

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и управления”



Дисциплина “Парадигмы и конструкции языков программирования”

Отчет по домашнему заданию
“Сервер на Flask”

Выполнил:

Студент группы ИУБ-34Б
Савинский А.Ю.

Преподаватель:
Нардид А.Н.

Москва 2025

Задание

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвященциальному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

Описание и тексты программы

Разворачивал сервер я на публичном хостинге [Render](#), при этом сам код лежит на GitHub и подгружается оттуда с зависимостями из requirements.txt.

Сама ссылка на веб приложение — <https://savinskii-pcpl-labs.onrender.com> (при долгой неактивности подгружается дольше)

Серверная часть веб-приложения

app.py

```
# Импорт основных компонентов Flask:  
# Flask – создание веб-приложения  
# render_template – генерация HTML на основе шаблонов  
# request – доступ к данным HTTP-запросов  
# redirect, url_for – перенаправление между страницами  
# session – хранение данных пользователя между запросами  
from flask import Flask, render_template, request, redirect, url_for, session  
  
# Импорт ORM SQLAlchemy для работы с базой данных  
from flask_sqlalchemy import SQLAlchemy  
  
# Импорт для работы с датой и временем  
from datetime import datetime  
  
# Импорт SQL-функций для подсчёта статистики  
from sqlalchemy import func  
  
# Импорт модуля для работы с путями файловой системы  
import os  
  
# Создание экземпляра веб-приложения Flask  
app = Flask(__name__)  
  
# Секретный ключ приложения.  
# Используется Flask для защиты данных сессии пользователя  
app.config["SECRET_KEY"] = "dev-secret-key"  
  
# Определение базовой директории проекта
```

```

BASE_DIR = os.path.abspath(os.path.dirname(__file__))

# Настройка подключения к базе данных SQLite
# База данных хранится в файле chat.db в директории проекта
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///{} + os.path.join(BASE_DIR,
"chat.db")

# Отключение отслеживания изменений объектов для снижения нагрузки
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False

# Инициализация объекта базы данных SQLAlchemy
db = SQLAlchemy(app)

# Модель данных сообщения
class Message(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    author = db.Column(db.String(50), nullable=False, default="Аноним")
    text = db.Column(db.Text, nullable=False)
    time = db.Column(db.String(5), nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)

    # Строковое представление объекта для удобства отладки
    def __repr__(self):
        return f"<Message {self.id} {self.author}>"

# Создание таблиц базы данных, если они ещё не существуют
with app.app_context():
    db.create_all()

MAX_MESSAGES = 100

# Регистрация пользователя
@app.route("/register", methods=["GET", "POST"])
def register():
    # Обработка отправки формы регистрации
    if request.method == "POST":
        # Получение имени пользователя из формы

```

```

username = request.form.get("username", "").strip()

# Проверка на пустое значение
if not username:
    error = "Введите ник."
    return render_template(
        "register.html",
        error=error,
        username=""
    )

# Сохранение имени пользователя в сессии
session["username"] = username

# Перенаправление пользователя на страницу чата
return redirect(url_for("chat"))

# Отображение формы регистрации (GET-запрос)
return render_template(
    "register.html",
    error=None,
    username=session.get("username", ""))

```

Основной чат

```

@app.route("/", methods=["GET", "POST"])
def chat():

    # Получение имени пользователя из сессии
    username = session.get("username")

    # Если пользователь не зарегистрирован – перенаправление на регистрацию
    if not username:
        return redirect(url_for("register"))

    # Обработка отправки сообщения
    if request.method == "POST":
        text = request.form.get("text", "").strip()

        if text:

