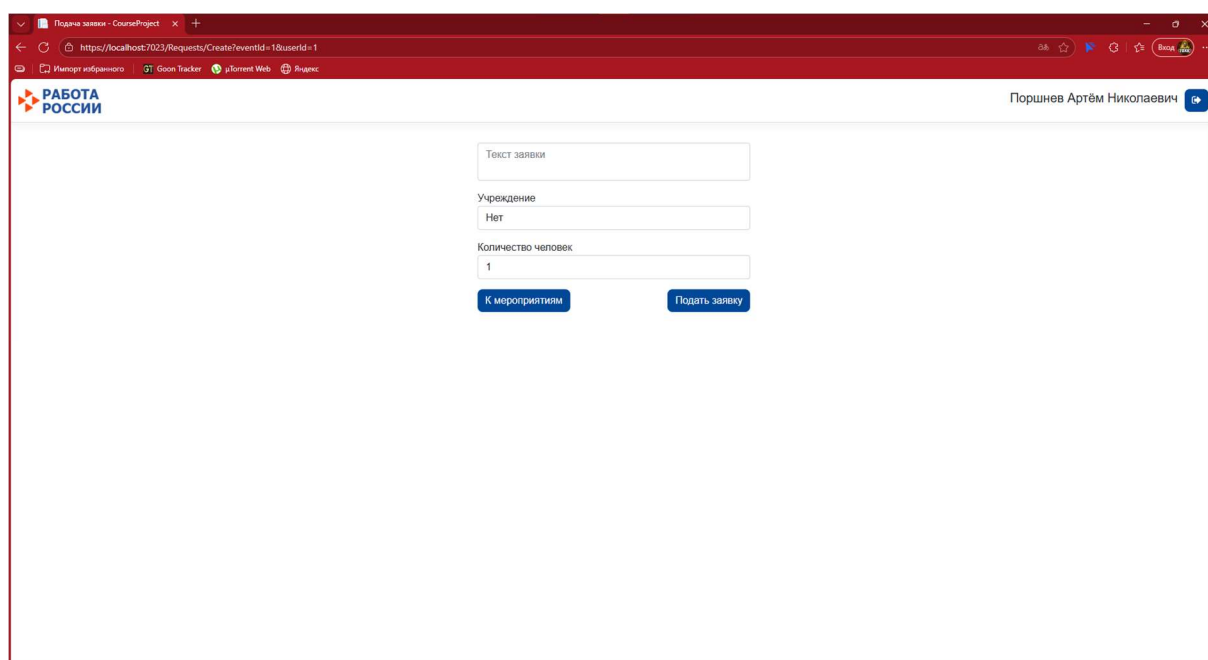
В случае успешного входа открывается страница мероприятий, обеспечивающая доступ к основной функциональности веб-приложения. Пользователю доступен раздел личного кабинета и возможность подачи заявки на мероприятие службы занятости. При нажатии на кнопку «Подать заявку» открывается страница для подачи заявки на мероприятие, представленная рисунком 13. Для ролей «Менеджер» и «Администратор» на этой странице дополнительно отображается кнопка «Создать», при нажатии на которую открывается страница добавления нового мероприятия (рисунок 14), генерации отчёта за текущий месяц о проведении мероприятий и

изменения информации о выбранном мероприятии. При нажатии на кнопку «Редактировать» откроется страница изменения информации о мероприятии, представленная на рисунке 15.



The screenshot shows a web browser window with the URL `https://localhost:7023/Users/Create`. The page features the 'РАБОТА РОССИИ' logo in the top left and a 'Войти' button in the top right. The main content area contains a registration form with the following fields: 'Фамилия', 'Имя', 'Отчество', 'Имя пользователя', and 'Пароль'. Below the 'Пароль' field, there is a link that says 'Аккаунт уже зарегистрирован' and a blue button labeled 'Зарегистрироваться'.

Рисунок 12 – Работа России. Вид страницы регистрации



The screenshot shows a web browser window with the URL `https://localhost:7023/Requests/Create?eventId=1&userId=1`. The page features the 'РАБОТА РОССИИ' logo in the top left and a user profile 'Поршнев Артём Николаевич' in the top right. The main content area contains an application form with the following fields: 'Текст заявки', 'Учреждение' (with a dropdown menu showing 'Нет'), and 'Количество человек' (with a dropdown menu showing '1'). At the bottom of the form, there are two blue buttons: 'К мероприятию' and 'Подать заявку'.

