Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №8**

Дисциплина: «Разработка профессиональных приложений»

Вариант №8

Выполнил:

студент группы ИВТАСбд-21

Зюзин Г.А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту.**

Необходимо модифицировать лабораторную работу №7:

1. Добавить средства редактирования для всех сущностей.

2. Добавить минимум две роли, соответствующих реальным ролям в выбранной предметной области. Добавить пользователей, обладающих этими ролями.

3. Реализовать средства регистрации пользователей, аутентификации.

4. Добавить средства авторизации и разграничения прав доступа разным ролям к разным сущностям.

**Описание реализации.**

Возможность редактирования всех сущностей была реализована ещё в предыдущей работе (см. отчёт по лабораторной работе 7).

При выполнении этой лабораторной работы были созданы две роли: Пациент и Врач.

|  |
| --- |
| from django.contrib.auth.models import Group  group\_name = 'Пациенты' if not Group.objects.filter(name=group\_name).exists():  group\_buyers = Group.objects.create(name=group\_name)   group\_name = 'Врачи' if not Group.objects.filter(name=group\_name).exists():  group\_managers = Group.objects.create(name=group\_name) |

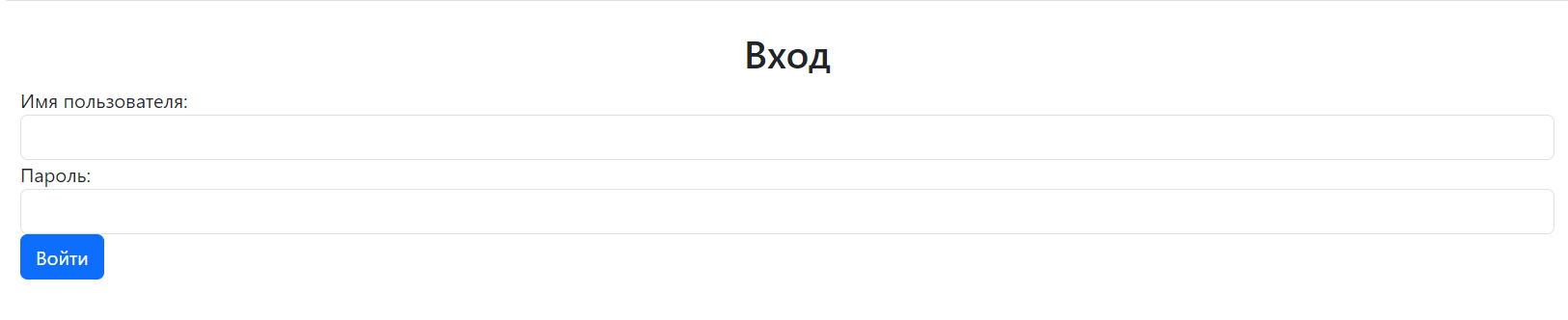
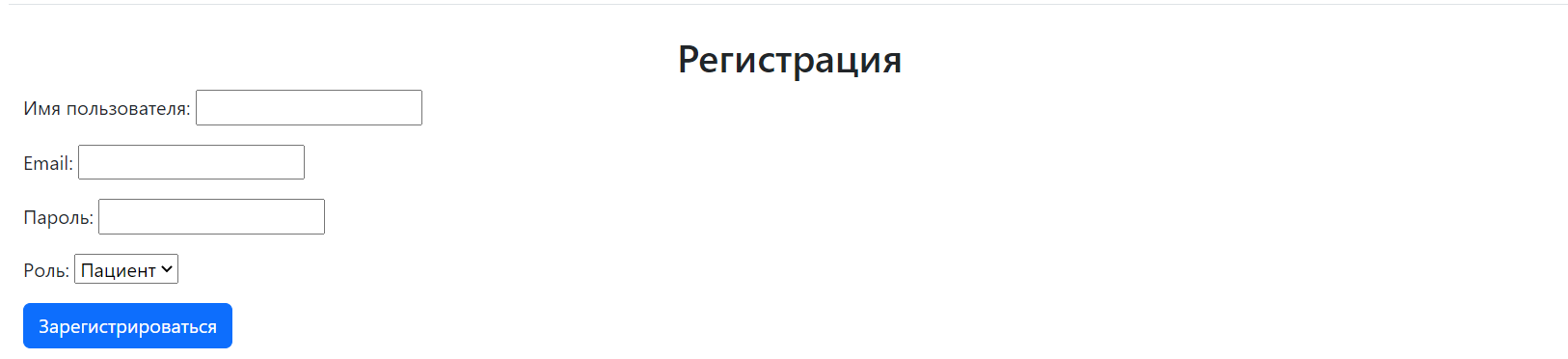
****Авторизация на сайте:

Рис. 1. Вход на сайт.