```

```
msg = Message(
    author=username,
    text=text,
    time=datetime.now().strftime("%H:%M")
)
db.session.add(msg)
db.session.commit()

# Ограничение количества сообщений в базе данных
total = Message.query.count()
if total > MAX_MESSAGES:
    to_delete = (
        Message.query
        .order_by(Message.created_at)
        .limit(total - MAX_MESSAGES)
        .all()
    )
    for m in to_delete:
        db.session.delete(m)
    db.session.commit()

# Перенаправление для предотвращения повторной отправки формы
return redirect(url_for("chat"))

# Получение последних сообщений для отображения
messages = (
    Message.query
    .order_by(Message.created_at.desc())
    .limit(MAX_MESSAGES)
    .all()
)
messages = list(reversed(messages))

# Отображение страницы чата
return render_template(
    "chat.html",
    messages=messages,
    username=username
)
```

```

# Статистика сообщений
@app.route("/stats")
def stats():
    # Подсчёт количества сообщений каждого пользователя
    stats = (
        db.session.query(
            Message.author,
            func.count(Message.id).label("count")
        )
        .group_by(Message.author)
        .order_by(func.count(Message.id).desc())
        .all()
    )

    total_messages = db.session.query(func.count(Message.id)).scalar() or 0

    current_user = session.get("username")

    # Отображение страницы статистики
    return render_template(
        "stats.html",
        stats=stats,
        total_messages=total_messages,
        current_user=current_user
    )

# Очистка чата
@app.route("/clear")
def clear():
    # Удаление всех сообщений из базы данных
    Message.query.delete()
    db.session.commit()

    # Перенаправление обратно в чат
    return redirect(url_for("chat"))

# Запуск приложения
if __name__ == "__main__":

```

```
# Запуск сервера в режиме отладки (для локальной разработки), далее будем деплоить
# через Render
app.run(debug=True)
```

Шаблоны, расположенные в каталоге `templates`, отвечают за формирование пользовательского интерфейса веб-приложения.

Они используются Flask совместно с шаблонизатором `Jinja2` для генерации HTML-страниц на основе данных, полученных от сервера.

base.html

```
<!doctype html>
<!-- Базовый HTML-шаблон веб-приложения -->
<html lang="ru">
<head>
    <!-- Установка кодировки документа -->
    <meta charset="utf-8">

    <!-- Заголовок страницы.
        Может быть переопределён в дочерних шаблонах -->
    <title>{% block title %}Мини-чат{% endblock %}</title>

    <!-- Встроенные стили оформления интерфейса -->
    <style>
        /* Общие стили страницы */
        body {
            font-family: system-ui, Arial, sans-serif;
            max-width: 800px;          /* Ограничение ширины страницы */
            margin: 0 auto;           /* Центрирование страницы */
            padding: 20px;
            background: #f4f4f4;       /* Цвет фона страницы */
        }

        /* Основной контейнер чата */
        .chat-container {
            background: #ffffff;
            padding: 0;
            border-radius: 10px;      /* Скругление углов */
            box-shadow: 0 2px 6px rgba(0,0,0,0.1); /* Тень */
        }
    
```

```
/* Внутренний контейнер, управляющий расположением элементов */
.chat-wrapper {
    display: flex;
    flex-direction: column;      /* Вертикальное расположение элементов */
    height: 90vh;                /* Высота окна чата */
    padding: 15px;
}

/* Верхняя панель с заголовком и служебной информацией */
.top-bar {
    margin-bottom: 10px;
    display: flex;
    justify-content: space-between;
}

/* Область отображения сообщений чата */
.messages-box {
    flex-grow: 1;                  /* Заполняет всё доступное пространство */
    overflow-y: auto;              /* Вертикальная прокрутка */
    padding-right: 5px;
    border: 1px solid #ddd;
    border-radius: 10px;
    background: #fafafa;
    padding: 10px;
}

/* Блок отдельного сообщения */
.message {
    padding: 4px 8px;
    margin-bottom: 4px;
    border-radius: 4px;
    background: #f0f0ff;
}

/* Метаданные сообщения (автор и время) */
.message-meta {
    font-size: 11px;
    color: #666;
    margin-bottom: 1px;
```

```
}

/* Форма отправки сообщений */
.send-form {
    margin-top: 10px;
    display: flex;
    flex-direction: column;
    gap: 8px;
}

/* Поля ввода текста */
input[type="text"], textarea {
    width: 98%;
    padding: 8px;
    border-radius: 6px;
    border: 1px solid #ccc;
    font-family: inherit;
}

/* Общие стили кнопок */
button {
    padding: 10px;
    border-radius: 6px;
    border: none;
    cursor: pointer;
}

/* Основная кнопка действия */
.btn-primary {
    background: #4f46e5;
    color: white;
    font-size: 16px;
}

</style>

<!-- Блок для подключения дополнительных стилей и скриптов
в дочерних шаблонах --&gt;
{% block head %}{% endblock %}

&lt;/head&gt;
&lt;body&gt;</pre>
```

```

<!-- Главный контейнер страницы -->
<div class="chat-container">

    <!-- Блок основного содержимого страницы.
        Заполняется дочерними шаблонами -->
    {% block content %}{% endblock %}

</div>
</body>
</html>