Рисунок 13 – Работа России. Вид страницы подачи заявки

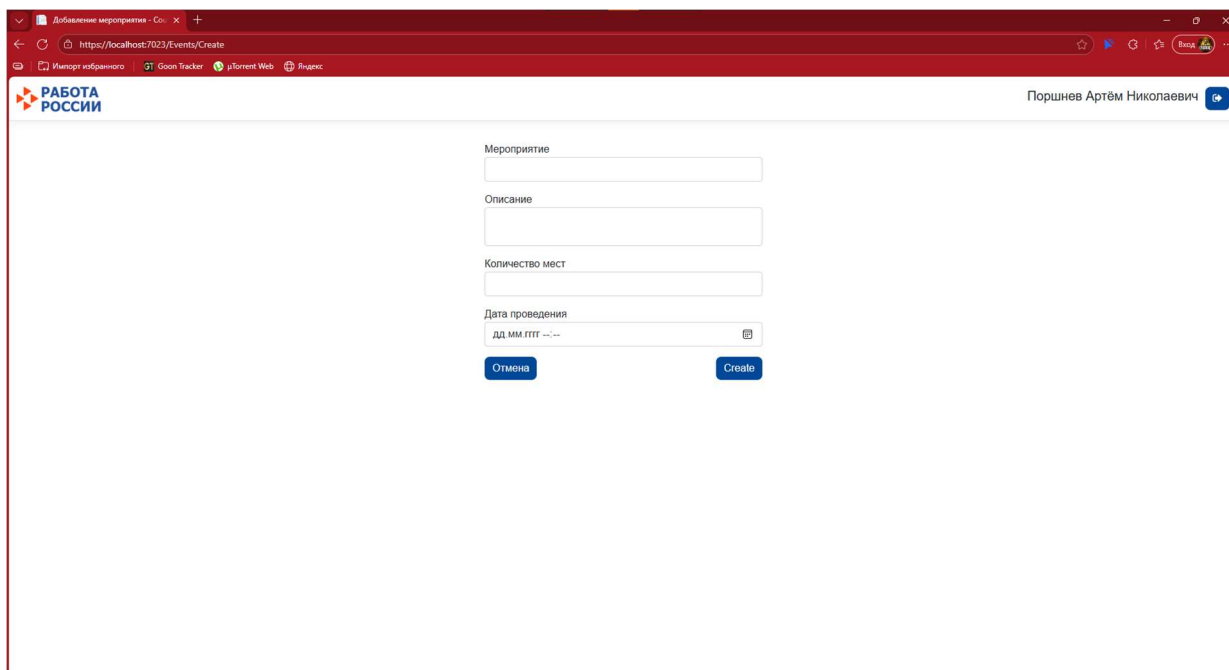


Рисунок 14 – Работа России. Вид страницы добавления нового мероприятия

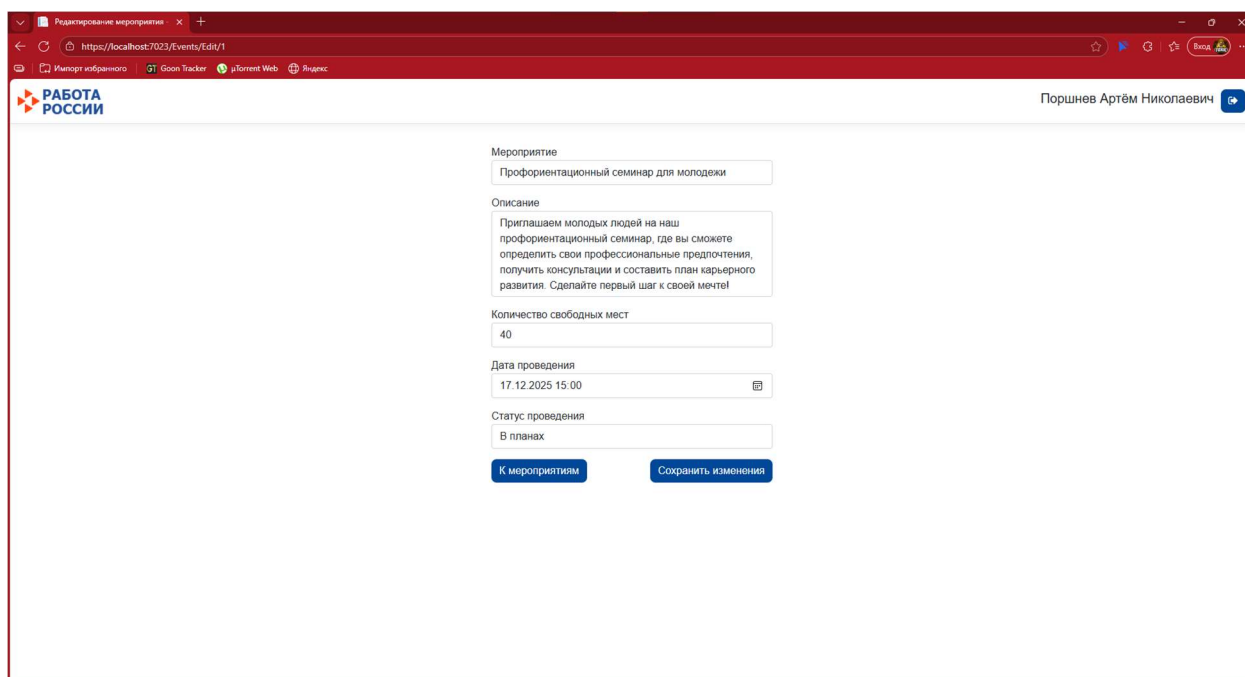


Рисунок 15 – Работа России. Вид страницы изменения данных мероприятия