Рис.2. Регистрация.

|  |
| --- |
| def exp(request):  return render(request, 'exp.html')  def logout\_view(request):  logout(request)  return redirect('login')  def register(request):  if request.method == 'POST':  form = RegistrationForm(request.POST)  if form.is\_valid():  username = form.cleaned\_data['username']  email = form.cleaned\_data['email']  password = form.cleaned\_data['password']  role = form.cleaned\_data['role']  # Создание пользователя  user = User.objects.create\_user(username=username, email=email, password=password)  # Назначение пользователя в соответствующую группу  if role == 'patient':  group = Group.objects.get(name='Пациенты')  else:  group = Group.objects.get(name='Врачи')  user.groups.add(group)  # Вход пользователя после успешной регистрации  user\_login(request, user)  return redirect('visit\_list')  else:  form = RegistrationForm()  return render(request, 'register.html', {'form': form})  def login(request):  if request.method == 'POST':  username = request.POST['username']  password = request.POST['password']  user = authenticate(request, username=username, password=password)  if user is not None:  user\_login(request, user)  return redirect('visit\_list')  else:  return render(request, 'login.html', {'error\_message': 'Неправильное имя пользователя или пароль.'})  return render(request, 'login.html') |

1. exp(request): Это представление просто отображает шаблон 'exp.html', используется в качестве сообщения пользователю о том, что у него нет доступа (Рис.3).
2. logout\_view(request): Это представление разлогинивает пользователя с помощью вызова logout(request) и затем перенаправляет его на страницу 'login'. Предполагается, что у вас есть шаблон URL с именем 'login', определенный в конфигурации URL вашего проекта Django.
3. register(request): В данном представлении происходит обработка регистрации пользователя. Если запрос методом POST, то создается экземпляр формы RegistrationForm, и если форма проходит валидацию, то происходит создание пользователя с указанными данными (имя пользователя, электронная почта, пароль) и назначение его в соответствующую группу в зависимости от выбранной роли (пациент или врач). Затем пользователь выполняет вход в систему и перенаправляется на страницу 'visit\_list', которую необходимо определить в вашей конфигурации URL.
4. login(request): Это представление обрабатывает вход пользователя в систему. Если запрос методом POST, то получаются значения имени пользователя и пароля из запроса, и происходит аутентификация пользователя. Если пользователь существует и введены правильные учетные данные, то он выполняет вход в систему и перенаправляется на страницу 'visit\_list'. В противном случае отображается шаблон 'login.html' с сообщением об ошибке.