```

chat.html

```

{% extends "base.html" %}

<!-- Наследование базового шаблона.
    Все общие элементы (каркас страницы, стили) берутся из base.html -->

{% block title %}Мини-чат{% endblock %}

<!-- Заголовок страницы, подставляется в <title> базового шаблона -->

{% block content %}
<!-- Основной блок содержимого страницы чата -->

<div class="chat-wrapper">

    <!-- Верхняя панель с названием чата и служебной информацией -->
    <div class="top-bar">
        <h1>Мини-чат</h1>

        <!-- Информация о текущем пользователе и навигационные ссылки -->
        <div class="small-text">
            Вы вошли как <strong>{{ username }}</strong><br>

            <!-- Ссылки на дополнительные функции -->
            <a href="{{ url_for('stats') }}>Статистика</a> ·
            <a href="{{ url_for('register') }}>Сменить ник</a> ·
            <a href="{{ url_for('clear') }}>Очистить чат</a><br>

            <!-- Отображение количества сообщений, переданных сервером -->

```

```

    Сообщений: {{ messages|length }}

</div>
</div>

<!-- Область отображения сообщений чата --&gt;
&lt;div id="messages-box" class="messages-box"&gt;

    <!-- Проверка наличия сообщений --&gt;
    {% if messages %}

        <!-- Цикл вывода всех сообщений --&gt;
        {% for msg in messages %}

            &lt;div class="message"&gt;
                <!-- Метаданные сообщения: автор и время --&gt;
                &lt;div class="message-meta"&gt;
                    &lt;strong&gt;{{ msg.author }}&lt;/strong&gt; в {{ msg.time }}
                &lt;/div&gt;

                <!-- Текст сообщения --&gt;
                &lt;p class="message-text"&gt;{{ msg.text }}&lt;/p&gt;
            &lt;/div&gt;

        {% endfor %}

    {% else %}

        <!-- Сообщение отображается, если чат пуст --&gt;
        &lt;p&gt;Пока нет сообщений. Напишите что-нибудь первым!&lt;/p&gt;
    {% endif %}

&lt;/div&gt;

<!-- Форма отправки нового сообщения --&gt;
&lt;form method="post" class="send-form" id="chat-form"&gt;

    <!-- Поле ввода текста сообщения --&gt;
    &lt;textarea name="text"
              rows="2"
              placeholder="Ваше сообщение"
              required&gt;&lt;/textarea&gt;

    <!-- Кнопка отправки --&gt;
    &lt;button type="submit" class="btn-primary"&gt;Отправить&lt;/button&gt;
&lt;/form&gt;
</pre>

```

```

</div>

<!-- Клиентский JavaScript-код -->
<script>

    document.addEventListener("DOMContentLoaded", function () {

        // Получение элементов формы и области сообщений
        const form = document.getElementById("chat-form");
        const textarea = form.querySelector('textarea[name="text"]');
        const box = document.getElementById("messages-box");

        // Автоматическая прокрутка чата к последнему сообщению
        if (box) {
            box.scrollTop = box.scrollHeight;
        }

        // Отправка сообщения по нажатию Enter
        // Shift + Enter используется для переноса строки
        if (textarea && form) {
            textarea.addEventListener("keydown", function (event) {
                if (event.key === "Enter" && !event.shiftKey) {
                    event.preventDefault();
                    form.submit();
                }
            });
        }

        // Периодическое обновление страницы для получения новых сообщений
        // Обновление не выполняется, если пользователь в данный момент вводит текст
        setInterval(function () {
            if (document.activeElement === textarea &&
                textarea &&
                textarea.value.trim() !== "") {
                return;
            }
            location.reload();
        }, 3000); // интервал обновления – 3 секунды
    });
</script>