Администратору также доступен раздел обработки заявок, страница которого представлена на рисунке 16.

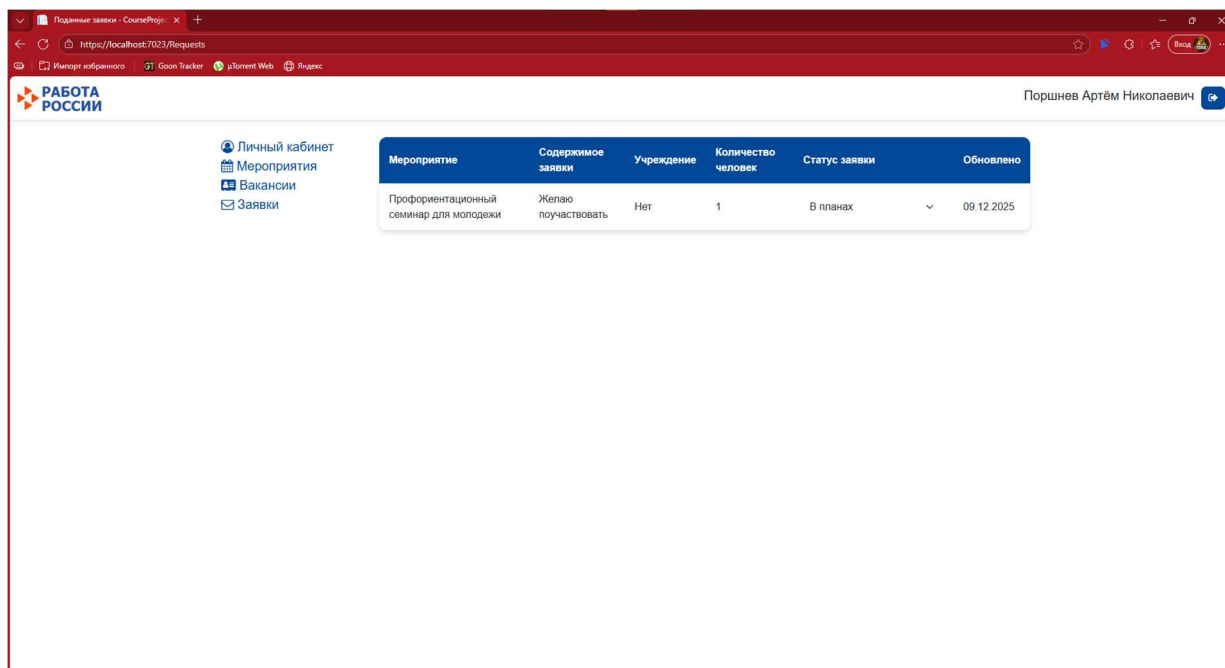


Рисунок 16 – Работа России. Вид страницы обработки заявок

В разделе «Личный кабинет» пользователь может изменить свои учётные данные и просмотреть список поданных заявок. Страница личного кабинета представлена рисунком 17.