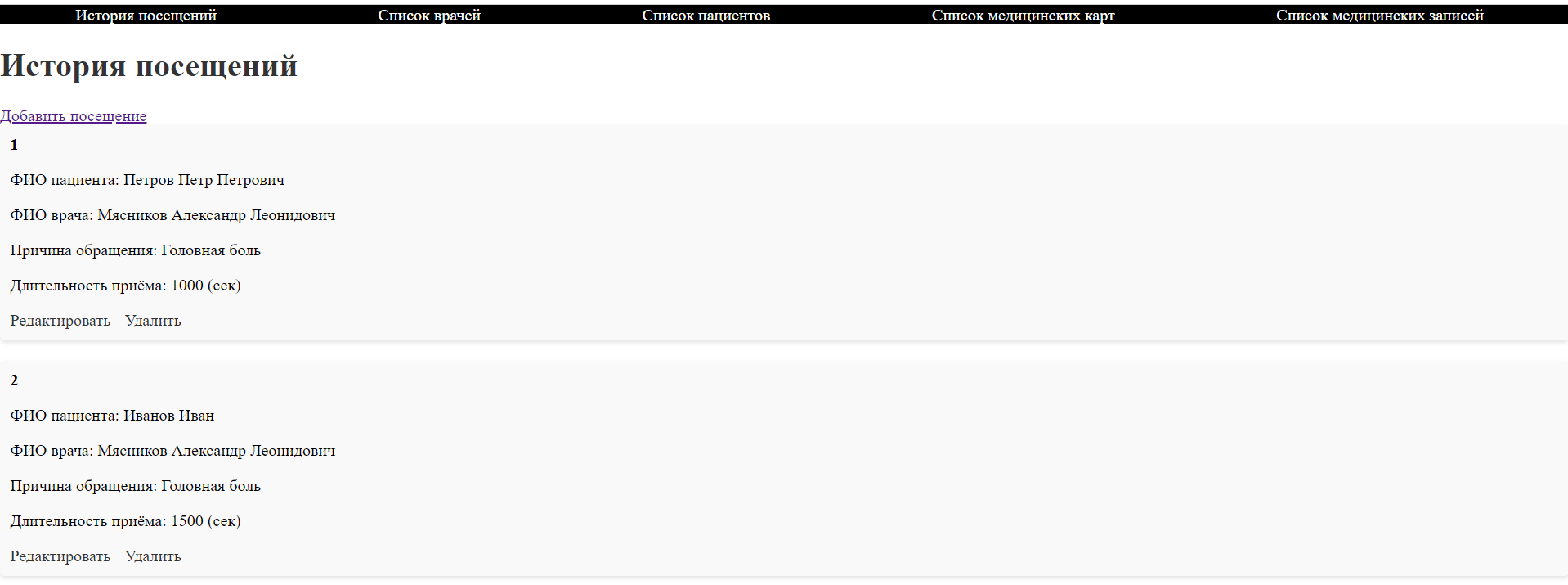
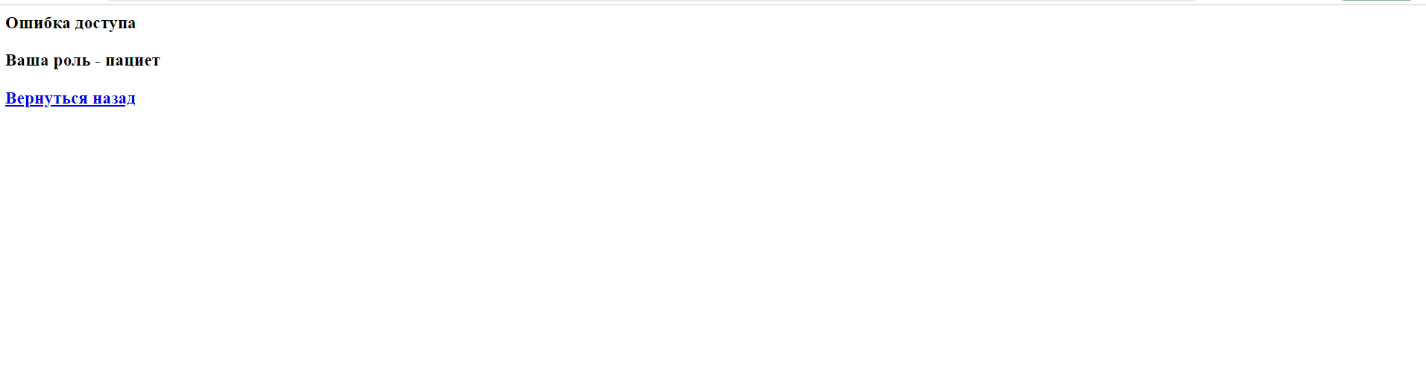
При входе на сайте отображается список посещений не в зависимости от роли:

Рис. 3. Отображение история посещений.

Пациентам нет доступа к остальным таблицам, в отличие от врачей, у которых есть доступ ко всем таблицам:  
Рис.4. Сообщение об отсутствии доступа.

**Описание возникших затруднений.**

Трудности при выполнении работы возникли, в основном, при реализации компонентов входа/регистрации пользователей и разграничении прав.

**Описание альтернативных способов решения.**  
Альтернативные способы решения задачи не рассматривались.

**Вывод.**

В модификации лабораторной работы №7 были внесены следующие изменения. Созданы роли "Врачи" и "Пациенты". Были созданы представления для регистрации и входа пользователей. Пользователи могут заполнить форму регистрации, и при успешной регистрации им будет предоставлен доступ к системе.