```

```
{% endblock %}  
<!-- Конец блока content -->
```

register.html

```
{% extends "base.html" %}  
<!-- Наследование базового шаблона.  
Общий каркас страницы и стили берутся из base.html -->  
  
{% block title %}Регистрация{% endblock %}  
<!-- Заголовок страницы, подставляемый в <title> базового шаблона -->  
  
{% block content %}  
<!-- Основной блок содержимого страницы регистрации -->  
  
<div class="chat-wrapper">  
  
<!-- Заголовок страницы -->  
<h1>Выбор никаОбратите внимание, что ник не может быть пустым и состоять из пробелов.</h1>  
  
<!-- Описание назначения страницы -->  
<p class="small-text">  
Выберите ник, под которым будете писать в чат.  
</p>  
  
<!-- Форма выбора имени пользователя -->  
<form method="post" class="send-form">  
  
<!-- Поле ввода имени пользователя -->  
<input type="text"  
name="username"  
placeholder="Ваш ник"  
value="{{ username or '' }}"  
required>  
  
<!-- Вывод сообщения об ошибке при некорректном вводе -->  
{% if error %}  
<p style="color: red; font-size: 14px;">{{ error }}</p>
```

```

    {% endif %}

    <!-- Кнопка подтверждения регистрации -->
    <button type="submit" class="btn-primary">Писать в чате</button>
</form>

<!-- Отображение текущего никна пользователя, если он уже сохранён в сессии -->
{% if username %}
<p class="small-text">
    Текущий ник: <strong>{{ username }}</strong>
</p>
{% endif %}

</div>
{% endblock %}
<!-- Конец блока content -->

```

stats.html

```

{% extends "base.html" %}
<!-- Наследование базового шаблона.

Общая структура страницы и стили берутся из base.html -->

{% block title %}Статистика сообщений{% endblock %}
<!-- Заголовок страницы, подставляется в <title> базового шаблона -->

{% block content %}
<!-- Основной блок содержимого страницы статистики -->

<div class="chat-wrapper">

<!-- Верхняя панель с заголовком страницы и служебной информацией -->
<div class="top-bar">
    <h1>Статистика</h1>

    <!-- Навигация и общая информация -->
    <div class="small-text">
        <!-- Ссылка для возврата в основной чат -->
        <a href="{{ url_for('chat') }}">← Вернуться в чат</a><br>

        <!-- Отображение общего количества сообщений в базе данных -->

```

```

    Всего сообщений в базе: {{ total_messages }}

</div>
</div>

<!-- Отображение имени текущего пользователя (если он определён) --&gt;
{%
  if current_user %}

  &lt;p&gt;&lt;strong&gt;Вы:&lt;/strong&gt; {{ current_user }}&lt;/p&gt;
{%
  endif %}

<!-- Проверка наличия статистических данных --&gt;
{%
  if stats %}

    <!-- Таблица статистики сообщений --&gt;
    &lt;table style="width:100%; border-collapse: collapse; font-size: 14px;"&gt;

      <!-- Заголовок таблицы --&gt;
      &lt;thead&gt;
        &lt;tr&gt;
          &lt;th style="text-align:left; border-bottom:1px solid #ddd;
padding:4px;"&gt;
            Ник
          &lt;/th&gt;
          &lt;th style="text-align:right; border-bottom:1px solid #ddd;
padding:4px;"&gt;
            Сообщений
          &lt;/th&gt;
        &lt;/tr&gt;
      &lt;/thead&gt;

      <!-- Тело таблицы --&gt;
      &lt;tbody&gt;

        <!-- Цикл перебора статистических данных --&gt;
        {%
          for author, count in stats %}

          <!-- Стока таблицы.
              Текущий пользователь выделяется цветом --&gt;
          &lt;tr {%
            if author == current_user %}style="background:#eef;"{%
            endif %}&gt;

            &lt;td style="padding:4px;"&gt;
</pre>