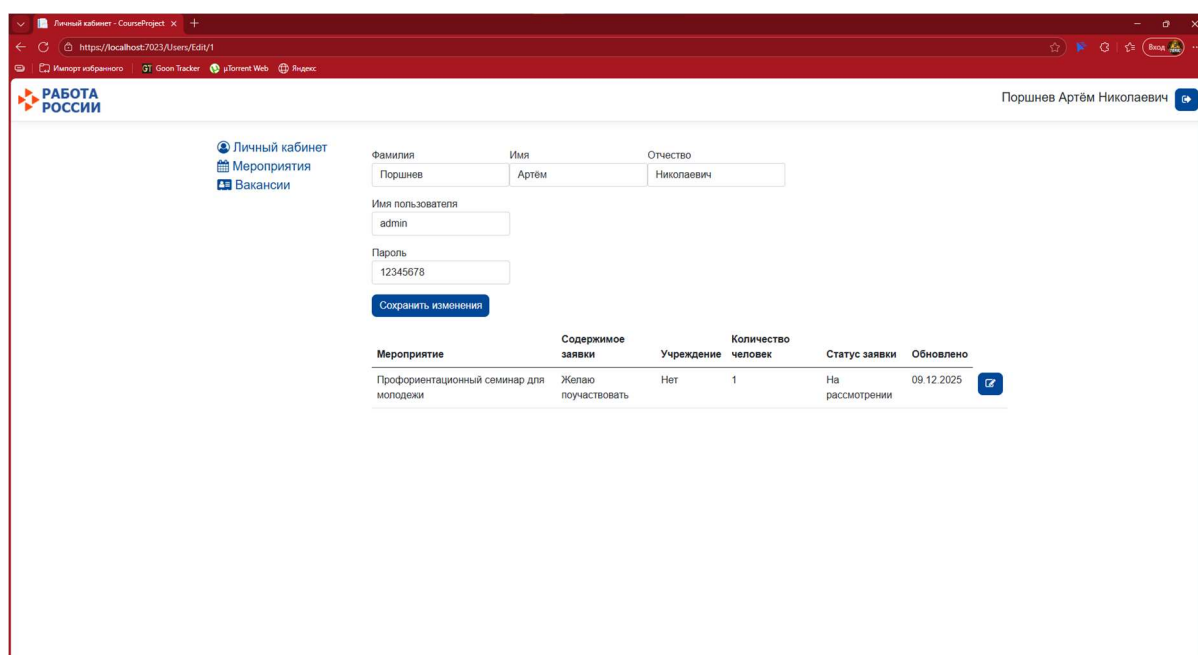
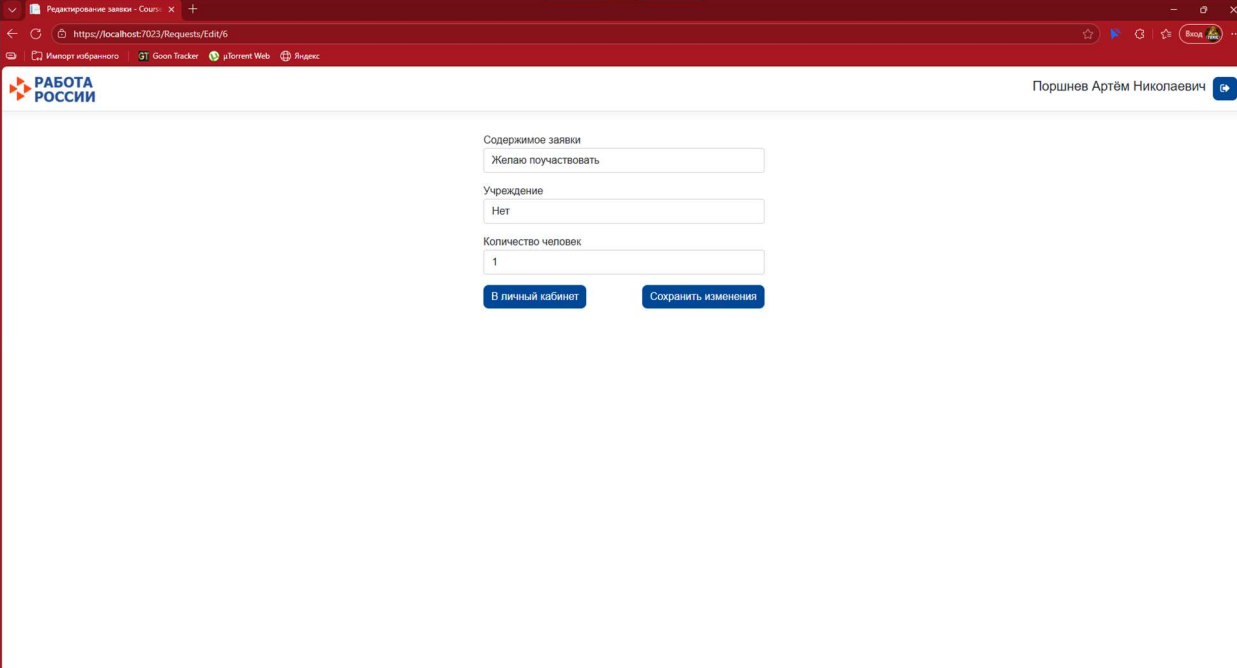