```

```
    {{ author }}
```

```
</td>
```

```
<td style="padding:4px; text-align:right;">
```

```
    {{ count }}
```

```
</td>
```

```
</tr>
```

```
{% endfor %}
```

```
</tbody>
```

```
</table>
```

```
{% else %}
```

```
<!-- Сообщение отображается, если в базе данных нет сообщений -->
```

```
<p>Пока нет сообщений, статистику показать нельзя.</p>
```

```
{% endif %}
```

```
</div>
```

```
{% endblock %}
```

```
<!-- Конец блока content -->
```

Экранные формы с примерами выполнения программы

Скриншот экрана с формой для выбора никнейма. Веб-страница загружена на savinskii-pcpl-labs.onrender.com/register. Форма имеет заголовок **Выбор никна**. Внутри формы есть инструкция: **Выберите ник, под которым будете писать в чат.** Ниже находится поле для ввода никнейма сplaceholderом **Ваш ник**, и кнопка **Писать в чате**.

Скриншот экрана с интерфейсом чата. Веб-страница загружена на savinskii-pcpl-labs.onrender.com. Заголовок страницы **Мини-чат**. В правом верхнем углу отображается сообщение: **Вы вошли как андрей**, [Статистика](#), [Сменить ник](#), [Очистить чат](#). Статус сообщений: **Сообщений: 6**. В центре страницы расположено сообщение с перепиской между пользователями **андрей** и **никита**:

андрей в 11:31
всем привет

андрей в 11:31
как дела

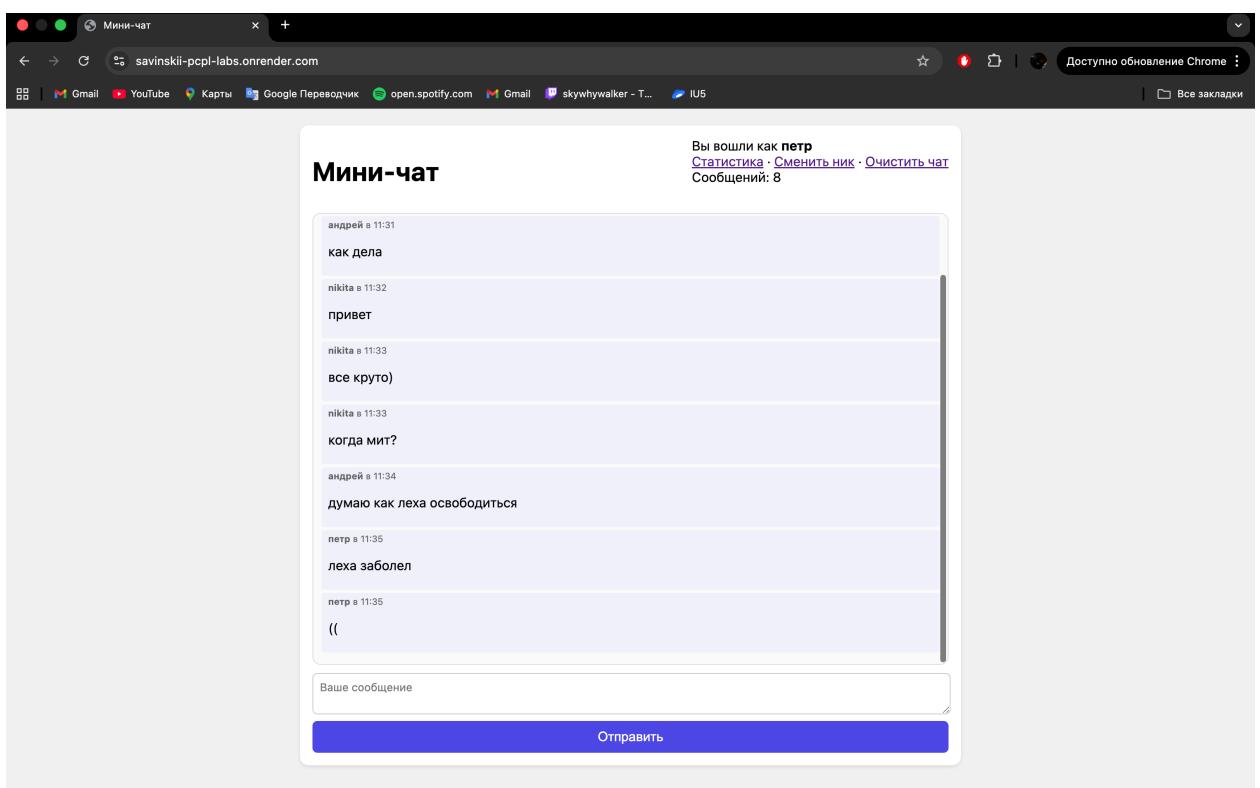
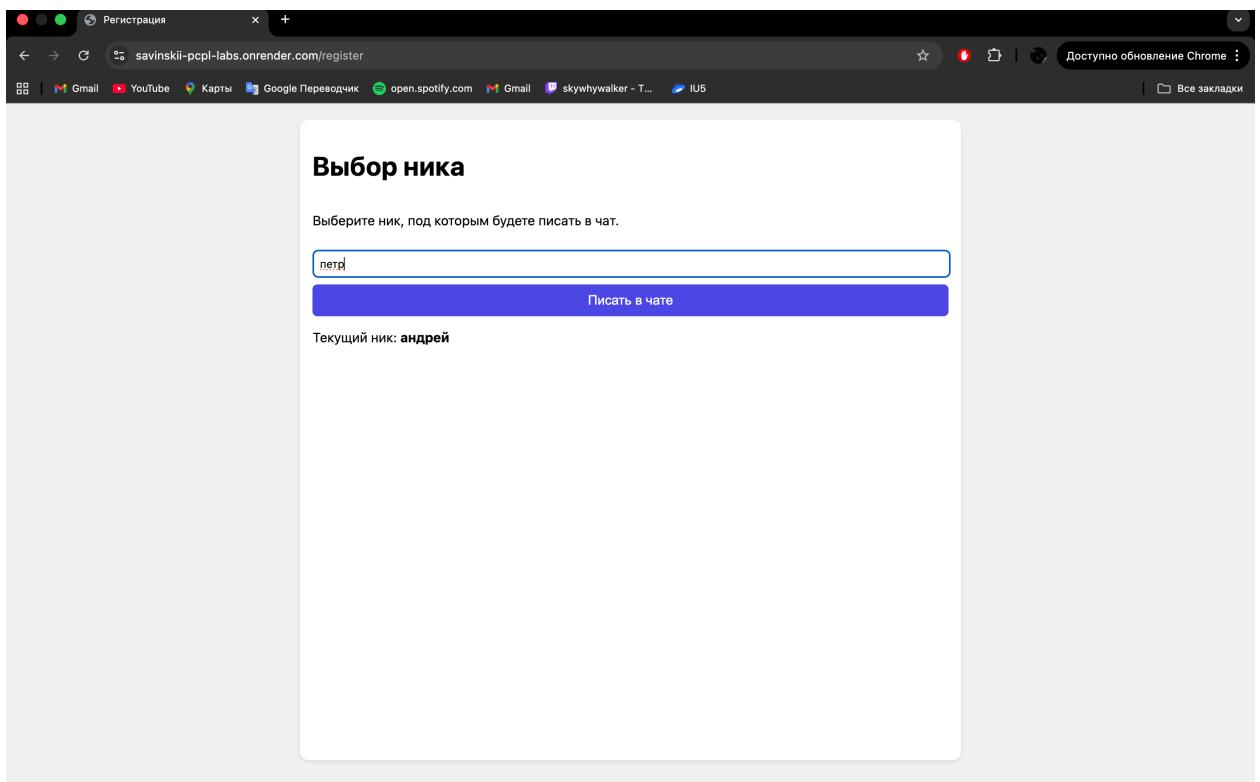
никита в 11:32
привет

никита в 11:33
все круто)

никита в 11:33
когда мит?

андрей в 11:34
думаю как леха освободиться

Внизу страницы есть поле для ввода сообщения сplaceholderом **Ваше сообщение** и кнопка **Отправить**.



← Вернуться в чат
Всего сообщений в базе: 8

Статистика

Всё: petr

Ник	Сообщений
андрей	3
nikita	3
petr	2