Рисунок 17 – Работа России. Вид страницы личного кабинета пользователя

При нажатии на кнопку «Редактировать» откроется страница редактирования заявки (рисунок 18)



Редктирование заявки - События

https://localhost:7023/Requests/Edit/6

История избранного | Goon Tracker | uTorrent Web | Яндекс

РАБОТА РОССИИ

Поршнев Артём Николаевич

Содержимое заявки

Желаю поучаствовать

Учреждение

Нет

Количество человек

1

В личный кабинет

Сохранить изменения

Рисунок 18 – Работа России. Вид страницы редактирования заявки

ЗАКЛЮЧЕНИЕ

Целью курсового проектирования являлась разработка веб-приложения подачи заявок на мероприятия службы занятости, предназначенного для просмотра списка текущих мероприятий и упрощения процесса подачи заявок.

Разработанный веб-сайт для службы занятости полностью соответствует актуальным требованиям современного цифрового взаимодействия между государственными учреждениями и гражданами. В ходе выполнения проекта удалось создать эффективный инструмент, который не только ускоряет обработку заявок на мероприятия, но и оптимизирует коммуникацию между сотрудниками кадрового центра и соискателями, обеспечивая последних всей необходимой информацией в удобном и структурированном виде.

Цель курсового проектирования достигнута, в процессе её достижения решены следующие задачи:

- проанализирована предметная область;
- определены требования к разрабатываемому программному средству;
- проектирована структура БД для хранения информации необходимой для функционирования веб-приложения;
- обеспечено удобство взаимодействия пользователя с приложением за счёт оптимизации навигации и логики интерфейса;
- реализована серверная часть, обеспечивающая выполнение бизнес-логики приложения и разграничение прав доступа;
- проведено тестирование разработанного программного продукта

В ходе работы был создан функциональный и удобный в использовании программный продукт, отвечающий актуальным требованиям к веб-сайтам и обеспечивающий пользователю удобство и простоту подачи заявок на мероприятия службы занятости.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бек, К. Экстремальное программирование: разработка через тестирование. – Санкт-Петербург : Питер, 2021. – 224 с. – URL: <https://ibooks.ru/bookshelf/376974/reading> (дата обращения: 27.11.2025). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.

2. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. – Москва : ФОРУМ : ИНФРА-М, 2025. – 400 с. – URL: <https://znanium.ru/catalog/product/2178802> (дата обращения: 10.11.2025). – Режим доступа: по подписке. – Текст : электронный.

3. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. – Москва : ФОРУМ : ИНФРА-М, 2024. – 368 с. – URL: <https://znanium.ru/catalog/product/2096940> (дата обращения: 3.11.2025). – Режим доступа: по подписке. – Текст : электронный.

4. Тидвелл, Д. Разработка интерфейсов. Паттерны проектирования. 3-е изд. – Санкт-Петербург : Питер, 2022. – 560 с. – URL: <https://ibooks.ru/bookshelf/386796/reading> (дата обращения: 10.11.2025). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

5. Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие. — Москва : КУРС : ИНФРА-М, 2024. — 336 с. — URL: <https://znanium.ru/catalog/product/2083407> (дата обращения: 15.11.2025). – Режим доступа: по подписке. – Текст : электронный.